A1500/A2500
Firmware Version 4.xx
Protocol Definition

Revision 1.8

alpha Zähler

Documentnumber:
Date of version: 08.10.2007 10:45
Product Manager: DEE / M Dr. Gerhard Eisenbeiß
Project Manager:
Author: DEE / E Frank Dewald
File: RCS//Doku_A1500/A1500FW4xx_Communication.doc

Release:

NOT RELEASED			
	Version	day/month/year	signature
Test			
Release			

COPYRIGHT

Printed in Germany

NOTICE

The information contained in this document is subject to change without notice.

ELSTER shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

ELSTER expressly disclaims all responsibility and liability for the installation, use, performance, maintenance and support of third party products. Customers are advised to make their own independent evaluation of such products.

Contents

1 Overview

1.1 System Overview

The communication protocol of the *alpha* meter A1500 and A2500 is realised according the EN 62056-21 standard. This document describes only the meter specific features.

The communication takes place between the meter and a hand-held-unit (HHU). The communication starts with a request of the HHU or pushing a button. Therefore the meter waits passive for a request.

1.2 Document Conventions

The following firmware versions for the A1500 and A2500 meter are supported in this document:

- Firmware Version 4.00
- Firmware Version 4.10
- Firmware Version 4.20
- Firmware Version 4.30
- Firmware Version 4.40
- Firmware Version 4.50

The expression FW 4.xx always signifies the latest supported Version.
In this case it's the Firmware Version 4.50!

2 Referenced Documents

Title	Version	Date
Electricity metering – data exchange for meter reading, tariff and load control – part 21: (former IEC1107)	EN 62056-21	05/2002
Electricity metering – data exchange for meter reading, tariff and load control – part 61: Object Identification System (OBIS)	EN 62056-61	06/2002
VDEW specification for electronic meters	2.0	12/1997
Product description “alpha meter A1500” A1500_PR_E.pdf	1.5	06/2006

3 General

3.1 Character format and Baud rates

3.1.1 Character format

Asynchronous serial bit (start-stop) transmission according to ISO 1177, half-duplex.
The character format ISO 1177: 1985 (1 start bit, 7 data bits, 1 stop bit)

3.1.2 Character security

With parity bit, even parity according to ISO 1177:1985

3.1.3 Data Transmission

According the 7 Bit sign format the contents must be transmitted in a hex dump format. The transmission of 1 Byte includes 2 Byte in a hex dump format, which must be transmitted.

3.1.4 Baud rates

The used baud rates are:

- 300, 600, 1200, 2400, 4800, 9600 Baud
- 19200 Baud (only electrical interfaces)

After the “Option Select Message” a switch of the initial baud rate (300 Baud) to the communication baud rate between the meter and the communication device (HHU) is possible.
This baud rate switch can be disabled and the meter can directly communicate with a higher baud rate. The reason for that feature is the use of simple telephone modems.

3.2 Communication modes

With the *alpha* meter A1500 Mode C and D, according the EN 62056-21, can be configurated.

The standard mode for the communication is Mode C. With Mode C and the “Option Select Message” the user can switch to the “Data Readout Mode” or the “Programming Mode”.

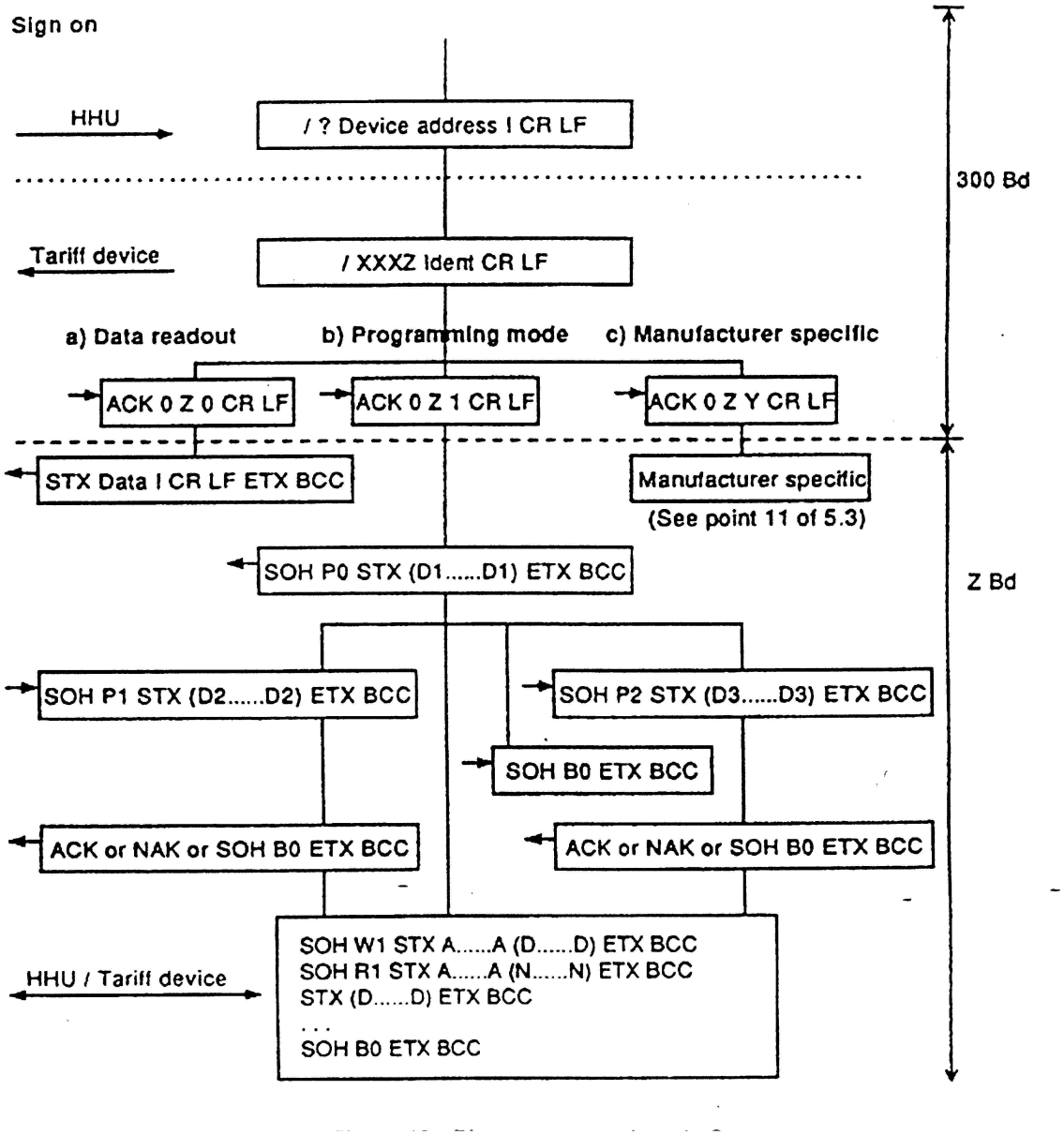

Fig. 1: meter reading using Mode C

If the alpha meter is configured according Mode D, he is transmitting his data after the alternate button is pressed during the scroll mode of the meter. The transmission baud rate is 2400 baud. After the end of the communication the meter goes automatically back to the scroll mode.
The Mode D is only useful for local meter reading with a HHU.

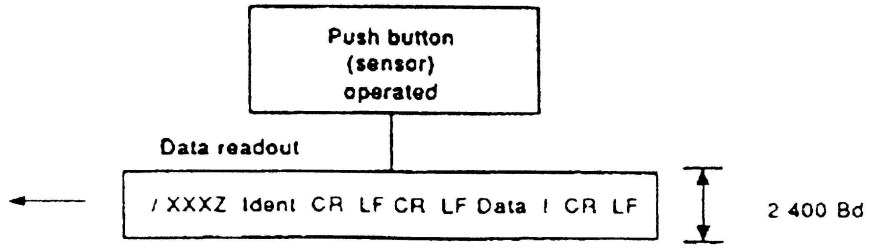

Fig. 2: meter reading using Mode D

3.3 Timeouts

The essential timeouts of the meter are realised according the EN 62056-21standard:

- The time between 2 succeeding signs should be lower than 1,5s
- The timeout in the standard mode between the select and request telegram is between 0,2..1,5s *)
- The timeout in the Programming Mode is approx. 60s.
- The timeout during the read and write of different meter classes, which is implemented in the manufacture specific mode is between 0,02..1,5s (60s).
- If the meter doesn't get the "Option Select Message" ("<ACK>0ZY<CR><LF>") after 1,5s, the alpha meter is sending his data in the "Data Readout Mode" with the initial baud rate.

*) By partial reading of load profile data, this time between the select and request telegram can be up to 6s.

3.4 Checksum

The checksum BCC is calculated according EN 62056-21 (ISO 1155).
Under the term of this standard the leading <SOH> or <STX> are not included in the calculation.

4 Message definitions

4.1 Mode C

4.1.1 Request Message

The Communication starts with the “Request Message” of the HHU to the meter:

HHU → Meter:
/?! or **/?nnnnnnnnnnnnnnn!**

with nn = device / communication address (up to 16 ASCII-characters, leading zeros and blanks are suppressed).

If the Request message includes a communication address, the meter will respond only if this address corresponds with his configured address.
The communication starts with the initial baud rate of the meter.

4.1.2 Identification Message

After the “Request message” the meter send back the “Identification Message” with his communication ID:

Meter → HHU:
/ABBz\@xxxxxxxxxxxxxx

with z = proposal of the communication baud rate

- z=0: 300Baud
- z=1: 600Baud
- z=2 1200Baud
- z=3: 2400Baud
- z=4: 4800Baud
- z=5: 9600Baud
- z=6: 19200Baud

and xxx ... = communication ID of the meter (14 ASCII-characters)

The characters „\@“ are used to identify that the communication is implemented according the “VDEW specification 2.0”, which means that the R5, W5, or R6 commands can be used.

4.1.3 Acknowledgement / Option Select Message

After the meter sends the “Identification Message” the HHU responds with:

HHU → Meter:
<ACK>0ZY<CR><LF>

with: Z = communication baud rate
(0 = 300BD, 1 = 600 Bd, 2 = 1200 Bd, 3 = 2400 Bd, 4 = 4800 Bd, 5 = 9600 Bd, 6=19200Bd)
Y = 0: switch to the “Data Readout Mode”
Y = 1: switch to the “Programming Mode”

If the HHU accepts the baud rate z, both communication devices (meter and HHU) switch to this communication baud rate. Otherwise the communication starts with the initial baud rate.

If the “Option Select Message” ("<ACK>0ZY<CR><LF>") is not received from the meter after 1,5s or if it is not the correct format, the meter sends his data in the “Data Readout Mode” with the initial baud rate.

4.1.4 Data Readout Mode: Data message

In the “Data Readout Mode” the data format is:

Meter → HHU:
<STX>Data!<CR><LF><ETX><BCC>

with "Data": 1 or more "Data Lines"
"Data Line": inclusive <CR><LF> max. of 80 characters with 1 or more "Data Sets"
"Data Set": [Address]([Value][*Unit]) (the information inside "[]" is optional).

4.1.5 Programming Mode: Acknowledgement Message

After a correct receipt of a “Data Message” in the Programming Mode, the meter sends back his “Acknowledge Message”

Meter → HHU:
<ACK>

4.1.6 Programming Mode: Repeat-request Message

After a not correct receipt of a “Data Message” in the Programming Mode, the meter sends back his “Repeat-request Message”

Meter → HHU:
<NAK>

Now the HHU should repeat the message. The number of repetitions is not fixed.

4.1.7 Programming Mode: Command Message

After the “Option Select Message”, which switches the meter to the “Programming Mode”, the meter responds with the “Programming Command Message”

Meter → HHU:
<SOH>P0<STX>(nnnnnnnn)<ETX><BCC>

with nn... = Serial number of the meter.

After the meter has sent his serial number, the HHU is asked for sending the password of the meter.

HHU → Meter:
<SOH>P1<STX>(pppppppp)<ETX><BCC>

with pp... = utility password of the meter (8 BCD characters)

or

HHU → Meter:
<SOH>P2<STX>(pppppppp)<ETX><BCC>

with pp... = manufacturer password of the meter (8 BCD characters)

An other possibility is, that the HHU sends directly the R5-, W5-, R6-commands to the meter

After the meter has received a correct command, he responds with a “Acknowledge Message”:

Meter → HHU:
<ACK>

In the next step the HHU is sending the next “Programming Command Message”, or (after a “Repeat-request Message”, <NAK>) the previous message will be repeated.

The communication ends with the ”Programming Command Message”

Meter → HHU:
<SOH>B0<ETX><BCC>

4.1.8 Programming Mode: Data Message

The answer of the meter to the “Programming Command Messages” send by the HHU is:

Meter → HHU:
<STX>Data<EOT>/<ETX><BCC>

with "Data": same as the data format in “Readout Mode”; 1 or several "Data Lines"

"Data Line": includes <CR><LF>, max. length 80 characters; 1 or several "Data Sets"

“Data Sets”: "[Address]([Value][*Unit])" ; definitions in "[]" are optional

Special “Data Messages” (Data Message for reading the meter classes):

HHU → Meter:
<STX>xxxx(0123..)<EOT>/<ETX><BCC>

If the required amount of data is higher than the available space in the “Data Message”, an <EOT> instead of <ETX> is transmitted. In that case the HHU will respond with a “Acknowledge Message” (see below). The next “Programming Command” can only be send after the end of the data transmission (<ETX>).

4.1.9 Programming Mode: Error Message

After special error conditions in the “Programming Command messages” the meter responds with an error message:

Meter → HHU:
<STX>(ERRORnn)<ETX><BCC>

with nn =
- 00: no valid command
- 01: unknown command
- 02: no access without manufacturer password
- 03: no access without hardware lock released
- 04: no valid class
- 05: no additional data available
- 06: command format not valid
- 07: function is not supported
- 08: demand reset not allowed
- 09: load profile initialisation not activated
- 10: ripple receiver not enabled
- 11: no valid time&date
- 12: no access to the desired storage
- 13: no access, because set mode was not activated by alternate button
- 14: no access without password
- 15: no access with closed terminal cover
- 16: no access due to configuration change denial

4.2 Mode D

4.2.1 Data message

In the Mode D the format of the “Data Message” is:

Meter → HHU:
/ABB3\@xxxxxxxxxxxxxx<CR><LF><CR><LF>Data!<CR><LF>

with xxx... = communication ID of the meter (14 ASCII characters, see “Identification Message”)

5 alpha meter commands

The alpha meter A1500 / A2500 supports following commands:

5.1 Data Readout Mode

5.1.1 Register Data Readout

Readout of configured register data. **(Supported in Firmware Versions 4.00 to 4.xx)**

Example: reading of the register data

Initial baud rate:	300 Baud
Communication baud rate:	4800 Baud
Identifier system of the meter:	OBIS

```
HHU:      /?!<CR><LF>
Meter:    /ABB4\@V4.40        <CR><LF>
HHU:      <ACK>040<CR><LF>
Meter:    <STX>F.F(00000000)<CR><LF>
          0.0.0(00000001)<CR><LF>
          0.9.1(14:45:59)<CR><LF>
          0.2.2(00-11-21)<CR><LF>
          1.8.1(000123.34)<CR><LF>
          1.8.2(000037.57)<CR><LF>
          ....
          2.8.2(000101.23)<CR><LF>
          !<CR><lf>
          <ETX><BCC><CR><LF>
```

5.1.2 Service Data Readout

Readout of configured service data. **(Supported in Firmware Versions 4.11 to 4.xx)**

Example: reading of the service data

Initial baud rate:	300 Baud
Communication baud rate:	4800 Baud
Identifier system of the meter:	OBIS

```
HHU:      /2!<CR><LF>
Meter:    /ABB4\@V4.40        <CR><LF>
HHU:      <ACK>040<CR><LF>
Meter:    <STX>32.7.0(228.3)<CR><LF>
          31.7.0(10.049)<CR><LF>
          ....
          34.7(49.99)<CR><LF>
          !<CR><lf>
          <ETX><BCC><CR><LF>
```

5.2 Programming Mode

5.2.1 Demand reset

After receiving the “Demand Reset” command the meter executes a demand reset by doing a snap shot of all energy and demand registers. 2 different commands for a demand reset are supported.

HHU → Meter: **(Supported in Firmware Versions 4.00 to 4.xx)**
<SOH>E2<STX>0001()<ETX><BCC>

HHU → Meter: **(Supported in Firmware Versions 4.00 to 4.xx)**
<SOH>W1<STX>S01()<ETX><BCC> **(standard command)**

Example: demand reset by using the W1-command

```
HHU:        /?!<CR><LF>
Meter:      /ABB4\@V4.40          <CR><LF>
HHU:        <ACK>041<CR><LF>
Meter:      <SOH>P0<STX>(00000231)<ETX><BCC>
HHU:        <SOH>P1<STX>(00000000)<ETX><BCC>
Meter:      <ACK>
HHU:        <SOH>W1<STX>S01()<ETX><BCC>
Meter:      <ACK>
HHU:        <SOH>B0<ETX><BCC>
```

5.2.2 Reset of power outage counter

With that command the following registers are set to zero:

- Counter for power outages
- Event registers (class 25)
- Power Fail
- Reverse Power

HHU → Meter: **(Supported in Firmware Versions 4.00 to 4.xx)**
<SOH>W1<STX>S02()<ETX><BCC>

Example: reset of power outage counters

```
HHU:        /?!<CR><LF>
Meter:      /ABB4\@V4.40          <CR><LF>
HHU:        <ACK>041<CR><LF>
Meter:      <SOH>P0<STX>(00000001)<ETX><BCC>
HHU:        <SOH>P1<STX>(00000000)<ETX><BCC>
Meter:      <ACK>
HHU:        <SOH>W1<STX>S02()<ETX><BCC>
Meter:      <ACK>
HHU:        <SOH>B0<ETX><BCC>
```

5.2.3 Reset of the power quality counters

With that command the following can be set to zero:

- Power quality counters (class 26)

HHU → Meter: **(Supported in Firmware Versions 4.30 to 4.xx)**

<SOH>W1<STX>S03()<ETX><BCC>

Example: reset of power quality counters

```
HHU:        /?!<CR><LF>
Meter:      /ABB4\@V4.40        <CR><LF>
HHU:        <ACK>041<CR><LF>
Meter:      <SOH>P0<STX>(00000001)<ETX><BCC>
HHU:        <SOH>P1<STX>(00000000)<ETX><BCC>
Meter:      <ACK>
HHU:        <SOH>W1<STX>S03()<ETX><BCC>
Meter:      <ACK>
HHU:        <SOH>B0<ETX><BCC>
```

5.2.4 Enable ripple receiver communication

With that command the optical interface will be enabled to communicate to the internal ripple receiver.

HHU → meter: **(Supported in Firmware Versions 4.00 to 4.xx)**

<SOH>W1<STX>S05(data)<ETX><BCC>

with data = duration of the ripple receiver communication mode

- **(mm)**
  - o mm = minutes (0..FF)

Example: enable optical port for LCR communication (duration: 20min)

```
HHU:            /?!<cr><LF>
meter:          /ABB4\@.....<cr><LF>
HHU:            <ACK>041<cr><LF>
Meter:          <SOH>P0<STX>(00000001)<ETX><BCC>
HHU:            <SOH>P1<STX>(00000000)<ETX><BCC>
Meter:          <ACK>
HHU:            <SOH>W1<STX>S05(14)<ETX><BCC>
Meter:          <ACK>
HHU:            <SOH>B0<ETX><BCC>
```

5.2.5 Set “time&date”

By using the W1 and W2-commands we have four different possibilities to set time&date of the meter.

HHU → Meter: **(Supported in Firmware Versions 4.00 to 4.xx)**
<SOH>W1<STX>S04(data)<ETX><BCC>

HHU → Meter: **(Supported in Firmware Versions 4.00 to 4.xx)**
<SOH>W1<STX>S06(data)<ETX><BCC>

HHU → Meter: **(Supported in Firmware Versions 4.00 to 4.xx)**
<SOH>W2<STX>C001(data)<ETX><BCC>

HHU → Meter: **(Supported in Firmware Versions 4.00 to 4.xx)**
<SOH>W2<STX>C003(data)<ETX><BCC> **(standard command)**

with data = time and date data

- **(yymmddhhmmss)**
  - yy = year (00..99)
  - mm = month (1..12)
  - dd = day (1..31)
  - hh = hour (0..23)
  - mm = minute (0..59)
  - ss = second (0..59)

Remark:

- After decoding of the command the meters starts a new demand period if the time difference between the meter and the transferred time is outside a time window (standard: +/-9s)

Example: set time&date by using the standard W2-command

Time and Date: 21.10.2000,20:10:35

```
HHU:      /?!<CR><LF>
Meter:    /ABB4\@V4.40        <CR><LF>
HHU:      <ACK>041<CR><LF>
Meter:    <SOH>P0<STX>(00000231)<ETX><BCC>
HHU:      <SOH>P1<STX>(00000000)<ETX><BCC>
Meter:    <ACK>
HHU:      <SOH>W2<STX>C003(001021201035)<ETX><BCC>
Meter:    <ACK>
HHU:      <SOH>B0<ETX><BCC>
```

5.2.6 Reset error status

With that command the error status of the meter can be reset:

HHU → Meter: **(Supported in Firmware Versions 4.00 to 4.xx)**

<SOH>W1<STX>S07()<ETX><BCC>

Example: reset error status

```
HHU:        /?!<CR><LF>
Meter:      /ABB4\@V4.40         <CR><LF>
HHU:        <ACK>041<CR><LF>
Meter:      <SOH>P0<STX>(00000001)<ETX><BCC>
HHU:        <SOH>P1<STX>(00000000)<ETX><BCC>
Meter:      <ACK>
HHU:        <SOH>W1<STX>S07()<ETX><BCC>
Meter:      <ACK>
HHU:        <SOH>B0<ETX><BCC>
```

5.2.7 Reset load profile and log file

With that command the load profile and log file will be reset.

HHU → Meter: **(Supported in Firmware Versions 4.00 to 4.xx)**

<SOH>W1<STX>S08()<ETX><BCC>

Example: reset load profile and log file

```
HHU:        /?!<CR><LF>
Meter:      /ABB4\@V4.40         <CR><LF>
HHU:        <ACK>041<CR><LF>
Meter:      <SOH>P0<STX>(00000001)<ETX><BCC>
HHU:        <SOH>P1<STX>(00000000)<ETX><BCC>
Meter:      <ACK>
HHU:        <SOH>W1<STX>S08()<ETX><BCC>
Meter:      <ACK>
HHU:        <SOH>B0<ETX><BCC>
```

5.2.8 Set “time & date & railway number” of the meter

With that command the time&date of the meter will be set and the railway number ttttttt (8 ASCII characters) will be stored in the log file of the meter. This command can be used for meters in railway applications.

HHU → Meter: **(Supported in Firmware Versions 4.00 to 4.xx)**
<SOH>W1<STX>S09(data)<ETX><BCC>

with data = time and date data

- **(tttttttyymmddhhmmss)**
  - tttttttt = railway num. (8 ASCII characters)
  - yy = year (00..99)
  - mm = month (1..12)
  - dd = day (1..31)
  - hh = hour (0..23)
  - mm = minute (0..59)
  - ss = second (0..59)

Remark:

- after the receipt of the command always a new demand period will be started

Example: set time&date and railway number

Time and Date: 21.10.2000,20:10:35
Railway Number 12345678

```
HHU:          /?!<cr><LF>
meter:        /ABB4\@.....<cr><LF>
HHU:          <ACK>041<cr><LF>
Meter:        <SOH>P0<STX>(00004301)<ETX><BCC>
HHU:          <SOH>P1<STX>(00000000)<ETX><BCC>
Meter:        <ACK>
HHU:          <SOH>W1<STX>S09(12345678991021201035)<ETX><BCC>
Meter:        <ACK>
HHU:          <SOH>B0<ETX><BCC>
```

5.2.9 Set “register value S0-input no. 1” of the meter

With that command the register value of the S0-input no. 1 of the meter will be set (9 bytes ,bcd)

HHU → Meter: **(Supported in Firmware Versions 4.20 to 4.xx)**
<SOH>W1<STX>S0A(data)<ETX><BCC>

with data = register value

- **(xxxxxxxxxxxxxxxxxx)**
  - xx.. = register value (9 Bytes bcd)

Example: set register value s0-input no. 1

```
HHU:          /?!<CR><LF>
Meter:        /ABB5\@V4.40          <CR><LF>
HHU:          <ACK>051<CR><LF>
meter:        <SOH>P0<STX>(00325085)<ETX>i Soll: i
HHU:          <SOH>P2<STX>(97771903)<ETX>g
meter:        <ACK>
HHU:          <SOH>W1<STX>S0A(001234560000000000)<ETX>C
meter:        <ACK>
HHU:          <SOH>B0<ETX>q
```

5.2.10 Set “register value S0-input no. 2” of the meter

With that command the register value of the S0-input no. 2 of the meter will be set (9 bytes ,bcd)

HHU → Meter: **(Supported in Firmware Versions 4.20 to 4.xx)**

<SOH>W1<STX>S0B(data)<ETX><BCC>

with data = register value

- **(xxxxxxxxxxxxxxxxx)**
  - xx.. = register value (9 Bytes bcd)

Example: set register value s0-input no. 2

```
HHU:          /?!<CR><LF>
Meter:        /ABB5\@V4.40        <CR><LF>
HHU:          <ACK>051<CR><LF>
meter:        <SOH>P0<STX>(00325085)<ETX>i Soll: i
HHU:          <SOH>P2<STX>(97771903)<ETX>g
meter:        <ACK>
HHU:          <SOH>W1<STX>S0B(001234560000000000)<ETX>C
meter:        <ACK>
HHU:          <SOH>B0<ETX>q
```

5.2.11 Set “register value S0-input no. 3” of the meter

With that command the register value of the S0-input no. 3 of the meter will be set (9 bytes ,bcd)

HHU → Meter: **(Supported in Firmware Versions 4.20 to 4.xx)**

<SOH>W1<STX>S0C(data)<ETX><BCC>

with data = register value

- **(xxxxxxxxxxxxxxxxxx)**
  - xx.. = register value (9 Bytes bcd)

Example: set register value s0-input no. 3

```
HHU:          /?!<CR><LF>
Meter:        /ABB5\@V4.40        <CR><LF>
HHU:          <ACK>051<CR><LF>
meter:        <SOH>P0<STX>(00325085)<ETX>i Soll: i
HHU:          <SOH>P2<STX>(97771903)<ETX>g
meter:        <ACK>
HHU:          <SOH>W1<STX>S0C(001234560000000000)<ETX>C
meter:        <ACK>
HHU:          <SOH>B0<ETX>q
```

5.2.12 Set “register value S0-input no. 4” of the meter

With that command the register value of the S0-input no. 4 of the meter will be set (9 bytes ,bcd)

HHU → Meter: **(Supported in Firmware Versions 4.50 to 4.xx)**

<SOH>W1<STX>S0H(data)<ETX><BCC>

with data = register value

- **(xxxxxxxxxxxxxxxxxx)**
  - xx.. = register value (9 Bytes bcd)

Example: set register value s0-input no. 4

```
HHU:          /?!<CR><LF>
Meter:        /ABB5\@V4.50          <CR><LF>
HHU:          <ACK>051<CR><LF>
meter:        <SOH>P0<STX>(00325085)<ETX>i Soll: i
HHU:          <SOH>P2<STX>(97771903)<ETX>g
meter:        <ACK>
HHU:          <SOH>W1<STX>S0H(0012345600000000000)<ETX>C
meter:        <ACK>
HHU:          <SOH>B0<ETX>q
```

5.2.13 Reset terminal cover opening counter

Resets the counter of the terminal cover openings, including the corresponding timestamp and the error message.

HHU → Meter: **(Supported in Firmware Versions 4.20 to 4.xx)**

<SOH>W1<STX>S0D()<ETX><BCC>

Example: reset terminal cover open counter

```
HHU:        /?!<CR><LF>
Meter:      /ABB4\@V4.40          <CR><LF>
HHU:        <ACK>041<CR><LF>
Meter:      <SOH>P0<STX>(00000001)<ETX><BCC>
HHU:        <SOH>P1<STX>(00000000)<ETX><BCC>
Meter:      <ACK>
HHU:        <SOH>W1<STX>S0D()<ETX><BCC>
Meter:      <ACK>
HHU:        <SOH>B0<ETX><BCC>
```

5.2.14 Read impulse constant

This command reads out the information to change the impulse constants for LED and pulse-output.

HHU → Meter: **(Supported in Firmware Versions 4.20 to 4.40)**

<SOH>R1<STX>S0F()<ETX><BCC>

Meter → HHU:

<STX>(data)<ETX><BCC>

with data = impulse constant data

- **(xxxxxxxxxxxxllooooovvvvcccc)**
  - x = xke (secondary Impulse constant)(bcd)
  - l = LED – divider (bcd)
  - o = output – divider (bcd)
  - v = voltage transducer value **(vv vv ; bcd)**
  - c = current transducer value **(vv vv ; bcd)**

Example: read impulse constant

xke	= 00000025000
LED-divider	= 01
output – divider	= 0002
voltage transducer value	= 0001
current transducer value	= 0001

```
HHU:      /?!<CR><LF>
Meter:    /ABB4\@V4.40        <CR><LF>
HHU:      <ACK>041<CR><LF>
Meter:    <SOH>P0<STX>(00000001)<ETX><BCC>
HHU:      <SOH>P1<STX>(00000000)<ETX><BCC>
Meter:    <ACK>
HHU:      <SOH>R1<STX>S0F()<ETX><BCC>
Meter:    <STX>(0000002500000100020001 0001)<ETX><BCC>
HHU:      <SOH>B0<ETX><BCC>
```

HHU → Meter: **(Supported in Firmware Versions 4.50 to 4.xx)**

<SOH>R1<STX>S0F()<ETX><BCC>

Meter → HHU:

<STX>(data)<ETX><BCC>

with data = impulse constant data

- **(xxxxxxxxxxxxllooooovvvvvvcccccc)**
  - x = xke (secondary Impulse constant)(bcd)
  - l = LED – divider (bcd)
  - o = output – divider (bcd)
  - v = voltage transducer value **(vv vv vv ; hex)**
  - c = current transducer value **(cc cc cc ; hex)**

Example: read impulse constant

xke	= 00000025000
LED-divider	= 01
output – divider	= 0002
voltage transducer value	= 333333
current transducer value	= 444444

```
HHU:      /?!<CR><LF>
Meter:    /ABB4\@V4.50        <CR><LF>
HHU:      <ACK>041<CR><LF>
Meter:    <SOH>P0<STX>(00000001)<ETX><BCC>
HHU:      <SOH>P1<STX>(00000000)<ETX><BCC>
Meter:    <ACK>
HHU:      <SOH>R1<STX>S0F()<ETX><BCC>
Meter:    <STX>(000000250000010002333333444444)<ETX><BCC>
HHU:      <SOH>B0<ETX><BCC>
```

5.2.15 Set impulse constant

This command changes the internal impulse constants for LED and pulse output.
This command will be accepted, if it is enabled by two parameters:

- general activation by personality and class 6
- activation for log book

HHU → Meter: **(Supported in Firmware Versions 4.20 to 4.40)**

<SOH>W1<STX>S0F(data)<ETX><BCC>

with data = impulse constant data

- **(xxxxxxxxxxxxllooooovvvvcccc)**
  - o x = xke (secondary Impulse constant)(bcd) [not used]
  - o l = LED – divider (bcd)
  - o o = output – divider (bcd)
  - o v = voltage transducer value **(vvvv bcd)** [not used]
  - o c = current transducer value **(cccc bcd)** [not used]

Example: set impulse constant

xke	= 00000025000
LED-divider	= 01
output – divider	= 0002
voltage transducer value	= 3333
current transducer value	= 4444

```
HHU:      /?!<CR><LF>
Meter:    /ABB4\@V4.50        <CR><LF>
HHU:      <ACK>041<CR><LF>
Meter:    <SOH>P0<STX>(00000001)<ETX><BCC>
HHU:      <SOH>P1<STX>(00000000)<ETX><BCC>
Meter:    <ACK>
HHU:      <SOH>W1<STX>S0F(0000002500000100023333444 4)<ETX><BCC>
Meter:    <ACK>
HHU:      <SOH>B0<ETX><BCC>
```

HHU → Meter: **(Supported in Firmware Versions 4.50 to 4.xx)**

<SOH>W1<STX>S0F(data)<ETX><BCC>

with data = impulse constant data

- **(xxxxxxxxxxxxllooooovvvvvvcccccc)**
  - o x = xke (secondary Impulse constant)(bcd) [not used]
  - o l = LED – divider (bcd)
  - o o = output – divider (bcd)
  - o v = voltage transducer value **(vv vv vv ; hex)** [not used]
  - o c = current transducer value **(cc cc cc ; hex)** [not used]

Example: set impulse constant

xke	= 00000025000
LED-divider	= 01
output – divider	= 0002
voltage transducer value	= 333333
current transducer value	= 444444

```
HHU:      /?!<CR><LF>
Meter:    /ABB4\@V4.50        <CR><LF>
HHU:      <ACK>041<CR><LF>
Meter:    <SOH>P0<STX>(00000001)<ETX><BCC>
HHU:      <SOH>P1<STX>(00000000)<ETX><BCC>
Meter:    <ACK>
HHU:      <SOH>W1<STX>S0F(000000250000010002333333444444)<ETX><BCC>
Meter:    <ACK>
HHU:      <SOH>B0<ETX><BCC>
```

5.2.16 Read meter class

The parameters of the meter are stored in different classes. To read a class of the meter the following format is used:

HHU → Meter: **(Supported in Firmware Versions 4.00 to 4.xx)**

<SOH>R3<STX>Cnnllllloooo()<ETX><BCC>

Meter → HHU:

<STX>xxxx(data)<ETX><BCC>

with nn = number of the selected class of the meter
llll = number of Bytes, which should be read
(0000: read all Bytes of the class)
oooo = Offset for the Bytes to be read
data = up to 64 ASCII characters

The numbers nn, llll, oooo identify the numbers of Bytes in the EEPROM of the meter, not the number of ASCII characters, which are transmitted (max 32 Byte = 20 hex).

Example: read class 3 of the meter

```
HHU:        /?!<CR><LF>
Meter:      /ABB4\@V4.40         <CR><LF>
HHU:        <ACK>041<CR><LF>
Meter:      <SOH>P0<STX>(00003201)<ETX><BCC>
HHU:        <SOH>P1<STX>(00000000)<ETX><BCC>
Meter:      <ACK>
HHU:        <SOH>R3<STX>C030000000000()<ETX><BCC>
Meter:      <STX>xxxx
                 (0003579000001C0000000250F8A0301800984530000000080040008 0A0003151)<EOT><BCC>
HHU:        <ACK>
Meter:      <STX>xxxx
                 (0633982450200000000000100000000DCFFF0000000000000000000A00F3000C0)<EOT><BCC>
HHU:        <ACK>
            ....
HHU:        <ACK>
Meter:      <STX>xxxx
                (3149245504004000000000010000DCFFF0000023000A00F330000)<ETX><BCC>
HHU:        <SOH>B0<ETX><BCC>
```

5.2.17 Write meter class

The configuration of the meter is stored in different classes.

To write a parameter class to the meter the following format is used:

HHU → Meter: **(Supported in Firmware Versions 4.00 to 4.xx)**

<SOH>W1<STX>Cnnllllloooo(data)<ETX><BCC>

with nn = number of the selected class of the meter
llll = number of Bytes, which should be written in the EEPROM of the meter
oooo = Offset for the Bytes to be written
data = up to 48 ASCII characters

The numbers nn, llll, oooo identify the numbers of Bytes in the EEPROM of the meter, not the number of ASCII characters, which are transmitted (max. 24 Byte = 18 hex).

Example: write class 4 to the meter

```
HHU:        /?!<CR><LF>
Meter:      /ABB5\@V4.40         <CR><LF>
HHU:        <ACK>051<CR><LF>
Meter:      <SOH>P0<STX>(00212630)<ETX>d Soll: d
HHU:        <SOH>P1<STX>(00000000)<ETX>a
Meter:      <ACK>
HHU:        <SOH>W1<STX>C0400180000
                        (30303030303031E40918264E000008400608100608020000)<ETX>
Meter:      <ACK>
HHU:        <SOH>W1<STX>C0400180018
                        (0300000800000A00000B0000010C01008414330D981A5903)<ETX>
Meter:      <ACK>
HHU:        <SOH>W1<STX>C0400180030
                        (60038820912008311131004103418450875004610761967O)<ETX>
Meter:      <ACK>
HHU:        <SOH>W1<STX>C0400180048
                        (997081808480009103919200950012111511000000000000)<ETX>
Meter:      <ACK>
HHU:        <SOH>W1<STX>C0400180060
                        (000000000000000000000000000000000000000000000000)<ETX>
Meter:      <ACK>
HHU:        <SOH>W1<STX>C0400180078
                        (000000000000000000000000000000000000000000000000)<ETX>
Meter:      <ACK>
HHU:        <SOH>W1<STX>C0400180090
                        (000000000000000000000000000000000000000000000000)<ETX>
Meter:      <ACK>
HHU:        <SOH>W1<STX>C04001800A8
                        (000000000000000000000000000000000000000000000000)<ETX>
Meter:      <ACK>
HHU:        <SOH>W1<STX>C04001800C0
                        (000000000000000000000000000000000000000000000000)<ETX>
Meter:      <ACK>
HHU:        <SOH>W1<STX>C04001800D8
                        (000000000000000000000000000000000000000000000000)<ETX>
Meter:      <ACK>
HHU:        <SOH>W1<STX>C04001800F0
                        (000000000000000000000000000000000000000000000000)<ETX>
Meter:      <ACK>
HHU:        <SOH>W1<STX>C0400150108
                        (0000300201820001820001820001820011111111 4C)<ETX>
Meter:      <ACK>
HHU:        <SOH>B0<ETX>q
```

5.2.18 Reset register data

With that command all registers of the meter (energy, demand register, ...) will be reset to zero.

HHU → Meter: **(Supported in Firmware Versions 4.00 to 4.xx)**

<SOH>W1<STX>P01()<ETX><BCC>

Example: reset registers

```
HHU:        /?!<CR><LF>
Meter:      /ABB4\@V4.40         <CR><LF>
HHU:        <ACK>041<CR><LF>
Meter:      <SOH>P0<STX>(00000001)<ETX><BCC>
HHU:        <SOH>P1<STX>(00000000)<ETX><BCC>
Meter:      <ACK>
HHU:        <SOH>W1<STX>P01()<ETX><BCC>
Meter:      <ACK>
HHU:        <SOH>B0<ETX><BCC>
```

5.2.19 Set Personality

This command can be used to upgrade the meter functionality.

HHU → Meter: **(Supported in Firmware Versions 4.00 to 4.xx)**

<SOH>W1<STX>P02(data)<ETX><BCC>

with data = 3 Bytes Personality (hex format, 3. Byte is CRC)

Example: Set personality (enable all parameters = “FFFF”)

```
HHU:        /?!<CR><LF>
Meter:      /ABB4\@V4.40         <CR><LF>
HHU:        <ACK>041<CR><LF>
Meter:      <SOH>P0<STX>(00000001)<ETX><BCC>
HHU:        <SOH>P1<STX>(00000000)<ETX><BCC>
Meter:      <ACK>
HHU:        <SOH>W1<STX>P02(FFFF00)<ETX><BCC>
Meter:      <ACK>
HHU:        <SOH>B0<ETX><BCC>
```

5.2.20 Certification mode

With that command the certification mode can be activated and several features of the meter to reduce testing time can be enabled.

HHU → Meter: **(Supported in Firmware Versions 4.00 to 4.xx)**

<SOH>W1<STX>P03(data)<ETX><BCC>

with data = configuration data for certification mode

- **(AABBCCDDEEFFGGGG)** certification mode data:

AA (1 Byte)	Format: '0 0 0 kyz div msk tar onoff' (8 Bits) with kyz = use pulse output divider from GG div = use LED divider from byteCC msk = use LED-quantity from byte DD tar = use tariff source from byte BB onoff = 1 certification mode ON, =0 certification Mode OFF
BB (1 Byte)	Select tariff source (**Bit 0, 1**) for demand: Format: 0 0 0 0 0 0 d d (2 Bits) with 11 = internal switch clock 10 = internal ripple control receiver 01 = external control inputs 00 = tariff according byte FF Select tariff source (**Bit 4, 5**) for energy: Format: 0 0 e e 0 0 0 0 (2 Bits) with 11 = internal switch clock 10 = internal ripple control receiver 01 = external control inputs 00 = tariff according byte EE
CC (1 Byte)	Pulse divider for LED range 0..FFhex (see below)
DD (1 Byte)	Format: 'P+ P- S+ S- Q4 Q3 Q2 Q1' (8 Bits) required LED quantity
EE (1 Byte)	Format: 0 0 0 0 r r 0 0 (8 Bits) with rr = 00 energy tariff T1 rr = 01 energy tariff T2 rr = 10 energy tariff T3 rr = 11 energy tariff T4
FF (1 Byte)	Format: 0 0 0 0 M1 M2 M3 M4 (8 Bits) required demand tariff M1..M4
GGGG (2 Bytes)	pulse divider for pulse outputs range: 0..FFFFhex

The LED-divider divides the internal energy pulses to the desired LED pulse constant. The parameterised LED divider can be read with the following command:

HHU → Meter:

<SOH>R3<STX>C0000010018()<ETX><BCC>

Meter → HHU:

<STX>xxxx(04)<ETX><BCC>

The increasing of the divider will decrease the pulse constant.

Example: Switch certification mode ON, tariff source = internal clock

LED quantity: +P, -P
LED pulse constant: max

```
HHU:      /?!<CR><LF>
Meter:    /ABB4\@V4.40        <CR><LF>
HHU:      <ACK>041<CR><LF>
Meter:    <SOH>P0<STX>(00000231)<ETX><BCC>
HHU:      <SOH>P1<STX>(00000000)<ETX><BCC>
Meter:    <ACK>
HHU:      <SOH>W1<STX>P03(1F0301C00000001)<ETX><BCC>
Meter:    <ACK>
HHU:      <SOH>B0<ETX><BCC>
```

5.3 R5, W5 Commands

5.3.1 Read single register data

The command to readout a single register is realised according the VDEW specification 2.0:

HHU → Meter: **(Supported in Firmware Versions 4.00 to 4.xx)**

<SOH>R5<STX>Kz()<ETX><BCC>

Meter → HHU:

<STX>Kz(data)<ETX><BCC>

with Kz = identifier (OBIS number)
data = read data (format depends on the requested value)

Example: read energy register 1.8.1 of the meter

Kz: 1.8.1
Result: 000234.51 kWh

```
HHU:      /?!<CR><LF>
Meter:    /ABB4\@V4.40        <CR><LF>
HHU:      <ACK>041<CR><LF>
Meter:    <SOH>P0<STX>(00000231)<ETX><BCC>
HHU:      <SOH>R5<STX>1.8.1()<ETX><BCC>
Meter:    <STX>1.8.1(000234.51*kWh)<ETX><BCC>
HHU:      <SOH>B0<ETX><BCC>
```

5.3.2 Read “time&date” (standard VDEW)

To read time&date we use the single register data readout (see 5.3.1).
The command to readout time&date is realised according the VDEW specification 2.0:

HHU → Meter: **(Supported in Firmware Versions 4.00 to 4.xx)**

<SOH>R5<STX>Kz()<ETX><BCC>

Meter → HHU:

<STX>Kz(data)<ETX><BCC>

with Kz = identifier
- identifier for time (0.9.1 or 1-1:0.9.1)
- identifier for date (0.9.2 or 1-1:0.9.2)

data = 2 different data formats
- **(hh:mm:ss)** for time
  - hh = hour (0..23)
  - mm = minute (0..59)
  - ss = second (0..59)
- **(yy-mm-dd)** for date
  - yy = year (00..99)
  - mm = month (1..12)
  - dd = day (1..31)

Remark:
- Independent from the configuration (with or without medium and channel) the meter response with his identifier and the required time/date
- The reading format for the time is configurable to (hh:mm) or (hh:mm:ss)

Example: read time&date of the meter

Identifier for the time: 0.9.1
Identifier for the date: 0.9.2
Time and Date: 21.11.2000, 15:23:57

```
HHU:      /?!<CR><LF>
Meter:    /ABB4\@V4.40          <CR><LF>
HHU:      <ACK>041<CR><LF>
Meter:    <SOH>P0<STX>(00000231)<ETX><BCC>
HHU:      <SOH>R5<STX>0.9.2()<ETX><BCC>
Meter:    <STX>0.9.2(00-11-21)<ETX><BCC>
HHU:      <SOH>R5<STX>0.9.1()<ETX><BCC>
Meter:    <STX>0.9.1(15:23:57)<ETX><BCC>
HHU:      <SOH>B0<ETX><BCC>
```

5.3.3 Set “time&date” (standard VDEW)

The command to set time&date is realised according the VDEW specification 2.0:

HHU → Meter: **(Supported in Firmware Versions 4.00 to 4.xx)**

<SOH>W5<STX>Kz(data)(pppppppp)<ETX><BCC>

with Kz = identifier
- identifier for time (0.9.1 or 1-1:0.9.1)
- identifier for date (0.9.2 or 1-1:0.9.2)

data = 2 different data formats
- **(shhmmss)** for time
  - hh = hour (0..23)
  - mm = minute (0..59)
  - ss = second (0..59)
  - s = season flag (will be ignored by the meter)
- **(syymmdd)** for date
  - yy = year (00..99)
  - mm = month (1..12)
  - dd = day (1..31)
  - s = season flag (will be ignored by the meter)

pp... = password of the meter

Remark:
- Independent from the configuration (with or without medium and channel) the identifier for time&date must not include medium&channel
- After receive of the W5-command the meters starts a new demand period only, if the time difference between the meter and the transferred time is outside a time window (standard: +/-9s)

Example: set time&date of the meter

Identifier for the time: 0.9.1
Identifier for the date: 0.9.2
Time and Date: 21.10.2000, 20:10:35

```
HHU:      /?!<CR><LF>
Meter:    /ABB4\@V4.40        <CR><LF>
HHU:      <ACK>041<CR><LF>
Meter:    <SOH>P0<STX>(00000231)<ETX><BCC>
HHU:      <SOH>P1<STX>(00000000)<ETX><BCC>
Meter:    <ACK>
HHU:      <SOH>W5<STX>0.9.2(0001021)(00000000)<ETX><BCC>
Meter:    <ACK>
HHU:      <SOH>W5<STX>0.9.1(0201035)(00000000)<ETX><BCC>
Meter:    <ACK>
HHU:      <SOH>B0<ETX><BCC><CR><LF>
```

5.4 Profiles & Log file

With the load profile memory approved by the german PTB, load profile memory contents are interrogated and output in conformity with EN 62056-21.
The contents of the output data records are formatted in terms of their data structure in conformity with EN 62056-61(OBIS).

FEATURES OF THE PROFILE MEMORY

- Load profiles are read out using the formatted "R5" command, which causes a load profile formatted with OBIS to be output. The reply generated by the meter here is given as a self-sufficient telegram.
- Recorded profiles can be deleted using the "W5 " command defined to supplement EN 62056-21.
  Please note: Erasing the load profile memory will automatically erase the log file, but has no affect on the instrumentation profile. Erasing the instrumentation profile has no affect on the load profile or log book.
- If the meter does not support the OBIS Identifier requested, it will return this as an echo response. The part contained in the reply telegram between the two brackets (which function as separators) is omitted completely.
- If in conjunction with the profile readout there is no entry in the inquiry for the OBIS Identifier of a measured value, the meter will respond with all available measured values in its profile.
- If the meter does not incorporate an internal device clock, then the following data will be output instead of the time stamp:
  - for the date of the string: "999999" (OBIS Format: D6)
  - for the time of the string: "999999" (OBIS Format: Z6)
  - for the time stamp of the string: "999999999999"
- The telegram formed as the reply corresponds to the form specified in OBIS. It contains in the "Address" field of the first data record the OBIS Identifier of the first profile excerpt of the reply. This is followed, in accordance with the definition specified in OBIS, by a header-specific number of bracketed additional values, to which are appended the likewise bracketed elements of the profile excerpt.
- If in the interval specified there is more than one section of the profile, then a new header will be inserted for each such section. The formation of new profile headers during profile transfer is explained with the events and status changes coded in the first 8 bits (Bits 7 to 0) of the profile status word. The time stamp in the header is assigned not to the transactions, but to the formation of the first profile value.
- The overall length of the telegram answered will depend on the size of the interval desired. The time stamps in the reply telegram are of the "ZSTs13" type. Output of the telegram's data always begins with the oldest interrogated value.
- If the order includes a request for a time range for which there are no entries, the meter will respond with "Kz (ERROR)".
- If the order requests an identifier that the meter does not support, the meter will merely supply the values for the Identifier it does know.

DEPICTION OF A LOAD PROFILE IN THE DATA TELEGRAM

The load profile memory processes the following characteristics:

- Measured values:
  1-8 channels (+P, -P, +S, -S, Q1, Q2, Q3, Q4, +Q, -Q)
- Memory depth:
  Depending on the combination of physical storage and parameter.
- Time period:
  Depending on demand integration period

The following types of measuring values can be stored in the load profile storage:

- Demand values per period
- Energy values per period
- Energy Register every period

KZ	(ZSTs13)	(S)	(RP)	(z)	$(KZ_1)(E_1)$ .. $(KZ_z)(E_z)$	(M_{W1})	...	(M_{WZ})

| <- Header of load profile entry -> | <-Meas.values of load profile-> |

* KZ	OBIS-Identifier "P.01"
* ZSTs13	Time stamp format of the oldest measured value
* S	Profile status word
* RP	Registration period in minutes
* z	Number of different measured values in one registration period
* KZ_n	Identifier of the measured values (without tariff particulars or preceding-value Identifier)
* E_1	Units of measured values
* M_{wn}	Measured values

Because of the following events the meter will send load profile data with the header information:

- With the beginning of every day
- After one of the status bits is set (the status bits are set at the end of the period)

Dependent on the FW Version of the meter the status bits of the load profile are defined as described below. All status bits are set at the end of the demand period:

Bit	Significance
b7	Power outage
b6	Power up
b5	Change of internal clock
b4	Demand Reset
b3	Seasonal switchover (summer/winter time)
b2	Measure value disturbed
b1	Running reserve exhausted
b0	Fatal device error

Remark:

- Status bit 2 “measure value disturbed” is always set when a demand period will be shortened or a power outage is longer than the demand period.
- Status bits 6 “power up” and 7 “power down“ will always be set if the event occurs during the demand period.

DEPICTION OF AN INSTRUMENTATION PROFILE IN THE DATA TELEGRAM

The instrumentation profile memory processes the following characteristics:

- Measured values:
  1-8 channels (all possible instrumentation values)

- Memory depth:
  Depending on the combination of physical storage and parameter.
- Time period:
  Depending on parameter

The following types of measuring values can be stored in the instrumentation profile storage:

- Minimum values per interval
- Maximum values per interval
- Average values per interval

KZ	(ZSTs13)	(S)	(RP)	(z)	$(KZ_1)(E_1)$ .. $(KZ_z)(E_z)$	(M_{W1})	...	(M_{WZ})

| <- Header of instrumentation profile entry -> | <-Meas.values of inst. profile-> |

* KZ OBIS-Identifier "P.02"
* ZSTs13 Time stamp format of the oldest measured value
* S Profile status word
* RP Registration period in minutes
* z Number of different measured values in one registration period
* KZ_n Identifier of the measured values (without tariff particulars or preceding-value Identifier)
* E_1 Units of measured values
* M_{wn} Measured values

Because of the following events the meter will send instrumentation profile data with the header information:

- With the beginning of every day
- After one of the status bits is set (the status bits are set at the end of the period)

Status bits of the instrumentation profile are defined as described below. All status bits are set at the end of the time period:

Bit	**Significance**
b7	Power down
b6	Power up
b5	Change of internal clock
b4	not used
b3	Seasonal switchover (summer/winter time)
b2	Measure value disturbed
b1	Running reserve exhausted
b0	Fatal device error

Remark:

- Status bit 2 “measure value disturbed” is always set when a demand period will be shortened or a power outage is longer than the demand period.
- Status bits 6 “power up” and 7 “power down“ will always be set if the event occurs during the demand period.

DEPICTION OF A LOG FILE IN THE DATA TELEGRAM

KZ	(ZSTs13)	(S)	()	(z)	$(KZ_1)(E_1)$ .. $(KZ_z)(E_z)$	(D_{w1})	...	(D_{wz})

| <- Header of log file entry -> | <- Additional log file data -> |

* KZ OBIS-Identifier "P.98"
* ZSTs13 Time stamp format of the entry
* S log file status bytes (2 or 4 bytes)
* z Number of additional log file data
* KZ_n Identifier of the additional log file data
* E_n Units of additional log file data
* M_{wn} additional log file data

Status bits of the log file:

```
0 0 0 0 0 0 0 0
| | | | | | | 1 -------------- fatal device error
| | | | | | | 2                running reserve exhausted
| | | | | | | 4                measuring value disturbed
| | | | | | | 8                seasonal switch over (summer/winter time)
| | | | | | 1 ---------------- demand reset
| | | | | | 2                  change of internal clock
| | | | | | 4                  power up
| | | | | | 8                  power down (3 phase)
| | | | | 1 ------------------ parameter changed
| | | | | 2                    error conditions: Fatal or non fatal error
| | | | | 4                    end of error conditions
| | | | | 8                    reverse power detected              (Up V4.30)
| | | | 1 -------------------- input event 1 detected             (Up V4.30)
| | | | 2                    reset of log file
| | | | 4                    reset of load profile
| | | | 8                    input event 2 detected              (Up V4.30)
| | | 1 ---------------------- change of Impulse constant
| | | 2                      terminal cover was opened
| | | 4                      unused (reserved for A1350 function)
| | | 8                      wrong password was used             (Up V4.30)
| | 1 ------------------------ unused (reserved for A1350 function)
| | 2                        phase L1 is missing
| | 4                        phase L2 is missing
| | 8                        phase L3 is missing
| n -------------------------- current season and energy tariff
|
|
|
|
|
1 --------------------------- demand tariff M4
2                             demand tariff M3
4                             demand tariff M2
8                             demand tariff M1
```

	T1	T2	T3	T4
Season 0	0	4	8	C
Season 1	1	5	9	D
Season 2	2	6	A	E
Season 3	3	7	B	F

5.4.1 Reset load profile (using R5-command)

With that command the load profile **and** log file will be reset. (see VDEW specification V2.0).

HHU → Meter: **(Supported in Firmware Versions 4.00 to 4.xx)**

<SOH>W5<STX>P.01()(pppppppp)<ETX><BCC>

with pp... = password of the meter

Example: reset load profile and log file

```
HHU:        /?!<CR><LF>
Meter:      /ABB4\@V4.40         <CR><LF>
HHU:        <ACK>041<CR><LF>
Meter:      <SOH>P0<STX>(00000001)<ETX>
HHU:        <SOH>P1<STX>(00000000)<ETX>
Meter:      <ACK>
HHU:        <SOH>W5<STX>P.01()(00000000)<ETX><BCC>
Meter:      <ACK>
HHU:        <SOH>B0<ETX><BCC>
```

5.4.2 Reset log file (using R5-command)

With that command the load profile **and** log file will be reset. (see VDEW specification V2.0).

HHU → Meter: **(Supported in Firmware Versions 4.00 to 4.xx)**

<SOH>W5<STX>P.98()(pppppppp)<ETX><BCC>

with pp... = password of the meter

Example: reset load profile and log file

```
HHU:        /?!<CR><LF>
Meter:      /ABB4\@V4.40         <CR><LF>
HHU:        <ACK>041<CR><LF>
Meter:      <SOH>P0<STX>(00000001)<ETX>
HHU:        <SOH>P1<STX>(00000000)<ETX>
Meter:      <ACK>
HHU:        <SOH>W5<STX>P.98()(00000000)<ETX><BCC>
Meter:      <ACK>
HHU:        <SOH>B0<ETX><BCC>
```

5.4.3 Reset instrumentation profile (using R5-command)

With that command the instrumentation profile will be reset. The reset of the instrumentation profile memory will not affect the load profile approved by the PTB.

HHU → Meter: **(Supported in Firmware Versions 4.00 to 4.xx)**

<SOH>W5<STX>P.02()(pppppppp)<ETX><BCC>

with pp... = password of the meter

Example: reset instrumentation profile

```
HHU:        /?!<CR><LF>
Meter:      /ABB4\@V4.40         <CR><LF>
HHU:        <ACK>041<CR><LF>
Meter:      <SOH>P0<STX>(00000001)<ETX>
HHU:        <SOH>P1<STX>(00000000)<ETX>
Meter:      <ACK>
HHU:        <SOH>W5<STX>P.02()(00000000)<ETX><BCC>
Meter:      <ACK>
HHU:        <SOH>B0<ETX><BCC>
```

5.4.4 Read profile or log file data (using R5-command)

With that command the profile data or log file data of the meter can be read out according the VDEW specification V2.0:

By reading the profile or log file a time window can be specified. Further, by reading the profile separate channels to read can be specified. (see VDEW specification V2.0).

HHU → Meter: **(Supported in Firmware Versions 4.00 to 4.xx)**

<SOH>R5<STX>Kz([syymmddhhmm];[syymmddhhmm])(Kz_1)..(Kz_x)<ETX><BCC>

with Kz = identifier
- Possible identifier:
- identifier for load profile (P.01 according VDEW-specification)
- identifier for instrumentation profile (P.02)
- identifier for log file (P.98 according VDEW-specification)
- "X" (identifier to get the load profile header, manufacturer-specific command)
- "Y" (identifier to get the log file header, manufacturer-specific command)
- "Z" (identifier to get the instrumentation profile header, manufacturer-specific command)

yy = year (00..99)
mm = month (1..12)
dd = day (1..31)
hh = hour (0..23)
mm = minute (0..59)
ss = second (0..59)
s = season flag (The season flag “s” will be ignored by the meter)
Kz_x = Identifier for single profile channels

Remark:
- The R5-command to readout profiles and log file can be send with or without a password
- The meter can be programmed that the user has to send the correct password to the meter, if he is using the R5-command for read out profiles and log file.

Example: readout load profile data

Identifier = P.01 (Load Profile)
Time window = from 13.10.2000, 00:15 -15.10.2000, 00:00

```
HHU:      /?!<CR><LF>
Meter:    /ABB4\@V4.40        <CR><LF>
HHU:      <ACK>041<CR><LF>
Meter:    <SOH>P0<STX>(00000001)<ETX>
HHU:      <SOH>P1<STX>(00000000)<ETX>
Meter:    <ACK>
HHU:      <SOH>R5<STX>P.01(00010130015;00010150000)<ETX><BCC>
Meter:    <STX>P.01(1001013001500)(00)(15)(3)(1.5)(kW)(2.5)(kW)(3.5)(kvar)<CR><LF>
          (1.202)(0.104)(0.980)<CR><LF>
          (0.657)(0.034)(0.002)<CR><LF>
          ..
          (1.334)(0.389)(0.394)<CR><LF>
          P.01(1001014001500)(00)(15)(3)(1.5)(kW)(2.5)(kW)    (3.5)(kvar)<CR><LF>
          (1.002)(0.104)(0.980)<CR><LF>
          (0.357)(0.035)(0.012)<CR><LF>
          ..
          (1.034)(0.189)(0.394)<CR><LF>
          <ETX>BCC>
HHU:      <SOH>B0<ETX><BCC>
```

Example: readout log file

Identifier = P.98 (Log File)
Time window = from 13.10.2000, 00:01 -15.10.2000, 16:00

```
HHU:        /?!<CR><LF>
Meter:      /ABB4\@V4.40         <CR><LF>
HHU:        <ACK>041<CR><LF>
Meter:      <SOH>P0<STX>(00000001)<ETX><BCC>
HHU:        <SOH>P1<STX>(00000000)<ETX><BCC>
Meter:      <ACK>
HHU:        <SOH>R5<STX>P.98(00011130001;00011211600)<ETX><BCC>
Meter:      <STX>P.98(0001121150349)(0100)()(0)<CR><LF>
            P.98(0001121150427)(0100)()(0)<CR><LF>
            P.98(0001121151104)(0014)()(0)<CR><LF>
            P.98(0001121151644)(0024)()(2)(0.9.2)()(0.9.1)()(0001121)(0151512)<CR><LF>
            P.98(0001121153312)(0084)()(0)<CR><LF>
            ..
            P.98(0001121154352)(0104)()(0)<CR><LF>
            <ETX>BCC>
HHU:        <SOH>B0<ETX><BCC>
```

Example: readout instrumentation profile data

Identifier = P.02 (Instrumentation Profile)
Time window = from 13.10.2000, 00:15 -15.10.2000, 00:00

```
HHU:        /?!<CR><LF>
Meter:      /ABB4\@V4.40         <CR><LF>
HHU:        <ACK>041<CR><LF>
Meter:      <SOH>P0<STX>(00000001)<ETX>
HHU:        <SOH>P1<STX>(00000000)<ETX>
Meter:      <ACK>
HHU:        <SOH>R5<STX>P.02(00010130015;00010150000)<ETX><BCC>
Meter:      <STX>P.02(1001013001500)(00)(15)(3)(32.5)(V)(31.5)(A)(34.7)(Hz)<CR><LF>
            (223.5)(10.043)(49.96)<CR><LF>
            (223.2)(10.045)(49.97)<CR><LF>
            (223.7)(10.049)(49.97)<CR><LF>
            ..
            (223.9)(10.043)(49.98)<CR><LF>
            P.02(1001014001500)(00)(15)(3)(32.5)(V)(31.5)(A)(34.7)(Hz)<CR><LF>
            (223.1)(10.045)(49.98)<CR><LF>
            (224.1)(10.044)(49.98)<CR><LF>
            (224.1)(10.043)(49.98)<CR><LF>
            ..
            (224.5)(10.045)(49.99)<CR><LF>
            <ETX><BCC>
HHU:        <SOH>B0<ETX><BCC>
```

5.4.5 Read profile or log file data (using R6-command)

Beside the R5-command the R6-command can be used for readout profile data or log file data too. The advantage of the R6-command is, that the readout data can be separated in different blocks with a separate checksum for every block. If one block is not transmitted correctly, only this block must be repeated (see VDEW specification V2.0).

HHU → Meter: **(Supported in Firmware Versions 4.00 to 4.xx)**

<SOH>R6<STX>Kz([syymmddhhmm];[syymmddhhmm];[xx])(Kz)..(Kz)<ETX><BCC>

with Kz = identifier

- Possible identifier:
- identifier for load profile (P.01 according VDEW-specification)
- identifier for instrumentation profile (P.02)
- identifier for log file (P.98 according VDEW-specification)
- "X" (identifier to get the load profile header, manufacturer-specific command)
- "Y" (identifier to get the log file header, manufacturer-specific command)
- "Z" (identifier to get the instrumentation profile header, manufacturer-specific command)

yy = year (00..99)
mm = month (1..12)
dd = day (1..31)
hh = hour (0..23)
mm = minute (0..59)
ss = second (0..59)
s = season flag (The season flag “s” will be ignored by the meter)
xx = block length
Kz_x = Identifier for single profile channels

The readout starts with the R6-command:

HHU → Meter:

<SOH>R6<STX>P.01(;;4)<ETX>

Behind the identifier for the profile or log file (p. ex. “P.01“ for the load profile) the starting and ending time stamp can be specified in the same way like the R5-command. Then the number of lines for one block must be defined (p. ex. „4“). This number will be interpreted from the alpha meter as the number of lines to be transmitted. The maximum block length is 1..99.

The meter answer of the first block is:

Meter → HHU:

<STX>P.01(sYYMMDDHHMMSS)(04)(15)(2)(1.8)(kWh)(2.8)(kWh)<CR><LF>
(06409.60)(06673.05)<CR><LF>
(06412.89)(06676.45)<CR><LF>
(06415.16)(06678.84)<CR><LF>
<EOT><BCC>

Every block ends with a <EOT> instead of <ETX> and the <BCC>.
After that the meter is waiting for the ”Acknowledgement Message” from the HHU. This can be either a <ACK> if the transmission was successful or a <NAK> if there was a problem

If the meter gets the <ACK> it goes on with sending the next block:

HHU → Meter:
<ACK>
Meter → HHU:
<STX>(06420.72)(06684.63)<CR><LF>
(06424.01)(06688.03)<CR><LF>
(06426.28)(06690.42)<CR><LF>
(06429.57)(06693.82)<CR><LF>
<EOT><BCC>

If the meter gets the <NAK> it will response with the repetition of the required block (Repeat Request Message):

HHU → Meter:
<NAK>
Meter → HHU:
<STX>P.01(sYYMMDDHHMMSS)(04)(15)(2)(1.8)(kWh)(2.8)(kWh)<CR><LF>
(06409.60)(06673.05)<CR><LF>
(06412.89)(06676.45)<CR><LF>
(06415.16)(06678.84)<CR><LF>
<EOT><BCC>

The last block can have a lower number of lines than specified. He ends with a <ETX>:

HHU → Meter:
<ACK>
Meter → HHU:
<STX>(06431.84)(06696.21)<CR><LF>
(06435.13)(06699.61)<CR><LF>
<ETX><BCC>

Example: readout load profile data

Identifier = P.01 (Load Profile)
Time window = from 13.10.2000, 00:15.-15.10.2000, 00:00
Block length = 4

```
HHU:        /?!<CR><LF>
Meter:      /ABB4\@V4.40         <CR><LF>
HHU:        <ACK>041<CR><LF>
Meter:      <SOH>P0<STX>(00000001)<ETX><BCC>
HHU:        <SOH>P1<STX>(00000000)<ETX><BCC>
Meter:      <ACK>
HHU:        <SOH>R6<STX>P.01(00010130015;00010150000;4)<ETX><BCC>
Meter:      <STX>P.01(1001013001500)(00)(15)(3)(1.5)(kW)(2.5)(kW)(3.5)(kvar)<CR><LF>
            (1.202)(0.104)(0.980)<CR><LF>
            (0.657)(0.034)(0.002)<CR><LF>
            (1.002)(0.104)(0.980)<CR><LF>
            <EOT>
HHU:        <ACK>
Meter:      <STX>(0.202)(0.104)(0.980)<CR><LF>
            (1.657)(1.034)(0.402)<CR><LF>
            (1.002)(0.104)(0.980)<CR><LF>
            (0.857)(0.434)(0.302)<CR><LF>
            <EOT>
HHU:        <ACK>
Meter:      <STX>...
            ..
Meter:      <STX>P.01(1001013001500(00)(15)(3)(1.5)(kW)(2.5)(kW)(3.5)(kvar)<CR><LF>
            (1.202)(0.104)(0.980)<CR><LF>
            (0.657)(0.034)(0.002)<CR><LF>
            (1.002)(0.104)(0.980)<CR><LF>
            (0.857)(0.434)(0.302)<CR><LF>
            <EOT>
HHU:        <ACK>
Meter:      <STX>()<ETX><STX><BCC>
HHU:        <SOH>B0<ETX><BCC>
```

Example: readout instrumentation profile data

Identifier = P.02 (Instrumentation Profile)
Time window = from 13.10.2000, 00:15.-15.10.2000, 00:00
Block length = 4

```
HHU:        /?!<CR><LF>
Meter:      /ABB4\@V4.40         <CR><LF>
HHU:        <ACK>041<CR><LF>
Meter:      <SOH>P0<STX>(00000001)<ETX><BCC>
HHU:        <SOH>P1<STX>(00000000)<ETX><BCC>
Meter:      <ACK>
HHU:        <SOH>R6<STX>P.02(00010130015;00010150000;4)<ETX><BCC>
Meter:      <STX>P.02(1001013001500)(00)(15)(3)(32.5)(V)(31.5)(A)(34.7)(Hz)<CR><LF>
            (223.5)(10.043)(49.96)<CR><LF>
            (223.2)(10.045)(49.97)<CR><LF>
            (223.7)(10.049)(49.97)<CR><LF>
            <EOT><BCC>
HHU:        <ACK>
Meter:      <STX>(223.5)(10.043)(49.96)<CR><LF>
            (223.2)(10.045)(49.97)<CR><LF>
            (223.7)(10.049)(49.97)<CR><LF>
            (223.2)(10.045)(49.97)<CR><LF>
            <EOT><BCC>
HHU:        <ACK>
Meter:      <STX>...
            ..
HHU:        <ACK>
Meter:      <STX>(223.2)(10.045)(49.97)<CR><LF>
            P.02(1001014001500)(00)(15)(3)(32.5)(V)(31.5)(A)(34.7)(Hz)<CR><LF>
            (223.2)(10.045)(49.97)<CR><LF>
            (223.7)(10.049)(49.97)<CR><LF>
            <EOT><BCC>
HHU:        <NAK>
Meter:      <STX>(223.2)(10.045)(49.97)<CR><LF>
            P.02(1001014001500)(00)(15)(3)(32.5)(V)(31.5)(A)(34.7)(Hz)<CR><LF>
            (223.2)(10.045)(49.97)<CR><LF>
            (223.7)(10.049)(49.97)<CR><LF>
            <EOT><BCC>
HHU:        <ACK>
Meter:      <STX>
            ..
HHU:        <ACK>
Meter:      <STX>(223.2)(10.045)(49.97)<CR><LF>
            (223.7)(10.049)(49.97)<CR><LF>
            <ETX><BCC>
HHU:        <SOH>B0<ETX><BCC>
```