RS 4 / Software

Information on the RS4 Protocol for external use

<table>
<thead>
<tr>
<th>Title</th>
<th>Information on the RS4 Protocol for external use</th>
</tr>
</thead>
<tbody>
<tr>
<td>Document No</td>
<td>xxx</td>
</tr>
<tr>
<td>Version No</td>
<td>1.0</td>
</tr>
<tr>
<td>Date</td>
<td>01.03.01</td>
</tr>
<tr>
<td>Status</td>
<td>checked (Brunner)</td>
</tr>
<tr>
<td>Author</td>
<td>Martin Kaul</td>
</tr>
<tr>
<td>Filename</td>
<td>rs4_protocolExtern.doc</td>
</tr>
</tbody>
</table>
## History

<table>
<thead>
<tr>
<th>Date</th>
<th>Vers.</th>
<th>Author</th>
<th>Status</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>01.03.2001</td>
<td>1.0</td>
<td>Martin Kaul</td>
<td>checked / Brunner</td>
<td>created using documents SER_COMM.DOC and CMDPROC.DOC</td>
</tr>
</tbody>
</table>
# Table of contents

1. Notice...................................................................................................................... .......................................4

2. Purpose of the present document ............................................................................................. ....................4

3. Requirements ................................................................................................................ ................................4

3.1. Universal requirements.................................................................................................... .........................4

3.2. Type of data.............................................................................................................. ................................4

3.3. Access restriction ........................................................................................................ .............................5

3.4. Protocol.................................................................................................................. ...................................5

3.5. Realization ............................................................................................................... .................................6

3.6. Characteristics of the interface.......................................................................................... .......................7

4. Analysis.................................................................................................................... ......................................8

4.1. Protocol.................................................................................................................. ...................................8

4.1.1. Requirement and impact .................................................................................................. ...................9

4.1.2. Specifications of the transmission protocol ............................................................................. ..........10

4.2. Structure of the user information ......................................................................................... ...................15

4.2.1. Strategies for the transmission of measurement values ............................................................. 15

4.2.2. Structure of the transmitted measurement contour ................................................................... 16

4.2.3. Transmitting errors ..................................................................................................... .......................17

4.3. Command description....................................................................................................... ......................18
1. Notice

This document is an excerpt from the documents SER_COMM.DOC and CMDPROC.DOC for external use, i.e. when an external organisation requires information regarding the structure of the RS4 protocol.

2. Purpose of the present document

This document describes the structure and protocol of the serial communication between the RS 4 and the PC configuration program.

It examines the requirements placed on the serial communication, analyses these requirements, uses this analysis to define a protocol for the serial communication and then designs the serial communication.

3. Requirements

Please find below a list of the different requirements to be met by the serial communication.

3.1. Universal requirements

Please find below a list of the universal requirements:

• Requirement 1000:
  Category: universal
  Description: The serial communication should facilitate a secure transmission of messages (with any content) between the RS 4 and a PC configuration program.

3.2. Type of data

Please find below a list of the requirements which specify the type of information to be transmitted:

• Requirement 2000:
  Category: Type of data
  Description: The serial communication must facilitate the transmission of configuration data.

• Requirement 2010:
  Category: Type of data
  Description: The serial communication must facilitate the transmission of device data.

• Requirement 2020:
  Category: Type of data
  Description: The serial communication must facilitate the output of measurement values.

• Requirement 2030:
  Category: Type of data
  Description: It should be possible to transmit measurement values sector-wise, i.e. it should be possible to suppress the output of measurement values from defined sectors.
3.3. Access restriction

Please find below a list of the requirements which apply to the access restriction of the RS 4:

- Requirement 3000:
  
  **Category:** Access restriction
  
  **Description:** When transmitting messages from the PC configuration program to the RS 4, the password for the current authorisation level must always be transmitted together with the message. Messages without a password must be ignored by the RS 4.

3.4. Protocol

Please find below a list of the requirements which are required for specifying the transmission protocol:

- Requirement 4000:
  
  **Category:** Protocol
  
  **Description:** Messages should be transmitted optionally with/without acknowledgement.

  The condition, whether the transmission takes place with or without acknowledgement, affects the transmitter (RS4 or PC program) of a message depending on the type of data/information to be transmitted.

  Security-critical data are, in this case, always transmitted with an acknowledgement. This setting is permanently programmed into the respective program.

- Requirement 4010:
  
  **Category:** Protocol
  
  **Description:** Messages are not transmitted within a network.

- Requirement 4020:
  
  **Category:** Protocol
  
  **Description:** Errors which could occur during the transmission must be detectable by the serial communication.

- Requirement 4030:
  
  **Category:** Protocol
  
  **Description:** Binary data should be used for the transmission of messages, i.e. the transmission makes use of all user bits of the serial interface.

- Requirement 4040:
  
  **Category:** Protocol
  
  **Description:** The connection of a terminal over the serial interface of the RS 4 is not intended.
• Requirement 4050:
  Category: Protocol
  Description: The following status information is transmitted each time the RS 4 performs an output function:
  - Operating status
  - Protected fields busy
  - Warning
  - Error
  - Restart-disable
  - Identifier of the current protected field pair

• Requirement 4060:
  Category: Protocol
  Description: The protocol should transmit information with as little redundancy as possible.

• Requirement 4070:
  Category: Protocol
  Description: It should be possible to transmit measurement values both word-wise (16 bit) as well as byte-wise (8 bit).
  16 bit: The resolution is 2 mm, i.e. 0 … approx. 65 m
  8 bit: The resolution is 16 mm, i.e. 0 … approx. 4 m

• Requirement 4080:
  Category: Protocol
  Description: The transmission protocol must be capable of synchronising the RS 4 and PC configuration program on start and stop if an error occurs during a previous transmission.

• Requirement 4090:
  Category: Protocol
  Description: Messages from RS 4 to PC and from PC to RS 4 have the same structure.

### 3.5. Realization

Please find below a list of requirements which are necessary for realizing the serial communication:

• Requirement 5000:
  Category: Realization
  Description: Only a single transmission buffer should be used in the RS 4 for the output of messages.

• Requirement 5010:
  Category: Realization
  Description: Only a single reception buffer should be used in the RS 4 for the reception of messages.
• Requirement 5020:
  Category: Realization
  Description: The reception buffer is not overwritten until a received message has been processed.

• Requirement 5030:
  Category: Realization
  Description: The transmission buffer is not overwritten until a message to be transmitted has been output.

• Requirement 5040:
  Category: Realization
  Description: The serial communication must function effectively and with an efficient use of resources.

3.6. Characteristics of the interface

Please find below a list of the requirements which are necessary for setting the serial interface.

• Requirement 6000:
  Category: Characteristics of the interface
  Description: The following serial interface settings should be available:
  - Baud rate 4800 baud and more
  - Data format: 8N1
4. **Analysis**

This chapter analyses the various requirements listed in chapter 0 *Requirements* and, using this analysis, establishes a transmission protocol for the communication between the RS 4 and the PC configuration program. The analysis is also used to create transmission strategies for the transmission of data.

4.1. **Protocol**

In general, a distinction must be made between the protocol and the interpretation of the transmitted data. The protocol defines a language which assists the RS 4 and the PC communication program in exchanging data with one another.

In this case it must be irrelevant to the protocol which type of data are being transmitted.

In order to achieve this, the protocol generates a frame for interpreting the data to be transmitted independent of the data. In this way the protocol defines for each message to be transmitted a header, which precedes the actual information, and a footer, which is appended to the actual information. Each message between the two communicating partners is composed of such a package with a header, the information itself and a footer.

The following diagram illustrates this:

Sending a message from the RS 4 to the PC configuration program:

![Diagram of Protocol](image)

Sending a message from the PC configuration program to the RS 4:

![Diagram of Protocol](image)

- [ ] Protocol relevant data
- [ ] Not protocol relevant data

Figure 0-1 Protocol relevance

In this chapter the requirements from chapter 0 *Requirements* are analysed and the header and footer formats specified according to these requirements.
4.1.1. Requirement and impact

This chapter lists the relevant requirements and their corresponding impacts. These impacts are then converted into a protocol specification.

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>3000</td>
<td>The frame must contain a field in which a password can be transmitted together with the message.</td>
</tr>
</tbody>
</table>
| 3000 4090   | It must be possible to set for each message whether or not a password is to be included in the transmission, as the password is only required for messages transmitted from the PC configuration program to the RS 4 and the protocol structure must be identical in both directions. Messages from PC to RS4 thus always include the password. Messages from RS4 to PC include no password.  
  -> Option |
| 4000        | It must be possible to set for each message whether or not an acknowledgement is desired. This setting is permanently programmed into the respective program.  
  -> Option |
| 4010        | An address can be omitted. |
| 4020        | A checksum must be transmitted within the frame in order to detect transmission errors.  
  A time-out is specified between the individual characters in order to detect a missing message end. |
| 4030 4040 4080 6000 | No extra bits may be used for identifying the frame, as all data bits of the individual characters are required for the data transmission.  
  Therefore, a token must be used for identifying the frame. A token is a special combination of characters which does not occur in the data transmission. As, in principle, all combinations of characters are possible when transmitting binary data, the corresponding combination of characters must be suppressed when transmitting the binary data.  
  It is best when the combination of characters representing a token appears as seldom as possible within the frame.  
  Readability of the messages on a terminal is not necessary, as the data can be transmitted in binary format without worry. |
Requirement: Impact

4050 | The following additional information must also be transmitted within the frame:
- Operating status of at least 4 different states
- act. personal safety field busy
- act. object safety field busy
- 2. personal safety field busy
- 2. object safety field busy
- selected pair of protected fields
- 2. pair of protected fields
- Warning
- Error
- Restart-disable

In order for everything to function quickly, the states of the protected fields + error + warning + restart-disable do not need to be transmitted when they are not set (i.e. "Off").

The operating status should always be included in the transmission.

-> Option

Table 0-1 Requirements and their impacts

4.1.2. Specifications of the transmission protocol

This chapter describes the specifications of the transmission protocol. For this purpose, the requirements from chapter 0 Requirements and the impacts from chapter 0 Requirement and impact are used.

Each message transmitted by the RS 4 to the PC configuration program or from the PC configuration program to the RS 4 has the following structure:

<table>
<thead>
<tr>
<th>End</th>
<th>Control char.</th>
<th>Actual information</th>
<th>Password</th>
<th>Option</th>
<th>Command</th>
<th>Start</th>
</tr>
</thead>
</table>

Two consecutive 0x00 are used as token characters, i.e. identifier of a special value (control character). Within the actual information, the occurrence of a \{ 0x00, 0x00 \} is extended with a 0xFF to avoid confusing with the token character, i.e. it must be prevented that a 0xFF follows a token in the protocol (by means of an appropriate specification). In this way, the occurrence of a \{ 0x00, 0x00, 0xFF \} can be unmistakably resolved as \{ 0x00, 0x00 \}. The additional 0xFF is also used in the calculation of the checksum.

Please find below a description of the structure

<table>
<thead>
<tr>
<th>Name</th>
<th>Numb. Bytes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start</td>
<td>2</td>
<td>The start character consists of the token (i.e. 0x00,0x00) Together with the following command character, which possesses a value range from 0x01 - 0xFE, the start sequence (consisting of start character and command character) can be unmistakably different from the end character and from a { 0x00, 0x00, 0xFF } in the user information.</td>
</tr>
<tr>
<td>Command</td>
<td>1</td>
<td>The command character possesses a value range from 0x01 - 0xFE. The command character specifies the method with which the user information which is to be transmitted must be interpreted. The command character is not analysed by the protocol, but rather according to the given operating status.</td>
</tr>
<tr>
<td>Name</td>
<td>Numb. Bytes</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| Option 1 | 1           | The option characters transmit certain control flags, e.g. status information, which is transmitted together with each message.  

The first option character is always transmitted and defines protocol information, e.g. the total number of option characters being transmitted. All other option characters are transmitted only when required by the transmission, i.e. when at least one status bit is set.  

This option character is always transmitted together with the message being transmitted and has the following structure:

```
<table>
<thead>
<tr>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

- **Bit 0**: Number of option fields, i.e. at least 1
- **Bit 1**: Presently operating status
  - 0: without password
  - 1: with password
- **Bit 2-4**: Field 0x00 indicates the currently active operating status and has the following structure:
  - 000 - no data (for messages from PC -> RS 4)
  - 001 - initialization
  - 010 - measurement operation
  - 011 - configuration
  - 100 - error/fault

As the number of option fields must be at least 1, the field **Option 1** never has the value 0x00.

| Option 2 | 1           | This option character transmits status information regarding the individual protected fields + error + warning + restart-disable.  

This option character is only transmitted when at least one flag is set, i.e. this field cannot possess the value 0x00. If this field is transmitted, the number of option fields in field **Option 1** must be at least 2. |
This option character has the following structure:

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>Act. personal safety field</td>
</tr>
<tr>
<td></td>
<td>0: free</td>
</tr>
<tr>
<td></td>
<td>1: busy</td>
</tr>
<tr>
<td>6</td>
<td>Act. object safety field</td>
</tr>
<tr>
<td></td>
<td>0: free</td>
</tr>
<tr>
<td></td>
<td>1: busy</td>
</tr>
<tr>
<td>5</td>
<td>Warning</td>
</tr>
<tr>
<td>4</td>
<td>Error</td>
</tr>
<tr>
<td>3</td>
<td>Restart, disable</td>
</tr>
<tr>
<td>2</td>
<td>2. Personal safety field</td>
</tr>
<tr>
<td></td>
<td>0: free</td>
</tr>
<tr>
<td></td>
<td>1: busy</td>
</tr>
<tr>
<td>1</td>
<td>2. Object safety field</td>
</tr>
<tr>
<td></td>
<td>0: free</td>
</tr>
<tr>
<td></td>
<td>1: busy</td>
</tr>
<tr>
<td>0</td>
<td>= 1 (for option 3)</td>
</tr>
</tbody>
</table>

This field has no meaning for messages from the PC to the RS 4.

Bit 7 is set to 1 when Option 3 needs to be transmitted to prevent Option 2 from taking the value 0.
<table>
<thead>
<tr>
<th>Name</th>
<th>Numb.</th>
<th>Bytes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Option 3</td>
<td>1</td>
<td></td>
<td>This option character transmits status information regarding the selected protected field pair.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>This option character is only transmitted when at least one flag is set, i.e. this field cannot possess the value 0x00. If this field is transmitted, the number of option fields in field Option 1 must be at least 3.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>This option character has the following structure:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><img src="image" alt="Option 3 Structure Diagram" /></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>As it is possible that two protected field pairs be simultaneously selected during protected field switching, two pieces of information are required for the selected protected field pair.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>for the selected protected field pairs:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>· 0: means nothing selected</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>· 1-4: means protected pair 1-4 selected</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>This field has no meaning for messages from the PC to the RS 4.</td>
</tr>
<tr>
<td>Password</td>
<td>8</td>
<td></td>
<td>The password is transmitted in this field when bit 5 of Option 1 is set. The number of characters is always 8.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>The structure of the individual characters is as follows:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><img src="image" alt="Password Structure Diagram" /></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>i.e. bit 7 of the individual characters is always set. This is necessary to prevent consecutive 0x00.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>The individual characters of the password thus have a value range from 0x00 - 0x7F.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>If the password has no content (e.g. authorisation level 1), 0xFF is transmitted 8 times.</td>
</tr>
<tr>
<td>Actual</td>
<td></td>
<td>variable</td>
<td>All actual information is transmitted with 8 bits. The actual information is not interpreted by the protocol, but rather depending on the command and the current operating status of a command process.</td>
</tr>
</tbody>
</table>
Table 0-2 Protocol structure

<table>
<thead>
<tr>
<th>Name</th>
<th>Numb. Bytes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Check characters</td>
<td>1</td>
<td>The check character corresponds to an XOR link of all transmitted characters, including command characters, password, option characters, i.e. following the start character to the last character preceding the check character. In order to prevent confusing with the end character, the check character must never possess the value 0x00. If the result of the XOR link is determined to be a 0x00, a 0xFF is transmitted and considered in the analysis. Comment: when transmitting configuration data, a CRC signature must be calculated and transmitted for the user information of all messages. This increases the level of security during the transmission of configuration data.</td>
</tr>
<tr>
<td>End</td>
<td>3</td>
<td>The end character consists of the token (i.e. 0x00 0x00) followed by a 0x00. The end character can, thus, be unmistakably different from the start sequence (start character and command character) and a (0x00, 0x00, 0xFF).</td>
</tr>
</tbody>
</table>

The protocol makes it possible for the transmitter and receiver to mutually synchronize one another. This is because the reception of (0x00, 0x00) followed by a character not equal to 0x00 or 0xFF always identifies the start of a message and the reception of (0x00, 0x00, 0x00) always identifies the end of a message.

A message which could not be correctly transmitted is not repeated by the protocol. The transmitter service is responsible for the repetition of faulty messages (for example, when transmitting the configuration data, the function which generated the message).
4.2. Structure of the user information

This chapter describes the function of the data which are exchanged with the aid of the transmission protocol between the RS 4 and the PC configuration program.

In general, the following types of data are exchanged:

- **Measurement values**
  During measurement operation, the measured values are transmitted from the RS 4 to the PC configuration program.

- **Configuration data**
  The PC configuration program loads the configuration data from the RS 4 and also stores them on the RS 4, i.e. the configuration data are transmitted both from the RS 4 to the PC configuration program as well as from the PC configuration program to the RS 4.

- **Diagnostic data**
  With the aid of the PC configuration program, it is possible to query various diagnostic values from the RS 4, i.e. the diagnostic data are transmitted upon request by the PC configuration program from the RS 4 to the PC configuration program.

Various commands are defined for the transmission of the various data and for the control of the RS 4. The received data can be unmistakably identified by the receiver with the aid of the commands (for a list of the commands, see document Command Processing (Level 2) CMDPROC.DOC)

4.2.1. Strategies for the transmission of measurement values

The RS 4 measures the distance to the next object over an area of 190°. The calculated distance results are internally analysed for protected field monitoring, but may, however, be transmitted for external analysis/display via the serial interface of the RS 4.

Please find below a list of the associated data volumes and transmission times:

- The incremental encoder possesses 1000 edge changes over the entire rotation
- The measurement range covers 190°, i.e. 529 measurement values are measured in the measurement range, i.e. 529 measurement values are measured per scan.
- The measurement values have a value range of 16 bits, where the resolution is approximately 2 mm, i.e. 2 bytes are required for the transmission of a measurement value.
- The measurement takes place at 25 rotations/sec. i.e. each rotation is 40 ms in duration. As a scan takes place over 190°, each scan is 21.1 ms in duration.
- Within a period of 21.1 ms, 529 measurement values are determined, i.e. 1 measurement value is determined every 40 µs.

Each scan produces a data volume of 2*529 bytes = 1058 bytes.
Plus the protocol overhead of 10 bytes (start+command+option+check character+end) yields a packet of 1068 bytes / scan for the transmission of all measurement values.

At a baud rate of 57600 baud, 185 ms are required for a packet, i.e. at a baud rate of 57600 baud not all measurement values for a scan can be transmitted.

**Realization:**

With the RS4, all measurement values are transmitted with 16 bits, where bit 0 specifies whether one of the active protected fields has been violated between the previously output measurement value and the given measurement value. These bits can be used on the PC when displaying the locations in the measurement value contour which violate an active protection field.
The resolution and size of the measurement value output can be defined in such a way that the output begins at a particular sector and ends at a particular sector and only every n-th value (beginning with the output start sector) is output. In this way the size of the measurement value output can be reduced.

The measurement value message is output by means of the RS4 protocol without acknowledgement. As the output of an entire measurement value message may take several rotations, depending on the quantity of data and the baud rate, the output of the measurement value message within the area being monitored is interrupted. While a measurement value is being output, scanning can, therefore, continue and the defined area monitored without overwriting the current measurement value output.

4.2.2. Structure of the transmitted measurement contour

This chapter describes the structure of the user data of the communication protocol during the transmission of a measurement contour from the RS4 to the PC configuration program.

Notice: Each measured contour possesses 529 measurement values (angle -5.04° - 185.04° with 0.36° resolution). Each individual measurement value possesses a number, beginning with 0, i.e. the measurement value at position -5.04° has the number 0 and the measurement value at position 0° has the number 14. Within a measurement message, it is not absolutely necessary that all values of the contour being measured be transmitted (see resolution, output start and output stop).

The following data are transmitted within a measurement contour:

<table>
<thead>
<tr>
<th>Address within the user data</th>
<th>Name</th>
<th>Description</th>
<th>Size in bytes</th>
</tr>
</thead>
</table>
| 0                           | Scan number | The scan number is incremented by one with each rotation of the RS4. This scan number can be used to determine the temporal separation between two output measurement contours. The actual scan number is 32 bytes in size. In order to prevent a smaller value from creating a double zero during the transmission of 4 bytes, filler bytes with the value 0xFE are inserted between the individual bytes of the scan number. The scan number is thus structured as follows:  
  - Bit 24-31  
  - 0xFE  
  - Bit 16-23  
  - 0xFE  
  - Bit 8-15  
  - 0xFE  
  - Bit 0-7  
  - 0xFE | 8          |
| 8                           | Resolution | This value specifies the distance between two individually transmitted measurement values. Assuming that the measurement contour is output beginning with measurement value 0 and ending with 528 and the resolution is set to value 4, the following values of the measured contour are then output:  
  - 0  
  - 4  
  - 8  
  - ...  
  - 520  
  - 524  
  - 528 | 1          |
| 9                           | Output start | This value specifies with which measurement value the output of the measured contour is to begin. | 2            |
Address within the user data | Name | Description | Size in bytes
--- | --- | --- | ---
11 | Output stop | This value specifies which measurement value is to be transmitted last. If the ratio of the number of measurement values to be output (as specified by output start and output stop) to the resolution is not divisible, the measurement value corresponding to the output stop is output in any case. | 2
13 | Measurement values | The individual measurement values are output in sequence, beginning with the measurement value corresponding to output start and ending with the measurement value corresponding to output stop. Each measurement value corresponds here to the distance measured at the respective angular position in millimetres. Bit 0 of a given measurement value specifies whether the measured contour of a protected field has been violated between the given measurement value and the preceding measurement value, i.e. bit 0 must be masked out when defining the measurement value as a distance value. | -

When transmitting 16-bit values within the message, the high byte is always transmitted first followed by the low byte.

### 4.2.3. Transmitting errors

Errors and warnings may occur due to various events. In the RS 4 errors are stored in an error buffer.

An error/warning consists of the following parts:

- **Number** (2 bytes)
  - Specifies the given error/warning
- **Parameter** (2 bytes)
  - Additional parameter for the given error/warning number, e.g. in the event of window soiling an indicator of which area of the window is soiled.
- **Error location** (2 bytes)
  - The location at which the error/warning occurred.

For 16-bit values, first the high byte and then the low byte is transmitted.

The error numbers + corresponding parameters are described in a separate realization document.

The PC configuration program can read out the error buffer of the RS 4. For this purpose, the PC configuration program sends a request to the RS 4. The RS 4 then sends the entire content of the error buffer to the PC configuration program.
4.3. Command description

This chapter lists all commands which are transmitted between the RS 4 and the PC configuration program.

The individual commands may possess a value of 0x01 - 0xFE - for further information, see document *Serial Communication (Level 3) (SER_COMM.DOC)*

<table>
<thead>
<tr>
<th>Command</th>
<th>Group</th>
<th>Significance (HEX)</th>
<th>Description</th>
<th>RS 4 → PC</th>
<th>PC → RS 4</th>
<th>with password</th>
<th>with acknowledgement</th>
<th>min. Access Level</th>
<th>Operating mode</th>
<th>Operating mode</th>
<th>with user information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mess 16 RT</td>
<td>Measurement</td>
<td>0x21</td>
<td>Transmission of the measurement values in 16-bit format with current scan number. This number is incremented by 1 with each rotation of the mirror and can be used to determine the distance between two successive measurement value outputs</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>M</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Error Occur</td>
<td>Error</td>
<td>0x53</td>
<td>This command is sent by the RS 4 to the PC configuration program in the event of an error. The error message is included in the transmission as a parameter (consisting of error number, parameter and error location).</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>M/KE</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Warning Occur</td>
<td>Error</td>
<td>0x54</td>
<td>This command is sent by the RS 4 to the PC configuration program in the event of a warning. The warning message is included in the transmission as a parameter (consisting of warning number, parameter and warning location).</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>M/KE</td>
<td>x</td>
<td></td>
</tr>
</tbody>
</table>

Table 0-3 Commands

If the recipient of a message does not know a command, it is ignored. If, as per the acknowledgement flag of the protocol, the transmitter expects an acknowledgement, the receiver sends in this case a negative acknowledgement.