SPA-Bus
Communication Protocol V2.5

Technical description

Issued 96-01-23
Modified 2001.07.31
Version C
Checked MP
Approved TK

Data subject to change without notice

SPA-bus Communication Protocol V2.5

Contents

APPENDICES:
1. Reserved SPA-bus data items.
2. SPA-bus hardware recommendations.
3. Implementation limitations.
4. Extensions: String transfer with exrended character set
5. SPA File Transfer

Updates

Version 1.0, 20.09.84
- Basic version

Version 2.0, 07.02.85
- External master facility omitted.
- Start-of-message and end-of-message characters altered.
- Numbering of events and use of time marking redefined.
- Categorization of event types omitted.
- Subtype 3 of slave message type N altered.
- Checksum computation method redefined.
- Length limitation of slave message F relaxed.

Version 2.1, 26.03.85
- Data category B meaning altered.
- Maximum length of message altered to be 255 characters. (Old value 256 characters.)

Version 2.2, 10.10.85
- Chapter 5 rewritten.
- Meaning of type N message error codes altered.
- Data definitions of categories I, O, S, V, and M redefined.
- Wait delays extended.
- Restrictions on slave data formats relaxed. (Fixed format of slave data canceled.)
- Master message length limited to one line.
- Facility for interrupting slave message send canceled.
- Appendix 1 included in documentation.

Version 2.3, 05.08.86
- Bus renamed as SPA bus.
- Data format of categories I, O, S, V, and M restricted. (Data for normal use to be given in decimal format.)

Version 2.4, 21.02.1992
The main reason for publishing this new version is to state in written format how the present SPA-bus implementations have been done.

- The confirmed data write to slaves using messages of types L, S and E was omitted.
- The possibility of "Interrupting slave message send" was omitted.
- The SPA-bus special characters are defined.
- The data presentation in categories I, O, S and V was clarified. Handling rules of data values in master and slave units was included.
- The possibility to send analog data together with events was included (so called analog events).
- New data category D was introduced to enable broadcasting of time and date.
- Data category A (active alarms) can now be requested also per channel.
- Appendix 1 was rewritten.
- Appendices 2 and 3 were included.

Version 2.5, 23.01.1996
Appendices 4 and 5 were included.

1.0 Introduction

The SPA-bus is was originally designed as a fieldbus in a distributed protection, control and event reporting system. The system may incorporate slave units as protective relays, control units and alarm units connected over the SPA-bus to a master unit. In addition to this the bus can be used to transfer data to and from other units of a secondary equipment system in a substation of an electrical power network.

Defined in this document is the communication protocol used in communications over the SPA bus.

Appendix 2 includes recommendations for the physical implementation of the bus. Usually - especially in protection and control systems of a substation- the physical implementation is fiber optic bus/loop. Some other media can also be used if the environment does not cause severe electrical interference problems.

2.0 General description

2.1 Bus and protocol definitions

The SPA bus uses an asynchronous serial communications protocol (1 start bit, 7 data bits + even parity, 1 stop bit) with data transfer rate of 9600 b/s. (In some cases a rate of 300, 1200, 2400 or 4800 bit/s can also be used). Messages on the bus consist of ASCII characters.

The bus can support one master and several slaves.

The basic construction of the protocol assumes that the slave
has no self-initiated need to talk to the master but the master is aware of the data contained in the slaves and, consequently, can request required data. In addition, the master can send data to the slave. Requesting by the master can be performed either by sequenced polling (e.g. for event information) or only on demand.

The master requests slave information using request messages and sends information to the slave in write messages. Furthermore, the master can send all slaves in common a broadcast message containing time or other data.

The inactive state of bus transmit and receive lines is a logical "1".

2.2 General characteristics of slaves

The bus supports the attached slave devices, which further have several inputs and outputs. The input information is supervised by assigning limit values and delays for the input signals. The slave incorporates a realtime clock for marking the recorded events.

A slave with several, almost independently operating entities can be divided into channels. Typically, each channel is assigned only one input and a few setting values. The assigned channels are numbered starting from 1. Common information from all channels of a slave is combined to channel 0.

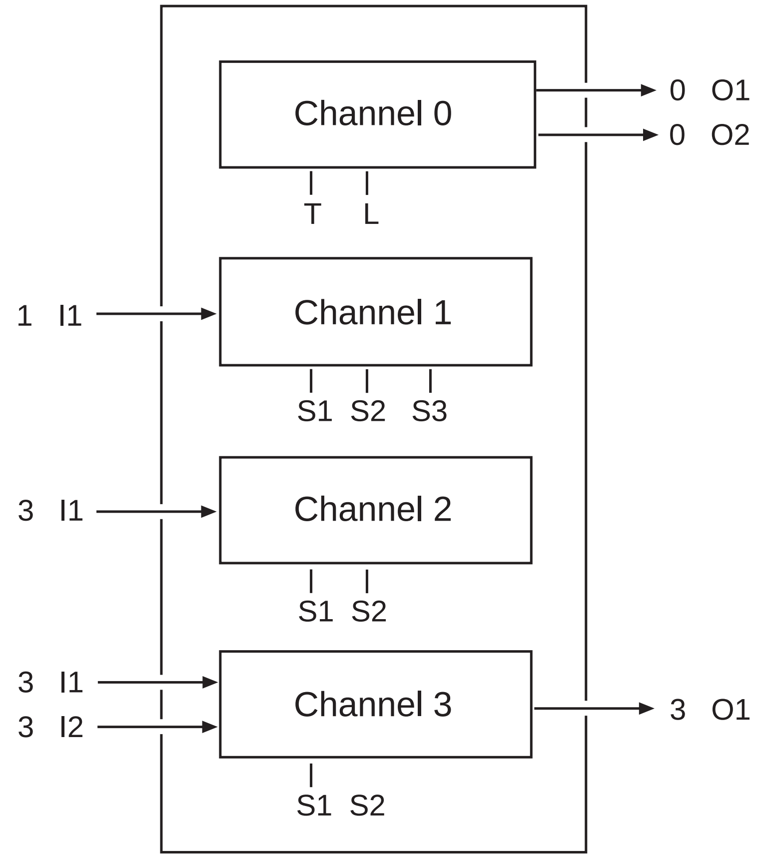

Figure 2.1 Logical configuration of a 3-channel slave.

Shown in Fig. 2.1 is a 3-channel slave in the logical form as "seen" by the protocol after assigning the inputs (I1, I2, etc.) and outputs (O1, O2, etc.) to the channels formed.

The depicted slave has four inputs, of which one is assigned to channel 1, one to channel 2, and two to channel 3. Of the three outputs, two are common to all channels, and consequently, assigned to channel 0 while one of the outputs is controlled by channel 3. Channel 1 is assigned three setting values (S1, S2, and S3) while channels 2 and 3 are assigned two setting values each. Channel 0 also is assigned event information L and time T.

3.0 Message formats

3.1 General

The messages only include printable ASCII characters (0AH, 0DH, 20H, ... , 7EH). Messages sent by the master are started with a begin bracket ">", and the slave messages start with characters "lf<". The master messages end with character "cr" (0DH) while the slave messages end with characters "crlf" (0DH and 0AH).

The slave can use character "&" to indicate continuation of the message on the next line. The maximum total length of the message is limited to 255 characters. No restrictions are set to the number of lines as far as the message maximum length is not exceeded.

Master messages are always in a single-line format.

The master message has the following format:

>nTe/eXm/m:xxxx/xxxx/xxx/xx:CCcr

- cr: Stop character
- CC: Checksum
- xxxx/xxxx/xxx/xx: Data part
- nTe/eXm/m: Header
- >: Start character

The slave message format is as follows:

lf<nT:xxxx/xxxx/xxx:CCcrlf

- crlf: Stop characters
- CC: Checksum
- xxxx/xxxx/xxx: Data part
- nT: Header
- lf<: Start characters

or

lf<nT:xxxx/xx&crlfxx/xxx:CCcrlf

- &: Continuation character

where
n = slave number
T = message type code
e = channel number
e/e = first/last channel
x = data category code (data type)
m = data number
m/m = first/last data

The delimiter character ":" separates the message header from the data part and the data part from the checksum, respectively.

Example 1
Setting 1 of channel 1 of slave 2 is requested by the master using message:
>2R1S1:XXcr

The slave replies the request by message:
lf<2D:10.1:XXcrlf

Example 2
Settings 1 and 2 of channel 1 of slave 2 are requested by the master using message:
>2R1S1/2:XXcr

The slave replies the request by message:
lf<2D:10.1/95:XXcrlf

Example 3
All settings of channel 0 of slave 2 are requested by the master using message:
>2RS:XXcr

Assuming, e.g. 3 settings to be included in channel 0, the slave replies the request by message:
lf<2D:11/3/234.88:XXcrlf

Note:
When the slave sends a message using more than one line, the end characters lf and cr are always sent at the end of the line, not as a start character for the next line.

3.2 Start and stop characters

The master messages are started with character ">" and ended with character "cr".

The slave message starts with characters "lf<" and end with characters "crlf". On all other message lines except the last line, these characters are preceded with character "&". On the last line, these characters are preceded with the checksum and character ":".

Note:
The master software can be implemented so that the receiving is triggered to slave reply message's character "<" and the first lf is ignored. This is possible because lf-characters are not calculated to the checksum.

3.3 Header content

The numbers are given in decimal format. The slave number and channel numbers may have max. three digits and the data numbers may have max. six digits. A single-letter code is used for designating the message type and the information category, respectively.

The slave number is separated from the channel numbers by the code letter of the message type. The channel numbers are separated from the data digits by the code letter of data category.

Slave number
The slave number is given in decimal format as 1...999.

Number 0 is not available. Number 900 is reserved as the broadcast address. This number is usually used to broadcast the clock message but can also be used for some other purposes.

The SPA-bus messages always carry the slave number.

Message type
ASCII character R, W, D, A or N indicating the type of the message (read, write, data, ack or nack).

Channel number
The channel number is given in decimal format as 0...999.

Only one channel number at a time can be issued if information associated with a specified channel is requested. Information associated with a string of several channels can be requested by issuing the numbers of the first and last desired channel in the request message.

When common information associated with all channels, that is by addressing channel 0, is requested, the channel number can be omitted. Equally, when the addressed slave has no other channels than channel 0, the number may be omitted.

If several channels are included in the slave, the specific channels are numbered starting from 1. Channel 0 is again reserved for requests concerning information common to all channels of the slave.

Data category
ASCII character I, O, S, V, M, C, F, T, D, L, B or A defining the logical data category (refer to chapter 5.1).

Data number
The data number is given in decimal format as 1...999999.

Data items in each data category of the slave are numbered. A string of several data items can be addressed by issuing the number of first and last item of the string. If the last number is omitted, the request is assumed by default to address all data of the data category. (Excluded are such data items which are numbered by specific numbers not forming a string of items. Thus, if the slave has data items S1, S2, S3, S4, and spefically numbered items S10 and S40, request >nnnRS:CCcr is replied with data S1...S4.)

3.4 Data part contents (also refer to Chapter 5)

The data part includes slave data items separated by character "/". If data is available from several channels, data from lowest numbered channel is sent first, then followed by the next channels in numbering order. The last data is followed by character ":".

Data transferred may be numerical or text data.

Some messages may be void of the data part. In this kind of messages, the checksum is given just next to the colon ":" after the header.

The data part of the message may also be empty in which case the colon after the data part is just next to the colon starting the data part ("::").

Note:
The data item may not contain SPA-bus special characters: cr (0DH), lf (0AH), ">", "<", ":", "/" or "&".

3.5 Checksum and parity bit

The correctness of messages is verified by including in the message a checksum and parity bits.

The checksum is sent as two ASCII characters which correspond to the hexadecimal digits obtained by XORing the bytes of the message. Excluded from the checksum count are the checksum itself and any of characters "cr", "lf", and "&".

The message sender may also omit checksum counting in which case letters "XX" are sent as checksum.

When counting the bytes, each byte is complemented with an 8th bit which is the parity bit. The parity bit is set so as to make the byte parity count even (that is, a byte with an even number of 1's). The byte checksum is counted before setting the parity bit. Correspondingly, during byte reception, the parity is checked and the parity bit is cleared.

4.0 Message types and use

4.1 Overview of master and slave messages

Master to slave message types:

- R (Read)	Data read from slave.
- W (Write)	Direct data write to slave.

Slave to master message types:

- D (Data)	Data message.
- N (Nack)	Negative acknowledgment.
- A (Acknowledge)	Message acknowledgment.

4.2 Communications between master and slaves

The master initiates the communications by sending a message to the bus. The slave, which recognizes its own slave number in master's message, responds by sending an appropriate reply message. Slave does not respond if it did not receive message start character (>), its own slave number and message end character (cr).

In case of parity or checksum error, slave either does not respond at all or responds with message of type N with error code 0. If the slave does not respond at all or responds with a message of type N with error code 0, the master resends its message until the slave responds with a desired reply message or the preset resend count of master message sendings is exhausted.

Messages sent by the slave are not replied with an acknowledgment by the master. If the master detects an error in the message sent by the slave, the master resends its message.

The communications are basically divided into two categories:
- data read from slave
- data write to slave.

Data read from the slave can use only one possible procedure.

Data write to the slave is possible using one of the two alternative procedures:
- direct data write to slave
- simultaneous data write to all slaves using the broadcast address

4.2.1 Data read from slave

Data read from slave

	Master	Slave
1. Master sends a read message	R ———>	
2. Slave replies with a data message	<———	D

Example:
When the master reads all setting 3 of channel 1 of slave 14, the following messages are sent:
Master requests: >14R1S3:XXcr
Slave replies: lf<14D:10.1:XXcrlf

4.2.2 Data write to slave

Direct data write to slave

	Master	Slave
1. Master sends data to slave	W ———>	
2. Slave acknowledges data	<———	A

Example:
Master desires to set output 5 of slave 14 by sending the message: >14W05:1:XXcr
Slave sets the output and acknowledges: lf<14A:XXcrlf

Simultaneous write to all slaves using the broadcast address

	Master	Slave
1. Master sends data to all slaves	W ———>	

(No acknowledgment is sent by the slaves.)

Example:
Master sends clock message: >900WT:37.271:XXcr
Master sends data to all slaves: >900WV251:1:XXcr

Note:
The use of broadcast messages (for other purposes than sending clock message) should be avoided, because the meaning of variables (V), settings (S) and outputs (O) is different in slaves of different type !

4.3 Contents of different types of messages

4.3.1 Message type R (Read)

A type R message contains no data part. The message has the following format:
>nRe/eXm/m:CCcr

4.3.2 Message type D (Data)

A type D message includes the data part but the header only contains the slave number and message type code. The message has the following format:
lf<nD:xxxx/xxxx/xx:CCcrlf

An empty data message has the following format:
lf<nD::CCcrlf

Generally, the master strives to request the slave information in so small pieces of information as to keep the slave reply message length under 255 bytes. If the requested data can't be included in one reply message, the slave should give message of type N, with error code 3.

4.3.3 Message type W (Write)

The message format is complete:
>nWe/eXm/m:xxxx/xxxx/xx:CCcr

4.3.4 Message type A (Ack)

The message format lacks the data part and contains in the header only the slave number and message type:
lf<nA:CCcrlf

4.3.5 Message type N (Nack)

A type N message has the following format:
lf<nN:x:CCcrlf

The data part of the message contains only a single digit, which indicates the cause of negative acknowledgment (error code).

The error codes in message type N are:
0 = error in checksum or parity

1 = slave busy
(The slave may reply with this kind of a type N message to a master's message of type R when a long reply message should be formed and the slave is currently busy with another task.)

2 = overflow of slave input buffer

3 = message from master too complicated for the slave (The slave may reply with this kind of a type N message when its communications program is intentionally simplified. Possible methods of simplification are, for instance, writing the communications program of the slave to recognize only such write messages which comprise only a single channel number and a single data item number)

4 = reserved for use in higher levels of the communication network

5 = syntax error
(Incorrect or unrecognized message type, unrecognized data type, error in channel number or data item numbers, syntax error in data part of message.)

6 = slave does not contain all data requested in the message
(The slave does not, for instance, include all addressed channels or data items, or some data addressed by a number is not available.)

7 = addressed data is impossible to write or read
(due to a permanent or temporary blockade)

A type N message with error code 7 can be issued as a reply to an data write message if the addressed information cannot be assigned a new value or is nonexistent.

A type N message with error code 7 can be issued as a reply to an data read message if the addressed data is existing and can (in general) be assigned a new value but it cannot be read.

8 = data in write message not validated

9 = undefined negative acknowledgment
(e.g., internal error in communications program)

4.4 Clock message

Second clock
To synchronize the slaves' realtime clocks to the master's main clock, requires the master to send the correct time to the common send address of the slaves using the type W message. The clock message is usually send with approximately 1 second intervals.

Clock message format:

>900WT:xx.xxx:CCcr

Time
tens of seconds, seconds,
1/10ths, 1/100ths, and 1/1000ths of seconds

In order to compensate for the delay in communications, the master does not send the time valid at the instant of sending but the time valid at the instant of receipt of the last character of the clock message by the slaves.

Date and time
To synchronize the slaves' full date and time to the master's main clock the master may broadcast message of format:

>900WD:yy-mo-dd hh.mm;ss.sss:CCcr

This message is usually sent with an interval of 10...60 s.

5.0 Data transfer in messages

5.1 Data categories

Data transferred by each slave is categorized into one of the following logical data categories:

Data category	Code letter (in message format)
Input data	I
Output data	O
Setting values	S
Variables (internal)	V
Memory data	M
Slave status (conditions)	C
Slave identification	F
Time	T
Date and time	D
Last events	L
Last events from backup buffer (repeated requesting)	B
Alarms valid	A

All slaves do not provide data of all categories since a wide latitude exists in data available from different kinds of slaves.

The slaves provide data of categories C, F, T, D, L, B, and A only on channel 0. Data of categories I, O, S, V, and M is available on both channel 0 and other channels.

5.2 Data belonging to different data categories

5.2.1 Data categories I, O, S, V, and M

Input data I:
Includes slave analog input values and digital input states.

Output data O:
Includes slave analog output values and digital output states.

Setting values S:
The setting values generally include such data, which sets the parameters of the slave to be compatible with the current operating situation.

Internal variables V:
The internal variables include such data which:
- gives complementing data on the supervised process or its events
- gives complementing data on the slave or its functions
- controls the functions of the slave in an operating situation
- programs some general functions of the slave.

Memory stored data M:
Includes memory stored measurement or status data. This kind of memory stored data is typically available only from datalogger type slaves.

Furthermore, data of category M can be used for transferring different kinds of complementary data, such as dimensional units and verbose parts of transferred data or programming data.

Data presentation formats of categories I, O, S and V:
Data intended for normal use (data number 1...100) is given in decimal numbers and transferred using ASCII characters.

Data intended for special applications (data number 101..999999) can also be formatted in some other manner. In these cases, the presentation format is given in the documentation of the product.

The numeric values may include a decimal point and sign (+ or -). The length of the integer part is 1...10 digits and the decimal part has a length of 0...10 digits.

General rules for sending data to the slaves:
- always at least one digit in the integer part
- remove unnecessary 0-digits from the integer part
- remove unnecessary 0-digits from the decimal part
- remove decimal point if it is not needed
- if there is a decimal point, it should never be the first or last character of the number
- the master unit should avoid sending data using unnecessary precision i.e. too many digits in the decimal part

Example:
The master unit should NOT send numbers like: 00120, 00.5, .6, 7., 8.00
The master unit should send the numbers in format: 120, 0.5, 0.6, 7, 8

Note:
The SPA-bus slave unit should be able to round and truncate data values it receives from the master unit. If the slave is anyhow not able to handle the received data value, it should discard the value and reply with N-message indicating error 8.

Note:
If the number of leading zeroes has some importance for the slave (e.g. the number of leading zeroes expresses the accuracy of a setting) it must be documented in the product documentation of the slave. To handle this kind of situation the master unit should be able to send the required number of leading zeroes.

Data presentation format of category M:
Data in category M is usually considered to be in string format, i.e. data may contain numbers and alphabets or other ASCII characters (20H...7EH) except SPA-bus special characters.

5.2.2 Data categories F, T, D, and C

Slave identification data F:
The slave type is identified by an identification code (max. length 10 characters). The ID code can be, for instance, the product code of the slave.

Time T:
Time is sent in the messages using the following format:

xx.xxx

- xxx — 1/10ths, 1/100ths, and 1/1000ths of second
- xx — Tens of seconds and seconds

Date and time D:
Date and time is sent in the messages using the following format:

yy-mo-dd hh.mm;ss.sss

- ss.sss — seconds and milliseconds
- mm — minutes
- hh — hours
- dd — day of month
- mo — month
- yy — year (two last digits)

Slave status C:
The slave status can be represented using two bits. In the messages, the status gets values 0, 1, 2 or 3.

Status bit 0: Slave reset or other situation, which may have caused loss of event data.

When this bit is set (C=1 or C=3), the slave always sends in conjunction with event requests also event xx.xxx E50 until the bit is cleared. The master must clear this bit by issuing message >nWC:0:CCcr.

Status bit 1: Indicates slave event buffer overflow.

When this bit is set (C=2 or C=3), the slave always sends in conjunction with event requests also event xx.xxx E51 until the bit is cleared. The master must clear this bit by issuing message >nWC:0:CCcr. Due to an overflow, the slave usually stops entering new events into the buffer, and the bit must be cleared to restart normal storage of events into the buffer.

5.2.3 Data categories L and B

Last events L:
Data category L includes events entered into the event buffer after the last request.

The slave sends the recent events starting from the oldest event. If all recent events do not fit into one message, rest of recent events will not be sent until during the next request.

Last events from backup buffer (repeated requesting) B:
Data category B includes same data as data category L.

Generally, the master requests the slave for latest events by addressing data category L. However, if the reply message of the slave contains an error, the master cannot re-request data category L but instead it must request data category B (read the following pages).

Event markings presentation format:
An event marking includes the following items:
<time><space><channel number><letter E><event number>

Where:
<time> = Tens of seconds, seconds, and 1/10ths, 1/100ths, and 1/1000ths of seconds. A period (.) is the delimiter between seconds and decimals of seconds.
<space> = Delimiter character.
<channel number> = Channel number associated with the event. The length of the channel number is 0...3 digits. If channel number is 0, it can be omitted.
<letter E> = Delimiter character
<event number> = The length of event number is 1 or 2 digits. The number can have value 0...63.

Example:
If the channels of analog alarm unit 23 are assigned two limit values, the events of channel 5 are coded, for instance, as follows:
Code 5E1 equals channel 5 signal below limit 1
Code 5E2 equals channel 5 signal above limit 1
Code 5E3 equals channel 5 signal below limit 2
Code 5E4 equals channel 5 signal above limit 2

For example, an event of channel 5 at time 10.236 with event code E2 and another event of the same channel at 11.555 with event code E4 would be coded in a type D message by the slave as:
lf<23D:10.236 5E2/11.555 5E4:CCcrlf

When events are requested from the slave and the event buffer of the slave is empty, the slave responds with an empty data message.

Example: lf<23D::CCcrlf

Analogue events:
It is also possible to send analogue data value together with an event.

In this case the event marking has the following format:
<time><space><ch.numb.><letter Q><event numb.><space><value>
<value> = Analog value following the normal rules of SPA-bus analog data
(see data presentation format of data categories I, O, S and V)

Example: 10.234 5Q6 5.05

The use of this event type is selected by a parameter (V-type). The parameter is default set to zero, which makes it possible to use the unit on a bus with a master that can not handle "analogue events".

The meaning of the analogue value depends on the device type. The value can be e.g. a measurement value, a fault current or an index number of the fault data record belonging to the event.

Numbering of events:
Events are numbered with numbers as 0...63. The interpretation of the event number is dependent on the slave type.

Event numbers E50...E54 of channel 0 are reserved for special purpose:
E50: Slave reset or hardware or program fault encountered in slave. As a result, event loss may have occurred.
E51: Overflow of slave event buffer.
E52: Temporary malfunction in the slave connection.
E53: No connection to the slave.
E54: Connection with the slave re-established.

Note:
Events E50 and E51 are reported in conjunction with event requests until the status bits corresponding to these event types are cleared. (Refer to data category C.)

Note:
Events E52, E53 and E54 are generated by the master unit of the bus.

5.2.4 Data category A

When the slave is requested for data category A, the slave responds by sending the event markings of the valid alarms without time information. The slave examines all possible causes that may have generated the alarm or alarms, and forms the markings of valid alarms on the basis of the examined information. Thus, the event markings are not retrieved from the slave's event buffer since some vital information may not anymore be included in the buffer contents.

Consequently, e.g. the events described on page 24, if still in an active state, would appear in a data message sent as a reply to a request of category A data as follows:

lf<23D:5E2/5E4:CCcrlf

Active alarms may be requested from all channels using message of format:
>nRA:CCcr (n=slave number)
or from one channel only using message of format:
>nRcA:CCcr, (n=slave number, c=channel number)

When the slave is requested for valid alarms, and no alarms are active, the slave responds with an empty data message of format:
lf<nD::CCcrlf (n=slave number)

Data category A can be used for example to print out an active alarm list from the master or to update the active alarm list of the master unit. This data category is especially useful with alarm annunciator units. Other type of slave units may or may not provide this information. If this information is used by the master depends on the application and the type of the master unit.

6.0 Delay durations

The maximum allowable delays and recommended maximum delays in communications are limited by the protocol as follows.

Time allowed for the slave to prepare a reply message after receiving a message from the master (reply timeout):
- absolute maximum time 50 ms
- recommended max. time 5 ms

Time interval between master's clock messages:
- absolute maximum time 10 s
- recommended max. time 1 s

Time allowed for a pause in message sending is 50ms, but it should be avoided. This allows e.g. the slave to send a reply message divided into several lines using 50 ms for preparing each line.

Appendix 1

Reserved SPA-bus data items:

All slave types:

Some items of data in data category V of channel 0 are reserved for special applications:

Data item	Description
V100	Alarm reset (when master writes value = 1)
V150	Flag for activating/deactivating the secondary settings of a protective relay (activate = 1, deactivate = 0)
V151	Store issued setting values into nonvolatile memory.
V200	Slave number
V201	Data transfer rate of slave (coded or true baudrate value)
V205	Program version number (string type)
V251	Perform/cancel preselected open/close operation (perform = 1, cancel = 0)
V252	Reset all mean value counting and minute counters
V253	Reset all pulse counters

Note: V251, V252 and V253 are sent using broadcast address.

Disturbance recorders:
The data stored by a disturbance recorder can be read by using data items V16, V20, M28, M29, M30 and M31:

Data item	Description
V20	Reset the block and line identifiers to the beginning of the oldest recording (write V20=1)
M28	Description of the oldest recording.
M29	Identifier of recording block.
M30	Identifier of recording line.
M31	Data item used to read one recording line.
V16	Erase oldest recording (write V16=0 and V16=1)

The disturbance recording is divided into blocks and each block in divided into lines for reading. One line includes one piece of the recorded data. The contents and format of the line depends on the type of the recorder.

A recording line pointed by identifiers M29 and M30 is read by issuing a read command to data item M31 (>nRM31:CCcr). After reading the identifiers are automatically updated to point to the next line of the recording. One recording can thus be read by issuing consecutive read commands to data item M31. A line can also be read by explicitly setting values for M29 and M30 before issuing the read command. The number of needed read commands can be calculated based on the information given by data item M28. (The format of M28 and M31 depend on the type of the recorder device.)

When the oldest recording has been successfully read, it can be erased by issuing two write commands to data item V16 (>nWV16:0:CCcr and >nWV16:1:XXcr). After this the block and the line identifier can be set to the beginning of the next oldest recording by issuing write command to data item V20 (>nWV20:1:CCcr).

Appendix 2

SPA-bus hardware recommendations:

A product including SPA-bus slave unit(s) or SPA-bus master unit is connected to the physical SPA-bus using a separate interface module called Bus Connection Module. This enables the use of different types of physical media e.g. glass or plastic fiber optic cable, RS-485 bus or current loop. The media is selected according to the requirements of the application (fiber optics is preferred).

Interface to the Bus Connection Module:
The interface between the Bus Connection Module and the product usually is:
- 9-pin female D-connector in the product
- signal levels: RS-485 or TTL (RS-485 is preferred)

(Also 25-pin D-connector and RS-232-C can be used, if suitable Bus Connection Module is available.)

The pin numbers of the SPA-bus/RS-485 connector are:

Pin	Usage
1	DATA A, data signal pair, signal A (+)
2	DATA B, data signal pair, signal B (-)
3	RTS A, request to send signal pair, signal A (+)
4	RTS B, request to send signal pair, signal B (-)
7	GND, signal ground for the power supply for the Bus Connection Module
8	+5V, power supply for the Bus Connection Module
9	+8V, power supply for the Bus Connection Module

Note: Usually only one of the power supply voltage alternatives is provided by the product.

The pin numbers of the SPA-bus/TTL connector are as follows:

Pin	Usage
2	TXD, transmit data, data from product
3	RXD, received data, data to product
7	GND, signal ground for the power supply for the Bus Connection Module
9	+8V, power supply for the Bus Connection Module

Fiber optic SPA-bus:
A fiber optic SPA-bus is built using one or more fiber optic loops.

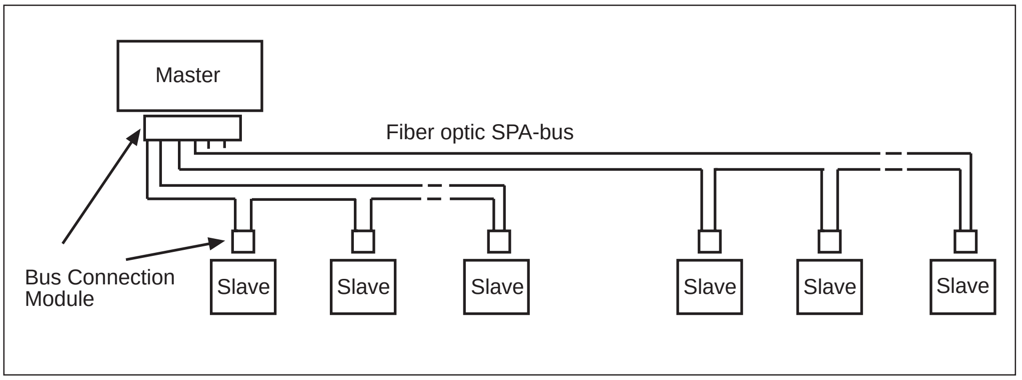

Figure 1. SPA-bus with two fiber optic loops.

The logical structure of Bus Connection Module is illustrated in the following figure. Logical "1"="dark" and "0"="light".

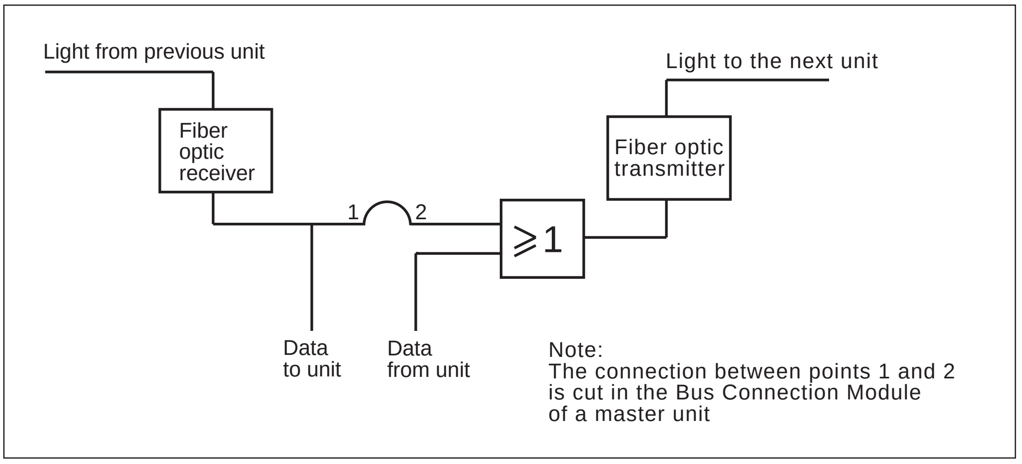

Figure 2. Logical structure of a Bus Connection Module.

Recommendation for plastic fiber connections:

Cable connector:	snap-in connector
Cable diameter:	1 m
Max. cable length:	30 m
Wavelength:	660 nm
Transmitted power (typ.):	-13 dBm
Receiver sensitivity:	-20 dBm

Recommendations for glass fiber connections:

Cable connector:	ST-connector
Cable diameter:	62.5/125 um
Max. cable length:	2000 m
Wavelength:	820-900 nm
Transmitted power (typ.):	-13 dBm
Receiver sensitivity:	-24 dBm

Appendix 3

Implementation limitations

In some master devices, which transfer the SPA-bus events to a higher level system using ANSI X3.28 protocol, the maximum number of channels per slave is limited to 127.

In some master devices the maximum data item number for such data items, which are defined to the data base of the master device, is 255.

In some slave devices the biggest available slave number is 99 or 254.

In a master unit of the SPA-bus it should be possible to set the reply timeout (e.g. 50...500 ms) and line idle time (e.g. 0...10 ms). The settable reply timeout is sometimes needed to be able to communicate with some slowly responding slave units. Line idle time is the delay time between the last character of the slave reply message and the start of the masters next message, this gives more time for the slave software and hardware to turn around from transmitting to receiving.

Appendix 4

Extensions

String transmission with extended character set:

SPA messages should only contain the printable ASCII characters with the hexadecimal codes 0A, 0D and 20 to 7E. In all this makes 98 characters out of which 11 characters are reserved for SPA use. However, personal computers can handle 256 characters.

In order to allow transmission of any of the 256 characters it is suggested that the hexadecimal ASCII code of the character is transmitted in two characters preceded by the '$' character as a "shift" character. The following examples serve as an illustration:

Original character		Transmitted character
3E	>	$3E
3C	<	$3C
3A	:	$3A
0D	carriage return	$0D

The suggested encoding method has been used with the generator protection REG 100. Here, exactly the same text strings are transferred to the personal computer as the ones presented on the terminal unit on the protection. In this way version conflicts are avoided between the software products in the protection and the personal computer.

Reserved characters in SPA-bus messages:

SPA messages should only contain the printable ASCII characters with the hexadecimal codes 0A, 0D and 20 to 7E. In all this makes 98 characters out of which 11 characters are reserved for SPA use. The reserved characters are used for transmission control, message delimiting and special purposes. The following characters are presently reserved:

Code hex	Reserved character	Comment
0A	line feed	Slave message start/stop character
0D	carriage return	Master/slave message stop character
21	!	For future use
22	"	For future use
24	$	ASCII code shift character
26	&	Continuation character
2F	/	Data separation character
3A	:	'Data part' delimiting character
3C	<	Slave message start character
3E	>	Master message start character
3F	?	For future use

Appendix 5

SPAFTR (FileTransfer for configuration files)

MASTER->SLAVE (write configuration to slave)

1) Master initiates the file transfer by sending the following message:
   >ssW0M200:ID/count:CCcr

where:
ss = slave number
ID = file identification
count = number of blocks, each block is 100 characters in ASCII-format
(the last block can be shorter than 100 chars.)
CC = checksum

In error situation the following NACK-messages are specified:
- device not ready for FTR (e.g. another FTR is going on) N1
- file with specified ID not available N6
- device in use/do not accept FTR in this mode (e.g. password not set) N7

Parity errors etc. are handled as usually.

2) After initiation master sends data blocks to the slave
 >ssW0M201:N/datablock:CCcr

where:
N = block number
- After initiation master has to start sending of the blocks within 10 s.
- If slave has not received the next block 10 s after it sent the ACK to the previous datablock, it will stop receiving the file.
- The minimum delay between two messages is 500 ms.
- The slave unit must take care of the block numbers so, that if block is received with wrong block number it will be ignored.
- When configuration is written to the slave, master has to always write the whole configuration file (safety reasons). Thus the first block number must always be one (1).

In error situation the following NACK-messages are specified:

- block coming too soon (busy)	N1
- extra block (more blocks sent that specified)	N5
- slave has canceled FTR (e.g. delay after previous msg > 10 s)	N7
- invalid block number	N8
- wrong data length	N8

After N1 and N8 master is allowed to resend the message. N5 and N7 tells that something has gone badly wrong, master has to start FTR from beginning.

Parity errors etc. are handled as usually.

3) After the transfer the master sends the 'end message'
 >ssW0M202:ID:CCcr

- The FTR end message must be sent within 10 s after the ACK of the last datablock.
- This message works as a confirmation message for the end of FTR.

In error situation the following NACK-messages are specified:

- block coming too soon (busy)	N1
- Wrong file ID	N5
- all blocks not received yet	N6
- slave has canceled FTR (e.g. delay after previous msg > 10 s)	N7

Parity errors etc. are handled as usually.

After N1 and N8 master is allowed to resend the message. N5 and N7 tells that something has gone badly wrong, master has to start FTR from beginning.
- The new configuration is stored to the nonvolatile memory using reserved SPA command V151.
- The configuration will be taken in use after RESET.

SLAVE->MASTER (read configuration from the slave)

1) Master sends the initiation message to the slave >ssW1M200:ID/N:CCcr
where:
ID = file identification
N = first block to read
In error situation the following NACK-messages are specified:
- device busy (e.g. another FTR going on) N1
- file with requested ID not available N6
- first block requested not available N6

Parity errors etc. are handled as usually.

2) Master reads the number of the blocks >ssR1M200:CCcr and slave answers with number of the datablocks in file, from first to read until the last in the file.

In error situation the following NACK-messages are specified:
- block coming too soon (busy) N1
- slave has canceled FTR (e.g. delay after previous msg > 10 s) N7

Parity errors etc. are handled as usually.

After receiving N1 master is allowed to do retransmission of the message.

3) Master reads the data blocks >ssR1M201:CCcr and slave answers with data block <ssD:N/datablock:CCcr

In error situation the following NACK-messages are specified:
- block coming too soon (busy) N1
- block requested does not exist (end of file reached) N6
- slave has canceled FTR (e.g. delay after previous msg > 10 s) N7

Parity errors etc. are handled as usually.

After receiving N1 master is allowed to do retransmission of the message.
- If master fails in receiving of the block it will send a new initiation message.
- New read of a datablock can be sent 500 ms after receiving the previous block.
- If the next datablock is not read within 10 s after the last datablock read, slave stops the file transfer and master has to initiate reading again. Channel number 0 is used for writing a configuration and channel number 1 is used for reading of the configuration.

The datablock number in the message, as well as end of FTR message, is an extra checking for the transmission. If slave/master receives the datablock with wrong number, it will notice that for some reason one block is missing.
Reading can be started/stopped from/to any block in a file without any errors. Writing is allowed only from first block to the last block. It is not possible write just few blocks of file.
It is allowed to split message in several parts using continuation character at the end of the data part.
The maximum size of the block is limited to the 100 characters, because long messages will block the bus for too long time. For the same reason there is also the delay between two datablock read/write.

ABB Substation Automation Oy
P.O.Box 699
FIN-65101 VAASA
Finland
Tel. +358 (0)10 22 4000
Fax.+358 (0)10 22 41094
www.abb.com/substationautomation

1MRS 750076-MTD EN