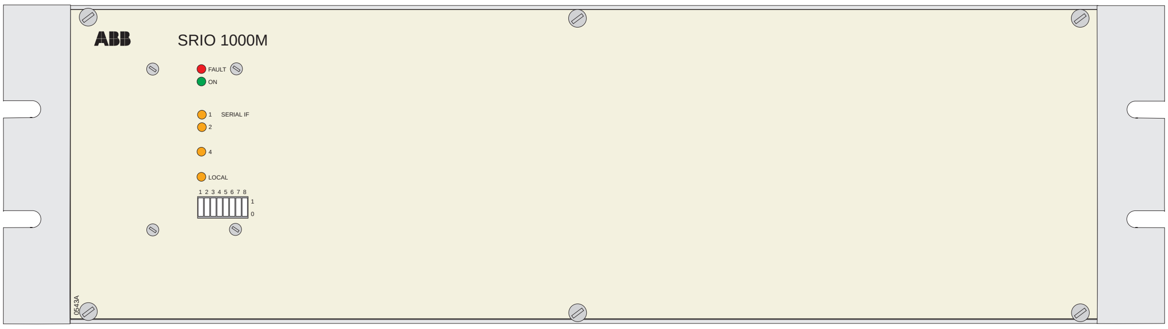
SRIO 500M, SRIO 1000M
Data communication and reporting unit

Programming manual



Issued 97-11-17
Version A (replaces 34 SRIO 1000M 2 EN1)
Checked SK
Approved OV

Data subject to change without notice

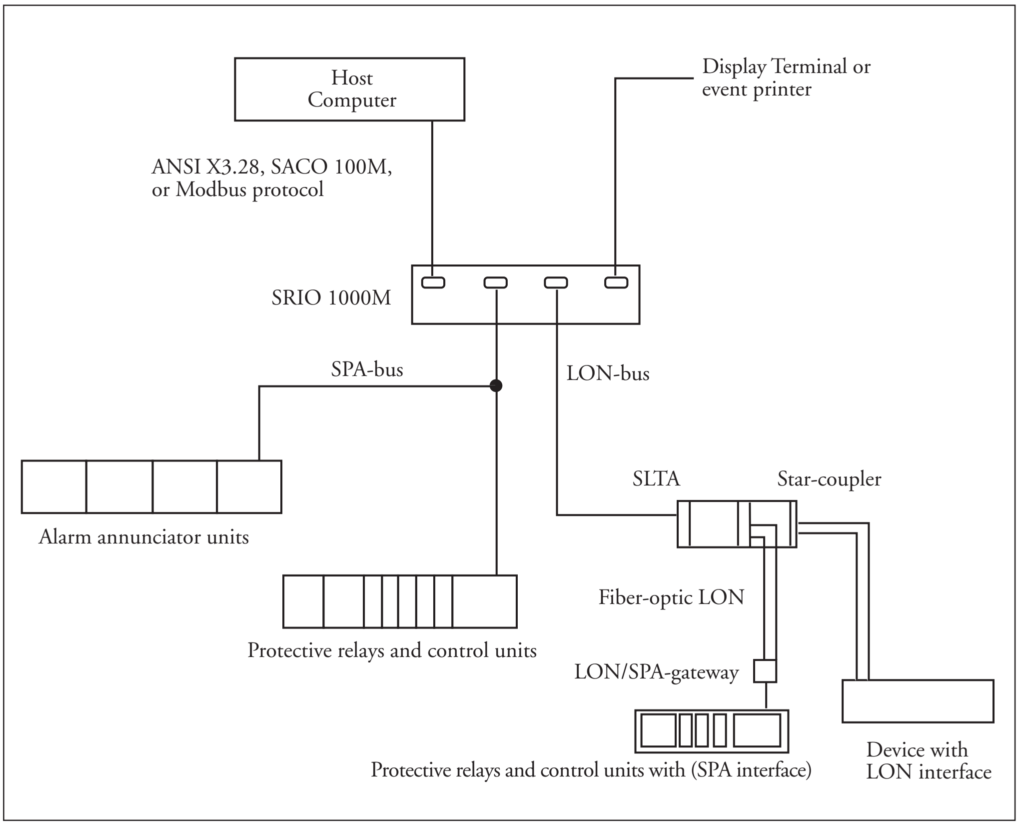
Table of Contents

Introduction

The SRIO 1000M is a data communicating and reporting unit for the substation protection and control systems. This system may incorporate slave devices such as protective relays, control units and annunciator units, which are able to communicate via the SPA-bus or LON-bus.

The task of the SRIO 1000M is to be the master unit of the SPA-bus and for LON-bus to act as network manager. SRIO connects the SPACOM system to a host computer and makes event reporting to an event printer. When LON interface is used the connection from SPACOM system to SRIO is made using SLTA and LON/SPA gateway.

The connection to the host computer can be done using ANSI X3.28, SACO 100M or Modbus protocol. The ANSI X3.28 used with SCS 100 or SPIDER MicroSCADA. The SACO 100M protocol is used to communicate with for example a personal computer or control system of another manufacturer. The Modbus protocol is used with PLCs, control systems or industrial MMI software packages.



SRIO 1000M with SPA and LON interfaces and local event printer.

Naming conventions

SPACOM system	A distributed substation secondary equipment system. It incorporates several slave units like protective relays, control units and alarm annunciator units connected over a serial bus, SPA-bus, to a master unit. The master unit is typically one of the control data communicators: SACO 100M, SPSC 500M or SRIO 1000M.
SACO 100M	An older 8051-based data communicator, which can communicate with its host computer using SACO 100M protocol. SACO 100M is located in a SACO 148D4 rack.
SPSC 500M	An older 80188-based data communicator, which can communicate with its host computer using ANSI X3.28 or SACO 100M protocol.
SPA-bus	An ASCII character based master-slave serial bus. The bus is built using fiber-optic cables or twisted pair (RS-485).
SPA-bus unit, SPA-bus slave	A device which can communicate via the SPA-bus as a slave.
LON-bus	A name given to our way of using LonWorks Network in Substation Automation applications, referring especially to the fiber-optic physical star network structure and our own message and data formats.
Master, Master device	Device connected to the SRIO using Modbus interface.
Slave, Slave device	SPA-bus unit or device connected to SRIO 1000M as slave using SACO 100M protocol. This term can also be used from SRIO 1000M when Modbus master is connected to it.
SCS	Substation Control System
PLC	Programmable Logic Controller
Host computer	The computer connected to SRIO 1000M using ANSI X3.28 or SACO 100M protocol, SRIO 1000M acts as the slave and the host acts as the master in the communication.
Host	Host computer
SLTA	Serial LONTALK Adapter
LON/SPA gateway	LON connection module for devices including SPA interface. It polls data from SPA-bus to its local database and spontaneously sends data to LON.

Program versions

Version	Notes
V1.0...V3.C	SPSC 500M.
V4.0...V4.Z	Versions which operate both in SPSC 500M and SRIO 1000M.
V5.0...V5.Z	New feature: Local event reporting
V6.0...	New feature: Event based data acquisition
V7.0...	New feature: Two priority levels of slaves to speed up the event acquisition from control units or other high priority slaves.
V8.0...	New features in ANSI X3.28 interface: Programming commands can be sent via the parameter data buffer. Data can be transferred to the host using so called multi event format which means that more data can be packed to one message to improve response time.
V8.31...	Implementation of LON interface.
V8.5...	New feature: Data transfer from SPA-bus to LON interface.
V9.2...	Implementation of Modbus protocol.

Note: The version of the program can be checked with VERS command.

Note: The latest version available on 1.5.1997 is V9.2.

Note: The PIC command still exists in V9.2. Its necessity is under investigation.

Echelon, LON, LonBuilder, LonManager, LonTalk, Neuron and 3150 are U.S. registered trademarks of Echelon Corporation. LonLink, LONMARK, LonSupport, LonUsers, LONWORKS, LonMaker, and 3120 are trademarks of Echelon Corporation.

1.0 Functions

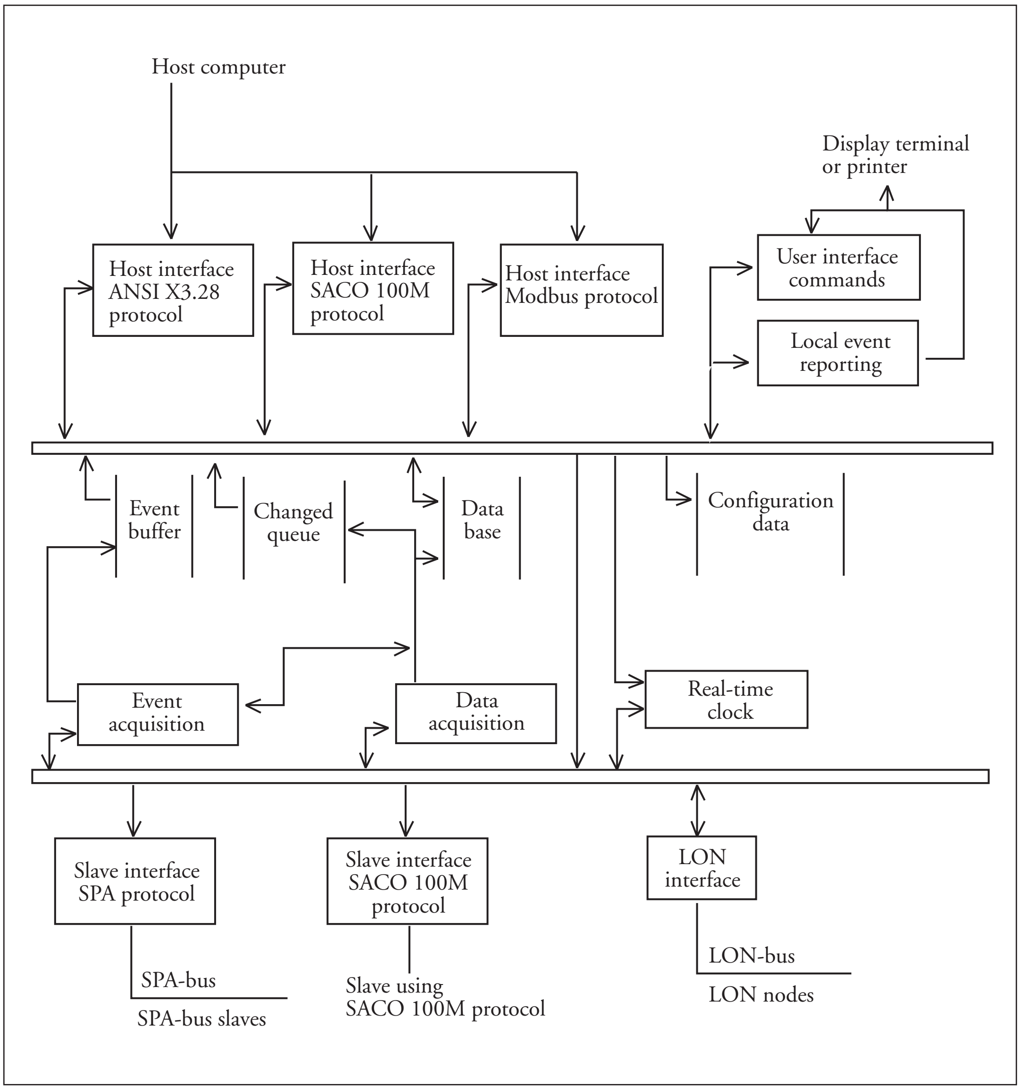


Figure 1.1. The functions of SRIO 1000M.

1.1 Event acquisition

1.1.1 Event acquisition principle, SPA-bus/SACO 100M protocol

SRIO 1000M polls events with time markings from the SPA-bus units defined by UNIT command and from slave devices connected to SRIO 1000M using SACO 100M protocol.

Using LON interface events are not polled, but instead the LON nodes spontaneously send explicit messages or network variables of type SNVT_alarm containing time stamped events. The events are sorted in time order and stored in the event buffer.

From the event buffer the events are delivered to the host computers or listed to event printers.

Time resolution between SPA-bus units connected to the same SRIO 1000M's serial interface is better than 10 ms. Time resolution between two units connected to two different SRIO 1000M's serial interfaces is better than 50 ms.

In addition to codes received from the slave devices, the data communicator reports some events associated with the slaves using the following event codes:

E52 = temporary malfunction in slave connection
E53 = no connection to slave
E54 = connection with the slave re-established

Note:
Beginning from program version V7.0 it is possible to divide the slaves into two groups: high priority and normal priority slaves. High priority slaves are polled faster than normal priority slaves. This improves response time for the data acquired from the high priority slaves (e.g. from control units).

Note:
All the SPA-bus units of the system under one SRIO 1000M must have unique slave numbers.

Note:
Modbus protocol does not support spontaneous event transmission to the host.

1.1.2 Event buffer

The event buffer is a fifo buffer with a capacity of 500 events. The events in the event buffer contain: year, month, day, hour, minute, seconds, milliseconds, unit number (slave number), channel number, event number.

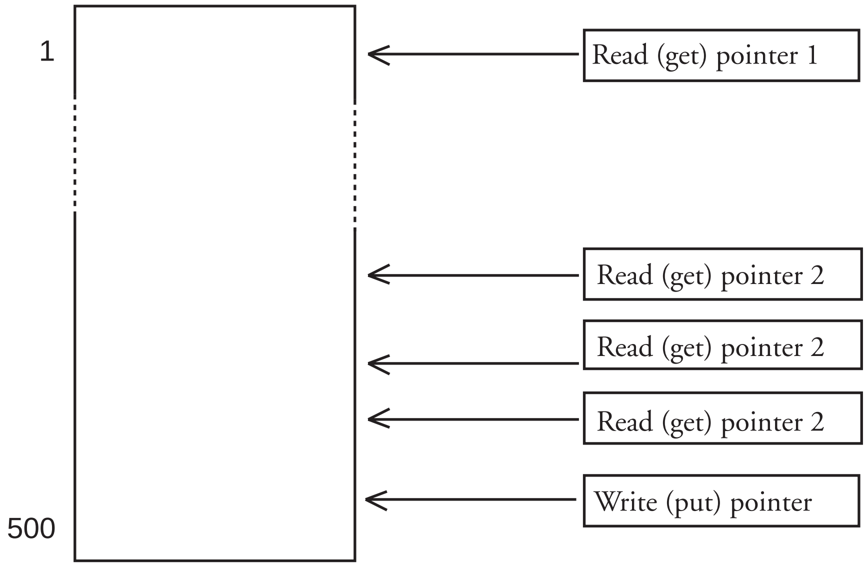


Figure 1.2. Event buffer and its pointers.

The events from the event buffer are read for different devices (e.g. local event printer and host computer) using different read pointers. If the write pointer of the buffer reaches the read pointer of a device due to communication problems with the device, an overflow occurs.

Overflow handling:

If the priority of the device is high, event polling is stopped and SRIO 1000M generates an overflow event including SRIO 1000M unit number, channel number zero and event number 51. Event polling is resumed after all the events have been read from the buffer.

If the priority of the device is low, then the read pointer is just pushed forward and new event is written on top of the old one. When the communication to the device starts operating again, it will get the latest 500 events.

The priority of the event printers is set with SET_PRINTER command. The priority of the host interface is normally high. If the connection to the host is lost for more than 10 minutes, then the priority of the interface is changed to low priority.

Note:
The overwritten description of the handling of the overflow situation of low priority devices is valid for program version V6.4 and later.

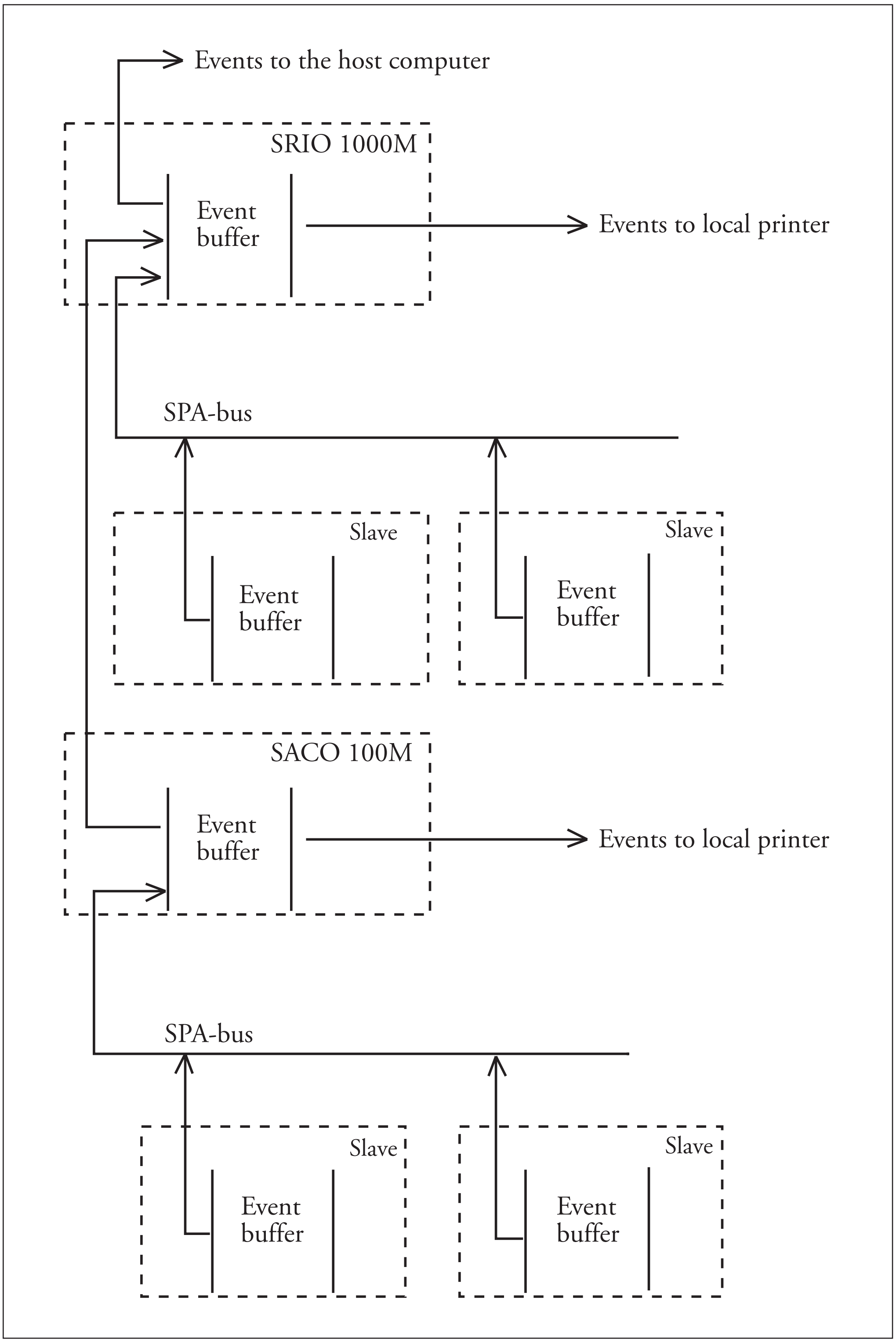


Figure 1.3. Illustration of event recording in a system which includes SRIO 1000M, SACO 100M and SPA-bus slave units.

1.2 Data acquisition

1.2.1 Database

The user can define up to 500 data items to the database. A data item can be one of the following data types:

- DI: digital input data (digital status data)
- AI: analog input data (measurement data)
- DO: digital output data (digital control data)
- AO: analog output data (analog control data)
- EV: event data (analog status data)

One DI or DO data item can hold 1...16 slave data bits from one slave unit. Digital data is stored to the database as 16 bits words. The first data bit of a slave data corresponds to bit 0 in the word.

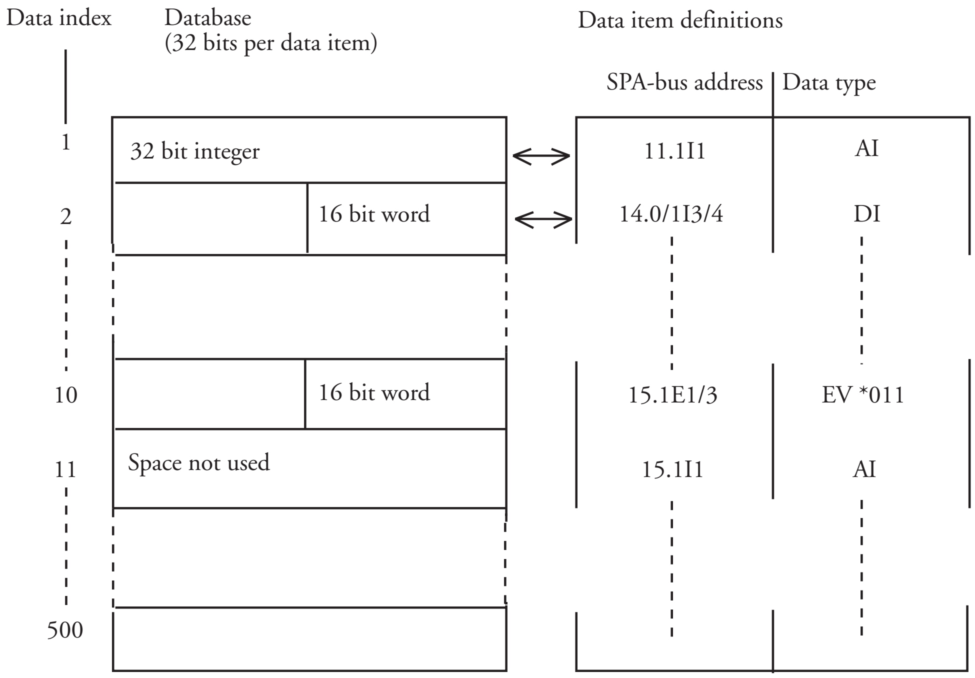
One AI or AO data item can hold one slave data. Analog data is stored to the database as 32 bit integer values. Slave data is multiplied by 1000 before storing to the database. (E.g. data value 1.65 from slave is stored as 1650.)

One EV data item can hold one analog status data value generated from slave's events. Event data is stored to the database as 16 bit integer values. Event data is also scaled by 1000. (E.g. EV data item which has value 2 is stored as 2000 in the database.)

Each data item of the database has also a time stamp in the time stamp table. The time stamp of AI or DI data is the time, when the data was polled. The time stamp of EV data is the time received with the event.

1.2.2 Data item definitions

The acquisition of data to the database is controlled with data item definitions. The definitions define e.g. the SPA-bus address and type of the data item. The definitions are created with DATA command.



Note:
The correspondence of SPA-bus message bits and DI data item bits of data item 2:

SPA-bus bit 14.0I3 = data item 2, bit 0
:   :
:   :
SPA-bus bit 14.1I4 = data item 2, bit 3

Note:
SPA-bus address contains slave unit number, channel number(s), data category identification letter (I, O, S, V,....) and data number(s).

Note:
Data item 10 is EV data and 11 is the corresponding data definition for reading the data directly from the slave in analog format.

Figure 1.4. The relation between the database and the data item definitions.

1.2.3 Data acquisition principle, SPA-bus/SACO 100M protocol

Cyclical data acquisition (polling)

SRIO 1000M acquires DI and AI data to the database by cyclical polling. A defined data item can be polled to the database continuously or periodically or the polling can be disabled.

Normally DO, AO data are not polled, the definitions are needed only to route the control message from the host computer to the slave devices. It is also possible that some DI, AI data are defined to the database but they are not polled. In this case the data item definition is only used to predefine the data for later read operation.

If the polling of a continuously polled data item fails two times, SRIO 1000M starts to poll the data item periodically. Continuous polling is started again after one successful poll.

Event based data acquisition

The EV data is acquired to the database by converting slave event codes (acquired by the event acquisition) to analog data values. The data conversion is done using data conversion tables, which are programmed by DATA_CONVERSION command.

The event base data acquisition may give wrong data values after special situations like:
- system reset
- slave reset (E50 from slave)
- communication error to the slave device (E54 from slave)

In these situations the EV data is acquired by polling the corresponding data (so called direct data) directly from the slaves. These data values are also converted using the data conversion tables before storing to the database.

After receiving E50 or E54 from one slave, it will be polled for direct data within about 1 minute. If E50 or E54 is received from more than 10 slaves, then all the EV-data items will be polled by direct polling.

In addition to special situations all the data items which correspond to the EV data items are also polled after every 15 minutes to guarantee the integrity of the database.

Auto poll

Data items can be connected together into data groups to get faster response for control operations. The group may include data items which are used for controlling an object, data items used to read data from the same object, and event data of the object. The events of the corresponding slaves are used to generate event based data of the object.

Auto poll is activated when SRIO 1000M receives a data write message, which writes data to a data item of a defined data group. Auto poll function polls data from the data items of the group and events from the slaves of the group some extra times. These extra polls are done in addition to the normal data poll and event poll sequences. The activation of a new auto poll aborts the possible previous auto poll sequence.

Changes in objects input data are detected usually in 1...2 seconds after the control operation, if the input data and control output data are in same data group.

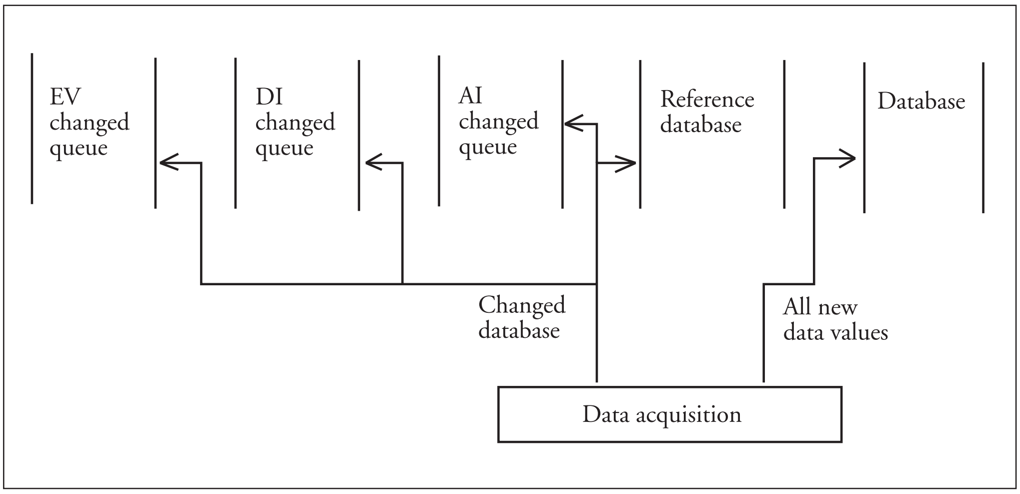
The data groups are defined using DATA_GROUP command. The maximum number of data groups is 100. The maximum number of data items or slaves in a group is 10.

The interval between the extra polls and the number of them are defined using SYSPAR command. The default values: interval 0.5 seconds, poll count 5. The possible ranges are: interval 0.5...10 seconds, count 1...10 times.

Data updating to the database, reference database and changed queue

If the data item is polled, it is always updated to the database (beginning from V6.0). If its value has changed compared to its old value, then it is also updated to a reference database and changed queue to be transferred to the host computer using spontaneous data transmission. A digital data item is updated to the reference database, if at least on bit has changed. An analog data item is updated to reference database, if the difference between the new value and the old value is greater than or equal to the user defined delta value.

Note:
If an analog data value goes under the delta value, then the data value in the reference database is reset to zero.



1.2.4 System response times

System response time

System response time is the time delay from the moment when a data changes in the slave to the moment when the change is updated to the corresponding data item of the database in SRIO 1000M.

With SRIO 1000M basic configuration:
- serial interface 1 is connected to SPIDER MicroSCADA
- all the slaves are connected to serial interface 2 with SPA-bus
- serial interface 3 is not used
- serial interface 4 is connected to a terminal (1200 b/s)
- SYSPAR P19 = 5/1

the response times can be calculated as follows:

Program version < V6.0, or event based data updating not used and SYSPAR P19= 1/5:
    response time
        = amount of updated data items * 100ms

Program version >= V6.0:
    response time for data items updated from events
        = amount of SPA-bus slaves * 50 MS
    response time for cyclically updated data items
        = amount of data items * 200 MS

Program version > V7.0:
    response time for data items updated from events of high priority slaves (usually control modules)
        = amount of high priority slaves * 70 MS
    response time for data items updated from events of normal priority slaves
        = amount of slaves * 200 MS
    response time for cyclically updated data items
        = amount of data items * 200 MS

Note:
There is always an additional delay before the change is detected in the host computer.

System response time for control operations:

The system response time for control operations is the time delay from the moment when data changes in slave after a control operation to the moment when the change is detected in SRIO 1000M.

The best response time can be reached, when the control output and the changed data are defined to the same data group. The response time for control operations is usually about one half of the corresponding normal response time.

1.3 Real-time clock

SRIO 1000M has a real-time clock. The clock contains the current time including time from year to millisecond.

The software driven time is supported by hardware clock chip supplied with a back-up battery. This clock chip is used to keep time during power off situations.

The accuracy of the clock is better than 2 seconds per day.

SRIO 1000M synchronizes the clocks of SACO 100M units every 10 seconds and the clocks of SPA-bus units every second. The synchronizing is done by sending a clock message with current time. The message transfer time is compensated in SRIO 1000M so that the time is correct when it is received by a SPACOM unit.

The clock can be set by the user through the operator interface. It can also be set by the host system to synchronize the clocks of the host and SRIO 1000M.

The clock can be synchronized with an external minute pulse. The pulse can be given using a potential free contact output which is closed, when the seconds change from 59.999 to 00.000. The length of the minute pulse must be from 10ms to 100ms. At the rising edge of the pulse the SRIO 1000M rounds its clock to the closest minute (seconds are set to 00.000.).The operation of the clock sync is programmed with SYSPAR 18 command.

The real time clock is set with CLOCK command.

1.4 Host interface, ANSI X3.28 protocol

The ANSI X3.28 protocol is described e.g. in "Allen-Bradley: 1771-811 PLC-2-family/ RS-232-C Interface module 1771-KG; Users' Manual". With this protocol the SRIO 1000M can communicate e.g. with ABB's Substation Control System SCS 100, ABB's Remote Control System SPIDER MicroSCADA and Allen-Bradley's PLC 2.

The protocol used by SRIO 1000M is the same as Allen-Bradley's protocol except that the data presentation ways have been extended.

Analog data in database is transferred in 32 bit integer format or in BCD format. Digital data is transferred in 16 bit binary format. Parameter data is transferred in ASCII format. Clock time is transferred in BCD format. Events with time markings are sent to the host using 32 bits for event identification and 32 bits for time stamp.

Data in database (process data) is transferred to the host either on request or spontaneously when a data change is detected. Parameter data is transferred only on request. Events are sent spontaneously.

1.4.1 Address map

The host interface using ANSI X3.28 protocol is based on an address map. The map defines the address ranges through which different types of slave data can be addressed by the host computer. The map also defines an individual address for every data item in the database. (Figure 1.5)

The address map defines the following address ranges:
- Digital input data, DI \
- Digital output data, DO \
- Analog event data, EV > Process data
- Analog input data, AI /
- Analog output data, AO /
- Event data
- Parameter data buffer
- Time
- Diagnostic data
- System parameters
- Object parameters

"DI", "DO", "EV", "AI" and "AO" address ranges are used to transfer data: status information, measurement values and control data to/from the database. Data may be spontaneously sent to the host. Data can be read or written by the host using block read command, block write command or bit write command.

"Event data" address range is used to send events with time markings from the event buffer to the host.

Through "Parameter data buffer" the host can send and receive SPACOM format messages to/from SPACOM units or send SRIO command lines and receive SRIO's reply data.

SPACOM messages are sent to read or write setting values or other parameter data of the SPACOM units. Transfer of SRIO command lines can be used to program the device.

Only the most important status information, measurement values and output control data is transferred via the database. Other SPACOM data is transferred using "Parameter data buffer".

The host can set the SRIO 1000M real-time clock through "Time".

Through "System parameters" and "Object parameters" the host may change SRIO 1000M's so called system or object parameters. "Diagnostic data" is used by host protocols diagnostic functions.

The address map defines the following default address ranges:

Data type	Start address decimal	(word address) octal	Length (words)	Format
DI data	0	0	500	16 bit binary
DO data	500	764	250	16 bit binary
EV data	750	1356	250	16 bit binary + timestamp
AI data	1000	1750	500	32 bit binary
AO data	1500	2734	500	32 bit binary
Parameter data buffer	2000	3720	168	ASCII
Time	2300	4374	9	BCD
Event data	2400	4540	4	32 bit data + timestamp
Diagnostic data	2500	4704	26	(see page 29)
System parameters	3000	5670	2000	16 bit binary
Object parameters	5000	11610	5500	16 bit binary
Object parameters	10500	24404	1000	32 bit binary
Object parameters	11500	26354	2700	16 bit binary

The address ranges can be changed with ADDRESS_MAP command. The format of AI or AO data can be changed with ANSI_DATA command.

Note:
Usually the EV data area is located on the last half of the DO data area defined by the ADDRESS_MAP command. So the ADDRESS_MAP command gives the length of DO data area as 500 instead of 250.

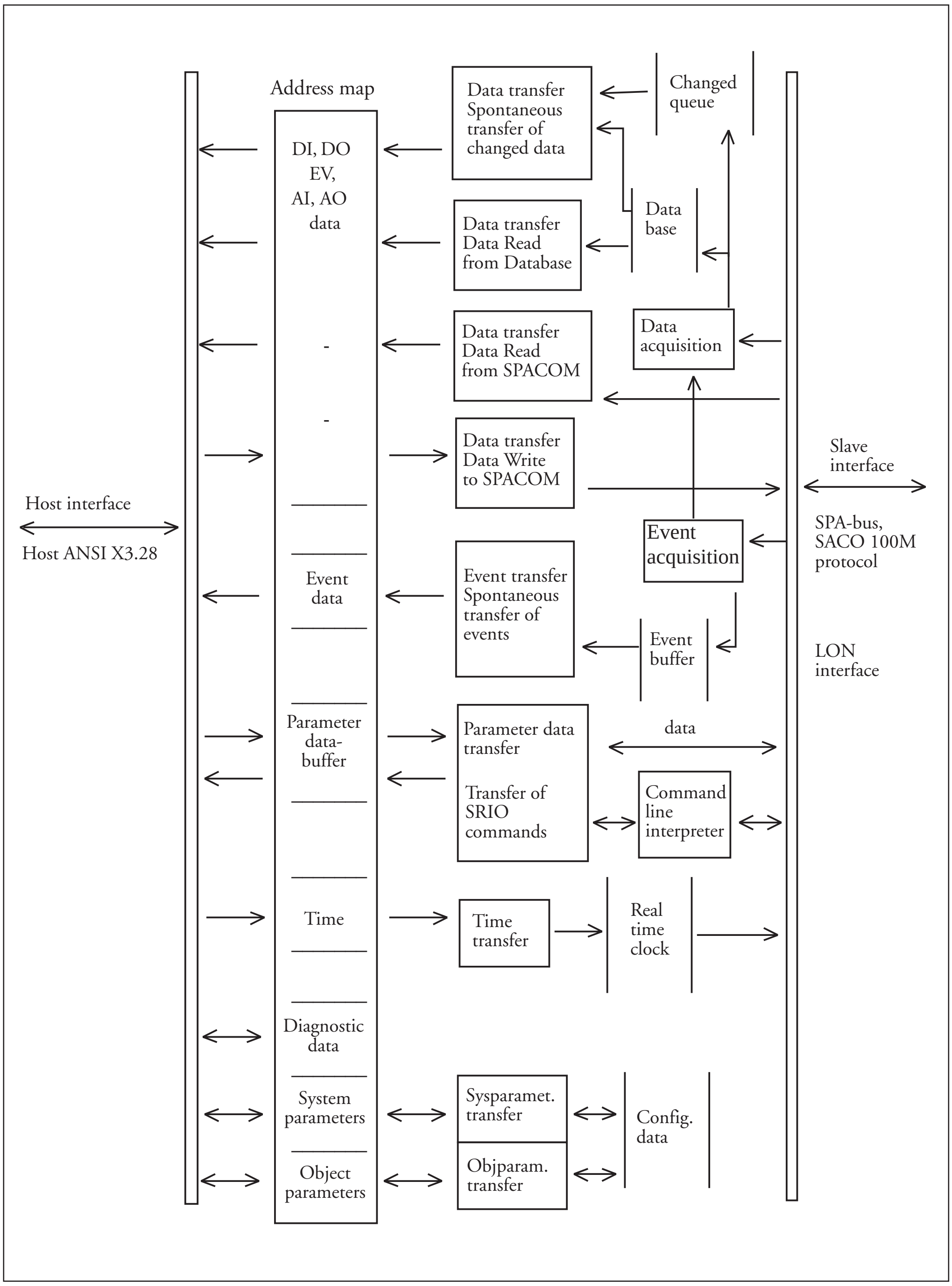


Figure 1.5. ANSI X3.28 host interface, functional block diagram.

1.4.2 Process data transfer

Addresses

For the host protocol (ANSI X3.28/AB) each data item of the database must be given a word address ranging from 1 to 32767.

The addresses of DI data items are defined to DI-address range,
the addresses of DO data items are defined to DO-address range,
the addresses of EV data items are defined to DO-address range,
the addresses of AI data items are defined to AI-address range,
the addresses of AO data items are defined to AO-address range.

Default addresses for the address ranges are:

DI-range:	0... 499	(octal: 0... 763) (500 * 16 bits)
DO-range:	500... 749	(octal: 764...1355) (250 * 16 bits)
EV-range:	750...999	(octal: 1356...1747)...(250 * 16 bits)
AI-range:	1000...1499	(octal: 1750...2733) (250 * 32 bits)
AO-range:	1500...1999	(octal: 2734...3717) (250 * 32 bits)

The user may also redefine the start and stop address of each address range.

(The order of the address ranges must be maintained: DI-range, DO-range, AI-range, AO-range.)

Figure 1.6 illustrates the address conversion from the ANSI address to the database index and to the corresponding SPACOM address.

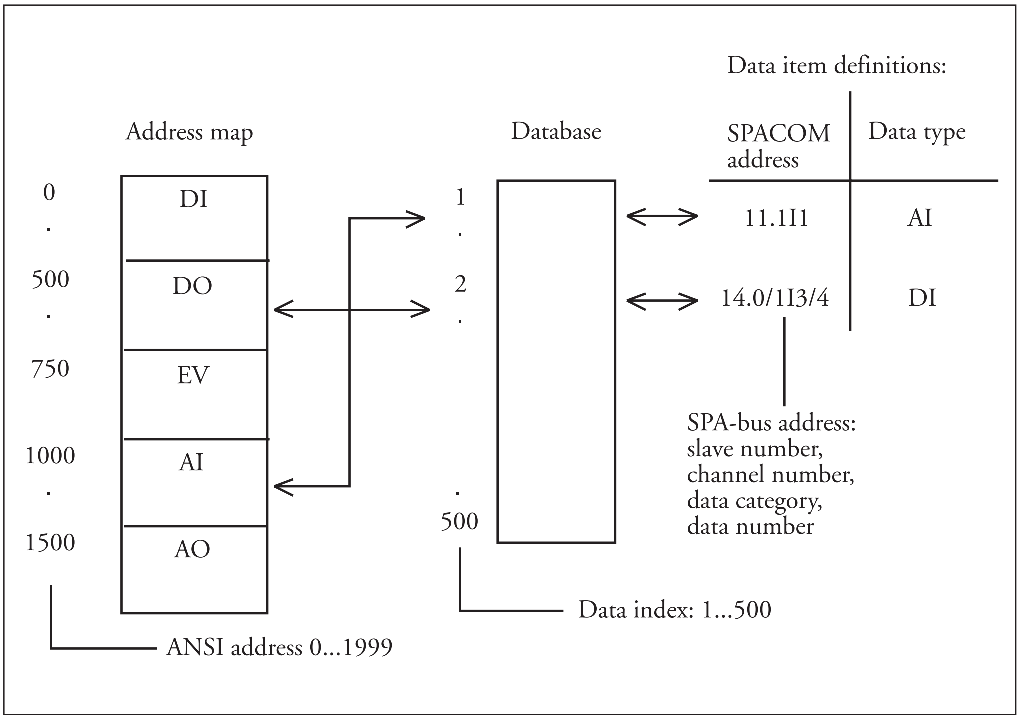


Figure 1.6. Address mapping: ANSI address/database/SPA-bus.

Modifying the Address map

The address ranges and individual data item addresses are defined using SRIO 1000M commands ADDRESS_MAP, ANSI_DATA and ANSI_ADDR.

Data read

The host can read a data from DI, DO, AI or AO address range. If the host reads many data items using one command then data is taken from the database. If one individual data item is read then data is read directly from the slave unit. For LON-interface the data is read from SRIO 1000M database, even if only one data item is requested.

Data is read using an unprotected block read command.

Command format:
DST SRC CMD STS TNS ADDR SIZE (CMD = 01H)

Reply format:
DST SRC CMD STS TNS DATA (max. 244 bytes) (CMD = 41H)

Data write

The host can set data bits in the DO address range and change analog data values in the AO address range. Only one bit or one analog value can be changed by one command message. The written data is not stored to the database, it is written directly to the slave unit.

The control commands sent to the LON-interface are converted to LON network variable format before sending them to LON.

A data bit is written using unprotected or protected bit write command.

Command format:
DST SRC CMD STS TNS ADDR Setmask Resetmask (CMD = 05H or 02H)
or
DST SRC CMD STS TNS ADDR Setmask Resetmask ADDR+1 Setmask Resetmask

Reply format:
DST SRC CMD STS TNS (CMD = 45H or 42H)

Setmask is an 8-bit value used to set a bit. Resetmask is a 8-bit mask used to reset a bit.

An analog value is written using unprotected or protected block write command.

Command format:
DST SRC CMD STS TNS ADDR DATA (CMD = 08H or 00H)

Reply format:
DST SRC CMD STS TNS (CMD = 48H or 40H)

Note:
The reply message is generated after data is sent to slave unit and slave reply message is received.

Spontaneous transmission of changed data

The new states of changed data bits in a digital data item or the new value of a changed analog data item are spontaneously sent to the host. An analog data item is sent, if the difference between the new value and the old value is greater than or equal to the user defined delta value.

The spontaneous transmission of a data item can be disabled or enabled. For a digital data item it can also be defined a mask which disables the spontaneous sending of individual bits.

(Note: All the bits are updated to the SRIO 1000M database.)

The spontaneous data transmission can also be globally disabled or enabled. These enable or disable bits are handled with DATA and SYSPAR commands.

Normal transmission:

The bits of a data word are sent to the host using unprotected bit write command message.

Command format: (CMD = 05H)
DST SRC CMD STS TNS ADDR Setmask Resetmask ADDR+1 Setmask Resetmask

Reply format:
DST SRC CMD STS TNS (CMD = 45H or 42H)

An analog value (AI or EV) is sent using unprotected block write.

Command format:
DST SRC CMD STS TNS ADDR DATA (CMD = 08H)

Reply format:
DST SRC CMD STS TNS (CMD = 48H)

If the transmission to the host does not succeed with the normal resendings, the data value is lost. If changed data values are generated faster than they can be sent to the host and the internal buffers are full, the newest changed data values are lost.

Multievent transmission:

A more efficient way of packing data to the message can be enabled by ANSI_SETUP command's "multi_event" parameter.

When multievent transmission is used, the data is always sent using unprotected block write. The general transmission format is:

Command format:
DST SRC CMD STS TNS DATA_BLOCK (CMD = 08H)

Reply format:
DST SRC CMD STS TNS (CMD = 48H)

DATA_BLOCK is a block of several data belonging to the same category (DI-, AI- or EV-data) with a maximum length of 128 bytes. Each data in DATA_BLOCK consists of byteaddress, value and possible timestamp.

In order to indicate end-of-data the value 32768 is added to the last byteaddress of the block. For the different data categories DATA_ BLOCK will look like this:

AI -data:
ADDR1 DATA1 ADDR2 DATA2 ... ADDRn+32768 DATAn
(DATA = 32 bit integer)

DI -data:
ADDR1 DATA1 ADDR2 DATA2 ... ADDRn+32768 DATAn
(DATA = 8 bit value: high nibble = bit no. 0..15,
low nibble = new bit value 0..1)

EV -data:
ADDR1 DATA1 TIME1 ADDR2 DATA2 TIME2 ... ADDRn+32768 DATAn TIMEn
(DATA = 16 bit integer, TIME = 32 bit word, Number of 100 microseconds since mid-night)

Note:
To ensure the data integrity between the host database and the SRIO 1000M database, the host should read all the data in SRIO 1000M database, when the host is restarted and also every now and then during the normal operation.

Process data format

The format of the data in message to/from the host system may be different from the format in the database. The data format can be modified with ANSI_DATA command. The possible alternatives are presented below.

DI and DO data: 16 bit binary.

AI and AO data: The AI or AO data can be in one of the three formats:

- 32 bit integer

```
    sxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx
    |              |
    |              |
   sign          value
(1=minus)
```

- 3 digit BCD

```
   0s00xxxx xxxxxxxx
    |        |
    |        |
   sign   3 digits
 (1=minus)
```

- 6 digit BCD

```
   0s00xxxx xxxxxxxx 0000xxxx xxxxxxxx
    |        |                 |
    |        |                 |
   sign   3 digits          3 digits
 (1=minus)
```

The data can have one of the following scales: 1, 10, 100, 1000 or 10000. The actual slave data value is multiplied with the scale before sending to the host. Data from the host is divided by the scale before sending to the slave unit. The user can select the suitable format and scale. The same format and scale are used for all AI and AO data items.

Note:
The default format: 32 bit integer. The default scale: 1000.

EV data:

The format of the EV data is the following:

Event data	Time stamp (2 * 16 bits)			
16 bits	second	minute	lsb milliseconds	msb milliseconds

The "Event data" is a 16 bit analog value, scaled with the same scale as AI-data values (default 1000).

The format of time stamp:

- normal transmission: second, minute, millisecond lsb, millisecond msb
- multievent transmission: 32 bit word, number of 100 microseconds since midnight

1.4.3 Event data transfer

All events from the event buffer are spontaneously sent to the host. One event is sent in one message. All events are sent through one address in address map "Event data".

The default address for "Event data" is 2400, decimal word address. This can be modified with ADDRESS_MAP command.

If the transmission to the host does not succeed with the normal resendings, the transmission is tried again periodically until transmission succeeds.

The events are sent to the host using unprotected block write command.

Command format: DST SRC CMD STS TNS ADDR DATA (8 bytes) (CMD = 08H)
Reply format: DST SRC CMD STS TNS (CMD = 48H)

The format of the event data is the following:

Event identification	Time stamp
32 bits	2 * 16 bits

The Event identification contains:

Byte 3	Byte 2	Byte 1	Byte 0
00000000	0UUUUUUU | SPACOM unit, (10 bits)	UUUccccc | channel, (7 bits)	ccEEEEEE | event number (6 bits)

The time stamp contains:

- normal transmission: seconds (byte 4), minute, millisecond lsb, millisecond msb (byte 7)
- multievent transmission: 32 bit word, number of 100 microseconds since midnight

In the message the bytes are sent byte 0 first and byte 7 last.

Note:
From the 32 bit integer Event identification the different parts can be calculated as follows:
- unit number = Event identification/8192
- channel = (Event identification/64) mod 128
- event number = Event identification mod 64

1.4.4 Parameter data transfer

Through "Parameter data buffer" the host can send and receive SPA-bus format messages to/from slave units. The structure of the buffer is illustrated in figure 1.7.

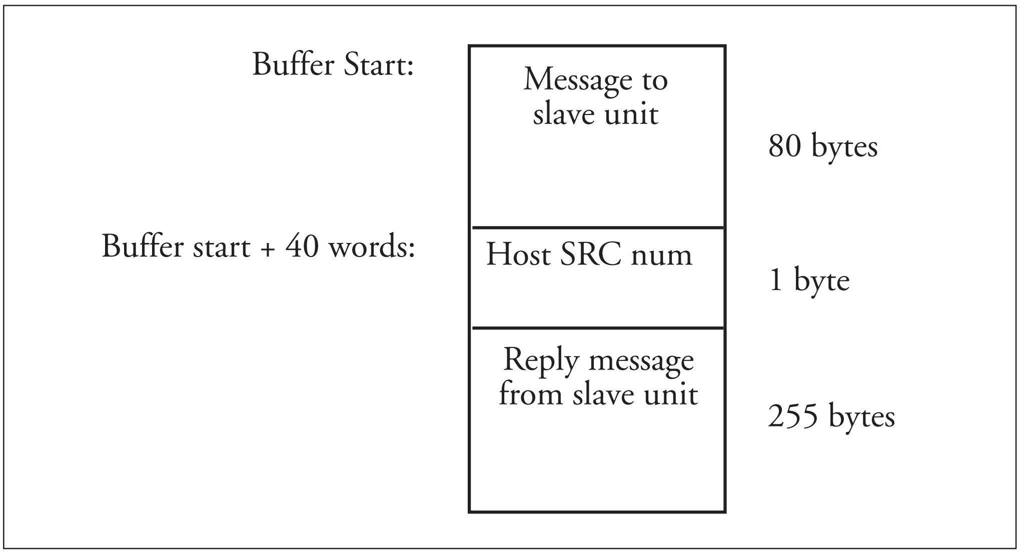


Figure 1.7. Parameter data buffer.

The host first writes a SPA-bus message to the beginning of the parameter data buffer and then reads the SPA-bus reply message from address: start of buffer + 40 words.

The host also gets back its SRC station number which can be used to check that the received reply message corresponds to the SPA-bus message it has sent. It is also possible for the host to read the contents of the buffer from the beginning of the buffer so that the host gets both the sent SPA-bus message and the received reply message (max. 244 bytes).

The default "Parameter data buffer" start address is 2000 and the default reply message start address is 2040 (decimal word addresses). These addresses can be modified with ADDRESS_MAP command.

The host can write SPA-bus message to the parameter data buffer using unprotected or protected block write command.

Command format:
DST SRC CMD STS TNS ADDR <SPACOM message> (CMD = 08H or 00H)

Reply format:
DST SRC CMD STS TNS (CMD = 48H or 40H)

The host can read the SPA-bus reply message using an unprotected block read command.

Command format:
DST SRC CMD STS TNS ADDR SIZE (CMD = 01H)

Reply format:
DST SRC CMD STS TNS <SPACOM message> (CMD = 41H)

The SPA-bus messages are transferred as data part of the messages. The format of the SPA-bus messages is the format used in SPA-bus protocol except: no start character or checksum is used, message ends with a CR-character (0DH).

Example: Message to slave: 14R1S1:<CR>
Reply form slave: 14D:22.10:<CR>

Note:
The host must read enough bytes from the parameter data buffer that the SPA-bus reply really fits to the reply message.

Note:
In order to transfer the parameters of a slave unit at least one data item must be defined for the unit. This definition is needed to find out the number of the serial interface to which the slave unit is connected.

The operation principle of the SRIO 1000M parameter data transfer is described in figure 1.8.

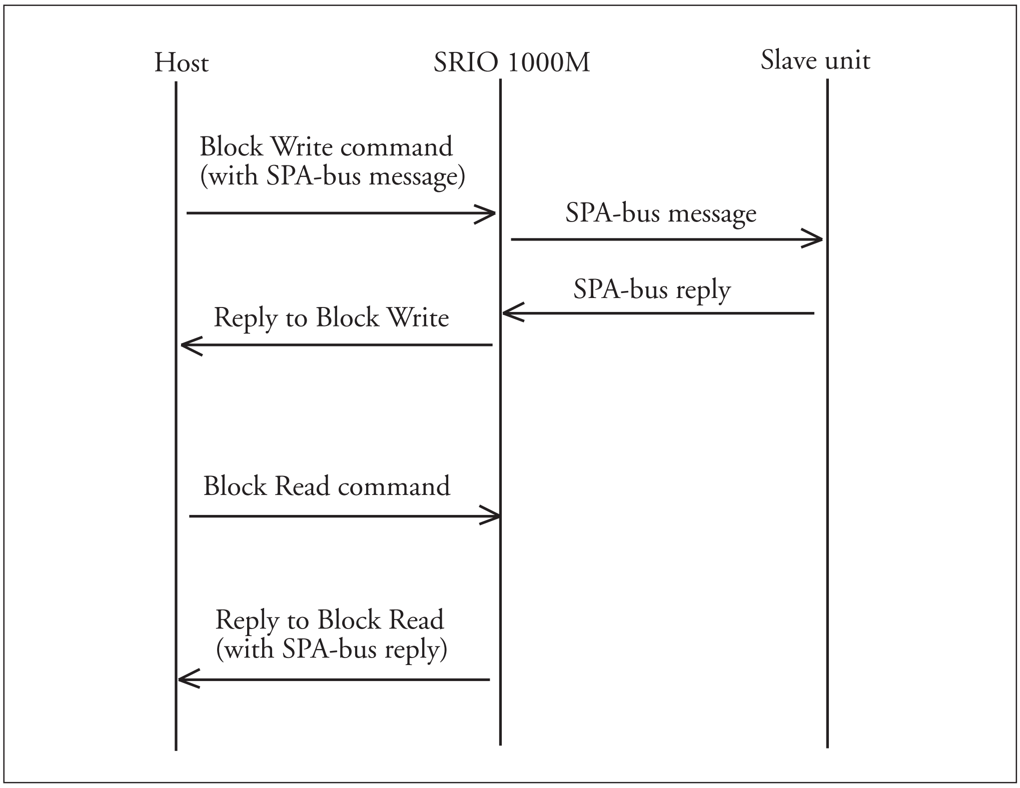


Figure 1.8. The operation principle of the SRIO 1000M parameter data transfer.

Note:
The SPA-bus reply is available in the buffer during one minute after "Reply to Block Write" has been sent. After the reply has been read the SRIO 1000M starts to wait for new Block write command with SPA-bus message.

1.4.5 Time transfer

The host can set SRIO 1000M real-time clock through address maps address entitled "Time".

Time is set using an unprotected or protected block write command.

Command format:
DST SRC CMD STS TNS ADDR DATA (9 words) (CMD = 08H or 00H)

Reply format:
DST SRC CMD STS TNS (CMD = 48H or 40H)

The format of the time is the following:

word 0 = status	(always 0)
word 1 = weekday	(1...7)
word 2 = year	(0...99)
word 3 = month	(1...12)
word 4 = day	(1...31)
word 5 = hour	(0...23)
word 6 = minute	(0...59)
word 7 = second	(0...59)
word 8 = milliseconds	(always 0)

All the words contain a 3 digit BCD value in format:

```
   0s00xxxx xxxxxxxx
     |          |
    sign     3 digits
 (always 0)
```

The default address for "Time" is 2300 (decimal word address). This address can be modified with ADDRESS_MAP command.

Note:
The whole time (9 words) must be sent in one block write command message.

Note:
The host may not set the time too often, because the milliseconds are always set to zero and the message transmission time from the host to the SRIO 1000M cannot be compensated.

After every time set operation there may become inconsistency in the time markings of the events.

A suitable time set period is 24 hours.

Note:
SRIO 1000M sets its clock according to the given time only if the difference between the given time and SRIO 1000M current time is more than one minute.

1.4.6 ANSI diagnostic and parameter messages

Diagnostic messages

The host can send to the SRIO 1000M the following diagnostic messages:
- Diagnostic Loop
- Diagnostic Reset
- Diagnostic Status
- Diagnostic Read

Diagnostic Loop

This command checks the integrity of the transmission over the communication link. The same data that is received is sent back.

Command format:
DST SRC CMD STS TNS FNC DATA (max. 243 bytes) (CMD = 06, FNC = 00)

Reply format:
DST SRC CMD STS TNS DATA (max. 243 bytes) (CMD = 46)

Diagnostic Reset

This command resets to zero all the diagnostic timers and counters.

Command format:
DST SRC CMD STS TNS FNC ADDR (CMD = 06, FNC = 07)

Reply format:
DST SRC CMD STS TNS (CMD = 46)

ADDR has to be the start address of the diagnostic counters (the same address as given in the diagnostic status reply message).

Diagnostic Status

This command can be used to read a block of status information from SRIO 1000M.

Command format:
DST SRC CMD STS TNS FNC (CMD = 06, FNC = 03)

Reply format:
DST SRC CMD STS TNS DATA (10 bytes) (CMD = 46)

The contents of the 10 data bytes of the reply message is listed in tables 1.1...1.3.

Byte	Meaning
1	Operating status of the SRIO 1000M as shown in table 1.2
2	Station type: SRIO 1000M = 80H
3,4	Zero
5,6	Zero
7,8	Starting address of diagnostic counters and timers
9	Program version and revision: bits 5...7 = version, bits 0...4 = revision (binary numbers)
10	Error code. (See table 1.3)

Table 1.1. Status bytes received by diagnostic status command.

Bits	Value	Meaning
0...2	1	Test (Local)
	2	Run (Normal operation)
	7	Error Run (Fault detected, SRIO 1000M runs)
3...4	0	Not used
5	0	Normal
	1	Error (Fault detected, operation stopped.)
6...7	0	Not used

Table 1.2. Status byte 1 bit values.

Error bit	Meaning
0	Fault on line A. (Serial interface A. Only SPCS 500M)
1	Fault on line 1. (Serial interface 1.)
2	Fault on line 2. (Serial interface 2.)
3	Fault on line 3. (Serial interface 3.)
4	Fault on line 4. (Serial interface 4.)
5	Fault on line 5. (Serial interface 5. Only SPCS 500M.)
6	Internal error.
7	Not used.

Table 1.3. Status byte 10, error code, bits.

Diagnostic Read

This command reads the diagnostic timers and counters (52 bytes).

Command format:
DST SRC CMD STS TNS FNC ADDR SIZE (CMD = 06, FNC = 01)

Reply format:
DST SRC CMD STS TNS DATA(52 bytes) (CMD = 46)

ADDR has to be the start address of the diagnostic counters (the same address as given in the diagnostic status reply message). SIZE has to be 52. The contents of these 52 bytes is shown in table 1.4.

Loc	Format	Counter type	Duplex
2	word	Transmitted messages	F/H
4	word	Received ACKS	F/H
7	byte	Received NACKS	F/H
9	byte	Transmit timeouts	F
10	byte	Transmitted ENQS	F
11	byte	Failed transmissions	F/H
13	word	Received messages	F/H
15	word	Transmitted ACKS	F/H
17	byte	Transmitted NACKS	F
18	byte	Received ENQS	F/H
21	byte	Parity errors + Framing errors + Overrun errors + Checksum errors + Buffer overflow errors	F/H
25	byte	No buffer	F/H
30	byte	Transmitted EOTS	H
The bytes that have not been mentioned contain value 0.			

Table 1.4. SRIO 1000M host interface diagnostic counters and timers as seen in diagnostic read reply message.

ANSI parameter messages

The host can send to SRIO 1000M the following command messages, which are used to change parameters of the host interface:
- Set ENQs
- Set NAKs
- Set Timeout
- Set Variables

Set ENQs

This command sets the maximum number of ENQs that SRIO 1000M issues per message transmission. The default setting is 3.

Command format:
DST SRC CMD STS TNS FNC DATA (1 byte) (CMD = 06, FNC = 06)

Reply format:
DST SRC CMD STS TNS (CMD = 46)

Set NAKs

This command sets the maximum number of NAKs that SRIO 1000M accepts per message transmission. The default setting is 3.

Command format:
DST SRC CMD STS TNS FNC DATA (1 byte) (CMD = 06, FNC = 05)

Reply format:
DST SRC CMD STS TNS (CMD = 46)

Set Timeout

This command sets the maximum amount of time that SRIO 1000M waits for an acknowledgment to its message transmission. The setting is expressed so that the actual second value is multiplied with 38.

Command format:
DST SRC CMD STS TNS FNC DATA (1 byte) (CMD = 06, FNC = 04)

Reply format:
DST SRC CMD STS TNS (CMD = 46)

Set Variables

This command is a combination of Set Timeout, Set NAKs and Set ENQs commands. It sets the maximum Timeout, ENQs and NAKs all at once. All three must be given in the message. The first byte sets the Timeout, the second sets the NAKs and the third sets the ENQs. If one of the data bytes is incorrect the other values are also rejected.

Command format:
DST SRC CMD STS TNS FNC DATA (3 bytes) (CMD = 06, FNC = 02)

Reply format:
DST SRC CMD STS TNS (CMD = 46)

1.4.7 System parameters

The host can read and write through "System parameters" address range a couple of general operation parameters of SRIO 1000M.

The default start address for "System parameters" is 3000. This address can be modified with ADDRESS_MAP command.

There are the following System parameters each using one word:

Word 0: Spontaneous event enable.
1 = spont. transmission of events is enabled (default)
0 = spont. transmission of events is disabled
Word 1: Spontaneous data enable.
1 = spont. transmission of data is enabled (default)
0 = spont. transmission of data is disabled
Word 2: Store command./ Reset command
1 = starts the storing of configuration data to EEROM memory
FFFF = resets the device (no response to host)
0 = no meaning
Word 3: Analog data format
0 = 32 bit integer (default)
1 = 3 digit BCD
2 = 6 digit BCD
Word 4: Analog data scale 1, 10, 100, 1000 or 10000 (1000 is default value)
Word 5: Timed polling interval 30...30000 seconds (60 is default value)

The parameters can be read using unprotected block read command.

Command format:
DST SRC CMD STS TNS ADDR SIZE (CMD = 01H)

Reply format:
DST SRC CMD STS TNS DATA (CMD = 41H)

The parameters can be written using unprotected or protected block write command.

Command format:
DST SRC CMD STS TNS ADDR DATA (CMD = 08H or 00H)

Reply format:
DST SRC CMD STS TNS (CMD = 48H or 40H)

Note:
If parameters are changed by the host, they can be stored in EEROM memory using system parameter "Store command".

Note:
If system parameters are read while parameter storing is in progress, word 2 has value 1, otherwise value 0.

1.4.8 Object parameters

The host can read or write through "Object parameters" address range the following definitions:

- data item definitions
- data group definitions
- event poll definitions

The default start address for "Object parameters" is 5000. This address can be modified with ADDRESS_MAP command.

The parameters can be read using unprotected block read command.

Command format:
DST SRC CMD STS TNS ADDR SIZE (CMD = 01H)

Reply format:
DST SRC CMD STS TNS DATA (CMD = 41H)

The parameters can be written using unprotected or protected block write command.

Command format:
DST SRC CMD STS TNS ADDR DATA (CMD = 08H or 00H)

Reply format:
DST SRC CMD STS TNS (CMD = 48H or 40H)

Note:
If parameters are changed by the host, they can be stored in EEROM memory using system parameter "Store command".

1.4.8.1 Data item definitions

For each of the 500 data items there are:

- ANSI address
- bus number
- SPACOM address
    - unit number
    - channel number 1
    - channel number 2
    - data category identifier
    - data number 1
    - data number 2
- data type/format
    - data type
    - data format
- delta value / bit mask
- status word

The parameters are mapped to "Object parameters" address range as shown in figure 1.9.

To read or write the parameters the host must use the data index (1...500) of the concerned data item. A data item is deleted, when its bus number is set to 0.

To avoid operation with inconsistent parameters, note the following:

- When a new data item is created, the status word must be given as the last parameter.
- When making changes to the parameters (other than status word) of a data item, it is recommended to act as follows:
  - read and save the status word
  - write 0 in the status word
  - change the parameters
  - write the original value in the status word

"Object parameters" start address	ANSI address - 500 words - each word contains the 16 bit ANSI word address of a data item
... + 500:	Bus number - 500 words - the number of the SRIO 1000M serial interface
... + 1500:	SPACOM address - 500 * 6 words - each 6 word group contains: Word 1: unit number Word 2: channel number 1 Word 3: channel number 2 Word 4: data category identifier (73=I, 79= O, 83=S, 86=V, 69=E) Word 5: data number 1 Word 6: data number 2
... + 4500:	Data type/format - 500 * 2 words - each 2 word grop contains: Word 1: data type (0=DI, 1=AI, 2=DO, 3=AO, 4=EV) Word 2: data format (0=binary, 1=decimal, 2=hex, >100=conversion table number for EV-data + 100)
... + 5500:	Delta value/bit mask/index of direct data - 500 * 32 bits
... + 6500:	Status word - 500 words

Figure 1.9. The object parameters defining the data items.

Delta value / bit mask / index of direct data

If the data item is a DI or DO data item then the delta value/bit mask contains:

```
00000000 00000000 xxxxxxxx xxxxxxxx
                          |
                     16 bit mask
```

If a bit in the bit mask is set, then the spontaneous sending of the corresponding data bit is disabled.

If a data item is an AI or AO data item then the delta value/bit mask contains a 32 bit delta value. The scale of the delta values is 1000: e.g. value 0.05 is presented as 50.

If a data item is an EV-data, then this item is the index of the direct data corresponding the EV-data.

Status word

The meaning of the bits of a status word are listed in table 1.5.

Bit	Meaning
0	Update enable
1	Continuous poll enable
2	Timed poll enable (Periodical poll enable)
3	Spontaneous transmission enable
4	-
5	-
6	Temporary error
7	Permanent error
8...15	-

Table 1.5. The bits of the status word in object parameters.

1.4.8.2 Data group definitions

There can be 100 data groups each containing 10 data items. The definitions are accessed through "Object parameters" address range beginning from address: "Object parameters" start address + 7000.

"Object parameters"
start address + 7000:

Data group definitions:
- 100 * 10 words
- 10 word group contains:
  word 1: data index 1
  word 2: data index 2
  .
  .
  .
  word 10: data index 10

Figure 1.10. The object parameters defining the data groups.

1.4.8.3 Event poll definitions

The total number of SPA-bus units which can be polled for event information is 300 (100 units per bus).

The required parameters for a unit are the following:
- bus number
- unit number
- unit type
- status

The parameters are accessed through "Object parameters" address range beginning from address: "Object parameters" start address + 8000.

"Object parameters" start address + 8000:	Event poll definitions: - 300 * 4 words - each 4 word group contains: word 1: bus number word 2: unit number word 3: unit type word 4: status word

Figure 1.11. The object parameters defining the data groups.

The possible unit types are the following:
0 = digital alarm unit SACO 16D_
1 = analog alarm unit SACO 16A_
2 = protective relay unit
3 = control unit of type SPTO or SPOC

The contents of the status word:

```
                lsb
0000 0000 PT00 000E
          | |     |
          | |     event poll enable bit
          | temporary error bit
          permanent error bit
```

1.4.9 Transfer of SRIO commands

Through "Parameter data buffer" the host can send commands to SRIO and read SRIO's responses to the commands. The structure of the buffer is illustrated in figure 1.12.

Buffer start:	Command to SRIO	80 bytes
Buffer start + 40 words:	Host SRC number	1 byte
	Reply from SRIO	255 bytes (only 80 bytes is used)

Figure 1.12. Parameter data buffer.

The host first writes command lines (SRIO command and required parameter data lines one at a time) to the beginning of the parameter data buffer and then reads the SRIO's reply from address: start of buffer + 40 words.

The host also gets back its SRC station number which can be used to check that the received reply message corresponds to the SRIO command it has sent. It is also possible for the host to read the contents of the buffer from the beginning of the buffer so that the host gets both the sent command and the received reply (max. 244 bytes).

The default "Parameter data buffer" start address is 2000 and the default reply message start address is 2040 (decimal word addresses). These addresses can be modified with ADDRESS_MAP command.

The host can write command lines to the parameter data buffer using unprotected or protected block write command.

Command format:
DST SRC CMD STS TNS ADDR <SRIO command> (CMD = 08H or 00H)

Reply format:
DST SRC CMD STS TNS (CMD = 48H or 40H)

The host can read the command reply using an unprotected block read command.

Command format:
DST SRC CMD STS TNS ADDR SIZE (CMD = 01H)

Reply format:
DST SRC CMD STS TNS <SRIO reply> (CMD = 41H)

The command lines are transferred as data part of the messages. The format of the command line sent to SRIO and reply data received from SRIO are exactly the same that would be used from the programming terminal interface except that reply data may contain nulls (00H), which must be ignored by the host device. Every command line must end with a CR-character (0DH).

Example: Command lines to SRIO: CLOCK<CR>
<CR>
Reply from SRIO: 80-12-31 23.55:00<CR>
-

Note:
The host must read at least 80 bytes from the parameter data buffer, because the SRIO's reply is transferred in blocks of max. 80 bytes. Unused bytes are filled with nulls (00H).

The operation principle of the SRIO command transfer is described in figure 1.13.

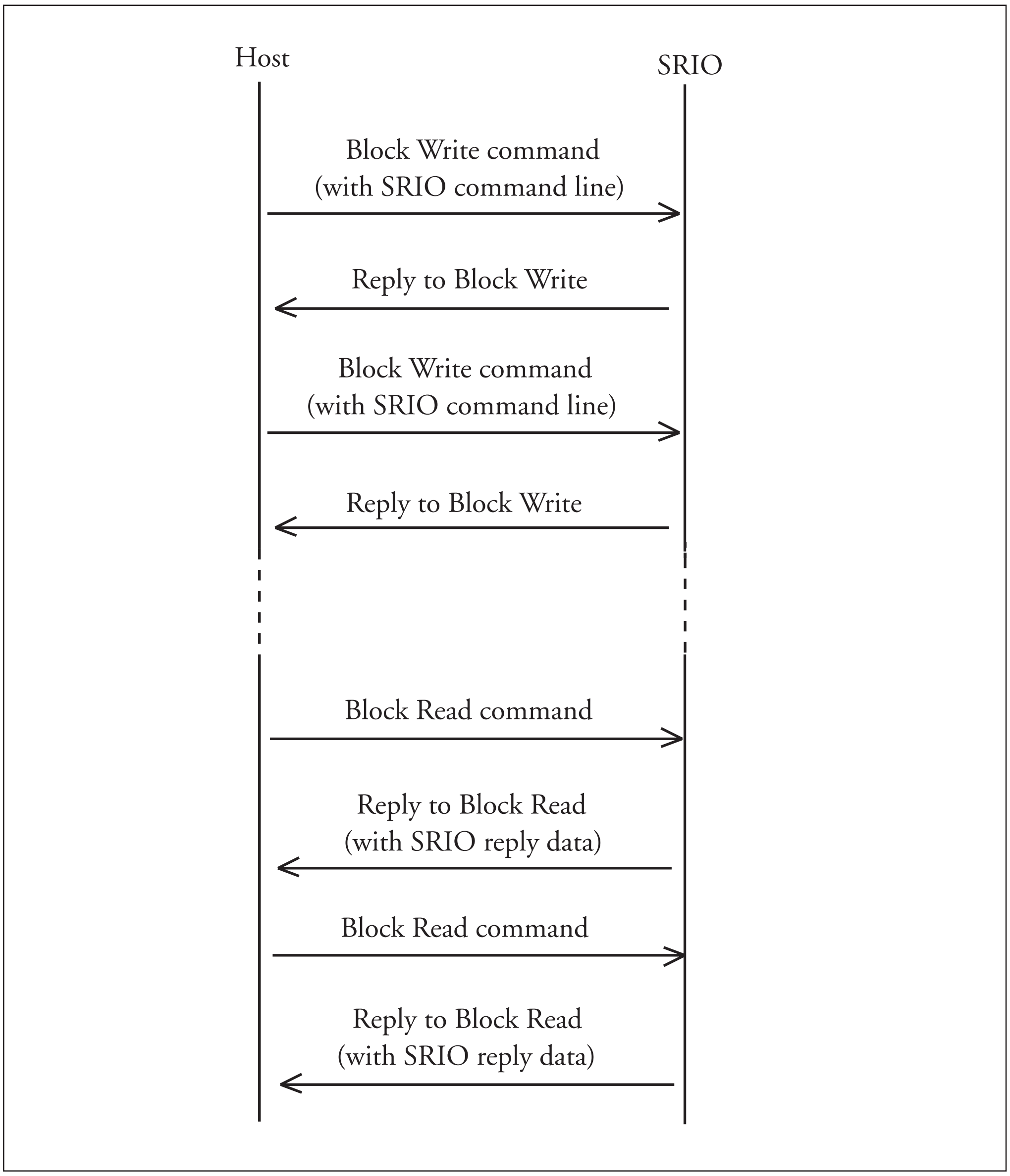


Figure 1.13 The operation principle used in transferring SRIO commands.

The SRIO's internal buffers used to store the reply data generated to received SRIO command lines can be cleared by sending #-character.

Example: Clear command to SRIO: #<CR>
Reply from SRIO: 00H,00H,00H,00H,00H,.....

If the host sends a command and the command is left "open" i.e. all the required parameter lines or final <CR> is not sent or the communication interface to the host suddenly breaks, then the command which was left "open" will be aborted and a prompt "-" will be displayed in reply message after timeout period of about 2 minutes.

The maximum number of characters SRIO can hold in its command input buffer is 300. This means that the host can send multiple command lines with combined length of 300 characters before it starts to read the reply data given by SRIO.

The maximum length of characters SRIO can hold in its command reply buffer is 1300. This means that, if the host does not read SRIO's reply data, then SRIO can execute commands from command input buffer until 1300 bytes of reply data has been generated.

If the host wants to download a command file to SRIO and it is not interested in reading the reply data from SRIO, it can send the file line by line and clear the command reply buffer by sending one #-character after every command line or before the total length of sent command lines reaches 300 (figure 1.14).

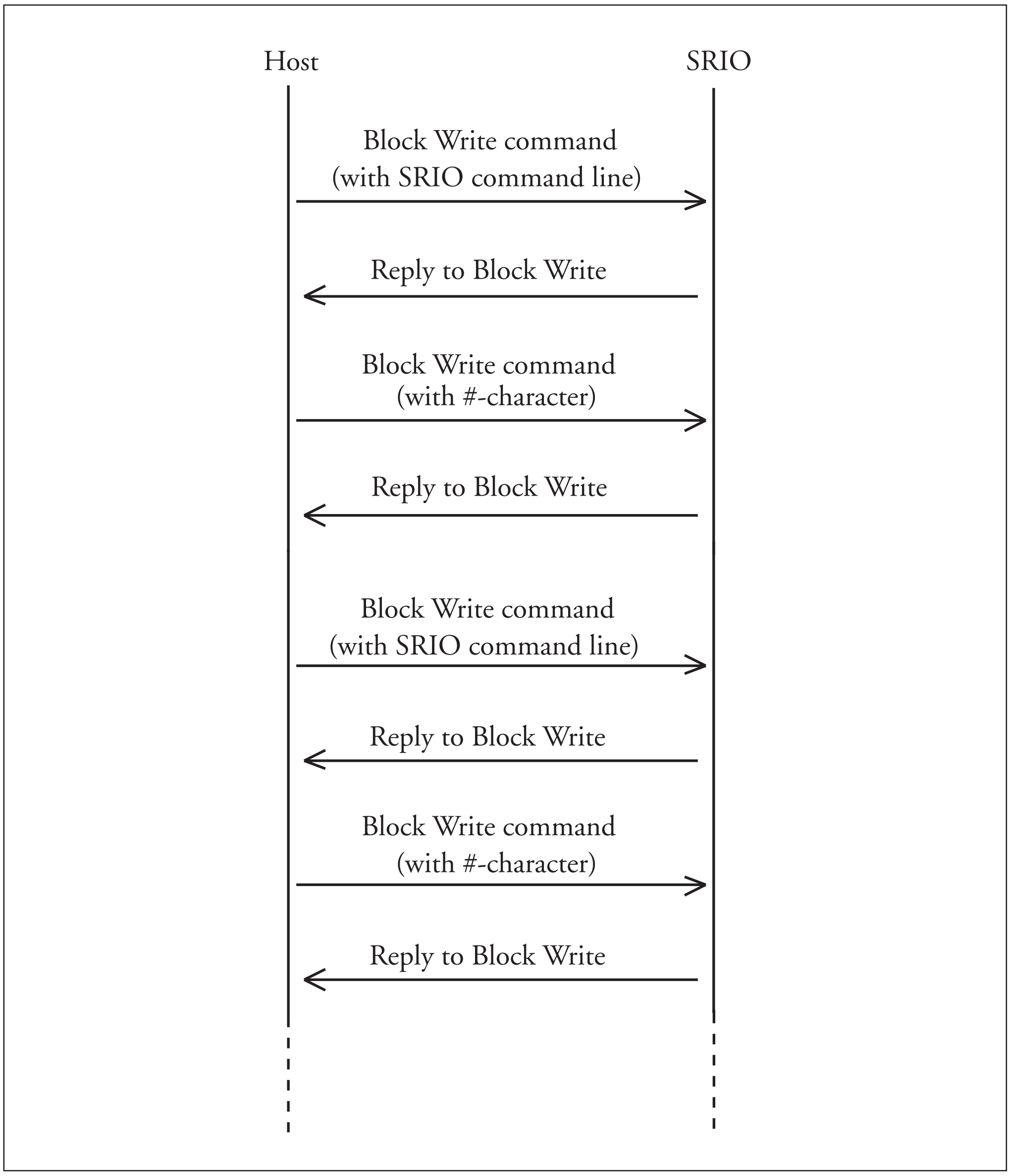


Figure 1.14. Command file downloading without reading the SRIO reply data.

1.5 Host interface, SACO 100M protocol

The SACO 100M protocol used for SRIO 1000M host interface is described in document: "SACO 100M communication protocol for data communicators SPSC 500M and SRIO 1000M."

Figure 1.15 describes the functions of the SACO 100M host interface. In principle there are two possibilities: the host computer can access data in SRIO 1000M or in slave units which are connected to the SRIO 1000M.

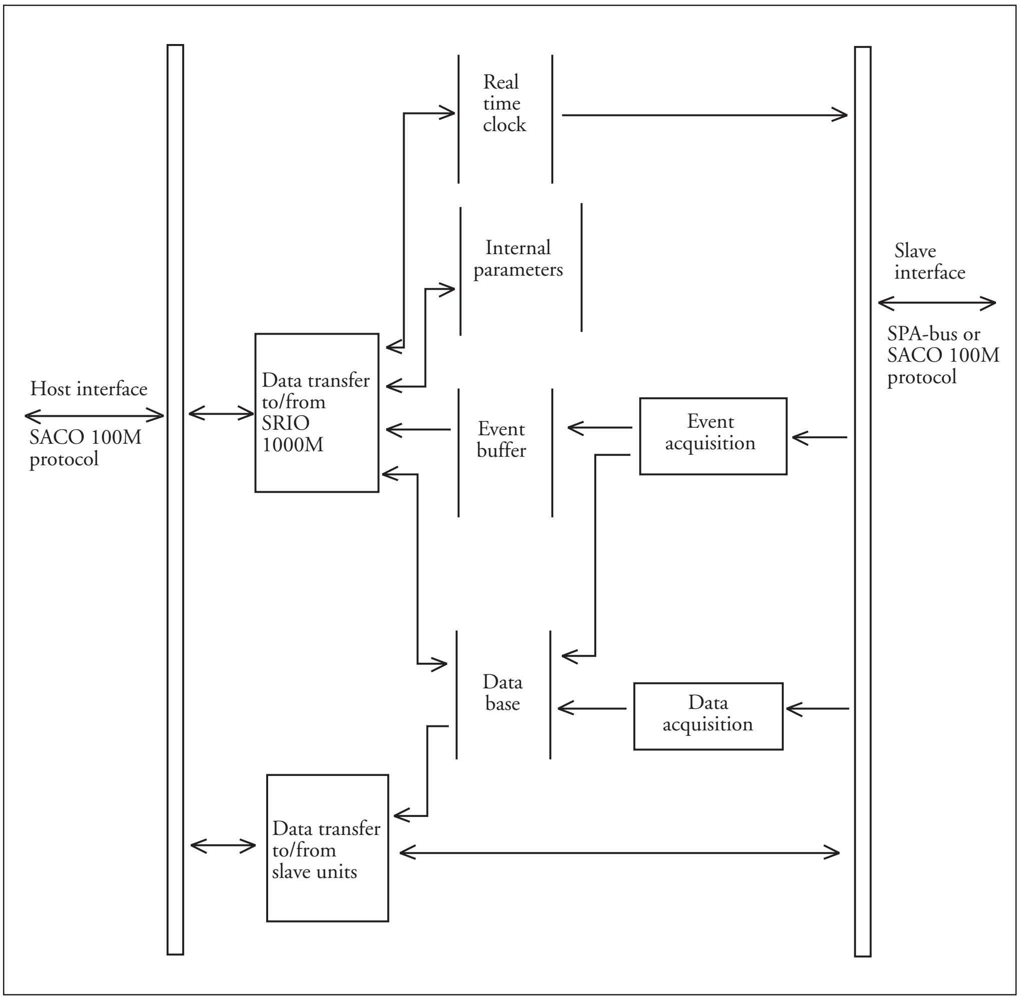


Figure 1.15. SACO 100M host interface, functional block diagram.

1.6 Host interface, Modbus protocol

Description of the Modbus protocol can be found e.g. from the Modicon Modbus Protocol Reference Guide PI-MBUS-300 Rev. D. The use of Modbus protocol allows communication between SRIO 500M or SRIO 1000M and different control systems.

The serial transmission mode is RTU (Remote Terminal Unit). The 32 bit analog data value is transferred in 16 bit signed integers with scaling factor of 100. Digital data is transferred in 16 bit binary format.

Data from SRIO database is sent to the Modbus master when requested. Notice that event data transfer to the master is not supported by this protocol.

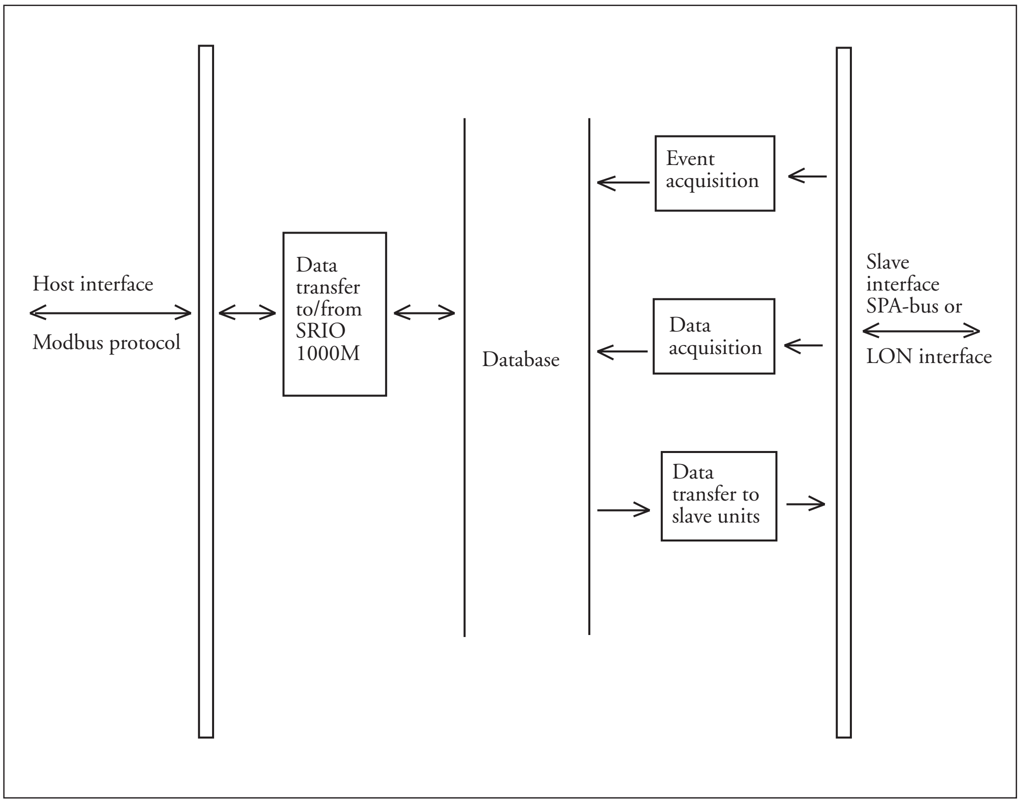


Figure 1.16. Modbus interface, functional block diagram.

1.6.1 Database

The 500 data items in SRIO database can be figured as:

- Digital input data, DI 16-bit
- Digital output data, DO 16-bit
- Analog input data, AI 32-bit
- Analog output data, AO 32-bit

For the digital input or output data the Modbus master considers the database as 8000 discrete inputs or outputs that are packed as 16 bits per one database item. The valid starting address for coils and discrete inputs is n*16+1. Where n is a natural number from 0 to 499.

The status of one coil or discrete input is packed so that the LSB of the database item corresponds to the lowest coil/input address and the status of coil/input with highest address is the MSB of the item.

The database address of SRIO 1000M	The coil/discrete input address of Modbus master
1	1...16
2	17...32
n	(n-1)*16+1...(n-1)*16+16
500	7985...8000

Table 1.6 Mapping of coils/discrete inputs in SRIO database.

For analog data SRIO database is seen by the Modbus master as 500 input or output registers mapped into 500 analog input or output items. When analog data is received from SRIO's slave device it is stored as 32 bit signed integer with scaling factor of 1000 into SRIO database. Because registers in Modbus are 16 bit long and have scaling factor of 100 conversion is performed before sending data to the Modbus master. Also conversion is performed when SRIO receives analog data from the master device.

The data item index in SRIO database must correspond to the address given to the Modbus input or holding register.

Note:
In the message frame the address value of coil, digital input or register is one less than the actual data item in SRIO database or defined in Modbus master.

1.6.2 Supported commands

Data read commands

Commands for reading digital data:

01 Read Coil Status and 02 Read Input Status

The command code 01 is used for reading digital output data from SRIO database. For reading digital input data sent by the SRIO's slave command code 02 is used.

Below can be seen examples of these two commands.

Query: 01 01 00 10 00 10 3C 03
Query: F6 02 00 10 00 10 78 03

The first bytes 01 and F6 (1 and 246) in these two example queries specify the SRIO devices to which the query is directed. The second bytes 01 and 02 are the actual function codes. The third and fourth bytes specify the address of starting coil/input. In the message framing coils and digital inputs are addressed starting at zero. This means that the first coil/input to be read in both of these examples is 17. In the SRIO database this corresponds to data item in index 2. The number of coils/inputs to be read is given in bytes five and six, the starting coil/input is included. Here the value is 16 in decimal format. CRC bytes are the last bytes in messages.

Response: 01 01 02 F9 2F B8 4C
Response: F6 02 02 BB FF 8B 08

The first two bytes in the response messages identify the slave device and the command to which this response message was created and sent to.

The third byte 02 is the number of data bytes in the response message.

This value can be up to FA, 250 in decimal format.

The fourth bytes F9 and BB in response messages describe the status of the first eight coils or input registers from the starting coil/input.

In the binary format the value F9 is 11111001. This means 17 is ON, coils 18 and 19 are OFF, and coils 20 through 24 are ON. In the binary format value BB in the second example is 10111011. The status of inputs 17 and 18 is ON, input 19 has status OFF, inputs 20, 21, 22 are ON, input 23 is OFF and the last input 24 in this byte is ON.

The fifth byte (2F and FF) gives the status for next eight coils/inputs. Also in this byte (and all following data bytes) the status of coil/input with lowest address is given in the LSB and the MSB has the value of highest coil/input. In the binary format the value 2F is 00101111. This means that the status of coils 25 through 28 is ON, coil 29 is OFF, coil 30 is ON and coils 31 and 32 are OFF. The value FF in the second example is in binary format 11111111 this means that all the inputs have status ON.

Bits are transmitted serially in the line from LSB to MSB. In other words from the starting coil/input to the coil/input with highest address x...x+7, x+8...x+15 etc.

Commands for reading analog data:

03 Read Holding Register and 04 Read Input Register

For reading analog output data from SRIO database command code 03 is used. Command code 04 is used for reading analog input data sent by the SRIO's slave.

Below can be seen examples of these two commands.

Query: 0F 03 00 02 00 04 E5 C9
Query: 01 04 00 06 00 06 11 C8

The first bytes 0F and 01 (15 and 1)are the slave addresses of the SRIO devices to which the query is directed. The second bytes 03 and 04 are the actual function codes. The third and fourth bytes specify the address of first holding/input register to be read. Also registers are addressed starting at zero. The first holding register to be read from SRIO database in the first example is the data item with index 3. In the second example the first input register to be read is item 7 in SRIO database. The number of holding/input registers to be read is given in bytes five and six. In these examples four and six registers are wanted to be read. As in all of the Modbus RTU mode messages CRC bytes are the last bytes in messages.

Response: 0F 03 08 00 34 01 54 02 30 02 C6 6C 83
Response: 01 04 0C 02 26 00 41 0F 0A 00 3D 00 50 0A 43 3C DE

These response messages are created by the SRIO with slave number 15 and SRIO number 1 for the commands 03 and 04. This information is given in first two bytes of the example response messages.

Third byte informs the number of data bytes in the response message. The first response, for command 03, has 8 data bytes in the data field. The second response has 12 data bytes. The maximum number of data bytes in one message is 250, this means that the maximum number of registers to be read at once is 125.

The bytes four and five form data value for the first register. This value for first example is 0.52. The second, third and fourth registers have values 3.40, 5.60, and 7.10. The second example has 0C data bytes, 12 in decimal format. The values from the first register to sixth register are 5.50, 0.65, 38.50, 0.61, 0.80 and 26.27. The last two bytes are CRC bytes.

Data write commands

The master device can set data bits in the Digital Output items and change Analog Output data values in SRIO database. Data is stored to the database and sent to slave device of SRIO. Write messages can be broadcast to several slave devices. Supported write commands are:

05 Force Single Coil and 06 Preset Single Register

Command code 05 is used for writing digital output data. Below can be seen an example of this command.

Query: 04 05 00 1A FF 00 AD FD

This request is directed to SRIO with slave number 04. The 05 is the command code. The third and fourth bytes form the number of coil to be set, forced. The coil to be set in the first example is coil 27, which corresponds to the 11th bit (counted from LSB) of data item 2 in SRIO database. The fifth and sixth bytes specify the requested state. The value FF 00 request the coil to be ON and 00 00 set the status OFF. Other values and combinations are illegal and will produce exception response illegal data value. No changes to coil's status will be made in this case.

For writing analog output data command code 06 is used. An example of this command is seen below.

Query: 0F 06 00 05 02 C6 18 F9

This request is sent to SRIO with slave number 15 in decimal format.

The third and fourth bytes form the holding register, analog output, which value is wanted to be preset. In this example the register 00 05 is item with index 6 in SRIO database. The requested preset value is specified with bytes five and six. Here the requested value is 02 C6 in hexadecimal format and 7.10 in decimal format. CRC bytes are also here the last two bytes.

The normal response messages sent by the SRIO for both of the writing request are copies of the queries. The response messages are sent to the host after acknowledgment is received from SRIO's slave device.

Note:
The new value is first stored to the database and then sent to SRIO's slaves. If negative acknowledgment is received or some other error occurs then the database is updated with the old value. Exception response can be sent to the host to inform it about the type of error.

1.6.3 Exception Responses

In addition to the normal responses SRIO may generate the following exception responses.

01 Illegal Function

This exception is generated when SRIO receives a query with a function code other than 01, 02, 03, 04, 05 or 06. Requested function is not supported by SRIO.

02 Illegal Data Address

Reasons for this exception response are the following:

- The data address received in the query is not allowable address for the slave. For digital data addresses less than 1 or over 8000, or the starting address is other than n*16+1. For analog data address less than 1 or over 500.
- The corresponding data item is not configured to the SRIO database.
- The number of inputs/outputs requested to be read at once exceeds for analog data 125 items and for digital data 2000 items.

03 Illegal Data Value

The value in the query data field is not allowable for the slave.

06 Device Busy

This exception response is generated and sent when SRIO is performing writing to database operation and next writing request is issued. To complete writing operation SRIO needs to receive acknowledgment from slave unit connected to other network.

Note:
It is up to the master device to handle the exception responses sent by SRIO. For example by sending subsequent retry.

1.7 Local event reporting

The SRIO 1000M can be programmed to make local event report on 1 or 2 event printer devices (device 1 and device 2). Note that SRIO 500M cannot be used for local event reporting.

The event report of SRIO 1000M may consist of:

- time
- event text consisting of the following parts
  - group text
  - slave text
  - channel text
  - event code text
- data from the slave units

Part of event text may be printed in front of the text to show the priority of the event.

An example of possible printout from the event recording program:

```
Front text                                                   Event
:                       Group       Slave                    code
:  Time                 text        text                     text
:  :                    :           :                        :
:  :                    :           :                        :
**89-10-30 11.00:31.472 FEEDER 5  OVERCURRENT RELAY          TRIP

                          Current before trip = 2.05  *  In
                          :                     :     :
                          :                     :     :
                          Event                 Data  Event
                          code                  from  code
                          text                  slave text
```

The operation of the event reporting is programmed using commands DEVICE, SET_PRINTER, PAGE_HEADER and T.

Text is written using the T command. The maximum amount of text is about 50 000 characters. The text memory is used in sections of 50 characters. There can be about 1000 texts if the maximum length of the texts is 50 characters. The maximum length of one text part is 250 characters. The maximum channel number for an alarm system using max. 50 characters long channel text is about 1000 channels.

Data from the slave units

The event report may contain data from the slave units. The user defines the data to be displayed by embedding SPA-bus messages in the event text, when writing the event texts with T command. The SPA-bus messages are written between special characters. When text is printed the message is sent to the SPA-bus and the received data is printed out as part of the event report. The most suitable place for embedding SPA-bus messages is often the event code text.

The embedded SPA-bus message can be any legal message, so data can also be sent to the slaves. This feature can be used e.g. to turn on an alarm lamp, when some special event is received from a slave unit.

1.8 LON interface

The LON interface is implemented using a Serial LonTalk Adapter (SLTA). The adapter is connected to the SRIO 1000M with RS-232 interface.

The SRIO 1000M LON interface supports the network variable data types used in the SPA-ZC 100/102, called LON/SPA-gateway in this text, and some other simple data types (e.g. 8/16 bit integer and 8/16 unsigned integer).

Incoming LSG data format	Handling in SRIO	Notes
LSG event type 0 (LSG SPA event)	Stored to event buffer. Sent to ANSI as normally.	
LSG event type 1 (LSG analog event)	Discarded.	SRIO accepts analog data as network variables.
LSG event type 2 (LSG digital event)	Data stored to data base. Data sent to ANSI.	If data defined as EV in SRIO, then data is sent with time stamp to ANSI
LSG event type 7 (LSG SPA analog event)	Discarded.	There is no support for Q-events in SRIO.
LSG event type 8 (LSG extended SPA event)	Discarded.	Not yet implemented.
NV type 83 = SNVT_state (16 bit digital data)	Data stored to data base. Data sent to ANSI.	In SRIO the data base definition may be DI or EV.
NV type 252 = NV_32b_analog (32 bit analog value with scale 1000)	Data stored to data base. Data sent to ANSI.	SRIO data base definition must be AI.
NV type 253 = NV_16b_analog (16 bit analog value with scale 100)	Data stored to data base. Data sent to ANSI.	SRIO data base definition must be AI. Scale 100 is not noticed by SRIO. Value sent to ANSI is original value divided by 10.
NV type 88 = SNVT alarm (SPA event)	Stored to event buffer.	Data is always stored as events.
NV type 88 = SNVT alarm (Analog SPA event)	Stored to event buffer.	Data is always stored as events.
NV type 88 = SNVT alarm (Digital event)	Stored to event buffer.	Data is always stored as events.
NV type 88 = SNVT alarm (Analog event)	Stored to event buffer.	Data is always stored as events.

Table 1.7. Handling of different LSG message types in SRIO

The task of the SRIO 1000M is to collect data from the network nodes and act as the network manager. The SRIO 1000M also connects the bus to a host computer using ANSI X3.28, Modbus, or SACO 100M protocols and reports events to an event printer.

LON based data acquisition

From LON point of view the SRIO database consists of 500 network variables.

Network variables representing measurement values (AI-data) and indications (DI-data or EV-data) sent by the LON nodes are received and stored to the SRIO database.

If the node is a LON/SPA-gateway then a network variable must be defined using LON/SPA-gateway-commands. (ref. LON/SPA-gateway User's Manual). LON/SPA-gateway send the network variable update when network variable is defined to be sent using LON_BIND-command. LON_BIND-command defines the sender and the receiver of the data, service type of the network variable message delivery, the network variable (NV) selector and the address table entry to the sender node.

LON_DATA-command defines LON database items. The direction, data format and the receiver of the network variable are defined into the LON database. A network variable is identified by the selector value. Correctly defined item in SRIO 1000M database that matches to LON database item is updated with the network variable.

In table 1.8. is presented the different SRIO database data type definitions and updating of different type of network variables.

LON/SPA-gateway	SRIO 1000M database				
nv_type	DI	AI	DO	AO	EV
DI_nv	is updated	0	0	0	is updated
DI_nv, ev	is updated	0	0	0	is updated
AI_nv	0	is updated	0	0	0
DO_nv	0	0	0	0	0

Table 1.8. LON/SPA-gateway network variables - SRIO 1000M database. A zero means that the database value is not updated.

AI-data is updated when the change of the measurement value is greater than a delta value defined by user.

DI-data is updated each time the indication is changed.

EV-data is updated from the events sent by LON node.

The explicit messages containing time stamped events sent by LON nodes are received and stored to the SRIO event buffer.

1.8.1 LON configuration data and network management

LON interface of the SRIO 1000M is configured/programmed using commands of the same type as other SRIO commands. All the commands used for configuring LON interface can be given from the local programming terminal or via the ANSI interface from the host computer.

The LON configuration data includes:

- LON node table
  (a table defining LON nodes, ref. LON_INSTALL command)
- network variable configuration table
  (one entry for each SRIO 1000M data item, ref. LON_DATA command)

Note:
The configuration data of LON/SPA-gateways is kept only in the memory of the gateways not in the memory of the SRIO 1000M, except the unit list configuration.

The network management functions supported by SRIO commands include:

- node installation (ref. LON_INSTALL command)
- network variable binding (ref. LON_BIND command)

Other network management functions can be executed by L command.

1.8.2 Sending clock messages

The clock synchronization of LON interface can be selected with SYSPAR command. The default selection is that SRIO 1000M LON interface broadcasts clock synchronization messages to LON once per second to synchronize the real-time clocks of LON nodes to the real-time clock of the SRIO 1000M. In system with two SRIO 1000M it is possible to receive time synchronization messages from LON. If time synchronization is not needed, it can be totally ignored (no sending and no received allowed).

1.8.3 Data transfer

Transparent transfer of messages

The SPA-bus messages or other foreign ASCII or binary messages can be transferred transparently between SRIO 1000M application programs and LON. For example SPA-bus message received through the ANSI X3.28 interface and addressed to a SPA-bus slave connected LON via a LON/SPA-gateway is further routed to LON and a response to it is routed back to ANSI X3.28 interface. Also Z command or > command can be used to send SPA-bus messages to SPA-device connected via LON/SPA-gateways to LON. If > command is used, the slaves must be defined to SRIO 1000M UNIT list using UNIT command.

Data transfer between SPA-bus and LON

SRIO 1000M supports LONWORKS' feature to send data from a node to another node. Data can be sent from SPA-bus side to LON and vice versa. Data transferring from SPA-bus to LON or from LON to SPA- bus does not prevent data transfer from/to ANSI interface.

Data from SPA-bus to LON

If data is transferred from SPA-bus to LON, SRIO 1000M programming is made in special way. A data item is programmed as an input type or EV data in SPA-bus side. On LON side the item is defined as an output network variable with LON_DATA command, see example.

Example of definition when data is transferred from SPA-bus to LON:

DATA command:

1	2 41.I1	DI	BIN	FFFFH	1 1 0 1 0 0
2	2 41.I2	AI	DEC	0.02	1 1 0 1 0 0
3	2 41.1E1/2	EV	001	*004	
4	2 41.I1	DI	BIN	FFFFH	0 0 0 0 0 0

LON_DATA command:

LON_DATA 1=OUT,0022,1,0,4,1
LON_DATA 2=OUT,0023,4,1,4,1
LON_DATA 3=OUT,0024,1,0,4,1

Now e.g. data 41.I1 is polled from SPA-bus and value is sent on LON using network variable selector 22.

Data from LON to SPA-bus

If data is transferred from LON to SPA-bus in binary format then only one data can be programmed to one data table item. Data is transferred in the lsb of the byte. In hexadecimal format all 16 bits are used. On LON side an item is defined as an input network variable with LON_DATA command. On SPA-side the item is defined as an output type data, see example. If an input type network variable is defined as an output type data on SPA-side, SRIO 1000M does not poll the network variable from LON, but it sends data to SPA-bus after incoming network variable update message.

Example of definition when data is transferred from LON to SPA-bus:

LON_DATA command:

LON_DATA 5=IN,0025,1,0,4,1
LON_DATA 6=IN,0026,1,0,4,1
LON_DATA 7=IN,0027,1,0,4,1
LON_DATA 8=IN,0028,4,0,4,1

DATA command:

5	2 41.1V1	DO	BIN	0000	0 0 0 0 0 0
6	2 41.1V2	DO	BIN	0000	0 0 0 0 0 0
7	2 41.1V3	DO	BIN	0000	0 0 0 0 0 0
8	2 41.S21	AO	DEC	0.02	0 0 0 0 0 0

1.8.4 Programming examples

Changing the node bit rate

Bit rate can be changed by writing a new value to the Neuron chip's configuration table. This is made using L-command with message code 6EH (= write memory).

Command:

L <addr.type>,<node>,<sub>,6E=2,0,8,1,1,XX where parameters for write memory are:

2 addressing mode
0 offset msb
8 offset lsb
1 write one byte
1 both checksum recalculation
XX new "comm_clock:input_clock"-value

XX depends on the Neuron chip's input clock and defines the comm_clock and input_clock fields of the configuration table. Values with the input clock value 5 MHz and 10 MHz are given in the table 1.9.

Note: A node start communicating with the new bit rate after reset.

Input_clock	5 MHz	10 MHz
Bit Rate bit/s	value of the XX	
1,25 M	--	05
625 K	04	0D
312,5 K	0C	15
156,3 K	14	1D
78,1 K	1C	25
39,1 K	24	2D
19,5 K	2C	35
9,8 K	34	3D

Table 1.9. "Comm_clock:input_clock"-values.

Resetting a node

A node can be reset using software reset. This is done with L-command, message code 6CH (set node mode).

Command:

Reset a LON/SPA-gateway:
L <addr.type>,<node>,<sub>,6C=2, where "2" is an action code and has a meaning "reset_node".

Reset a SLTA:
L 0,0,0,6C=2

2.0 Commands

2.1 Command line interpreter

2.1.1 Connecting a display terminal

SRIO 1000M programming is done using a display terminal. The terminal can be connected to serial interface 4.

By switching the front panel switch 1 to ON position the communication parameters of the serial interface 4 are set to: 1200 b/s, 8 data bits, no parity, 1 stop bit.

The display terminal must be set to operate in half duplex or "local echo" mode, because SRIO 1000M does not echo the characters sent by the terminal.

After setting the terminal parameters and connecting the serial cable, SRIO 1000M should display a "-" -character, when the Return/Enter key of the terminal is pressed. This character is the SRIO 1000M command line interpreters prompt character. After displaying the prompt the SRIO 1000M is ready to receive commands entered from the terminal.

SRIO 1000M command line interpreter is able to receive commands also from serial interfaces 1, 2 or 3. In order to use these interfaces the mode of the interface must be set with BUS_MODE and SETUP command and the RS-232 / current loop selection must be set as required by the terminal.

When SRIO 1000M power is turned on or it is resetted or the mode of an interface is changed to TERMINAL mode then the following text is displayed:

```
NNNN    Strömberg     NNNN
NNNN    SRIO 1000M    NNNN
```

where N = the number of the serial interface (1...4).

Note:
The display terminal must recognize the Scandinavian characters according to the following list:

Scandinavian character	Hex code	Decimal code	The corresponding standard ASCII character	IMB PC code
Ä	5B	91	[	8EH 142
Ö	5C	92	\	99H 153
Å	5D	93	]	8FH 143
ä	7B	123	{	84H 132
ö	7C	124	|	94H 148
å	7D	125	}	86H 134

2.1.2 Entering commands

The general syntax of the commands is:

```
-<command> < > <parameters>
     |       |       |
     |       |       |
     |       |       Parameters for the command
     |       A space to separate the command and the parameters
     The command word
```

Example:

- SETUP 1

Every command line must be ended by pressing Return or Enter key. When pressing this key the terminal must send a CR -character (Carriage Return, 0DH) to the SRIO 1000M.

All the possible commands are displayed with a HELP command. Help for a specific command can be obtained with a H-parameter, for example:

- SETUP H

Typing errors can be corrected by moving the cursor backwards with BACKSPACE key and typing correct character on top of the erroneous character. Pressing BACKSPACE key must send code 08H to SRIO 1000M.

Commands and their parameters are usually typed with capital letters but also noncapitals can be used.

One command can be active only on one serial interface at a time.

If a command is entered from one interface and it is already active on another interface then the command line interpreter displays text: "Command reserved, wait a moment and try again".

If command word is unknown (typing error), the a question mark followed by the prompt is displayed: ? -

2.1.3 Example systems and programming

2.1.3.1 General example

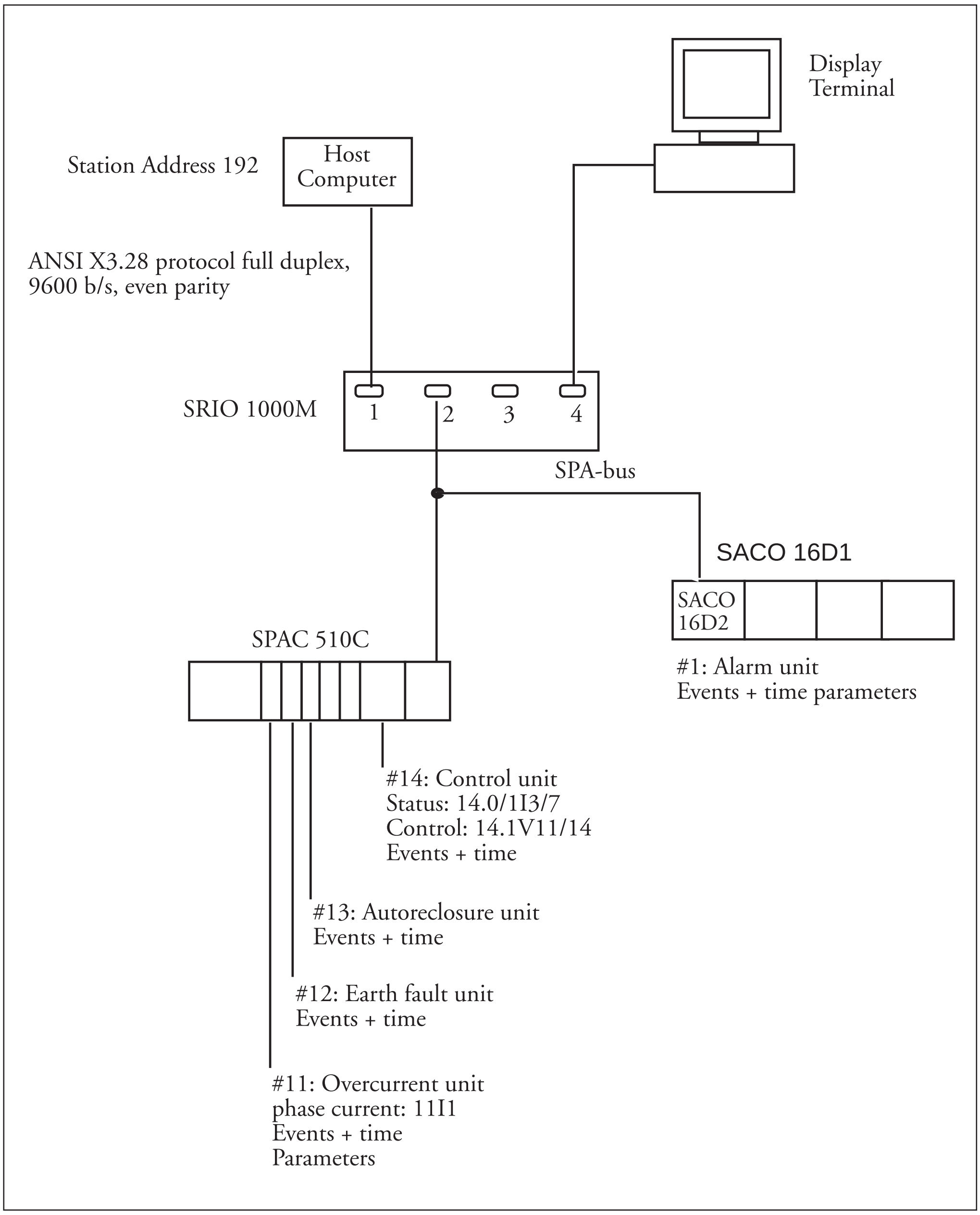


Figure 2.1. Small example system, which has been used in some of the programming examples.

2.1.3.2 Modbus interface

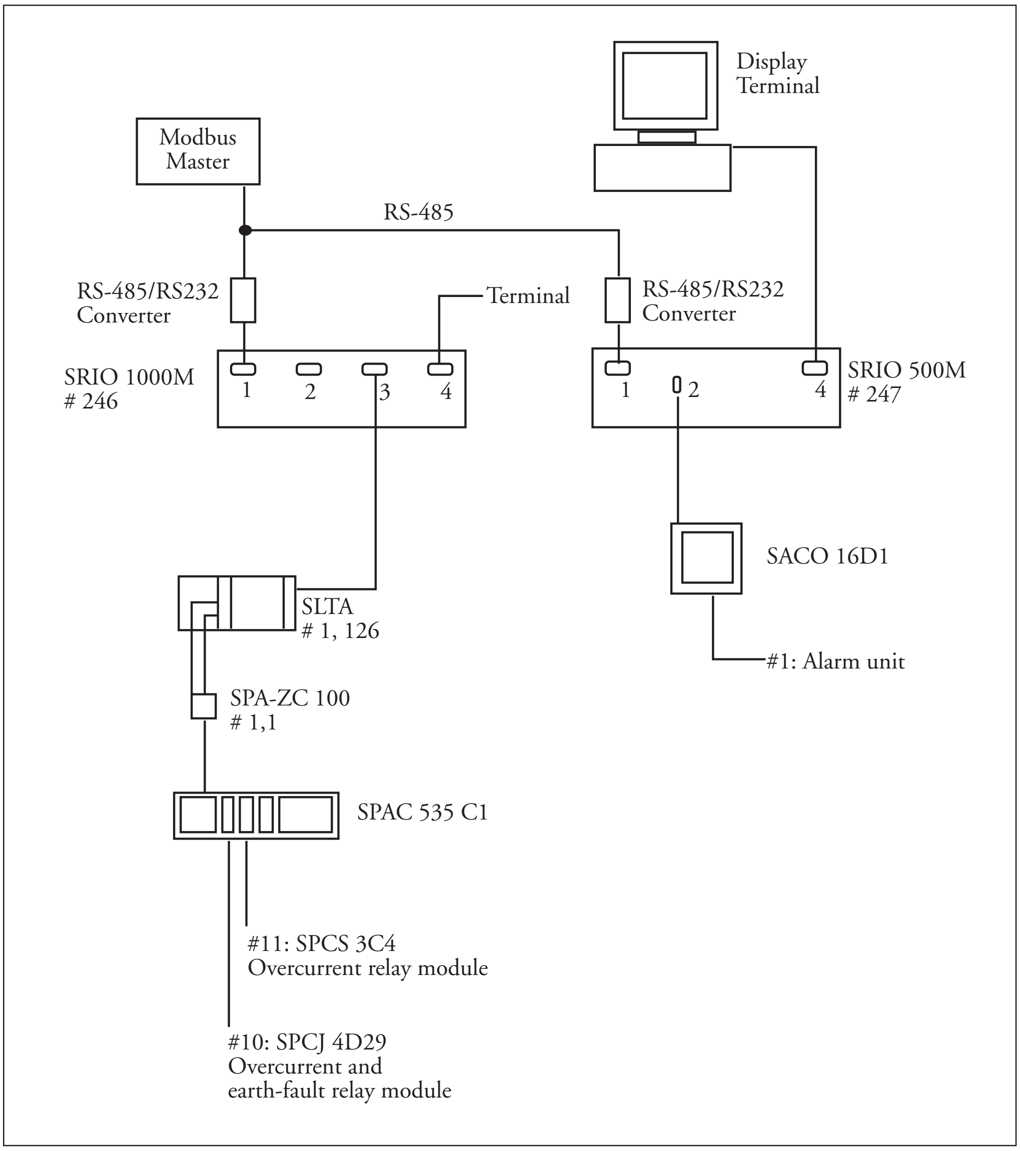


Figure 2.2. Example system, which has been used in the following programming example of Modbus interface.

An example configuration of SRIO 1000M:

Bus mode settings for SRIO 1000M

<bus>	<mode>	
1	11	(MODBUS Slave mode)
2	0	(NULL mode)
3	2	(LON SLTA mode)
4	10	(TERMINAL mode)

MODBUS_SETUP command used for setting the slave number and communication parameters.

slave no,	baudr,	parity,	CTS,	use,	use,	CRC
1..247	: 150	: 0=none	: delay	: CTS	: DCD	: order
	: 200	: 1=odd	:	: 0=no	: 0=no	: 0=LO/HI
slave	: 300	: 2=even	: 0...	: 1=yes	: 1=yes	: 1=HI/LO
station	: .	:	: 32767	:	:	:
number	: .	:	:	:	:	:
	: 9600	:	:	:	:	:

246,9600,2,0,0,0,0

Units programmed for the event poll list

```
-UNIT
3,10,2,1     SPCJ 4D29
3,11,2,1     SPCS 3C4

-DATA C 259

Index (1..500)
:     Bus (1..5)
:     :  Slave.C1/C2  Data Category Data 1/Data 2 (SPA-bus. addr.)
:     :  :            Data Type ( DI, AI, DO, AO, EV)
:     :  :            :   Data Format ( BIN, DEC, HEX) or Conversion table number
:     :  :            :   :
:     :  :            :   :       Delta or Mask ( dec or hex number (H))
:     :  :            :   :       :      Moveindex ( * + index)
:     :  :            :   :       :       Update Enable (0,1)
:     :  :            :   :       :       :  Continuous Poll Enable (0,1)
:     :  :            :   :       :       :  :  Timed Poll Enable (0,1)
:     :  :            :   :       :       :  :  :  Spont. Transf.Enable (0,1)
:     :  :            :   :       :       :  :  :  :  Temporary Error (0,1)
:     :  :            :   :       :       :  :  :  :  :  Permanent Error (0,1)
259  3  10S21         AO  DEC     0       1  0  0  0  0  0
260  3  10S22         AO  DEC     0       1  0  0  0  0  0
261  3  10S23         AO  DEC     0       1  0  0  0  0  0
262  3  10S24         AO  DEC     0       1  0  0  0  0  0
.
.
.
380  3  11S11         AI  DEC     0.001   1  1  1  1  0  0
381  3  11S12         AI  DEC     0.001   1  1  1  1  0  0
382  3  11S13         AI  DEC     0.001   1  1  1  1  0  0
383  3  11S14         AI  DEC     0.001   1  1  1  1  0  0
```

To configure data items used by LON interface to SRIO's database

```
- LD 380=IN,2,4,2,1,1
- LD 381=IN,3,4,2,1,1
- LD 382=IN,4,4,2,1,1
- LD 383=IN,5,4,2,1,1
```

Commands used to install SLTA and LON/SPA-gateway

```
-LON_INSTALL 126=126,1,0
-LI 1=1,1,2
```

LON units

```
LON_UNIT   1,  1,0, 10,2      protective relay SPCJ 4D29
LON_UNIT   1,  1,1, 11,2      protective relay SPCS 3C4

- LG 1 1 43=DA AI,0=11.0S11,0.01
- LG 1 1 43=DA AI,1=11.0S12,0.01
- LG 1 1 43=DA AI,2=11.0S13,0.01
- LG 1 1 43=DA AI,3=11.0S14,0.01
```

ref. LON/SPA gateway User's Manual

The configuration of SRIO 500M:

Used Bus modes

```
<bus>  <mode>
  1      11      (MODBUS Slave mode)
  2       4      (SPA Master mode)
  3       0      (NULL mode)
  4      10      (TERMINAL mode)
```

- MS (Modbus setup command)

```
slave no,  baudr, parity,    CTS,     use,     use,     CRC
1..247     :  150 : 0=none  : delay   : CTS    : DCD    : order
           :  200 : 1=odd   :         : 0=no   : 0=no   : 0=LO/HI
slave      :  300 : 2=even  : 0...    : 1=yes  : 1=yes  : 1=HI/LO
station    :    . :         : 32767   :        :        :
number     :    . :         :         :        :        :
           : 9600 :         :         :        :        :
247,9600,2,0,0,0,0
```

- UNIT

```
2,1,0,1      digital alarm unit SACO 16D1
```

- DATA C 500

```
Index (1..500)
:   Bus (1..5)
:   :  Slave.C1/C2 Data Category Data 1/Data 2 (SPA-bus. addr.)
:   :  :          Data Type ( DI, AI, DO, AO, EV)
:   :  :          :   Data Format ( BIN, DEC, HEX) or Conversion table number
:   :  :          :   :
:   :  :          :   :       Delta or Mask ( dec or hex number (H))
:   :  :          :   :       :      Moveindex ( * + index)
:   :  :          :   :       :        Update Enable (0,1)
:   :  :          :   :       :        :  Continuous Poll Enable (0,1)
:   :  :          :   :       :        :  :  Timed Poll Enable (0,1)
:   :  :          :   :       :        :  :  :  Spont. Transf.Enable (0,1)
:   :  :          :   :       :        :  :  :  :  Temporary Error (0,1)
:   :  :          :   :       :        :  :  :  :  :  Permanent Error (0,1)
500 2  1.1/16I1   DI  BIN     0000H    1  1  1  1  0  0
```

Detailed information about these commands can be found from this chapter and from LON/SPA gateway User's Manual.

2.2 General commands

HELP command

Abbreviation: H
Task: HELP command displays all the available commands.
Format: HELP

Note: No help text is available for HELP command.

VERS command

Abbreviation: V
Task: VERS command displays the version number of the SRIO 1000M software.
Format: VERS

Note: No help text is available for VERS command.

CLOCK command

Abbreviation: C
Task: CLOCK command can be used to display or set date, time and weekday of the SRIO 1000M internal real time clock.
Format: CLOCK <parameter>

<parameter> =		(No parameter, clock displayed.)
	W	(Weekday also displayed or set.)
	H	(Help.)
	F	(Timer fix value can be set.)

Remarks:
The command displays the date and time in the following format:
yy-mo-dd hh.mm:ss
or
yy-mo-dd hh.mm:ss Weekday

To leave the clock as is, press Return/Enter key. To change the clock, type new clock data and press Return/Enter. If you type any information (for example just the year and month) the remaining fields are left unchanged. Any delimiter character (e.g. space) can be used between the fields.

If you type a valid clock data line, the clock is set and the command line interpreter prompt is displayed. If the clock data line is invalid two question marks and the prompt is displayed: ?? -

When command is executed with F parameter, current time fix value is returned. To change the fix value, type a value in the range -1000...1000 and press Return/Enter. The fix value is used to set the timer count cycle. If the clock is running too slow the fix value should be given value between -1000...0. If the clock is running too fast, the value should be between 1000...0. One step in fix value changes the clock rate with a ratio of 1/20000 (about 4 seconds per day).

Example:
The following example changes the clock from "80-12-31 23.55:00 Tuesday to "88-10-12 12.10:10 Wednesday".

```
- CLOCK W
  80-12-31 23.55:00 Tuesday
  88 10 12 12 10 10 Wednesday
  -
```

Note: The weekday names must be given in format:
Monday, Tuesday, Wednesday, Thursday, Friday, Saturday or Sunday.

STORE command

Abbreviation: ST
Task: STORE command stores the setup and configuration data given by setup or programming commands to EEPROM memory.
Format: STORE <parameter>

<parameter> = (No parameters: check if ready.)
S (Start storing.)
F (Start Fast storing.)
B (Break storing.)
H (Help.)

Remarks:
Storing is started by typing "STORE S". After storing is started some other commands except setup and programming commands can be used. The storing takes about 10...20 minutes.

If storing is started by typing "STORE F", then only changed setup and configuration data are stored. In this way the storing usually takes only some seconds!

The progress of store operation can be checked by writing "STORE" with no parameters. If store is finished a text "Ready xxxxx bytes stored" is displayed.

If you want stop the storing for example to make some more setup or configuration changes, type "STORE B".

Example 1:
The following starts storing.
- STORE S
 Start 00000 bytes stored

Example 2:
The following checks if storing is ready.
- STORE
 Wait 00917 bytes stored

Removing stored data

To remove all data stored to EEROM memory do the following:

- change "EEROM-memory id-code" using SYSPAR command
- give RESET command
- give STORE command

By these actions the setup and configuration variables in EEROM memory and in RAM memory are reset or set to their default values.

To remove some data programmed with a programming command, there usually is a possibility to give the command with R or RA paramete

2.3 Serial interface setup commands

BUS_MODE command

Abbreviation: BM
Task: Displays and changes the operating modes of the serial interfaces.
Format: BUS_MODE <parameters>

Parameters:

H	(Help.	)
L	(List current bus modes.	)
C <X>	(Change mode of the bus X. X = 1...5, number of the serial interface, bus.)	

Remarks:
To display or change the mode of a bus, type BUS_MODE C X, where X is the number of the bus (serial interface). The command headerlines and the current bus mode of the specified bus are displayed. To leave the bus mode as it is, press Return/Enter. To change the bus mode type new bus mode number.

Example:
Change mode of serial bus 3 from SACO 100M Master mode to TERMINAL mode.

```
- BUS_ MODE C 3
 <bus   modes>
   0    = NULL mode
   1    = ANSI slave mode          (bus 1 only)
   2    = LON SLTA mode            (bus 3)
   3    = reserved
   4    = SPA Master mode
   5    = SPA Slave  mode
   6    = Fast SPA Master mode   (bus 2 only)
   7    = SACO 100M Slave multidrop mode
   8    = SACO 100M Master mode
   9    = SACO 100M Slave mode
  10    = TERMINAL mode
  11    = MODBUS Slave mode  (bus 1,3 or 4)

 <bus> <mode>
   1    8   (SACO 100M Master mode)
   1    10
   -
```

Note:
To program the serial interface 1 to TERMINAL mode give the following commands: BUS_MODE C 1 = 10, SYSPAR 1 = 0, SYSPAR 2 = 0 and ANSI_SETUP = 0,....

Note:
Only one bus can be programmed in SPA-bus mode, usually it is bus number 2.

SETUP command

Abbreviation: SE
Task: SETUP command is used to display or change the communication parameters of serial interfaces.
Format: SETUP <parameter>
<parameter> = 1...5 (Serial interface number.)
H (Help.)

Remarks:
The command headerlines and the current setup parameters are displayed.

To leave the parameters as they are, press Return/Enter. To change parameters, type new parameter values separated with commas. If you only type a few new values (for example only mode and baudrate), the remaining parameters are left unchanged.

If you type valid parameters, then the parameters of the interface are changed, the interface is resetted and the command line interpreter prompt is displayed.

If you type invalid parameters, then the two question marks followed by the name of the parameter and the prompt are displayed.

If <cts> or <dcd> parameter is set, then CTS or DCD handshaking signal is used.

If <aut.lf> parameter is set, then SRIO 1000M sends a LF character after every CR character.

<resend> parameter determines the maximum number of times a message is resent in error situations.

Note: By switching the front panel switch 1 to ON position the communication parameters of the interface 4 are set to: text mode, 1200 b/s, 8 data bits, no parity, 1 stop bit.

Note: In versions older than V4.0 the SETUP command is also used to set the operating mode of the serial interface.

The available modes are the following:
Mode 0 = Serial interface is not used
Mode 1 = Master of "SACO 100M interface protocol".
Mode 2 = Slave of "SACO 100M interface protocol".
Mode 3 = Master of SPA-bus.
Mode 4 = Slave of SPA-bus.
Mode 5 = Text mode.

If serial interface is in text mode, then the SRIO 1000M command line interpreter can be given commands using the interface.

Example 1:
The following changes the baudrate of serial interface 2 from 1200 to 9600.

```
- SETUP 2

<baud>,   <parity>,  <sbits>, <cts>,  <dcd>,  <aut.lf>,   <timeo>,     <res>
  300     : 0=none   :  1     :  0 = not used             : 1...       : 0...10
 1200     : 1= odd   :  2     :  1 = used                 : 32767ms    :
 2400     : 2=even   :        :                           :            :
 4800     :          :        :  Notice:                  : Notice:
 9600     :          :        :  <aut.lf> only in         : <timeout> and
          :          :        :  mode 5.                  : <resend> not
          :          :        :                           : in mode 5.
1200,0,1,0,0,1,0,0
9600
-
```

Example 2:
The following shows what is displayed if an invalid baudrate is given.

```
- SETUP 2

<baud>,   <parity>,  <sbits>, <cts>,  <dcd>,  <aut.lf>,   <timeo>,     <res>
  300     : 0=none   :  1     :  0 = not used             : 1...       : 0...10
 1200     : 1= odd   :  2     :  1 = used                 : 32767ms    :
 2400     : 2=even   :        :                           :            :
 4800     :          :        :  Notice:                  : Notice:
 9600     :          :        :  <aut.lf> only in         : <timeout> and
          :          :        :  mode 5.                  : <resend> not
          :          :        :                           : in mode 5.
1200,0,1,0,0,1,0,0
2500
?? <baudrate>
-
```

ANSI_SETUP command

Abbreviation: AS
Task: ANSI_SETUP command is used to display or change the host interface communication parameters.
Format: ANSI_SETUP <parameter>
<parameter> = (No parameter, normal operation.)
H (Help.)

Remarks:
The command headerlines and the current setup parameters are displayed. To leave the parameters as they are, press Return/Enter. To change parameters, type new parameter value separated with commas. If you type only some new values (for example only mode and own number), the remaining parameters are left unchanged. If you type valid parameters, then the parameters of the interface are changed and the command line interpreter prompt is displayed. If you type invalid parameters, then two question marks followed by the name of the parameter and the prompt are displayed.

Only <own num>, <host num> and <CTS delay> parameters are taken in use immediately. If some other parameters are changed store the new parameters with STORE command and after store is complete reset the device.

Parameter "mode" sets the interface to half duplex of full duplex mode. "Own num" defines the station number of the SRIO. "Host num" defines the station number of the ANSI X3.28 master device.

"baudr" defines the used baudrate, valid values are: 50, 75, 110, 150, 200, 300, 600, 1200, 2400, 4800 and 9600. "check" defines the used error checking method. "parity" defines the use of parity checking. "CTS delay" defines the delay after CTS handshaking signal is valid until transmission is started. If "emb. resp" is 1, short response messages from SRIO can be embedded in data messages. "RTS keepup" defines the number of null characters sent after the end of the message to keep the RTS handshaking signal active after message is sent. If "Multi event" is 1 then AI, DI and EV data and events are sent using so called multi event format (ref. chapter "Process data transfer" and "Event data transfer").

Example:
The following changes the own station number and host station number to new values.

```
- ANSI_SETUP

mode, own num,   host num, baudr., check,   parity,   CTS,    embedded, RTS,          Multi
0=hd: 1..255     :1...255  :    50 : 0=CRC : 0=none :delay  : response :"keepup"      :event
1=fd  :          :         :   110 : 1=BCC : 1=odd  :       :          :              :
      :          :         :   .   :       : 2=even :1...   : 0=no     : Number of    : 0=no
      :          :         :   :   :       :        :32767: 1=yes      : extra chars  : 1=yes
      :          :         : 9600  :       :        :msec              :              : 0...9
1,21,193,9600,1,2,300,1
1,4,5
-
```

MODBUS_SETUP command

Abbreviation:	MS
Task:	MODBUS_SETUP command is used to display or change the Modbus communication parameters.
Format:	MODBUS_SETUP <parameter>

<parameter> = (No parameter, normal operation.)
H (Help.)

Communication parameters in the table have following meanings:

<slave no> = address of the SRIO device in Modbus network, 1-247

<baudr> = baudrate of serial communication
150, 200, 300, 600, 1200, 2400, 4800, and 9600

<parity> = odd, even or no parity can be used
0 = none
1 = odd
2 = even

<CTS delay> = delay between activating CTS line by modem and starting transmitting response by SRIO. Valid values are from 0 to 32767.

<CTS use> = use of CTS line, supported by port 1
0 = no
1 = yes

<DCD use> = use of DCD line
0 = no
1 = yes

<CRC order> = order of CRC bytes in the message frame.
0 = Low order byte/High order byte
1 = High order byte/Low order byte

Remarks:
The command first displays the current parameters. To leave parameters as they are just press Enter/Return. If changes are needed, type parameter values and separate them by commas. If only some new values from the beginning are typed, then the remaining parameters are left unchanged. If parameters were valid then by using STORE and RESET commands in the aforementioned order values are stored to EEPROM. If there was an error/s in the parameters typed two question marks and incorrect parameter will be produced.

Example:
The slave address of SRIO is changed from 240 to 2 and baudrate is changed to 9600, other parameters are kept the same.

```
- MODBUS_SETUP

slave no,       baudr,  parity,   CTS,     use,    use,    CRC
1..247          :  150  : 0=none  : delay  : CTS   : DCD   : order
                :  200  : 1=odd   :        : 0=no  : 0=no  : 0=LO/HI
slave           :  300  : 2=even  : 0...   : 1=yes : 1=yes : 1=HI/LO
station         :    .  :         : 32767  :       :       :
number          :    .  :         :        :       :       :
                : 9600  :         :        :       :       :
240,1200,2,0,0,0,0
2,9600
-
```

Note:
By using the default value for CRC order the low order byte is the first CRC byte to be send in the message frame and the last byte of the message frame is the high order CRC byte.

2.4 Commands to set the general operating parameters

SYSPAR command

Abbreviation: SY
Task: SYSPAR command displays and changes some system parameters.
Format: SYSPAR <parameter>
<parameter> = (Start from parameter number 1 .)
N (Start from parameter number N.)
H (Help.)

Remarks:
The current parameters are displayed one by one. To leave the displayed parameter as it is, press only Return/Enter. To change the parameter, type in a new parameter value and press Return/Enter. To abort the command, type . and press Return/Enter.

Command is completed, when all the parameters have been displayed or when it is aborted by typing '.'..

P1: Spontaneous event enable = 1

New value ? (0...1)

Enables or disables the spontaneous sending of events to the host computer using ANSI X3.28 protocol.

P2: Spontaneous data enable = 1

New value ? (0...1)

Enables or disables the spontaneous sending of changed data to the host computer using ANSI X3.28 protocol.

P3: Periodical data poll cycle = 60

New value ? (30...30000 seconds)

The poll cycle used to data items, which have the "Timed poll enable" flag set (refer to DATA command). The same cycle is used for continuously polled data items if their polling has failed two times.

P4: SPACOM slave number for SRIO = 901

New value ? (1...999)

The slave number of SRIO 1000M, if SRIO 1000M is used as a slave device in SACO 100M protocol.

P5: Event sort delay = 1000

New value ? (1...30000 milliseconds)

Delay before a new event is stored to event buffer and sorted in time order. If the events should be delivered to the host computers or printed to event printer in time order, then this delay should be longer than two times the event poll cycle of the system.

P6: EEROM-memory ready flag = 1

Changing not possible. Press Return/Enter.

If this flag is 1, then the device was started with setup and configuration data from EEROM memory. If flag is 0, then the device was started using default setup and configuration data.

P7: Battery RAM ready flag = 1

Changing not possible. Press Return/Enter.

If this flag is 1, then the event buffer was started with the data in Battery RAM.

P8: EEROM id-string = CONFIGURATION_DATA_STORED

If the string is changed, the configurations will be lost during the next device reset !

To change the string, type any character.

To leave it as it is, press only Return/Enter. ?

To destroy all the programmed data from SRIO 1000M's EEROM memory do the following:
- change "EEROM id-string" with SYSPAR command
- give RESET command
- give STORE command
- wait until storing is ready

P9: Battery RAM id -string

=TEST_TEXT1234567890zzzzzUUUUUUUUUUwwwwwwwwwwpppppppppp

If the string is changed, the contents of the event buffer will be lost during the next reset!

To change the string, type any character. To leave it as it is, press only Return/Enter. ?

To destroy all the data from SRIO 1000M's Battery RAM memory do the following:
- change "Battery RAM id-string" with SYSPAR command
- give RESET command

P10: Acceptable Ansi Reply STS (1) = 64

New value ? (0...255)

Acceptable STS byte value in ANSI X3.28 protocols reply message in addition to STS byte value 0.

P11: Acceptable Ansi Reply STS (2) = 80

New value ? (0...255)

Acceptable STS byte value in ANSI X3.28 protocols reply message in addition to STS byte value 0.

P12: Acceptable Ansi Reply STS (3) = 0

New value ? (0...255)

Acceptable STS byte value in ANSI X3.28 protocols reply message in addition to STS byte value 0.

P13: Auto poll interval = 500

New value ? (500...10000 milliseconds)

Refer to chapter "Data acquisition principle"/ "Auto poll".

P14: Auto poll count = 5

New value ? (1...10)

Refer to chapter "Data acquisition principle"/ "Auto poll".

P15: Optimise sending of digital data = 1

New value ? (0...1)

If this parameter is 1: If the value of a data item containing DI-data is changed more often than it can be sent to the host, then all the changes of the data item are not sent to the host computer using ANSI X3.28 protocol.

P16: Optimise sending of analog data = 1

New value ? (0...1)

If this parameter is 1: If the value of a data item containing AI-data is changed more often than it can be sent to the host, then all the changes of the data item are not sent to the host computer using ANSI X3.28 protocol.

P17: Digital data high priority enable = 1

New value ? (0...1)

If this parameter is 1, then the DI-data has higher priority than AI-data in ANSI X3.28 interface.

P18: Type of hardware synchronization in/out = 0

(0 = not in use, 1 = sync out, 2 = sync in)

New value ? (0...2)

This parameter is used to define the use of the real-time clocks hardware synchronization possibility:
0 = no hw-synchronization is used
1 = SRIO 1000M gives a hw-synchronization pulse once per minute. This pulse can be used to synchronize other SRIO 1000M devices.
2 = SRIO 1000M is programmed to receive a hardware synchronization pulse once per minute from another device.

P19: Event poll / Data poll ratio = 5/1

Valid ratios (1/5, 2/4, 3/3, 4/2, 5/1)

New value for ratio ?

If this parameter has been given value n/m, then the event acquisition program sends messages to the SPA-bus in bursts of n messages and the data acquisition program sends messages in bursts of m messages.

When this parameter is given different values the systems response time for data updated from events or for cyclically updated data can be changed. To get the best possible response time for events, give value 5/1 for this parameter.

P20: Line idle time before next message is transmitted = 5 ms

New value? (1ms..10ms)

There are some SPA-bus slave units which are not able to receive a new message immediately after sending a reply message. To avoid communication errors with these slaves a small delay must be generated before a message is sent from SRIO 1000M to the SPA-bus.

This parameter can be used to insert a delay of 1...10 milliseconds before a message is sent.

P21: Enable control commands from host computer using ANSI X3.28 protocol also in local mode = 0

New value? (0..1)

Normally the SRIO 1000M is put to local mode to disable control commands from the host computers. This parameter can anyhow be used to enable the control commands also in local mode.

P22: ANSI Eventbased data retransmissions = 0

New value ? (0...999)

If the sending of a EV-data message to the ANSI X3.28 host fails, then the message is resent by the communication program automatically 3 times. If all communication programs retransmissions fail, then a new EV-data message is built and sent as many times as defined by this parameter.

P23: ANSI Digital data retransmissions = 0

New value ? (0...999)

If the sending of a DI-data message to the ANSI X3.28 host fails, then the message is resent by the communication program automatically 3 times. If all communication programs retransmissions fail, then a new DI-data message is built and sent as many times as defined by this parameter.

P24: ANSI Analog data retransmissions = 0

New value ? (0...999)

If the sending of an AI-data message to the ANSI X3.28 host fails, then the message is resent by the communication program automatically 3 times. If all communication programs retransmissions fail, then a new AI-data message is built and sent as many times as defined by this parameter.

P25: Slave priority of digital alarm units = 0

New value ? (0 = normal, 1 = high)

If priority is high, these units will be polled in high priority event poll cycle (see UNIT command).

P26: Slave priority of analog alarm units = 0

New value ? (0 = normal, 1 = high)

If priority is high, these units will be polled in high priority event poll cycle (see UNIT command).

P27: Slave priority of relay units = 0

New value ? (0 = normal, 1 = high)

If priority is high, these units will be polled in high priority event poll cycle (see UNIT command).

P28: Slave priority of control units = 1

New value ? (0 = normal, 1 = high)

If priority is high, these units will be polled in high priority event poll cycle (see UNIT command).

P29: Time synchronization of LON-bus = 0

0 = sync out, SRIO broadcasts clock synchronization messages to LON to synchronize the real-time clocks.
1 = sync in, time sync message sent by LON, used in systems with two or more SRIOs
2 = not in use

New value ? (0...2)

P30: Subnet number of LON-bus time sync. = 0

0 = domain broadcast, clock sync. is sent to all nodes in all subnets
1..255 = subnet broadcast, clock sync. message is sent to nodes in the specified subnet

New value ? (0...255)

P31: Delay of LON-bus time synchronization = 1

Defines how often the clock sync. message is broadcast to the network. Default value is once per second.

New value ? (1...30 seconds)

P32: Extended SACO 100M event timestamp = 0

0 = not used

1 = extended event timestamp is used

If yes, then the events transmitted via the SACO 100M protocol will have a timestamp consisting of date and time information.

Example
>901RL:CC<cr>
<901D:**97-05-15** 12.32;32.123 56E037...

SIGNAL command

Abbreviation: SI
Tasks: Programming of the SRIO hardware inputs (I1...I7). Sending of signal messages to alarm system slaves (slave type 0 or 1).
Format: SIGNAL <parameters>

Parameters:

PRG :	Programming of inputs.	
RES :	Send reset message	(>nWV100:1:CCcr)
ACK :	Send acknowledgment message	(>nWV3:1:CCcr)
SIL :	Send silence message	(>nWV2:1:CCcr)
TON :	Send test on message	(>nWV4:1:CCcr)
TOF :	Send test off message	(>nWV4:0:CCcr)
ION :	Send ilock on message	(>nWV11:1:CCcr)
IOF :	Send ilock off message	(>nWV11:0:CCcr)
LOC :	Send local message	(>nWV1:2:CCcr)
REM :	Send remote message	(>nWV1:3:CCcr)
L/R :	Send local/remote message	(>nWV1:1:CCcr)

Example:
The following resets the alarm unit alarms:

```
- SIGNAL RES
  -
```

Example:
The following programs the SRIO 1000M input 1 to be used as the LOCAL input of the SRIO 1000M.

```
- SIGNAL PRG

SIGNAL NAME:         SILENCE
                     :  ACKNOWLEDGMENT
                     :  : RESET
                     :  :  :  TEST
                     :  :  :  :  LOCAL
                     :  :  :  :  :  REMOTE
                     :  :  :  :  :  ILOCK
INPUT NUMBER:        0,0,0,0,0,0
NEW ASSIGNMENT >     0,0,0,0,1,0
-
```

Note:
Input I7 can also be programmed as clock sync. input by SYSPAR 18.

Note:
The local/remote switch of the substation can be wired to the hardware inputs (I1...I7) of SRIO or to the inputs of a slave device like SACO or SPOC. In the first case SIGNAL PRG is used to program the operation of the SRIO. In the latter case the LOCAL_REMOTE command must be used.

LOCAL_REMOTE command

Abbreviation: LR
Task: Displays and changes the database data item and its value, which is used to set the device in local mode.
Format: LOCAL_REMOTE

Function:
The command displays <data ind.> is the data index, which is used to read the local/remote state of the device, possible values are 0...500. If value 0 is given the local/remote state is not read from the database. The command also displays <value>, which is the data value, which corresponds to the local state (0..32000). When command is activated also the current state is displayed.

Example:
-LOCAL_REMOTE<data ind.>,<value>, State is REMOTE. (Type CR to exit.)
2,1
3,0 (This must be typed in.)
3,0 State is LOCAL.
Now you can exit the cmd by typing CR or make another change of the parameters.
-

Note:
When the mode is LOCAL, then the frontpanel TEST LED is flashing (1 s. period).

Note:
If some SRIO 1000M hardware inputs are programmed as LOCAL or REMOTE inputs with SIGNAL PRG-command, then the definition made with LOCAL_REMOTE-command is not used.

2.5 Commands to program the event and data acquisition

Event acquisition from SPA-bus

To get the events from the units connected to SRIO 1000M with SPA-bus, the units must be programmed to SRIO 1000M's event poll list using

UNIT command.

The unit number or "SPACOM slave number" for SRIO 1000M is programmed using SYSPAR command. This number is used e.g. in the event which is generated when SRIO 1000M's event buffer overflows.

The "Event sort delay" is programmed using SYSPAR command.

Example:
In the example system of figure 2.1 the event poll list would be programmed so that the UNIT command would produce the following listing.

-UNIT L
2,1,0,1
2,11,2,1
2,12,2,1
2,13,2,1
2,14,3,1
-

Event acquisition from the slave units using SACO 100M protocol

To get events with time markings from the SACO 100M, the SACO 100M must be programmed with SACO 100M's UNIT command to poll events from the slave units connected to it. SRIO 1000M polls event from SACO 100M automatically, if the mode of the used SRIO 1000M serial interface is "Master of SACO 100M interface protocol".

The numbers of the SACO 100M units connected to the SRIO 1000M must be as follows:

SRIO 1000M serial interface 2: SACO 100M number 902
SRIO 1000M serial interface 3: SACO 100M number 903
SRIO 1000M serial interface 4: SACO 100M number 904

The number of SACO 100M can be changed using SRIO 1000M

Z command.

Note:
Remember to store SACO 100M parameters using SACO 100M STORE command after changing the number.

Data acquisition

The status information, measurement data and control output data are defined as data items to the SRIO 1000M database.

The data item definitions are made with DATA command.

The periodical data poll cycle is programmed with SYSPAR command.

To get use of the SRIO 1000M's auto poll function the related data items are grouped together to data groups.

The data groups are defined with DATA_GROUP command.

The auto poll interval and count are programmed with SYSPAR command.

Example:
In the example system of figure 2.1 the data items and data groups would be programmed so that the DATA and DATA_GROUP commands would produce the following listings.

Note:
The first data item definition is needed only to enable the transfer of parameter data from alarm unit 1.

```
- DATA L 1 5

Index (1..500)
:     Bus (1..5)
:     :  Slave.C1/C2 Data Category Data 1/Data 2 (SPA-bus. addr.)
:     :  :            Data Type ( DI, AI, DO, AO, EV)
:     :  :            :   Data Format ( BIN, DEC, HEX) or Conversion table number
:     :  :            :   :
:     :  :            :   :     Delta or Mask ( dec or hex number (H))
:     :  :            :   :     :     Moveindex ( * + index)
:     :  :            :   :     :       Update Enable (0,1)
:     :  :            :   :     :     : Continuous Poll Enable (0,1)
:     :  :            :   :     :     : : Timed Poll Enable (0,1)
:     :  :            :   :     :     : : : Spont. Transf.Enable (0,1)
:     :  :            :   :     :     : : : : Temporary Error (0,1)
:     :  :            :   :     :     : : : : : Permanent Error (0,1)
1     2  1.1/16I1     DI  BIN   0000H 0 0 0 0 0 0
2     2  13I1         AI  DEC   0.03  1 1 0 1 0 0
3     2  14.0/1I3/7   DI  B     FC1CH 1 1 0 1 0 0
4     2  14.1V11/14   DO  B     0000H 0 0 0 0 0 0
-
```

```
- DATA_GROUP L 1 2
1  2,3,4
2
-
```

UNIT command

Abbreviation: U
Task: UNIT command is used to program the event poll list. The list defines the SPA-bus unit which are polled for events by SRIO 1000M.
Format: UNIT <parameter>

<parameter> = D (Define.)
L (List.)
L P (List in page mode.)
H (Help.)
RA (Remove all.)

Remarks:
With D-parameter the command displays the following header line:
<bus>,<unit>,<type>,<event enable>

After the header line you can enter one or more unit definitions. The definition can be used to change or remove an existing definition of a unit or to define a completely new definition.

To abort from the command only Return/Enter is typed.

The definition contains:

<bus> = the number of the serial interface to which the SPA-bus is connected (1...5)

<unit> = the unit (or slave) number of the SPA-bus unit or SPACOM unit (1...999)

<type> = the type of the unit:
0: digital alarm unit of type SACO 16D
1: analog alarm unit of type SACO 16A
2: protective relay unit
3: control unit (e.g. of type SPTO or SPOC)
100: data communicator of type SACO 100M

<event enable> = the event enable/disable/remove parameter
0: the unit is defined but not polled
1: the unit is polled for events
9: the unit is removed from the event poll list

When L parameter is given, all the units in the event poll list are listed in format: <bus>,<unit>,<type>,<event enable>,<status>.

If SRIO 1000M does not succeed to communicate with the unit, then the <status> is the following text:

"COMMUNICATION ERROR".

Note:
The priority (high or normal) of each slave type can be set with SYSPAR 25...SYSPAR 28. Usually the priority of the units of type 3 is high and priority of other types is low. This improves the systems response time, when the state of the breaker or isolator changes.

Note:
The maximum number of unit definitions is 100.

Note:
All the units (slaves) of the SPACOM system must have a unique unit number.

Example:
The following adds unit 123 on serial interface 2 to the event poll list.

- UNIT D
<bus>,<unit>,<type>,<event enable>
2,123,1,1
-

DATA command

Abbreviation: DA

Task: DATA command defines, lists, changes or removes data item definitions. Data item definitions define what data is polled from the SPACOM system to the SRIO 1000M database. (See chapter 3.1.)

Format: DATA <parameters>

<parameters> = D (Define a new data item.)
H (Help.)
C <I> (Change definition of data item I.)
R <I> <J> (Remove data item definitions I to J.)
L <I> <J> (List data item definitions I to J.)
L <I> <J> P (List data item definitions I to J. The listing stops after every page. Listing continues when any key is pressed.)
<I>: 1...500, index of the first item
<J>: 1...500, index of the last item
B (List with database values.)
E (Reset error status.)

Remarks:
When command is activated with C or D parameter, some headerlines are first displayed to point the starting columns for the different fields of data item definition. Every field is started from the defined column and the space between the fields is filled with spaces.

With D-parameter: After the headline, a new data item definition can be entered. If the entered definition is accepted, it is displayed with the given data index and you can enter the next definition. If it is not accepted two question marks are displayed and you can try again.

With C-parameter: The headerlines and the existing definition of data item <I> are displayed. To change the definition, type new definition line and press Return/Enter. If the given definition is accepted, it and the next definition are displayed. If it is not accepted, two question marks are displayed and you can try again. After entering an accepted definition you can continue by changing the next definition which is displayed together with the accepted one. Both Define and Change functions are aborted, if only Return/Enter is entered instead of a whole definition line.

One data item definition contains the following:

Data index:
Number 1...500 given to each data item.

Bus:
SRIO 1000M serial interface 1 to 4 through which defined data is available.

SPACOM address:
The address of SPACOM data according to SPA-bus protocol:
(Slave number, Channel number(s), Data category, Data number(s))

Data type:
DI = digital input data
DO = digital output data
AI = analog input data
AO = analog output data
EV = data updated from events

Data format or Conversion table number:
The format in which data is read or written from or to SPACOM unit.

DI-data:	BIN or HEX
DO-data:	BIN
AI-data:	DEC
EV-data:	the number of the conversion table, which is used to convert the defined events and corresponding "direct" data values before storing to the database

Delta or Mask or Moveindex:

For AI/AO-data:	The delta value that is used as criteria with database updating and spontaneous sending of data to the host. Minimum delta value is 0.001.
For DI/DO-data:	A hexadecimal bit mask. It disables the spontaneous sending of the bits, which have the mask bit set.
For EV-data:	The data index of the corresponding "direct" data definition.

Update enable:
1 = data item is polled
0 = data item is not polled

Continuous poll enable:
1 = data item is polled with maximum speed
0 = data item is not polled continuously

Timed poll enable:
1 = data item is polled periodically
0 = data item is not polled periodically

Spontaneous transfer enable:
1 = data item is spontaneously sent to the host
0 = data item is not sent spontaneously

Temporary error:
Temporary communication error.

Permanent error:
Permanent communication error.

Note:
With EV-data instead of the Update enable/Continuos poll enable....-parameters the command displays spaces or error code:

Error 1: E-code could not be converted. (Not found in DATA_CONVERSION tables.)

Error 3: Direct data not found in DATA_CONVERSION tables or direct data not of same format as declared.

Note:
With both Define and Change functions the characters from the preceding definition line are copied to the new definition, if "_" -characters are typed to the new line.

Note:
Data bits which can be addressed by one SPACOM message can be defined to one digital data item. This includes data which belongs to one slave, one data category, successive channel numbers and successive data numbers.

If digital data item is used to read data from SPACOM unit, the data item definition must be done so that all the bits of the data item can be read by one message.

If digital data item is only used to write data to SPACOM unit, it is not necessary that all the bits can be read or written by one SPACOM message, because only one bit is written at a time. It is anyhow necessary that all the bits can be addressed (read or written in theory) by one SPACOM message.

Note:
After defining the necessary data item definitions, clear the unused portion of the data item definition table with DATA command using R-parameter.

Example:
If the system contains data item definitions 1...50, clear the unused table as follows:
-DATA R 51 500
-

Note:
When a data item is removed (DATA R) then the ANSI address of the data item is set to 77777Q.

Example 1:
The following defines two new data items.

Data item 1 is used to read the current of the feeder line from overcurrent relay.

Data item 2 is used to read the status information from the control unit SPTO 1C1 as normal digital input data. This kind of data item would be used to get the status of the truck, breaker and earthing switch in compact bit format.

Data items 3..5 are used to convert events from SPTO 1C1 to analog status data (EV data). Data items 6..8 are used to read the EV data using direct read from the slave. These data definitions are needed to get the status data of each object in analog format with time stamp.

Note:
Data item 2 and data items 3...8 are two different alternatives to get the status data to the host computer. Usually only one of the alternatives is used.

```
-DATA D

Index (1..500)
:     Bus (1..5)
:     :  Slave.C1/C2 Data Category Data 1/Data 2 (SPA-bus. addr.)
:     :  :             Data Type ( DI, AI, DO, AO, EV)
:     :  :             :   Data Format ( BIN, DEC, HEX) or Conversion table number
:     :  :             :   :
:     :  :             :   :     Delta or Mask ( dec or hex number (H))
:     :  :             :   :     :      Moveindex ( * + index)
:     :  :             :   :     :        Update Enable (0,1)
:     :  :             :   :     :        :  Continuous Poll Enable (0,1)
:     :  :             :   :     :        :  :  Timed Poll Enable (0,1)
:     :  :             :   :     :        :  :  :  Spont. Transf.Enable (0,1)
:     :  :             :   :     :        :  :  :  :  Temporary Error (0,1)
:     :  :             :   :     :        :  :  :  :  :  Permanent Error (0,1)
      2  11I1          AI  DEC   0.02     1  1  0  1  0  0
1     2  11I1          AI  DEC   0.02     1  1  0  1  0  0
      2  14.0/1I3/7    DI  BIN   FC1CH    1  1  0  1  0  0
2     2  14.0/1I3/7    DI  BIN   FC1CH    1  1  0  1  0  0
      2  14.1E1/3      EV  001   *006
3     2  14.1E1/3      EV  001   *006
      2  14.1E4/6      EV  001   *007
4     2  14.1E4/6      EV  001   *007
      2  14.0E1/3      EV  001   *008
5     2  14.0E1/3      EV  001   *008
      2  14.1I1        AI  DEC   1        0  0  0  0  0  0
6     2  14.1I1        AI  DEC   1        0  0  0  0  0  0
      2  14.1I2        AI  DEC   1        0  0  0  0  0  0
7     2  14.1I2        AI  DEC   1        0  0  0  0  0  0
      2  14I1          AI  DEC   1        0  0  0  0  0  0
8     2  14I1          AI  DEC   1        0  0  0  0  0  0
-
```

Example 2: The following changes the data item 1 to be polled continuously and sent spontaneously to the host.

```
-DATA C 1

Index (1..500)
:   Bus (1..5)
:   :  Slave.C1/C2 Data Category Data 1/Data 2 (SPA-bus. addr.)
:   :  :          Data Type ( DI, AI, DO, AO, EV)
:   :  :          :   Data Format ( BIN, DEC, HEX) or Conversion table number
:   :  :          :   :
:   :  :          :   :     Delta or Mask ( dec or hex number (H))
:   :  :          :   :     :     Moveindex ( * + index)
:   :  :          :   :     :       Update Enable (0,1)
:   :  :          :   :     :       :  Continuous Poll Enable (0,1)
:   :  :          :   :     :       :  :  Timed Poll Enable (0,1)
:   :  :          :   :     :       :  :  :  Spont. Transf.Enable (0,1)
:   :  :          :   :     :       :  :  :  :  Temporary Error (0,1)
:   :  :          :   :     :       :  :  :  :  :  Permanent Error (0,1)
1   2  11I1       AI  DEC   0.02    1  0  0  0  0  0
    ____________________________1  1  0  1  0  0
1   2  11I1       AI  DEC   0.02    1  1  0  1  0  0
2   2  14.0/1I3/7 DI  BIN   FC1CH   1  1  0  1  0  0
-
```

Example 3: The following lists the data item definitions 1...9.

```
-DATA L 1 9

Index (1..500)
:   Bus (1..5)
:   :  Slave.C1/C2 Data Category Data 1/Data 2 (SPA-bus. addr.)
:   :  :          Data Type ( DI, AI, DO, AO, EV)
:   :  :          :   Data Format ( BIN, DEC, HEX) or Conversion table number
:   :  :          :   :
:   :  :          :   :     Delta or Mask ( dec or hex number (H))
:   :  :          :   :     :     Moveindex ( * + index)
:   :  :          :   :     :       Update Enable (0,1)
:   :  :          :   :     :       :  Continuous Poll Enable (0,1)
:   :  :          :   :     :       :  :  Timed Poll Enable (0,1)
:   :  :          :   :     :       :  :  :  Spont. Transf.Enable (0,1)
:   :  :          :   :     :       :  :  :  :  Temporary Error (0,1)
:   :  :          :   :     :       :  :  :  :  :  Permanent Error (0,1)
1   2  11I1       AI  DEC   0.02    1  1  0  1  0  0
2   2  14.0/1I3/7 DI  BIN   FC1CH   1  1  0  1  0  0
3   2  14.1E1/3   EV  001   *006
4   2  14.1E4/6   EV  001   *007
5   2  14.0E1/3   EV  001   *008
6   2  14.1I1     AI  DEC   0.02    0  0  0  0  0  0
7   2  14.1I2     AI  DEC   0.02    0  0  0  0  0  0
8   2  14I1       AI  DEC   0.02    0  0  0  0  0  0
9   0  0X0        DI  BIN   0000H   0  0  0  0  0  0
```

DATA_CONVERSION command

Abbreviation: DC

Task: Defines the data conversion tables to be used for converting event codes to EV data.

Format: DATA_CONVERSION <parameters>

<parameters> = D <I>: Define data conversion table.
R <I> <J>: Remove conversion tables.
H: Print help text
L <I> <J> P: List conversion tables. If P is given then the listing stops after every page. Listing continues when the user hits Return/Enter.
<I>: 1..50 = index of the first table
<J>: 1..50 = index of the last table.

Remarks:
When command is activated with D parameter, then a header line and a possible existing definition are displayed. To leave the definition as it is press Return/Enter. To change the definition, type in new definition.

The format of a definition is:
<conversion data> = E<event number>[, <direct data>]
Conversion data is an analog value 0..65535.
Event number is an analog value 1..99.
If direct data is defined it may have one of two formats.

a) Analog value 0..65535
b) Digital value between '0' and '1111111111111111'
   Digital value is given in ' brackets

If the data in the database can be the same as the event code, then the definition can be given in format 'X = EX'. In this case no other definitions can be made in the same conversion table.

The comment text (max. 39 chars) can be given at any moment in ' brackets and the old text is immediately overwritten.

When the definitions of one table are made then press Return/Enter. The header and the definitions of the next table are displayed. If Return/Enter is given as the first command after entering a table, then the whole command is aborted.

Example 1:
Data conversion table to convert the events and corresponding direct data of control unit of type SPTO 1C1 to EV data. The conversion table is designed keeping in mind the following rules:

State	Data to database	Event from slave	Direct data from slave
Earthing disconnector open	1	E1	0 (Channel 0 data I1)
Earthing disconnector closed	2	E2	1 (Channel 0 data I1)
Earthing disconnector undefined	3	E3	2 (Channel 0 data I1)
Earthing disconnector read blocked	3	--	3 (Channel 0 data I1)
Circuit breaker open	1	E1	0 (Channel 0 data I1)
Circuit breaker closed	2	E2	1 (Channel 0 data I1)
Circuit breaker undefined	3	E3	2 (Channel 0 data I1)
Circuit breaker read blocked	3	--	3 (Channel 0 data I1)
Busbar disconnector open	1	E4	0 (Channel 0 data I2)
Busbar disconnector closed	2	E5	1 (Channel 0 data I2)
Busbar disconnector undefined	3	E6	2 (Channel 0 data I2)
Busbar disconnector read blocked	3	--	3 (Channel 0 data I2)

```
-DATA_CONVERSION D 1

CONVERSION TABLE No 1
'SPTO 1C1 conversion table'
1=E1,0
2=E2,1
3=E3,2
3=E3,3
1=E4,0
2=E5,1
3=E6,2

CONVERSION TABLE No 2
-
```

DATA_GROUP command

Abbreviation: DG
Task: DATA_GROUP command displays and modifies the data group definitions.
Format: DATA_GROUP <parameters>

<parameters> = C (Change data group definitions beginning from data group I.)
H (Help.)
L <I> <J> (List data group definitions I to J.)
L P <I> <J> (List data group definition I to J.)
The listing stops after every page.
Listing continues when any key is pressed)
<I>: 1...100, the first data group
<J>: 1...100, the last data group

Remarks:
A data group is a list of related data item indexes and slave numbers:
<index>,<index>,E<slave>,E<slave>.

With C-parameter the command displays headerlines and the group number followed by the list of the group members.

To leave the definition as it is, press only Enter.

To change the definition, type in the data indexes separated with commas or spaces and press Return/Enter.

To remove the definition, type 0 and press Return/Enter.

After Return/Enter is pressed, the next data group is displayed. Now you can change also this data group definition or leave the command by pressing Return/Enter.

Example 1:
The following removes all data group definitions.

```
-DATA_GROUP RA
       -
```

Example 2:
The following modifies data group 10. The group now contain data items 3,4,5 and slave 14. It is changed to contain data items 2,4,5 and slave 14. Data group 11 is not changed.

```
-DATA_GROUP C 10
10 3,4,5,E14
10 2,4,5,E14
11 6,7,8,E15
11
```

2.6 Commands to program the LON interface

The programming of the SRIO 1000M LON interface is done using the following LON commands:

- L
- LON_INSTALL
- LON_BIND
- LON_DATA
- LON_GWCOM
- LON_UNIT

L command is used to send/receive LON messages to/from LON interface. It is also used for testing and with some network management functions.

LON_INSTALL command is used to install LON nodes to LONWORKS™ network. LON_INSTALL command updates "LON node table" of SRIO 1000M.

LON_BIND command is used to bind network variables from point to point.

LON_DATA command is used to define the contents of the "network variable configuration table" of the network variables located in the SRIO 1000M data base.

LON_GWCOM command is used to send any message to the network as an ASCII string with <CR>. Maximum length for outgoing message is 45 characters and for incoming message 60 characters.

In addition to the specialized LON commands the following other commands of the SRIO 1000M must be used in order to make LON interface work properly:

- BUS_MODE	change the mode of the used serial interface to LON interface
- SETUP	setup communication parameters for LON interface
- DATA	define data items of the SRIO 1000M data base
- UNIT	define SPA-bus slaves connected to SRIO 1000M via LON/SPA-gateways

---

L command

Task: Send/receive message to LON interface.
Format: L <addr.type>,<node>,<sub>,<message code>,=<data>
or L <parameter>

A space is also valid separator instead of comma.

<addr.type>	= addressing type in hex format 0 = unassigned (for local SLTA address) 1 = subnet node addressing 2 = Neuron, ID addressing 3 = broadcast >=80H = group addressing (value = 80H+group size)
<node>	= node number in hex format (anything if broadcast message)
<sub>	= subnet number in hex format (1 = the default subnet)
<message code>	= explicit message code in hex format or 80H + NV_selector msb bits (NV_update) C0H + NV_selector msb bits (NV_poll)
<data>	= data bytes of the message in hex format (if message is NV_message then the first data byte includes the lsb byte of the NV_selector)
<parameter>	H = Display Help text S = Returns 16 named driver status words

Depending on the message and address type, a response message or a completion message is shown on the terminal.

Response message format:

First line: <length><~length><command_type|command_queue><msg_header 3B>
<network_address 11B><success/fail_code>
Second line: <data bytes><checksum>

A network address contains always the message format. It can also contain, depending on the message, the source address (2B), the destination address (1-4B) and data bytes.

The response message for "L S" command contains 16 counters which are:

ALERT NI_ACK C_SUM LENGTH ?MESS ?CHAR X_OFFS R_TIMO
R_UART SND_FU NO_RSP REC_FU SLCP_O SLIH_O SLCP_I SLIH_I

Where the counters have the following meaning:

ALERT	No ALERT_ACK received for sent ALERT. This indicates failure in communication between SRIO 1000M and SLTA.
NI_ACK	No free buffers on SLTA. Communication error or overload.
C_SUM	Received message has wrong checksum. Transfer error.
LENGTH	Message length different from length byte in message. Probably transfer error. Creates other errors.
?MESS	Unidentified message.
?CHAR	Unidentified control characters.
X_OFFS	Sent Xoffs. The SL_REC_BUFFER is almost full. Serious overload.
R_TIMO	Receive timeout. Usual if a character is lost or message length is wrong.
R_UART	UART error when receiving. Parity, framing or overrun error. Creates other errors
SND_FU	SL_SND_BUFFER full. Too many messages sent at a time.
NO_RSP	No response or completion message from SLTA.
REC_FU	SL_REC_BUFFER full. Overload. Incoming message is lost.
SLCP_O	Number of messages sent by SLCP to SLIH.
SLIH_O	Number of messages sent by SLIH to SLCP. Should match with SLCP_O.
SLCP_I	Number of messages received from SLIH.
SLIH_I	Number of messages received from SLCP by SLIH. If greater than SLCP_I then an application message has not fetched the message from SLCP.

Remarks:
This command should be used very carefully, because mistakes might set a node to the unconfigured /applicationless state.

Examples:
Query NV_config Table entry 0, node 5, subnet1:
l 2 5 1 68 0
Response: 13 EC 16 6x 09 xx xx xx xx xx xx xx xx xx xx xx xx 38
12 34 09 xx
Last byte of the first line is the success/failed response code of the message. The two first bytes on the second line contain the selector 1234H, third byte is the index to address entry 9. The last byte on line two is the checksum.

LON_INSTALL command

Abbreviation: LI
Task: Install LON node to the network.
Format: LON_INSTALL<index>=(<x>,<n1>,...<n6>,)<node>,<subnet>,<type>
or LON_INSTALL <parameter>(=<index>,<length>)

A space is also valid separator instead of comma.

<index> = index to the SRIO 1000M node table in decimal format

<x> = optional parameter. Value must be set to 0 if there is no Service Pin - message available and Neuron ID must be given by user.

<n1>...<n6> = optional Neuron ID hex bytes, if param. <x> = 0.

<node> = node number of the node to be installed in decimal format

<subnet> = subnet number in hex format (normally = 1)

<type> = node type in hex format
0 = Local SLTA
1 = Neuron base LON node
2 = LON/SPA-gateway
3 = SLTA based node
4 = MIP based node

<parameter> H = Display Help text
L = List address table from <index>,<length> entries
R = Remove installations from entry <index>,<length> of entries

Remarks:
The address table index must match to the node number when local subnet is used (node index < 128). The local SLTA must be installed before any other node is accessed. The maximum number of LON/SPA-gateways that can be installed in a local subnet is 127.

Examples:
Install Neuron based node, node table index 2, node number 2, subnet 1.
LI 2=2,1,1

List the address table from index 1, length of 3 items.
LI L=1 3
Response: LON_INSTALL 001=001,001,1=00 12 0B 01 00 FD, OK status OK
LON_INSTALL 002=002,001,1=00 08 0B 01 00 FD, ERR comm. error
LON_INSTALL 003=255,255,9=FF FF FF FF FF FF, not installed

List the address table from index 126, length of 1 item.
LI L=126 1

Response: LON_INSTALL 126=126,001,0=** SLTA **, OK status OK

LON_BIND command

Abbreviation: LB

Task: Bind two network variables together by NV_selector or set the NV_selector of a single network variable. Connection is made from point to point or broadcast from one node to the subnet.

Format: LB <sel>=<service>,<sn>,<ss>,<si>(,<ai>(,<rn>,<rs>,<ri>))
or LON_BIND H to display the Help text.

A space is also valid separator instead of comma.

<sel> = selector value to be assigned to the defined network variable(s) in hex format (8xxxH = priority)

<service> = service type of NV_message delivery.
0 = ACKD
1 = UNACKD_RPT
2 = UNACKD

<sn> = sender node's node number in decimal format

<ss> = sender node's subnet number in decimal format

<si> = sender node's network variable index in decimal format

<ai> = sender address entry number in decimal format (0..3)

<rn> = receiver node's node number in decimal format
(0 = variable is broadcast to the subnet)

<rs> = receiver node's subnet in decimal format

<ri> = receiver node's network variable index in decimal format

The sender is the node to have an output network variable.

If all parameters are given, the sender's and the receiver's NV_Configuration_tables to the index are defined and the address table entry is created to sender node.

If parameters <rn>, <rs> and <ri> are missing, the address table entry is not created, but the NV_Configuration_table to the index is defined. This can only be used if the address table entry is already created by previous LON_BIND or L command.

If only first five parameters (<sel>,<service><sn>,<ss>,<si>) are given, an input network variable is defined. The network variable can be broadcast from another node or from the host to the network.

Remarks:
The sender node's Neuron Address Table is searched to find a reference to the received node before creating a new address table entry.

Examples:
Bind the NV_output (16) from node 2 in subnet 1 to the NV_input (32) of node 1 in subnet 1 by the selector number 1, using address_table_entry 1. Service type acknowledge.

LB 1=0,2,1,16,1,1,1,32

Bind the NV_output (1) from node 2 in subnet 1 using address_table_entry 1. Service type acknowledged.

LB 2=0,2,1,1,1

Set the selector to the NV_input (33) in node 2 in subnet 1. Service type unacknowledged.

LB 2=1,2,1,33

Set the selector to the NV_output (10) in node 2 in subnet 1. Send NV using address table entry 1. Service type acknowledge.

LB 2=0,2,1,10,0

Note:
NV_index is the identification of the network variable inside LON/SPA-gateway. NV_selector is defined with LB command.

Note:
SLTA node number may not be used. LON_DATA command must be used for configuration NVs in SRIO-SLTA node. If network variables are wanted to be sent to the SRIO (via) SLTA, the address table entry 0 (contains the master address) can be used. See examples.

LON_DATA command

Abbreviation: LD
Task: Define the network variable configuration data for the SRIO database data items used by LON interface.
Format: LON_DATA <index>=<dir>,<sel>,<type>,<ser>,<node>,<sub>
or LON_DATA <parameter>(=<index>,<length>)

A space is also valid separator instead of comma.

<index> = 1...500, SRIO database index in decimal format

<dir> = direction of the network variable, from SRIO point of view.
IN = input network variable
OUT = output network variable

<sel> = NV_selector in hex format, 0-2FFFH, +8000H for priority

<type> = format of the NV_data in decimal format
0 = UINT8
1 = UINT16
2 = INT8
3 = INT16
4 = INT 32
5 = SNVT_alarm

<ser> = service type in decimal format
0 = NV is sent using ACKD service
1 = NV is sent using UNACKD_RPT service
2 = NV is sent using UNACKD service

<node> = the node number of the receiver/sender node in decimal format
(0 = NV is broadcast to the network)

<sub> = the subnet number of the receiver/sender node in decimal format

<parameter> H = Display Help text
L = List the <length> of defined items from <index>
R = Remove activation from <index>,<length> of items

Examples:
Define data index 3, input NV, selector 2, type INT16, service ACKD, receiving node 2, receiving subnet 1.

LD 3=IN,2,3,0,2,1

List 3 defined items from index 5.

LD L 5 3

Response:
LON_DATA 007= IN,8012,3,0,004,001, UNBOUND
LON_DATA 008= IN,0333,3,1,006,002, BOUND
LON_DATA 121=OUT,0135,0,2,123,001, OK

The table contains binding status for input NVs for output NVs appear text OK. The binding status of the input NVs is updated using background polling. The binding status is set to UNBOUND until SRIO 1000M receives the response to the background poll, then the status is set to BOUND. If SRIO will not get a response to polls, status is set to UNBOUND.

Note:
The LON/SPA-gateway network variables 30H, 31H and 32H (the nv_warning, the nv_clock and the nv_gw_status) do not have to be mapped into database.

Note:
Zero as a node number is accepted only if direction of the data is OUT and the service type is UNACKD or UNACKD_RPT. The subnet must match to the SLTA subnet.

Note:
If type of data is set to 5, SNVT-alarm, then SRIO is able to receive time stamped alarms and events through this data item. The received events are converted to SRIO's internal event format as described below and stored to SRIO's event buffer.

SPA slave number = derived from SNVT_alarm "location" field.
- If "location" = sss, where sss is a decimal number (001...999) then SPA slave number = SSS.
- If "location" does not contain a decimal number in the last three characters:
  The SPA slave number will be 900 + the node number (901...999). When the node number is greater than 99, then the SPA slave number is the maximum SPA slave number 999.

SPA channel number = SNVT_alarm "object id" field

SPA event code = SNVT_alarm "alarm type" field

LON_GWCOM command

Abbreviation: LG
Task: Send any message with <CR> to LON nodes.
Format: LON_GWCOM <node>,<sub>,<code>=<command>
or LON_GWCOM H to display the Help text.

A space is also valid separator instead of comma.

<node> = the node number of LON/SPA-gateway in decimal format

<sub> = the subnet number of LON/SPA-gateway in decimal format

<code> = explicit message in hex format
41H = send SPA-message to the slave unit
43H = send message to LON/SPA-gateway

Remarks:
The node number and the subnet number must be given in decimal format. The message code must be given in hex format.

Maximum length of a response message is 60 characters.

If UN command is sent it will also update SRIO 1000M LON_UNIT list.

Examples:
Send UN-command to LON/SPA-gateway, node number 3 in subnet 1.

LG 3 1 43=UN 2=32,2

Send RF-command to the slave unit 35, connected via LON/SPA-gateway, node number 3 in subnet 1.

LG 3 1 41=>35RF:XX

LON_UNIT command

Abbreviation: LU
Task: Define LON_UNIT list to SRIO 1000M.
Format: LON_UNIT <nod>,<sub>,<index>,<sl_nr>,<sl_type>
or LON_UNIT <parameters>

A space is also valid separator instead of comma.

<nod> = node number

<sub> = subnet number

<index> = index in gateway unit list (0..7)

<sl_nr> = slave number

<sl_type> = slave type
0 = SACO 16D
1 = SACO 16A
2 = protective relay
3 = control unit

<parameter> H = Display Help text
L = List all defined units
L P = List all defined units, one page at a time, listing continues when user hits RETURN/ENTER
R<n>,<s> = Remove unit definitions from node number <n>, subnet<s>
R A = Remove all unit definitions

Remarks:
Accept answer (00) to UN command (sent using LG) also updates LON_unit_list.

Examples:
Define slave number 10, type SACO 16D in node 1, subnet.
LU 1,1,4,10,0

List units
LON_UNIT 1 1 2 8 3
LON_UNIT 1 1 3 12 2
LON_UNIT 1 1 4 10 0
LON_UNIT 1 1 7 30 1
LON_UNIT 5 1 0 45 0
LON_UNIT 5 1 1 11 0

Send RF-command to the slave unit 35, connected via LON/SPA-gateway, node number 3 in subnet 1.

LG 3 1 41=>35RF:XX

Configuration commands may also return following error texts:

??	syntax error in command
SLCP Send Error	a driver buffer is full, sending on LON failed
Response timeout	answer in 10 seconds, a communication error (or node does not exist)
Press node service pin	a request to press the node service pin (LI-command)
Illegal parameter value	a parameter out of range
Too large SLTA index	SLTA index must be <128 for the local subnet
Wrong local subnet	a subnet must match to existing SLTA subnet
Illegal subnet	not a local subnet if index >127
Node installed	successful installation completed
Local SLTA not installed	SLTA must be installed before nodes
Local node ID <> Index	in local subnet node index and node number must match
Node not installed	node must be installed before configuration commands can be executed
Gw reply timeout	no answer from gateway in 15 seconds; (usually this means syntax error in gateway command
Index already reserved	node or SLTA with this index is already installed (LI-command) or data item with this index is already defined (LD-command)

2.7 Commands to program the ANSI X3.28 host interface

The address ranges which are used to transfer data with the host computer can be programmed using

ADDRESS_MAP command.

The DI-, DO-, AI- and AO-data ranges and the format and scale of analog data can be programmed using

ANSI_DATA command.

Note:
Usually the address ranges and analog data format and scale are left to their default values.

The Ansi addresses of the individual data items of the database are programmed using

ANSI_ADDR command.

The disabling or enabling of spontaneous event or data sending globally is done using SYSPAR command. The disabling or enabling of spontaneous sending of individual data items, or bits of a data items is done with DATA command. Also the delta value which controls the spontaneous sending of analog data is programmed with DATA command.

Example:
In the example system of figure 2.1 the Ansi addresses would be programmed so that the ANSI_ADDR command would produce the following listing.

```
-ANSI_ADDR L 1 5

Index (1...500)
:  Ansi/AB address
:  :        Bus
:  :        :  SPACOM address
:  :        :  :
1  00000Q  2  1.1/16I1
2  01750Q  2  13I1
3  00001Q  2  14.0/1I3/7
4  00764Q  2  14.1V11/14
-
```

Parameter data transfer

To transfer parameter data via ANSI X3.28 host interface to or from a slave unit, at least one data item must be defined for the unit. This is to route the parameter data messages to the correct serial interface of the SRIO 1000M.

If you want to transfer the parameters of such a unit, which has no data item in the database, you must define a "dummy data definition" with Update Enable = 0, Continuous Poll Enable = 0, Timed Poll Enable = 0 and Spontaneous Transfer Enable = 0. A "dummy data definition" for slave 1 on bus 2 could be done e.g. in the following way:

```
-DATA D

Index (1..500)
:    Bus (1..5)
:    :  Slave.C1/C2 Data Category Data 1/Data 2 (SPA-bus. addr.)
:    :  :          Data Type ( DI, AI, DO, AO, EV)
:    :  :          :   Data Format ( BIN, DEC, HEX) or Conversion table number
:    :  :          :   :
:    :  :          :   :     Delta or Mask ( dec or hex number (H))
:    :  :          :   :     :    Moveindex ( * + index)
:    :  :          :   :     :      Update Enable (0,1)
:    :  :          :   :     :      :  Continuous Poll Enable (0,1)
:    :  :          :   :     :      :  :  Timed Poll Enable (0,1)
:    :  :          :   :     :      :  :  :  Spont. Transf.Enable (0,1)
:    :  :          :   :     :      :  :  :  :  Temporary Error (0,1)
:    :  :          :   :     :      :  :  :  :  :  Permanent Error (0,1)
     2  1.1I1      DI  BIN   0000H  0  0  0  0  0  0
```

ADDRESS_MAP command

Abbreviation: AM
Task: Displays and changes the address rages defined in the SRIO 1000M "Address map". The address map defines the address ranges through which different types of SPACOM system data can be addressed by the host.
Format: ADDRESS_MAP <parameter>

<parameter> =		(No parameter, display map.)
	C	(Change address map.)
	H	(Display help text.)

Remarks:
If no parameter is given, the current address map is displayed.

If C parameter is given, the command starts listing the start and end addresses of the different address ranges. To leave the displayed address as it is, press Enter. To change the displayed address type in a new value for the address.

If the given value is invalid the command displays text: "Bad address !!" and "New address ?" and you can try again.

If only Return/Enter or a valid new address is typed, the address is changed and the next address is displayed. All the Address map addresses are gone through in the same way until the command line interpreter prompt is displayed.

Note:
All the addresses handled by ADDRESS_MAP command are decimal word addresses in the range 0...32767.

Example:
The following is an example of a typical "Address map".

-ADDRESS_MAP

RANGE	START	END	LENGTH
DI-range	0	499	(500 words)
DO-range	500	999	(500 words)
AI-range	1000	1499	(500 words)
AO-range	1500	1999	(500 words)
Event data	2400		(4 words)
Parameter data buffer	2000		(168 words)
-Messages to SPACOM	2000		(40 words)
-Messages from SPACOM	2040		(128 words)
Time	2300		(9 words)
Diagnostic data	2500		(26 words)
System parameters	3000		(2000 words)
Object parameters	5000		(9200 words)

-

ANSI_DATA command

Abbreviation: AN
Task: ANSI_DATA command displays and modifies the "Address map" addresses and the data format and scale used in communications with the host computer.
Format: ANSI_DATA <parameter>

```
<parameter> = L  (List      )
              C  (Change )
              H  (Help      )
```

Remarks:
The command displays the DI address range, DO address range, AI address range, AO address range, Analog data format and Analog data scale. The addresses are displayed as octal word addresses.

If the command was activated with C parameter, new address values and format and scale can be given. Each value must be typed into the field defined by the headerlines. All the fields must be given. The addresses can be decimal, octal or hexadecimal word addresses.

If an invalid line is entered, the command displays two question marks ??, and you can try again. If a valid line is entered, the command displays the line again and you can make new changes. To leave the values as they are, press only Return/Enter.

Example:
The following example shows how to change the AO address range using ANSI_DATA command.

```
-ANSI_DATA C

DI-range
:             DO-range
:             :             AI-range
:             :             :             AO-range
:             :             :             :             Analog data format
:             :             :             :             (long integer=0,
:             :             :             :             3bcd=1, 6bcd=2)
:             :             :             :             :      Analog data
:             :             :             :             :      (scale 1 10 100
:             :             :             :             :      1000 10000)
:             :             :             :             :      :
:             :             :             :             +--------+:
:             :             :             :             :      :
00000Q-00763Q 00764Q-01747Q 01750Q-02733Q 02734Q-03717Q 0      1000
0Q-763Q       764Q-1747Q    1750Q-2733Q   2734Q-2738Q   0      1000
00000Q-00763Q 00764Q-01747Q 01750Q-02733Q 02734Q-02738Q 0      1000
-
```

ANSI_ADDR command

Abbreviation: AA

Task: ANSI_ADDR command displays or modifies the Ansi addresses of the data items. These Ansi addresses are used by the host protocol (ANSI X3.28/AB) to address data of the substation. The addresses must be in range 0...32767 when interpreted as decimal word addresses. The corresponding byte address range is 0...65534.

Format: ANSI_ADDR <parameters>

<parameters> = C (Change Ansi addresses beginning from data item I.)
H (Help.)
L <I> <J> (List Ansi addresses of data items I to J)
L <I> <J> P (List Ansi addresses of data items I to J. The listing stops after every page. Listing continues when any key is pressed.)
<I>: 1...500, index of the first item
<J>: 1...500, index of the last item

Remarks:
If command is entered with L parameter, then the Ansi addresses of data items I to J are displayed. Together with the Ansi addresses it is also displayed the SPACOM addresses of the data items. The Ansi addresses are displayed as octal word addresses.

If command is entered with C parameter, then the Ansi address and the SPACOM address of the data item I are displayed. To leave the Ansi address as it is, press Return/Enter. To change the Ansi address, type new address and press Return/Enter. The address can be given in decimal, octal or hexadecimal format. An octal address must be followed by a Q-character, a hexadecimal address must be followed by a H-character.

If the new address is invalid, two question marks are displayed and you can try again. If the new address is valid, the new Ansi address in octal format and the SPACOM address are displayed together with the Ansi address and SPACOM address of the next data item. Now you can continue by typing a new Ansi address for the next data item or you can quit by pressing Return/Enter.

Note: The SPACOM addresses can not be modified with ANSI_ADDR command. They can be modified with DATA command.

Note: The address of 32 bit analog data values must be even word addresses. One 32 bit analog data value is transferred using two words.

Note: The ANSI address of the "direct" data items used to read EV-data directly from slave units must be 77777Q.

Example:
The following shows how to change the Ansi address of data item 1 from 77777Q to 100 (144Q) using ANSI_ADDR command.

```
-ANSI_ADDR C 1

Index (1...500)
:   Ansi/AB address
:   :          Bus
:   :          :   SPACOM address
:   :          :   :
1   77777Q     2   20.0/0I1/1
    100
1   00144Q     2   20.0/0I1/1
2   77777Q     2   22.0/1I3/7
```

2.8 Control commands for local event reporting

Note: These commands are not applicable in SRIO 500M.

SET_PRINTER command

Abbreviation: SP
Task: Setup command for event printer devices 1 and 2.
Format: SET_PRINTER <parameter>
<parameter> = 1...2 (Device number.)
H (Help.)

Remarks:
Parameters which can be changed by the command:

PRINTER IN USE: Event reports are printed out on this device.

TEXT ENABLE: User defined event text is generated if this setting is 1, else a standard event report (time + event) is printed.

CHARACTERS/LINE: Maximum width of the event report text.

LINES/PAGE: Number of lines between FORM FEEDS. If No FORM FEED is desired then give the value 99.

PAGE NUMBERING: Prints a page number in the upper right corner after every FORM FEED.

PAGE HEADER: Prints a header to each page during FORM FEED. The header text is defined with the PAGE_HEADER <dev> command.

LIFESIGN: The device makes an extra printout approx. every 24 hours with time and PAGE HEADER text to assure that the printer connection is operating.

PRIORITY: Describes the action which is taken if the internal event buffer overflows due to problems with this device (paper out, printer cable disconnected etc.):
- LOW priority: Oldest events are lost. New events will be collected.
- HIGH priority: Oldest events are not lost. New events are not collected until all the event buffer empty.

Example:
Changing of the number of lines on an event listing page.

```
-SET_PRINTER 1

Printer In Use      ................... (1 - Yes, 0 - No)
:  Text Enable      ................... (1 - Yes, 0 - No)
:  :  Characters/Line ................ (28...199)
:  :  :    Lines/Page ................. (10...98, 99 - No Form Feed)
:  :  :    :   Page Numbering ... (1 - Yes, 0 - No)
:  :  :    :   : Page Header ........ (1 - Yes, 0 - No)
:  :  :    :   : :  Lifesign ........... (1 - Yes, 0 - No)
:  :  :    :   : :  :  Priority ........ (1 - High, 0 - Low)
:  :  :    :   : :  :  :
1,1,120,65,1,1,0,0
1,1,120,70
-
```

Note:
To enable the SRIO 1000M to supervise the state of the printer, the <dcd> handshaking signal must be programmed "in use" by SETUP command. The "printer ready" signal, which usually is the DTR-signal of the printers RS-232 interface, must be connected to the DCD-signal of the SRIO 1000M RS-232 interface. If current loop is used, then the printers "ready" loop must be connected to the SRIO 1000M "Ready+" and "Ready-" signals.

DEVICE command

Abbreviation: DE
Task: Displays and changes printer device to serial interface assignment.
Format: DEVICE <parameter>
<parameter > = (No parameter, display or change.)
H (Help.)

Remarks:
The current device to serial interface assignment is displayed, and the user can change the assignment.

To leave the assignment as it is, press Return/Enter key. To change the assignment, type new serial interface numbers. If there is something wrong with the input line the command prints an error message.

Example:
The following programs the event printer 1 to serial interface 4 and event printer 2 to serial interface 3.

-DEVICE

DEVICE NUMBER 1,2,3,4,5
SERIAL INTERFACE NUMBER 1,2,3,4,5
NEW ASSIGNMENT >4,3,3,3,3

PAGE_HEADER command

Abbreviation: PH
Task: Defines the page header for printer devices 1 and 2.
Header text is printed out on top of every event listing page.
(SET_PRINTER parameter PAGE HEADER must also be 1 !)
Format: PAGE_HEADER <parameter>
<Parameters> 1 - Page header for device 1.
2 - Page header for device 2.
H - Help text.

Remarks:
First give the command PAGE_HEADER 1 (or 2).

The old text or a "No text" message is printed out.

Write a new header text + RETURN or just RETURN to exit without any changes to the text or write a $ (dollar sign) character to delete the text.

Example:
Write a header for event printer 1 pages.

-PAGE_HEADER 1

No Text
*********** Vasa Mottagningsstation ***********
-

T command

Task: Edit and list event texts. Define channel groups.

Format: T <parameters>

<Parameters>

H	: Help text.
s/s.c/cEe	: Write text. s(lave number) = 1..999, c(hannel number) = 0..999, e(vent code number) = 1..99
Gn	: Write text to group number n.
D Gn	: Define text group number n.
L	: List event or group texts.
R A	: Remove all texts and group definitions.
R Gn	: Remove text and definition from group number n.
R s/s.c/cEe	: Remove text.

Function: Texts can be written or changed in two different ways:

1. Write new text in a same command line

-T 12 =<new text for slave>
-T 12.3 =<new text for channel>
-T 12.3E4 =<new text for event code>
-T 12/15.3E4=<new text for event code>
-T 12.3/16E4 =<new text for event code>
-T 12/15.3/16E4=<new text for event code>
-T G3=<new text for channel group>

2. Write new text in next command line

-T 12.3E4
<old text>
<new text>

There can be special characters in text:

\$n	= carriage return
\$fxx	= "xx" front marks for event, always two (2)
\$>xxxxx\$<	= "xxxxx" SPA-bus message, length free e.g. 10R11S12:

Textgroups can be defined:
-T D G3
<old definition>
<new definition>
-
Definition format: e.g. 1,2.1,2.2,3.0,3.1,4
(slave 1 all channels, slave 2 channels 1 and 2...)
There can be max. 10 items in a group definition.

Texts can be listed to the terminal:
-T L (list all texts to the terminal)
-T L 2 (list all texts of slave number 2)
-T L 2.1E4 (list text of event 2.1E4)

Texts and group definitions can be removed:
-T R A (remove all texts and group definitions)
-T R G4 (remove text and def. from group number 4)
-T R 2/4.1/16E2 (remove event text)

Note:
It is not enough to define just the "channel texts" for the alarm channels also the event code texts must be defined. These are usually defined in a group format as shown by the next example.

Example 1:
Texts for a SACO alarm system consisting of three 16 channel alarm units (slave numbers 10,11 and 12) can be defined in the following manner.

Texts for the event codes, ($f=printed before the time stamp):
-T 10/12.1/16E1=$f**
-T 10/12.1/16E2=$f..   (two space characters)

Texts for the channels:
-T 10.1=TRANSFORMER GAS RELAY
-T 10.2=GENERAL OIL ALARM
. etc...

Texts for the special events coming from "channel 0":
-T 10/12.0E50= UNIT RESTART
-T 10/12.0E51= EVENT BUFFER OVERFLOW
-T 10/12.0E53= COMMUNICATION ERROR$f**
-T 10/12.0E54= COMMUNICATION OK

-T 10.0=SACO 16D2 NUMBER 10,
-T 11.0=SACO 16D2 NUMBER 11,
-T 12.0=SACO 16D2 NUMBER 12,

When alarm on unit 10 channel 1 goes on and then off after about five minutes, the following texts would be printed:
**89-10-30 11.00:31.472 TRANSFORMER GAS RELAY
89-10-30 11.05:44.371 TRANSFORMER GAS RELAY

If connection to alarm unit 12 is broken for about 10 minutes, then the following texts would be printed:
**89-10-30 11.00:31.472 SACO 16D2 NUMBER 12, COMMUNICATION ERROR
89-10-30 11.10:01.222 SACO 16D2 NUMBER 12, COMMUNICATION OK

Example 2:
Texts for activating SACO alarm lamp and/or reflash relay from protective relay events.

If a trip event E3 from overcurrent unit number 121 should activate alarm unit 10 channel 16, then the following texts could be used:

```
-T 121.0=FEEDER 2, OVERCURRENT UNIT
-T 121.0E3= TRIP$f**$>10W16I1:1:$<$>10W16I1:0:$<
                       |            |
                     alarm        alarm
                  channel on   channel off
```

When overcurrent unit trips, then the following text would be printed. Channel 16 in alarm unit 10 is activated just before the text is printed out.

**89-10-30 11.00:31.472 FEEDER 2, OVERCURRENT UNIT TRIP

Example 3:
Producing text from four different elements: group text, slave text, channel text and event code text.

Let's suppose we have slaves 10, 11 and 12. Each of them has 16 channels. Each channel can produce event codes E1 and E2.

Defining a group and text for it:
-T D G1
10,11,12
-
-T G1=Group 1,
-

Slave texts (note space character before text):
-T 10= Slave 10,
-T 11= Slave 11,
-T 12= Slave 12,

Channel texts:
-T 10.1= Channel 1,
-T 10.2= Channel 2,
. etc...

Event code texts:
-T 10/12.1/16E1= Event E1$f**
-T 10/12.1/16E2= Event E2$f..

If there comes an event E1 in slave 12 channel 8, then the following would be printed out:

**89-10-30 11.00:31.472 Group 1, Slave 12, Channel 8, Event E1

2.9 Diagnostic commands

DIAGNOSTIC command

Abbreviation: D
Task: DIAGNOSTIC command displays or resets the diagnostic counters of serial interfaces 1...5. (Serial interface 5 can only be found in SPSC 500M.)
Format: DIAGNOSTIC <parameter>
<parameter> = (No parameter, display.)
RA (Reset all counters.)
RX (Reset counters of line X. X = 1...5)
H (Help.)

Remarks:
When either the total number of errors or total number of messages reaches value 2'147'483'647, all the diagnostic counters of the interface are reset.

Note:
S-bus 1 is the same as serial interface 1 etc.

RESPONSE command

Abbreviation: RE

Task: RESPONSE command displays or resets the response time and transmit + receive time counters of serial interfaces 1...4.

Format: RESPONSE <parameter>

<parameter> =		(No parameter, counters.)
	RA	(Reset all time counters.)
	RX	(Reset counters of line X. X = 1...5)
	H	(Help.)

Remarks:
The response time and transmit+receive time are mean values calculated as follows: new mean time value = (old mean time value + new time) / 2.

The response time includes the actual response time and the transfer time of 3 message bytes.

Note:
Response time is only calculated in "SPA-bus master" or "SACO 100M Master" modes.

ANSI_DIAGNOSTIC command

Abbreviation: AD

Task: ANSI_DIAGNOSTIC command displays or resets the diagnostic counters of the serial interface used for host interface with ANSI X3.28 protocol.

Some other parameters are also displayed.

Format: ANSI_DIAGNOSTIC <parameter>

<parameter> =		(No parameter, display.)
	R	(Reset all counters.)
	H	(Help.)

Remarks:
The maximum value of counters is 65535. These counters do not reset automatically.

3.0 Utility commands

Z command

Task: Z command sends a message to one of the serial interfaces 2...4 and displays the received reply message.

```
Format: Z <s> <message>
           |    |
           |    message to be sent
           |
           the number of the serial interface
```

Remarks:
The command sends the given message to the defined serial interface and displays the received reply message.

The message to be sent depends on the operating mode of the used serial interface. If the interface is in SPA-bus or SACO 100M interface mode, then the message is given in the format used in these protocols except that no start character or checksum is given.

If some error occurs the command displays one of the following error texts:
"Send error"
"No reply"
"Timeout"

Example 1:
The following sends to serial interface 2 a message which reads the "slave identification data" F from slave 1. The reply comes from alarm unit SACO 16D2.

-Z 2 1RF:
1D:SACO 16D2:
-

Example 2:
The following sends to serial interface 2 message which changes the "slave number" of SACO 100M from 901 to new value 902.

-Z 2 901WV200:902:
902A:
-

X command

Task: X command sends a command message to serial interface A and displays the received reply message.

Format: X <parameter>
<parameter> = H (Help.)
<DST> <CMD> <data> (Sends a message.)

Remarks:
Command format: X <DST> <CMD> <data>
Reply format: <DST> <SRC> <STS> <data>

<DST> = destination station number (two hex digits)
<SRC> = source station number (two hex digits)
<CMD> = command byte (two hex digits)
01=block read,
08=unprotected block write,
00=protected block write,
05=unprotected bit write,
02=protected bit write
<STS> = status byte (two hex digits)
<data> = "data" bytes in hex format in a command message data bytes may be e.g.
ADDR,SIZE
ADDR,data
ADDR,Set-mask,Reset-mask,ADDR+1,Set-mask,Reset-mask

If some error occurs the command displays one of the following error texts:
"Send error"
"??"
"Too long reply"
"Timeout"

Note:
Words and longinteger (32 bit) values are given low byte first. The numbers are always hexadecimal numbers.

Example 1:
Sending of a block read command.
-X C1 01 0E02 04
15 C1 41 00 00000000
-

Example 2:
Sending of a unprotected block write command.
-X C1 08 0E02 05000000
15 C1 48 00
-

Example 3:
Sending of a unprotected bit write command.
-X C1 05 0200 0100
15 C1 45 00
-

EVENT_MONITOR command

Abbreviation: EM
Task: EVENT_MONITOR command list events to the terminal.
Format: EVENT_MONITOR <parameter>

<parameter> = (No parameter, list.)
H (Help.)
R (Reset event buffer.)

Remarks:
The events are listed in format:
yy-mo-dd hh.mm:ss.sss <slave>.<channel>E<event number>

The listing is paused, when all the new events have been listed or after 20 events. To stop the listing and leave the command during the pause, press Return/Enter.

When listing is restarted, all the new events that came during the time the command was inactive are listed- provided that the event buffer did not overflow or it was not resetted.

Note:
No user text is available.

Example:
The following is a sample listing produced by the command.

-EVENT_MONITOR

88-02-22 17.10:09.889 004.013 E03
88-02-22 17.10:09.959 004.013 E04
88-02-22 17.10:14.948 004.013 E03
88-02-22 17.10:15.203 004.013 E04
-

IOTEST command

Abbreviation: IOT
Tasks: The current state of the SRIO hardware inputs and outputs can be read with this command. The output relays can also be turned on/off individually. This command is used only in testing the SRIO hardware.
Format: IOTEST

EXEC_TIMES command

Abbreviation: ET
Tasks: List or reset records with measured execution times of given processes.
Format: EXEC_TIMES <parameters>

a) to display help text
   -EXEC_TIMES H
b) to reset measured times
   -EXEC_TIMES RA = of all processes
   -EXEC_TIMES R n = of process number n
   -EXEC_TIMES R n = of processes from n to m
c) to list measured execution times
   -EXEC_TIMES L n = of process number n
   -EXEC_TIMES L n m = of processes from n to m

The following information is displayed for a given process:
- maximum execution time,
- minimum execution time,
- average execution time,
- number of recent executions used to compute the average time.

Note:
This command is only used for software testing.

RESET command

Task: RESET command resets the SRIO 1000M.
Format: RESET <parameter>
<parameter> = (No parameter, normal operation.)
H (Help.)

Remarks:
The command causes either software or hardware reset. Hardware reset is done if the device contains a hardware watch-dog.

Note:
All configuration data that has not been stored in EEPROM memory is lost !

A command

Task: Displays the active alarms of the alarm units.
Format: A <parameter>

<parameter> =		(No parameter, display active alarms.)
	H	(Display help text.)

Remarks:
The command list the active alarms of those slave units, which have been defined using the UNIT command.

The active alarms are requested from the SPA-bus slaves using SPA-bus "RA-message".

The alarms are listed using texts defined by T command.

Example:
-A
**A001 OIL LEVEL
*A214 GENERATOR COOLING AIR FLOW DISTURBANCE

EVENT_POINTER command

Abbreviation: EP
Task: Display, change and reset the event buffers read (get) pointers.
Format: EVENT_POINTER <parameter>

<parameter> =	H	(Display help text.)
	L	(List event pointers.)
	C n	(Change the value of pointer n.)
	R n	(Reset event pointer n.)
	RA	(Reset all event pointers.)

Remarks:
This command is used in system startup phase to empty the event buffer.

Command can also be used to reprint e.g. 25 last events: move pointer backwards using C-parameter.

> -command (SPA-command)

Task: > -command sends a given message to the SPA-bus slave and displays the received reply message.

Format: The command format is the same as the master message format in the SPA-bus or SACO 100M protocol:

```
>......XXcr
|   |   |  |
|   |   |  end character CR (Return/Enter)
|   |   checksum (or two "X"-characters)
|   message header and data part
start character
```

Remarks:
The number of the serial bus to which the message is sent is searched from the event poll list (UNIT-list). If the bus number is not found from the UNIT-list or if the slave does not give any reply then no reply message is displayed.

After sending the message and displaying the received reply, the command line interpreter starts to wait for the next SPA command and no prompt is displayed. To display the prompt and return to normal command mode, press Return/Enter.

Example:
Send message to slave number 2.
>2RF:XX
<2D:SACO 16D1:XX

Note:
The reply message is in SACO 100M format: start character is "<" and end characters are "CR,LF".

Note:
The <aut.lf>-parameter in the setup of the used serial interface must be 1.

Note:
This command can also be used to read event from the SRIO event buffer, using format >numRL:XX.

LON_QUERY_ID command

Abbreviation: LQ

Task: Request selected nodes to respond with a message containing their unique Neuron ID and Program ID, and list the found nodes on the terminal.

Format: LON_QUERY_ID <mode>
or LON_QUERY_ID H to display the Help text.

<mode> = nodes to respond.
1 = all nodes of the network
2 = all unconfigured nodes of network

SACO 100M communication protocol for data communicators SPSC 500M and SRIO 1000M V1.1

ABB Network Partner

SACO 100M communication protocol for data communicators SPSC 500M and SRIO 1000M

Table of Contents

Versions:

Version	Date	Signed	Changes
V1.0	06.11.1989	O. Vähämäki	- Original version. - Based on "SACO 100M remote control and automation interface communication protocol V1.3, 01.02.1988"
V1.1	11.09.1990	O. Vähämäki	- The maximum speed was reduced to 4800 b/s.

Note:
This protocol is implemented in SPSC 500M version V3.A or later and in SRIO 1000M version V4.0 or later.

1.0 Introduction

The serial interfaces of the data communicators can be used for a connection with a host computer (master system), e.g. a remote control unit of a substation, programmable logic controller, or a personal computer. Multiple host computers can be connected to one data communicator, if the other host is using ANSI X3.28 protocol and the other is using SACO 100M protocol.

Overview of communications protocol

The electrical interface is a serial RS-232-C interface. The data transfer rate is 300, 1200, 2400 or 4800 bit/s. In principle the protocol is the same as SPA-bus protocol.

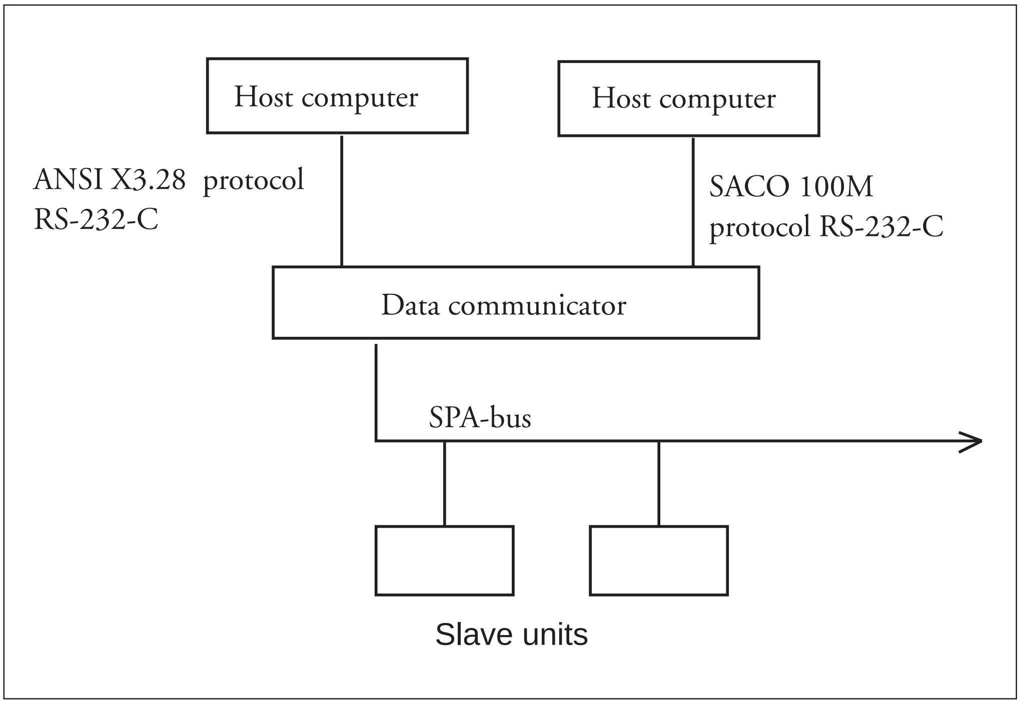


Figure 1. Connection of data communicator to a host computer.

Communication is always initiated by the master. The master system sends either data read or write messages which are responded by the data communicator with a data message or an appropriate acknowledgment message.

Data transferred

The host computer may address its message to either the data communicator or a slave unit attached to it.

Messages addressed to the data communicator can be used for reading event data and for writing or reading database or other internal data of the data communicator.

Messages addressed to the slave units allow the host to read or write data directly to/from the slave units.

2.0 Differences between SACO 100M communication protocol and SPA-bus communication protocol

Basically, the SACO 100M communication protocol is the SPA-bus communication protocol with some slight modifications. The SPA-bus protocol is described in document "SPA-bus communication protocol Vx.x".

The protocol hierarchy assumes the host computer to act as the master to which the data communicator is a slave. However, also the slaves of the data communicator are addressable as slaves by the host computer.

The basic format of messages is equivalent to those of the SPA-bus protocol.

2.1 Message length and start/stop characters

The maximum length of host computer messages is 80 characters. The messages start with character ">" and end with character "cr". The host computer messages have always only one line.

Usually the data communicator does not echo the characters sent to it.

The maximum length of the data communicator messages is 255 characters. The messages start with character "<". The message ends with characters "crlf", which are not preceded with character "&".

2.2 Communications initialization and exiting

Communication in enabled, if the used serial interface is programmed to operating mode "slave of SACO 100M protocol".

The operating mode of the serial interfaces of the data communicator is set using SETUP and/or BUS_MODE commands.

2.3 Data transfer rate and checks

Transfer rate: 300, 1200, 2400 or 4800 b/s

Checksum: the same as in SPA-bus

Parity bit: even (no parity or odd parity can also be used)

2.4 Delay durations in communications

The messages addressed to the slaves are delayed approximately with 100 milliseconds by data communicator. To the messages addressed to the data communicator itself it responds in about 100 milliseconds.

3.0 Messages to slave units

When a message by the host computer is addressed to a slave which is attached to the data communicator, the data communicator normally communicates the message to the slave and relays the reply message from the slave to the host computer. If the data requested by the type R message is already in the database of the data communicator, then the data for the reply message is read from the database and message is not communicated to the slave.

If the slave does not respond, the data communicator sends a message of type N with error code 4. (Refer to SPA-bus communication protocol, page 11). This special reply message incorporates the data communicators own slave number as the slave number.

By using the data read messages, type R messages, the host computer may request all other information except event data of the slaves, data of category L. The host computer can send data to the slaves directly by using the direct write messages, unconfirmed write messages of type W.

If the data sent to the slave is defined as a data item of the data communicator's database and if the item is a member of a data group, then the data write initiates so called auto poll function.

4.0 Messages to the data communicator

A message addressed to the data communicator can access the following information:

Information	Channel	Code	Read or Write	Values
Device name	0	F	R	Text
Events from event buffer	0	L	R	Time, slave, channel, event
Events from back-up event buffer	0	B	R	Time, slave, channel, event
Time	0	T	R,W	Year, month, day, hour, minute, seconds, milliseconds (Note 1.)
Local/Remote state	0	V1	R	1 = local state 2 = remote state
EEROM store	0	V151	R,W	1 = starting store (W) 1 = storing 0 = store done (R)
Communication speed	0	V201	R,W	300, 1200, 2400, 4800
Version number	0	V205	R	Text
Slave number of the data communicator	0	V200	R,W	901...998 (Note 2.)
Database data bit0	1...500	I1	R,W	0, 1
Database data bit1	1...500	I2	R,W	0, 1
Database data bit2	1...500	I3	R,W	0, 1
Database data bit3	1...500	I4	R,W	0, 1
Database data bit4	1...500	I5	R,W	0, 1
Database data bit5	1...500	I6	R,W	0, 1
Database data bit6	1...500	I7	R,W	0, 1
Database data bit7	1...500	I8	R,W	0, 1
Database data bit8	1...500	I9	R,W	0, 1
Database data bit9	1...500	I10	R,W	0, 1
Database data bit10	1...500	I11	R,W	0, 1
Database data bit11	1...500	I12	R,W	0, 1
Database data bit12	1...500	I13	R,W	0, 1
Database data bit13	1...500	I14	R,W	0, 1
Database data bit14	1...500	I15	R,W	0, 1
Database data bit15	1...500	I16	R,W	0, 1
Database data in analog format	1...500	I17	R,W	-2147483.647... +2147483.647 (Note 3.)

Note 1: Clock message format: >nnnWT:yy-mo-dd hh.mm;ss.sss:XXcr
Time can also be sent using message of type: >900WT:ss.sss.XXcr.

Note 2: Numbers 1...899 can also be used as slave number of the data communicator, but usually the numbers of the data communicators are >= 901.

Note 3: The resolution of data in database is 0.001.

4.1 Event data

Using a message addressed to the data communicator, the host computer can request time-marked event data.

The requesting is done on data categories L and B. The requesting message is the normal data read message addressed to the data communicator. (E.g. >901RL:XXcr.)

Data category L:

New events stored after the last received request are available in data category L. The data communicator sends the new events starting from the oldest. If all new events do not fit into one message, the rest of new events will be sent in reply to the next request.

If the host computer desires to retrieve all events stored by the data communicator, the master must read data category L at regular interval.

Data category B:

Data request of data category B includes same data as data request of the last requested data of category L.

When the data communicator is requested for data category L, and an error is detected in the reply message, data requesting cannot be repeated on category L data but the requesting must be imposed on category B data.

4.1.1 Event presentation format

The event marking includes the following information:

<time>< ><slave number><.><channel number><event code>

<time>	= The time marking consists of the time in units of hours, minutes, seconds, and milliseconds in the form of 13.42;10.401. If the time marking is unverifiable, the seconds and its parts in the marking are replaced by marking ??.???.
< >	= delimiter (space)
<slave number>	= number 1...999 (1...3 digits) The number is either the slave number or the data communicator identification number
<.>	= delimiter
<channel number>	= number 0...999 (1...3 digits)
<event code>	= letter E and number 1...63 (1 or 2 digits)

For instance, an event E2 at time 13.42;10.401 on channel 5 of slave 5 and an event E49 at time 13.42;11.500 on channel 16 of slave 99 would be seen in the message in the following format:

<901D:13.42;10.401 005.05E02/&crlf
13.42;11.500 099.16E49:XXcrlf

If the data communicator has not recorded any new events after the last requesting, it replies with an empty data message.

>901D::CCcr

4.1.2 Event codes

The event code alternatives for messages received from different slave units are presented in the corresponding product documents of the slaves. Generally, the events associated with the actual process are coded as E1...E10. In addition, all slaves use on channel 0 codes E50 and E51 for indicating the following information:

E50 = hardware or program fault in slave
E51 = event buffer overflow in slave

In addition to codes received from the slaves, the data communicator reports some events associated with the slaves using the following event codes:

E13 = event generated from event matrix of slave type 0
E52 = temporary malfunction in slave connection
E53 = no connection available with the slave
E54 = connection with the slave available after a break

The special situations of the data communicator itself generates the following event codes:

E51 = event buffer overflow of data communicator

4.2 Database data

4.2.1 Direct database addressing

The database data can be accessed using channel numbers 1...500, data category I and data numbers 1...17.

Each channel corresponds to one data item of the database.

Each data item can be read either in binary format or in analog format using data codes I1...I17:

I1...I16 = bits 0...15 of the data item
I17 = the value of the data item in analog format

Example:
If data communicators own slave address is 901, then message
>901W142I4:1:XXcr sets bit 3 of data item 142.
Message >901R142I17:XXcr reads data item 142 in analog format.

4.2.2 Indirect database addressing

The database data can be read also indirectly by type R messages addressed to slaves. If a R message request for such a slave data which has been defined as one data item of the data base, then data is read from the database and not from the slave.

Example:
Let's suppose that the data communicators' data item 7 has been defined as follows:

```
index
:    bus
:    :  SPACOM address
:    :  :             Data type ...
:    :  :             :
72   2  14.0/1I3/7    DI        ...
```

Now the reply data for a host computer's message >14R0/1I3/7:XXcr is read from database data item 72 bits 0...9. (Channel 72, I1...I10.)

ANSI X3.28 protocol, lower layers

General

SRIO has one serial interface for communications with the host. The used protocol resembles ANSI X3.28 protocol. This protocol is described e.g. in "Allen-Bradley: 1771-811 PLC-2-family/RS-232-C Interface module 1771-KG; User's Manual". The electrical connection of the interface is RS 232.

The operating mode can be either full duplex or half duplex. Baudrate can be selected from 50 to 9600 b/s. Error checking method can be either CRC or BCC with even or odd parity. Character length is always 8 bits and 1 stop bit is always used.

The station address of the SRIO and the host can be selected from 1 to 254. All spontaneous data and events are sent to the defined host station address.

Used commands and STS byte values

The used commands of the protocol are listed in table 2.1. The STS (status) byte in SRIO reply messages can get the values listed in table 2.2.

Command name	Command bytes CMD	FNC	Accepted by SRIO	Sent by SRIO to the host.
Unprotected Block Read	01		Yes	
Unprotected Block Write	08		Yes	Yes
Unprotected Bit Write	05		Yes	Yes
Protected Block Write	00		Yes	
Protected Bit Write	02		Yes	
Set ENQs	06	06	Yes	
Set NAKs	06	05	Yes	
Set Timeouts	06	04	Yes	
Set Variables	06	02	Yes	
Diagnostic Counter Reset	06	07	Yes	
Diagnostic Loop	06	00	Yes	
Diagnostic Read	06	01	Yes	
Diagnostic Status	06	03	Yes	

Table 2.1. The commands accepted and sent by SRIO.

STS byte	Meaning
00	No errors
10	Error in CMD, FNC, SIZE or message length.
40	Error in SPACOM communication.
50	Error in address or data conversion.

Table 2.2. The STS byte.

Message format

The protocol is a character oriented protocol that uses the following ASCII control characters extended to eight bits by adding a zero for bit 7:

SOH = 01H, STX = 02H, ETX = 03H, EOT = 04H,
ENQ = 05H, ACK = 06H, DLE = 10H, NAK = 15H.

Additionally a block check character (BCC) or two-byte cyclic redundancy check (CRC) field is used at the end of each packet for error checking.

General message format in full duplex protocol is the following:

DLE STX DST SRC CMD STS TNS data DLE ETX BCC/CRC.

The "data" contains FNC ADDR DATA, ADDR DATA, DATA or nothing.

General master (host) message format in half duplex protocol is:

DLE SOH STN DLE STX DST SRC CMD STS TNS data DLE ETX BCC/CRC

General slave (SRIO) message format in half duplex protocol is the same as general message format in full duplex protocol.

Special messages:

ACK message: DLE ACK
NAK message: DLE NAK
ENQ message: DLE ENQ
Poll message: DLE ENQ STN BCC
EOT message: DLE EOT

STN = slave station number
DST = destination station number
SRC = source station number
CMD = command byte
TNS = transaction number (16 bits)
FNC = function code
ADDR = data address (16 bits)
DATA = data bytes

In a reply message the CMD byte is equal to the corresponding command messages CMD byte + 40H.

Note:
Data value "10H" (=DLE) is transmitted as "10H 10H" to separate it from control character DLE.

Operation principle

The communication is based on a command/reply principle. The communication between the host and SRIO is illustrated in the following figures.

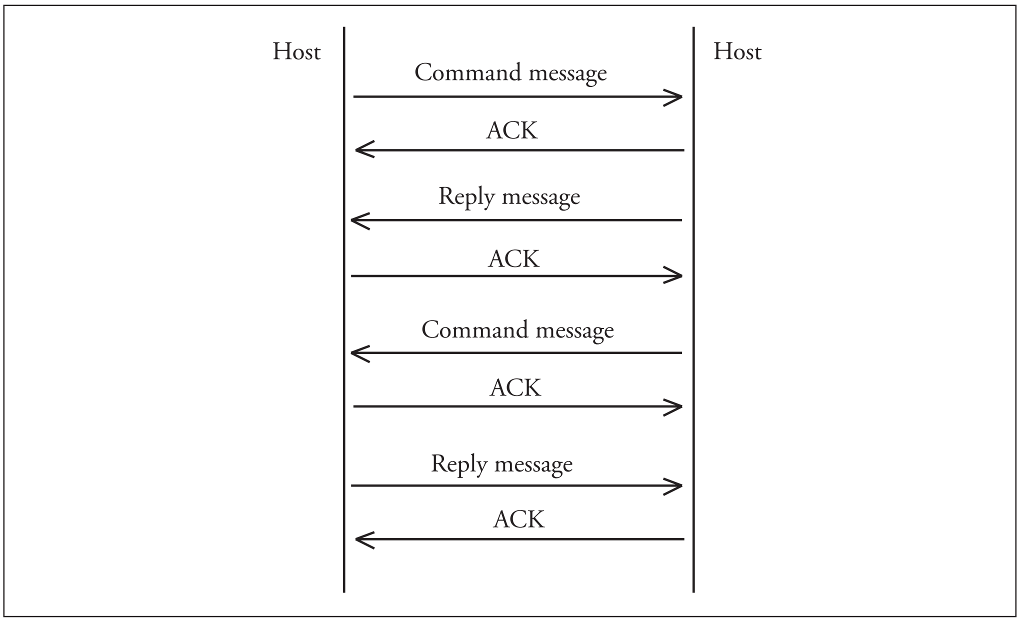


Figure. Communications between the host and the SRIO using full duplex protocol.

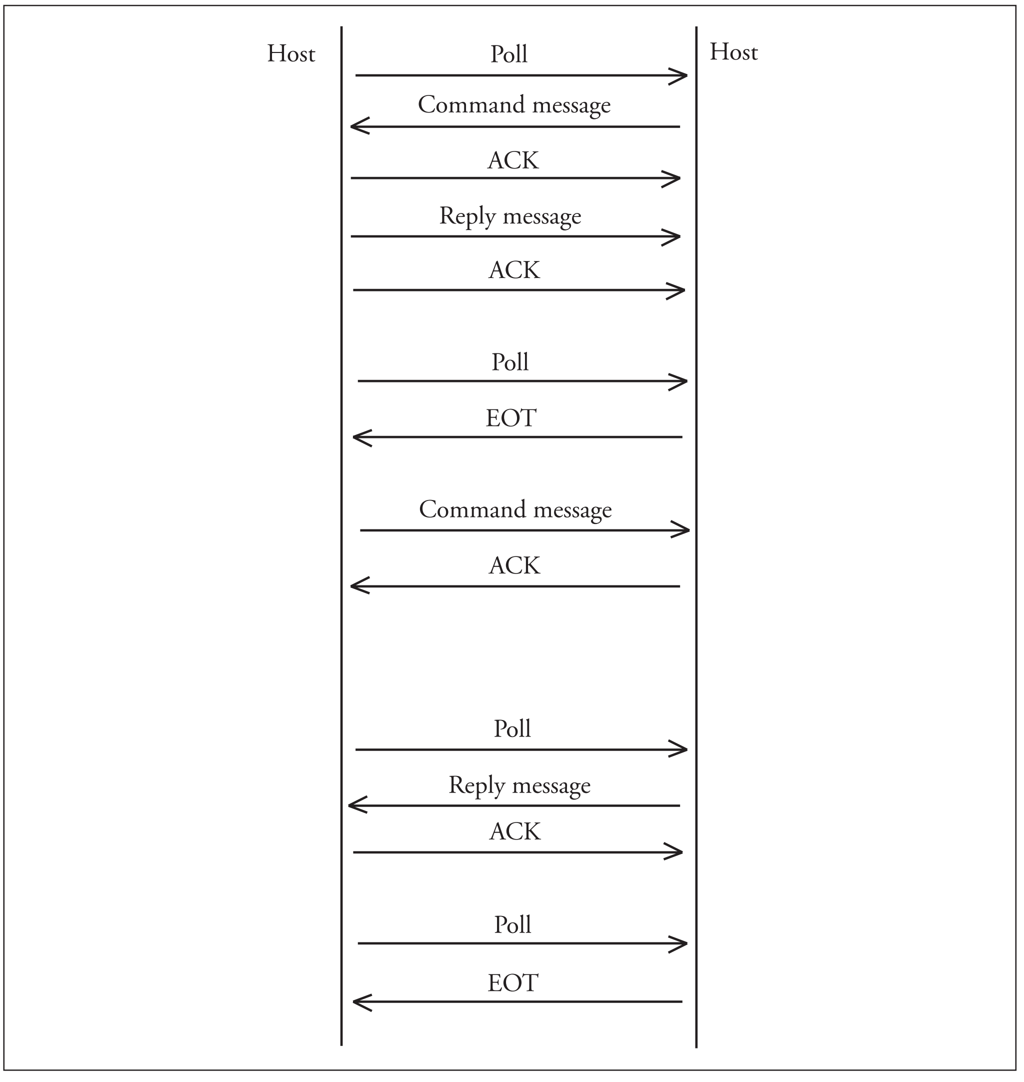


Figure. Communications between the host and the SRIO using half duplex protocol

ABB Oy
Substation Automation
P.O.Box 699
FIN-65101 VAASA
Finland
Tel. +358 (0)10 22 11
Fax.+358 (0)10 22 41094
www.abb.com/substationautomation

1MRS 750885-MUM EN