stage accompany

Stage Accompany

SAnet™

Network for Audio, Video and Lighting Equipment

General Information

Version 1.0

stage accompany

Published by:

Stage Accompany B.V.
Training & Documentation
Anodeweg 4
1627 LJ Hoorn, The Netherlands

First Printing, March 1989

Printed in The Netherlands

stage accompany

Preface

The introduction of intelligent professional audio equipment and the wish to control multiple devices from a central position creates the need for remote control of these devices. Existing computer interfaces and protocols are unsuited for pro audio applications or suffer from high complexity and price. This document describes a newly developed remote control network for professional sound, light and motor/servo equipment, called SAnet (Stage Accompany network).

Other manufacturers are encouraged to implement the network in their own equipment. For these purposes Stage Accompany provides detailed SAnet information, SAnet software interfaces, SAnet device drivers and a SAnet function library. Also, information about commands and responses of Stage Accompany equipment that is provided with a SAnet interface is available from Stage Accompany to enable software developers to write their own control programs.

No doubt SAnet will be compared with the well known MIDI interface. However a comparison is not appropriate, because MIDI is a typical synchronisation interface while SAnet is a typical communication interface. Of course SAnet can be used as a synchronisation interface too.

stage accompany

Table of Contents

Table of Contents

1 Introduction

In 1985, Stage Accompany introduced the PPE-2400; the first commercial available digital controlled analog parametric equaliser. To enable remote control of the presets, the equaliser was provided with a MIDI interface.

During the development of new devices like the Blue Box, however, the need arose to control the complete set of system parameters remotely. We were also looking for a way to efficiently control a large PA system consisting of Blue Boxes.

1.1 Motivation

This paragraph gives a brief overview of the most important design specifications, the existing interfaces and the final decision to develop SAnet.

The four most important interface specifications were as follows:

- The interface needs to have a network-like (multipoint) character to allow for efficient connections.
- The number of separately controllable devices needs to be at least one hundred to control, for example, a PA system of one hundred Blue Boxes.
- The data transfer has to be transparant and error free.
- The maximum distance between the connected devices has to be 500 meters (1600 feet).

The MIDI interface provides for individual control of system parameters with the "exclusive messages mode". But MIDI and other existing interfaces like RS-232 are unsuited to realise efficient connections, as this would require massive bundles of cable. Also, with these interfaces it is not easy to implement a protocol that provides for error correction. Finally, maximum allowable distances between the devices are too short.

Stage Accompany will take care of the administration of the different manufacturer codes. A unique manufacturer code and a list of assigned manufacturer codes can be requested from Stage Accompany. Besides, each manufacturer is free to choose its device type and device serial number.

The SAnet controller uses the identity code to create data links with the network devices. As long as the data link exists, the primary connects the identity code of a device to a unique station address that is used by the 8344 Serial Interface Controller to transfer data to the specific device.

Also the synchronous interfaces known from administrative applications like Ethernet, Starlan, etc. are less suitable because of their high complexity and therefore high cost.

After evaluation of the existing interfaces, we decided to develop a simple and professional network around a special component from Intel; the 8344 Serial Interface Controller.
This controller uses a subset of the standard IBM Synchronous Data Link Control (SDLC) protocol. The most simple application needs only one external component; an inexpensive line driver.

The main features of this network, called SAnet (Stage Accompany Network), are:

- Serial multipoint configuration (efficient wiring)
- Asynchronous block transmission (little overhead)
- Error free data transfer (correction by retransmission)
- High transmission speed (375 kbits/second)
- Distances up to 500 meter (1600 feet)

This manual includes a description of the Physical Layer, the Data Link Layer, and the Network Layer of SAnet according to the OSI (Open Systems Interconnection) reference model.

1.2 Configuration

There are two basic configurations for a communication channel: point-to-point and multipoint. RS-232 is an example of a point-to-point configuration; one device transmits information to one other device. A multipoint configuration, however, is a data link with three or more devices.

In addition, a point-to-point or multipoint configuration can operate either alternate or two-way simultaneous. In two-way alternate operation, the stations take turns transmitting, one at the time. Two-way simultaneous operation allows two stations to transmit and receive at the same time.

A two-way simultaneous configuration is called duplex. A two-way alternate configuration is called half-duplex. SAnet uses a half-duplex multipoint configuration (see Figure 1-1).

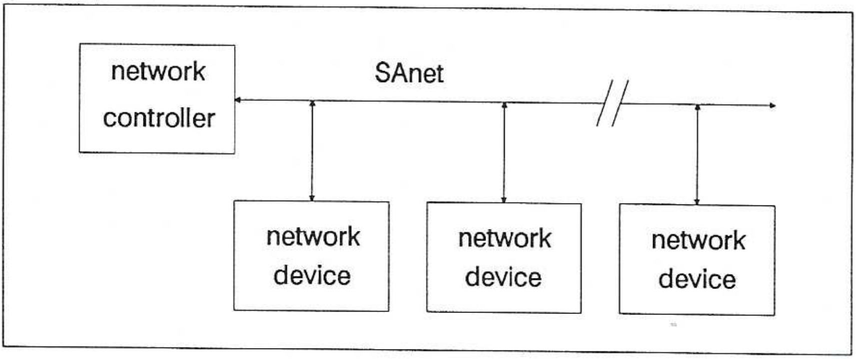

Figure 1-1 SAnet Half-duplex Multipoint Configuration.

A multipoint configuration implies that all network devices as well as the network controller are connected to each other. This is an advantage because the SAnet cable can be routed from controller to device, from device to device, and so on. Therefore SAnet is just one connection that goes from one device to another with each device having a SAnet input and a SAnet output that are internally connected without any buffering electronics.

Because all devices within SAnet are connected by the same cable to the network controller there has to be a way to distinguish them from one another. This is achieved by introducing a unique identity code per device consisting of a manufacturer code, a device type, and a serial number.

A maximum number of 256 manufacturer codes may be used to allow devices from 256 different manufacturers to be connected to SAnet. Per manufacturer, 256 different device types can be used to allow for 256 different types of devices like equalisers, amplifiers, etc... Per device type, 65536 different devices can be used so 65536 equal devices can be uniquely addressed.

2 Physical Layer

The physical layer is concerned with transmitting raw bits over a communication channel. The design issues have to do with making sure that when one side sends a 1 bit, it is received by the other side as a 1 bit, not as a 0 bit. Typical questions here are how many volts should be used to represent a 1 and how many for a 0, how many microseconds a bit occupies, how the connections are made, how many pins the network connector has and what each pin is used for.

2.1 Mechanical Medium

The mechanical medium of SAnet consists of 2-wire coax shielded cable (twinax) and 4-pins XLR connectors. The male connector serves as signal output while the female one serves as signal input. The pin functions of the XLR input and output connectors are as follows:

pin 1 = SAnet ground (cable shield)
pin 2 = reserved for supplying a remote control
pin 3 = non-inverting SAnet connection (+)
pin 4 = inverting SAnet connection (-)

When pin 2 is used to supply a remote control, a 3-wire shielded cable is needed.

The serial transmission speed is 375 kbits/sec, while the maximum allowable distance is 500 meters (1600 feet). This distance can be increased by the use of bidirectional repeaters.

2.2 Electrical Interface

The electrical interface has been implemented according to RS-422. This interface uses a symmetrical line and differential line drivers with a line voltage of 5 volts.

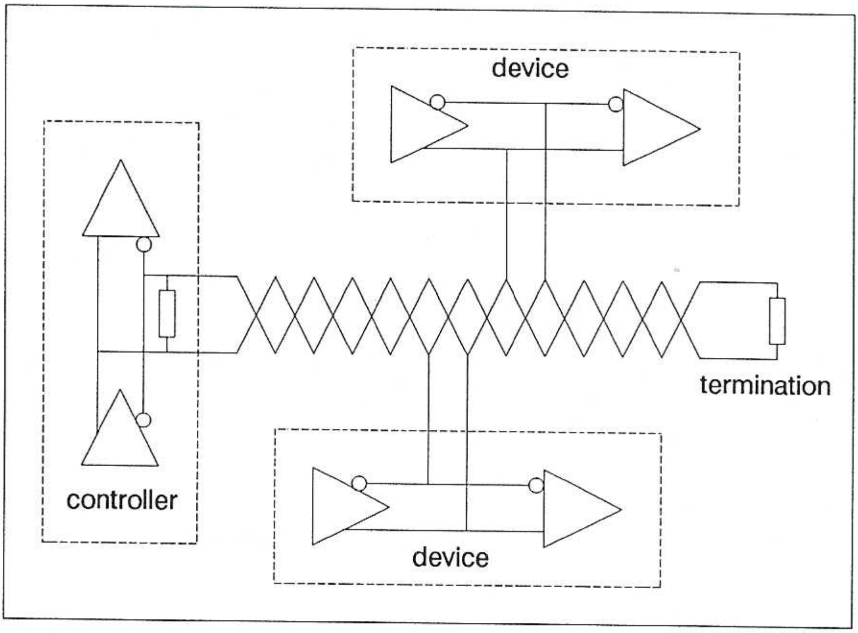

Figure 2-1 SAnet Electrical interface according to RS-422.

Figure 2-1 shows the electrical interface with the differential line drivers.

The output impedance of the primary line driver has to be 75 ohm between the non-inverting and the inverting connections of SAnet. The end of the line should always be terminated with a 75 ohm resistor.

3 Data Link Layer

The task of the data link layer is to take a raw transmission facility and transform it into a line that appears free of transmission errors to the network layer (see next Chapter). It accomplishes this task by breaking the input data up into data frames, transmitting the frames sequentially, and processing the acknowledgement frames sent back by the receiver. Since the physical layer merely accepts and transmits a stream of bits without regard to meaning or structure, it is up to the data link layer to create and recognise frame boundaries. This can be accomplished by attaching special bit patterns to the beginning and end of the frame.

The protocol used in SAnet is based on the IBM Synchronous Data Link Control (SDLC) protocol. This protocol has been accepted as a standard protocol for many high level teleprocessing systems.

Four basic concepts are fundamental to the understanding of the Synchronous Data Link Control protocol. They are:

- The definitions of primary and secondary stations and their responsibilities
- The definitions of the transmission states that affect information transfer
- The way that information is formatted into groups for transfer
- The way these formatted groupings are organised into larger sequences

This chapter describes these four concepts.

3.1 Primary and Secondary Stations

Two types of stations are used in SAnet: a primary station and secondary stations. A primary station has the responsibility for controlling SAnet; it issues commands. Secondary stations receive commands and return responses. All communications on a data link are from the primary station to one or more secondary stations, and from a secondary station to the primary station. There can be only one primary station in SAnet at one time.

The primary contineously transmits requests for data to the secondary stations. These requests are called polls. Every time a secondary station is polled, it may transmit data to the primary station. All secondary stations are polled sequentially. After the "last" secondary station has been polled, the primary station will start all over with the "first" secondary station. Due to the high poll speed it seems that all secondary stations are being polled simultaneously.

Both primary station and the secondary station are based on the 8344 Microcontroller from Intel. The 8344 contains an industry standard 8051 Central Processing Unit and a Serial Interface Unit which is used to communicate with SAnet. The Serial Interface Unit of the 8344 Microcontroller can operate in either of two modes, FLEXIBLE mode or AUTO mode.

The primary station operates in FLEXIBLE mode. In FLEXIBLE mode, reception or transmission of data by the Serial Interface Unit is performed under the control of the Central Processing Unit. The FLEXIBLE mode provides greater flexibility with regard to the kinds of operation permitted.

The secondary station operates in AUTO mode. In AUTO mode, the Serial Interface Unit performs in hardware a subset of the Synchronous Data Link Control protocol called the Normal Response Mode. The AUTO mode enables the Serial Interface Unit to recognise and respond to certain kinds of SDLC commands without intervention from the 8344's Central Processing Unit. The Central Processing Unit can almost entirely be used by the application software. AUTO mode provides a faster turnaround time and a simplified software interface.

3.2 Transmission Frames

All transmissions in SAnet are organised in a specific format called an SDLC frame (see Figure 3-1 on the next page). This format enables the receiving station to determine where the transmission starts and stops, whether the transmission is for that station, what actions to be performed with the transmission, specific information for that station, and data that is used to check whether the frame was received without error. Each SDLC transmission frame has the same specific format and is made up of the following fields:

- A beginning flag (F) that indicates the beginning of the frame
- An address (A) field that identifies the secondary station that is sending (or is to receive) the frame
- A control (C) field that specifies the purpose of the particular frame
- An optional information (I) field that contains information data
- A frame check sequence (FCS) field that enables the receiving station to check the transmission accuracy of the frame
- An ending flag (F) that signals the end of the frame

Each of these fields contain either 8 bits or a multiple of 8 bits (see Figure 3-1).

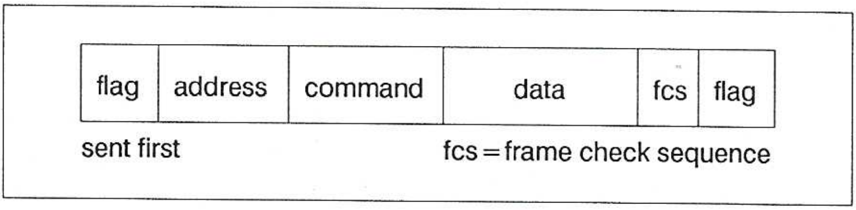

Figure 3-1 SDLC Transmission Frame.

3.2.1 Flags

The beginning flag and the ending flag enclose the SDLC frame. The beginning flag serves as a reference for the position of the A (address) and C (control) fields and initiates transmission error checking. The ending flag terminates the check for transmission errors. Both beginning and ending flags have a unique binary pattern.

3.2.2 Address Field

The address field of an SDLC frame follows immediately after the beginning flag. The address that is sent is always the address of the secondary station on the data link. If the primary station is transmitting the frame, the address is similar to the main address on a letter - it tells who a letter is to. If a secondary is transmitting the frame, the address is similar to the return address on a letter - it tells who the message is from.

In the case of SAnet, a secondary station may have a station address whose value is between 0 and 251. Address 0 is the "no stations address". No secondary station is assigned this as its station address. Address 255 is the "broadcast address" which is not used in SAnet. Addresses 251, 252, 253, and 254 are reserved for future use.

3.2.3 Control Field

Following the address field in an SDLC frame comes the control field. The control field defines the function of the frame. The control field can be in one of three formats: unnumbered format, supervisory format, or information transfer format.

Unnumbered Format

Unnumbered-format frames are used for functions such as:

- Initialising secondary stations
- Controlling the response mode of secondary stations
- Reporting certain procedural errors

Supervisory Format

Frames with a control field of the supervisory format are used to assist in the transfer of information in that they are used to confirm preceding frames carrying information. The frames of the supervisory format do not carry information themselves. These frames are used to confirm received frames, convey ready or busy conditions, and to report frame numbering errors indicating that a numbered information frame was received out of its proper sequence (described later).

Information Transfer Format

Frames with a control field of the information transfer format are - as the name implies - the vehicle for information transfer in SDLC. The control field, besides indicating the format, contains send and receive counts (Ns and Nr), which are used to ensure that these frames are received in their proper order (Ns) and to confirm accepted information frames (Nr). Paragraph 3.3 gives more details on frame numbering.

3.2.4 Information Field

Following the control field, there may or may not be an information field. The supervisory format does not contain an information field.
Data to be transferred on the data link is contained in the information field of a frame. The information field does not have a set length, but must be a multiple of 8 bits (byte). In each byte the low-order bit is sent first and the high-order bit is sent last. In the case of SAnet, the maximum number of data bytes per information frame is 64.

3.2.5 Frame Check Sequence Field

Following the information field (or control field if there is no information field) is the frame check sequence field. The purpose of the frame check sequence (FCS) field is to check the received frame for errors that may have been introduced by the communication channel. This field contains a 16-bits check sequence that is the result of a computation on the contents of the A, C, and I fields at the transmitter. The computation method used is called cyclic redundancy checking (CRC). The receiver performs a similar computation and checks its results. The receiver accepts no frame that is found to be in error.
The FCS field is followed by the ending flag, closing the frame.

3.3 Frame Numbering

A provision is made for transmitting a sequence of numbered information frames and making sure that they are received in the proper order.

A station transmitting numbered information frames counts each such frame, and sends the count with the frame. This count is a sequence number known as Ns. This sequence number is checked at the receiver for missing or duplicated frames.

A station receiving numbered information frames accepts each numbered information frame that it receives (that is error-free and in-sequence) and advances its receive count for each such frame. The receiver count is called Nr. If the received frame is error-free, a receiving station's Nr count is the same as the Ns count that it will receive in the next numbered information frame; that is, a count of one greater than the Ns count of the last frame received. The receiver confirms accepted numbered information frames by returning its Nr count to the transmitting station.

The Nr count at the receiving station advances when a frame is checked and found to be error-free and in sequence; Nr then becomes the count of the "next-expected" frame and should agree with the next incoming Ns count. If the incoming Ns does not agree with Nr, the frame is out of sequence and Nr does not advance. Out-of-sequence frames are not accepted. The receiver does, however, accept the incoming Nr count (for confirmation purposes) if the out-of-sequence frame is otherwise error free.

The Nr and Ns counts of both stations are initialised to 0 by control of the primary station. At other times, the counts advance as numbered frames are sent and received.

In the case of SAnet, the data link only allows one outstanding frame before it must receive an acknowledgement. Immediate acknowledgement allows the secondary station interface to use the AUTO mode. In addition, one outstanding frame uses less memory for buffering, and the software becomes easier to manage.

4 Network Layer

The network layer, sometimes called the communication subnet layer, controls the operation of the subnet. Among other things, it determines the chief characteristics of the primary-secondary station interface. Furthermore it describes how to initiate and end communications.

4.1 Secondary Station Startup and Normal Modes

When power is first applied to a secondary station, it enters STARTUP mode. A secondary station in STARTUP mode is not able to communicate according to the SDLC protocol. It has to be "opened" by the primary station first. The "open" command puts the secondary station in the NORMAL (SDLC) mode. The primary uses the secondary station's identity code (see paragraph 1-2) to initiate SDLC communications so that each secondary station can be addressed individually. The next paragraph describes the "open" command.

4.2 Initiating Communications

The SDLC protocol expects the secondary station to have a unique one-byte station address. Normally, this station address is selected using a DIP switch.

In the case of SAnet, however, it is not convenient to use unique one-byte station identifiers. In paragraph 1.2 it was already mentioned that each device has a unique (4-bytes) identity code. In some way this identity code has to be connected to the station address for as long as the data link exists. The open command provides a way for this connection by transmitting the specified identity code and the station address that is going to be used.

Any secondary station that is in STARTUP mode will receive the open command. It will compare the received identity code with its own identity code. If they match it will take the one-byte station address and make it its own station address. Further it re-initialises its Serial Interface Unit to be able to receive NORMAL (SDLC) commands. Now if the secondary station's user state equals OPEN (see Chapter 6) normal communications are possible through the use of the assigned station address.

The close command is like the open command except that it changes the secondary station's mode from NORMAL (SDLC) to STARTUP.

4.3 Scanning for Serial Numbers

As already mentioned in the previous paragraph the open command needs to have the identity code of the secondary station that is going to be opened. Usually the user must provide the secondary station's indentity codes. To avoid the situation that a user has to search for a large number of identity codes in his equipment SAnet provides a "scanning for identity codes" feature. The user may select a specified manufacturer code and device type and use the scanning command to look for all available serial numbers.
These serial numbers combined with the manufacturer code and the device type can be used to open one or more secondary stations.

4.4 Secondary to Secondary Data Transfer

Because the 8344 Serial Interface Unit of the secondary station is capable of operating in FLEXIBLE mode as well as in AUTO mode (see Chapter 3) it is possible to transfer data from one secondary station to another.
A primary station is not used in this situation so one of the secondary stations has to be the data link controller. This can be done by switching a secondary station to MASTER mode. All other secondary stations which are in STARTUP mode are automatically in SLAVE mode.

A secondary station in MASTER mode is allowed to initiate transmission of data. All other secondary stations receive the data simultaneously. This feature is very usefull to track a large amount of the same devices. Think for example of a PA system with one hundred Blue Boxes. One or more Blue Boxes can be switched to MASTER mode to control all other Blue Boxes.

Remember that the secondary to secondary data transfer is not acknowledged by the receiving station.

5 Network Controller: Primary

SAnet is controlled by a special SAnet controller card that fits in an IBM (-compatible) PC. The controller card (primary station) contains the 8344 Serial Interface Processor that controls SAnet. The software for this processor can be downloaded from the PC to simplify updates, specials, etc...

The PC and the controller share a common Random Access Memory (RAM) to exchange commands, responses, errors, and network data. Because each processor has an address and data path to the RAM, it is called a Dual Port RAM (DPR). When the PC wants to command the primary, it writes the command to the DPR and interrupts the primary. Then the primary tries to get access and reads the command. Also when the primary wants to return responses, errors, or network data, it writes the data to the DPR and interrupts the PC. The PC tries to get access to the DPR and reads the data for further interpretation.

5.1 Operating Modes

When the controller is reset it starts up in boot mode. In this mode the controller can be programmed with new normal mode software. In boot mode the primary can switch to one of four normal operating modes that are listed here:

- return mode
- SAnet controller mode
- SAnet monitor mode without CRC check
- SAnet monitor mode with CRC check

Each mode is briefly described in the following text:

- return mode

In the return mode the primary reads all bytes from the commands area of the DPR and writes them to the responses area. There are no primary commands and responses in this mode. The return mode is usefull for testing the primary-PC interface, PC SAnet device drivers, etc...

- SAnet controller mode

The primary acts as SAnet supervisor in the SAnet controller mode that is described in this chapter.

- SAnet monitor mode without CRC check

In SAnet monitor mode without CRC check the primary listens to SAnet and sends all received SDLC frames inclusive the CRC word to the PC. This is a way to monitor or debug SAnet in real time. Of course you need a second PC with primary to control the examined SAnet. There are no primary commands and responses in this mode.

- SAnet monitor mode with CRC check

This command does exactly the same as the SAnet monitor mode without CRC check, except that it only receives SDLC frames with a correct CRC. The CRC is not returned to the PC.

5.2 PC-Primary Communications

The first byte that the PC writes to the DPR commands area is a station address or group number. The station address is used to determine if either data has to be transmitted or a primary command has to be handled. A station address from 1 up to and included 250 means data to be transmitted to the specified station. A station address equal to 255 means a primary command. Station addresses 0, 251, 252, 253 and 254 are reserved for future use and may not be used!

The format of a "send-data string" is as follows:

<station adress> or <group number>, <n>, <data_1>, [..., <data_n>]

Station address or group number contains the destination of the data. The item "n" contains the length of the data string in bytes. Its range is 1 up to and included 64. Data length 0 is not allowed! The first byte contains a group number if the most significant bit of "n" is set. In that case is the range of "n" 129 up to and included 192. The data is sent to all stations that are member of the specified group. The range of group number is 0 up to and included 31. Group 0 is the default group and contains all opened stations.

The format of a "command string" is as follows:

255, <n+1>, <command>, [<argument_1>, ...<argument_n>]

Item "n" contains the length of the argument string in bytes. Its range is 0 up to and included 255. The arguments can contain station addresses, group numbers, values, flags, etc...

The first byte that the primary writes to the DPR responses area is a station address. The station address is used to determine if either received data has to be transferred or a primary response/error has to be returned. A station address from 1 up to and included 250 means received data to be transferred from the specified station. A station address equal to 255 means a primary response/error. Station addresses 0, 251, 252, 253 and 254 are reserved for future use and are not used!

The format of a "received-data string" is as follows:

<station address>, <n>, <data_1>, [..., <data_n>]

Station address contains the station the data is coming from. Item "n" contains the length of the data string in bytes. Its range is 1 up to and included 64.

The format of a "response/error string" is as follows:

255, <n+1>, <response> or <error>, [<argument_1>, ...<argument_n>]

Item "n" contains the length of the argument string in bytes. Its range is 0 up to and included 255. The arguments can contain commands, station addresses, group numbers, values, flags, etc...

5.3 Boot Mode Commands

The SAnet boot mode (see paragraph 5-1) commands are listed here:

- get operating mode
- get software version
- read comment string
- write comment string
- switch to normal mode
- write normal program record
- write normal program checksum

A brief explanation of each command is given in the following text.

- get operating mode

This command causes the primary to return its operating mode.

- get software version

This command causes the primary to return both its boot software version and its normal software version.

- write comment string

With this command a comment string can be added to the normal mode software during software programming. For example, this string may contain update comments, date of programming, reminders, etc... The maximum length of the string is 80 characters.

- read comment string

This command returns the above mentioned comment string.

- program normal mode software record

This command writes a normal mode software record to a specified address in the EEPROM. It returns the last address written and a flag that indicates if the programming was successfull. The maximum length of the record is 32 bytes.

- program normal mode software checksum

After programming the normal mode software, this command must be used to generate the normal mode software checksum. The checksum is written in EEPROM.

- switch to normal mode

This command must be used to switch from boot mode to the specified normal mode. Each time before the controller switches from boot mode to normal mode, the normal mode software checksum is verified.

5.4 Primary Commands

The SAnet primary commands are written by the PC to the commands area of the DPR. The primary reads the commands and starts the interpretation. The SAnet primary commands are listed here:

- get operating mode
- get software version
- turn send-arguments check on/off
- set station timeout reply time
- set station watchdog delay
- scan for stations
- open a station for communications
- close a station
- define a group of stations
- undefine a group of stations
- define the fast-polled stations group
- undefine the fast-polled stations group

A brief explanation of each command is given in the following text.

- get operating mode

This command causes the primary to return its operating mode.

- get software version

This command causes the primary to return both its boot software version and its normal software version.

- turn send-arguments check on/off

When data is transferred from PC to primary, the primary may check the arguments on illegal station addresses, group numbers, data length, etc. The primary will return an appropriate error message if an error occurs and the send-arguments checking is on. The default state of the send-arguments checking is on. To get higher data throughput the send-arguments checking can be turned off (and on) with the "send-arguments check on/off" command.

- set station timeout reply time

The primary will generate a timeout error if it loses contact with a secondary station. The rate of timeout errors can be set in steps of 40 milliseconds with the "station timeout reply time" command. For example with a reply of 25 the primary will generate a timeout error every second (25 * 40 milliseconds). The station timeout time increases slightly when the number of stations connected to SAnet increase.

- set station watchdog delay

The primary continuously sends watchdog frames to all stations to indicate that it is still polling. This is done because the secondary stations which are in AUTO mode can not determine that they are being polled by the primary station. The watchdog mechanism causes the secondary station to automatically return to STARTUP mode after a specific time when it is not being polled by the primary station anymore. This feature is very usefull to start all over after a network failure due to cable problems. More information about the watchdog mechanism can be obtained from Stage Accompany.

- scan for stations

This command causes the primary to scan SAnet for connected stations with a specific manufacturer code and device type. The primary returns all serial numbers up to a predefined maximum. In this way the programmer or user does not have to know all serial numbers of the secondary stations. More information about station scanning, manufacturer code, device type and serial numbers can be found in paragraph 4.3.

- open a station for communications

This command causes the primary to open a secondary station for communications using the station's manufacturer code, device type and serial number. After it has been opened, the primary starts polling the secondary station and data can be transmitted to and received from the station.

- close a station

The communications with a station can be ended using the "close station" command. This causes the station to return to STARTUP mode (see chapter 6). The primary stops polling the station.

- define a group of stations

A group of stations can be defined for transmitting the same data to more than 1 station. Group 0 is the default group. It contains all opened stations, so when a station is opened it is added to group 0.
A maximum of 31 groups can be defined (1-31). The memory for the groups can hold at most 1000 stations and is allocated dynamically. If the memory is fragmented after successive group defines and undefines, a "not enough memory" error may occur. The best way to handle this error is to undefine all groups and redefine them.

- undefine a group of stations

When a group of stations is not used anymore it can be removed using the "undefine group" command.

- define the fast-polled stations group

This command causes the primary to define a group of stations which are polled every 20 milliseconds. Commonly, this fast-polled group will contain one or more stations that have to be monitored in real time. The fast-polled stations are not removed from there normal group.

- undefine the fast-polled stations group

The fast-polled group can be undefined using this command.

5.5 Primary Responses and Errors

As a result of command interpretation the primary may generate responses or errors. The three responses are listed here:

- return response
- done response
- debug response

A brief explanation of each response is given in the following text.

- return response

A return response is used to return response data generated by a command. Examples are the commands: get operating mode, get software version, scan for stations, open station, etc...

- done response

A done response is used to signal the end of the interpretation of a command. An example is the scan for stations command.

- debug response

A debug response is used to send debug data from primary to PC. The arguments contain a data length and a data string. It is only used during software development.

errors: commands:	0	1	2	3	4	5	6
command length error	X	X	X	X	X	X	X
data length error	X						
illegal station address error	X	X	X	X		X	
duplicate station address error				X		X	
illegal station address string err.				X		X	
station not open error	X		X	X		X	
station already open error		X					
illegal group number error	X			X	X		
group already defined error				X			
group not defined error	X				X		
not enough memory error				X			

Figure 5-1 Primary commands versus Errors

Besides responses there are a number of errors that may occur during the primary command interpretation. Figure 5-1 shows an overview of primary commands and there possible errors. The commands are listed on the next page:

0 - send data to station or group (not a primary command)
1 - open a secondary station for communications
2 - close a secondary station; stop communications
3 - define a group of stations
4 - undefine a group of stations
5 - define a group of fast-polled stations
6 - undefine a group of fast-polled stations

All primary command errors or responses are followed by the command that caused the error or response and some optional arguments like station addresses or group numbers. A brief explanation of each error is given in the following text.

- illegal command error

An illegal command error occurs when the primary receives an undefined command from the PC.

- command length error

A command length error occurs when the primary receives a command from the PC with a wrong argument length number.

- data length error

A data length error occurs when the length of the data string in a send data command equals 0 or is greater than 64.

- illegal station address error

An illegal station address error occurs when a station address equal to 0 or greater than 250 is used as an argument to various commands.

- duplicate station address error

A duplicate station address error occurs when there are two equal station addresses in a station address string.

- illegal station address string error

An illegal station address string error occurs when the length of the station address string equals 0 or is greater than 250.

- station not open error

A station not open error occurs when a not opened station is used as an argument to various commands.

- station already open error

A station already open error occurs when an open station command on an already opened station is interpreted.

- illegal group number error

An illegal group number error occurs when a group number greater than 31 is used as an argument to various commands.

- group already defined error

A group already defined error occurs when a define group command on an already defined group is interpreted.

- group not defined error

A group not defined error occurs when an undefined group is used as an argument to various commands.

- not enough memory error

A not enough memory error occurs when there is not enough primary system memory (RAM) to execute a command.

5.6 Network Errors

Network errors are generated by the primary when there is a problem with one or more secondary stations. The network errors are listed here:

- watchdog error
- send data error

All network errors are followed by an SDLC error (see next paragraph) and the station address of the station that caused the error. A brief explanation of each error is given in the following text.

- watchdog error

A watchdog error occurs when the primary is not able to transmit a watchdog frame to a station.

- send data error

A send data error occurs when the primary is not able to transmit a data frame to a station.

5.7 SDLC Errors

When there is a communication problem with a secondary station, SDLC errors specify the error. The SDLC errors are listed here:

- timeout error
- data link error
- not ready error
- no unnumbered acknowledgement error
- send sequence error
- receive sequence error
- buffer overrun error
- unknown command error

All SDLC errors are followed by the station address that caused the error. A brief explanation of each error is given in the following text.

- timeout error

A timeout error occurs when a station does not respond within 10 milli-seconds to a primary poll command.

- data link error

A data link error occurs when the primary is not able to put a station into Normal Response Mode (NRM).

- not ready error

A not ready error occurs when the primary transmits a data frame to a station whose receive buffer is full.

- no unnumbered acknowledgement error

A no unnumbered acknowledgement error occurs when a station does not respond with an Unnumbered Acknowledgement (UA) to a Set Normal Response Mode (SNRM) command or a Go Disconnected Mode (DISC) command.

- send sequence error

A send sequence error occurs when the number of data frames sent by the primary are not equal to the number of data frames received by a station.

- receive sequence error

A receive sequence error occurs when the number of data frames received by the primary are not equal to the number of data frames sent by a station.

- buffer overrun error

A buffer overrun error occurs when a station receives a data frame from the primary with a data length exceeding 64 bytes.

- unknown command error

An unknown command error occurs when a station receives an undefined SDLC command from the primary.

5.8 PC SAnet Controller Board

This paragraph describes the hardware of the SAnet controller board that fits in an IBM (-compatible) PC.

5.8.1 PC I/O Address DIP Switch Settings

The DPR (Dual Port RAM) arbitration takes 8 addresses of the PC's I/O space. With the DIP switch on the primary PC board any 16-bytes block can be selected in the I/O space of the PC from address 300 hex to 400 hex. Only the lower 8 bits of the selected memory block are used. The factory setting (default) is 390 hex.
The DPR itself uses 8 Kbytes of PC memory space. With the present primary board this 8 Kbytes block can be mapped within the D000 hex memory segment. The PC SAnet device driver determines where the 8 kbytes block is located in the D000 hex memory segment (8 options).

5.8.2 Opto Couplers and SAnet Connections

When using 500m of cable, interference may cause malfunction of the SAnet controller PC. To avoid this problem the total SAnet has been separated from the PC's circuit by means of opto couplers.

The line driver that drives SAnet is a bidirectional symmetrical receiver-transmitter that is protected against short circuit and high common mode voltages from outside.

The configuration of the 9-pins sub-D connector is as follows:

1 - Shield
2 - Shield
3 - Shield
4 - Inverting output (-)
5 - Non inverting output (+)
6 - Shield
7 - Shield
8 - Inverting output (-)
9 - Non inverting output (+)

5.8.3 Status LEDs

The primary board has five status LEDs which show the status of the Dual Port RAM arbitration. The function of each LED from high to low is to indicate that:

- the primary has Dual Port RAM access (yellow)
- the PC has Dual Port RAM access (yellow)
- the Dual Port RAM is in use (red)
- there is data for the PC available (green)
- there is data for the primary available (green)

6 Network Station: Secondary

As the primary station, the secondary station is based on the 8344 Serial Interface Controller from Intel. Existing 8051 applications could add high performance SAnet communications capability by linking the SAnet interface to the existing software and providing additional software to be able to communicate with SAnet.

The SAnet Secondary Station interface was written as a general purpose SAnet driver. It was written to be linked to an application module. The application software implements the actual application in addition to interfacing to SAnet. The SAnet interface is independent of the main application, it just provides the SAnet communications.

6.1 SAnet Interface Open and Closed States

The secondary station SAnet interface can be in two states: OPEN or CLOSED. If the SAnet interface is closed SAnet can not be used for communications. The secondary station does not respond to an open command from the primary. If the secondary station application wants to communicate using SAnet it changes the SAnet interface state from CLOSED to OPEN. Now if the secondary station receives an open command from the primary it will go from STARTUP mode into NORMAL (SDLC) mode to make communications possible.

6.2 SAnet Interface Routines

The SAnet interface consists of two parts: a communications interface and a user interface. The communications interface is the part of the software which controls the SAnet communications. It handles link access, command recognition/response, acknowledgements, and error recovery. The user interface provides ten functions which the application programmer may use to communicate using SAnet. These functions are common to many I/O drivers like keyboard/CRT, and asynchronous communication drivers.

Only the user interface is described here. A block diagram of the user interface software is shown in Figure 6-1 on the next page.

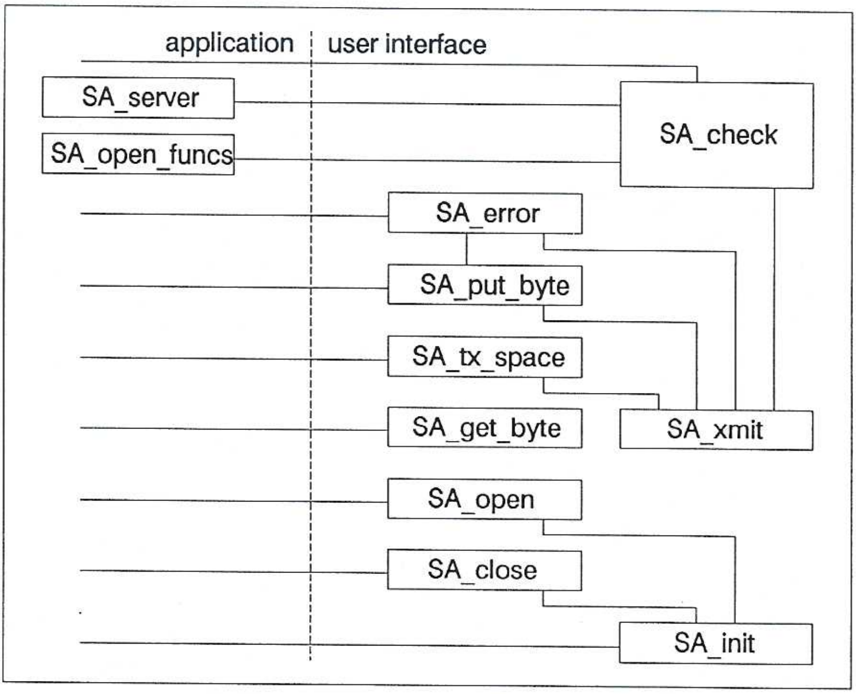

Figure 6-1 SAnet User Interface Software Block Diagram.

Two or more routines connected by a solid line means the routine above calls the routine below. The left part is the application part. The routines in the application part must be written by the application programmer. The lines that go from left to right mean that the routines in the right part are called from the application.

The routines of the user interface are listed here:

- SA_init
- SA_open
- SA_close
- SA_check
- SA_error
- SA_put_byte
- SA_get_byte
- SA_tx_space
- SA_open_funcs
- SA_server

The application calls the SA_open routine to change the SAnet interface state to OPEN and to enable SAnet communications.
If the application wants to end communications it calls SA_close to change the SAnet interface state to CLOSED and to abort SAnet communications.

The main loop of the application program must contain the SA_check routine call to check the SAnet interface for received data. The application programmer has to provide for the SA_server and SA_open_funcs routines to actually handle the received data. Both routines are application dependent.

The application can read bytes from SAnet using the SA_get_byte routine. Transmission space is checked by the SA_tx_space routine. It is not possible to split a data block and transmit both parts with different frames. Space for a data block must be checked before writing the block to the transmit buffer. The data bytes can be written to the transmit buffer using the SA_put_byte routine.

If an unknown or illegal command is interpreted the SA_error routine can be called. This routine returns the command with the most significant bit set to the primary and aborts the interpretation.

The following text gives a description of each routine.

- SA_init

SA_init is called from the SA_open and SA_close routines and initialises the SIU hardware and special purpose registers. Usually it is also called from the SA_server routine.

- SA_open

The SA_open routine sets the user state of the station to OPEN and initialises the SIU hardware by calling SA_init. Now when a Set Normal Response Mode command from the primary is received the station responds with an Unnumbered Acknowledgement and enters into Normal Response Mode.

- SA_close

SA_close sets the user state to CLOSED and puts the SIU out of AUTO mode by calling SA_init. This causes the primary to disconnect the station. All communications are aborted.

- SA_check

SA_check compares the put and get pointers of the circular receive buffer to test if any data has been received. If this is the case SA_check calls the SA_server routine. After handling the data SA_check looks for an information frame to transmit. If the station has just been opened SA_check calls the SA_open_funcs routine to perform unique actions.

- SA_error

SA_error can be called by the application software if an error occurs during data interpretation. The most significant bit of the illegal or unknown command is set and the command is returned to the primarysx. After that interpretation is aborted by resetting both the put and the get pointer of the circular receive buffer.

- SA_put_byte

This routine writes one byte to the Serial Interface Unit's transmit buffer and increments the transmit buffer index. Be sure that there is space for a byte in the transmit buffer using the SA_tx_space routine before calling the SA_put_byte routine.

- SA_get_byte

The SA_get_byte routine reads one byte from the receive buffer if the put and the get pointer are not equal and increments the get pointer. If both pointers are equal this routine returns 0 without incrementing the get pointer.

- SA_tx_space

SA_tx_space checks for enough transmit buffer space for a specified amount of bytes. It transmits the buffer first if there is not enough space. It returns true if there is enough space, otherwise it will return false.

- SA_open_funcs

SA_open_funcs is called by the SA_check routine just after the station has been opened. This routine must be written by the application programmer. Usually it contains statements to initialise the watchdog mechanism, refresh the displays, etc...

- SA_server

SA_server is called by the SA_check routine if data must be interpreted. This routine must be written by the application programmer. Usually it contains a command decoder.

6.3 Secondary Station Hardware

Like the primary station, the secondary station uses the 8344 Serial Interface Controller from Intel. The industry standard 8051 part of the 8344 can be used as main controller or input/output controller of the station. Figure 6-2 on the next page shows the hardware schematic diagram of the 8344 with the SAnet line transceiver.

There are many possibilities to implement the secondary station. Think, for example, of a multi-processor device with an 8344 Serial Interface Controller as a dedicated SAnet processor. Besides using the 8344, there are a number of different serial interface controllers which can be used. Examples are the 8273 Programmable HDLC/SDLC Protocol Controller from Intel, the Z8530 Serial Communications Controller from Zilog, etc... Notice that those components have lower performance than the 8344 because they handle network commands by software in stead of hardware.

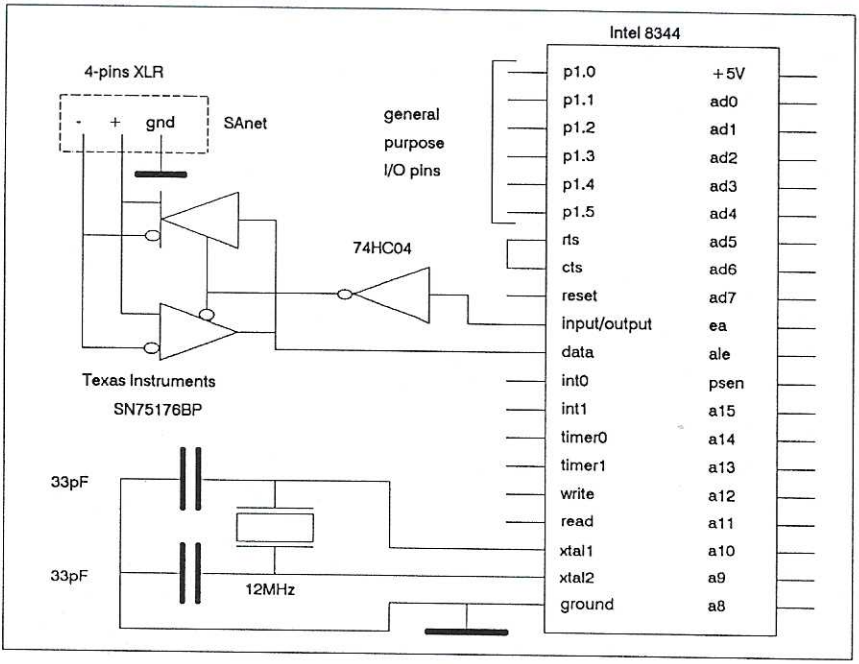

Figure 6-2 Secondary Station Hardware Schematic Diagram.

7 PC SAnet Programming Support

To support the programming of SAnet applications on the PC, an SAnet toolkit is available from Stage Accompany in the near future. It inludes a device driver, a function library and a demonstration program written in C.

7.1 SAnet Function Library

The SAnet function library is intended to be used in a Microsoft C programming environment. The following functions are available at this moment:

- SA_log_start
- SA_log_stop
- SA_log_txt
- SA_open
- SA_close
- SA_open_station
- SA_close_station
- SA_write_ready
- SA_write
- SA_read_ready
- SA_read
- SA_send_station
- SA_send_all

A brief description of these functions is given in the following text.

- SA_log_start

The SA_log_start function opens a log file and enables the PC application programmer to log the network data for detailed examination.

- SA_log_stop

The SA_log_stop function closes the current log file and ends the logging of network data.

- SA_log_txt

The SA_log_txt function can be used to write a debug text string to the current log file to ease the interpretation of large log files.

- SA_open

The SA_open function resets the primary board and puts it in a specified mode. It also opens a file handle to enable communication with the SAnet device driver.

- SA_close

The SA_close function ends SAnet communication by closing the file handle of the SAnet device driver.

- SA_open_station

The SA_open_station function initiates communication between the PC and a secondary station. The secondary stations manufacturers code, device type, serial number and a station address are used as parameters.

- SA_close_station

The SA_close_station function ends communication with a secondary station.

- SA_write_ready

The SA_write_ready function is used to determine if data can be written to the SAnet primary by checking the Dual Port RAM semaphore and the primary "data read" flag.

- SA_write

The SA_write function is used to actually write data to the SAnet primary.

- SA_read_ready

The SA_read_ready function determines if there is data available in the SAnet device driver.

- SA_read

The SA_read function is used to actually read data from the SAnet primary.

- SA_send_station

The SA_send_station function is used to transmit data to one or more secondary stations.

- SA_send_all

The SA_send_all function is used to transmit data to all opened secondary stations.

7.2 SAnet Device Driver

The SAnet device driver provides an interface between the hardware of the primary board and the SAnet function library. It must be included in the file CONFIG.SYS. The following parameters can be adjusted:

- Dual Port RAM memory segment
- I/O base address
- Interrupt number
- Receive buffer size
- Device driver name

8 Applications

SAnet is a universal network with numerous applications. This chapter tries to present some examples.

From a central point equipment can be controlled separately or in groups. The kind of control depends on the application. Stage Accompany use an IBM (-compatible) PC as the central operating point of SAnet. By means of this PC all parameters of our processor controlled equipment can be controlled remotely. An advantage using a PC as central operating point is the fact that settings of all parameters can be stored on floppy- or harddisk. Presets can be created by storing combinations of parameters. The use of presets will decrease sound check time significantly. Besides recalling presets manually, it is possible to synchronise them to a master like SMPTE, MIDI or a Compact Disc player.

Another advantage using a PC is to make system control a lot easier by using the graphic capabilities of the PC's screen. Think for example of a parametric equaliser with its many parameters. In stead of showing the user a list of gains, central frequencies and Q-factors, it is possible to edit and show the actual frequency response.

Monitoring is another important application of SAnet. Routine tasks like power and temperature monitoring can be done by the PC. If a threshold is exceeded, the PC can automatically perform some action to avoid undesired situations.

All Stage Accompany processor controlled equipment have an internal log-book that stores several system parameters like hours of operation, duration of overloads, etc. These data can be accessed through SAnet and can be used as a basis for a maintenance system.

Another application of SAnet is downloading of control software to update a secondary station. Of course this implies the use of a reprogrammable program memory. Besides updating secondary software, it is easy to develop custom made programs that can be downloaded as well.

SAnet can also be used without a controller card by turning a secondary station into a primary station. For instance, a Blue Box that has been switched to MASTER mode can be used to control other Blue Boxes. In this way a complete PA system of Blue Boxes can be controlled by one Blue Box.