Boehringer Mannheim GmbH

Keysys

Clinical Chemistry Analyzer

Host Interface Manual

Keysys Analyzer Host Interface Manual

Including features of version 1.00.15

BM / ID. No. 1709984-001

Document:	BM_KH1.DOC
Author:	Claus Mahlmann
Printed:	28.07.97

Boehringer Mannheim
Department LI-TD

Reformated by: A.B. Williams - docuware GmbH

1. Introduction

This manual describes the connection of the **Boehringer Mannheim GmbH Keysys Clinical Chemistry Analyzer** to a 3rd Party Laboratory Host System via an RS232C asynchronous serial port. It details the communication protocols, the required hardware, and considerations to be taken when interfacing different computer systems.

The following fundamental questions are considered:

Hardware

- What interface hardware is required?

How is the physical connection established?

Operation and Control of Data Transmission

- How to set transfer and communication parameters?

What is the format of the transmitted data and how is it used?

- What type of data can be transmitted?
- How is the data transfer initiated?

Software

- What communication protocol is used?
- What are the Host's software requirements?

The **Keysys Clinical Chemistry Analyzer** Host Interface is a reliable and intuitive method of transferring data to and from a Host computer.

Difficulties with installation or questions can be referred to the Boehringer Mannheim Service Department or alternatively the Service Management of the Central Marketing Department of Boehringer Mannheim GmbH (Germany):

Boehringer Mannheim GmbH
Technical Product Management / Data Technique
Sandhofer Straße 116
D-68305 Mannheim

Tel: (49) 621 / 759-2464
Fax: (49) 621 / 759-4394

2. Basic Operation

The Keysys Analyzer is a state-of-the-art clinical chemistry Analyzer for low to medium volume use. It functions via an embedded computer system that controls the hardware and the user interface. The user interface comprises of a 40 x 20 LCD alphanumeric display and a touch sensitive keyboard.

The Keysys Analyzer may be used in a variety of system configurations:

2.1 Standalone System

The Keysys is a standalone system controlled entirely from the integrated user interface. The operator can request and run tests, print results and perform maintenance with the unit itself. It is suitable for hospitals or small, specialized or satellite laboratories that do not have an electronic information infrastructure.

2.2 RS232 Host Connection

The Keysys uses an RS232C interface to connect to an existing network. The Analyzer can retrieve test requests from the Host system in batch or on-line mode. The results are returned as soon as they are available for processing and storage by the Host. This allows the Host to implement the more sophisticated operations such as advanced reporting capabilities, quality control programs, etc. The Host system may be a personal computer a mini-computer or a mainframe.

- The Analyzer can receive or transmit data via the RS232 interface and simultaneously perform an analysis. Test requests can be made by the built-in touch sensitive keyboard or, in batch mode, via the Host interface. The Analyzer can also interrogate the Host for the bar-coded samples that have no test selection information available.
- The Personal Computer can be located up to 15 meters away from the Keysys Analyzer (RS232 cabling limit) and still allow low cost hardware to reliably operate the Host communication line with adequate speed. The Analyzer has a wide range of functions. It can relay information to and from the Analyzer to a more powerful computer system and convert the protocols if required. Alternatively it can take on all or part of the Host’s functionality.
- A minicomputer or mainframe based Host system allows the Keysys to fully utilize an existing laboratory specific infrastructure. It is possible to directly connect the Analyzer to most of these systems. However, it is generally more efficient and cost effective to use a Personal Computer.

2.3 Ethernet Host Connection

2.3.1 Introduction

Using the Keysys with a built-in Ethernet interface allows integration with a LAN (local area network) environment. The Keysys can then become part of a greater network infrastructure. The basic operation is the same as described in “RS232 Host Connection”, but the LAN provides faster and more reliable transmission of data. Several Analyzers, PCs, mini-computers or mainframes can all be connected to the same network and share resources. A single computer can fulfill the role of “Host” by communicating directly (point to point) with the Keysys Analyzer.

2.3.2 The Communication Mechanism of the RPC Transmission Layer

All data communications shall use the “string” type only. The string data are defined in chapter 5.2.

The IP address of the “host” is specified in the “System Parameters” screen in the Keysys user interface. The default of the system software is “192.1.1.1”. The user interface allows to specify a host name. If this string is not empty, the host name will be used.

The IP address of the Keysys analyzer is also specified in the “System Parameters” screen in the Keysys user interface. The default of the system software is again “192.1.1.1”. The user interface allows to specify a Keysys name. If this string is not empty, the Keysys name will be used.

Keysys uses TCP/IP protocol for ONC Sun RPC.

- Keysys has a server task to receive command from a host by RPC. Keysys calls a “`registerrpc()`” function with a program number “0x29950101”, a version number “1” and a “`xdr_wrapstring`” parameter to register a server task.
- Keysys has a client task to transmit data to a host by RPC. Keysys calls a “`clnt_create()`” function with a program number “0x3F000000”, a version number “1” and a protocol TCP to create a client task. And Keysys calls a “`clnt_call()`” function with a program number “0x3F000000” and a “`xdr_wrapstring`” parameter.
- A host should have a client task to send command to Keysys. A host should call a “`clnt_create()`” function with a program number “0x29950101”, a version number “1” and a protocol TCP to create a client task. And a host should call a “`clnt_call()`” function with a program number “0x29950101” and a “`xdr_wrapstring`” parameter.
- A host should have a server task to receive data from Keysys. A host should call a “`registerrpc()`” function with a program number “0x29950101”, a version number “1” and a “`xdr_wrapstring`” parameter to register a server task.

2.3.3 ECOM - A low level ethernet communication module

This description is intended to be used by software developers who like to build applications which need to communicate with the Keysys analyzer via the ethernet port.

The ECOM module encapsulates the NEWT RPC- and TCP/IP-stack. It frees the programmer from the need to have full acknowledge about the detailed underlying communication mechanism. The ECOM module may be directly used in programs written in the C or in the C++ programming language. It is compatible with the Microsoft Visual C++ compiler V1.5, but it can be easily adapted to other development environments.

2.3.4 Requisite Knowledge

The material in this manual and the source-code assumes that you are familiar with developing in C/C++ for the Microsoft Windows operating system. It is helpful if you have basic knowledge about TCP/IP networking.

2.3.5 Support Diskette

The following software components are available on a diskette which can be ordered from Technical Service department (Tel: 2464, Fax: 4394)

File name	Description
RPCINTER.C	C coded RPC interface to the Keysys analyzers
RPCINTER.H	Declarations for RPCINTER.C
RPC.MAK	Make file for ECOM objects using Microsoft C
RPC_KS.X	RPC program, version and function declarations for the Keysys client
RPC_PC.X	RPC program, version and function declarations for the PC service

2.3.6 System Requirements for development

- A 386-PC or higher running under Microsoft Windows for Workgroups V3.11.
- An ethernet network card.
- At least 8 MB of RAM and 100 MB of free disk space for the compiler and the TCP/IP stack.
- Microsoft Visual C/C++ Compiler V1.5 including the Microsoft Windows SDK.
- Chameleon TCP/IP stack V4.01 of Net*Manage*, Inc. for Microsoft Windows for Workgroups.
- NEWT RPC- and TCP/IP-SDK V3.11.2 of Net*Manage*, Inc.

2.3.7 Before you begin

Before you start developing your own application you should install all hardware and setup all software components. You should follow the installation guides provided with the software.

It is important that you assign an unique TCP/IP address to your PC and to the Keysys analyzer. The Keysys analyzer must be setup for ethernet communication and you have to enter the TCP/IP addresses of the analyzer and of your PC.

After this basic setup has been completed it will be a good idea to test the functionality of the TCP/IP communication with the Chameleon PING-utility.

In the next step you should compile the ECOM objects using the makefile RPC.MAK. The calling sequence for this task from the DOS command line is:

```
NMAKE -f RPC.MAK
```

2.3.8 Building your own application

You must link our application with the following libraries and objects:

Name	Source
RPC_PC_S.OBJ	ECOM, generated by RPC.MAK
RPC_KS_C.OBJ	ECOM, generated by RPC.MAK
RPCInter.OBJ	ECOM, generated by RPC.MAK
RPC.LIB	NEWT SDK
NMPCIP.LIB	NEWT SDK

2.3.9 Coding guidelines

Before the first command can be transmitted to the Keysys analyzer using the ECOM function *RpcTransmit* you must call the ECOM function *RpcCtorInterface* to initialize the RPC communication. This function requires that you provide a pointer to a call-back function which will be invoked if a message has been received from the Keysys analyzer. Before quitting your application you should call the ECOM function *RpcDtorInterface* to shutdown the RPC communication.

2.3.10 ECOM Reference

Name:	**enum RpcErrorCodes**
Purpose:	Providing identifiers for error codes returned by various ECOM functions
Enumerators:	
NoRpcError	Indicates a successful completion.
InitializationFailed	Installation of the RPC service has failed.
ClientCreateFailed	The NEWT clnt_create function has failed.
RemoteCallFailed	The ECOM rpc_keysys_1 function has returned NULL.
AbnormalReturnValue	The Keysys analyzer hasn’t returned 0 in response to a RPC message.

Name:	**RpcCtorInterface**
Purpose:	Initializes the RPC communication.
Return value:	see enum RpcErrorCodes for a detailed description.
Arguments:	
unsigned hApplication	Handle of the application instance
pRpcReceive pReceiveCallback	Pointer to a call-back function with the prototype
int ReceiveCallbackFunction (const char * pMessage);	This functions is invoked via the RPC mechanism by the Keysys analyzer if it want’s to transmit a message to the PC. The function must return 1 if it is successful and 0 otherwise. The argument pMessage contains a pointer to the zero terminated message received from the Keysys analyzer. The storage referenced by pMessage may not be used after the returning from the receive call-back function.

Name:	**RpcDtorInterface**
Purpose:	Shuts down the RPC communication.
Return value:	None

Name:	**RpcTransmit**
Purpose:	Transmits a string to the Keysys analyzer.
Return value:	see enum RpcErrorCodes for a detailed description.
Arguments:	
const char * pKeysysAddress	Zero terminated string containing the TCP/IP address or the host name of the Keysys analyzer. Note: If you like to use host names, you have to setup the hosts file which is located in your windows directory.
const char * pMessage	Zero terminated string containing the message (command) to be transmitted to the Keysys analyzer.
const char ** ppErrorMessage	The address referenced by *ppErrorMessage must be large enough to store a pointer to a character and the storage must be allocated by the caller. Upon return *ppErrorMessage contains NULL if no error message is available. Otherwise the address of the zero terminated TCP/IP error message string is store there. The error message is valid until the next ECOM function call.

3. Communication Modes

The Analyzer can be used in three communication modes independent of the Host ”type” (RS232C or Ethernet):

3.1 Standalone Unit (No Host Connection)

The Keysys is fully functional as a *stand alone* unit. Test selection data is entered via the keyboard and the results are output, either when available or at a later date, by the integrated printer. There are no patient oriented data handling or quality control facilities. There is, however, limited inventory control for reagents.

In this mode the work flow is:

- Input the test requests at the Analyzer keyboard.
- Place the samples on the instrument.
- The Analyzer processes the samples and prints out the results on the integrated printer.

3.2 Host Connection: Batch Test Selection Mode

In this mode it is possible to download test selection data in a batch before (or during) a run. This is possible for all samples including STAT, calibrator and control samples. Test selection data entered via the keyboard, and the batch download data is merged into one common test selection list (“local Analyzer database”). After the measurement is completed the results are uploaded to the Host in “real-time” (RES frame). Transmission can be repeated later in a batch. If the connection to the Host fails the Analyzer continues to function. An alarm is issued after a certain period.

In this mode the work flow is:

- Download test requests from the Host to the Analyzer.
- Place the samples on the instrument.
- The Analyzer processes the samples and sends the results to the Host and optionally prints the results on the integrated printer. Re-transmission of the results in a batch is also optional.

3.3 Host Connection: Real-Time Test Selection Mode

In this mode the Analyzer reads the bar-codes for each sample (normal, STAT and control) and requests the Host to provide test selection data. The Host data is merged with the test selections entered at the keyboard. On completion of the measurement the results are uploaded to the Host in “real-time” (RES frame). Transmission can be repeated later in a batch. If the connection to the Host fails the Analyzer continues to function. An alarm is issued after a certain period.

In this mode the work flow is:

- Place samples on the instrument
- The Analyzer processes the samples and requests the Host in real-time for the test selection data. The results are transferred to the Host and the results are optionally output on the integrated printer. Re-transmission of the results in a batch is also optional.

The Analyzer operation mode is selected via the **SYSTEM PARAMETERS** screen. It is important to note that the Host protocol is independent of the operating modes. **All defined messages** exchanged between the Analyzer and Host are **functional at all times**.

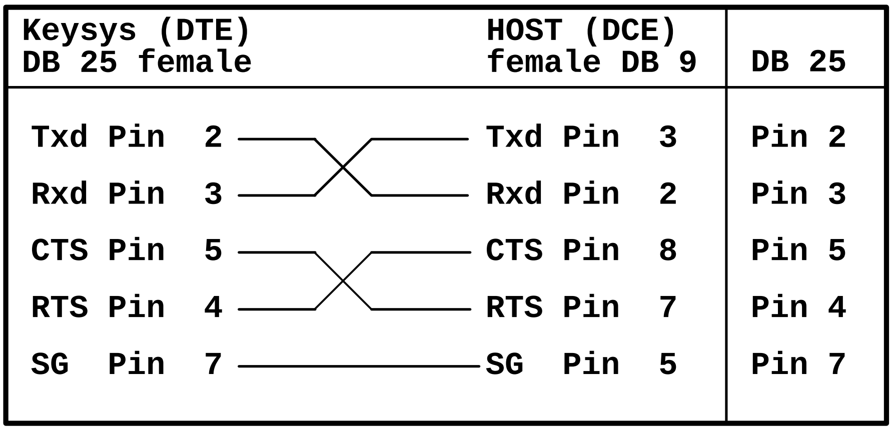
Result data can be transmitted to the Host in two data formats in both real-time or batch mode. The data formats is selected via the Keysys user interface are:

1. **Result Data Transfer**
   All results are transferred to the Host and stored locally on the Analyzer, i.e., for later re-transmission.
2. **Reaction Data Transfer**
   All calculated results and all the original absorbance data is transferred to the Host. Printout of data is **disabled** as the “Host” could potentially modify the calculated data after processing the raw absorbance data. Repeat transmission of the data to the Host is possible.

4. Interface Setup

4.1 Connection Cable

Figure 1 shows the pinout of the cable to connect the Analyzer to the Host.



Keysys (DTE) DB 25 female	HOST (DCE) female DB 9	DB 25
Txd Pin 2	Txd Pin 3	Pin 2
Rxd Pin 3	Rxd Pin 2	Pin 3
CTS Pin 5	CTS Pin 8	Pin 5
RTS Pin 4	RTS Pin 7	Pin 4
SG Pin 7	SG Pin 5	Pin 7

Figure 1: **Connection cable**

Attach the cable to the Host connector on the right-hand side of the Analyzer.

4.2 Pin Description

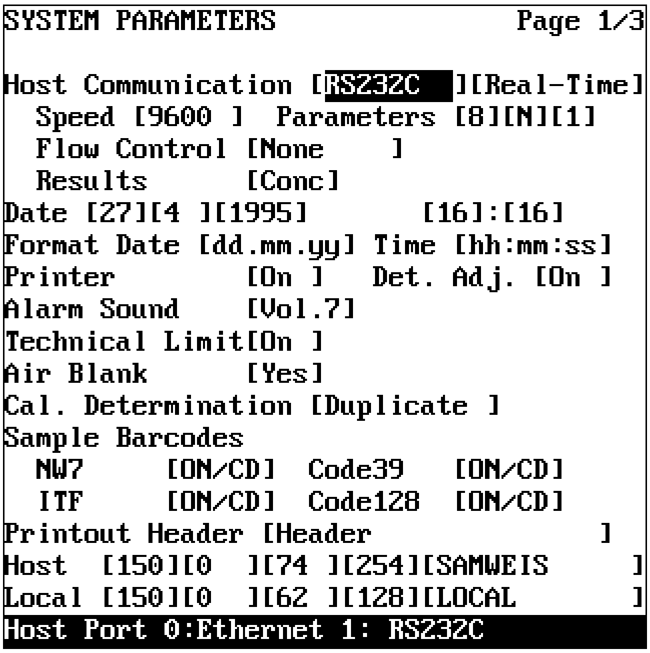
Table 1 describes the used pins of the RS232 serial interface.

Pin no	Signal	Explanation	Direction
2	Txd	Transmit Data	out
3	Rxd	Receive Data	in
4	RTS	Request to Send	out
5	CTS	Clear to Send	in
7	SG	Signal Ground	-

Table 1: **Pin description of the Keysys Analyzer 25-pin plug**

4.3 Setup of Communication Parameters

All Host interface settings are made on the **SYSTEM PARAMETER** screen.



```
SYSTEM PARAMETERS                  Page 1/3

Host Communication [RS232C   ][Real-Time]
  Speed [9600 ]  Parameters [8][N][1]
  Flow Control [None     ]
  Results      [Conc]
Date [27][4 ][1995]       [16]:[16]
Format Date [dd.mm.yy] Time [hh:mm:ss]
Printer        [On ]   Det. Adj. [On ]
Alarm Sound    [Vol.7]
Technical Limit[On ]
Air Blank      [Yes]
Cal. Determination [Duplicate ]
Sample Barcodes
  NW7      [ON/CD]  Code39   [ON/CD]
  ITF      [ON/CD]  Code128  [ON/CD]
Printout Header [Header            ]
Host  [150][0  ][74 ][254][SAMWEIS     ]
Local [150][0  ][62 ][128][LOCAL       ]
Host Port 0:Ethernet 1: RS232C
```

Figure 2: The SYSTEM PARAMETER screen

The following settings can be done:

Host Communication

Host Port	0: Ethernet 1: RS232C
Communication	0: Off 1: Batch 2: Real-Time

Speed

Baud rate	1:1200 2:2400 3:4800 4:9600 5:19200

Parameters

Data Bits	1:7 2:8
Parity check	0: None 1: Odd 2: Even
Stop Bits	1:1 2:2

Flow Control

	0: None 1: RTS/CTS 2: XON/XOFF 3: both

Results

Format	1: Concentration values 2: Raw data

Communication is initiated from **SYSTEM PARAMETER** screen by switching ‘Host Communication’ from [off] to [Real-Time] or [Batch].

5. Software Protocol

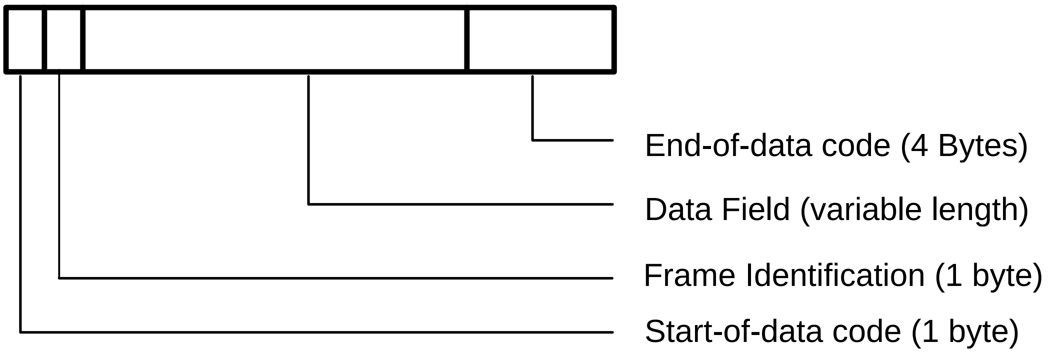
The RS232 interface allows point-to-point connection between the Keysys and a “Host” (this may be a Personal Computer). A communication protocol is necessary to establish a reliable connection over the physical RS232 link. This protocol is called the “frame transmission layer” and is described below.

The frame transmission layer guarantees reliable delivery of bi-directional data. The data messages between the two devices are called “commands”. The command set implemented on the Keysys is called the “command protocol”. It is identical for both the RS232 and Ethernet interfaces. (The Ethernet Interface uses RPCs over a ONC protocol, guaranteeing reliable data transfer. The “frame transmission layer” is, therefore, not strictly necessary).

This chapter describes the RS232 frame transmission protocol layer.

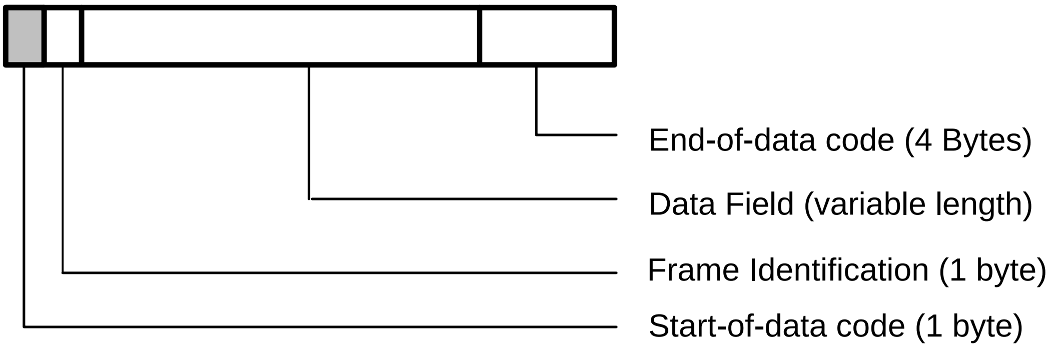
5.1 Frame Structure

All data between the Analyzer and the Host computer system is transferred in **frames**. A frame consists of at least 6 bytes. The maximum length is not fixed but is typically about 500 bytes. Each frame is identified by a **frame ID**. A frame contains the following elements:


End-of-data code (4 Bytes)
Data Field (variable length)
Frame Identification (1 byte)
Start-of-data code (1 byte)

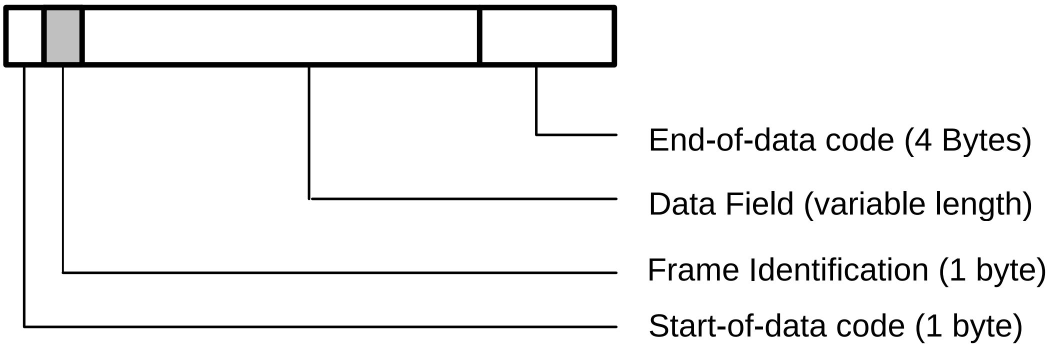
The **DATA FIELD** is empty for messages without data (control frames ***ACK***, ***NAK,*** see Table 3).

5.1.1 Start-of-data Code


End-of-data code (4 Bytes)
Data Field (variable length)
Frame Identification (1 byte)
Start-of-data code (1 byte)

Every message sent from the Keysys or the Host begins with the **START-OF-DATA CODE** (STX **=** 02H).

5.1.2 Frame Identification


End-of-data code (4 Bytes)
Data Field (variable length)
Frame Identification (1 byte)
Start-of-data code (1 byte)

5.1.2.1 Frame Character

The frame character holds the purpose or content of message.

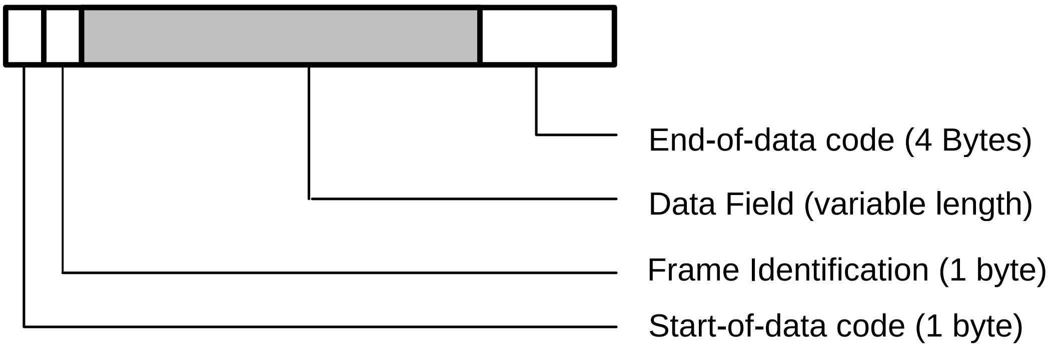
Mnemonic	Name	Char.	ASCII Code	Sender	Note
COM	Command	$	24H	Analyzer or Host	data as specified by the command protocol

Table 2: **Command Frames with a data field**

Mnemonic	Name	Char.	ASCII Code	Sender	Note
ACK	Acknowledge	>	3EH	Analyzer or Host	last valid frame received
NAK	Not acknowledge	?	3FH	Analyzer or Host	last invalid frame received - re-send request

Table 3: **Data Flow Control Frames without a data field**

5.1.3 Data Field


End-of-data code (4 Bytes)
Data Field (variable length)
Frame Identification (1 byte)
Start-of-data code (1 byte)

Data is transferred between the Analyzer and the Host computer system as **commands**. A command is the same for both RS232 and Ethernet communication and consists of a string of ASCII characters. Each string has a 3-digit **command identifier (CID)** followed by a colon and the remaining **command elements (CE)**. Each command element (or parameter) uses the ASCII characters 33 to 126 (decimal). Command elements are separated by delimiters. The only legal **delimiter** is the **space** character (“ ”, 0x20 hexadecimal).Non-printable characters (0x00 - 0x1F) and the space character itself (0x20) are coded by escape sequences. Each escape sequence is calculated as follows:

> Coded string = ESC followed by C.
> C = c + 0x40,
> where ESC is the character (0x1B), and ‘c’ is the ASCII code of the character to be coded.

A space, therefore, is represented by the two-byte escape sequence (0x1B60).

Non-printable characters and space are, however, valid in Sample ID’s and Reagent ID’s. Space is valid in following strings:

- Sample ID,
- Reagent name,
- Versions strings (application, system software, Host interface),
- Application name.

The characters 0x00 - 0x7F are valid for Sample ID’s, Reagent ID’s. The characters 0x20 - 0x7F are valid for versions, reagent and application names.

The maximum number of command elements is not fixed and depends on the command used. A typical command looks like this:

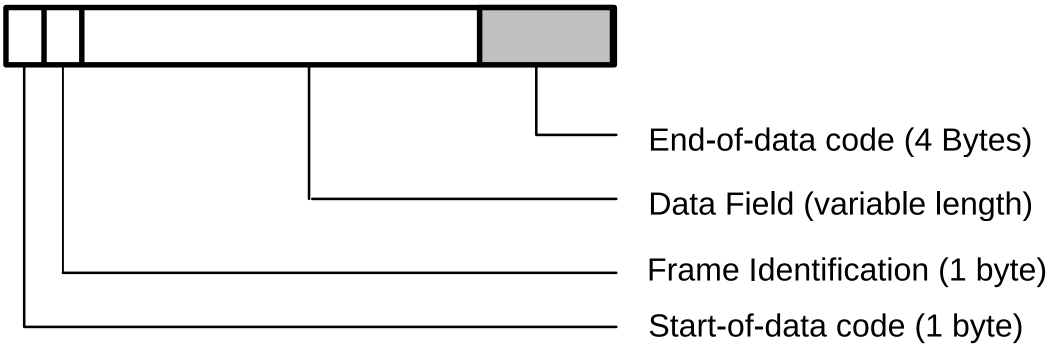
CID	**CE1**	**CE2**	. . .	**CEn**

The CID alone identifies the purpose of a command. An ASCII string, therefore, might look like this:

```
“REE: 0 123456789ABC 000 12 1 1994 0”
```

This is a Host request to transmit the results for sample “1234567890ABC”, for all methods, and all samples available dated 12 January 1994.

5.1.4 End-of-data Code


End-of-data code (4 Bytes)
Data Field (variable length)
Frame Identification (1 byte)
Start-of-data code (1 byte)

The **END-OF-DATA** code is placed at the end of each message sent from the Analyzer or Host. Its format is shown below:

Code	ASCII	Bytes
`[ETX][CKSH][CKSL][CR]`	`03H [high][low] 0DH`	4

Table 4: Format of the end-of-data code

Checksum Calculation:

[CKSH][CKSL] = Checksum high/low

The calculation is made as follows:

> Add the unsigned binary values of all bytes between the STX and ETX (i.e. all data bytes and the frame ID, but not the STX and ETX). Take the modulo 256 value and represent the result as a two digit hexadecimal string. Store the ASCII representation of the right character in CKSL and the left character in CKSH. Valid characters are “0” to “9” and “A” to “F”.
> It is essential for the reliable detection of the frame start and end that an STX or ETX byte is never allowed to occur in any other part of the frame, i.e. the Frame ID or within the data.

Example:

> If the added checksum is 25000 decimal, the hexadecimal equivalent is 61A8H. CKSL therefore becomes “8” and CKSH “A”
>
> End-of-data code [ETX]A8[CR]

5.2 Composition of the Data Field

5.2.1 Command Overview

The following tables give a summary of the Host interface commands. The command identifier is three-character mnemonic followed by a colon. The structure of the individual commands is detailed in the following chapters.

Data Control Commands

CID	Sender	Meaning
ERR	Host or Analyzer	used to signal an error condition

Test Selection Commands

CID	Sender	Meaning
TSE	Analyzer	test selection request
TSD	Host	test selection data

Result Commands

CID	Sender	Meaning
REE	Host	result request
RES	Analyzer	result data

Status Information Commands

CID	Sender	Meaning
STE	Host	status request
STS	Analyzer	status information transmission
SET	Host	set Analyzer status

Reagent Information Commands

CID	Sender	Meaning
RGE	Host	reagent status request
REG	Analyzer	reagent status transmission

Calibration Information Commands

CID	Sender	Meaning
CAE	Host	calibration factors request
CAL	Analyzer	calibration factors transmission
CAL	Host	calibration factors transmission

Application Data Commands

CID	Sender	Meaning
APR	Host	request to send application data
APD	Analyzer	application data transmission
APD	Host	application data transmission

Profile Information Commands

CID	Sender	Meaning
PDE	Host	request to send profile settings
PDA	Analyzer	profile data transmission
PDA	Host	profile data transmission

Calculated Tests Information Commands

CID	Sender	Meaning
CDE	Host	request to send settings for calculated data
CDA	Analyzer	calculated tests data transmission
CDA	Host	calculated tests data transmission

Inventory Information Commands

CID	Sender	Meaning
IVE	Host	inventory request
INV	Analyzer	inventory of applications

Note:

The Analyzer returns an ERR 14 (Analyzer busy) if the underlined commands are performed during a run.

5.2.2 Data Types

The following paragraphs specify each command in detail. The data representation tables use the following definitions:

char ASCII code (20H-7FH)

string [n] Variable length ASCII string with a maximum of n characters. An empty (NULL) string is represented by “??”.

int ASCII representation of an integer value (e.g. “1”, “-4567876654”).

double An ASCII string representation of an IEEE double precision floating point real. Factor values sent between the Host and the Analyzer are stored in this form. Unless otherwise stated the resolution is 13 (max. precision of a double). The format is:

“SD.DDDDDDDDDDDDePNN” or
“SD.DDDDDDDDDDDDeP1NN”, where,

S :	sign,	“-“, “+” or blank
D :	mantissa,	“0” to “9”, number of digits specified by the resolution. Default is 13.
P :	sign,	“+” or “-“
N :	exponent,	“0” to “9”.

The ‘C’ format used to generate the string is “%.12e”. Please refer to a C Programming Language Manual.

S ASCII sign of the number, either “-“, “+” or blank.

D Digit in the range of “0”-“9”

SID Sample ID Number: ASCII string of maximum 12 bytes length represented by string [12].
Note: Sample ID Numbers are not restricted to the digits “0” - “9” but may also contain printable characters like “A”, “%” or unprintable characters such as 0x00 - 0x1F. ID Numbers should be sorted alphabetically.
The following sample ID strings have a special meaning and are therefore **reserved**:

“997001” - “997999” :	**Control** Samples
“998001” - “998999” :	**Standard** Samples
“999001” - “999999”:	**STAT** Samples

CON Concentration data represented in double precision. The resolution is specified in chemistry parameters. The format is:
“D.DDDDe+DD” or
“D.DDDDe-DD” with a maximum resolution of 5.
The ‘C’ format used to generate the string is “%.(R-1)e” (where R is a Resolution determined by CHEMISTRY PARAMETERS; please refer to C Programming manual).
The string “????????” is used when no data is available.

ABS Absorbance data represented in double precision with a fixed resolution of 5. The format is
“SD.DDDDe+DD” or
“SD.DDDDe-DD”.
The ‘C’ format used to generate the string is“%.4e”. Please refer to a C Programming Language Manual.

Note: Keysys receives floating point values from the Host interface using the C format “%le”. Please refer to a C Programming Language Manual.

5.2.3 Command Details

5.2.3.1 Data Control Commands

5.2.3.1.1 ERR Command

This command is notifies the recipient of an error condition.

Format:

Item	Frame	Command	Function	Error Code
Char	$	ERR:	0	
Type	char	char	char	int
Length	1	3	1	1-2
Details				1

Details	Meaning
1	Error Codes: 0 unknown data frame received 11 no data for last REE query 13 REE already busy 14 abort last query (from Host only) 80 Received message is too long (element number is wrong) 81 CID error: 1) Received sentence is less than 6 bytes: CID (3 bytes) + ':' (1-byte) + Delimiter (1-byte) + Function_code (1-byte) 2) The character following CID should be a colon(:). 3) The character following the colon(:) should be the delimiter. 82 Detected wrong code: 1) Data code in frame should contain ESC or '0x20' to '0x7F'. 2) ESC code should be used only in a string type element. 3) End of element is the ESC code (there is no data following the ESC code). 4) Following the ESC code should be character '0x40' to '0x60'. 83 Data length is wrong in each element: 1) Maximum data length of element 1 is too long. 2) In case of a fixed length element, the data length is incorrect. 3) There is no data following the delimiter. 84 Abnormal data: 1) Data type is incorrect (there is an alpha-numeric code in the integer type). 2) Value in data is over the limit (maximum and minimum value depending on data type is wrong). 3) Character number in data is over the limit (character number depending on data type is wrong). 4) There is an invalid value in data. 85 Received ERR command except ERR(13). 86 Detected incorrect value in SET command. 87 Detected incorrect value in CAL command. 88 In TSD command, received non-requested sample ID to delete. 89 In TSD command, received test request which is not assigned. 90 Analyzer has detected another fatal error on command processing (Analyzer sends this information just as Analyzer detects it)

5.2.3.2 Test Selection Commands

5.2.3.2.1 TSE Command: Test selection request

The Analyzer uses this frame to request a sample's test selection information from the Host.

Format:

Item	Frame	Command	Function	Sample ID
Char	$	TSE:	0	
Type	char	char	char	SID
Length	1	3	1	1-12
Details				1

Details	Meaning
1	Sample ID for normal, STAT and control samples

5.2.3.2.2 TSD Command: Test selection data

The Host uses this frame to transfer test selection data to the Analyzer. The Sample ID and a set of 0 to 113 application codes from 001 to 945 and 961 to 998 are transferred. The received test selection merges with any test selection currently on the Analyzer. When the Analyzer is in batch mode and supplementary tests are required to those selected by the Host, then a manual selection can be made on the Analyzer before or after the batch download of the test requests.

Note: Selecting zero tests for a sample can be used to **cancel the test selection information** for that sample, including any manually selected tests.

There is no frame to select the same set of tests for multiple samples.
Format:

Item	Frame	Command	Function	Sample ID	No. of test	Test 1	...	Test n
Char	$	TSD:	0					
Type	char	char	char	SID	int	string[3]		string[3]
Length	1	3	1	1-12	1-3	1-3		1-3
Details				1	2	3		

Details	Meaning
1	Sample ID for normal, STAT and control samples
2	Number of tests n: 0-113 (75 + 20 calculated + 18 profiles max.) if 0, the existing test selection is canceled
3	An array [1 to n] of method ID numbers 1-945 and 961-998 for the selected tests.

5.2.3.3 Result Commands

5.2.3.3.1 REE Command: Result request

This frame is used by the Host to request the re-transmission of a specific sample result stored on the Analyzer. It is possible to request the transmission of all results for this sample or only selected methods. An equal number of RES frames will be transmitted if all tests are requested.

An REE request will result in an ERR frame with code 11 in the following circumstances:

1. The sample on the ASP has not been accepted (not read by BCR and not scheduled).
2. The sample on the ASP has not been accepted but has been read by the BCR (not scheduled).

A REE request will result in a RES frame containing “????????” string in the data field in the following circumstances:

1. The sample is accepted but testing has not yet started.
2. The sample is accepted and started to test already and some methods are being tested.
3. There is a sample which is accepted and some methods are finished. Methods which are finished give valid results, the others give “????????”.

Please note that all analytical results are stored on the Analyzer until they are deleted. It is therefore possible that the same sample ID is run on subsequent days. It is even possible in some cases to run the same sample several times during a day. The REE frame therefore contains a day designator and an index that allows searching for the n-th. sample run on a specific day. If the day and/or the index for the request is defined as “all” then an REE query may result in the transmission of many RES frames.

If a match is not found for a specific REE request, an ERR frame with error code 11 is transmitted. Only one REE request can be submitted to the Analyzer at any one time. If an RES request is still being processed then further REE requests will generate an error frame ERR with error code 12. The Keysys can store a large quantity of data. The transmission of data from the last request issued can be aborted by sending an ERR frame with function code 13 from the Host.

The REE frame is flexible. It can be used to **delete** data from the local data base of the Analyzer. In this mode the function code of the REE frame is set to **deletion** instead of **query**. This allows data handling on the Host without the danger of the local database overflowing. If the deletion function is used, the Host can send an REE frame requesting information about the data currently in the Analyzer database. The data can be deleted by sending the same REE command set to deletion.

Format:

Item	Frame	Command	Function	Sample ID	Sel. test	Day	Month	Year	Index
Char	$	REE:							
Type	char	char	char	SID	int	int	int	int	int
Length	1	3	1	1-12	1-3	1-2	1-2	1-4	1-2
Details			1	2	3	4	5	6	7

Details	Meaning
1	Function character: “0”: query “1”: query with original absorbance data “D”: delete
2	Sample ID for normal, STAT, standard and control samples
3	Method ID: “001-945” and “961” - “980”, 0 = all tests
4	1-31, 0 = all days
5	1-12, 0 = all months
6	1994-2093, 0 = all years
7	1-99, 0 = all data (1 = 1st result during this day, 2 = 2nd result, etc.)

5.2.3.3.2 RES Command: Result data

This frame is used to transmit the results for a test item of a single sample to the Host immediately the result is available. This frame can be also be sent in response to a request (REE frame) from the Host or when transferred at the Analyzer to the Host.

Note that this frame comes in two different formats for either routine samples, STAT samples and control results, or for absorbance data. The format is similar in both cases.

There are two formats for the transfer of routine data; the result data format and reaction data format. The reaction data format contains all raw absorbance data in addition to the calculated result value. The transfer mode is defined in the **SYSTEM PARAMETERS** screen.

Format:

Item	Frame	Command	Function	Format	Sample ID	Test	Data	Unit	Flags	Index	Day	Month	Year	Hours	Minutes	Seconds	ABS 1	ABS 22
Char	$	RES:																
Type	char	char	char	char	SID	int	CON / ABS	int	string[8]	int	int	int	int		char	char	ABS	ABS
Length	1	3	1	1	1-12	1-3	1-2	1-2	8	1-2	1-2	1-2	1-4		1	3		
Details			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	15

Details	**Meaning**
1	Function character: “A” = Normal measurement, Real time “a” = Normal measurement, Re-transmission “C” = Rerun, Real time “c” = Rerun, Re-transmission
2	Result format: “0” = Concentration data “1” = Original absorbances
3	Sample ID for samples, standards and controls. For samples this is a 3-12 character alphanumeric code. For standards, controls and STAT samples this is a 6-digit numerical code.
4	Routine Data & ABS Data: “001” - “945” Routine Data: “961” - “980”
5	Routine data: CON format ABS Data: ABS format
6	1-24 1 = mg/ml, 2 = ug/ml, 3= ng/ml, 4 = g/dl , 5 = mg/dl, 6 = ug/dl, 7 = ng/dl 8=g/l, 9 = mg/l, 10 = ug/l, 11 = mmol/l, 12 = umol/l, 13 = nmol/l, 14 = mval/l, 15 = ukat, 16 = nkat, 17=u/l, 18=u/g, 19=u/ml, 20 = mu/ml, 21 = uu/ml, 22=%, 23 = mg/g, 24=g/mol, 25 = AU, 26 = mAU, 27 = mExt, 28 = - / -
7	Pos. 0: Sample short “0” = No flag “S” = Sample short Pos. 1: Reagent short “0” = No flag “R” = Reagent short Pos. 2: Technical range“0” = No flag “<” = Technical range low “>” = Technical range high Pos. 3: Normal range “0” = No flag “L” = Normal range low “H” = Normal range high Pos. 4: Linearity “0” = No flag “L” = Linearity flag Pos. 5: Prone Check “0” = No flag “P” = Prozone Flag Pos. 6: Absorbance check “0” = No flag “B” = Absorbance flag at begin “E” = Absorbance flag at end “X” = Abs. flag at begin and end Pos. 7: “0” = No flag “E” = Memory error flag
8	1-99
9	1-31
10	1-12
11	1994-2093
12	0-24
13	0-59
14	0-59
15	W-A a set of 22 absorbance values

If original absorbance data is transmitted (in the “data format” field) then items 1 to 14 of the data frame are identical. The end of the data frame is different (Detail 15).

5.2.3.4 Instrument Status Commands

5.2.3.4.1 STE Command: Status request

The Host uses this frame to interrogate the Analyzer about its operating condition.

Format:

Item	Frame	Command	Function
Char	$	STE:	0
Type	char	char	char
Length	1	3	1
Details			

5.2.3.4.2 STS Command: Status information

The Analyzer uses this frame to inform the Host about its operating condition.

Format:

Item	Frame	Command	Function	Status	Sample No	Analyzer Version	Host Version
Char	$	STS:	0				
Type	char	char	char	string[8]	int	string[13]	string[13]
Length	1	3	1	8	1-2	1-13	1-131
Details				1	2	3	4

Details	Meaning
1	Pos. 0-4: Operating status “00000” = Idle “00001” = sampling “00011” = sampling end Pos. 5: Test selection mode “1” = Real Time “2” = batch Pos. 6: Original Absorbances “0” = No “1” = Yes Pos. 7: Cycle time “1” = 30 second cycle time “2” = 60 second cycle time
2	current sample being pipetted 1-30, 0 = not sampling
3	e.g., “V. 1.12.87A” (characters in the range 0x21 - 0x7E are valid)
4	e.g. “1.0 Rev. B”

5.2.3.4.3 SET Command: Set Analyzer status

This frame is used by the Host to set the Analyzer status. If the same status can be set on the Analyzer, this will overwrite the selected status. This allows the Host to exercise limited control over the Analyzer.

Format:

Item	Frame	Command	Function	Status
Char	$	SET:	0	
Type	char	char	char	string[8]
Length	1	3	1	8
Details				1

Details	Meaning
1	Status Byte with positions 0 to 7: Pos. 0-4: “0” Pos. 5: Set test selection mode “1” = Real Time “2” = batch Pos. 6: Set original absorbances mode “0” = No “1” = Yes Pos. 7: Set cycle time “1” = 30 second cycle time “2” = 60 second cycle time

5.2.3.5 Reagent Information Commands

5.2.3.5.1 RGE Command: Reagent status request

The Host uses this frame to interrogate the Analyzer about its reagent inventory status.

Format:

Item	Frame	Command	Function
Char	$	RGE:	0
Type	char	char	char
Length	1	4	1
Details			

5.2.3.5.2 REG Command: Reagent Status Information

This frame is used by the Analyzer to transfer current reagent inventory information.

Format:

Item	Frame	Command	Function	# Bottles	Bar-code 1	Vol. 1	Bar-code 47	Vol. 47
Char	$	REG:	0	47				
Type	char	char	char	int	string[21]	int	string[21]	int
Length	1	4	1	2	21	1-3	21	1-3
Details				1	2	3	2	3

Details	Meaning
1	Number of bottles is always 47
2	The bottle bar-code is a character string; “@” if bottle is absent
3	Number of tests for reagents, volume in 0.1 ml for wash solutions and dilutents. Range is 0-999

5.2.3.6 Calibration Information Commands

5.2.3.6.1 CAE Command: Calibration factors request

This frame is used by the Host to ask the Analyzer to send the calibration data currently used for a certain method.

Format:

Item	Frame	Command	Function	App. ID
Char	$	CAE:	0	
Type	char	char	char	string[3]
Length	1	4	1	3
Details				1

Details	Meaning
1	Application ID: “001”-”945”

5.2.3.6.2 CAL Command: Calibration factors transmission

This frame is used by the Analyzer to send the current calibration data. The Host can also send this frame and can **edit and overwrite** the data used by the Analyzer. Please note that a valid calibration does *not* have any flags!

This frame has a second use . It is sent by the Analyzer after every calibration calculation, valid or otherwise. This allows the Host to recognize that calibrations are performed, even if they fail. If a calibration fails the flags are set accordingly, but the factors F1 = F2 = F3 = F4 are set to zero (“0.0”).
Format:

Item	**Frame**	**Command**	**Function**	**App. ID**	**Cal Type**	**Flags**	
Char	$	CAL:	0				
Type	char	char	char	string[3]	int	string[8]	➔
Length	1	4	1	1-3	1	8	
Details				1	2	3	

Item		**Cal code 1**	**Target 1**	**ABS 1**	**Cal code 6**	**Target 6**	**ABS 6**	
Char								
Type	➔	string[3]	double	ABS	string[3]	double	ABS	➔
Length		1-3			1-3			
Details		4	5	6	4	5	6	

Item		**Factor 1**	**Factor 4**
Char			
Type	➔	double	double
Length			
Details		7	7

Details	**Meaning**
1	Application ID: “001”-”945”
2	Calibration Type: 1 = straight, 2 = linear, 3 = comp, 4 = hyperbole, 5 = logit 0, 6 = logit1, 7 = logit2, 8 = spline
3	Flags: Pos. 0: “0” = No flag “D” = Duplicate Limit Pos. 1: “0” = No flag “S” = Sensitivity Limit Pos. 2: “0” = No flag “F” = Fit Limit Pos. 3-7: “0”
4	Calibrator Code number: “001”-”999”; “000” means absent
5	Target value
6	Measured absorbance
7	Linear Calibration: F1 = K, F2 = S1 Abs, F3 = F4 = 0.0 Nonlinear Calibration: F1 = A, F2 = B, F3 = C, F4 = D Spline Calibration: F1 = F2 = F3 = F4 = 0.0 If a calibration fails (D, S or F flag set): F1 = F2 = F3 = F4 = 0.0

5.2.3.7 Application Data Commands

5.2.3.7.1 APR Command: Application data request

This frame is used by the Host to request the Analyzer to send the currently application data for the specified application.

Format:

Item	Frame	Command	Function	App. ID
Char	$	APR:	0	
Type	char	char	char	string[3]
Length	1	4	1	3
Details				1

Details	Meaning
1	Application ID: “001”-”945”

5.2.3.7.2 APD Command: Application data transmission

This frame is used by the Host or Analyzer to transfer application data and synchronize the data on both systems after an update. The user interface intentionally blocks the operator from saving the application data on floppy disk. The application data is transferred in coded form to prevent the use of the Host interface to circumvent this restriction.

The data is transmitted in blocks of 512 bytes. The same data format is used on the application floppy disk. The data is coded.

Format:

Item	Frame	Command	Function	App. Version	App. ID	App. data
Char	$	APD:	0			
Type	char	char	char	string[13]	string[3]	string[512]
Length	1	4	1	1-13	3	512
Details				1	2	3

Details	Meaning
1	e.g.: “1.00.000”
2	Application ID: “001” - “945”
3	each byte of the 256 byte encrypted application data is converted into a 2 byte hexadecimal ASCII string. The encryption algorithm uses the code of the “cipher.c” module.

5.2.3.8 Profile Data Commands

5.2.3.8.1 PDE Command: Request to send profile settings

This frame is used by the Host to request the Analyzer to send the specified profile settings.

Format:

Item	Frame	Command	Function	App. ID
Char	$	PDE:	0	
Type	char	char	char	string[3]
Length	1	4	1	3
Details				1

Details	Meaning
1	Application ID: “981”-”998”

5.2.3.8.2 PDA Command: Profile data transmission

This frame is used by the Host or Analyzer to transfer profile data. It synchronizes the data on both systems after an update.

Format:

Item	Frame	Command	Function	App. ID	App. name	# Methods	App. ID 1	App. ID n
Char	$	PDA:	0					
Type	char	char	char	string[3]	string[5]	int	string[3]	string[3]
Length	1	4	1	1-3	1-5	1-2	1-3	1-3
Details				1	2	3	4	4

Details	Meaning
1	Application ID: “981” - “998”
2	5-character Application name
3	Number of selected methods: 0-75
4	A set [1 to n] of method ID numbers 1-945 for the selected methods separated by a delimiter.

5.2.3.9 Calculated Tests Data Commands

5.2.3.9.1 CDE Command: Request settings for calculated data

This frame is used by the Host to request the Analyzer to send the settings for calculated tests for the specified "method".

Format:

Item	Frame	Command	Function	App. ID
Char	$	CDE:	0	
Type	char	char	char	string[3]
Length	1	4	1	3
Details				1

Details	Meaning
1	Application ID: "961"-"980"

5.2.3.9.2 CDA Command: Calculated tests data transmission

This frame is used by the Host or Analyzer to transfer settings for calculated tests. It allows synchronization of the data after the systems have been updated.
Format:

Item	Frame	Command	Function	App. ID	App. name	Range low	Range high
Char	$	CDA:	0				
Type	char	char	char	string[3]	string[5]	double	double ➔
Length	1	4	1	1-3	1-5		
Details				1	2	3	4

Item	Unit	Resolution	Form 1	Form n
Char				
Type	➔ int	int	string[4]	string[4]
Length	1-2	1	1-4	1-4
Details	5	6	7	7

Details	Meaning
1	Application ID: "961" - "980"
2	5-character Application name
3	Normal range low
4	Normal range high
5	Unit: 1-24
6	Resolution: 1-5
7	18 strings of max. 4 characters each: A three digit number is a method ID. A one digit character is an operator (set of "*", "/", "-", "+", "(", ")"). Any other number is a floating point number. All individual strings are separated with a delimiter.

5.2.3.10 Inventory Data Commands

5.2.3.10.1 IVE Command: Inventory request

This frame is used by the Host to request the Analyzer to send application codes and status information of available applications.
Format:

Item	Frame	Command	Function
Char	$	IVE:	0
Type	char	char	char
Length	1	4	1
Details			

5.2.3.10.2 INV Command: Inventory of applications

This frame is used by the Analyzer to send the application codes of the applications currently available on the Analyzer (i.e. with assigned channel and chemistry parameters set).
Format:

Item	Frame	Command	Function	# App.	App. ID 1	On Board flag 1	App. ID n	On Board flag n
Char	$	INV:	0					
Type	char	char	char	int	string[3]	int	string[3]	int
Length	1	4	1	1-2	1-3	1	1-3	1
Details				1	2	3	2	3

Details	Meaning
1	Number of Applications: 0-75
2	Application ID: “001” - “945”
3	On the reagent tray / Status flag: “0” = absent, “1” = all necessary reagents on the reagent tray

5.3 Data Transmission Control Procedure

5.3.1 Establishment of Data Link

After switching the **'Host Communication'** from [Off] to [Real-Time] or [Batch] on the **SYSTEM PARAMETERS** screen, the Analyzer sends the ***ACK*** frame to the Host to initiate communication. The Host must answer within 5 seconds but not earlier than 20 msec, usually with an ***ACK*** frame.

The Analyzer and the Host continue to transmit alternately:

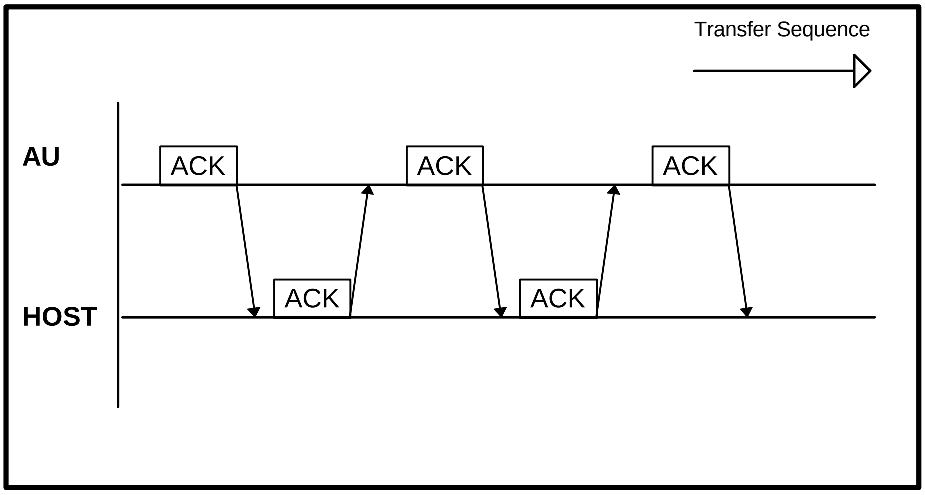


Figure 3: **Common communication timing**

5.3.2 Command Transmission

All frames except ***NAK*** (not acknowledge) indicate that the last message was good. This allows an extremely efficient interface, as shown below:

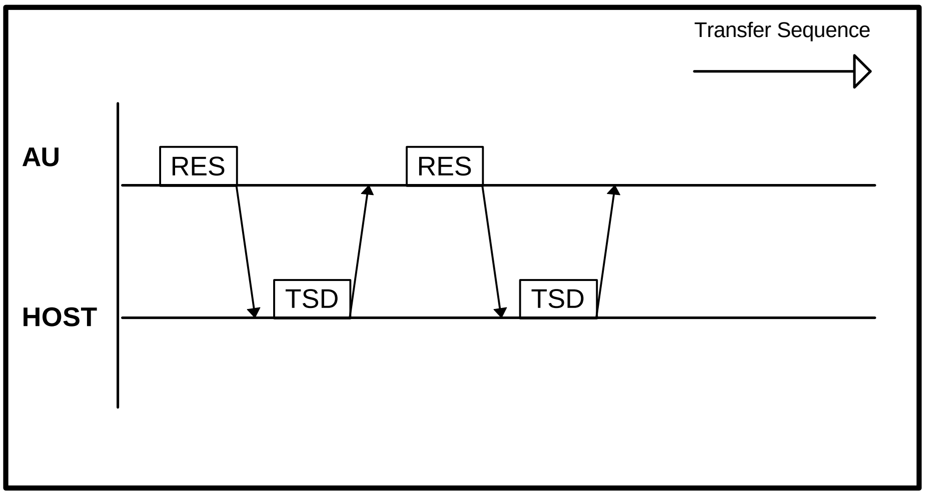


Figure 4: **Command Transmission**

5.3.3 Re-sending Request

Should an error condition occur, e.g., receipt of a corrupted frame, either side will respond with a ***NAK*** (not acknowledge) frame. The other side is then expected to repeat the transmission of the last frame. Transmission will continue after receiving an acknowledgment that the last message was good. The ***NAK*** frame can be considered a *re-send request* frame.

It is important to realize, that if a ***NAK*** frame is itself not understood and the response is another ***NAK*** frame, a deadlock situation occurs ("deadly embrace").Only ***NAK*** frames are exchanged over the Host connection. To avoid this situation either side will stop the communication if the response to a ***NAK*** frame is a ***NAK*** frame. The Analyzer will not restart the communication. However, the Host can restart communication after a 5 second pause.

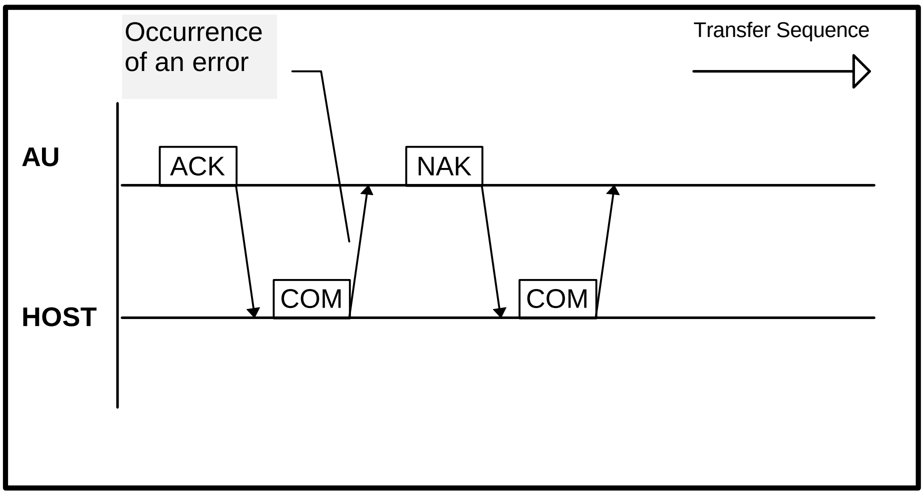


Figure 5: **Re-sending request with NAK frame from Analyzer to Host**

5.3.4 Termination of Communication

The communication is terminated by switching the **'Host Communication'** option on the **SYSTEM PARAMETERS** screen to [Off].

Should a time-out condition occur, i.e., no response within 5 seconds, both sides will stop the communication and assume the last message (if any) was not received. The Host can restart the communication after an additional waiting time of 5 seconds by sending another ***ACK*** frame.

6. Host Interface Functions

The following table shows the relationship between the Analyzer operating status and the data exchange through the Host interface:

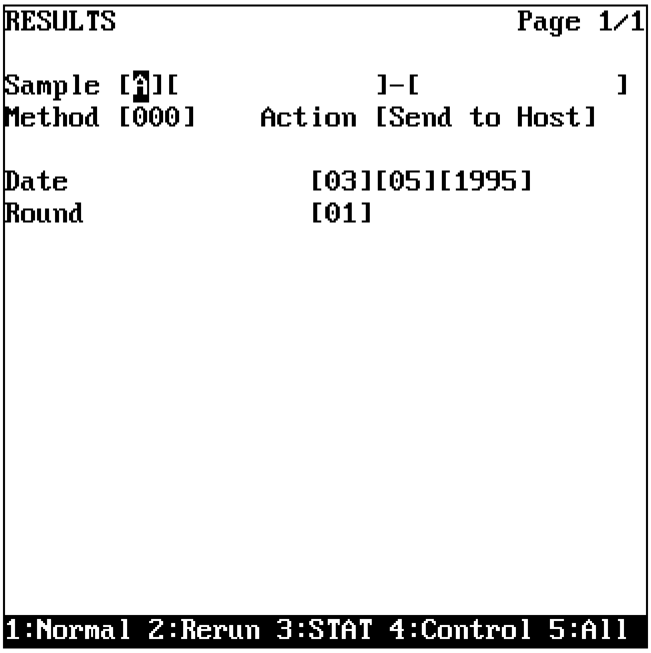
	Standalone	**Batch Mode**	**Real-time**
Test Selection			
Keyboard	YES	YES	YES
Batch download[1]	NO	YES	YES
Real-time query[2]	NO	NO	YES
Test results / Calibration Results			
Printer[3]	YES	YES	YES
Hard Disk storage	YES	YES	YES
Batch upload[1]	NO	YES	YES
Real-time upload[2]	NO	YES	YES

Notes:

1 Batch download of test selection data is possible at any time (initiated from Host). Batch upload of all available data is possible at any time from the **RESULTS** screen (see below).

2 Real-time communication performed while the instrument is analyzing.

3 Only if Printer = “On” on the **SYSTEM PARAMETERS** screen



```
RESULTS                              Page 1/1

Sample [A][            ]-[             ]
Method [000]    Action [Send to Host]

Date                 [03][05][1995]
Round                [01]













1:Normal 2:Rerun 3:STAT 4:Control 5:All
```

Figure 6: Results screen

The following settings are made in the **RESULTS** screen displayed above:

Sample

1: Normal ,2: Rerun, 3: STAT, 4: Control, 5: All
From Sample Number 001-999999999999

To Sample Number 001-999999999999

Method

Method Code 001-945 / 000=All

Action

0: Off 1: Print 2: Print Abs 3: Send to Host

Date

Day 1-31, 0 = Any

Month 1-12, 0 = Any

Year 1994-2093, 0 = Any

Round

Run Number 1-99, 0 = All

7. Example Traces

Example 1 : Data Control Command / ERR
Example 2 : Test Selection Commands / TSE, TSD
Example 3 : Result Commands / REE, RES
Example 4 : Instrument Status Commands / STE, STS, SET
Example 5 : Reagent Information Commands / RGE, REG
Example 6 : Calibration Data Commands / CAE, CAL
Example 7 : Application Data Commands / APR, APD
Example 8 : Profile Data Commands / PDE, PDA
Example 9 : Calculated Tests Data Commands / CDE, CDA
Example 10 : Inventory Information Commands / IVE, INV

Common explanations for the following Keysys communication traces:

Format

1st column	Sender of text (AU = Analyzer Unit)
2nd column	Sending time
3rd column	Trace data

Replacement of Control characters

Mnemonic	meaning	replaced ASCII code
[STX]	start of text	02H
[ETX]	end of text	03H
[CR]	carriage return	0DH
·	space	20H

Table 5: **Communication trace details**

The communication trace was acquired with the Interface Test Program ‘HOST_KEY.EXE’ (developed by Technical Product Management / Data Technique Department)

7.1 Data Control Command / ERR

Result request, result commands and abortion of result transmission by the Host with ERR 13 command

```
AU   13:49:05,25 [STX]>[ETX]3E[CR]
Host 13:49:05,30 [STX]$REE:·0·000001·0·0·0·0·0[ETX]5B[CR]

AU   13:49:06,07 [STX]>[ETX]3E[CR]
Host 13:49:08,26 [STX]>[ETX]3E[CR]

AU   13:49:08,82 [STX]$RES:·a·0·000001·930·3.86e+02·17·000H0000·1·2·5·1995
                 ·11·4·51[ETX]83[CR]

Host 13:49:09,08 [STX]>[ETX]3E[CR]
AU   13:49:09,91 [STX]$RES:·a·0·000001·943·0e+00·1·00000000·1·2·5·1995·11·
                 5·21[ETX]95[CR]
Host 13:49:10,07 [STX]>[ETX]3E[CR]
AU   13:49:10,73 [STX]$RES:·a·0·000001·917·5.21e+02·0·000H0000·1·3·5·1995·
                 10·47·56[ETX]83[CR]
Host 13:49:11,01 [STX]$ERR:·0·13[ETX]1B[CR]
```

7.2 Test Selection Commands / TSE, TSD

Requests for test selections and test selections from Host for the samples 000001, 000002 and 000003

```
AU   10:33:07,62 [STX]$TSE:·0·000001[ETX]DB[CR]
Host 10:33:10,26 [STX]$TSD:·0·000001·4·917·921·923·928[ETX]2C[CR]

AU   10:33:10,80 [STX]$TSE:·0·000002[ETX]DC[CR]
Host 10:33:11,01 [STX]$TSD:·0·000002·4·917·921·923·928[ETX]2D[CR]

AU   10:33:11,30 [STX]$TSE:·0·000003[ETX]DD[CR]
Host 10:33:11,41 [STX]$TSD:·0·000003·4·917·921·923·928[ETX]2E[CR]
```

7.3 Result Commands / REE, RES

Result Request for concentration result format and concentration result from instrument

```
AU   09:52:53,75 [STX]>[ETX]3E[CR]
Host 09:52:54,03 [STX]$REE:·0·998001·930·0·0·0·0[ETX]E1[CR]

AU   09:52:54,57 [STX]>[ETX]3E[CR]
Host 09:52:57,05 [STX]>[ETX]3E[CR]

AU   09:52:57,71 [STX]$RES:·a·0·998001·930·-7.8000e-03·17·00000000·1·3·5·1
                 995·9·43·41[ETX]1D[CR]
Host 09:52:58,08 [STX]>[ETX]3E[CR]
```

Result request for absorbance result format and absorbance result from instrument

```
AU   09:54:39,71 [STX]>[ETX]3E[CR]
Host 09:54:40,25 [STX]$REE:·1·998001·930·3·5·1995·1[ETX]93[CR]

AU   09:54:41,07 [STX]>[ETX]3E[CR]
Host 09:54:43,28 [STX]>[ETX]3E[CR]

AU   09:54:44,26 [STX]$RES:·a·1·998001·930·-7.8000e-03·17·00000000·1·3·5·1
                 995·9·43·41·1.0483e+00·1.0282e+00·1.0046e+00·1.0046e+00·1
                 .0008e+00·9.9800e-01·9.9440e-01·9.9180e-01·9.8820e-01·9.8
                 400e-01·9.7970e-01·9.7100e-01·9.6620e-01·9.6360e-01·9.593
                 0e-01·9.5710e-01·9.5120e-01·9.4770e-01·9.4360e-01·9.3880e
                 -01·9.3610e-01·9.3370e-01[ETX]17[CR]
Host 09:54:45,46 [STX]>[ETX]3E[CR]
```

Result request for a sample for which not all tests have been measured. Result for test 930 is available. Tests 917, 921 and 928 are not yet available.

```
AU   10:37:08,07 [STX]>[ETX]3E[CR]
Host 10:37:08,14 [STX]$REE:·0·000001·0·0·0·0·0[ETX]5B[CR]

AU   10:37:08,62 [STX]>[ETX]3E[CR]
Host 10:37:11,10 [STX]>[ETX]3E[CR]

AU   10:37:11,71 [STX]$RES:·a·0·000001·917·????????·5·00000000·1·3·5·1995·
                 10·38·50[ETX]AA[CR]

Host 10:37:12,19 [STX]>[ETX]3E[CR]
AU   10:37:12,80 [STX]$RES:·a·0·000001·921·????????·5·00000000·1·3·5·1995·
                 10·38·50[ETX]A5[CR]
Host 10:37:13,03 [STX]>[ETX]3E[CR]
AU   10:37:13,19 [STX]$RES:·a·0·000001·928·????????·5·00000000·1·3·5·1995·
                 10·38·50[ETX]AC[CR]
Host 10:37:13,41 [STX]>[ETX]3E[CR]

AU   10:37:13,85 [STX]$RES:·a·0·000001·930·3.86e+02·0·000H0000·1·2·5·1995·
                 11·4·51[ETX]4B[CR]
Host 10:37:14,12 [STX]>[ETX]3E[CR]
```

7.4 Instrument Status Commands / STE, STS, SET

Status request, Status information from instrument and the command to set the status from Host to instrument

```
AU   10:06:51,96 [STX]>[ETX]3E[CR]
Host 10:06:52,07 [STX]$STE:·0[ETX]9A[CR]

AU   10:06:52,85 [STX]$STS:·0·00000101·0·01.00.00·0.1[ETX]E6[CR]
Host 10:06:53,01 [STX]>[ETX]3E[CR]

AU   10:06:53,55 [STX]>[ETX]3E[CR]
Host 10:06:56,08 [STX]>[ETX]3E[CR]

...

AU   10:07:15,85 [STX]>[ETX]3E[CR]
Host 10:07:16,14 [STX]$SET:·0·00000212[ETX]3F[CR]
```

7.5 Reagent Information Commands / RGE, REG

Reagent Information request and reagent information from instrument

```
AU   10:19:42,73 [STX]>[ETX]3E[CR]
Host 10:19:43,01 [STX]$RGE:·0[ETX]8C[CR]

AU   10:19:43,78 [STX]$REG:·0·47·378654112010683086·330·378654802027383856
                 •330·@·0·@·0·819653167005956346·245·819653065006927046·24
                 5·@·0·@·0·286123456000666849·300·@·0·@·0·@·0·@·0·@·0·@·0·
                 •@·0·@·0·@·0·@·0·@·0·@·0·@·0·@·0·@·0·@·0·@·0·@·0·@·0·@·0·@
                 0·@·0·@·0·@·0·@·0·@·0·@·0·@·0·@·0·@·0·@·0·@·0·@·0·@·0·16
                 8653603007700076·700·168653602007300026·700·@·0·@·0[ETX]1
                 9[CR]
Host 10:19:45,71 [STX]>[ETX]3E[CR]
```

7.6 Calibration Data Commands / CAE, CAL

Request for Calibration data, Calibration data from instrument and Calibration data from Host to instrument

```
AU   11:03:05,71 [STX]>[ETX]3E[CR]
Host 11:03:06,26 [STX]$CAE:·0·917[ETX]38[CR]

AU   11:03:07,08 [STX]>[ETX]3E[CR]
Host 11:03:09,01 [STX]>[ETX]3E[CR]

AU   11:03:09,78 [STX]$CAL:·0·917·2·00000000·1·0.000000000000e+00·4.0000e-
                 04·3·5.370000000000e+02·6.3800e-02·0·0.000000000000e+00·0
                 .0000e+00·0·0.000000000000e+00·0.0000e+00·0·0.00000000000
                 0e+00·0.0000e+00·0·0.000000000000e+00·0.0000e+00·8.470e+0
                 3·4.0000e-04·0.000000000000e+00·0.000000000000e+00[ETX]15
                 [CR]
Host 11:03:10,37 [STX]>[ETX]3E[CR]

AU   11:03:10,66 [STX]>[ETX]3E[CR]
Host 11:03:13,12 [STX]>[ETX]3E[CR]

AU   11:03:14,28 [STX]>[ETX]3E[CR]
Host 11:03:16,14 [STX]>[ETX]3E[CR]

AU   11:03:16,76 [STX]>[ETX]3E[CR]
Host 11:03:19,12 [STX]>[ETX]3E[CR]

AU   11:03:20,00 [STX]>[ETX]3E[CR]
Host 11:03:20,55 [STX]>[ETX]3E[CR]

AU   11:03:21,08 [STX]>[ETX]3E[CR]
Host 11:03:21,19 [STX]$CAL:·0·917·2·00000000·1·0.0000e+00·4.0000e-04·3·5.3
                 700e+02·6.3800e-02·0·0.0000e+00·0.0000e+00·0·0.0000e+00·0
                 .0000e+00·0·0.0000e+00·0.0000e+00·0·0.0000e+00·0.0000e+00
                 8.4700e+03·4.0000e-04·0.0000e+00·0.0000e+00[ETX]45[CR]
```

7.7 Application Data Commands / APR, APD

Request for Application data, Application data from instrument and Application data from Host to instrument

```
AU   11:03:53,78 [STX]>[ETX]3E[CR]
Host 11:03:54,10 [STX]$APR:·0·901[ETX]4B[CR]

AU   11:03:54,37 [STX]>[ETX]3E[CR]
Host 11:03:56,03 [STX]>[ETX]3E[CR]

AU   11:03:56,96 [STX]$APD:·0·??·901·41534c4f200000000000000000013030000000
                 0000000000004069000000000000000000000000000000004082c000000000
                 003f847ae147ae147b000000000000000000000000000000003ff0000
                 000000000000000000000000f94a762a69dae5b2aeac699929434a8d
                 3717cf0bf07c1dfeeaac6ae4528d4192280ebfaa0fa166a8700ecad04
                 72df76e4fd56cb3a0f7f36936125cc49dd12957a7cfca023848448d73
                 11061cfa809808784d5c72e0973a4236ff3ff6b10ca5ae875ac903bbc
                 7b821010003008700870000000000000000000000000000000000000
                 000000000000000000000000000000000000000000000000000000000
                 0000000000000000000[ETX]77[CR]
Host 11:03:57,57 [STX]>[ETX]3E[CR]

AU   11:03:58,21 [STX]>[ETX]3E[CR]
Host 11:04:00,14 [STX]>[ETX]3E[CR]

AU   11:04:00,96 [STX]>[ETX]3E[CR]
Host 11:04:03,17 [STX]>[ETX]3E[CR]

AU   11:04:03,71 [STX]>[ETX]3E[CR]
Host 11:04:06,19 [STX]>[ETX]3E[CR]

AU   11:04:07,01 [STX]>[ETX]3E[CR]
Host 11:04:07,12 [STX]$APD:·0·??·901·41534c4f200000000000000000013030000000
                 0000000000004069000000000000000000000000000000004082c000000000
                 003f847ae147ae147b000000000000000000000000000000003ff0000
                 000000000000000000000000f94a762a69dae5b2aeac699929434a8d
                 3717cf0bf07c1dfeeaac6ae4528d4192280ebfaa0fa166a8700ecad04
                 72df76e4fd56cb3a0f7f36936125cc49dd12957a7cfca023848448d73
                 11061cfa809808784d5c72e0973a4236ff3ff6b10ca5ae875ac903bbc
                 7b821010003008700870000000000000000000000000000000000000
                 000000000000000000000000000000000000000000000000000000000
                 0000000000000000000[ETX]77[CR]
```

7.8 Profile Data Commands / PDE, PDA

Request for Profile data, Profile data from instrument and Profile data from Host to instrument

```
AU   11:03:30,21 [STX]>[ETX]3E[CR]
Host 11:03:30,26 [STX]$PDE:·0·981[ETX]49[CR]

AU   11:03:30,71 [STX]>[ETX]3E[CR]
Host 11:03:33,23 [STX]>[ETX]3E[CR]

AU   11:03:33,78 [STX]$PDA:·0·981·LIP ` `·2·943·930[ETX]0E[CR]
Host 11:03:34,28 [STX]>[ETX]3E[CR]
AU   11:03:35,10 [STX]>[ETX]3E[CR]
Host 11:03:37,07 [STX]>[ETX]3E[CR]

AU   11:03:37,62 [STX]>[ETX]3E[CR]
Host 11:03:40,10 [STX]>[ETX]3E[CR]

AU   11:03:40,32 [STX]>[ETX]3E[CR]
Host 11:03:42,01 [STX]>[ETX]3E[CR]

AU   11:03:42,51 [STX]>[ETX]3E[CR]
Host 11:03:43,12 [STX]$PDA:·0·981·LIP ` `·2·943·930[ETX]0E[CR]
```

The character represents the Escape character (code 1B hex, 27 dec)

7.9 Calculated Tests Data Commands / CDE, CDA

Request for Calculated tests data, Calculated tests data from instrument and Calculated test data from Host to instrument

```
AU   11:02:40,71 [STX]>[ETX]3E[CR]
Host 11:02:41,26 [STX]$CDE:·0·961[ETX]3A[CR]

AU   11:02:41,82 [STX]>[ETX]3E[CR]
Host 11:02:44,01 [STX]>[ETX]3E[CR]

AU   11:02:44,35 [STX]$CDA:·0·961·HBA1 `·3.900000095367e+00·5.699999809265
                 e+00·22·1·0.93·*·910·/·911·*·100.·+·0.92[ETX]D3[CR]
Host 11:02:44,62 [STX]>[ETX]3E[CR]

AU   11:02:45,12 [STX]>[ETX]3E[CR]
Host 11:02:47,03 [STX]>[ETX]3E[CR]

AU   11:02:47,85 [STX]>[ETX]3E[CR]
Host 11:02:50,05 [STX]>[ETX]3E[CR]

AU   11:02:50,60 [STX]>[ETX]3E[CR]
Host 11:02:53,07 [STX]>[ETX]3E[CR]

AU   11:02:53,62 [STX]>[ETX]3E[CR]
Host 11:02:54,17 [STX]$CDA:·0·961·HBA1 `·3.900000000000e+00·5.700000000000
                 e+00·22·1·0.93·*·910·/·911·*·100.·+·0.92[ETX]6B[CR]
```

The character   represents the Escape character (code 1B hex, 27 dec)

7.10 Inventory Information Commands / IVE, INV

Request for Inventory Information and Inventory information from instrument

```
AU   09:21:01,80 [STX]>[ETX]3E[CR]
Host 09:21:02,07 [STX]$IVE:·0[ETX]92[CR]

AU   09:21:03,23 [STX]$INV:·0·43·901·1·902·1·903·0·904·1·905·0·906·1·907·1
                 •908·0·909·0·910·1·911·1·912·1·913·0·914·0·915·0·916·1·91
                 7·1·918·1·919·1·920·1·921·1·922·1·923·1·924·1·925·1·926·1
                 •927·1·928·1·929·1·930·1·931·1·932·1·933·0·934·0·935·0·93
                 6·0·937·0·938·0·939·0·940·0·943·1·944·1·945·0[ETX]CC[CR]
Host 09:21:05,53 [STX]>[ETX]3E[CR]
```

8. Figures and Tables

Appendix A - ASCII Chart

Char	Hex	Dec	Char	Hex	Dec	Char	Hex	Dec	Char	Hex	Dec	Char	Hex	Dec	Char	Hex	Dec	Char	Hex	Dec	Char	Hex	Dec
NUL	00	0	DLE	10	16	Space	20	32	0	30	48	@	40	64	P	50	80	`	60	96	p	70	112
SOH	01	1	DC1	11	17	!	21	33	1	31	49	A	41	65	Q	51	81	a	61	97	q	71	113
STX	02	2	DC2	12	18	″	22	34	2	32	50	B	42	66	R	52	82	b	62	98	r	72	114
ETX	03	3	DC3	13	19	#	23	35	3	33	51	C	43	67	S	53	83	c	63	99	s	73	115
EOT	04	4	DC4	14	20	$	24	36	4	34	52	D	44	68	T	54	84	d	64	100	t	74	116
ENQ	05	5	NAK	15	21	%	25	37	5	35	53	E	45	69	U	55	85	e	65	101	u	75	117
ACK	06	6	SYN	16	22	&	26	38	6	36	54	F	46	70	V	56	86	f	66	102	v	76	118
BEL	07	7	ETB	17	23	′	27	39	7	37	55	G	47	71	W	57	87	g	67	103	w	77	119
BS	08	8	CAN	18	24	(	28	40	8	38	56	H	48	72	X	58	88	h	68	104	x	78	120
HT	09	9	EM	19	25	)	29	41	9	39	57	I	49	73	Y	59	89	i	69	105	y	79	121
LF	0A	10	SUB	1A	26	*	2A	42	:	3A	58	J	4A	74	Z	5A	90	j	6A	106	z	7A	122
VT	0B	11	ESC	1B	27	+	2B	43	;	3B	59	K	4B	75	[	5B	91	k	6B	107	{	7B	123
FF	0C	12	FS	1C	28	,	2C	44	<	3C	60	L	4C	76	\	5C	92	l	6C	108	|	7C	124
CR	0D	13	GS	1D	29	-	2D	45	=	3D	61	M	4D	77	]	5D	93	m	6D	109	}	7D	125
SO	0E	14	RS	1E	30	.	2E	46	>	3E	62	N	4E	78	^	5E	94	n	6E	110	~	7E	126
SI	0F	15	US	1F	31	/	2F	47	?	3F	63	O	4F	79	_	5F	95	o	6F	111	DEL	7F	127

Appendix B - Table of Communication Errors

Error Code	Description
528-50-000	HC Task Init Error
528-50-001 to 528-50-002	HC Task Delete Error
528-50-003	HC 232C Open Error
528-50-004	HC 232C Init Error
528-50-005 to 528-50-007	HC Task Init Error
528-50-008 to 528-50-012	HC Task Delete Error
528-50-013	HC Port Flush Error
528-50-014	HC Port Init Error
528-50-015 to 528-50-016	HC Port Set Err(RATE)
528-50-017	HC Port Set Err(NONE)
528-50-018	HC Port Set Err(RTS)
528-50-019	HC Port Set Err(XON)
528-50-021 to 528-50-023	HC Port Set Err(BOTH)
528-50-024	HC Port Set Err(DATA)
528-50-025	HC Port Set Err(STOP)
528-50-026	HC Port Set Err(PRTY)
528-50-028	HC 232C Read Error
528-50-029	HC 232C Buffer Full
528-50-031	HC Data Code Error
528-50-032	HC CID Code Error
528-50-033 to 528-50-34	HC Task Init Error
528-50-044	HC Message Full
528-50-045 to 528-50-050	HC RPC Server Error
528-50-052	HC Task Init Error
528-50-055	HC Msg Receive Error
528-50-056	HC Msg Send Error
528-50-057 to 528-50-058	HC Msg Receive Error
528-50-059 to 528-50-065	HC Task Init Error
528-50-066 to 528-50-068	HC Task Delete Error
528-50-069 to 528-50-072	HC Msg Receive Error

528-50-073	HC Illegal access
528-50-075	HC Data Write Error
528-50-076	HC Data Read Error
528-50-077	HC Data Write Error
528-50-078 to 528-50-082	HC Data Read Error
528-50-083	HC Data Write Error
528-50-086 to 528-50-087	HC Data Send Error
528-50-090	HC Illegal Status
528-50-091	HC Illegal Data(SET)
528-50-094	HC 232C Send Error
528-50-095	HC Data Read Err(RTDB)
528-50-103	HC Msg Receive Error
528-50-112	HC Illegal Host Name
528-50-113	HC Illegal Local Name
528-50-204	HC Data-Flow Timeout
528-50-201	HC Illegal Code (STX)
528-50-202	HC Illegal CR (ETX)
528-50-203	HC Illegal ETX (CR)
528-50-210	HC Illegal STX (ETX)
528-50-211	HC Illegal STX (CR)
528-50-222	HC Char Timeout (ETX)
528-50-223	HC Char Timeout (CR)
528-50-224	HC Sum-Code less
528-50-228	HC Sum-Code Error
528-50-232	HC Sum-Code over
528-50-300	HC Illegal Data recv.
528-50-301	HC Illegal ERR recv.
203-20-001	Result Store Error
203-20-002	Realtime CAL Send Err
203-20-003	Realtime RES Send Err