COBAS *MIRA*

HOST INTERFACE

DESCRIPTION

Information in this document is subject to change without notice.

1.   MAIN TABLE OF CONTENTS

This page is intentionally blank!

2. BASICS

2.1. TABLE OF CONTENTS

This page is intentionally blank!

2.2. HOW TO USE THIS MANUAL

This manual contains the following Host Interface message references:

COBAS MIRA Plus	*Section 3.*
COBAS INTEGRA 400 and 700	*Section 4.*
COBAS AMPLICOR	*Section 5.*

The sections *'2. Basics'* and *'6. Appendix'* refer to all COBAS instruments.

The communication protocol is similar for all COBAS instruments. Read this chapter first if you are not familiar with COBAS Host interfaces.

If you already know the COBAS Host interface structure and method of communication, go directly to each instrument's own reference section. Each instrument has a specific set-up and some specific block and line structures.

You may find the *'Example'* section and the COBAS Host simulator program of further help.

Syntax:

The following terms are used throughout the manual:

'COBAS instrument' or *'instrument'*
    Refers to COBAS Mira / Mira Plus, COBAS Integra / Integra 400 and/or COBAS Amplicor.

'Host computer', *'Host system'*, *'LIS/Host'* or *'Host'*
    Refers to Laboratory Information System or other computer system being connected to the COBAS instrument.

You will find the following symbols throughout this manual:

<LF>   All characters between the two angle brackets indicate an ASCII control character.

☐   This symbol indicates a blank (or space) character and is used as a separator between single parameters.

This symbol indicates also a blank (or space) character and is used as a place holder inside a single parameter and is part of the parameter.

The following block composition notation is used:

Single data line which is repeated from m up to n times:

```
                              ┌──────────┐ ┄┄┄┄┄┄┄┄┄ Line code
                              │    96    │
                         ────▶│Response  │────▶  next line code
                              │  Mess.   │
        minimum iteration ┄┄┄ │  m │  n  │ ┄┄┄  maximum iteration
                              └──────────┘
```

Several data lines which are repeated from x up to y times:

```
                      ┌──────────┐
                  ◀───│  x │  y  │┄┄┄┄
    minimum iteration └──────────┘    maximum iteration
```

Example 1:
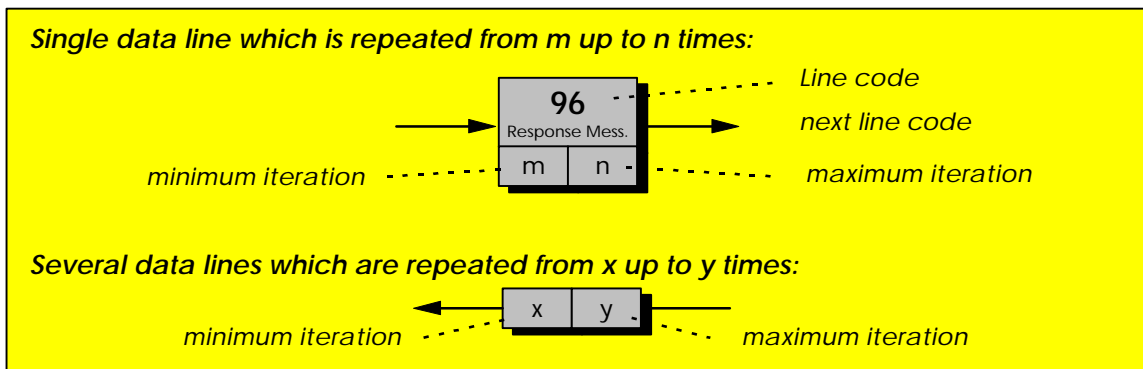
```
┌──────────────┐   ┌──────────────┐   ┌──────────────┐
│      53      │   │      55      │   │      00      │
│Def_Samp.Order│──▶│Def_Test Order│──▶│ Result Data  │──▶
│   1  │   1   │   │   1  │   1   │   │   1  │   1   │
└──────────────┘   └──────────────┘   └──────────────┘
```
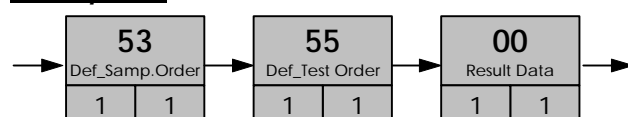
This function therefore contains three data lines:

53 Data string	min = max iteration -> line 53 exists 1 time
55 Data string	min = max iteration -> 1 time
00 Data string	min = max iteration -> 1 time

Example 2:

For some blocks, different structures are possible. Each structure represents a special function. The different functions depend either on the database set-up or on the kind of request / response.

```
┌──────────────┐   ┌──────────────┐   ┌──────────────┐   ┌──────────────┐
│      55      │   │      01      │   │      02      │   │      00      │
│Def_Test Order│──▶│ Result Time  │─┬▶│Contr. Identif│──▶│ Result Data  │─┬──▶
│   1  │   n   │   │   1  │   1   │ │ │   0  │   1   │   │   1  │   1   │ │
└──────────────┘   └──────────────┘ │ └──────────────┘   └──────────────┘ │
                                    │         ┌──────────┐                 │
                                    └─────────│  1 │  n  │─────────────────┘
                                              └──────────┘
```
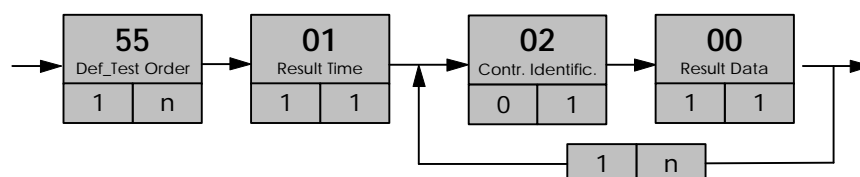
With *n=2* this function can look like this:

55 Data string	at least 1 time	
55 Data string	maximum 2 times	
01 Data string	min = max -> 1 time	
02 Data string	optional (0 or 1 time)	
00 Data string	min=max -> 1 time	at least 1 time
02 Data string	optional (0 or 1 time)	maximum 2 times
00 Data string	min=max -> 1 time	

Or it may look like this:

55 Data string	at least 1 time	
01 Data string	min = max -> 1 time	
00 Data string	min=max -> 1 time	at least 1 time
00 Data string	min=max -> 1 time	maximum 2 times

2.3.   COBAS HOST INTERFACE SIMULATOR PROGRAM

We recommend using the COBAS Interface Simulator program available with this manual to help program and test the Host interface.

COSIM is a Windows program which runs on any PC or Notebook (Windows 3.1, Windows 95, Windows NT). It allows to simulate a Host or a COBAS instrument and can also supervise communications between a Host and a COBAS:

- Operating as the Host, COSIM can send a request to the COBAS instrument and receive the instrument's response.
- In Instrument Mode, COSIM can be used by the host programmer to test his program without the need for a COBAS instrument.
- In Monitor Mode you can monitor and check the communication between the Host and the Instrument (two COM-ports are required).

The current version of COSIM allows to simulate either a COBAS Integra, Integra 400, Core, Core II or a Mira Plus.

User instructions are available from the integrated COSIM Online Help. Press F1, or choose Help from the main menu.

Installation:

- Start *'install.exe'*
- You will be asked where to install the program. Type in any directory (DOS 8.3 format) or accept the default path *'c:\cosim'*.

De-installation:

- All required files and information are kept within the directory specified during the installation procedure, e.g. *'c:\cosim'* → i.e. to un-install the COSIM program delete its directory and all files in it.

2.4.   INTRODUCTION

All COBAS instruments covered in this documentation are equipped with a RS232 serial interface for connection to a Host computer. This document explains the details of communication between Host computer and COBAS instrument.

2.4.1.   Communication Concept

The communication is based on a *Master / Slave* configuration, where the Host acts as Master. A Master / Slave communication is a data communication based on *Request / Response* dialogue. Following a request from the Master (Host) the Slave (COBAS) will transmit a specific response message:

- Assuming there are no transmission errors and the message can be handled correctly, the COBAS instrument will respond with a *Positive Acknowledgement Message*.

- If for any reason an error occurs the COBAS instrument will respond with a *Negative Acknowledgement Message*.

- If the Host does not receive a response on a request within a certain time, a *Time Out* should be activated.

The required error handling is described in section *'2.7.1 Communication Checks and Error Handling'*.

COBAS Integra 400

COBAS Integra 700

COBAS Mira Plus

COBAS Amplicor

Request

Response

HOST
Master

COBAS System
Slave
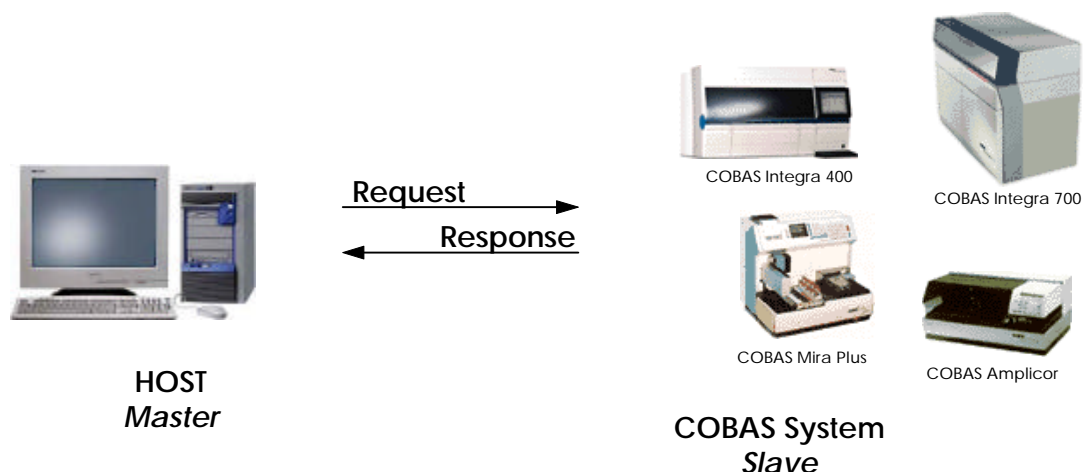
> *Note:*
> *The **COBAS MIRA** instrument can also act as a **Master**. However the Master / Slave mode guarantees safer and easier communication. Therefore we strongly recommend using the Master / Slave configuration.*

2.5.   MESSAGE STRUCTURE

The general message structure described below is valid for all COBAS instruments.

Information transferred inside the message is based on an 8- or 7 bit ASCII code (see also Appendix A). Characters with ASCII-Code 0..31, 127 and 255 are disallowed within data fields.

All messages are based on the same general configuration. The size of a message is variable. Each message contains several lines.

Generally a message contains:
- Block Header
- Block Data
- Block Check

Each message starts with a **<SOH>** (Start of Header) and ends with **<EOT>** (End of Transmission).

The block data begins with a **<STX>** (Start of Text) and concludes with an **<ETX>** (End of Text).

<SOH><LF>

Block Header*<LF>* (1 line)

<STX><LF>

Block Data*<LF>* (0..n lines)

<ETX><LF>

Block Check*<LF>* (0 or 2 lines)

<EOT><LF>

The Block Check can be enabled/disabled in the interface set-up menu. It is not available for COBAS

Sign	ASCII-Code (decimal)
<SOH>	001
<STX>	002
<ETX>	003
<EOT>	004
<LF>	010
<CR>	013

Each line closes with the line terminator **<LF>** or alternatively with **<CR><LF>**. The line terminator must be configured in the interface set-up menu accordingly.

The following example shows a complete block (Result response from a COBAS INTEGRA).

Start of Header	*<SOH> <LF>*
Block Header	**09□COBAS INTEGRA □04**<LF>
Start of Text	*<STX> <LF>*
	53□Order#211044711□20/10/93□SER<LF>
Block Data	**55□178**<LF>
	00□+3.234000E+01□mg/dl□004□023□014□000<LF>
End of Text	*<ETX> <LF>*
Block Check	**1**<LF>
	811<LF>
End of Transm.	*<EOT> <LF>*

Using this example an explanation of each item is given below.

2.5.1.   **Block Header**

The Block Header contains the following information:

> **Block Code:**
> **04** is a two digit numerical code to identify the type of information which follows. The block codes depend on the Instrument and are described in the *'Message Reference'* chapter of each instrument.

09□COBAS INTEGRA □04<LF>

> **Instrument Identifier:**
> **COBAS INTEGRA** is a sixteen digit alpha numerical. It is helpful in case more than one COBAS Instrument of the same type is connected to the Host.
> ⇒ The instrument's identifier can be defined in the instrument set-up menu.
> ⇒ The Host may use any 16 digit string. The instrument does not refer to it.

> **Instrument Code:**
> **09** is a two digit numerical code with a fixed value. It identifies the type of COBAS Instrument. An overview of the existing codes is documented in *'Appendix D'*.

The number of characters of the Block Header is fixed. If the instrument identifier is not used in its full length, remaining digits must be filled with blanks!

2.5.2. Block Data

The Block Data contains the main information. The size of this part is variable.

53□ Order#211044711□20/10/93□SER*<LF>*
55□178*<LF>*
00□+3.234000E+01□mg/dl□004□023□014□000*<LF>*

Depending on the Block Type the Block Data has 0, 1 or several data lines.

The length of a line depends on the type of the line code. Each line code corresponds to a fixed line length.

The content of each line is identified by a two digit numerical code called **Line Code**. These two digits are located at the beginning of every line and are separated by a space from the rest of the line.

A complete list of available line codes and their description is given in the *'Message Reference'* of each instrument.

Note:
Some of the line codes contain optional parameters. If any of the optional parameters are not used they can be left out. All other partially or unused parameters have to be filled out with blanks.

2.5.3. Block Check

1*<LF>*
811*<LF>*

The block check contains a *Sequence Counter (SC)* and a *Block Check Sum (BCS)*. These values are used for transmission control purposes.

The detailed information on how to use SC and BCS can be found in chapter *'2.6 Communication Control'*.

The Block Check is not available for COBAS MIRA.

2.6.   COMMUNICATION CONTROL

To maintain safe data communication, COBAS Instruments provide the following communication control mechanism:

– Message Control
  · Parity check
  · Sequence counter (SC)
  · Block check sum (BCS)
  · Check of block and line composition (only COBAS instrument)
– Flow Control
  · Hardware handshaking
  · Software handshaking
– Time-out control (only Host system)

2.6.1.   Message Control

2.6.1.1.   Parity Check

This is the most common method employed for detecting errors on asynchronous transmission. The number of active data bits in each character is inspected prior to transmission and the parity bit is computed. This is then added so that the total number of binary 1's in the complete character frame is either odd or even according to the pre-selection in the interface configuration. The receiving device will then perform the same inspection and compare the computed values.

Any mismatch will generate a parity error. COBAS instruments will transmit an error message (block 99, line 99) or in the case of COBAS MIRA an error message will be printed. The Host system's behaviour is described in section *'2.7.1 Communication Checks and Error Handling'*.

Important Note

COBAS INTEGRA 700:

On some Hewlett-Packard data stations the parity check does not work. If parity EVEN or ODD is selected no data will be transmitted. Therefore:

On COBAS INTEGRA 700 select Parity NONE!

2.6.1.2. Sequence Counter and Block Check Sum

This section is not valid for COBAS MIRA users!

Sequence Counter:

The Sequence Counter (SC) allows the Host to notify the COBAS instrument that there was a loss of data, i.e. transmission errors from the Host to the COBAS instrument or vice versa. Both SCs (Host and COBAS instrument) must be updated equally, otherwise the COBAS instrument will retransmit the previous sent block. The sequence counter is a one digit number, which alternates between ′**0**′ and ′**1**′. The mathematical presentation looks as follows:

$SC = (inc\ SC)\ mod\ 2$

The resulting use and behaviour of the sequence counter is shown in chapter *'2.7 Protocol Dialogue'*.

Block Check Sum:

The Block Check Sum (BCS) is a 3-digit number value, ASCII coded.

The BCS contains the mathematical sum of all values of the message starts with <SOH> ... until <SC> (<CR>) <LF> (Sequence Counter with line terminator) modulo 1000.

$BCS = sum\ (\ ord(<SOH>)\ ..\ ord(<LF>))\ mod\ 1000$

If any mismatch occurs the receiving station has to react as described in the following chapter *'2.7 Protocol Dialogue'*.

Enable / Disable SC and BCS:

The Block Check Sum BCS and The Sequence Counter SC are activated or de-activated in the Host Interface Configuration menu (BCS and SC always come together!).

BCS ON   COBAS Instrument sends Block Check Sum and Sequence Counter.

BCS OFF   COBAS Instrument sends neither Block Check Sum nor Sequence Counter.

> *Note:*
> *We strongly recommend that you* **switch the block check on**.
> ***Otherwise you cannot assure that*** *all data (e.g. results)*
> *are correctly transmitted to the Host system.*

ON/OFF refers only to transmission from the COBAS Instrument. When the Host sends a BCS/SC, it will be evaluated regardless of the pre-selection of the COBAS Instrument.

2.6.1.3. Check of Block and Line Composition

Following any mismatch in block and line composition, COBAS systems will transmit an error message (block 99, line 99) containing the appropriate error code. In the case of COBAS MIRA an error message will be printed. Whenever a block or line composition error occurs, this indicates that the Host software is not completely configured as described in this manual and it has to be modified accordingly.

Except relevant configuration settings have been changed, COBAS instruments never change the block and line composition. (You will find the configuration description in the instrument's *'Set-up Parameter'* section.)

2.6.2. Hardware and Software Handshaking

If the COBAS instrument or Host system is currently not able to receive data for any reason it must be able to stop the communication. This is covered by the flow control. The flow control is independent from the content of a message.

Two different possibilities are provided, either **Hardware Handshaking** or **Software Handshaking**. You will find the description of both in *'Appendix A'*.

> *Suggestion:*
> *Selecting Software **Handshaking** may simplify*
> *RS-232 cable wiring and Host SW handling.*

2.6.3. Receive Time Out

An exactly defined response time cannot be guaranteed. However, a returned response can be expected within 60 seconds. We recommend to **set the Host's time-out to 180 seconds**.

> *Note:*
> *Time-out control is only foreseen on the Host side.*
>
> *Incompletely received messages (messages without <EOT> termination) will* **not** *be acknowledged by the COBAS instruments → Host Time-out!*

2.7.   PROTOCOL DIALOGUE

2.7.1.   Communication Checks and Error Handling

The Error Handling can be explained by one sentence:

> *Note:*
> *Whenever the* **Host detects any transmission error, it has to re-transmits the previous message** *keeping the same Sequence Counter as before.*

Possible Transmission Errors:

- either a wrong block check sum, i.e. a corrupted message was received,
- a parity error was detected,
- a general error message (block 99) was received,
- or a time-out occurred.

The diagram on the following page shows a suggestion on how to implement the above described behaviour.

Suggestion on how to implement the Host's communication handling:

Synchronization:

Program Start

Synchronize the Host's Sequence Counter to the
COBAS Instrument Sequence Counter
(procedure explained in section *2.7.2*)

Normal Operation:

Update Sequence Counter,
Send request message

Receive response message
from COBAS Instrument

*Message
OK?*

No

Yes

Process the received data

Transmission Errors Detected:

Initialize an Error Counter

Increment Error Counter

*Error Counter
³ 8 ?*

Yes

No

Repeat the previously sent message,
it **must** contain the same Sequence
Counter as before → this will initialize
the retransmit of data

Receive response message
from COBAS Instrument

*Message
OK?*

No

Yes

2.7.2. Synchronisation between Host and COBAS Instrument

Note that the protocol only works properly when the Sequence Counters of both, Host system and COBAS instrument, have the same value. Therefore, whenever the Host's communication software is started up, the first step will be to synchronise the SCs. The following procedure is recommended:

Synchronisation

Host System:

Host initializes communication by sending a Sync-Block:

```
<SOH><LF>
09□COBAS INTEGRA    □00<LF>
<STX><LF>
<ETX><LF>
<EOT><LF>
```

A Sync-Block is effectively an Idle-Block *without block check*, i.e. both, Sequence Counter (SC) and Block Check Sum are omitted.

COBAS Instrument:

Instrument sends as a response an Idle Block *containing a block check*:

```
<SOH><LF>
09□INTEGRA 30-1051 □00<LF>
<STX><LF>
<ETX><LF>
1<LF>
299<LF>
<EOT><LF>
```

The COBAS Instrument will insert its current Sequence Counter, i.e. it will insert either 0 or 1.

The value of the received SC has to be used to initialize the Host's SC, i.e. Host SC = received SC

Normal Operation

The Host's SC has to be incremented as described in section *2.6.1.2* and to be added to the next transmitted request, e.g. to a result request as shown:

```
<SOH><LF>
09□COBAS INTEGRA    □09<LF>
<STX><LF>
10□01<LF>
<ETX><LF>
0<LF>
624<LF>
<EOT><LF>
```

Following the reception of a complete block the Instrument will increment its own SC, i.e. the Instrument's response will contain the same SC as the request:

```
<SOH><LF>
09□INTEGRA 30-1051 □00<LF>
<STX><LF>
<ETX><LF>
0<LF>
298<LF>
<EOT><LF>
```

Next Host request has to contain a SC of 1 → Response: 1, next Request: 0 → Response: 0, and so on ...

This page is intentionally blank!

3. MESSAGE REFERENCE COBAS MIRA PLUS

3.1. TABLE OF CONTENTS

This page is intentionally blank!

3.2. QUICK START

The flow chart given below shows how to quickly set up a communication between the COBAS MIRA Plus and a Host computer.

- **Prepare an interface cable** based on the recommendations given in the APPENDIX C
- **Connect the cable to the Host and the COBAS instrument**

Enable Host Interface transmission
<PROG> <6> <3>
Status : ON

and
Select operation mode

P63	OUTPUT MODE		MODIFY
P R I N T E R			
STATUS	:	ON	
NUMBER OF COPIES	:	1	
RESULT AUTO.MODE	:	SPL/CAL/CS	
QC AUTO.MODE	:	STATIST	
I N T E R F A C E			
STATUS	:	ON	
REMOTE CTRL.OUTP	:	ON	
RESULT AUTO.MODE	:	**SPL/CAL/CS**	
<PRINT> -> SEND	:	ON	
<CLOSE> -> SEND	:	OFF	
<1> <2> <3> <4>			

'Automatic Random Transmit Mode'

P63	OUTPUT MODE		MODIFY
P R I N T E R			
STATUS	:	ON	
NUMBER OF COPIES	:	1	
RESULT AUTO.MODE	:	SPL/CAL/CS	
QC AUTO.MODE	:	STATIST	
I N T E R F A C E			
STATUS	:	ON	
REMOTE CTRL.OUTP	:	**OFF**	
RESULT AUTO.MODE	:	**OFF**	
<PRINT> -> SEND	:	**OFF**	
<CLOSE> -> SEND	:	**OFF**	
<1> <2> <3> <4>			

'Master Slave Mode'

P63	OUTPUT MODE	MODIFY
INTERFACE CONFIGURATION		
BAUD RATE	:	4800
DATA BITS	:	8
STOP BITS	:	1
PARITY	:	NONE
HANDSHAKE	:	XON/XOFF
LINE TERMINATION	:	LF
POWER OFF/ON TO ACTIVATE CHANGED INTERFACE CONFIGURATION		
<1> <2> <3> <4>		

- **Set-up interface parameter** as required: <PROG> <6> <3> < >
- Do not use Baudrates above 4800 Bits/s. Loss of Data may occur if the analyzer is running!

If you are using **Bar Codes**, set-up the instrument as described in *'ID Type and Barcode Set-up'*

If you modified anything, **power OFF and ON the instrument**

Load your application, e.g. LIS or terminal program. As a first check **send an idle block** (block code 00)

3.3. GENERAL

3.3.1. Connecting the Instrument to the Host Computer

The interface port to connect to the LIS/Host is a SUB-D male type with 25 pins. The connector is named *HOST - RS-232* on the backside of the instrument. The pin assignment is referred in the section *'6.4. APPENDIX C'*. The following picture illustrates the connector part at the rear of the instrument.

Host Connector
RS 232
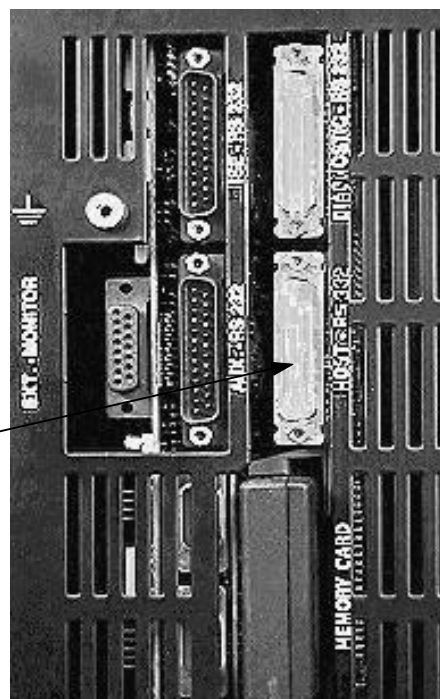
3.3.2. Set-up Parameter

Only the parameters that affect the Host interface or the work flow with a LIS/Host will be described.

> *Note:*
> *Some changes take effect after the next instrument **restart**! A message will appears on the screen for these parameters.*

How to change parameters:

- Press <PROG> and the corresponding numbers to access the required menu.
- Press <F1> MODIFY and the cursor will appear at the first parameter.
- Press <ENTER>, <↓> or <↑> to navigate through the parameters.
- Parameters can be modified by <F3> STEP PAR.
- Press <PRINT> to obtain a hard-copy of the OUTPUT MODE if desired.
- Press <ESCAPE> to exit the OUTPUT MODE.

3.3.2.1. Enable Host Interface Transmission

The Host Interface is enabled in the OUTPUT MODE menu. The OUTPUT MODE is used to define the printer's and the RS-232 serial interface's data output configuration. To enable the interface set INTERFACE STATUS to ON.

> *Note:*
> *The interface configuration can only be changed*
> *with an operator priority level 4.*

- Press <PROG> <6> <3> to access the OUTPUT MODE.

- Enable the interface by setting the INTERFACE STATUS to ON.

```
P63              OUTPUT MODE
                                      MODIFY

P R I N T E R
STATUS               :   ON
NUMBER OF COPIES     :   1
RESULT AUTO.MODE     :   SPL/CAL/CS
QC AUTO.MODE         :   STATIST

I N T E R F A C E
STATUS               :   ON
REMOTE CTRL.OUTP     :   ON
RESULT AUTO.MODE     :   SPL/CAL/CS
<PRINT> -> SEND      :   ON
<CLOSE> -> SEND      :   OFF


      <1>     <2>      <3>       <4>
```

3.3.2.2. Interface Parameters

- Press <PROG> <6> <3> <↓> to access the OUTPUT MODE page 2.

Parameter	Range			
Baudrate	<1>	75	<6>	1200
	<2>	110	<7>	2400
	<3>	150	<8>	4800
	<4>	300	<9>	9600
	<5>	600		
Data Bits	<1>	5	<3>	7
	<2>	6	<4>	8
Stop Bits	<1>	1	<3>	2
	<2>	1,5		
Parity	<1>	None	<3>	Odd
	<2>	Even		
Handshake	<1> HW only		<2>	SW
Line Termin.	<1>	LF	<2>	CR LF

```
P63              OUTPUT MODE
                                      MODIFY

INTERFACE CONFIGURATION


BAUD RATE            :   4800
DATA BITS            :   8
STOP BITS            :   1
PARITY               :   NONE
HANDSHAKE            :   XON/XOFF
LINE TERMINATION     :   LF


POWER OFF/ON TO ACTIVATE CHANGED
INTERFACE CONFIGURATION


      <1>     <2>      <3>       <4>
```

> *Note:*
> *We do not recommend to use 9600 Baud. Loss of data may occur if the*
> *analyzer is running.*

3.3.2.3. Communication Mode Set-up

COBAS MIRA Plus supports two basic communication modes, *Automatic Random Transmit Mode* and *Master Slave Mode*.

> *Note:*
> *Up to software version 9215.DA only the Automatic Random Transmit Mode was available.*

Automatic Random Transmit Mode:

With Automatic Random Transmit Mode, results will be transferred either as soon as they are generated or - if Print Send is enabled - when printing out results.

Set-up:

- Press <PROG> <6> <3> to access the OUTPUT MODE.
- To modify parameters please refer to the description above *'How to change parameters'*.

Parameters:

(All parameters are activated by setting INTERFACE STATUS to ON)

- REMOTE CTRL. OUTPUT:
    - ON: Additional commands and information is available: Rack loading, Executable test, Messages errors, Start, Stop, Abort and Initialization.
    - OFF: The additional commands are not available.

```
P63           OUTPUT MODE           MODIFY

PRINTER
STATUS                 :  ON
NUMBER OF COPIES       :  1
RESULT AUTO.MODE       :  SPL/CAL/CS
QC AUTO.MODE           :  STATIST

INTERFACE
STATUS                 :  ON
REMOTE CTRL.OUTP       :  ON
RESULT AUTO.MODE       :  SPL/CAL/CS
<PRINT> -> SEND        :  ON
<CLOSE> -> SEND        :  OFF


   <1>      <2>      <3>      <4>
```

- RESULT AUTO.MODE:
    SAMPLES: Automatic transmission of sample results.
    SPL/CAL/CS: Automatic transmission of calibration, control and sample results.
    RAW DATA: Automatic transmission of calibration, control and sample results with absorbency readings.
    (OFF: Not used with Automatic Random Transmit Mode.)

- <PRINT> -> SEND:
    ON: The <PRINT> key activates data for transmission. Transmission must be initiated from the PATIENT INTERIM REPORT.
    OFF: No transmission when <PRINT> is pressed.

- <CLOSE> -> SEND:
    ON: Quality control data is transmitted when the QUALITY CONTROL files are closed.
    OFF: Quality control data is not transmitted when the QUALITY CONTROL files are closed.

Master Slave Mode:

- Master Slave simply refers to which machine controls the communication, and which one is being controlled. This means:

  – The LIS/Host REQUESTS specific information (it is the Master)

  – The Instrument RESPONDS with only the specific requested information (it is the Slave)

- Master Slave has a big advantage compared with Random Transmit Mode:

  – Unlike Master Slave mode, Random Transmit Mode forces the LIS/Host to accept information even though it may not be able to do so. This may cause loss of data, data queues, etc..

  – With Master Slave Mode results are only transmitted on a specific request from the LIS/Host.

Set-up:

- Press <PROG> <6> <3> to access the OUTPUT MODE.

- To modify parameters please refer to the description above *'How to change parameters'*.

Parameters:

- Master Slave requires the *'Remote Control Output'*, *'Result Auto Mode'*, *'Print -> send'* and *'Close -> send'* set to OFF.

```
┌─────────────────────────────────┬──────────┐
│ P63          OUTPUT MODE        │          │
│                                 │ MODIFY   │
│                                 │          │
│ P R I N T E R                   ├──────────┤
│ STATUS              :  ON        │          │
│ NUMBER OF COPIES    :  1         │          │
│ RESULT AUTO.MODE    :  SPL/CAL/CS│          │
│ QC AUTO.MODE        :  STATIST   │          │
│                                 ├──────────┤
│ I N T E R F A C E               │          │
│ STATUS              :  ON        │          │
│ REMOTE CTRL.OUTP    :  OFF       │          │
│ RESULT AUTO.MODE    :  OFF       ├──────────┤
│ <PRINT> -> SEND     :  OFF       │          │
│ <CLOSE> -> SEND     :  OFF       │          │
│                                 │          │
│                                 ├──────────┤
│  <1>      <2>      <3>     <4>   │          │
└─────────────────────────────────┴──────────┘
```

3.3.2.4. Identification Type and Barcode Set-up

COBAS MIRA's patient identification can either be alphanumeric or only numeric. Unlike the alphanumeric identification the numeric identification is increased automatically when orders are entered manually. Please refer also to *'Line Code 50: Order Information'* description.

When using barcode any position on any rack can be selected to put a sample on. The LIS/Host does not need to specify a Sample Cup Number (position on the rack).

Set-up:

- Press <PROG> <6> <5> <↓> to access the SET UP PARAMETERS.
- To modify parameters please refer to section *'3.3.2 Set-up Parameter'*, *'How to change parameters'*.

Parameters:

- Barcode operation is enabled by setting SAMPLE-BARCODE and ID OF SAMPLE to ON.

- ID TYPE:
  - Numeric: Numbers only , automatic increase if orders are entered manually
  - Alpha: Letters supported, no automatic increase

- NB OF CHAR: up to 10 (length of the used (part of the) barcode)

- (SEARCH FOR: does not influence the Host Interface)

- PARTIAL SEL: Right/Left (only used if barcode ID is longer than defined by NB OF CHAR, either the right or the left part of the barcode is used as Patient ID)

```
┌─────────────────────────────────┬──────────┐
│ P65        SET UP PARAMETERS     │          │
│                                  │  MODIFY  │
│                                  ├──────────┤
│ PATIENT                          │          │
│ REPORT "ROUTINE"    :  ON        │          │
│ REPORT "STAT"       :  ON        │          │
│                                  ├──────────┤
│ IDENTIFICATION                   │          │
│ SAMPLE-BARCODE      :  ON        │          │
│ ID OF SAMPLE        :  ON        │          │
│    ID TYPE          :  NUMERIC   ├──────────┤
│    NB. OF CHAR.     :  10        │          │
│    SEARCH FOR       :  IDENTIFICATION       │
│    PARTIAL SEL.     :  RIGHT     │          │
│                                  │          │
├──────────────────────────────────┤          │
│  <1>    <2>    <3>    <4>         │          │
└──────────────────────────────────┴──────────┘
```

3.3.3.   Programmer's Guideline

3.3.3.1.   Time Out

An exactly defined response time cannot be guaranteed. However, a returned response can be expected within 20 - 30 seconds.

3.3.3.2.   Result Request

Automatic Random Transmit Mode:

With *Automatic Random Transmit Mode* there is no result request available. The result will be transferred either as soon as the result is generated or - if Print Send is enabled - when printing out results.

Master Slave Mode:

With *Master Slave Mode* Results are only transmitted upon a request from the LIS/Host. This is done by means of the Result Request Block 09.

The LIS/Host has to transmit this result request periodically.

It is not possible to request specific test results, which also means each result will be transmitted only once.

In the case of a particular test the calibration results are transmitted first, followed by the control results, then the test results.

The instrument carries out all test orders automatically in a random-access process within an instrument cycle time of 25 seconds.

As a result of this, we recommend using the following *procedure and timing for the result request loop*:
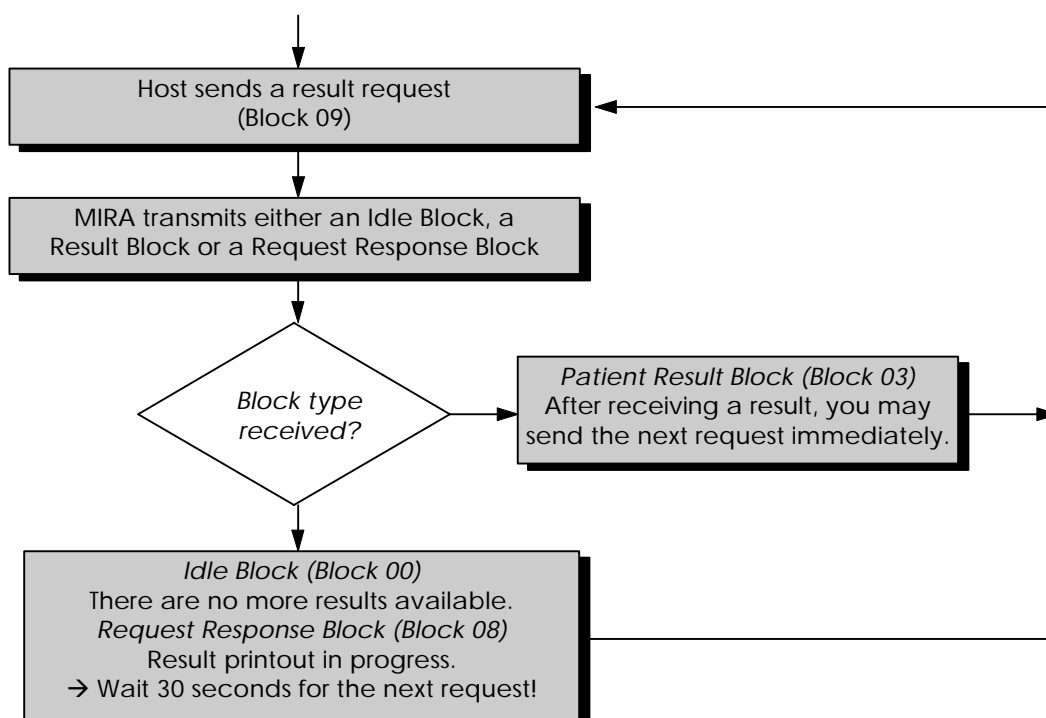
3.3.3.3. Order Entry and Order Extension

Orders are entered by means of Order Entry Block 10. They can be entered either into the Routine or the STAT worklist.

An existing order is extended by block 10 as well.

Tests and rack positions must be defined on COBAS MIRA Plus in advance. The instrument will not accept Order Entries containing un-defined tests and/or rack positions!

Additional <u>important</u> information can be found in *'Line Code 50: Order Information'* description.

3.3.3.4. Using Samples with Barcode Identification

When using barcode, after pushing the START button, the instrument starts reading all barcode labels containing the Patient Identifications.

The LIS/Host does not need to specify a Sample Cup Number (position on the rack). Therefore, the Sample Cup Number has to be set to '0000'. The Sample Cup Number is part of *'Line Code 50: Order Information'*.

3.4. BLOCK CODES

3.4.1. Overview Block Codes

Group	Type	Block Content	Instrument	Host
Result processing	00	Idle block	✗	✗
	01	Calibration and control results	✗	
	03	Patient results	✗	
	05	Quality control results	✗	
	07	Quality control result request		✗
	08	Request response	✗	
	09	Patient result request		✗
Order processing	10	Order entry		✗
	11	Order entry confirmation	✗	
	15	Order deletion		✗
	16	Order deletion confirmation	✗	
Status processing	70	Rack information	✗	
	71	Error message	✗	
	79	Rack information request		✗
Control processing	94	Instrument control		✗
	95	Instrument initialization		✗

> *Note:*
> *Do not use block codes 20, 21, 23, 29, 30, 31, 39, 80, 81 and 89. They are used for service and maintenance purposes.*

3.4.2. Protocol Dialogue

This chapter contains an overview about the dialogue between the LIS/Host and COBAS MIRA Plus.

The instrument supports two basic modes, the *Automatic Random Transmit Mode* and the *Master Slave Mode*.

> *Note:*
> *Up to software version 9215.DA only the Automatic Random Transmit Mode was available.*

3.4.2.1. Automatic Random Transmit Mode

Host System

Request →
← *Response*
← *Result*

COBAS MIRA
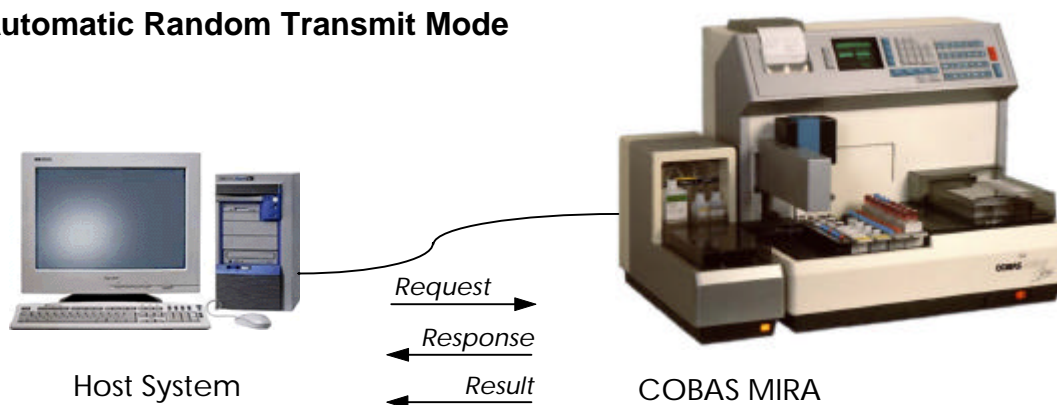
LIS/Host	Involved Block Codes	MIRA PLUS
Connection Test		
Idle Block	00	
	00	Idle Block
Patient result		
	03	Patient result
Calibration and Control results		
	01	Calibration and control results
Quality control results		
	05	Quality control result

Order Entry		
Order entry	10	
	11	Order entry confirmation

Order Deletion		
Order deletion	15	
	16	Order deletion confirm.

Rack Information		
	70	Rack information

Error Message		
	71	Error message

Control Processing		
Instrument control	94	
Instrument initialization	95	

3.4.2.2. Master Slave Mode (only with software version ^{3} 9215.DA)

Host System
Master

Request
Response

COBAS MIRA
Slave
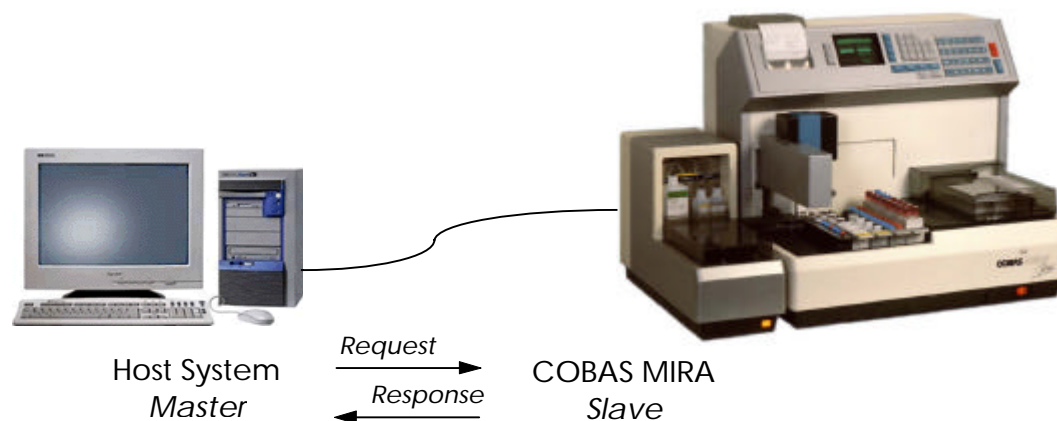
LIS/Host Request	*Involved Block Codes*	*MIRA PLUS Response*

Connection Test		
Idle Block	00	
	00	Idle Block

Patient result		
Patient result request	09	
	00	Idle Block (when no results exist)
	03, 08	Patient result or request response

Quality control results		
Quality control result req.	07	
	00	Idle Block (when no results exist)
	05, 08	Quality control result or Request response

Order Entry		
Order entry	10	
	11	Order entry confirmation

Order Deletion		
Order deletion	15	
	16	Order deletion confirm.

Rack Information		
Rack information request	79	
	70	Rack information

Note:
The data blocks **01** *(calibration and control results),*
03 *(Patient result) and* **71** *(Error message)* **are only**
available in Automatic Random Transmit Mode.

3.4.3. Composition of the Blocks

Please find block composition examples in section *'3.7 Examples'*.

The description of the used symbol notification can be found in section *'2.2 CONVENTIONS FOR THIS MANUAL'*.

3.4.3.1. Block 00: Idle Block

Block 00 is used for connection test, or as an answer to result request, if no results are available.

Block 00 has **no** lines.

3.4.3.2. Block 01: Calibration and Control Results

Only available with *Automatic Random Transmit Mode.* This block is transmitted to the LIS/Host whenever a calibration or control result is available.

ABS-Calibration and/or ABS-Control Result:
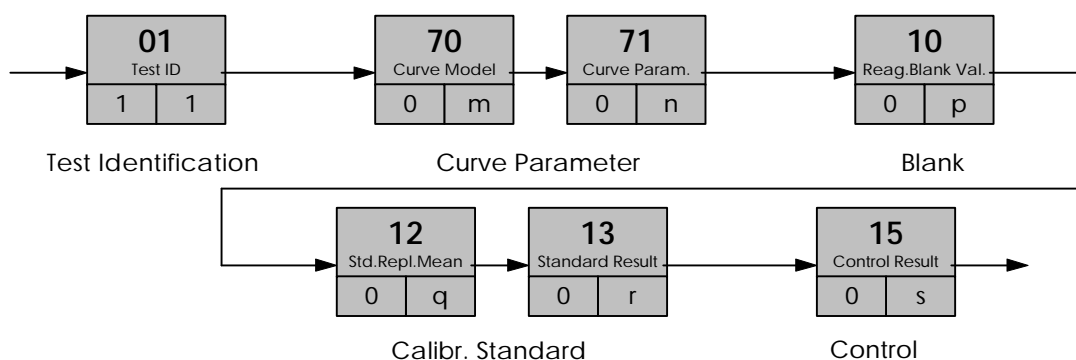
m = 0..1
n = 0..5
p = 0..1
q = 0..8
r = 0..24
s = 0..3

Depending on the test definitions (only Calibration or only Control Result, or Calibration and Control Result, etc.)

ISE-Calibration Result:
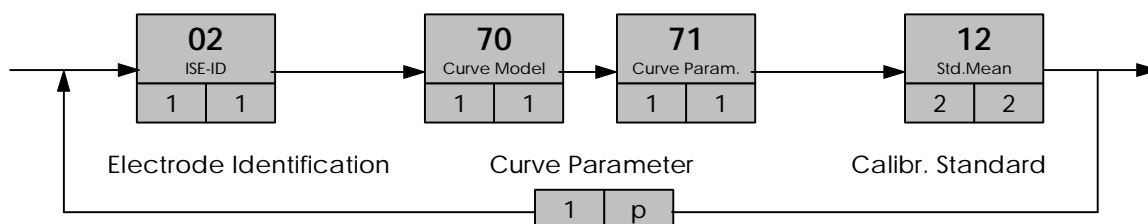
p = 1..3 (number of electrodes)

ISE-Control Result:
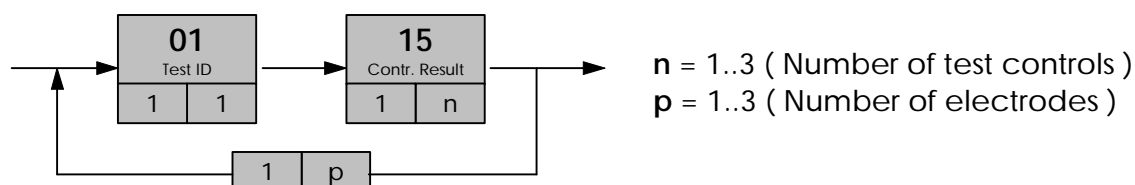
n = 1..3 (Number of test controls)
p = 1..3 (Number of electrodes)

> *Note:*
> *Raw data description is not included in this manual.*
> *Please call your local Roche service center if specific information is needed.*

3.4.3.3. Block 03: Patient Result

Automatic Random Transmit Mode: The block 03 is transmitted to the LIS/Host as soon as all results of a sample tube are available.

Master Slave Mode: These results are not transmitted until the Host sent a patient result request block 09.
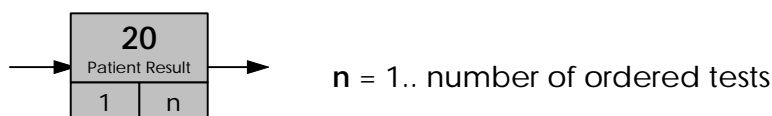
n = 1.. number of ordered tests

> *Note:*
> *Raw data description is not included in this manual.*
> *Please call your local Roche service center if specific information is needed.*

3.4.3.4. Block 05: Quality Control Result

Automatic Random Transmit Mode: The LIS/Host gets this message when closing the quality control (depending on the interface set-up).

Master Slave Mode: Block 05 is transmitted as a response to a quality control result request.

n = 1.. number of test controls
t = 1.. number of days in QC statistics
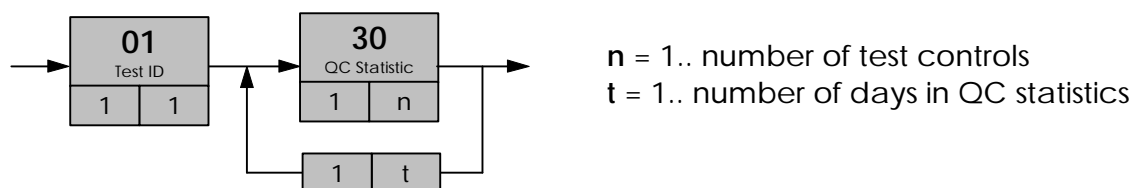
3.4.3.5. Block 07: Quality Control Result Request [1]

With *Master Slave Mode* quality control results are only transmitted after receiving a quality control result request block 07. The block 07 contains one line, the line:

```
       01
     Test ID
   ┌────┬────┐
   │ 1  │ 1  │
   └────┴────┘
```
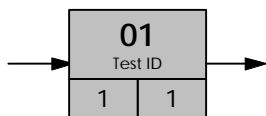
3.4.3.6. Block 08: Request Response [1]

A possible error of a Result Request (Block 07 or Block 09) will be answered by a Request Response block (Block 08). The block 08 contains one line, the line:

```
        96
   Response Msg.
   ┌────┬────┐
   │ 1  │ 1  │
   └────┴────┘
```
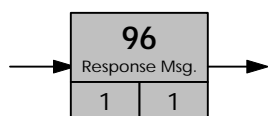
The line contains the corresponding error code.

3.4.3.7. Block 09: Patient Result Request[1]

With *Master Slave Mode* results are only transmitted after receiving a patient result request block 09.

Block 09 has *no* lines.

Please find the required result request procedure and timing in section *'3.3.3.2 Result Request'*.

3.4.3.8. Block 10: Order Entry

The Block 10 is used to enter orders into the worklist. The block 10 contains one line, the line:

```
        50
    Order Info.
   ┌────┬────┐
   │ 1  │ 1  │
   └────┴────┘
```

3.4.3.9. Block 11: Order Entry Confirmation

The block 11 is the positive or negative confirmation of an order entry. The block 11 contains one line, the line:

```
        50
    Order Info.
   ┌────┬────┐
   │ 1  │ 1  │
   └────┴────┘
```

---

[1] only with the software release 9215.DA

3.4.3.10.   Block 15: Order Deletion

The block 16 contains one line, the line:
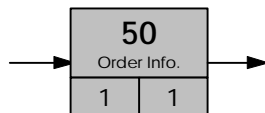
3.4.3.11.   Block 16: Order Deletion Confirmation

The block 16 is the positive or negative confirmation of an order deletion. The block 16 contains one line, the line:

3.4.3.12.   Block 70: Rack Information

The block 70 is used to receive information about the available racks on the COBAS MIRA Plus. This block is either transmitted in *Master Slave Mode* as a response to a block 79 (rack information request) or is transmitted in *Automatic Random Transmit Mode* with Remote CTRL. OUTP = ON.

n = 1 .. max. number of active tests

The rack numbers and all available tests (test numbers and test names) on these racks are sent back. The information is available as soon as the code of a newly inserted rack code has been read.

3.4.3.13.   Block 71: Error Message

All Error Messages which are printed or shown on the Screen can also be sent to the Host if *Automatic Random Transmit Mode* and '*Remote CTRL. OUTP = ON'* is selected. The block 71 has one line, the line:
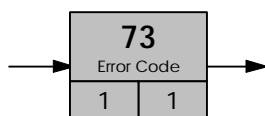
3.4.3.14. Block 79: Rack Information Request[1]

With *Master Slave Mode* the rack information request allows the LIS/Host, to get the actual slot configuration on the COBAS MIRA Plus.

Block 79 has *no* lines.

The rack numbers of the racks in the slots are sent back.

3.4.3.15. Block 94: Instrument Control

The block 94 is used to control the work flow of the COBAS MIRA Plus. This block allows the Host to start, stop or abort the analysis. The block contains one line, the line:

```
      ┌──────────────┐
  ──▶ │      65      │ ──▶
      │ Instr.Ctrl.Com. │
      ├──────┬───────┤
      │  1   │   1   │
      └──────┴───────┘
```
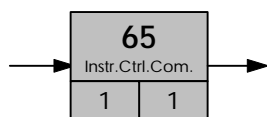
3.4.3.16. Block 95: Instrument Initialization

The block 95 is used to initialize the complete memory of the COBAS MIRA Plus. It contains one line, the line:

```
      ┌──────────────┐
  ──▶ │      66      │ ──▶
      │ Instr.Init.Com. │
      ├──────┬───────┤
      │  1   │   1   │
      └──────┴───────┘
```

---
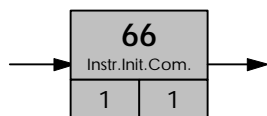
[1] only with the software release 9215.DA

This page is intentionally blank!

3.5. LINE CODES

3.5.1. Overview Line Codes

Line code	Content of line
01	Test Identification Number and Name
02	Ion Selective Electrode Identification
10	Reagent Blank Value
12	Standard, Replicate Mean
13	Standard Result
15	Control Result
20	Patient Result
30	Quality Control Statistics
50	Order Information
65	Instrument Control Command
66	Instrument Initialization Command
70	Curve Model
71	Curve Parameter
72	Rack Numbers
73	Error Code
96	Response Message

> *Note:*
> *The line codes 40, 41, 42 and 45 are used when working with raw data.*
> *If you need specific information please call your local ROCHE service center.*

3.5.2. Composition of Lines

3.5.2.1. Line Code 01: Test Identification Number and Name

Description	Format	Range
Start of this line	**01**	
	<space>	
Test number	US2	A1..Z9
	<space>	
Test name	US4	Capital A...Z, 0...9
End of this line	**<LF>**	

Example: **01□A1□GLUC**_{<LF>}

3.5.2.2. Line Code 02: Ion Selective Electrode Identification

Description	Format	Range
Start of this line	**02**	
	<space>	
Electrode position	I2	0..99
	<space>	
Electrode name	US4	Capital A...Z, 0...9
End of this line	**<LF>**	

Example: **02□01□NA··**_{<LF>}

3.5.2.3. Line Code 10: Reagent Blank Value

Description	Format	Range
Start of this line	**10**	
	<space>	
Blank result	F12	-9.99999E-38...+9.99999E+38
	<space>	
Result type	US1	"O" Original result "A" Additional result
	<space>	
Flag	US1	see section 'Tables, Flag Table'
	<space>	
Remark code	I2	see section 'Tables, Remark Table'
End of this line	**<LF>**	

Example: **10□-1.90000E-03□O□N□00**_{<LF>}

3.5.2.4. Line Code 12: Standard, Replicate Mean

Description	Format	Range
Start of this line	**12**	
	<space>	
Standard number	I2	0..99
	<space>	
Concentration	F12	-9.99999E-38...+9.99999E+38
	<space>	
Number of digits behind decimal point in the original instrument result	I2	
	<space>	
Unit code	I2	see section *'Tables, Unit Table'*
	<space>	
Deviation in %	F12	-9.99999E-38...+9.99999E+38
	<space>	
Result type	US1	"O" Original result "A" Additional result
	<space>	
Flag	US1	see section *'Tables, Flag Table'*
	<space>	
Remark code	I2	see section *'Tables, Remark Table'*
End of this line	**<LF>**	

Example: 12□01□+7.02000E+00□02□04□+0.00000E+00□O□N□00<LF>

3.5.2.5. Line Code 13: Standard Result

Description	Format	Range
Start of this line	**13**	
	<space>	
Standard number	I2	
	<space>	
Concentration	F12	-9.99999E-38...+9.99999E+38
	<space>	
Number of digits after decimal point in the original instrument result	I2	
	<space>	
Unit code	I2	see section *'Tables, Unit Table'*
	<space>	

....

See continuation on next page!

Description	Format	Range
Result type	US1	"O" Original result
		"A" Additional result
	<space>	
Flag	US1	see section *'Tables, Flag Table'*
	<space>	
Remark code	I2	see section *'Tables, Remark Table'*
End of this line	**<LF>**	

Example: **13□01□+7.02000E+00□02□04□O□N□00<LF>**

3.5.2.6. Line Code 15: Control Result

Description	Format	Range
Start of this line	**15**	
	<space>	
Control number	I2	00..03
	<space>	
Concentration	F12	-9.99999E-38...+9.99999E+38
	<space>	
Number of digits after decimal point in the original instrument result	I2	
	<space>	
Unit code	I2	see section *'Tables, Unit Table'*
	<space>	
Result type	US1	"O" Original result
		"A" Additional result
	<space>	
Flag	US1	see section *'Tables, Flag Table'*
	<space>	
Remark code	I2	see section *'Tables, Remark Table'*
End of this line	**<LF>**	

Example: **15□01□+4.01093E+00□02□04□O□N□00<LF>**

3.5.2.7. Line Code 20: Patient Result

Description	Format	Range
Start of this line	**20**	
	<space>	
Order type	US1	"R" Routine
		"M" Stat with routine rack
	<space>	
Test number	S2	A1..Z9
	<space>	
Test name	US4	Capital A...Z, 0...9
	<space>	
Test result index	I2	
	<space>	
Sample cup number	I4	
	<space>	
Patient identification	US10	Capital A...Z, 0...9
	<space>	
Concentration	F12	-9.99999E-38...+9.99999E+38
	<space>	
Number of digits after decimal point in the original instrument result	I2	
	<space>	
Unit code	I2	see section 'Tables, Unit Table'
	<space>	
Result type	US1	"O" Original result
		"A" Additional result
	<space>	
Flag	US1	see section 'Tables, Flag Table'
	<space>	
Remark code	I2	see section 'Tables, Remark Table'
End of this line	**<LF>**	

Example: **20□R□B1□CA··□01□0152□ANDERSON·L□+1.73208E+00□02□03□O□N□00**<LF>

3.5.2.8.　Line Code 30: Quality Control Statistics

Description	Format	Range
Start of this line	**30**	
	<space>	
Control number	I2	00..03
	<space>	
Day	I2	00..31
	<space>	
Month	I2	00..12
	<space>	
Year	I2	00..99
	<space>	
Mean value	F12	-9.99999E-38...+9.99999E+38
	<space>	
Minimum value	F12	-9.99999E-38...+9.99999E+38
	<space>	
Maximum value	F12	-9.99999E-38...+9.99999E+38
	<space>	
Number of control values	I2	
End of this line	**<LF>**	

Example: **30□02□12□03□94□+2.10322E+01□+1.33454E+01□+3.44332E+01□12**<LF>

3.5.2.9. Line Code 50: Order Information

Description	Format	Range
Start of this line	**50**	
	<space>	
Sample cup number	I4	
	<space>	
Patient identification	US10	**Details: see the following explanation!**
	<space>	
Order type	US1	"R" Routine "M" Stat with routine rack
	<space>	
Test number 1	S2	A1..Z9
Optional:	<space>	
Test number 2	S2	A1..Z9
	<space>	
Test number 3	S2	A1..Z9
	<space>	
Test number 4	S2	A1..Z9
	<space>	
Test number 5	S2	A1..Z9
	<space>	
Test number 6	S2	A1..Z9
	<space>	
Test number 7	S2	A1..Z9
	<space>	
Test number 8	S2	A1..Z9
End of this line	**<LF>**	

Example:

- Order Entry (Block 10):
  50□0018□WYSS·HELEN□R□A2□A3□Z1<LF>

- Order Entry Confirmation (Block 11) → Order entry <u>accepted</u>:
  50□0018□WYSS·HELEN□R□ □ □ <LF>

- Order Entry Confirmation (Block 11) → Order entry <u>rejected</u>:
  50□0018□WYSS·HELEN□R□A2□A3□Z1<LF>

- Order Entry Confirmation (Block 11) → Order entry <u>partially accepted</u>:
  50□0018□WYSS·HELEN□R□ □A3□ <LF>

Order Entry Accepted / Rejected:

- If an order was accepted, the Order Entry Confirmation will contain <space>s instead of the transmitted test number.

- If it was not accepted, the rejected test number will be still included in the response.

Patient Identification:

- General:
  - The <u>transmitted</u> Sample ID must contain 10 characters! Even if NB OF CHAR is set to less than 10 *(P65 Set Up Parameters, Page 2)*, the information must be expanded to 10 characters with trailing <space>'s.
  - The actual identification can not be longer than defined by NB OF CHAR.
  - The identification must be left aligned.

- If the sample's ID TYPE is set to **ALPHA** *(P65 Set Up Parameters, Page 2)*:
  - The sample identification may not contain characters other than capital letters (A..Z) and numbers (0..9). Special characters are not supported.
  - If the identification is shorter than NB OF CHAR it must be expanded with trailing <space>'s.
  - Valid examples (NB OF CHAR equals 6): 'ID01   ', 'ABCDEF   ' or 'A   '.
    Not allowed (NB OF CHAR equals 6): 'ABCDEFGH   ' or '   ABCDEF'.

- If the sample's ID TYPE is set to **NUMERIC**:
  - The sample identification may only contain numbers (0..9).
  - If the number used is shorter than NB OF CHAR it must be expanded with leading zeros.
  - Valid examples (NB OF CHAR equals 6): '001234   ' or '000001   '.
    Not allowed (NB OF CHAR equals 6): '12345678   ', '   123456' or '   1234   '.

> *Note:*
> *A maximum of 8 test entries per order information line are possible. If more than 8 tests have to be entered, the order must be divided into several order entries (complete blocks).*
>
> ******
>
> *Test and sample cup numbers (rack positions) used in order entries must already be defined on the instrument. Otherwise some tests or even the complete order might be refused.*

3.5.2.10. Line Code 65: Instrument Control Command

Description	Format	Range
Start of this line	**65**	
Command	<space> I2	01 = start analysis 02 = stop analysis 03 = abort analysis
End of this line	**<LF>**	

Example: 65□02<LF>

3.5.2.11. Line Code 66: Instrument Initialization Command

Description	Format	Range
Start of this line	**66**	
Command	<space> I2	01 = initialize memory
End of this line	**<LF>**	

Example: 66□01<LF>

3.5.2.12. Line Code 70: Curve Model

Description	Format	Range
Start of this line	**70**	
Evaluation model	<space> I2	details, see table below
End of this line	**<LF>**	

Example: 70□01<LF>

Evaluation Model	Significance
01	linear (photometry)
02	linear (ion selective electrode)
11	4 parameter log/logit
12	5 parameter log/logit
13	5 parameter exponential
15	liner interpolation
16	linear regression
17	factor

3.5.2.13. Line Code 71: Curve Parameters

Description	Format	Range
Start of this line	**71**	
Evaluation parameter	\<space\> I2	details, see table below
Parameter value; for ISE: slope	\<space\> F12	-9.99999E-38...+9.99999E+38
End of this line	**\<LF\>**	

Example: 71☐02☐+5.14307E+00\<LF\>

Evaluation Parameter	Significance
01	factor
02	correction factor
10 - 14	parameter according to model; see below

Parameter Model	10	11	12	13	14
11	RO	Kc	a	b	
12	RO	Kc	a	b	c
13	RO	KI	a	b	c
15					
16	RO	a			
17	F				

3.5.2.14. Line Code 72: Rack Numbers

Start of this line	**72**
Rack number, position 1	\<space\> I4
Rack number, position 2	\<space\> I4
Rack number, position 3	\<space\> I4
Rack number, position 4	\<space\> I4
End of this line	**\<LF\>**

Example: 72☐0226☐0006☐0511☐0512\<LF\>

3.5.2.15. Line Code 73: Error Code

Description	Format	Range
Start of this line	**73**	
	<space>	
Error code	I4	Please refer to section '*Tables*'
End of this line	**<LF>**	

Example: 73□0087_{LF}

3.5.2.16. Line Code 96: Response Message[1]

Description	Format	Range
Start of this line	**96**	
	<space>	
Request Error Code	I2	details, see table below
End of this line	**<LF>**	

Example: 96□24_{LF}

Error Code	Significance
24	Test not available (Response to block 07)
33	Patient result not available at the time → result is currently being printed (Response to block 09)

3.5.3. Data Formats

Format-Code	Format-Description
USx	x-character ASCII-String, left adjusted Only uppercase characters A..Z and numbers 0..9 Length x is mandatory, a shorter string has to be exceeded with <space>s to the length x. Example: US10 "EVANS "
Ix	ASCII-formatted x-digit integer value, right adjusted, Length x is mandatory, shorter integer-value has to be exceeded with < 0 on the left to the length x. Example: I5 "00069"
SRx.y	ASCII-formatted real value with sign, right adjusted x = Length of the whole data record y = Number of digits after the decimal point (optional) Length x is mandatory, shorter real-value has to be exceeded with 0 on the left to the length x. The decimal point is also mandatory. Example: SR7.4: "-7.7123"
Fx	ASCII formatted floating point number, right adjusted x = Length of the whole data record Length x is mandatory, shorter real-value has to be exceeded with 0 on the left to the length x: The sign "+" or "-" for exponent is mandatory Decimal point and exponent "E" are mandatory Mantissa has exactly one digit before the decimal point in the range 1..9 Between mantissa and exponent is no <space> allowed Example: F15: "-00007.4321E-02"
Hx	x-digit hexadecimal value Example: H2 "1C"

3.6. TABLES

3.6.1. Error Code Table

Error Code	Description	Error Code	Description
0000	No error	0046	ROM check failed
0001	Photometer overflow	0047	Reference < limit
0002	Photometer offset too low	0048	Level detection error
0003	Photometer overflow fix again	0049	Printer motor defect
0004	Photometer timing invalid	0050	Bad temperatures
0005	Photometer offset too high	0051	Bad analyzer temperature
0006	Photometer no response	0052	Bad reagent temperature
0007	Photometer no interrupt	0053	Analyzer temp. out of range
0008	Time cyclic measurement	0054	Reagent temp. out of range
0009	Time-out filter change	0055	Reag./analys.temp.not within rng.
0010	Time-out to measurement	0056	Host interface no response
0011	Background overflow sample	0057	Host interface protocol error
0012	Background underflow sample	0058	Instrument busy
0013	Background overflow ref.	0059	Communicat. control no response
0014	Background underflow ref.	0060	Invalid sample barcode pos:___
0015	Transfer not initialized	0061	Diluent empty-dil:____
0016	P-Sample not initialized	0062	ISE-activator empty-pos:___
0017	P-Reagent not initialized	0063	Sample empty-pos:___
0018	Analyzer not initialized	0064	Control empty-pos:___
0019	Filter not initialized	0065	Program error:___
0020	X-Motor init. not possible	0066	Reagent empty-test:___
0021	Analyzer init. not possible	0067	ISE-electrode:____ over range
0022	Filter init. not possible	0068	ISE-A/D conv:___ time-out
0023	Y-Motor init. not possible	0069	Invalid sample position:____
0024	P-Sample init. not possible	0070	Standard-empty pos:___
0025		0071	Start reagent 1 empty-test
0026		0072	Start reagent 2 empty-test
0027	P-Reagent init. not possible	0073	Blank solution empty-pos:___
0028		0074	Motor-ctrl II over cur:___
0029	Analyzer cover open	0075	Adjust-wash___not possible
0030	Transfer area blocked	0076	Adjust-anal.___not possible
0031	Analyzer step error	0077	Adjust-barc.___not possible
0032	Filter step error	0078	Adjust-rack.___not possible
0033	Y-Motor step error	0079	No worklist entry-pos:___
0034	P-Sample step error	0080	(BELL)
0035	Z-Sample position error	0081	ISE-interface no response
0036		0082	ISE-module no response
0037	P-Reagent step error	0083	ISE-Module busy
0038	Z-Reagent position error	0084	ISE-interface parity error
0039	Z-Barcode position error	0085	ISE-ram error
0040	Motor-ctrl I no response	0086	ISE-leakage-error
0041	Motor-ctrl II no response	0087	ISE cleaner empty
0042	Motor-ctrl II data error	0088	ISE etcher empty
0043	No suitable racks found	0089	Result error in test result!
0044	Not enough empty cuvettes	0090	Leave test result, please!
0045		0091	Invalid barcode input

0092	Incorrect ratio test group		0130	Segment input empty
0093	Incorrect ratio formula		0131	Segment output full
0094	Worklist complete		0132	Changer X-motor time-out
0095	Missing rack		0133	Changer Y-motor time-out
0096	Tip cleaner empty		0134	Changer Z-motor time-out
0097	No water available		0135	Changer X not in position
0098	Test result occupied		0136	Changer Y not in position
0099	99 Rack reader no response[1]		0137	Changer Z not in position
0100	Memory capacity overflow		0138	Change/cooling-contr.no response
0101	Instrument initialized		0139	Changer timing invalid
0102	Test doesn't exist		0140	Rack temperature too high
0103	Ratio doesn't exist		0141	Routine worklist not complete
0104	Profile doesn't exist		0142	Stat worklist not complete
0105	Sample not found		0143	Invalid input
0106	Test not found		0144	Motor-ctrl II hardware error
0107	No rack found		0145	Motor-ctrl II lightb.-error
0108	Test/rack programming error		0146	R-S needles not parallel
0109	Rack Placement		0147	Touch error
0110	Low CPU-battery		0148	Adjust-racked. not possible
0111	Printer busy		0149	R-S needles distance is not 8 mm
0112	Privileged action		0150	LD error-samp___kHz[1]
0113	Test locked		0151	LD error-reag___kHz[1]
0114	Test in worklist and quality		0152	
0115	Ratio in worklist			
0116	All test in worklist			
0117	Instrument in process			
0118	Instrument checks in process			
0119	Worklist not empty			
0120	Multi-IF-PCB/barcode reader missing or not adjusted[1]			
0121	Test/dilution already assigned			
0122	Confirm by <ENTER>			
0123	<ENTER> to delete the already printed results			
0124	First adjust R-S needles[1]			
0125	ISE tests not assignable			
0126	Test in worklist			
0127	Test in quality			
0128	Printer lever open			
0129	No printer paper			

---

[1] only with the software release 9215.DA

3.6.2. Flag Table

Flag	significance
N	no Flag
C	calibration/control error
D	disabled
R	recalculated result
U	unit changed
A	accepted result
∧	concentrated result
∨	diluted result

Note:
The flags Ù and Ú are assigned as ASCII-values 135 and 136. This applies to 8 bit data format only; in case of 7 bit format ASCII-values 7 (bell) and 8 (BS) are transmitted.

3.6.3. Remark Table

Remark value	significance
00	no remark
01	high abs
02	non linear
03	noise
04	sign
05	> reaction limit
06	high activity
07	< test range
08	> test range
09	< reagent range
10	> reagent range
11	< blank range
12	> blank range
13	< deviation
14	> deviation
15	< conv limit
16	> conv limit
17	antigen excess
18	sample limit
19	< normal range
20	> normal range
21	calculation error
22	antigen calc error
23	< calculation range
24	> calculation range
31	< slope range
32	> slope range
34	unstable reference
35	unstable
36	< test range (check = off)
37	> test range (check = off)

Note:
A result value is to be proceeded by a < or a > as applicable, if accompanied by remarks 7, 8, 17[1] , 36, 37.

[1] only with the software release 9215.DA

3.6.4.   Unit Table

Unit value	significance	Unit value	significance
02	mol/l	21	U/l
03	mmol/l	22	mU/l
04	μmol/l	23	U/ml
05	nmol/l	24	mU/ml
06	pmol/l	25	IU/l
07	g/l	26	mIU/l
08	mg/l	27	IU/ml
09	μg/l	28	mIU/ml
10	ng/l	29	mval/l
11	g/dl	30	meq/l
12	mg/dl	31	ΔA
13	μg/dl	32	ΔA/min
14	ng/dl	33	%
15	mg/ml	34	Δ%
16	μg/ml	35	s
17	ng/ml	36	kU/l
18	pg/ml	37	kIU/l
19	μkat/l		
20	nkat/l		

This page is intentionally blank!

3.7. EXAMPLES

This chapter shows some communication examples between a HOST computer and COBAS MIRA Plus.

3.7.1. Automatic Random Transmit Mode and Master Slave Mode

The following examples are valid for the *Automatic Random Transmit Mode* and the *Master Slave Mode*.

3.7.1.1. Idle Block

Host	<SOH><LF>	
Request	08□COBAS·MIRA·HOST·□00<LF>	
10:17:10	<STX><LF>	*A simple communication check can be performed by sending an Idle Block*
	<ETX><LF>	
Idle Block	<EOT><LF>	

Instrument	<SOH><LF>	
Response	08□COBAS·MIRA·PLUS·□00<LF>	
10:17:12	<STX><LF>	*The response will also be an Idle block.*
	<ETX><LF>	
Idle Block	<EOT><LF>	

3.7.1.2. Order Entry

Only one order entry per Order Entry block

An Order Information line may have up to 8 test

Host Request 10:17:15	<SOH>_{\<LF>} 08□COBAS·MIRA·HOST·□10_{\<LF>} <STX>_{\<LF>} 50□0007□MEIER·CARL□R□A1□A3□A8□A9_{\<LF>}
Order *Entry*	<ETX>_{\<LF>} <EOT>_{\<LF>}

Please, refer to Line 50 description for important information regarding the Patient ID's use and the construction of the Patient ID

If barcodes are used set the Sample Cup number to '0000' !!.

Instrument Response 10:17:18	<SOH>_{\<LF>} 08□COBAS·MIRA·PLUS·□11_{\<LF>} <STX>_{\<LF>} 50□0007□MEIER·CARL□R□··□··□A8□··_{\<LF>}
Ord.Entry *Confirm.*	<ETX>_{\<LF>} <EOT>_{\<LF>}

Test was accepted

*Test NOT accepted
(e.g. test not defined)*

3.7.1.3. Order Deletion

An Order Information line my have up to 8 test

Host Request 10:17:21	<SOH>_{\<LF>} 08□COBAS·MIRA·HOST·□15_{\<LF>} <STX>_{\<LF>} 50□0018□WYSS·HELEN□R□A2□A3 _{\<LF>}
Order *Deletion*	<ETX>_{\<LF>} <EOT>_{\<LF>}

Only one order deletion per Order Deletion block

Instrument Response 10:17:24	<SOH>_{\<LF>} 08□COBAS·MIRA·PLUS·□16_{\<LF>} <STX>_{\<LF>} 50□0018□WYSS·HELEN□R□··□·· _{\<LF>}
Ord.Delet. *Confirm.*	<ETX>_{\<LF>} <EOT>_{\<LF>}

Tests were successfully deleted

3.7.2.   Automatic Random Transmit Mode

The following examples are valid for the *Automatic Random Transmit Mode.*
Output Mode P63: *'Result Auto Mode = ON'.*

3.7.2.1.   Patient result

Instrument	`<SOH>`_{‹LF›}
Message	`08□COBAS·MIRA·PLUS·□03`_{‹LF›}
10:34:48	`<STX>`_{‹LF›}
	`20□R□G1□GLUC□01□0001□ANDERSON·L□+5.14307E+00□02□03□O□N□00`_{‹LF›}
Patient	`<ETX>`_{‹LF›}
Result	`<EOT>`_{‹LF›}

COBAS MIRA Plus sends the results in the sequence as they have been finished.

Instrument	`<SOH>`_{‹LF›}
Message	`08□COBAS·MIRA·PLUS·□03`_{‹LF›}
10:35:18	`<STX>`_{‹LF›}
	`20□R□Z1□CA··□01□0001□ANDERSON·L□+2.12381E+00□02□03□O□N□00`_{‹LF›}
	`20□R□Z1□CA··□02□0001□ANDERSON·L□+1.98234E+00□02□03□O□N□00`_{‹LF›}
Patient	`<ETX>`_{‹LF›}
Result	`<EOT>`_{‹LF›}

Instrument	`<SOH>`_{‹LF›}
Message	`08□COBAS·MIRA·PLUS·□03`_{‹LF›}
10:39:12	`<STX>`_{‹LF›}
	`20□R□A1□AST·□01□0002□WILKINSON·□+1.70325E+00□02□21□O□N□00`_{‹LF›}
Patient	`<ETX>`_{‹LF›}
Result	`<EOT>`_{‹LF›}

3.7.3. Master Slave Mode

The following examples are valid for the *Master Slave Mode*.

3.7.3.1. Patient result

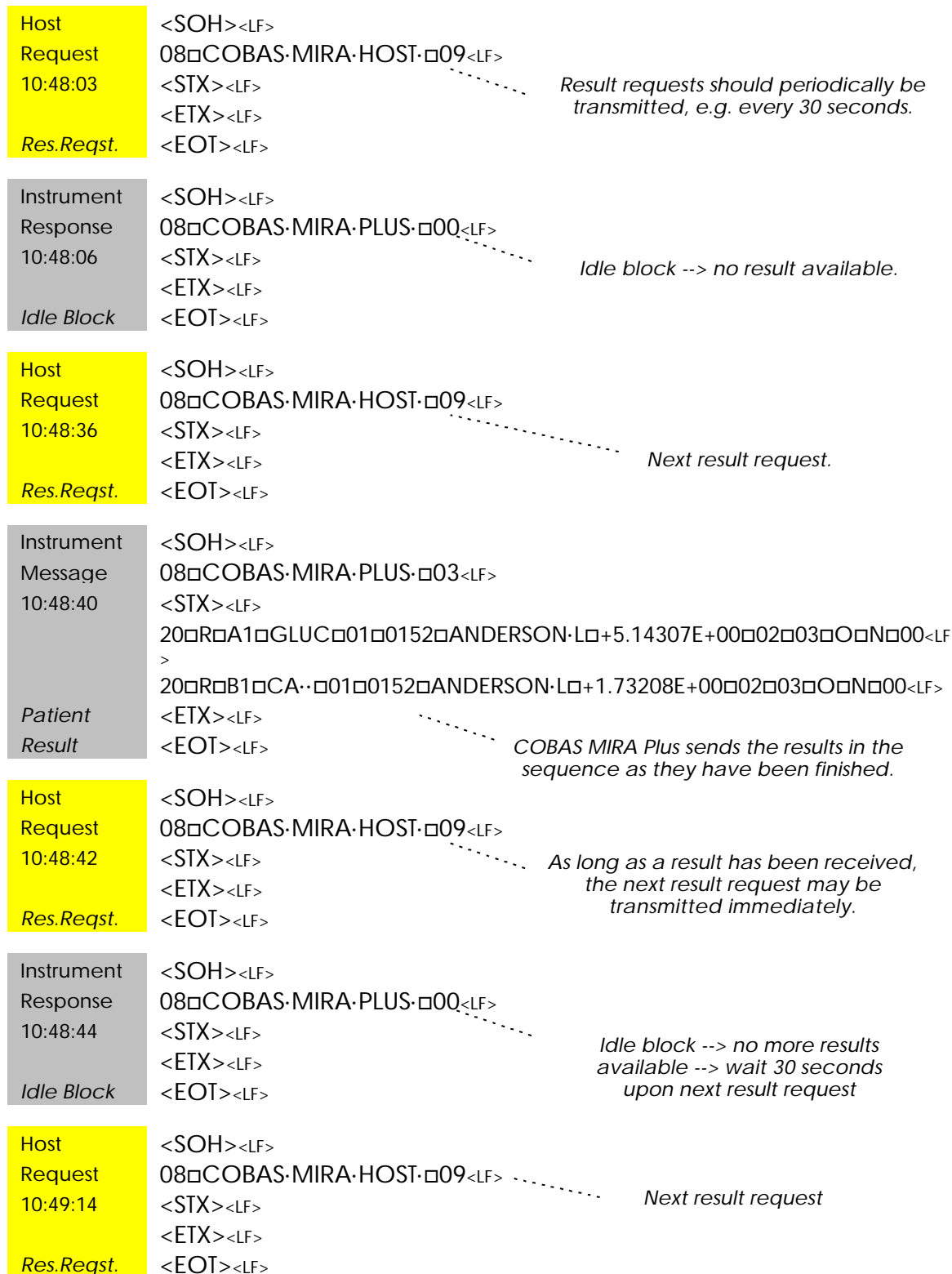|		
Host Request 10:48:03 *Res.Reqst.*	`<SOH>`_{<LF>} `08□COBAS·MIRA·HOST·□09`_{<LF>} `<STX>`_{<LF>} `<ETX>`_{<LF>} `<EOT>`_{<LF>}	*Result requests should periodically be transmitted, e.g. every 30 seconds.*
Instrument Response 10:48:06 *Idle Block*	`<SOH>`_{<LF>} `08□COBAS·MIRA·PLUS·□00`_{<LF>} `<STX>`_{<LF>} `<ETX>`_{<LF>} `<EOT>`_{<LF>}	*Idle block --> no result available.*
Host Request 10:48:36 *Res.Reqst.*	`<SOH>`_{<LF>} `08□COBAS·MIRA·HOST·□09`_{<LF>} `<STX>`_{<LF>} `<ETX>`_{<LF>} `<EOT>`_{<LF>}	*Next result request.*
Instrument Message 10:48:40 *Patient Result*	`<SOH>`_{<LF>} `08□COBAS·MIRA·PLUS·□03`_{<LF>} `<STX>`_{<LF>} `20□R□A1□GLUC□01□0152□ANDERSON·L□+5.14307E+00□02□03□O□N□00`_{<LF>} `20□R□B1□CA··□01□0152□ANDERSON·L□+1.73208E+00□02□03□O□N□00`_{<LF>} `<ETX>`_{<LF>} `<EOT>`_{<LF>}	*COBAS MIRA Plus sends the results in the sequence as they have been finished.*
Host Request 10:48:42 *Res.Reqst.*	`<SOH>`_{<LF>} `08□COBAS·MIRA·HOST·□09`_{<LF>} `<STX>`_{<LF>} `<ETX>`_{<LF>} `<EOT>`_{<LF>}	*As long as a result has been received, the next result request may be transmitted immediately.*
Instrument Response 10:48:44 *Idle Block*	`<SOH>`_{<LF>} `08□COBAS·MIRA·PLUS·□00`_{<LF>} `<STX>`_{<LF>} `<ETX>`_{<LF>} `<EOT>`_{<LF>}	*Idle block --> no more results available --> wait 30 seconds upon next result request*
Host Request 10:49:14 *Res.Reqst.*	`<SOH>`_{<LF>} `08□COBAS·MIRA·HOST·□09`_{<LF>} `<STX>`_{<LF>} `<ETX>`_{<LF>} `<EOT>`_{<LF>}	*Next result request*

... and so on ...

3.7.3.2. Rack information

<table>
<tr><td>Host</td><td><SOH>_{<LF>}</td></tr>
<tr><td>Request</td><td>08□COBAS·MIRA·HOST·□79_{<LF>}</td></tr>
<tr><td>10:52:09</td><td><STX>_{<LF>}</td></tr>
<tr><td>*Rack Info.*</td><td><ETX>_{<LF>}</td></tr>
<tr><td>*Request*</td><td><EOT>_{<LF>}</td></tr>
</table>

Instrument	<SOH>_{<LF>}
Response	08□COBAS·MIRA·PLUS·□70_{<LF>}
10:52:13	<STX>_{<LF>} *List of the racks on the instrument*
	72□0226□0006□0511□0512_{<LF>}
	01□A1□GLUC_{<LF>} *List of the available tests*
	01□Z1□AST·_{<LF>}
	<ETX>_{<LF>}
Rack Info.	<EOT>_{<LF>}

3.7.3.3. Quality control request

Host **Request** **10:52:09**	`<SOH><LF>` `08□COBAS·MIRA·HOST·□07<LF>` `<STX><LF>` `01□C1□CREA<LF>`
QC Result *Request*	`<ETX><LF>` `<EOT><LF>`

Instrument Response 10:52:13	`<SOH><LF>` `08□COBAS·MIRA·PLUS·□08<LF>` `<STX><LF>` `96□24<LF>` `<ETX><LF>`	*Test not defined!*
QC Result	`<EOT><LF>`	

Host **Request** **10:52:09**	`<SOH><LF>` `08□COBAS·MIRA·HOST·□07<LF>` `<STX><LF>` `01□A1□GLUC<LF>`
QC Result *Request*	`<ETX><LF>` `<EOT><LF>`

Instrument Response 10:52:13	`<SOH><LF>` `08□COBAS·MIRA·PLUS·□05<LF>` `<STX><LF>` `01□A1□GLUC<LF>` `30□01□12□03□94□+2.10322E+01□+1.33454E+01□+3.44332E+01□12<LF>` `30□02□12□03□94□+2.10322E+01□+1.33454E+01□+3.44332E+01□12<LF>` … … `<ETX><LF>`
QC Result	`<EOT><LF>`

# 4. APPENDIX

4.1. TABLE OF CONTENTS

4.2.   APPENDIX A

4.2.1.   Serial Interface RS-232 (CCITT V.24)

4.2.1.1.   Overview

COBAS instrument serial interfaces are based on international standards. The relevant standards are EIA RS-232D and CCITT V.24. This appendix is an extract of these standards.

The serial interface RS-232 is an asynchronous interface with character transfer. The characters are transmitted in single bits. The transfer starts with a start bit and ends with one or two stop bits. There is also a possibility to detect a bit failure with the parity bit. This is selectable as even, odd or none.
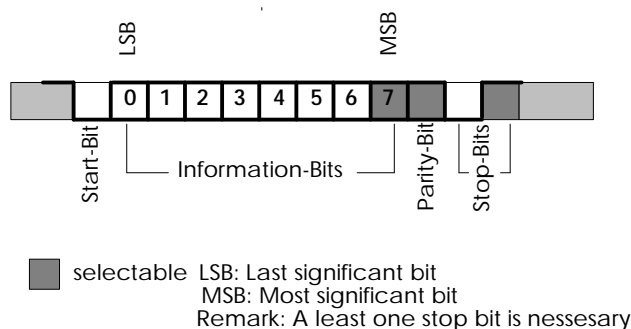
4.2.1.2.   Signal Levels

The signal levels for *data* lines are shown below:
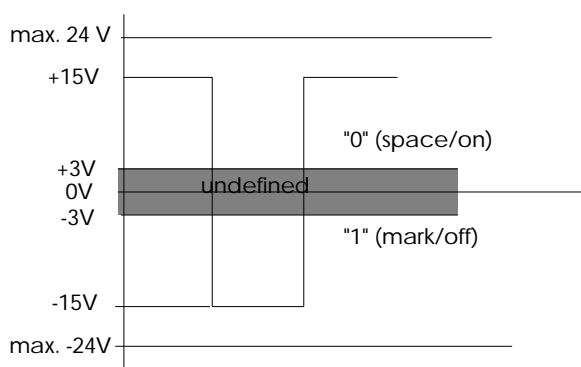
For the *control and status* lines the signal levels are as follows:

"ON"	+3V to +15V
"OFF"	-3V to -15V

4.2.1.3. Signal Description

The following table shows the most used signals There are about 25 different signals defined in the standard of CCITT V.24. Normally only a part of these are used.

CCITT	EIA	Signal name	Description
101	AA	PG	Protective Ground
102	AB	SG	Signal ground or common return
103	BA	TxD	Transmitted data
104	BB	RxD	Received data
105	CA	RTS	Request to send
106	CB	CTS	Ready for sending (clear to send)
107	CC	DSR	Data set ready
108/2	CD	DTR	Data terminal ready
109	CF	DCD	Receive line signal detector
125	CE	RI	Ring indicator

Supported by COBAS Instruments

4.2.1.4. Hardware Handshaking

The hardware handshaking works as follows:

1. The sender sets its RTS line to active. This signals the receiver to prepare to receive data from the sender station.

2. If the receiver is ready to receive data it sets the DTR line to active. This signals the sender the receiver's readiness, and the sender starts data transmission.

3. As long as the sender's CTS signal is active it is allowed to transmit data.

4. If the receive buffer in the receive station is full (or better still, before it is full!) the receiver stops the transmission by setting the DTR line to inactive and then to active again when he is again ready to receive data.

4.2.1.5. Software Handshaking

The software handshaking operates as follows:

Sender:
After receiving a <XOFF> sign the send procedure is interrupted until the receipt of an <XON> sign.

Receiver:
To interrupt a reception a <XOFF> sign is sent. Further reception can be obtained by sending a <XON> sign.

<XON> : CHAR(17), DC1
<XOFF> : CHAR(19), DC3

> *Note:*
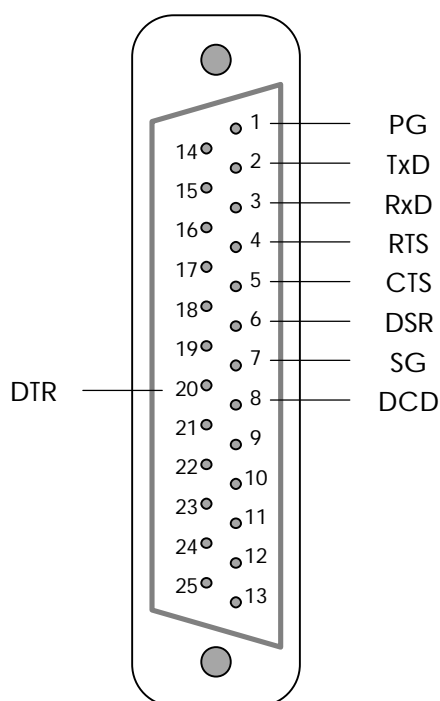> *As soon as the COBAS instrument software is started, the Host Interface is ready to receive data. No initial SW handshaking is required.*
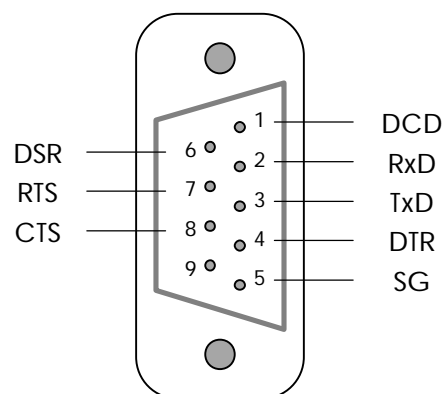
4.2.1.6. Pin Assignment

The correct pin assignment, is defined in standard ISO 2110 for 25-pole connector and in standard ISO 4902 for 9-pole connector, is shown below:

Pin assignment 25 pin connector:

Pin assignment 9 pin connector:

4.3. APPENDIX B

4.3.1. ASCII-Code

American **N**ational **S**tandard **C**ode for **I**nformation **I**nterchange. The following table shows the seven-bit code in accordance with the US standard (ANSI X3.4) for the Interchange of Information among Communication Devices. The first 32 values are reserved for control codes.

NUL 000D 00H ^@	SOH ☺ 001D 01H ^A	STX ☻ 002D 02H ^B	ETX ♥ 003D 03H ^C	EOT ♦ 004D 04H ^D	ENQ ♣ 005D 05H ^E	ACK ♠ 006D 06H ^F	BEL ● 007D 07H ^G
BS ◘ 008D 08H ^H	HT ○ 009D 09H ^I	LF ◙ 010D 0AH ^J	VT ♂ 011D 0BH ^K	FF ♀ 012D 0CH ^L	CR ♪ 013D 0DH ^M	SO ♫ 014D 0EH ^N	SI ☼ 015D 0FH ^O
DLE ► 016D 10H ^P	DC1 ◄ 017D 11H ^Q	DC2 ↕ 018D 12H ^R	DC3 ‼ 019D 13H ^S	DC4 ¶ 020D 14H ^T	NAK § 021D 15H ^U	SYN 022D 16H ^V	ETB ↨ 023D 17H ^W
CAN ↑ 024D 18H ^X	EM ↓ 025D 19H ^Y	SUB → 026D 1AH ^Z	ESC ← 027D 1BH ^{	FS ∠ 028D 1CH ^\	GS ↔ 029D 1DH ^}	RS ▲ 030D 1EH ^^	US ▼ 031D 1FH ^_
032D 20H	! 033D 21H !	" 034D 22H "	# 035D 23H #	$ 036D 24H $	% 037D 25H %	& 038D 26H &	' 039D 27H '
(040D 28H (	) 041D 29H)	* 042D 2AH *	+ 043D 2BH +	, 044D 2CH ,	- 045D 2DH -	. 046D 2EH .	/ 047D 2FH /
0 048D 30H 0	1 049D 31H 1	2 050D 32H 2	3 051D 33H 3	4 052D 34H 4	5 053D 35H 5	6 054D 36H 6	7 055D 37H 7
8 056D 38H 8	9 057D 39H 9	: 058D 3AH :	; 059D 3BH ;	< 060D 3CH <	= 061D 3DH =	> 062D 3EH >	? 063D 3FH ?
@ 064D 40H @	A 065D 41H A	B 066D 42H B	C 067D 43H C	D 068D 44H D	E 069D 45H E	F 070D 46H F	G 071D 47H G
H 072D 48H H	I 073D 49H I	J 074D 4AH J	K 075D 4BH K	L 076D 4CH L	M 077D 4DH M	N 078D 4EH N	O 079D 4FH O
P 080D 50H P	Q 081D 51H Q	R 082D 52H R	S 083D 53H S	T 084D 54H T	U 085D 55H U	V 086D 56H V	W 087D 57H W
X 088D 58H X	Y 089D 59H Y	Z 090D 5AH Z	[091D 5BH [	\ 092D 5CH \	] 093D 5DH]	^ 094D 5EH ^	_ 095D 5FH _
` 096D 60H `	a 097D 61H a	b 098D 62H b	c 099D 63H c	d 100D 64H d	e 101D 65H e	f 102D 66H f	g 103D 67H g
h 104D 68H h	i 105D 69H i	j 106D 6AH j	k 107D 6BH k	l 108D 6CH l	m 109D 6DH m	n 110D 6EH n	o 111D 6FH o
p 112D 70H p	q 113D 71H q	r 114D 72H r	s 115D 73H s	t 116D 74H t	u 117D 75H u	v 118D 76H v	w 119D 77H w
x 120D 78H x	y 121D 79H y	z 122D 7AH z	{ 123D 7BH {	| 124D 7CH |	} 125D 7DH }	~ 126D 7EH ~	127D 7FH

Description of a field:

Denotation
PC-Sign

| S O H | ☺ | 001D 01H ^A |

Decimal
Hexadecimal
Keyboard

Several 8-bit variants exist, including IBM`s PC Extended ASCII character set, in which even the control codes have graphic assignments. Other specific variants include Hewlett-Packard`s Roman-8 and ECMA-94 (European Computer Manufacters`Association).

Remark:

The fields with shading are used as control characters for the communication protocol between a COBAS Instrument and a host computer.

<SOH>	Start of Header
<STX>	Start of Text
<ETX>	End of Text
<EOT>	End of Transmission
<LF>	Line Feed
<CR>	Carriage Return
<space>	Space
<DC1>	XON
<DC3>	XOFF

4.4. APPENDIX C

4.4.1. RS-232 Cable

Detailed information to perform the correct hardware connection between COBAS Instrument and a Host system is given below.

General:

The customary connector types are mentioned in the previous appendix section. Some of the computer manufacturers use female SUB-D connectors and others male connectors. Both have the same number of pins. The most popular kind are those with 9 pole and 25 pole connectors. There are also lot of different connectors in use.

We recommend buying an off-the-shelf RS-232 cable. It will work with any COBAS instrument.

Connector housing:

To prevent electromagnetic interference we strongly recommend using full metal housing.

Connector type:

COBAS AMPLICOR : SUB-D male type, 9 pin
COBAS INTEGRA : SUB-D male type, 9 pin
COBAS MIRA : SUB-D male type, 25 pin

Cable type and length:

To reduce electromagnetic interference we recommend using shielded twisted pair cable with low inductance. The cable length should not exceed 15m.

Cable shield:

We recommend connecting the cable shield with the SUB-D chassis of the interface connector on *both* sides. This is different from generally accepted practice, but following our latest knowledge concerning EMI we recommend this way of wiring the cable shield in order to suppress EMI interference.

Connect the cable shield *only* to one SUB-D chassis if for any reason the shield voltage level is not the same on both interface connectors.

Ensure that there is *NO* connection between the cable shield and signal ground.

Handshaking:

We recommend using SW-Handshaking (XON, XOFF). Thus only a three wire cable is necessary.

Nevertheless the HW handshaking conditions must be fulfilled. The operating system usually treats the HW handshaking conditions with highest priority.
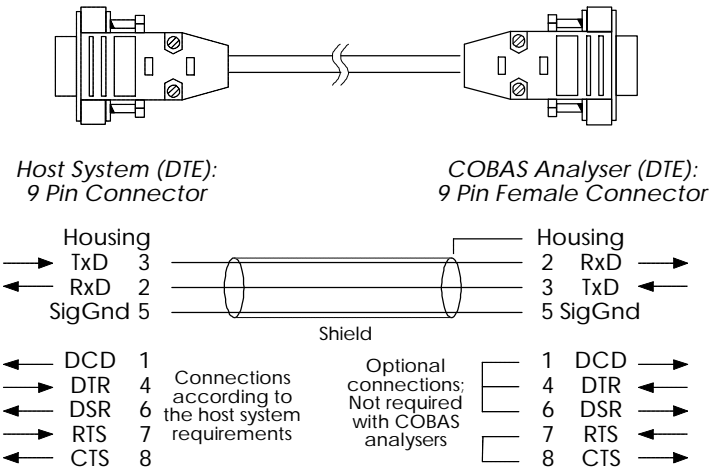
Cable Configuration:

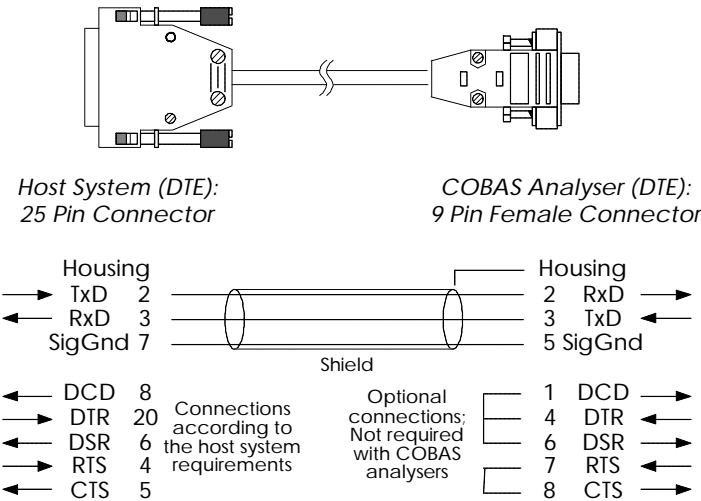The following pictures show the correct cable wiring for most applications:

> *Note:*
> *Please ensure, that the suggested wiring matches the host system requirements as well.* **Attention**; *Some Host systems are configured as DCE-devices, i.e. TxD and RxD are exchanged !!*
>
> *The shown wiring is correct for all COBAS instruments (COBAS CORE, COBAS AMPLICOR, COBAS INTEGRA, COBAS MIRA)*

Pin-Assignment RS-232 SW-Handshaking (9-pin/9-pin):

Host System (DTE):
9 Pin Connector

COBAS Analyser (DTE):
9 Pin Female Connector

Housing					Housing	
→	TxD	3			2	RxD →
←	RxD	2			3	TxD ←
	SigGnd	5			5	SigGnd

Shield

←	DCD	1	Connections according to the host system requirements	Optional connections; Not required with COBAS analysers	1	DCD →
→	DTR	4			4	DTR →
←	DSR	6			6	DSR →
→	RTS	7			7	RTS ←
←	CTS	8			8	CTS →

Pin-Assignment RS-232 SW-Handshaking (25-pin/9-pin):

Host System (DTE):
25 Pin Connector

COBAS Analyser (DTE):
9 Pin Female Connector

Housing					Housing	
→	TxD	2			2	RxD →
←	RxD	3			3	TxD ←
	SigGnd	7			5	SigGnd

Shield

←	DCD	8	Connections according to the host system requirements	Optional connections; Not required with COBAS analysers	1	DCD →
→	DTR	20			4	DTR ←
←	DSR	6			6	DSR →
→	RTS	4			7	RTS ←
←	CTS	5			8	CTS →

r

Pin-Assignment RS-232 HW-Handshaking (9-pin/9-pin):

Host System (DTE):
9 Pin Connector

COBAS Analyser (DTE):
9 Pin Female Connector

Housing			Shield	Housing		
→	TxD	3		2	RxD	→
←	RxD	2		3	TxD	←
	SigGnd	5		5	SigGnd	
→	DTR	4		6	DSR	→
←	DSR	6		4	DTR	←
→	RTS	7		1	DCD	→
←	CTS	8		7	RTS	←
←	DCD	1		8	CTS	→

Pin-Assignment RS-232 HW-Handshaking (25-pin/9-pin):

Host System (DTE):
25 Pin Connector

COBAS Analyser (DTE):
9 Pin Female Connector

Housing			Shield	Housing		
→	TxD	2		2	RxD	→
←	RxD	3		3	TxD	←
	SigGnd	7		5	SigGnd	
→	DTR	20		6	DSR	→
←	DSR	6		4	DTR	←
→	RTS	4		1	DCD	→
←	CTS	5		7	RTS	←
←	DCD	8		8	CTS	→

4.5.   APPENDIX D

4.5.1.   COBAS Instrument Code

The following table shows the codes used to identify the different COBAS instruments.

Instrument Code	Name of Instrument
01	COBAS FARA
02	COBAS MIRA
03	COBAS BACT
04	COBAS MICRO
05	EIA - Photometer
06	COBAS CORE
07	COBAS MIRA `89 / PLUS
08	COBAS MIRA S / PLUS CC
09	COBAS INTEGRA 700
10	COBAS MFC
11	
12	COBAS PRIME
13	DP25
14	COBAS INTEGRA 400
15	COBAS AMPLICOR
16	COBAS CORE II
17	
18	