22 SOFTWARE - SERIAL COMMUNICATION: ANALYZER <-> HOST COMPUTER

Important Notice:

Any modification to the Variable Serial Protocol is restricted to qualified personnel only. The Company guaranties the correct performance of the internal serial protocol. The responsibility for any malfunction arising out of any modifications to the scripts of the Variable Serial Protocol rests with the customer.

WARNING

This information regards the setting up of the barcode for sample tubes identification. The reading of the sample barcode label has the same progression as patient code.
For example: Once a patient code of 15 characters has been entered, then a code of 8 characters followed by 7 empty spaces to reach the 15 characters is sent.
The code read on the barcode label must have the same sequence 8 + 7 for correct detection.

IMPORTANT NOTICE ON DATA TRANSMISSION

Any character below the "space" (value 32 decimal or 0x20 hexadecimal – the “space” is not included) at the beginning or at the end of a patient code will be deleted. In this way it will be possible to have a patient code shorter that 15 characters, without being obliged to use the variable protocol. In case, after eliminating these characters, the resulting patient code is empty, it will automatically be changed into '{0}'

22.1 GENERAL

The analyzer allows bi-directional communication through RS 232C serial connection with any host computer.

The particular feature of the dialog is that it is always the host computer, which initiates the communication for either transmitting patient list or for receiving the results.

To initiate any communication the host computer will have to send to analyzer the character STX (0x02) and expect the character ACK (0x06) as a response. At this point the host computer will send data to the analyzer and terminate the communication by sending the character EOT (0x04).

It is important to remember that any communication is followed by a response from the analyzer.

It must be noted that if the parameter to be transmitted is shorter in length than the length requirement of the communication protocol than a space must be added before or after. For example the analysis have length 4, therefore to send a code GLY one must add a space after to reach the length of 4 characters.

PATIENT TRANSMISSION

- Start communication with sequence STX<->ACK.....
- Send patient code..... (15 characters)
- Send list type for patient insertion..... ("T" for Routine or "R" for STAT)
- Send type of serum..... ("S" for Serum or "U" for Urine)
- Send if the patient is a clone..... ("Y" for Yes or "N" for No)
- Transmit position of cup..... ("00" unknown)
- Send number of tests to be executed..... (from "01" to "99")
- Send codes of tests to be performed..... (4 characters)
- Send Check-Sum..... (3 characters)
- Send end transmission character EOT.....
- Wait for response from the analyzer..... (2 characters)

If the communication is successful then the analyzer responds with character "Y" followed by a byte, which identifies the position where patient has been inserted. In case the communication was unsuccessful, then the analyzer responds with "N" followed by a byte identifying the type of error. The possible errors generated by the analyzer in response to the invalid insertion of patient are as follows:

0x01	Check-Sum Error
0x02	Unknown Command
0x03	Routine/STAT field Error
0x04	Serum/Urine field Error
0x05	Clone Yes/No field Error
0x06	Cup position Error
0x07	Number of Analysis field Error
0x08	Wrong Number of Test
0x09	Position already in execution
0x0A	Cloning impossible
0x0B	Code duplicated
0x0C	One or more analysis not present in the analyzer
0x0D	One or more analysis not present in the current plate
0x0E	Too many analysis for the patient
0x0F	Constant data not valid
0x10	Variable data not valid
0x11	Variable data in analysis not valid
0x12	No patient to repeat
0x13	The serum field in the patient to be repeated is different from the memorized one
0x14	Patient to be repeated, but the list is already full
0x15	Patient to be repeated, but the list is different
0x16	The assigned position is already in use
0x17	Already existing or performed patient, it is not a clone and belongs to a supplementary (extra) list
0x18	Already performed patient, but no repetitions or clones are active

For example to send a patient with code 000000000000001, serum type and with analysis GLY, BUN and CHO onto the STATS list, then one must send the following sequence of characters (excluding initial sequence STX<->ACK):

000000000000001RSN0003GLU BUN CHO 142<EOT>

Where:

000000000000001.....	Patient code
R.....	Identifies STATS list
S.....	Identifies the type of patient (in this case: Serum)
N.....	Identifies that the patient is not a clone
00.....	Unknown position (the analyzer will insert the patient in a convenient position)
03.....	Identifies the number of test to be executed.
GLY, BUN, CHO.....	Test codes (observe the space after each code to reach the 4 characters limit)
142.....	Identifies the Check-Sum
<EOT>.....	This character ends communication

22.2 RESULTS RECEPTION

There are three commands for receiving reports from the analyzer:

R.....	Reception of next available report
L.....	Reception of the last report sent (in case of reception problems)
A.....	Reception of the first available report (in case one desires to receive again all the reports)

The commands R, L, and A require standard communication or the procedure STX<->ACK and the character EOT to end communication.

As a response to one of these three commands the analyzer sends the requested report (if available) or the character NAK (0x15) if there is no report to be sent. It must be borne in mind that after a run test the reports are not immediately available for transmission as these need validation. To do this, go to Utility menu, RS232 and enable the option "Accept result to be sent". This operation must always be performed after each run test or groups of run test.

There is also an additional option for performing validation operation automatically. Go to Setup of the analyzer (Menu Utility, Setup Analyzer), go to the Serial (fourth from the left) and enable the option "All results must be sent automatically (without validate)" at the bottom of the page.

In case of positive response to the request for a report the analyzer will transmit the data which will have a different format, depending on the selection made in the **Send also the controls** option (enabled or disabled - see xxx).

Data transmitted with the Send controls option disabled:

Patient code.....	15 characters
List type.....	"T" for Routine or "R" for STATS
Sample type.....	"S" for Serum or "U" for Urine
Number of reports.....	3 characters
For each report:	
Analysis code.....	4 characters
Result.....	7 characters
Check-Sum.....	3 characters
<EOT>	

The following is an example of eventual response to the data sent in "Sending a patient":

000000000000001RS003GLU 000.000BUN 0010.10CHO 00100.0253<EOT>

000000000000001.....	Patient code
R.....	Identifies STATS list
S.....	Identifies the type of patient (in this case: Serum)
003.....	Identifies numbers of reports
GLU.....	First test code
000.000.....	GLU test result
BUN.....	Second test code
0010.10.....	BUN test result
CHO.....	Third test code
00100.0.....	CHO test result
253.....	Identifies the Check-Sum
<EOT>.....	This character ends communication

transmitted with the Send controls option enabled:

Patient code.....	15 characters
List type.....	"T" for Routine or "R" for STATS
Sample type.....	"S" for Serum or "U" for Urine
Control sample	“Y” for control or “N” for normal sample
Known Control	“K” for Known or “U” for Unknown
Control level	3 characters
Control lot number	20 characters
Number of reports.....	3 characters
For each report:	
Analysis code.....	4 characters
Result.....	7 characters
Check-Sum.....	3 characters
<EOT>	

Note: when sending a sample, which is not a control, the control lines (fields) will have the following values:

Control Sample	“N”
Known Control	“K”
Control level	“001”
Control lot number	“ “

The following is an example of eventual response to the data sent in "Sending a patient":

000000000000001RSNK001 003GLU 000.000BUN 0010.10CHO 00100.0063<EOT>

Where:

000000000000001.	Patient code
R..	Identifies STATS list
S..	Identifies the type of patient (in this case: Serum)
N	The sample is not a quality control
K	Type of control: known
001	Control level 1
	Control lot number (20 spaces)
003.	Identifies numbers of reports
GLU.....	First test code
000.000.....	GLU test result
BUN.....	Second test code
0010.10.....	BUN test result
CHO.....	Third test code
00100.0.....	CHO test result
063.....	Identifies the Check-Sum
<EOT>.....	This character ends communication

22.3 CALCULATION OF CHECK-SUM

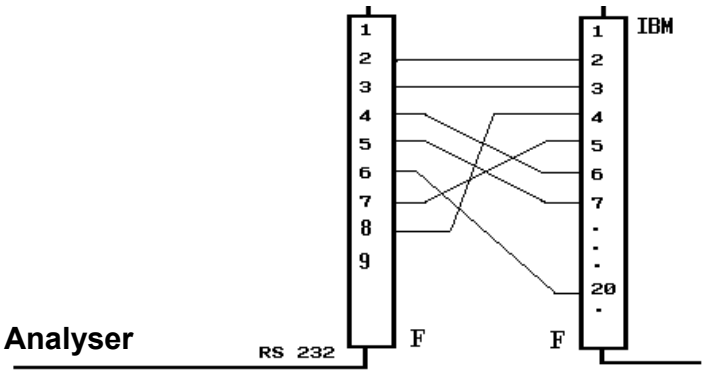
This procedure calculates a Control code in accordance with the transmitted or received data. An algebraic sum of ASCII values of all the sent characters is executed. For example the character "A" has ASCII value 65 - 0x41.

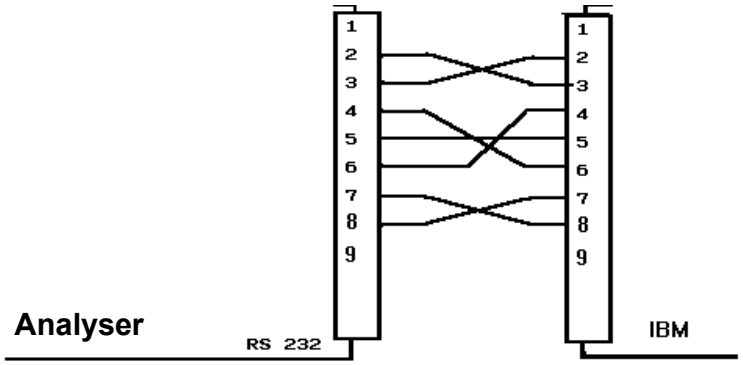
Consequently the module 256 of the found value is executed (balance of dividing the value by 256). This is the Check-Sum to be sent.

IMPORTANT NOTE FOR VARIABLE COMMUNICATION PROTCOL

THE TERMINATING CHARACTERS OF VARIABLE LENGTH STRINGS DO NOT ENTER INTO THE CHECKSUM CALCULATION

22.4 WIRING DIAGRAM OF INTERFACE CABLE





22.5 VARIABLE COMMUNICATION PROTOCOL

Introduction
The variable serial protocol has been designed to provide the user with possibility to personalize the transmitted and received data from the analyzer.

ATTENTION: WHEN A VARIABLE COMMUNICATION PROTOCOL IS CREATED OR MODIFIED (BOTH IN TRANSMITTING AND/OR RECEIVING), IT IS ABSOLUTELY NECESSARY TO SAVE BOTH TRANSMITTING AND RECEIVING SCRIPS AND TO COMPILE AGAIN BOTH OF THEM, EVEN THOU ONLY ONE HAS BEEN MODIFIED.

The default scripts are similar to those of the standard serial protocol, but please note that the character used to set the type of list is different between the two protocols. In the standard internal protocol, the character "T" identifies the routine list and "R" identifies the urgent samples list. In the variable Protocol, the character "R" identifies the routine list and "S" identifies the urgent samples list.

The user can transmit or receive in addition to preset data (patient code, analysis code, results etc.), also the simple text strings and/or characters in order to meet the personal requirements.

Not only the user can decide to send or receive numerical information (for example number of tests) not as single byte but as a preset numerical string or vice versa.

For example the user can decide to receive something like:

"Initiate analysis data"
<Analysis data true and typical>
"End analysis data"

Where the phrases "Initiate analysis data" and "End analysis data" do not refer to any preset data by the analyzer but serve only for monitoring communication process (can be useful for inserting specific markers on those programs which obtain information from text files).

It is obvious that the protocol of initiation and end of communication, the commands for the request of report, and the analyzer responses in case of error or success remain identical to the usual preset serial communication.

NOTE:

a) If a check-sum is omitted in a communication then the analyzer will not control it.
b) The following numbers have been used to represent the error codes relevant to sending a patient to the analyzer as regards the parameters not part of the standard serial communication:

0x0F	Data (constant) sent to a TAG #Char, #String or #Stringn does not fall within the possible values range
0x10	Data (variable) sent to a TAG #Char, #String or #Stringn is not valid
0x11	An analysis variable is outside the **SET BEGIN/END** relative to the analysis

Notes regarding Scripts
A script is a text document. Each one of the single commands must each reside in a different line and be complete. In other words a single command cannot be divided into more lines.

Stringn 'Name'|$10 -> *Valid line*

Stringn 'Name'|$10 **char** 'A' -> *Invalid line*

Stringn 'Name' -> *Invalid Command*
|$10

An editor for writing, modifications, saving and compiling of one or more scripts is accessible inside the program (setup function). In any case it is possible to write a script with any text editor (DOS or Windows) like Notepad of Windows or the EDITOR of the DOS. It is not possible to import documents written with UNIX as the characters used for going to the next line are different from the ones used by the DOS or Windows.

CAUTION!

If one wants to use the script stored in a removable disk (for example floppy disk) then it will be necessary to copy it on the hard disk.

TYPE OF DATA

Character: Identifies a single character, can pass as printable character (enclosed between single apostrophes), as decimal ASCII value (followed by symbol **$**) or else hexadecimal ASCII value (followed by **0x**).
If for example we want to identify the character A **(decimal value** 65 **or hexadecimal value** 41**) then we can write** 'A', $65 **or** 0x41.

String: Identifies a sequence of printable characters enclosed in single apostrophes, for example: **'this is a string'**.

Comment: Identifies a portion of test (preceded by a character **;** which will not be compiled but will serve as note only for the programmer.

Variables: These are particular sequence of characters preceded by the symbol **#**, which will be used by the program for storing internal information (patient code, analysis name and etc.), refer to "TABLE 1 - TRANSMISSION/RECEPTION".

There are also variables for direct uses, which allow for identification of any character below ASCII 32 (space) to facilitate the writing of the script (for example, one can use the variable **#EOT** to identify the character **$4**), see "TABLE 2 - INTERNAL VARIABLES".

SCRIPT FUNCTIONS

String: Identifies a string of variable length ending with a particular character.

Syntax:
String <string>**I**<Terminator>

Where
<String> Transmit/receive string
<Terminator> End character

Note:
It is not possible to use the variables like parameter <Terminator>

Example:
String 'Hello Word'|$0
String 'My String'|'@'
String #Variable1|0x10

Stringn: Identifies a string of fixed length

Syntax:
Stringn <*String*>|<*Length*>

Where
<String> Transmit/receive string
<Length> String length

Note:
If the length of the text strings is less than the data length then a series of spaces will be added on the right to reach the data preset length. In case the text string is longer than the data length then the string end will be cut off to match the data length.

If the length of the numerical values is less than the data length then a series of characters '0' will be added to the left to reach the preset data dimension. In case the length of the numerical values string is longer than the data length then the string will be truncated to match the data length.

It is not possible to use variables as parameter <Length>.

Example:
Stringn 'Hello Word'|$40
Stringn #Variable1|0x10

Char: Identifies a single character (or single byte)

Syntax:
Char <*Character*>

Example:
Char 'H'
Char $20
Char 0x10
Char #STX

Set: Identifies the beginning and the end of the group of repetitive commands

Syntax

Set Begin<*Name of group*>
Begin repetitive group

Set End<*Name of group*>
End repetitive group

Note:
Actually the **ANALYSESDATA** is the unique SET present, which identifies the analysis in transmission/reception.
Only one command **SET BEGIN** and one command **SET END** can be present in a script.
A script must always contain the command **SET**.
The variable **PATIENTNUMBERTEST** must be present before the command **SET**.

COMPILATION ERRORS

One or more errors due to incorrect script writing or the system error may show up during compilation of a script. The compiler shows the error code, the description of error, and the line where it has been detected.
The following table shows the error codes, description, and the possible causes:

Error Code	Description	Possible Causes
1	Unknown command	An invalid command has been inserted in the commands of script.
2	A string request	A string as first parameter for String or Stringn command has not been inserted.
3	A number request	A string like parameter <lunghezza - length> of command Stringn has been inserted.
4	Invalid number format	Inserted invalid decimal or hexadecimal number.
5	Excessive data	a) Inserted more than two parameters for command String or Stringn. b) Inserted more than one parameter for the command Char or Set.
6	Invalid data	A string for command Char has been inserted.
7	String Terminator Request	The end (') character of a string not found.
8	Too little data	a) Inserted less than two parameters for command String or Stringn. b) No parameter inserted for command Char or Set.
9	Invalid String Length	The string length for Stringn command is less than 0 or more than 128.
10	Empty string	a) An empty string inserted for the command String or Stringn. b) Inserted a character identified as "
11	Unknown variable	a) Tried to transfer an invalid variable in the list of internal variables. b) Tried to use a transmission variable in the script of reception or vice versa.
12	Damaged file	Hard disk error. Contact Sales/Service.
13	Unknown file	Internal error. Probably damaged program. Reinstall the program. If the problem persists contact sales/service.
14	Incorrect identifier in the SET command	a) The text SET BEGIN or SET END not written. b) A different value from ANALYSEDATA transferred as <Group name> for the SET command.
15	Damaged exit file	Hard disk error. Contact sales/service.
16	SET command not closed	The SET END not inserted in the script.
17	Too many SET commands	More than one SET BEGIN command inserted.
18	SET command not found	The SET BEGIN command not found in the script.
19	Incorrect variable for SET command	A different value from ANALYSEDATA transferred as <Group name> for the SET command.
20	Variable not found before the SET command	The highlighted variable required in the script before the SET BEGIN command.
21	The variable must be String type	The highlighted variable must be String type, not Char
22	Already occupied position	An already occupied position on the plate has been entered
23	Patient exists but in different lists	An already existing code (or already executed) on the plate has been entered, but the list is different.
24	Patient executed but no repetition	An already processed code has been entered without indicating a repetition or a clone.

TABLE 1 – TRANSMISSION

The following variables are used for the transmission of a report from analyzer to the host computer:

Variable	Usage	Type of valid data
PATIENTCODE	Patient Code	String
PATIENTNAME	Patient Name	String
PATIENTSURNAME	Patient Surname	String
PATIENTGROUP	Group [1]	String Character
PATIENTLISTTYPE	List [2]	String Character
PATIENTTYPE	Method Type [3]	String Character
PATIENTNOTE	Descriptive Note	String
PATIENTNUMBERTEST	Number of Results	String Character
CHECKSUM	Check-Sum	String Character
ANALYSESCODE	Analysis Code	String
ANALYSENAME	Analysis name	String
ANALYSESTYPE	Analysis Type [4]	String Character
ANALYSESCONCENTRATION1	1st Concentration	String
ANALYSESCONCENTRATION2	2nd Concentration	String
ANALYSESFLAGS1	Flag 1st Result	String
ANALYSESFLAGS2	Flag 2nd Result	String
ANALYSESMINVALUE	Minimum Value	String
ANALYSESMAXVALUE	Maximum Value	String
ANALYSESUM1	1st Unit of Measurement	String
ANALYSESUM2	2nd unit of measurement	String
ANALYSESUMFACTOR	Unit Factor	String
ANALYSES2RESULT	Does the 2nd result exists? [5]	String Character
ANALYSESSERUMTYPE	Method Type [3]	String Character
ANALYSESURINE24H	Urine in 24/h	String
ISCONTROL	Indentifies if it is a QC [6]	String Character
CONTROLTYPE	Known or Unknown control [7]	String Character
CONTROLLEVEL	Control level	String Character
CONTROLLOTNAME	Control lot number	String

[1] Identifies Male, Female or Child (Select one of these):
'M' : Male
'F' : Female
'C' : Child

[2] Identifies Routine or STAT (Select one of these):
'R' : Routine
'S' : STAT

Transmitting patient from archive will always have identifier of Routine.

[3] Identifies Serum or Urine (Select one of these):
'S' : Serum
'U' : Urine

[4] Identifies Clinical Chemistry, or Relation Analysis (Select one of these):
'C' : Clinical Chemistry
'R' : Relation Analysis

[5] Identifies if the 2nd result exists or not (Select one of these):
'Y' : 2nd result exists
'N' : 2nd result does not exist

- If only the final result is desired then always refer to variables pertaining to 2nd result.
- In case of the absence of 2nd result then its variables will have the same values of the 1st result.

[6] Identifies if it is a quality control or a sample (only one of these)
"Y": Quality control
"N": Normal sample
[7] Indentifies if it is a Known or an Unknown quality control
"K": Known Control
"U": Unknown control

TABLE 2 – RECEPTION

The following variables are used for reception of a patient by the analyzer:

Variable	Usage	Type of valid data
PATIENTCODE	Patient Code	String
PATIENTNAME	Patient Name	String
PATIENTSURNAME	Patient Surname	String
PATIENTLISTTYPE	List [1]	String Character
PATIENTGROUP	Group [2]	String Character
PATIENTTYPE	Method Type [3]	String Character
PATIENTURINE24H	Urine in 24/h	String
PATIENTNOTE	Descriptive Note	String
PATIENTISCONTROL	If the patient is a control [4]	String Character
PATIENTCONTROLKNOK	If it is a known control [5]	String Character
PATIENTCONTROLLEVEL	Control Level [6]	String Character
PATIENTCLONE	If it is a clone [7]	String Character
PATIENTCUPPOSITION	Vial (Cup) position[8]	String Character
PATIENTNUMBERTEST	Number of test	String Character
CHECKSUM	Check-Sum	String Character
ANALYSESCODE	Analysis Code	String

[1] Identifies Routine or STAT (Select only one of these):

$0	: Routine
$1	: STAT
'0'	: Routine
'1'	: STAT
'R'	: Routine
'S'	: STAT
'ROUTINE'	: Routine
'STAT'	: STAT

[2] Identifies Male, Female or Child (Select only one of these):

$0	: Male
$1	: Female
$2	: Child
'0'	: Male
'1'	: Female
'2'	: Child
'M'	: Male
'F'	: Female
'C'	: Child
'MAN'	: Male
'FEMALE'	: Female
'CHILD'	: Child

[3] Identifies Serum or Urine (Select only one of these):

$0	: SERUM
$1	: URINE
'0'	: SERUM
'1'	: URINE
'S'	: SERUM
'U'	: URINE
'SERUM'	: SERUM
'URINE'	: URINE

(4) Identifies a Control or a Sample (Select only one of these):

$0	: Sample
$1	: Control
'0'	: Sample
'1'	: Control
'N'	: Sample
'Y'	: Control
'S'	: Sample
'C'	: Control
'NO'	: Sample
'YES'	: Control
'SAMPLE'	: Sample
'CONTROL'	: Control

(5) Identifies a Known or Unknown Control (Select only one of these):

$0	: Unknown
$1	: Known
'0'	: Unknown
'1'	: Known
'N'	: Unknown
'Y'	: Known
'U'	: Unknown
'K'	: Known
'NO'	: Unknown
'YES'	: Known
'UNKNOW'	: Unknown
'KNOW'	: Known

(6) Identifies Control Level (Select only one of these):

$1	: Level 1
$2	: Level 2
$3	: Level 3
'1'	: Level 1
'2'	: Level 2
'3'	: Level 3
'L'	: Level 1
'N'	: Level 2
'A'	: Level 3
'LOW'	: Level 1
'NORMAL'	: Level 2
'ABNORMAL'	: Level 3

(7) Identifies if it is a Clone (Select only one of these):

$0	: Normal
$1	: Clone
'0'	: Normal
'1'	: Clone
'N'	: Normal
'Y'	: Clone
'NOCLONE'	: Normal
'CLONE'	: Clone

(8) If this field has the value 0, the position is automatically assigned by the analyser
Note:

It is possible to repeat the patient if:

1. It is not present among the executed patients
2. No free position exists on the plate
3. The patient has already been inserted in the current list
4. The lists of execution are different
5. The Serum/Urine field is different
6. It is also selected as clone

TABLE 3 – INTERNAL VARIABLES

Variables	Decimal	Hexadecimal
NUL	$00	0x00
SOH	$01	0x01
STX	$02	0x02
ETX	$03	0x03
EOT	$04	0x04
ENQ	$05	0x05
ACK	$06	0x06
BEL	$07	0x07
BS	$08	0x08
TAB	$09	0x09
LF	$10	0x0A
VF	$11	0x0B
FF	$12	0x0C
CR	$13	0x0D
SO	$14	0x0E
SI	$15	0x0F
DLE	$16	0x10
DC1	$17	0x11
DC2	$18	0x12
DC3	$19	0x13
DC4	$20	0x14
NAK	$21	0x15
SYN	$22	0x16
ETB	$23	0x17
CAN	$24	0x18
EM	$25	0x19
SUB	$26	0x1A
ESC	$27	0x1B
FS	$28	0x1C
GS	$29	0x1D
RS	$30	0x1E
US	$31	0x1F

SCRIPT EXAMPLES

The examples outlined here are the transformation in script of the standard routine of the patient reception by the analyzer.

Stringn #PatientCode|$15
Char #PatientListType
Char #PatientType
Char #PatientClone
Stringn #PatientCupPosition|$2
Stringn #PatientNumberTest|$2

Set #BeginAnalysesData
Stringn #AnalysesCode|$4
Set #EndAnalysesData

Stringn #CheckSum|$3

The following are the details of the above Scripts:

Stringn #PatientCode|$15
Patient Code of fixed length equal to 15 characters

Char #PatientListType
Type of list (Routine/STAT) as single character

Char #PatientType
Serum type (Serum/Urine) as single character

Char #PatientClone
Identifies if the patient is or is not a clone (single character)

Stringn #PatientCupPosition|$2
Position of serum cup (string of fixed length equal to 2 characters)

Stringn #PatientNumberTest|$2

Number of tests to be executed (string of fixed length equal to 2 characters)

Set #BeginAnalysesData
Beginning of analysis codes

Stringn #AnalysesCode|$4
An analysis code of fixed length equal to 4 characters. It must be entered for each type of test as per qty indicated in the #PatientNumberTest.

Set #EndAnalysesData
End of analysis codes

Stringn #CheckSum|$3
Check-Sum (transferred as a string of fixed length equal to 3 characters)

The following examples are the transformation in script of the standard routine for the transmission of a report by the analyzer to the host computer:

Stringn #PatientCode|$15
Char #PatientType
stringn #PatientNumberTest|$3

Set #BeginAnalysesData
Stringn #AnalysesCode|$04
Stringn #AnalysesConcentration2|$7
Set #EndAnalysesData

Stringn #CheckSum|$3

The details of the above scripts are as follows:

Stringn #PatientCode|$15
Patient Code of fixed length equal to 15 characters

Char #PatientType
Serum type (Serum/Urine) as single character

stringn #PatientNumberTest|$3
Number of results to be sent (a string of fixed length equal to 3 characters)

Set #BeginAnalysesData
Beginning of zone repeated for the number of results to be sent (see #PatientNumberTest)

Stringn #AnalysesCode|$04
An analysis code of fixed length equal to 4 characters

Stringn #AnalysesConcentration2|$7
Concentration referred to the analysis code as per #AnalysesCode (a string of fixed length equal to 7 characters)

Set #EndAnalysesData
End of zone repeated for the number of results to be sent

Stringn #CheckSum|$3
Check-Sum (transferred as a string of fixed length equal to 3 characters)

22.6 SERIAL COMMUNICATION TEST PROGRAMS

22.6.1 PROGRAM COMUNICA.EXE:

It is a simple communication program for sending command characters to the analyzer and receive any response.

At the start the only input to the program is the number of the communication port (from 1 to 4).

A blue screen divided into two sections is displayed. In the upper section the characters coming from the analyzer are displayed, while the lower section displays the characters sent to the analyzer.

The only special keys used are F1 to clear the screen and F10 for exiting the program.

The special characters (with values less than 32) are displayed in ASCII notations along with their values.

For example the Character EOT - value 4 - will be shown as EOT (4).

To send a special character (with values less than 32 or higher than 124) it is necessary to keep pressed the ALT key and simultaneously to write the value of the character to be sent using numerical keys. For example to send EOT it is necessary to keep the ALT key pressed and simultaneously enter the value 4 through the numerical key and then release the ALT key.

22.6.2 PROGRAM BTPLUS.EXE:

It is a simple communication program that simulates the host computer. At the start it is necessary to identify the number of communication port (from 1 to 4) and the desired procedure (Transmission or Reception).

In case the Transmission is selected the program will ask for patient code (from 1 to 15 characters), the test number (from 1 to 9) and the relevant analysis code for each test (for example: BUN).

It is a good practice to use the same analysis codes, which the analyzer has memorized in the plate actually in use, if otherwise then an error will result in the transmission phase.

Now the program will execute an initialization procedure of communication with the analyzer, will send patient data and wait for the outcome of transmission.

At the end the screen will display the outcome of the operation or show the position number of the plate where the patient has been inserted or explanation of error code sent by the instrument (for example: Patient Code Duplicated).

If the Reception procedure is selected, then the program will begin initialization of communication with analyzer, will ask for data of the next report ready for serial dispatch and show data of relevant downloaded report.

If there are no reports to be received, then a relevant message will be displayed.

Every time the program waits for a response from the analyzer, in case of problem it is possible to abort the current operation by simply pressing the Esc (Escape) key.

NOTE:

The program BTPLUS.EXE works only with the internal serial protocol. The option "Send controls" in the Setup must be disabled.

Both the programs must reside in the computer connected serially to the analyzer through appropriate cable indicated in the Operators Manual.

The computer must be an IBM compatible equipped with DOS operative system: Windows 95, Windows 98, or Windows 2000. The operating systems such as MAC, UNIX, Windows ME or XP are not supported.

Since the programs operate in DOS ambience, therefore in case the Windows operating system is used then it will be necessary to open a DOS shell (the command Prompts of MS-DOS is found in the menu Programs, Accessories - accessed through the Start button on the bottom left of the screen).

Both the programs use serial port with the following setups:

Baude-Rate.....	9600
Stop-Bits.....	1
Parity.....	None
Hand-shake.....	Hardware

IMPORTANT NOTICE: These two programs are in the installation disk under Utility folder.

THE PRINTER MUST BE INSTALLED BEFORE THE OPERATIVE PROGRAM!

Open the Printers window ("Start" button, "Settings", "Printers") and click on "New Printer", then follow instructions. Otherwise, insert the printer's installation disk into the driver and follow the printer's setup instructions.