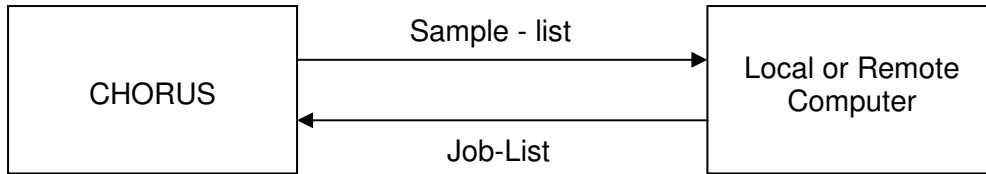


INDEX

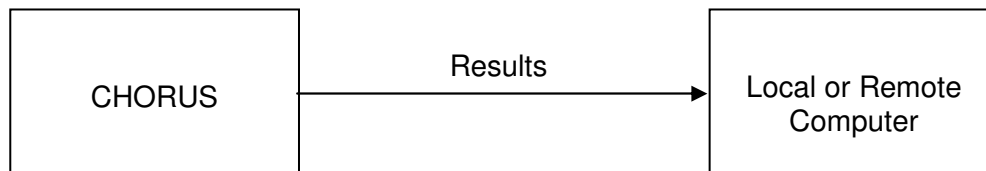
1	Introduction	2
2	Communication protocol	2
3	Interface programming	3
4	Serial communication frame.....	4
5	Enquiry.....	5
6	Sample programming.....	6
7	Result management.....	9
8	Test application.....	11

1 INTRODUCTION

Programming of samples



Filing of results



2 COMMUNICATION PROTOCOL

The following document describes the protocol for serial communication between the CHORUS instrument, manufactured by Diesse Diagnostica Senese S.p.A., and a local or remote computer. The term “local Computer” means a Personal Computer positioned near the CHORUS. The term “remote Computer” means a (generally large-scale) computer, the terminals of which are normally placed at the reception, and are able to handle acceptance of patients and most of the computerized instruments in the different laboratories. In this case, the term “Host-Computer” is often used.

In general, the term “**CHORUS Hosting**” means the interface between the analyzer and the computer, to allow the tests to be programmed on the instrument and corresponding results to be received with the aid of a Computer. In particular, the computer may implement one or both of the following features:

1. receive the sample list from the CHORUS and send back the related Job-list
2. receive sample results

This exchange of data takes place through a serial protocol using a RS-232C connection.



3 INTERFACE PROGRAMMING

Pinout of the Host connection cable is shown in the following picture (pins shown as unconnected may or may not be connected because not used):



Standard Serial Cable

Data exchange is implemented by a RS232C serial interface that is programmed with the following parameters:

bitrate: set up by WinChorus++ (v 6.43 or above) through the hardware parameter *miscellanea\hosting speed* between the following values (default bitrate is 38400 baud):

bitrate	Parameter
38400 baud	0
19200 baud	1
9600 baud	2
4800 baud	3
2400 baud	4
1200 baud	5
600 baud	6

n bit: 8

stop bit: 1

parity bit: no

serial port: set up by WinChorus++ (v 6.43 and following) through the hardware parameter *Miscellanea \ Hosting serial port* between the following values (default serial port is HOST):

Port	Parameter
HOST	1
EXT-SERV	2

Note:

bitrate setting affects *HOST* connector only.
EXT-SERV connector bitrate is fixed at 38400 baud.

Note:

HOST connector requires firmware version 2.09 rev10 or above.

Note:

Used byte order is *little-endian*: bytes are stored from the less significant to the most (e.g. a word field of value **258** is coded as **0x0201**).

4 SERIAL COMMUNICATION FRAME

Note:

variable characters are highlighted in bold inside each frame, frame-specific characters are highlighted in gray.

The frame structure used in the serial communication is as follows:

Sender \longrightarrow Receiver

<i>STX</i>	<i>N</i>	<i>CMD</i>	<i>D₁</i>	<i>D₂</i>	<i>...</i>	<i>D_{N-1}</i>	<i>CS</i>
------------	----------	------------	----------------------	----------------------	------------	------------------------	-----------

Field	Description	Value	Data type	Bytes
<i>STX</i>	Start of Text (frame start character)	0x02	byte	1
<i>N</i>	Frame lenght (without STX, N, CS character)		byte	1
<i>CMD</i>	Command code		byte	1
<i>D₁..D_{N-1}</i>	Eventual data characters		byte[]	N-1
<i>CS</i>	Frame check-sum (without STX, CS character), calculated as a XOR of N+1 frame characters		byte	1

Note:

frame value descriptions use the hexadecimal notation **0xGH**, which represents value **(G * 16) + H**, and is used to define non quantitative values.

Note:

string data types include a terminating character **0x00**.

Eg: string **text** is represented by the sequence **0x74 0x65 0x78 0x74 0x00**.

5 ENQUIRY

Before sending the sample list (see chapter 6) or session results (see chapter 7), the CHORUS checks Computer availability with the following frame:

CHORUS → Computer

STX	2	ENQ	InstrID	CS
-----	---	-----	---------	----

Field	Description	Value	Data type	Bytes
ENQ	Enquiry	0x05	byte	1
InstrID	CHORUS instrument identifier	0x43	byte	1

The Computer must answer with the following frame:

Computer → CHORUS

STX	1	EOT	CS
-----	---	-----	----

Field	Description	Value	Data type	Bytes
EOT	End of Text	0x04	byte	1

If, after sending an enquiry frame, the Computer doesn't answer, a timeout error message will be displayed.

Example:

CHORUS → Computer

0x02	0x02	0x05	0x43	0x44
------	------	------	------	------

Computer → CHORUS

0x02	0x01	0x04	0x05
------	------	------	------

6 SAMPLE PROGRAMMING

After receiving each sample code inserted by the user, the Computer sends the CHORUS the list of tests to be executed on that sample. This is called Job-List (J-List).

After establishing a connection between the Computer and the CHORUS (see chapter 5), the instrument sends, for each inserted sample, the following frame:

CHORUS → Computer

<i>STX</i>	<i>21</i>	<i>JListCmd</i>	<i>StorableRec</i>	<i>SampleCode</i>	<i>CS</i>
------------	-----------	-----------------	--------------------	-------------------	-----------

Field	Description	Value	Data type	Bytes
<i>JListCmd</i>	J-List record request command	0xD2	byte	1
<i>StorableRec</i>	Remaining test slots		byte	1
<i>SampleCode</i>	Sample code (including terminating character)		string	19

The Computer answers with the following frame:

Computer → CHORUS

<i>STX</i>	$(2 * K) + 21$	<i>EOT</i>	<i>SampleCode</i>	<i>PedFlag</i>	<i>TestID₁</i>	<i>TestID₂</i>	<i>...</i>	<i>TestID_K</i>	<i>CS</i>
------------	----------------	------------	-------------------	----------------	---------------------------	---------------------------	------------	---------------------------	-----------

Field	Description	Value	Data type	Bytes
<i>PedFlag</i>	Pediatric sample flag (available in firmware version 2.09 rev10 and above)	0..1	byte	1
<i>TestID₁..TestID_K</i>	Identifiers of tests to be executed on this sample	1..999	word[]	2 * K

Where K is the number of tests to be executed on the sample. If, for a certain sample, a single test has to be executed, the frame will contain only one *TestID* record.

Note:

A table must be created in the Computer of the enabled valid tests, such as Name test / ID Test, in order to be able to send to the instrument the identification of the tests to be performed.

Note:

The sample code handled by CHORUS must be of 15 characters or less. However, in order to handle also the quality controls (QC), the code must be large enough to contain 18 characters. Therefore the SampleCode field contains 19 characters (including trailing character).

Note:

Since the total number of storable tests of the CHORUS is limited, the Computer must not send a number of *TestID* records greater than *StorableRec*. There is therefore the possibility that not all the relative ID for a given sample can be sent, for reasons of space. In a subsequent work cycle, the Computer will start to send the data again from the point where it had stopped.

After receiving last sample data, the CHORUS sends the following frame:

CHORUS → Computer

STX	2	JlistEnd	InstrID	CS
-----	---	----------	---------	----

Field	Description	Value	Data type	Bytes
JlistEnd	J-List end command	0xD3	byte	1

The Computer closes the connection with the following frame:

Computer → CHORUS

STX	1	EOT	CS
-----	---	-----	----

Example:

no tests have to be executed on sample S1;

CMV-G (ID 1) has to be executed on sample S2;

CMV-G (ID 1) and RV-G (ID 2) have to be executed on sample S3.

CHORUS → Computer

0x02	0x15	0xD2	0x3A	0x53	0x31	0x00	0x00	0x00	0x00	0x00	0x00
0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x9F

Computer → CHORUS

0x02	0x15	0x04	0x53	0x31	0x00	0x00	0x00	0x00	0x00	0x00	0x00
0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x73

CHORUS → Computer

0x02	0x15	0xD2	0x3A	0x53	0x32	0x00	0x00	0x00	0x00	0x00	0x00
0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x9C

Computer → CHORUS

0x02	0x17	0x04	0x53	0x32	0x00	0x00	0x00	0x00	0x00	0x00	0x00
0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x01	0x00

CHORUS → Computer

0x02	0x15	0xD2	0x3A	0x53	0x33	0x00	0x00	0x00	0x00	0x00	0x00
0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x9D

Computer → CHORUS

0x02	0x19	0x04	0x53	0x33	0x00	0x00	0x00	0x00	0x00	0x00	0x00
0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x01	0x00	0x02

CHORUS → Computer

0x02	0x02	0xD3	0x43	0x92
------	------	------	------	------

Computer → CHORUS

0x02	0x01	0x04	0x05
------	------	------	------

7 RESULT MANAGEMENT

After each cycle, in the result window, or anytime from the session archive, the CHORUS can send the Computer sample results, in order to archive or elaborate them.

After establishing connection between the Computer and the CHORUS (see chapter 5), the instrument sends the following frame for each sample processed in this session:

CHORUS → Computer

STX	50	ResFrame	SampleCode	TestDesc	Report	Titre	MeasUnit	CS
-----	----	----------	------------	----------	--------	-------	----------	----

Field	Description	Value	Data type	Bytes
ResFrame	Result text sending command	0xD7	byte	1
TestDesc	Test description (including terminating character)		string	7
Report	Refertation (P = positive, N = negative, D = doubtful)		char	1
Titre	Titration (including terminating character)		string	12
MeasUnit	Measure unit (including terminating character)		string	10

The Computer answers with the following frame:

Computer → CHORUS

STX	1	EOT	CS
-----	---	-----	----

After sending all result frames, the CHORUS sends the following closing frame:

CHORUS → Computer

STX	2	ResEnd	InstrID	CS
-----	---	--------	---------	----

Field	Description	Value	Data type	Bytes
ResEnd	Result text sending end command	0xD8	byte	1

The Computer closes the connection with the following frame:

Computer → CHORUS

STX	1	EOT	CS
-----	---	-----	----

Example:

CMV-G test (with negative refertation and titre 0.1 IU/ml) has been executed on sample S2;
 CMV-G (with negative refertation and titre 0.2 IU/ml) and RV-G (with positive refertation and titre 20.0 IU/ml) tests have been executed on sample S3.

CHORUS → Computer

0x02	0x32	0xD7	0x53	0x32	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x43	0x4D	0x56	0x2D	0x47	0x00
0x00	0x4E	0x30	0x2E	0x31	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
0x49	0x55	0x2F	0x6D	0x6C	0x00	0x00	0x00	0x00	0x00	0x00	0xE5		

Computer → CHORUS

0x02	0x01	0x04	0x05
------	------	------	------

CHORUS → Computer

0x02	0x32	0xD7	0x53	0x33	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x43	0x4D	0x56	0x2D	0x47	0x00
0x00	0x4E	0x30	0x2E	0x32	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
0x49	0x55	0x2F	0x6D	0x6C	0x00	0x00	0x00	0x00	0x00	0x00	0xE7		

Computer → CHORUS

0x02	0x01	0x04	0x05
------	------	------	------

CHORUS → Computer

0x02	0x32	0xD7	0x53	0x33	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x52	0x56	0x2D	0x47	0x00	0x00
0x00	0x50	0x32	0x30	0x2E	0x30	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
0x49	0x55	0x2F	0x6D	0x6C	0x00	0x00	0x00	0x00	0x00	0x00	0x95		

Computer → CHORUS

0x02	0x01	0x04	0x05
------	------	------	------

CHORUS → Computer

0x02	0x02	0xD8	0x43	0x99
------	------	------	------	------

Computer → CHORUS

0x02	0x01	0x04	0x05
------	------	------	------

8 TEST APPLICATION

The test application attached to this documentation will let you verify the correct behaviour of either a Chorus or a Computer.

Chorus check procedure:

- switch on the Chorus;
- ensure the C-List is not full and contains at least one sample;
- ensure the archive contains at least one session;
- connect one serial cable endpoint to HOST or EXT-SERV connector on the Chorus (see chapter 3);
- connect the other serial cable endpoint to the PC that will be used to run the test application;
- run the test application and select *Chorus* in *Target* field;
- specify in *Serial port* field the address of the serial port used to connect one serial cable endpoint;
- select Chorus connection speed in *Serial speed* field (see chapter 3);
- configure fields in *Sample Programming* group as needed;
- select *Begin test* and follow instructions displayed on status bar.

The text area at the bottom of the application will show any information received from the Chorus during the test. When test is finished an error message will be displayed in case communication with the Chorus could not be established or anomalies have been detected in received data. You may repeat the test, eventually changing one or more fields, selecting *Restart test*.

Computer check procedure:

- properly configure the Computer and ensure it is ready to accept connections;
- connect one serial cable endpoint to the Computer;
- connect the other serial cable endpoint to the PC that will be used to run the test application;
- run the test application and select *Host* in *Target* field;
- specify in *Serial port* field the address of the serial port used to connect one serial cable endpoint;
- select Computer connection speed in *Serial speed* field;
- configure fields in *Sample Info* and *Sample Result* groups as needed;
- select *Begin test*.

The text area at the bottom of the application will show any information received from the Computer during the test. When test is finished an error message will be displayed in case communication with the Computer could not be established or anomalies have been detected in received data. You may repeat the test, eventually changing one or more fields, selecting *Restart test*.

Application sources are also available, written in C++ language, as well as project files, created with Turbo C++ (<http://www.turboexplorer.com/cpp>) development environment.