

Technical Specification Nr:	X00803.DOC
Device Name:	JRC SIO PROTOCOL
Old Device Name:	----
Device Type:	----
Revision level:	1.2
Last change:	04.02.2000
Creation Date:	20.02.1998
Author:	Dieter Prifling

1 Description

Specification of the serial I/O protocol of the **JUMPtec**[®] Remote Control (JRC) feature.

Revision History

Rev.	Date	Subject	Changed by
1.0	20.02.98	Initial release based on JRCINT.DOC version 1.6 / 1998.02.19 by DP	H. Eß
1.0	11.03.98	Command acknowledge timeout period corrected to 3 seconds.	H. Eß
1.1	30.07.98	Delayed Ack can be extended, flash write with verify	DP
1.2	04.02.2000	Added Board Info command 0x1b	DP

Hardware Layer

The communication between the client and the host utilizes the machine's serial ports with these parameters:

- 3-wire RX-TX connection
- 8 bits
- no parity
- 1 stop bit
- baud rates between 1200 and 115200 bauds

Connect Phase

In order to initiate the host mode the host will continually transmit these two bytes with 1200 baud followed by approximately a 40 ms delay:

CA	CB
----	----

CA always 0xCF
CB always 0xD0

During the boot phase the client will check all up to four COM ports that have been detected by the BIOS for the presence of these two characters. The client polls each port about 110 ms for the presence of any character. If it detects characters it will try to read in the two connect bytes 3 times in a row. If this is successful the client will then answer at 1200 baud with a standard *Maximum Baud Rate* reply (See below for details). This contains the maximum baud rate that the client is capable of.

After this the client will enter a loop that polls for Generic Command Blocks.

In the current implementation the host will always send a *Set Baud Rate* command to increase the baud rate to the largest common baud rate. Note that the client will send a *Positive Ack* reply with the previous baud rate. After that acknowledge both will switch to that baud rate. The client will also turn on any FIFO buffering at this time.

The current implementation of the host software will always issue a *Version* command followed by a *Query Need Password* and an optional number of *Check Password* commands.

The host then sends a *Resume* command with a parameter zero. This causes the client to send a *Delayed Ack* response and continue with further system initialization. After that the client sends a regular *Positive Ack* and enters another loop that polls for Generic Command Blocks.

The host machine will save the COM port number and selected baud rate to the jrc.ini file and return to the Command Prompt to await further commands.

This concludes the connect phase and leaves the connection open for further Generic Command Blocks. The host will control the protocol on the serial port.

If the connect is initiated by a *server* command the Resume command will have its parameter set to 2 which causes the client to hook some BIOS interrupts and continue system initialization without entering a second poll loop.

In this scenario the client will always issue a *Set Cursor* command and then a *Query Disk Change* command. The host can acknowledge the later positively if remote disk drive emulation is desired. The client will control the protocol on the serial port.

Command Blocks

Generic Command Block

Almost all data between the client and the host is sent in block with the following byte structure:

ST	LL	LH	CD	D0	..	Dn	C1	C2	ED
----	----	----	----	----	----	----	----	----	----

ST always 0xD1
LL length low byte (D0..Dn)
LH length high byte
CD command byte
D0..Dn zero or more command specific bytes
C1 Fletcher checksum (LL..Dn)
C2 Fletcher checksum
ED always 0xD2

Each of this command blocks must be acknowledged by a response block from the opposite side within 3 seconds. If no response is received the command is retransmitted up to 3 times.

Display Character Shortcut Command Block

This block is only sent from the client to the host as a shortcut for the *Display Character* command:

DC	cc
----	----

DC always 0xD3
cc character to display

Alert and Poll Command Command Block

Depending on the type of connect (Host Mode or Image Mode) either the client or the host will control the serial protocol. Only the controlling party is allowed to issue commands at random. If the other party wishes to send a command block it must send an *Alert* single byte command that notifies the controller of this need. The *Alert* command should be repeated every 500 ms. The controller will then send a *Poll Command* single byte command at its earliest convenience to give the other party a chance to issue a generic command.

This behavior is necessary to prevent each party from sending a command at almost the same time which is very likely to lead to time outs while each party keeps waiting for a reply.

In the current implementation this method is used while in host mode by the host to send *Key Event* commands to the client.

Alert command:

AL

AL always 0xD4

Poll Command command:

PC

PC always 0xD5

Defined Commands and Replies

Overview

- 00 Negative Ack
 - [1] error# (0:tx error, resend)
- 01 Positive Ack
- 02 Maximum Baud Rate (Connect)
 - [4] version, 0, 0, baudrate/1200
- 03 Set Baud Rate
 - [1] baudrate/1200
 - reply is sent with old baud rate
- 04 Resume (Disconnect)
 - [1] mode (0: early loop, 1: regular loop;; 2: server resume)
- 05 Reboot
- 06 Query Mem Size
 - [1] mem#
 - > OK with length [4]
- 07 Read Mem
 - [5] mem#, offset[2], count[2]
 - > OK with count bytes
- 08 Read Mem Reply (*obsolete*)
- 09 Write Mem
 - [5+] mem#, offset[2], count[2], bytes
- 0a Query Disk Size
 - [2] drive#, type
 - > OK with number of sectors [4], [[2] secs/track, [2] heads, [2] cyls, [drive#]]
 - the optional parameters are used for disk emulation.
- 0b Read Disk Sector
 - [7] drive#, type, sect#[4], number of sects[1]
 - > OK with bytes
- 0c Read Disk Sector Reply (*obsolete*)
- 0d Write Disk Sector
 - [7+] drive#, type, sect#[4], number of sects[1], bytes
- 0e Key Event
 - [2] ascii, scan
- 0f Display Chars
 - [1+] attrib, characters
- 10 Set Cursor
 - [3] x, y, show
- 11 Clear Screen
- 12 Delayed Ack
 - [1] number of seconds for delay
- 13 Scroll Window
 - [7] lines [2], attrib, x0, y0, x1, y1
- 14 Format Disk Sector
 - [7] drive#, type, sect#[4], number of sects[1]
- 15 Disk Changed
 - [2] drive#, type
- 16 Version
 - > OK with BIOS ver[4], JRC ver[4], Protocol Ver [4]
- 17 Query Disk Change
 - [2] drive#, type
 - > OK with change flag [1]
- 18 Query Protected Sectors
 - [2] drive#, type
 - > OK with starting sector [4], number of sectors [4]

19 Query Need Password

-> OK with need pwd [1]

1a Check Password

[+] password

1b Board Info

-> [124+] JIDABOARDINFO structure (implicit size, optional class list)

Command	H->C Cmd	C->H Reply	C->H Cmd	H->C Reply
00 Negative Ack		I		H
01 Positive Ack		I		H
02 Maximum Baud Rate (Connect)		I		
03 Set Baud Rate	I/H			
04 Resume (Disconnect)	I/H			
05 Reboot	I		H	
06 Query Mem Size	I			
07 Read Mem	I			
09 Write Mem	I			
0a Query Disk Size	I		H	
0b Read Disk Sector	I		H	
0d Write Disk Sector	I		H	
0e Key Event	H			
0f Display Chars			H	
10 Set Cursor			H	
11 Clear Screen			H	
12 Delayed Ack		I		H
13 Scroll Window			H	
14 Format Disk Sector	I			
15 Disk Changed	H			
16 Version	I/H			
17 Query Disk Change			H	
18 Query Protected Sectors	I			
19 Query Need Password	I/H			
1a Check Password	I/H			
1b Board Info	I/H			

H->C: Host to Client transmission

C->H: Client to Host transmission

Cmd: Command

I: Used in Image Mode

H: Used in Host Mode

0x00 - Negative Ack

Parameters: 1 bytes

BYTE error#

0: Transmission error, please resend command

1: Generic Error

Returns:

Description:

The previously sent command was not received properly or was not executed due to an error.

0x01 - Positive Ack

Parameters: 0+ bytes

Returns:

Description:

The requested command was completed successfully. Any number of bytes can be returned depending on the issued command. Usually the bytes are the result of an read operation.

0x02 - Maximum Baud Rate (Connect)

Parameters: 4 bytes

BYTE version number

0 at the moment

BYTE 0

BYTE 0

BYTE baudrate/1200

Example: 8 equals 9600 baud

Returns:

Description:

This is sent as an positive acknowledge by the client to the host.

0x03 - Set Baud Rate

Parameters: 1 bytes

BYTE baudrate/1200

Returns:

Description:

The host instructs the client to change the current baud rate. The OK reply is sent with old baud rate. This is usually the first command sent by the host to the client.

0x04 - Resume (Disconnect)

Parameters: 1 bytes

BYTE mode

0: early loop

1: regular loop

2: server resume

Returns:

Description:

The client drops out of it command dispatch loop and continues with the current boot process.

0x05 - Reboot

Parameters: 0 bytes

Returns:**Description:**

The client reboots, i.e. start a new boot process.

0x06 - Query Mem Size

Parameters: 1 bytes

BYTE mem#

0: CMOS

1: EEPROM CMOS mirror

2: NVRAM of JIDA board #1

n: NVRAM of JIDA board #n-1

Returns:

Positive Ack containing DWORD with length

Description:

Queries the total size of the given memory.

0x07 - Read Mem

Parameters: 5 bytes

BYTE mem#

0: CMOS

1: EEPROM CMOS mirror

2: NVRAM of JIDA board #1

n: NVRAM of JIDA board #n-1

WORD offset

WORD count

Returns:

Positive Ack containing count bytes

Description:

Read the specified memory.

0x09 - Write Mem

Parameters: 5+ bytes

BYTE mem#

0: CMOS

1: EEPROM CMOS mirror

2: NVRAM of JIDA board #1

n: NVRAM of JIDA board #n-1

WORD offset

WORD count

bytes

Returns:**Description:**

Write the specified memory.

0x0a - Query Disk Size

Parameters: 2 bytes

BYTE drive#

0: Floppy A:

1: Floppy B:

0x80: Hard Drive (Usually C:)

0x81: Hard Drive (Probably D:)

BYTE type

0: drive number specifies INT 13h drive number

0xff: Full Flash Memory

Returns:*Positive Ack* containing DWORD with number of sectors

For disk emulation, the host will provide the following information to simplify the client code:

WORD secs/track

WORD heads

WORD cyls

BYTE drive#

The drive is optional again and allows the host to override the clients suggested drive number

Description:

Queries the number of sectors of the specified drive

0x0b - Read Disk Sector

Parameters: 7 bytes

BYTE drive#

0: Floppy A:

1: Floppy B:

0x80: Hard Drive (Usually C:)

0x81: Hard Drive (Probably D:)

BYTE type

0: drive number specifies INT 13h drive number

0xff: Full Flash Memory

DWORD sect#

logical sector number starting with 0.

BYTE number of sectors

must be 1 at the moment.

Returns:*Positive Ack* containing sector bytes**Description:**

Read the given sector.

0x0d - Write Disk Sector

Parameters: 7+ bytes

BYTE drive#

0: Floppy A:

1: Floppy B:

0x80: Hard Drive (Usually C:)

0x81: Hard Drive (Probably D:)

BYTE type

0: drive number specifies INT 13h drive number
0xfe: Full Flash Memory with verify
0xff: Full Flash Memory
DWORD sect#
logical sector number starting with 0.
BYTE number of sects
must be 1 at the moment.
bytes

Returns:**Description:**

Write the given sector to the drive.

0x0e - Key Event

Parameters: 2 bytes
BYTE ASCII code
BYTE scan code

Returns:**Description:**

The passed key has been pressed on the host.

0x0f - Display Chars

Parameters: 1+ bytes
BYTE color attribute for all characters
BYTE characters

Returns:**Description:**

Display the passed characters on the host.

0x10 - Set Cursor

Parameters: 3 bytes
BYTE x
0 based cursor position
BYTE y
0 based cursor position
BYTE show
1 shows the cursor, 0 hides the cursor.

Returns:**Description:**

Set the cursor on the client, positions are 0 based.

0x11 - Clear Screen

Parameters: 0 bytes

Returns:

Description:

Clear the host screen.

0x12 - Delayed Ack

Parameters: 1 bytes

BYTE delay in seconds

Returns:

Description:

The requested command was received successfully. However execution of the command might take longer. The parameter specifies an additional number of seconds that the sender needs to wait for the completion of the command. A regular *Positive Ack* or *Negative Ack* will be sent after the execution of the command. *Delayed Ack* responses can be repeated to increase the delay further.

0x13 - Scroll Window

Parameters: 1 bytes

WORD lines

=0 clear the window

>0 scroll up (last line will be empty)

<0 scroll down (first line will be empty)

BYTE attribute

BYTE x0

BYTE y0

BYTE x1

BYTE y1

Returns:

Description:

Scrolls or clears the requested window in 0 based coordinates.

0x14 - Format Disk

Parameters: 7 bytes

BYTE drive#

must be 0

BYTE type

0xff: Full Flash Memory

DWORD sect#

logical sector number starting with 0.

BYTE number of sects

must be 128 at the moment.

Returns:

Description:

This is implemented for the flash disk only at the moment. The number of sectors must be 128 and sector number a multiple of 128.

0x15 - Disk Changed

Parameters: 2 bytes

BYTE drive#

0: Floppy A:

1: Floppy B:

0x80: Hard Drive (Usually C:)

0x81: Hard Drive (Probably D:)

BYTE type

0: drive number specifies INT 13h drive number

Returns:**Description:**

The emulated disk in the drive has changed.

0x16 - Version

Parameters: 0 bytes**Returns:**

DWORD BIOS Version

DWORD JRC Version

DWORD Protocol Version

Description:

Retrieves the version numbers of the above client components. The LSB (i.e. first byte) of each contains the major version number, then the minor number and two more bytes for sub minor version information.

0x17 - Query Disk Change

Parameters: 2 bytes

BYTE drive#

0: Floppy A:

1: Floppy B:

0x80: Hard Drive (Usually C:)

0x81: Hard Drive (Probably D:)

BYTE type

0: drive number specifies INT 13h drive number

Returns:

BYTE changeflag

0: disk not changed

1: disk has changed

0xff: not disk in drive

Description:

Retrieves the disk change status.

0x18 - Query Protected Sectors

Parameters: 0 bytes

BYTE drive#

must be 0

BYTE type

0xff: Flash Memory

Returns:

DWORD startsector

DWORD sectorcount

Description:

Retrieves the number of sectors which are protected on the client. Currently this is used only for the flash memory. This is the protected area and usually contains the BIOS. The host uses this information to prevent accidental overwriting of the BIOS on the client. The host command flashread, flashwrite and flasherase require a switch /bios to allow the host to write to these areas. Please note that with this switch the sector numbering is different: block 0 contains the system ROM, block 1 the extension, the flash disk usually starts at block 2. Without this switch the flash disk starts with block 0.

0x19 - Query Need Password

Parameters: 0 bytes

Returns:

BYTE flag

0: no password needed

1: need password

Description:

The host queries the client if a password verification is required.

0x1a - Check Password

Parameters: n bytes

BYTES password

Returns:

Description:

The client should check the given password and return a positive ack if it is acceptable.

0x1b – Board Info

Parameters: 0 bytes

Returns: 124+ bytes (JIDABOARDINFO structure)

DWORD dwFlags

Reserved. Always 0.

BYTE[16] szPrimaryClass

Primary class name. Usually "CPU".

BYTE[16] szBoard

BYTE[16] szBoardSub

Subname of the board or empty. Usually the same as the board name.

BYTE[16] szManufacturer
Manufacturer name. Usually "JUMP".

WORD[8] stManufacturingDate (DOS date in the 3rd WORD)

WORD[8] stLastRepairDate (DOS date in the 3rd WORD)
Date that the system was last repaired or refurbished. Valid only if later then the manufacturing date.

BYTE[16] szSerialNumber
10 character unique serial number of the board.

WORD wHardwareRevision
Hardware revision number. Major number in upper byte. Minor number in lower byte.

WORD wFirmwareRevision
Firmware revision number. Major number in upper byte. Minor number in lower byte.

WORD wJidaRevision
BIOS Jida interface revision number. Major number in upper byte. Minor number in lower byte.

WORD wFeatureNumber
Feature Number of the BIOS or the driver. Usually 0 or 1.

Description:

Returns the core of the JIDABOARDINFO structure as defined in the JIDA Win32 API specification.