RespeQt protocols

Jochen Schäfer (JoSch)

June 2019

Contents

Part I
Introduction

This article documents the protocols user by RespeQt and SpartaDos X to communicate with each other.

Part II
RespeQt Client (RCl)

1 Introduction

RCl always uses the device id 0x46 (P7: ?) to communicate.

2 Command overview

Cmd	Name	Parameters	Return value
0x93	Send datetime	None	Date and Time
0x94	Swap disks	Aux1 = slot#, Aux2 = slot#	Success or failure
0x95	Unmount disk	Aux1 = slot#	Success or failure
0x96	Mount disk	Aux1 = slot#	Success or failure
0x97	Create and mount image	cf. description	Success or failure
0x98	Toggle auto commit	None	Success or failure

3 Details

This section details all the commands used in RCl

3.1 0x93 - Send date and time

This command requests the date and time from the server. The client then sets the date and time on the Atari.

Parameters None

Return value The server returns a dataframe containing 6 bytes with the recent date and time as follows:

Index	Size	Meaning
0	Byte	Day of month
1	Byte	Month
2	Byte	Year without century
3	Byte	Hours
4	Byte	Minutes
5	Byte	Seconds

3.2 0x94 - Swap disk images

This command swaps two images in their places

Parameters Aux 1 and Aux 2 contain the respective slot indices of the disk images to swap. Their value is always between 1 and 9 or between 26 and 31.

Return value If one of the image indices is out of range, the server sends a NAK. If the operation is successful, the server sends an ACK.

3.3 0x95 - Umount disk images

This command unmount one or all disk images

Parameters Aux 1 contains the slot index of the disk image to be unmounted. The value is always between 1 and 9 or between 26 and 31 or is -6 / 249. If the slot index is -6 all disk images are unmounted.

Return value If the slot index is out of range, the server sends a NAK. If the specified image or in case of index -6 any image has to be saved, then nothing gets unmounted and the server sends a NAK. If the operation is successful, then the server sends an ACK.

3.4 0x96 - Mount disk image

This command mounts a disk image in the next free slot.

Parameters If Aux 1 is 0, the server mounts the image with the file name sent in the following data frame in the next free slot. If Aux 1 is not 0, then the slot index of the last mount is requested.

Return value If Aux 1 is 0, the server sends a NAK, if the mount was not possible, otherwise the server sends an ACK. The filename has to be in 8.3 format. If Aux 2 is not 0, the server returns an ACK, followed by a data frame of size 1 Byte, containing the slot index.

3.5 0x97 - Create and mount disk image

This command creates a disk image as specified and then mounts it in the next free slot

Parameters If Aux 1 is 0, the server mounts the image with the file name sent in the following data frame in the next free slot. The filename has to be in 8.3 format. The requested image format is appended to the filename by '.' and the type byte. If Aux 1 is not 0, then the slot index of the last mount is requested.

Type number	Disk format
1	SSSD
2	SSES
3	SSDD
4	DSDD
5	DDHD
6	QDHD

Return value If Aux 1 is 0, the server sends a NAK, if the mount was not possible, otherwise the server sends an ACK. The filename has to be in 8.3 format. If Aux 2 is not 0, the server returns an ACK, followed by a data frame of size 1 Byte, containing the slot index.

3.6 0x98 - Toggle auto commit

This command toggles the auto commit flag.

Parameters Aux 1 contains the slot index of the auto commit flag to be toggled. The value is always between 1 and 9 or between 26 and 31 or is -6 / 249. If the slot index is -6 the auto commit flag of all slots are toggled.

Return value If the slot index is out of range, the server sends a NAK. If the operation is successful, then the server sends an ACK.

4 Proposals

This sections details proposals for new commands to be used in RCl.

Part III
PCLINK

5 Introduction

PCLINK always uses the device id 0x6F (XXX: ?) to communicate. The protocol used by PCLINK is DOS2DOS (cf *BIBREF to DOS2DOS website*)

6 Command overview

Cmd	Name	Parameters	Return value
0x00	FREAD		data frame with read data
0x01	FWRITE		success or failure
0x02	FSEEK		success or failure
0x03	FTELL		file position
0x04	FLEN		file length
0x06	FNEXT		
0x07	FCLOSE		success or failure
0x08	INIT		success or failure
0x09	FOPEN		success or failure
0x0A	FFIRST		first directory entry
0x0B	RENAME		success or failure
0x0C	REMOVE		success or failure
0x0D	CHMOD		success or failure
0x0E	MKDIR		success or failure
0x0F	RMDIR		success or failure
0x10	CHDIR		success or failure
0x11	GETCWD		current working directory
0x13	DFREE		success or failure
0x14	CHVOL		success or failure

7 Protocol sequence

On the SIO protocol layer, the commands as described above are wrapped into the following commands:

Cmd	Name	Parameters	Return value
? 0x3F	Get Hi-Speed	None	Hi-Speed setting
P 0x50	Begin subcommand	Aux1 = Buffer size, Aux 2 = protocol version, device #	Success or failure
R 0x52	End subcommand	Aux1 = Buffer size, Aux 2 = protocol version, device #	Success or failure
S 0x53	Get Status	None	Status

Every command begins with sending a P with the destination unit number (0-15) in the lower 4 bits of AUX2. Upon command ACK the client sends a data frame, which contains the command byte and its respective parameters (cf 8.1). The size of this data frame has to match the value sent in AUX 1 of the P command, because not all commands are using the complete parameter structure. If an error occurs, the server records the error.

The client then sends a status request to the server to get the status of the addressed device. The server also sends a possible correction of the parameter structure size for adoption by the client. If errors occured, the client is able to abort the operation.

The client ends the sub command by sending an R to the addressed device.

If a P is send to a device before the preceding sub command is ended with R, the server is sending an command NAK.

8 Data structures

8.1 Parameter buffer

All sub commands use the following structure to submit the parameters to their respective sub command. The client chooses how much of the structure is needed by supplying a structure size. The server can correct the client on the structure size needed for the operation.

name	type	function
fno	byte	sub command number
handle	byte	handle to opened file or directory
f1, f2, f3, f4, f5, f6	byte	general purpose bytes
fmode	byte	Open file mode, cf 8.4
fatr1	byte	Search file mask, cf 8.2
fatr2	byte	Transferred file attributes, cf 8.3
name	string of length 12	File or directory name
names	string of length 12	File or directory name
path	string of length 65	Path to file or directory

8.2 fatr1

This bit mask selects the attributes, when files are searched. Value 0x00 selects all files.

Value	Name	Meaning
0x01	RA_PROTECT	Protected files
0x02	RA_HIDDEN	Hidden files
0x04	RA_ARCHIVED	Files with the Archive attribute set
0x08	RA_SUBDIR	Sub directories
0x10	RA_NO_PROTECT	Not protected files
0x20	RA_NO_HIDDEN	Not hidden files
0x40	RA_NO_ARCHIVED	File without the Archive attribute set
0x80	RA_NO_SUBDIR	Not sub directories

8.3 fatr2

This bit mask repesent the transferred file attributes to be set.

Value	Name	Meaning
0x01	SA_PROTECT	Protect file
0x02	SA_HIDE	Hide file
0x04	SA_ARCHIVE	Set Archive flag on file
0x10	SA_UNPROTECT	Unprotect file
0x20	SA_UNHIDE	Unhide file
0x40	RA_UNARCHIVE	Remove Archive flag on file

8.4 fmode

Value	Meaning
0x04	Read
0x08	Write [1]
0x09	Append
0x0C	Read and Write

9 Details

9.1 0x00 - FREAD Read from file or directory

This sub command reads from a file or directory, which was opened for the requested unit before. If no file was opened before, the server answers with a command NAK. If the opened handle is to a directory, the directory is read (). If the read was successful, the server answers with a command ACK.

direcotry read means what?

[1] Obviously, 0x09 really uses Bit 0, which is used as Append flag. Should we only record the value making sense or the bit masks? 0x0D is plausible for Read and Write with appending

Parameters Parameter buffer, cf. 8.1

Return value

9.2 0x01 - FWRITE Write to file

This sub command writes to a file or directory, which was opened for the requested unit before. If no file or directory was opened before, the server answers with a command NAK. If the opened handle is to a directory, the operation is ignored. If the write was successful, the server answers with a command ACK.

Parameters Parameter buffer, cf. 8.1

Return value

9.3 0x02 - FSEEK Seek to position in file

This sub command sets the position of a file or directory, which was opened for the requested unit before. If no file or directory was opened before, the server answers with a command NAK. If setting the position was successful, the server answers with a command ACK.

Parameters Parameter buffer, cf. 8.1

Return value

9.4 0x03 - FTELL Return the position in file

This sub command returns the current position of the handle to a file or directory. If no file or directory was opened before, the server answers with a command NAK. If setting the position was successful, the server answers with a command ACK.

Parameters Parameter buffer, cf. 8.1

Return value The position into the file or directory is returned in 3 bytes, starting with the lowest bits first. better wording

9.5 0x04 - FLEN Return file length

This sub command returns the current size of the handle to a file or directory. If no file or directory was opened before, the server answers with a command NAK. If setting the position was successful, the server answers with a command ACK.

Parameters Parameter buffer, cf. 8.1

Return value The position into the file or directory is returned in 3 bytes, starting with the lowest bits first. () better wording

9.6 0x06 - FNEXT Read next directory entry

This sub command returns the next directory entry. If no file or directory was opened before, the server answers with a command NAK. If setting the position was successful, the server answers with a command ACK.

Parameters Parameter buffer, cf. 8.1

Return value The server sends back a data frame beginning with the handle (1 byte), followed by the directory entry (25 bytes).

9.7 0x07 - FCLOSE Close the file handle

Parameters Parameter buffer, cf. 8.1

Return value

9.8 0x08 - INIT Initialize unit data structures

Parameters Parameter buffer, cf. 8.1

Return value

9.9 0x09 - FOPEN Open file handle

Parameters Parameter buffer, cf. 8.1

Return value

9.10 0x0A - FFIRST Read first directory entry

Parameters Parameter buffer, cf. 8.1

Return value

9.11 0x0B - RENAME Rename file or directory

Parameters Parameter buffer, cf. 8.1

Return value

9.12 0x0C - REMOVE Remove file

Parameters Parameter buffer, cf. 8.1

Return value

9.13 0x0D - CHMOD Change file permissions

Parameters Parameter buffer, cf. 8.1

Return value

9.14 0x0E - MKDIR Create a directory

Parameters Parameter buffer, cf. 8.1

Return value

9.15 0x0F - RMDIR Remove directory

Parameters Parameter buffer, cf. 8.1

Return value

9.16 0x10 - CHDIR Change the current working directory

Parameters Parameter buffer, cf. 8.1

Return value

9.17 0x11 - GETCWD Return the current working directory

Parameters Parameter buffer, cf. 8.1

Return value

9.18 0x13 - DFREE Return information about unit

Parameters Parameter buffer, cf. 8.1

Return value

9.19 0x14 - CHVOL Change the current volume / drive

Parameters Parameter buffer, cf. 8.1

Return value

Part IV
Smartdevice

10 Introduction

Smartdevice always uses the device id 0x45 (P6: ?)

11 Command overview

Cmd	Name	Parameters	Return value
0x55	Submit URL	AUX = URL length, data frame with URL	Success or failure
0x93	Send datetime	None	Date and Time

12 Details

12.1 0x55 - Submit URL

This command submits a URL for opening by the host systems default application for the URL's scheme.

Parameters AUX 1 contains the LSB of the URL length, while AUX 2 contains the MSB of the URL length. The client sends a data frame containing the URL upon command ACK.

Return value If the URL is longer than 2000 bytes, the server answers with a command NAK. Otherwise, the server answers with a command ACK.

12.2 0x93 - Send date and time

This command requests the date and time from the server. The client then sets the date and time on the Atari.

Parameters None

Return value The server returns a dataframe containing 6 bytes with the recent date and time as follows:

Index	Size	Meaning
0	Byte	Day of month
1	Byte	Month
2	Byte	Year without century
3	Byte	Hours
4	Byte	Minutes
5	Byte	Seconds