R8C ROM Monitor interface V1.0

Document ID:	TSW0191-006
Status:	Released
Version:	1.0
Date:	9 Oct 2003

ALTIUM BV

1 Commands

This document describes the R8C ROM monitor interface v1.0.

Characters send and received are by default in hexadecimal format, and hence limited to the range `[0-9a-fA-F]`. Only the download and dump commands may contain binary data in the full character range `[0,256>`, with the exception of the following values:

```
XOFF = 0x11
XON  = 0x13
ESC  = 0x1B
!    = 0x21
```

These characters shall be escaped using the escape character (0x1B) followed by their value incremented with the constant value 0x40, e.g. the binary value 0x21 results in the sequence 0x1B, 0x61.

Table 1.1 lists all monitor commands.

CMD_CONF	(00) Read configuration	
Response	`(0V<nr>+.<nr>+<ID-string>?[\n<setting>]*): Ok` `(8V<nr>+.<nr>+<ID-string>?[\n<setting>]*): Syntax error`	
CMD_REST	(01) Reset CPU	

Response	`(0V?.?<ID>[\n<setting>]*): Ok`
	`(4): Program is running`
	`(8): Syntax error`	

CMD_ERASE	(02bb) Erase/Download mode bb = blocks to erase, each bit representing a single block.	

Response	`(0): Ok`
	`(4): Program is running`
	`(8): Syntax error`	

Erase specified blocks, then enters download mode, after which data can be downloaded using one or more download lines with the following syntax.		
	CMD_DWNH	.aaaalldd...ddcc Program flash aaaa = start address ll = number of data bytes (0 equals 256 bytes) dd...dd = binary data cc = checksum

	Response	`(0): Ok`
		`(2): Checksum error`
		`(4): Program is running`
		`(8): Syntax error`

	When a new line does not start with a '.' the monitor will leave download mode and returns to standard mode.	
CMD_BS	(03iiaaaa) Set breakpoint ii = breakpoint nr aaaa = break address	
Response	`(0): Ok` `(8): Syntax error`	

CMD_BC	(04ii) Clear breakpoint ii = breakpoint number
Response	(0): Ok (8): Syntax error
CMD_GO	(05) Go
Response	(0): Ok (4): Program is running (8): Syntax error
CMD_STAT	(06) Get program status
Response	(00): Ok, program running (01): Ok, program stopped (80): Syntax error, program running (81): Syntax error, program stopped
CMD_STOP	(07) Stop program
Response	(0aaaa): Ok, stopped at address 'aaaa' (8aaaa): Syntax error, stopped at address 'aaaa'
CMD_STEP	(08) Single step
Response	(0): Ok, did one single step (4): Program is running (8): Syntax error, no single step done
CMD_REG	(09) Show all registers
Response	(0[vvvv]+): Ok vvvv+ values of all registers concatenated (4): Program is running (8): Syntax error
CMD_REGI	(0Aii) Show register ii = register nr.
Response	(0vvvv): Ok vvvv register value (4): Program is running (8): Syntax error
CMD_REGSET	(0Biivvvv) Set register ii = register nr. vvvv = value
Response	(0): Ok (4): Program is running (8): Syntax error
CMD_DUMP	(0Caaaall) Dump from address aaaa = address ll = length (0 length equals 256 bytes)
Response	(0d*cc): Ok d*: binary data cc: checksum (8): Syntax error
CMD_FILL	(0Daaaavv) Fill at address aaaa = address vv = byte value
Response	(0): Ok (4): Program is running (8): Syntax error

Table 1.1: R8C monitor commands

Table 1.2 lists the register numbers that can be used in the CMD_REGI and CMD_REGSET commands.

Register	Number	Nibbles
R0	00	4
R1	01	4
R2	02	4
R3	03	4
A0	04	4
A1	05	4
SB	06	4
FB	07	4
PC	08	4
FLG	09	4
USP	0A	4
ISP	0B	4
INTB	0C	4

Table 1.2: Register numbers

1.1 Configuration command

The CMD_CONF configuration commands allows for returning target specific settings to the debugger. The version string directly after the status '0' character is obligatory, a following identification string and one or more setting strings are optional. This leads to the following syntax for the CMD_CONF response string:

```
(0V<digit>+.<digit>+[<string>]?[\n<string>=<string>]*])
```

Table 1.3 shows a list of possible options and values. All numerical values should be returned in hexadecimal format like other values returned by the monitor.

An example response string is shown here:

```
(0V3.0 TASKING ROM Monitor
TG=3DK-R8C11
CPU=R5F21114FP
RE=F000,1000)
```

Option	Value	Description
TG	<string>	Target board identification
CPU	<string>	CPU name
OCD	<val>	OCD startup value
CM0	<val>	CM0 startup value
CM1	<val>	CM1 startup value
PM0	<val>	PM0 startup value
RE	<start>,<size>	Area to be reserved
CO	<comment-string>	Any comment string

Table 1.3: Optional ROM Monitor settings