

**NAME**

`rxm` – DEC RX02 Floppy Disk System Emulator

**SYNOPSIS**

```
rxm [ -v ] [ -h ] [ -t RX01 | RX02 | RX03 ] [ -t DMA | PIO ] [ -x ] [ -0 filename ] [ -d SD | DD ] [ -1 filename ] [ -d SD | DD ]
```

**DESCRIPTION**

`rxm` emulates the operation of a Digital Equipment Corporation RX02 Floppy Disk System. The emulator requires an interface module to connecting a standard PC parallel port to one of several RX disk interface cards. Schematics for the interface are available from the author.

The emulator should present to the host computer a faithful representation of either an RX01 or an RX02 dual floppy drive system. At certain points in development it has been shown that it could operate with various OS/8 and RT-11 handlers and 2.11BSD drivers. The RX8E, RX28, RX11, RX211, RXV11, and RXV21 are known to have worked at one time during development, but have not all been tested with the latest version.

No experience exists with some systems that included RX interfaces other than the ones listed. These include the VT78, VT278 and DECmates.

`rxm` can be configured from the command line. After initiation, `rxm`'s attached files and file densities can be modified by issuing an interrupt signal from the keyboard, usually `^C`. This stops the emulation and presents a command prompt.

During normal execution of the emulator, it can be in one of two states: interruptable and uninterruptable. The emulator is uninterruptable when in the midst of a command execution. It becomes interruptable while it is waiting for the issuance of the next command or when the interface signal `RX_INIT` is held active.

When the emulator is interruptable, receipt of an interrupt signal ceases normal processing and presents a command prompt. When the emulator is uninterruptable, several interrupt signals in quick succession will cause the emulator to present a command prompt with a warning that it is in an uninterruptable state. This allows control to be regained when the host processor stops in the middle of an uninterruptable command sequence. Extreme caution should be exercised when at the command prompt when in the uninterruptable state because some actions could cause corruption of the disk images.

**OPTIONS**

- v** Provide additional information related to program operation. Multiple occurrences increase the amount to information.
- h** Print a brief summary of command options.
- t *type*** Specify the emulation type desired, where *type* is one of **RX01**, **RX02** or **RX03**.
- t *mode*** Specify the emulation transfer mode desired, where *mode* is one of **DMA** or **PIO**.
- 0 *filename***  
Use file *filename* as an image for drive unit 0. If the file does not exist, create it, and clear it. If the file exists, use the file size to determine the density to emulate. If the size is not appropriate, then a warning is issued and the emulator uses a cleared memory image. In this case, this image cannot be saved to disk.
- 1 *filename***  
Use file *filename* as an image for drive unit 1.
- d *density***  
Set the density of the immediately preceding unit. *density* is one of *SD* or *DD*. Used to indicate the size of an image file that is to be created or expanded. The option has no effect if the file already exists.
- x** Used to enable the expansion of an image file to the full size of an emulated disk. This option will only enlarge images and never truncate them.

## COMMAND PROMPT

The emulator can be stopped by an interrupt signal. When interrupted, a command prompt allows the emulator state to be modified, continued, or exited.

**c** Continue from where the emulator was stopped.

**0** [*fi lename* | .] [*SD* | *DD*] [*DMA* | *PIO*]

Change the file attached to emulated floppy drive 0. *fi lename* designates a new file to attach to the drive. If *fi lename* is a single ".", the currently attached file is detached and closed and the drive is made **Not Ready**. If *fi lename* refers to a file already present, the size is examined and the emulated density set appropriately. If the file does not exist, it is created and cleared. *SD* causes any created disk emulation to be single density. *DD* causes any created disk emulation to be double density. *DMA* causes transfers to occur using DMA. *PIO* causes transfers to occur using PIO. The transfer mode is fixed by the host interface in use and cannot be changed arbitrarily.

**1** [*fi lename* | .] [*SD* | *DD*] [*DMA* | *PIO*]

Change the file attached to emulated floppy drive 1.

**p** Print the contents of the emulated RX02 registers.

**s** Print the file attachment for the emulated floppy drives.

**v n** Change the verbosity level of debug messages where 0 is minimum verbosity and 3 is maximum. Setting this value too high can cause timing problems with the emulator resulting in protocol violations and loss of synchronization with the host.

**q** Quit, saving the files to disk.

## FILES

*rx0.dsk* The default file for drive unit 0.

*rx1.dsk* The default file for drive unit 1.

## NOTES

The interface schematic is available from the author. There has also been a third party that has made a PCB layout. This may or may not be generally available.

The machine running the emulator is responsible for bit-banging the serial data stream to and from the host computer. This can place a high load on that machine. For best results a dedicated machine is recommended. The author has had quite a bit of success with a 400Mhz Pentium II.

The RX01/02 serial protocol is paced by the RX01/02 drive. This relaxes some of the real time requirements of the emulator since they are dictated by the emulator. It has been found though, that some handlers, especially boot loaders, violate the protocol in ways that are acceptable to real hardware but can break the emulator.

## BUGS

Sometimes when RT11 is booted from an emulated drive, it hangs. It seems that the boot code issues a disk read command followed by an empty buffer command for each sector that is to be read. Between the read and empty commands, the driver waits for the RX\_DONE bit. This bit seems to become TRUE before the RXV21 is ready to accept another command, and so the empty buffer command is sometimes lost, resulting in a hang.

The latest version has not been tested with all the possible hardware/software combinations.

## AUTHOR

Charles H. Dickman, chd@nktelco.net