Gemini Microcomputers Ltd

18 Woodside Road Amersham Bucks HP7 0BH England.

The Gemini MultiBoard Microsystem

RP/M V2.0

SOFTWARE MANUAL for GM811/GM813

GM811
GM813
Issue 1
06-07-82

STUDENT COPY 2.

ROM PROGRAM FOR MICROCOMPUTERS (RP/M) V2.0 JUNE 1982

RP/M is a resident operating system for microcomputers. The concept behind RP/M is that software should be compatiole between computers regardless of whether or not disks are used. RP/M therefore has been written to appear to application programs as much like the Digital Research CP/M disk operating system as possible. However the code is quite different and the commands are simple and do not resemble those provided by CP/M. It is supplied as a 4K byte ROM which resides at address F000H.

The success of RP/M is demonstrated by the software compatibility achieved. RP/M can for example support the Microsoft Disk Basic interpreter without a single change, and the interpreter can load and save Basic programs and sequential files on cassette, while believing that a disk is in use. The Digital Research ZSID dynamic debugging program is another example of CP/M software that can be run under RP/M. The only significant restrictions of RP/M are that random access files are not supported, only one sequential file may be open at a time, and that file must be read or written sequentially from the start. These restrictions are of course inevitable because of the nature of a cassette tape.

Some of the benefits of the design of RP/M are that:-

(a) all software which runs under RP/M will run under CP/M, so that conversion to a disk system is easy. The CP/M programs RC and WC read and write cassette files in RP/M format.

(b) much software is available which was written for CP/M but can be run under RP/M.

(c) software can be developed, tested and run under CP/M and placed in ROM or on tape for use on a system with RP/M and no disk.

Features of RP/M include:-

(a) the ability to boot a disk system, so that it is still needed if a disk system is added. The Gemini disk card with either Pertec compatible double density double sided drives or Micropolis compatible double density double tracking drives are supported.

(b) automatic disk boot on power-on or on reset, if a disk card is plugged in to the system.

(c) operation with either the GM811 CPU card or the GM813 CPU plus RAM card, automatically detecting which is in use.

(d) full support for the Gemini video card, providing special features such as full screen editing and the ability to print the contents of the screen.

(e) the ability to operate without a video card.

(f) support for serial printers, allowing for handshake, and parallel printers with a Centronics compatible interface.

RP/M also provides a comprehensive set of commands for:-

(a) booting a disk.

(b) reading and writing files to cassette.

(c) modifying and manipulating the contents of memory.

(d) operating the I/O ports.

(e) executing programs.

THE KEYBOARD

RP/M automatically supports keyboards attached to the computer in any of three ways. You can attach a Gemini ASCII encoded keyboard to the video card, and this is the normal configuration. In the case of the GM811 CPU card, you can attach a similar keyboard directly to this card as well. In addition you can attach a keyboard through the RS232 serial interface. In this case the data rate must be 300 bps, although this can be altered by use of the U command (see below).

Operation of the keyboard should be obvious, apart from a number of special features:-

(a) to generate a control character, hold down the Control key and then press the letter key required.

(b) to tabulate the input, enter the tab character, Control-I.

(c) to turn the printer on or off, type Control-P. It is essential to have a printer attached to the system before typing Control-P as otherwise the lack of a handshake signal will make the system lock up and you will have to press Reset.

(d) to pause the output display, type Control-S, then type Control-S again to continue, or type Control-C to stop the program.

(e) to stop a program, type Control-C as the first character on an input line. This restarts RP/M.

(f) input lines may be edited by using the Backspace key to move back, Control-X to backspace to the start of the line, Control-U to ignore the line entered, Control-R to retype the line being entered, and Control-E to give a physical line feed which is not input to the program.

(g) the Del key acts as a method of backspacing when using a terminal. The characters backspaced over are echoed.

(h) the Return key must be pressed to enter an input line. Line Feed has the same effect.

(i) Control-@ (Null) enters screen edit mode, described below. The cursor movement keys operate only in screen edit mode.

(j) Control-@ followed by Control-B activates a screen dump, which means that the current contents of the screen are printed. See below.

(k) to activate the Escape sequences described in the intelligent video card documentation, press Control-@, then ESC and then the required command character. This will work only for Escape sequences with one character sent after the Escape, and also should not be used for Escape sequences to which the video card replies.

RESETTING RP/M

When the system is switched on or the reset button is pressed, RP/M takes control. First it initialises the video card, which displays an identifying message at the top of the screen. Then it finds out how much memory is available by testing until it finds the end, and displays the size of memory and the header message. This process does not change the contents of memory, so that programs are not overwritten. Then RP/M initialises its workspace, the keyboard and the UART. If there is a disk card in the system it automatically attempts to boot a disk. If this fails then an appropriate message is output as from the B command. Then the "** RP/M ready **" message is output, followed by a line with an asterisk and a blinking cursor.

RP/M COMMANDS

RP/M displays an asterisk followed by a blinking cursor when it is waiting for a command. The commands available are described below, in several groups:-

- Disk command (B)
- Cassette commands (R, W, L)
- Memory commands (D, S ,C ,F, P)
- Port commands (U, O, Q)
- Execution commands (G, I)

Commands are carefully checked for many possible errors. Error messages are "No such command", "Too many/few values" and "What?". In the descriptions given below, values in square brackets "[]" are optional. Do not type in the brackets! Hex means hexadecimal (base 16), and hex values are shown followed by "H". When entering hex values, do not type in the "H". Values must be separated by "white space" which means one or more spaces, commas or tabs. Commands may be up to 126 characters long.

DISK COMMAND

B

Boot disk. This command executes the boot sector on the disk placed in drive A. When the sector has been read into memory and is about to be executed, the message "Executing boot" is output. If the command fails then an error message is output. "No disk" means that either there is no disk card in the system, or there is no disk in drive A, or the door is not closed, or the disk has been inserted the wrong way round, or the disk is not rotating, or there is a hardware failure. "Bad disk" means that the boot sector on the disk cannot be read. This is likely to be because the disk has not been formatted correctly for the drive used. "Wrong disk" means that the boot sector has been read in correctly but it has been found not to contain a valid boot program and is therefore not the correct disk. An attempt to boot a disk, whether successful or not, does not change the contents of the program area. This allows a disk to be booted and then a CP/M Save command to be issued, providing another means of moving data between an RP/M and a CP/M system.

CASSETTE COMMANDS

R filename

Read a cassette file with the filename specified. Filenames must be one to eight characters long, and only letters and numbers may be used. The message "Insert file: filename" is displayed, followed by "Start playing. Press return". Obey the instructions and then press the Return key. As each record is read, "*" is displayed. But if a bad record is read, "?" is displayed, and if a record has already been read "-" is displayed. If the file is too large for the amount of memory in the system, then "No memory" is displayed and the command is aborted. If a record is read which is from a later position in the file than the next record expected, then "Rewind and retry" is displayed. This is an instruction to rewind the tape to a position before the bad record, and then start playing again. If the wrong file is being read, then the message "Wrong file: filename" is displayed. Because a 16 bit checksum is included in each record, and because of the checking performed, it should be impossible to load bad data. Simply follow the instructions given on the screen. When the last record in the file has been read, "." is displayed followed by the messages "Remove file: filename" and "Press return". Stop the cassette recorder and then press the Return key.

W filename

Write a cassette file with the filename specified. Filenames must be one to eight characters long, and only letters and numbers may be used. The message "Insert file: filename" is displayed, followed by "Start recording. Press return". Obey the instructions and then press the Return key. As each record is written to the file, "*" is displayed. When the file has been written to tape, the messages "Remove file: filename" and "Press return" are displayed. Stop the cassette recorder and then press the Return key.

L [length]

Display or set the length of the program. The R command stores the length of the last program read in. This value is used by the W command to determine the size of the program to be written. Programs always start at address 0100H. To find the length of the current program, enter L by itself. The number of records in the program is displayed in hex. To set the length of the program enter the number of records in hex. Note that each record is 128 (80H) bytes long, so a 2K (2048 byte) program has 16 (10H) records.

MEMORY COMMANDS

D [start] [end]

Display the contents of memory, in hex and ASCII. Each line shows the address of the first byte, then the values of 16 bytes in hex, then the same values in ASCII, if they are printable. Start and end addresses may be specified. If no end address is entered, eight lines are output. If the command follows a previous D command, and no start address is entered, then the command displays the next block of memory. Remember that you can pause the output by using Control-S (see above). The command may be cancelled during output by typing a Space.

S start

Set or examine values in memory, starting at the "start" address entered. The address of the memory location is output, followed by the value at that location in hex, and then the ASCII equivalent, in quotes, if the value is printable. The command then waits for an input line. Press Return to step through subsequent locations. Alternatively, enter one or more values in hex, and these will be loaded into memory. Enter a double quote followed by an ASCII character, and that value will be loaded. Enter "-" to decrement the address. Enter "/" followed by an address, and the address will change to the value entered. All these options may be combined in a single line. To end the S command, end the input line with ".".

C from to length

Copy data from the "from" address to the "to" address, for "length" bytes. The copying process is "intelligent", which means that copying is always done so that no data is overwritten during copying.

F start end char

Fill memory from "start" to "end" with the value "char".

P [start] [page] [length]

Move a firmware package to address 0100H for execution. The firmware is taken from "start" address on page "page", and "length" bytes are moved. If no length is entered then 24K bytes are moved. The page must be 0, 1, 2 or 3, and if no page is entered then the data is read from page 0. If no start address is entered then the value C000H is used.

PORT COMMANDS

U [D or C] [speed]

Initialise the UART. On reset the UART is set to use the RS232 interface at 300 bps.* The U D command sets the UART to use the RS232 interface at the speed specified. If no speed is entered then the speed is left unchanged at its previous value. Speeds allowed are: 50, 75, 110, 134 (giving 134.5 bps), 150, 300, 600, 1200, 1800, 2000, 2400, 3600, 4800, 7200, 9600, 192F (giving 19200 bps), 384F (giving 38400 bps), and 560F (giving 56000 bps). The U C command sets the UART to use the cassette interface at 1200 bps. Since the cassette interface will operate correctly only at 1200 bps, no speed may be entered. The U C command is not normally used since the R and W commands and the RP/M system routines automatically switch to and from the cassette interface as needed.

O port value

Output a value to a port. Do not use this command unless you are confident that you understand the effect of your actions, as you could in theory destroy devices attached to the system such as disk controllers.

Q port

Input from a port and display the value received in hex.

EXECUTION COMMANDS

G [start]

Execute a routine or program at the start address specified. If no start address is entered, then the program at address 0100H is executed. This command should not be used to execute programs which expect a command string or file control block to be supplied.

I [command string]

Execute the program starting at address 0100H. This is the command used to execute RP/M programs. The default file control block is initialised with the filename, if one is entered. The command string, which starts with the second character in the input line, is converted to upper case and stored at address 0081H so that the program may use it. The length of the command string is stored at 0080H.

Note that if a program executes the instruction FFH then a jump instruction at address 0038H provided by RP/M will return control to RP/M and display the message "** Trap at XXXX" where "XXXX" is the address of the FFH instruction.

* Initialisation now set to 2400 baud.

SCREEN EDITING

To use the screen editing feature, press Control-@ before starting to enter an input line. The cursor changes to a blinking block to show that you are in screen editing mode. The cursor movement keys will then allow you to move the cursor up, down, left or right. If you hold down the Shift key and press the cursor left or right keys you will delete or insert characters in a line. If you hold down the Control key and press the cursor up or down keys, you will delete or insert lines on the screen. There are many other video card functions which you can activate, and these are described in full in the video card documentation. Do not use Escape sequences which require more than one character to be sent after the Escape or which produce a reply from the video card.

When you have finished editing the screen, press the Return key. The line on which the cursor is positioned will automatically be entered as input to RP/M and screen editing mode will be terminated. Alternatively, press Control-C, which cancels screen editing mode and returns a Control-C to RP/M or the program being executed.

Screen editing can be used not only when entering commands to RP/M, but also when executing programs. For example if you use the Microsoft Disk Basic, you can list lines on the screen, and then use screen editing to alter and reenter them. This is often a faster and simpler means of editing a program than by using the Basic Edit command.

When entering a line using screen editing, the following prompts on the left of the screen are automatically ignored: "*", "£", "-" and "X>" where "X" is an upper case letter. These are the most common prompts output by CP/M programs, and RP/M itself uses the asterisk prompt.

The character used to activate screen editing can be altered by using the S command (see above) to change the value at address 004CH. The initial value is 00H, which is a Null character.

SCREEN DUMP

The contents of the screen may be printed by pressing Control-@ followed by Control-B. The Control-@ enters screen editing mode and the Control-B performs the screen dump and cancels screen editing mode. The entire screen display is output to the printer, with a horizontal line printed above and below the screen image. Trailing blanks on each line are automatically suppressed to optimise printing time.

It is essential to have a printer attached to the system as otherwise the lack of a handshake signal will make the system lock up and you will have to press Reset.

The character used to activate the screen dump may be altered, by using the S command (see above) to change the value at address 004DH. The initial value is 02H, which is Control-B.

PRINTER SUPPORT

RP/M supports a serial printer and/or keyboard attached to the RS232 interface. The speed defaults to 300 bps on reset. Use the U D command (see above) to alter the speed. The printer is used by entering Control-P (see Keyboard section above), or Control-@ and Control-B (see Screen Dump section above) or by system calls from a program (see RP/M System Routines below).

In order to support printers which provide a handshake signal, the CTS input signal to the UART is examined by RP/M. When this signal is low (0V) no characters are output, and the system locks up. If your printer does not have a handshake signal, you must connect the CTS input (pin 8) to 5V (pin 2) on the serial interface connector. Note that the other lines required are RS232 ground (pin 11), and RS232 out (pin 6). Also, if the printer has a keyboard, this is connected to RS232 in (pin 3).

RP/M also supports a parallel printer using the Centronics interface conventions. Port B is the output data port, and port A is used to control the data transfer. Bit 0 of port A is connected to the BUSY line from the printer, and bit 1 is the strobe which indicates to the printer that data is available. To select a parallel printer, use the S command to change location 0003H (IOBYTE). Bit 7 is tested to determine which type of printer to use. If it is 1 then the parallel printer is selected, so change the value from 01H to 81H or 03H to 83H.

A special feature of the printer support is that form feed works with printers which do not have a page throw mechanism. RP/M does this by counting the line feeds output and automatically supplying the necessary number of blank lines when a form feed is encountered. The number of lines per page is set to 66 on reset. This value may be altered by using the S command (see above) to change the value at address 0042H from 66 (42H) to the page length required.

OPERATION WITHOUT A VIDEO CARD

RP/M is designed to be used with an intelligent video card. However for some applications it is useful to be able to operate the system with minimal hardware. RP/M will now operate without a video card if bit 0 of location 0003H (IOBYTE) is set to 0. Since this needs to be set on power-on, this bit is automatically flipped if a link has been installed on the CPU card. This is the Ring Indicator link attached to the Modem Status Port. If this link is made then the serial port is used as the console device. Note that serial handshaking is used, so ensure that this is provided. If there is a video card in the system it will be reset but it will not be used and its keyboard will not operate. If there is a disk card then an attempt will be made to boot a disk, as normal. All messages are output to the serial printer, and input may be from a serial keyboard or from a keyboard on the CPU card (GM811 only).

It is even possible to operate the system with no video card and no serial terminal, but with a keyboard on the CPU card (GM811 only) and a parallel printer for output, by turning on the computer and immediately using the S command without being able to see the serial output, to turn on the parallel printer.

There are several limitations on the facilities provided if there is no video card. No screen editing mode or screen dump is available. Also, no cassette input or output will operate, and use of the R or W commands results in an error message.

MEMORY MAP

This list shows the memory organisation used by RP/M. Since part of this depends on the size of the system, in this list X is the address 256 (100H) bytes below the top of memory, and Y is the address 128 (80H) bytes below the top of memory. The addresses are in hex.

Address	Name	Meaning
0000 to 0002	JBOOT	Jump to JWBOOT (below) to restart RP/M
0001 to 0002	ABOOT	Address of second jump in I/O jump table
0003	IOBYTE	See description below
0004		Reserved
0005 to 0007	JRPM	Jump to JRPM2 (below) which jumps to the RP/M system routines
0006 to 0007	ARPM	Address of logical top of memory
0008 to 0037		Reserved
0038 to 003A	JDIS	Jump to trap message in RP/M
003B to 005B		RP/M workspace - reserved
005C to 007F	FCB	Default file control block
005D to 0064	FNAME	File name
0065 to 007B		Reserved
007C	CREC	Current record to read/write
007D to 007F		Reserved
0080 to 00FF	DFIO	Default file I/O buffer and console input buffer
0100 to (X-1)	PROG	Program area available for the user
X to (X+2)	JRPM2	Jump to RP/M system routines
(X+3) to (Y-1)	EDBUF	Screen editing buffer
Y to (Y+17H)	TJIO	Table of jumps to I/O routines:-
Y	JCBOOT	Jump to RP/M reset
Y+03H	JWBOOT	Jump to restart RP/M
Y+06H	JCONST	Jump to console status
Y+09H	JCONIN	Jump to console input
Y+0CH	JCONOU	Jump to console output
Y+0FH	JLIST	Jump to list device
Y+12H	JPUNCH	Jump to punch device
Y+15H	JREAD	Jump to reader device
Y+40H	CSTACK	Top of stack for RP/M commands
Y+80H	SSTACK	Top of RP/M system stack and top of memory

RP/M SYSTEM ROUTINES

RP/M provides a large number of system routines which programs can call through a standard mechanism. This method is identical to that adopted by the Digital Research CP/M disk operating system, and it is the high compatibility of these system routines which makes it possible to execute many programs written for CP/M under the control of RP/M.

To call an RP/M system routine, set the C register to the number of the routine required, and set the DE or E registers to the input value (if any). Then call address 0005H which jumps to RP/M. When the routine returns, the A register contains the value returned by the routine (if any). Note that all registers are altered by calling RP/M system routines.

This section lists each of the RP/M system routines. If an attempt is made to call a routine which does not exist, RP/M ignores the request. A number of routines perform no function, and in this case the value returned is 0.

Note that you may end a program in three ways. You can restart RP/M by jumping to address 0000H or by calling system routine 0, or if you have saved the original stack pointer you can return directly to the RP/M command processor.

Remember that programs must start by setting the stack pointer to the top of an area of free memory within the program area.

Routine 0 (00H) Restart RP/M

End the program and restart RP/M. This is equivalent to jumping to address 0000H.

Routine 1 (01H) Console Input

Obtain a character in A from the keyboard. Characters are echoed to the console output device.

Routine 2 (02H) Console Output

Output the character in E to the screen, and if Control-P has been used, to the list device as well.

Routine 3 (03H) Reader Input

Obtain a character in A from the serial input port.

Routine 4 (04H) Punch Output

Output the character in E to the serial output port.

Routine 5 (05H) List Output

Output the character in E to the printer.

Routine 6 (06H) Direct Console I/O

If E is FFH then examine the keyboard status. If no character is ready, set A to 0, otherwise set A to the character. If E is not FFH then output E to the screen. This routine should not normally be used, because the RP/M functions such as Control-P and Control-S do not operate.

Routine 7 (07H) Get I/O Byte

Set A to the value of the IOBYTE (at 0003H).

Routine 8 (08H) Set I/O Byte

Set the IOBYTE (at 0003H) to the value in E.

Routine 9 (09H) Print String

Output the string which starts at DE to the screen, and if Control-P has been used, to the printer as well. The string must be terminated by the character "$".

Routine 10 (0AH) Read Console Buffer

Obtain an input buffer from the keyboard, and store this at DE. The value at DE must have been set to the maximum number of characters in the buffer. The byte after this is returned containing the number of characters in the buffer, and this is followed by the characters themselves.

Routine 11 (0BH) Get Console Status

Examine the keyboard status and set A to 0 if no character is ready, or to FFH if a character is ready.

Routine 12 (0CH) No Function

Routine 13 (0DH) Reset File I/O System

Set the data address for file I/O to the default value of 0080H.

Routine 14 (0EH) No Function

Routine 15 (0FH) Open File

Open the cassette file for input or output, and tell the user to insert the cassette. DE must point to the start of the File Control Block (FCB), which is normally at 005CH. A is set to 0. The filename must have been set in the FCB at FCB+1, normally 005DH. Note that the current record field at FCB+20H, normally 007CH, must be set to 0 by the program.

Routine 16 (10H) Close File

Close the cassette file and display a message telling the user to remove the cassette. DE must point to the start of the FCB. A is set to 0. Output files must be closed by calling this routine. If the file has not been opened, then the message "Invalid I/O" is displayed and RP/M is restarted.

Routines 17-19 (11H-13H) No Function

Routine 20 (14H) Read Cassette Record

Read the next record from the cassette file. DE must point to the start of the FCB. A is set to 0, unless end of file is reached, when A is set to 1. The data is stored at the current file I/O data address, which defaults to 0080H but may be changed by function 26 (below). If the file has not been opened, or is open but has been written to, then the message "Invalid I/O" is displayed and RP/M is restarted. Note that RP/M automatically handles cassette read errors and ensures that files are read without error.

Routine 21 (15H) Write Cassette Record

Write the next record to the cassette file. DE must point to the start of the FCB. A is set to 0. The data is transmitted from the current file I/O data address, which defaults to 0080H but may be changed by function 26 (below). If the file has not been opened, or is open but has been read from, then the message "Invalid I/O" is displayed and RP/M is restarted. Note that because of the nature of cassette files, a program reading a file must issue a read instruction before the start of a block is reached on the tape. Although RP/M automatically inserts an interblock gap, this may be inadequate if the software processes the previous record slowly. The solution is simply to insert a delay after each write instruction. This has proved necessary only when a Basic interpreter processes data files, and it is easy to insert the necessary delays into the Basic program.

Routine 22 (16H) Make File

This routine is identical to the Open File routine described above.

Routines 23-25 (17H-19H) No Function

Routine 26 (1AH) Set Data Address

Set the data address for file I/O to the value in DE. This address is the start of the 128 (80H) byte area used for records read or written to cassette.

RP/M INPUT/OUTPUT ROUTINES

The input and output routines in RP/M are accessed by the system routines through a table of jump instructions which may be altered. This table starts 128 (80H) bytes below the top of memory. It should be located by examining the address stored at address 0001H, which points to the second jump instruction in the table. The table is shown in the memory map above.

In addition, two routines may be added to RP/M by a skilled programmer with the ability to produce modified EPROMs. Locate the first unprogrammed location at the end of RP/M. If the value at this location is not FFH then it is called at the end of the RP/M reset sequence. This routine could for example initialise an interrupt system or alter the I/O jump table. The second routine starts at the fourth unprogrammed location and is executed when RP/M is restarted if the value at this location is not FFH. This arrangement allows the first routine to start with a jump instruction which avoids the fourth byte.

The following information briefly describes the requirements for the I/O routines, in case you wish to alter the I/O jump table to use your own routines.

JCBOOT

Perform RP/M reset. This is normally a jump to the start of RP/M, and should not be altered.

JWBOOT

Restart RP/M. This routine is executed when a program ends by jumping to address 0000H, system routine 0 is executed, or Control-C is used to stop a program. You are strongly advised to leave this alone!

JCONST

Examine the status of the console input device. Return the A register set to FFH if a character is ready, or 00H if no character is ready. Remember that the pending character must be stored. The RP/M routine supports keyboards attached to the CPU card (GM811 only), the video card and the serial input port. It also supports screen editing.

JCONIN

Obtain an input character from the console input device and return with the character in the A register, with no parity. The character might already have been stored by the status routine above. The RP/M routine supports keyboards attached to the CPU card (GM811 only), the video card and the serial input port. It also supports screen editing and screen dump.

JCONOU

Output the character in the C register to the console output device. The RP/M routine supports the video card or a serial printer if there is no video card. If a form feed (OCH) is output to the video card then it is converted to a carriage return and line feed.

JLIST

Output the character in the C register to the list device. The RP/M routine supports a printer attached to the serial output port, with all characters sent with even parity, and handshaking using the CTS input signal. RP/M also supports a parallel printer. It provides support for form feed by supplying the necessary number of blank lines.

JPUNCH

Output the character in the C register to the punch device. The RP/M routine sends the character to the serial output port.

JREAD

Return a character from the reader device in register A. The RP/M routine waits for an input character from the serial input port.

SUPPORT OF IOBYTE

The memory location at 0003H is known as the IOBYTE. RP/M uses this value to store information about the hardware configuration, as follows:-

Bit 0: 0 = no video card, 1 = video card. On reset the Ring Indicator link is tested and bit 0 is flipped if the link is installed, which indicates that there is no video card.

Bit 1: 0 = GM811 CPU card with keyboard port, 1 = GM813 CPU and RAM card without a keyboard port. On reset a test is made to determine if the CPU card has the GM813 memory addressing capability. If it has then bit 1 is set and this disables scanning of the keyboard port on the CPU card, since GM813 has no such port.

Bit 7: 0 = serial printer, 1 = parallel printer. Bit 7 is not modified by RP/M and may be changed by the user to indicate that a parallel printer is being used.

The initial value of IOBYTE is 01H, although on reset bits 0 and 1 are automatically set according to the hardware configuration. Only bit 7 may be altered by the user.

FIXED LOCATIONS IN RP/M

For certain applications it may be convenient to modify certain default values used by RP/M. These may easily be changed when using RP/M by using the appropriate command, but if access to an EPROM programmer is available then the initial values may be changed. These are stored at fixed locations in RP/M.

F009H contains the two byte value used as the UART divisor. The normal value is 417 decimal, 01A1H, which is stored as A1 01. This gives a speed of 300 bps (30 characters per second).

F00BH contains the initial value of the IOBYTE. This is 01H, giving a serial printer. Change to 81H for a parallel printer.

F00CH contains the number of lines per page. This is set to 66 decimal, 42H.

CONCLUSION

We hope that you enjoy using RP/M and find that it meets your needs. We are always happy to learn of any genuine errors in RP/M so that we can continue to improve it. However it has been rigorously tested and we hope that this new and more advanced version will prove to be as free of errors as the original version.

We are also interested to learn of any original high quality software which you develop to run under RP/M. Several excellent system software programs such as GBASIC and GEM-DEBUG have already been produced by independent suppliers, and we wish to encourage this by providing whatever technical assistance that you may require.

* This has been altered to give a UART speed of 2400 baud.