MCX BASIC
SERIAL PORT FILE PROTOCOL

Revised December 30, 2010

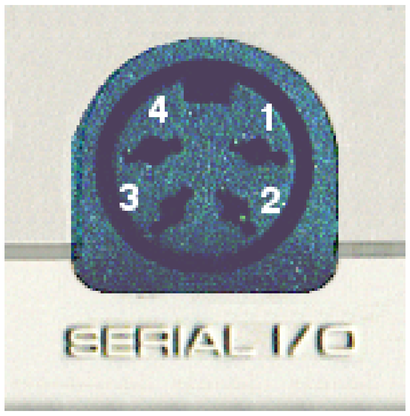
OVERVIEW

This document describes the serial port filing protocol implemented in MCX Basic. This protocol provides a new method for loading and saving files on the MC-10 utilizing the built-in serial port.

The MC-10 must be connected via serial cable to a host system running a compliant server. The cable used must be wired as NULL MODEM in which the host’s TxD output is connected to the MC-10’s RxD input and vise-versa. The GND pin of the MC-10’s serial port must also be connected to GND on the host. No handshaking signals are used. DriveWire cables for the TRS-80 Color Computer are compatible with this system.

MC-10 to Host Cabling

MC-10	Host DB-9	Host DB-25
Pin 2 (RxD)	Pin 3 (TxD)	Pin 2 (TxD)
Pin 3 (GND)	Pin 5 (GND)	Pin 7 (GND)
Pin 4 (TxD)	Pin 2 (RxD)	Pin 3 (RxD)



MC-10 Serial Connector

The MC-10 communicates with the host at 38,400 bps, using 8 data bits, no parity and 1 stop bit (8-N-1). All transactions between the MC-10 and the host are initiated by the MC-10. Every request begins with an Attention byte (hex 21, ascii '!') and is followed by at least one command byte. The server must respond to requests within 2 seconds, otherwise a time-out occurs. A time-out or invalid response may result in the request being sent again.

LOAD FILE:

```
===============================================================================
 COMMAND:  attention, cmd, mode, nameLength, name
-------------------------------------------------------------------------------
      attention   | ! (ascii)  21 (hex)
      cmd         | L (ascii)  4C (hex)
      mode        | 00 = LOAD,  02 = LOADM,  04 = LOAD*,  08 = LOAD SCREEN
      nameLength  | number of name bytes that follow (1..255)
      name        | file identifier string

===============================================================================
 RESPONSE: address, size, checksum
-------------------------------------------------------------------------------
      address     | Block load address MSB..
                  | .. LSB
      size        | Block size MSB..
                  | .. LSB
      checksum    | Data Checksum MSB..
                  | Data Checksum LSB or Error Code
```

MCX Basic initiates a Load File operation by transmitting a 4 byte command sequence followed by the ASCII character string identifying the requested file. The server attempts to locate the specified file and verify that it is of an appropriate type for the given mode. The server is responsible for any needed translation of the file into the raw data bytes expected by the MC-10.

The server responds to a LOAD FILE command by transmitting a 6 byte packet which either describes the first data block from the file or indicates an error condition. An error condition is indicated by a block size of 0 and a non-zero error code in the LSB of the checksum field (see the list of error codes below).

The block descriptor for a successful response consists of the load address, size and checksum of the block's data. The load address is only relevant for mode 02 (LOADM) and is ignored by the MC-10 in all other modes. The checksum is a 16-bit value computed by adding together all of the unsigned byte values from the block's data.

The size of a data block for a Load File operation is an implementation detail of the server. The MC-10 will accept data blocks of any length. The server may choose to provide the entire file as a single block or divide it into smaller pieces. If a checksum error were to occur during the transmission of a block, the entire block must be transmitted again. For this reason it is a good idea to place some limit on the block size.

GET DATA BLOCK:

```
==============================================================================
 COMMAND:  attention, cmd, file#
------------------------------------------------------------------------------
      attention   | ! (ascii)  21 (hex)
      cmd         | G (ascii)  47 (hex)
      file#       | The file number (0-15) associated with the request.

==============================================================================
 RESPONSE: data
------------------------------------------------------------------------------
      data        | contents of current data block
```

When the MC-10 is ready to download the contents of a data block, it sends this 3 byte command sequence to the server.

The file# provided in the command will be 0 when requesting data from the file identified by the most recent LOAD FILE command. If the file# is non-zero then it references a specific data file identified through a previous OPEN DATA FILE command.

The server responds by sending the contents of the current data block. The current data block is the one whose block descriptor was most recently provided to the MC-10 in response to a LOAD FILE or PREPARE NEXT BLOCK command for the given file number.

The MC-10 may re-send this command if a communications error occurs or if a checksum of the data does not match the one provided in the block descriptor.

PREPARE NEXT BLOCK:

```
==============================================================================
 COMMAND:  attention, cmd, file#
------------------------------------------------------------------------------
      attention   | ! (ascii)  21 (hex)
      cmd         | N (ascii)  4E (hex)
      file#       | The file number (0-15) associated with the request.

==============================================================================
 RESPONSE: address, size, checksum
------------------------------------------------------------------------------
      address     | Block load address MSB..
                  | ..LSB
      size        | Block size MSB..
                  | ..LSB
      checksum    | Data Checksum MSB..
                  | Data Checksum LSB or Error Code
```

When the MC-10 requires another block from a file, it will send this 3 byte command sequence asking the server to prepare the next block.

The file# provided in the command will be 0 when requesting data from the file identified by the most recent LOAD FILE command. If the file# is non-zero then it references a specific data file identified through a previous OPEN DATA FILE command.

The server responds by transmitting the 6 byte block descriptor for the file's next sequential block. A size of 0 returned in the block descriptor indicates that the End-Of-File has been reached or that some error has occurred. The checksum will be 0 for an EOF response, otherwise a non-zero error code is returned in the checksum LSB.

The address field of an EOF response contains the default EXEC address when loading a machine language file.

SAVE FILE:

```
==============================================================================
 COMMAND:  attention, cmd, mode, nameLength, exec, size, name
------------------------------------------------------------------------------
      attention   | ! (ascii)  21 (hex)
      cmd         | S (ascii)  53 (hex)
      mode        | 00 = SAVE,  02 = SAVEM,  04 = SAVE*,  08 = SAVE SCREEN
      nameLength  | number of name bytes (1..255)
      exec        | ML Exec Address MSB..
                  | ..LSB
      size        | File Size or ML Load Address MSB..
                  | ..LSB
      name        | file identifier string

==============================================================================
 RESPONSE: status
------------------------------------------------------------------------------
      status      | Error code or 0 if no error
```

MCX Basic initiates a Save File operation by transmitting an 8 byte command sequence followed by the ASCII character string identifying the destination file.

The exec field is only relevant for mode 02 (SAVEM) and will be 0 for all other modes. In modes other than 02, the size field contains the total size of the file's data. In mode 02, the size field contains the Load Address for the machine language file.

The server validates the file name string and determines if such a file can be created or replaced. The server responds by sending back a single status byte indicating whether or not the operation can proceed. The status byte is 0 if the operation may proceed or a non-zero error code if the operation was denied (see the list of error codes below).

WRITE BLOCK:

```
==============================================================================
 COMMAND:  attention, cmd, file#, size, data
------------------------------------------------------------------------------
      attention   | ! (ascii)  21 (hex)
      cmd         | W (ascii)  57 (hex)
      file#       | The file number (0-15) associated with the request.
      size        | Block size MSB..
                  | ..LSB
      data        | Zero or more Data bytes as indicated by size.

==============================================================================
 RESPONSE: checksum
------------------------------------------------------------------------------
      checksum    | Data checksum MSB..
                  | ..LSB
```

The MC-10 sends WRITE BLOCK commands to supply file data during a Save File operation or for a data file that has been opened using the Output or Append modes. Each block is transmitted as a 4 byte command sequence followed by the block's data bytes. The number of data bytes in a block sent by the MC-10 during a Save File operation will not exceed 1024. For data files, the block size will not exceed 256.

The server returns a 2 byte response which is the 16 bit checksum of the block's data bytes. The server computes the checksum by adding together all of the received data bytes as unsigned values.

The MC-10 compares the checksum response with its own computation to determine if the command was successful. If the checksums don't match then the same data may be sent again in a Write Retry command. The Write Retry command is identical to the Write Block command except the command byte is a lower case 'w' (77 hex) rather than 'W'.

When all of the file's data blocks have been successfully sent, the MC-10 sends an empty data block (size = 0) to indicate EOF. The server responds to the EOF block by returning a checksum of 0.

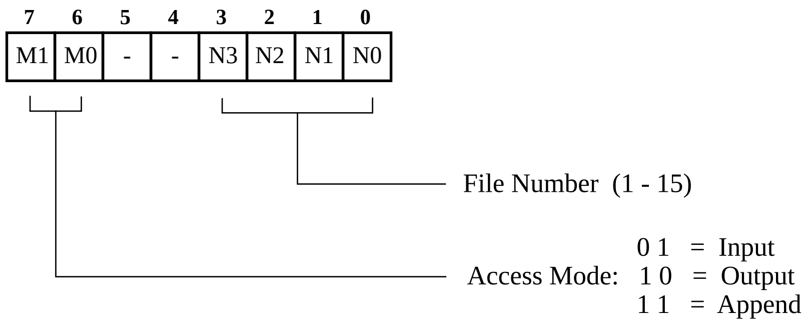
OPEN DATA FILE:

```
==============================================================================
 COMMAND:  attention, cmd, file#/mode, nameLength, name
------------------------------------------------------------------------------
      attention   | ! (ascii)  21 (hex)
      cmd         | O (ascii)  4F (hex)
      file#/mode  | File Number and Access Mode
      nameLength  | Number of name bytes that follow (1..255)
      name        | File identifier string

==============================================================================
 RESPONSE: status
------------------------------------------------------------------------------
      status      | Error code or 0 if no error
```

Execution of an OPEN statement in MCX Basic causes this 4 byte command sequence to be sent to the server followed by the ASCII character string identifying the file.

The file#/mode byte provides both a file reference number and the access mode being requested. The byte is encoded as follows:



The server validates the file name string and determines if such a file exists or can be created. The server responds by sending back a single status byte indicating whether or not the operation was successful. The status byte is 0 if the file was opened or a non-zero error code if the operation was denied (see the list of error codes below).

Once a data file has been opened, the MC-10 will either request (Input) or supply (Output/Append) sequential data blocks in the same manner as a LOAD FILE or SAVE FILE operation. **The block size for data files must not exceed 256 bytes.**

For files opened using the Output or Append modes, the MC-10 sends a zero-length data block when the file is closed. No notification is sent to the server when an Input file is closed. The server can close an Input file after returning an EOF response to the PREPARE NEXT BLOCK command, or when another OPEN DATA FILE command is received for the same file number.

DIR FILE REQUEST:

```
==============================================================================
 COMMAND:  attention, cmd, flag, argLength, argString
------------------------------------------------------------------------------
      attention   | ! (ascii)  21 (hex)
      cmd         | F (ascii)  46 (hex)
      flag        | 00 = First File,  FF = Next File
      argLength   | Number of bytes in the following string argument (1..255)
      argString   | Optional string argument

==============================================================================
 RESPONSE: status, nameLength
------------------------------------------------------------------------------
      status      | Error code or 0 if no error
      nameLength  | Length of file name string ready for download
```

When a DIR command is entered in MCX Basic, the MC-10 sends a 4 byte command sequence followed by the contents of an optional string argument. The flag byte in the command sequence will be 00 when requesting the first file in a directory or FF in subsequent requests for additional files in the same directory.

The server sends back a 2 byte response to these requests. A successful response consists of a status code of 0 (no error) and the length of the name of one file in the directory. After receiving this response, the MC-10 will send a Retrieve Name command (see below) to download the file name string and then issue another Dir File Request command. When there are no more files remaining, the server returns a nameLength of 0 in the response.

If the server returns a non-zero error code in the response, the MC-10 will abort the DIR operation and report the error.

RETRIEVE NAME:

```
==============================================================================
 COMMAND:  attention, cmd, length
------------------------------------------------------------------------------
      attention   | ! (ascii)  21 (hex)
      cmd         | $ (ascii)  24 (hex)
      length      | Expected string length.

==============================================================================
 RESPONSE: nameString
------------------------------------------------------------------------------
      nameString  | ASCII string
```

Upon receiving a response to a Dir File Request or Directory Name Request in which the name length field is not 0, the MC-10 will follow up by sending a Retrieve Name command to download the related name string.

The server responds by simply sending back the content of the ASCII string. The number of bytes returned must equal the value specified by length.

DIRECTORY NAME REQUEST:

```
==============================================================================
 COMMAND:  attention, cmd, flag, reserved
------------------------------------------------------------------------------
      attention   | ! (ascii)  21 (hex)
      cmd         | D (ascii)  44 (hex)
      flag        | 00 = First Directory,  FF = Next Directory
      reserved    | This byte is currently unused

==============================================================================
 RESPONSE: status, nameLength
------------------------------------------------------------------------------
      status      | Error code or 0 if no error
      nameLength  | Length of directory name string ready for download
```

When a DIRLIST command is entered in MCX Basic, the MC-10 sends this 4 byte command sequence to the server. The flag byte in the command sequence will be 00 when requesting the name of the first available directory or FF in subsequent requests for additional directories.

The server sends back a 2 byte response to these requests. A successful response consists of a status code of 0 (no error) and the length of the name of one directory. After receiving this response, the MC-10 will send a Retrieve Name command (see above) to download the directory name string and then issue another Directory Name Request command. When there are no more directories remaining, the server returns a nameLength of 0 in the response.

If the server returns a non-zero error code in the response, the MC-10 will abort the DIRLIST operation and report the error.

SET CURRENT DIRECTORY:

```
==============================================================================
 COMMAND:  attention, cmd, reserved, nameLength, dirName
------------------------------------------------------------------------------
      attention   | ! (ascii)  21 (hex)
      cmd         | C (ascii)  43 (hex)
      reserved    | This byte is currently unused
      nameLength  | Number of bytes in the following name string (1..255)
      dirName     | Directory name string

==============================================================================
 RESPONSE: status
------------------------------------------------------------------------------
      status      | Error code or 0 if no error
```

When a SETDIR command is entered in MCX Basic, the MC-10 sends a 4 byte command sequence followed by an ASCII string identifying the requested directory.

The server responds by sending a single status byte. The status byte is 0 for a successful operation or one of the error codes listed below.

ERROR CODES (decimal):

8	'FC'	Illegal Function Call
34	'IO'	Input/Output Error
36	'FM'	Wrong File Mode
38	'DN'	Invalid File Number
40	'NE'	File or Directory does Not Exist
42	'WP'	Write Protected
44	'FN'	Badly formed File or Directory Name
46	'FS'	File System Error
48	'IE'	Input past End of File
50	'FD'	Unacceptable File Data
52	'AO'	File Already Open
54	'NO'	File is Not Open
56	'DS'	Direct Statement in file