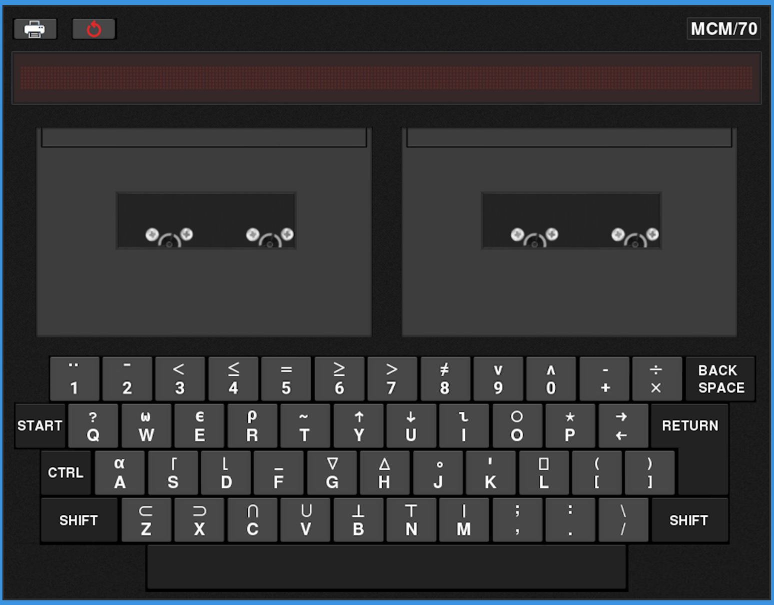
Emulator Series

MCM/70E

An MCM/70 Emulator



York University Computer Museum
Toronto, Canada

MCM/70E
An MCM/70 Emulator

Version 2.1
December 2019

Zbigniew Stachniak

York University Computer Museum
York University, Toronto, Ontario, Canada

CONTENTS

Introduction

The MCM/70 computer, designed by a Canadian company Micro Computer Machines Inc. (MCM) in the period between 1972 and 73, is the earliest example of a microcomputer manufactured specifically for personal use. From the hardware and software engineering point of view it does not have much in common with early hobby computers, such as the SCELBI-8H or MITS Altair 8800, except that all these computers were microprocessor based. By the time the Altair 8800 kit was offered to hobbyists in the early 1975, with its 256 bytes of RAM memory and no high-level programming language capability, the MCM microcomputers were providing in-house APL support for applications ranging from engineering design, modeling and simulation to investment analysis and education. By 1976, the MCM computers were utilized by, among other companies, institutions and organizations, Chevron Oil Research Company, Firestone, IBM, Hospital for Sick Children (Toronto), Government of Ontario (Ministry of the Environment), Mutual Life Insurance Company of New York, Ontario Hydro-Electric Power Commission, NASA Goddard Space Flight Centre, U.S. Army, the Computer Center of the Academy of Science of the USSR, and several North-American Universities.

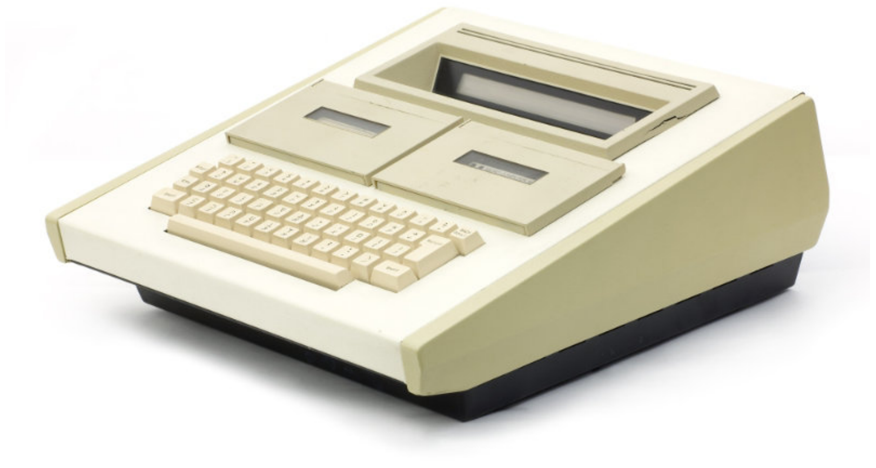


Fig. 1. The MCM/70 computer

The official announcement of the MCM/70 came on September 25, 1973, in Toronto. The computer was unveiled in New York on September 27th and, the following day, in Boston. One of its early prototypes was demonstrated to the APL community in May of 1973 during the Fifth International APL Users' Conference in Toronto. Another prototype was touring Europe in August and September of that year, when the MCM team was showcasing the machine in Holland, Germany, Switzerland, France, Italy, and the U.K. Because MCM was the earliest company to commercially offer a versatile computer for personal use it was also among the first firms to wrestle with the issue of defining personal computing which makes the company as well as its products and policies historically significant. For more information on the MCM/70 computer and the corporate history of MCM, see [3].

Zbigniew Stachniak December 2019

1 The MCM/70 – technical specifications

The MCM/70 was offered in several configurations: with one or two built-in cassette drives, and 2KB, 4KB, or 8KB of RAM.

Hardware

- CPU: Intel 8008,
- ROM: 32KB,
- RAM: 2KB, 4KB, or 8KB,
- display: Burroughs Self-Scan, model C4047, 32 characters, built-in,
- external storage: up to two digital cassette drives, MFE Computer Access Systems, Model 250, built-in,
- APL keyboard: 46 keys, IBM 2741 layout, built-in.

Software

The MCM/70 was sold with systems software (stored in computer's ROM) and some applications software on digital cassettes.

- operating system (OS): EASY and AVS in ROM,
- programming languages: MCM/APL interpreter in ROM,
- APL applications libraries on cassettes including finance, statistics, mathematics, engineering, computer aided instructions, and games libraries,
- word processor: TEXT 70 and TEXT 700 on cassettes,
- printer/plotter support software.

MCM/APL

All MCM computers, including the MCM/70, were sold with an APL programming language interpreter–the MCM/APL. An MCM/70 user interacted with the computer via this interpreter and OS. Therefore, in order to operate the emulator, some knowledge of APL is paramount. The emulator is distributed with *MCM/70 Users' Guide* published by MCM in 1974 which should be used to learn the MCM/APL.

The following APL terminology is used thought this guide.

- **an APL object**: an APL function (program) or data,
- **workspace**: a part of RAM designated for storage of user-defined APL objects as well as for the execution of user's programs,
- **a group**: a collection of APL objects; each group is stored on a tape with a unique identifier which is an integer between 0 and 255.

2 The MCM/70E emulator

The MCM/70 emulator (MCM/70E) was developed at York University Computer Museum between 2018-19. It is a result of an extensive study of the MCM/70 hardware and of a software recovery project that has recovered and preserved MCM software from approximately 100 digital MCM cassettes donated to the museum.

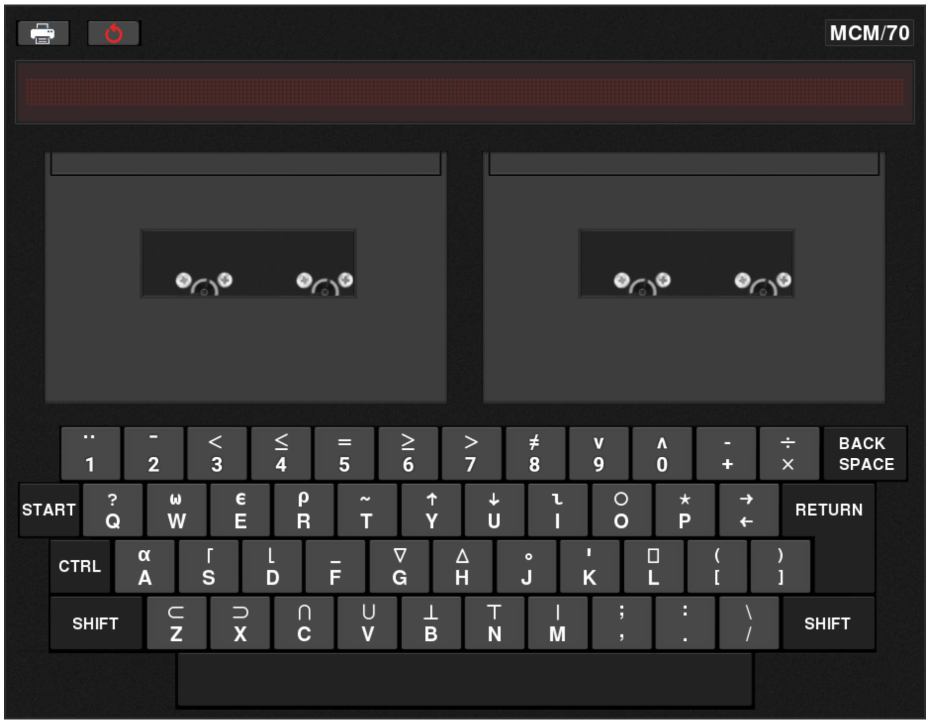

MCM/70
1 2 3 4 5 6 7 8 9 0 + × BACK SPACE
START Q W E R T Y U I O P ← RETURN
CTRL A S D F G H J K L [ ]
SHIFT Z X C V B N M , . / SHIFT

Fig. 2. The MCM/70E emulator

The MCM/70E emulates the original MCM/70 hardware with 8KB of RAM with high historical accuracy. In particular, the memory management, the keyboard, and the display functionality is accurate and so is the emulation of the MCP-132 printer plotter. The emulator operates under the original systems software and is using the original tape storage organization (reconstructed during the above mentioned software preservation project).

Installation

Currently, the emulator is available for Linux platform only. It is written in C and requires OpenGL library and GLUT (or FreeGLUT) utility toolkit which have to be installed prior to the compilation of the emulator.

Extract the tarball mcm70E.tgz. This will create the directory mcm70E_v2.0 containing four directories:

– **APL_F**: directory containing APL fonts used by the emulator,
– **Documentation**: directory containing this guide, the MCM users' guides [4] and [5], and the listings of the *demo.tp* and *utils.tp* tapes,
– **images**: directory containing all the images required by the emulator,
– **Tapes**: directory containing three tapes that the emulator can operate with: the empty formatted tape *empty.tp*, the utility tape *utils.tp*, and an original MCM demonstration tape *demo.tp*.

The mcm70E_v2.0 directory should also include these files:

– the emulator's source files,
– **ROM6k** and **ROMs0-C**: the emulator's ROM files,
– **MCM70E_software_license**: the emulator's software license.

To compile the emulator execute:

```
gcc mcm.c -lGL -lglut -o mcm
```

which will create the executable file **mcm**.

MCM/70E keyboard

The emulator uses the host computer's QWERTY keyboard. Because the APL character set contains several symbols that do not appear on QWERTY keyboards, it is recommended that a keyboard overlay conforming to the MCM/70 keyboard layout is made or key-cup stickers are affixed as shown below.

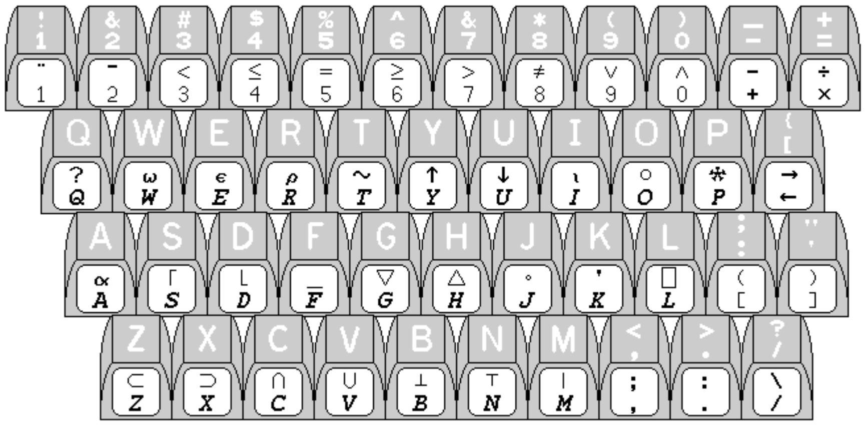


Fig. 3. APL symbols on a QWERTY keyboard are shown as stickers affixed to the front of the keys (black symbols on white stickers). A modified image from http://www.quadibloc.com/comp/kyb05.htm

Starting the emulator

Run **mcm** program. When the main emulator's window is displayed, press START key (implemented as Tab on the QWERTY keyboard). The emulator should display the message **MCM/APL**. Then press RETURN.

Note: the MCM/70 computer did not have an on/off switch. Instead, the computer was started by pressing the START key and was turned off with the OS command ⎕OFF.

The emulator is now ready to accept OS commands and execute APL programs. Try

2+2 followed by RETURN

See also **The demonstration tape** section that describes some sample programs on the demo.tp tape.

Adjusting the emulator's speed

The emulator's speed can be adjusted by changing the value of the parameter **emulator_speed** in the header file mcm.h. The higher the value, the slower the emulator's speed will be. The original MCM/70 hardware executes the

$$0.7 \div \iota 255$$

instruction (and shows the first line of results) in approximately 50 seconds. If desired, adjust the value of **emulator_speed** to get the required speed.

"Garbage" on the MCM/70's display

One of the main obstacles faced by MCM engineers during the design of the MCM/70 was a small amount of memory that the Intel 8008 microprocessor could address directly (just 16KB). The MCM/70's display functionality required an allocation of 222 Bytes of RAM for the display. Therefore when the computer was executing instructions that did not involve displaying outputs, then the display memory was also used to store some data temporarily.

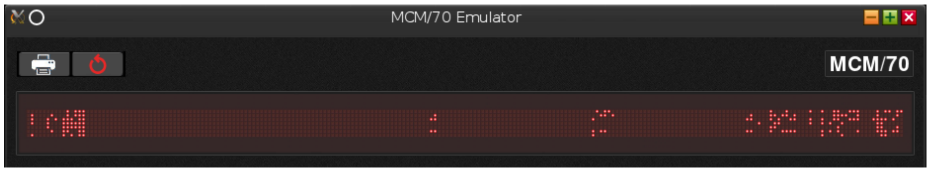


Fig. 4. "Garbage" displayed during the execution of an APL program.

Because during an execution of an APL program, the computer was continuously refreshing its display, the Self-Scan display was showing strange patterns ("garbage", see Figure 4) – these patterns were the values of bytes stored temporarily in the display memory region of RAM.

3 MCM tape organization and sample OS commands

Logical organization of MCM tape storage required all APL objects to be stored in groups – a group being a collection of APL objects that might share some common utility. For example, in one group a user might have collected games exclusively while placing various printing and plotting routines into another group. The MCM/70's operating system provided a rudimentary set of commands to manipulate groups and their content, commands such as group creation and deletion, listing of a group's content, writing to, reading from, and deleting objects from a group. Some of these commands are described in the next section. For other OS commands, consult *MCM/70 Users' Guide* [5].

Tape mounting, closing, ejecting, and formatting

The MCM/70 computers were equipped with up to two digital cassette drives. Their purpose was not only to store APL objects but also to offer virtual memory to extend user's workspace to over 100KB. (In the early 1970s, the virtual memory was available only on some mainframe computers such as the IBM System\370 Models 158 and 168.)

Tape mounting. Follow these steps:

1. Start MCM/70E.
2. Left click the selected tape drive to open its lid.
3. Right click the selected drive to access tape menu.
4. Select a tape.
5. Left click the selected tape drive to close its lid.

NOTE: The emulator requires that tape names are at most 16-character long. It will not load tapes with longer names.

Ejecting a tape. To eject a tape:

1. Left click the selected tape drive to open its lid.
2. Access tape menu by right clicking the selected drive.
3. Select "eject".
4. Left click the selected tape drive to close its lid.

By ejecting a tape, its content will be saved in the tape's file located in the tapes directory **Tapes** of the emulator. However, some information about the ejected tape's content (e.g. about tape directories) will remain in the workspace. To remove such information, one has to clear the workspace using the OS command

□WC

Clearing the workspace is necessary if another tape is to be mounted on the same drive.

Tape formatting. Before a tape could be used with an MCM/70, the tape had to be formatted. The MCM/70E emulator is distributed with an empty and formatted tape *empty.tp*. However, such a tape can be also created by inserting any tape into the emulator's left drive and issuing an OS command to format the tape. This is done by following these steps:

1. Insert a tape into the left drive.
2. Verify that this is indeed the tape you want to format by viewing its content:
   □XN ι0
   If this is not the tape that you want to format, eject it.
3. Clear the workspace with the command
   □WC
   This has to be done because, in step 2, the MCM/70 copied some directories into the workspace and an attempt to format the tape would result in TAPE ERROR.
4. Enter the OS format command
   (ι0) □XI 0

The formatting operation takes a few minutes to complete. Furthermore, formatting will remove all the information previously stored on the tape and will initialize all the tape directories.

If a user wants only to "softly" erased a tape, i.e. to clear the tape's main directory that lists all the groups on the tape while leaving all remaining data unchanged, then issue the command

(ι0) □XI 1

Closing a tape. The MCM/70 computer can modify the content of a mounted tape (e.g. store and delete APL objects). However, these changes become permanent only when the tape is closed. The tape eject action will not accomplish that because all information about tape modification are initially stored in the workspace rather than directly on the tape. However, there are several ways to make all the modifications permanent. The first one is to use the OS command □WC which has already been discussed above. It's purpose is to clear the entire workspace. However, before that cleanup is done, MCM/70 makes all the changes done to a tape permanent.

Note: although the selected tape has been closed, it remains mounted until it's ejected.

Another way of closing a tape is to use the OS command

```
⎕OFF
```

whose purpose is to turn the MCM/70 off. As in the case of ⎕WC, the tape remains mounted and requires its ejection to be saved into a file.

Tape related OS commands

After a tape has been mounted, its content can be loaded, viewed, and modified in several ways. Here are some essential tape-oriented OS commands.

A. Listing all the groups on a tape.

```
⎕XN ⍳0          for a tape mounted on the left drive
⎕XN[2] ⍳0          for a tape mounted on the right drive
```

B. Listing the content of a group with group identifier n.

```
⎕XN n          for a tape mounted on the left drive
⎕XN[2] n          for a tape mounted on the right drive
```

C. Loading an APL object named xxx in group n into workspace.

```
n ⎕XR ”xxx”          for a tape mounted on the left drive
n ⎕XR[2] ”xxx”          for a tape mounted on the right drive
```

D. Loading all the APL objects in group n.

```
n ⎕XR ⎕XN n          for a tape mounted on the left drive
n ⎕XR[2] ⎕XN[2] n          for a tape mounted on the right drive
```

E. Saving an APL object xxx in group n.

```
n ⎕XW ”xxx”          for a tape mounted on the left drive
n ⎕XW[2] ”xxx”          for a tape mounted on the right drive
```

F. Deleting an APL object xxx from group n.

```
n ⎕XD ”xxx”          for a tape mounted on the left drive
n ⎕XD[2] ”xxx”          for a tape mounted on the right drive
```

You can delete all the objects from a group nnn by issuing the command:

```
n ⎕XD ⎕XN n          for a tape mounted on the left drive
n ⎕XD[2] ⎕XN[2] n          for a tape mounted on the right drive
```

GROUP and TAPE copy

A group can be copied from one tape, say t1.tp to another tape, say t2.tp using an APL program △CP stored in group 0 of tape utils.tp (provided with this emulator). Here are the steps to copy a single group, say group 120, from t1.tp to t2.tp.

1. If t1.tp has both △CP and SORT codes in its group 0, then skip this step and go to step 2. Otherwise, copy these programs from utils.tp as follows:
   - mount tape utils.tp in the left drive and t1.tp in the right drive;
   - activate AVS (A Virtual System) by issuing
     ⎕XS 0
   - enter
     0 ⎕*XW*[2] "△*CP*"
     (this will copy △*CP* from utils.tp to group 0 of t1.tp) followed by
     0 ⎕*XW*[2] "*SOR*"
     (this will copy *SORT* from utils.tp to group 0 of t1.tp).
   - eject utils.tp;
   - enter
     ⎕WC
   - make changes to t1.tp permanent by ejecting t1.tp;

   Now, the group 120 is ready to be copied from t1.p to t2.tp.
2. mount tape t1.tp in the left drive and t2.tp in the right drive;
3. activate AVS (A Virtual System) by issuing a command
   ⎕XS 0
4. execute
   2 △CP 120
   this will copy the group 120 from t1.tp to t2.tp;
5. enter
   ⎕WC
   to make changes to t2.tp permanent;
6. eject both tapes (this will also save the modified t2.tp in the tape's files).

To copy the entire tape t1.tp to t2.tp replace 2 △CP 120 in step 4 in the above procedure with

2 △CP ⍳0

Other useful OS commands and error messages

⎕FN – list all function names currently residing in the workspace,
⎕VA – list all variable names currently residing in the workspace,
⎕WA – return workspace available in KB.

The MCM/70 can issue several error messages depending on the type of a problem. Consult *MCM/70 Users' Guide* for all error categories and their meanings.

4 The demonstration tape

The current release of the MCM70E emulator includes a demonstration tape–demo.tp–containing sample programs from three main software categories: utilities, communications, and applications. This tape was created in late 1975 by MCM for its distributors as promotional and sales aids. The group 80 contains several games, such as HOR (HORSE, possibly the world's earliest game implemented on a microcomputer), HAN (HANGMAN), GUN, etc. Here is the complete listing of this group:

```
GROUP 80:
J3   CLE   SL   GUN   RD   RN   MOO   CRA   RL   HOR   F   M△
C   X3   HIT   F△   CD   D△   H△   PLA   PAY   TOT   LN   N
STR   ST   ALP   HAN   WOR   GO   ADD   DAY   MG4   MAG   AGG
```

Because the size of group 80 is more than 8KB, all its programs cannot be loaded into RAM at the same time. However, this group can be made part of the MCM/70's virtual memory without any need to load programs directly into RAM. To make that happen, type

```
□XS 80
```

Now, simply type a game's name, say HAN, and you are ready to play HANGMAN. Please be patient with the garbage on the display and slow execution – in 1974, that was the state of the art in PC computing.

5 Interrupting program execution

The MCM/70's OS offers a procedure to interrupt the execution of programs in cases when such interruptions are desired (e.g. errors in programs, printing large amounts of date, etc.). To interrupt an execution of a program, a user simply presses the CRTL key until the display freezes. Then a user has two options:

(←) This option requires a user to press the ← key;
it causes the computer to finish the execution of the line of code at which the suspension took place, after which the computer displays that line.

(→) This option requires a user to press the → key (SHIFT + ←);
it causes the computer to display the message 'INTERRUPT'. Pressing Return key will cause the computer to display the line of code at which the suspension took place.

Because GLUT utility toolkit used for the implementation of the emulator considers CTRL key press as a key modifier rather than an ordinary key press, pressing CRTL key alone cannot be detected. Therefore, in the above interrupt procedure the emulator simulates the pressing of CTRL key by pressing simultaneously CTRL key and some alphanumeric key. This simulation of the CTRL key applies only to the interrupt procedure.

6 The MCP-132 printer/plotter

In 1975, MCM offered its MCP-132 printer/plotter (in fact, the device was the Diablo HyType I printer). The MCM/70E emulator includes the emulation of this printer. By clicking the printer icon in the top left corner of the emulator, the printer's window is displayed, as shown in Figure 5.

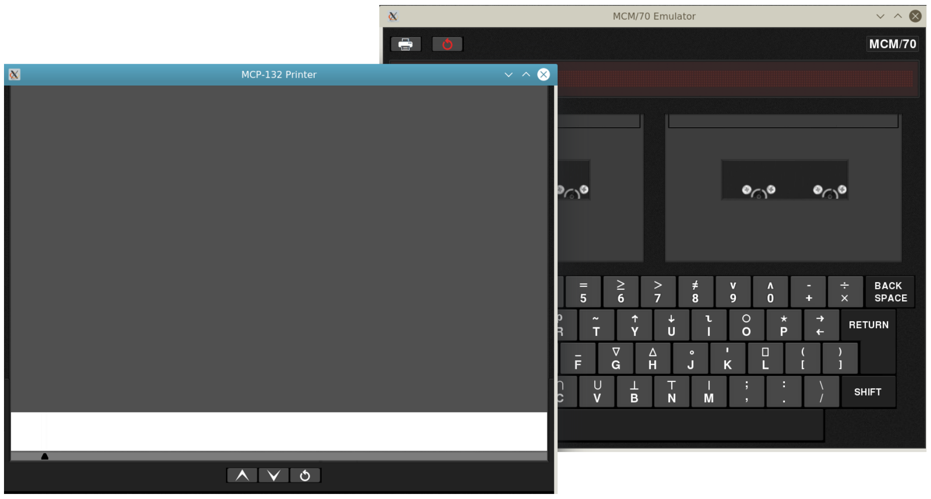


Fig. 5. The emulation of the MCM MCP-132 printer.

The printer's window contains three buttons for scrolling printer's paper up and down, and for resetting the printer. These three operations are describe blow.

Scroll paper up one line: this action causes the paper to move up one line without any loss of printed information.

Scroll paper down one line: this action causes the paper to move down one line without any loss of printed information. However, when scrolling down is attempted but not possible (out of paper), the reset button will change its color to red and printing will not be possible until the printer is reset.

reset printer: clicking the reset button resets the printer: a new and empty roll of printer paper is inserted into the printer and printing can be resumed; all previously printed information is lost.

Connecting and initializing the printer

By clicking the printer icon, the MCP-132 printer is **connected** to the MCM/70E emulator but it is not ready to print anything. The MCM/70E has to be instructed to **redirect the output** to the printer and this is accomplished by issuing the command:

$$\square OUT\ \ 1$$

This instruction tries to interface the MCM/70 with the output device 1 (which is the default identifier of a printer). The printer responds by providing the MCM/70 with two pieces of information: the answer back code (which is 66 for this printer) and printer's current status (initially, its value is 241, indicating that the printer is ready). In short, after issuing the $\square OUT$ 1 command, the display should show three values: 1, 66, 241. Now, the printer is ready to print. Typing

$$\square \leftarrow' HELLO\ \ APL\ \ WORLD'$$

should cause the printer to print the $HELLO\ \ APL\ \ WORLD$ message.

Next, load the tape demo.pt. Make group 1 active by issuing the command:

$$\square XS\ \ 1$$

Then enter BUN to execute the program BUN which prints... yes, an image of a *Playboy* bunny logo as an ASCII art (see Figure 6).

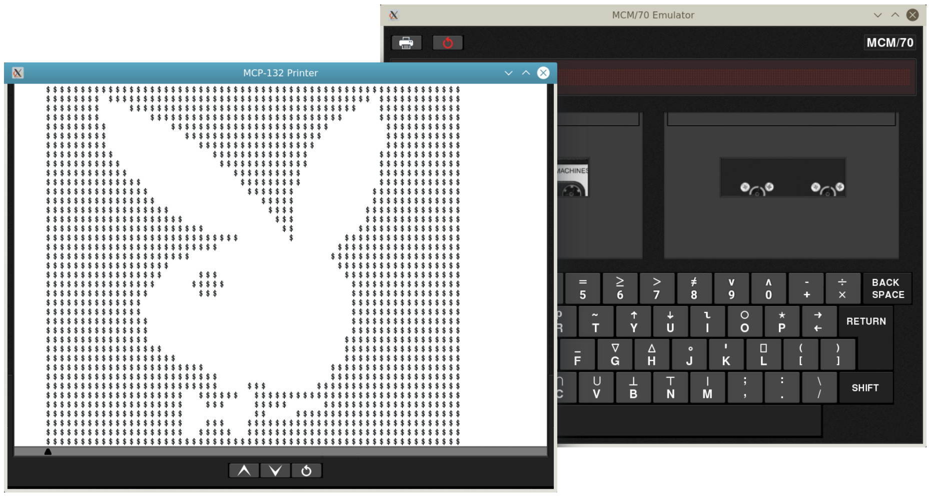


Fig. 6. *Playboy* bunny logo printed on the MCM MCP-132 printer.

Since October 1967, when *Playboy* magazine published an ad featuring the logo in the ASCII format, several versions of the logo had been created and circulated all over the computer industry and academia. BUN is one of the earliest examples of an ASCII art software written, or adapted for a microcomputer. It instructs a printer to print spaces and '\$' characters in 60-character-long lines resulting in the printout depicted in Figure 7. More information on this and other early MCM software in [1] and [2].

Printing and plotting on the MCP-132 printer can be customized in a number of ways: the number of characters or lines to be printed per inch can be set as well as several other parameters. Please consult the MCP-132 manual [4], included in the emulator's package, for details.

Plotting on the MCP-132

The MCP-132 manual describes plotting features of the MCP-132 printer. Before studying the manual in detail, you can do a few simple plotting exercises, described below. To this end, you need mount and initialize the tape utils.tp. The group 201 on this tape contains printer facilities that have to be made available to the MCM/70E first. Instead of loading all these programs into RAM, we can instruct the computer to virtually extend its memory to include the entire group 8 (as the active group). All of that can be done by following these steps:

1. mount the tape utils.tp in the left drive;
2. activate AVS (A Virtual System) and make the group 201 active by issuing ⎕XS 201

Next, make sure that the printer is selected, as described above. Now, we are ready to plot.

First, enter the command

$$1\ \ 0\ \ TITLE\ \ 'HELLO\ \ APL\ \ WORLD'$$

which should plot a large title 'HELLO APL WORLD'. Then enter

$$0.5\ \ \bar{\ }0.5\ \ BOX\ \ 4\ \ 0.75$$

Finally scroll the paper a few lines up to reveal a framed title as shown in Figure 7.

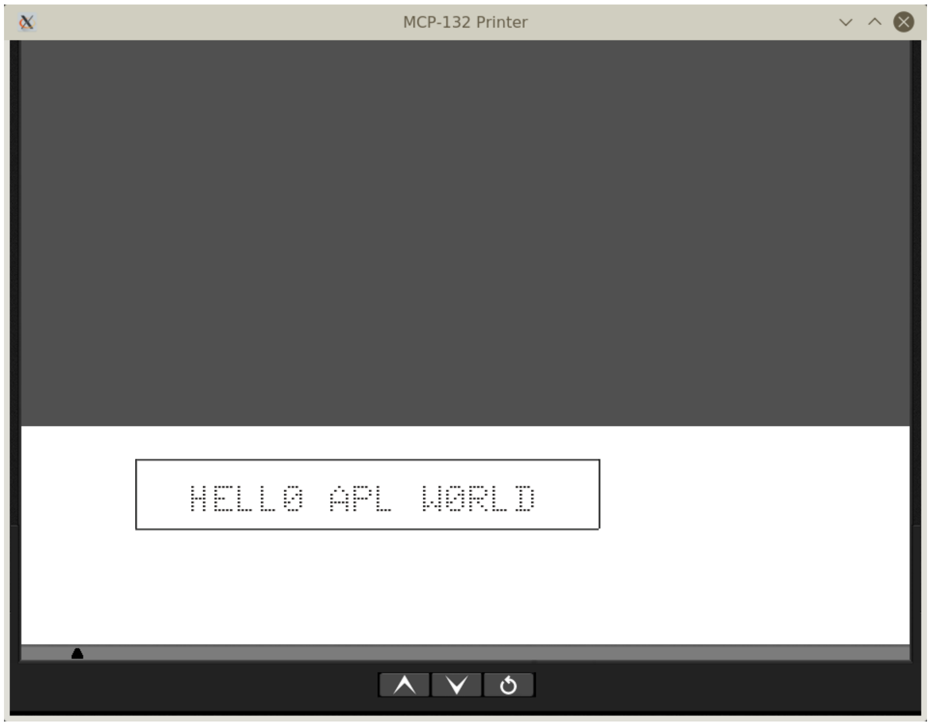


Fig. 7. Plotting on the MCP-132 printer.

In the next exercise, we shall plot the graph of the function 2X as a 4” by 4” graph using 20 points. To this end, follow these steps (assuming that the tape utils.tp is mounted in the left drive and that the group 201 has been made active with the ⎕XS 201 command):

1. type PLOT EQU
2. answer EQUATION:
   by writing the definition of the function Xx2
3. answer DEFINE INDEPENDENT VARIABLE:
   by typing X ← ι 20
4. answer PLOT CHARACTER(S):o
   by pressing RETURN
5. answer WIDTH,HEIGHT (INCHES): by typing 4 4
6. answer X AXIS TITLE:
   by typing TEMPERATURE
7. answer Y AXIS TITLE:
   by typing ICECREAM

Figure 8 shows the plot of this function.

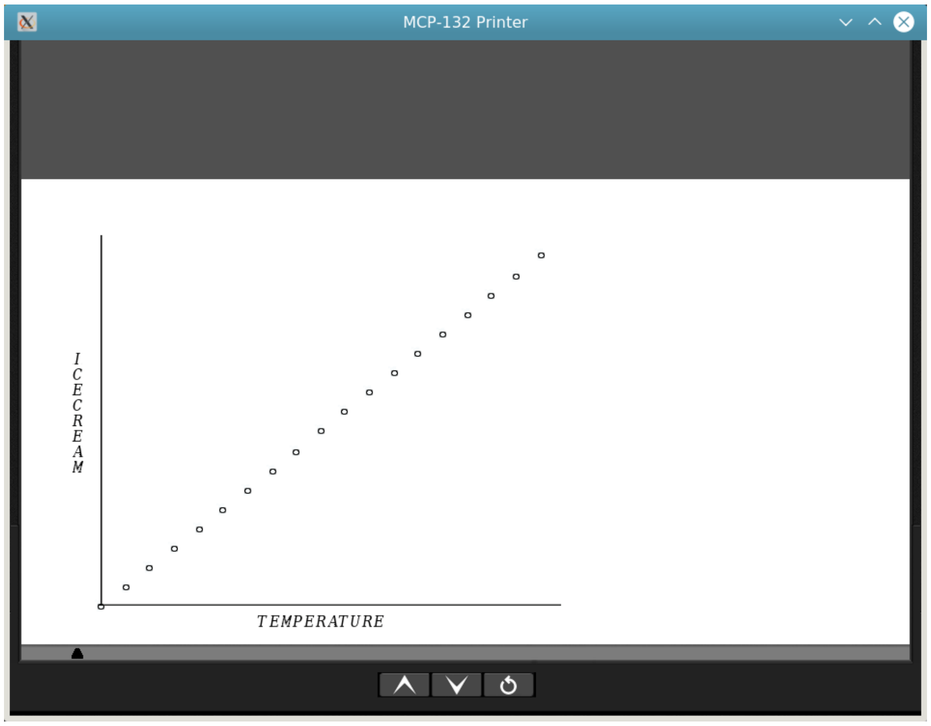


Fig. 8. Plotting the graph of a function on the MCP-132 printer.

7 Power fail protection

The MCM was among the first companies to equip its computers with battery-operated power fail protection systems. Not much is known about such a system for the MCM/70 computer from primary sources. It is only known that this system was implemented as part of the MCM/70's switching power supply and that it was under control of AVS component of the computer's operating system. Sporadic information about power fail protection can be found only in the promotional. For example, one of the MCM/800 promotional brochures describes this system as follows:

> *Transient Power Loss – System continues under battery power.*
> *Extended Power Loss – Initiate orderly shutdown; workspace saved on AVS-active cassette tape; automatically reloads and continues when power restored.*

Some related information can be extracted form the computer's manual (see [5], Appendix B). Fortunately, experiments with the MCM/70E revealed sufficient details about this power fail protection system from the user point of view. What follows are steps to be taken to simulate power failure and recovery from such a failure while operating an MCM/70.

Making power fail protection active

The MCM/70E simulates power failure using two function keys:

F1 Pressing the F1 key, simulates both cutting off power to the computer as well as low battery power.
F2 Pressing the F2 key, simulates both the restoration of power to the computer as well as making batteries fully charged.

To activate power protection system, a cassette has to be loaded into the left drive and AVS has to be activated with, say, this instruction

□XS 0

When AVS is inactive, then pressing F1 will only cause the MCM/70 to perform orderly shutdown of its operations, whether a cassette is mounted in the left drive or not. The entire workspace (residing in RAM) is lost which can be seen when power is restored (by pressing F2), START key is pressed, and the content of workspace is checked, say with □ VA (list defined variables stored in the workspace) or □ FN (list defined functions stored in the workspace).

Simulating power failure with AVS active

Let us assume that a tape, say empty.tp, has been mounted in the left drive and AVS has been activated. Create a variable B in the workspace wit, say, this instruction

$$B \leftarrow 70$$

Now, simulate power failure by pressing F1. The MCM/70 stores the content of the workspace as well as other vital information (the session) on empty.tp. Then it shuts its operations down. Leave the cassette in its drive.

Finally, ”restore power” by pressing F2, then turn the MCM/70 on by pressing START. The computer restores the entire session form the tape. You can verify this by checking if B is in the workspace (type B).

In the above experiment, the tape empty.tp was left in the drive. This is not necessary. You can remove it with the session saved but please do it by ejecting it (this will physically save the modified empty.tp in the Tapes directory of the emulator). Then, ready the next section.

AVS tapes

In the event of a power failure and AVS active, the MCM/70 not only saves the session and also ”marks” the tapes as an AVS tape – meaning a tape with a saved session. To use such a tape, the tape has to be loaded into the left drive **before** the START key is pressed. When a tape is present in a drive and

START is pressed OS is checking whether or not a tape is an AVS type. If it is not, OS does nothing. If it is, the OS restores the session stored on such a tape.

AVS tapes cannot be used for anything else but for restoring sessions, whether or not they contain additional groups. Trying to list groups on an AVS tape will result in an error. To turn an AVS tape back into a "normal" tape, follow these two steps with AVS mounted:

– deactivate AVS with the command ⎕XF ι0
– unload tape by ejecting it (select "eject" from the tape menu).

Acknowledgments

There are only a few MCM/70s left in computer museums and private collections. Unfortunately, most of them are in poor technical condition that prevents computer historians and vintage computing enthusiasts from using these machines directly. Because of its high historical accuracy, the MCM/70E emulator provides a unique tool not only for experimentation with the MCM/70 as an early personal computing hardware but also for the study of early microcomputer software including personal software.

Several individuals contributed significantly to the development of the emulator. My special thanks go to the vintage computer engineer Josh Bensadon – without his expertise and many hours devoted to the unraveling of the MCM/70's hardware design mysteries, this project would be hard to complete. I would also like to thank former MCM engineers Gord Ramer and Andre Arpin for sharing their MCM design expertise, and to Cam Farnell for testing the emulator and clarifying several MCM/70 software and hardware features including the ⎕ ZZ instruction. Finally, my thanks go to Santo Nucifora for performing several crucial tests on his working! MCM/70.

Toronto, 2019

References

[1] Stachniak, Z. Software Recovery and Beyond: The MCM/70 Case, *IEEE Annals of the History of Computing* vol. 41, nr. 4 (2019), pp. 110-118.,
[2] Stachniak, Z. MCM on Personal Software, *IEEE Annals of the History of Computing*, vol. 39, no. 1 (2017), pp. 29–51.
[3] Stachniak, Z. *Inventing the PC: the MCM/70 Story*, McGill-Queen's University Press (2011).
[4] *MCP-132 Printer, Users Guide*, rev. 1., MCM, June 1977.
[5] *MCM/70 Users' Guide*, MCM (1974).