Commodore

T·H·E

EMULATOR

TO EMULATE THE BBC MICRO ON THE COMMODORE AMIGA

THE EMULATOR DOCUMENTATION VERSION 1.2

COMMODORE BUSINESS MACHINES (UK) LTD.

Commodore Business Machines (UK) Ltd.
Commodore House, The Switchback,
Gardner Road, Maidenhead,
Berks SL6 7XA

Introduction

THE EMULATOR is a program which will give your Amiga the appearance of a BBC Model B with BASIC 2 and DFS. THE EMULATOR will run BBC BASIC with the most commonly used VDU and *FX calls, and call 6502 subroutines.

It runs as a PROCESS on its own screen within the Amiga multi-tasking operating system, which means you can run THE EMULATOR at the same time as other Amiga programs such as editors and Workbench accessories. Therefore, if you have an Amiga 2000 with a Bridgeboard and 3 Mbytes of memory, you can run BBC applications, PC applications, and native Amiga applications at the same time — with no data transfer problems between the environments. You can even run several EMULATORs at once.

The principal objective in creating THE EMULATOR is to run existing educational software, written in BBC BASIC or in a mixture of BBC BASIC and 6502 assembler. This has been achieved without any degradation of the Amiga's native advantages, such as split screens and multi-tasking.

THE EMULATOR strategy has been to create a "bridge" from the BBC operating system to the Amiga's operating environment. This means THE EMULATOR can run nearly all "legal" BBC software which uses BASIC and the OS. However, THE EMULATOR cannot run "illegal" software that dodges the OS and goes directly to the BBC screen memory or to the hardware. In practice, this means THE EMULATOR can run nearly all educational programs written in BASIC or a mixture of BASIC and assembler, but not programs such as arcade-style games.

Principal components of THE EMULATOR software are as follows:

Beebos: A "bridge" from the BBC operating system through to the Amiga graphics libraries, maths libraries, etc. Nearly all BBC calls are supported, including OSBYTE (*FX calls).

DFS: "Disk Furtling System" — a bridge from DFS through to AmigaDOS.

BASIC: A new implementation of BBC BASIC, re-coded from scratch in 68000 assembler. It is as compatible as possible with BBC BASIC 2.

6502: This emulation of the 6502 chip — written in 68000 assembler — enables programs to call assembler subroutines.

Other elements in the software are a "CLI environment" entered by *GO, and a simple 6502 monitor, entered by *MONITOR.

This documentation assumes basic familiarity with the Amiga environment as explained in the Amiga User Guide; it also assumes familiarity with the BBC computer. Material which needs a more advanced understanding of the Amiga are flagged as "Tech notes" — ignore these if you want to.

To get started double-click on THE EMULATOR Icon. If you are running on a single drive system you may be prompted to reinsert your Workbench to allow THE EMULATOR to load certain Amiga system libraries. You will then be in THE EMULATOR.

Note that THE EMULATOR expects to find BASIC in the same drawer as itself. If THE EMULATOR can't find BASIC a CLI window will appear with a '*' prompt.

To PAUSE THE EMULATOR, perform a mouse-click anywhere in THE EMULATOR screen. This will also present you with an Amiga "drag bar" at the top of the screen which can be used to pull the screen up and down, or rearrange it relative to the Workbench and any other screens you may have open. With THE EMULATOR in pause mode, it is possible to run various accessories from the Amiga Workbench and Extras disk, including Palette and GraphicDump utilities. To get rid of the drag bar and run THE EMULATOR, click again anywhere in THE EMULATOR screen below the drag bar.

To EXIT THE EMULATOR, enter the special command:

```
*CLOSEDOWN
```

Tech Note: In order to run properly, THE EMULATOR needs to have at least 300 kbytes free memory, and to be able to find 'mathstrans.library' in its LIBS: directory. If these conditions are not met, it will fail to open. To gain maximum speed in 6502 emulator, THE EMULATOR uses one or two elegant routines — this means THE EMULATOR will run on a system with a 68010 or 68020 unless you disable instruction caching (68020), and run a utility to catch MOVE SR instructions (68010 and 68020).

The keyboard

A good idea when first entering THE EMULATOR is to press CAPS LOCK — this will allow you to talk to BASIC sensibly.

THE EMULATOR pays attention to your Keymap and Preferences settings as determined by your Amiga Workbench. Your Preferences determine such attributes as display positioning and key repeat delays — these can be changed from within THE EMULATOR by *FX 11 and 12 if necessary. The numeric keypad works as you would expect.

Certain keys are "stolen" by THE EMULATOR to have special effects. These are as follows:

Function keys	— as BBC function keys; F10 used for F0
Cursor keys	— as BBC cursor keys for copy key editing
Backspace	— acts like BBC DELETE
Del	— acts like copy key
Help	— acts like BREAK

Note that some protection has been applied to the BREAK facility. To BREAK it is necessary to hold down the left Amiga key (CBM key on some A500s) then press the 'help' key. This performs a "warm boot" like BBC BREAK; you can also perform a "cold boot" by holding down the 'ctrl' key and left Amiga then pressing the help' key. Cold boot differs from warm boot in that it resets the programmable function keys and resets the DFS — note that this involves a few seconds pause for disk activity.

Graphics modes

THE EMULATOR supports the following graphics modes:

0	640 x 256	2	colours	80x32
1	320 x 256	4	colours	40x32
2	320 x 256	16	colours	20x32
3	text only	2	colours	80x25
4	320 x 256	2	colours	40x32
5	320 x 256	4	colours	20x32
6	text only	2	colours	40x25
7	Teletext display			40x25

In modes 2 and 5 the horizontal resolution is actually double the original. The double width (20 column) characters are rendered accurately.

The BBC's Teletext (Mode 7) display is software emulated by THE EMULATOR. Mode 7 is the fastest mode on the BBC, but the slowest mode on THE EMULATOR. This is because the Amiga has to emulate character-based hardware with pixel-based software.

The speed of THE EMULATOR Mode 7 should be fast enough for all practical application purposes; however, it is not recommended for program editing. For this reason, the "power up default" for THE EMULATOR has been set to Mode 6.

VDU Drivers

The following VDU calls are supported:

VDU 0	No effect
VDU 1	Next character to printer only
VDU 2	Turn printer on
VDU 3	Turn printer off
VDU 4	Text at text cursor
VDU 5	Text at graphics cursor
VDU 6	Enable screen output
VDU 7	Make a beep
VDU 8	Backspace cursor
VDU 9	Forwardspace cursor
VDU 10	Move down cursor
VDU 11	Move up cursor
VDU 12	Clear text window
VDU 13	Carriage return
VDU 14	Page mode on
VDU 15	Page mode off
VDU 16	Clear graphics window
VDU 17	Change, text colour
VDU 18	Change graphics colour — GCOL 1 to 3 TEXT ONLY
VDU 19	Change colour palette
VDU 20	Restore default palette
VDU 21	Disable screen output
VDU 22	MODE
VDU 23	User-defined character
VDU 24	Define graphics window
VDU 25	PLOT
VDU 26	Default windows
VDU 27	No effect
VDU 28	Define text window
VDU 29	Define graphics origin
VDU 30	Home cursor
VDU 31	Move cursor
VDU 127	Backspace and delete

Note that THE EMULATOR character set is always fully "exploded" — which means any character from 32 to 126 and 128 to 255 can be redefined using VDU 23. One example is VDU 23,1 which is used to switch the cursor on/off is supported.

GCOL modes 1 to 3 are not available from the Amiga graphics library and not supported, except for the special case of text output at the graphics cursor.

Operating System Commands

The following standard commands are supported via THE EMULATOR CLI:

*CAT <drive>
*BASIC
*CODE [X, Y]
*EXEC <file>
*FX A [,X,Y]
*GO [<address>]
*HELP [<subject>]
*KEY N <string>
*LOAD <file> [<address>]
*LINE <text>
*MOTOR 0/1 — NO EFFECT
*OPT X,Y — OPT 4 ONLY
*RUN <file>
*ROM — NO EFFECT
*SAVE <file><start><end/+1n>[<x>][<1d>]
*SPOOL <file>
*TAPE — NO EFFECT
*TV X,Y — NO EFFECT

GO by itself will enter a special "CLIPrompt" environment, with a "" prompt — to get back to BASIC use '*BASIC'.

A number of special commands are also supported; these are discussed elsewhere in this document.

*AEXEC [<file>]
*ASPOOL [<file>]
*CLOSEDOWN
*CONNECT [<baud>]
*FORCECLOSE
*ICOPY <source> <dest> <files>
*IMPORT <source> <dest>
*MONITOR
*SEVER

ASPOOL and AEXEC

*ASPOOL and *AEXEC are like their DFS equivalents, but take an AmigaDOS path-name as a parameter, and work in AmigaDOS compatible format. They are used to transfer text files between THE EMULATOR and the standard Amiga environments. For example, to export a program called Spire to the Amiga RAM disk in AmigaDOS compatible format, use

```
*ASPOOL RAM:Spire
LIST
*ASPOOL
```

This file could then be edited using an Amiga editor such as MicroEmacs; it could then be imported back again by

```
NEW
*AEXEC RAM:Spire
```

Printer redirection by CTRL-B talks directly to the serial and parallel ports, ignoring Preferences settings. To get a listing using the printer device which pays attention to Preferences, use

```
*ASPOOL PRT:
LIST
*ASPOOL
```

Alternatively, you might prefer to ASPOOL to a RAM: file as above, start a new CLI on the Workbench, then on this CLI enter

```
COPY RAM:Spire PRT:
```

This will allow you to print as a background process, without holding up THE EMULATOR.

If you have Workbench 1.3, you may wish to experiment communicating directly between THE EMULATOR and other Amiga processes by means of "pipes". For example, try pulling down THE EMULATOR in front of an AmigaDOS CLI window, load a BASIC program in THE EMULATOR, then type

```
      TYPE PIPE:A          in the AmigaDOS CLI
then  *ASPOOL PIPE:A       in THE EMULATOR
      LIST
      *ASPOOL
```

Disk Furtling System (DFS)

DFS allows an Amiga "drawer" or directory to appear as a complete DFS disk to THE EMULATOR. The normal DFS limitation of 31 files per drawer has been increased to 63 files per drawer. It is also possible to fill an Amiga disk with as many DFS drawers as you like.

THE EMULATOR startup default is to assign DFS drives 0,1,2,3 to Amiga drawers Drive0, Drive1, Drive2 and Drive3. This can be changed later using *ASSIGN — eg. to assign DFS drive 0 to the Amiga RAM disk use

*ASSIGN 0 RAM:

or, more generally,

*ASSIGN <DFS drive number> <AmigaDOS path name>

The DFS expects to find its files in the form

B.SPIRE (<directory>.<name>),

and its catalog information in a special file DFSCat. If you try to assign to a drawer without a DFSCat you will be presented with a warning message "No catalog". If you try to access a drive which has not been assigned or which has no catalog you will get an error message "Drive not valid". If you forget where a drive is assigned, you can find out by typing

*ASSIGN <drive>

It is possible to access a DFS drawer from outside THE EMULATOR eg. from the Amiga CLI, or from a utility program such as CLIMate. If you have made changes in the drawer from outside THE EMULATOR, then you can build a new DFSCat by using the command

*MAKE <drive>

For example, to get a valid DFS "disk" in the Amiga RAM disk, you should

*ASSIGN 0 RAM:
*MAKE 0

If any files of the form <dir>.<name> are found in the specified directory, MAKE will prompt with the filename; you can then enter a (hex) load address followed optionally by an execution address, or just press return if this has no relevance.

If you have added or deleted files to an existing DFS drawer from outside THE EMULATOR, you can update the DFSCat to reflect just the changes made by

*REMAKE <drive>

Standard DFS and UTILS commands supported are:

*ACCESS <files> [L]
*COMPACT <drive>
*COPY <sdrive> <ddrive> <files>
*DELETE <file>
*DIR <dir>
*DRIVE <drive>
*ENABLE
*INFO <files>
*LIB <lib>
*RENAME <from> <to>
*TITLE <name>
*DISC
*DISK
*BUILD <file>
*DUMP <file>
*LIST <file>
*TYPE <file>

*BACKUP, *DESTROY and *WIPE are not supported — it is easier to perform this sort of operation from the Amiga CLI (or CLIMate etc), then fix up the DFS directory using *REMAKE. Individual DFS Drawers can be copied from disk to disk by dragging the "disk drive" Icons.

THE EMULATOR non-standard DFS commands are as follows:

*ASSIGN <drive> [<path>]
*BOOT [<drive>]
*MAKE <drive>
*REMAKE <drive>

Note that the characters $ and / have special meanings to AmigaDOS — to avoid confusing the system, these characters appear as pound signs (Amiga ALT-L) and as broken vertical bars (ALT-I) in AmigaDOS directories respectively.

Serial and Parallel Port Facilities

Calls involving the serial port and printer are routed to the Amiga serial and parallel devices. These behave in a way which is similar, but not identical to the BBC. In particular, note the following:

The Amiga serial hardware doesn't support "split" baud rates — so *FX 7 and *FX 8 are in fact equivalent in THE EMULATOR.

In order to support multi-tasking access, the serial and parallel devices are not "opened" until they are actually needed. This means a call such as *FX 138 to insert a character in the serial device buffer won't have any effect unless the device has been enabled by *FX 2,2.

THE EMULATOR printer support ignores the Amiga printer device and goes straight to the serial or parallel devices — this is to support programs which use their own printer drivers which expect to talk directly to the printer. This means that printer Preferences settings have no effect, so if you have a serial port printer you must inform the emulator explicitly, using *FX 5.

Calls such as *FX 21 (flush buffers), *FX 138 (insert character into buffer) and *FX 145 (get character from buffer) may not have the same effect as the original. *FX 152 (examine buffer status) is not supported.

Import facilities

THE EMULATOR contains special commands to allow existing files to be transferred from a BBC with DFS, by means of a serial cable. PLEASE NOTE that this is intended to allow you to transfer your own files, and public domain software; using it to transfer commercial software is almost certainly a breach of a licencing agreement — in exactly the same way as making unauthorized copies by any other mechanism.

In order to transfer files to which you are legitimately entitled, you will need a serial cable to run between the BBC's RS423 and the Amiga's RS232 ports.

To make a cable you will need to connect the following :

BBC	data in	pin A	to Amiga TXD	pin 2
BBC	data out	pin B	to Amiga RXD	pin 3
BBC	0V	pin C	to Amiga system 0V	pin 7

Also connect BBC pin D to BBC pin E (CTS to RTS).

Code transfer is supported via the following commands:

*CONNECT
*SEVER
*IMPORT
*ICOPY

To connect at the default baud rate (9600 baud), ensure the lead is connected correctly, enter THE EMULATOR, and type

*CONNECT

This will prompt you to enter a command on your BBC (*FX 2,1), then press any key on the Amiga. Should the connection fail please check that the cable is secure. Connection can be terminated by pressing ESCAPE.

To connect at a different baud rate — eg. 1200 baud, type

*CONNECT 1200

then follow the instructions printed. Baud rates supported are 150, 300, 1200, 2400, 4800, 9600 and (in theory) 19200.

Use of the lower baud rates might allow *CONNECT to be used over a modem link. However, this is not recommended except over error correcting modems, as the error checking in the program itself is quite simple.

To disconnect "politely" and free the remote machine, enter

*SEVER

To disconnect "rudely" perform a BREAK (left-Amiga HELP) on THE EMULATOR. The far end will then have to be manually reset by pressing BREAK.

Having connected, it is then possible to copy files. To copy an entire disk from remote drive 1 to THE EMULATOR drive 0 use

*IMPORT 1 0

Note that this will erase the previous contents of THE EMULATOR drive 0. More generally, use

*IMPORT <remote source drive> <THE EMULATOR destination drive>

To copy files selectively, use *ICOPY. To copy files from directory B on remote drive 1 to THE EMULATOR drive 0 use

ICOPY 1 0 B.

More generally, the syntax is

*ICOPY <source> <destination> <files>

where <files> is an Ambiguous File Specification as used by *COPY.

IMPORT or ICOPY can be terminated politely by pressing ESC, which will cause termination after the current file has been completed, or rudely by pressing BREAK (left-Amiga HELP), which will terminate immediately. In the latter case, it will be necessary to reset the remote machine (press BREAK) then reconnect before trying again.

NOTE that IMPORT and ICOPY are designed to work with standard DFS on the remote machine. Non-standard systems are not guaranteed to work correctly.

Boot Facilities

If you are using the system's default ASSIGNs, then having imported a disk to drive 0, you can then "boot" the disk by the special command

*BOOT 0

Alternatively, you can both invoke THE EMULATOR and autoboot the disk by double-clicking on the corresponding "5 1/4 disk" icon on the Amiga Workbench. This will start THE EMULATOR and boot the disk. As soon as the application is finished — at the point where it would return to the BASIC '>' prompt — THE EMULATOR will exit back to the Amiga Workbench.

Tech Note — More about Booting

Readers reasonably familiar with the Amiga may appreciate the following additional information about booting:

The "5 1/4 disk" Icons are Amiga project Icons, set with stack size 16000, and Default Tool = THE EMULATOR. You can use IconEd, or various public domain Icon edit tools to create custom imagery for boot icons for particular programs.

The effect of double clicking on a '5 1/4 disk' icon "Boot1" (or whatever) is in fact to force THE EMULATOR automatically to AExec file "Boot1" on entry. The default contents of this file are:

```
*FORCECLOSE
*ASSIGN 0 DRIVE1
*BOOT 0
```

but you can change this for special purposes using an editor like Emacs. The trick of forcing THE EMULATOR to close down instead of returning to the BASIC '>' prompt (or CLI environment '*' prompt) is achieved using the special command

```
*FORCECLOSE
```

Note that if a *FORCECLOSE has been performed, then an ordinary BREAK (LEFT-AMIGA-HELP) will exit THE EMULATOR. However, a hard break (CTRL-LEFT-AMIGA-HELP) will override *FORCECLOSE, and stay in the system.

It is also possible to run THE EMULATOR from the Amiga CLI environment, using a command such as

```
THE EMULATOR Boot0
```

where "Boot0" is the name of a file to AExec on entry. NOTE that in order to do this, you should set the stack to at least 16000 using the STACK command. Readers familiar with the Amiga's s:start-up-sequence will realize that it is therefore possible to create a "turnkey" disk, that will automatically boot up THE EMULATOR application after a disk is inserted on the power-up prompt for Workbench.

BBC BASIC

THE EMULATOR BASIC is a new version of BBC BASIC, coded in 68000 assembler. The default value HIMEM is &F800, which is also the value of OSLIWM. The default value of PAGE is &1000, and the minimum allowed is &0800, which is the value of OSHiWM. This gives around 60kbytes of BASIC workspace.

Keywords supported are as follows:

ABS	ACS	ADVAL	AND
ASC	ASN	ATN	AUTO
BGET	BPUT	CALL	CHAIN
CHR$	CLEAR	CLG	CLOSE
CLS	COLOUR	COS	COUNT
DATA	DEF	DEG	DELETE
DIM	DIV	DRAW	ELSE
END	ENDPROC	ENVELOPE	EOF
EOR	ERL	ERR	ERROR
EVAL	EXP	EXT	FALSE
FN	FOR	GCOL	GET
GET$	GOSUB	GOTO	HIMEM
IF	INKEY	INKEY$	INPUT
INSTR(	INT	LEFT$(	LEN
LET	LINE	LIST	LN
LOAD	LOCAL	LOG	LOMEM
MID$(	MOD	MODE	MOVE
NEW	NEXT	NOT	OFF
OLD	ON	OPENIN	OPENOUT
OPENUP	OR	OSCLI	PAGE
PAGE	PI	PLOT	POINT(
POS	PRINT	PROC	PTR
RAD	READ	REM	RENUMBER
REPEAT	REPORT	RESTORE	RETURN
RIGHT$(	RND	RUN	SAVE
SGN	SIN	SOUND	SPC
SQR	STEP	STOP	STR$
STRING$(	TAB(	TAN	THEN
TIME	TO	TRACE	TRUE
UNTIL	USR	VAL	VDU
VPOS	WIDTH		

All THE EMULATOR * commands, including *FX, are also available from BASIC.

Some areas of special consideration are:

Floating Point Precision — THE EMULATOR uses the Amiga's "fast floating point" maths routines — these are VERY fast, but work to only four bytes precision, as opposed to five bytes on the BBC. The default value of @% is therefore set to &060A as opposed to &090A, giving six rather than nine decimal places of accuracy.

CALL with parameters — The simple cases of CALL (eg. CALL P%, CALL &3000) work fine; more unusual cases where CALL is

passed a list of variables are not supported, and give an error message "Not implemented".

ENVELOPE — Passed to the OS, but not implemented at present.

ADVAL command — always returns zero.

Negative INKEY — always returns zero.

GCOL modes 1 to 3 — not supported by the OS, except for text at graphics.

6502 Monitor

A simple 6502 monitor is included, loosely based on the old PET monitors like Supermon(!) Entry is via *MONITOR; commands are

*	execute CLI command
Q	exit monitor
M	display memory
.	alter memory
R	display registers
;	alter registers
D	disassemble
G	go

Commands are typically followed by one or more hex parameters, eg.

M 0000 0100	display page zero
M 0200	display block from $0200
M	display next block
D 8000 8020	disassemble from $8000 to $8020
D A000	disassemble next block from $A000
D	disassemble next block
G 8000	execute from $8000
G	execute from current PC

'.' and ';' are designed so that register and memory contents can be changed by cursoring up, then using the copy key from the start of the line in question.

Operating System

Calls currently supported under the 6502 emulator are as follows:

GSINIT	— Fully supported
GSREAD	— Fully supported
OSFIND	— All DFS compatible options
OSGBPB	— All DFS compatible options
OSBPUT	— Fully supported
OSBGET	— Fully supported
OSARGS	— All DFS compatible options
OSFILE	— All DFS compatible options
OSRDCH	— Fully supported, including redirection
OSASCI	— Fully supported
OSNEWL	— Fully supported
OSWRCH	— Fully supported, including redirection
OSWORD	— All calls except ENVELOPE supported
OSBYTE	— Most documented calls supported
OSCLI	— Fully supported

OSBYTE calls supported are as follows:

OSBYTE &00 Identify operating system version
OSBYTE &01 Read/write the user flag
OSBYTE &02 Select input stream
OSBYTE &03 Select output stream
OSBYTE &04 Enable/disable cursor editing
OSBYTE &05 Select printer destination
OSBYTE &06 Set character ignored by printer
OSBYTE &07 Set RS423 baud rate
OSBYTE &08 Set RS423 baud rate
OSBYTE &09 Set duration of mark state of flashing colours
OSBYTE &0A Set duration of space state of flashing colours
OSBYTE &0B Set auto-repeat delay
OSBYTE &0C Set auto-repeat speed
OSBYTE &0F Flush selected buffer class
OSBYTE &12 Reset soft keys
OSBYTE &13 Wait for vertical sync
OSBYTE &14 Explode soft character RAM allocation
OSBYTE &15 Flush selected buffer
OSBYTE &77 Close any SPOOL or EXEC files
OSBYTE &7C Clear ESCAPE condition
OSBYTE &7D Set ESCAPE condition
OSBYTE &7E Acknowledge ESCAPE condition
OSBYTE &7F Check for end-of-file
OSBYTE &81 Read key with time limit — not negative values
OSBYTE &82 Read machine high order address
OSBYTE &83 Read top of operating system RAM address (OSHHWM)
OSBYTE &84 Read bottom of display RAM address (OSLWM)
OSBYTE &85 Read bottom of display RAM address for a specified mode
OSBYTE &86 Read text cursor position (POS and VPOS)
OSBYTE &87 Read character at text cursor position
OSBYTE &88 Execute code indirected via USERV
OSBYTE &8A Insert character into buffer
OSBYTE &91 Get character from buffer
OSBYTE &D3 Read/write BELL channel
OSBYTE &D4 Read/write BELL amplitude/envelope
OSBYTE &D5 Read/write BELL pitch
OSBYTE &D6 Read/write BELL duration
OSBYTE &D9 Read/write lines since last page halt
OSBYTE &DA Zero number of items in VDU queue

OSBYTE &DC Read/write ESCAPE character
OSBYTE &E1 Read/write function key status
OSBYTE &E2 Read/write SHIFT+function key status
OSBYTE &E3 Read/write CTRL+function key status
OSBYTE &E4 Read/write CTRL+SHIFT+function key status
OSBYTE &E5 Read/write status of ESCAPE key
OSBYTE &E6 Read/write ESCAPE effects
OSBYTE &FD Read/write hard/soft break

OSWORD calls supported are as follows:

OSWORD &00 Read line from input stream to memory
OSWORD &01 Read system clock
OSWORD &02 Write system clock
OSWORD &03 Read interval timer
OSWORD &04 Write interval timer
OSWORD &07 Generate a sound
OSWORD &09 Read pixel colour
OSWORD &0A Read character definition
OSWORD &0B Read the VDU palette
OSWORD &0C Write the VDU palette
OSWORD &0D Read current/previous graphics cursor position

Credits for THE EMULATOR are as follows: Rahman Haleem and Peter Talbot of CBM (UK) made it happen; Mike Bolley and Dave Parkinson of Ariadne Software made it work and the Amiga made it possible.

Thanks to CHRIS for the Icons, to HANAFI for moral support, and to HARRY for dinner at Gennaros. . . .

THE EMULATOR

THE EMULATOR enables you to write/run software compatible with a BBC Model B with BASIC 2 and DFS. It runs as a PROCESS on its own screen within the Amiga multi-tasking operating system; thus you can run it AT THE SAME TIME as other Amiga programs like editors and Workbench accessories, or tasks such as PC applications (with optional XT Bridgeboard), with easy transfer of data between the different environments. You can even run several copies of THE EMULATOR simultaneously if you so wish.

THE EMULATOR runs pure BBC Basic programs faster than the original micro, while pure 6502 emulation is slower (than the original). Most educational programs combine Basic, machine code and OS calls and run at a respectable speed. More information is available in the document file on the disk.

Components of THE EMULATOR

Principal components of THE EMULATOR software are as follows:

Beebos: A "bridge" from the BBC operating system through to the Amiga graphics libraries, maths libraries, etc.

DFS: "Disk furtling system" — a bridge from DFS through to AmigaDOS.

BASIC: A new implementation of BBC BASIC — coded from scratch in 68000 assembler, with a high degree of compatibility with BBC BASIC 2. This is a very fast implementation.

6502: An emulation of the 6502 chip, written in 68000 assembler — enables programs which call assembler subroutines to work properly, though not as fast as the original.

Other elements in the software are a "CLI environment" entered by *GO, and a simple 6502 monitor, entered by *MONITOR.

THE EMULATOR Capabilities

This version of THE EMULATOR supports the following:

Graphics modes 0-7 are all supported.

VDU drivers: Most commonly used ones supported.

Operating system calls: Most commonly used calls supported

GCOL modes 1 to 3 not supported.

THE EMULATOR will support software written to the original specifications of the BBC micro but is not guaranteed to run programs not complying with these specifications (as with some games).

Communications

THE EMULATOR allows communication between its environment and the Amiga environment through *ASPOOL and *AEXEC. Communication between two emulators running on the same machine is also possible.

Transferring your programs from your BBC micro to THE EMULATOR is easily achievable through the special commands *CONNECT, *IMPORT and *ICOPY, but this requires a serial cable.