DATEL ELECTRONICS

AMIGA ACTION REPLAY

INSTRUCTION MANUAL

Amiga Action Replay

INSTRUCTION MANUAL

IMPORTANT
READ THIS FIRST

The Amiga Action Replay is available in many countries of the world and therefore has several keyboard modes. Please check that your cartridge is in the mode for your country i.e. American / UK or German. After pressing the freeze button then press F9 key to change between the two and select the mode for your country. Now either continue within the menu or press X to exit. The cartridge will stay in the chosen mode until you next power off.

WARNING
1988 COPYRIGHT ACT
Datel Electronics neither condones or authorises the use of it's products for the reproduction of copyright material. The back-up facilities of this product are designed to reproduce only software such as public domain material, the users own programs or software where permission to make a back-up has been clearly given.
It is illegal to make copies, even for your own use, of copyright material, without the expressed permission of the copyright owner, or their licencee.

CONTENTS

1 INSTALLATION

BEFORE PLUGGING OR UN-PLUGGING ANYTHING INTO YOUR AMIGA, ALWAYS MAKE SURE THAT THE POWER TO THE COMPUTER IS SWITCHED OFF. FAILURE TO OBSERVE THIS PRECAUTION WILL RESULT IN DAMAGE TO YOUR SYSTEM.

The Action Replay cartridge fits into the expansion socket on the left hand side on the Amiga 500. This is situated underneath a small removeable cover. This cover is easily removed and should then be kept in a safe place.

On the Amiga 1000 the expansion port is situated on the right hand side of the computer. Also remember that you must first load Kickstart.

OPERATION

The Action Replay interface has only a few, easy to use controls.

Slow Motion

The slow motion function is brought into play by moving the small flick switch into the "on" position. The amount of effect, i.e. the factor by which the processor is slowed, is controlled by the small knob adjacent to the switch. The "active" LED, as the name suggests, indicates that the "slowmo" has been activated.

The results from using the slowmo can be both interesting and varied depending upon the construction of the program that you are operating on. In most cases the gradation of speed will be nice and smooth without any disruption. In others, the movement can become quite jerky and sporadic especially at slower speeds.

It is probably best not to use the slowmo function when loading or saving from the disk drive, since some commercial programs use their own fast load routines which will be adversely affected by interrupting processor time. In any event, it is unlikely that you would wish to slow down either of these operations.

2 FREEZER

Freezer Operation

The cartridge also features a freezer button which, when pressed, will completely halt the program in operation and take you directly into the Action Replay main function section. The freeze function will work on the majority of programs but you may find a small number of programs that cannot be frozen due to their construction.

Once the button is pressed, you will be presented with a blue screen showing the Datel copyright message and below this is the flashing cursor prompt. At this stage, you have available a huge range of features designed to allow you to examine, manipulate and even save out the program presently in memory.

Pressing the HELP key at this stage will produce an on-screen list of all the commands available. We will briefly explain each of the functions, though not necessarily in the order in which they appear on screen. Using the SHIFT key will pause the listing.

COMMANDS.

F1	Clears the actual picture.
F2	Puts the cursor in the top lefthand corner.
F9	Keyboard setting USA/UK or German.
F10	Switch screens
HELP	Short Help listing. (use shift to pause)
DEL	Clears the character at the cursor position.
BACKSP	Clears the character in front of the cursor position.
TAB	Inserts a space at the cursor position.
CURSOR KEYS	Move the cursor in any direction.
ESCAPE	Escape from running the Action Replay command.
X	Will return you to the program in memory at its frozen point to resume.
XX	Completely switches out the cartridge hardware.

If you completey switch out the hardware (XX), then the only way to return to Action Replay is by a complete hardware reset.

3 SAVING THE PROGRAM

Saving the Program in Memory to Disk.

Since Action Replay uses its own special FDOS format in order to increase disk speed and disk capacity, it is necessary to first of all produce a disk with FDOS format. It should be noted that like the ordinary format command, the FDOS format will destroy any existing data on the disk you are about to format.

SF "string"

This will format the disk and give it the label named by the string. You will be asked "Do you wish to continue?", at which point you should answer "Y" <RETURN> to continue or "N" <RETURN> if you wish to abort the format.

It is not unusual for the format command to take as little as 3 seconds to complete.

SA"string"

This command saves the current program in memory under the name defined by "string "to the current disk drive selected. Before saving to a pre-formatted FDOS diskette, the program will be crunched to save disk space. This process is indicated by colour flashing of the screen. Be sure that you have enough disk space for the program to fit. If you only have 512K of memory then this will require 70-80 half tracks free on your FDOS disk. If you have 512K expansion memory, you will need up to 140 half tracks free.

If you wish to make space on an already formatted FDOS disk, you can use the delete command to delete any unwanted programs on the disk.

SD "string"

This will delete the program defined by the name string.

SM "string", Start End

This command allows you to save just part of the computer memory to an FDOS disk. Again the name will be given as denoted by the string. The block of memory saved must be a single area without interruption and its length must not exceed 940KB.

4 RELOADING A PROGRAM

SP"string" (,screen number(screen height))

This command saves the current picture under the name denoted by string to the current disk drive. The picture is saved dependent upon the screen number which has been notified to it. If no number is notified, then the first picture is saved. If the picture height designated isn't correctly recognised, then Action Replay will choose the default value. By using the P command (see later) you can check the parameters for the picture displayed and use these within the SP command. Many pictures shown within programs are in fact dual pictures overlaid to produce the desired effect. In these cases these should be cut and saved as two separate pictures, P1 and P2.

SS"string",channel list

This saves the current sound sample designated to the channel selected under the name notified by string. The Amiga has four such sound channels and each sample can be first heard with the H command (see later).

After any of the above commands, Action Replay will show "DISK OK". You can now return to the frozen program with the X <RETURN> command.

Reloading a frozen program

To reload a frozen program, first enter the Action Replay by pressing the freeze button. Now insert the FDOS disk that contains the required program into the current drive. Use the command LR to load and execute the frozen program.

LR"name"<RETURN>

Obviously you should use the name given to the program when you saved it, instead of "name". If you are not sure of the name you used, then use the LD command to read the directory of the currently inserted FDOS disk.

5 TRAINER

LD <RETURN>

If you wish to have the program you have saved to load independently of the Action Replay hardware, then it will need to be converted from an FDOS file to a conventional Amiga DOS file. For details see the FDOS support disk section.

LM "string",address

This loads a program notified by the name "string" but this time at a particular address.

LA"string"

This will load the frozen program notified by the name "string" but this time the program will not run automatically. To run the program, you would use the X command to return to it and execute it. This command is useful in cases where you may wish to reload a program for examination but not necessarily to run it such as programs that load extra files. See hints and tips later.

T command - Trainer

a) TS value
b) T value
c) TX

Example (a) starts the train mode and searches for the value notified. The notified value represents a counter value for which you are searching. With the TS/T command, you can get the address where the value for which you searched is located. With this information you can use the M command (see later) to increment or decrement this value in order to produce infinite lives etc.

5 TRAINER

Example (b) continues the trainer program with the notified value, for example you could enter a second value as the number of lives currently in your program decrease, in this way the trainer mode could deliver to you a more accurate address for the location containing lives, since it has two values to work upon, i.e. a value that say was 5 but is now 3 is more likely to be that of the lives parameter. If the trainer mode was not successful, then it will report as such. In this case you can either re-start the train mode (example A) or use the TX command to exit the trainer.

Example (c) TX will exit the train mode.

```
SAMPLE:

TS!3
FIRST TRAIN PASS
CHANGE COUNTVALUE NEXT TIME
SEARCH UPTO :005444
TRAIN MODE ACTIVE  READY
```

Now use the X command to return to the program and play until 1 life has been lost. Then freeze the program again and this time use the T command to enter the new value of remaining lives, i.e. 2:

```
T!2
SEARCHED UPTO :080000
00014428 NEXT LINE TRAIN MODE ACTIVE
READY
```

In some cases, the trainer will deliver a number of possible addresses and you should make a note of each of them, since at this stage you have only come up with addresses which had a value of 3 last time that you checked and now have a value of 2. This of course could still be a large number of addresses. At this stage, you can resume the program and lose one more life and then re-freeze the program and, using the T command, look for locations which now contain a value of 1. As you will see, the number of addresses will reduce and may in fact have now become just one. In this case, you can be almost sure that you have found the correct location for the address which contains the variable for the number of lives.

To change such an address, we would use the M command (see later) and in our example we would say

```
M 14428 <RETURN>
```

The screen will now display the 8 bytes contained in address location 14428 and the subsequent seven locations. Position the cursor over the value for the first address, i.e. 14428 and you can now insert a new binary hex value and thus give yourself any number of lives (providing of course you have in fact found the correct location). Don't get carried away with the number of lives since some programs will not accept high values in this location and will hang up. You can experiment to see just how high a value you can insert. Now type X <RETURN> and you will return to the frozen program where you can check to see if you have had the desired effect. Note that the new value that you have inserted may not show until you next lose a life, move to new level or some other interval.

Obviously any value for lives, ammunition, or fuel can be treated in this way and the value can be either an increasing or decreasing one.

If you don't get the required results from the trainer mode, then see the Hints & Tips section later.

TF command - Absolute Trainer

a) TF address
b) TFD address

Example (a) shows the addresses of menumonic commands where Command "SUBQ.X£X,ADDRESS" or "SUBI.X#X,ADDRESS" is used to subtract the contents of the notified address (look also at T command for more help).

Example (b) is like (a) but additional mnemonics are removed from the program.

5 TRAINER

TM command - Editing the Address Marker

a) TM
b) TMS address
c) TMD address

Example (a) shows the current memory marker.

Example (b) puts a memory marker in the next inserted value. There are 10 memory markers possible in a sort of notepad so that you can write all important addresses into it. They are saved by the SA command and are reloaded by using LA/LR commands (see later).

Example (c) will delete the memory marker as pointed to by the notified address. If it doesn't exist at that address, the message "Memory counter not found" will appear.

6 MONITOR COMMANDS

MACHINE CODE MONITOR

The following section is concerned with the Machine Code Monitor, Assembler and Dis-assembler functions of the cartridge together with other toolkit type routines. If you are not conversant with machine language, then this section may not be of interest to you. Our explanation of the monitor and assembler/ dis-assembler commands are brief because persons familiar with these tools will not need a more full explanation.

Digital Input Format

Action Replay accepts number input as hexadecimal, decimal and binary. The default for numbers input without qualification is hexadecimal.

To express a number as hexadecimal, then prefix the dollar sign ("$") to it.

To express a number as decimal, then prefix exclamation mark ("!") to it.

To express a number as binary, then prefix with percent sign ("%").

The leading zeros before any address and number input can be cancelled.

Sample $00052340 = $52340.

The calculation commands will automatically cope with mixed type number inputs to give a hex result.

Sample ? $1000 - $100 + !256 -%1 = $FFF

A command Assemble

Syntax A Address

Calls the direct assembler. It will show the chosen address. Now you can make the mnemonic input in the usual way of programming. There is one exception - the register list of the MOVEM command can also be written in a different way if the normal length of the inputting doesn't fit.

For the data register list, write "D" followed by 8 zeros (00000000) or 8 ones (11111111) to mark the chosen data register.

D0/D2-D4 = D10111000
D6/D7 = D00000011

For the address register list, write "A" followed by 8 zeros or 8 ones (00000000) or (11111111).

A2/A4 = A00101000
A0-A6 = A11111110

The combination of the data register list and address register list is very easy while you write both lists with empty signs one after the other.

D0-D2/A4 = D11100000A00001000

Press the <RETURN> key without input to exit the direct assembler.

Sample:

```
A 70000
;070000 ADDQ.L £1,D0
;070002 RTS
;070004
```

B Command - Set/Clear Breakpoint

Syntax:
a) B
b) BS Address
c) BD Address

a) Shows the current setting of breakpoints.

b) Puts a breakpoint at the chosen address. If no breakpoint setting is possible, a message will appear "NO FREE ENTRY". You can set up to five breakpoints at a time.

c) Clears a breakpoint at the chosen address. If a breakpoint was not set, a message "BREAKPOINT NOT FOUND" appears.

The breakpoints are only activated by leaving Action Replay. Only breakpoints that were set at the current PC are not active because when you leave Action Replay it would recall Action Replay immediately.

If the program being run reaches a set breakpoint, the program will be automatically interrupted and the module editor will be activated. Otherwise a message appears "BREAKPOINT RAISED AT" and the address of the breakpoint will be shown.

Sample:

```
BS 4348
BREAKPOINT INSERTED!
B
BREAKPOINTS:
004348
```

C Command - Copper-Ass/Disassembler

Disassembled, the Copperlist at the given address. If cipher 1 or 2 is given instead of the address, the first address or second address is in use, for instance.

The results of this list are directly processed by the Copper Assembler. You can edit the Copperlist directly.

What is COPPER? The Copper is an Amiga special chip that works like a processor.

The Copper normally is used for screen picture construction.

What is a Copperlist? The Copperlist is a Copper program (see below).

6 MONITOR COMMANDS

Syntax of COPPER Commands:

data: = data that has to be written
adr : = offset chip registers
WAIT (x,y,xmask,ymask,bfd)
x: = vertical beam position
y: = horizontal beam position
xmask:= vertical mask register
ymask:= horizontal mask register
bfd: = Blitter finished disable bit
SKIP (x,y,xmask,ymask,bfd)
parameter looks for WAIT command.

Sample - changing background colour of workbench:

```
c 2
~00004436 WAIT ($0,$2f,$1fc,$7f,$1)
~0000443A MOVE £$005A,$180
```

to change

After exiting from Action Replay you will see the changing background colour.

D Command Dissasemble

Syntax: D (0 address)

Disassembled the 68000 machine program at the given address. If no address is given, it will continue the disassembling at the last line (previous D command). Should the first D command be given without an address, the address will be accepted where the program would continue before exiting Action Replay.

When you give the cipher 0 instead of an address, the Program Counter is put in.

Sample:
```
D 0
~C01234 SUBQ.W £1,$000500
~C0123A BEQ C01200.S
```

G Command - Jump to Address

Syntax: G(address)

Continue the interrupted program at the given address.

If no address is given, the current Program Counter is used and a normal exit of Action Replay.

Use this command very carefully because an invalid address will cause the interrupted program to hang.

Sample:
```
G FC00D2
```
Jump in Kickstart Reset routine = softreset

H Command - Replay Current Sound

Syntax: H Channel

Replays the current sound of the notified channel which was playing in the last run program.

The sound can only be stopped by pressing the ESC key. The channel number should be between 1 and 4.

Sample:
```
H 3
```

I Command - Copy Memory Area

Syntax: I BEGIN END TARGET

Copies the given memory area specified to a target area.

Sample: I 100 200 1000
copies $100 - $1FF (= 100 bytes) to $1000

J Command - Editing Sprites

Syntax: J Spritenr1 Adr1 (Spritenr2 Adr2)

Declares a sprite the number spritenr1 or the sprite that is stored in memory at Adr1 and shows it on the screen as an editable listing, where the sprite colours are represented as ciphers 0 to 3.

If a second sprite number of address is notified, the sprite will be notified as ATTACHED SPRITE and its colours will be notified by ciphers 1 to 9 and A to F. In this case the colour 0 indicates a transparent colour.

Sample: originates from workbench:

```
J 0
~001480 1111110000000000
~001482 1222221000000000
~001484 1333321000000000
~001486 1333210000000000
~001488 1333321000000000
~00148A 1331332100000000
~00148C 0110133210000000
~00148E 0000013321000000
~001490 0000001332100000
~001492 0000000131000000
~001494 0000000010000000
~001496 0000000000000000
```

You can edit the mouse cursor as you want to.

M Command - Show and Edit Memory Area

Syntax: M Address

Shows memory from notified address. You can edit directly in the listing. Don't forget to press <RETURN> when memory addresses are changed.

Sample:

```
M 234
:00000234 11 20 2A 00 10 FF FF FD ..........
```

N Command - Shows Memory as Text

Syntax:

a) N Address
b) NO Constant
c) NQ Address

a) Shows the memory notified addresses as text. The text can be accessed if you add a constant to every memory Byte, previously notified using the NO Command.

The results will be interpreted as ASCII code.

If no constant is specified, the constant will default to 0, meaning normally the memory will be interpreted as ASCII-TEXT.

b) The notified constant used in the N command can be set. A value can also be used that was set by the FR command as Offset.

c) Shows memory at notified addresses as TEXT. Unprintable characters are ???
The screen list can be halted by <SHIFT> and quit <EXIT> by <ESC>.

At the end of the NQ Command, the last examined memory area will be shown.

Sample:

```
M 1234
:001234 41 42 43 44 00 00 00 00 ABCD......
N 1234
;001234 ABCD................................................
NO 1
N 1
N 1234
;001234 BCDE.................................................
```

6 MONITOR COMMANDS

O Command - Fill Memory

Syntax: O string, Start End
Fill the entire memory area specified with the specified string. If the string is smaller than the memory area it is to fill, then the string will be repeated in memory.

```
Sample:     O "ACTION REPLAY", 0 80000
        N 0
        ;000000 ACTION REPLAYACTION REPLAYACTION REPLAY
        O 0,    0 8
        M 0
        :000000 00 00 00 00 00 00 00 00 ...............
```

P Command - Show/Set Screen

Syntax: a) P (Screen number)
b) PS
c) PC (Screen number)

a) Represents the current screen. Sometimes a screen is built up of more than one screen, like overlapping workbench screens. These can only be shown one by one. The screen you would like to see is indicated by the screen number. If you leave the screen number blank, the first screen will be shown. During the screen show, you can set the lowest bar by the left and right mouse buttons.

When you leave the screen show by the ESC key, the height of the notified picture will be shown. This value should be notified too when you save a picture (SP Command). When you use this setting, only this picture is saved and no other data.

Normally Action Replay selects the correct height of the picture. Only in exceptional circumstances do you have to correct it yourself.

b) When you exit Action Replay, the screen parameter will be set at the notified values. This is only necessary when Action Replay doesn't recognise the way the screen parameter was set. You will recognise it by shivering graphics in the program when you exit Action Replay.

6 MONITOR COMMANDS

With the PS Command, standard or special parameters can be put in. You can experiment with it and the program that is running will not be affected, only the picture building is manipulated.

Only use the PS command after the X/G Command when the program is continuing normally and where, for example, the picture is too long or move too far up.

INPUT and OUTPUT of PS-COMMAND
VER.STRT = Screens starting line vertical (Lath-ray) normally = !41
HOR.STRT = Screens starting line horizontal (lath-ray)
WIDTH = Size of displayed screen horizontal
normally x = !320 or x = !640
HEIGHT = Size of displayed screen vertical
normally y = !200 or y = !256

c) Builds the current screen. Some screens are built up by more than one overlapping picture, like the workbench screens. These can only be shown one by one.

The screen which you want to see can be selected by screen numbers. If you leave the screen number blank, the first picture will be displayed. During the screen show, you can select a cut area with your mouse, where height and width are displayed in screenpoints after <ESC>.

By pressing the left mouse key, you can move the left top border of the cut area. Pressing the right mouse key you can move the right lower border. With this function you can measure for example the powerbars in games to use the results with the built in Trainer, and to train energy at this time.

To get good results for the trainer, make sure that you set a screen cut areaPIXEL precisely.The result is in decimal so it can be used in the trainer (TS) & (T) inputs to try and find the address that contains the value for the powerbar.

Sample:

```
P 1
PICTUREHEIGHT: !123
READY

PC 1
WIDTH : X + !123
HEIGHT: Y = !010
READY
```

Q Command - Show/Edit CPU Register

Syntax: R (register value)

If notified, it sets the CPU register to the notified value and lists several registers on the screen, e.g.

Data register: D0-D7
Address register : A0-A7
User Stack Pointer: USP
Status Register: SR
Program Counter: PC
Flags in SR: FT, FS, FV, FC, FZ or FO
Interrupt Mask: FI

Sample:

```
R D0 1234
D0 = 00001234 D1 = 00000000
D2 = 11111111 D3 = 22222222
D4 = 33333333 D5 = 44444444
D6 = 77777777 D7 = 88888888
A0 = 00000000 A1 = 00000000
A2 = 00000000 A3 = 00000000
A4 = 00000000 A5 = 00000000
A6 = 00000000 A7 = 000014a6
PC = FC00D@ USP = 00123430 SR = 0010
T=0 S=0 I=000 X=0 N=1 Z=0 V=0 C=0
R PC FC00D@ (The PC is set to Reset routine)
```

6 MONITOR COMMANDS

U Commands

The U commands are very useful since they will display all of the important system status registers together with other useful information such as chip memory, disk drives, current tasks etc etc.

Syntax a) U (picturenumber)   b) UD
c) UI   d) UL
e) UP   f) UR
g) UT   h) UM

a) Shows the important system parameters which fix the screen building. There are also overlapping screens like workbench screens. The system parameters cover only a few single screen, specified by a picture number. If you leave out the picture number, the first picture is to be analysed. The colours and addresses of BITplanes of the the current screen are shown and also sprite and audio data and the current position of drive read/write heads.The read/write head position is very useful to identify the track on which a program you are loading has stalled.

The outputs of U commands

DIWSTRT	= Upper left corner of the display windows
DIWSTOP	= Lower right corner of the display windows
DDFSTRT	= Start Bitplane-DMAs (x-position)
DDFSTOP	= End Bitplane-DMAs (x-position)
BITPLANES	= Addresses of Bitplanes (1-6 Bitplanes)
COLORS	= Colour register 00 to colour register 31
SPRITES	= Addresses of sprite data (8 sprites)
BPLCON0	= Bitplane Control register 0
BPLCON1	= Bitplane Control register 1 (scroll value)
BPLCON2	= Bitplane Control register 2 (priority)
DMACON	= DMA Control register
INTREQ	= Interrupt request
BPLMOD1	= Bitplane Module for odd Bitplanes
BPLMOD2	= Bitplane Module for even Bitplanes
BLTCON0	= Blitter control register 0
BLTCON1	= Blitter control register 1
DSKSYNC	= Disk Synchronisation sample
CLXCON	= Sprite collide control register

INTENA	= Interrupt enable
ADKCON	= Audio, Disk and UART control register
TDF0	= Track display for DF0 (!FFF = not found yet)
TDF1	= Track display for DF1 (!FFF = not found yet)
AUDADR	= Audio addresses
AUDLEN	= Length of audio data
AUDPER	= Period time
AUDVOL	= Volume

b) Shows a list of current devices.

c) Shows a list of interrupts.

d) Shows a list of current libraries.

e) Shows a list of current ports.

f) Shows a list of current resources.

g) Shows the current task, subdivided into Running, Ready and Waiting Task. The current program can be found as the first in the tasks list (RUNNING TASK).

h) Shows free memory.

If the current program doesn't use the Amiga Kernal System, several values can't be calculated. An error message will appear "EXECBASE NOT VALID!"

Sample
```
UM
CHIPMEM   FASTMEM TOTALMEM
!010010   !000200   !010210
```

V Command - Compare Memory Area

Syntax V Start End Target

Compares memory location from start to end with the memory area of the target address. Differences are displayed with their addresses in the target memory area.

Sample

```
M 100
:000100 12 11 01 12 12 01 00 00 ..........
V 1000 103 103
000104
READY.
```

W Command - Show/Edit CIA Register

Syntax W (register number)
Shows the contents of the notified registers of both CIA's on screen. The contents of the registers can be changed directly in the list. If no register number is specified, CIA register pair $0 is displayed. The first binary value shown is always the contents of CIA-A, the second CIA-B.

What is a CIA?

By the CIA (Complex Interface Adaptor) it actually two 8520 IC's. The job of the CIA's is to handle the disk signals and more free programmable timers are available.

Sample

```
W
'0 %11111100 %11111111
             ^
set to 1
If exit Action Replay the Power light goes out.
```

Y Command - Show/Edit Memory Area Binary

Syntax
a) Y Addresses
b) YS (Byte number)

a) Shows the memory binary at the specified address and shows as much as wanted with the YS Command. It can be edited directly in the list. Don't forget <RETURN>.

b) Set the Byte number for the Y command on the specified value. You can choose between 1 and 4. If a byte number is specified, the current set value is displayed directly. The bytes are not displayed but the number of bytes in bits (Byte=8Bit).

Sample

```
Y 100
.000100 %0011001100110011
YS 1
Y 100
.000100 %00110011
.000101 %00110011
YS
CURRENT BIT WIDTH: !08
```

Z Command - Set Options

Syntax

```
a) Z
b) ZD (0/1)
c) ZM (0/1)
d) ZV (0/1)
e) ZF (0/1)
```

a) Shows the current options such as current drives, reset virus test, clear memory and FastMem.

b) Sets the current drive to DF0 or DF1.

c) Activate (1) or de-activate the memory clear function by Reset. If the memory clear function is active, the whole memory (RAM) is filled with Byte #$00, possible virus or reset free memory resident programs are cleared now.

d) Activate (1) or de-activate the reset virus test. If switched on, the Reset Virus Test recognizes a virus, the screen will blink and then the reset free virus will be de-activated.

e) Switches a connected internal memory expansion of 512KB on (1) or out (0).

Sample

```
Z
RESETVIRUSTEST
CLEARMEMORY
DISK DF1:
FAST RAM ACTIVE!
ZD 0
Z
```

RESETVIRUSTEST
CLEARMEMORY
DISK DF0:
FAST RAM ACTIVE!

? Command - Calculation

Syntax ? Value (= - * / Value) ...

Calculates the specified term without notice of "point or stripe" and displays the results in Binary, Decimal and Hexadecimal.

```
Sample        ? !10+4*2
              %000000000000000000000000011100
              !28 $1C
```

7 APPENDIX

APPENDIX - Messages after activating Action Replay

After you activate Action Replay, the following messages can appear:

1) "DANGER! RESIDENT PROGRAM ALLOCATED (VIRUS)"
A reset fixed program or data module is recognized. Usually you can say it's a virus, BUT it also could be a reset fixed RAM-d

2) "BREAKPOINT RAISED AT ADR: 12345678"
You have set a breakpoint at address $12345678 and the program has now reached address $12345678. The program is interrupted automatically and the Action Replay is now activated.

EXCEPTION: Memory areas (SM-command) are saved byte by byte in Amigados-format, because for memory areas IFF-format doesn't exist.

8 FDOS SUPPORT DISK

USING THE FDOS SUPPORT DISK

CONVERTING PROGRAMS TO STAND-ALONE

To enable the FDOS backup programs that you have created to reload without the presence of the cartridge, it is necessary to run the TOOLDISK supplied with your cartridge.

This disk will autoboot into a menu. From this menu use the F2 command and you will be asked to insert the FDOS disk that you wish to install the boot onto. It takes only a few seconds and the FDOS disk will then reload the program(s) on it without the cartridge. The FDOS disk, when booted, will now present a menu of the program(s) on it which can be selected with the mouse.

CONVERTING FILES TO IFF-FORMAT

To use drawings, sounds or the saved pictures,within music or drawing packages you must convert them from FDOS-files into Amigados, as follows:

1) Prepare an Amigados formatted disk with enough free space to save. Use the program FORMAT from your workbench as normal.

2) Boot now from the disk "TOOLDISK".

3) Inert the convertible FDOS-disk in the current drive and press F1. The files contained on the disk are listed.

4) Choose the name of the file that is to be converted (without extension name).

5) After loading the FDOS-file, insert the prepared Amigados disk in any available drive and give the filename with path (e.g. for internal drives DF0).

After this process, the Amigados-disk contains the FDOS-file, in IFF-format.

9 HINTS & TIPS

HINTS & TIPS

Before saving a program, it is better to switch any memory extensions off to avoid unnecessary saving and subsequent loading of large areas of memory.

If you use the Memory Clear Option (MZ1) then the cruncher, upon freezing, will work more efficiently.

If you have different memory expansion than the normal 0.5Meg FastRam ($C00000), you will have to save the additional areas by hand because Action Replay can only save 512K of chip memory.

If you use the SM command to save a block of memory, then you have to use the LM command to re-load it.

It is advisable, because of virus danger, to leave the Reset Virus Test and the Reset Memory Function on, so that a virus has no chance to stay within the system. Remember, if a virus is recognised, it is indicated as a flickering on the screen.

In the Trainer Mode, if you don't get the results by entering the exact number of screen lives, you could try adding the life value plus 1. Remember to use the "!" for decimal values. Some counters used by programmers use the BCD number format. For example, the on-screen value would be, say, 978. In the BCD case, remember not to use any prefix such as "!".

Normally, player data such as lives, energy or score are close together in memory so if you have found the score, the chances are the addresses containing the other values are nearby.

Some counters are inverted, i.e. they go up instead of down.

The second screen picture can be selected with the HELP key and this can be used for keeping notes etc. Remember that during the Trainer Mode the second screen is not available.

Always use 2DD diskettes.

9 HINTS & TIPS

If you have an original program which is not completely memory resident, i.e. loads extra data from the disk, then it is not really suited to being saved with Action Replay. However, a full backup can be achieved in some cases by using the following method:

First copy the entire original disk by either using the workbench or a good disk copier, if you have one. Put this copy to one side. Now load the original program up to the point where you would like to freeze it and then, after pressing the Freeze button, save the program (SA) out to an FDOS disk. When you now reload your backup, use the LA command to reload the program but not to run. Now exchange the disk in the drive for the complete backup disk that you made originally. If you then use the X or G command, the program will execute and should find any support or data files required on the diskette in the drive. For obvious reasons, this will not work in all cases but may be useful where it is important to make a backup of a particular program.

10 ADDITIONAL FEATURES

A number of additional features have been added to allow the editing of the frozen picture. When 'P' has been entered you will be taken into the new Mempeeker mode. Using the list of commands herewith, the picture can be manipulated in many ways. To search for a picture use the cursor keys to scroll through memory. When the required results are achieved, then use the 'SPM' command (see manual) to save the picture in its modified form.

Pressing the Help key in this mode will superimpose a Parameter Status Table onto the picture. As you move through memory or around the picture itself this screen will display the current status of location, bitplanes, dimensions etc. This table can be repositioned on the screen using the mouse and the table will not be saved with the picture.

"Commands for Mempeeker v2.1"

a	autoplane
b	increase brightness
shift & b	decrease brightness
c	increase colorreg
d	dual playfield on
shift & d	dual playfield off
e	increase right border
shift & e	decrease right border
f	fast plane up
shift & f	fast plane down
h	hold and modify (ham) on
shift & h	hold and modify (ham) off
i	invert all colours
l	lores mode on
shift & l	hires mode on
m	increase modulo
n	decrease modulo
s	decrease left border
shift & s	increase left border
w	white helpscreen
shift & w	black helpscreen
x	decrease colorreg
0	unlock all planes
shift & 0	lock all planes
1	lock plane 1
shift & 1	unlock plane 1
2	lock plane 2
shift & 2	unlock plane 2
3	lock plane 3

10 ADDITIONAL FEATURES

shift & 3	unlock plane 3
4	lock plane 4
shift & 4	unlock plane 4
5	lock plane 5
shift & 5	unlock plane 5
6	lock plane 6
shift & 6	unlock plane 6
7	increase red color
shift & 7	decrease red color
8	increase blue color
shift & 8	decrease blue color
9	increase green color
shift & 9	decrease green color
F1	set to colors
F10	randon colors
left	rotate picture left
right	rotate picture right
up	scroll picture up
shift & up	scroll picture up fast
down	scroll picture down
shift & down	scroll picture down fast
del	hide helpscreen
left mousebutton	incease pic height
right mousebutton	decrease pic height

set helpscreen with mouse on position

esc	quit mempeeker
help	show helpscreen

to save mempeeker-picture quit mempeeker
with<esc> and save pic with 'SPM' command

A Ram tester command 'QR Start End' has been added and is simple to use e.g.:-

QR 10000 20000

will test the Ram within these parameters and will return the message 'Ram is OK'

The command 'UE' has been added also and this will produce a list of Processor Exceptions and also Trap Vectors.

PBALL/FI –

DATEL
ELECTRONICS

FENTON INDUSTRIAL ESTATE, GOVAN ROAD,
FENTON, STOKE-ON-TRENT, ST4 2RS.
TEL 0782 744707. FAX 0782 744292.