

---

# **DIALOGA V1.0**

**LOGICA**  
Copyright 1993

Logica Diagnostic for Amiga - Technical Documentation

They have contributed to the realisation of DIALOGA: Stefano Bornacin, Fabio Biasutti, Simone Tasselli, Andrea Sanavia.

Copyright 1993 Logica. All rights reserved. Reproduction, adaptation or translation of this manual is prohibited without prior written permission of Logica. Logica makes no warranty of merchantability and fitness for a particular purpose. Logica shall not be liable for any errors or for incidental, consequential, special or indirect damages resulting from possession, use or malfunction of this product.

English Version

First Edition: March 1993

Amiga and Commodore-Amiga are trademarks of Commodore-Amiga, Inc.  
DIALOGA and Logica are trademarks of Logica s.n.c.  
IHT is a trademark of IHT Gruppo Editoriale

Logica

Via Pinali, 13

33170 Pordenone ITALY

[info@logicasnc.it](mailto:info@logicasnc.it)

---

Description	Page
Premise .....	1
How to use this manual .....	3
Part 1. Introduction.....	1-1
1.1. The power of DIALOGA .....	1-3
1.2. Limits of DIALOGA .....	1-3
1.3. Components of DIALOGA .....	1-4
Part 2. Tutorial .....	2-1
2.1. System requirements .....	2-3
2.2. DIALOGA installation .....	2-3
2.3. Startup of DIALOGA .....	2-3
2.4. How to use the keyboard and the mouse .....	2-5
2.5. How to use DIALOGA.....	2-6
Part 3. Reference .....	3-1
3.1. START-UP .....	3-3
3.1.1. CPU test.....	3-5
3.1.2. Checksum .....	3-5
3.1.3. Hardware registers .....	3-5
3.1.4. Overlay test.....	3-5
3.1.5. Exception vectors.....	3-5
3.1.6. VHPOS register .....	3-5
3.1.7. CHIP RAM .....	3-6
3.1.8. RAM test.....	3-6
3.1.9. Memory expansions ranger.....	3-6
3.1.10. DMACON test .....	3-7
3.1.11. Vertical Blanking test .....	3-7
3.1.12. Blitter test.....	3-7
3.1.13. Display set-up .....	3-8
3.1.14. Keyboard Start-up test .....	3-8
3.1.15. Internal Disk Drive .....	3-8
3.2. End of START-UP .....	3-9
3.2.1. System info 1 .....	3-11
3.2.2. System info 2 .....	3-15
3.2.3. Chip test.....	3-17
3.2.4. CPU vectors test .....	3-19
3.2.5. Amiga model.....	3-23

3.3.	Analysis.....	3-25
3.3.1.	Main menu .....	3-27
3.3.2.	Keyboard test.....	3-29
3.3.3.	Cia Timer test.....	3-33
3.3.4.	Internal disk drive test .....	3-35
3.3.5.	Parallel test.....	3-41
3.3.6.	Serial test .....	3-45
3.3.7.	External disk drive port test.....	3-47
3.3.8.	Control port test .....	3-51
3.3.9.	Memory test .....	3-53
3.3.10.	Display test .....	3-59
3.3.11.	Audio test.....	3-63
3.3.12.	All in one test.....	3-65
3.3.13.	AutoTest .....	3-67
3.3.14.	About .....	3-69
3.4.	Exceptions .....	3-71
	COLOR CODES .....	4-1
	ADKCON - INTENA - DMACON .....	5-1
	KEYS CODES .....	6-1
	CONDITIONS OF USE.....	7-1

**Premise**

Logica welcomes you in the world of DIALOGA, the first diagnostic operative system for Amiga computers.

Please, in order to obtain the best results, first read carefully the instructions before you start using your new diagnostic system.

This manual assumes a certain degree of technical knowledge of Amiga computers hardware and, less important, software knowledge.

<b>CAUTION:</b> <b>THIS PRODUCT IS DESIGNED FOR USERS QUALIFIED IN AMIGA REPAIRING. THE MISUSE MAY DAMAGE THE COMPUTER.</b>
---

<b>IMPORTANT:</b> <b>THE INSTALLATION OF DIALOGA REQUIRES TO OPEN UP THE COMPUTER, THUS VOIDING THE COMMODORE WARRANTY.</b>
---



## How to use this manual

Subject of the manual is DIALOGA diagnostic system. In this manual we are going to show you how to use and how to handle the diagnostic output. This product cannot in anyway substitute the technical knowledge of the computer repairer to whom is left the responsibility of the correct use of the information obtained from DIALOGA.

This manual is designed to be used in the following way:

- Read Part 1 once to have a complete view of the product characteristics.
- Read Part 2 to gain a basic understanding of DIALOGA operations.
- Then use part 3 to find other topics in which you're interested.

To obtain the best results, please read all the manual at least once.

The manual is divided in three parts and five appendix.

- Part 1. Introduction - contains information on the characteristics, the goals and the limits of the diagnostic
- Part 2. Tutorial - describes the processes of installing the ROM card and the external hardware, shows how to use the keyboard and the mouse, guide the user to a first use of DIALOGA
- Part 3. Reference - covers all aspects of DIALOGA code contained in the ROM card and the correct interpretation of the diagnostic output
- Appendix A: Color Code - contains the color codes used by the diagnostic
- Appendix B: Tables - contains the ADCON, INTENA, INTREQ and DMACON register tables.
- Appendix C: Keys Code - contains the keys codes used in the analysis of the keyboard.
- Appendix D: Conditions of use - contains the conditions of use of this product for the user.



---

## **Part 1. Introduction**

**In this section are illustrated the power and the limits of the diagnostic.**



## **1.1. The power of DIALOGA**

DIALOGA is a powerful but easy to use diagnostic system to analyse Amiga systems.

This diagnostic system is designed to satisfy the needs of Amiga systems repairers such as a powerful tool that is easy to use, fast and thrust worth without the need of expensive equipment.

Our target is to fulfil the scarcity of useful information given by the operative system to the repairer and to create an efficient diagnostic tool that works also with a malfunctioning system but not completely compromised. In many situations the Kickstart blocks itself or reset itself in the power up phase.

The traditional diagnostic programs, loaded from disk, required a working operative system.

DIALOGA works in conditions where others fail. This is possible because it is designed as an operative system with the purpose of analysing the workings of the computer to obtain the information on the system failures needed by the computer repairer.

## **1.2. Limits of DIALOGA**

There is one set-back, DIALOGA is a diagnostic partially based on a computer program, due to this, it is requested that the system's core elements (CPU, part of the RAM, ecc.) are still working.

Anyway DIALOGA is and remain a very powerful diagnostic system, necessary in Amiga computer repairing.

### **1.3. Components of DIALOGA**

The diagnostic system is made of an external control box to connect with the ports on the backside of the computer, and a ROM card with the firmware to install internally at the place of the Kickstart.

Please check the contents of the box to be sure to have a complete diagnostic system in all his components.

The box carry:

- one ROM card
- one external control box
- one technical manual

If any of these elements is missing or is altered, please call immediately your retail dealer.

#### **Internal ROM card**

The internal ROM card has a custom chip containing the firmware, to insert in the socket of the Kickstart.

#### **External Control Box**

The external control box have on a side three cables of the parallel port, serial port and external disk drive port.

On the opposite side there is a connector of the parallel port. For future use.

On the top there are rows of led concerning the three connectors

---

## **Part 2. Tutorial**

**In this section are illustrated the basic elements to know for use your diagnostic system.**



## 2.1. System requirements

The product works with any Amiga computer model 500, 500 plus, 2000B and 2000C.

The basic system requirements are the microprocessor and the essential communication elements operating. It is also required a working game port or keyboard. It is necessary at least 90 kbytes of functioning Chip RAM for system data.

The diagnostic system is able to work without the screen display. In this case the only available information are those of the led on the computer and on the external control box.

## 2.2. DIALOGA installation

We advise to install the diagnostic on a working computer the first time that you use it, so it is possible to test and to practice with the available operating procedures in normal conditions.

- Open the computer subject to analysis, locate the Kickstart ROM, check the version part number.

Version	Part Number
1.2	315093-01
1.3	315093-02
2.0	390979-01

- Change the Kickstart Rom with the diagnostic ROM card paying attention to the right side.
- Insert the three cables present on the side of the external control box to the corresponding parallel, serial and external disk drive ports on the backside of the computer.
- Connect the keyboard and the mouse to the computer.
- Insert an empty diskette in the internal disk drive df0:
- Switch on the computer.

DIALOGA works as the original Amiga operating system normally resident in the ROM, it start to operate when the computer is turned on or is resetted.

## 2.3. Startup of DIALOGA

If the computer is seriously damaged, it may not startup because the diagnostic is partially based on software. Hence, it needs a minimum part of the hardware working.

It is possible that even if you installed and connected DIALOGA correctly, the only result is a black screen. In this case it is important to understand if the diagnostic did operate and blocked itself without any warning or didn't even start. During the startup (3.1.1 e 3.1.2) DIALOGA performs crucial tests that may fail without any warning to the user. It is possible that everything works fine and it is only the display hardware damaged. Anyway if the yellow leds of the external control box are all off and you heard the internal disk mechanic working it is sure that the diagnostic started normally.

## 2.4. How to use the keyboard and the mouse

The user interface is simple and easy to use. The user is always informed when it is needed his decision.

In general the request is displayed in this way:

```
Press a Mouse button
Press any key
Press the left mouse button
```

In order to grant a greater flexibility to user preferences, it is possible to use the mouse or the keyboard to make the selection.

DIALOGA adapt the screen messages to the availability or not of the keyboard from

```
Press any key or a mouse button
```

to

```
Press a mouse button.
```

It is possible to use to the diagnostic without the keyboard except, of course the keyboard test.

The available mouse actions are:

**quick start:** It is possible to make scroll continuous of the messages that appear on the screen in the startup phase without the user confirmation until the choice of the Amiga model subject to analysis (3.2.5). To do so press the left button during the colored strips scroll at the power up of the computer (3.1.6). If quick start is selected, when the diagnostic finds any anomalies during the startup sequence the scroll returns normal.

**quick exit:** To speed up or to exit from a selected test press together the two mouse buttons.

## 2.5. How to use DIALOGA

After the diagnostic installation, insert in the internal disk drive df0: an empty disk write enabled.

At the power up of the computer the firmware of the diagnostic starts the analysis. Check on the external control box that the green power leds of the ports are all on. If not, verify the absence of power in the ports with the power leds off. The red led switch on only for about a second.

All the other leds present on the external control box must be off, if not check the CIA 8520 chips.

The diagnostic at start-up communicates the results with a color code and by blinking of the power led on the computer.

### Startup

If on power up phase the diagnostic stops with a screen colored of:

BLUE	CPU malfunction
RED	EPROM code integrity check-up failed
PINK	CIA malfunction (Overlay)

The video test presents the screen divided in five colored stripes, red green, blue, gray and black. the colors are rotated to verify the correct working of the electronic beam.

Memory test makes appear on the screen one of these cases:

Irregular colored strips	correct working
Green with blinking led	90K contiguous non available, CHIP RAM malfunctioning
Shortly some green stripes	90K contiguous available, CHIP RAM malfunctioning
Shortly some red stripes	90K contiguous available, Address error

In case DMACON or Vertical Blanking interrupt malfunction the screen appears:

YELLOW with the led on	malfunction of the DMACON register
WHITE with the led on	malfunction of the Vblank interrupt

In this case it is impossible to proceed with the analysis without eliminating the error cause.

The screen appears brown for some seconds while the diagnostic is testing the Blitter.

After that we have one of these cases:

BLACK	Blitter working correctly
BROWN with the led on	Blitter malfunction

The screen of the diagnostic is formed by an area in the upper part of the screen with the LOGICA logos in 16 colors hires non interlaced, a second area placed in the remaining part in two colors hires non interlaced where the diagnostic messages are showed.

The screen shows the message:

The DIAGNOSTIC Kernel v1.0

Programmed by Bornacin Stefano in 1991/1992

All Rights reserved to LOGICA S.N.C

If the main screen shows up with all it's parts, the Copper and the graphics circuits in normal mode are working correctly.

If the CAPS LOCK key led blinks during the keyboard test, the keyboard is not working correctly.

**System Info 1**

The diagnostic presents a table with the results obtained from the analysis of the system up to now.

These are:

Total Ram available:  
Chip Ram available:  
Expansion Ram available  
System Stack located at:  
Copper List located at:  
Screen Planes located at:  
Kernel Data buffer located at:  
Disk Drive Df0  
Keyboard Startup

**System Info 2**

This table shows more information concerning the state of the system with the label on the right side indicating if the value is correct (OK) or not (WRONG).

Status Register(CPU) is:  
DMACON(DMA Control Reg.) is:  
Intena(Interrupt Enable Reg.) is:  
Interq(Interrupt Request Reg.) is:  
Adkcon(Disk-Audio Control Reg.) is:  
Cia A Timer A (After Reset) is:  
Cia A Timer B (After Reset) is:  
Cia B Timer A (After Reset) is:  
Cia B Timer B (After Reset) is:

**Chip RAM Test**

Shows a table with the results of the RAM test.

These values are:

Memory Test Start Address:  
Memory Test End Address:  
First Error Address:  
Last Error Address:  
Data Errors Number:  
Address Error Test Result:

### Exception Vectors Analysis

DIALOGA installs all the 256 vectors table of the CPU. If everything works correctly it appears the message:

```
All Vectors correctly installed
```

otherwise if a vector installation isn't possible due to damaged RAM, it appears a screen:

BLUE1	with 3 blinking	Bus error
-------	-----------------	-----------

Once correctly installed the vectors the diagnostic proceeds with the analysis of all CPU exceptions.

### Exception Vectors

The table shows the following exceptions:

```
Address Error
Illegal instruction
Division by Zero
CHK instruction
Overflow Trap
Privilege Violation
Trace On
Line A Emulation
Line F Emulation
```

If the analysis of those exceptions fails, the system blocks itself and it appear a screen:

GREEN1	with 4 blinking	Address error
PURPLE	with 5 blinking	Illegal instruction
PINK1	with 6 blinking	Division by Zero
ORANGE	with 7 blinking	CHK instruction
BLUE2	with 8 blinking	TrapV instruction
GREEN2	with 9 blinking	Privilege Violation
RED1	with 10 blinking	Trace On
BROWN	with 11 blinking	Line A Emulation
WHITE1	with 12 blinking	Line F Emulation

The CPU doesn't work correctly.

**Trap Vectors**

The table shows the following exceptions:

```
Trap #0 generated      Exception correctly executed
                        :
Trap #15 generated     Exception correctly executed
```

If the analysis stops, the exception is not correctly executed and it appears a screen:

GRAY	with 13 blinking	Not initialised interrupt
GRAY1	with 14 blinking	Ingiustified interrupt

**Interrupt Vectors**

These vectors are used by a CPU instruction.

The table shows the following exceptions:

```
Interrupt #1 generated Exception correctly executed
                        :
Interrupt #6 generated Exception correctly executed
```

In case of malfunction it appears a screen:

YELLOW1	with 15 blinking	Interrupt (1-7)
---------	------------------	-----------------

At the end the following message appears:

Exception Test completed

**Amiga model to analyse**

The diagnostic asks the user to choose the Amiga model.

The choice is made by mouse buttons or keyboard:

	Game Port 1/Port 2	Keyboard
Amiga 500	Right mouse button	Right Amiga key
Amiga 2000	Left mouse button	Left Amiga key

**Analysis Selection Menu**

Finished the first group of analysis, the user can choose one of the test available on the screen. The choice is made by mouse scrolling the cursor or pressing the keyboard number related to the desired option.

Possible choices are:

```

KEYBOARD Test
CIA Test
INTERNAL DRIVE Test
PARALLEL PORT Test
SERIAL PORT Test
EXTERNAL DRIVE PORT Test
CONTROL PORT 1 Test
CONTROL PORT 2 Test
MEMORY Test
DISPLAY Test
AUDIO Test
ALL IN ONE Test
AUTO Test
ABOUT

```

We suggest to choose ALL IN ONE Test. So the diagnostic executes all the available analysis. Please, follow the instructions on the screen.

**Conclusion**

We suggest you to try all the possible analysis one by one so you can gain familiarity with them. Remember that you cannot cause any damage using DIALOGA.

Good work.



---

## **Part 3. Reference**

**In this section are explained the evaluation and working of the available analysis of the diagnostic.**



### **3.1. START-UP**

In this phase DIALOGA starts to test progressively the system from the elements whose correct performance is necessary to proceed to the next phase.

At power up of the computer, the diagnostic makes the following operations.



### 3.1.1. CPU test

Here are executed the most important operations made by the microprocessor during normal working time. If there are anomalies the diagnostic signals them with a completely blue screen (\$00f).

Clearly, it can not be always possible to do so, because the CPU malfunction can cause unexpected behaviour.

### 3.1.2. Checksum

In this phase is verified the integrity of the data present in the firmware contained in the diagnostic ROM. If the checksum is incorrect the screen becomes red (\$f00).

### 3.1.3. Hardware registers

Initialisation of CIA, DMACON, INTENA, INTREQ and ADKCON registers in a way so DIALOGA finds standard values.

For a correct initialisation of the system it is necessary the presence in the internal disk drive DF0: of a blank write-enabled diskette.

After this phase, the external control box leds must be all off, otherwise the CIA (8520) or related circuits are faulty.

### 3.1.4. Overlay test

At the power up or after a reset, due to Overlay, the system doesn't see the RAM at address \$00000000, in its place appears the ROM memory area.

The Overlay is disabled, so the system can see the RAM at the address \$00000000. DIALOGA signals the impossibility to proceed with a pink screen(\$d57) in case of inability to disable the Overlay. In this case, the problem probably concern the CIA (A) and related circuits.

### 3.1.5. Exception vectors

DIALOGA proceed with the installation of the fundamental 68000 exception vectors with the objective to signal to the user anomalies like: unexpected interrupts, bus errors, address errors, unknown instructions, ecc.

### 3.1.6. VHPOS register

The correct working of the VHPOS register (horizontal and vertical position of the electronic beam) and the correct visualisation of the colors is verified.

The screen is divided in five different horizontal colored stripes.

They are rotated vertically to verify the colors in all the possible positions.

Showed colors on the screen must be:

COLORS	HEX CODE (RGB)
Red	\$F00
Green	\$0F0
Blue	\$00F
Gray	\$777
Black	\$000

### 3.1.7. CHIP RAM

The diagnostic checks the amount of available CHIP RAM to the system. The procedure used is able to detect CHIP RAM independently from the type of installed Agnus.

### 3.1.8. RAM test

RAM found previously is analysed to locate at least 70 Kbyte of CONTIGUOUS working CHIP RAM in any area of the memory.

During this phase, the regular execution of the analysis is signalled by irregular colored stripes on the screen (similar to those generated by crunchers or decrunchers).

DIALOGA signal the absence of 70 Kbyte of working contiguous RAM with a green screen (\$0f0) and blinks the computer power led. DIALOGA cannot proceed further more.

If there are some faulty RAM locations detected during this phase, the diagnostic signals this shortly with some green stripes on the screen.

### 3.1.9. Ranger memory expansions

The presence in the system of ranger type expansion memory is verified (RAM memory configured at the address \$c00000).

Please note that if the expansion is seriously damaged, the diagnostic may not be able to correctly detect the presence and can consider missing the expansion.

### **3.1.10. DMACON test**

Verifies the functions of the DMACON register.

In case of anomalies DIALOGA signals the impossibility to proceed with a yellow screen (\$cc0), the led ... on and halts itself.

The DMACON register malfunction makes impossible to check the correct working of the DMA for the screen allocation, for the Blitter handling, Audio, Sprites and the read/write from/to the disk drives.

### **3.1.11. Vertical Blanking test**

Verifies the correct working of the Vertical Blanking interrupt.

In case of incorrect working, DIALOGA halts itself and signals the problem with a completely white screen (\$fff) and the power led on.

This interrupt is generated every 1/50 of second in a PAL system and it is useful for graphic operations synchronization such as screen refreshing.

### **3.1.12. Blitter test**

The memory area reserved to the screen (not detectable yet) is filled with random data and successively cleared with the Blitter.

All the DMA channels of the Blitter, the most frequent logical operations, masks, shift and blit of different sizes are used. In absence of problems the screen must be perfectly clean and black.

The test checks the BLIT DONE flag, in fact prior to start the Blitter DIALOGA waits the clearing of this signal. While the diagnostic is waiting, the screen is brown (\$642) and the led blinks.

The waiting time is related to the type of installed Agnus in the system and to the type of startup of DIALOGA. The diagnostic can start with a power up or with a simple hard reset.

The correct working of the Blitter is verified also successively with the operations of clearing the screen and scrolling the text using the Blitter copy facility.

Possible working anomalies in these operations can be caused almost surely by a faulty Blitter.

### 3.1.13. Display set-up

The display must be formed by a stripe in the higher part of the screen with four not interlaced hires bitplanes (16 colors) showing:

- LOGICA logos centred,
- the diagnostic version number
- the name of the software author

A second part that occupies the remaining screen area is formed by one not interlaced hires bitplane where are displayed the diagnostic messages.

It presents this text:

```
The DIAGNOSTIC Kernel v1.0
```

```
Programmed by Bornacin Stefano in 1991/1992
```

```
All Rights reserved to LOGICA S.N.C
```

```
Press any Key or a Mouse Button to Continue
```

If the display appears correctly we can be sure that copper is working correctly, infact:

- In the copper list that handle the screen are fully used all the instructions recognised by the Copper: MOVE, WAIT, SKIP;
- The display hardware of Amiga in normal conditions works fine (DMA channels, control registers, bitplanes pointers, color registers, ecc.)

### 3.1.14. Keyboard Start-up test

Verifies the synchronization operations between the keyboard and the computer. In case of keyboard malfunction the keyboard is disabled and the user can use only the mouse to perform input operations.

### 3.1.15. Internal Disk Drive

Correct response to the identification request of the internal disk drive of Amiga (Df0:) at the diagnostic is verified.

### **3.2. End of START-UP**

The initialisation phase of DIALOGA is terminated.

The diagnostic is able to communicate with the user by the screen, without the need to use color codes replaced by more clear text messages. The essential parts of the system are working, but it's necessary to proceed with other analysis of the single parts to detect other problems not discovered yet.



### 3.2.1. System info 1

The screen presents the results of the performed analysis:

#### SYSTEM INFOS 1

```

-----
Total Ram available:           $00100000 (1 M)
Chip Ram available:           $00100000 (1 M)
Expansion Ram available:      None
System Stack located at:     $00100000
Copper List located at:      $000F2C32
Screen Planes located at:    $000EE772
Kernel Data buffer located at: $000F712C
Disk Drive Df0:              Detected
Keyboard Startup              Done
-----

```

Press any Key or a Mouse Button to Continue

These are:

Total Ram available:

The total RAM detected is given by the CHIP RAM plus the additional expansion RAM at the address \$C00000.

Chip Ram available:

Size of detected CHIP RAM.

Expansion Ram available:

Size of FAST RAM at the address \$C00000.

System Stack located at:

Shows the address at which is located the stack used by the diagnostic to operate. In normal conditions it's located at the highest CHIP RAM address detected.

Copper List located at:

Memory location of the copper list used to handle the display.

Screen Planes located at:

Start address of bitplanes data.

Kernel Data buffer located at:

Start of memory location used by the diagnostic to store internal variables.

Disk Drive Df0: Detected/None.

Shows if the internal disk drive has been detected correctly or not.

Keyboard Startup:

Shows the results of the Keyboard Startup test.

The Keyboard Startup presents one of the following messages:

Done

Shows a correct working of the keyboard signals.

Autotest Failed

The Autotest of the keyboard has failed, this situation is also communicated to the user by the keyboard with the blinking of the CAPS LOCK led. The number of blinking indicates the detected error type.

Blinking Number	Malfunction type
1	ROM checksum failed
2	RAM test failed
3	Watchdog timer test failed
4	There is a short between two files or one of the special test

No Keyboard Communication

After a fixed period, no synchronisation signal is received from the keyboard.

Startup Code Not Received

The initial startup-sequence code is not received, the keyboard is synchronised with the computer but didn't sent the initial startup-sequence code \$fd. After this, the keyboard sends the codes of the pressed keys during the power up.

End Startup Not Received

The end startup-sequence code is not received, it happens if from the reception of the initial startup code \$fd after a reasonable period of time is not received the \$fe code (end startup-sequence code). Please note that in

---

normal conditions the \$fd and \$fe codes are sent one after the other, if any keys have been pressed during the power up phase.

#### Too Much Chars Received

Too many characters codes received, the keyboard continues to send pressed or released keys codes without a plausible reason.



### 3.2.2. System info 2

The screen shows the system registers values useful to detect possible anomalies:

```

                                SYSTEM INFOS 2
-----
Status Register (CPU) is:           $2704   Ok
DMACON (DMA Control Reg.) is:      $03C0   Ok
Intena (Interrupt Enable Reg.) is: $0000   Ok
Interq (Interrupt Request Reg.) is: $0060   Ok
Adkcon (Disk-Audio Control Reg. is: $0000   Ok
Cia A Timer A (After Reset) is:    $FFFF   Ok
Cia A Timer B (After Reset) is:    $FFFF   Ok
Cia B Timer A (After Reset) is:    $FFFF   Ok
Cia B Timer B (After Reset) is:    $FFFF   Ok
-----

```

Press any Key or a Mouse Button to Continue

These are:

Status Register (CPU) is:

Shows the contents of the CPU State register. The value must be evaluated as a 16 bit binary number. It is important to verify bit 12 that shows the state of the microprocessor (Supervisor or User) and the interrupt enabling mask. Normally the correct value is 1 (supervisor) for bit 12 and 111(\$7) (all the interrupts disabled) for the interrupt enabling mask.

On the side of the hexadecimal number DIALOGA signals if the detected value is normally expected (OK) or not (WRONG).

DMACON (DMA Control Reg.) is:

Reports the contents of the DMACON register. It shows which DMA channels are enabled and which not. See appendix B.

Intena (Interrupt Enable Reg.) is:

Shows with a hexadecimal value the enabled interrupts in the system.

If the interrupt request is enabled by the Intena register, the CPU control is passed to the routine that should handle it, otherwise it is ignored. See appendix B.

Interq (Interrupt Request Reg.) is:

Shows with a hexadecimal value the requested interrupts. Which interrupts have been generated in the system. See appendix B.

Adkcon (Disk-Audio Control Reg.) is:

This register controls some parameters for the use of the disk drives and for the variations of width and frequency of sounds generated by the system. See appendix B.

Cia x Timer x (After Reset) is:

Shows the contents of the CIA timers counters after system initialisation. If they have been resetted correctly, they are suppose to carry \$FFFF (-1 decimal value).

### 3.2.3. Chip test

CHIP RAM TEST

```

-----
Memory Test Start Address:           $00000400
Memory Test End Address:             $00100000
First Error Address:                 None
Last Error Address:                 None
Data Errors Number:                 Done
Address Error Test Result           Done
-----

```

Press any Key or a Mouse Button to Continue

Reports the results of the analysis of the CHIP RAM:

Memory Test Start Address:

Shows the starting address (\$400) of the test that is the lowest RAM address after the 68000 vectors table.

Memory Test End Address:

Shows the final address of the test, the highest CHIP RAM location found.

First Error Address:

Shows the address of the first error found.

Last Error Address:

Shows the address of the last error found.

Data Errors Number:

Shows the number of the detected errors.

Address Error Test Result:

Detects the interruption on one or more addresses of the bus.

TESTING ADDRESS

WARNING !! ADDRESS ERRORS DETECTED IN CHIP MEMORY  
BE CAREFUL ! THE DIAGNOSTIC COULD MALFUNCTION !!

### 3.2.4. CPU vectors test

#### EXCEPTION VECTORS TEST

---

All Vectors correctly installed

Press a Mouse Button to Continue

The table of 256 vectors of the CPU is installed in the first Kbyte of memory of the system and verified.

The unsuccessful installation of a vector is signalled by the diagnostic with the message:

Memory Error on installing vector N.

At the end of the list of the incorrectly installed vectors appears:

#### EXCEPTION VECTORS TEST

---

Installation completed, some errors occurred

Press a Mouse Button to Continue

DIALOGA generates events that cause the use of the 'exception' procedures. If the procedure is not completed correctly the system halts itself. To restart is necessary to reset diagnostic.

The 'exception' is an event of particular importance that request the immediate attention of the CPU and the suspension of the work done until that moment.

We can divide them in three groups, they are:

-----  
 EXCEPTION VECTORS TEST  
 -----

Address Error generated	Exception correctly executed
Illegal instruction generated	Exception correctly executed
Division by Zero generated	Exception correctly executed
CHK instruction generated	Exception correctly executed
Overflow Trap generated	Exception correctly executed
Privilege Violation generated	Exception correctly executed
Trace On generated	Exception correctly executed
Line A Emulation generated	Exception correctly executed
Line F Emulation generated	Exception correctly executed

Press a Mouse Button to Continue

**exception vectors:**

they are those vectors that cause the classic 'GURU MEDITATION' generated from errors of the CPU: bus error, address error, illegal instruction, division by zero, CHK instruction, TRAPV instruction, privilege violation, trace, line A and line F emulation.

If the execution of these operations stops the computer, it means that the procedure is not been executed correctly and probably the CPU is faulty.

-----  
 EXCEPTION VECTORS TEST  
 -----

Testing....

Trap #1 generated	Exception correctly executed
Trap #2 generated	Exception correctly executed
Trap #3 generated	Exception correctly executed
Trap #4 generated	Exception correctly executed
Trap #5 generated	Exception correctly executed
Trap #6 generated	Exception correctly executed
Trap #7 generated	Exception correctly executed
Trap #8 generated	Exception correctly executed
Trap #9 generated	Exception correctly executed
Trap #10 generated	Exception correctly executed
Trap #11 generated	Exception correctly executed
Trap #12 generated	Exception correctly executed
Trap #13 generated	Exception correctly executed
Trap #14 generated	Exception correctly executed

Press a Mouse Button to Continue

**trap vectors:**

these vectors are used by the instruction TRAP #xx of the CPU (xx = 0 - 15). They are little used in the Amiga architecture.

## EXCEPTION VECTORS TEST

-----  
Testing....

Interrupt #1 generated	Exception correctly executed
Interrupt #2 generated	Exception correctly executed
Interrupt #3 generated	Exception correctly executed
Interrupt #4 generated	Exception correctly executed
Interrupt #5 generated	Exception correctly executed
Interrupt #6 generated	Exception correctly executed

Press a Mouse Button to Continue

**interrupt vectors:**

these are the level 1-7 interrupt vectors used by the 68000 microprocessors.

The level 7 interrupt (interrupt non maskable) is not tested because this can be generated only externally to the computer.

## EXCEPTION VECTORS TEST

-----  
Exception Test completed

Press a Mouse Button to Continue



### 3.2.5. Amiga model

The user selects in which model of Amiga is installed DIALOGA, the successive analysis are adequated to the chosen model.

The choice is made with the mouse buttons or with the keyboard:

	Game Port 1/Port 2	Keyboard
Amiga 500	Right mouse button	Right Amiga Key
Amiga 2000	Left mouse button	Left Amiga Key

\* \* \* \* SELECT TEST TYPE \* \* \* \*

AMIGA 2000 TEST

AMIGA 500 TEST

Press

Press

Left Button

Right Button



### **3.3. Analysis**

DIALOGA waits the user selection. Here are made more incisive and specialised analysis on the system. It is possible to analyse partially, to analyse all, or to test the system in a continuous loop.



### 3.3.1. Main menu

```

* * * * AMIGA xxxx TEST SELECTION MENU * * * *
1.....KEYBOARD Test 8.....CONTROL PORT 2 Test
2.....CIA Test 9.....MEMORY Test
3.....INTERNAL DRIVE Test 0.....DISPLAY Test
4.....PARALLEL PORT Test F1.....AUDIO Test
5.....SERIAL PORT Test F2.....ALL IN ONE Test
6...EXTERNAL DRIVE PORT Test F3.....AUTO Test
7.....CONTROL PORT 1 Test HELP.....ABOUT

      Use Keyboard or Mouse To Perform Selection

```

The user can choose the available options in two ways:

**keyboard:**

the choice must be made pressing the indicated key for the analysis to execute.

**mouse:**

the choice must be made moving the mouse vertically, the cursor highlight the desired option, to confirm press a mouse button.

In case the keyboard is not available the message changes from

```

      Use Keyboard or Mouse to Perform Selection
to

```

```

Keyboard DISABLED. Use the MOUSE to Perform Selection

```

If a wrong option is selected, it appears:

```

Invalid Selection

```



### 3.3.2. Keyboard test

The analysis verifies the correct working of the keyboard.

Pressing the two mouse buttons at the same time you can abort the current test

```
Keyboard Test Aborted
```

The diagnostic controls the results of the keyboard initialization made at the power up of DIALOGA.

If there were detected error codes, the initialization is executed again to verify the presence of errors.

Depending on the type of error detected (see 3.2.1), DIALOGA may continue the analysis or signal to the user the impossibility to proceed.

The successive phases of the analysis are not done with:

```
Keyboard Failed. Test is Impossible.
No Signals from Keyboard. Test is Impossible.
```

DIALOGA proceed with the successive phases of the analysis with:

```
WARNING!! End Startup Not Received
WARNING!! End Startup Code Received
WARNING!! Too Much Chars Received
```

In this case appears on the screen:

```
Keyboard Test is successfully,BUT a condition error was
detected during the Keyboard Startup Sequence
DO YOU WANT ENABLE THE KEYBOARD AS INPUT DEVICE?

NO, USE MOUSE          YES, USE IT
  Press                Press
Left Button            Right Button
```

Comes up the main screen of the test; this shows a representation of the keyboard with the keys object of verification. The indication of the key to press is on the low part of the screen to the left, while the hexadecimal code corresponding to the pressed key is on the right .

\* \* \* \* KEYBOARD TEST \* \* \* \*

```

ESC                                F7
    1                                9  -          DEL
        W                            I            ]
CTRL          D                    J
L . SH          V      N                R . SH DOWN
ALT L . AM                                R . AM ALT

```

Press the CTRL Key

Keyboard Code:\$

The keyboard analysis is made by pressing sequentially the keys prompted on the screen.

Keeping pressed the key, on the screen appear the indication

Key xxxx Pressed.

where xxxx is the name of the key and the correspondent hexadecimal code.  
Releasing it comes up the message

Key xxxx Released.

with the hexadecimal code of the release formed in this way: key code + \$80. It is possible to compare these codes with the table in the appendix C.

When releasing the key it is verified the capacity of recover of synchronization between the computer and keyboard. This is done by loosing it intentionally and then attempting to restore the correct communication.

Normally it appears for a while the code \$f9 which the keyboard uses to signal the lost of synchronization.

If it is not possible to restore synchronization, there is a continuous release of code \$f9 from the keyboard.

Keys can be pressed also at higher speed than that is required by the diagnostic. The keyboard buffer is able to store up to 10 characters . If this limit is overcome, the keyboard generates a code \$fa to signal this. DIALOGA shows this message:

Keyboard Buffer Overflow

Pressing an incorrect key appears the message:

Invalid Key Pressed.

and the diagnostic requests to press the key again.

If this happens four times with the same key, the test is considered failed and it appears the message

Keyboard Test Failed

If the key is detected correctly, the diagnostic erases it from the screen and proceeds to the next one until all requested keys are tested successfully.

Keyboard Test completed successfully



### 3.3.3. Cia Timer test

The CIA 8520 chips (Complex Adapter Interface) in Amiga system are of particular importance, to them is assigned the task of handling the input-output communications like: serial, parallel, disk drive, keyboard, mouse.

The three timers present in each one of the chips are able to give precise timing to the system.

It is clear that most of the problems that may concern the Amiga computer can be caused by the malfunction of these chips.

In this analysis the objective is to verify only the correct working of the timers, the tests of input-output handling are made by more efficient options chosen from the main menu.

On the initial screen are indicated progressively the type of analysis and his outcome.

It is possible to exit the current test keeping pressed two mouse buttons at the same time. It appears the message:

Cia Test Aborted

In relation to each single CIA, the analysis concern timer A, timer B (16 bit timer) and the TOD (24 bit timer).

With each one of these is reported a hexadecimal number that shows the value contained in the timer at the end of the counting time. The result may change slightly between two equal computer models.

The used algorithm is able to certify correctly the counting capacity of the timer eliminating the influence of the working modality of the timer and of the total workload of the system.

The table at the end of the analysis presents:

\* \* \* \* TIMERS TEST \* \* \* \*

CIA A (ODD):

Timer A	\$000039AD	Correct Value
Timer B	\$000039AD	Correct Value
Timer TOD	\$000100B4	Correct Value

CIA B (EVEN):

Timer A	\$000039AD	Correct Value
Timer B	\$000039AD	Correct Value
Timer TOD	\$0001DC2D	Correct Value

Press any key or a Mouse Button to Continue

The valuation of the state of the timer is reported on the right side of the counted value and it is expression of three base situations:

Doesn't count:  
the timer doesn't even starts.

Wrong Value:  
the timer doesn't counts regularly; where for 'regularly' must be understand a medium value present usually on Amiga computers and not a value calculated on the optimum characteristics of the timer.

Correct Value:  
the timer counts regularly; the offered timings are precise or presents very little differences with the optimum values.

### 3.3.4. Internal disk drive test

The analysis has the objective to verify the correct working of the internal disk drive of the computer and this means to check different components of the system like: mechanic of the disk drive, Gary chip (disk drive controller), Agnus and Paula chips for the control registers and DMA of the disk drive, CIA chip for timing and input-output control.

Prior to start the analysis, the diagnostic verifies that:

- the drive has been detected correctly during the power up (3.1.15). If not, the diagnostic retries to detect it.
- the ADKCON register works correctly
- the disk drive DMA is operative
- the write protection tab on the disk is disabled
- a CIA timer is available and working for the timing duties

**WARNING:** Prior to start the analysis of the disk drive is better to execute a CIA test with the objective to avoid possible damages that the disk drive may receive due to wrong temporization of disk drive heads movement .

In case that one of these conditions is not satisfied the analysis is not executed and appears one of the following messages:

```
Drive Not Detected. Can't Do Test
Read/Write ADKCON Register doesn't match.Can't do Test
Can't Active Disk DMA. Test is Impossible
Can't Direct Disk DMA. Test is Impossible
Can't use CIA Timers. Test is Impossible
ALL CIA Timers don't Count. Test is Impossible
```

DIALOGA returns to the main menu.

If all conditions are satisfied, the user can make his choice between two possible analysis or return to the main menu:

\* \* \* \* DRIVE TEST SELECTION MENU \* \* \* \*

```

1.....ULTRA FAST SPEED TEST      3.....EXIT
2.....DEEP READ/WRITE TEST      .....
```

Use Keyboard or Mouse to Perform Selection

ULTRA FAST SPEED TEST:

the analysis verifies the capacity of the drive to work in extreme conditions, at the limit of timings indicated normally for disk drives used on Amiga systems. A complete writing of the disk is made in 30 seconds. (in optimal conditions) In this test is made no control on data written/read from/to the disk drive but only to the correct response of the disk drive mechanic.

The heading of the screen presents:

\* \* \* \* ULTRA FAST SPEED DRIVE TEST \* \* \* \*

Heads are positioned on track 0, in order to read tracks from 0 to 79 and backwards. After that the test goes on with the reading process signalling those tracks not read correctly after a certain number of attempts. In this analysis due to the high speed at which data is treated and in relation to the type of the disk used it is possible to obtain "strange" results. It is possible that happens that:

- reading proceeds at a regular and constant speed
- reading is slow with stops due to the difficulty to identify tracks written too fast
- a track once is read correctly, once is considered wrong

It is important for the validity of the analysis that the disk drive is able to execute what requested without blocking itself and that there isn't an excessive number of tracks considered wrong.

**DEEP READ/WRITE TEST:**

the test verifies data written/read from/to the disk drive with methods frequently used on Nibble type copiers.

The heading of the screen presents:

```
* * * * DEEP READ/WRITE DRIVE TEST * * * *
```

The analysis is made always on tracks 0, 40, 79 and on two tracks chosen in a random way. For each track the diagnostic writes on each side of the disk. With the reading process is verified the correspondence with the original data.

**EXIT:**

returns to the main menu.

Once selected the type of analysis to execute, appears the main screen of the test which reports all the basic information regarding the state of the disk drive.

```
* * * * XXXXXXXXXXXX DRIVE TEST * * * *
```

**DISK INFOS:**

Disk Status:	Inserted
Disk Protection:	Disabled
Disk Motor:	Off
Disk Side:	Lower
Head Movement Direction:	Outward
Cylinder Number:	00
N. of Tracks Not Recognized:	None

Action: Reading

Messages: Done. Press any key or a Mouse Button to Exit

They are:

Disk Status: Inserted / Not Inserted

State of the disk drive, disk inserted in the drive or not

Disk Protection: Active / Disabled / ??

Write protection: Protected disk, not protected, non inserted. In case that the disk is protected or not inserted, the diagnostic request to remove the write protection to start the analysis. DIALOGA is structured to permit the extraction or the insertion of the disk in any moment allowing a normal use of the Amiga disk drive.

Disk Motor: On / Off

Shows the state of the disk drive motor.

Disk Side: upper / lower

Shows which side of the disk is operative.

Head direction: inward / outward

Shows in which direction are moving the heads of the disk drive

Cylinder Number: XX

Shows on which cylinder the heads are positioned (0-79)

Errors:

If the selected test is DEEP READ WRITE/DISK DRIVE TEST  
it is showed the number of errors that are detected in the last read track.

If the selected test is ULTRA FAST SPEED TEST  
it is showed the total number of tracks readied incorrectly.

Actions:

Shows the action the diagnostic is doing in that moment.

They are:

Waiting for User Action

Reading

Writing

Verifying

Turning On drive Motor

Searching for Cylinder

Messages :

More indications given to the user. They are:

ALL OK

Insert disk in drive 0

Disk is WRITE PROTECTED, Deprotect it

Some Errors on this Track

The track is recognised but there are some data that result to be wrong. The track is damaged, the diagnostic repeats the writing and successively verifies it up to five times. If each try fails, the diagnostic signal the impossibility of reading the track and pass to the next one.

No Errors on this Track

Sector Not Recognized

The sector read doesn't present the necessary data for identification.

Less than 11 Sectors Detected

Can't Read this Track

Retrying to test this Track

Index Pulse Signal Not Detected

Drive Ready Signal Not Detected

Disk DMA Enabled Signal Not Received

Write Enable Signal Not Received

Write Enable Signal Received While Reading!!

Note: Test operates using WORD SYNCH synchronization type with value \$4489 (classic value MFM).

The most important values concerning the timing are:

- side change of the disk: waiting time = 100 microseconds
- direction change of the heads: waiting time = 18 milliseconds
- step pulse : waiting time = 3 milliseconds before a new step pulse

If the disk drive DMA doesn't work, the diagnostic waits the end of the DMA operations. This may happen when persists the message

#### Disk DMA Working

In this case to avoid the block of the diagnostic the user has the possibility to force to terminate operations pressing at the same time the two mouse buttons. If this occurs data can be written/read in a faulty way, so the diagnostic advise the user of the error with a message. Pressing at the same time the two mouse buttons can be used in normal conditions to terminate more quickly the current test. The results of the analysis on the tested tracks while the two mouse buttons where pressed are not valid.

Please remember that the activity of the disk drive can be checked also by the led of the external control box regarding the disk drive port. This is possible because all the signals sent to a disk drive, are sent also to all the others present in the system. It is also possible to check the leds that show the enabling to write and MFM data flow to the disk during the write phase.

### 3.3.5. Parallel test

The analysis has the objective to verify the integrity of the parallel port of Amiga. This is made activating and deactivating sequentially the lines presents on the parallel port that is possible to test via software. These are connected externally to a series of leds showing the condition of the connected lines. This kind of approach needs the intervention of the user to verify the correct working of the examined line because it is not possible to do it via software. In other terms it is necessary to check that when the state of the line changes, on screen, there is a corresponding change of the led connected to this line.

```

* * * * PARALLEL PORT TEST * * * *

```

PIN	SIGNAL	STATUS
2	D0	HIGH
3	D1	HIGH
4	D2	HIGH
5	D3	HIGH
6	D4	HIGH
7	D5	HIGH
8	D6	HIGH
9	D7	HIGH
11	BUSY	HIGH
12	POUT	HIGH
13	SEL	HIGH
1	STROBE	HIGH
10	ACK (*)	TESTED OK

Action:

Signals with an asterisk (\*) are analysed by the diagnostic via software.

Once selected this test, it appears a screen in which are reported signals that are object of verification, the pin of the port where these are connected and the state of the related connected led.

On the lower part of the screen on the left is reported the action that the diagnostic is doing in that moment in order to inform the user.

Setting

Set the logical level 0

Cleaning

Set the logical level 1

Waiting for Signal to test ACK

Reading Parallel Port to generate STROBE signal

None

No action

The possible messages that can be reported relating to the state of the external led are:

LOW

Led is on

HIGH

Led is off

???

Inability to detect the state of the led. This happens for those signals that change continuously or for those with the initial state unknown.

TESTED OK

It refers to the signal that can be verified internally via software; it means that the test has been successful.

FAILED

Like the previous case but shows the failure of the test.

TESTED

Shows that is generated the necessary condition for the test of the signal but is not able to learn the outcome via software.

Signals tested by the analysis of the parallel port are:

d0 , d1 , d2 , d3 , d4 , d5 , d6 , d7 , BUSY , POUT , SEL , STROBE , ACK

but only ACK is verified via software.

It is possible to speed up the analysis keeping pressed the two mouse buttons at the same time.



### 3.3.6. Serial test

The analysis has the objective to verify the functions of the serial port. It is made in the same way of the analysis of the parallel port.

The signals with an asterisk (\*) are analysed by the diagnostic via software.

\* \* \* \* SERIAL PORT TEST \* \* \* \*

PIN	SIGNAL	STATUS
4	RTS	LOW
20	DTR	LOW
6	DSR (*)	LOW
5	CTS (*)	LOW
8	CD (*)	LOW
2	TXD (*)	HIGH
3	RXD (*)	TESTED OK

Action:

On the lower part of the screen on the left is reported the action that the diagnostic is doing in that moment in order to inform the user.

Testing RXD signal at 108 bps

Testing RXD signal at 9600 bps

Testing RXD signal at 31500 bps (Midi)

Testing RXD signal at 100000 bps

Writing to Serial to generate TXD signal

The possible messages that can be reported about the state of the external leds are:

LOW  
Led is on

HIGH  
Led is off

FAILED  
Shows the failure of the test.

???  
Inability to detect the state led. This happens for those signals that change continuously or for those with the initial state unknown.

Testing  
Analysis is running

Signals analysed are:

RTS, DTR, DSR, CTS, CD, TXD, RXD of these only the first two must be verified by the user while others are tested directly by the diagnostic.

On the external control box are present the signals RTS, DTR, TXD connected to the related led.

For the others signals the diagnostic verifies via software if these are set with the default state.

For the signal RXD analysis, data is transmitted to output on the line TXD with four different speeds, 108 bps, 9600 bps, 31500 bps, 100000 bps. Trough the external control box these data pass to the line RXD.

During the verification of both lines TXD and RXD the connected led to the first one blinks with the coincident sending of the 0 and 1 in the line TXD. At higher speeds correspond less visible variations of the led state .

It is possible to speed up the current analysis keeping pressed the two mouse buttons at the same time.

### 3.3.7. External disk drive port test

The analysis permits to verify the integrity of the external disk drive port. This test is made in the same way as the other two previous analysis.

It changes slightly depending on the Amiga model analysed, on Amiga 500 the checked signals are:

STEP, DIR, SIDE, MTRX, CHNG, WPRO, TRK0, RDY, SEL1, SEL2, SEL3, INDEX, DKWEB, DKWDB

\* \* \* \* AMIGA 500 EXTERNAL DRIVE PORT TEST \* \* \* \*

PIN	SIGNAL	STATUS
18	STEP	HIGH
19	DIR	HIGH
13	SIDE	HIGH
8	MTRX	HIGH
11	CHNG	HIGH
14	WPRO	HIGH
15	TRK0	HIGH
1	RDY	HIGH
21	SEL1	HIGH
9	SEL2	HIGH
20	SEL3	HIGH
22	INDEX ( * )	HIGH
16	DKWEB	HIGH
17	DKWDB	HIGH

Action:

On Amiga 2000, due to a different signal mapping of the port (caused by the possible presence in this model of two internal disk drives), they are:

STEP, DIR, SIDE, MTRX, CHNG, WPRO, TRK0, RDY, SEL2, SEL3, INDEX,  
DWEB, DKWDB

\* \* \* \* AMIGA 2000 EXTERNAL DRIVE PORT TEST \* \* \* \*

PIN	SIGNAL	STATUS
18	STEP	HIGH
19	DIR	HIGH
13	SIDE	HIGH
8	MTRX	HIGH
11	CHNG	HIGH
14	WPRO	HIGH
15	TRK0	HIGH
1	RDY	HIGH
21	SEL2	HIGH
9	SEL3	HIGH
22	INDEX ( * )	HIGH
16	DKWEB	HIGH
17	DKWDB	HIGH

Action:

In both cases signal INDEX is the only to be verified directly via software.

The possible messages that can be reported about the state of the external leds are:

LOW  
Led is on

HIGH  
Led is off

On the lower part of the screen on the left is reported the action the diagnostic is doing in that moment in order to inform the user.

Cleaning  
Set the logical level at 1

Setting  
Set the logical level at 0

Waiting for Signal to test INDEX  
Waits the signal to analyse INDEX

Testing DKWEB and DKWDB Signals  
Analysis of the signals DKWEB and DKWDB

It is possible to speed up the current analysis keeping pressed the two mouse buttons at the same time.



### 3.3.8. Control port test

This test permits to verify the correct working of the devices connected to the game ports as mouse or joystick. On the main menu are present two options that allow to analyse port 1 or port 2.

```
* * * * CONTROL PORT 1 TEST * * * *
```

```
Mouse:           (If a Mouse is  connected)
```

```
X Pos. Counter = 212    Y Pos. Counter = 28
```

```
Left Button: Released   Right Button: Released
```

```
Joystick:         (If a joystick is connected)
```

```
Joy Position: Centered
```

```
Fire Button: Released
```

```
Press any Key to Exit
```

After the selection, it appears a screen divided in two sections, mouse and joystick.

#### Mouse section

There are viewed the values of the internal Amiga counters related to the X and Y position of the mouse pointer and the indication regarding the state of the right and left mouse buttons.

```
Pressed  
Released
```

#### Joystick section

It is viewed the position assumed by the joystick (forward, backward, left forward, ecc.) and the state of the fire button.

Centered  
Down  
Up  
and Left  
and Right

To exit the analysis press a key on the keyboard if available, otherwise the two mouse buttons at the same time if the keyboard is not connected or is seriously damaged.

### 3.3.9. Memory test

Permits a full verification of the memory selected from the user (CHIP or EXPANSION memory).

At the selection of the test there is the possibility to choose the type of memory to test:

```
* * * * SELECT MEMORY BLOCK TO TEST * * * *
```

```
CHIP MEMORY AT $000000      EXPANSION MEMORY AT $C00000
```

```
Press
```

```
Press
```

```
Left Button
```

```
Right Button
```

#### CHIP Memory

All the CHIP memory present in the system that has been found during the startup phase of the diagnostic (3.1.7) is tested byte by byte. This memory includes also the table of the vectors of the CPU previously treated above (\$0-\$400) and all the CHIP memory used by the diagnostic itself to execute his functions.

#### EXPANSION Memory

With the term expansion memory is indicated memory located at the address \$c00000. Actually memory at the address \$200000 is not tested, because this kind of memory make itself visible to the system trough a particular procedure called "AutoConfig" which is not implemented in the diagnostic.

The diagnostic prior to proceed, executes the same analysis made for the first time during the Startup phase (3.1.9) to verify the presence of the expansion at the address \$c00000.

If the expansion is not detected, the user is informed with a message.

```

Expansion Memory NOT FOUND
Do You Want Execute The Test Anyway ??
REMEMBER, IF NO EXPANSION IS CONNECTED, THE SYSTEM
WILL CRASH!!

```

```

NO, EXIT FROM TEST
Press
Left Button

```

```

YES, DO IT !!
Press
Right Button

```

The cause of the missed detection of the expansion by the diagnostic can be of the absence of it or of the presence of too many damaged memory locations.

The choice to continue in case that the expansion is not installed halts the system.

The user can select the capacity in Kbyte of the expansion subject to analysis or return to the main menu.

In case of the Amiga 500 are available these options:

```

1..512 K Bytes ($C00000-$C80000)
2.....1 M Byte ($C00000-$D00000)
3...1,5 M Byte ($C00000-$D80000)
4...1,7 M Byte ($C00000-$DC0000)
5.....EXIT TO MAIN MENU

```

In case of the Amiga 2000 are available these options:

```

1..512 K Bytes ($C00000-$C80000)
2.....EXIT TO MAIN MENU

```

Terminated the initial selections, it is presented to the user a screen similar the one used during the end of Startup phase (3.2.3). The heading changes depending the type of memory that is verified (CHIP or EXPANSION).

## XXXX RAM TEST

Memory Test Start Address:	\$00000000
Memory Test End Address:	\$00100000
First Error Address:	None
Last Error Address:	None
Data Errors Number:	Done
Address Error Test Result	Done

-----

Test Completed

Press any Key or a Mouse Button to Continue

Possible messages that can be reported depending on the analysis Address Error Test are:

No Errors  
Addresses lines working

Errors Found  
Address lines corrupted

Not Done Yet  
Analysis waiting to be executed

Can't do Test  
Impossibility to execute the analysis

Not Done  
Analysis not executed

On the lower part of the screen on the center is reported the action that the diagnostic is performing in that moment in order to inform the user.

```
Now Testing Ram Addresses..
```

```
Test Completed
```

During the analysis, the diagnostic to signal his activity blinks the background of the upper part of the screen, in the lower part of the screen indicates the memory zone under analysis and provides to update immediately the information concerning the detected errors. When an error is found the diagnostic increases the related counter by one

```
Data Errors Number: xx
```

and updates the hexadecimal values related to the memory addresses indicating the first and the last error detected.

```
First Error Address: xx
```

```
Last Error Address: xx
```

Memory to test is ideally divided in banks of 512 K each one (example: the first 1024 K of RAM memory are divided in banks 0-1).

Memory is analysed with a word step by writing and reading different bit patterns in a way to be certain of the integrity of the memory under exam.

At the end of the analysis, if no errors are detected, DIALOGA returns to the main menu, otherwise it displays a second screen where are reported more detailed information related to the detected errors.

```
----- DETAILED RAM ERRORS REPORT -----  
BITS   BANK 0 BANK 1 BANK 2 BANK 3 BANK 4 BANK 5 BANK 6  
BANK 7  
  
d0  
d1  
d2  
d3  
d4  
d5  
d6  
d7  
d8  
d9  
d10  
d11  
d12  
d13  
d14  
d15
```

Errors are grouped accordingly to the bank in which they have been detected and the wrong bits of the word. In this way it is possible to locate the defective memory chips and to provide to their substitution.

If there are faulty bits it is viewed the number of errors in decimal value.

If a certain bit is correct, on the screen, the number of errors is replaced by the symbol

XXXXXXXX

If a memory bank is not used it comes out the message

----Bank UNUSED----

Pressing the two mouse buttons at the same time, the analysis is interrupted and appears the message

BREAK..Test NOT Completed

If until interruption time were detected memory errors, the diagnostic shows the detailed report screen.



### 3.3.10. Display test

Allows to test and verify all the Amiga graphic modes, those standard and those not, those made available from the new graphic chips and the correct working of the sprites.

The option Display Test of the main menu bring forward a second menu where is possible to select the desired graphic mode.

\* \* \* \* DISPLAY PREFERENCES SELECTION MENU \* \* \* \*

1.....LOW RESOLUTION Screen	6.....DUAL PLAYFIELD Screen
2.....MEDIUM RESOLUTION Screen	7.....OVERSCAN Screen
3..HIGH RESOLUTION LACED Screen	8.....SPRITES Screen
4.....HOLD AND MODIFY Screen	9.....AUTO DISPLAY
5.....EXTRA HALF BRITE Screen	0.....EXIT TO MAIN MENU

Use Keyboard or Mouse to Perform Selection

The selection of one of the available options changes the resolution of the graphic screen.

The display is divided in three parts:

- higher part in hires not interlaced with 4 bitplanes and 16 colors
- central part in hires interlaced with 1 bitplane and 2 colors
- lower part with characteristic related to the selection made by the user.

This configuration of screen can be kept also while executing all the other available analysis in the main menu.

In this way it is possible to analyse the computer with different operative configurations and different workload.

The graphic modes in high resolution produce a lower availability of the DMA resource that are taken away from the other system components (CPU, Blitter, copper, sprites, ecc.) . The speed of execution of the system decrease.

Try to execute the CIA Test with different graphic modes to observe the variation of the values displayed by the counter. They change eloquently in function of the selected type of graphic resolution.

The activation of the graphic mode depends on two conditions:

- the correct installation of level three interrupt for interlace handling
- the presence of working CHIP memory higher than 256K to contain the bitplanes of the screen to view.

The absence of one of this two condition doesn't permit the visualisation of the desired graphic screen and the user is warned by a message.

```
Can't Install Display Interrupt. Option not available
      Not enough Chip Memory for Screen
```

These are the different existing options on the display menu:

Low Resolution screen:

Activates a screen in low resolution 320x102 with 5 bitplane, 32 colors. The yellow lines viewed on the screen allow to verify the dimension of the single pixel. Please note that text indicating the display characteristics is printed using always the same font 9x9. It is possible to check the display characteristics also looking at the characters aspect.

Medium Resolution screen:

Allow to restore the main work screen disabling the alternative display.

High Resolution Laced Screen:

Activates the screen in high resolution, 640x204 interlaced, 4 planes 16 colors.

Ham screen:

Activates the hold and modify screen, 320x102 not interlaced, 6 bitplanes and 4096 colors.

Ehb screen:

Activates the extra half brite screen, 320x102 not interlaced, 6 bitplanes 64 colors, the screen presents the 32 base colors in the upper part and the 32 half brightness colors in the lower one.

Dual Playfield screen:

Activates the dual playfield screen, 320x102 not interlaced, 6 bitplanes 8 independent colors for screen. With this graphic mode are created two independent screens (playfield) formed by a maximum of three bitplanes each. To test the activation of dual playfield mode, the two screen are switched between them at regular interval acting on the video priority register.

Overscan screen:

Activate the overscan mode screen, 736x102 not interlaced, hires 4 bitplanes, 16 colors. With this analysis it is generated the most wide display that the system can handle. To be able to observe all the display it is necessary to change the centering of the screen on the monitor. It is the graphic mode that uses the highest quantity of DMA resources, infact, if sprites are activated, only five will be displayed at the place of eight due to insufficient DMA resources. In the first Amiga versions the fifth sprite may appear corrupted.

#### Sprites on/off:

Allows to activate the sprites or to deactivate them if previously activated. These are represented by numbers that go from 1 to 8. They are displayed in an area on the lower part of the screen using four colors each, taken from the highest 16 color registers. Depending on the color used for the different screens, sprites change their color if the active display uses also the 16-32 registers for its colors. The activated sprites with the medres (normal) screen can be partially covered by text of other current analysis.

#### Autodisplay:

Allow to test automatically all graphic modes without the user selection; each screen is displayed either with activated sprites either with deactivated sprites. Activating the procedure appears the message:

```
Auto Display Procedure Active
```

At the end of the analysis the diagnostic returns in medres mode. Pressing a mouse button the procedure speeds up to the end.

#### Exit to main menu:

Permits to return in the main menu leaving active the options selected by the user.



### 3.3.11. Audio test

Allows to verify the integrity of the audio functions.

```
* * * * AUDIO TEST * * * *
```

	Status	Volume	Note	Period	Freq.(HZ)
Channel 1:	Off	45	F#2	0302	05965
Channel 2:	Off	30	F#2	0151	11931
Channel 3:	Off	40	C#3	0202	08860
Channel 4:	Off	30	A#2	0240	07457

Audio Filter: On

Press any Key or a Mouse Button to Continue

At the selection of this analysis appear a screen where are showed for each one of the four audio channels:

status: on / off

Shows if the channel is enabled or not.

volume:

Shows the volume level of the channel (0-64)

note:

Shows the note played in the related channel using the english notation. For example c-3 is equivalent to do in the third octave, a#2 is la diesis in the second octave.

Period:

Shows the period used to generate the note; this is the value inserted in the related audio registers.

Freq. (HZ) :

Shows the frequency in hertz of the played note.

The analysis is made by playing all notes in each channel with a different volume starting from c-1 to b-3. With each change of channel is modified the state of the audio filter to check that it works correctly.

Then some chords are played at different volume using all four channel at the same time. If the audio DMA works correctly it shouldn't be heard strange sounds or metallic sounds.

### 3.3.12. All in one test

This option allows the user to make all the analysis sequentially without the need to select them one by one from the main menu. This option has been inserted for operative reasons.

\* \* \* \* ALL IN ONE PROCEDURE \* \* \* \*

This option lets you perform ALL TESTS in sequence without going back to main menu every time a test is finished

NO, EXIT!

YES, DO IT!

Press

Press

Left Button

Right Button

The selection of this option from the main menu make appear the screen where the user can confirm or cancel the procedure.

The choice is made with the mouse buttons or with the keyboard:

	Game Port 1/Port 2	Keyboard
Confirm	Left mouse button	Left Amiga key
Cancel	Right mouse button	Right Amiga key



### 3.3.13. AutoTest

This procedure is used to analyse the system in a continuous way with the objective to check the integrity for a certain period of time. The procedure terminates when an analysis doesn't ends with the correct values or when the user decides to stop the execution.

Analysis of this procedure are:

Memory test (only Chip ram), Cia test, Parallel test, Extern test, Serial test and Df0 test (with Ultra Fast Speed test).

It appears a message screen and the diagnostic waits the user to confirm or to cancel the procedure.

```
* * * * AUTO TEST PROCEDURE * * * *
```

This option lets you run some tests until an Error condition occurs. However, you can stop the Auto test while the Auto Test Screen appears before each Test is executed

```
BY PRESSING BOTH THE MOUSE BUTTONS
```

```
Press any Key or a Mouse Button to Continue
```

The diagnostic presents another screen where it shows which is the next analysis to execute. This screen comes up for all the analysis.

```
* * * * AUTO TEST SCREEN * * * *
```

```
PRESS BOTH MOUSE BUTTONS TO EXIT
```

```
OR WAIT FOR
```

```
MEMORY TEST
```

It's possible to return to the main menu by pressing both mouse buttons together. If the analysis discovers anomalies in the system, it appear a message where the diagnostic shows the terminated analysis and the impossibility to continue.

```
MEMORY TEST Failed. Auto Testing STOPPED
CIA TEST Failed. Auto Testing STOPPED
PARALLEL TEST Failed. Auto Testing STOPPED
EXTERNAL DRIVE TEST Failed. Auto Testing STOPPED
SERIAL TEST Failed. Auto Testing STOPPED
DRIVE 0 TEST Failed. Auto Testing STOPPED
```

### 3.3.14. About

Shows a screen with the copyright and credits of the author and his development team.

THE DIAGNOSTIC V1.0

A SOFTWARE DIAGNOSTIC FOR AMIGA 500/2000 (PAL, CPU, 68000)  
ALL CONCEPTS AND PROGRAMMING IN PURE ASSEMBLER BY  
BORNACIN STEFANO

HARDWARE INTERFACES BY  
BIASUTTI FABIO  
TASELLI SIMONE

MANUAL BY  
SANAVIA ANDREA AND BORNACIN STEFANO

PROJECT DEVELOPED FOR  
LOGICA S.N.C. IN 1991/92

Press any Key or a Mouse Button to Continue



### 3.4. Exceptions

Exceptions can come up anytime when the computer subject to analysis is not working correctly.

First of all when they are generated the diagnostic halt its activity, it signals to the user the situation and waits for an external reset to start again. The message is delivered in a different way if we are in the startup phase or this phase is terminated.

In the first case the exceptions are signalled with color codes listed in Appendix A. In the second case it appear on the screen a message similar to this one:

```
THE BOSS MEDITATION,unexpected interrupt occurred
Can't continue PC was at $xxxxxxx
```

with the blinking of the power led of the computer in a way similar to normal GURU MEDITATION of Amiga.



## COLOR CODES

During the first phase of initialization of the diagnostic it is not available the system RAM due to the fact that it is not tested yet . For this reason DIALOGA uses a color code system to signal any anomalies clearly.

### COLOR CODES

COLOR CODE	RGB (HEX)	BLINKING LED	BLINKING. SCREEN	DESCRIPTION
blue	\$00f	no	no	CPU malfunction
red	\$f00	no	no	EPROM Checksum error
green	\$0f0	yes	no	Memory error
pink	\$d57	no	no	Overlay error
yellow	\$cc0	no	no	DMACON error
brown	\$642	yes	no	Blitter error (if continuous)
white	\$fff	no	no	Vblank error

### EXCEPTIONS

COLOR CODE	RGB (HEX)	BLINKING LED	BLINKING. SCREEN	DESCRIPTION
blue1	\$44b	yes	yes (3)	Bus error
green1	\$3da	yes	yes (4)	Address error
purple	\$506	yes	yes (5)	Illegal instruction
pink1	\$c09	yes	yes (6)	Division by Zero
orange	\$fa6	yes	yes (7)	CHK instruction
blue2	\$00f	yes	yes (8)	TrapV instruction
green2	\$080	yes	yes (9)	Privilege Violation
red1	\$c00	yes	yes (10)	Trace
brown1	\$540	yes	yes (11)	Line A Emulation
white1	\$fff	yes	yes (12)	Line F Emulation
gray	\$555	yes	yes (13)	Not initialized interrupt
gray1	\$bbb	yes	yes (14)	Ingiustified interrupt
yellow1	\$ff0	yes	yes (15)	Interrupt (1-7)

The numbers between (XX) shows the number of binkings.



**ADKCON - INTENA - DMACON**

DIALOGA gives the register values in hexadecimal form. To evaluate correctly these values they must be converted to binary form.

For example consider the value \$013f. This is equal to:

hex	0				1				3				f			
bin	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
bit	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00

## ADKCON

Bits	Function										
15	Set/ Clear control bit. Determines if bits written with a 1 get set or cleared. Bits written with a zero are always unchanged.										
14-13	PRECOMP 1-0 <table style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th><u>Code</u></th> <th><u>PRECOMP Value</u></th> </tr> </thead> <tbody> <tr> <td>00</td> <td>none</td> </tr> <tr> <td>01</td> <td>140 ns</td> </tr> <tr> <td>10</td> <td>280 ns</td> </tr> <tr> <td>11</td> <td>560 ns</td> </tr> </tbody> </table>	<u>Code</u>	<u>PRECOMP Value</u>	00	none	01	140 ns	10	280 ns	11	560 ns
<u>Code</u>	<u>PRECOMP Value</u>										
00	none										
01	140 ns										
10	280 ns										
11	560 ns										
12	MFMPREC 1 = M5FM precomp 0 = GCR precomp										
11	UARTBRK Forces a UART break (clears TXD) if true.										
10	WORDSYNC Enables disk read synchronizing on a word equal to DISK SYNC CODE, located in address (\$7e)										
09	MSBSYNC Enables disk read synchronizing on MSB (most signif bit). Applicable type GCR.										
08	FAST Disk data clock rate control. 1 = fast (2us) 0 = slow(4us). fast for MFM, slow for MFM and GCR.										
07	Uses audio channel 3 to modulate nothing.										
06	Uses audio channel 2 to modulate period of channel 3.										
05	Uses audio channel 1 to modulate period of channel 2.										
04	Uses audio channel 0 to modulate period of channel 1.										
03	Uses audio channel 3 to modulate nothing.										
02	Uses audio channel 2 to modulate period of channel 3.										
01	Uses audio channel 1 to modulate period of channel 2.										
00	Uses audio channel 0 to modulate period of channel 1.										

## INTENA - INTREQ

Bits	Function
15	Set/Clear control bit
14	INTEN Master interrupt (enable only, no request)
13	EXTER 6 External interrupt
12	DSKSYN 5 Disk sync register (DSKSYNC) matches disk data
11	RBF 5 Serial port receive buffer full
10	AUD3 4 Audio channel 3 block finished
09	AUD2 4 Audio channel 2 block finished
08	AUD1 4 Audio channel 1 block finished
07	AUD0 4 Audio channel 0 block finished
06	BLIT 3 Blitter finished
05	VERTB 3 Start of vertical blank
04	COPER 3 Copper
03	PORTS 2 I/O ports and timers
02	SOFT 1 Reserved for software-initiated interrupt
01	DSKBLK 1 Disk block finished
00	TBE 1 Serial port transmit buffer empty

## DMACON

Bits	Function
15	Set / Clear control bit
14	BBUSY Blitter busy status bit (read only)
13	BZERO Blitter logic zero status bit (read only)
12	X
11	X
10	BLTPRI Blitter DMA priority (over CPU micro) (also called "blitter nasty") Disables /BLS pin, preventing micro from stealing any bus cycles while Blitter DMA is running.
09	DMAEN Enable all DMA below
08	BPLEN Bit plane DMA enable
07	COPEN Copper DMA enable
06	BLTEN Blitter DMA enable
05	SPREN Sprite DMA enable
04	DSKEN Disk drive DMA enable
03	AUD3EN Audio channel 3 DMA enable
02	AUD2EN Audio channel 2 DMA enable
01	AUD1EN Audio channel 1 DMA enable
00	AUD0EN Audio channel 0 DMA enable

## **KEYS CODES**



## CONDITIONS OF USE

The code, the external equipment, and the documentation are exclusive propriety of Logica. Copyright 1993.

No part of the code, or the external equipment, or the documentation can be reproduced, transmitted, transcribed, memorized or translated in any language or programming language, in any form or in any way, optical, electronic, magnetic, biological, chemical, mechanic, acoustic, manually or other, without a written authorisation of Logica.

The user is owner of the diagnostic, but due to this agreement Logica keeps for itself the propriety of the code memorised on the EPROM. It is absolutely forbidden to duplicate in any way the code.

The product comprising of code, external equipment and documentation can be transferred to another user only if this user accept this conditions.

The authorisation of Logica is transferred to the new user. The former user doesn't maintain any rights on the transferred material.

It is forbidden to change, convert, translate, adapt, reengineerize, decompile, disassemble also partially the code, the external equipment, and the documentation.

Logica doesn't grant that the product is completely bug less or that it meets the requirements of the user.

Any warranty explicit or implicit with exception of those here expressly written are excluded.

In no way the Logica can be held responsible of damages directly or indirectly caused by the use or missed use of the code, the external equipment, and the documentation independently from the fact that the possibility of occurrence has been notify to Logica.



Note

---



Note

---

