

PETSCII Doctor 64

User manual

A free diagnostic cartridge for the Commodore 64: it tests the memory without using it, then every chip, draws your board with the chips to change, and tests with you the keyboard, the joysticks, the sound and the picture.

Version 0.4.10 beta · November 5, 2026

© 2026 P2i Lab · BSD 3-Clause

Contents

1. What PETSCII Doctor 64 is 3

2. Before you start 3

What you need 3

3. What is in the zip 4

4. Getting your cartridge ready 5

At a glance 5

Kung Fu Flash 5

EasyFlash (the original, or a copy) 5

EasyFlash 3: not on a C64 that shows nothing 6

TeensyROM 6

1541 Ultimate II or II+ 6

An EPROM on an Ultimax cartridge board 6

To try it: VICE, MiSTer, Ultimate 64, C64 Ultimate 6

5. Using it, step by step 7

The keys on step 2's screen 7

6. Step 1: the memory 8

The border colour 8

What each fault looks like 8

7. Step 2: the full check-up 10

The list 10

The lines under the list 11

The drawing of your board (F1) 11

The details (F3) 12

Again and again (F7) 12

8. Step 3: the tests with you 13

The keyboard 13

The joysticks, paddles and mouse 13

The sound 14

The picture 14

The report 15

9. What each test checks, in plain words 16

Step 1: the memory, without using it 16

Step 2: the full check-up 16

Step 3: what only you can judge 17

10. The words on the screens 18

Step 1: a memory chip is broken 18

Step 2: the list and the lines under it 18

Step 2: the details (F3) 20

Step 2: the drawing of your board (F1) 21

Step 3's screens 21

11. The cartridge's LED 22

From power-on to the result 22

What is your LED doing? 22

How to count the flashes 23

Step 1: which memory chip 23
Step 2: which line 23
12. When the screen stays black 23
13. Where the chips are, board by board 25
The five boards, seen from above 25
14. The C64's parts and other words 28
15. Tell us what it found 28
16. The music, the licence, thanks 28
Appendix: for experts, how each test works 30
Step 1, without RAM 30
Step 2, the full check-up 30
Step 3, the tests with you 31

1. What PETSCII Doctor 64 is

PETSCII Doctor 64 is a free diagnostic cartridge for the Commodore 64. It finds the broken parts of a C64, tells you in plain words which chip to look at, and shows where that chip sits on your board. Every detail is there too, for the expert.

It works in three steps:

1. **Step 1, the memory, without using it** (a few seconds, by itself). For a C64 whose screen stays black. Like Commodore's Dead Test, it tests the colour RAM, the memory chips, the memory's wiring and the first 4 KB of RAM, and names the chip or the line to look at.
2. **Step 2, the full check-up** (about 35 seconds, by itself), once the first 4 KB work: all 64 KB of RAM, the three ROMs, the PLA, both CIAs, the clock and its 9 V AC, the VIC-II, the SID and the serial port. A drawing of your board then shows each chip: green if it works, red and blinking if it is broken.
3. **Step 3, the tests with you** (a few minutes): what the C64 cannot judge alone. The keyboard, the joysticks (paddles and a 1351 mouse too), the sound and the picture. Then a report and, if everything works, the C64 says "All system tests are OK" out loud.

Dead Test and DesTestMAX only test the first 4 KB of RAM: the Ultimax mode they run in shows the processor nothing more. PETSCII Doctor 64 leaves that mode once those 4 KB are proved, to test the rest. It was written from scratch: it holds no code from Commodore's Dead Test or diagnostic cartridges.

> **A beta.** PETSCII Doctor 64 has been tested a lot, but in emulators and on a C64 simulated by software, with faults put in on purpose. It now needs real C64s that do not work. Try it on yours and, once you have found the real fault, tell us what PETSCII Doctor 64 said: right or wrong, each report makes it better (chapter 15).

The cartridge's screens are in English. This manual quotes them like this: `GOOD NEWS: THE MEMORY WORKS.` Keys look like this: `F1`, `SPACE`.

2. Before you start

> - **Switch off and unplug the C64** before you plug in or pull out a cartridge, and before you open the C64.
> - If you can, **measure the power supply first**: a worn-out Commodore power brick can give more than 5 V and kill the chips. On the user port (at the back, the leftmost socket as you face the C64), top row: pin 1 is ground, pin 2 must read 4.9 to 5.2 V DC, pins 10 and 11 about 9 to 10 V AC between them. Above 5.3 V, unplug at once and change the power supply.
> - **Unplug the rest**: disk drives, printer, other cartridges. A drive on the serial port can make its test fail. Joysticks, paddles or a mouse may stay: step 3 tests them (plug them in with the C64 switched off).

What you need

- A way to start PETSCII Doctor 64 (chapter 4): a Kung Fu Flash, an EasyFlash (the original or a copy, not an EasyFlash 3), a TeensyROM, a 1541 Ultimate II or II+, or an EPROM on an Ultimax cartridge board (step 1 only).
- For a flash cartridge: a C64 that works, once, to prepare it (a friend's, or yours before it broke).
- A TV or monitor, with its sound on for step 3; joysticks, if you want their ports tested.
- For the drawing of your board: its number, printed on it ("ASSY NO. 250407", for example). You read it once the C64 is open; PETSCII Doctor 64 asks for it the first time you press `F1`.

3. What is in the zip

File	What it is
`petscii-doctor-64.crt`	**The cartridge**: all three steps, in the EasyFlash format. For a Kung Fu Flash, an EasyFlash, a TeensyROM, a 1541 Ultimate II or II+, VICE, MiSTer, the Ultimate 64 and the C64 Ultimate.
`petscii-doctor-64-step1.bin`	Step 1 alone, 8 KB, to burn into an EPROM (2764 or 27C64) on an Ultimax cartridge board, like a Dead Test board. It repeats its tests for ever. Its screen says V0.4: step 1 has not changed since that version.
`petscii-doctor-64-step1.crt`	The same step 1 alone, as a cartridge image (Ultimax).
`README.md`	The full guide, in English, with every technical detail.
`LICENSE`	The licence (BSD 3-Clause), and the terms of the tune.
`musique/README.md`	“Ode to C64”: Jeff’s permission, and what it covers.
`voice-all-system-tests-are-ok.wav`	The voice the C64 speaks when everything works, to hear on a PC.
`PETSCII-Doctor-64-0.4.10-manual-en.pdf` `PETSCII-Doctor-64-0.4.10-manuel-fr.pdf`	This manual, in English and in French.

4. Getting your cartridge ready

A broken C64 often cannot show a flash cartridge's menu: that menu is a C64 program, and it needs the memory and the ROMs that may be the broken parts. So prepare the cartridge **on a C64 that works** (a friend's, or yours before it broke) so that it starts PETSCII Doctor 64 by itself at power-on. Then switch off, plug it into the broken C64, and switch on. Step 1 needs nothing of that C64 but its processor (the 6510), its clock, the PLA and the cartridge port.

At a glance

Your device	Starts it by itself on a C64 that shows nothing	Its LED gives the results	What runs
Kung Fu Flash	Yes, once started in "Run (VIC-II/C128 mode)"	Yes	All three steps
EasyFlash (the original, or a copy such as the EasyFlash 1 EVO)	Yes, its jumper on BOOT	Yes	All three steps
EasyFlash 3	No: its own menu always starts first	—	All three steps, on a C64 that still shows its menu
TeensyROM	Yes, with its Auto-Launch (its firmware 0.6.6 or later; not yet tried)	May not	All three steps (tried on a real C64)
1541 Ultimate II, II+	Yes, after "Copy to Flash" (not yet tried on a real one)	May not	All three steps
EPROM on an Ultimax board	Yes, always	No LED	Step 1 alone, again and again
VICE, MiSTer, Ultimate 64, C64 Ultimate	For trying it only: their chips are emulated or inside an FPGA, and every test passes (on a MiSTer, the PLA says **?**: one rare case differs)	—	All three steps

Kung Fu Flash

1. Copy `petscii-doctor-64.crt` to its microSD card.
2. On the working C64, switch on and press its **MENU** button: its launcher opens.
3. Choose `petscii-doctor-64.crt`, press **SHIFT**+**RETURN** for the file's options, and choose **Run (VIC-II/C128 mode)**, not the plain "Run": step 1's screens are read by the VIC-II from the cartridge, which the Kung Fu Flash only allows in that mode.
4. PETSCII Doctor 64 starts. The Kung Fu Flash keeps it: from then on it starts it by itself, in the same mode, at every power-on, without its menu, even on the broken C64. Leave the microSD card in.

Its **MENU** button brings its menu back. Its LED gives the results when the C64 shows no picture (chapter 11).

EasyFlash (the original, or a copy)

1. On the working C64, set its jumper to PROG (DISABLE on some versions).
2. Start EasyProg from a disk and write `petscii-doctor-64.crt` into the cartridge.
3. Switch off and set the jumper to BOOT: it now starts PETSCII Doctor 64 at every power-on.

Its LED gives the results too (chapter 11).

EasyFlash 3: not on a C64 that shows nothing

Its own menu starts at every power-on (and from its MENU button), and that menu runs on the C64: a C64 that cannot show it cannot start anything from it. On a C64 that still shows that menu, write `petscii-doctor-64.crt` into one of its slots with EasyProg, and start it with that slot's key.

TeensyROM

1. Copy `petscii-doctor-64.crt` to its microSD card.
2. On the working C64, switch on: the TeensyROM's menu opens. Go to the file and press `SHIFT`+`A`: its **Auto-Launch** now starts that file. In its settings, check that Auto-Launch is on.
3. From then on it starts PETSCII Doctor 64 by itself at every power-on, without its menu, even on the broken C64. This needs its firmware 0.6.6 or later: update it first if it is older.

Its menu button brings its menu back, and its settings turn Auto-Launch off. Without a C64 that works, an `autolaunch.txt` file at the top of the microSD card does what `SHIFT`+`A` does (the TeensyROM's documentation gives an example). A tester has run PETSCII Doctor 64 on a real C64 from a TeensyROM; its Auto-Launch has not been tried with it yet: please tell us how it goes.

1541 Ultimate II or II+

1. Copy `petscii-doctor-64.crt` to its USB stick or card.
2. On the working C64, open its menu, go to the file, press `RETURN` and choose **Copy to Flash**: it becomes the cartridge it starts at every power-on.
3. To undo it: `F2`, "C64 and Cartridge Settings", "Cartridge".

Not yet tried on a real Ultimate: please tell us how it goes.

An EPROM on an Ultimax cartridge board

Burn `petscii-doctor-64-step1.bin` into a 2764 or a 27C64 on an Ultimax cartridge board, like a Dead Test board: step 1 alone. Nothing to set: it starts by itself and repeats its tests for ever, one green block per good pass.

To try it: VICE, MiSTer, Ultimate 64, C64 Ultimate

VICE: File, Attach cartridge image, `petscii-doctor-64.crt` (or `x64sc -cartcrt petscii-doctor-64.crt`). **MiSTer** (C64 core), **Ultimate 64**, **C64 Ultimate**: load `petscii-doctor-64.crt` as a cartridge. There the C64's chips live inside an FPGA (or are emulated) and every test passes, but one on a MiSTer: the PLA says **?**, `PLA: 1 RARE CASE DIFFERS (A MISTER?)`. Its C64 core maps the memory as the PLA does except in one rare case (a 16K cartridge with $01 = $31: the characters at $D000 instead of memory): nothing to repair. They are for trying the cartridge, not for repairs.

5. Using it, step by step

1. Switch off, plug in the cartridge, switch on. Step 1 runs: the border changes colour at each test, then `GOOD NEWS: THE MEMORY WORKS.` and step 2 starts by itself.
2. Step 2 tests all the memory first (about 30 seconds: a yellow bar fills up), then the rest. Each line gets **OK** (green), **BAD** (red) or **?** (yellow). The border ends green (all good), red (something is broken) or yellow (a doubt).
3. Read the lines under the list: what is broken, in plain words.
4. Look at the drawing of your board (`F1`), or the details (`F3`), or run the tests again (the keys below).
5. Press `SPACE` (or fire on a joystick) for step 3, the tests with you. It ends with a report, then goes back to the menu of its tests: `6 BACK TO THE CHECK-UP` returns to step 2's screen, which then shows their four results too, and their faults among its lines.

The keys on step 2's screen

Key	What it does
`F1`	The drawing of your board, with the chips to change. The first time, it asks which board your C64 has (chapter 7).
`F3`	The details, for experts (chapter 7).
`F5`	Test again.
`F7`	Test again and again, to catch faults that come and go: some chips only fail once warm, so leave it 30 minutes. Hold `F7` to stop (let it go first if you still hold it from the start).
`SPACE`, fire	Step 3, the tests with you. Fire on a joystick in either port works too.

A key that stays down (SHIFT LOCK, a stuck key, a joystick held) is ignored until it is let go. `SHIFT`, `CTRL` and `C=` alone are not keys, as for the C64 itself. A joystick in port 1 is not taken for keys. During step 1, `RESTORE` starts step 1 again.

6. Step 1: the memory

Step 1 runs the way Dead Test does, in Ultimax mode: it needs no working RAM, only the processor, its clock, the PLA and the cartridge port. The screen lists its four tests, `COLOR MEMORY`, `MEMORY CHIPS`, `MEMORY WIRING` and `FIRST 4 KB OF MEMORY`, each with **OK** or **BAD**. After a **BAD**, the next ones do not run. Its title says V0.4: step 1 has not changed since that version.



Step 1 passed. This is step 1 alone, from the EPROM: it repeats, one green block per good pass.
The full cartridge goes on to step 2.

The border colour

The border colour is the test running. If the C64 stops, the border keeps it, even when the screen shows nothing readable. When a test finds a fault, the border flashes (second table).

Border	Test
white	Start: the processor runs and reads the cartridge
yellow	The colour RAM ($D800-$DBFF)
cyan	The memory chips (data lines D0-D7)
purple	The memory's wiring (address lines A0-A11)
orange	Every memory cell of the first 4 KB
green	Step 1 passed (step 2 comes next)

The border flashes	The fault
red and black, slowly (half a second each)	The colour RAM
red and black, fast	The memory's wiring
red, with groups of 1 to 8 white flashes	A memory chip, counted as Dead Test does (chapter 11 gives the chip for each count)
red, steady	Step 1 passed, but step 2 could not start (see below)

What each fault looks like

A broken memory chip. It is shown in red as it sits on the board, for every kind of board:

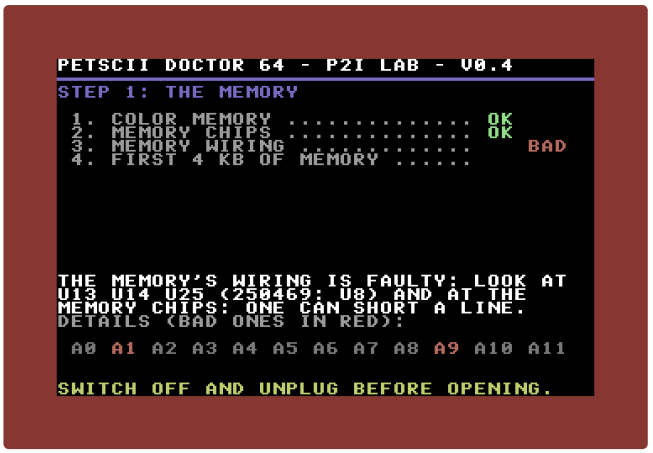
- 8 chips (boards 326298, 250407, 250425), one per data bit: U21 (bit 0), U9, U22, U10, U23, U11, U24, U12 (bit 7);
- board 250466: U10 holds bits 0-3, U9 bits 4-7;
- board 250469: U10 holds bits 0-3, U11 bits 4-7.

The border also flashes as many times as Dead Test flashes its screen, 1 flash for bit 7 (U12) … 8 flashes for bit 0 (U21), for the repair guides written for Dead Test. All 8 in red: check the memory's +5 V, RAS and CAS first, as eight chips rarely die together.

The memory's wiring (address lines A0-A11): the bad lines in red. A memory pin MA0-MA3 carries A0 and A8, A1 and A9, and so on: a bad pin shows as such a pair. Suspects: the multiplexers U13, U14, U25 (74LS257/258; on a 250469 board, U8), or a memory chip shorting the line.



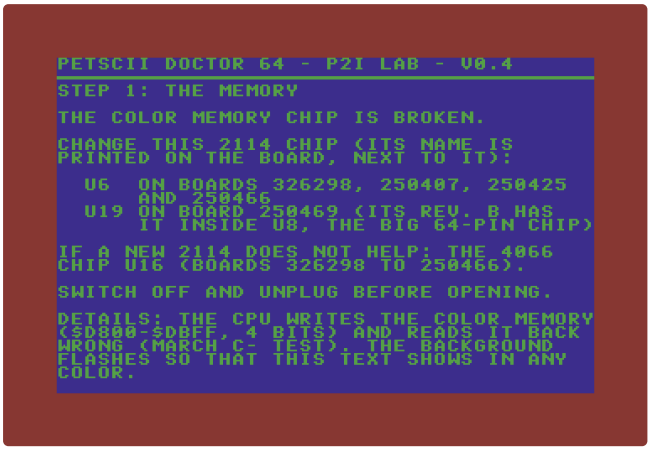
Data bit 7 is broken: U12 on the 8-chip boards, U11 on a 250469, U9 on a 250466. The pages line shows where the errors are.



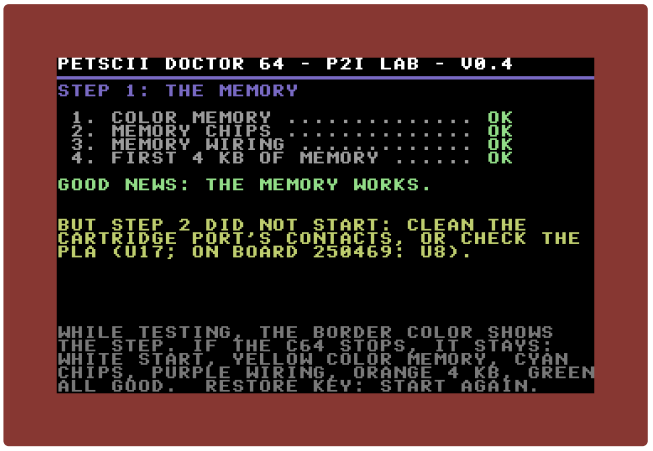
A1 and A9 in red: the pair a memory pin MA1 carries.

The colour RAM has its own screen. Replace the 2114: U6 on boards 326298, 250407, 250425 and 250466, U19 on a 250469 (on its Rev. B, the colour RAM is inside U8). If a new 2114 does not help: the 4066 switch U16. The screen's background flashes, so that its text shows whatever colours the broken chip gives it.

Step 2 did not start (red border, `BUT STEP 2 DID NOT START`): the cartridge could not switch the C64 to 16K mode. Clean the cartridge port's contacts; else suspect the PLA (U17; U8 on a 250469).



The colour RAM is broken: which chip to change on each board.



The memory works, but step 2 did not start.

`RESTORE` starts step 1 again. Step 1 alone (the EPROM) repeats its tests for ever, one green block per good pass.

7. Step 2: the full check-up

Step 2 starts by itself after step 1. It tests all the memory first (about 30 seconds: a yellow bar fills up), then each chip. Each line gets **OK**, **BAD** or **?**.



Every line OK. The keys are at the bottom of the screen.

The list

Line	What it tests	If it is BAD
0. Step 1	The first 4 KB, the colour RAM	(step 2 only runs if they work)
1. All the memory	The RAM $1000-$FFFF, even under the ROMs	A memory chip (`F1` shows which), or the memory's wiring
2. BASIC ROM	Its checksum	The BASIC ROM chip
3. KERNAL ROM	Its checksum	The KERNAL ROM chip
4. Character ROM	Its checksum	The character ROM chip
5. PLA	Which chip answers where, in the 24 memory configurations	The PLA
6. CIA 1	Timers, interrupts (IRQ)	CIA 1 (keyboard, joysticks)
7. CIA 2	Timers, interrupts (NMI)	CIA 2 (serial port, user port)
8. Clock and its 9 V AC	Each CIA's time-of-day clock	Stopped: that CIA; both: their gate (U27; U23 on a 250469), or no 9 V AC (the fuse F1). At a wrong speed: their gate first, then that CIA
9. VIC-II	Raster, its interrupt, sprite collisions	The VIC-II
10. SID	Voice 3 read back	The SID
11. Serial port	CLK and DATA pulled and read back	The 7406 (first unplug the drives)

A ? means: not broken, but not one we know. A ROM with another checksum whose structure is sound is another version (a JiffyDOS KERNAL, a custom character set…): normal if you changed it. Three replacement KERNALs are known, **OK** with their name in the details (`F3`): JiffyDOS, DolphinDOS 2.0 and SpeedDOS Plus 2.7. The PLA test says **?** when too many memory faults leave nothing to compare, and on a MiSTer (its one difference, above). The VIC-II says **?** when the interrupt line (IRQ), which it shares with CIA 1, stays low even with every interrupt off: CIA 1 is then usually **BAD** too, by far the likelier to hold it (if a new CIA 1 does not cure it, the VIC-II may). That **?** is a fault all the same: the summary says `BROKEN` and the border is red.

A SID **replacement** (SwinSID, ARMSID, FPGASID…) may answer **BAD** to the SID test if it cannot give voice 3 back: that is not a fault. The keyboard, the joysticks, the sound and the picture need your eyes and ears: step 3. Not tested yet (they need a plug): the user port and the cassette port.

The lines under the list

- `GOOD NEWS: ALL THESE PARTS WORK.` Then what is still to check with your eyes and ears, or the results of step 3 once you have run it.
- `BROKEN - F1 SHOWS WHERE ON THE BOARD:` then each fault in plain words, in red, and the doubts in yellow.
- `SOME TESTS DID NOT RUN: F5 RUNS THEM.` A loop (`F7`) was stopped before the end of its first round.



CIA 1 is broken. The VIC-II gets a ? because the interrupt line it shares with CIA 1 stays low.

The drawing of your board (F1)

The first time, PETSCII Doctor 64 asks which board your C64 has. Switch off and unplug the C64, open it, and find the number printed on the board (“ASSY NO. 250407”, for example). Then press `1` to `5`:

Key	Board	Which C64
`1`	326298	The first, 1982
`2`	250407	Brown case, the most common
`3`	250425	Brown case, later
`4`	250466	C64C, long board
`5`	250469	C64C, short board

The screen also gives a hint from the SID it found, for example `THIS C64 HAS A 6581 SID: BOARDS 1-4.` (none when the SID’s model could not be told).



Which board is in your C64?



Board 250407: U10 (memory, data bit 3) is broken.

The board is seen from above, its back (the ports) at the top, as you see it with the C64 open in front of you. Each chip carries its name, as printed on the board next to it (U9, U18…), and what it is (CIA, ROM, CPU…):

- a **green** name: tested, it works;
- a **red** chip, blinking: broken, change it (its full name is below the drawing);
- a grey name: not tested (glue chips, the clock circuit…);
- a **yellow** name: in doubt (another ROM version; the VIC-II when the interrupt line is held low; a CIA whose clock runs at a wrong speed).

Boards 326298, 250407 and 250469 are drawn from the positions of their parts. The drawings of 250425 and 250466 are approximate, and say so: the names printed on your board are the right ones. On the drawing, `F1` shows the list of the five boards again, `1` to `5` change the board at once, and the other keys go back.

The details (F3)

For experts: the memory's bad bits, the address lines A12-A15, the first error and a map of the 256 pages (X: a bad cell; green: the 16 pages of step 1); each ROM's checksum (CRC-16) and version; the PLA's first wrong read; the sub-tests of the CIAs, the VIC-II, the SID and the serial port (in red: failed); the clocks' tenths of a second; in a loop, how many passes each test failed. Any key goes back.



The details of a C64 that works: every ROM known (901226-01, 901227-03, 901225-01), a 6581 SID.

Again and again (F7)

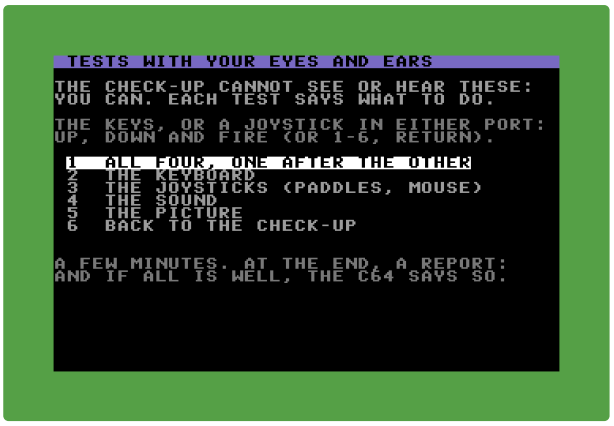
`F7` runs the check-up in a loop, to catch faults that come and go: some chips only fail once warm, so leave it running 30 minutes. The line being tested shows `TEST`; the details (`F3`) count how many passes each test failed. Hold `F7` to stop. Stopped before the end of its first round, the tests not finished show "." (`NOT RUN` in the details), and `F5` runs them all.



Pass 3 of a loop: the SID failed in a pass, the BASIC ROM is being tested.

8. Step 3: the tests with you

SPACE (or fire) on step 2's screen opens their menu: 1 all four, one after the other, or 2 to 5 one of them, 6 back to the check-up. Everything works with the keyboard or with a joystick in either port (up, down, fire): one of them may be broken.



The menu of the tests with you.

The keyboard

The keyboard is drawn as it is, 66 keys. Press each one once: it turns **white** while down, then **green**. **Red**: down since the start, it stays down (stuck); IS SHIFT LOCK ON? if it is LEFT SHIFT. RESTORE too. SHIFT LOCK: lock it a second, then unlock it (it is the same contact as LEFT SHIFT: both light up). Under the drawing: the last key's name and, for experts, its place in the keyboard's matrix (PA, its column on CIA 1's port A; PB, its row on port B).

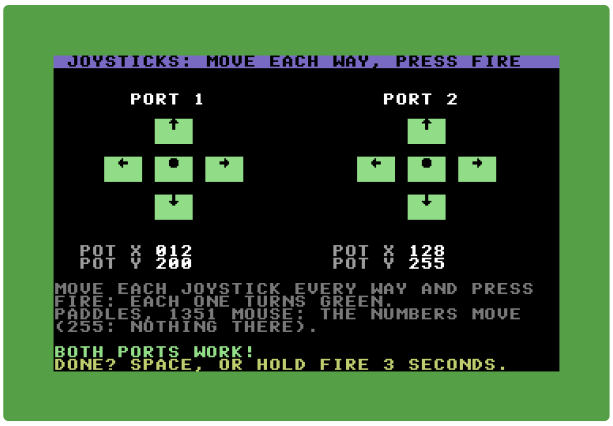
It ends by itself when the 66 have worked; else hold SPACE (or RETURN, or fire) 2 seconds. A stuck key: **BAD**. A whole column or row of keys dead: **BAD** (the keyboard's cable and plug, or CIA 1). Hidden by a joystick held in its port: **?** (unplug it and try again). Keys never pressed: **?** (dead, or not pressed).



All 66 keys work.

The joysticks, paddles and mouse

Two crosses, port 1 and port 2: move each joystick every way and press fire, each part turns green. Paddles and a 1351 mouse: their numbers (POT X, POT Y) move; 255 means nothing is there. It ends when both ports have worked, or with SPACE or RETURN, or fire held 3 seconds. A way that never came, or one on all the time: **?** (the joystick, try another; else the port or CIA 1). No joystick: SPACE or RETURN, and the result is "." (not tested).



Both ports work.

The sound

Turn the volume up. Five parts:

1. Voice 1 plays an A (440 Hz) as triangle, sawtooth, pulse and noise.
2. Voice 2, the same.
3. Voice 3, the same.
4. The filter: low, band and high pass, its frequency going up and down.
5. The three voices together: “Ode to C64” by Jeff (Søren Lund), 2003, a C64 classic, played from its original file with his kind permission, each voice’s meter moving as it plays. The tune runs in the RAM: if the memory test failed, P2i Lab’s arrangement of “In the Hall of the Mountain King” (Edvard Grieg, 1875) plays instead.

A key stops it (Jeff’s tune stops by itself after 3 minutes at most). Then `DID YOU HEAR ALL FIVE?` `Y` yes, `N` no, `A` play them again. After `N`, `WHICH ONE WAS SILENT OR WRONG?` asks for the part (1 to 5): **BAD**, the SID; for the filter, the SID or its two capacitors. Nothing at all: first the TV’s volume, the audio cable and its plugs.



Part 5: “Ode to C64”. Each voice’s meter moves as it plays (here voices 1 and 3).

The picture

The 16 colours, each named (black and white framed); the C64’s own letters, both sets, read by the VIC-II from the character ROM; 8 balls (sprites) going right. `COLOURS AS NAMED, LETTERS, BALLS OK?` `Y` yes, or `N` and what: `1` the colours (the VIC-II, the video cable, the TV’s PAL or NTSC setting), `2` the letters (the character ROM or the VIC-II), `3` the balls (the VIC-II).



The picture test.

The report

The check-up and the four tests, OK, BAD, ? or "." (not run), and for each fault what to check. **If everything works** (the check-up, a ROM of another version allowed, the keyboard, the sound, the picture; the joysticks too, unless you have none), a green banner, and the C64 says "All system tests are OK" in ECCGMS's voice (P2i Lab's voice 1), for about two seconds.



Everything works.



A stuck key: what to do.

Any key or fire goes on, back to the menu of the tests; 6 goes back to the check-up, whose screen then shows the four results too.



Step 2's screen after the tests with you.

9. What each test checks, in plain words

A C64 is a few dozen chips on one board, joined by wires. Each test looks at one of them, or at the wires between them, the way a repairer would with a probe, but from the inside: the processor writes, reads back, and compares with what it should find.

Step 1: the memory, without using it

Every program needs memory: when it is broken, nothing can run, not even a diagnostic program. So step 1, like Dead Test, uses none. It runs in Ultimax mode, straight from the cartridge, and keeps its few values in the video chip's registers. It only needs the processor and its clock, the PLA, the cartridge port, and the VIC-II's registers: step 1 keeps its values there and times its waits on the picture's lines.

On the screen	What it checks	If it is BAD
`COLOR MEMORY`	A small chip of its own (a 2114) keeps the colour of each of the 1000 characters on the screen, 4 bits each. The test fills it with patterns and reads them back in a set order (March C-), which finds a cell that forgets and a cell that changes its neighbour.	Wrong colours, or no readable text: the 2114 (or its 4066 switch: U16; U21 on a 250469).
`MEMORY CHIPS`	A byte is 8 bits, D0 to D7. On most boards each bit lives in a chip of its own: bit 0 in U21, bit 7 in U12 (a C64C has two chips of 4 bits). The test writes 12 patterns (all 0, all 1, every other bit, a single 1 moving along) and reads each one back at once: a bit read back wrong, even once, points to its chip.	That chip, in red on the screen.
`MEMORY WIRING`	To reach a byte, the processor sends its address on 16 wires, A0 to A15; step 1 checks A0 to A11. The memory chips receive it in two halves on 8 pins, MA0 to MA7, through multiplexer chips: pin MA1, for example, carries A1 and then A9. If a wire is cut or touches another, two addresses fall on the same byte: what is written in one shows in the other. The test gives 13 bytes each their own value, and looks for those that share.	The lines in red. A bad pin MA0 to MA3 shows as a pair (A1 and A9, for example); a bad MA4 to MA7 as a single line, A4 to A7: its partner, A12 to A15, is step 2's. The multiplexers U13, U14, U25 (U8 on a 250469), or a memory chip shorting the line.
`FIRST 4 KB OF MEMORY`	Every cell of the first 4 KB, the part the rest of the cartridge needs (its stack, its variables, the screen), with March C-.	The chip of the bad bit, as for MEMORY CHIPS (the PAGES line shows where).

Step 2: the full check-up

Once the first 4 KB are proved, the cartridge leaves Ultimax mode and can use them: it now tests the rest of the C64, one part at a time.

On the screen	What it checks	If it is BAD
`0. STEP 1: 4 KB, COLOR MEMORY`	Step 1's results, again.	(Step 2 only runs when they are good.)
`1. ALL THE MEMORY (64 KB)`	Every byte from 4 KB to 64 KB, even those hidden under the ROMs and the input/output chips. First the four high address lines, A12 to A15; then a stronger test (March LR), with three patterns ($00/$FF, $55/$AA, $33/$CC): the 4-bit chips of a C64C have faults between their own bits that only some patterns show.	A memory chip (`F1` shows which), or the memory's wiring.

On the screen	What it checks	If it is BAD

`2. BASIC ROM`
`3. KERNAL ROM`
`4. CHARACTER ROM`
	The three ROMs hold the BASIC language, the KERNAL (the C64's own system: keyboard, screen, disk) and the shapes of the letters. The test reads each one whole and computes its checksum (CRC-16), a 16-bit fingerprint, compared with the versions we know.	That ROM chip. A `?` is another version that looks sound (a JiffyDOS KERNAL, a custom font).

`5. PLA (MEMORY MAP)`	The processor sees 64 KB at a time, but a C64 holds more: the RAM, three ROMs, the input/output chips, a cartridge. For each address the PLA chooses which one answers, from five signals: LORAM, HIRAM and CHAREN from the processor, GAME and EXROM from the cartridge. The test tries their 24 useful combinations and checks who answers at $8000, $A000, $D000 and $E000.	The PLA (a broken processor port, $01, would look the same).
`6. CIA 1 (KEYBOARD, JOYSTICKS)`	An input/output chip: it reads the keyboard and the joysticks, and its timer gives the system its beat (an interrupt, the IRQ, 60 times a second). The test runs its two timers, checks that each one raises its flag when it ends, and that its interrupt reaches the processor.	CIA 1 (`F1` shows where).
`7. CIA 2 (SERIAL, USER PORT)`	The same chip, for the serial port (the disk drives) and the user port. Its interrupt is the NMI, the one RESTORE also makes.	CIA 2.
`8. CLOCK AND ITS 9 V AC`	Each CIA has a time-of-day clock that counts tenths of a second. It counts the mains' 50 or 60 Hz, which come from the power supply's 9 V AC through a small gate chip. The test sets both clocks, waits one second and reads them: 8 to 12 tenths is right.	One stopped: that CIA. Both stopped: their gate (U27; U23 on a 250469), or no 9 V AC (the fuse F1, the power supply). At a wrong speed: their gate first (its signal may be poor), then that CIA.
`9. VIC-II (VIDEO CHIP)`	The video chip: it draws the picture line by line, and makes the processor's clock. The test counts its lines (312 on PAL, 263 or 262 on NTSC), checks its line interrupt (the raster IRQ) and that it sees sprites touch each other and the text.	The VIC-II. A `?` when the interrupt line it shares with CIA 1 stays low: CIA 1 is the likelier culprit.
`10. SID (SOUND CHIP)`	The sound chip. The processor can read its voice 3 back: the test checks that its noise changes, and that its volume envelope stays near 0 with the note off and climbs to the top with the note on. It also tells the model: a 6581 (the first C64s) or an 8580 (later ones); `?` when voice 3 cannot be read back.	The SID. A SID replacement (SwinSID, ARMSID…) that cannot give voice 3 back may answer BAD: that is not a fault.
`11. SERIAL PORT (DISK DRIVE)`	The port of the disk drives and printers. CIA 2 pulls its two lines, CLK and DATA, down through a 7406 chip: the test pulls each one, reads it back, then lets go, and the line must come back up.	The 7406. Unplug the drives first: a drive can hold a line down.

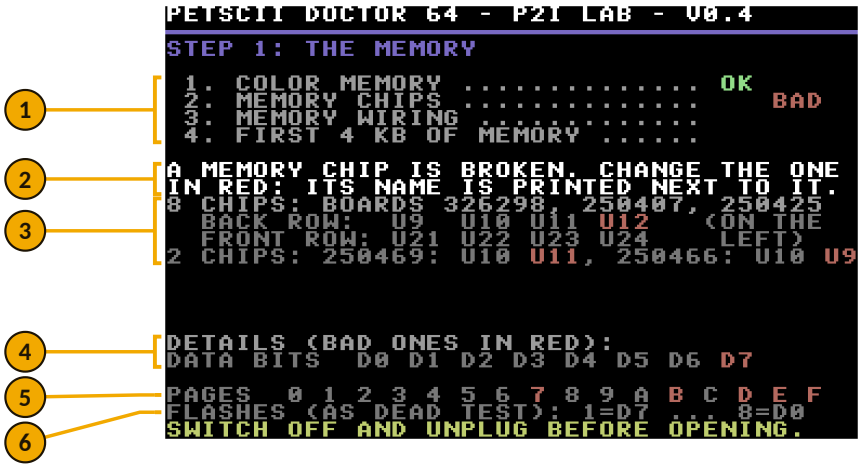
Step 3: what only you can judge

A C64 cannot see its own picture or hear its own sound, and it cannot press its own keys: step 3 asks you. The keyboard test reads the keyboard's matrix through CIA 1 (8 columns by 8 rows: each key joins one column to one row; RESTORE has a wire of its own, and SHIFT LOCK holds LEFT SHIFT's contact). The joystick test reads both ports through CIA 1, and the paddles' or the mouse's numbers through the SID. The sound test plays each part of the SID in turn; the picture test shows what the VIC-II draws: the 16 colours, the character ROM's letters, sprites.

10. The words on the screens

The cartridge's screens are in English, in short words. Here they are, screen by screen: the numbers in the pictures match the lists below them.

Step 1: a memory chip is broken



Step 1 found a broken data bit.

1. The four tests and their result, **OK** or **BAD**; after a BAD the next ones did not run.
2. What is broken, in plain words: here a memory chip, the one in red.
3. Where it is. `8 CHIPS`, on boards 326298, 250407 and 250425: the `BACK ROW` (U9 to U12) and the `FRONT ROW` (U21 to U24), on the left of the board. `2 CHIPS`: on a C64C (250469, 250466). The broken one is in red; its name is printed on the board next to it.
4. `DATA BITS D0 ... D7`: the 8 bits of a byte, the broken one in red (here D7, bit 7).
5. `PAGES 0 ... F`: the first 4 KB, in 16 pages of 256 bytes; in red, the pages where an error was found.
6. `FLASHES (AS DEAD TEST): 1=D7 ... 8=D0`: how many times the border (and the LED) flashes for each bit, as Dead Test counts.

Step 1's other words: `PASSES WITHOUT ERROR` (step 1 alone, from the EPROM: a green block per good pass); on the wiring screen, `A0 ... A11`, the address lines, the bad ones in red, and `U13 U14 U25 (250469: U8)`, the multiplexers.

Step 2: the list and the lines under it

Each line's badge: **OK**, it works; **BAD**, broken; **?**, a doubt (not broken, but not what we know); ".", not run; `TEST`, being tested. In a loop the second row reads, for example, `LOOP, PASS 00003. HOLD F7 TO STOP.`: the pass being run.

Under the list	What it means
`A MEMORY CHIP`	Line 1: F1 shows which, in red; F3 the bad bits and pages.
`MEMORY WIRING: A MULTIPLEXER CHIP`	Line 1: an address line A12-A15 is faulty: U13, U14, U25 (U8 on a 250469), or a memory chip shorting the line.

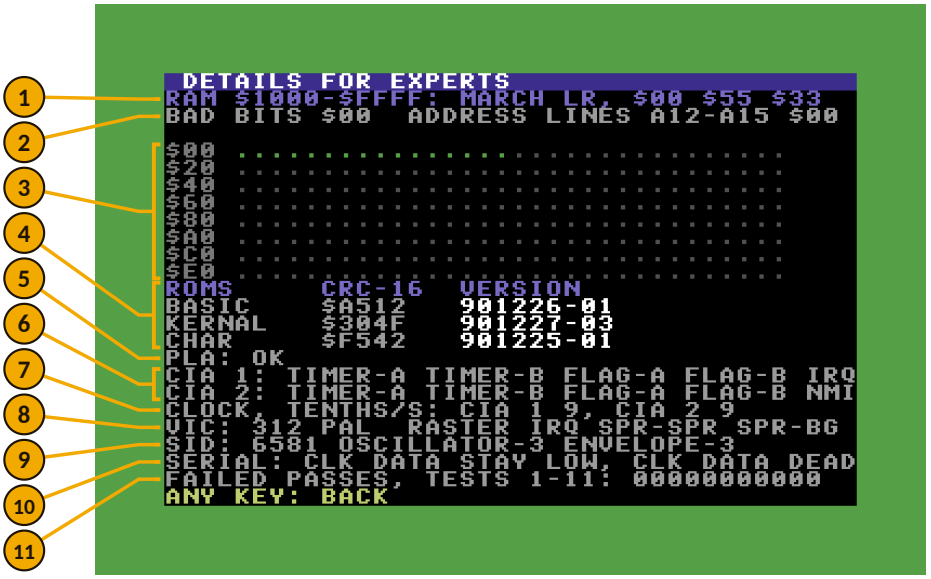
`THE BASIC ROM CHIP`
`THE KERNAL ROM CHIP`
`THE CHARACTER ROM CHIP`
	That ROM matches no version we know, and does not look sound: change the chip.

`THE PLA: IT MIXES UP THE MEMORY`	A wrong chip answered somewhere (F3 shows the first one).
`CIA 1 (KEYBOARD AND JOYSTICK CHIP)` `CIA 2 (SERIAL AND USER PORT CHIP)`	That CIA failed a timer, a flag or its interrupt (F3 tells which).
`CIA 1: ITS CLOCK DOES NOT RUN` `CIA 2: ITS CLOCK DOES NOT RUN`	That CIA's clock is stopped: the CIA.
`CLOCKS STOPPED: THEIR GATE, OR NO 9 V AC`	Neither clock counts: their gate chip, or no 9 V AC (the fuse F1).
`CIA 1: ITS CLOCK RUNS AT A WRONG SPEED` `CIA 2: ITS CLOCK RUNS AT A WRONG SPEED`	That CIA's clock counts, but not 8 to 12 tenths in a second, or it gives an impossible time. On the drawing their gate is red (U27; U23 on a 250469): the signal both clocks count may be poor; and that CIA is yellow. If the CIAs are in sockets, swap them: if the fault follows the chip (the other CIA's clock is now the wrong one), change that chip; if not, look at the gate and the parts around it, and the CIA's pin 19 (its clock input).
`CLOCKS AT A WRONG SPEED: THEIR GATE`	Both clocks count, but not 8 to 12 tenths in a second: the signal from their gate (U27; U23 on a 250469).
`THE VIC-II VIDEO CHIP`	The VIC-II failed a sub-test (F3).
`THE SID (OR A SID REPLACEMENT)`	The SID failed; a replacement may not give voice 3 back.
`SERIAL: 7406 CHIP (UNPLUG DRIVES FIRST)`	A serial line misbehaves: unplug the drives and test again; then the 7406.

`BASIC ROM: A VERSION WE DO NOT KNOW`
`KERNAL: NOT ONE WE KNOW (JIFFYDOS?)`
`CHARACTERS: A FONT WE DO NOT KNOW`
	A doubt, in yellow: another version that looks sound. Normal if you changed that ROM.

`PLA: NOT TESTED (TOO MANY RAM FAULTS)`	A doubt: the memory is too broken to compare.
`PLA: 1 RARE CASE DIFFERS (A MISTER?)`	Not a fault: the memory map of a MiSTer's C64 core, which differs from the PLA's in one rare case only (a 16K cartridge with $01 = $31, the characters at $D000 instead of memory). A real PLA that fails does not fail there only.
`VIC-II: IT MAY HOLD THE IRQ LINE LOW`	A fault all the same: the interrupt line stays low. CIA 1 is usually BAD too, and the likelier culprit.

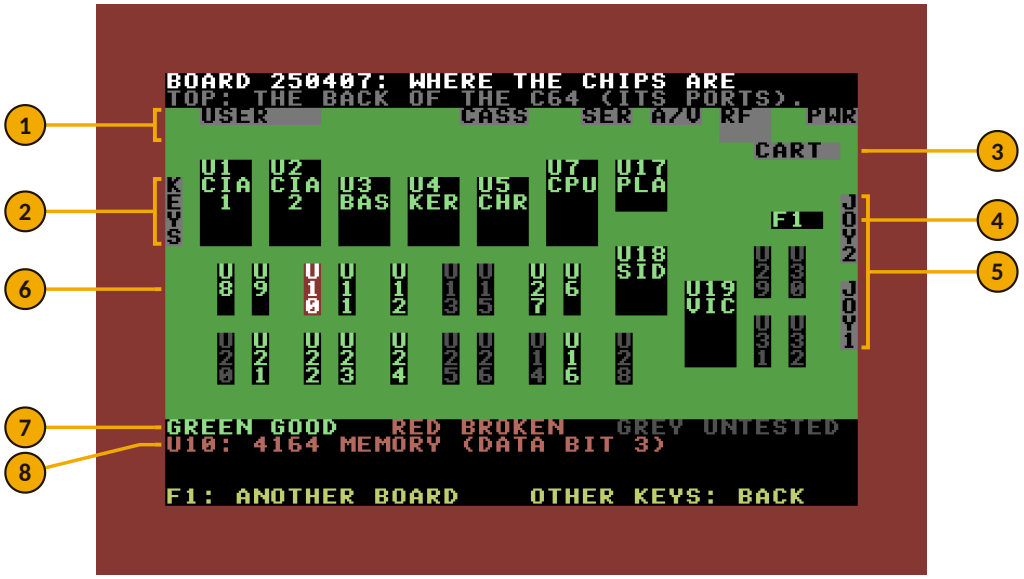
Step 2: the details (F3)



The details of a C64 that works.

1. `RAM $1000-$FFFF: MARCH LR, $00 $55 $33`: the memory tested (from 4 KB to the top), the test (March LR) and its three patterns.
2. `BAD BITS $00`: the broken data bits as a hexadecimal number, adding $01 for bit 0, $02 bit 1, $04 bit 2, $08 bit 3, $10 bit 4, $20 bit 5, $40 bit 6, $80 bit 7; $00, none. `ADDRESS LINES A12-A15 $00`: the broken high address lines, the same way: $10 A12, $20 A13, $40 A14, $80 A15. After an error, the next row gives the first one: `FIRST AT` its address, `WROTE` the value written, `READ` the value read back.
3. The map of the memory: 256 dots, one for each page of 256 bytes, 32 to a row (8 KB); each row starts with its address ($00 for $0000, $20 for $2000…). Green: the 16 pages step 1 tested. An X: a page with a bad byte.
4. `ROMS CRC-16 VERSION`: each ROM's checksum, and the version it matches (its Commodore part number), or `ANOTHER VERSION`, or `BAD`.
5. `PLA:` `OK`, or its first wrong answer: the cartridge's mode (`NO CART`, `8K`, `16K`), the processor's port ($01), the address, what answered and what should have: `RAM`, `ROM`, `CART` or `I/O`.
6. `CIA 1`, `CIA 2`: `TIMER-A` and `TIMER-B`, the two timers count down; `FLAG-A` and `FLAG-B`, each timer raises its flag at its end; `IRQ`, `NMI`, the CIA's interrupt reaches the processor. In red: failed.
7. `CLOCK, TENTHS/S`: how many tenths of a second each CIA's clock counted in one second: 8 to 12 is right; 0, stopped; in red, wrong; `250+`, 25 seconds or more; `???`, an impossible time (a digit above 9): that clock counts wildly.
8. `VIC:` how many lines the VIC-II draws, and its standard (`312 PAL`; 263 or 262 NTSC); `RASTER`, it flags a chosen line; `IRQ`, that line's interrupt; `SPR-SPR`, two sprites touching; `SPR-BG`, a sprite touching the text. In red: failed.
9. `SID:` its model (`6581` or `8580`; `?` when oscillator 3 failed: then it cannot be told); `OSCILLATOR-3`, voice 3's noise read back; `ENVELOPE-3`, voice 3's envelope read back. In red: failed.
10. `SERIAL:` `CLK DATA STAY LOW`, a line that stays down when let go; `CLK DATA DEAD`, a line the C64 cannot pull down. The one in red failed.
11. `FAILED PASSES, TESTS 1-11`: in a loop (`F7`), how many passes each of tests 1 to 11 failed, a digit each (+ for 10 or more).

Step 2: the drawing of your board (F1)



Board 250407, with a broken memory chip.

1. The ports at the back of the C64, at the top of the drawing: `USER`, the user port; `CASS`, the cassette port; `SER`, the serial port (disk drives); `A/V`, audio and video; `RF`, the TV output; `PWR`, the power socket.
2. `KEYS`: the keyboard's plug.
3. `CART`: the cartridge port.
4. `F1`: the fuse of the 9 V AC.
5. `JOY2`, `JOY1`: the joystick ports.
6. The chips: their name (U1, U2…) and what they are: `CIA`; `BAS`, `KER`, `CHR`, the BASIC, KERNAL and character ROMs; `CPU`, the 6510; `PLA`; `SID`; `VIC`, the VIC-II; the memory chips (U9 to U12 and U21 to U24 on this board). Small chips are judged too: U6, the colour RAM, and U16, its switch; U8, the serial port's 7406; U27, the clocks' gate; the multiplexers U13, U14 and U25, green when test 1 passes, red when an address line A12-A15 fails (grey here: with a broken memory chip, the wiring is not proved). Grey: the chips no test judges (glue chips, the clock circuit).
7. `GREEN GOOD`, `RED BROKEN`, `GREY UNTESTED`: what the colours mean (yellow: a doubt).
8. The broken chip's full name: here `U10: 4164 MEMORY (DATA BIT 3)`.

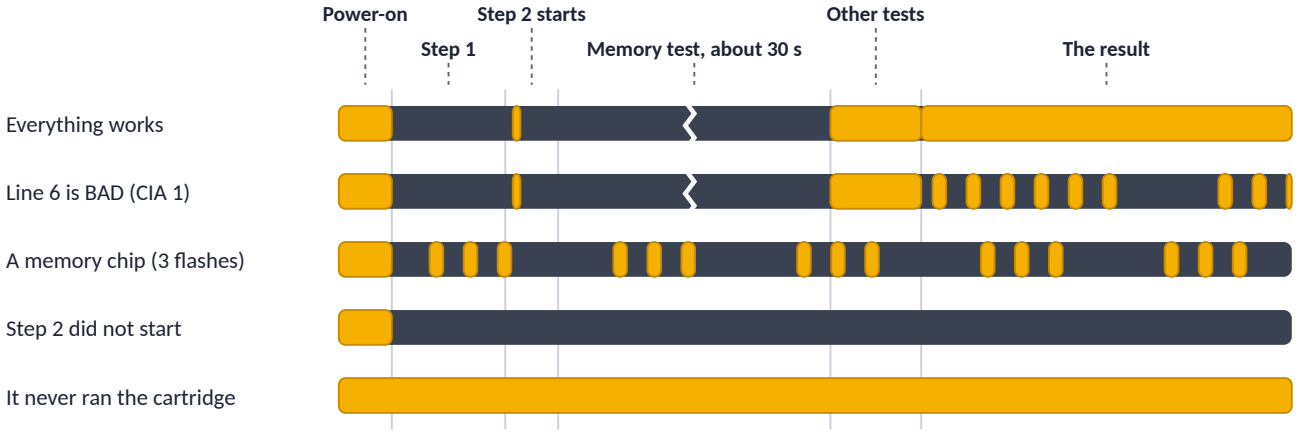
Step 3's screens

- **The keyboard**. `LAST KEY: Q PA7 PB6`: the last key pressed, and where it sits in the keyboard's matrix: `PA`, its column (a line of CIA 1's port A, 0 to 7); `PB`, its row (port B). A whole column or row dead points to that line, its pin on the keyboard's plug, or CIA 1. `KEYS LEFT`: how many keys are still to press.
- **The joysticks**. `POT X`, `POT Y`: the port's two analogue inputs, read by the SID, for paddles or a 1351 mouse (0 to 255; 255, nothing plugged in).
- **The sound**. `VOICE 1` to `3`: the SID's three voices; `TRIANGLE, SAW, PULSE, NOISE`: its four waveforms, the shapes of its sound (soft, buzzing, hollow, a hiss); `FILTER: LOW, BAND AND HIGH PASS`: its filter keeps the low, the middle or the high sounds; `NOW:` the part playing.
- **The picture**. The 16 colours, named; the letters, both sets of the character ROM (capitals and graphics; small letters and capitals); the balls are sprites, small pictures the VIC-II draws over the screen.
- **The report**. `THE CHECK-UP (11 TESTS)`: step 2's result.

11. The cartridge's LED

The LED of a Kung Fu Flash, or of an EasyFlash, tells what the screen would, for a C64 that shows **no picture at all** (a dead video output, a bad cable). It speaks during steps 1 and 2 only: the tests with you need the screen anyway. Other devices may not show it (the Ultimate, the TeensyROM, VICE).

From power-on to the result



Five C64s, one a lane, not to scale. The first box is the Kung Fu Flash's LED at power-on, lit (an EasyFlash's is dark there). Step 1 takes a few seconds, in the dark. When it passes, the LED blinks once, briefly, as step 2 starts, stays dark during its memory test (about 30 seconds; less when the memory is broken), lights for the other tests (a few seconds; it may flicker), then gives the result until a key is pressed.

What is your LED doing?

1 Right after power-on, the LED... (a Kung Fu Flash's; an EasyFlash's is dark from the start: go to question 2)

never changes	The processor never ran the cartridge: the VIC-II (it makes the processor's clock), the clock chip, the 6510, the PLA, the power supply, the cartridge port.
goes dark	Step 1 runs: go on to question 2.

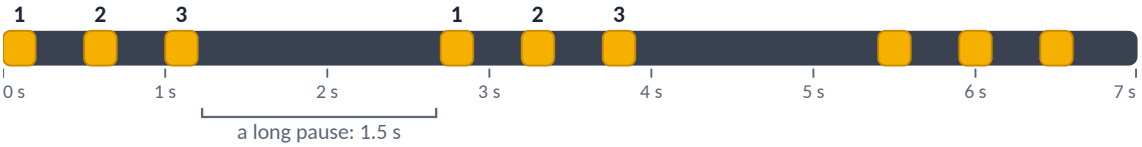
2 During the next seconds (step 1), it...

blinks slowly half a second lit, half dark	The colour RAM is broken.
blinks fast about three times a second	The memory's wiring is faulty.
blinks in groups 1 to 8 flashes, a long pause	A memory chip is broken: count the flashes, then find the chip in the first table below.
blinks once, briefly then stays dark about 30 s	Step 1 passed, and step 2 tests the memory: go on to question 3.
stays dark a minute or more	Step 2 did not start: the cartridge port, the PLA. Or step 1 hangs: the VIC-II, whose registers it needs. On an EasyFlash it can also mean that the processor never ran the cartridge: a Kung Fu Flash tells them apart.

3 **After the memory test (about 30 seconds dark; less when it finds a fault) and a few seconds lit, it…**

	**stays lit**	The check-up found nothing BAD. A doubt (?) does not show on the LED.
	**blinks in groups** 1 to 11 flashes, a long pause	A line of step 2's list is BAD: count the flashes, then find the line in the second table below.

How to count the flashes



Each flash lasts 0.2 s, 0.3 s apart. After the last one of a group comes a long pause (1.5 s), then the group again. Count the flashes between two long pauses: here 3. (On an NTSC C64 it all goes a little faster.) In step 1, when several bits are broken, their groups take turns, one for each bad bit: on a C64C, a broken chip of 4 bits gives up to four groups in turn (for example 1, 2, 3 and 4 flashes), all for the same chip in the table. All eight groups in turn: first check what the eight chips share, their 5 V and their RAS and CAS signals; eight chips rarely die together.

Step 1: which memory chip

Flashes	Data bit	Boards 326298, 250407, 250425	Board 250466	Board 250469
1	7	U12	U9	U11
2	6	U24	U9	U11
3	5	U11	U9	U11
4	4	U23	U9	U11
5	3	U10	U10	U10
6	2	U22	U10	U10
7	1	U9	U10	U10
8	0	U21	U10	U10

Step 2: which line

The first BAD line of step 2's list: see chapter 7 and chapter 9 for what it tests and what to change.

Flashes	Line BAD	Flashes	Line BAD	Flashes	Line BAD
1	All the memory	5	PLA	9	VIC-II
2	BASIC ROM	6	CIA 1	10	SID
3	KERNAL ROM	7	CIA 2	11	Serial port
4	Character ROM	8	Clock and its 9 V AC		

12. When the screen stays black

Often no cartridge can run then: Dead Test and DesTestMAX need the same chips. But only the picture may be missing (its output, the cable, the TV): the cartridge's LED then gives the results (chapter 11). Look, in this order:

1. **The border**: any colour from step 1's table tells how far the C64 went (chapter 6). No picture at all: **the cartridge's LED** (chapter 11).
2. **The power supply**: the power LED, then 5 V and 9 V AC on the user port (chapter 2).

3. **The fuse** F1 inside the C64 (9 V AC).
4. **The cartridge port**: is the cartridge pushed in all the way? Clean the contacts.
5. **The video**: the cable, the TV or monitor's input; try the other output (RF or composite).
6. **The chips every program needs**: the VIC-II (it also makes the processor's clock: no VIC-II, no clock), the clock chip on later boards (8701), the 6510 (8500 on a 250469), the PLA (906114-01; on a 250469, 251715-01, or 252535-01 which also holds the colour RAM). A chip much hotter than the others is suspect; socketed ones can be swapped with a C64 that works.

13. Where the chips are, board by board

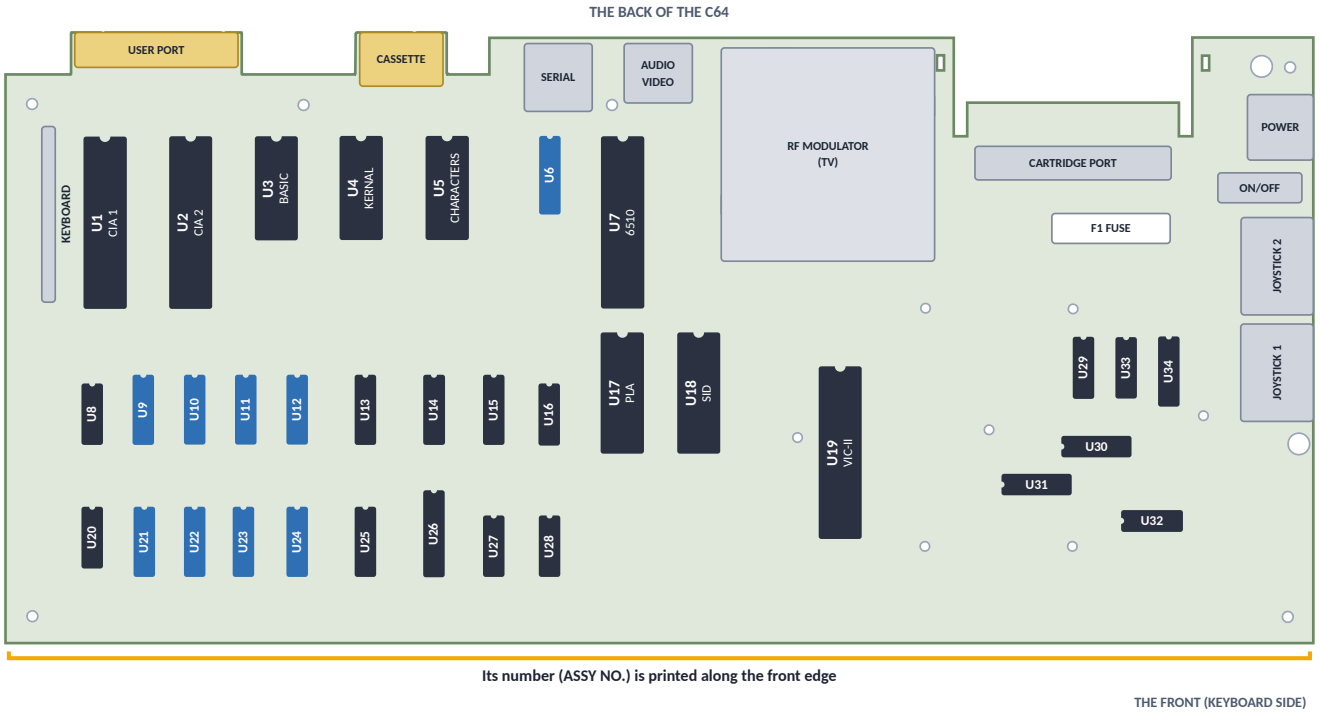
The parts this manual names, on the five boards PETSCII Doctor 64 knows. The drawing (F1) shows all the others (the ROMs, the CIAs, the VIC-II, the SID, the processor) on your board, and so do the plans below.

Part	Boards 326298, 250407, 250425	Board 250466	Board 250469
Which C64	326298: the first, 1982; 250407: brown case, the most common; 250425: brown case, later	C64C, long board	C64C, short board
Memory chips	8 chips, one per data bit: U21 (bit 0), U9, U22, U10, U23, U11, U24, U12 (bit 7)	2 chips: U10 (bits 0-3), U9 (bits 4-7)	2 chips: U10 (bits 0-3), U11 (bits 4-7)
Colour RAM (2114)	U6	U6	U19 (Rev. B: inside U8)
Colour RAM's switch (4066)	U16	U16	U21
Address multiplexers (74LS257/258)	U13, U14, U25	U13, U14, U25	U8
PLA	U17 (906114-01)	U17 (906114-01)	U8 (251715-01, or 252535-01 which also holds the colour RAM)
The CIAs' clock gate (from the 9 V AC)	U27 (7408) and its transistor	U27 (7408) and its transistor	U23 (74LS14)

The five boards, seen from above

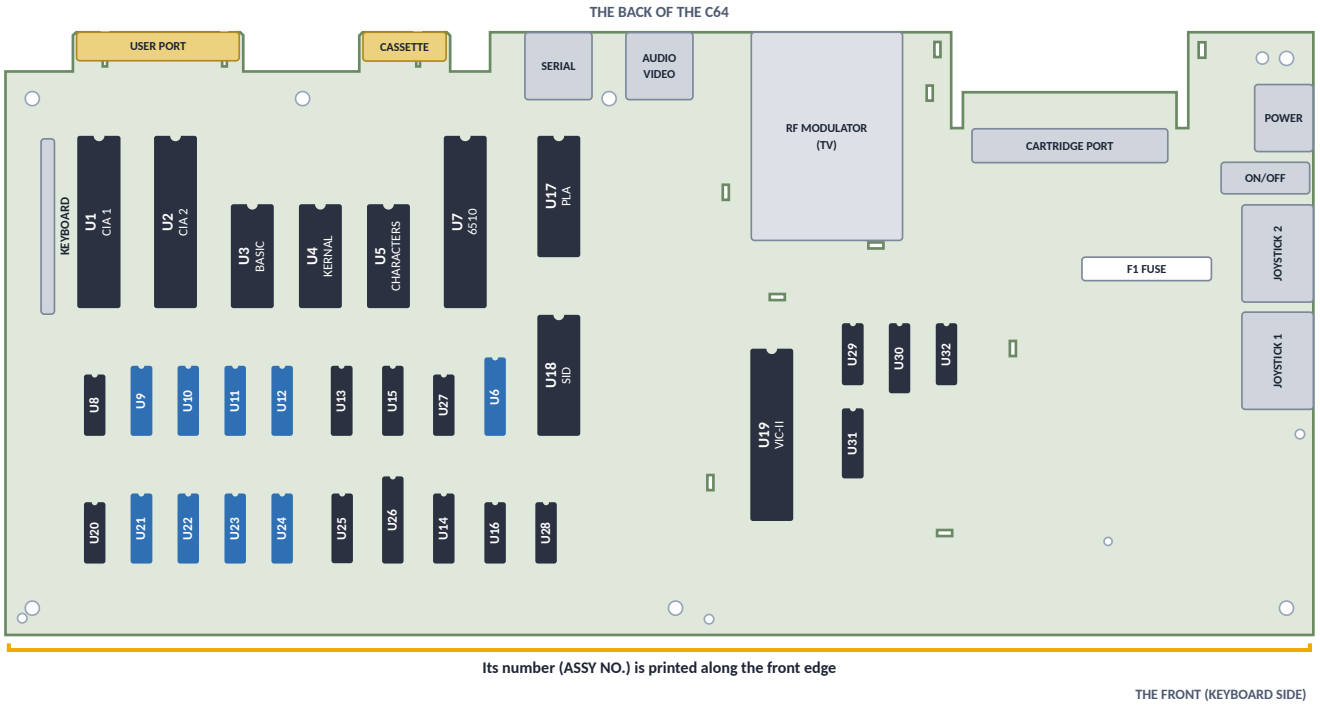
Each board as you see it with the C64 open in front of you: its back, with the ports, at the top. Blue: the memory chips and the colour RAM. Every chip's name (U1, U2…) is also printed on the board, next to it; the notch at one end of a chip marks its pin 1. The board's own number (ASSY NO.) is printed on it, near its front edge; it may be hidden under the shield that covers the board.

326298: the first C64s (1982)



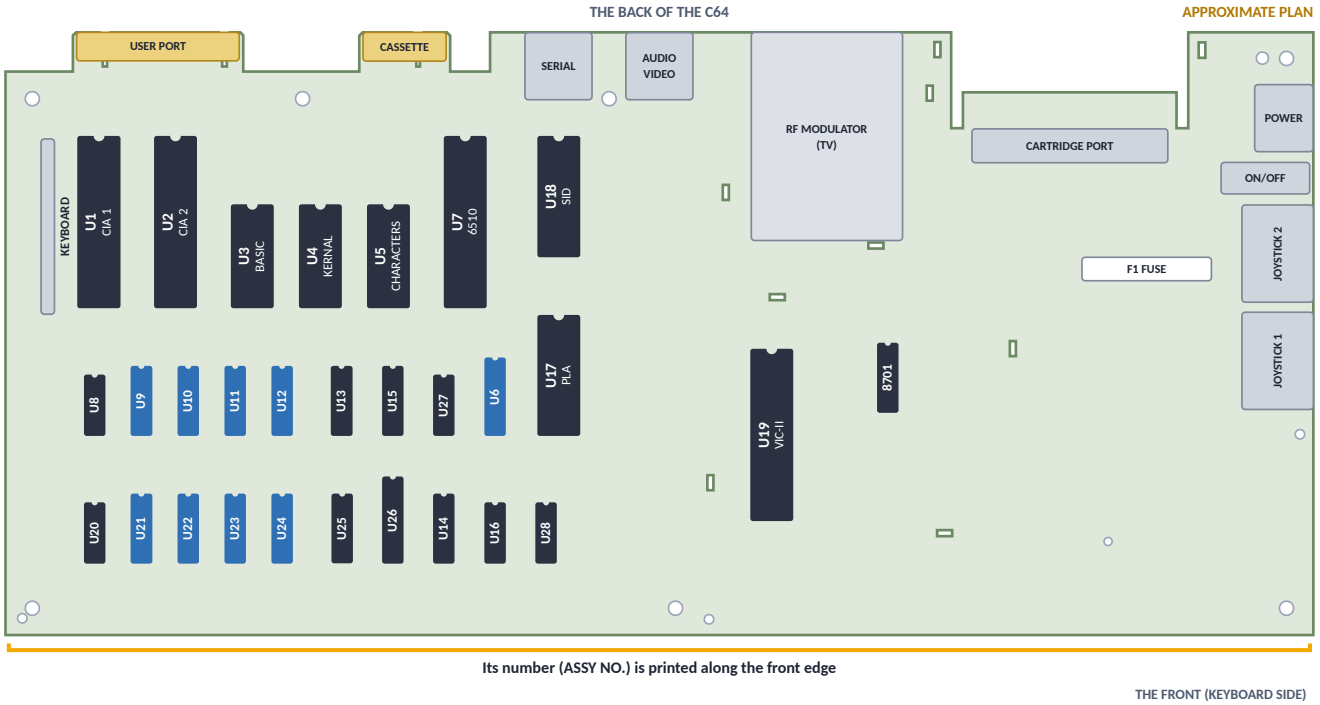
To scale. Eight memory chips in two rows, on the left; the colour RAM, U6, beside the processor; the clock circuit, small chips on the right; a 5-pin audio/video socket.

250407: brown case, the most common



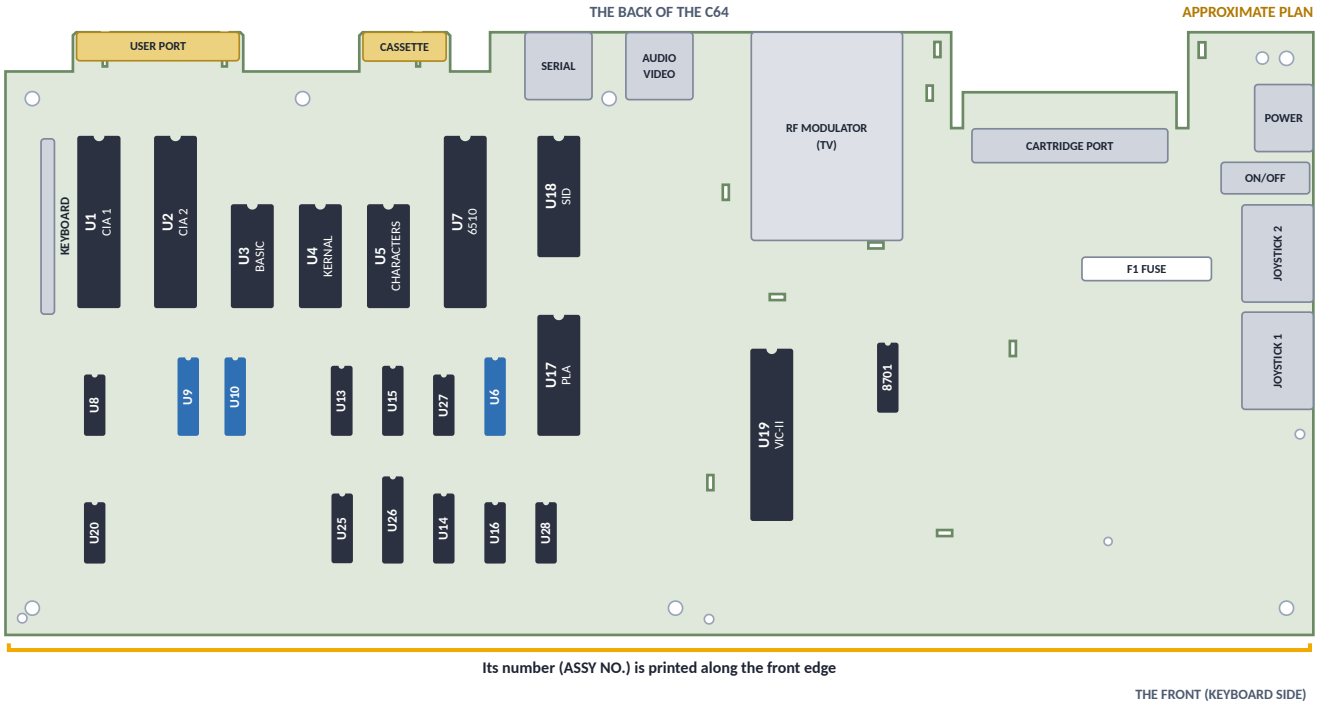
To scale. Eight memory chips in two rows; the colour RAM, U6, among the small chips; the PLA near the serial port; the clock circuit, four small chips (U29 to U32) beside the VIC-II; an 8-pin audio/video socket.

250425: brown case, later



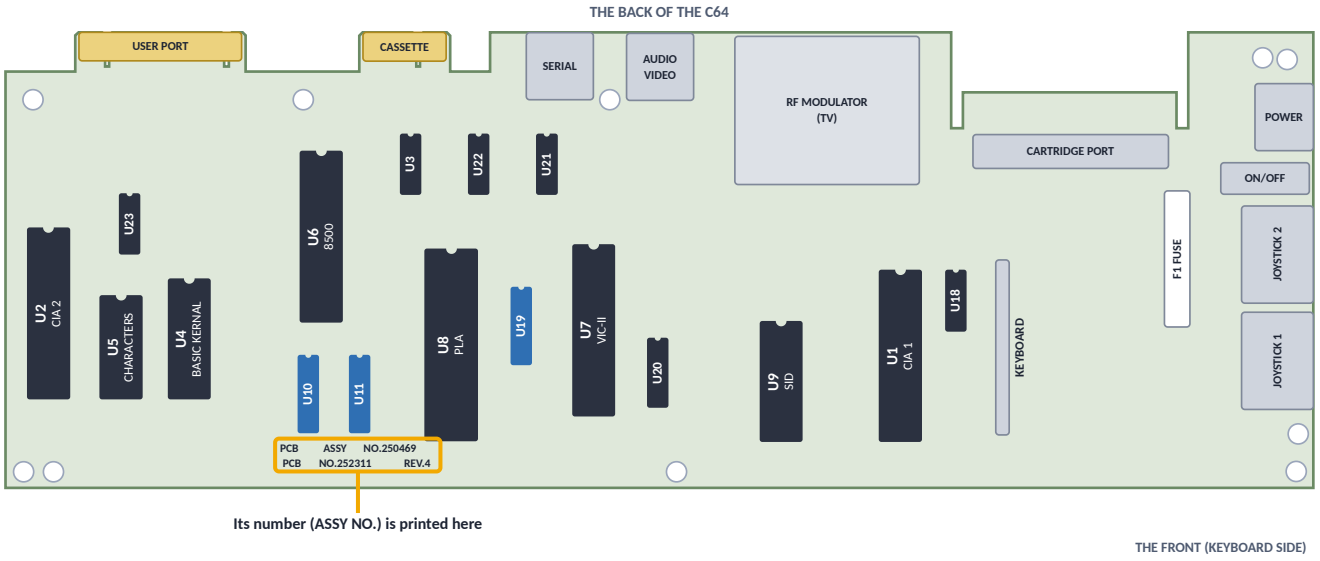
Approximate: drawn from the 250407, with what changed. One clock chip, the 8701, instead of four; the PLA and the SID swapped; the VIC-II no longer in a metal can.

250466: C64C, long board



Approximate, like the 250425. Two memory chips of 4 bits (U9, U10) instead of eight, which leaves an empty space on the left, at the front.

250469: C64C, short board



To scale (a Rev. 4). Shorter than the others; two memory chips of 4 bits (U10, U11); a large PLA of 64 pins, U8; the fuse upright beside the cartridge port. Its number: in the orange frame.

The plans give each part the place it has in the boards' KiCad replicas: 326298 Rev. A by Bo Herrmannsen, 250407 by bwack and Der Alte Bastler, 250469 by bwack. The differences of the 250425 and the 250466 come from Dave Farquhar's and Breadbox64's pages on the boards, and from Ray Carlsen's list.

14. The C64's parts and other words

6510	The C64's processor (an 8500 on the C64C short board, 250469).
VIC-II	The video chip. It also makes the processor's clock: without it, nothing runs.
SID	The sound chip: a 6581 in the first C64s, an 8580 in later ones.
CIA 1, CIA 2	The two input and output chips. CIA 1 reads the keyboard and the joysticks; CIA 2 drives the serial port (the disk drives) and the user port. Each has a time-of-day clock, paced by the 9 V AC.
PLA	The chip that decides which chip answers at each address (the memory map).
ROMs	BASIC, the KERNAL (the C64's own system) and the character ROM (the shapes of the letters).
Colour RAM	A small memory chip (2114) that holds the colour of each character on the screen.
Data lines, address lines	The wires between the processor and the memory: the bits of a byte (D0-D7), and of its address (A0-A15).
Ultimax mode	A cartridge mode in which the processor only sees 4 KB of RAM, the input and output chips and the cartridge. Dead Test and step 1 run in it, so they need almost nothing to work.
EasyFlash	A flash cartridge, and its file format: the Kung Fu Flash, the EasyFlash 3, the TeensyROM and the Ultimate can load it too.
March test	A way to test memory: every cell written and read in a set order, which finds faults a simple pattern misses (step 1 uses March C-, step 2 March LR).
PAL, NTSC	The European and American TV standards: a C64 of each has its own clock and picture.

15. Tell us what it found

The most useful report: what PETSCII Doctor 64 said, and what the real fault was. Tell us also:

- your C64: its model and the number of its board (printed on it: ASSY NO.), PAL or NTSC;
- how you started it: Kung Fu Flash, EasyFlash, TeensyROM, Ultimate, EPROM…;
- what the screen showed: the text, the border colour, the LED (a link to a photo helps a lot).

The form (Google Forms, no account needed):
docs.google.com/forms/d/e/1FAIpQLSfHFYOXpPCF9X_YEhc6v1TzR9DOVDiAjtrNWo6OOyzTzWJvYQ/viewform

The page, with the latest version and its comments: p2i-lab.com/c64/petscii-doctor-64.html

16. The music, the licence, thanks

"Ode to C64" by Jeff (Søren Lund), 2003, released by Viruz, plays in the sound test with his kind permission (2 November 2026, relayed by Klas). The tune belongs to Søren Lund and is not covered by PETSCII Doctor 64's licence. His permission covers PETSCII Doctor 64 in its current form only, as P2i Lab gives it away, free of charge (its cartridge image and its source, copied and shared for free). Selling PETSCII Doctor 64 with the tune, on a cartridge or in any other form, is not covered: that needs his own permission, as does any other use of the tune (`musique/README.md`). Thank you, Jeff.

PETSCII Doctor 64: © 2026 P2i Lab, BSD 3-Clause licence (`LICENSE`). Written from scratch: no code from Commodore's Dead Test or diagnostic cartridges. The voice is ECCGMS's (P2i Lab's voice 1). "In the Hall of the Mountain King": Edvard Grieg, 1875, in the public domain; the arrangement is P2i Lab's.

The facts about the boards and the chips come from the C64-Wiki, Ray Carlsen's list of boards, the KiCad replicas of boards 326298 (Bo Herrmannsen), 250407 (bwack, Der Alte Bastler) and 250469 (bwack), which also give the plans their places, Dave Farquhar's and Breadbox64's pages on the boards, the source of VICE, that of the Kung Fu Flash, the TeensyROM's documentation, and the Commodore 64 Programmer's Reference Guide (all cited in `README.md`). Commodore 64, Kung Fu Flash, EasyFlash, TeensyROM, Ultimate, MiSTer, Dead Test and the other names belong to their owners.

Appendix: for experts, how each test works

The short version; `README.md` has the rest.

Step 1, without RAM

- **Ultimax mode**: the 6510 sees 4 KB of RAM ($0000-$0FFF), the I/O and the cartridge ($E000-$FFFF). No stack and no zero page until they are tested: the few values kept live in the VIC-II's sprite registers ($D000-$D00F, harmless while the sprites are off); subroutines return with `JMP ($D00E)`, one level deep (there is no stack).
- **The screen needs no RAM**: in Ultimax mode the VIC-II reads the cartridge's top 4 KB at $3000-$3FFF, so the screens and the font come from the ROM. The colour RAM, tested first, shows and hides the fixed texts.
- **The tests**: the colour RAM, March C- on its 1024 cells of 4 bits. The data lines: 12 patterns ($00, $FF, $55, $AA, a walking 1) written and read back at once in 13 cells. The address lines: those 13 cells ($0F00, and $0F00 with one address bit flipped) each get their own byte; a cell that reads another's byte shares it, and their different address bits are the bad lines; then again around $00FF, so that two shorted lines both show. Every cell: March C- ($00 and $55) on the zero page and the stack, then on $0200-$0FFF.
- **On the EasyFlash**, step 1 is bank 0 (the EasyFlash starts in Ultimax mode). Once it passes, a few bytes copied to $0200 select bank 1 and 16K mode, check step 2's signature and start it.

Step 2, the full check-up

- **16K mode**: the program at $8000-$BFFF uses only the proved 4 KB; a RAM kernel at $0C00 runs what needs the cartridge off. `RESTORE` cannot get in the way: CIA 2's NMI is never acknowledged, so the NMI line stays low.
- **RAM** $1000-$FFFF (cartridge off, $01 = $30: all RAM, also under BASIC, the I/O and the KERNAL): first the address lines A12-A15, then March LR {(w0); dn(r0,w1); up(r1,w0,r0,w1); up(r1,w0); up(r0,w1,r1,w0); up(r0)}, which also finds linked faults, with three backgrounds: $00/$FF, $55/$AA, $33/$CC (the 4-bit chips of the C64C boards have faults between their own bits that only some patterns show).
- **ROMs**: CRC-16/CCITT against the versions we know: BASIC 901226-01; KERNAL 901227-01, -02, -03, 251104-04 (SX-64), 390852-01 (C64 GS), 901246-01 (Educator 64), 906145-02 (Japan); characters 901225-01, 906143-02 (Japan). Another checksum: **?** if the structure is sound, else **BAD**.
- **PLA**: for the cartridge off, 8K and 16K, times the 8 values of LORAM, HIRAM and CHAREN (24 configurations), the 6510 reads $8000, $A000, $D000 and $E000 and must find what the PLA's equations say: RAM, a ROM, the cartridge or the I/O. (A broken 6510 port, $01, would look the same.)
- **CIAs**: timers A and B count down; one-shot ends set the interrupt flags; CIA 1's interrupt reaches the 6510 (IRQ), CIA 2's too (NMI).
- **Clock**: both time-of-day clocks set to 1:00:00.0, then read after 50 frames (PAL) or 60 (NTSC), minutes, seconds and tenths: 8 to 12 tenths is right. A digit above 9 (above 5 for the tens) is an impossible time (`???`); 25 seconds or more show as `250+`. One at 0: that CIA; both: their gate, or no 9 V AC. On a breadbin (brown case) that shows a picture, the 9 V AC is there (the VIC-II's 12 V is made from it): look at the gate first. One at a wrong speed: their gate, and that CIA in doubt; both: their gate.
- **VIC-II**: the highest raster line (311 on PAL, 262 or 261 on NTSC), the raster flag and its interrupt, sprite collisions.
- **SID**: voice 3's noise read back changes; its envelope ($D41C) stays near 0 with the gate off, then reaches the sustain level 15 ($FF) with the gate on; the model from oscillator 3 right after a restart (3 on a 6581, 2 on an 8580); none when oscillator 3 failed (`?`).
- **Serial port**: CIA 2 pulls CLK, then DATA, through the 7406, and reads them back; let go, they must be high.

Step 3, the tests with you

- **Keyboard**: port 2 read with no column chosen, port 1 before and after the 8 columns, so that a joystick is not taken for keys. `RESTORE`'s NMI is let through for this test only. SHIFT LOCK: LEFT SHIFT's contact held a second.
- **Paddles**: CIA 1 chooses the pair the SID reads ($D419, $D41A), a port a frame.
- **Sound**: A4 (440 Hz) on each voice, 25 frames each waveform; the filter with voice 1's sawtooth A2. Jeff's tune: its own player, copied to the RAM and called each frame. The SID's registers cannot be read back, so for the meters its PLAY runs twice a frame: once with the RAM everywhere ($01 = $34, its writes to the SID landing in the RAM under the I/O), its bytes then put back, and once for the SID, which gets exactly the same writes.
- **Picture**: the character ROM's two sets switched in on the way down the screen, 8 sprites.
- **The voice**: "All system tests are OK." from ECCGMS's model of the C64's voice synthesizer, 4-bit values sent to the volume ($D418) every 152 cycles (158 on NTSC); on an 8580, its three voices are held at a level that the volume modulates, else the voice would be very faint.