

DesTest-KSP
(Keyboard-Stick-Paddle)
[and Mouse] [and Light-pen]

Matthew Desmond
factorofmatt.com

Table of Contents

Introduction.....	3
Building a DesTestKSP Cartridge.....	3
Running the Diagnostics.....	4
Keyboard Test.....	5
Joystick Test.....	7
Paddle Test.....	8
Mouse Test.....	9
Lightpen Test.....	9

Introduction

DesTest-KSP (Keyboard-Stick-Paddle), as its name suggests, is a test program for the C64 Keyboard, Joysticks and Paddles. And mice. And lightpens.

'KSP is a one-stop-shop for testing most** human-interface devices that can be plugged in to your C64, C128 or Ultimax machine.

All versions are functionally identical with the slight exception that the Ultimax version does not attempt to display Time-Of-Day information for its non-existent CIA#2. Additionally, the labelling for the control-ports is Front/Back for the C64 and Left/Right for the 'MAX.

** Perhaps not most, just the ones I have. Except the SNES adapter that came with PETSCII Robots. I may add that one day.

Note: *This release isn't really a release – consider it more of a teaser for a release yet-to-come. The keyboard, joystick and paddle tests work just about perfectly. The mouse test will certainly indicate if your 1351 mouse is good or not but the crosshair will have erratic behaviour if the mouse is moved too quickly. The lightpen test will also at least confirm that the circuitry is working though there is currently no mechanism to calibrate the location of the lightpen with the position of the crosshair. These issues are expected to be addressed before the first 'proper' release.*

Building a DesTestKSP Cartridge

Three variants of the DesTestKSP binary are provided in the distribution archive:

- `destest-ksp-exrom-c64.rom`
- `destest-ksp-game-c64.rom`
- `destest-ksp-max.rom`

Filename	Executable Address	/GAME (CNT)	/EXROM (/CIAX)	/IO2 (/CIA)	/ROML	/ROMH
		(pin 8)	(pin 9)	(pin 10)	(pin 11)	(pin B)
<code>destest-ksp-exrom-c64.rom</code>	\$8000	High	Low	N/C	Yes	No
<code>destest-ksp-game-c64.rom</code>	\$E000	Low	High	N/C	No	Yes
<code>destest-ksp-max.rom</code>	\$E000	N/C	Tie to 10	Tie to 9	No	Yes

A Versa64Cart or equivalent is an ideal platform upon which to build a DesTestKSP cartridge. For the C64 just the /EXROM or /GAME and /ROML or /ROMH jumpers need to be correct and you'll be well on your way.

There is no practical difference between the /EXROM and /GAME images – they function the same way. The choice to use one over the other may simply be down to what configuration of cartridge you happen to have at hand.

The story for the UltiMAX image is a little different. While mostly compatible, the expansion ports for a C64 and a MAX machine are subtly different. The MAX machine has no /GAME or /EXROM signals (since all MAX cartridges are, by definition, /GAME carts – it is the “GAME” system to which that signal refers). Additionally, MAX cart pins 9 and 10 must be tied together in order for the CIA to work properly. Lastly, the PROM chip-select should be tied to /ROMH (pin **B**).

Running the Diagnostics

The first and only screen you'll see when you run DesTest-KSP is shown below.



Towards the top of the screen is a continuous display of the Time Of Day information coming from each of the two CIAs (or the single one for an UltiMAX). All of the input devices tested with this program use the CIA inputs to some degree or another so times that don't count-up, make no sense (are garbled) or are significantly different from each can be a strong indication of a bad CIA or associated logic circuitry. For the C64 the displayed times will be yellow if they appear OK and red if they do not. It is normal for them to flash red on occasion.

Next on the screen is a simple depiction of the the C64 keyboard.

Below the keyboard are the control-port indicators.

Lastly comes the scrolly-message. This message is really just intended to remind you of the special key-combinations used to test specific control-port peripherals.

Key-combinations? In a keyboard test? Yes. But it isn't nearly as daft as it seems. The keyboard is not needed in order to decently test joysticks (or 1350 joy-mice), paddles and 1351 Mice in both control ports.

The keyboard-combinations allow switching between which kind of device and which control port is in charge of updating the crosshair sprite. So you only need the keyboard to work (and then only certain keys) if you like the visual-feedback of the crosshair or want to test a light-pen.

All key combinations require pressing either of the two meta-keys (CTRL or C= (the Commodore key)) and a specific key depending on the required action:

Key Combination	Action (Crosshair mode)
C= + S or CTRL+S	Select "Stick" crosshair mode. The crosshair is placed in joystick mode.
C= + P or CTRL+P	Select "Paddle" crosshair mode. The crosshair is placed in paddle mode.
C= + M or CTRL+M	Select "Mouse" crosshair mode. The crosshair is placed in mouse mode.
C= + L or CTRL+L	Select "Light-pen" crosshair mode. The crosshair is placed in light-pen mode.
C= + 1 or CTRL+1	Select Control Port 1 (Front or Left). This port controls the crosshair.
C= + 2 or CTRL+2	Select Control Port 2 (Rear or Right). This port controls the crosshair.
C= + F1 or CTRL+F1	Cycle through the crosshair -modes and ports.

You can see the currently selected control-port and device-mode by looking directly between the two control port indicators:

"Fr" for Front, "Bk" for Back (or "Lf" or Left or "Rt" for Right on the MAX).

"Stick" for joystick, "Paddle" for paddle, "Mouse" for mouse and "Lt-Pen" for light-pen.

Keyboard Test

The keyboard test display is the central-focus of the KSP suite and potentially the part that will be used the most. Stubborn, sticky or plain-broken keys are most vexing and a good keyboard test will aid immeasurably in diagnosing any fault.



All of the keys on a C64 are represented in the keyboard display, though some short-forms are used for some of the key-caps:

Display Legend	Real Key Legend
HO	HOME
DL	DEL
CTRL	CTRL or CONTROL on some C64Cs (who knew?)
REST	RESTORE
ST	STOP
SL	SHIFT LOCK (two lines, one meaning. Unlike the other two-line keys)
SHFT	SHIFT

Note: The key-caps displayed are that of the unshifted keys only.

All of the keys start off being displayed in light grey. Any key that is actively being pressed will be displayed in yellow. Once the key is released it will be displayed in Black. Thus it is possible to tell the difference between what hasn't been pressed, what has been pressed and what is currently being pressed. The same is true for the Control Port displays too.

The RESTORE key can be pressed and released just like any other key, though due to its nature it is handled a little differently: The screen display will act like you've released the RESTORE key

even if you hold it down. This is due to how the RESTORE key is indirectly wired to the NMI control signal not into the CIA as with all other keys.

Note: The RESTORE key may require a short, sharp “stab” to activate rather than a regular “press”. This is absolutely 100% by design on all breadbin models of the C64 and you are by no means doing anyone any favours by “fixing” it to work more like a C64C. It is the C64C boards that are broken by comparison, not the breadbin boards.

Note: There is no way to reset the keyboard (or control port) displays to a “nothing pressed” state. Since this is predominantly a cartridge-based solution cycling the power or pressing a reset button is a quick and effective way to achieve this.

Joystick Test

The joystick directions are represented by the small cross-shaped arrangement of characters in the control-port region of the screen. The 4 directions (**Up**, **Right**, **Down** and **Left**) are represented plus the **Fire** button in the middle.



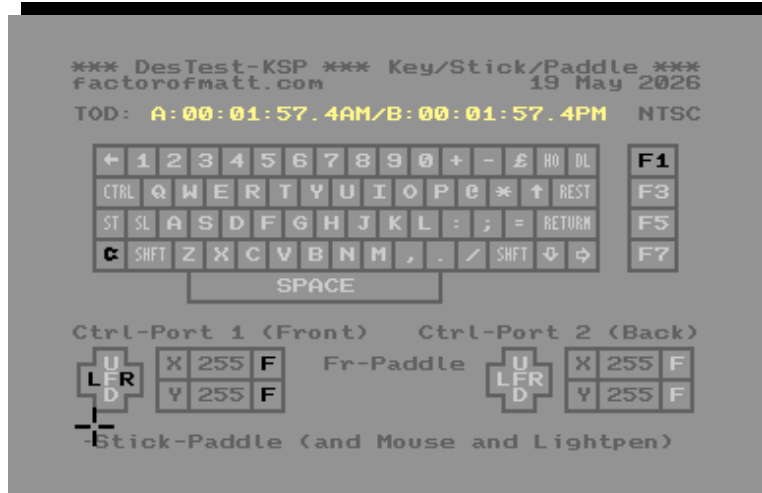
Note: you will notice that the Left and Right joystick buttons also activate the X and Y Fire-button indications for the paddles/mouse immediately to their right.

If you have **Fr-Stick**, **Bk-Stick**, **Lf-Stick** or **Rt-Stick** crosshair mode selected, then activating the directions on your joystick will move the crosshair accordingly. The crosshair will stop at the edges of the display.

Note: This test is also applicable for the 1350 joy-mouse and the 1351 mouse (in joy-mouse mode).

Paddle Test

Paddles come in pairs one is considered to be the X paddle and the other to be the Y paddle. The paddles are attached to the SID which performs the Analogue to Digital conversion. Non-functional paddles may indicate a bad SID.



Each control port shows “position” information for each of the X and Y paddles plus their **Fire** button. Ideally the position number will range between 0 and 255 and will represent the entire spin from lock-to-lock. In practice some paddles won’t reach the entire 0-255 range and/or may have dead-zones at the end of their travel. It is up to you how bad you feel your paddles are.

Note: You may notice that even when entirely still, the position value may jump up and down by one – this is expected and a consequence of how the SID’s AtoD converters work.

The fire buttons for each paddle are indicated to the right of the position information. You will notice that they also trigger the joystick **Left** and **Right** indicators. This is since they use the same lines into the CIA.

If you have **Fr-Paddle**, **Bk-Paddle**, **Lf-Paddle** or **Rt-Paddle** crosshair mode selected, then activating the paddles will move the crosshair accordingly. The X paddle moves the crosshair left and right where the Y paddle moves it up and down. The crosshair will stop at the edges of the display.

Mouse Test

This test is specifically for the 1351 proportional mouse or similar. The older, crappier 1350 mouse must be tested by the joystick test.



As with the paddles, each control port shows “position” information for each of the X and Y directions the mouse can move plus the two mouse buttons (**Fire** for the left button, **Up** for the right button). The position number will vary from 64-ish to 192-ish (*no two of my C64s and C128s get exactly the same numbers*) – though the difference between the highest and lowest number is generally 128. As with the paddles, the number itself will vary by 1 even when the mouse is stationary. You will notice that movements (left, right, up and down) will change the X and Y position values up or down somewhat (and they wrap at the 64ish and 192ish values as above).

If you have **Fr-Mouse**, **Bk-Mouse**, **Lf-Mouse** or **Rt-Mouse** crosshair mode selected, then moving the mouse will move the crosshair accordingly. The crosshair will stop at the edges of the display.

Note: Rapid movement of the mouse will cause the crosshair to act erratically and will jump all over the place. ***This is not an indication of a faulty mouse but is a software error that I haven't yet managed to find the time to fix.***

Light-pen Test

Not many people have a light-pen. This is a shame. The promise of the late '70s and '80s was a lighten in every hand. We need to bring them back. If you are fortunate enough to have a light-pen, then DesTest-KSP has you covered.



To start, there are several kinds of light-pens that operate slightly differently.

- No button. These pens will send position information to the computer whenever they are close enough to the monitor.
- Button to enable. These pens will send position information to the computer when close to the monitor AND the button is pressed.
- Separate Button. These pens will send position information to the computer whenever they are close to the monitor. Additionally, the button will activate one of the traditional joystick buttons (usually **Left** or **Up**).

Note: the **Fire** button would seem like the logical choice to use for the lightpen button, but this signal is actually used by the light-pen. and the VIC-II to signal that position information is available.

For the light-pen. test to operate, the crosshair mode must be set to “Fr-Lt-Pen”. This is quickly achieved by pressing **C= + L**. The crosshair should then follow the position of the light-pen. as it is brought up to the monitor (possibly with the button pressed, depending on the type of light-pen.). The front (left) control port position indicator will display where the light-pen. is being detected. The X position will be half of the actual position due to a limitation of the VIC-II chip not having 9-bits to represent it.

Note: traditional light-pens only work with good, old-fashioned CRT monitors.

Note: The marvellous VICE C64 emulator doesn't do a very good job of emulating light-pen. input. The reported position is good, but the scan-line (raster) at which the detection is made is always 50 when it should match the actual Y coordinate. This messes up the display a little.

Note: There is currently no way to calibrate the position of the crosshair to exactly match that of the light-pen. You will see the crosshair move, and it will track the light-pen. quite well, but it will probably be a little down and to the right. ***This is***

not an indication of a faulty light-pen. but is feature that I haven't yet managed to find the time to add.