ASCII Keyboard Tester
Manual

Test vintage parallel ASCII Keyboards with ease and see how they actually work.

Assembly

Make sure you have all the parts, see the *Bill of Materials*. Always solder by vertical height of the components, from low to high:

- **Resistors** first, check the color codes to match the values (R1 - R16). Use a plastic lead bend tool, or bend the leads by hand, before inserting the resistors into the board.
- **Sockets** second, make sure their orientation is correct (U1 - U3, J3, J4). Check if the notch of the socket matches the outline on the board. Insert the IC's later.
- **Capacitors** next, make sure the cathode (-) side of the capacitor is in the shaded, 'filled' half of the circle on the board (C1, C2). The anode (+) should be in the 'empty' half of the circle.
- Then the **LED**'s (D1 - D12), the long leg is the anode (+) and the short leg the cathode (-). The base of the LED is not a full circle, but has a flat edge. This edge (cathode side) should match the outline on the board.
- **Pin headers** next (J1 - J3, J6). The 4-pin horizontal header is for the power supply (J1), the rest are vertical. Solder one pin first. Then melt the solder of this pin again while moving the header on the other side in the right angle. Finish by soldering the rest of the pins. Watch out to not touch the same pin that is being soldered, it gets hot!
- Insert the **IC**'s (U1 - U3, J3, J4), make sure the notch of the IC (first-pin side) matches the notch-side of the socket (and the outline of the board). It is handy to use an IC straightener tool to bend the legs of the IC correctly before inserting them.
- Finish the tester by mounting the **Standoffs**.

Electronic workings

The core of the tester is the 74LS374 IC, an 8 bit flip-flop [1], at location U1. This flip-flop stores the 8-bit (1 byte) ASCII **data value** coming from the attached keyboard when pressing a key. At key release, the value is still being displayed by the LED's D1-D8 (and stored in this IC) until you press another key. When the tester is first powered on, the displayed value is arbitrary before the first key press.

With the two white LED's D9 and D10 the tester visualizes the **strobe signal** coming from the keyboard. At the moment a key is pressed, the keyboard generates a very short electric pulse signal to indicate a key is being pressed.

This signal can be low-*going* (default high) or high-*going* (default low), as indicated by the sequence in which the LEDs turn on. For example, when the strobe signal is default low and gets high at a keypress, then LED High (D10) is lit first. At the time of this signal change, the data bits are set on the data lines by the keyboard. In order to 'see' this signal, its duration is extended by the 74LS123 IC, a retriggerable monostable multivibrator [2], at location U3. To be precise, a negative/positive edge-triggered one-shot is used [3].

The keyboard's **clear screen** and **reset** signals are fed through a 74LS00 NAND-gate [4], at location U2. The outputs of this IC are the LED's D11 and D12, which visualize both signals respectively. The clear screen signal is a positive going signal, while the reset signal is a negative going signal, *see* Figure 1.

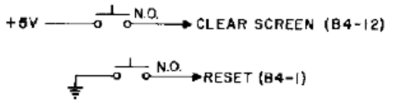


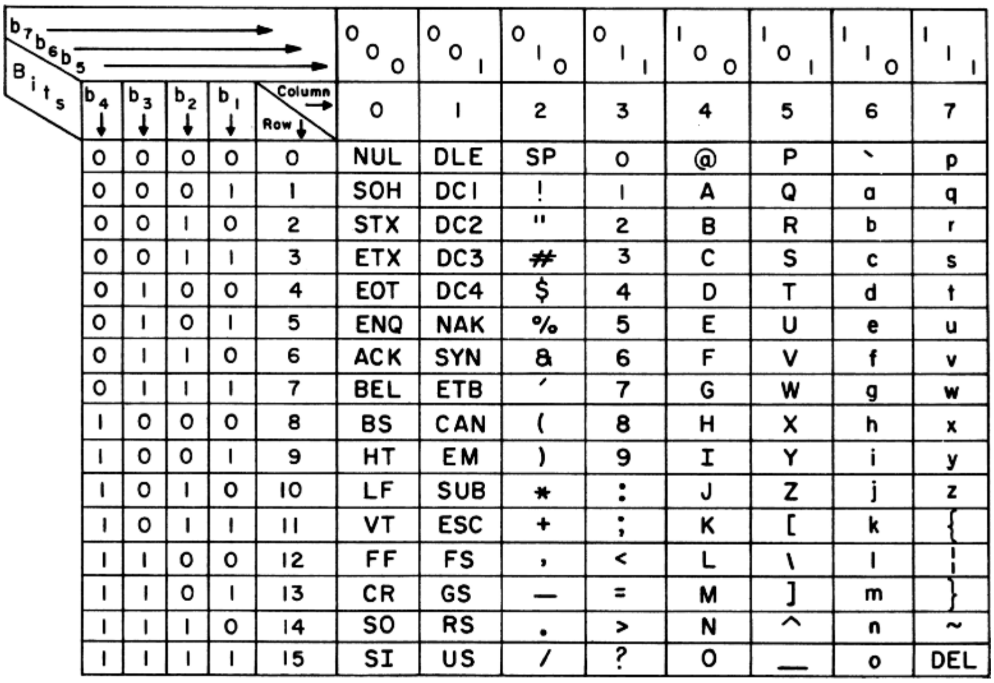
Figure 1: Apple-1 Clear Screen and Reset signals [5]

v1.1

Usage

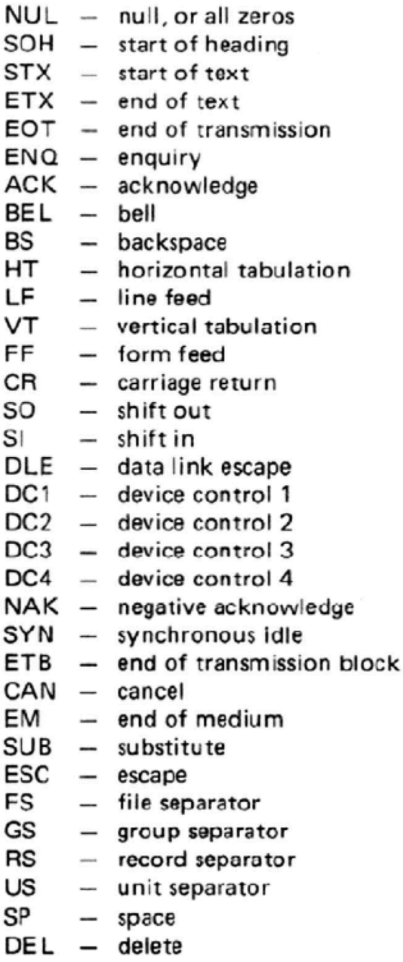
This keyboard tester can be used to test **parallel ASCII** keyboards. For example, the keyboards used with the Apple-1 and Apple][. These keyboards send ASCII data to the computer in parallel form (i.e. all bits through a separate cable at the same time) and were used back in the 70s and 80s. Nowadays keyboards send data in *serial* form, bit after bit through a single cable.

When a key is pressed, the circuitry of the keyboard generates an **ASCII** code and sends this to the computer. The original ASCII standard encodes 128 characters into a **7-bit code**, *see* Figure 2. Most of the characters can be displayed or printed. Some of them are *control characters*, like ‘BS’ being backspace and ‘CR’ being carriage return, *see* Figure 3.



Bits b7 b6 b5 →					0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
b4 ↓	b3 ↓	b2 ↓	b1 ↓	Column → Row ↓	0	1	2	3	4	5	6	7
0	0	0	0	0	NUL	DLE	SP	0	@	P	`	p
0	0	0	1	1	SOH	DC1	!	1	A	Q	a	q
0	0	1	0	2	STX	DC2	"	2	B	R	b	r
0	0	1	1	3	ETX	DC3	#	3	C	S	c	s
0	1	0	0	4	EOT	DC4	$	4	D	T	d	t
0	1	0	1	5	ENQ	NAK	%	5	E	U	e	u
0	1	1	0	6	ACK	SYN	&	6	F	V	f	v
0	1	1	1	7	BEL	ETB	'	7	G	W	g	w
1	0	0	0	8	BS	CAN	(	8	H	X	h	x
1	0	0	1	9	HT	EM	)	9	I	Y	i	y
1	0	1	0	10	LF	SUB	*	:	J	Z	j	z
1	0	1	1	11	VT	ESC	+	;	K	[	k	{
1	1	0	0	12	FF	FS	,	<	L	\	l	¦
1	1	0	1	13	CR	GS	—	=	M	]	m	}
1	1	1	0	14	SO	RS	.	>	N	^	n	~
1	1	1	1	15	SI	US	/	?	O	_	o	DEL

Figure 2: The original 7-bit ASCII character encoding from 1967[6]



NUL	null, or all zeros
SOH	start of heading
STX	start of text
ETX	end of text
EOT	end of transmission
ENQ	enquiry
ACK	acknowledge
BEL	bell
BS	backspace
HT	horizontal tabulation
LF	line feed
VT	vertical tabulation
FF	form feed
CR	carriage return
SO	shift out
SI	shift in
DLE	data link escape
DC1	device control 1
DC2	device control 2
DC3	device control 3
DC4	device control 4
NAK	negative acknowledge
SYN	synchronous idle
ETB	end of transmission block
CAN	cancel
EM	end of medium
SUB	substitute
ESC	escape
FS	file separator
GS	group separator
RS	record separator
US	unit separator
SP	space
DEL	delete

Figure 3: Explanation of the control characters

The **8th bit** is used by some keyboards as well. It can be a *parity bit*, an error detecting code for the computer to see if the transmitted character code is correct. Or it can be part of the encoded character, as in *extended* ASCII[7] encodings, which provide more (or even different) characters than shown in the above figure.

The red LEDs of this keyboard tester show the full byte (8 bits) of data coming from the keyboard. The default value of bit 8 is zero (the LED is off), for the case it is not connected and a 7-bit input is used.

Power supply

The keyboard tester needs a (custom built) power supply and should be connected to pin header J1 (or J2, these can be used interchangeably). The board itself works with just **+5V** being connected. However, most vintage keyboards need **-12V** as well, and sometimes **+12V**. For example, a suitable power supply to use is the Mean Well PT-65B[8], which provides all three different voltages.

On the next page, *see* Table 1. It gives an overview of the voltages *supplied* by the board’s connectors and *needed* by several keyboards. This list is not exhaustive, see the specific keyboard and/or computer documentation to discover which voltages are needed.

Connector and keyboards	+5V	-12V	+12V
Apple-1 / J4	✓	✓	✓
Newton Computer’s Datanetics Rev B Keyboard replica	✓		
Original Datanetics Rev B using MM5740/AAE encoder [9]	✓	✓	
Apple][/ ITT 2020 / J5	✓	✓	
Early Apple][keyboards with MM5740/AEE encoder	✓	✓	
Later Apple][keyboards with separate plug-in boards using a SMC KR3600 encoder	✓	✓	
Power out / J2	✓	✓	✓
General Instrument AY-5-2376 [10] based keyboards, like SWTPC’s KBD-5 [11] and Elektor’s ASCII keyboard [12]	✓	✓	
National MM5740/AAx based	✓	✓	

Table 1: Overview of supplied (in bold) and needed voltages

Connecting a keyboard

There are three methods to connect a keyboard (use only one method at the same time):

1. use the DIP-16 socket for an **Apple-1** keyboard at J4,
2. the DIP-16 socket for an **Apple][** keyboard at J5
3. or use the pin headers J2, J3 and J6 to connect **uncommon** or **custom** keyboards.

For methods **i** and **ii**, the two DIP sockets provide power to the connected keyboard, in the same way the computer (Apple-1 or Apple][, respectively) does. *See* Figures 4 and 5 for the pinout of these two connectors [5, 13]. Important: these figures are upside down, i.e. the first pin is bottom right on the board (marked with a white dot) while it is top left in the figure.

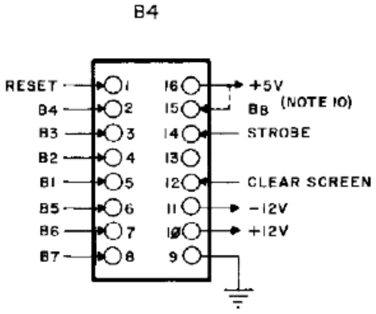


Figure 4: Pinout of the keyboard connector for the Apple-1 (J4) [5]

The Bx pins are the data lines, where the ASCII code gets transmitted. In the documentation out there [13], the first data pin is sometimes referred to as B0 instead of B1. The Apple-1 and Apple][accept capital letters only, so it is normal when the lowercase ASCII codes do not show up. Read more about the strobe signal in the chapter ‘Electronic workings’ on the first page.

Some differences; the Apple-1 makes use of the clear screen signal, while both use the reset signal, *see* Figure 1 on the first page. And the Apple-1 only supplies +12V.

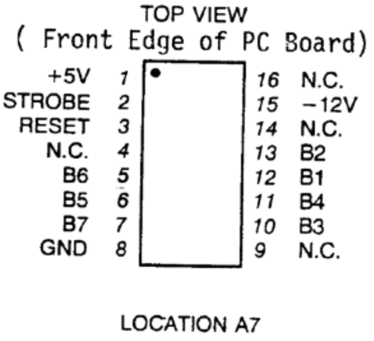


Figure 5: Pinout of the keyboard connector for the Apple][(J5) [13]

For example, if you have a self-made keyboard, you can test if its DIP connector is correctly wired for an Apple-1 or Apple][, using method i or ii respectively.

For method **iii**, data and strobe input from the keyboard is connected at pin header J3, power is supplied to the keyboard through the pins at J2. Optionally, the clear and reset signals (matching Apple-1 and Apple][behavior, *see* Figure 1) can be provided at J6.

Make sure all needed voltages are supplied by the power supply at J1, *see* the previous chapter, otherwise the keyboard will not work. On the previous page, *see* Table 1 for some examples of keyboards that can be connected. The forum Deskthority has a posting of MMcM where a couple of more examples of parallel keyboards are listed [14], use it as a source of inspiration.

Breadboard area

Just as the Apple-1 and the first Apple]['s this board has a breadboard / prototype area. Custom circuits can be added here. For example: signal convertors, lowercase to uppercase ASCII translators, a 'stepper' to probe an 8 bit address bus or an 8 bit counter. It even has room for a complete Hex display [15].

On the sides of the breadboard area there is a voltage rail and there are a couple of solder-holes to pick-up necessary signals. The IC at location U2 even has a free NAND gate to use. All clearly marked on the board.

Hex Display Adapter

This tester can be extended with a two-digit Hex display adapter board. This adapter board fits on top of this board in socket U1. It displays the Hex value that corresponds to the binary value displayed by the red LEDs. See the website for more information.

DISCLAIMER

The use of our products and schematics is at your own risk. **We expect our audience and buyers to have experience with electronics and electricity.** The product images shown on the website are for illustration purposes only and may not be an exact representation of the actual product, which is subject to enhancement. Colors (of parts) can differ because of changes in supply.

In no event shall 8BitFlux.com be liable in warranty, contract, negligence, strict liability, tort or otherwise, for any damages, whether incidental or consequential, which are alleged to be caused by one or more of our products, beyond the cost to the purchaser of the subject product or products in question.

REFERENCE LIST

1. Texas Instruments (1975, October). *OCTAL D-TYPE TRANSPARENT LATCHES AND EDGE-TRIGGERED FLIP-FLOPS*. Retrieved December 27, 2023, from https://www.ti.com/lit/gpn/SN54LS373-SP
2. Texas Instruments (1983, December). *RETRIGGERABLE MONOSTABLE MULTIVIBRATORS*. Retrieved December 27, 2023, from https://www.ti.com/lit/gpn/SN54LS123-SP
3. Texas Instruments (1997, March). *Designing With the SN54/74LS123*, p. 13. Retrieved December 27, 2023, from https://www.ti.com/lit/pdf/sdla006
4. Texas Instruments (2017, May). *Quadruple 2-Input Positive-NAND Gates.* Retrieved January 8, 2024, from https://www.ti.com/lit/gpn/SN54LS00-SP
5. Apple Computer Company (1976). *Apple-1 Operation Manual*, *Schematic Diagram, Power Supply.*
6. Wikipedia (2024, January). *ASCII*. Retrieved January 8, 2024, from https://en.wikipedia.org/wiki/ASCII
7. Wikipedia (2024, January). *Extended ASCII*. Retrieved January 8, 2024, from https://en.wikipedia.org/wiki/Extended_ASCII
8. Mean Well (2024). *PT-65B AC-DC Triple output Open frame power supply*. Retrieved January 9, 2024, from https://www.meanwell-web.com/nl-nl/ac-dc-triple-output-open-frame-power-supply-output-pt--65b
9. M. Willegal. *Datanetics Rev B Keyboard*. Retrieved January 9, 2024, from http://www.willegal.net/appleii/apple1-datanetics.htm
10. General Instrument. *AY-5-2376*. Retrieved January 9, 2024, from https://telcontar.net/KBK/General_Instrument/docs/General%20Instrument%20AY-5-2376%20specifications.pdf
11. SWTPC. *KBD-5 Keyboard*. Retrieved January 9, 2024, from https://deramp.com/swtpc.com/CT_1024/KBD5_Assembly.pdf
12. Elektor (1978, November). *ASCII Keyboard*. Retrieved January 9, 2024, from https://www.applefritter.com/appleii-box/APPLE1/Keyboard/ASCIIkeyboard.pdf
13. Apple Computer Inc. (1979). A*pple II Reference Manual*, p. 103.
14. MMcM (2021, February). *Some parallel ASCII keyboards*. Retrieved January 10, 2024, from https://deskthority.net/viewtopic.php?t=25289
15. Macnoyd (2024, January). Retrieved January 8, 2024 from https://www.applefritter.com/comment/105994#comment-105994