Hardware

COLLABORATORS

	TITLE : Hardware	

ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		July 23, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

Hardware

1.1 Amiga® Hardware Reference Manual: G Keyboard Interface

This appendix contains the keyboard interface specification for A1000, A500, A2000 and A3000.

The keyboard plugs into the Amiga computer via a cable with four primary connections . The four wires provide 5-volt power, ground, and signals called KCLK (keyboard clock) and KDAT (keyboard data). KCLK is unidirectional and always driven by the keyboard; KDAT is driven by both the keyboard and the computer. Both signals are open-collector; there are pullup resistors in both the keyboard (inside the keyboard microprocessor) and the computer.

@{ " Keyboard Communications " link G-1}
@{ " Keycodes " link G-2}
@{ " Caps Lock Key " link G-3}
@{ " Out-of-Sync Condition " link G-4}
@{ " Power-Up Sequence " link G-5}
@{ " Reset Warning " link G-6}
@{ " Hard Reset " link G-7}
@{ " Matrix Table " link G-8}
@{ " Special Codes " link G-9}

1.2 G Keyboard Interface / Keyboard Communications

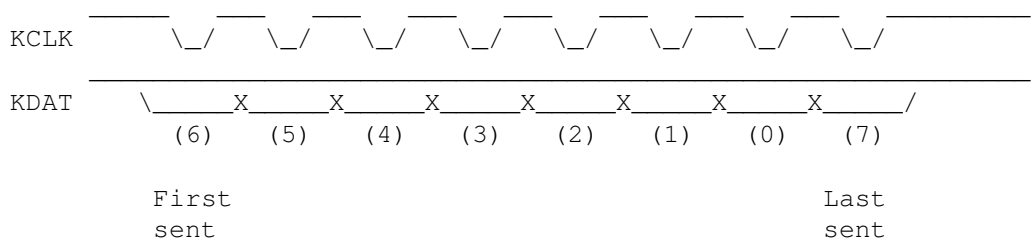
The keyboard transmits 8-bit data words serially to the main unit. Before the transmission starts, both KCLK and KDAT are high. The keyboard starts the transmission by putting out the first data bit (on KDAT), followed by a pulse on KCLK (low then high); then it puts out the second data bit and pulses KCLK until all eight data bits have been sent. After the end of the last KCLK pulse, the keyboard pulls KDAT high again.

When the computer has received the eighth bit, it must pulse KDAT low for at least 1 (one) microsecond, as a handshake signal to the keyboard. The handshake detection on the keyboard end will typically use a hardware latch. The keyboard must be able to detect pulses greater than or equal to 1 microsecond. Software MUST pulse the line low for 85 microseconds to ensure compatibility with all keyboard models.

All codes transmitted to the computer are rotated one bit before transmission. The transmitted order is therefore 6-5-4-3-2-1-0-7. The

reason for this is to transmit the up/down flag last, in order to cause a key-up code to be transmitted in case the keyboard is forced to restore lost sync (explained in more detail below).

The KDAT line is active low; that is, a high level (+5V) is interpreted as 0, and a low level (0V) is interpreted as 1.



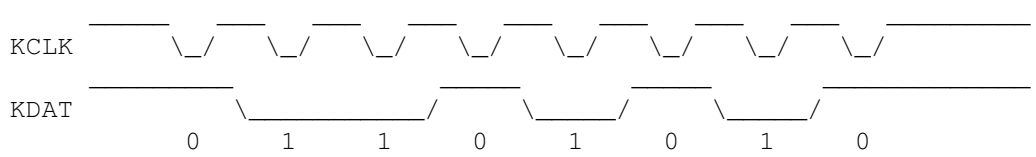
The keyboard processor sets the KDAT line about 20 microseconds before it pulls KCLK low. KCLK stays low for about 20 microseconds, then goes high again. The processor waits another 20 microseconds before changing KDAT.

Therefore, the bit rate during transmission is about 60 microseconds per bit, or 17 kbits/sec.

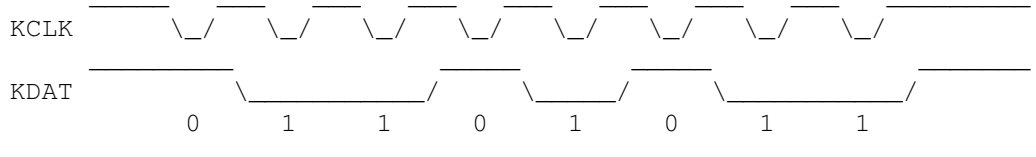
1.3 G Keyboard Interface / Keycodes

Each key has a keycode associated with it (see accompanying table). Keycodes are always 7 bits long. The eighth bit is a "key-up"/"key-down" flag; a 0 (high level) means that the key was pushed down, and a 1 (low level) means the key was released (the Caps Lock key is different -- see below).

For example, here is a diagram of the "B" key being pushed down. The keycode for "B" is $35 = 00110101; due to the rotation of the byte, the bits transmitted are 01101010.



In the next example, the B key is released. The keycode is still $35, except that bit 7 is set to indicate "key-up," resulting in a code of $B5 = 10110101. After rotating, the transmission will be 01101011:



1.4 G Keyboard Interface / Caps Lock Key

This key is different from all the others in that it generates a keycode only when it is pushed down, never when it is released. However, the up/down bit is still used. When pushing the Caps Lock key turns on the Caps Lock LED, the up/down bit will be 0; when pushing Caps Lock shuts off the LED, the up/down bit will be 1.

1.5 G Keyboard Interface / Out-of-Sync Condition

Noise or other glitches may cause the keyboard to get out of sync with the computer. This means that the keyboard is finished transmitting a code, but the computer is somewhere in the middle of receiving it.

If this happens, the keyboard will not receive its handshake pulse at the end of its transmission. If the handshake pulse does not arrive within 143 ms of the last clock of the transmission, the keyboard will assume that the computer is still waiting for the rest of the transmission and is therefore out of sync. The keyboard will then attempt to restore sync by going into "resync mode." In this mode, the keyboard clocks out a 1 and waits for a handshake pulse. If none arrives within 143 ms, it clocks out another 1 and waits again. This process will continue until a handshake pulse arrives.

Once sync is restored, the keyboard will have clocked a garbage character into the computer. That is why the key-up/key-down flag is always transmitted last. Since the keyboard clocks out 1's to restore sync, the garbage character thus transmitted will appear as a key release, which is less dangerous than a key hit.

Whenever the keyboard detects that it has lost sync, it will assume that the computer failed to receive the keycode that it had been trying to transmit. Since the computer is unable to detect lost sync, it is the keyboard's responsibility to inform the computer of the disaster. It does this by transmitting a "lost sync" code (value $F9 = 11111001) to the computer. Then it retransmits the code that had been garbled.

> About Lost Sync.
> ----------
> The only reason to transmit the "lost sync" code to the computer is to alert the software that something may be screwed up. The "lost sync" code does not help the recovery process, because the garbage key code can't be deleted, and the correct key code could simply be retransmitted without telling the computer that there was an error in the previous one.

1.6 G Keyboard Interface / Power-Up Sequence

There are two possible ways for the keyboard to be powered up under normal circumstances: <1> the computer can be turned on with the keyboard plugged in, or <2> the keyboard can be plugged into an already "on" computer. The keyboard and computer must handle either case without causing any upset.

The first thing the keyboard does on power-up is to perform a self-test. This involves a ROM checksum test, simple RAM test, and watchdog timer test. Whenever the keyboard is powered up (or restarted -- see below), it must not transmit anything until it has achieved synchronization with the computer. The way it does this is by slowly clocking out 1 bits , as described above, until it receives a handshake pulse.

If the keyboard is plugged in before power-up, the keyboard may continue this process for several minutes as the computer struggles to boot up and get running. The keyboard must continue clocking out 1s for however long is necessary, until it receives its handshake.

If the keyboard is plugged in after power-up, no more than eight clocks will be needed to achieve sync. In this case, however, the computer may be in any state imaginable but must not be adversely affected by the garbage character it will receive. Again, because it receives a key release, the damage should be minimal. The keyboard driver must anticipate this happening and handle it, as should any application that uses raw keycodes.

Warning:

The keyboard must not transmit a " lost sync " code after re-synchronizing due to a power-up or restart ; only after re-synchronizing due to a handshake time-out.

Once the keyboard and computer are in sync, the keyboard must inform the computer of the results of the self-test. If the self-test failed for any reason, a "selftest failed" code (value $FC = 11111100) is transmitted (the keyboard does not wait for a handshake pulse after sending the "selftest failed" code). After this, the keyboard processor goes into a loop in which it blinks the Caps Lock LED to inform the user of the failure. The blinks are coded as bursts of one, two, three, or four blinks, approximately one burst per second:

One blink	ROM checksum failure.
Two blinks	RAM test failed.
Three blinks	Watchdog timer test failed.
Four blinks	A short exists between two row lines or one of the seven special keys (not implemented).

If the self-test succeeds, then the keyboard will proceed to transmit any keys that are currently down. First, it sends an "initiate power-up key stream" code (value $FD = 11111101), followed by the key codes of all depressed keys (with keyup/down set to "down" for each key). After all keys are sent (usually there won't be any at all), a "terminate key stream" code (value $FE = 11111110) is sent. Finally, the Caps Lock LED is shut off. This marks the end of the start-up sequence, and normal processing commences.

The usual sequence of events will therefore be: power-up; synchronize; transmit "initiate power-up key stream" ($FD); transmit "terminate key stream" ($FE).

1.7 G Keyboard Interface / Reset Warning

```
   About Reset Warning.
   --------------------
   This feature is available on some A1000 and A2000 keyboards. You
   cannot rely on this feature for all Amigas.
```

The keyboard has the additional task of resetting the computer on the command of the user. The user initiates Reset Warning by simultaneously pressing the Ctrl key and the two Amiga keys.

The keyboard responds to this input by syncing up any pending transmit operations. The keyboard then sends a "reset warning" to the Amiga. This action alerts the Amiga software to finish up any pending operations (such as disk DMA) and prepare for reset.

A specific sequence of operations ensure that the Amiga is in a state where it can respond to the reset warning. The keyboard sends two actual "reset warning" keycodes. The Amiga must handshake to the first code like any normal keystroke, else the keyboard goes directly to Hard Reset . On the second "reset warning" code the Amiga must drive KDAT low within 250 milliseconds, else the keyboard goes directly to Hard Reset . If all the tests are passed, the Amiga has 10 full seconds to do emergency processing. When the Amiga pulls KDAT high again, the keyboard finally asserts hard reset .

If the Amiga fails to pull KDAT high within 10 seconds, Hard Reset is asserted anyway.

1.8 G Keyboard Interface / Hard Reset

```
   About Hard Reset.
   -----------------
   Hard Reset happens after  Reset Warning . Valid for all keyboards
   except the Amiga 500.
```

The keyboard Hard Resets the Amiga by pulling KCLK low and starting a 500 millisecond timer. When one or more of the keys is released and 500 milliseconds have passed, the keyboard will release KCLK. 500 milliseconds is the minimum time KCLK must be held low. The maximum KCLK time depends on how long the user holds the three reset keys down. Circuitry on the Amiga motherboard detects the 500 millisecond KCLK pulse.

After releasing KCLK, the keyboard jumps to its start-up code (internal RESET). This will initialize the keyboard in the same way as cold power-on .

```
   NOTE:
   -----
   The keyboard must resend the " powerup key stream "!
```

1.9 G Keyboard Interface / Matrix Table

Column	Row 5 (Bit 7)	Row 4 (Bit 6)	Row 3 (Bit 5)	Row 2 (Bit 4)	Row 1 (Bit 3)	Row 0 (Bit 2)
15 (PD.7)	(spare)	(spare)	(spare)	(spare)	(spare)	(spare)
	(0E)	(1C)	(2C)	(47)	(48)	(49)
14 (PD.6)	* note 1	<SHIFT> note 2	CAPS LOCK	TAB	~ `	ESC
	(5D)	(30)	(62)	(42)	(00)	(45)
13 (PD.5)	+ note 1	Z	A	Q	! 1	(note 1
	(5E)	(31)	(20)	(10)	(01)	(5A)
12 (PD.4)	9 note 3	X	S	W	@ 2	F1
	(3F)	(32)	(21)	(11)	(02)	(50)
11 (PD.3)	6 note 3	C	D	E	# 3	F2
	(2F)	(33)	(22)	(12)	(03)	(51)
10 (PD.2)	3 note 3	V	F	R	$ 4	F3
	(1F)	(34)	(23)	(13)	(04)	(52)
9 (PD.1)	. note 3	B	G	T	% 5	F4
	(3C)	(35)	(24)	(14)	(05)	(53)
8 (PD.0)	8 note 3	N	H	Y	^ 6	F5
	(3E)	(36)	(25)	(15)	(06)	(54)
7 (PC.7)	5 note 3	M	J	U	& 7	) note 1
	(2E)	(37)	(26)	(16)	(07)	(5B)
6 (PC.6)	2 note 3	< ,	K	I	* 8	F6
	(1E)	(38)	(27)	(17)	(08)	(55)
5 (PC.5)	ENTER note 3	> .	L	O	(9	/ note 1
	(43)	(39)	(28)	(18)	(09)	(5C)
4 (PC.4)	7 note 3	? /	: ;	P	) 0	F7
	(3D)	(3A)	(29)	(19)	(0A)	(56)
3 (PC.3)	4 note 3	(spare)	" ’	{ [	_ -	F8
	(2D)	(3B)	(2A)	(1A)	(0B)	(57)

2 (PC.2)	1 note 3 (1D)	SPACE BAR (40)	<RET> note 2 (2B)	}] (1B)	+ = (0C)	F9 (58)
1 (PC.1)	0 note 3 (0F)	BACK SPACE (41)	DEL (46)	RETURN (44)	\ (0D)	F10 (59)
0 (PC.0)	- note 3 (4A)	CURS DOWN (4D)	CURS RIGHT (4E)	CURS LEFT (4F)	CURS UP (4C)	HELP (5F)

note 1: A500, A2000 and A3000 keyboards only (numeric pad)
note 2: International keyboards only (these keys are cutouts of the larger key on the US ASCII version.) The key that generates $30 is cut out of the left Shift key. Key $2B is cut out of return. These keys are labeled with country-specific markings.
note 3: Numeric pad.

The following table shows which keys are independently readable. These keys never generate ghosts or phantoms.

(Bit 6)	(Bit 5)	(Bit 4)	(Bit 3)	(Bit 2)	(Bit 1)	(Bit 0)
LEFT AMIGA (66)	LEFT ALT (64)	LEFT SHIFT (60)	CTRL (63)	RIGHT AMIGA (67)	RIGHT ALT (65)	RIGHT SHIFT (61)

1.10 G Keyboard Interface / Special Codes

The special codes that the keyboard uses to communicate with the main unit are summarized here.

About the special codes.

The special codes are 8-bit numbers; there is no up/down flag associated with them. However, the transmission bit order is the same as previously described.

Code	Name	Meaning
78	Reset warning .	Ctrl-Amiga-Amiga has been hit - computer will be reset in 10 seconds. (see text)
F9	Last key code bad, next code is the same code	retransmitted (used when keyboard and main unit get out of sync).
FA	Keyboard output buffer overflow	
FB	Unused (was controller failure)	
FC	Keyboard selftest failed	

```
FD      Initiate  power-up key stream  (keys pressed at powerup)
FE      Terminate  power-up key stream
FF      Unused (was interrupt)
```