SECTION 4

ACCESS.bus

Keyboard Device Protocol Specification

TABLE OF CONTENTS

4.0 Introduction

4.0.1 Design Objectives

The keyboard device protocol described in this specification defines standard messages for reporting keystrokes and controlling keyboard peripherals. The keyboard device protocol attempts to define the simplest set of functions from which common industry standard keyboard interfaces can be built. The following principles were used to guide the design:

1. Provide sufficient functional completeness to support existing user interfaces.
2. Reduce complexity in human terms wherever possible. Existing keyboard drivers do not utilize many of the features provided. Some have introduced errors in handling the subtle interactions defined.
3. Minimize state information that must be modeled in both the keyboard and the host. This is to avoid synchronization problems.
4. Minimize memory required in the keyboard. Cost per bit is much lower on the host.

4.0.2 Generic Keyboard Overview

A generic keyboard consists of an array of key stations assigned numbers between 8 and 255. When any key station transitions between open and closed, the entire list of key stations currently closed or depressed is transmitted to the host.

In addition to reporting key stations, the generic keyboard device can support simple feedback mechanisms such as keyclicks, bells, and light-emitting diodes. These mechanisms are controlled explicitly from the host so that minimal keyboard state modeling is required. The capabilities information is used to identify the keyboard mapping table and the feedback mechanisms available. The keyboard mapping table can also be stored in the keyboard itself as part of the capabilities string.

4.1 Key Event Reporting

Each key is assigned a unique 8-bit number (8-255). The first eight (8) codes are reserved for other keyboard functions. On each key transition, up or down, the keyboard will report the complete state of the key array as a list of zero to ten key stations that are currently down.

Example: user enters the modified keystroke Alt-Shift-A

transition		report
Alt	down	Alt
Shift	down	Alt Shift
'A'	down	Alt Shift A
'A'	up	Alt Shift
Shift	up	Alt
Alt	up	<empty list>

This reporting scheme is functionally complete in that the host can detect every key transition and it provides the full state of the keyboard on each report. No special resynchronization reports are needed.

To simplify generating the reports, keys can be reported in any order.

4.2 Auto Repeat

Auto-repeat is the responsibility of the host. The common model is to have appropriate keys begin auto-repeating at a constant <rate> if held down for longer than some start up <delay>. The auto-repeat <rate> and <delay> are often user setable.

4.3 Keyclick and Bell

Keyclick is handled manually by the host sending the command to click after each appropriate key transition or auto repeated keystroke. In order for keyclick to appear instantaneous, keyclick should occur within 100 milliseconds of the corresponding key transition. The keyclick volume is specified by the host on each command so there is no state to set or remember in the keyboard.

Bell is handled manually by the host sending the command to sound the bell as needed. The bell volume is specified by the host on each command so there is no state to set or remember in the keyboard.

4.4 Capabilities Information

The keywords defined in this section have standard meanings within the ACCESS.bus Generic Keyboard Device Protocol.

Keyword	Meaning
keymap()	The "keymap()" entry identifies the keymap table that should be used to translate keycodes to characters or keystroke functions.
feedback()	The "feedback()" entry lists and describes the feedback mechanisms available for use by host software.
click()	The "click()" entry indicates the keyboard provides a host controlled click feature. The parenthesis contain an integer representing the maximum volume setting. Zero is assumed to be the minimum.
bell()	The "bell()" entry indicates the keyboard provides a host controlled bell feature. The parenthesis contain an integer representing the maximum volume setting. Zero is assumed to be minimum.
led()	The "led()" entry indicates the keyboard provides one or more host controlled LED

indicators. The parenthesis contain a list of tagged labels giving the bit number in the illumination mask and corresponding indicator label or name. The following labels are currently defined:

hold, lock, compose, wait, NumLock, CapsLock, ScrollLock

4.5 Timing and Exceptions (Guideline)

The keyboard microprocessor scans the key array repeatedly to detect key transitions. If more than one key transition is detected during a single scan, all key transitions will be reported together as part of the new keyboard state. In this case, the order of key transitions cannot be determined.

When the microprocessor completes a scan of the key array and has detected one or more key transitions, it will try to assume bus mastership to send a keyboard report. Since other devices may be using the bus, it could take some time before the keyboard is allowed to become bus master. The processor may restart scanning the key array during this time, and wait to be interrupted when bus mastership has been granted.

It is assumed devices attached to the desktop bus will have an opportunity to report during each frame refresh interval. This is necessary to track locator devices smoothly. For 60 Hz video, this interval is 16.6 milliseconds.

4.6 Host Commands to Keyboard

4.6.1 Application Click (Optional)

The Application Click command instructs the keyboard to generate a click sound (device defined Control/Status, P=l).

Op-code: 01 Data: l-byte click volume.

4.6.2 Application Bell (Optional)

The Application Bell command instructs the keyboard to generate a bell sound (device defined Control Status, P=l).

Op-code: 02 Data: l-byte bell volume.

4.6.3 Application LEDs (Optional)

The Application LEDs command instructs the keyboard to illuminate one or more LED indicators (device defined Control/Status, P=1).

Format:

Op-code:	03
Data:	1-byte illumination mask: 0=off, 1=on

4.6.4 Application Poll

The Application Poll command instructs the keyboard to report its current state showing which keys are currently down (device defined Control/Status, P=1). See Keyboard State Report.

Op-code:	04
Data:	none

4.7 Keyboard To Host Data

4.7.1 Keyboard State Report (Device Data Stream)

The keyboard State Report transmits a list of up to ten keycode (8-255) for the keys that are currently down. Code value zero (0) means the key list is empty, no keys are down.

4.7.2 Keyboard Output Error

The Keyboard Output Error message indicates that the keyboard has detected a key state it cannot report. This might occur because more than ten keys are being held down simultaneously or a possible phantom key has been detected. The keyboard will transmit a valid key state report as soon as the condition preventing the key state from being sent is corrected. Keyboard Output Error is a device defined Control/Status message.

4.7.3 Keyboard Self Test (Attention) Report

At power up, or upon receiving a "Reset" command, the keyboard will test its electronics and firmware and report the results as an "Attention" report (see ACCESS.bus Description and Protocol Specification). The keyboard uses the following predefined status values in its attention report:

0	Success
1	ROM checksum error detected
2	RAM error detected
3	Key down error detected

If a ROM checksum error is detected, the second data byte will give the computed non-zero ROM checksum.

If a key down error is detected, the second data byte will give the keycode of the first down key detected (8-255).

These same parameters are used for Application Reply in response to an Applicaltion Test command. In this case, the previously stored power-up test results are used.

The keyboard shall ignore incoming messages during self test. Messages addressed to the device will not be acknowledged.

Start up guidelines: If a ROM or RAM error is detected, the keyboard may attempt to start normal operation anyway. If a key down error is detected, the keyboard will continue scanning the key array until all keys are up, and then transmit a test report with status code zero indicating success.

4.8 Keyboard Mapping Tables

The keyboard mapping tables in this section are not mandatory, but recommended for compatibility with industry standard keyboard layouts. Additional layouts can be inc luded in this specification by approval of the ACCESS.bus Standard Committee.

FIGURE 4.1: EPC Numbered Assignment of Key Positions

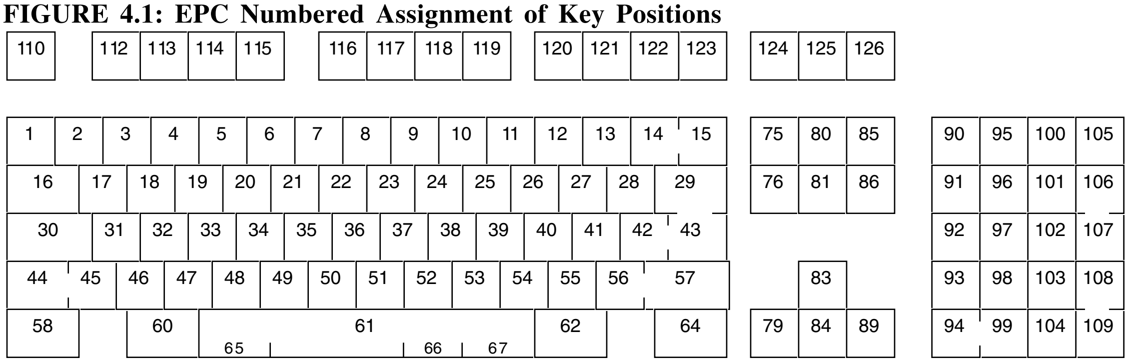

FIGURE 4.2: EPC Typical Domestic US Layout

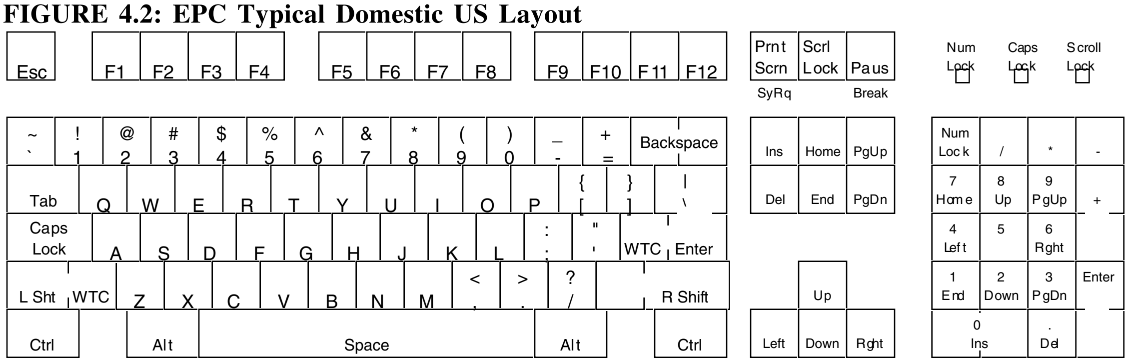

Table 4.1: EPC Suggested Keycodes

Key #	Legend	Key Code	Position
1	`~	BF	E00
2	1	CO	E01
3	2	C5	E02
4	3	CB	E03
5	4	D0	E04
6	5	D6	E05
7	6	DB	E06
8	7	E0	E07
9	8	E5	E08
10	9	EA	E09
11	0	EF	E10
12	- _	F9	E11
13	= +	F5	E12
14		F4	E13
17	Q	C1	D01

18	W	C6	D02
19	E	CC	D03
20	R	D 1	D04
21	T	D7	D05
22	Y	DC	D06
23	U	E 1	D07
24	1	E6	D08
25	0	EB	D09
26	P	F0	D10
27	[~	FA	D11
28	] }	F6	D12
29	\I	F1	D13
31	A	C2	C01
32	S	C7	C02
33	D	CD	C03
34	F	D2	C04
35	G	D8	C05
36	H	D D	C06
37	J	E2	C07
38	K	E7	C08
39	L	EC	C09
40	;:	F2	C10
41	' "	FB	C11
42		F7	C12
45		C9	B00
46	Z	C3	B01
47	X	C8	B02
48	C	CE	B03
49	V	D3	B04
50	B	D9	B05
51	N	DE	B06
52	M	E3	B07
53	, ~	E8	B08
54	. >	ED	B09
55	/ ?	F3	B10
56		FC	B11
15	Backspace	BC	
16	Tab	BE	
58	Left Ctrl	AF	
64	Right Ctrl	62	
30	Lock	B0	
43	Return	BD	
44	Left Shift	AE	
57	Right Shift	AB	
60	Left Alt	AC	
61	Space	D4	
62	Right Alt (Alt Gr)	B2	
65		5E	
66		5F	
67		60	
75	Insert	8B	
76	Delete	8C	
80	Home	8A	
81	End	8D	
85	Pg Up	8E	

86	Pg Dn	8F
83	Up Arrow	A A
84	Down Arrow	A9
79	Left Arrow	A7
89	Right Arrow	A8
90	Num Lock	A1
95	1	A2
100	~	A3
105	-	A4
91	Home 7	9D
96	Up 8	9E
101	PgUp 9	9F
106	+	A0
92	Left 4	99
97	5	9A
102	Right 6	9B
107		9C
93	End 1	96
94		91
98	Down 2	97
103	PgDn 3	98
108	Enter	95
109		93
99	Ins 0	92
104	Del .	94
110	Esc	7D
112	F1	56
113	F2	57
114	F3	58
115	F4	59
116	F5	5A
117	F6	64
118	F7	65
119	F8	66
120	F9	67
121	F10	68
122	F11	71
123	F12	72
124	Print Screen	73
125	Scroll Lock	74
126	Pause	7C

DEC convention uses Lock as a local Extend key so that Lock/Fl through Lock/F10 are synonyms for Fll through F20 on Enhanced PC keyboards that do not have 20 top row function key

4.8.1 DEC LK501 LAYOUT

Figure 4.3: LK501 Numbered Assignment of Key Positions

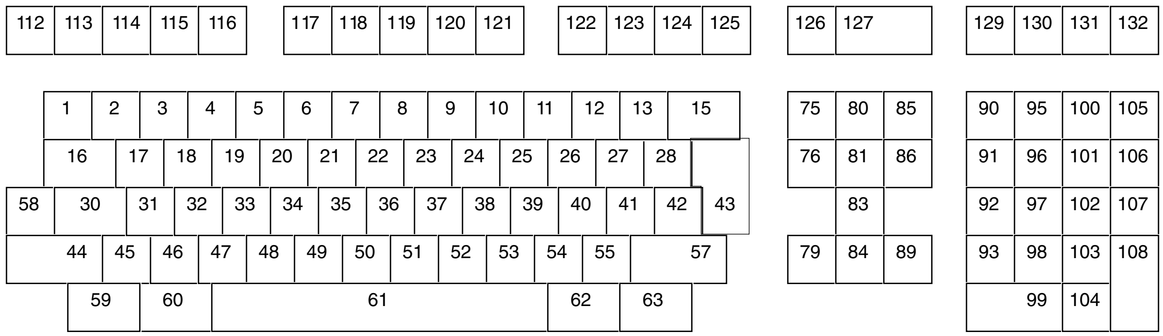

Figure 4.4: LK501 Typical North American Key Legends

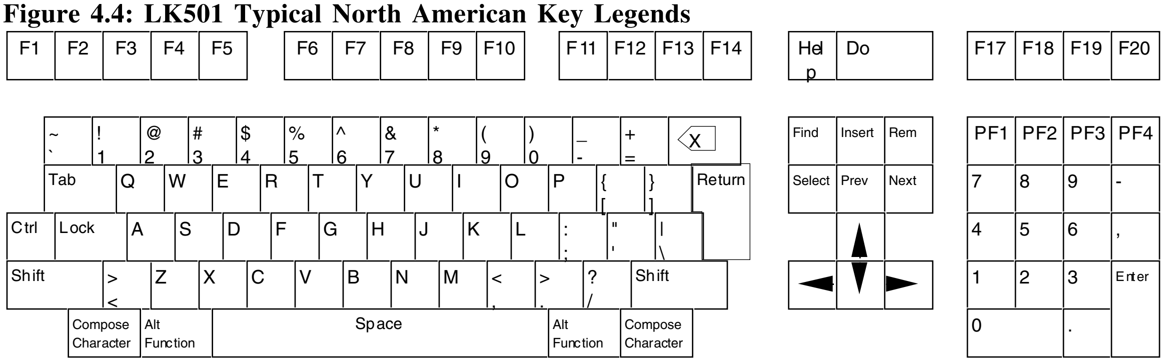

Table 4.2: LK501 Suggested Keycodes

Key #	Legend	Key Code	Position
	' ~	BF	E00
2	1	CO	E01
3	2	C5	E02
4	3	CB	E03
5	4	D0	E04
6	5	D6	E05
7	6	D B	E06
8	7	E0	E07
9	8	E5	E08
10	9	EA	E09
11	0	EF	E10
12	-	F9	E11
13	= +	F5	E12
17	a	C1	D01
18	W	C6	D02
19	E	CC	D03
20	R	D1	D04
21	T	D7	D05
22	Y	DC	D06
23	U	E1	D07
24	1	E6	D08
25	0	EB	D09
26	P	F0	D10
27	[{	F A	D11
28	]}	F6	D12
31	A	C2	C01

32	S	C7	C02
33	D	CD	C03
34	F	D2	C04
35	G	D8	C05
36	H	DD	C06
37	J	E2	C07
38	K	E7	C08
39	L	EC	C09
40	;:	F2	C10
41	' "	FB	C11
42	\I	F7	C12
45	<,	C9	B00
46	Z	C3	B01
47	X	C8	B02
48	C	CE	B03
49	V	D3	B04
50	B	D9	B05
51	N	DE	B06
52	M	E3	B07
53	, <	E8	B08
54	. >	ED	B09
55	/ ?	F3	B10
15	<X]	BC	
16	Tab	BE	
58	Ctrl	AF	
30	Lock	B0	
43	Return	BD	
44	Left Shift	AE	
57	Right Shift	AB	
59	Left Compose	B1	
60	Left Alt	AC	
61	Space	D4	
62	Right Alt	B2	
63	Right Compose	AD	

75	Find	8A
80	Insert Here	8B
85	Remove	8C
76	Select	8D
81	Prev	8E
86	Next	8F
83	Up Arrow	AA
84	Down Arrow	A9
79	Left Arrow	A7
89	Right Arrow	A8

90	PF1	A1
95	PF2	A2
100	PF3	A3
105	PF4	A4
91	KP7	9D
96	KP8	9E
101	KP9	9F
106	KP-	A0
92	KP4	99
97	KP5	9A
102	KP6	9B

107	KP,	9C
93	KP1	96
98	KP2	97
103	KP3	98
108	KPEnter	95
99	KP0	92
104	KP.	94
112	F1	56
113	F2	57
114	F3	58
115	F4	59
116	F5	5A
117	F6	64
118	F7	65
119	F8	66
120	F9	67
121	F10	68
122	F11	71
123	F12	72
124	F13	73
125	F14	74
126	Help	7C
127	Do	7D
128		7E
129	F17	80
130	F18	81
131	F19	82
132	F20	83

APPENDIX 4.A

LK501 DESIGN EXAMPLE

This appendix describes an example ACCESS.bus keyboard implementation and should not be viewed as specifying any conformance requirements.

4.A.1 LK501 Capabilities Information

prot (keyb) model(LK501-AA) type (keyboard) keymap (LK501-??)
feedback(click(7)bell(7)1ed(4(hold)3(1cck)))

The keyboard model number LK501-?? is used as the keymap identifier. The model suffix (??) is replaced by a two character country layout designation stored in non-volatile memory when the keyboard was manufactured.

The keyboard supports the "Application Click" and "Application Bell" commands with volume levels 0-7 (0=off, 7=maximum).

The keyboard supports two LED indicators labelled "Hold" and "Lock". These are illuminated by setting bits four and three in the illumination mask where bit one is the least significant bit.

4.A.2 Scan Timing

The keyboard is expected to be able to detect keys depressed for as little as twenty milliseconds (20 ms). Multiple keys should be detected in correct order when the interkeying times are not less than 8.3 ms. (120 Hz scan rate). Since the keyboard scan rate is 120 Hz, it is sometimes necessary to save a new key transition while waiting for the previous keyboard state to be reported. It is only necessary to "save" one scan worth of key transitions since the keyboard is expected to be allowed to report within a single frame time (16.6 ms at 60 Hz).

4.A.3 Keyclick and Bell

Keyclick and bell on the LK501 use the same transducer. Because of that, neither sound may be interrupted by the other. A single keyclick and bell may be remembered (pending) while the bell or previous keyclick is sounding. If additional keyclick or bell commands are received before the previous bell or keyclick has finished (including the beep space), they will be discarded. If both a keyclick and bell are pending when the previous beep space completes, the bell will sound next.

4.A.4 Write Non-volatile Memory (Vendor Reserved)

This command instructs the keyboard to write the specified six character string to its nonvolatile memory. The first two characters are used as the country layout designation and the remaining four characters are used as the device serial number. ControVStatus Vendor Reserved Command, P=l, Length=7.

Op-code: C3h Data: 6-byte data string.

The LK501 keyboard contains a 93C06 EEPROM used to provide non-volatile memory. While writing to the EEPROM, the keyboard will ignore the I2C bus. After programming, the keyboard will reset itself generating a power up attention message. The host can use the Identify Request message to verify that the new EEPROM data has been successfully written.

If the first byte of the EEPROM is FFh, the keyboard will assume the EEPROM is erased or not present and proceed without it. In this case, the model number will report as LK501-??, and the keyboard will generate a pseudo random negative number for the device number.

Programming Example: Assume the keyboard to be programmed has been assigned device address 52h.

Message Bytes	Comment
52	Destination address, message to device to be programmed
50	Source address, from host device
87	Control message (P=l), data length 7 bytes
C3	Vendor reserved op-code C3h, write EEPROM
41	"A" First byte of model variation as in LK501-AA
41	"A" Second byte of model variation
00	Start of 4 byte device number, SN=l in this example
00	
00	
01	
??	Check byte, equal to the XOR of the previous bytes.

4.A.5 Keyboard Generated Hardware Signals

The LK501 keyboard can be used to generate hardware Reset, Halt, and Attention signals if the I2C to computer interface supports them.

On the LK501 and similar keyboards:

CTRL-ALT-DEL	Signal 1, "Reset"
CTRL-ALT-RETURN	Signal 2, "Halt"
CTRL-ALT-SHIFT-DELETE	Signal 3, "Attention"

Refer to the ACCESS.bus Description and Protocol Specification for a description of these hardware signals.