

## 1. Overview of Features

The keyboard device has an interrupt routine which scans the keyboard matrix every video frame (20ms). This detects key presses and buffers a single character.

It supports programming of the eight function keys. Each one can be programmed separately for shifted and unshifted use giving effectively sixteen function keys. If the string programmed into any one of these function keys is of zero length then instead of returning characters, this function key will cause a software interrupt when it is pressed.

The eight function keys also each return a specific code if used with the CTRL or ALT keys, giving effectively another 16 functions.

The keyboard driver treats the joystick as if it were four cursor keys and provides diagonal movement by alternating two cursor codes. Autorepeat is supported on all keys. Both the delay until autorepeat begins, and the autorepeat speed can be altered.

The keyboard provides audible feedback by triggering the sound device to produce a click whenever a key is pressed. This can be disabled by the user.

There is no editing of lines since input is purely character, not line orientated. Any editing functions must be performed by the calling program or device.

## 2. Character Input

All input is done using the EXOS read character and read block calls. Read block is supported for compatibility with other devices although it is not very likely to be used. The keyboard is not an output device and so will not accept write function calls.

With the exception of the function keys which can be programmed with arbitrary strings, each key produces a single ASCII code. Many keys will produce different codes when used in conjunction with the CTRL, SHIFT or ALT keys.

## 2.1 Lock Modes

The keyboard is always in one of four modes; Normal, shift-lock, caps-lock or alt-lock. The default mode is normal. The mode can be changed by various key combinations:

```
CTRL LOCK - Enters Caps-lock mode.
SHIFT LOCK - Enters Shift-lock mode.
ALT LOCK - Enters Alt-lock mode.
LOCK - Returns to Normal mode.
```

When the keyboard is in any of the lock modes then it behaves as if the appropriate SHIFT, CTRL or ALT key was held down permanently. If the appropriate key actually is held down during a lock mode then it temporarily counteracts the effect of the lock. Thus for example in SHIFT LOCK mode the action of the SHIFT key is effectively reversed. In this example if the CTRL key is used while in SHIFT LOCK mode it will behave as if it was in NORMAL mode. This applies to all other combinations.

The current lock mode is indicated on the status line the first six characters of which are reserved for the keyboard. It displays the word SHIFT, CAPS or ALT as appropriate and is blank for normal mode.

There is an EXOS variable LOCK\_KEY which is always set to the current lock status according to the following codes:

```
0 - Un-locked
1 - CAPS lock
2 - SHIFT lock
8 - ALT lock
```

If this EXOS variable is changed by the user then the next keyboard interrupt will update the lock mode appropriately. Any values other than the above which are put into the variable will be changed to one of the four allowed values.

## 2.2 Key Codes

These are the ASCII codes returned by each key both normally, and with SHIFT, CTRL and ALT. (All values are in hexadecimal.)

Key	NORMAL	SHIFT	CONTROL	ALT
1	31	21	31	31
2	32	22	32	32
3	33	23	33	33
4	34	24	34	34
5	35	25	35	35
6	36	26	36	36
7	37	27	37	37
8	38	28	38	38
9	39	29	39	39
0	30	5F	1F	9F
-	2D	3D	2D	2D
^	5E	7E	1E	9E
@	40	60	00	80
[	5B	7B	1B	9B
;	3B	2B	3B	3B
:	3A	2A	3A	3A
]	5D	7D	1D	9D
\	5C	7C	1C	9C
,	2C	3C	2C	2C
.	2E	3E	2E	2E
/	2F	3F	2F	2F
space	20	20	20	20
A	61	41	01	81
B	62	42	02	82
C	63	43	03	83
D	64	44	04	84
E	65	45	05	85
F	66	46	06	86
G	67	47	07	87
H	68	48	08	88
I	69	49	09	89
J	6A	4A	0A	8A
K	6B	4B	0B	8B
L	6C	4C	0C	8C
M	6D	4D	0D	8D
N	6E	4E	0E	8E
O	6F	4F	0F	8F
P	70	50	10	90
Q	71	51	11	91
R	72	52	12	92
S	73	53	13	93
T	74	54	14	94
U	75	55	15	95
V	76	56	16	96
W	77	57	17	97

X	78	58	18	98
Y	79	59	19	99
Z	7A	5A	1A	9A
ENTER	0D	0D	0D	0D
ESC	1B	1B	1B	1B
TAB	09	09	09	09
-----				
DEL	A0	A1	A2	A3
ERASE	A4	A5	A6	A7
INS	A8	A9	AA	AB
STOP	03	03	03	03
joy up	B0	B1	B2	B3
joy down	B4	B5	B6	B7
joy left	B8	B9	BA	BB
joy right	BC	BD	BE	BF
Function 1	.	.	F0	F8
Function 2	.	.	F1	F9
Function 3	.	.	F2	FA
Function 4	.	.	F3	FB
Function 5	.	.	F4	FC
Function 6	.	.	F5	FD
Function 7	.	.	F6	FE
Function 8	.	.	F7	FF

### 3. Special Features

#### 3.1 Keyclick control

When the interrupt routine detects a key press it calls a routine called KEYCLICK in the sound driver which produces an audible click. This routine is only called if the EXOS variable KEY\_CLICK is zero. Thus setting this variable to a non-zero value will disable key click.

#### 3.2 Autorepeat Control

Autorepeat is controlled by two EXOS variables. DELAY\_KEY is the delay until autorepeat starts in 1/50 seconds. If it is zero then autorepeat is disabled.

The other EXOS variable is RATE\_KEY which is the delay between each repetition of the key, also in 1/50 seconds.

### 3.3 Function key control

There are sixteen logical programmable function keys numbered 0 to 15. Keys 0 to 7 refer to the basic function keys, 8 to 15 are the shifted versions. Any one of these may be programmed with a string of characters (which may include control codes etc.) using a special function call. The default string for all keys is a null string.

Parameters: B = 8 (Special function code)  
 C = Function key number (0..15)  
 DE = Pointer to string (Length byte first)

Returns: A = Status

The maximum length for each programmed string is 23 characters excluding the length byte. An error will be returned if the string is too long.

If the programmed string is of zero length (the null string) then this function key will cause a software interrupt when it is pressed. The software interrupt code will be 10h for function key 0, up to 1Fh for function key 15.

### 3.4 Stop Key Control

There is an EXOS variable called STOP\_IRQ which controls the action of the STOP key. If it is non-zero then the stop key simply returns the ASCII Ctrl-C code (03h) in the same way as all other keys. If STOP\_IRQ is zero then instead of this a software interrupt is caused, with software interrupt code 20h.

### 3.5 Hold Key Control

When the hold key is pressed the keyboard driver will hang up in its interrupt routine until the key is pressed again. This will thus freeze any listing etc. which is being produced. When it hangs up it will call the routine SILENCE in the sound driver to ensure that the DAVE chip is not left in a noisy state.

When the hold key is pressed it displays the message "HOLD" in place of the current lock mode on the status line. This message will be replaced by the correct lock mode message (which is blank for normal mode) when interrupts start again. If the STOP key is pressed while in HOLD mode then this will force an exit from HOLD mode, and will then respond to the STOP key in the normal way.



### 3.6 Normal Key Software Interrupts

When a normal key is pressed the character code for it is simply put in the buffer. However if the EXOS variable `KEY_IRQ` is non-zero then as well as returning the character code, a software interrupt will be caused with software interrupt code 21h.

### 3.7 Direct Joystick Reading

A special function call is provided which will directly read the joystick on the main keyboard, or one of the two external joysticks on the control ports. The parameters for this are:

```
Parameters:  B = 9  (Special function call)
              C = 0  (internal joystick)
              = 1  (external joystick 1)
              = 2  (external joystick 2)
Returns:     A = Status
              C = b0 - Set if RIGHT presses
                b1 - Set if LEFT pressed
                b2 - Set if DOWN pressed
                b3 - Set if UP pressed
                b4 - Set if FIRE pressed
                b5..b6 - Clear
```

Note that for the internal joystick the "fire" button is in fact the space bar.