

KeyWarrior - Custom Scancode Tables

Status: August 12th 2008

1. Introduction

Keyboards are organized as matrixes of switches. The placement of the keys in the matrix is mostly arbitrary with just a few constraints to keep keys from interfering with each other.

The keyboard controller or encoder does read the matrix and translates the matrix coordinates into the proper scancodes. Scancodes are not ASCII characters but merely a standardized code for each key, telling the system that a certain key was pressed. It is then up to the keyboard driver to turn this information into actual characters. Scancodes are fixed for a type of interface. Different interfaces (ADB, USB, PS/2) use different scancodes and PS/2 does even support three software selectable scancode sets.

To minimize the workload in adapting KeyWarrior to a specific keyboard layout it uses a Master Translation Table. This Master Translation Table does translate the product specific coordinates into a fixed set of scancodes. Since USB has the most complete definition of scancodes the USB usage codes are utilized for the Master Translation Table. The USB usage codes are converted to the codes of ADB and PS/2 as necessary.

The use of the USB usage codes reduces the adaption of a specific keyboard to KeyWarrior to generating a single keyboard specific table.

Optionally if the Function-Shift keys are used a second and maybe a third table may be needed.

2. Definition of terms

The following terms will be used throughout this document as defined here:

Row -	Horizontal scan line, or X scanline. KeyWarrior supports 8 rows.
Column -	Vertical scanline, or Y scanline, KeyWarrior supports 8 or 16 columns.
Physical Scancode -	The physical position of a key in the matrix. Physical scancodes are calculated as: $X + (Y * 8)$ or $X + (Y * 12)$ for KeyWarrior 20
Logical Scancode -	A USB usage code as specified by the "Device Class Definition for Human Interface Devices". KeyWarrior translates Physical Scancodes to logical Scancodes before further processing them.
Function-Shift -	KeyWarrior models may support one or two keys to switch to alternate scancode tables. This allows small keyboards with fewer keys to still offer all functions of full sized keyboards. The availability of Function-Shift depends on the KeyWarrior model, refer to the datasheet for details.
Master Translation Table -	The scancode table which is used to translate Physical Scancodes to Logical Scancodes. If Function-Shift keys are used the Master Translation Table contains two or three codesets.

3. Building a Master Translation Table

The Master Translation Table uses the physical scancode of a key as the index to the table and has the desired USB usage code as the entry in that position. A Master Translation Table has the size of the matrix supported by the individual KeyWarrior variant times the number of Function-Shift levels. I.e. the KeyWarrior 16, which supports matrixes with up to 8 x 16 keys and one additional Function-Shift level (two levels total), has a Master Translation Table with 256 entries. The first 128 entries of that table are used when the Function-Shift key is not pressed, the second half is used when the Function-Shift key is pressed.

A Function-Shift key is assigned by placing the respective keycode in both parts of the Master Translation Table. KeyWarrior 16 and KeyWarrior Combo support only the Function-Shift 1 key. KeyWarrior 8 supports both Function Shift keys and has a matrix size of 8 x 8. This results in a Master Translation Table with three codesets with 64 entries each for a total table size of 192 bytes.

KeyWarrior 20 does not support Function-Shift keys, it uses a single table with 240 entries.

When creating the Master Translation Table keep in mind that all Function-Shift levels of the table have to contain all scancodes which should be generated for that Function-shift level. Keys that do not change their

code must have identical codes in all Function-Shift levels of the Master Translation Table.

The Function-Shift key always has to have its code in all levels of the Master Translation Table. A Function-Shift key may not change its code between the levels, unpredictable results may occur.

In the attached table you will find the usage codes and the key numbers (not scancodes!) which they do represent on the standard (ISO) AT and Apple Extended ISO keyboards.

Usage Code	Function	AT-101 Ref.	Apple Ext. Ref.
\$00	Reserved - KeyWarrior sends no code	N/A	N/A
\$01	Reserved	N/A	N/A
\$02	Reserved	N/A	N/A
\$03	Reserved	N/A	N/A
\$04	Keyboard d a and A	31	61
\$05	Keyboard d b and B	50	82
\$06	Keyboard d c and C	48	80
\$07	Keyboard d d and D	33	63
\$08	Keyboard d e and E	19	42
\$09	Keyboard d f and F	34	64
\$0A	Keyboard d g and G	35	65
\$0B	Keyboard d h and H	36	66
\$0C	Keyboard d i and I	24	47
\$0D	Keyboard d j and J	37	67
\$0E	Keyboard d k and K	38	68
\$0F	Keyboard d l and L	39	69
\$10	Keyboard d m and M	52	84
\$11	Keyboard d n and N	51	83
\$12	Keyboard d o and O	25	48
\$13	Keyboard d p and P	26	49
\$14	Keyboard d q and Q	17	40
\$15	Keyboard d r and R	20	43
\$16	Keyboard d s and S	32	62
\$17	Keyboard d t and T	21	44
\$18	Keyboard d u and U	23	46
\$19	Keyboard d v and V	49	81
\$1A	Keyboard d w and W	18	41
\$1B	Keyboard d x and X	47	79
\$1C	Keyboard d y and Y German: z and Z	22	45
\$1D	Keyboard d z and Z German: y and Y	46	78
\$1E	Keyboard d 1 and !	2	19
\$1F	Keyboard d 2 and @ German: 2 and "	3	20
\$20	Keyboard d 3 and # German: 3 and §	4	21
\$21	Keyboard d 4 and \$	5	22
\$22	Keyboard d 5 and %	6	23
\$23	Keyboard d 6 and ^ German: 6 and &	7	24
\$24	Keyboard d 7 and & German: 7 and /	8	25
\$25	Keyboard d 8 and * German: 8 and (9	26
\$26	Keyboard d 9 and (German: 9 and)	10	27
\$27	Keyboard d 0 and) German: 0 and =	11	28
\$28	Keyboard d Return	43	72
\$29	Keyboard d ESCAPE	110	1
\$2A	Keyboard d Backspace	15	31
\$2B	Keyboard d Tab	16	39
\$2C	Keyboard d SPACE	61	96
\$2D	Keyboard d - and _ German: ß and ?	12	29
\$2E	Keyboard d = and + German: ´ and `	13	30

Usage Code	Function	AT-101 Ref.	Apple Ext. Ref.
\$2F	Keyboard [and { German: ü and Ü	27	50
\$30	Keyboard] and } German: + and *	28	51
\$31	Keyboard \ and German: # and '	29	52
\$32	Keyboard (non US) # and ~	42	52
\$33	Keyboard ; and : German: ö and Ö	40	70
\$34	Keyboard ' and " German: ä and Ä	41	71
\$35	Keyboard ` and ~ German: ^and °	1	106
\$36	Keyboard , and < German: , and ;	53	85
\$37	Keyboard . and > German: . and :	54	86
\$38	Keyboard / and ? German: - and _	55	87
\$39	Keyboard Caps Lock (not locking)	30	60
\$3A	Keyboard F1	112	2
\$3B	Keyboard F2	113	3
\$3C	Keyboard F3	114	4
\$3D	Keyboard F4	115	5
\$3E	Keyboard F5	116	6
\$3F	Keyboard F6	117	7
\$40	Keyboard F7	118	8
\$41	Keyboard F8	119	9
\$42	Keyboard F9	120	10
\$43	Keyboard F10	121	11
\$44	Keyboard F11	122	12
\$45	Keyboard F12	123	13
\$46	Keyboard Print Screen (Mac: F13)	124	14
\$47	Keyboard Scroll Lock (Mac: F14)	125	15
\$48	Keyboard Pause (Mac: F15)	126	16
\$49	Keyboard Insert (Mac: Help)	75	32
\$4A	Keyboard Home	80	33
\$4B	Keyboard PageUp	85	34
\$4C	Keyboard Delete Forward	76	53
\$4D	Keyboard End	81	54
\$4E	Keyboard PageDown	86	55
\$4F	Keyboard RightArrow	89	102
\$50	Keyboard LeftArrow	79	100
\$51	Keyboard DownArrow	84	101
\$52	Keyboard UpArrow	83	89
\$53	Keypad NumLock and Clear	90	35
\$54	Keypad /	95	37
\$55	Keypad *	100	38
\$56	Keypad -	105	59
\$57	Keypad +	106	76
\$58	Keypad ENTER	108	105
\$59	Keypad 1 and End	93	90
\$5A	Keypad 2 and DownArrow	98	91
\$5B	Keypad 3 and PageDown	103	92
\$5C	Keypad 4 and LeftArrow	92	73
\$5D	Keypad 5	97	74

Usage Code	Function	AT-101 Ref.	Apple Ext. Ref.
\$5E	Keypad 6 and RightArrow	102	75
\$5F	Keypad 7 and Home	91	56
\$60	Keypad 8 and UpArrow	96	57
\$61	Keypad 9 and PageUp	101	58
\$62	Keypad 0 and Insert	99	103
\$63	Keypad . and Delete	104	104
\$64	Keyboard non US \ and German: < and >	45	18
\$65	Keyboard Application (Menu)	129	N/A
\$66	Keyboard Power	N/A	17
\$67	Keypad =	N/A	36
\$68	Keyboard F13	N/A	(14)*
\$69	Keyboard F14	N/A	(15)*
\$6A	Keyboard F15	N/A	(16)*
\$6B	Keyboard F16	N/A	N/A
\$6C	Keyboard F17	N/A	N/A
\$6D	Keyboard F18	N/A	N/A
\$6E	Keyboard F19	N/A	N/A
\$6F	Keyboard F20	N/A	N/A
\$70	Keyboard F21	N/A	N/A
\$71	Keyboard F22	N/A	N/A
\$72	Keyboard F23	N/A	N/A
\$73	Keyboard F24	N/A	N/A
\$74	Keyboard Execute	N/A	N/A
\$75	Keyboard Help	N/A	N/A
\$76	Keyboard Menu	N/A	N/A
\$77	Keyboard Select	N/A	N/A
\$78	Keyboard Stop	N/A	N/A
\$79	Keyboard Again	N/A	N/A
\$7A	Keyboard Undo	N/A	N/A
\$7B	Keyboard Cut	N/A	N/A
\$7C	Keyboard Copy	N/A	N/A
\$7D	Keyboard Paste	N/A	N/A
\$7E	Keyboard Find	N/A	N/A
\$7F	Keyboard Mute	N/A	N/A
\$80	Keyboard Volume Up	N/A	N/A
\$81	Keyboard Volume Down	N/A	N/A
\$82	Keyboard Locking Caps Lock **	N/A	N/A
\$83	Keyboard Locking Num Lock **	N/A	N/A
\$84	Keyboard Locking Scroll Lock **	N/A	N/A
\$85	Keypad Comma	N/A	N/A
\$86	Keypad Equal Sign	N/A	N/A
\$87	Keyboard Kanji 1	N/A	N/A
\$88	Keyboard Kanji 2	N/A	N/A
\$89	Keyboard Kanji 3	N/A	N/A
\$8A	Keyboard Kanji 4	N/A	N/A
\$8B	Keyboard Kanji 5	N/A	N/A
\$8C	Keyboard Kanji 6	N/A	N/A

Usage Code	Function	AT-101 Ref.	Apple Ext. Ref.
\$8D	Keyboard Kanji 7	N/A	N/A
\$8E	Keyboard Kanji 8	N/A	N/A
\$8F	Keyboard Kanji 9	N/A	N/A
\$90	Keyboard LANG1	N/A	N/A
\$91	Keyboard LANG2	N/A	N/A
\$92	Keyboard LANG3	N/A	N/A
\$93	Keyboard LANG4	N/A	N/A
\$94	Keyboard LANG5	N/A	N/A
\$95	Keyboard LANG6	N/A	N/A
\$96	Keyboard LANG7	N/A	N/A
\$97	Keyboard LANG8	N/A	N/A
\$98	Keyboard LANG9	N/A	N/A
\$99	Keyboard Alternate Erase	N/A	N/A
\$9A	Keyboard SysReq/Attention	N/A	N/A
\$9B	Keyboard Cancel	N/A	N/A
\$9C	Keyboard Clear	N/A	N/A
\$9D	Keyboard Prior	N/A	N/A
\$9E	Keyboard Return	N/A	N/A
\$9F	Keyboard Separator	N/A	N/A
\$A0	Keyboard Out	N/A	N/A
\$A1	Keyboard Oper	N/A	N/A
\$A2	Keyboard Clear /Again	N/A	N/A
\$A3	Keyboard CrSel /Props	N/A	N/A
\$A4	Keyboard ExSel	N/A	N/A
\$A5- DF	Reserved		
\$E0	Keyboard LeftControl	58	93
\$E1	Keyboard LeftShift	44	77
\$E2	Keyboard LeftAlt (Option)	60	94
\$E3	Keyboard LeftGUI (Windows/Command)	127	95
\$E4	Keyboard RightControl	64	99
\$E5	Keyboard RightShift	57	88
\$E6	Keyboard RightAlt (Option)	62	98
\$E7	Keyboard RightGUI (Windows/Command)	128	97
\$E8- FF	Reserved		

Usage Code	Function	AT-101 Ref.	Apple Ext. Ref.
	Special Codes for KeyWarrior		
\$BO- \$DF	Macros for Commander	N/A	N/A
\$E8	LeftControl break - Commander only	N/A	N/A
\$E9	LeftShift break - Commander only	N/A	N/A
\$EA	LeftAlt break - Commander only	N/A	N/A
\$EB	LeftGUI break - Commander only	N/A	N/A
\$EC	RightControl break - Commander only	N/A	N/A
\$ED	RightShift break - Commander only	N/A	N/A
\$EE	RightAlt break - Commander only	N/A	N/A
\$EF	RightGUI break - Commander only	N/A	N/A
\$FB	Middle Mouse Button * * *	N/A	N/A
\$FC	Right Mouse Button * * *	N/A	N/A
\$FD	Left Mouse Button * * *	N/A	N/A
\$FE	Function Shift 2 * * * *	N/A	N/A
\$FF	Function Shift 1 * * * * *	N/A	N/A

* These codes should not be used for cross platform compatibility. MacOS does decode F13-15 usage codes properly but not all operating systems may do this. Use the "Print", "Scroll Lock", "Pause" codes instead, those are properly accepted by all currently available systems.

** KeyWarrior does not support locking keys.

*** Mouse button codes can be used on KeyWarrior Combo chips only.

**** FN2 is available only on KeyWarrior 8 variants and KeyWarrior Commander.

***** FN1 is not available on KeyWarrior 20 variants.

4. Mouse support with KeyWarrior Combo

The KeyWarrior Combo chips require the mouse buttons to be inside the matrix. Use the mouse button codes and place them in the Master Translation table to assign the mouse buttons. Make sure to place the mouse button codes in both parts of the table, otherwise pressing or releasing the Function Shift key can alter the state of a pressed mouse button.

5. Master Translation Table format

The Master Translation Table should be supplied by the customer in electronic form. Only Master Translation Tables that arrive properly formatted allow us to process your order for the standard price.

Several files with an empty Master Translation Table are available on our website. The files vary with the matrix size and there is a second set for KeyWarrior Operator controllers. Please use the appropriate file to create your Master Translation Table and email it with your order. Do not change the formatting of the file, just put your usage codes in the proper places of the table.

Master Translation Tables that are not directly useable for generating the KeyWarrior code are subject to surcharges.

6. KeyWarrior Flex programming

The utility program for downloading the Master Translation Table into KeyWarrior Flex controllers does use the same source file as the preprogrammed KeyWarrior chips. To configure a KeyWarrior Flex controller use the appropriate empty file and fill in the usage codes. Then use the Flex utility to download this file into the KeyWarrior Flex.

The identical file format makes KeyWarrior Flex a good option for prototyping KeyWarrior controllers.

If the EEPROM should be programmed with a device programmer rather than in-circuit a hex file containing the Master Translation Table is necessary. Such a file can be generated either manually or by running the Master Translation Table file through an appropriate assembler.

6.1 KeyWarrior programming utility on Windows

If you are using the KeyWarrior programming tool on Windows be aware that downloading of the data into KeyWarrior happens only when the keyboard is plugged in while the program is running and has a Master Translation Table loaded.

To program a KeyWarrior Flex or Operator on Win 98 first start the tool. Then select "Open" from the "File" Menu and load a Master Translation Table. Now connect the KeyWarrior based keyboard to download the Master Translation Table.

7. KeyWarrior Operator programming

The Master Translation Table for KeyWarrior Operator has eight bytes for every matrix coordinate. You need to use the appropriate empty files to generate a Master Translation Table for an Operator controller.

Usage codes programmed into the Operator Master Translation Table are send left to right when the key is pressed and right to left when the key is released (if this applies to the active interface). Any \$00 in the Master Translation Table will be handled as a No-Op and will not generate any code.

KeyWarrior Operator has no support for releasing a key in a sequence (i.e. Shift down, A, Shift Up, b is not possible). It does also not support entries which have the same key more than once, so it is not possible to generate "aa" from a single keystroke.

8 KeyWarrior Commander programming

The Master Translation Table of KeyWarrior Commander is mostly identical to that of KeyWarrior and KeyWarrior Flex. There are single byte entries for each key. Though any of the keys in the matrix may be assigned a special code that points to one of 48 macros. The codes \$B0 to \$DF do not directly generate a keycode but point to the macros.

Each of the 48 macros may be set to either "stable mode" or "typing mode" and may have up to 31 codes.

In stable mode all of the codes will be send as key downs when the key is pressed and as key ups when the key is released again, identical to KeyWarrior Operator. This mode allows special keys that do auto repeat or signal codes (i.e. ctrl-alt-del), it does not allow things like entering "http://" or "Hello" as individual codes can not be repeated or released.

Typing mode does generate a stream of key down and up codes from a macro. All keys except Ctrl, Alt, Shift and GUI do get released again immediatley, this allows to repeat the same key multiple times in a macro. The modifiers (Ctrl, Alt, Shift, GUI) do get send as a key down event when their code is encountered in a macro. The key up code for them is not send until either the key is released again or a special release code is encountered in the macro.

Legal Stuff

This document is ©1999-2008 by Code Mercenaries. The information contained herein is subject to change without notice. Code Mercenaries makes no claims as to the completeness or correctness of the information contained in this document.

Code Mercenaries assumes no responsibility for the use of any circuitry other than circuitry embodied in a Code Mercenaries product. Nor does it convey or imply any license under patent or other rights.

Code Mercenaries products may not be used in any medical apparatus or other technical products that are critical for the functioning of lifesaving or supporting systems. We define these systems as such that in the case of failure may lead to the death or injury of a person. Incorporation in such a system requires the explicit written permission of the president of Code Mercenaries.

Trademarks used in this document are properties of their respective owners.

Code Mercenaries
Hard- und Software GmbH
Karl-Marx-Str. 147a
12529 Schönefeld
Germany
Tel: x49-3379-20509-20

Fax: x49-33790-20509-30
Mail: support@codemerces.com
Web: www.codemerces.com

HRB 16007 P
Geschäftsführer: Guido Körber, Christian Lucht