Reference > Language > Functions > Usb > Keyboard > Keyboardmodifiers

Keyboard Modifiers and Special Keys

Description

When given a printable ASCII character as an argument, the functions `Keyboard.write()`, `Keyboard.press()` and `Keyboard.release()` simulate actuations on the corresponding keys. These functions can also handle ASCII characters that require pressing a key in combination with Shift or, on international keyboards, AltGr. For example:

```
Keyboard.write('a');  // press and release the 'A' key
Keyboard.write('A');  // press Shift and 'A', then release both
```

A typical keyboard, however, has many keys that do not match a printable ASCII character. In order to simulate those keys, the library provides a set of macros that can be passed as arguments to `Keyboard.write()`, `Keyboard.press()` and `Keyboard.release()`. For example, the key combination Shift+F2 can be generated by:

```
Keyboard.press(KEY_LEFT_SHIFT);  // press and hold Shift
Keyboard.press(KEY_F2);          // press and hold F2
Keyboard.releaseAll();           // release both
```

Note that, in order to press multiple keys simultaneously, one has to use `Keyboard.press()` rather than `Keyboard.write()`, as the latter just “hits” the keys (it presses and immediately releases them).

The available macros are listed below:

Keyboard modifiers

These keys are meant to modify the normal action of another key when the two are pressed in combination.

KEY	HEXADECIMAL VALUE	DECIMAL VALUE	NOTES
KEY_LEFT_CTRL	0x80	128	
KEY_LEFT_SHIFT	0x81	129	

KEY	HEXADECIMAL VALUE	DECIMAL VALUE	NOTES
KEY_LEFT_ALT	0x82	130	Option (⌥) on Mac
KEY_LEFT_GUI	0x83	131	OS logo, Command (⌘) on Mac
KEY_RIGHT_CTRL	0x84	132	
KEY_RIGHT_SHIFT	0x85	133	
KEY_RIGHT_ALT	0x86	134	also AltGr, Option (⌥) on Mac
KEY_RIGHT_GUI	0x87	135	OS logo, Command (⌘) on Mac

Special keys

These are all the keys that do not match a printable ASCII character and are not modifiers.

Within the alphanumeric cluster

KEY	HEXADECIMAL VALUE	DECIMAL VALUE
KEY_TAB	0xB3	179
KEY_CAPS_LOCK	0xC1	193
KEY_BACKSPACE	0xB2	178
KEY_RETURN	0xB0	176
KEY_MENU	0xED	237

Navigation cluster

KEY	HEXADECIMAL VALUE	DECIMAL VALUE
KEY_INSERT	0xD1	209
KEY_DELETE	0xD4	212
KEY_HOME	0xD2	210
KEY_END	0xD5	213
KEY_PAGE_UP	0xD3	211
KEY_PAGE_DOWN	0xD6	214
KEY_UP_ARROW	0xDA	218
KEY_DOWN_ARROW	0xD9	217
KEY_LEFT_ARROW	0xD8	216
KEY_RIGHT_ARROW	0xD7	215

Numeric keypad

KEY	HEXADECIMAL VALUE	DECIMAL VALUE
KEY_NUM_LOCK	0xDB	219
KEY_KP_SLASH	0xDC	220
KEY_KP_ASTERISK	0xDD	221
KEY_KP_MINUS	0xDE	222
KEY_KP_PLUS	0xDF	223
KEY_KP_ENTER	0xE0	224
KEY_KP_1	0xE1	225
KEY_KP_2	0xE2	226
KEY_KP_3	0xE3	227
KEY_KP_4	0xE4	228
KEY_KP_5	0xE5	229
KEY_KP_6	0xE6	230

KEY	HEXADECIMAL VALUE	DECIMAL VALUE
KEY_KP_7	0xE7	231
KEY_KP_8	0xE8	232
KEY_KP_9	0xE9	233
KEY_KP_0	0xEA	234
KEY_KP_DOT	0xEB	235

Escape and function keys

The library can simulate function keys up to F24.

KEY	HEXADECIMAL VALUE	DECIMAL VALUE
KEY_ESC	0xB1	177
KEY_F1	0xC2	194
KEY_F2	0xC3	195
KEY_F3	0xC4	196
KEY_F4	0xC5	197
KEY_F5	0xC6	198
KEY_F6	0xC7	199
KEY_F7	0xC8	200
KEY_F8	0xC9	201
KEY_F9	0xCA	202
KEY_F10	0xCB	203
KEY_F11	0xCC	204
KEY_F12	0xCD	205
KEY_F13	0xF0	240
KEY_F14	0xF1	241
KEY_F15	0xF2	242

KEY	HEXADECIMAL VALUE	DECIMAL VALUE
KEY_F16	0xF3	243
KEY_F17	0xF4	244
KEY_F18	0xF5	245
KEY_F19	0xF6	246
KEY_F20	0xF7	247
KEY_F21	0xF8	248
KEY_F22	0xF9	249
KEY_F23	0xFA	250
KEY_F24	0xFB	251

Function control keys

These are three keys that sit above the navigation cluster.

KEY	HEXADECIMAL VALUE	DECIMAL VALUE	NOTES
KEY_PRINT_SCREEN	0xCE	206	Print Screen or PrtSc / SysRq
KEY_SCROLL_LOCK	0xCF	207	
KEY_PAUSE	0xD0	208	Pause / Break

International keyboard layouts

Some national layouts define extra keys. For example, the Swedish and Danish layouts define `KEY_A_RING` as `0xB7`, which is the key to the right of “P”, labeled “Å” on those layouts and “{”/“[” on the US layout. In order to use those definitions, one has to include the proper Keyboard_*.h file. For example:

```
#include <Keyboard.h>
#include <Keyboard_sv_SE.h> // extra key definitions from Swedish layou
```