

# Linux problems with USB HID keyboards and patches

**Source:** <http://linux.derkeiler.com/Mailing-Lists/Kernel/2003-07/2464.html>

---

**From:** Diego SANTA CRUZ (*Diego.SantaCruz\_at\_epfl.ch*)

**Date:** 07/29/03

To: vojtech@suse.cz, linux-kernel@vger.kernel.org

Date: 29 Jul 2003 13:37:13 +0200

[Not subscribed to l-k, so please CC me]

Hi,

I've been experiencing some problems with a Logitech iTouch Internet Navigator USB connected keyboard (model Y-BF37). I solved some of them by patching the kernel, while others I don't know how to handle. Kernel is an ACPI enabled version of RH 2.4.20-19.9.

They problems are:

\* Problem 1. Pressing Shift+Alt\_R generates a fake Cltr\_R key press event with no Ctrl\_R key up event, so the Ctrl key gets stuck. Pressing the Ctrl\_R key just after generates only a key release event and then everything gets back to normal. I added some printk messages to keybdev, hid-input and hid-core and this is what I got:

```
# ---> Press Right Shift
hid_core.c:hid_input_field: count = 8
hidinput_hid_event: page 0x7, usage 0xe5, code 54
hidinput_hid_event: key 54 (0x36), down = 1
hid_core.c:hid_input_field: count = 6
hidinput_hid_event: key 54 (0x36), down = 2
# ---> Press Right Alt
# but Right Ctrl synthetized in HW or hid_input_field()
hid_core.c:hid_input_field: count = 8
hidinput_hid_event: page 0x7, usage 0xe4, code 97
keybdev event key 97 (0x61), down = 1
hidinput_hid_event: page 0x7, usage 0xe6, code 100
keybdev event key 100 (0x64), down = 1
hid_core.c:hid_input_field: count = 6
# ---> Release Right Alt
hid_core.c:hid_input_field: count = 8
hidinput_hid_event: page 0x7, usage 0xe6, code 100
```

## Linux-Kernel: Linux problems with USB HID keyboards and patches

```
keybdev event key 100 (0x64), down = 0
hid_core.c:hid_input_field: count = 6
# ---> Release Right Shift
hid_core.c:hid_input_field: count = 8
hidinput_hid_event: page 0x7, usage 0xe5, code 54
hidinput_hid_event: key 54 (0x36), down = 0
hid_core.c:hid_input_field: count = 6
# ---> Press Right Shift
hid_core.c:hid_input_field: count = 8
hidinput_hid_event: page 0x7, usage 0xe5, code 54
keybdev event key 54 (0x36), down = 1
hid_core.c:hid_input_field: count = 6
keybdev event key 54 (0x36), down = 2
# ---> Press Right Alt
hid_core.c:hid_input_field: count = 8
hidinput_hid_event: page 0x7, usage 0xe6, code 100
keybdev event key 100 (0x64), down = 1
hid_core.c:hid_input_field: count = 6
# ---> Release Right Alt
hid_core.c:hid_input_field: count = 8
hidinput_hid_event: page 0x7, usage 0xe6, code 100
keybdev event key 100 (0x64), down = 0
hid_core.c:hid_input_field: count = 6
# ---> Release Right Shift
hid_core.c:hid_input_field: count = 8
hidinput_hid_event: page 0x7, usage 0xe5, code 54
keybdev event key 54 (0x36), down = 0
hid_core.c:hid_input_field: count = 6
# ---> Press and Release Right Ctrl, but only release reported
hid_core.c:hid_input_field: count = 8
hidinput_hid_event: page 0x7, usage 0xe4, code 97
keybdev event key 97 (0x61), down = 0
hid_core.c:hid_input_field: count = 6
```

In the above count is the count variable in `hid_input_field()`, page and usage the page and usage parts of the `usage->hid` value in `hidinput_hid_event()` and code is the `usage->code` value in the same function. Finally key is the value of code in `keybdev_event()`, decimal and hex.

Any ideas what's going wrong? Is it a buggy keyboard? Or a problem in `hid_input_field()`? Or something else?

\* Problem 2: The mapping of HID keys to event keys and from event keys to PS/2 scancodes are not very nice for "internet" keyboards. I reworked them to recognize many more HID keys and use the standard MS scancodes, supplemented by the MS Office keyboard mappings too. That is implemented by the patch `linux-2.4.20-18.9.acpi.4-input.patch`. These mappings could be worth having in 2.4 and also in 2.6 (2.6 has better mappings, but many are still non-standard).

## Linux-Kernel: Linux problems with USB HID keyboards and patches

\* Problem 3: My keyboard (and I guess some others too) report some of the special keys as buttons. They get assigned the BTN\_\* event keys. However, these are interpreted by mousedev as regular mouse buttons and are ignored by keybdev, which is quite annoying. The attached linux-2.4.20-18.9.acpi.4-keyb-but.patch makes mousedev ignore those and assigns KEY\_PROG\* event keys to those in keybdev, so that they behave as programmable keys.

Any feedback about the patches is appreciated.

Best,

Diego

--

-----  
Diego Santa Cruz - PhD  
Publications available at <http://ltswww.epfl.ch/~dsanta>  
Signal Processing Institute (LTS1 / ITS / STI)  
Swiss Federal Institute of Technology (EPFL)  
EPFL - STI - ITS, CH-1015 Lausanne, Switzerland  
E-mail: Diego.SantaCruz@epfl.ch  
Phone: +41 - 21 - 693 26 57  
Fax: +41 - 21 - 693 76 00  
-----

D Recognize more keys on USB keyboards, in particular many keys  
D defined in the HID consumer page.

D Use MS standard and MS Office keyboard USB HID to PS/2 scancode  
D mapping, plus custom mapping for other keys, this makes the scancodes  
D for USB keyboards compatible with most PS/2 keyboards.

D

--- linux-2.4.20-18.9.acpi.4/include/linux/input.h.orig 2003-07-24 13:06:49.000000000 +0200

+++ linux-2.4.20-18.9.acpi.4/include/linux/input.h 2003-07-28 08:14:48.000000000 +0200

@@ -7,6 +7,9 @@

\* Copyright (c) 1999-2000 Vojtech Pavlik

\*

\* Sponsored by SuSE

+ \*

+ \* 2003.07.29, Diego Santa Cruz <Diego.SantaCruz@epfl.ch>

+ \* Updated USB HID to event keycode mapping

\*/

/\*

@@ -305,11 +308,58 @@

#define KEY\_SUSPEND 205

#define KEY\_CLOSE 206

-#define KEY\_UNKNOWN 220

+/\* #define KEY\_UNKNOWN 220 \*/

#define KEY\_BRIGHTNESSDOWN 224

```
#define KEY_BRIGHTNESSUP 225

+/* Extra key definitions from kernel 2.6 */
+#define KEY_PLAY 207
+#define KEY_FASTFORWARD 208
+#define KEY_BASSBOOST 209
+#define KEY_PRINT 210
+#define KEY_HP 211
+#define KEY_CAMERA 212
+#define KEY_SOUND 213
+#define KEY_QUESTION 214
+#define KEY_EMAIL 215
+#define KEY_CHAT 216
+#define KEY_SEARCH 217
+#define KEY_CONNECT 218
+#define KEY_FINANCE 219
+#define KEY_SPORT 220
+#define KEY_SHOP 221
+#define KEY_ALTERASE 222
+#define KEY_CANCEL 223
+#define KEY_MEDIA 226
+
+/* Extra key definitions (private) */
+#define KEY_SELECT 199
+#define KEY_WORDPROCESSOR 227
+#define KEY_SPREADSHEET 228
+#define KEY_CALENDAR 229
+#define KEY_LANBROWSER 230
+#define KEY_LOGOFF 231
+#define KEY_NEXTTASK 232
+#define KEY_PREVTASK 233
+#define KEY_DOCUMENTS 234
+#define KEY_DESKTOP 235
+#define KEY_SPELLCHECK 236
+#define KEY_PICTURES 237
+#define KEY_MUSIC 238
+#define KEY_NEW 239
+#define KEY_SAVE 240
+#define KEY_HISTORY 241
+#define KEY_REPLY 242
+#define KEY_FORWARDMSG 243
+#define KEY_SEND 244
+#define KEY_ZOOMIN 245
+#define KEY_ZOOMOUT 246
+#define KEY_ZOOM 247
+#define KEY_SCROLL 248
+
+#define KEY_UNKNOWN 255
+
#define BTN_MISC 0x100
#define BTN_0 0x100
```

## Linux–Kernel: Linux problems with USB HID keyboards and patches

```
#define BTN_1 0x101
--- linux-2.4.20-18.9.acpi.4/drivers/usb/hid-input.c.orig 2003-07-21 15:09:26.000000000 +0200
+++ linux-2.4.20-18.9.acpi.4/drivers/usb/hid-input.c 2003-07-28 08:15:01.000000000 +0200
@@ -6,6 +6,9 @@
 * USB HID to Linux Input mapping module
 *
 * Sponsored by SuSE
+ *
+ * 2003.07.29, Diego Santa Cruz <Diego.SantaCruz@epfl.ch>
+ * Updated USB HID to event keycode mapping
 */

/*
@@ -37,25 +40,37 @@

#include "hid.h"

-#define unk KEY_UNKNOWN
-
static unsigned char hid_keyboard[256] = {
- 0, 0, 0, 0, 30, 48, 46, 32, 18, 33, 34, 35, 23, 36, 37, 38,
- 50, 49, 24, 25, 16, 19, 31, 20, 22, 47, 17, 45, 21, 44, 2, 3,
- 4, 5, 6, 7, 8, 9, 10, 11, 28, 1, 14, 15, 57, 12, 13, 26,
- 27, 43, 84, 39, 40, 41, 51, 52, 53, 58, 59, 60, 61, 62, 63, 64,
- 65, 66, 67, 68, 87, 88, 99, 70,119,110,102,104,111,107,109,106,
- 105,108,103, 69, 98, 55, 74, 78, 96, 79, 80, 81, 75, 76, 77, 71,
- 72, 73, 82, 83, 86,127,116,117, 85, 89, 90, 91, 92, 93, 94, 95,
- 120,121,122,123,134,138,130,132,128,129,131,137,133,135,136,113,
- 115,114,unk,unk,unk,124,unk,181,182,183,184,185,186,187,188,189,
- 190,191,192,193,194,195,196,197,198,unk,unk,unk,unk,unk,unk,unk,
- unk,unk,unk,unk,unk,unk,unk,unk,unk,unk,unk,unk,unk,unk,unk,
- unk,unk,unk,unk,unk,unk,unk,unk,unk,unk,unk,unk,unk,unk,unk,
- unk,unk,unk,unk,unk,unk,unk,unk,unk,unk,unk,unk,unk,unk,unk,
- unk,unk,unk,unk,unk,unk,unk,unk,unk,unk,unk,unk,unk,unk,unk,
- 29, 42, 56,125, 97, 54,100,126,164,166,165,163,161,115,114,113,
- 150,158,159,128,136,177,178,176,142,152,173,140,unk,unk,unk,unk
+ /* 0x00 */ KEY_RESERVED, KEY_RESERVED, KEY_RESERVED, KEY_RESERVED, KEY_A,
KEY_B, KEY_C, KEY_D,
+ /* 0x08 */ KEY_E, KEY_F, KEY_G, KEY_H, KEY_I, KEY_J, KEY_K, KEY_L,
+ /* 0x10 */ KEY_M, KEY_N, KEY_O, KEY_P, KEY_Q, KEY_R, KEY_S, KEY_T,
+ /* 0x18 */ KEY_U, KEY_V, KEY_W, KEY_X, KEY_Y, KEY_Z, KEY_1, KEY_2,
+ /* 0x20 */ KEY_3, KEY_4, KEY_5, KEY_6, KEY_7, KEY_8, KEY_9, KEY_0,
+ /* 0x28 */ KEY_ENTER, KEY_ESC, KEY_BACKSPACE, KEY_TAB, KEY_SPACE, KEY_MINUS,
KEY_EQUAL, KEY_LEFTBRACE,
+ /* 0x30 */ KEY_RIGHTBRACE, KEY_BACKSLASH, KEY_103RD, KEY_SEMICOLON,
KEY_APOSTROPHE, KEY_GRAVE, KEY_COMMA, KEY_DOT,
+ /* 0x38 */ KEY_SLASH, KEY_CAPSLOCK, KEY_F1, KEY_F2, KEY_F3, KEY_F4, KEY_F5, KEY_F6,
+ /* 0x40 */ KEY_F7, KEY_F8, KEY_F9, KEY_F10, KEY_F11, KEY_F12, KEY_SYSRQ,
KEY_SCROLLLOCK,
+ /* 0x48 */ KEY_PAUSE, KEY_INSERT, KEY_HOME, KEY_PAGEUP, KEY_DELETE, KEY_END,
KEY_PAGEDOWN, KEY_RIGHT,
```

## Linux–Kernel: Linux problems with USB HID keyboards and patches

```
+ /* 0x50 */ KEY_LEFT, KEY_DOWN, KEY_UP, KEY_NUMLOCK, KEY_KPSLASH,  
KEY_KPASTERISK, KEY_KPMINUS, KEY_KPPLUS,  
+ /* 0x58 */ KEY_KPENTER, KEY_KP1, KEY_KP2, KEY_KP3, KEY_KP4, KEY_KP5, KEY_KP6,  
KEY_KP7,  
+ /* 0x60 */ KEY_KP8, KEY_KP9, KEY_KP0, KEY_KPDOT, KEY_102ND, KEY_COMPOSE,  
KEY_RESERVED, KEY_KPEQUAL,  
+ /* 0x68 */ KEY_F13, KEY_F14, KEY_F15, KEY_F16, KEY_F17, KEY_F18, KEY_F19, KEY_F20,  
+ /* 0x70 */ KEY_F21, KEY_F22, KEY_F23, KEY_F24, KEY_MACRO, KEY_HELP, KEY_MENU,  
KEY_SELECT,  
+ /* 0x78 */ KEY_STOP, KEY_AGAIN, KEY_UNDO, KEY_CUT, KEY_COPY, KEY_PASTE,  
KEY_FIND, KEY_MUTE,  
+ /* 0x80 */ KEY_VOLUMEUP, KEY_VOLUMEDOWN, KEY_UNKNOWN, KEY_UNKNOWN,  
KEY_UNKNOWN, KEY_KPCOMMA, KEY_UNKNOWN, KEY_INTL1,  
+ /* 0x88 */ KEY_INTL2, KEY_INTL3, KEY_INTL4, KEY_INTL5, KEY_INTL6, KEY_INTL7,  
KEY_INTL8, KEY_INTL9,  
+ /* 0x90 */ KEY_LANG1, KEY_LANG2, KEY_LANG3, KEY_LANG4, KEY_LANG5, KEY_LANG6,  
KEY_LANG7, KEY_LANG8,  
+ /* 0x98 */ KEY_LANG9, KEY_ALTERASE, KEY_UNKNOWN, KEY_UNKNOWN,  
KEY_UNKNOWN, KEY_CANCEL, KEY_UNKNOWN, KEY_UNKNOWN,  
+ /* 0xA0 */ KEY_UNKNOWN, KEY_UNKNOWN, KEY_UNKNOWN, KEY_UNKNOWN,  
KEY_UNKNOWN, KEY_RESERVED, KEY_RESERVED, KEY_RESERVED,  
+ /* 0xA8 */ KEY_RESERVED, KEY_RESERVED, KEY_RESERVED, KEY_RESERVED,  
KEY_RESERVED, KEY_RESERVED, KEY_RESERVED, KEY_RESERVED,  
+ /* 0xB0 */ KEY_UNKNOWN, KEY_UNKNOWN, KEY_UNKNOWN, KEY_UNKNOWN,  
KEY_UNKNOWN, KEY_UNKNOWN, KEY_KPLEFTPAREN, KEY_KPRIGHTPAREN,  
+ /* 0xB8 */ KEY_UNKNOWN, KEY_UNKNOWN, KEY_UNKNOWN, KEY_UNKNOWN,  
KEY_UNKNOWN, KEY_UNKNOWN, KEY_UNKNOWN, KEY_UNKNOWN,  
+ /* 0xC0 */ KEY_UNKNOWN, KEY_UNKNOWN, KEY_UNKNOWN, KEY_UNKNOWN,  
KEY_UNKNOWN, KEY_UNKNOWN, KEY_UNKNOWN, KEY_UNKNOWN,  
+ /* 0xC8 */ KEY_UNKNOWN, KEY_UNKNOWN, KEY_UNKNOWN, KEY_UNKNOWN,  
KEY_UNKNOWN, KEY_UNKNOWN, KEY_UNKNOWN, KEY_UNKNOWN,  
+ /* 0xD0 */ KEY_UNKNOWN, KEY_UNKNOWN, KEY_UNKNOWN, KEY_UNKNOWN,  
KEY_UNKNOWN, KEY_UNKNOWN, KEY_UNKNOWN, KEY_KPPLUSMINUS,  
+ /* 0xD8 */ KEY_UNKNOWN, KEY_UNKNOWN, KEY_UNKNOWN, KEY_UNKNOWN,  
KEY_UNKNOWN, KEY_UNKNOWN, KEY_RESERVED, KEY_RESERVED,  
+ /* 0xE0 */ KEY_LEFTCTRL, KEY_LEFTSHIFT, KEY_LEFTALT, KEY_LEFTMETA,  
KEY_RIGHTCTRL, KEY_RIGHTSHIFT, KEY_RIGHTALT, KEY_RIGHTMETA  
+ /* 0xE8–0xFFFF */ /* reserved (KEY_RESERVED == 0) */  
};
```

```
static struct {  
@@ -200,11 +215,13 @@  
  
        switch (usage->hid & HID_USAGE) {  
            case 0x000: usage->code = 0; break;  
- case 0x034: usage->code = KEY_SLEEP; break;  
- case 0x036: usage->code = BTN_MISC; break;  
+  
            case 0x08a: usage->code = KEY_WWW; break;  
- case 0x095: usage->code = KEY_HELP; break;
```

## Linux–Kernel: Linux problems with USB HID keyboards and patches

```
+ case 0x0b0: usage->code = KEY_PLAYCD; break;
+ case 0x0b1: usage->code = KEY_PAUSECD; break;
+ case 0x0b2: usage->code = KEY_RECORD; break;
+ case 0x0b3: usage->code = KEY_FASTFORWARD; break;
      case 0x0b4: usage->code = KEY_REWIND; break;
      case 0x0b5: usage->code = KEY_NEXTSONG; break;
      case 0x0b6: usage->code = KEY_PREVIOUSSONG; break;
@@ -213,26 +230,78 @@
      case 0x0cd: usage->code = KEY_PLAYPAUSE; break;

      case 0x0e2: usage->code = KEY_MUTE; break;
+ case 0x0e5: usage->code = KEY_BASSBOOST; break;
      case 0x0e9: usage->code = KEY_VOLUMEUP; break;
      case 0x0ea: usage->code = KEY_VOLUMEDOWN; break;

      case 0x183: usage->code = KEY_CONFIG; break;
+ case 0x184: usage->code = KEY_WORDPROCESSOR; break;
+ case 0x186: usage->code = KEY_SPREADSHEET; break;
      case 0x18a: usage->code = KEY_MAIL; break;
+ case 0x18e: usage->code = KEY_CALENDAR; break;
+ case 0x191: usage->code = KEY_FINANCE; break;
      case 0x192: usage->code = KEY_CALC; break;
- case 0x194: usage->code = KEY_FILE; break;
+ case 0x194: usage->code = KEY_COMPUTER; break;
+ case 0x195: usage->code = KEY_LANBROWSER; break;
+ case 0x196: usage->code = KEY_WWW; break;
+ case 0x197: usage->code = KEY_CONNECT; break;
+ case 0x198: usage->code = KEY_CAMERA; break;
+ case 0x199: usage->code = KEY_CHAT; break;
+ case 0x19a: usage->code = KEY_PHONE; break;
+ case 0x19c: usage->code = KEY_LOGOFF; break;
+ case 0x19f: usage->code = KEY_SETUP; break;
+ case 0x1a0: usage->code = KEY_MSDOS; break;
+
+ case 0x1a2: usage->code = KEY_CYCLEWINDOWS; break;
+ case 0x1a3: usage->code = KEY_NEXTTASK; break;
+ case 0x1a4: usage->code = KEY_PREVTASK; break;
+
+ case 0x1a7: usage->code = KEY_DOCUMENTS; break;
+ case 0x1aa: usage->code = KEY_DESKTOP; break;
+ case 0x1ab: usage->code = KEY_SPELLCHECK; break;
+ case 0x1b4: usage->code = KEY_FILE; break;
+ case 0x1b6: usage->code = KEY_PICTURES; break;
+ case 0x1b7: usage->code = KEY_MUSIC; break;
+
+ case 0x201: usage->code = KEY_NEW; break;
+ case 0x202: usage->code = KEY_OPEN; break;
+ case 0x203: usage->code = KEY_CLOSE; break;
+ case 0x204: usage->code = KEY_EXIT; break;
+ case 0x207: usage->code = KEY_SAVE; break;
+ case 0x208: usage->code = KEY_PRINT; break;
```

## Linux–Kernel: Linux problems with USB HID keyboards and patches

```
+ case 0x209: usage->code = KEY_PROPS; break;

        case 0x21a: usage->code = KEY_UNDO; break;
        case 0x21b: usage->code = KEY_COPY; break;
        case 0x21c: usage->code = KEY_CUT; break;
        case 0x21d: usage->code = KEY_PASTE; break;
+ case 0x21f: usage->code = KEY_FIND; break;

- case 0x221: usage->code = KEY_FIND; break;
+ case 0x221: usage->code = KEY_SEARCH; break;
        case 0x223: usage->code = KEY_HOMEPAGE; break;
        case 0x224: usage->code = KEY_BACK; break;
        case 0x225: usage->code = KEY_FORWARD; break;
        case 0x226: usage->code = KEY_STOP; break;
        case 0x227: usage->code = KEY_REFRESH; break;
        case 0x22a: usage->code = KEY_BOOKMARKS; break;
+ case 0x22b: usage->code = KEY_HISTORY; break;
+
+ case 0x22d: usage->code = KEY_ZOOMIN; break;
+ case 0x22e: usage->code = KEY_ZOOMOUT; break;
+ case 0x22f: usage->code = KEY_ZOOM; break;
+ case 0x233: usage->code = KEY_SCROLLUP; break;
+ case 0x234: usage->code = KEY_SCROLLDOWN; break;
+ case 0x235: usage->code = KEY_SCROLL; break;
+
+ case 0x260: usage->code = KEY_SHOP; break;
+ case 0x279: usage->code = KEY_AGAIN; break;
+
+ case 0x289: usage->code = KEY_REPLY; break;
+ case 0x28b: usage->code = KEY_FORWARDMSG; break;
+ case 0x28c: usage->code = KEY_SEND; break;
+
+ case 0x28e: usage->code = KEY_SENDFILE; break;
+ case 0x28f: usage->code = KEY_XFER; break;

        default: usage->code = KEY_UNKNOWN; break;

--- linux-2.4.20-18.9.acpi.4/drivers/input/keybdev.c.orig 2003-07-25 19:15:55.000000000 +0200
+++ linux-2.4.20-18.9.acpi.4/drivers/input/keybdev.c 2003-07-28 08:11:30.000000000 +0200
@@ -6,6 +6,10 @@
 * Input driver to keyboard driver binding.
 *
 * Sponsored by SuSE
+ *
+ * 2003.07.29, Diego Santa Cruz <Diego.SantaCruz@epfl.ch>
+ * Use MS std. USB HID to PS/2 scancode mappings, supplemented
+ * with MS Office keyboard and custom mappings for other keys.
 */

/*
@@ -50,22 +54,184 @@
```

```

static int jp_kbd_109 = 1; /* Yes, .jp is the default. See 51142. */

-static unsigned short x86_keycodes[256] =
- { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
- 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31,
- 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47,
- 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63,
- 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79,
- 80, 81, 82, 83, 43, 85, 86, 87, 88, 115, 119, 120, 121, 375, 123, 90,
- 284, 285, 309, 298, 312, 91, 327, 328, 329, 331, 333, 335, 336, 337, 338, 339,
- 367, 294, 293, 286, 350, 92, 334, 512, 116, 377, 109, 111, 373, 347, 348, 349,
- 360, 93, 94, 95, 98, 376, 100, 101, 357, 316, 354, 304, 289, 102, 351, 355,
- 103, 104, 105, 275, 281, 272, 306, 106, 274, 107, 288, 364, 358, 363, 362, 361,
- 291, 108, 381, 290, 287, 292, 279, 305, 280, 99, 112, 257, 258, 113, 270, 114,
- 118, 117, 125, 374, 379, 259, 260, 261, 262, 263, 264, 265, 266, 267, 268, 269,
- 271, 273, 276, 277, 278, 282, 283, 295, 296, 297, 299, 300, 301, 302, 303, 307,
- 308, 310, 313, 314, 315, 317, 318, 319, 320, 321, 322, 323, 324, 325, 326, 330,
- 332, 340, 341, 342, 343, 344, 345, 346, 356, 359, 365, 368, 369, 370, 371, 372 };
+/* Based on Microsoft HID to PS/2 scancode mapping from
+ * <http://www.microsoft.com/hwdev/download/tech/input/translate.pdf>
+ * with addition of MS Office keyboard scancodes from
+ * <http://www.chipx86.com/linuxstuff/officekb/> (X scancodes converted
+ * to PS/2 scancodes using XFree86 source code).
+ *
+ * Exceptions:
+ *
+ * LANG1 72 (MS says f2 for make, no code for break)
+ * LANG2 71 (MS says f1 for make, no code for break)
+ * SYSRQ special handling in emulate_raw()
+ * PAUSE special handling in emulate_raw()
+ *
+ * Additions:
+ *
+ * F16 e0 11
+ * F17 e0 12
+ * F18 e0 77
+ * F19 e0 1f
+ * F20 5a
+ * MACRO e0 6f
+ * KPPLUSMINUS e0 4e
+ * F21 74
+ * F22 67
+ * F23 6d
+ * F24 6f
+ * FRONT 5b
+ * HELP e0 3b (MS Office keyb.)
+ * MENU e0 4c
+ * DELETEFILE 69
+ * COFFEE e0 36
+ * DIRECTION 6b

```

## Linux–Kernel: Linux problems with USB HID keyboards and patches

- + \* CLOSECD e0 59
- + \* EJECTCLOSECD e0 5a
- + \* ISO 68
- + \* MOVE 72
- + \* EDIT 6a
- + \* KPLEFTPAREN e0 76
- + \* KPRIGHTPAREN e0 62
- + \* INTL7 62
- + \* INTL8 63
- + \* INTL9 64
- + \* LANG6 65
- + \* LANG7 66
- + \* LANG8 6c
- + \* LANG9 6e
- + \* SUSPEND e0 6e
- + \* HP e0 3c (Office Home in MS Office keyb.)
- + \* SOUND 55
- + \* QUESTION e0 3b (same as HELP)
- + \* SPORT e0 2a
- + \* ALTERASE e0 79
- + \* CANCEL 71
- + \* BRIGHTNESSDOWN e0 74
- + \* BRIGHTNESSUP e0 75
- + \* SELECT 75
- + \*
- + \* PROG1 e0 70
- + \* PROG2 e0 71
- + \* PROG3 e0 72
- + \* PROG4 e0 73
- + \*
- + \* UNKNOWN e0 79 (same as ALTERASE)
- + \*
- + \* Special keys associated to HID functions in consumer page (0x0c):
- + \*
- + \* KEY HID name HID code Scancode
- + \*
- + \* MUTE Mute e2 e0 20 (MS std.)
- + \* VOLUMEDOWN Volume Decrement ea e0 2e (MS std.)
- + \* VOLUMEUP Volume Increment e9 e0 30 (MS std.)
- + \* STOP AC Stop 226 e0 68 (MS std.)
- + \* AGAIN AC Redo/Repeat 279 e0 07 (MS Office keyb.)
- + \* PROPS AC Properties 209 e0 56
- + \* UNDO AC Undo 21a e0 08 (MS Office keyb.)
- + \* COPY AC Copy 21b e0 18 (MS Office keyb.)
- + \* OPEN AC Open 202 e0 3f (MS Office keyb.)
- + \* PASTE AC Paste 21d e0 0a (MS Office keyb.)
- + \* FIND AC Find 21f e0 78
- + \* CUT AC Cut 21c e0 17 (MS Office keyb.)
- + \* CALC AL Calculator 192 e0 21 (MS std.)
- + \* SETUP AL Control Panel 19f e0 55
- + \* FILE AL File Browser 1b4 e0 05 (MS Office keyb.)

## Linux–Kernel: Linux problems with USB HID keyboards and patches

- + \* SENDFILE AC Upload 28e e0 01
- + \* XFER AC Download 28f e0 02
- + \* WWW AL Internet Browser 196 e0 03
- + \* MSDOS AL Command line processor 1a0 e0 04
- + \* CYCLEWINDOWS AL Select Task / Application 1a4 e0 3d (MS Office keyb.)
- + \* MAIL / EMAIL AL Email Reader 18a e0 6c (MS std.)
- + \* BOOKMARKS AC Bookmarks 22a e0 66 (MS std.)
- + \* COMPUTER AL Local Machine Browser 194 e0 6b (MS std.)
- + \* BACK AC Back 224 e0 6a (MS std.)
- + \* FORWARD AC Forward 225 e0 69 (MS std.)
- + \* EJECTCD Eject b8 e0 29
- + \* NEXTSONG Scan Next Track b5 e0 19 (MS std.)
- + \* PLAYPAUSE Play/Pause cd e0 22 (MS std.)
- + \* PREVIOUSSONG Scan Previous Track b6 e0 10 (MS std.)
- + \* STOPCD Stop b7 e0 24 (MS std.)
- + \* RECORD Record b2 e0 2b
- + \* REWIND Rewind b4 e0 2c
- + \* PHONE AL Telephony/Dialer 19a e0 54
- + \* CONFIG / MEDIA AL Consumer Control Conf. 183 e0 6d (MS std.)
- + \* HOMEPAGE AC Home 223 e0 32 (MS std.)
- + \* REFRESH AC Refresh 227 e0 67 (MS std.)
- + \* EXIT AC Exit 204 e0 34
- + \* SCROLLUP AC Scroll Up 233 e0 39
- + \* SCROLLDOWN AC Scroll Down 234 e0 3a
- + \* PLAYCD / PLAY Play b0 e0 2d
- + \* PAUSECD Pause b1 e0 2f
- + \* CLOSE AC Close 203 e0 40 (MS Office keyb.)
- + \* FASTFORWARD Fast Forward b3 e0 31
- + \* BASSBOOST Bass Boost e5 e0 33
- + \* PRINT AC Print 208 e0 58 (MS Office keyb.)
- + \* CAMERA AL Network Conference 198 e0 06
- + \* CHAT AL Network Chat 199 e0 28
- + \* SEARCH AC Search 221 e0 65 (MS std.)
- + \* CONNECT AL ISP Connect 197 e0 25
- + \* FINANCE AL Checkbook/Finance 191 e0 26
- + \* SHOP AC Catalog 260 e0 27
- + \* WORDPROCESSOR AL Word Processor 184 e0 13 (MS Office keyb.)
- + \* SPREADSHEET AL Spreadsheet 186 e0 14 (MS Office keyb.)
- + \* CALENDAR AL Calendar/Schedule 18e e0 15 (MS Office keyb.)
- + \* LANBROWSER AL LAN/WAN Browser 195 e0 1b
- + \* LOGOFF AL Logoff 19c e0 16 (MS Office keyb.)
- + \* NEXTTASK AL Next Task/Application 1a3 e0 1e (MS Office keyb.)
- + \* PREVTASK AL Previous Task/Application 1a4 e0 09 (MS Office keyb.)
- + \* DOCUMENTS AL Documents 1a7 e0 0c
- + \* DESKTOP AL Desktop 1aa e0 0d
- + \* SPELLCHECK AL Spell Check 1ab e0 23 (MS Office keyb.)
- + \* PICTURES AL Pictures (non–std.) 1b6 e0 0e
- + \* MUSIC AL Music (non–std.) 1b7 e0 0f
- + \* NEW AC New 201 e0 3e (MS Office keyb.)
- + \* SAVE AC Save 207 e0 57 (MS Office keyb.)
- + \* HISTORY AC History 22b e0 1a

## Linux–Kernel: Linux problems with USB HID keyboards and patches

```
+ * REPLY AC Reply 289 e0 41 (MS Office keyb.)
+ * FORWARDMSG AC Forward Msg 28b e0 42 (MS Office keyb.)
+ * SEND AC Send 28c e0 43 (MS Office keyb.)
+ * ZOOMIN AC Zoom in 22d e0 44
+ * ZOOMOUT AC Zoom out 22e e0 45
+ * ZOOM AC Zoom 22f e0 4a
+ * SCROLL AC Scroll 235 e0 0b (MS Office keyb.)
+ *
+ **/
+
+static unsigned short x86_keycodes[256] = {
+ /* 0x00 */ 0x000, 0x001, 0x002, 0x003, 0x004, 0x005, 0x006, 0x007,
+ /* 0x08 */ 0x008, 0x009, 0x00A, 0x00B, 0x00C, 0x00D, 0x00E, 0x00F,
+ /* 0x10 */ 0x010, 0x011, 0x012, 0x013, 0x014, 0x015, 0x016, 0x017,
+ /* 0x18 */ 0x018, 0x019, 0x01A, 0x01B, 0x01C, 0x01D, 0x01E, 0x01F,
+ /* 0x20 */ 0x020, 0x021, 0x022, 0x023, 0x024, 0x025, 0x026, 0x027,
+ /* 0x28 */ 0x028, 0x029, 0x02A, 0x02B, 0x02C, 0x02D, 0x02E, 0x02F,
+ /* 0x30 */ 0x030, 0x031, 0x032, 0x033, 0x034, 0x035, 0x036, 0x037,
+ /* 0x38 */ 0x038, 0x039, 0x03A, 0x03B, 0x03C, 0x03D, 0x03E, 0x03F,
+ /* 0x40 */ 0x040, 0x041, 0x042, 0x043, 0x044, 0x045, 0x046, 0x047,
+ /* 0x48 */ 0x048, 0x049, 0x04A, 0x04B, 0x04C, 0x04D, 0x04E, 0x04F,
+ /* 0x50 */ 0x050, 0x051, 0x052, 0x053, 0x054, 0x055, 0x056, 0x057,
+ /* 0x58 */ 0x058, 0x059, 0x05A, 0x05B, 0x05C, 0x05D, 0x05E, 0x05F,
+ /* 0x60 */ 0x060, 0x061, 0x062, 0x063, 0x064, 0x065, 0x066, 0x067,
+ /* 0x68 */ 0x068, 0x069, 0x06A, 0x06B, 0x06C, 0x06D, 0x06E, 0x06F,
+ /* 0x70 */ 0x070, 0x071, 0x072, 0x073, 0x074, 0x075, 0x076, 0x077,
+ /* 0x78 */ 0x078, 0x079, 0x07A, 0x07B, 0x07C, 0x07D, 0x07E, 0x07F,
+ /* 0x80 */ 0x080, 0x081, 0x082, 0x083, 0x084, 0x085, 0x086, 0x087,
+ /* 0x88 */ 0x088, 0x089, 0x08A, 0x08B, 0x08C, 0x08D, 0x08E, 0x08F,
+ /* 0x90 */ 0x090, 0x091, 0x092, 0x093, 0x094, 0x095, 0x096, 0x097,
+ /* 0x98 */ 0x098, 0x099, 0x09A, 0x09B, 0x09C, 0x09D, 0x09E, 0x09F,
+ /* 0xA0 */ 0x0A0, 0x0A1, 0x0A2, 0x0A3, 0x0A4, 0x0A5, 0x0A6, 0x0A7,
+ /* 0xA8 */ 0x0A8, 0x0A9, 0x0AA, 0x0AB, 0x0AC, 0x0AD, 0x0AE, 0x0AF,
+ /* 0xB0 */ 0x0B0, 0x0B1, 0x0B2, 0x0B3, 0x0B4, 0x0B5, 0x0B6, 0x0B7,
+ /* 0xB8 */ 0x0B8, 0x0B9, 0x0BA, 0x0BB, 0x0BC, 0x0BD, 0x0BE, 0x0BF,
+ /* 0xC0 */ 0x0C0, 0x0C1, 0x0C2, 0x0C3, 0x0C4, 0x0C5, 0x0C6, 0x0C7,
+ /* 0xC8 */ 0x0C8, 0x0C9, 0x0CA, 0x0CB, 0x0CC, 0x0CD, 0x0CE, 0x0CF,
+ /* 0xD0 */ 0x0D0, 0x0D1, 0x0D2, 0x0D3, 0x0D4, 0x0D5, 0x0D6, 0x0D7,
+ /* 0xD8 */ 0x0D8, 0x0D9, 0x0DA, 0x0DB, 0x0DC, 0x0DD, 0x0DE, 0x0DF,
+ /* 0xE0 */ 0x0E0, 0x0E1, 0x0E2, 0x0E3, 0x0E4, 0x0E5, 0x0E6, 0x0E7,
+ /* 0xE8 */ 0x0E8, 0x0E9, 0x0EA, 0x0EB, 0x0EC, 0x0ED, 0x0EE, 0x0EF,
+ /* 0xF0 */ 0x0F0, 0x0F1, 0x0F2, 0x0F3, 0x0F4, 0x0F5, 0x0F6, 0x0F7,
+ /* 0xF8 */ 0x0F8, 0x0F9, 0x0FA, 0x0FB, 0x0FC, 0x0FD, 0x0FE, 0x0FF,
+};

#ifdef CONFIG_MAC_EMUMOUSEBTN
extern int mac_hid_mouse_emulate_buttons(int, int, int);
@@ -225,13 +391,8 @@
    kbd_ledfunc = keybdev_ledfunc;
    kbd_refresh_leds();
#endif
```

## Linux-Kernel: Linux problems with USB HID keyboards and patches

```
- if (jp_kbd_109) {
- x86_keycodes[0xb5] = 0x73; /* backslash, underscore */
- x86_keycodes[0xb6] = 0x70;
- x86_keycodes[0xb7] = 0x7d; /* Yen, pipe */
- x86_keycodes[0xb8] = 0x79;
- x86_keycodes[0xb9] = 0x7b;
- }
+ if (!jp_kbd_109)
+ printk(KERN_WARNING "keybdev: legacy jp_kbd_109 option = 0 ignored (jp_kbd_109 always
enabled)\n");

    return 0;
}
```

D Do not map keyboard buttons to PS/2 mouse buttons but to

D programmable keys.

D

```
--- linux-2.4.20-18.9.acpi.4/drivers/input/keybdev.c.orig 2003-07-28 08:11:30.000000000 +0200
+++ linux-2.4.20-18.9.acpi.4/drivers/input/keybdev.c 2003-07-28 08:11:21.000000000 +0200
@@ -337,6+337,13 @@
{
```

```
    if (type != EV_KEY) return;
```

```
+ switch (code) { /* map keyboard buttons to programmable keys */
```

```
+ case BTN_0: code = KEY_PROG1; break;
```

```
+ case BTN_1: code = KEY_PROG2; break;
```

```
+ case BTN_2: code = KEY_PROG3; break;
```

```
+ case BTN_3: code = KEY_PROG4; break;
```

```
+ }
```

```
+
```

```
    if (emulate_raw(code, down))
```

```
        printk(KERN_WARNING "keyboard.c: can't emulate rawmode for keycode %d\n", code);
```

```
--- linux-2.4.20-18.9.acpi.4/drivers/input/mousedev.c.orig 2003-07-25 19:15:36.000000000 +0200
```

```
+++ linux-2.4.20-18.9.acpi.4/drivers/input/mousedev.c 2003-07-25 19:17:42.000000000 +0200
```

```
@@ -122,17+122,12 @@
```

```
    case EV_KEY:
```

```
        switch (code) {
```

```
- case BTN_0:
```

```
            case BTN_TOUCH:
```

```
            case BTN_LEFT: index = 0; break;
```

```
- case BTN_4:
```

```
            case BTN_EXTRA: if (list->mode == 2) { index = 4; break; }
```

```
            case BTN_STYLUS:
```

```
- case BTN_1:
```

```
            case BTN_RIGHT: index = 1; break;
```

```
- case BTN_3:
```

```
            case BTN_SIDE: if (list->mode == 2) { index = 3; break; }
```

## Linux-Kernel: Linux problems with USB HID keyboards and patches

– case BTN\_2:

```
case BTN_STYLUS2:  
case BTN_MIDDLE: index = 2; break;  
default: return;
```

–

To unsubscribe from this list: send the line "unsubscribe linux-kernel" in the body of a message to majordomo@vger.kernel.org

More majordomo info at <http://vger.kernel.org/majordomo-info.html>

Please read the FAQ at <http://www.tux.org/lkml/>