

```

/*
 * $Id: leds.c,v 1.1 2000/08/17 18:56:37 vojtech Exp $
 *
 * Copyright (c) 1999-2000 Vojtech Pavlik
 *
 * Leds'n'buttons test program
 */

/*
 * This program is free software; you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software
 * Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
 *
 * Should you need to contact me, the author, you can do so either by
 * e-mail - mail your message to <vojtech@ucw.cz>, or by paper mail:
 * Vojtech Pavlik, Ucitelska 1576, Prague 8, 182 00 Czech Republic
 */

#include <linux/input.h>

#include <string.h>
#include <fcntl.h>
#include <unistd.h>
#include <stdio.h>

int main (int argc, char **argv)
{
    int fd, rd, i;
    struct input_event ev[64];
    int version;
    unsigned short id[4];
    char name[256] = "Unknown";

    if (argc < 2) {
        printf ("Usage: leds /dev/inputX\n");
        printf ("Where X = input device number\n");
        exit (1);
    }

    if ((fd = open(argv[argc - 1], O_RDWR)) < 0) {
        perror("leds");
        exit(1);
    }

    ioctl(fd, EVIOCGVERSION, &version);
    printf("Input driver version is %d.%d.%d\n",
           version >> 16, (version >> 8) & 0xff, version & 0xff);

    ioctl(fd, EVIOCGID, id);
    printf("Input device ID: bus 0x%x vendor 0x%x product 0x%x version 0x%x\n",
           id[ID_BUS], id[ID_VENDOR], id[ID_PRODUCT], id[ID_VERSION]);

    ioctl(fd, EVIOCGNAME(sizeof(name)), name);
    printf("Input device name: \"%s\"\n", name);

    if (id[ID_BUS] != BUS_USB || id[ID_VENDOR] != 0x4242 || id[ID_PRODUCT] != 0x4201) {
        fprintf(stderr, "leds: %s is not a Leds & Buttons device\n", argv[argc - 1]);
        return -1;
    }

    printf("Testing ... (interrupt to exit)\n");

    while (1) {
        rd = read(fd, ev, sizeof(struct input_event) * 64);

        if (rd < sizeof(struct input_event)) {
            perror("leds: error reading");
            exit (1);
        }
    }
}

```

```
for (i = 0; i < rd / sizeof(struct input_event); i++) {  
    printf("%s %d: %s\n",  
        ev[i].type == EV_KEY ? "Button" : "LED  ",  
        ev[i].code - (ev[i].type == EV_KEY ? BTN_MISC : LED_MISC),  
        ev[i].value ? "on" : "off");  
  
    if (ev[i].type == EV_KEY) {  
        ev[i].type = EV_LED;  
        ev[i].code += LED_MISC - BTN_MISC;  
  
        write(fd, ev + i, sizeof(struct input_event));  
    }  
}  
}
```