Evmapy

Evmapy is a tool used by Batocera to map inputs from the controller to a virtual keyboard (although, its capable of quite a bit more than that). Refer to evmapy's readme for more information. This is primarily used to enable the use of the emulator's shortcuts if it does not natively support hotkey binding onto the controller's buttons (for example, Dolphin does not use evmapy).

The emulator's "keys" file, if used, is located at `package/batocera/<emulators or ports>/<emulator or port>/<system shortname>.<emulator>.keys`. This file must then be copied from `$(BR2_EXTERNAL_BATOCERA_PATH)/package/batocera/emulators/<emulator>/<system shortname>.<emulator>.keys` to `$(TARGET_DIR)/usr/share/evmapy` in its makefile. For example, with Citra (the 3DS emulator):

citra.mk

```
define CITRA_EVMAP
    # Create the directory if it doesn't already exist.
    mkdir -p $(TARGET_DIR)/usr/share/evmapy
    # Copy, while making all the parent folders, the keys file to the
correct location.
    cp -prn
$(BR2_EXTERNAL_BATOCERA_PATH)/package/batocera/emulators/citra/3ds.citr
a.keys \
        $(TARGET_DIR)/usr/share/evmapy
endef

CITRA_POST_INSTALL_TARGET_HOOKS = CITRA_EVMAP
```

Evmapy uses a simple syntax for its events, and indentation/spacing is not enforced (but it's still recommended for readability). First, everything must be inside of a curly bracket:

```
{
```

then the controller defined. In Batocera, defining this as "`actions_player1`" will automatically use whichever controller is currently assigned to player 1. Then, its triggers and actions are contained within a square bracket:

```
    "actions_player1": [
```

From there on, every trigger and action is contained inside of another curly bracket. For instance:

```
        {
            "trigger": ["hotkey", "start"],
            "type": "key",
            "target": ["KEY_LEFTCTRL", "KEY_Q"],
            "description": "Exit emulator"
```

```
        },
```

and additional trigger and actions are separated by commas, like a list, until the last one is reached and the remaining brackets are closed off:

```
        {
            "trigger": "r3",
            "type": "key",
            "target": "BTN_LEFT",
            "description": "Mouse Click"
        }
    ]
}
```

Input triggers

Batocera uses `es_input`'s generic terms to refer to its controller's inputs. Though not everything is intuitive, is a list of the human-readable inputs against the "`trigger`" code they use in the keys file:

Input	**Trigger name**
[HOTKEY]	hotkey
[START]	start
[SELECT]	select
	b
	a
	y
	x
D-pad Down	down
D-pad Right	right
D-pad Up	up
D-pad Left	left
All left analog stick movement	joystick1
All right analog stick movement	joystick2
LS Down	joystick1down
LS Right	joystick1right
LS Up	joystick1up
LS Left	joystick1left
RS Down	joystick2down
RS Right	joystick2right
RS Up	joystick2up
RS Left	joystick2left
[L1]	pageup
[R1]	pagedown
[L2]	l2
[R2]	r2
[L3] (LS click-in)	l3

Input	Trigger name
[R3] (RS click-in)	r3

This is but the tip of the iceberg of events that evmapy can parse. Refer to its documentation for more details.

Output events

Keys

The bread and butter of events evmapy will be sending out. Type in the name of the key desired to be pressed. If multiple keys are needed, separate them with commas in a list.

The most common modifier keys: KEY_LEFTCTRL, KEY_LEFTSHIFT, KEY_LEFTALT, KEY_RIGHTSHIFT, KEY_ENTER, KEY_PAGEDOWN, etc.

The most common exit shortcuts are [“KEY_LEFTALT”, “KEY_F4”] and [“KEY_LEFTCTRL”, “KEY_Q”].

Here's a sample of the /usr/include/linux/input-event-codes.h file from a standard installation of Solus. This should cover all the possible output events that can be handled by evmapy.

Click to reveal

```
/* SPDX-License-Identifier: GPL-2.0-only WITH Linux-syscall-note */
/*
 * Input event codes
 *
 *    *** IMPORTANT ***
 * This file is not only included from C-code but also from devicetree
source
 * files. As such this file MUST only contain comments and defines.
 *
 * Copyright (c) 1999-2002 Vojtech Pavlik
 * Copyright (c) 2015 Hans de Goede <hdegoede@redhat.com>
 *
 * This program is free software; you can redistribute it and/or modify
it
 * under the terms of the GNU General Public License version 2 as
published by
 * the Free Software Foundation.
 */
#ifndef _INPUT_EVENT_CODES_H
#define _INPUT_EVENT_CODES_H

/*
 * Device properties and quirks
 */
```

```
#define INPUT_PROP_POINTER      0x00    /* needs a pointer */
#define INPUT_PROP_DIRECT       0x01    /* direct input devices */
#define INPUT_PROP_BUTTONPAD        0x02    /* has button(s) under pad
*/
#define INPUT_PROP_SEMI_MT      0x03    /* touch rectangle only */
#define INPUT_PROP_TOPBUTTONPAD     0x04    /* softbuttons at top of
pad */
#define INPUT_PROP_POINTING_STICK   0x05    /* is a pointing stick */
#define INPUT_PROP_ACCELEROMETER    0x06    /* has accelerometer */

#define INPUT_PROP_MAX          0x1f
#define INPUT_PROP_CNT          (INPUT_PROP_MAX + 1)

/*
 * Event types
 */

#define EV_SYN          0x00
#define EV_KEY          0x01
#define EV_REL          0x02
#define EV_ABS          0x03
#define EV_MSC          0x04
#define EV_SW           0x05
#define EV_LED          0x11
#define EV_SND          0x12
#define EV_REP          0x14
#define EV_FF           0x15
#define EV_PWR          0x16
#define EV_FF_STATUS        0x17
#define EV_MAX          0x1f
#define EV_CNT          (EV_MAX+1)

/*
 * Synchronization events.
 */

#define SYN_REPORT      0
#define SYN_CONFIG      1
#define SYN_MT_REPORT       2
#define SYN_DROPPED     3
#define SYN_MAX         0xf
#define SYN_CNT         (SYN_MAX+1)

/*
 * Keys and buttons
 *
 * Most of the keys/buttons are modeled after USB HUT 1.12
 * (see http://www.usb.org/developers/hidpage).
 * Abbreviations in the comments:
 * AC - Application Control
 * AL - Application Launch Button
```

```
 * SC - System Control
 */

#define KEY_RESERVED        0
#define KEY_ESC         1
#define KEY_1           2
#define KEY_2           3
#define KEY_3           4
#define KEY_4           5
#define KEY_5           6
#define KEY_6           7
#define KEY_7           8
#define KEY_8           9
#define KEY_9           10
#define KEY_0           11
#define KEY_MINUS       12
#define KEY_EQUAL       13
#define KEY_BACKSPACE       14
#define KEY_TAB         15
#define KEY_Q           16
#define KEY_W           17
#define KEY_E           18
#define KEY_R           19
#define KEY_T           20
#define KEY_Y           21
#define KEY_U           22
#define KEY_I           23
#define KEY_O           24
#define KEY_P           25
#define KEY_LEFTBRACE       26
#define KEY_RIGHTBRACE      27
#define KEY_ENTER       28
#define KEY_LEFTCTRL        29
#define KEY_A           30
#define KEY_S           31
#define KEY_D           32
#define KEY_F           33
#define KEY_G           34
#define KEY_H           35
#define KEY_J           36
#define KEY_K           37
#define KEY_L           38
#define KEY_SEMICOLON       39
#define KEY_APOSTROPHE      40
#define KEY_GRAVE       41
#define KEY_LEFTSHIFT       42
#define KEY_BACKSLASH       43
#define KEY_Z           44
#define KEY_X           45
#define KEY_C           46
#define KEY_V           47
```

```
#define KEY_B           48
#define KEY_N           49
#define KEY_M           50
#define KEY_COMMA       51
#define KEY_DOT         52
#define KEY_SLASH       53
#define KEY_RIGHTSHIFT      54
#define KEY_KPASTERISK      55
#define KEY_LEFTALT     56
#define KEY_SPACE       57
#define KEY_CAPSLOCK        58
#define KEY_F1          59
#define KEY_F2          60
#define KEY_F3          61
#define KEY_F4          62
#define KEY_F5          63
#define KEY_F6          64
#define KEY_F7          65
#define KEY_F8          66
#define KEY_F9          67
#define KEY_F10         68
#define KEY_NUMLOCK     69
#define KEY_SCROLLLOCK      70
#define KEY_KP7         71
#define KEY_KP8         72
#define KEY_KP9         73
#define KEY_KPMINUS     74
#define KEY_KP4         75
#define KEY_KP5         76
#define KEY_KP6         77
#define KEY_KPPLUS      78
#define KEY_KP1         79
#define KEY_KP2         80
#define KEY_KP3         81
#define KEY_KP0         82
#define KEY_KPDOT       83

#define KEY_ZENKAKUHANKAKU  85
#define KEY_102ND       86
#define KEY_F11         87
#define KEY_F12         88
#define KEY_RO          89
#define KEY_KATAKANA        90
#define KEY_HIRAGANA        91
#define KEY_HENKAN      92
#define KEY_KATAKANAHIRAGANA    93
#define KEY_MUHENKAN        94
#define KEY_KPJPCOMMA       95
#define KEY_KPENTER     96
#define KEY_RIGHTCTRL       97
#define KEY_KPSLASH     98
```

```
#define KEY_SYSRQ       99
#define KEY_RIGHTALT        100
#define KEY_LINEFEED        101
#define KEY_HOME        102
#define KEY_UP          103
#define KEY_PAGEUP      104
#define KEY_LEFT        105
#define KEY_RIGHT       106
#define KEY_END         107
#define KEY_DOWN        108
#define KEY_PAGEDOWN        109
#define KEY_INSERT      110
#define KEY_DELETE      111
#define KEY_MACRO       112
#define KEY_MUTE        113
#define KEY_VOLUMEDOWN      114
#define KEY_VOLUMEUP        115
#define KEY_POWER       116    /* SC System Power Down */
#define KEY_KPEQUAL     117
#define KEY_KPPLUSMINUS     118
#define KEY_PAUSE       119
#define KEY_SCALE       120    /* AL Compiz Scale (Expose) */

#define KEY_KPCOMMA     121
#define KEY_HANGEUL     122
#define KEY_HANGUEL     KEY_HANGEUL
#define KEY_HANJA       123
#define KEY_YEN         124
#define KEY_LEFTMETA        125
#define KEY_RIGHTMETA       126
#define KEY_COMPOSE     127

#define KEY_STOP        128    /* AC Stop */
#define KEY_AGAIN       129
#define KEY_PROPS       130    /* AC Properties */
#define KEY_UNDO        131    /* AC Undo */
#define KEY_FRONT       132
#define KEY_COPY        133    /* AC Copy */
#define KEY_OPEN        134    /* AC Open */
#define KEY_PASTE       135    /* AC Paste */
#define KEY_FIND        136    /* AC Search */
#define KEY_CUT         137    /* AC Cut */
#define KEY_HELP        138    /* AL Integrated Help Center */
#define KEY_MENU        139    /* Menu (show menu) */
#define KEY_CALC        140    /* AL Calculator */
#define KEY_SETUP       141
#define KEY_SLEEP       142    /* SC System Sleep */
#define KEY_WAKEUP      143    /* System Wake Up */
#define KEY_FILE        144    /* AL Local Machine Browser */
#define KEY_SENDFILE        145
#define KEY_DELETEFILE      146
```

```
#define KEY_XFER        147
#define KEY_PROG1       148
#define KEY_PROG2       149
#define KEY_WWW         150    /* AL Internet Browser */
#define KEY_MSDOS       151
#define KEY_COFFEE      152    /* AL Terminal Lock/Screensaver */
#define KEY_SCREENLOCK      KEY_COFFEE
#define KEY_ROTATE_DISPLAY  153    /* Display orientation for e.g.
tablets */
#define KEY_DIRECTION       KEY_ROTATE_DISPLAY
#define KEY_CYCLEWINDOWS    154
#define KEY_MAIL        155
#define KEY_BOOKMARKS       156    /* AC Bookmarks */
#define KEY_COMPUTER        157
#define KEY_BACK        158    /* AC Back */
#define KEY_FORWARD     159    /* AC Forward */
#define KEY_CLOSECD     160
#define KEY_EJECTCD     161
#define KEY_EJECTCLOSECD    162
#define KEY_NEXTSONG        163
#define KEY_PLAYPAUSE       164
#define KEY_PREVIOUSSONG    165
#define KEY_STOPCD      166
#define KEY_RECORD      167
#define KEY_REWIND      168
#define KEY_PHONE       169    /* Media Select Telephone */
#define KEY_ISO         170
#define KEY_CONFIG      171    /* AL Consumer Control Configuration */
#define KEY_HOMEPAGE        172    /* AC Home */
#define KEY_REFRESH     173    /* AC Refresh */
#define KEY_EXIT        174    /* AC Exit */
#define KEY_MOVE        175
#define KEY_EDIT        176
#define KEY_SCROLLUP        177
#define KEY_SCROLLDOWN      178
#define KEY_KPLEFTPAREN     179
#define KEY_KPRIGHTPAREN    180
#define KEY_NEW         181    /* AC New */
#define KEY_REDO        182    /* AC Redo/Repeat */

#define KEY_F13         183
#define KEY_F14         184
#define KEY_F15         185
#define KEY_F16         186
#define KEY_F17         187
#define KEY_F18         188
#define KEY_F19         189
#define KEY_F20         190
#define KEY_F21         191
#define KEY_F22         192
#define KEY_F23         193
```

```
#define KEY_F24         194

#define KEY_PLAYCD      200
#define KEY_PAUSECD     201
#define KEY_PROG3       202
#define KEY_PROG4       203
#define KEY_DASHBOARD       204    /* AL Dashboard */
#define KEY_SUSPEND     205
#define KEY_CLOSE       206    /* AC Close */
#define KEY_PLAY        207
#define KEY_FASTFORWARD     208
#define KEY_BASSBOOST       209
#define KEY_PRINT       210    /* AC Print */
#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    /* AL Checkbook/Finance */
#define KEY_SPORT       220
#define KEY_SHOP        221
#define KEY_ALTERASE        222
#define KEY_CANCEL      223    /* AC Cancel */
#define KEY_BRIGHTNESSDOWN  224
#define KEY_BRIGHTNESSUP    225
#define KEY_MEDIA       226

#define KEY_SWITCHVIDEOMODE 227    /* Cycle between available video
                      outputs (Monitor/LCD/TV-out/etc) */
#define KEY_KBDILLUMTOGGLE  228
#define KEY_KBDILLUMDOWN    229
#define KEY_KBDILLUMUP      230

#define KEY_SEND        231    /* AC Send */
#define KEY_REPLY       232    /* AC Reply */
#define KEY_FORWARDMAIL     233    /* AC Forward Msg */
#define KEY_SAVE        234    /* AC Save */
#define KEY_DOCUMENTS       235

#define KEY_BATTERY     236

#define KEY_BLUETOOTH       237
#define KEY_WLAN        238
#define KEY_UWB         239

#define KEY_UNKNOWN     240

#define KEY_VIDEO_NEXT      241    /* drive next video source */
```

```
#define KEY_VIDEO_PREV      242    /* drive previous video source */
#define KEY_BRIGHTNESS_CYCLE    243    /* brightness up, after max is
min */
#define KEY_BRIGHTNESS_AUTO 244    /* Set Auto Brightness: manual
                      brightness control is off,
                      rely on ambient */
#define KEY_BRIGHTNESS_ZERO KEY_BRIGHTNESS_AUTO
#define KEY_DISPLAY_OFF     245    /* display device to off state */

#define KEY_WWAN        246    /* Wireless WAN (LTE, UMTS, GSM, etc.)
*/
#define KEY_WIMAX       KEY_WWAN
#define KEY_RFKILL      247    /* Key that controls all radios */

#define KEY_MICMUTE     248    /* Mute / unmute the microphone */

/* Code 255 is reserved for special needs of AT keyboard driver */

#define BTN_MISC        0x100
#define BTN_0           0x100
#define BTN_1           0x101
#define BTN_2           0x102
#define BTN_3           0x103
#define BTN_4           0x104
#define BTN_5           0x105
#define BTN_6           0x106
#define BTN_7           0x107
#define BTN_8           0x108
#define BTN_9           0x109

#define BTN_MOUSE       0x110
#define BTN_LEFT        0x110
#define BTN_RIGHT       0x111
#define BTN_MIDDLE      0x112
#define BTN_SIDE        0x113
#define BTN_EXTRA       0x114
#define BTN_FORWARD     0x115
#define BTN_BACK        0x116
#define BTN_TASK        0x117

#define BTN_JOYSTICK        0x120
#define BTN_TRIGGER     0x120
#define BTN_THUMB       0x121
#define BTN_THUMB2      0x122
#define BTN_TOP         0x123
#define BTN_TOP2        0x124
#define BTN_PINKIE      0x125
#define BTN_BASE        0x126
#define BTN_BASE2       0x127
#define BTN_BASE3       0x128
#define BTN_BASE4       0x129
```

```
#define BTN_BASE5       0x12a
#define BTN_BASE6       0x12b
#define BTN_DEAD        0x12f

#define BTN_GAMEPAD     0x130
#define BTN_SOUTH       0x130
#define BTN_A           BTN_SOUTH
#define BTN_EAST        0x131
#define BTN_B           BTN_EAST
#define BTN_C           0x132
#define BTN_NORTH       0x133
#define BTN_X           BTN_NORTH
#define BTN_WEST        0x134
#define BTN_Y           BTN_WEST
#define BTN_Z           0x135
#define BTN_TL          0x136
#define BTN_TR          0x137
#define BTN_TL2         0x138
#define BTN_TR2         0x139
#define BTN_SELECT      0x13a
#define BTN_START       0x13b
#define BTN_MODE        0x13c
#define BTN_THUMBL      0x13d
#define BTN_THUMBR      0x13e

#define BTN_DIGI        0x140
#define BTN_TOOL_PEN        0x140
#define BTN_TOOL_RUBBER     0x141
#define BTN_TOOL_BRUSH      0x142
#define BTN_TOOL_PENCIL     0x143
#define BTN_TOOL_AIRBRUSH   0x144
#define BTN_TOOL_FINGER     0x145
#define BTN_TOOL_MOUSE      0x146
#define BTN_TOOL_LENS       0x147
#define BTN_TOOL_QUINTTAP   0x148    /* Five fingers on trackpad */
#define BTN_STYLUS3     0x149
#define BTN_TOUCH       0x14a
#define BTN_STYLUS      0x14b
#define BTN_STYLUS2     0x14c
#define BTN_TOOL_DOUBLETAP  0x14d
#define BTN_TOOL_TRIPLETAP  0x14e
#define BTN_TOOL_QUADTAP    0x14f    /* Four fingers on trackpad */

#define BTN_WHEEL       0x150
#define BTN_GEAR_DOWN       0x150
#define BTN_GEAR_UP     0x151

#define KEY_OK          0x160
#define KEY_SELECT      0x161
#define KEY_GOTO        0x162
#define KEY_CLEAR       0x163
```

```
#define KEY_POWER2      0x164
#define KEY_OPTION      0x165
#define KEY_INFO        0x166    /* AL OEM Features/Tips/Tutorial */
#define KEY_TIME        0x167
#define KEY_VENDOR      0x168
#define KEY_ARCHIVE     0x169
#define KEY_PROGRAM     0x16a    /* Media Select Program Guide */
#define KEY_CHANNEL     0x16b
#define KEY_FAVORITES       0x16c
#define KEY_EPG         0x16d
#define KEY_PVR         0x16e    /* Media Select Home */
#define KEY_MHP         0x16f
#define KEY_LANGUAGE        0x170
#define KEY_TITLE       0x171
#define KEY_SUBTITLE        0x172
#define KEY_ANGLE       0x173
#define KEY_FULL_SCREEN     0x174    /* AC View Toggle */
#define KEY_ZOOM        KEY_FULL_SCREEN
#define KEY_MODE        0x175
#define KEY_KEYBOARD        0x176
#define KEY_ASPECT_RATIO    0x177    /* HUTRR37: Aspect */
#define KEY_SCREEN      KEY_ASPECT_RATIO
#define KEY_PC          0x178    /* Media Select Computer */
#define KEY_TV          0x179    /* Media Select TV */
#define KEY_TV2         0x17a    /* Media Select Cable */
#define KEY_VCR         0x17b    /* Media Select VCR */
#define KEY_VCR2        0x17c    /* VCR Plus */
#define KEY_SAT         0x17d    /* Media Select Satellite */
#define KEY_SAT2        0x17e
#define KEY_CD          0x17f    /* Media Select CD */
#define KEY_TAPE        0x180    /* Media Select Tape */
#define KEY_RADIO       0x181
#define KEY_TUNER       0x182    /* Media Select Tuner */
#define KEY_PLAYER      0x183
#define KEY_TEXT        0x184
#define KEY_DVD         0x185    /* Media Select DVD */
#define KEY_AUX         0x186
#define KEY_MP3         0x187
#define KEY_AUDIO       0x188    /* AL Audio Browser */
#define KEY_VIDEO       0x189    /* AL Movie Browser */
#define KEY_DIRECTORY       0x18a
#define KEY_LIST        0x18b
#define KEY_MEMO        0x18c    /* Media Select Messages */
#define KEY_CALENDAR        0x18d
#define KEY_RED         0x18e
#define KEY_GREEN       0x18f
#define KEY_YELLOW      0x190
#define KEY_BLUE        0x191
#define KEY_CHANNELUP       0x192    /* Channel Increment */
#define KEY_CHANNELDOWN     0x193    /* Channel Decrement */
#define KEY_FIRST       0x194
```

```
#define KEY_LAST        0x195    /* Recall Last */
#define KEY_AB          0x196
#define KEY_NEXT        0x197
#define KEY_RESTART     0x198
#define KEY_SLOW        0x199
#define KEY_SHUFFLE     0x19a
#define KEY_BREAK       0x19b
#define KEY_PREVIOUS        0x19c
#define KEY_DIGITS      0x19d
#define KEY_TEEN        0x19e
#define KEY_TWEN        0x19f
#define KEY_VIDEOPHONE      0x1a0    /* Media Select Video Phone */
#define KEY_GAMES       0x1a1    /* Media Select Games */
#define KEY_ZOOMIN      0x1a2    /* AC Zoom In */
#define KEY_ZOOMOUT     0x1a3    /* AC Zoom Out */
#define KEY_ZOOMRESET       0x1a4    /* AC Zoom */
#define KEY_WORDPROCESSOR   0x1a5    /* AL Word Processor */
#define KEY_EDITOR      0x1a6    /* AL Text Editor */
#define KEY_SPREADSHEET     0x1a7    /* AL Spreadsheet */
#define KEY_GRAPHICSEDITOR  0x1a8    /* AL Graphics Editor */
#define KEY_PRESENTATION    0x1a9    /* AL Presentation App */
#define KEY_DATABASE        0x1aa    /* AL Database App */
#define KEY_NEWS        0x1ab    /* AL Newsreader */
#define KEY_VOICEMAIL       0x1ac    /* AL Voicemail */
#define KEY_ADDRESSBOOK     0x1ad    /* AL Contacts/Address Book */
#define KEY_MESSENGER       0x1ae    /* AL Instant Messaging */
#define KEY_DISPLAYTOGGLE   0x1af    /* Turn display (LCD) on and off
*/
#define KEY_BRIGHTNESS_TOGGLE   KEY_DISPLAYTOGGLE
#define KEY_SPELLCHECK      0x1b0   /* AL Spell Check */
#define KEY_LOGOFF      0x1b1   /* AL Logoff */

#define KEY_DOLLAR      0x1b2
#define KEY_EURO        0x1b3

#define KEY_FRAMEBACK       0x1b4    /* Consumer - transport controls
*/
#define KEY_FRAMEFORWARD    0x1b5
#define KEY_CONTEXT_MENU    0x1b6    /* GenDesc - system context menu
*/
#define KEY_MEDIA_REPEAT    0x1b7    /* Consumer - transport control */
#define KEY_10CHANNELSUP    0x1b8    /* 10 channels up (10+) */
#define KEY_10CHANNELSDOWN  0x1b9    /* 10 channels down (10-) */
#define KEY_IMAGES      0x1ba    /* AL Image Browser */
#define KEY_NOTIFICATION_CENTER 0x1bc    /* Show/hide the notification
center */
#define KEY_PICKUP_PHONE    0x1bd    /* Answer incoming call */
#define KEY_HANGUP_PHONE    0x1be    /* Decline incoming call */

#define KEY_DEL_EOL     0x1c0
#define KEY_DEL_EOS     0x1c1
```

```
#define KEY_INS_LINE        0x1c2
#define KEY_DEL_LINE        0x1c3

#define KEY_FN          0x1d0
#define KEY_FN_ESC      0x1d1
#define KEY_FN_F1       0x1d2
#define KEY_FN_F2       0x1d3
#define KEY_FN_F3       0x1d4
#define KEY_FN_F4       0x1d5
#define KEY_FN_F5       0x1d6
#define KEY_FN_F6       0x1d7
#define KEY_FN_F7       0x1d8
#define KEY_FN_F8       0x1d9
#define KEY_FN_F9       0x1da
#define KEY_FN_F10      0x1db
#define KEY_FN_F11      0x1dc
#define KEY_FN_F12      0x1dd
#define KEY_FN_1        0x1de
#define KEY_FN_2        0x1df
#define KEY_FN_D        0x1e0
#define KEY_FN_E        0x1e1
#define KEY_FN_F        0x1e2
#define KEY_FN_S        0x1e3
#define KEY_FN_B        0x1e4
#define KEY_FN_RIGHT_SHIFT  0x1e5

#define KEY_BRL_DOT1        0x1f1
#define KEY_BRL_DOT2        0x1f2
#define KEY_BRL_DOT3        0x1f3
#define KEY_BRL_DOT4        0x1f4
#define KEY_BRL_DOT5        0x1f5
#define KEY_BRL_DOT6        0x1f6
#define KEY_BRL_DOT7        0x1f7
#define KEY_BRL_DOT8        0x1f8
#define KEY_BRL_DOT9        0x1f9
#define KEY_BRL_DOT10       0x1fa

#define KEY_NUMERIC_0       0x200    /* used by phones, remote
controls, */
#define KEY_NUMERIC_1       0x201    /* and other keypads */
#define KEY_NUMERIC_2       0x202
#define KEY_NUMERIC_3       0x203
#define KEY_NUMERIC_4       0x204
#define KEY_NUMERIC_5       0x205
#define KEY_NUMERIC_6       0x206
#define KEY_NUMERIC_7       0x207
#define KEY_NUMERIC_8       0x208
#define KEY_NUMERIC_9       0x209
#define KEY_NUMERIC_STAR    0x20a
#define KEY_NUMERIC_POUND   0x20b
#define KEY_NUMERIC_A       0x20c    /* Phone key A - HUT Telephony
```

```
0xb9 */
#define KEY_NUMERIC_B       0x20d
#define KEY_NUMERIC_C       0x20e
#define KEY_NUMERIC_D       0x20f

#define KEY_CAMERA_FOCUS    0x210
#define KEY_WPS_BUTTON      0x211    /* WiFi Protected Setup key */

#define KEY_TOUCHPAD_TOGGLE 0x212    /* Request switch touchpad on or
off */
#define KEY_TOUCHPAD_ON     0x213
#define KEY_TOUCHPAD_OFF    0x214

#define KEY_CAMERA_ZOOMIN   0x215
#define KEY_CAMERA_ZOOMOUT  0x216
#define KEY_CAMERA_UP       0x217
#define KEY_CAMERA_DOWN     0x218
#define KEY_CAMERA_LEFT     0x219
#define KEY_CAMERA_RIGHT    0x21a

#define KEY_ATTENDANT_ON    0x21b
#define KEY_ATTENDANT_OFF   0x21c
#define KEY_ATTENDANT_TOGGLE    0x21d    /* Attendant call on or off */
#define KEY_LIGHTS_TOGGLE   0x21e    /* Reading light on or off */

#define BTN_DPAD_UP     0x220
#define BTN_DPAD_DOWN       0x221
#define BTN_DPAD_LEFT       0x222
#define BTN_DPAD_RIGHT      0x223

#define KEY_ALS_TOGGLE      0x230    /* Ambient light sensor */
#define KEY_ROTATE_LOCK_TOGGLE  0x231    /* Display rotation lock */

#define KEY_BUTTONCONFIG        0x240    /* AL Button Configuration */
#define KEY_TASKMANAGER     0x241    /* AL Task/Project Manager */
#define KEY_JOURNAL     0x242    /* AL Log/Journal/Timecard */
#define KEY_CONTROLPANEL        0x243    /* AL Control Panel */
#define KEY_APPSELECT       0x244    /* AL Select Task/Application */
#define KEY_SCREENSAVER     0x245    /* AL Screen Saver */
#define KEY_VOICECOMMAND        0x246    /* Listening Voice Command */
#define KEY_ASSISTANT       0x247    /* AL Context-aware desktop
assistant */
#define KEY_KBD_LAYOUT_NEXT 0x248    /* AC Next Keyboard Layout Select
*/
#define KEY_EMOJI_PICKER    0x249    /* Show/hide emoji picker
(HUTRR101) */

#define KEY_BRIGHTNESS_MIN      0x250    /* Set Brightness to Minimum
*/
#define KEY_BRIGHTNESS_MAX      0x251    /* Set Brightness to Maximum
*/
```

```
#define KEY_KBDINPUTASSIST_PREV     0x260
#define KEY_KBDINPUTASSIST_NEXT     0x261
#define KEY_KBDINPUTASSIST_PREVGROUP        0x262
#define KEY_KBDINPUTASSIST_NEXTGROUP        0x263
#define KEY_KBDINPUTASSIST_ACCEPT       0x264
#define KEY_KBDINPUTASSIST_CANCEL       0x265

/* Diagonal movement keys */
#define KEY_RIGHT_UP            0x266
#define KEY_RIGHT_DOWN          0x267
#define KEY_LEFT_UP         0x268
#define KEY_LEFT_DOWN           0x269

#define KEY_ROOT_MENU           0x26a /* Show Device's Root Menu */
/* Show Top Menu of the Media (e.g. DVD) */
#define KEY_MEDIA_TOP_MENU      0x26b
#define KEY_NUMERIC_11          0x26c
#define KEY_NUMERIC_12          0x26d
/*
 * Toggle Audio Description: refers to an audio service that helps
blind and
 * visually impaired consumers understand the action in a program.
Note: in
 * some countries this is referred to as "Video Description".
 */
#define KEY_AUDIO_DESC          0x26e
#define KEY_3D_MODE         0x26f
#define KEY_NEXT_FAVORITE       0x270
#define KEY_STOP_RECORD         0x271
#define KEY_PAUSE_RECORD        0x272
#define KEY_VOD             0x273 /* Video on Demand */
#define KEY_UNMUTE          0x274
#define KEY_FASTREVERSE         0x275
#define KEY_SLOWREVERSE         0x276
/*
 * Control a data application associated with the currently viewed
channel,
 * e.g. teletext or data broadcast application (MHEG, MHP, HbbTV, etc.)
 */
#define KEY_DATA            0x277
#define KEY_ONSCREEN_KEYBOARD       0x278
/* Electronic privacy screen control */
#define KEY_PRIVACY_SCREEN_TOGGLE   0x279

/* Select an area of screen to be copied */
#define KEY_SELECTIVE_SCREENSHOT    0x27a

/*
 * Some keyboards have keys which do not have a defined meaning, these
keys
 * are intended to be programmed / bound to macros by the user. For
```

```
most
 * keyboards with these macro-keys the key-sequence to inject, or
action to
 * take, is all handled by software on the host side. So from the
kernel's
 * point of view these are just normal keys.
 *
 * The KEY_MACRO# codes below are intended for such keys, which may be
labeled
 * e.g. G1-G18, or S1 - S30. The KEY_MACRO# codes MUST NOT be used for
keys
 * where the marking on the key does indicate a defined meaning /
purpose.
 *
 * The KEY_MACRO# codes MUST also NOT be used as fallback for when no
existing
 * KEY_FOO define matches the marking / purpose. In this case a new
KEY_FOO
 * define MUST be added.
 */
#define KEY_MACRO1          0x290
#define KEY_MACRO2          0x291
#define KEY_MACRO3          0x292
#define KEY_MACRO4          0x293
#define KEY_MACRO5          0x294
#define KEY_MACRO6          0x295
#define KEY_MACRO7          0x296
#define KEY_MACRO8          0x297
#define KEY_MACRO9          0x298
#define KEY_MACRO10         0x299
#define KEY_MACRO11         0x29a
#define KEY_MACRO12         0x29b
#define KEY_MACRO13         0x29c
#define KEY_MACRO14         0x29d
#define KEY_MACRO15         0x29e
#define KEY_MACRO16         0x29f
#define KEY_MACRO17         0x2a0
#define KEY_MACRO18         0x2a1
#define KEY_MACRO19         0x2a2
#define KEY_MACRO20         0x2a3
#define KEY_MACRO21         0x2a4
#define KEY_MACRO22         0x2a5
#define KEY_MACRO23         0x2a6
#define KEY_MACRO24         0x2a7
#define KEY_MACRO25         0x2a8
#define KEY_MACRO26         0x2a9
#define KEY_MACRO27         0x2aa
#define KEY_MACRO28         0x2ab
#define KEY_MACRO29         0x2ac
#define KEY_MACRO30         0x2ad
```

```
/*
 * Some keyboards with the macro-keys described above have some extra
keys
 * for controlling the host-side software responsible for the macro
handling:
 * -A macro recording start/stop key. Note that not all keyboards which
emit
 *  KEY_MACRO_RECORD_START will also emit KEY_MACRO_RECORD_STOP if
 *  KEY_MACRO_RECORD_STOP is not advertised, then
KEY_MACRO_RECORD_START
 *  should be interpreted as a recording start/stop toggle;
 * -Keys for switching between different macro (pre)sets, either a key
for
 *  cycling through the configured presets or keys to directly select a
preset.
 */
#define KEY_MACRO_RECORD_START      0x2b0
#define KEY_MACRO_RECORD_STOP       0x2b1
#define KEY_MACRO_PRESET_CYCLE      0x2b2
#define KEY_MACRO_PRESET1       0x2b3
#define KEY_MACRO_PRESET2       0x2b4
#define KEY_MACRO_PRESET3       0x2b5

/*
 * Some keyboards have a buildin LCD panel where the contents are
controlled
 * by the host. Often these have a number of keys directly below the
LCD
 * intended for controlling a menu shown on the LCD. These keys often
don't
 * have any labeling so we just name them KEY_KBD_LCD_MENU#
 */
#define KEY_KBD_LCD_MENU1       0x2b8
#define KEY_KBD_LCD_MENU2       0x2b9
#define KEY_KBD_LCD_MENU3       0x2ba
#define KEY_KBD_LCD_MENU4       0x2bb
#define KEY_KBD_LCD_MENU5       0x2bc

#define BTN_TRIGGER_HAPPY       0x2c0
#define BTN_TRIGGER_HAPPY1      0x2c0
#define BTN_TRIGGER_HAPPY2      0x2c1
#define BTN_TRIGGER_HAPPY3      0x2c2
#define BTN_TRIGGER_HAPPY4      0x2c3
#define BTN_TRIGGER_HAPPY5      0x2c4
#define BTN_TRIGGER_HAPPY6      0x2c5
#define BTN_TRIGGER_HAPPY7      0x2c6
#define BTN_TRIGGER_HAPPY8      0x2c7
#define BTN_TRIGGER_HAPPY9      0x2c8
#define BTN_TRIGGER_HAPPY10     0x2c9
#define BTN_TRIGGER_HAPPY11     0x2ca
#define BTN_TRIGGER_HAPPY12     0x2cb
```

```
#define BTN_TRIGGER_HAPPY13     0x2cc
#define BTN_TRIGGER_HAPPY14     0x2cd
#define BTN_TRIGGER_HAPPY15     0x2ce
#define BTN_TRIGGER_HAPPY16     0x2cf
#define BTN_TRIGGER_HAPPY17     0x2d0
#define BTN_TRIGGER_HAPPY18     0x2d1
#define BTN_TRIGGER_HAPPY19     0x2d2
#define BTN_TRIGGER_HAPPY20     0x2d3
#define BTN_TRIGGER_HAPPY21     0x2d4
#define BTN_TRIGGER_HAPPY22     0x2d5
#define BTN_TRIGGER_HAPPY23     0x2d6
#define BTN_TRIGGER_HAPPY24     0x2d7
#define BTN_TRIGGER_HAPPY25     0x2d8
#define BTN_TRIGGER_HAPPY26     0x2d9
#define BTN_TRIGGER_HAPPY27     0x2da
#define BTN_TRIGGER_HAPPY28     0x2db
#define BTN_TRIGGER_HAPPY29     0x2dc
#define BTN_TRIGGER_HAPPY30     0x2dd
#define BTN_TRIGGER_HAPPY31     0x2de
#define BTN_TRIGGER_HAPPY32     0x2df
#define BTN_TRIGGER_HAPPY33     0x2e0
#define BTN_TRIGGER_HAPPY34     0x2e1
#define BTN_TRIGGER_HAPPY35     0x2e2
#define BTN_TRIGGER_HAPPY36     0x2e3
#define BTN_TRIGGER_HAPPY37     0x2e4
#define BTN_TRIGGER_HAPPY38     0x2e5
#define BTN_TRIGGER_HAPPY39     0x2e6
#define BTN_TRIGGER_HAPPY40     0x2e7

/* We avoid low common keys in module aliases so they don't get huge.
*/
#define KEY_MIN_INTERESTING KEY_MUTE
#define KEY_MAX         0x2ff
#define KEY_CNT         (KEY_MAX+1)

/*
 * Relative axes
 */

#define REL_X           0x00
#define REL_Y           0x01
#define REL_Z           0x02
#define REL_RX          0x03
#define REL_RY          0x04
#define REL_RZ          0x05
#define REL_HWHEEL      0x06
#define REL_DIAL        0x07
#define REL_WHEEL       0x08
#define REL_MISC        0x09
/*
 * 0x0a is reserved and should not be used in input drivers.
```

```
 * It was used by HID as REL_MISC+1 and userspace needs to detect if
 * the next REL_* event is correct or is just REL_MISC + n.
 * We define here REL_RESERVED so userspace can rely on it and detect
 * the situation described above.
 */
#define REL_RESERVED        0x0a
#define REL_WHEEL_HI_RES    0x0b
#define REL_HWHEEL_HI_RES   0x0c
#define REL_MAX         0x0f
#define REL_CNT         (REL_MAX+1)

/*
 * Absolute axes
 */

#define ABS_X           0x00
#define ABS_Y           0x01
#define ABS_Z           0x02
#define ABS_RX          0x03
#define ABS_RY          0x04
#define ABS_RZ          0x05
#define ABS_THROTTLE        0x06
#define ABS_RUDDER      0x07
#define ABS_WHEEL       0x08
#define ABS_GAS         0x09
#define ABS_BRAKE       0x0a
#define ABS_HAT0X       0x10
#define ABS_HAT0Y       0x11
#define ABS_HAT1X       0x12
#define ABS_HAT1Y       0x13
#define ABS_HAT2X       0x14
#define ABS_HAT2Y       0x15
#define ABS_HAT3X       0x16
#define ABS_HAT3Y       0x17
#define ABS_PRESSURE        0x18
#define ABS_DISTANCE        0x19
#define ABS_TILT_X      0x1a
#define ABS_TILT_Y      0x1b
#define ABS_TOOL_WIDTH      0x1c

#define ABS_VOLUME      0x20

#define ABS_MISC        0x28

/*
 * 0x2e is reserved and should not be used in input drivers.
 * It was used by HID as ABS_MISC+6 and userspace needs to detect if
 * the next ABS_* event is correct or is just ABS_MISC + n.
 * We define here ABS_RESERVED so userspace can rely on it and detect
 * the situation described above.
 */
```

```
#define ABS_RESERVED        0x2e

#define ABS_MT_SLOT     0x2f    /* MT slot being modified */
#define ABS_MT_TOUCH_MAJOR  0x30    /* Major axis of touching ellipse
*/
#define ABS_MT_TOUCH_MINOR  0x31    /* Minor axis (omit if circular) */
#define ABS_MT_WIDTH_MAJOR  0x32    /* Major axis of approaching
ellipse */
#define ABS_MT_WIDTH_MINOR  0x33    /* Minor axis (omit if circular) */
#define ABS_MT_ORIENTATION  0x34    /* Ellipse orientation */
#define ABS_MT_POSITION_X   0x35    /* Center X touch position */
#define ABS_MT_POSITION_Y   0x36    /* Center Y touch position */
#define ABS_MT_TOOL_TYPE    0x37    /* Type of touching device */
#define ABS_MT_BLOB_ID      0x38    /* Group a set of packets as a blob
*/
#define ABS_MT_TRACKING_ID  0x39    /* Unique ID of initiated contact
*/
#define ABS_MT_PRESSURE     0x3a    /* Pressure on contact area */
#define ABS_MT_DISTANCE     0x3b    /* Contact hover distance */
#define ABS_MT_TOOL_X       0x3c    /* Center X tool position */
#define ABS_MT_TOOL_Y       0x3d    /* Center Y tool position */


#define ABS_MAX         0x3f
#define ABS_CNT         (ABS_MAX+1)

/*
 * Switch events
 */

#define SW_LID          0x00  /* set = lid shut */
#define SW_TABLET_MODE      0x01  /* set = tablet mode */
#define SW_HEADPHONE_INSERT 0x02  /* set = inserted */
#define SW_RFKILL_ALL       0x03  /* rfkill master switch, type "any"
                    set = radio enabled */
#define SW_RADIO        SW_RFKILL_ALL    /* deprecated */
#define SW_MICROPHONE_INSERT    0x04  /* set = inserted */
#define SW_DOCK         0x05  /* set = plugged into dock */
#define SW_LINEOUT_INSERT   0x06  /* set = inserted */
#define SW_JACK_PHYSICAL_INSERT 0x07  /* set = mechanical switch set */
#define SW_VIDEOOUT_INSERT  0x08  /* set = inserted */
#define SW_CAMERA_LENS_COVER    0x09  /* set = lens covered */
#define SW_KEYPAD_SLIDE     0x0a  /* set = keypad slide out */
#define SW_FRONT_PROXIMITY  0x0b  /* set = front proximity sensor
active */
#define SW_ROTATE_LOCK      0x0c  /* set = rotate locked/disabled */
#define SW_LINEIN_INSERT    0x0d  /* set = inserted */
#define SW_MUTE_DEVICE      0x0e  /* set = device disabled */
#define SW_PEN_INSERTED     0x0f  /* set = pen inserted */
#define SW_MACHINE_COVER    0x10  /* set = cover closed */
#define SW_MAX          0x10
```

```
#define SW_CNT          (SW_MAX+1)

/*
 * Misc events
 */

#define MSC_SERIAL      0x00
#define MSC_PULSELED        0x01
#define MSC_GESTURE     0x02
#define MSC_RAW         0x03
#define MSC_SCAN        0x04
#define MSC_TIMESTAMP       0x05
#define MSC_MAX         0x07
#define MSC_CNT         (MSC_MAX+1)

/*
 * LEDs
 */

#define LED_NUML        0x00
#define LED_CAPSL       0x01
#define LED_SCROLLL     0x02
#define LED_COMPOSE     0x03
#define LED_KANA        0x04
#define LED_SLEEP       0x05
#define LED_SUSPEND     0x06
#define LED_MUTE        0x07
#define LED_MISC        0x08
#define LED_MAIL        0x09
#define LED_CHARGING        0x0a
#define LED_MAX         0x0f
#define LED_CNT         (LED_MAX+1)

/*
 * Autorepeat values
 */

#define REP_DELAY       0x00
#define REP_PERIOD      0x01
#define REP_MAX         0x01
#define REP_CNT         (REP_MAX+1)

/*
 * Sounds
 */

#define SND_CLICK       0x00
#define SND_BELL        0x01
#define SND_TONE        0x02
#define SND_MAX         0x07
#define SND_CNT         (SND_MAX+1)
```

```
#endif
```

Execute

In case the emulator in question supports no keyboard shortcuts to perform actions, evmapy is capable of running commands directly. This should only be used as a last-resort, and typically only to kill the emulator.

To execute a command instead of pressing a key, replace the type “key” with “exec”, then type out the desired command enclosed in quotation marks as the target. For example, from the Sonic Retro engines (which do not feature keyboard shortcuts):

```
{
        "trigger": ["hotkey", "start"],
        "type": "exec",
        "target": "killall -9 sonic2013; killall -9 soniccd"
}
```

From:
https://www.wiki.batocera.org/ - **Batocera.linux - Wiki**

Permanent link:
https://www.wiki.batocera.org/evmapy?rev=1649670295

Last update: **2022/04/11 09:44**