

USB HID.

More detailed information on USB HID and HID reports can be found:

http://www.usb.org/developers/hidpage/Hut1_12v2.pdf

http://www.usb.org/developers/hidpage/HID1_11.pdf

HID Keyboard Report.

Byte:	Function:
0	<i>Modifiers</i>
1	Reserved (0x00)
2	<i>Key1</i>
3	<i>Key2</i>
4	<i>Key3</i>
5	<i>Key4</i>
6	<i>Key5</i>
7	<i>Key6</i>

Modifiers byte:

Bit	Function:
0	Left CTRL
1	Left SHIFT
2	Left ALT
3	Left GUI (Win key)
4	Right CTRL
5	Right SHIFT
6	Right ALT
7	Right GUI (Win key)

Examples:

For keycodes look-up-table, see HID Usages 1.11 PDF (page 53).

All keys released:

0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
------	------	------	------	------	------	------	------

Key “A” pressed:

0x00	0x00	0x04	0x00	0x00	0x00	0x00	0x00
------	------	------	------	------	------	------	------

Keys “A” and “B” pressed at the same time:

0x00	0x00	0x04	0x05	0x00	0x00	0x00	0x00
------	------	------	------	------	------	------	------

Modifier keys “Left CTRL”, “Left ALT” and key “Delete” pressed at the same time:

0x05	0x00	0x4C	0x00	0x00	0x00	0x00	0x00
------	------	------	------	------	------	------	------

How to use keyboard reports.

Each keyboard report represents current state of the keyboard. When USB host receives:

0x00	0x00	0x1C	0x00	0x00	0x00	0x00	0x00
------	------	------	------	------	------	------	------

it will assume that key “Y” is being pressed, until next report is received. This is NOT equivalent to: typing “Y” character.

When typing text, you should send sequence of reports that will represent actions of human using physical keyboard:

1. Press *Modifiers* keys (if necessary):

0x02	0x00	0x00	0x00	0x00	0x00	0x00	0x00
------	------	------	------	------	------	------	------

This step is introduced because some USB hosts may not recognize correctly event when *Modifiers* and *Keys* are pressed at exactly the same moment.

2. Keep *Modifiers* keys pressed, press desired *Key*

0x02	0x00	0x1C	0x00	0x00	0x00	0x00	0x00
------	------	------	------	------	------	------	------

3. Release all keys:

0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
------	------	------	------	------	------	------	------

As soon as keys are released, character “Y” will appear. You should always release all keys. Otherwise USB host will assume that keys send in last report are constantly being held, what will result in typing the character multiple times (“YYYYYYYY....”).

Remember that you are NOT sending ASCII / Unicode characters, but keyboard state.

You must take into account:

1. Keyboard layout.

Depending on what keyboard layout is currently being used by USB host, it may be necessary to use different keycodes and even key combinations to produce some characters. For example: when German layout is active, you must use “Z” keycode in order to type “Y” character and vice versa. Some characters are not available in certain layouts. Important: it is not possible for USB keyboard interface to know what layout is being used by USB host. Such information must be provided manually by user.

2. Current state of keyboard LEDs: NumLock, CapsLock and ScrollLock.

When CapsLock is ON, you will get capital characters when “Shift” is not used and small characters when “Shift” is used. Depending on your application it may be a good idea to automatically detect CapsLock state and apply correction (password manager) or not (remote control).

SHORT HID Keyboard Report.

In most cases, only *Modifiers* and *Key1* bytes are used. This format allows to remove unnecessary 0x00s and as a result, use shorter packets. This is particularly important in upcoming Bluetooth Low Energy version.

Byte:	Function:
0	<i>Modifiers</i>
1	<i>Key1</i>

Such report will be changed into standard HID report before sending to USB host:

Modifiers	0x00	Key1	0x00	0x00	0x00	0x00	0x00
-----------	------	------	------	------	------	------	------

Remarks:

Use only with: *CMD* Queue SHORT keyboard reports (0x2C).

Available only in 0.96 and later firmware versions.

Press and Release event:

Allows to even further reduce payload size.

Byte:	Function:
0	<i>Modifiers</i>
1	<i>Key1</i>

Such report will be changed into 3 keyboard reports before being put into keyboard buffer:

<i>Modifiers</i>	0x00	0x00	0x00	0x00	0x00	0x00	0x00
------------------	------	------	------	------	------	------	------

<i>Modifiers</i>	0x00	<i>Key1</i>	0x00	0x00	0x00	0x00	0x00
------------------	------	-------------	------	------	------	------	------

0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
------	------	------	------	------	------	------	------

Remarks:

Use only with: *CMD* Queue Press and Release events (0x2D).

Available only in 0.96 and later firmware versions.

Each event will be changed into 3 keyboard reports. Take this into account when managing keyboard buffer!

HID Mouse Report.

Byte:	Function:
0	<i>Buttons</i>
1	<i>X displacement</i>
2	<i>Y displacement</i>
3	<i>Scroll displacement</i>

Buttons byte:

Bit	Function:
0	Left button
1	Right button
2	Middle button
3	Button 1
4	Button 2
5	Button 3
6	Button 4
7	Button 5

Buttons behaves the same way as keys in keyboard report: USB host will assume that state of buttons remains the same, until it receives next mouse report.

X displacement, *Y displacement* and *Scroll displacement* behaves in a different way: USB host will change respective axis and will not modify it further until it receives next mouse report.

Examples:

All buttons released:

0x0	0x00	0x00	0x00
-----	------	------	------

Left mouse button pressed:

0x01	0x00	0x00	0x00
------	------	------	------

Mouse moved by 1, X axis:

0x00	0x01	0x00	0x00
------	------	------	------

Mouse moved by 5, Y axis, while Right button is being pressed:

0x02	0x00	0x05	0x00
------	------	------	------

Remarks:

HID Mouse reports sent to InputStick must be always 4 bytes long, even when BOOT protocol is set by USB host. In such case value of *Scroll displacement* will be removed by InputStick in order to match mouse boot protocol (3 bytes, scroll wheel is not supported), but it must be provided anyway (use 0x00).

HID Consumer Report:

Byte:	Function:
0	Report ID (0x01)
1	Usage code (LSB)
2	Usage code (MSB)

Notes:

For usage codes, see HID Usages 1.11 PDF (page 75).

HID System Report:

Byte:	Function:
0	Report ID (0x02)
1	Usage code (LSB)
2	Usage code (MSB)

HID Gamepad Report:

Byte:	Function:
0	Report ID (0x03)
1	Buttons 1-8
2	Buttons 9-16
3	X axis
4	Y axis
5	Z axis
6	rX axis

Notes:

Gamepad functionality is available since 0.97 firmware version. Gamepad shares endpoint and report queue (buffer) with consumer control.

Keyboard layouts.

It is not possible for USB keyboard interface to know what layout is currently used by USB host. Such information must be always provided manually by user.

Example:

Following keyboard reports were sent to the USB host:

“Left Shift” is pressed:

0x02	0x00	0x00	0x00	0x00	0x00	0x00	0x00
------	------	------	------	------	------	------	------

“Left Shift” and key “3” are pressed:

0x02	0x00	0x20	0x00	0x00	0x00	0x00	0x00
------	------	------	------	------	------	------	------

All keys released:

0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
------	------	------	------	------	------	------	------

Assume that text editor is currently opened. Depending on what keyboard layout is currently used by the USB host, following character will appear:

US keyboard layout (en-US): #
UK keyboard layout (en-GB): £
Polish keyboard layout (pl-PL): #
German keyboard layout (de-DE): \$

Look-up table:

When typing text using InputStick, use following procedure:

Unicode character → Use look-up table → Generate appropriate HID keyboard reports.

Use tools like “Microsoft Keyboard Layout Creator” to obtain data for different keyboard layouts.

Examples:

“,.” = 0x00, 0x00, 0x00, 0x00, 0x00 (Keyboard report must consist of 8 bytes).

Key names are based on US layout.

Bold indicates cases which differs from US layout.

US keyboard layout (en-US):

Character	Report #1	Report #2	Report #3
:			
a	0x00, 0x00, 0x04, .. (Key „A”)	0x00, 0x00, 0x00, ..	0x00, 0x00, 0x00, ..
A	0x02, 0x00, 0x00, .. (Left Shift)	0x02, 0x00, 0x04, .. (Left Shift + Key „A”)	0x00, 0x00, 0x00, ..
ą	Character not available		
ä	Character not available		
y	0x00, 0x00, 0x1C, .. (Key „Y”)	0x00, 0x00, 0x00, ..	0x00, 0x00, 0x00, ..
3	0x00, 0x00, 0x20, .. (Key „3”)	0x00, 0x00, 0x00, ..	0x00, 0x00, 0x00, ..
#	0x02, 0x00, 0x00, .. (Left Shift)	0x02, 0x00, 0x20, .. (Left Shift + Key „3”)	0x00, 0x00, 0x00, ..
<	0x02, 0x00, 0x00, .. (Left Shift)	0x02, 0x00, 0x36, .. (Left Shift + Key „Coma”)	0x00, 0x00, 0x00, ..
~	0x02, 0x00, 0x00, .. (Left Shift)	0x02, 0x00, 0x35, .. (Left Shift + Key „Grave Accent”)	0x00, 0x00, 0x2C, .. (Key „Spacebar”)

UK keyboard layout (en-GB):

Character	Report #1	Report #2	Report #3
:			
a	0x00, 0x00, 0x04, .. (Key „A”)	0x00, 0x00, 0x00, ..	0x00, 0x00, 0x00, ..
A	0x02, 0x00, 0x00, .. (Left Shift)	0x02, 0x00, 0x04, .. (Left Shift + Key „A”)	0x00, 0x00, 0x00, ..
ą	Character not available		
ä	Character not available		
y	0x00, 0x00, 0x1C, .. (Key „Y”)	0x00, 0x00, 0x00, ..	0x00, 0x00, 0x00, ..
3	0x00, 0x00, 0x20, .. (Key „3”)	0x00, 0x00, 0x00, ..	0x00, 0x00, 0x00, ..
#	0x00, 0x00, 0x31, .. (Key „Backslash”)	0x00, 0x00, 0x00, ..	0x00, 0x00, 0x00, ..
<	0x02, 0x00, 0x00, .. (Left Shift)	0x02, 0x00, 0x36, .. (Left Shift + Key „Coma”)	0x00, 0x00, 0x00, ..
~	0x02, 0x00, 0x00, .. (Left Shift)	0x02, 0x00, 0x31, .. (Left Shift + Key „Backslash”)	0x00, 0x00, 0x00, ..

Polish keyboard layout (pl-PL):

Character	Report #1	Report #2	Report #3
:			
a	0x00, 0x00, 0x04, .. (Key „A”)	0x00, 0x00, 0x00, ..	0x00, 0x00, 0x00, ..
A	0x02, 0x00, 0x00, .. (Left Shift)	0x02, 0x00, 0x04, .. (Left Shift + Key „A”)	0x00, 0x00, 0x00, ..
ą	0x40, 0x00, 0x00, .. (Right Alt)	0x40, 0x00, 0x04, ..	0x00, 0x00, 0x00, ..
ä	Character not available		
y	0x00, 0x00, 0x1C, .. (Key „Y”)	0x00, 0x00, 0x00, ..	0x00, 0x00, 0x00, ..
3	0x00, 0x00, 0x20, .. (Key „3”)	0x00, 0x00, 0x00, ..	0x00, 0x00, 0x00, ..
#	0x02, 0x00, 0x00, .. (Left Shift)	0x02, 0x00, 0x20, .. (Left Shift + Key „3”)	0x00, 0x00, 0x00, ..
<	0x02, 0x00, 0x00, .. (Left Shift)	0x02, 0x00, 0x36, .. (Left Shift + Key „Coma”)	0x00, 0x00, 0x00, ..
~	0x02, 0x00, 0x00, .. (Left Shift)	0x02, 0x00, 0x35, .. (Left Shift + Key „Grave Accent”)	0x00, 0x00, 0x2C, .. (Key „Spacebar”)

German keyboard layout (de-DE):

Character	Report #1	Report #2	Report #3
:			
a	0x00, 0x00, 0x04, .. (Key „A“)	0x00, 0x00, 0x00	0x00, 0x00, 0x00, ..
A	0x02, 0x00, 0x00, .. (Left Shift)	0x02, 0x00, 0x04 (Left Shift + Key „A“)	0x00, 0x00, 0x00, ..
ä	Character not available		
ä	0x00, 0x00, 0x34, .. (Key „Apostrophe“)	0x00, 0x00, 0x00, ..	0x00, 0x00, 0x00, ..
y	0x00, 0x00, 0x1D, .. (Key „Z“)	0x00, 0x00, 0x00, ..	0x00, 0x00, 0x00, ..
3	0x00, 0x00, 0x20, .. (Key „3“)	0x00, 0x00, 0x00, ..	0x00, 0x00, 0x00, ..
#	0x00, 0x00, 0x31, .. (Key „Backslash“)	0x00, 0x00, 0x00, ..	0x00, 0x00, 0x00, ..
<	0x00, 0x00, 0x64, .. (Key „Non-US backslash“)	0x00, 0x00, 0x00, ..	0x00, 0x00, 0x00, ..
~	0x40, 0x00, 0x00, .. (Right Alt)	0x40, 0x00, 0x30 (Right Alt + Key „Right Bracket“)	0x00, 0x00, 0x00, ..