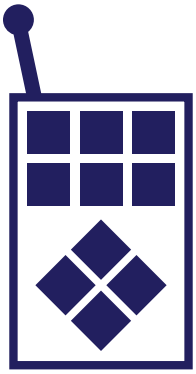

kcKeyboard 2.1
Firmware User Guide

Jan 23, 2014

Wireless Keyboard, Mouse, & Media Controller Firmware
HID Bluetooth Profile

Contents

1 Features

kcKeyboard firmware offers fully qualified Bluetooth Human Interface Device (HID), and can communicate with any Bluetooth device that supports this profile.

A kcKeyboard device provides both the UART interface and PIO interface for transmitting key press information. AT Commands are accepted via UART for configuration and operations.

2 PIO Inputs

PIO pins are LOW by default. Key presses are edge triggered, with a LOW to HIGH signal indicating a key press, and a HIGH to LOW signal indicating a key release.

Any ASCII character can be assigned to available PIO pins. Assignments will be available via AT Command shortly, but is currently not available. Please contact us for custom assignments.

PIO 0	Device Firmware Update (DFU). HIGH to start firmware update via USB. Firmware updates require a firmware image, download application, and special USB driver.
PIO 1	Reconnect button feature. HIGH to start reconnections.

3 PIO Outputs

PIO 4	Connecting indicator. HIGH during pairing and connection phase.
PIO 5	Connected indicator. HIGH when connected.

4 Uart Input

Uart inputs operate in two modes: CommandMode and DataMode. In CommandMode, the Uart accepts AT Commands for configuration and control. When wirelessly connected to a remote device, kcKeyboard will automatically switch DataMode. Sending the escape byte, 0xFA from DataMode will switch kcKeyboard to CommandMode. Also, the disconnect byte, 0xFF, in DataMode can disconnect a wireless link.

When in DataMode, bytes for complete Hid frames can be sent to the Uart for complete control of Keyboard, Mouse, Joystick, and Media characters and functions. These frames require one designated byte prefix, to indicate the type of frame data that follows. Multiple frames can be provided as a single string of bytes.

Bytes received from the UART during DataMode are interpreted as follows:

0x00 – 0x1F	Ignored, except for Backspace (0x08), Carriage Return (0x0C), and Escape (0x1B).
0x20 – 0x7F	All ASCII characters are translated to HID.
0x80 – 0xEF	Ignored.
0xF0 – 0xFF	kcKeyboard Custom: 0xF0 = Raw hid frame bytes to follow. 0xF1 = Hid keyboard frame. Trailing 7 bytes will form a keyboard frame. 0XF2 = Hid mouse frame. Trailing 4 bytes will form a mouse frame. 0XF3 = Hid media frame. Trailing 2 bytes will form a media frame. 0XF4 = Hid joystick frame. Trailing 4 bytes will form a joystick frame. (not implemented yet) 0XF5 = Reserved. 0XF6 = Reserved. 0XF7 = Reserved. 0XF8 = Reserved. 0XF9 = Reserved. 0XFA = Escape, switch from DataMode to CommandMode. 0XFB = Reserved. 0XFC = Reserved. 0XFD = Reserved. 0XFE = Reserved. 0XFF = Disconnect.

5 ASCII Character Translation

When connected, the Uart accepts ASCII characters, translates strings into individual HID key codes, and transmits them to the HID receiver device. Each character is sent individually as a key press, and immediately followed by a key release.

Most basic ASCII printable characters are available for sending including `0x08`, `0x0C`, `0x1B`, and `0x20` up to `0x7F`.

The maximum rate is about 50 characters per second (which is 100 HID frames by sending both the key press and release).

6 Hid Keyboard Frame

Byte 1	Report ID	0xF1 indicates keyboard. kcKeyboard will convert to 0x01.
Byte 2	Key Modifier	Bit[7] = Right GUI Bit[6] = Right Alt Bit[5] = Right Shift Bit[4] = Right Ctrl Bit[3] = Left GUI Bit[2] = Left Alt Bit[1] = Left Shift Bit[0] = Left Ctrl
Byte 3	Reserved	0x00
Byte 4	Keycode 1	Channel 1 Hid Keycode. Example 0x04 is letter ‘a’ or ‘A’ depending on the Key Modifier.
Byte 5	Keycode 2	Channel 2 Hid Keycode.
Byte 6	Keycode 3	Channel 3 Hid Keycode.
Byte 7	Keycode 4	Channel 4 Hid Keycode.
Byte 8	Keycode 5	Channel 5 Hid Keycode.
Byte 9	Keycode 6	Channel 6 Hid Keycode.

Example.

Letter ‘A’ key press, send these 9 bytes to the Uart.

```
0xF1 02 00 04 00 00 00 00 00
```

Letter ‘A’ key release, send these 9 bytes to the Uart.

```
0xF1 02 00 00 00 00 00 00 00
```

Letters ‘B’ and ‘C’ key press, send these 9 bytes to the Uart.

```
0xF1 02 00 05 06 00 00 00 00
```

Letter ‘B’ key release, letter ‘C’ continues to press, send these 9 bytes to the Uart.

```
0xF1 02 00 00 06 00 00 00 00
```

7 Hid Mouse Frame

Byte 1	Report ID	0xF2 indicates mouse. kcKeyboard will convert to 0x02.
Byte 2	Buttons	Bit[7] – Bit[5] are Mouse Buttons 1 - 3
Byte 3	X	X Movement Change
Byte 4	Y	Y Movement Change
Byte 5	Scroll Wheel	-127 to 128 Scroll Change

Example.

Mouse pointer moving up, send these 9 bytes to the Uart.

```
0xF2 00 10 00 00
```

8 Hid Media Frame

Byte 1	Report ID	0xF3 indicates media. kcKeyboard will convert to 0x03.
Byte 2	Feature	Bit[7] = CD Eject (iPhone keypad toggle) Bit[6] = Mute Bit[5] = Volume Up Bit[4] = Volume Down Bit[3] = Play / Pause Bit[2] = Stop Bit[1] = Previous Track Bit[0] = Next Track
Byte 3	Feature	Future Use

Example.

Volume up, send , send these 3 bytes to the Uart.

```
0xF3 20 00
```

Note: some devices will repeat the key press feature until the key release is sent.

```
0xF3 00 00
```

9 Raw HID Frame

Byte 1	Report ID	0xF0 indicates Raw Hid Frame. kcKeyboard will drop this byte.
Byte 2	Size	The number of bytes to follow.
Byte 3+	Frame	The Hid frame bytes to send directly.

Example.

Letter ‘A’ & ‘B’ keyboard key press, using Raw Hid Frame.

```
0xF0 08 01 00 04 05 00 00 00 00
```

Keyboard key release, using Raw Hid Frame.

```
0xF0 08 01 00 00 00 00 00 00 00
```

10 User Guide Version

Version	Changes
Jan 18, 2013	kcKeyboard v2.1
Jan 23, 2014	Formatting

KC Wirefree Corporation
2640 W Medtronic Way
Tempe, Arizona 85281

(602) 386-2640

www.kcwirefree.com
info@kcwirefree.com
