KPod USB Application Interface Specification

© 2016 Elecraft Inc - P. Saffren N6HZ

Overview

The KPod USB device interface use the simple generic HID protocol of communicating with a USB host. All command and return data use 8 byte EP0 packets. Commands are sent to the KPod using an 8 byte command packet, and data is then returned in an 8 byte report packet. After every command is sent, a USB read must be performed, even for commands that return no data.

USB Device Information (from the device descriptor)

USB spec: 2.0
class code: 0x00
sub class: 0x00
protocol: 0x00
EP0 pkt size: 0x08
Vendor ID: 0x04D8
Product ID: 0xF12D

Command Packet (PC → KPod)

The command packet structure consists of a 1 byte command and 7 bytes of data:

```c
struct USB_CmdPacket
{
   unsigned char   cmd;
   unsigned char   data[7];
};
```

`cmd` holds the command (see command list)
`data` holds optional arguments for the command
**Command List** (details shown on next page)

- Get Update: 'u'
- Get ID: '='
- Get Version: 'v'
- Force FW load: 'b'
- Reset: 'r'
- Configure: 'C'
- LED/Aux Control: 'O'
- Beep Control: 'Z'

**Report Packet** (KPod → PC)

```c
struct USB_ReportPkt
{
    unsigned char    cmd;     // command reply
    int16_t   ticks;         // encoder tick count, signed 16 bit
    unsigned char controls; // button, tap/hold and rocker state
    unsigned char spare[4];  // spares, TBD
};
```

- **cmd** - holds copy of command that requested this report
- **ticks** – accumulated encoder ticks, positive = CW, negative = CCW
- **controls:**

<table>
<thead>
<tr>
<th>bit 7</th>
<th>bit 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>unused</td>
<td>Rocker_1</td>
</tr>
</tbody>
</table>

- **Button:**
  - 0x01 = button 1 press
  - 0x02 = button 2 press
  - .......
  - 0x08 = button 8 press

- **Tap/Hold:**
  - 0 = tap
  - 1 = hold

- **Rocker:**
  - 00 = center (VFO B)
  - 01 = right (RIT/XIT)
  - 10 = left (VFO A)
  - 11 = error
Command Reference

'\u' get update - Signals the KPod to return an update report. The KPod will set the returned packet's cmd to 'u' if there was a new event (encoder, button, or rocker), otherwise the cmd will be set to 0.

'=' get ID - Returns the string “KPOD” in the report packet. The report packet is cast as 1 byte command and 7 bytes data:

```c
struct id_report_packet
{
    unsigned char cmd; // will contain ‘=’
    char id_string[7]; // will contain “KPOD”
};
```

'v' get version - The version of the KPod application firmware is returned encoded as BCD in the report’s ‘ticks’ field. (example: 1.08 = 108)

'b' jump to bootloader – Internal Use Only

'r' reset – forces a hard reset of the KPod. This will reset the USB interface.

'C' Configure – The first byte of the command packet data field holds a bit pattern which is used to control various configuration settings.

<table>
<thead>
<tr>
<th>b7</th>
<th>b6</th>
<th>b5</th>
<th>b4</th>
<th>b3</th>
<th>b2</th>
<th>b1</th>
<th>b0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>SCALE</td>
<td>MUTE</td>
</tr>
</tbody>
</table>

SCALE Writing a 1 at this bit position sets the encoder scale at 100 counts per revolution, while writing a 0 sets the default of 200 counts.

MUTE Writing a 1 at this bit position enables beeper mute, while writing a 0 disables mute.
'O'  LED/Aux control – The first byte of the command packet data field holds a bit pattern which is used to turn on/off the LEDs and/or the Aux Outputs.

<table>
<thead>
<tr>
<th>b7</th>
<th>b6</th>
<th>b5</th>
<th>b4</th>
<th>b3</th>
<th>b2</th>
<th>b1</th>
<th>b0</th>
</tr>
</thead>
<tbody>
<tr>
<td>LEDR</td>
<td>LED_4</td>
<td>LED_3</td>
<td>LED_2</td>
<td>LED_1</td>
<td>AUX_3</td>
<td>AUX_2</td>
<td>AUX_1</td>
</tr>
</tbody>
</table>

AUX 1-3  Writing a 1 at this bit position turns the AUX_n output on (grounds the pin), while a 0 will turn it off, opens the connection to ground.

LED 1-4  Writing a 1 turns the LED on, 0 turns it off. (see LEDR below)

LEDR  Writing a 1, LEDs will be controlled by the KPod's rocker switch. Writing a 0, LEDs will be controlled by LED bit commands.

Note: LED D4 is always controlled by LED bit commands.

'Z'  Beep Control – The first 3 bytes of the data portion of the command packet hold the parameters:

data[0] – tone frequency:
0 = 1000 Hz
1 = 1500 Hz
2 = 2000 Hz
3-255 = 500 Hz

data[1] – tone level:
0 = low
1 = medium
2-255 = high

data[2] – duration:
0-255 * 10 ms

(end)