

Busylight/Kuando Box HID communication protocol specification

USB Vendor ID (VID): 0x27BB
USB Product ID (PID): 0x3BCA (BusyLight Alpha model)
0x3BCB (BusyLight UC model)
0x3BCC (Kuando Box)
0x3BCD (BusyLight Omega model)
0x3BC0 (Bootloader for all of the above models)

Outgoing HID package (from PC):

56+8 bytes arranged in 7 steps of 8 bytes each + 8 bytes of additional data.
C-language style definition of the 7 steps is shown below.

```
struct {
    unsigned char cmd;           // bitwise interpretation:
                                // 0b1000xxxx: Keep alive signal = no change. xxxx = timeout in seconds.
                                // 0b01000000: Start Bootloader command
                                // 0b00100000: Reset Device command
                                // 0b00010aaa: Jump to step aaa when the step is executed.
    unsigned char repeat;       // Execute this step for repeat number of times [1..255]
    unsigned char red_int;      // RED LED intensity expressed as PWM on time [0..100] 0-100%
    unsigned char green_int;    // GREEN LED intensity expressed as PWM on time [0..100] <=> 0-100%
    unsigned char blue_int;     // BLUE LED intensity expressed as PWM on time [0..100] <=> 0-100%
    unsigned char on_time;      // ON time in 0.1 second steps [0..255] <=> 0-25.5sec
    unsigned char off_time;     // OFF time in 0.1 second steps [0..255] <=> 0-25.5sec
    unsigned char ringtone;     // bitwise interpretation: 0bcbbbbbee
                                // bbbb is the ringtone/IM alert tone number to be played.
                                // If bit c = 1 the audio setting is changed.
                                // if bit c = 0 the audio setting is ignored.
                                // The volume level is defined by bits eee. Setting eee=000 stops
                                // playing the ringtone. Setting eee=111 is the highest volume level.
                                // There might be fewer than 16 ringtones/IM alert tones available.
                                // Selecting an undefined ringtone is just ignored.
                                // Ringtones are repeated automatically.
} step[7];
```

8 bytes of additional data (index 56 .. 63) carries the following information:

Index 56: Kuando box sensitivity (0-31) 0=high sensitivity 31=low sensitivity.
Index 57: Kuando box timeout (1-30). In seconds.
Index 58: Kuando box triggertime (1-250). In milliseconds.
Index 59-61: Currently no explicit use – set to 0xFF.
Index 62: MSB of 16 bit checksum.
Index 63: LSB of 16 bit checksum.

Checksum calculation & check:

```
int16 checksum = 0;

for(i=0;i++;i<62)
    checksum += incoming_usb_data[i];

if(incoming_usb_data[62] == (checksum / 256) && incoming_usb_data[63] == (checksum % 256))
    { checksum approved ... }
```

Example (BusyLight data):

- Step 0: Blink blue (20% intensity) 3 times (DC:0.5s on 0.5s off), turn off audio; then jump to step 1
- Step 1: Blink red (80% intensity) 5 times (DC:1.5s on 0.5s off), audio setting ignored; then jump to step 2
- Step 2: Show green (100% intensity) steady on for 10 seconds and play ringtone #4 at volume level 3; then jump to step 0

Will be encoded as (hex):

```
11 03 00 00 33 05 05 80 ... step 0
12 05 50 00 00 0F 05 00 ... step 1
10 02 00 64 00 32 00 A3 ... step 2
00 00 00 00 00 00 00 00 ... step 3 unused set to 00
00 00 00 00 00 00 00 00 ... step 4 unused set to 00
00 00 00 00 00 00 00 00 ... step 5 unused set to 00
00 00 00 00 00 00 00 00 ... step 6 unused set to 00
00 00 FF FF FF FF 06 93 ... Additional data including checksum
```

Incoming HID package:

Applies to: BusyLight Alpha, BusyLight UC, Kuando Box and BusyLight Omega.

64 bytes. C-language style definition is shown below.

```
struct {
  unsigned char status;           // [0,1] Kuando box status. "0"=no line activity detected, "1"=line activity detected
                                  // BusyLight will set this to "0"
  unsigned char product_id[3];    // [0-9,A-Z] "001"=Busylight. "002"=Kuando Box
  unsigned char customer_id[8];   // [0-9,A-Z] Customer Identification e.g. "PLENOM00"
  unsigned char model_[4];        // [0-9,A-Z] Model (customer version)
  unsigned char serial_[8];       // [0-9] Serial number e.g. "00000001"
  unsigned char mfg_id[8];        // [0-9,A-Z] Manufacturer ID e.g. "DASAN000"
  unsigned char mfg_date[8];      // [0-9] YYYYMMDD e.g. "20150331"
  unsigned char sw_rev[6];        // [0-9] Software revision MMmmmm "011234" <=> Rev. 01.1234
  unsigned char reserved[18];     // Reserved for future purposes; always set to "0".
} info;
```