**Disclaimer**

**THIS SPECIFICATION IS LICENSED AND PROVIDED BY LOGITECH "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY ANDFITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL LOGITECH OR ANY OF ITS AFFILIATED COMPANIES BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SPECIFICATION, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.**

**[0x4301] Solar Kbd Dashboard DRAFT SPEC**

max\_num\_of\_reports,report\_period = [0]SetLightMeasure(max\_num\_of\_reports, report\_period)

LED\_ID = [1]SetLED(LED\_ID)

**Application Specific Event**

**Broadcast Event**

BattStatus = [0]BatteryBroadcastEvent

BattLightStatus = [1]BattLightMeasureBroadcastEvent

BattLightStatus = [2]CheckLightButtonBroadcastEvent

**FUNCTIONS**

SetLightMeasure

**Summary**

Gives the current light measure in lux (min 0 = 0 lux, max 511 = 511 lux)

**Parameters**

max\_num\_of\_reports [8bits] Maximum number of BattLightMeasureEvents that will be sent after function call (0 = cancel event reporting)

report\_period [8bits] Minimum time (1s units) between two consecutive BattLightMeasureEvents (0 = cancel event reporting)

**Errors**

None

**Remarks**

max\_num\_of\_reports counter must be reset when exiting disconnected mode.

example:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |
|  | 85 |  | 411 | 03 | 06 | 0 | X | 10 03 06 0F 05 02 00 SetLightMeasure |
|  |  |  | 438 | 03 | 06 | 0 | R | 11 03 06 0F 05 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |
|  | 87 |  | 442 |  |  |  | 3 | SetLightMeasure response  [ num\_of\_reports: 5, period: 2 seconds] |

**Request**

|  |  |  |
| --- | --- | --- |
| 0 | 1 | 2 |
| max\_num\_of\_reports | report\_period | 0 |

**Response**

|  |  |  |
| --- | --- | --- |
| 0 | 1 | 2 |
| max\_num\_of\_reports | report\_period | 0 |

SetLED

**Summary**

Light the selected LED for the duration (hard-coded in FW)

**Parameters**

LED\_ID [8bits] Target LED (0=LEDS\_OFF, 1=RED, 2=ORANGE, 3=GREEN)

**Returns**

[void]

**Errors**none

**Remarks**

SetLED is supposed to be used only to override the FW CheckLight LED display. FW will wait 250ms before displaying CheckLight status on LEDs, to let host SW enough time to react to the CheckLightButtonBroadcastEvent.  
If SW sends the SetLED before the 250ms timeout expires, the FW does not display CheckLight status on LEDs at all, but if timeout expires, FW displays its own CheckLight status.

**Request**

|  |  |  |
| --- | --- | --- |
| 0 | 1 | 2 |
| LED\_ID | 0 | 0 |

**Response**

|  |  |  |
| --- | --- | --- |
| 0 | 1 | 2 |
| LED\_ID | 0 | 0 |

**EVENTS**

BatteryBroadcastEvent

Reports battery capacity to host SW spontaneously each 90s when keyboard is being used A report is also sent at power-up and when exiting disconnected mode.

**LightLevelMeasure bytes are set to 0.**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  | |
|  | 132 | 07 12:22:46 | 837 | 03 | 06 | **Notif 0** | R | (91000)11 03 06 00 60 00 00 06 72 47 4F 4F 44 00 00 00 00 00 00 00 | |
|  | 65 |  |  |  |  |  | |  |  | |
|  | 71 |  |  |  |  |  |  |  | |
|  | 135 |  | 838 |  |  |  | 1 | Broadcast Event[BattLightBroadcastEvent] received [BatteryLevel:  96%, LightLevel: 0, BattADC: 2.67793237653V]  K750 received after 71.8300018311 s | |

BattLightMeasureEvent

Reports battery and light level to host SW according to parameters provided by SetLightMeasure function

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |
|  | 108 |  | 079 | 03 | 06 | **Notif 1** | R | (3000)11 03 06 10 60 01 3F 06 72 47 4F 4F 44 00 00 00 00 00 00 00 |
|  | 65 |  | 079 |  |  |  | 3 |  |
|  | 71 |  | 079 |  |  |  | 3 |  |
|  | 111 |  | 080 |  |  |  | 1 | Broadcast Event[BattLightMeasureBroadcastEvent] received  [BatteryLevel: 96%, LightLevel: 319, BattADC: 2.67793237653V]  K750 received after 0.0 s |

CheckLightButtonBroadcastEvent

Flag that CheckLight button has been pressed and report also battery and light level to SW

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
| 132 |  | 006 | 03 | 06 | **Notif 2** | R | (91000)11 03 06 20 60 00 00 06 72 47 4F 4F 44 00 00 00 00 00 00 00 |
| 65 |  | 006 |  |  |  | 3 |  |
| 71 |  | 006 |  |  |  | 3 |  |
| 135 |  | 007 |  |  |  | 1 | Broadcast Event[CheckLightButtonBroadcastEvent] received  [BatteryLevel: 96%, LightLevel: 0, BattADC: 2.67793237653V]  K750 received after 0.0 s |

**Report**

BatteryLevel [8bits] The remaining battery capacity as defined in [Notes Link](Notes:///C1257618003EF8F0/2DB07D495B7E9985C1256C280073A40C/1D901CF7C5517EA4C125766D00425D3F),

LightLevelMeasure [16bits] Gives the current light measure in lux

**Remarks**

See BattLightMeasureEvent remarks

**NOTE:** When CheckLight button is pressed, the HID code page 0x0C, usage 0x01F0 is also sent (dummy keystroke, sent for FW internal reasons)

**Report**

|  |  |  |
| --- | --- | --- |
| 0 | 1 | 2 |
| BatteryLevel | LightLevelMeasure MSB | LightLevelMeasure LSB |