Request Submission Template

Request #:	HUTRR39
Title:	HID Sensor Usage Tables
Spec Release:	HID 1.12
Received:	05 May, 2011
Requester:	Jim Trethewey
Company:	Intel Corporation
Phone:	+1 503 264 4636
Fax:	+1 503 264 4230
Email:	jim.r.trethewey@intel.com
Current Status:	Approved
Priority:	Normal
Submitted:	05 May, 2011
Voting Starts:	23 June 2011
Voting Ends:	30 June 2011
Required Voter:	HID Chair Nathan Sherman nathans@microsoft.com
Required Voter:	Intel Corporation Steve McGowan steve.mcgowan@intel.com
Required Voter:	ST Microelectronics Bo Kang bo.kang@st.com
Response:	Original submission modified due to feedback, replaced with this version, received 17 June 2011
Approved by voting companies	3 Yes votes1 No vote

Notes on Approval Procedure

HID WG On Line Voting Procedures:

1. Votes are on a per company basis.
2. Each Review Request shall have attached a Required Voter List that is the result of recruiting by the HID Chair and submitter of members of the USB IF. Required Voter List must include the HID Chair plus 2 companies (other than the submitter) plus any others designated by the HID Chair at the Chair's discretion. The Required Voter List ensures that a quorum is available to approve the Request.
3. Impose a 7-calendar-day posting time limit for new Review Requests. HID Chair or designate must post the RR within 7 calendar days. HID Chair or designate must work with the submitter to make sure the request is valid prior to posting. Valid review request must include all fields marked as required in the template. A new template will be adopted that requires at least the following fields: Change Text, Required Voter List, Review Period End Date and Voting End Date, Submittal Date, Submitter, Review Request Title and RR Number.
4. If a RR approval process stalls, the HID Chair may call a face-to-face meeting or conference call to decide the issue. Submitter may request that this take place.
5. Impose a minimum 15-calendar-day review period on a posted RR prior to the voting period. At HID Chair discretion, changes to the RR may require this review period to restart.
6. The Chair will accept votes via documentable means such as mail or e-mail during the 7 calendar days after the close of the review period. If a Required Voter does not vote during the period, then there is no quorum and the Chair may pursue the absent required voter and extend the voting

period. The Chair may designate a substitute for the absent voter and extend the voting period if necessary.

Summary

A new HID Usage Page for various types of sensors is proposed.

Background

Computing devices are increasingly incorporating one or more sensors to enhance end-user usage models. These include, but are not limited to: accelerometer, gyrometer, compass, and ambient light sensor.

Operating systems are beginning to support standardized APIs (application programming interfaces) to sensors, as examples: the Microsoft* Windows* 7 Sensor Framework, the MeeGo* Sensor Framework, and the Android* Sensor Framework. *[* indicates may be trademarks or registered trademarks of their respective companies.]*

Standardization of HID usages for sensors would allow (but not require) sensor hardware vendors to provide a consistent Plug And Play interface at the USB boundary, thereby enabling some operating systems to incorporate common device drivers that could be reused between vendors, alleviating any need for the vendors to provide the drivers themselves.

Approach

Because the Microsoft Windows 7 Sensor Framework has the most comprehensive set of defined sensors to date; the approach of this proposal is to begin with an equivalent set and add some reasonable extensions.

Table of Contents

Table of Tables

Table of Figures

1. Sensor Page (0x20)

This page provides usages for sensors. This section is Normative, meaning that it is the formal description of HID Usages for Sensors.

See Also

For background information about Sensors, including a Glossary and conceptual object model, please see Section 2.

For a discussion of how communication with Sensors is mapped onto HID mechanisms, please see Section 3.

For illustrative examples of HID Report Descriptors for various types of Sensors that incorporate these Usages, please see Section 4.

The Usage IDs are numerically segregated into sections for convenience.

- The lowest-numbered IDs from 0x00 to 0xFF are Usages applied to Collections and represent sensor objects (may equate to sensor *Categories* or *Types*).
- The IDs from 0x0100 to 0x07FF are Usages applied to *Properties* and *Data Fields*. These are grouped by the sensor *Category* where the Usages are commonly employed, but this arrangement is arbitrary. Usages may be reported by any sensor (or more than one sensor) if it makes sense to do so.
- The IDs from 0x0800 to 0x0FFF are *Selector* Usages used with *Properties* or *Data Fields* that are *Named Array* enumerations.
- The IDs from 0x1000 to 0xEFFF are *Properties* or *Data Fields* from the 0x0100 – 0x0FFF range with "Modifiers" OR-ed in to the top 4 bits.
- The IDs from 0xF000 upward are reserved for proprietary use by vendors.

Usage ID	Usage Name	Usage Type	Section
00	**Undefined**		
01	**Sensor**	CA,CP	1.1
02-0F	***Sensor: Reserved***		
	(for Properties commonly used with *all Sensors, please look at Usage range 0300 – 03ff)*		1.5
	(Data Field ***Timestamp*** *is also commonly used with all Sensors, and its Usage is 0529)*		1.15
10	**Biometric**	CA,CP	1.1,1.6
11	**Biometric: Human Presence**	CA,CP	1.1,1.6
12	**Biometric: Human Proximity**	CA,CP	1.1,1.6
13	**Biometric: Human Touch**	CA,CP	1.1,1.6
14-1F	***Biometric: Reserved***		
	(for Data Fields commonly used with Biometric sensors, please look at Usage range 04b0 – 04cf)		1.6
20	**Electrical**	CA,CP	1.1,1.7

21	**Electrical: Capacitance**	CA,CP	1.1,1.7
22	**Electrical: Current**	CA,CP	1.1,1.7
23	**Electrical: Power**	CA,CP	1.1,1.7
24	**Electrical: Inductance**	CA,CP	1.1,1.7
25	**Electrical: Resistance**	CA,CP	1.1,1.7
26	**Electrical: Voltage**	CA,CP	1.1,1.7
27	**Electrical: Potentiometer**	CA,CP	1.1,1.7
28	**Electrical: Frequency**	CA,CP	1.1,1.7
29	**Electrical: Period**	CA,CP	1.1,1.7
2A-2F	***Electrical: Reserved***		
	(for Data Fields commonly used with Electrical sensors, please look at Usage range 0500 – 051f)		1.7
30	**Environmental**	CA,CP	1.1,1.8
31	**Environmental: Atmospheric Pressure**	CA,CP	1.1,1.8
32	**Environmental: Humidity**	CA,CP	1.1,1.8
33	**Environmental: Temperature**	CA,CP	1.1,1.8
34	**Environmental: Wind Direction**	CA,CP	1.1,1.8
35	**Environmental: Wind Speed**	CA,CP	1.1,1.8
36-3F	***Environmental: Reserved***		
	(for Data Fields commonly used with Environmental sensors, please look at Usage range 0430 – 043f)		1.8
	(for Properties commonly used with Environmental sensors, please look at Usage range 0440 – 044f)		1.8
40	**Light**	CA,CP	1.1,1.9
41	**Light: Ambient Light**	CA,CP	1.1,1.9
42	**Light: Consumer Infrared**	CA,CP	1.1,1.9
43-4F	***Light: Reserved***		
	(for Data Fields commonly used with Light sensors, please look at Usage range 04d0 – 04ef)		1.9
	*(Property **Response Curve** is also commonly used with Light sensors, and its Usage is 0318)*		1.5
50	**Location**	CA,CP	1.1,1.10
51	**Location: Broadcast**	CA,CP	1.1,1.10
52	**Location: Dead Reckoning**	CA,CP	1.1,1.10
53	**Location: GPS (Global Positioning System)**	CA,CP	1.1,1.10
54	**Location: Lookup**	CA,CP	1.1,1.10
55	**Location: Other**	CA,CP	1.1,1.10
56	**Location: Static**	CA,CP	1.1,1.10
57	**Location: Triangulation**	CA,CP	1.1,1.10
58-5F	***Location: Reserved***		
	(for Data Fields commonly used with Location sensors, please look at Usage range 0400 – 0429)		1.10
	(for Properties commonly used with Location sensors, please look at Usage range 042a – 042f)		1.10
60	**Mechanical**	CA,CP	1.1,1.11
61	**Mechanical: Boolean Switch**	CA,CP	1.1,1.11
62	**Mechanical: Boolean Switch Array**	CA,CP	1.1,1.11
63	**Mechanical: Multivalue Switch**	CA,CP	1.1,1.11
64	**Mechanical: Force**	CA,CP	1.1,1.11
65	**Mechanical: Pressure**	CA,CP	1.1,1.11

66	**Mechanical: Strain**	CA,CP	1.1,1.11
67	**Mechanical: Weight**	CA,CP	1.1,1.11
68	**Mechanical: Haptic Vibrator**	CA,CP	1.1,1.11
69	**Mechanical: Hall Effect Switch**	CA,CP	1.1,1.11
6A-6F	***Mechanical: Reserved***		
	(for Data Fields commonly used with Mechanical sensors, please look at Usage range 0490 – 049f)		1.11
	(for Properties commonly used with Mechanical sensors, please look at Usage range 04a0 – 04af)		1.11
70	**Motion**	CA,CP	1.1,1.12
71	**Motion: Accelerometer 1D**	CA,CP	1.1,1.12
72	**Motion: Accelerometer 2D**	CA,CP	1.1,1.12
73	**Motion: Accelerometer 3D**	CA,CP	1.1,1.12
74	**Motion: Gyrometer 1D**	CA,CP	1.1,1.12
75	**Motion: Gyrometer 2D**	CA,CP	1.1,1.12
76	**Motion: Gyrometer 3D**	CA,CP	1.1,1.12
77	**Motion: Motion Detector**	CA,CP	1.1,1.12
78	**Motion: Speedometer**	CA,CP	1.1,1.12
79	**Motion: Accelerometer** *(any number of axes)*	CA,CP	1.1,1.12
7A	**Motion: Gyrometer** *(any number of axes)*	CA,CP	1.1,1.12
7B-7F	***Motion: Reserved***		
	(for Data Fields commonly used with Motion sensors, please look at Usage range 0450 – 046f)		1.12
80	**Orientation**	CA,CP	1.1,1.13
81	**Orientation: Compass 1D**	CA,CP	1.1,1.13
82	**Orientation: Compass 2D**	CA,CP	1.1,1.13
83	**Orientation: Compass 3D**	CA,CP	1.1,1.13
84	**Orientation: Inclinometer 1D**	CA,CP	1.1,1.13
85	**Orientation: Inclinometer 2D**	CA,CP	1.1,1.13
86	**Orientation: Inclinometer 3D**	CA,CP	1.1,1.13
87	**Orientation: Distance 1D**	CA,CP	1.1,1.13
88	**Orientation: Distance 2D**	CA,CP	1.1,1.13
89	**Orientation: Distance 3D**	CA,CP	1.1,1.13
8A	**Orientation: Device Orientation**	CA,CP	1.1,1.13
8B	**Orientation: Compass** *(any number of axes)*	CA,CP	1.1,1.13
8C	**Orientation: Inclinometer** *(any number of axes)*	CA,CP	1.1,1.13
8D	**Orientation: Distance** *(any number of axes)*	CA,CP	1.1,1.13
8E-8F	***Orientation: Reserved***		
	(for Data Fields commonly used with Orientation sensors, please look at Usage range 0470 – 048f)		1.13
90	**Scanner**	CA,CP	1.1,1.14
91	**Scanner: Barcode**	CA,CP	1.1,1.14
92	**Scanner: RFID**	CA,CP	1.1,1.14
93	**Scanner: NFC**	CA,CP	1.1,1.14
94-9F	***Scanner: Reserved***		
	(for Data Fields commonly used with Scanner sensors, please look at Usage range 04f0 – 04f7)		1.14
	(for Properties commonly used with Scanner sensors, please look at Usage range 04f8 – 04ff)		1.14
A0	**Time**	CA,CP	1.1,1.15

A1	**Time: Alarm Timer**	CA,CP	1.1,1.15
A2	**Time: Real Time Clock**	CA,CP	1.1,1.15
A3-AF	***Time: Reserved***		
	(for Data Fields commonly used with Time sensors, please look at Usage range 0520 – 052f)		1.15
	(for Properties commonly used with Time sensors, please look at Usage range 0530 – 053f)		1.15
B0-DF	***Reserved***		
E0	**Other**	CA,CP	1.1
E1	**Other: Custom**	CA,CP	1.1,1.16
E2	**Other: Generic**	CA,CP	1.1,1.17
E3	**Other: Generic Enumerator**	CA,CP	1.1,1.17
E4-EF	***Other: Reserved***		
	(for Data Fields commonly used with Custom sensors, please look at Usage range 0540 – 055f)		1.16
	(for Properties commonly used with Generic sensors, please look at Usage range 0560 – 057f)		1.17
F0-FF	***Reserved for Vendors/OEMs***	CA,CP	
	(for Vendor-reserved Data Fields and Properties commonly used with Vendor-reserved sensors, please use Usage range f000-ffff)		
	(Modifiers are Usage Switches used in conjunction with other Usages. The value of the Modifier is OR-ed in to the top 4 bits of the un-modified Usage ID)		
0	Modifier: None	US	1.2
1	Modifier: Change Sensitivity Absolute	US	1.2
2	Modifier: Maximum	US	1.2
3	Modifier: Minimum	US	1.2
4	Modifier: Accuracy	US	1.2
5	Modifier: Resolution	US	1.2
6	Modifier: Threshold High	US	1.2
7	Modifier: Threshold Low	US	1.2
8	Modifier: Calibration Offset	US	1.2
9	Modifier: Calibration Multiplier	US	1.2
A	Modifier: Report Interval	US	1.2
B	Modifier: Frequency Max	US	1.2
C	Modifier: Period Max	US	1.2
D	Modifier: Change Sensitivity Percent of Range	US	1.2
E	Modifier: Change Sensitivity Percent Relative	US	1.2
F	*Modifier: Reserved for Vendors/OEMs*	US	1.2
	(These Events are commonly used by all Sensors)		
0200	Event		
0201	Event: Sensor State	NAry	1.3
0800	Sensor State: Undefined	Sel	1.3
0801	Sensor State: Ready	Sel	1.3
0802	Sensor State: Not Available	Sel	1.3
0803	Sensor State: No Data	Sel	1.3
0804	Sensor State: Initializing	Sel	1.3
0805	Sensor State: Access Denied	Sel	1.3

0806	Sensor State: Error	Sel	1.3
0202	Event: Sensor Event	NAry	1.4
0810	Sensor Event: Unknown	Sel	1.4
0811	Sensor Event: State Changed	Sel	1.4
0812	Sensor Event: Property Changed	Sel	1.4
0813	Sensor Event: Data Updated	Sel	1.4
0814	Sensor Event: Poll Response	Sel	1.4
0815	Sensor Event: Change Sensitivity	Sel	1.4
0816	Sensor Event: Range Maximum Reached	Sel	1.4
0817	Sensor Event: Range Minimum Reached	Sel	1.4
0818	Sensor Event: High Threshold Cross Upward	Sel	1.4
0819	Sensor Event: High Threshold Cross Downward	Sel	1.4
081A	Sensor Event: Low Threshold Cross Upward	Sel	1.4
081B	Sensor Event: Low Threshold Cross Downward	Sel	1.4
081C	Sensor Event: Zero Threshold Cross Upward	Sel	1.4
081D	Sensor Event: Zero Threshold Cross Downward	Sel	1.4
081E	Sensor Event: Period Exceeded	Sel	1.4
081F	Sensor Event: Frequency Exceeded	Sel	1.4
0820	Sensor Event: Complex Trigger	Sel	1.4
0203-02FF	*Event: Reserved*		
	(These Properties are commonly used by all Sensors)		
0300	Property		1.5
0301	Property: Friendly Name	SV	1.5
0302	Property: Persistent Unique ID	DV	1.5
0303	Property: Sensor Status	DV	1.5
0304	Property: Minimum Report Interval *(default Unit: milliseconds)*	SV	1.5
0305	Property: Sensor Manufacturer	SV	1.5
0306	Property: Sensor Model	SV	1.5
0307	Property: Sensor Serial Number	SV	1.5
0308	Property: Sensor Description	SV	1.5
0309	Property: Sensor Connection Type	NAry	1.5
0830	Connection Type: PC Integrated	Sel	1.5
0831	Connection Type: PC Attached	Sel	1.5
0832	Connection Type: PC External	Sel	1.5
030A	Property: Sensor Device Path	DV	1.5
030B	Property: Hardware Revision	SV	1.5
030C	Property: Firmware Version	SV	1.5
030D	Property: Release Date	SV	1.5
030E	Property: Report Interval *(default Unit: milliseconds)*	DV	1.5
030F	Property: Change Sensitivity Absolute	DV	1.5
0310	Property: Change Sensitivity Percent of Range	DV	1.5
0311	Property: Change Sensitivity Percent Relative	DV	1.5
0312	Property: Accuracy	DV	1.5
0313	Property: Resolution	DV	1.5
0314	Property: Maximum	DV	1.5
0315	Property: Minimum	DV	1.5
0316	Property: Reporting State	NAry	1.5
0840	Reporting State: Report No Events	Sel	1.5
0841	Reporting State: Report All Events	Sel	1.5
0842	Reporting State: Report Threshold Events	Sel	1.5
0843	Reporting State: Wake On No Events	Sel	1.5

0844	Reporting State: Wake On All Events	Sel	1.5
0845	Reporting State: Wake On Threshold Events	Sel	1.5
0317	Property: Sampling Rate *(default Unit: milliseconds)*	DV	1.5
0318	Property: Response Curve	DV	1.5
0319	Property: Power State	NAry	1.5
0850	Power State: Undefined	Sel	1.5
0851	Power State: D0 Full Power	Sel	1.5
0852	Power State: D1 Low Power	Sel	1.5
0853	Power State: D2 Standby Power with Wakeup	Sel	1.5
0854	Power State: D3 Sleep with Wakeup	Sel	1.5
0855	Power State: D4 Power Off	Sel	1.5
031A-03FF	*Property: Reserved*		
	(These Data Fields are commonly used by Location sensors)		
0400	Data Field: Location	SV	1.10
0401	*Data Field: Location Reserved*		
0402	Data Field: Altitude Antenna Sea Level *(default Unit: meters)*	SV	1.10
0403	Data Field: Differential Reference Station ID	SV	1.10
0404	Data Field: Altitude Ellipsoid Error *(default Unit: meters)*	SV	1.10
0405	Data Field: Altitude Ellipsoid *(default Unit: meters)*	SV	1.10
0406	Data Field: Altitude Sea Level Error *(default Unit: meters)*	SV	1.10
0407	Data Field: Altitude Sea Level *(default Unit: meters)*	SV	1.10
0408	Data Field: Differential GPS Data Age *(default Unit: seconds)*	SV	1.10
0409	Data Field: Error Radius *(default Unit: meters)*	SV	1.10
040A	Data Field: Fix Quality	NAry	1.10
0870	Fix Quality: No Fix	Sel	1.10
0871	Fix Quality: GPS	Sel	1.10
0872	Fix Quality: DGPS	Sel	1.10
040B	Data Field: Fix Type	NAry	1.10
0880	Fix Type: No Fix	Sel	1.10
0881	Fix Type: GPS SPS Mode, Fix Valid	Sel	1.10
0882	Fix Type: DGPS SPS Mode, Fix Valid	Sel	1.10
0883	Fix Type: GPS PPS Mode, Fix Valid	Sel	1.10
0884	Fix Type: Real Time Kinematic	Sel	1.10
0885	Fix Type: Float RTK	Sel	1.10
0886	Fix Type: Estimated (dead reckoned)	Sel	1.10
0887	Fix Type: Manual Input Mode	Sel	1.10
0888	Fix Type: Simulator Mode	Sel	1.10
040C	Data Field: Geoidal Separation *(default Unit: meters)*	SV	1.10
040D	Data Field: GPS Operation Mode	NAry	1.10
0890	GPS Operation Mode: Manual	Sel	1.10
0891	GPS Operation Mode: Automatic	Sel	1.10
040E	Data Field: GPS Selection Mode	SV	1.10
08A0	GPS Selection Mode: Autonomous	Sel	1.10
08A1	GPS Selection Mode: DGPS	Sel	1.10
08A2	GPS Selection Mode: Estimated (dead reckoned)	Sel	1.10
08A3	GPS Selection Mode: Manual Input	Sel	1.10

08A4	GPS Selection Mode: Simulator	Sel	1.10
08A5	GPS Selection Mode: Data Not Valid	Sel	1.10
040F	Data Field: GPS Status	NAry	1.10
08B0	GPS Status: Data Valid	Sel	1.10
08B1	GPS Status: Data Not Valid	Sel	1.10
0410	Data Field: Position Dilution of Precision	SV	1.10
0411	Data Field: Horizontal Dilution of Precision	SV	1.10
0412	Data Field: Vertical Dilution of Precision	SV	1.10
0413	Data Field: Latitude *(default Unit: degrees)*	SV	1.10
0414	Data Field: Longitude *(default Unit: degrees)*	SV	1.10
0415	Data Field: True Heading *(default Unit: degrees)*	SV	1.10
0416	Data Field: Magnetic Heading *(default Unit: degrees)*	SV	1.10
0417	Data Field: Magnetic Variation *(default Unit: degrees)*	SV	1.10
0418	Data Field: Speed *(default Unit: knots)*	SV	1.10
0419	Data Field: Satellites in View	SV	1.10
041A	Data Field: Satellites in View Azimuth	SV	1.10
041B	Data Field: Satellites in View Elevation	SV	1.10
041C	Data Field: Satellites in View IDs	SV	1.10
041D	Data Field: Satellites in View PRNs	SV	1.10
041E	Data Field: Satellites in View S/N Ratios	SV	1.10
041F	Data Field: Satellites Used Count	SV	1.10
0420	Data Field: Satellites Used PRNs	SV	1.10
0421	Data Field: NMEA Sentence	SV	1.10
0422	Data Field: Address Line 1	SV	1.10
0423	Data Field: Address Line 2	SV	1.10
0424	Data Field: City	SV	1.10
0425	Data Field: State or Province	SV	1.10
0426	Data Field: Country or Region *(ISO 3166)*	SV	1.10
0427	Data Field: Postal Code	SV	1.10
0428-0429	*Data Field: Location Reserved*		1.10
	(These Properties are commonly used by Location sensors)		
042A	Property: Location		1.10
042B	Property: Location Desired Accuracy	NAry	1.10
0860	Accuracy: Default	Sel	1.10
0861	Accuracy: High	Sel	1.10
0862	Accuracy: Medium	Sel	1.10
0863	Accuracy: Low	Sel	1.10
042C-042F	*Property: Location Reserved*		
	(These Data Fields are commonly used by Environmental sensors)		
0430	Data Field: Environmental	SV	1.8
0431	Data Field: Atmospheric Pressure *(default Unit: bars)*	SV	1.8
0432	*Data Field: Reserved*		1.8
0433	Data Field: Relative Humidity *(percent)*	SV	1.8
0434	Data Field: Temperature *(default Unit: degrees Celsius)*	SV	1.8
0435	Data Field: Wind Direction *(default Unit: degrees)*	SV	1.8
0436	Data Field: Wind Speed *(default Unit:*	SV	1.8

	meters/second)		
0437-043F	*Data Field: Environmental Reserved*		
	(These Properties are commonly used by Environmental sensors)		
0440	Property: Environmental	SV	1.8
0441	Property: Reference Pressure *(default Unit: bars)*	SV	1.8
0442-044F	*Property: Environmental Reserved*		
	(These Data Fields are commonly used by Motion sensors)		
0450	Data Field: Motion	SV	1.12
0451	Data Field: Motion State	SF	1.12
0452	Data Field: Acceleration *(default Unit: G's)*	SV	1.12
0453	Data Field: Acceleration Axis X *(default Unit: G's)*	SV	1.12
0454	Data Field: Acceleration Axis Y *(default Unit: G's)*	SV	1.12
0455	Data Field: Acceleration Axis Z *(default Unit: G's)*	SV	1.12
0456	Data Field: Angular Velocity *(default Unit: degrees/second)*	SV	1.12
0457	Data Field: Angular Velocity about X Axis *(default Unit: degrees/second)*	SV	1.12
0458	Data Field: Angular Velocity about Y Axis *(default Unit: degrees/second)*	SV	1.12
0459	Data Field: Angular Velocity about Z Axis *(default Unit: degrees/second)*	SV	1.12
045A	Data Field: Angular Position *(default Unit: degrees)*	SV	1.12
045B	Data Field: Angular Position about X Axis *(default Unit: degrees)*	SV	1.12
045C	Data Field: Angular Position about Y Axis *(default Unit: degrees)*	SV	1.12
045D	Data Field: Angular Position about Z Axis *(default Unit: degrees)*	SV	1.12
045E	Data Field: Motion Speed *(default Unit: meters/second)*	SV	1.12
045F	Data Field: Motion Intensity *(percent)*	SV	1.12
0460-046F	*Data Field: Motion Reserved*		
	(These Data Fields are commonly used by Orientation sensors)		
0470	Data Field: Orientation	SV	1.13
0471	Data Field: Heading *(default Unit: degrees)*	SV	1.13
0472	Data Field: Heading X Axis *(default Unit: degrees)*	SV	1.13
4073	Data Field: Heading Y Axis *(default Unit: degrees)*	SV	1.13
0474	Data Field: Heading Z Axis *(default Unit: degrees)*	SV	1.13
0475	Data Field: Heading Compensated Magnetic North *(default Unit: degrees)*	SV	1.13
0476	Data Field: Heading Compensated True North *(default Unit: degrees)*	SV	1.13
0477	Data Field: Heading Magnetic North *(default Unit: degrees)*	SV	1.13
0478	Data Field: Heading True North *(default Unit: degrees)*	SV	1.13
0479	Data Field: Distance *(default Unit: meters)*	SV	1.13
047A	Data Field: Distance X Axis *(default Unit: meters)*	SV	1.13
047B	Data Field: Distance Y Axis *(default Unit: meters)*	SV	1.13

047C	Data Field: Distance Z Axis *(default Unit: meters)*	SV	1.13
047D	Data Field: Distance Out-of-Range	SF	1.13
047E	Data Field: Tilt *(default Unit: degrees)*	SV	1.13
047F	Data Field: Tilt X Axis *(default Unit: degrees)*	SV	1.13
0480	Data Field: Tilt Y Axis *(default Unit: degrees)*	SV	1.13
0481	Data Field: Tilt Z Axis *(default Unit: degrees)*	SV	1.13
0482	Data Field: Rotation Matrix	SV	1.13
0483	Data Field: Quaternion	SV	1.13
0484	Data Field: Magnetic Flux *(default Unit: milligauss)*	SV	1.13
0485	Data Field: Magnetic Flux X Axis *(default Unit: milligauss)*	SV	1.13
0486	Data Field: Magnetic Flux Y Axis *(default Unit: milligauss)*	SV	1.13
0487	Data Field: Magnetic Flux Z Axis *(default Unit: milligauss)*	SV	1.13
0488-048F	*Data Field: Orientation Reserved*		
	(These Data Fields are commonly used by Mechanical sensors)		
0490	Data Field: Mechanical	SV	1.11
0491	Data Field: Boolean Switch State	SF	1.11
0492	Data Field: Boolean Switch Array States	SV	1.11
0493	Data Field: Multivalue Switch Value	SV	1.11
0494	Data Field: Force *(default Unit: Newtons)*	SV	1.11
0495	Data Field: Absolute Pressure *(default Unit: Pascals)*	SV	1.11
0496	Data Field: Gauge Pressure *(default Unit: Pascals)*	SV	1.11
0497	Data Field: Strain *(percent)*	SV	1.11
0498	Data Field: Weight *(default Unit: kilograms)*	SV	1.11
0498-049F	*Data Field: Mechanical Reserved*		1.11
	(These Properties are commonly used by Mechanical sensors)		1.11
04A0	Property: Mechanical	DV	1.11
04A1	Property: Vibration State	DF	1.11
04A2	Property: Forward Vibration Speed *(percent)*	DV	1.11
04A3	Property: Backward Vibration Speed *(percent)*	DV	1.11
04A4-04AF	*Property: Mechanical Reserved*		
	(These Data Fields are commonly used by Biometric sensors)		
04B0	Data Field: Biometric	SV	1.6
04B1	Data Field: Human Presence	SF	1.6
04B2	Data Field: Human Proximity Range *(default Unit: meters)*	SV	1.6
04B3	Data Field: Human Proximity Out of Range	SF	1.6
04B4	Data Field: Human Touch State	SF	1.6
04B5-04CF	*Data Field: Biometric Reserved*		
	(These Data Fields are commonly used by Light sensors)		
04D0	Data Field: Light	SV	1.9
04D1	Data Field: Illuminance *(default Unit: Lux)*	SV	1.9
04D2	Data Field: Color Temperature *(default Unit:*	SV	1.9

	degrees Kelvin)		
04D3	Data Field: Chromaticity	SV	1.9
04D4	Data Field: Chromaticity X *(default Unit: CIE 1931 x)*	SV	1.9
04D5	Data Field: Chromaticity Y *(default Unit: CIE 1931 y)*	SV	1.9
04D6	Data Field: Consumer IR Sentence Receive	SV	1.9
04D7-04DF	*Data Field: Light Reserved*		
	(These Properties are commonly used by Light sensors)		
04E0	Property: Light	DV	1.9
04E1	Property: Consumer IR Sentence Send	DV	1.9
04E2-04EF	*Property: Light Reserved*		
	*(Property **Response Curve** is also commonly used with Light sensors; it is Usage 0318)*		1.5
	(These Data Fields are commonly used by Scanner sensors)		
04F0	Data Field: Scanner	SV	1.14
04F1	Data Field: RFID Tag 40 Bit	SV	1.14
04F2	Data Field: NFC Sentence Receive	SV	1.14
04F3-04F7	*Data Field: Scanner Reserved*		
	(These Properties are commonly used by Scanner sensors)		
04F8	Property: Scanner	SV	1.14
04F9	Property: NFC Sentence Send	SV	1.14
04FA-04FF	*Property: Scanner Reserved*		
	(These Data Fields are commonly used by Electrical sensors)		
0500	Data Field: Electrical	SV	1.7
0501	Data Field: Capacitance *(default Unit: Farads)*	SV	1.7
0502	Data Field: Current *(default Unit: Amperes)*	SV	1.7
0503	Data Field: Electrical Power *(default Unit: Watts)*	SV	1.7
0504	Data Field: Inductance *(default Unit: Henrys)*	SV	1.7
0505	Data Field: Resistance *(default Unit: Ohms)*	SV	1.7
0506	Data Field: Voltage *(default Unit: Volts)*	SV	1.7
0507	Data Field: Frequency *(default Unit: Hertz)*	SV	1.7
0508	Data Field: Period *(default Unit: milliseconds)*	SV	1.7
0509	Data Field: Percent of Range	SV	1.7
050A-051F	*Data Field: Electrical Reserved*		
	(These Data Fields are commonly used by Time sensors)		
0520	Data Field: Time	SV	1.15
0521	Data Field: Year	SV	1.15
0522	Data Field: Month	SV	1.15
0523	Data Field: Day	SV	1.15
0524	Data Field: Day of Week	NAry	1.15
08C0	Day of Week: Sunday	Sel	1.15
08C1	Day of Week: Monday	Sel	1.15
08C2	Day of Week: Tuesday	Sel	1.15
08C3	Day of Week: Wednesday	Sel	1.15

08C4	Day of Week: Thursday	Sel	1.15
08C5	Day of Week: Friday	Sel	1.15
08C6	Day of Week: Saturday	Sel	1.15
0525	Data Field: Hour	SV	1.15
0526	Data Field: Minute	SV	1.15
0527	Data Field: Second	SV	1.15
0528	Data Field: Millisecond	SV	1.15
0529	Data Field: Timestamp	SV	1.15
052A	Data Field: Julian Day of Year	SV	1.15
052A-052F	*Data Field: Time Reserved*		
	(These Properties are commonly used by Time sensors)		
0530	Property: Time	DV	1.15
0531	Property: Time Zone Offset from UTC *(default Unit: minutes)*	DV	1.15
0532	Property: Time Zone Name	DV	1.15
0533	Property: Daylight Savings Time Observed	DF	1.15
0534	Property: Time Trim Adjustment	DV	1.15
0535	Property: Arm Alarm	DF	1.15
0535-053F	*Property: Time Reserved*		
	(These Data Fields are commonly used by Custom sensors)		
0540	Data Field: Custom	SV	1.16
0541	Data Field: Custom Usage	SV	1.16
0542	Data Field: Custom Boolean Array	SV	1.16
0543	Data Field: Custom Value	SV	1.16
0544	Data Field: Custom Value 1	SV	1.16
0545	Data Field: Custom Value 2	SV	1.16
0546	Data Field: Custom Value 3	SV	1.16
0547	Data Field: Custom Value 4	SV	1.16
0548	Data Field: Custom Value 5	SV	1.16
0549	Data Field: Custom Value 6	SV	1.16
054A-055F	*Data Field: Custom Reserved*		
	(These Data Fields are commonly used by Generic sensors)		
0560	Data Field: Generic	SV	1.17
0561	Data Field: Generic GUID or PROPERTYKEY	SV	1.17
0562	Data Field: Generic Category GUID	SV	1.17
0563	Data Field: Generic Type GUID	SV	1.17
0564	Data Field: Generic Event PROPERTYKEY	SV	1.17
0565	Data Field: Generic Property PROPERTYKEY	SV	1.17
0566	Data Field: Generic Data Field PROPERTYKEY	SV	1.17
0567	Data Field: Generic Event	SV	1.17
0568	Data Field: Generic Property	SV	1.17
0569	Data Field: Generic Data Field	SV	1.17
056A	Data Field: Enumerator Table Row Index	SV	1.17
056B	Data Field: Enumerator Table Row Count	SV	1.17
056C	Data Field: Generic GUID or PROPERTYKEY kind	NAry	1.17
08D0	Kind: Category	Sel	1.17
08D1	Kind: Type	Sel	1.17
08D2	Kind: Event	Sel	1.17

08D3	Kind: Property	Sel	1.17
08D4	Kind: Data Field	Sel	1.17
056D	Data Field: Generic GUID	SV	1.17
056E	Data Field: Generic PROPERTYKEY	SV	1.17
056F	Data Field: Generic Top Level Collection ID	SV	1.17
0570	Data Field: Generic Report ID	SV	1.17
0571	Data Field: Generic Report Item Position Index	SV	1.17
0572	Data Field: Generic Firmware VARTYPE	NAry	1.17
0900	VT_NULL: Empty	Sel	1.17
0901	VT_BOOL: Boolean	Sel	1.17
0902	VT_UI1: Byte	Sel	1.17
0903	VT_I1: Character	Sel	1.17
0904	VT_UI2: Unsigned Short	Sel	1.17
0905	VT_I2: Short	Sel	1.17
0906	VT_UI4: Unsigned Long	Sel	1.17
0907	VT_I4: Long	Sel	1.17
0908	VT_UI8: Unsigned Long Long	Sel	1.17
0909	VT_I8: Long Long	Sel	1.17
090A	VT_R4: Float	Sel	1.17
090B	VT_R8: Double	Sel	1.17
090C	VT_WSTR: Wide String	Sel	1.17
090D	VT_STR: Narrow String	Sel	1.17
090E	VT_CLSID: Guid	Sel	1.17
090F	VT_VECTOR|VT_UI1: Opaque Structure	Sel	1.17
0910	VT_F16E0: HID 16-bit Float with Unit Exponent 0	Sel	1.17
0911	VT_F16E1: HID 16-bit Float with Unit Exponent 1	Sel	1.17
0912	VT_F16E2: HID 16-bit Float with Unit Exponent 2	Sel	1.17
0913	VT_F16E3: HID 16-bit Float with Unit Exponent 3	Sel	1.17
0914	VT_F16E4: HID 16-bit Float with Unit Exponent 4	Sel	1.17
0915	VT_F16E5: HID 16-bit Float with Unit Exponent 5	Sel	1.17
0916	VT_F16E6: HID 16-bit Float with Unit Exponent 6	Sel	1.17
0917	VT_F16E7: HID 16-bit Float with Unit Exponent 7	Sel	1.17
0918	VT_F16E8: HID 16-bit Float with Unit Exponent 8	Sel	1.17
0919	VT_F16E9: HID 16-bit Float with Unit Exponent 9	Sel	1.17
091A	VT_F16EA: HID 16-bit Float with Unit Exponent A	Sel	1.17
091B	VT_F16EB: HID 16-bit Float with Unit Exponent B	Sel	1.17
091C	VT_F16EC: HID 16-bit Float with Unit Exponent C	Sel	1.17
091D	VT_F16ED: HID 16-bit Float with Unit Exponent D	Sel	1.17
091E	VT_F16EE: HID 16-bit Float with Unit Exponent E	Sel	1.17
091F	VT_F16EF: HID 16-bit Float with Unit Exponent F	Sel	1.17
0920	VT_F32E0: HID 32-bit Float with Unit Exponent 0	Sel	1.17
0921	VT_F32E1: HID 32-bit Float with Unit Exponent 1	Sel	1.17
0922	VT_F32E2: HID 32-bit Float with Unit Exponent 2	Sel	1.17
0923	VT_F32E3: HID 32-bit Float with Unit Exponent 3	Sel	1.17
0924	VT_F32E4: HID 32-bit Float with Unit Exponent 4	Sel	1.17
0925	VT_F32E5: HID 32-bit Float with Unit Exponent 5	Sel	1.17
0926	VT_F32E6: HID 32-bit Float with Unit Exponent 6	Sel	1.17
0927	VT_F32E7: HID 32-bit Float with Unit Exponent 7	Sel	1.17
0928	VT_F32E8: HID 32-bit Float with Unit Exponent 8	Sel	1.17
0929	VT_F32E9: HID 32-bit Float with Unit Exponent 9	Sel	1.17
092A	VT_F32EA: HID 32-bit Float with Unit Exponent A	Sel	1.17
092B	VT_F32EB: HID 32-bit Float with Unit Exponent B	Sel	1.17
092C	VT_F32EC: HID 32-bit Float with Unit Exponent C	Sel	1.17

092D	VT_F32ED: HID 32-bit Float with Unit Exponent D	Sel	1.17
092E	VT_F32EE: HID 32-bit Float with Unit Exponent E	Sel	1.17
092F	VT_F32EF: HID 32-bit Float with Unit Exponent F	Sel	1.17
0573	Data Field: Generic Unit of Measure	NAry	1.17
0940	Unit: Not Specified	Sel	1.17
0941	Unit: Lux	Sel	1.17
0942	Unit: Degrees Kelvin	Sel	1.17
0943	Unit: Degrees Celsius	Sel	1.17
0944	Unit: Pascal	Sel	1.17
0945	Unit: Newton	Sel	1.17
0946	Unit: Meters/Second	Sel	1.17
0947	Unit: Kilogram	Sel	1.17
0948	Unit: Meter	Sel	1.17
0949	Unit: Meters/Second/Second	Sel	1.17
094A	Unit: Farad	Sel	1.17
094B	Unit: Ampere	Sel	1.17
094C	Unit: Watt	Sel	1.17
094D	Unit: Henry	Sel	1.17
094E	Unit: Ohm	Sel	1.17
094F	Unit: Volt	Sel	1.17
0950	Unit: Hertz	Sel	1.17
0951	Unit: Bar	Sel	1.17
0952	Unit: Degrees Anti-clockwise	Sel	1.17
0953	Unit: Degrees Clockwise	Sel	1.17
0954	Unit: Degrees	Sel	1.17
0955	Unit: Degrees/Second	Sel	1.17
0956	Unit: Degrees/Second/Second	Sel	1.17
0957	Unit: Knot	Sel	1.17
0958	Unit: Percent	Sel	1.17
0959	Unit: Second	Sel	1.17
095A	Unit: Millisecond	Sel	1.17
095B	Unit: G	Sel	1.17
095C	Unit: Bytes	Sel	1.17
095D	Unit: Milligauss	Sel	1.17
095E	Unit: Bits	Sel	1.17
0574	Data Field: Generic Unit Exponent	NAry	1.17
0970	Exponent 0: 1	Sel	1.17
0971	Exponent 1: 10	Sel	1.17
0972	Exponent 2: 100	Sel	1.17
0973	Exponent 3: 1 000	Sel	1.17
0974	Exponent 4: 10 000	Sel	1.17
0975	Exponent 5: 100 000	Sel	1.17
0976	Exponent 6: 1 000 000	Sel	1.17
0977	Exponent 7: 10 000 000	Sel	1.17
0978	Exponent 8: 0.00 000 001	Sel	1.17
0979	Exponent 9: 0.0 000 001	Sel	1.17
097A	Exponent A: 0.000 001	Sel	1.17
097B	Exponent B: 0.00 001	Sel	1.17
097C	Exponent C: 0.0 001	Sel	1.17
097D	Exponent D: 0.001	Sel	1.17
097E	Exponent E: 0.01	Sel	1.17
097F	Exponent F: 0.1	Sel	1.17
0575	Data Field: Generic Report Size	SV	1.17

0576	Data Field: Generic Report Count	SV	1.17
0577-057F	*Data Field: Generic Reserved*		
	(These Properties are commonly used by Generic sensors)		
0580	Property: Generic	DV	1.17
0581	Property: Enumerator Table Row Index	DV	1.17
0582	Property: Enumerator Table Row Count	SV	1.17
0583-058F	*Property: Generic Reserved*		
0590-07FF	*Reserved for future use as Data Fields and Properties*		
0800-0FF	*Reserved for use as Selection Values*		
1000-EFFF	*Reserved for use as "Data Fields with Modifiers"*		1.2
F000-FFFF	*Reserved for Vendors/OEMs*		

Table 1. HID Usages for Sensors, Properties, Data Fields, and Selection Values

1.1 Sensor Device Usages

Sensor	CA,CP – An application-level or physical collection that identifies a device that aggregates one or more sensors on one sensor board; for example, a sensor hub.
Biometric	CA,CP – An application-level or physical collection that identifies a device that detects biometric information.
Biometric: Human Presence	CA,CP – An application-level or physical collection that identifies a device that detects human presence (Boolean yes or no).
Biometric: Human Proximity	CA,CP – An application-level or physical collection that identifies a device that detects human proximity (range of values).
Biometric: Human Touch	CA,CP – An application-level or physical collection that identifies a device that registers human touch. *This is not to be confused with single-touch or multi-touch digitizers that provide finger position coordinates.*
Electrical	CA,CP – An application-level or physical collection that identifies a device that measures electrical information.
Electrical: Capacitance	CA,CP – An application-level or physical collection that identifies a device that measures electrical capacitance.
Electrical: Current	CA,CP – An application-level or physical collection that identifies a device that measures electrical current, such as an ammeter.
Electrical: Power	CA,CP – An application-level or physical collection that identifies a device that measures electrical power, such as a wattmeter.
Electrical: Inductance	CA,CP – An application-level or physical collection that identifies a device that measures electrical inductance.
Electrical: Resistance	CA,CP – An application-level or physical collection that identifies a device that measures electrical resistance, such as an ohmmeter or a potentiometer.
Electrical: Voltage	CA,CP – An application-level or physical collection that identifies a device that measures electrical voltage, such as a voltmeter.

Electrical: Potentiometer	CA,CP – An application-level or physical collection that identifies a device that measures percent of range, such as a potentiometer.
Electrical: Frequency	CA,CP – An application-level or physical collection that identifies a device that measures electrical frequency, such as a frequency meter.
Electrical: Period	CA,CP – An application-level or physical collection that identifies a device that measures electrical period, such as a period meter.
Environmental	CA,CP – An application-level or physical collection that identifies a device that measures environmental information.
Environmental: Atmospheric Pressure	CA,CP – An application-level or physical collection that identifies a device that measures atmospheric pressure, such as a barometer.
Environmental: Humidity	CA,CP – An application-level or physical collection that identifies a device that measures humidity, such as a hygrometer.
Environmental: Temperature	CA,CP – An application-level or physical collection that identifies a device that measures temperature, such as a thermometer or a thermocouple.
Environmental: Wind Direction	CA,CP – An application-level or physical collection that identifies a device that measures wind direction, such as a weather vane.
Environmental: Wind Speed	CA,CP – An application-level or physical collection that identifies a device that measures wind speed, such as an anemometer.
Light	CA,CP – An application-level or physical collection that identifies a device that measures light information.
Light: Ambient Light	CA,CP – An application-level or physical collection that identifies a device that detects ambient light.
Light: Consumer Infrared	CA,CP – An application-level or physical collection that identifies a device that can transmit and receive Consumer Infrared signals, e.g., for controlling TVs and stereo equipment.
Location	CA,CP – An application-level or physical collection that identifies a device that can report location information.
Location: Broadcast	CA,CP – An application-level or physical collection that identifies a device that detect location information using transmissions such as television or radio frequencies (for example, cellular telephone).
Location: Dead Reckoning	CA,CP – An application-level or physical collection that identifies a virtual device that calculates the current location using aggregated motion data from multiple physical sensors (such as GPS, accelerometer, gyro, compass, altimeter).
Location: GPS	CA,CP – An application-level or physical collection that identifies a device that detects the current location using the GPS (Global Positioning Satellite) system.
Location: Lookup	CA,CP – An application-level or physical collection that identifies a device that detects the current location using the computers current IP Address.
Location: Other	CA,CP – An application-level or physical collection that identifies a device that detects the current location using other means.
Location: Static	CA,CP – An application-level or physical collection that identifies a device that use end-user provided information such as Civic

	Address to report the current location.
Location: Triangulation	CA,CP – An application-level or physical collection that identifies a device that detects the current location using triangulation techniques, such as cellular phone tower proximities.
Mechanical	CA,CP – An application-level or physical collection that identifies a device that can report mechanical information.
Mechanical: Boolean Switch	CA,CP – An application-level or physical collection that identifies a device that can switch between two states: on and off.
Mechanical: Boolean Switch Array	CA,CP – An application-level or physical collection that identifies an array of devices each of which can switch between two states: on and off.
Mechanical: Multivalue Switch	CA,CP – An application-level or physical collection that identifies a device that can switch between multiple states.
Mechanical: Force	CA,CP – An application-level or physical collection that identifies a device that measures force.
Mechanical: Pressure	CA,CP – An application-level or physical collection that identifies a device that measures pressure.
Mechanical: Strain	CA,CP – An application-level or physical collection that identifies a device that measures strain.
Mechanical: Weight	CA,CP – An application-level or physical collection that identifies a device that measures weight.
Mechanical: Haptic Vibrator	CA,CP – An application-level or physical collection that identifies a vibrator device that can provide Haptic feedback.
Mechanical: Hall Effect Switch	CA,CP – An application-level or physical collection that identifies a Hall Effect (magnetic proximity) detector switch.
Motion	CA,CP – An application-level or physical collection that identifies a device that measures motion information.
Motion: Accelerometer	CA,CP – An application-level or physical collection that identifies a device that measures linear acceleration along any number of axes.
Motion: Accelerometer 1D	CA,CP – An application-level or physical collection that identifies a device that measures linear acceleration along 1 axis.
Motion: Accelerometer 2D	CA,CP – An application-level or physical collection that identifies a device that measures linear acceleration along 2 axes.
Motion: Accelerometer 3D	CA,CP – An application-level or physical collection that identifies a device that measures linear acceleration along 3 axes.
Motion: Gyrometer	CA,CP – An application-level or physical collection that identifies a device that measures angular acceleration or velocity about any number of axes.
Motion: Gyrometer 1D	CA,CP – An application-level or physical collection that identifies a device that measures angular acceleration or velocity about 1 axis.
Motion: Gyrometer 2D	CA,CP – An application-level or physical collection that identifies a device that measures angular acceleration or velocity about 2 axes.

Motion: Gyrometer 3D	CA,CP – An application-level or physical collection that identifies a device that measures angular acceleration or velocity about 3 axes.
Motion: Motion Detector	CA,CP – An application-level or physical collection that identifies a device that detects motion (Boolean yes or no).
Motion: Speedometer	CA,CP – An application-level or physical collection that identifies a device that measures velocity.
Orientation	CA,CP – An application-level or physical collection that identifies a device that measures orientation information.
Orientation: Compass	CA,CP – An application-level or physical collection that identifies a compass with any number of axes.
Orientation: Compass 1D	CA,CP – An application-level or physical collection that identifies a one-axis compass.
Orientation: Compass 2D	CA,CP – An application-level or physical collection that identifies a two-axis compass.
Orientation: Compass 3D	CA,CP – An application-level or physical collection that identifies a three-axis compass.
Orientation: Inclinometer	CA,CP – An application-level or physical collection that identifies a tilt meter with any number of axes.
Orientation: Inclinometer 1D	CA,CP – An application-level or physical collection that identifies a one-axis tilt meter.
Orientation: Inclinometer 2D	CA,CP – An application-level or physical collection that identifies a two-axis tilt meter.
Orientation: Inclinometer 3D	CA,CP – An application-level or physical collection that identifies a three-axis tilt meter.
Orientation: Distance	CA,CP – An application-level or physical collection that identifies a device that measures distance using any number of axes.
Orientation: Distance 1D	CA,CP – An application-level or physical collection that identifies a device that measures distance using one axis.
Orientation: Distance 2D	CA,CP – An application-level or physical collection that identifies a device that measures distance using two axes.
Orientation: Distance 3D	CA,CP – An application-level or physical collection that identifies a device that measures distance using three axes.
Orientation: Device Orientation	CA,CP – An application-level or physical collection that identifies a device that measures device orientation in three axes.
Scanner	CA,CP – An application-level or physical collection that identifies a device that reports information from scanning devices.
Scanner: Barcode	CA,CP – An application-level or physical collection that identifies a device that is used for optical scanning of bar codes. It is strongly recommended that barcode scanners report input data and symbology information using the previously defined *HID Point of Sale Usage Tables* specification (*Reference Document [4])* and its associated HID Usage Page 0x8C.
Scanner: RFID	CA,CP – An application-level or physical collection that identifies a device that is used for radio-frequency scanning of tags.

Scanner: NFC	CA,CP – An application-level or physical collection that identifies a Near-Field Communication reader device. Such a device can communicate with other NFC-enabled devices over short distances. Some NFC devices are also able to read RFID tags.
Time	CA,CP – An application-level or physical collection that identifies a device that can report time, such as a typical RTC (Real Time Clock) / Time of Day Clock.
Time: Alarm Timer	CA,CP – An application-level or physical collection that identifies a device that can report information at a particular time or after a certain amount of time has passed.
Time: Real Time Clock	CA,CP – An application-level or physical collection that identifies a device that can report current time, most often used for timestamping sensor samples.
Other	CA,CP – An application-level or physical collection that identifies a device that does not fit into any of the other pre-defined categories.
Other: Custom	CA,CP – An application-level or physical collection that identifies a device that conforms to the "custom" sensor specification (*see Section 4.2.6*).
Other: Generic	CA,CP – An application-level or physical collection that identifies a device that conforms to the "generic" sensor specification (*see Section 4.2.7*).
Other: Generic Enumerator	CA,CP – An application-level or physical collection that identifies a device that conforms to the "generic enumerator" specification (*see Section 4.2.7.1*).

1.2 Sensor Field Usages: Modifiers

These fields are optionally supported by all sensors. The meaning is common for all sensors. Modifiers are used to change the meaning of a data field. This permits a single data field to take on some number of additional meanings depending on the usage of that data field.

The modifier is used to change the meaning of a data field as follows:

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Modifier				Data Field Usage											

Table 2. Modifiers composed as the top 4 bits of Data Field Usage

Modifier: None	US – The information contained in the data field is the unmodified meaning for that data field.
Modifier: Change Sensitivity Absolute	US – Specifies the change sensitivity set for a particular data field. Units are the same as the data field being modified. For example, if the data field is "Temperature, Degrees Celsius", and the absolute sensitivity is "3" then that would mean "change of ±3 Degrees Celsius".
Modifier: Maximum	US – The information contained in the data field is the maximum value for that data field.

Modifier: Minimum	US – The information contained in the data field is the minimum value for that data field.
Modifier: Accuracy	US – The information contained in the data field specifies the absolute accuracy with which that data field is reported.
Modifier: Resolution	US – The information contained in the data field specifies the absolute precision with which that data field is reported.
Modifier: Threshold High	US – The information contained in the data field is the high threshold value for that data field.
Modifier: Threshold Low	US – The information contained in the data field is the low threshold value for that data field.
Modifier: Calibration Offset	The information contained in the data field specifies the calibration offset applied to the data normally reported in that data field.
Modifier: Calibration Multiplier	The information contained in the data field specifies the calibration multiplier applied to the data normally reported in that data field.
Modifier: Report Interval	US – Specifies the Report Interval set for a particular data field.
Modifier: Frequency Max	US – Specifies the maximum frequency for a particular data field. Usually used as a time oriented threshold to indicate an event has occurred more often than required.
Modifier: Period Max	US – Specifies the maximum period for a particular data field. Usually used as a maximum threshold to indicate an event has not occurred.
Modifier: Change Sensitivity Percent of Range	US – Specifies the change sensitivity set for a particular data field. Units are a percentage of the Minimum to Maximum range. For example, if the data field is “Temperature, Degrees Celsius”, the Minimum is -4.0, the Maximum is +40.0, and the percent of range sensitivity is “5” then that would mean “change of 5% of -4.0 to +40.0 Degrees Celsius”, (i.e., ±2.2 Degrees Celsius).
Modifier: Change Sensitivity Percent Relative	US – Specifies the change sensitivity set for a particular data field. Units are a percentage of the “prior reading”. For example, if the data field is “Temperature, Degrees Celsius”, the prior reading was +24.0, and the percent relative sensitivity is “4” then that would mean “change of 4% from 24.0 Degrees Celsius”, (i.e., ±0.96 Degrees Celsius).

These fields are optionally supported by all sensors. The meaning is common for all sensors.

1.3 Sensor Field Usages: States

These fields are optionally supported by all sensors. The meaning is common for all sensors.

The sensor state field is usually part of the Input report Event and indicates the current state of the sensor.

Sensor State	DV – Specifies a sensor state as determined by the table below.

The Selection Usages (0800-0806) are used to select one enumeration value for the Sensor State field (0201) to indicate the current sensor state.

Sel Usage	State Name	Comment
0800	Unknown	The sensor state is unknown
0801	Not Available	The sensor not available
0802	Ready	Sensor is able to provide new complete and accurate data
0803	No Data	The sensor is available, but is not yet providing data. It is not known in what timeframe data will, if ever, be provided
0804	Initializing	The sensor is available, but is not yet providing data due to initialization activities. It is expected the sensor will provide data, but the timeframe in which that data will be available is not know
0805	Access Denied	In the case where an ID must be provided to access sensor data, and the requester fails to match the ID, this state will be returned
0806	Error	The sensor has encountered a major error. The sensor may recover from the state, but the time frame for recovery is unknown
0807-080F	*Reserved for future use*	*Reserved for future use*

Table 3. Selection Values for Sensor State Usage

1.4 Sensor Field Usages: Events

These fields are optionally supported by all sensors. The meaning is common for all sensors.

The sensor event field us usually part of the Input report Event and indicate the reason for the receipt of the input report.

Sensor Event	DV – Specifies a sensor event as determined by the table below.

These Selection Usages (0810-0820) are used to select one enumeration value for the Sensor Event field (0202) to indicate the sensor event.

Sel Usage	Event Name	Comment
0810	Unknown	The sensor event type is not known
0811	State Changed	The sensor state as specified in (20.5) has changed
0812	Property Changed	A property value has changed
0813	Data Updated	A data field has changed
0814	Poll Response	The most current sensor data is being returned as the result of a poll request (Get Input)
0815	Change Sensitivity	The change sensitivity has been exceeded for a data field
0816	Max Reached	The maximum for a data field has been reached
0817	Min Reached	The minimum for a data field has been reached
0818	High Threshold Cross Above	The high threshold set for a data field has been crossed to above the threshold from below the threshold
0819	High Threshold Cross Below	The high threshold set for a data field has been crossed to below the threshold from above the threshold
081A	Low Threshold Cross Above	The low threshold set for a data field has been crossed to above the threshold from below the

		threshold
081B	Low Threshold Cross Below	The low threshold set for a data field has been crossed to below the threshold from above the threshold
081C	Zero Threshold Cross Above	The zero point for a data field has been crossed to above the zero point from at or below the zero point
081D	Zero Threshold Cross Below	The zero point for a data field has been touched from above the zero point
081E	Period Exceeded	The maximum period set for a data field has been exceeded
081F	Frequency Exceeded	The maximum frequency set for a data field has been exceeded
0820	Complex Event	A complex combination of vendor-defined circumstances has occurred
0821-082F	*Reserved for future use*	*Reserved for future use*

Table 4. Selection Values for Sensor Event Usage

1.5 Sensor Field Usages: Properties

These fields are optionally supported by all sensors. The meaning is common for all sensors.

Friendly Name	SV – Specifies a textual string name of the device in a human-friendly wording.
Persistent Unique ID	DV – Uniquely identifies the device instance with which the sensor is associated. You can use this to tell apart multiple identical sensors attached to the same computer. Typically this value will be either dynamically stored by the operating system into the USB device shortly after reset/power-up or assigned by the manufacturer at the time the device is manufactured.
Sensor Status	DV – Specifies the current sensor status, as defined by the implementer. Not to be confused with the **Sensor State** *Data Field* that has standardized enumeration values.
Minimum Report Interval	SV – Specifies the minimum allowed elapsed time for periodic sensor Input Report generation. Default unit of measure is milliseconds; can be overridden using explicit Unit and/or Unit Exponent.
Sensor Manufacturer	SV – Specifies a textual string name of the manufacturer of a sensor device. For USB-based sensor devices, this may be the same as the MANUFACTURER USB String Descriptor, but could differ in two cases: (1) when a vendor manufactures a sensor module that incorporates a sensor chip from a third-party manufacturer; or (2) when a vendor manufactures a sensor hub that contains an aggregation of sensors from one or more other manufacturers.
Sensor Model	SV – Specifies a textual string name of the model of a sensor device. For USB-based sensor devices, this may be the same as the PRODUCT USB String Descriptor, but could differ in two cases: (1) when a vendor manufactures a sensor module that incorporates a sensor chip from a third-party manufacturer; or (2) when a vendor manufactures a sensor hub that contains an aggregation of sensors from one or more other manufacturers.

Sensor Serial Number	SV – Specifies a textual string name of the Serial Number ID of a sensor device. For USB-based sensor devices, this may be the same as the SERIAL NUMBER USB String Descriptor, but could differ in two cases: (1) when a vendor manufactures a sensor module that incorporates a sensor chip from a third-party manufacturer; or (2) when a vendor manufactures a sensor hub that contains an aggregation of sensors from one or more other manufacturers.
Sensor Description	SV – Specifies a textual string description of the sensor function.
Sensor Connection Type	NAry – Specifies the current connection type: • Sel – PC Integrated= integrated inside the computer, • Sel – PC Attached = attached to the computer through a peripheral device (for example, with a special docking connector), • Sel – PC External = connected by means of an external interface such as a network connection. USB HID sensor devices should usually be type 0 or type 1.
Sensor Device Path	DV – Uniquely identifies the device instance with which the sensor is associated. You can use this to tell apart multiple identical sensors attached to the same computer. Typically this value will be dynamically stored by the operating system into the USB device shortly after reset/power-up.
Hardware Revision	SV – Specifies a textual string name of the hardware revision of a sensor device.
Firmware Version	SV – Specifies a textual string name of the firmware version of a sensor device.
Release Date	SV – Specifies a textual string name of the release date of a sensor device.
Report Interval	DV – Specifies the elapsed time for periodic sensor Input Report generation, in milliseconds. A value of 0 means “set/use device default value”, not 0 milliseconds.
Change Sensitivity Absolute	DV – Specifies the absolute amount that by which a data field should change before an event (such as an asynchronous Input Report) is generated. Absolute sensitivity values are expressed using the same units as the corresponding data field, unless otherwise documented. This form of change sensitivity usually applies to all related data fields rather than to individual data fields.
Change Sensitivity percent of Range	DV – Specifies the percent relative to the overall range of a data field that a data field should change before an event (such as an asynchronous Input Report) is generated. Percent of range Sensitivity values are expressed in percent of range, with range typically being the maximum value minus the minimum value of the data field when expressed as absolute values. This form of change sensitivity usually applies to all related data fields rather than to individual data fields.
Change Sensitivity relative percent	DV – Specifies the percent relative to the current sensor absolute value of a data field that a data field should change before an event (such as an asynchronous Input Report) is generated. This form of

	change sensitivity usually applies to all related data fields rather than to individual data fields.
Sensor Accuracy	DV – Specifies the accuracy of sensor values by representing possible variation from true values. Accuracy values are expressed using the same units as the corresponding data field, except when otherwise documented. This form of accuracy a usually applies to all related data fields rather than to individual data fields.
Sensor Resolution	DV – Resolution represents sensitivity to change in the data field. Resolution values are expressed by using the same units as the data field, except when otherwise documented.
Range Maximum	DV – Specifies the maximum value that can be produced by the sensor. Range maximum is expressed using the same units as the corresponding data field unless otherwise documented. This form of Range Maximum usually applies to all related data fields rather than individual data fields.
Range Minimum	DV – Specifies the minimum value that can be produced by the sensor. Range minimum is expressed using the same units as the corresponding data field unless otherwise documented. This form of Range Minimum usually applies to all related data fields rather than individual data fields.
Reporting State	NAry – Indicates the current reporting state of the sensor. The reporting state may be: • Sel – Report No Events = no asynchronous Input reports are sent, • Sel – Report All Events = all Input reports are sent without any filtering, • Sel – Report Threshold Events = Input reports are sent only when it exceeds a pre-programmed threshold, • Sel – Wake On No Events = no asynchronous Input reports are sent and a Wake On event is never performed, • Sel – Wake On All Events = all Input reports are sent without any filtering and a Wake On event is performed, • Sel – Wake On Threshold Events = Input reports are sent only when it exceeds a pre-programmed threshold and a Wake On event is performed. - *See* **Power State** *to see interactions with Wake On capability*
Sampling Rate	DV – Sampling rate indicates the rate at which the sensor is physically sampled. This is not necessarily the same as the rate at which samples are reported using asynchronous Input reports. Default unit of measure is milliseconds; can be overridden using explicit Unit and/or Unit Exponent.
Response Curve	DV – Reports pairs of values that provide a mapping between value levels and desired output.
Power State	NAry – Indicates the current power state of the sensor. The power state may be:

	• Sel – Undefined = the device power state is currently unknown or undefined, • Sel – D0 Full Power = the device is in full power operation, • Sel – D1 Low Power = the device is in a low power operation mode, • Sel – D2 Standby Power with Wakeup = the device is at a standby power mode (e.g., halted and awaiting interrupts) and can be awakened, • Sel – D3 Sleep with Wakeup = the device is in a sleep mode and can be awakened, • Sel – D4 Power Off = the device is completely powered off and cannot be awakened. - *See* **Reporting State** *for valid Wake On settings*

1.6 Biometric Sensor Field Usages

These fields are commonly supported by biometric sensors.

Human Presence	SF – TRUE when a human is using the computer, otherwise FALSE.
Human Proximity Range	SV – Distance between a human and the computer. Default unit of measure is meters; can be overridden using explicit Unit and/or Unit Exponent.
Human Proximity Out-of-Range	SF – TRUE when the sensor measuring human proximity range indicates "out of range" meaning the value provided as Human Proximity Range may not be accurate.
Human Touch State	SF – TRUE when the touch sensor is being touched, otherwise FALSE. *This is not to be confused with single-touch or multi-touch digitizers that provide finger position coordinates.*

See Also
For more information about HID single-touch or multi-touch digitizers; please refer to *HUTTR30 Addition of usages related to touch digitizers* (*Reference Document [5]*) and *HUTTR34 Addition of usages related to multi-touch digitizers* (*Reference Document [6]*).

1.7 Electrical Sensor Field Usages

These fields are commonly supported by electrical sensors.

Capacitance	SV – Measures electrical capacitance. Default unit of measure is Farads; can be overridden using explicit Unit and/or Unit Exponent.
Current	SV – Measures electrical current. Default unit of measure is Amperes; can be overridden using explicit Unit and/or Unit Exponent.
Electrical Power	SV – Measures electrical power. Default unit of measure is Watts;

	can be overridden using explicit Unit and/or Unit Exponent.
Inductance	SV – Measures electrical inductance. Default unit of measure is Henrys; can be overridden using explicit Unit and/or Unit Exponent.
Resistance	SV – Measures electrical resistance. Default unit of measure is Ohms; can be overridden using explicit Unit and/or Unit Exponent.
Voltage	SV – Measures electrical voltage. Default unit of measure is Volts; can be overridden using explicit Unit and/or Unit Exponent.
Frequency	SV – Measures electrical frequency. Default unit of measure is Hertz; can be overridden using explicit Unit and/or Unit Exponent.
Period	SV – Measures electrical period. Default unit of measure is milliseconds; can be overridden using explicit Unit and/or Unit Exponent.
Percent of Range	SV – Measures the percent of range provided by a value, such as the position of a potentiometer with respect to the overall physical range of that potentiometer. Can be scaled with a Unit Exponent.

1.8 Environmental Sensor Field Usages

These fields are commonly supported by environmental sensors.

Atmospheric Pressure	SV – Measures atmospheric pressure. Default unit of measure is bars; can be overridden using explicit Unit and/or Unit Exponent.
Reference Pressure	DV – Specifies reference atmospheric pressure at sea level, nominally “1976 US Standard Atmosphere” air pressure at Sea Level Pressure. Default unit of measure is bars; can be overridden using explicit Unit and/or Unit Exponent.
Relative Humidity Percent	SV – Measures relative humidity as a percentage.
Temperature	SV – Measures temperature. Default unit of measure is degrees Celsius; can be overridden using explicit Unit and/or Unit Exponent.
Wind Direction	SV – Measures wind direction relative to magnetic north. Default unit of measure is degrees; can be overridden using explicit Unit and/or Unit Exponent.
Wind Speed	SV – Measures wind speed. Default unit of measure is meters/second; can be overridden using explicit Unit and/or Unit Exponent.

1.9 Light Sensor Field Usages

These fields are commonly supported by light sensors.

Illuminance	SV – Measures illuminance (light level, i.e., luminance per square area). Default unit of measure is Lux; can be overridden using explicit Unit and/or Unit Exponent.

Color Temperature	SV – Measures the color temperature. Default unit of measure is degrees Kelvin; can be overridden using explicit Unit and/or Unit Exponent.
Chromaticity	SV – Chromaticity without respect to which axis it occurs in. This is usually used as a composite value for specifying min, max and accuracy for chromaticity.
Chromaticity X	SV – Measures chromaticity in the X axis as defined by the CIE 1931 specification. Can be scaled with a Unit Exponent.
Chromaticity Y	SV – Measures chromaticity in the Y axis as defined by the CIE 1931 specification. Can be scaled with a Unit Exponent.
Consumer IR Sentence Receive	SV – Data message received from a Consumer Infrared controller. Data type is an opaque counted array of bytes; interpretation will depend on host-based middleware.
Consumer IR Sentence Send	DV – Data message sent to a Consumer Infrared controller. Data type is an opaque counted array of bytes; interpretation will depend on host-based middleware.

1.10 Location Sensor Field Usages

These fields are commonly supported by location sensors.

Location Desired Accuracy	NAry – Indicates the type of accuracy handling desired by a client application: • Sel – Default = indicates that the sensor should use its own default accuracy policy, • Sel – High = indicates that the sensor should optimize for the most accurate location report possible, even if it consumes more energy, costs more money, or uses more connection bandwidth, • Sel – Medium = indicates that the sensor should strike a balance between accuracy and power consumption, • Sel – Low = indicates that the sensor should reduce accuracy thereby optimizing for power utilization.
Altitude Antenna Sealevel	SV – Indicates altitude of the antenna, references to sea level. Default unit of measure is meters; can be overridden using explicit Unit and/or Unit Exponent.
Deferential Reference Station ID	SV – Indicates ID of the differential reference station. The range is 0000 to 1023.
Altitude Ellipsoid Error	SV – Indicates altitude error referenced to the World Geodetic System (WGS 84) reference ellipsoid. Default unit of measure is meters; can be overridden using explicit Unit and/or Unit Exponent.
Altitude Ellipsoid	SV – Indicates altitude referenced to the World Geodetic System (WGS 84) reference ellipsoid. Default unit of measure is meters; can be overridden using explicit Unit and/or Unit Exponent.
Altitude Sealevel Error	SV – Indicates altitude error referenced to sea level. Default unit of measure is meters; can be overridden using explicit Unit and/or

	Unit Exponent.
Altitude Sealevel	SV – Indicates altitude referenced to sea level. Default unit of measure is meters; can be overridden using explicit Unit and/or Unit Exponent.
DPGS Data Age	SV – Indicates age of differential GPS data. Default unit of measure is seconds; can be overridden using explicit Unit and/or Unit Exponent.
Error Radius	SV – Indicates accuracy of latitude and longitude values. Default unit of measure is meters; can be overridden using explicit Unit and/or Unit Exponent. A value of 0 means that the accuracy level is not currently known.
Fix Quality	NAry – Indicates fix quality: • Sel – No Fix, • Sel – GPS, • Sel – DPGS.
Fix Type	NAry – Indicates fix type: • Sel – No Fix, • Sel – GPS SPS Mode, Fix Valid, • Sel – DGPS SPS Mode, Fix Valid, • Sel – GPS PPS Mode, Fix Valid, • Sel – Real Time Kinematic, • Sel – Float RTK, • Sel – Estimated (dead reckoned), • Sel – Manual Input Mode, • Sel – Simulator Mode.
Geoidal Separation	SV – Indicates the difference between the World Geodetic System (WGS 84) ellipsoid and mean sea level. Values less than zero indicate that mean sea level is below the reference ellipsoid. Default unit of measure is meters; can be overridden using explicit Unit and/or Unit Exponent.
GPS Operation Mode	NAry – Indicates GPS operation mode: • Sel – Manual = Manually set for 2D or 3D mode , • Sel – Automatic = Automatically can switch between 2D and 3D modes.
GPS Selection Mode	NAry – Indicates GPS selection mode: • Sel – Autonomous, • Sel – DGPS, • Sel – Estimated (dead reckoned), • Sel – Manual Input, • Sel – Simulator, • Sel – Data Not Valid.
GPS Status	NAry – Indicates current GPS data status: • Sel – Data Valid,

	• Sel – Data Not Valid.
Position Dilution of Precision	SV – Indicates the position dilution of precision.
Horizontal Dilution of Precision	SV – Indicates the horizontal dilution of precision.
Vertical Dilution of Precision	SV – Indicates the vertical dilution of precision.
Latitude	SV – Indicates the latitude. Default unit of measure is degrees; can be overridden using explicit Unit and/or Unit Exponent. North is positive; South is negative.
Longitude	SV – Indicates longitude. Default unit of measure is degrees; can be overridden using explicit Unit and/or Unit Exponent. East is positive; West is negative.
True Heading	SV – Indicates the current heading in relation to true north. Default unit of measure is degrees; can be overridden using explicit Unit and/or Unit Exponent.
Magnetic Heading	SV – Indicates the heading in relation to magnetic north. Default unit of measure is degrees; can be overridden using explicit Unit and/or Unit Exponent.
Magnetic Variation	SV – Indicates the magnetic variation from true north. East is positive; West is negative. Default unit of measure is degrees; can be overridden using explicit Unit and/or Unit Exponent.
Speed	SV – Indicates the current speed. Default unit of measure is knots; can be overridden using explicit Unit and/or Unit Exponent.
Satellites in View	SV – Indicates the number of GPS satellites currently in view.
Satellites in View Azimuth	SV – Indicates the azimuth of GPS satellites currently in view.
Satellites in View Elevation	SV – Indicates the elevation of GPS satellites currently in view.
Satellites in View IDs	SV – Indicates the ID of GPS satellites currently in view.
Satellites in View PRNs	SV – Indicates the Pseudo-Random Noise codes of GPS satellites currently in view.
Satellites in View S/N Ratios	DV – Indicates the Signal-to-Noise Ratio of GPS satellites currently in view.
Satellites Used Count	SV – Indicates the number of GPS satellites that are currently being used to calculate a location solution.
Satellites Used PRNs	SV – Indicates the Pseudo-Random Noise codes of the GPS satellites currently being used to calculate a location solution.
NMEA Sentence	SV – Indicates the current NMEA sentence string.
Address Line 1	SV – Indicates street address, first line.
Address Line 2	SV – Indicates street address, second line.
City	SV – Indicates city.
State or Province	SV – Indicates state or province.

Country or Region	SV – Indicates country or region represented as an ISO 3166 1-alpha-2 country/region code.
Postal Code	SV – Indicates the postal code.

1.11 Mechanical Sensor Field Usages

These fields are commonly supported by mechanical sensors.

Boolean Switch State	SF – Reports the on/off state of a Boolean switch.
Boolean Switch Array State	SV – Reports the on/off state of each of an array of Boolean switches.
Multivalue Switch Value	SV – Reports the multivalue state of a Multivalue switch.
Force	SV – Measures force. Default unit of measure is Newtons; can be overridden using explicit Unit and/or Unit Exponent.
Absolute Pressure	SV – Measures absolute pressure. Default unit of measure is Pascals; can be overridden using explicit Unit and/or Unit Exponent.
Gauge Pressure	SV – Measures relative gauge pressure. Default unit of measure is Pascals; can be overridden using explicit Unit and/or Unit Exponent.
Strain	SV – Measures strain (in percent). Can be scaled with a Unit Exponent.
Weight	SV – Measures weight. Default unit of measure is kilograms; can be overridden using explicit Unit and/or Unit Exponent.
Vibration State	DF – The on/off state of a Haptic feedback vibrator.
Forward Vibration Speed	DV – The forward speed of the vibrator (in percent). Can be scaled with a Unit Exponent.
Backward Vibration Speed	DV – The backward speed of the vibrator (in percent). Can be scaled with a Unit Exponent. Some haptic motors do not support both forward and backward motion. For those that do, setting *both* forward and backward speeds to non-zero values simultaneously has a vendor-defined behavior.

1.12 Motion Sensor Field Usages

These fields are commonly supported by motion sensors.

Motion State	SF – A flag indicating presence or absence of motion.
Motion Intensity	SV – A positive number indicating intensity of motion if motion is detected (in percent), otherwise 0. Can be scaled with a Unit Exponent.
Acceleration	SV – Linear acceleration magnitude without respect to which axis it occurs in. This is usually used as a composite value for specifying min, max and accuracy for related accelerations. Default unit of measure is G's; can be overridden using explicit

	Unit and/or Unit Exponent.
Acceleration Axis X	SV – Linear acceleration along the X axis. Default unit of measure is G's; can be overridden using explicit Unit and/or Unit Exponent.
Acceleration Axis Y	SV – Linear acceleration along the Y axis. Default unit of measure is G's; can be overridden using explicit Unit and/or Unit Exponent.
Acceleration Axis Z	SV – Linear acceleration along the Z axis. Default unit of measure is G's; can be overridden using explicit Unit and/or Unit Exponent.
Angular Velocity	SV – Angular velocity magnitude without respect to which axis it occurs in. This is usually used as a composite value for specifying min, max and accuracy for related velocity. Default unit of measure is degrees/second; can be overridden using explicit Unit and/or Unit Exponent.
Angular Velocity X Axis	SV – Angular velocity about the X axis. Default unit of measure is degrees/second; can be overridden using explicit Unit and/or Unit Exponent.
Angular Velocity Y Axis	SV – Angular velocity about the Y axis. Default unit of measure is degrees/second; can be overridden using explicit Unit and/or Unit Exponent.
Angular Velocity Z Axis	SV – Angular velocity about the Z axis. Default unit of measure is degrees/second; can be overridden using explicit Unit and/or Unit Exponent.
Angular Position	SV – Angular position without respect to which axis it occurs in. This is usually used as a composite value for specifying min, max and accuracy for related position. Default unit of measure is degrees; can be overridden using explicit Unit and/or Unit Exponent.
Angular Position X Axis	SV – Angular position about the roll axis. Default unit of measure is degrees; can be overridden using explicit Unit and/or Unit Exponent.
Angular Position Y Axis	SV – Angular position about the pitch axis. Default unit of measure is degrees; can be overridden using explicit Unit and/or Unit Exponent.
Angular Position Z Axis	SV – Angular position about the yaw axis. Default unit of measure is degrees; can be overridden using explicit Unit and/or Unit Exponent.
Speed	SV – Velocity magnitude without respect to direction. Default unit of measure is meters/second; can be overridden using explicit Unit and/or Unit Exponent.

1.13 Orientation Sensor Field Usages

These fields are commonly supported by orientation sensors.

Heading	SV – Indicates the compass heading without respect to which axis it occurs in. This is usually used as a composite value for

	specifying min, max and accuracy for related axes. Default unit of measure is degrees; can be overridden using explicit Unit and/or Unit Exponent.
Heading X Axis	SV – Indicates the compass X axis heading. Default unit of measure is degrees; can be overridden using explicit Unit and/or Unit Exponent.
Heading Y Axis	SV – Indicates the compass Y axis heading. Default unit of measure is degrees; can be overridden using explicit Unit and/or Unit Exponent.
Heading Z Axis	SV – Indicates the compass Z axis heading. Default unit of measure is degrees; can be overridden using explicit Unit and/or Unit Exponent.
Heading Compensated Magnetic North	SV – Indicates compass magnetic heading has been compensated for tilt with respect to earth normal. Default unit of measure is degrees; can be overridden using explicit Unit and/or Unit Exponent.
Heading Compensated True North	SV – Indicates compass true north heading has been compensated for tilt with respect to earth normal. Default unit of measure is degrees; can be overridden using explicit Unit and/or Unit Exponent.
Heading Magnetic North	SV – Indicates compass magnetic heading is not compensated. Default unit of measure is degrees; can be overridden using explicit Unit and/or Unit Exponent.
Heading True North	SV – Indicates compass true north heading is not compensated. Default unit of measure is degrees; can be overridden using explicit Unit and/or Unit Exponent.
Distance	SV – Indicates the distance magnitude without respect to which axis it occurs in. This is usually used as a composite value for specifying min, max and accuracy for related axes. Default unit of measure is meters; can be overridden using explicit Unit and/or Unit Exponent.
Distance X Axis	SV – Indicates the X axis distance. Default unit of measure is meters; can be overridden using explicit Unit and/or Unit Exponent.
Distance Y Axis	SV – Indicates the Y axis distance. Default unit of measure is meters; can be overridden using explicit Unit and/or Unit Exponent.
Distance Z Axis	SV – Indicates the Z axis distance. Default unit of measure is meters; can be overridden using explicit Unit and/or Unit Exponent.
Distance Out-of-Range	SF – TRUE when the sensor measuring distance indicates “out of range” meaning the value provided as Distance may not be accurate.
Tilt	SV – Indicates the inclinometer angle without respect to which axis it occurs in. This is usually used as a composite value for specifying min, max and accuracy for related axes. Default unit of measure is degrees; can be overridden using explicit Unit and/or Unit Exponent.

Tilt X Axis	SV – Indicates the inclinometer X axis angle. Default unit of measure is degrees; can be overridden using explicit Unit and/or Unit Exponent.
Tilt Y Axis	SV – Indicates the inclinometer Y axis angle. Default unit of measure is degrees; can be overridden using explicit Unit and/or Unit Exponent.
Tilt Z Axis	SV – Indicates the inclinometer Z axis angle. Default unit of measure is degrees; can be overridden using explicit Unit and/or Unit Exponent.
Rotation Matrix	SV – A 3 x 3 matrix of numbers, all ranging in value from -1.0 to 1.0, representing rotation within a 3D space. No units are specified and scaling is by the Unit Exponent usage.
Quaternion	SV – A matrix of 4 values (x, y, z and w, all ranging in value from -1.0 to 1.0) that represent rotation in space about a unit vector. No units are specified and scaling is by the Unit Exponent usage.
Magnetic Flux	SV – Indicates the magnetic field strength without respect to which axis it occurs in. This is usually used as a composite value for specifying min, max and accuracy for related axis. Default unit of measure is milligauss; can be overridden using explicit Unit and/or Unit Exponent.
Magnetic Flux X Axis	SV – Indicates the X axis magnetic field strength. Default unit of measure is milligauss; can be overridden using explicit Unit and/or Unit Exponent.
Magnetic Flux Y Axis	SV – Indicates the Y axis magnetic field strength. Default unit of measure is milligauss; can be overridden using explicit Unit and/or Unit Exponent.
Magnetic Flux Z Axis	SV – Indicates the Z axis magnetic field strength. Default unit of measure is milligauss; can be overridden using explicit Unit and/or Unit Exponent.

In addition to the field usages listed above, the following usages are commonly used with Orientation sensors (and are “normally” found under Location sensors).

Page ID (hex)	Page Name	Usage (hex)	Usage Name	Comment
20	Sensors	0416	Magnetic Heading	Combined magnetic heading (synthesis of X, Y, and Z data).
20	Sensors	0417	Magnetic Variation	Magnetic declination compared to true north.

Table 5. Other Common Usages for Orientation Sensors

1.14 Scanner Sensor Field Usages

These fields are commonly supported by scanner sensors.

RFID Tag 40 Bit	SV – Indicates the 40-bit radio frequency ID tag value.
NFC Sentence Receive	SV – Data message received from an NFC controller. Data type is an opaque counted array of bytes; interpretation will depend on

	host-based middleware (HCI specification protocol is typical).
NFC Sentence Send	DV – Data message sent to an NFC controller. Data type is an opaque counted array of bytes; interpretation will depend on host-based middleware (HCI specification protocol is typical).

See Also
In addition to the field usages listed above, usages from the Bar Code Scanner Usage Page (0x8C) as documented in Section 3 of the *Universal Serial Bus HID Point of Sale Usage Tables* specification (*Reference Document [4]*) may be used for Barcode Scanners.

1.15 Time Sensor Field Usages

These fields are commonly supported by time sensors.

Year	SV – Indicates the current year.
Month	SV – Indicates the current month (1 – 12).
Day	SV – Indicates the current day of the month (1 – 31).
Day of Week	NAry – Indicates the current day of the week: • Sel – Sunday, • Sel – Monday, • Sel – Tuesday, • Sel – Wednesday, • Sel – Thursday, • Sel – Friday, • Sel – Saturday.
Hour	SV – Indicates the current hour (00 – 23).
Minute	SV – Indicates the current minute (00 – 59).
Second	SV – Indicates the current second (00 – 59).
Millisecond	SV – Indicates the current millisecond (000 – 999).
Timestamp	SV – Indicates the current time (UTC) expressed in a format compliant to the "C" language library `_time64()` function (i.e., the number of seconds since 1/1/1970 00:00:00 UTC).
Julian Day of Year	SV – Indicates the day of the year (1 – 366).
Time Zone Offset From UTC	DV – Specifies the local time zone offset from UTC. Default unit of measure is minutes; can be overridden using explicit Unit and/or Unit Exponent.
Time Zone Name	DV – Specifies the textual name of the local time zone.
Daylight Savings Time Observed	DF – Specifies whether or not Daylight Savings Time or Summer Time is observed in the local area.
Time Trim Adjustment	DV – Specifies a trim factor used to correct inaccuracies in the

	Real-Time Clock. It is a signed unit-less value, and implementation dependent.
Arm Alarm	DF – Specifies whether the Alarm function should be armed (TRUE) or disarmed (FALSE). The alarm is automatically disarmed when it expires (i.e., "goes off").

1.16 Custom Sensor Field Usages

These fields are commonly supported by custom sensors.

Custom Usage	SV – Indicates the HID Sensor Usage.
Custom Boolean Array	SV – Reports the on/off state of each of an array of Boolean variables.
Custom Value	SV – Custom value without respect to which specific custom value field is being used. This is usually used as a composite value for specifying min, max and accuracy for custom values. Units are specified by the Units usage and scaling by the Unit Exponent usage.
Custom Value 1	SV – A first custom value. Units are specified by the Units usage and scaling by the Unit Exponent usage.
Custom Value 2	SV – A second custom value. Units are specified by the Units usage and scaling by the Unit Exponent usage.
Custom Value 3	SV – A third custom value. Units are specified by the Units usage and scaling by the Unit Exponent usage.
Custom Value 4	SV – A fourth custom value. Units are specified by the Units usage and scaling by the Unit Exponent usage.
Custom Value 5	SV – A fifth custom value. Units are specified by the Units usage and scaling by the Unit Exponent usage.
Custom Value 6	SV – A sixth custom value. Units are specified by the Units usage and scaling by the Unit Exponent usage.

1.17 Generic Sensor Field Usages

These fields are commonly supported by generic sensors.

Generic GUID or PROPERTYKEY	SV – a 30-byte structure GUID_OR_PROPERTYKEY defined in Section 4.2.7, below.
Generic GUID or PROPERTYKEY kind	NAry – Indicates what kind of GUID or PROPERTYKEY is being used: • Sel – Sensor Category GUID, • Sel – Sensor Type GUID, • Sel – Sensor Event PROPERTYKEY, • Sel – Sensor Property PROPERTYKEY, • Sel – Sensor Data Field PROPERTYKEY.

Generic GUID	SV – A 16-byte GUID. May be a Category GUID or a Type GUID; as specified by **Generic GUID or PROPERTYKEY kind**.
Generic Category GUID	SV – A 16-byte GUID used to specify an "inline" sensor category. The GUID is followed by a field that indicates the sensor category value assigned to that GUID.
Generic Type GUID	SV – A 16-byte GUID used to specify an "inline" sensor type. The GUID is followed by a field that indicates the sensor type value assigned to that GUID.
Generic PROPERTYKEY	SV – A 20-byte PROPERTYKEY. May be an Event PROPERTYKEY, Property PROPERTYKEY, or a Data Field PROPERTYKEY; as specified by **Generic GUID or PROPERTYKEY kind**.
Generic Event PROPERTYKEY	SV – A 20-byte PROPERTYKEY used to specify an "inline" sensor event. The PROPERTYKEY is followed by a field that indicates the event value assigned to that PROPERTYKEY.
Generic Property PROPERTYKEY	SV – A 20-byte PROPERTYKEY used to specify an "inline" sensor property. The PROPERTYKEY is followed by a field that indicates the property value assigned to that PROPERTYKEY.
Generic Data Field PROPERTYKEY	SV – A 20-byte PROPERTYKEY used to specify an "inline" sensor data field. The PROPERTYKEY is followed by a field that indicates the data field value assigned to that PROPERTYKEY.
Generic Event	SV – Usage ID for the field that follows the **Generic Event PROPERTYKEY**.
Generic Property	SV – Usage ID for the field that follows the **Generic Property PROPERTYKEY**.
Generic Data Field	SV – Usage ID for the field that follows the **Generic Data Field PROPERTYKEY**.
Enumerator Table Row Index	SV (when Data Field), DV (when Property) – When using the "Enumerator" top-level-collection strategy (*see Section 4.2.7.1, below*), this usage specifies the Row index of the Enumerator's table.
Enumerator Table Row Count	SV – When using the "Enumerator" top-level-collection strategy (*see Section 4.2.7.1, below*), this usage specifies the total count of Rows in the Enumerator's table.
Generic Top Level Collection ID	SV – Identifies the HID Top Level Collection ID for the Row in the Enumerator's table.
Generic Report ID	SV – Identifies the HID Report ID for the Row in the Enumerator's table.
Generic Report Item Position Index	SV – Indicates the 1-based sequential position of the Property or Data Field in its Report.
Generic Firmware VARTYPE	NAry – Identifies the firmware data type associated with the Property or Data Field in the Row in the Enumerator's table. • Sel – VT_NULL: Empty, • Sel – VT_BOOL: Boolean, • Sel – VT_UI1: Byte,

	• Sel – VT_I1: Character, • Sel – VT_UI2: Unsigned Short, • Sel – VT_I2: Short, • Sel – VT_UI4: Unsigned Long, • Sel – VT_I4: Long, • Sel – VT_UI8: Unsigned Long Long, • Sel – VT_I8: Long Long, • Sel – VT_R4: Float, • Sel – VT_R8: Double, • Sel – VT_WSTR: Wide String, • Sel – VT_STR: Narrow String, • Sel – VT_CLSID: Guid, • Sel – VT_VECTOR|VT_UI1: Opaque Structure, • Sel – VT_F16E0: HID 16-bit Float with Unit Exponent 0, • Sel – VT_F16E1: HID 16-bit Float with Unit Exponent 1, • Sel – VT_F16E2: HID 16-bit Float with Unit Exponent 2, • Sel – VT_F16E3: HID 16-bit Float with Unit Exponent 3, • Sel – VT_F16E4: HID 16-bit Float with Unit Exponent 4, • Sel – VT_F16E5: HID 16-bit Float with Unit Exponent 5, • Sel – VT_F16E6: HID 16-bit Float with Unit Exponent 6, • Sel – VT_F16E7: HID 16-bit Float with Unit Exponent 7, • Sel – VT_F16E8: HID 16-bit Float with Unit Exponent 8, • Sel – VT_F16E9: HID 16-bit Float with Unit Exponent 9, • Sel – VT_F16EA: HID 16-bit Float with Unit Exponent A, • Sel – VT_F16EB: HID 16-bit Float with Unit Exponent B, • Sel – VT_F16EC: HID 16-bit Float with Unit Exponent C, • Sel – VT_F16ED: HID 16-bit Float with Unit Exponent D, • Sel – VT_F16EE: HID 16-bit Float with Unit Exponent E, • Sel – VT_F16EF: HID 16-bit Float with Unit Exponent F, • Sel – VT_F32E0: HID 32-bit Float with Unit Exponent 0, • Sel – VT_F32E1: HID 32-bit Float with Unit Exponent 1, • Sel – VT_F32E2: HID 32-bit Float with Unit Exponent 2, • Sel – VT_F32E3: HID 32-bit Float with Unit Exponent 3, • Sel – VT_F32E4: HID 32-bit Float with Unit Exponent 4, • Sel – VT_F32E5: HID 32-bit Float with Unit Exponent 5, • Sel – VT_F32E6: HID 32-bit Float with Unit Exponent 6, • Sel – VT_F32E7: HID 32-bit Float with Unit Exponent 7, • Sel – VT_F32E8: HID 32-bit Float with Unit Exponent 8, • Sel – VT_F32E9: HID 32-bit Float with Unit Exponent 9, • Sel – VT_F32EA: HID 32-bit Float with Unit Exponent A, • Sel – VT_F32EB: HID 32-bit Float with Unit Exponent B, • Sel – VT_F32EC: HID 32-bit Float with Unit Exponent C, • Sel – VT_F32ED: HID 32-bit Float with Unit Exponent D, • Sel – VT_F32EE: HID 32-bit Float with Unit Exponent E, • Sel – VT_F32EF: HID 32-bit Float with Unit Exponent F.
Generic Unit of Measure	NAry – Indicates the HID Unit for the Row in the Enumerator's table. These are used in lieu of explicit `Unit()` declarations in the HID Report Descriptor for Generic Sensors. • Sel – Not Specified, • Sel – Lux, • Sel – Degrees Kelvin,

	• Sel – Degrees Celsius, • Sel – Pascal, • Sel – Newton, • Sel – Meters/Second, • Sel – Kilogram, • Sel – Meter, • Sel – Meters/Second/Second, • Sel – Farad, • Sel – Ampere, • Sel – Watt, • Sel – Henry, • Sel – Ohm, • Sel – Volt, • Sel – Hertz, • Sel – Bar, • Sel – Degrees Anti-clockwise, • Sel – Degrees Clockwise, • Sel – Degrees, • Sel – Degrees/Second, • Sel – Degrees/Second/Second, • Sel – Knot, • Sel – Percent, • Sel – Second, • Sel – Millisecond, • Sel – G, • Sel – Bytes, • Sel – Milligauss, • Sel – Bits.
Generic Unit Exponent	NAry – Indicates the HID Unite Exponent for the Row in the Enumerator's table. • Sel – Exponent 0: 1 • Sel – Exponent 1: 10 • Sel – Exponent 2: 100 • Sel – Exponent 3: 1 000 • Sel – Exponent 4: 10 000 • Sel – Exponent 5: 100 000 • Sel – Exponent 6: 1 000 000 • Sel – Exponnet 7: 10 000 000 • Sel – Exponent 8: 0.00 000 001 • Sel – Exponent 9: 0.0 000 001 • Sel – Exponent A: 0.000 001 • Sel – Exponent B: 0.00 001 • Sel – Exponent C: 0.0 001 • Sel – Exponent D: 0.001 • Sel – Exponent E: 0.01 • Sel – Exponent F: 0.1
Generic Report Size	SV – Indicates the HID Report Size for the Row in the Enumerator's table.
Generic Report Count	SV – Indicates the HID Report Count for the Row in the Enumerator's table.

2. Sensor Backgrounder

This section describes Sensor terminology and the conceptual object model associated with the HID Sensor Usages. This section is Informative, which means it is only for orientation and guidance.

2.1 Glossary

A number of terms specific to the Sensor subject matter are used in the context of this document. The following table provides a list of those terms, and the intended meaning.

Accelerometer	A device that measures acceleration along one or more linear axes (traditionally called X, Y, and Z). When the device is still, it can measure the acceleration of the Earth's gravity, and can be used to calculate the device's orientation (expressed in angles traditionally called *Pitch* and *Roll*). When the device is in motion, can also measure the speed ups and slow downs. Note that an accelerometer cannot detect the difference between perfectly still and a perfectly constant speed.
Actuator	A device that causes an output. Technically not a true *Sensor* (which acquires an input), but some actuators are represented as Sensors for convenience.
Altimeter	Another name for *Barometer*.
Altitude	Height with respect to some reference level of the Earth. The most common reference levels are the *Geoid* (especially when reported by a *GPS*) and mean sea level (when reported by a *Barometer*).
Ambient Light Sensor	Measures the amount of light striking the sensor, from which can be deduced the amount of ambient light in the local environment.
Ambient Temperature Sensor	Another name for *Thermometer*.
Atmospheric Pressure Sensor	Another name for *Barometer*.
Anemometer	Measures wind speed. Also called a *Wind Speed* sensor.
Barometer	A sensor that measures the pressure exerted by the weight of the atmosphere. Also called an *Atmospheric Pressure* sensor. The output can be affected by local weather conditions. Cold weather results in low pressure; warm weather results in high pressure. The output is also affected by *Altitude*; the higher you are, the less weight of atmosphere is pressing down. As a result, barometers can also be used to calculate *Altitude* and therefore are also called *Altimeter*s.
Category	A first-order level of description of a *Sensor*'s *Type*. The *Categories* of *Sensor*s are: **All**, **Biometric**, **Electrical**, **Environmental**, **Light**, **Location**, **Mechanical**, **Motion**, **Orientation**, **Scanner**, **Time**, **Custom**, and **Generic**. The Application Programming Interfaces (APIs) of modern operating systems may provide a way of searching for sensors (for example, by scanning all of the HID Top Level Collections and/or all of the Logical sub-Collections) based on the *Category* description. This may be convenient and adequate based on application programmer needs. A more comprehensive search technique is to

	examine all of the sensors and select those which support the *Data Fields* which are needed to implement the application.
Chromaticity	A method of describing the color of light. The scale commonly used is called CIE 1931, and colors are represented by X, Y coordinates on the chart. Also called Chromacity.
Compass	Measures magnetic flux along one or more linear axes (traditionally called X, Y, and Z). Can be used to calculate *Heading* with respect to the Earth's Magnetic North Pole. It can be used to calculate the device's orientation (expressed in an angle traditionally called *Yaw*).
Consumer IR	A Consumer IR sensor uses infrared signals to transmit short messages between devices. These devices are typically consumer electronics components, such as TVs, VCRs, DVD players, hand-held Remote Controls, and so on. You can use the Consumer IR sensor to send and receive messages to such consumer electronics components. Typically, you would send messages to devices such as TVs, VCRs, and DVD players; while instead receiving messages from hand-held Remote Controls. Almost all Consumer IR message codes are vendor proprietary. To interoperate with any given consumer electronics device, it is necessary to consult a huge code database (usually must be licensed from a provider, and downloaded via the Internet). For example, the "Power On/Off" code for a Sony TV will be different from the "Power On/Off" code for a Toshiba TV, and even from the "Power On/Off" code for a Sony DVD player. Consumer IR is not to be confused with IrDA, which is a different (incompatible) infrared communications protocol. Consumer IR devices may more commonly be represented as a HID Remote Control device instead of as a HID Sensor.
Data Field	*Data Field*s are the real-time acquired data values from *Sensor*s. The application program can acquire them synchronously by "polling", or receive them asynchronously as and when sent by the sensors themselves. HID Input Reports contain one or more *Data Field*s as Input Report Items.
Dead Reckoning	A technique whereby various sensors are used to **estimate** position of a device in motion, typically while a GPS signal is temporarily unavailable (for example, while driving through a tunnel or after having walked indoors). A combination of one or more of *Accelerometer*, *Gyro*, *Compass*, and *Altimeter* are typically used.
Ellipsoid	Another name for the *Geoid*.
Enumerator	A "virtual sensor" object used by the *Sensor Hub* device driver to enumerate at run-time all of the *Sensors*, their *Properties*, and *Data Fields* using their Generic descriptions (rather than describing them thoroughly using HID Report Descriptor attributes).
Geoid	A simplified mathematical model used to describe the "egg"-like shape of the Earth. Altitude as reported by a GPS is with respect to the *Geoid*. This may vary slightly from true altitude above mean sea level. The *Geoid* is also called the *Ellipsoid*. The difference between the Geoid and true sea level is called *Geoidal*

	Separation or *Ellipsoid Error*.
GPS	A device that receives (never sends) weak radio signals from a series of satellites orbiting high above the Earth. Together these form the Global Positioning System. Based on the content of the signals sent by the satellites, a GPS receiver is able to compute its location (*Latitude*, *Longitude*, and *Altitude*) on the Earth and speed (in knots). At least four satellites must be “acquired” in order for the GPS to “get a location fix”. This may take some time, and is affected by how clear the view is to the satellites (streets surrounded by tall buildings, or locations inside buildings made of metal and concrete are problematic).
Gyro	A device that measures Angular Velocity *around* one or more linear axes (traditionally called X, Y, and Z). A *Gyro* is also called a *Gyroscope* or a *Gyrometer*.
Gyrometer	Another name for *Gyro*.
Gyroscope	Another name for *Gyro*.
Hall Effect Sensor	A binary switch that can detect the nearby presence of a magnet. Classic use is as a laptop lid-closed detector.
Haptic Vibrator Motor	An *Actuator* that rotates a motor with an off-balance weight on the axle. This results in the types of vibrations that you may be familiar with your cellular mobile phone producing. Typically used when in “silent mode” to indicate the arrival of an email, SMS message, and so on.
Heading	Angle with respect to North. Magnetic Heading is expressed with respect to the Earth’s Magnetic North Pole, while True Heading is expressed with respect to the Earth’s Geographic North Pole. Note that the Magnetic North Pole and the Geographic North Pole are not at the same location on the globe. The North Magnetic Pole moves slowly over time due to constant magnetic changes in the Earth's core. In 2001, it was near Ellesmere Island in northern Canada. The difference between the two poles is called *Magnetic Declination* or *Magnetic Variation*. To calculate the True Heading, you have to subtract the Magnetic Declination from the reported Magnetic Heading. The correct value of *Magnetic Declination* depends upon your position on the globe. Heading is also called *Azimuth*, especially when referring to GPS satellite positions in the sky.
(Human) Presence	A sensor used to detect the presence (Boolean true or false) of a human in front of the sensor. This is typically done using reflection of infrared or ultrasonic waves.
(Human) Proximity	A sensor used to detect the linear distance that a human is away from (in front of) the sensor. This is typically done using reflection of infrared or ultrasonic waves.
Humidity Sensor	Another name for *Hygrometer*.
Hygrometer	A sensor used to measure Relative Humidity, the percent saturation of water in the atmosphere.
Inclinometer	A sensor used to measure angular tilt with respect to one or more axes (traditionally called X, Y, and Z).
Latitude	A component of position on the Earth, measured as degrees of

	inclination or declination from the Equator. Latitudes north of the Equator are positive; Latitudes south of the Equator are negative.
Longitude	A component of position on the Earth, measured as degrees away from the Prime Meridian. Longitudes east of the Prime Meridian are positive (up until 180°); Longitudes west of the Prime Meridian are negative.
Magnetic Declination	Another name for *Magnetic Variation*.
Magnetic Variation	The angle described by the difference in *Heading* between the Earth's Magnetic North Pole and the Geographic North Pole. It depends upon your position on the globe.
Magnetometer	Another name for *Compass*.
NFC	Near-Field Communications device. Uses magnetic resonance and/or radio frequency to interact with another device at very short distance (typically 5 cm or less). NFC is used for "sharing", "pairing", and "transactions": 1. Sharing small bursts of non-private data such as electronic Business Cards; 2. Pairing two devices securely prior to performing higher-level protocol messaging; or 3. For performing real-time electronic micro-payments such as subway fares.
Pitch	Angle up or down from "straight ahead". Positive pitch angles slant upward, and negative pitch angles slant downward. The terminology comes from use in airplanes. Technically, *Pitch* only can be calculated from X, Y, Z accelerometer coordinates that have first been normalized to the "NED" (X=North, Y=East, Z=Down) coordinate system. Also called *Elevation*, especially when referring to GPS satellite positions in the sky.
Property	*Properties* are *Sensor* identification or configuration values. The application program can read ("get") or write ("set") them synchronously. Some *Properties* are Read/Write, and others are Read-Only. HID Feature Reports contain one or more *Properties* as Feature Report Items.
Proximity Sensor	A *Sensor* that detects something else nearby. Most often, the sensor detects a Human (see *Human Presence* and *Human Proximity*). *NFC* Sensors can be used to detect the (very near) proximity of another NFC-enabled device.
Real Time Clock	A device that measures time. It may also have Alarm capabilities. For the purpose of a *Sensor Hub*, the *Real Time Clock* is used to *UTC* timestamp all the samples acquired from the *Sensors*.
RFID	Radio Frequency IDentification device. *RFID* Readers are able to acquire a small burst of data from non-self-powered *RFID* "tags" such as found in employee badges and some credit cards. Many *NFC* Readers are also able to read *RFID* tags.
Roll	Angle right or left from "flat". Positive roll angles bank rightward, and negative roll angles bank leftward. The terminology comes from use in airplanes. Technically, *Roll* only can be calculated from X, Y, Z accelerometer coordinates that have first been normalized to the "NED" (X=North, Y=East,

	Z=Down) coordinate system.
Selection (Value)	The manner in which enumeration values are expressed in HID. A Usage that can take on one of multiple enumeration values is defined as **NAry** and the enumerated choices are defined as **Sel** (*Selection*s). The numeric value actually reported for the *Data Field* is the zero-based index of the list of *Selection*s shown in the HID Report Descriptor.
Sensor	A device that acquires an input, measuring some physical phenomena. Although not strictly a true Sensor, *Actuators* are also represented as Sensors for convenience.
Sensor Hub	A *Sensor Hub* is a small microcontroller that attaches to multiple physical *Sensor*s, provides a single physical I/O connection to the PC, and provides a homogenized object-oriented sensor abstraction to the device driver(s). Functionally, the *Sensor Hub* offers sensor aggregation, PC CPU offload, and enhanced triggering options. The Sensor Hub "virtual sensor" may itself have its own *Properties* and *Data Fields* that control the management of all the *Sensors* in an aggregated manner.
Thermometer	A device that measures the *Ambient Temperature* of the atmosphere.
Triangulation	A technique whereby location is estimated by describing circles around a radio source such as at least three cellular telephone provider's transmitting antennae. The radius of the circle (distance from each transmitter antenna) is proportional to the signal strength. The intersection of the three described circles "triangulates" the position of the receiving device.
Type	A detailed level of description of a *Sensor*'s intended function. The Application Programming Interfaces (APIs) of modern operating systems may provide a way of searching for sensors (for example, by scanning all of the HID Top Level Collections and/or all of the Logical sub-Collections) based on the *Type* description. This may be convenient and adequate based on application programmer needs. A more comprehensive search technique is to examine all of the sensors and select those which support the *Data Fields* which are needed to implement the application.
UTC Time	Universal Coordinated Time, also known as Greenwich Mean Time ("GMT"). All sensor samples are timestamped when acquired, with UTC. This is the only way to perform consistent sample analysis, especially if and when you are in motion (you might cross a time zone or the International Date Line, for example). Unless you live in England, UTC time is likely to be different than your PC's local time. Even in England during Summer Time (called Daylight Savings Time in the US) the local time differs from UTC by one hour. Basing all calculations on UTC means that you don't have to account for all the (error prone) complexity of time zones and semi-yearly "spring forward" and "fall back" time shifts. Also consider the multiplied complexity of processing data from a network of sensors that is spread across a wide geographic region or across the entire planet. A single consistent time reference is a necessity.

Wind Direction sensor	Another name for *Wind Vane*.
Wind Speed sensor	Another name for *Anemometer*.
Wind Vane	A device that measures the direction from which the wind is blowing. Traditionally this is reported with respect to the Magnetic North Pole, the same as reported by a *Compass*. A *Wind Vane* is also called a *Wind Direction* sensor or Weather Vane.
Yaw	Rotation around the vertical axis; the angle right or left from "straight ahead". The terminology comes from use in airplanes. Technically, *Yaw* only can be calculated from compass coordinates that have first been normalized to the "NED" (X=North, Y=East, Z=Down) coordinate system.

2.2 Sensor Taxonomy and Object Model

Sensors are described using object-oriented principles. The following table describes the objects and the iconography that will be used.

	Sensor	The simplest "base class" of all sensor objects is **Sensor**. The next-level "sub classes" of *Sensor* are the *Categories*; for example **Motion** sensor. Additional "sub classing" can be done as convenient, for example **Accelerometer**, or even more detailed **3D Accelerometer**. Specifying this Usage in the HID Report Descriptor is an informational "hint" to the application programmer, and any level of detail may be provided by the implementer.
	Property	Properties are sensor identification values that may be read, or sensor configuration setting values that may be read and written. HID Feature Reports are used to transfer Properties as Feature Report Items.
	Data Field	Data Fields are sensor acquired data values, which may only be read. HID Input Reports are used to transfer Data Fields as Input Report Items.
	Selection	Properties or Data Fields that can take on one of a finite number of enumeration values define each possible value as a HID Selection Usage.

The first-level sub-classes of *Sensor* are the *Categories*. Application programmers may search for a sensor of interest based on its Category and use it.

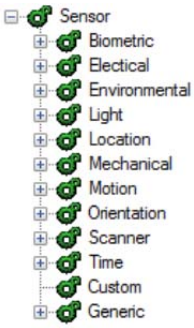

Figure 1. Sensor Categories

The full expanded hierarchy tree of sensor *Type*s contains further sub-classes of *Sensor*. Application programmers may also search for a sensor of interest based on its *Type* and use it.

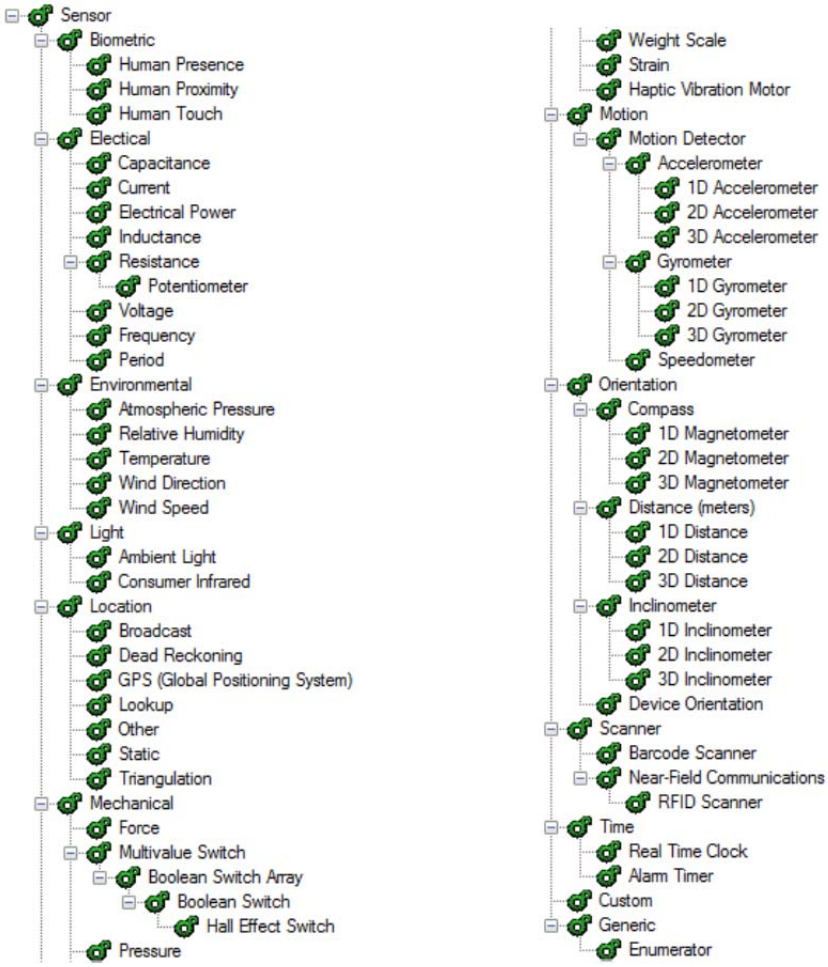

Figure 2. Sensor Types

The following diagram illustrates when all Properties, Data Fields, and Selections for all sensor classes are shown.

Properties and Data Fields are shown underneath the sensor classes where they are *typically* encountered, but this is arbitrary. It is possible that specific Properties or Data Fields may appear underneath any sensor that the implementer feels makes sense.

For example, many Barometers also contain an integrated ambient temperature sensor, so the **Temperature (degrees Celsius)** Data Field could appear underneath a **Barometer** as easily as it could be found underneath a **Thermometer**.

Another example that commonly occurs is that **Magnetic Heading (degrees)** and **True Heading (degrees)** are reported by a **Compass** and also by a **GPS**.

In order to deal with this uncertainty, the recommended approach for programmers is to scan all sensor collections for the Data Fields of greatest interest to the needs of the application. Once they are located (regardless of which sensor they are reported underneath), use those.

- Sensor
  - Event State (selection list)
    - State: Undefined
    - State: Ready
    - State: Not Available
    - State: No Data
    - State: Initializing
    - State: Access Denied
    - State: Error
  - Event ID (selection list)
    - Event: Unknown
    - Event: State Changed
    - Event: Property Changed
    - Event: Data Updated
    - Event: Poll Response
    - Event: Change Sensitivity
    - Event: Range Maximum Reached
    - Event: Range Minimum Reached
    - Event: High Threshold Cross Upward
    - Event: High Threshold Cross Downward
    - Event: Low Threshold Cross Upward
    - Event: Low Threshold Cross Downward
    - Event: Zero Threshold Cross Upward
    - Event: Zero Threshold Cross Downward
    - Event: Period Exceeded
    - Event: Frequency Exceeded
    - Event: Complex Trigger
  - Timestamp
  - Friendly Name
  - Persistent Unique ID
  - Status
  - Minimum Report Interval (milliseconds)
  - Manufacturer
  - Model
  - Serial Number
  - Description
  - Connection Type (selection list)
    - ConnType: PC Integrated
    - ConnType: PC Attached
    - ConnType: PC External
  - Device Path
  - Hardware Revision
  - Firmware Revision
  - Release Date
  - Current Report Interval (milliseconds)
  - Change Sensitivity Absolute
  - Change Sensitivity Percent of Range
  - Change Sensitivity Percent Relative
  - Accuracy
  - Resolution
  - Range Maximum
  - Range Minimum
  - Reporting State (selection list)
    - ReportState: Report No Events
    - ReportState: Report All Events
    - ReportState: Report Threshold Events
    - ReportState: Wake On No Events
    - ReportState: Wake On All Events
    - ReportState: Wake On Threshold Events
  - Sampling Rate (millisconds)
  - Response Curve
  - Power State (selection list)
    - PowerState: Undefined
    - PowerState: D0 Full Power
    - PowerState: D1 Low Power
    - PowerState: D2 Standby Power with Wakeup
    - PowerState: D3 Sleep with Wakeup
    - PowerState: D4 Power Off
- Biometric
  - Human Presence
    - Human Presence (boolean)
  - Human Proximity
    - Human Proximity Range (meters)
    - Human Proximity Out Of Range (boolean)
  - Human Touch
    - Human Touch State (boolean)
- Electical
  - Percent of Range
  - Capacitance
    - Capacitance (farad)
  - Current
    - Current (amps)
  - Electrical Power
    - Electrical Power (watts)
  - Inductance
    - Inductance (henry)
  - Resistance
    - Resistance (ohms)
    - Potentiometer
  - Voltage
    - Voltage (volts)
  - Frequency
    - Frequency (hertz)
  - Period
    - Period (milliseconds)
- Environmental
  - Atmospheric Pressure
    - Atmospheric Pressure (bars)
    - Reference Pressure (bars)
  - Relative Humidity
    - Humidity (percent)
  - Temperature
    - Temperature (degrees Celsius)
  - Wind Direction
    - Wind Direction (degrees)
  - Wind Speed
    - Wind Speed (meters/second)
- Light
  - Ambient Light
    - Illuminance (lux)
    - Light Chromaticity
    - Chromaticity X (CIE 1931 x)
    - Chromaticity Y (CIE 1931 y)
    - Light Color Temperature (degrees Kelvin)
  - Light Response Curve
  - Consumer Infrared
    - CIR Sentence Receive (counted array of bytes)
    - CIR Sentence Send (counted array of bytes)
- Location
  - Location Desired Accuracy (selection list)
    - Accuracy: Default
    - Accuracy: High
    - Accuracy: Medium
    - Accuracy: Low
  - Broadcast
  - Dead Reckoning
  - GPS (Global Positioning System)
    - NMEA Sentence (NMEA-0183)

- Latitude (degrees)
- Longitude (degrees)
- Magnetic Heading (degrees)
- Magnetic Variation (derees)
- True Heading (degrees)
- Speed (knots)
- Altitude above Sea Level (meters)
- Altitude above Sea Level Error (meters)
- Altitude Ellipsoid (meters)
- Altitude Ellipsoid Error (meters)
- Differential GPS Data Age (seconds)
- Differential Reference Station ID
- Error Radius (meters)
- Fix Quality (selection list)
  - GpsFixQ: No Fix
  - GpsFixQ: GPS
  - GpsFixQ: DGPS
- Fix Type (selection list)
  - GpsFixType: No Fix
  - GpsFixType: GPS SPS Mode, Fix Valid
  - GpsFixType: DGPS SPS Mode, Fix Valid
  - GpsFixType: GPS PPS Mode, Fix Valid
  - GpsFixType: Real Time Kinematic
  - GpsFixType: Float RTK
  - GpsFixType: Estimated (dead reckoned)
  - GpsFixType: Manual Input Mode
  - GpsFixType: Simulator Mode
- Geoidal Separation (meters)
- GPS Operation Mode (selection list)
  - GpsOpMode: Manual
  - GpsOpMode: Automatic
- GPS Selection Mode (selection list)
  - GpsSelMode: Autonomous
  - GpsSelMode: DPGS
  - GpsSelMode: Estimated (dead reckoned)
  - GpsSelMode: Manual Input
  - GpsSelMode: Simulator
  - GpsSelMode: Data Not Valid
- GPS Status (selection list)
  - GpsStatus: Data Valid
  - GpsStatus: Data is Not Valid
- Horizontal Dilution of Precision
- Vertical Dilution of Precision
- Position Dilution of Precision
- Satellites in View
- Satellites in View Azimuth (array of degrees)
- Satellites in View Elevation (array of degrees)
- Satellites in View IDs (array)
- Satellites in View PRNs (array)
- Satellites in View S/N Ratios (array)
- Satellites Used Count
- Satellites Used PRNs (array)
- Lookup
- Other
- Static
  - Address 1
  - Address 2
  - City
  - State or Province
  - Country or Region (ISO 3166)
  - Postal Code
- Triangulation
  - Antenna Altitude (meters above sea level)
- Mechanical
  - Force
    - Force (newtons)
  - Multivalue Switch
    - Multivalue Switch State
    - Boolean Switch Array
      - Boolean Switch Array States (array of boolean)
      - Boolean Switch
        - Boolean Switch State (boolean)
        - Hall Effect Switch
  - Pressure
    - Absolute Pressure (pascals)
    - Gauge Pressure (pascals)
  - Weight Scale
    - Weight (kilograms)
  - Strain
    - Strain (percent)
  - Haptic Vibration Motor
    - Vibration State (boolean)
    - Forward Vibration Speed (percent)
    - Backward Vibration Speed (percent)
- Motion
  - Motion Detector
    - Motion State (boolean)
    - Motion Intensity (percent)
    - Accelerometer
      - Linear Acceleration Magnitude (G's)
      - Linear Acceleration along X axis (G's)
      - Linear Acceleration along Y axis (G's)
      - Linear Acceleration along Z axis (G's)
      - 1D Accelerometer
      - 2D Accelerometer
      - 3D Accelerometer
    - Gyrometer
      - Angular Velocity Magnitude (degrees/sec)
      - Angular Velocity about X axis (degrees/sec)
      - Angular Velocity about Y axis (degrees/sec)
      - Angular Velocity about Z axis (degrees/sec)
      - Angular Position about X axis (degrees)
      - Angular Position about Y axis (degrees)
      - Angular Position about Z axis (degrees)
      - 1D Gyrometer
      - 2D Gyrometer
      - 3D Gyrometer
    - Speedometer
      - Speed (meters/second)
- Orientation
  - Compass
    - Heading (degrees)
    - Heading along X axis (degrees)
    - Heading along Y axis (degrees)
    - Heading along Z axis (degrees)
    - Heading Compensated toward Magnetic North (degrees)
    - Heading Compensated toward True North (degrees)
    - Heading toward Magnetic North (degrees)
    - Heading toward True North (degrees)
    - 1D Magnetometer
    - 2D Magnetometer
    - 3D Magnetometer
    - Magnetic Flux Magnitude (milligauss)
    - Magnetic Flux along X axis (milligauss)
    - Magnetic Flux along Y axis (milligauss)
    - Magnetic Flux along Z axis (milligauss)
  - Distance (meters)
    - Distance Magnitude (meters)
    - Distance along X axis (meters)

- Distance along Y axis (meters)
- Distance along Z axis (meters)
- Distance Out Of Range (boolean)
- 1D Distance
- 2D Distance
- 3D Distance
- Inclinometer
  - Tilt (degrees)
  - Tilt from X axis (degrees)
  - Tilt from Y axis (degrees)
  - Tilt from Z axis (degrees)
  - 1D Inclinometer
  - 2D Inclinometer
  - 3D Inclinometer
- Rotation Matrix (3x3 matrix of IEEE 32-bit float coefficients)
- Quaternion (1x4 vector of IEEE 32-bit float coefficients)
- Device Orientation
- Scanner
  - Barcode Scanner
  - Near-Field Communications
    - NFC Sentence Receive (counted array of bytes)
    - NFC Sentence Send (counted array of bytes)
    - RFID Scanner
      - RFID Tag 40-bit
- Time
  - Timestamp
  - Year
  - Month
  - Day
  - Day of Week
    - DayOfWeek: Sunday
    - DayOfWeek: Monday
    - DayOfWeek: Tuesday
    - DayOfWeek: Wednesday
    - DayOfWeek: Thursday
    - DayOfWeek: Friday
    - DayOfWeek: Saturday
  - Julian Day of Year
  - Hour
  - Minute
  - Second
  - Millisecond
  - Real Time Clock
    - Time Zone Offset from UTC (minutes)
    - Time Zone Name
    - Daylight Savings Time Observed (boolean)
    - Time Trim Adjustment
  - Alarm Timer
    - Arm Alarm (boolean)
- Custom
  - Custom Usage
  - Custom Boolean Array
  - Custom Value
  - Custom Value 1
  - Custom Value 2
  - Custom Value 3
  - Custom Value 4
  - Custom Value 5
  - Custom Value 6
- Generic
  - Generic GUID or PROPERTYKEY kind (selection list)
    - Kind: Category
    - Kind: Type
    - Kind: Event
    - Kind: Property
    - Kind: Data Field
  - Generic GUID
    - Generic Category GUID
    - Generic Type GUID
  - Generic PROPERTYKEY
    - Generic Event PROPERTYKEY
    - Generic Property PROPERTYKEY
    - Generic Data Field PROPERTYKEY
  - Generic Top Level Collection ID
  - Generic Report ID
  - Generic Report Item Position Index
  - Generic Firmware VARTYPE (selection list)
    - VT_NULL: Empty
    - VT_BOOL: Boolean
    - VT_UI1: Byte
    - VT_I1: Character
    - VT_UI2: Unsigned Short
    - VT_I2: Short
    - VT_UI4: Unsigned Long
    - VT_I4: Long
    - VT_UI8: Unsigned Long Long
    - VT_I8: Long Long
    - VT_R4: Float
    - VT_R8: Double
    - VT_WSTR: Wide String
    - VT_STR: Narrow String
    - VT_CLSID: Guid
    - VT_VECTOR|VT_UI1: Opaque Structure
    - VT_F16E0: HID 16-bit Float with Unit Exponent 0
    - VT_F16E1: HID 16-bit Float with Unit Exponent 1
    - VT_F16E2: HID 16-bit Float with Unit Exponent 2
    - VT_F16E3: HID 16-bit Float with Unit Exponent 3
    - VT_F16E4: HID 16-bit Float with Unit Exponent 4
    - VT_F16E5: HID 16-bit Float with Unit Exponent 5
    - VT_F16E6: HID 16-bit Float with Unit Exponent 6
    - VT_F16E7: HID 16-bit Float with Unit Exponent 7
    - VT_F16E8: HID 16-bit Float with Unit Exponent 8
    - VT_F16E9: HID 16-bit Float with Unit Exponent 9
    - VT_F16EA: HID 16-bit Float with Unit Exponent A
    - VT_F16EB: HID 16-bit Float with Unit Exponent B
    - VT_F16EC: HID 16-bit Float with Unit Exponent C
    - VT_F16ED: HID 16-bit Float with Unit Exponent D
    - VT_F16EE: HID 16-bit Float with Unit Exponent E
    - VT_F16EF: HID 16-bit Float with Unit Exponent F
    - VT_F32E0: HID 32-bit Float with Unit Exponent 0
    - VT_F32E1: HID 32-bit Float with Unit Exponent 1
    - VT_F32E2: HID 32-bit Float with Unit Exponent 2
    - VT_F32E3: HID 32-bit Float with Unit Exponent 3
    - VT_F32E4: HID 32-bit Float with Unit Exponent 4
    - VT_F32E5: HID 32-bit Float with Unit Exponent 5
    - VT_F32E6: HID 32-bit Float with Unit Exponent 6
    - VT_F32E7: HID 32-bit Float with Unit Exponent 7
    - VT_F32E8: HID 32-bit Float with Unit Exponent 8
    - VT_F32E9: HID 32-bit Float with Unit Exponent 9
    - VT_F32EA: HID 32-bit Float with Unit Exponent A
    - VT_F32EB: HID 32-bit Float with Unit Exponent B
    - VT_F32EC: HID 32-bit Float with Unit Exponent C
    - VT_F32ED: HID 32-bit Float with Unit Exponent D
    - VT_F32EE: HID 32-bit Float with Unit Exponent E
    - VT_F32EF: HID 32-bit Float with Unit Exponent F
  - Generic Unit Of Measure (selection list)
    - Unit: Not Specified
    - Unit: Lux
    - Unit: Degrees Kelvin
    - Unit: Degrees Celsius

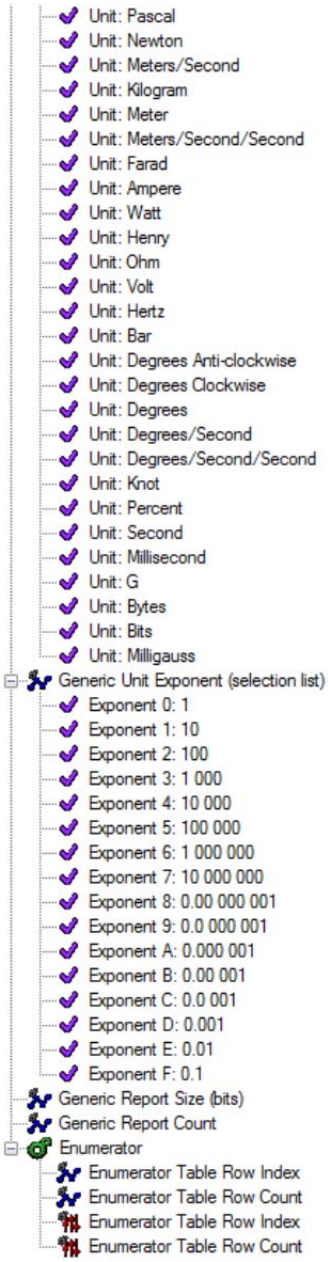

Figure 3. Sensor Properties, Data Fields, and Selection Values

3. Sensor Interaction via HID

This section describes how communication with Sensors is mapped to HID mechanisms. This section is Informative, which means it is only for orientation and guidance.

3.1 Related Documents

The following documents provide additional information related to or referenced by this document. All of the documents listed below are available on the Internet at http://www.usb.org.

[1] Universal Serial Bus Specification
[2] Device Class Definition for Human Interface Devices (HID)
[3] Universal Serial Bus HID Usage Tables
[4] Universal Serial Bus HID Point of Sale Usage Tables
[5] HUTTR30 Addition of usages related to touch digitizers
[6] HUTTR34 Addition of usages related to multi-touch digitizers
[7] Device Class Definition for Physical Interface Devices (PID)

3.2 Functional Overview

The Functional connection model for the Sensor class is below. There are two roles of communication. The host computer initiates requests. One or more sensor devices reply to the requests.

HID defines communication channels called Pipes. Every host and device must have a bidirectional Control Pipe. Devices may use additional Pipes for unidirectional communications.

For Sensor Devices, the HID Input Pipe is mandatory, and the HID Output Pipe is optional.

Each end of a Pipe is called an Endpoint.

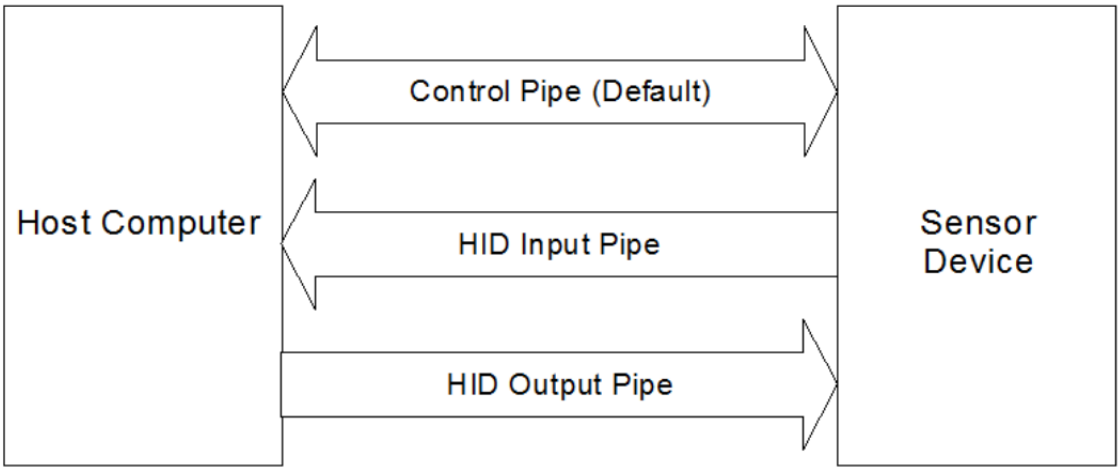

Figure 4. HID Functional Model

The data transfer mechanism for the Sensor class is based on the HID class Report Descriptors.

Communication with a Sensor device is identical to that of a HID device.

See Also

For more information about USB bus operation concepts and terminology; please refer to the *Universal Serial Bus Specification* (*Reference Document [1]*).

For more information on Human Interface Device concepts, terminology, and technical details; please refer to the *Device Class Definition for Human Interface Devices* specification (*Reference Document [2]*), including particularly Section 4.4.

3.3 HID Logical Devices

Upon first connection and/or detection, the Host Computer queries the Sensor Device for identification information. The device returns a HID Report Descriptor as defined by the HID Specification.

See Also
For more information about HID Report Descriptors; please refer to Section 6.2.2 of the *Device Class Definition for Human Interface Devices* specification (*Reference Document [2]*).

The HID Report Descriptor contains metadata describing the layout of one or more logical devices, the *Reports* that each logical device can transfer along with their *Report IDs*, and the characteristics of *Report Items* within each Report.

Important components of the HID Report Descriptor are one or more elements called *Collections*. A Collection can be used to describe a Sensor logical device.

Collections may be nested within other collections, and such are declared as either `Collection(Physical)` or `Collection(Logical)`.

The un-nested Collections (those not contained within any other Collection) are known as *Top Level Collections* (or "TLC" for short) and are customarily declared as `Collection(Application)`.

Top Level Collections are unique because most modern Operating Systems treat them as a logical device object; they expect and attempt to load a device driver to service the device.

There are two strategies that can be used to organize the definitions of sensors into Collections.

Strategy 1: nesting

A single Top Level Collection can be defined, where the Sensor logical devices are incorporated as nested sub-Collections.

The implications of this approach are that a single device driver will be loaded for the single Top Level Collection, and that device driver must parse the sub-Collections to determine how to communicate with the individual Sensor logical devices.

```
Usage Page (Sensors)
Usage (Sensor)
Collection (Application)
   Usage (Accelerometer)
   Collection (Physical)
      …
   End Collection
   Usage (Compass)
   Collection (Physical)
      …
   End Collection
   Usage (Gyro)
   Collection (Physical)
      …
   End Collection
End Collection
```

Figure 5. One TLC, 3 sub-Collections

Strategy 2: no nesting

Multiple Top Level Collections can be defined, one per Sensor logical device.

The implications of this approach are that a device driver must be loaded for each of the individual Top Level Collections.

They could be completely separate device drivers, one per Sensor logical device, or they could be multiple instances of the same device driver that is able to communicate with sensors of different types.

```
Usage Page (Sensors)
Usage (Accelerometer)
Collection (Application)
   …
End Collection
Usage (Compass)
Collection (Application)
   …
End Collection
Usage (Gyro)
Collection (Application)
   …
End Collection
```

Figure 6. Three TLCs

See Also
For full examples of HID Report Descriptors illustrating multiple HID sensor logical devices; please refer to Section 4.2.5, below.

3.4 HID Reports

HID communication transactions are accomplished by transferring Reports. On the USB bus, every HID Report begins with a SETUP packet on the Control Pipe, followed by a DATA packet on the Control Pipe, Input Pipe, or Output Pipe. Inside the SETUP packet are some fields that indicate:

- The HID-protocol request transfer type, such as GET_REPORT or SET_REPORT;
- The report type, such as Input, Output, or Feature;
- A numeric identifier for the report, called a Report ID, which is always the first byte of the of the DATA packet;
- The total length of the DATA packet (actual length sent, or expected length desired to receive).

On other busses, this discriminating information may be transmitted in a slightly different manner.

See Also

For more information about SETUP packets; please refer to Section 9.3 of the *Universal Serial Bus Specification* (*Reference Document [1]*).

For more information about HID Reports; please refer to Sections 5.2, 5.6, 7.2.1, and 7.2.2 of the *Device Class Definition for Human Interface Devices* specification (*Reference Document [2]*).

The combination of the transfer type and report type determine the nature of the communication transaction, as summarized in the following table.

Report Type	*Transfer for a GET_REPORT*	*Transfer for a SET_REPORT*
Input	Via: HID Interrupt In Pipe Data Direction: Device → Host "HID GET INPUT REPORT"	N/A
Output	N/A	Via: HID Interrupt Out pipe Data Direction: Host → Device "HID SET OUTPUT REPORT"
Feature	Via: Control pipe Data Direction: Device → Host "HID GET FEATURE REPORT"	Via: Control pipe Data Direction: Host → Device "HID SET FEATURE REPORT"

Table 6. HID Transfer and Report Types

3.5 HID Report IDs

The first byte of each report (byte position 0) is always the Report ID, which is used to differentiate reports from one another. Report IDs are assigned by the device implementer.

For a sensor physical device which represents more than one logical device, each of the logical devices must have a disjoint set of Report IDs. This can be done by judiciously assigning ranges of Report ID numbers for each logical device. The following table shows an example of only one of many possible Report ID allocation schemes that an implementer may choose.

Logical sensor device	**Report ID range**
First	0x10 – 0x1f
Second	0x20 – 0x2f

Third	0x30 – 0x3f

Table 7. A Report ID allocation scheme example

Because the SETUP packet has fields describing the request type separate from the Report ID, it is OK to "re use" Report IDs for Input Reports, Output Reports, and Feature Reports. For example, a sensor can have an Input Report with Report ID 0x01, an Output Report with Report ID 0x01, and a Feature Report with Report ID 0x01. These are all separate Reports with different contents. If you find this too confusing, you may wish to assign different Report IDs, for example: an Input Report with Report ID 0x01, an Output Report with Report ID 0x02, and a Feature Report with Report ID 0x03. Either technique is allowed.

If there is only one logical device, and that device only has a single Input Report, Output Report, and/or Feature Report, then the Report ID is not mandatory and must be filled with the special value zero.

3.6 HID Report Items

Each report may incorporate one or more pieces of data, called Items. The Items always follow the Report ID byte, i.e., the Items start at byte position 1 and work upward.

See Also
For more information about HID Report Items; please refer to Sections 5.2 and 5.3 of the *Device Class Definition for Human Interface Devices* specification (*Reference Document [2]*).

Input Reports are sent from the device to the host. Input Reports may be sent "asynchronously" by the device to the host (in actuality, the host bus controller periodically polls for these). The host may also "demand" an Input Report by invoking a HID GET INPUT REPORT request. Sensor ***Data Fields***, i.e., real-time acquired sensor data samples, are equated to HID ***Input Report Items***.

Output Reports are sent from the host to the device. Output Reports are optional. In the context of Sensors, Output Reports may be used to send functional commands to the Sensor by invoking the HID SET OUTPUT REPORT request. In practice, configuration of sensors is more commonly done using the HID SET FEATURE REPORT request instead.

See Also
For more information about HID Output Reports that may optionally be supported by some sensor device implementations; please refer to the *Device Class Definition for Physical Interface Devices* (PID) specification (*Reference Document [7]*). In that document, it describes how various types of Output Reports could be used for advanced features such as downloading detailed Haptic feedback waveforms, waveform shape "envelopes", and "condition" definitions that can be used as complex triggers. Please note that PID Usages appear on Usage Page 0x0F, not the Sensor Usage Page 0x20.

Feature Reports can be used to transfer non-real-time Properties between the device and the host. They are optional, meaning that it is possible to define a sensor collection that communicates solely with Input Reports. Feature Reports are also bi-directional, and sub-divided into *Get Feature Reports* and *Set Feature Reports*, depending upon direction.

Get Feature Reports are sent from the device to the host. They are used to get Properties from a sensor by invoking the HID GET FEATURE REPORT request. This might include identifying information such as the sensor's manufacturer and model number, or reading back previously set configuration settings. Sensor ***Properties*** are equated to HID ***Feature Report Items***.

Set Feature Reports are sent from the host to the device. They are used to set configuration Properties into a sensor by invoking the HID SET FEATURE REPORT. This might include the desired sample rate and reporting preferences. Sensor ***Properties*** are equated to HID ***Feature Report Items***.

3.6.1 HID Report Item packing options

Items come in varying sizes, depending upon their data types, and whether they are a scalar or an array. This is specified by the Report Size and Report Count values in the Report Descriptor.

Examples for common data types (non-exhaustive list):

Conventional Data Type	HID Report Size (bits)	HID Report Count
boolean[1]	1	1
unsigned char	8	1
char	8	1
unsigned short	16	1
short	16	1
unsigned long	32	1
long	32	1
unsigned long long	64	1
long long	64	1
float[2]	32	1
double	64	1
String; char[*n*]	8	*n*
Wide String; wchar_t[*n*]	16	*n*
Array of long; long[*n*]	32	*n*

Table 8. Common Data Types expressed as Report Size and Report Count

One strategy is to pack as many Items into a Report as will fit in the available space. Another strategy is to only put a single Item in each Report. The following are examples of each of these strategies.

Byte Position	Bit Position in the byte							
	7	6	5	4	3	2	1	0
0	Report ID = 0x01							
1	Data Field = 0x12345678							
2								
3								
4								

Table 9. Input Report with a single scalar Data Field of Report Size 32, Report Count 1

Byte Position	Bit Position in the byte							
	7	6	5	4	3	2	1	0
0	Report ID = 0x23							
1	Data Field 1 = 0xFF00							
2								

[1] For ease of programming, it is customary to byte-align Boolean values in HID reports. For example, if 3 Boolean bits are needed, they are followed by 5 unused "padding" bits so that the total fills up to an even byte boundary.

[2] HID has a special technique for representing floating point numbers, which is described in Section 4.2.1 *Values, Types, and Unit Exponents* below.

3	Data Field 2 = 0x5343
4	

Table 10. Input Report with 2 separate scalar Data Fields of Report Size 16, Report Count 1

Byte Position	Bit Position in the byte							
	7	6	5	4	3	2	1	0
0	Report ID = 0x12							
1	Data Field = ‘H’							
2	‘e’							
3	‘l’							
4	‘l’							
5	‘o’							
6	‘\0’							

Table 11. Input Report with a single array Data Field of Report Size 8, Report Count 6 (narrow character string “Hello”)

Byte Position	Bit Position in the byte							
	7	6	5	4	3	2	1	0
0	Report ID = 0x74							
1	Property = 0x00000000c4216527							
2								
3								
4								
5								
6								
7								
8								

Table 12. Feature Report with single scalar Property of Report Size 64, Report Count 1

Byte Position	Bit Position in the byte							
	7	6	5	4	3	2	1	0
0	Report ID = 0xa3							
1	Property 1 = 0x33							
2	Property 2 [0] = 0xFFFFFFFF							
3								
4								
5								
6	Property 2 [1] = 0x87654321							
7								
8								
9								

Table 13. Feature Report with 2 Properties, one a scalar of Report Size 8, Report Count 1 and one an array of Report Size 32, Report Count 2

Appendices at the end of this document show examples of HID Report Descriptors for various types of sensors. For simplicity’s sake, most of these examples are silent about the use of Report IDs. The examples presume:

- A Report ID of zero for both the Input Reports and Feature Reports;
- All the Data Fields are packed as Items into a single Input Report;

- All the Properties are packed as Items into a single Feature Report.

It should be appreciated by the reader that:

- A device with multiple sensors will be forced to use separate Report IDs to disambiguate the Reports (see Section 4.2.5 “Sensor Collections” below for an example of this);
- Even a device with a single sensor may use separate Report IDs for the Data Fields and Properties, as needed due to the size of the data to be transferred.

3.7 HID Usages

HID Collections and HID Report Items can be described by a number of different characteristics. These characteristics are described in the HID Report Descriptor. One of the most important of these is the *Usage*. The HID Usage tells what a given Collection or Report Item means (its semantics).

For:	The HID Usage appears:	And it tells:
Collections	Before the `Collection(Application)` or `Collection(Physical)` descriptor item.	What sensor *Type* the Collection describes.
Report Items	Before the `Input()` or `Feature()` descriptor item.[3]	What sensor *Property* or *Data Field* the Report Item describes.

Table 14. Usages applied to Collections and Report Items

See Also

For more information about HID Usages; please refer to the *USB HID Usage Tables* specification (*Reference Document [3]*).

For sensor-specific HID Usages; please refer to Section 1 *Sensor Usages*, above.

3.7.1 HID Usage Types

There are different varieties of usages when applied to Collections or Report Items. This is called the *Usage Type*, as listed in the following table:

Usage Type		Descriptor item flags	Description of HID use	Sensor-related use
CA	Collection	Application	Top level collection	*Sensor* object (*Category* or *Type*)
CP		Physical	Nested collection	
CL		Logical	Nested collection	
US	Usage Switch	*N/A*	Modifies a Usage	*Data Field* modifiers and threshold *Properties*
SF	Static Flag	Constant, Variable, Absolute	A read-only single bit value	Boolean *Data Field* or read-only *Property*
SV	Static Value	Constant, Variable, Absolute	A read-only multi-bit value	Other typed *Data Field* or read-only *Property*

[3] According to the *USB HID Usage Tables* specification (*Reference Document [3]*), a Usage may also be employed to describe an `Output()` descriptor item. But because `Output()` descriptor items are not defined for the HID Sensor Usage Page, it was not mentioned in the table.

DF	Dynamic Flag	Data, Variable, Absolute	A read/write single bit value	Boolean read/write *Property*
DV	Dynamic Value	Data, Variable, Absolute	A read/write multi-bit value	Other typed read/write *Property*
NAry	Named Array	Constant, Array, Absolute *or* Data, Array, Absolute	An enumerated value	Enumerated-value *Data Field* or *Property*
Sel	Selection	*N/A*	One of the selection choices for a Named Array	One of the selections for the enumerated-value of a *Data Field* or *Property*

Table 15. HID Usage Types

These Usage Types appear in the listed Sensor Usage Page (*see Section 1, above*). They give some information about how the different types of Usages should be employed.

See Also
For more information about HID Usage Types; please refer to Section 3.4.2, 3.4.3 of the *USB HID Usage Tables* specification (*Reference Document [3]*).

3.7.2 HID Selectors

One particularly interesting Usage Type is the *Selector* (marked as "Sel" in the tables) and its parent, the *Named Array* (marked as "NAry" in the tables).

For those familiar with high-level programming languages such as C++, the Named Array is analogous to a variable with an enumeration type; the Selectors are analogous to the various enumeration constants that the variable may take on.

One aspect that is unique about HID Selectors is that the actual numeric values of the Report Items that are passed in the HID Reports depend upon both the *order* in which the Selectors are declared, and interaction with the **Logical Minimum** and **Logical Maximum** declarations. Here is an example for the *Sensor State* Usage:

```
.
.
.
HID_LOGICAL_MIN_8(0),         // lowest numerical value of this enumeration will be 0
HID_LOGICAL_MAX_8(6),         // highest numerical value of this enumeration will be 6
HID_REPORT_SIZE(8),           // Input Report Item will be 8 bits long
HID_REPORT_COUNT(1),          // and there are 1 of them
HID_USAGE_SENSOR_STATE,       // NAry – the Current Sensor State reported as Data Field in Input Report
HID_COLLECTION(Logical),
          HID_USAGE_SENSOR_STATE_UNKNOWN,         // Sel - this will take on value 0
          HID_USAGE_SENSOR_STATE_READY,           // Sel - this will take on value 1
          HID_USAGE_SENSOR_STATE_NOT_AVAILABLE,   // Sel - this will take on value 2
          HID_USAGE_SENSOR_STATE_NO_DATA,         // Sel - this will take on value 3
          HID_USAGE_SENSOR_STATE_INITIALIZING,    // Sel - this will take on value 4
          HID_USAGE_SENSOR_STATE_ACCESS_DENIED,   // Sel - this will take on value 5
          HID_USAGE_SENSOR_STATE_ERROR,           // Sel - this will take on value 6
          HID_INPUT(Const_Arr_Abs),               // Input Item – takes on characteristics previously declared
HID_END_COLLECTION,
.
.
.
```

Note that for maximum interoperability of implementations, the following assumptions and restrictions are in force for sensors:

- The Named Array Usage is applied to a **Logical Collection**.
- All the defined Selectors for each Named Array must be declared (don't omit any);
- Always start the **Logical Minimum** at zero;

- Each Named Array takes on only one value ("radio button" style), so **Report Count** must be 1;
- Input Report Item or Feature Report Item must have "Array" flag (use **Const_Arr_Abs** or **Data_Arr_Abs**).

See Also
For more information about HID Selectors; please refer to Section 3.4.2.1, 3.4.3.1, and A.4 of the *USB HID Usage Tables* specification (*Reference Document [3]*).

3.8 HID Usage Page

All HID Usages for sensors are defined to occur on HID Usage Page 0x20.

3.9 HID Units

Usages have by definition a default Unit of Measure, which will be mentioned in the Usages Table.

For example, a Thermometer sensor may have a *Data Field* (Input Report Item) Usage called "Current Temperature", and its default Unit of Measure may be "Degrees Celsius".

If not explicitly overridden, the programmer should assume the default Unit of Measure.

It is possible to override the default Unit of Measure by explicitly including a `Unit()` descriptor item in the HID Report Descriptor. For example, "Degrees Celsius" could be overridden by declaring `Unit(Kelvin)`.

See Also
For more information about HID Units; please refer to Sections 6.2.2.7 of the *Device Class Definition for Human Interface Devices* specification (*Reference Document [2]*).

Usages state specifically whether *Units* are required to define a time base or other reference. A *Units* requirement implies the definition or Units, Physical Maximum, Physical Minimum and Unit Exponent descriptor items.

Units are **global** items that persist from main item to main item. *Units* can be disabled by setting Units to "None", and Physical Maximum, Physical Minimum and Unit Exponent equal to 0.

Normalized Usages use Logical Minimum and Logical Maximum to define their range of values. These values are automatically scaled by an application to fit the range of values required for the target operation and do not require *Units*. This is useful where, for example, the Usage is expressed as a Percent; use a Logical Minimum of 0 and a Logical Maximum of 100.

To declare a vector (Vx, Vy, or Vz) as a linear velocity or acceleration, define the *System* field of the Unit item associated with the vector usage to be either *SI Linear* (1) or *English Linear* (3). To declare vectors as angular velocity or acceleration, define the *System* field of the Unit item to be either *SI Rotation* (2) or *English Rotation* (4).

The HID Specification describes a scheme whereby complex Units of Measure can be constructed by composition of simple Units of Measure for various measurement dimensions (*System*, *Length*, *Mass*, *Time*, *Temperature*, *Current*, and *Luminosity*).

Here are some Units of Measure appropriate for sensors that can be used as overrides:

Sensor Unit of Measure	**What it measures**	**Composed as**	**Unit Exp.**	**"Nibble code" for measurement dimension**							**HID Report Descriptor Item**
				6 cd	5 A	4 K	3 s	2 g	1 cm	0 sys	
None	*anything*	*default for Usage*									`Unit(None)` `65 00`
Lux	Illuminance	candela / m²	2	1					-2	1	`Unit(Lux)` `67 e1 00 00 01`
Degrees Kelvin	Temperature	Kelvin	0			1				1	`Unit(Kelvin)` `67 01 00 01 00`
Degrees Fahrenheit	Temperature	Fahrenheit	0			1				3	`Unit(Fahrenheit)` `67 03 00 01 00`
Pascal	Pressure	kg / m · s²	5				-2	1	-1	1	`Unit(Pascal)` `66 f1 e1`
Newton	Force	kg · m / s²	5				-2	1	1	1	`Unit(Newton)` `66 11 e1`
m/s	Linear Velocity	m / s	2				-1		1	1	`Unit(M_per_s)` `66 11 f0`
m/s²	Linear Acceleration	m / s²	2				-2		1	1	`Unit(M_per_s2)` `66 11 e0`
Farad	Capacitance	s^4 · A² / kg · m²	5		2		4	-1	-2	1	`Unit(Farad)` `67 e1 4f 20 00`
Ampere	Current	A	0		1					1	`Unit(Ampere)` `67 01 00 10 00`
Watt	Electric Power	kg · m² / s³	5				-3	1	2	1	`Unit(Watt)` `66 21 d1`
Henry	Inductance	kg · m² / s² · A²	5		-2		-2	1	2	1	`Unit(Henry)` `67 21 e1 e0 00`
Ohm	Resistance	kg · m² / s³ · A²	5		-2		-3	1	2	1	`Unit(Ohm)` `67 21 d1 e0 00`
Volt	Voltage	kg · m² / s³ · A	5		-1		-3	1	2	1	`Unit(Volt)` `67 21 d1 f0 00`
Hertz	Frequency	1 / s	0				-1			1	`Unit(Hertz)` `66 01 f0`
Degrees	Angle	degrees	0						1	4	`Unit(Degree)` `65 14`
Degrees/s	Angular Velocity	degrees / s	0				-1		1	4	`Unit(D_per_s)` `66 14 f0`
Degrees/s²	Angular Acceleration	degrees / s²	0				-2		1	4	`Unit(D_per_s2)` `66 14 e0`
Radians	Angle	radians	0						1	2	`Unit(Radian)` `65 12`
Radians/s	Angular Velocity	radians / s	0				-1		1	2	`Unit(R_per_s)` `66 12 f0`
Radians/s²	Angular Acceleration	radians / s²	0				-2		1	2	`Unit(R_per_s2)` `66 12 e0`
second	Time	s	0				1			1	`Unit(Second)` `66 01 10`
Gauss	Magnetic Flux Density	g / s² · A	0		-1		-2	1		1	`Unit(Gauss)` `67 01 e1 f0 00`
Gram	Mass	g	0					1		1	`Unit(gram)` `66 01 01`
Centimeter	Distance	cm	0						1	1	`Unit(centimeter)` `65 11`

Table 16. Common Units of Measure and HID expressions

Note that without a multiplicative and/or additive constant, it is not possible to use *only* `Unit()` to represent an override Unit of Measure for:

- Degrees Celsius (Celsius = Kelvin - 273.15)
- Kilogram
  - Kilograms can be expressed as `Unit(gram)` with `UnitExponent(0x03)`
- Meters

  - Meters can be expressed as `Unit(centimeter)` with `UnitExponent(0x02)`
- Millibars, Bars
  - Millibars can be expressed as `Unit(Pascals)` with `UnitExponent(0x02)`
  - Bars can be expressed as `Unit(Pascals)` with `UnitExponent(0x05)`
- Knots (Knot = 1852 m / 3600 s)
- G's (G = 9.8 m/s²)
- Millisecond
  - Milliseconds can be expressed as `Unit(second)` with `UnitExponent(0x0D)`
- Milligauss
  - Milligauss can be expressed as `Unit(gauss)` with `UnitExponent(0x0D)`

In such cases, it is better for the default Unit of Measure to be one of these few above, and override it if necessary to a Unit of Measure for which there is a HID-expressible `Unit()`.

3.10 HID Unit Exponents

Unit Exponents can be used to scale the numeric value of a Report Item by a power of ten according to the following table:

Unit Exponent Argument	Power of Ten (Scientific Notation)	Power of Ten (Decimal Notation)
0x00	1 × 10E0	1
0x01	1 × 10E1	10
0x02	1 × 10E2	100
0x03	1 × 10E3	1 000
0x04	1 × 10E4	10 000
0x05	1 × 10E5	100 000
0x06	1 × 10E6	1 000 000
0x07	1 × 10E7	10 000 000
0x08	1 × 10E-8	0.00 000 001
0x09	1 × 10E-7	0.0 000 001
0x0A	1 × 10E-6	0.000 001
0x0B	1 × 10E-5	0.00 001
0x0C	1 × 10E-4	0.0 001
0x0D	1 × 10E-3	0.001
0x0E	1 × 10E-2	0.01
0x0F	1 × 10E-1	0.1

Table 17. HID Unit Exponent encoding and meanings

A Report Item that has a *default* Unit of Measure can be combined with a HID Unit Exponent to scale the Unit of Measure by a power of ten. Likewise, *explicit* Units of Measure declared with a `Unit()` descriptor item can be also combined with HID Unit Exponents to scale the Unit of Measure by a power of ten. For example:

- `Unit(Second)` with `UnitExponent(0x0D)` gives `Unit(Milliseconds)`,
- `Unit(gram)` with `UnitExponent(0x03)` gives `Unit(kilogram)`,
- `Unit(Centimeter)` with `UnitExponent(0x02)` gives `Unit(Meter)`.

See Also
For more information about HID Unit Exponents; please refer to Section 6.2.2.7 of the *Device Class*

Definition for Human Interface Devices specification (*Reference Document [2]*).

3.11 3D Coordinates and Compass Points

Some sensors (most notably: accelerometer, gyro, and compass) report X, Y, and Z coordinates in a 3D space.

At the chip-level, each of these sensors has their own individual and unique 3D coordinate systems that they use to report their "raw" data.

When reporting such coordinates in a HID Report, by convention they are expressed using a "X = East, Y = South, Z = Down" (or "ESD") ordering. This makes it necessary to re-order the coordinates from the "native" system of the chip to the HID ordering prior to transmission.

See Also
For more information about coordinate ordering; please refer to Section 5.9 of the *Device Class Definition for Human Interface Devices* specification (*Reference Document [2]*).

The HID coordinate ordering system is natural for use in 3D computer graphics.

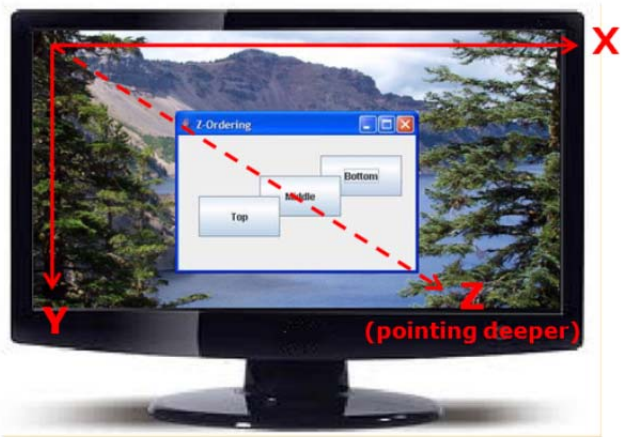

Figure 7. The 3D coordinate system used by computer graphics is "ESD"

Some application programmers may desire to have the X, Y, Z data in yet a different 3D coordinate system that makes more sense for any given application.

For example, navigation/mapping applications and 3D games typically prefer to use the same 3D coordinate system used by aeronautics (airplanes) and maritime (ships) that is called "NED" ("X = North, Y = East, Z = Down").

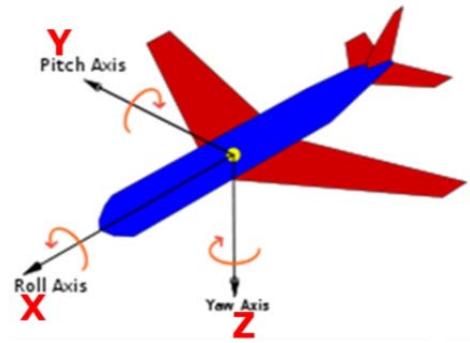

Figure 8. The preferred 3D coordinate system used by airplanes is “NED”

It is possible to translate between the “ESD” and the “NED” (or any other) coordinate system by using matrix arithmetic, for example:

$$\begin{bmatrix} X_{NED} \\ Y_{NED} \\ Z_{NED} \end{bmatrix} = \begin{bmatrix} X_{ESD} \\ Y_{ESD} \\ Z_{ESD} \end{bmatrix} * \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Figure 9. Rotation Matrix translation from "ESD" to "NED"

4. Illustrative Examples

This section is Informative, meaning that they provide unofficial guidelines for how to use the Usages in Section 1, above. It is offered to resolve potential ambiguities and to foster compatibility between separate implementations from multiple vendors.

Section 4.1 provides a sample “C” language “include file” which can be used by application developers.

Section 4.2 provides descriptions of some special constructions such as: Modifiers, Thresholds, Custom Sensors, and Generic Sensors.

Section 4.3 provides illustrative examples of HID Report Descriptors for various common sensor types. All of these examples employ use of the “include file” described in Section 4.1. Some of them employ use of the special constructions described in Section 4.2.

4.1 Include File Definitions

These definitions are used in common by all report descriptors in these Appendices. These #defines are intended to be compiled by an ANSI “C” compiler pre-processor.

```
////////////////////////////////////////////////////////////////////////////////
//
// HidSensorSpec.h : Defines compliant with HID Sensor Spec.
//
////////////////////////////////////////////////////////////////////////////////

#ifndef _HIDSENSORSPEC_H_
#define _HIDSENSORSPEC_H_

#define HID_USAGE_PAGE_SENSOR                                          0x05,0x20

//sensor category usages
#define HID_USAGE_SENSOR_TYPE_COLLECTION                               0x09,0x01
//sensor category biometric
#define HID_USAGE_SENSOR_CATEGORY_BIOMETRIC                            0x09,0x10
#define HID_USAGE_SENSOR_TYPE_BIOMETRIC_PRESENCE                       0x09,0x11
#define HID_USAGE_SENSOR_TYPE_BIOMETRIC_PROXIMITY                      0x09,0x12
#define HID_USAGE_SENSOR_TYPE_BIOMETRIC_TOUCH                          0x09,0x13
//sensor category electrical
#define HID_USAGE_SENSOR_CATEGORY_ELECTRICAL                           0x09,0x20
#define HID_USAGE_SENSOR_TYPE_ELECTRICAL_CAPACITANCE                   0x09,0x21
#define HID_USAGE_SENSOR_TYPE_ELECTRICAL_CURRENT                       0x09,0x22
#define HID_USAGE_SENSOR_TYPE_ELECTRICAL_POWER                         0x09,0x23
#define HID_USAGE_SENSOR_TYPE_ELECTRICAL_INDUCTANCE                    0x09,0x24
#define HID_USAGE_SENSOR_TYPE_ELECTRICAL_RESISTANCE                    0x09,0x25
#define HID_USAGE_SENSOR_TYPE_ELECTRICAL_VOLTAGE                       0x09,0x26
#define HID_USAGE_SENSOR_TYPE_ELECTRICAL_POTENTIOMETER                 0x09,0x27
#define HID_USAGE_SENSOR_TYPE_ELECTRICAL_FREQUENCY                     0x09,0x28
#define HID_USAGE_SENSOR_TYPE_ELECTRICAL_PERIOD                        0x09,0x29
//sensor category environmental
#define HID_USAGE_SENSOR_CATEGORY_ENVIRONMENTAL                        0x09,0x30
#define HID_USAGE_SENSOR_TYPE_ENVIRONMENTAL_ATMOSPHERIC_PRESSURE       0x09,0x31
#define HID_USAGE_SENSOR_TYPE_ENVIRONMENTAL_HUMIDITY                   0x09,0x32
#define HID_USAGE_SENSOR_TYPE_ENVIRONMENTAL_TEMPERATURE                0x09,0x33
#define HID_USAGE_SENSOR_TYPE_ENVIRONMENTAL_WIND_DIRECTION             0x09,0x34
#define HID_USAGE_SENSOR_TYPE_ENVIRONMENTAL_WIND_SPEED                 0x09,0x35
//sensor category light
#define HID_USAGE_SENSOR_CATEGORY_LIGHT                                0x09,0x40
#define HID_USAGE_SENSOR_TYPE_LIGHT_AMBIENTLIGHT                       0x09,0x41
#define HID_USAGE_SENSOR_TYPE_LIGHT_CONSUMER_INFRARED                  0x09,0x42
//sensor category location
#define HID_USAGE_SENSOR_CATEGORY_LOCATION                             0x09,0x50
#define HID_USAGE_SENSOR_TYPE_LOCATION_BROADCAST                       0x09,0x51
#define HID_USAGE_SENSOR_TYPE_LOCATION_DEAD_RECKONING                  0x09,0x52
#define HID_USAGE_SENSOR_TYPE_LOCATION_GPS                             0x09,0x53
#define HID_USAGE_SENSOR_TYPE_LOCATION_LOOKUP                          0x09,0x54
#define HID_USAGE_SENSOR_TYPE_LOCATION_OTHER                           0x09,0x55
#define HID_USAGE_SENSOR_TYPE_LOCATION_STATIC                          0x09,0x56
#define HID_USAGE_SENSOR_TYPE_LOCATION_TRIANGULATION                   0x09,0x57
//sensor category mechanical
#define HID_USAGE_SENSOR_CATEGORY_MECHANICAL                           0x09,0x60
#define HID_USAGE_SENSOR_TYPE_MECHANICAL_BOOLEAN_SWITCH                0x09,0x61
#define HID_USAGE_SENSOR_TYPE_MECHANICAL_BOOLEAN_SWITCH_ARRAY          0x09,0x62
#define HID_USAGE_SENSOR_TYPE_MECHANICAL_MULTIVALUE_SWITCH             0x09,0x63
#define HID_USAGE_SENSOR_TYPE_MECHANICAL_FORCE                         0x09,0x64
#define HID_USAGE_SENSOR_TYPE_MECHANICAL_PRESSURE                      0x09,0x65
#define HID_USAGE_SENSOR_TYPE_MECHANICAL_STRAIN                        0x09,0x66
```

```
#define HID_USAGE_SENSOR_TYPE_MECHANICAL_SCALE_WEIGHT                          0x09,0x67
#define HID_USAGE_SENSOR_TYPE_MECHANICAL_VIBRATOR                              0x09,0x68
#define HID_USAGE_SENSOR_TYPE_MECHANICAL_HALL_EFFECT_SWITCH                    0x09,0x69
//sensor category motion
#define HID_USAGE_SENSOR_CATEGORY_MOTION                                       0x09,0x70
#define HID_USAGE_SENSOR_TYPE_MOTION_ACCELEROMETER_1D                          0x09,0x71
#define HID_USAGE_SENSOR_TYPE_MOTION_ACCELEROMETER_2D                          0x09,0x72
#define HID_USAGE_SENSOR_TYPE_MOTION_ACCELEROMETER_3D                          0x09,0x73
#define HID_USAGE_SENSOR_TYPE_MOTION_GYROMETER_1D                              0x09,0x74
#define HID_USAGE_SENSOR_TYPE_MOTION_GYROMETER_2D                              0x09,0x75
#define HID_USAGE_SENSOR_TYPE_MOTION_GYROMETER_3D                              0x09,0x76
#define HID_USAGE_SENSOR_TYPE_MOTION_MOTION_DETECTOR                           0x09,0x77
#define HID_USAGE_SENSOR_TYPE_MOTION_SPEEDOMETER                               0x09,0x78
#define HID_USAGE_SENSOR_TYPE_MOTION_ACCELEROMETER                             0x09,0x79
#define HID_USAGE_SENSOR_TYPE_MOTION_GYROMETER                                 0x09,0x7A
//sensor category orientation
#define HID_USAGE_SENSOR_CATEGORY_ORIENTATION                                  0x09,0x80
#define HID_USAGE_SENSOR_TYPE_ORIENTATION_COMPASS_1D                           0x09,0x81
#define HID_USAGE_SENSOR_TYPE_ORIENTATION_COMPASS_2D                           0x09,0x82
#define HID_USAGE_SENSOR_TYPE_ORIENTATION_COMPASS_3D                           0x09,0x83
#define HID_USAGE_SENSOR_TYPE_ORIENTATION_INCLINOMETER_1D                      0x09,0x84
#define HID_USAGE_SENSOR_TYPE_ORIENTATION_INCLINOMETER_2D                      0x09,0x85
#define HID_USAGE_SENSOR_TYPE_ORIENTATION_INCLINOMETER_3D                      0x09,0x86
#define HID_USAGE_SENSOR_TYPE_ORIENTATION_DISTANCE_1D                          0x09,0x87
#define HID_USAGE_SENSOR_TYPE_ORIENTATION_DISTANCE_2D                          0x09,0x88
#define HID_USAGE_SENSOR_TYPE_ORIENTATION_DISTANCE_3D                          0x09,0x89
#define HID_USAGE_SENSOR_TYPE_ORIENTATION_DEVICE_ORIENTATION                   0x09,0x8A
#define HID_USAGE_SENSOR_TYPE_ORIENTATION_COMPASS                              0x09,0x8B
#define HID_USAGE_SENSOR_TYPE_ORIENTATION_INCLINOMETER                         0x09,0x8C
#define HID_USAGE_SENSOR_TYPE_ORIENTATION_DISTANCE                             0x09,0x8D
//sensor category scanner
#define HID_USAGE_SENSOR_CATEGORY_SCANNER                                      0x09,0x90
#define HID_USAGE_SENSOR_TYPE_SCANNER_BARCODE                                  0x09,0x91
#define HID_USAGE_SENSOR_TYPE_SCANNER_RFID                                     0x09,0x92
#define HID_USAGE_SENSOR_TYPE_SCANNER_NFC                                      0x09,0x93
//sensor category time
#define HID_USAGE_SENSOR_CATEGORY_TIME                                         0x09,0xA0
#define HID_USAGE_SENSOR_TYPE_TIME_ALARM                                       0x09,0xA1
#define HID_USAGE_SENSOR_TYPE_TIME_RTC                                         0x09,0xA2
//sensor category other
#define HID_USAGE_SENSOR_CATEGORY_OTHER                                        0x09,0xE0
#define HID_USAGE_SENSOR_TYPE_OTHER_CUSTOM                                     0x09,0xE1
#define HID_USAGE_SENSOR_TYPE_OTHER_GENERIC                                    0x09,0xE2
#define HID_USAGE_SENSOR_TYPE_OTHER_GENERIC_ENUMERATOR                         0x09,0xE3

//unit usages
#define HID_USAGE_SENSOR_UNITS_NOT_SPECIFIED                                   0x65,0x00                // Unit
#define HID_USAGE_SENSOR_UNITS_LUX                                             0x67,0xE1,0x00,0x00,0x01 // Unit
#define HID_USAGE_SENSOR_UNITS_KELVIN                                          0x67,0x01,0x00,0x01,0x00 // Unit
#define HID_USAGE_SENSOR_UNITS_FAHRENHEIT                                      0x67,0x03,0x00,0x01,0x00 // Unit
#define HID_USAGE_SENSOR_UNITS_PASCAL                                          0x66,0xF1,0xE1           // Unit
#define HID_USAGE_SENSOR_UNITS_NEWTON                                          0x66,0x11,0xE1           // Unit
#define HID_USAGE_SENSOR_UNITS_METERS_PER_SECOND                               0x66,0x11,0xF0           // Unit
#define HID_USAGE_SENSOR_UNITS_METERS_PER_SEC_SQRD                             0x66,0x11,0xE0           // Unit
#define HID_USAGE_SENSOR_UNITS_FARAD                                           0x67,0xE1,0x4F,0x20,0x00 // Unit
#define HID_USAGE_SENSOR_UNITS_AMPERE                                          0x67,0x01,0x00,0x10,0x00 // Unit
#define HID_USAGE_SENSOR_UNITS_WATT                                            0x66,0x21,0xD1           // Unit
#define HID_USAGE_SENSOR_UNITS_HENRY                                           0x67,0x21,0xE1,0xE0,0x00 // Unit
#define HID_USAGE_SENSOR_UNITS_OHM                                             0x67,0x21,0xD1,0xE0,0x00 // Unit
#define HID_USAGE_SENSOR_UNITS_VOLT                                            0x67,0x21,0xD1,0xF0,0x00 // Unit
#define HID_USAGE_SENSOR_UNITS_HERTZ                                           0x66,0x01,0xF0           // Unit
#define HID_USAGE_SENSOR_UNITS_DEGREES                                         0x65,0x14                // Unit
#define HID_USAGE_SENSOR_UNITS_DEGREES_PER_SECOND                              0x66,0x14,0xF0           // Unit
#define HID_USAGE_SENSOR_UNITS_DEGREES_PER_SEC_SQRD                            0x66,0x14,0xE0           // Unit
#define HID_USAGE_SENSOR_UNITS_RADIANS                                         0x65,0x12                // Unit
#define HID_USAGE_SENSOR_UNITS_RADIANS_PER_SECOND                              0x66,0x12,0xF0           // Unit
#define HID_USAGE_SENSOR_UNITS_RADIANS_PER_SEC_SQRD                            0x66,0x12,0xE0           // Unit
#define HID_USAGE_SENSOR_UNITS_SECOND                                          0x66,0x01,0x10           // Unit
#define HID_USAGE_SENSOR_UNITS_GAUSS                                           0x67,0x01,0xE1,0xF0,0x00 // Unit
#define HID_USAGE_SENSOR_UNITS_GRAM                                            0x66,0x01,0x01           // Unit
#define HID_USAGE_SENSOR_UNITS_CENTIMETER                                      0x65,0x11                // Unit
#ifdef DEFINE_NON_HID_UNITS
#define HID_USAGE_SENSOR_UNITS_CELSIUS                     “Use Unit(Kelvin) and subtract 273.15”
#define HID_USAGE_SENSOR_UNITS_KILOGRAM                    “Use Unit(gram) and UnitExponent(0x03)”
#define HID_USAGE_SENSOR_UNITS_METER                       “Use Unit(centimeter) and UnitExponent(0x02)”
#define HID_USAGE_SENSOR_UNITS_BAR                         “Use Unit(Pascal) and UnitExponent(0x05)”
#define HID_USAGE_SENSOR_UNITS_KNOT                        “Use Unit(m/s) and multiply by 1852/3600”
#define HID_USAGE_SENSOR_UNITS_PERCENT                     “Use Unit(Not_Specified)”
#define HID_USAGE_SENSOR_UNITS_G                           “Use Unit(m/s2) and divide by 9.8”
#define HID_USAGE_SENSOR_UNITS_MILLISECOND                 “Use Unit(second) and UnitExponent(0x0D)”
#define HID_USAGE_SENSOR_UNITS_MILLIGAUSS                  “Use Unit(Gauss) and UnitExponent(0x0D)”
#endif

//data type usage switches -- we use them as modifiers for sensor properties & data fields
//to create thresholds, for example.
//NOTE: the usage tables actually define these as two bytes, but in order
//to get the define macros to work so these are ‘or-ed’ these are defined
//here as only one byte.
#define HID_USAGE_SENSOR_DATA_MOD_NONE                                         0x00 // US
#define HID_USAGE_SENSOR_DATA_MOD_CHANGE_SENSITIVITY_ABS                       0x10 // US
#define HID_USAGE_SENSOR_DATA_MOD_MAX                                          0x20 // US
#define HID_USAGE_SENSOR_DATA_MOD_MIN                                          0x30 // US
#define HID_USAGE_SENSOR_DATA_MOD_ACCURACY                                     0x40 // US
#define HID_USAGE_SENSOR_DATA_MOD_RESOLUTION                                   0x50 // US
#define HID_USAGE_SENSOR_DATA_MOD_THRESHOLD_HIGH                               0x60 // US
```

```
#define HID_USAGE_SENSOR_DATA_MOD_THRESHOLD_LOW                                0x70 // US
#define HID_USAGE_SENSOR_DATA_MOD_CALIBRATION_OFFSET                           0x80 // US
#define HID_USAGE_SENSOR_DATA_MOD_CALIBRATION_MULTIPLIER                       0x90 // US
#define HID_USAGE_SENSOR_DATA_MOD_REPORT_INTERVAL                              0xA0 // US
#define HID_USAGE_SENSOR_DATA_MOD_FREQUENCY_MAX                                0xB0 // US
#define HID_USAGE_SENSOR_DATA_MOD_PERIOD_MAX                                   0xC0 // US
#define HID_USAGE_SENSOR_DATA_MOD_CHANGE_SENSITIVITY_RANGE_PCT                 0xD0 // US
#define HID_USAGE_SENSOR_DATA_MOD_CHANGE_SENSITIVITY_REL_PCT                   0xE0 // US
#define HID_USAGE_SENSOR_DATA_MOD_VENDOR_RESERVED                              0xF0 // US

//state usages
#define HID_USAGE_SENSOR_STATE                                                 0x0A,0x01,0x02 // NAry
//state selectors
#define HID_USAGE_SENSOR_STATE_UNKNOWN                                         0x0A,0x00,0x08 // Sel
#define HID_USAGE_SENSOR_STATE_READY                                           0x0A,0x01,0x08 // Sel
#define HID_USAGE_SENSOR_STATE_NOT_AVAILABLE                                   0x0A,0x02,0x08 // Sel
#define HID_USAGE_SENSOR_STATE_NO_DATA                                         0x0A,0x03,0x08 // Sel
#define HID_USAGE_SENSOR_STATE_INITIALIZING                                    0x0A,0x04,0x08 // Sel
#define HID_USAGE_SENSOR_STATE_ACCESS_DENIED                                   0x0A,0x05,0x08 // Sel
#define HID_USAGE_SENSOR_STATE_ERROR                                           0x0A,0x06,0x08 // Sel

//event usages
#define HID_USAGE_SENSOR_EVENT                                                 0x0A,0x02,0x02 // NAry
//event selectors
#define HID_USAGE_SENSOR_EVENT_UNKNOWN                                         0x0A,0x10,0x08 // Sel
#define HID_USAGE_SENSOR_EVENT_STATE_CHANGED                                   0x0A,0x11,0x08 // Sel
#define HID_USAGE_SENSOR_EVENT_PROPERTY_CHANGED                                0x0A,0x12,0x08 // Sel
#define HID_USAGE_SENSOR_EVENT_DATA_UPDATED                                    0x0A,0x13,0x08 // Sel
#define HID_USAGE_SENSOR_EVENT_POLL_RESPONSE                                   0x0A,0x14,0x08 // Sel
#define HID_USAGE_SENSOR_EVENT_CHANGE_SENSITIVITY                              0x0A,0x15,0x08 // Sel
#define HID_USAGE_SENSOR_EVENT_MAX_REACHED                                     0x0A,0x16,0x08 // Sel
#define HID_USAGE_SENSOR_EVENT_MIN_REACHED                                     0x0A,0x17,0x08 // Sel
#define HID_USAGE_SENSOR_EVENT_HIGH_THRESHOLD_CROSS_UPWARD                     0x0A,0x18,0x08 // Sel
#define HID_USAGE_SENSOR_EVENT_HIGH_THESHOLD_CROSS_ABOVE    HID_USAGE_SENSOR_EVENT_HIGH_THRESHOLD_CROSS_UPWARD
#define HID_USAGE_SENSOR_EVENT_HIGH_THRESHOLD_CROSS_DOWNWARD                   0x0A,0x19,0x08 // Sel
#define HID_USAGE_SENSOR_EVENT_HIGH_THRESHOLD_CROSS_BELOW   HID_USAGE_SENSOR_EVENT_HIGH_THRESHOLD_CROSS_DOWNWARD
#define HID_USAGE_SENSOR_EVENT_LOW_THRESHOLD_CROSS_UPWARD                      0x0A,0x1A,0x08 // Sel
#define HID_USAGE_SENSOR_EVENT_LOW_THRESHOLD_CROSS_ABOVE    HID_USAGE_SENSOR_EVENT_LOW_THRESHOLD_CROSS_UPWARD
#define HID_USAGE_SENSOR_EVENT_LOW_THRESHOLD_CROSS_DOWNWARD                    0x0A,0x1B,0x08 // Sel
#define HID_USAGE_SENSOR_EVENT_LOW_THRESHOLD_CROSS_BELOW    HID_USAGE_SENSOR_EVENT_LOW_THRESHOLD_CROSS_DOWNWARD
#define HID_USAGE_SENSOR_EVENT_ZERO_THRESHOLD_CROSS_UPWARD                     0x0A,0x1C,0x08 // Sel
#define HID_USAGE_SENSOR_EVENT_ZERO_THRESHOLD_CROSS_ABOVE   HID_USAGE_SENSOR_EVENT_ZERO_THRESHOLD_CROSS_UPWARD
#define HID_USAGE_SENSOR_EVENT_ZERO_THRESHOLD_CROSS_DOWNWARD                   0x0A,0x1D,0x08 // Sel
#define HID_USAGE_SENSOR_EVENT_ZERO_THRESHOLD_CROSS_BELOW   HID_USAGE_SENSOR_EVENT_ZERO_THRESHOLD_CROSS_DOWNWARD
#define HID_USAGE_SENSOR_EVENT_PERIOD_EXCEEDED                                 0x0A,0x1E,0x08 // Sel
#define HID_USAGE_SENSOR_EVENT_FREQUENCY_EXCEEDED                              0x0A,0x1F,0x08 // Sel
#define HID_USAGE_SENSOR_EVENT_COMPLEX_TRIGGER                                 0x0A,0x20,0x08 // Sel

//property usages (get/set feature report)
#define HID_USAGE_SENSOR_PROPERTY                                              0x0A,0x00,0x03
#define HID_USAGE_SENSOR_PROPERTY_FRIENDLY_NAME                                0x0A,0x01,0x03
#define HID_USAGE_SENSOR_PROPERTY_PERSISTENT_UNIQUE_ID                         0x0A,0x02,0x03
#define HID_USAGE_SENSOR_PROPERTY_SENSOR_STATUS                                0x0A,0x03,0x03
#define HID_USAGE_SENSOR_PROPERTY_MINIMUM_REPORT_INTERVAL                      0x0A,0x04,0x03
#define HID_USAGE_SENSOR_PROPERTY_SENSOR_MANUFACTURER                          0x0A,0x05,0x03
#define HID_USAGE_SENSOR_PROPERTY_SENSOR_MODEL                                 0x0A,0x06,0x03
#define HID_USAGE_SENSOR_PROPERTY_SENSOR_SERIAL_NUMBER                         0x0A,0x07,0x03
#define HID_USAGE_SENSOR_PROPERTY_SENSOR_DESCRIPTION                           0x0A,0x08,0x03
#define HID_USAGE_SENSOR_PROPERTY_SENSOR_CONNECTION_TYPE                       0x0A,0x09,0x03 // NAry
//begin connection type selectors
#define HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_INTEGRATED                0x0A,0x30,0x08 // Sel
#define HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_ATTACHED                  0x0A,0x31,0x08 // Sel
#define HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_EXTERNAL                  0x0A,0x32,0x08 // Sel
//end connection type selectors
#define HID_USAGE_SENSOR_PROPERTY_SENSOR_DEVICE_PATH                           0x0A,0x0A,0x03
#define HID_USAGE_SENSOR_PROPERTY_HARDWARE_REVISION                            0x0A,0x0B,0x03
#define HID_USAGE_SENSOR_PROPERTY_FIRMWARE_VERSION                             0x0A,0x0C,0x03
#define HID_USAGE_SENSOR_PROPERTY_RELEASE_DATE                                 0x0A,0x0D,0x03
#define HID_USAGE_SENSOR_PROPERTY_REPORT_INTERVAL                              0x0A,0x0E,0x03
#define HID_USAGE_SENSOR_PROPERTY_CHANGE_SENSITIVITY_ABS                       0x0A,0x0F,0x03
#define HID_USAGE_SENSOR_PROPERTY_CHANGE_SENSITIVITY_RANGE_PCT                 0x0A,0x10,0x03
#define HID_USAGE_SENSOR_PROPERTY_CHANGE_SENSITIVITY_REL_PCT                   0x0A,0x11,0x03
#define HID_USAGE_SENSOR_PROPERTY_ACCURACY                                     0x0A,0x12,0x03
#define HID_USAGE_SENSOR_PROPERTY_RESOLUTION                                   0x0A,0x13,0x03
#define HID_USAGE_SENSOR_PROPERTY_RANGE_MAXIMUM                                0x0A,0x14,0x03
#define HID_USAGE_SENSOR_PROPERTY_RANGE_MINIMUM                                0x0A,0x15,0x03
#define HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE                              0x0A,0x16,0x03 // NAry
//begin reporting state selectors
#define HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_NO_EVENTS                    0x0A,0x40,0x08 // Sel
#define HID_USAGE_REPORTING_STATE_ON_NONE                   HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_NO_EVENTS
#define HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS                   0x0A,0x41,0x08 // Sel
#define HID_USAGE_REPORTING_STATE_ON_ALL                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS
#define HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_THRESHOLD_EVENTS             0x0A,0x42,0x08 // Sel
#define HID_USAGE_REPORTING_STATE_ON_THRESHOLD              HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_THRESHOLD_EVENTS
#define HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_NO_EVENTS_WAKE               0x0A,0x43,0x08 // Sel
#define HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS_WAKE              0x0A,0x44,0x08 // Sel
#define HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_THRESHOLD_EVENTS_WAKE        0x0A,0x45,0x08 // Sel
//end reporting state selectors
#define HID_USAGE_SENSOR_PROPERTY_SAMPLING_RATE                                0x0A,0x17,0x03
#define HID_USAGE_SENSOR_PROPERTY_RESPONSE_CURVE                               0x0A,0x18,0x03
#define HID_USAGE_SENSOR_PROPERTY_POWER_STATE                                  0x0A,0x19,0x03 // NAry
//begin power state selectors
#define HID_USAGE_SENSOR_PROPERTY_POWER_STATE_UNDEFINED                        0x0A,0x50,0x08 // Sel
#define HID_USAGE_SENSOR_PROPERTY_POWER_STATE_D0_FULL_POWER                    0x0A,0x51,0x08 // Sel
#define HID_USAGE_SENSOR_PROPERTY_POWER_STATE_D1_LOW_POWER                     0x0A,0x52,0x08 // Sel
```

```
#define HID_USAGE_SENSOR_PROPERTY_POWER_STATE_D2_STANDBY_WITH_WAKE          0x0A,0x53,0x08 // Sel
#define HID_USAGE_SENSOR_PROPERTY_POWER_STATE_D3_SLEEP_WITH_WAKE            0x0A,0x54,0x08 // Sel
#define HID_USAGE_SENSOR_PROPERTY_POWER_STATE_D4_POWER_OFF                  0x0A,0x55,0x08 // Sel
//end power state selectors

//data type location
//data field usages (input report
#define HID_USAGE_SENSOR_DATA_LOCATION                                      0x0A,0x00,0x04
#define HID_USAGE_SENSOR_DATA_LOCATION_DESIRED_ACCURACY                     0x0A,0x01,0x04
#define HID_USAGE_SENSOR_DATA_LOCATION_ALTITUDE_ANTENNA_SEALEVEL            0x0A,0x02,0x04
#define HID_USAGE_SENSOR_DATA_LOCATION_DIFFERENTIAL_REFERENCE_STATION_ID    0x0A,0x03,0x04
#define HID_USAGE_SENSOR_DATA_LOCATION_ALTITIDE_ELIPSOID_ERROR              0x0A,0x04,0x04
#define HID_USAGE_SENSOR_DATA_LOCATION_ALTITIDE_ELIPSOID                    0x0A,0x05,0x04
#define HID_USAGE_SENSOR_DATA_LOCATION_ALTITUDE_SEALEVEL_ERROR              0x0A,0x06,0x04
#define HID_USAGE_SENSOR_DATA_LOCATION_ALTITUDE_SEALEVEL                    0x0A,0x07,0x04
#define HID_USAGE_SENSOR_DATA_LOCATION_DGPS_DATA_AGE                        0x0A,0x08,0x04
#define HID_USAGE_SENSOR_DATA_LOCATION_ERROR_RADIUS                         0x0A,0x09,0x04
#define HID_USAGE_SENSOR_DATA_LOCATION_FIX_QUALITY                          0x0A,0x0A,0x04 // NAry
//begin fix quality selectors
#define HID_USAGE_SENSOR_DATA_FIX_QUALITY_NO_FIX                            0x0A,0x70,0x08 // Sel
#define HID_USAGE_SENSOR_DATA_FIX_QUALITY_GPS                               0x0A,0x71,0x08 // Sel
#define HID_USAGE_SENSOR_DATA_FIX_QUALITY_DGPS                              0x0A,0x72,0x08 // Sel
//end fix quality selectors
#define HID_USAGE_SENSOR_DATA_LOCATION_FIX_TYPE                             0x0A,0x0B,0x04 // NAry
//begin fix type selectors
#define HID_USAGE_SENSOR_DATA_FIX_TYPE_NO_FIX                               0x0A,0x80,0x08 // Sel
#define HID_USAGE_SENSOR_DATA_FIX_TYPE_GPS_SPS_MODE_FIX_VALID               0x0A,0x81,0x08 // Sel
#define HID_USAGE_SENSOR_DATA_FIX_TYPE_DGPS_SPS_MODE_FIX_VALID              0x0A,0x82,0x08 // Sel
#define HID_USAGE_SENSOR_DATA_FIX_TYPE_GPS_PPS_MODE_FIX_VALID               0x0A,0x83,0x08 // Sel
#define HID_USAGE_SENSOR_DATA_FIX_TYPE_REAL_TIME_KINEMATIC                  0x0A,0x84,0x08 // Sel
#define HID_USAGE_SENSOR_DATA_FIX_TYPE_FLOAT_RTK                            0x0A,0x85,0x08 // Sel
#define HID_USAGE_SENSOR_DATA_FIX_TYPE_ESTIMATED_DEAD_RECKONING             0x0A,0x86,0x08 // Sel
#define HID_USAGE_SENSOR_DATA_FIX_TYPE_MANUAL_INPUT_MODE                    0x0A,0x87,0x08 // Sel
#define HID_USAGE_SENSOR_DATA_FIX_TYPE_SIMULATOR_MODE                       0x0A,0x88,0x08 // Sel
//end fix type selectors
#define HID_USAGE_SENSOR_DATA_LOCATION_GEOIDAL_SEPARATION                   0x0A,0x0C,0x04
#define HID_USAGE_SENSOR_DATA_LOCATION_GPS_OPERATION_MODE                   0x0A,0x0D,0x04 // NAry
//begin gps operation mode selectors
#define HID_USAGE_SENSOR_DATA_GPS_OP_MODE_MANUAL                            0x0A,0x90,0x08 // Sel
#define HID_USAGE_SENSOR_DATA_GPS_OP_MODE_AUTOMATIC                         0x0A,0x91,0x08 // Sel
//end gps operation mode selectors
#define HID_USAGE_SENSOR_DATA_LOCATION_GPS_SELECTION_MODE                   0x0A,0x0E,0x04 // NAry
//begin gps selection mode selectors
#define HID_USAGE_SENSOR_DATA_GPS_SEL_MODE_AUTONOMOUS                       0x0A,0xA0,0x08 // Sel
#define HID_USAGE_SENSOR_DATA_GPS_SEL_MODE_DGPS                             0x0A,0xA1,0x08 // Sel
#define HID_USAGE_SENSOR_DATA_GPS_SEL_MODE_ESTIMATED_DEAD_RECKONING         0x0A,0xA2,0x08 // Sel
#define HID_USAGE_SENSOR_DATA_GPS_SEL_MODE_MANUAL_INPUT                     0x0A,0xA3,0x08 // Sel
#define HID_USAGE_SENSOR_DATA_GPS_SEL_MODE_SIMULATOR                        0x0A,0xA4,0x08 // Sel
#define HID_USAGE_SENSOR_DATA_GPS_SEL_MODE_DATA_NOT_VALID                   0x0A,0xA5,0x08 // Sel
//end gps selection mode selectors
#define HID_USAGE_SENSOR_DATA_LOCATION_GPS_STATUS                           0x0A,0x0F,0x04 // NAry
//begin gps status selectors
#define HID_USAGE_SENSOR_DATA_GPS_STATUS_DATA_VALID                         0x0A,0xB0,0x08 // Sel
#define HID_USAGE_SENSOR_DATA_GPS_STATUS_DATA_NOT_VALID                     0x0A,0xB1,0x08 // Sel
//end gps status selectors
#define HID_USAGE_SENSOR_DATA_LOCATION_POSITION_DILUTION_OF_PRECISION       0x0A,0x10,0x04
#define HID_USAGE_SENSOR_DATA_LOCATION_HORIZONTAL_DILUTION_OF_PRECISION     0x0A,0x11,0x04
#define HID_USAGE_SENSOR_DATA_LOCATION_VERTICAL_DILUTION_OF_PRECISION       0x0A,0x12,0x04
#define HID_USAGE_SENSOR_DATA_LOCATION_LATITUDE                             0x0A,0x13,0x04
#define HID_USAGE_SENSOR_DATA_LOCATION_LONGITUDE                            0x0A,0x14,0x04
#define HID_USAGE_SENSOR_DATA_LOCATION_TRUE_HEADING                         0x0A,0x15,0x04
#define HID_USAGE_SENSOR_DATA_LOCATION_MAGNETIC_HEADING                     0x0A,0x16,0x04
#define HID_USAGE_SENSOR_DATA_LOCATION_MAGNETIC_VARIATION                   0x0A,0x17,0x04
#define HID_USAGE_SENSOR_DATA_LOCATION_SPEED                                0x0A,0x18,0x04
#define HID_USAGE_SENSOR_DATA_LOCATION_SATELLITES_IN_VIEW                   0x0A,0x19,0x04
#define HID_USAGE_SENSOR_DATA_LOCATION_SATELLITES_IN_VIEW_AZIMUTH           0x0A,0x1A,0x04
#define HID_USAGE_SENSOR_DATA_LOCATION_SATELLITES_IN_VIEW_ELEVATION         0x0A,0x1B,0x04
#define HID_USAGE_SENSOR_DATA_LOCATION_SATELLITES_IN_VIEW_ID                0x0A,0x1C,0x04
#define HID_USAGE_SENSOR_DATA_LOCATION_SATELLITES_IN_VIEW_PRNs              0x0A,0x1D,0x04
#define HID_USAGE_SENSOR_DATA_LOCATION_SATELLITES_IN_VIEW_STN_RATIO         0x0A,0x1E,0x04
#define HID_USAGE_SENSOR_DATA_LOCATION_SATELLITES_USED_COUNT                0x0A,0x1F,0x04
#define HID_USAGE_SENSOR_DATA_LOCATION_SATELLITES_USED_PRNs                 0x0A,0x20,0x04
#define HID_USAGE_SENSOR_DATA_LOCATION_NMEA_SENTENCE                        0x0A,0x21,0x04
#define HID_USAGE_SENSOR_DATA_LOCATION_ADDRESS_LINE_1                       0x0A,0x22,0x04
#define HID_USAGE_SENSOR_DATA_LOCATION_ADDRESS_LINE_2                       0x0A,0x23,0x04
#define HID_USAGE_SENSOR_DATA_LOCATION_CITY                                 0x0A,0x24,0x04
#define HID_USAGE_SENSOR_DATA_LOCATION_STATE_OR_PROVINCE                    0x0A,0x25,0x04
#define HID_USAGE_SENSOR_DATA_LOCATION_COUNTRY_OR_REGION                    0x0A,0x26,0x04
#define HID_USAGE_SENSOR_DATA_LOCATION_POSTAL_CODE                          0x0A,0x27,0x04
//property usages (get/set feature report)
#define HID_USAGE_SENSOR_PROPERTY_LOCATION                                  0x0A,0x2A,0x04
#define HID_USAGE_SENSOR_PROPERTY_LOCATION_DESIRED_ACCURACY                 0x0A,0x2B,0x04 // NAry
//begin location desired accuracy selectors
#define HID_USAGE_SENSOR_DESIRED_ACCURACY_DEFAULT                           0x0A,0x60,0x08 // Sel
#define HID_USAGE_SENSOR_DESIRED_ACCURACY_HIGH                              0x0A,0x61,0x08 // Sel
#define HID_USAGE_SENSOR_DESIRED_ACCURACY_MEDIUM                            0x0A,0x62,0x08 // Sel
#define HID_USAGE_SENSOR_DESIRED_ACCURACY_LOW                               0x0A,0x63,0x08 // Sel
//end location desired accuracy selectors

//data type environmental
//data field usages (input report)
#define HID_USAGE_SENSOR_DATA_ENVIRONMENTAL                                 0x0A,0x30,0x04
#define HID_USAGE_SENSOR_DATA_ENVIRONMENTAL_ATMOSPHERIC_PRESSURE            0x0A,0x31,0x04
#define HID_USAGE_SENSOR_DATA_ENVIRONMENTAL_REFERENCE_PRESSURE              0x0A,0x32,0x04
#define HID_USAGE_SENSOR_DATA_ENVIRONMENTAL_RELATIVE_HUMIDITY               0x0A,0x33,0x04
```

```
#define HID_USAGE_SENSOR_DATA_ENVIRONMENTAL_TEMPERATURE                    0x0A,0x34,0x04
#define HID_USAGE_SENSOR_DATA_ENVIRONMENTAL_WIND_DIRECTION                 0x0A,0x35,0x04
#define HID_USAGE_SENSOR_DATA_ENVIRONMENTAL_WIND_SPEED                     0x0A,0x36,0x04
//property usages (get/set feature report)
#define HID_USAGE_SENSOR_PROPERTY_ENVIRONMENTAL                            0x0A,0x40,0x04
#define HID_USAGE_SENSOR_PROPERTY_ENVIRONMENTAL_REFERENCE_PRESSURE         0x0A,0x41,0x04

//data type motion
//data field usages (input report)
#define HID_USAGE_SENSOR_DATA_MOTION                                       0x0A,0x50,0x04
#define HID_USAGE_SENSOR_DATA_MOTION_STATE                                 0x0A,0x51,0x04
#define HID_USAGE_SENSOR_DATA_MOTION_ACCELERATION                          0x0A,0x52,0x04
#define HID_USAGE_SENSOR_DATA_MOTION_ACCELERATION_X_AXIS                   0x0A,0x53,0x04
#define HID_USAGE_SENSOR_DATA_MOTION_ACCELERATION_Y_AXIS                   0x0A,0x54,0x04
#define HID_USAGE_SENSOR_DATA_MOTION_ACCELERATION_Z_AXIS                   0x0A,0x55,0x04
#define HID_USAGE_SENSOR_DATA_MOTION_ANGULAR_VELOCITY                      0x0A,0x56,0x04
#define HID_USAGE_SENSOR_DATA_MOTION_ANGULAR_VELOCITY_X_AXIS               0x0A,0x57,0x04
#define HID_USAGE_SENSOR_DATA_MOTION_ANGULAR_VELOCITY_Y_AXIS               0x0A,0x58,0x04
#define HID_USAGE_SENSOR_DATA_MOTION_ANGULAR_VELOCITY_Z_AXIS               0x0A,0x59,0x04
#define HID_USAGE_SENSOR_DATA_MOTION_ANGULAR_POSITION                      0x0A,0x5A,0x04
#define HID_USAGE_SENSOR_DATA_MOTION_ANGULAR_POSITION_X_AXIS               0x0A,0x5B,0x04
#define HID_USAGE_SENSOR_DATA_MOTION_ANGULAR_POSITION_Y_AXIS               0x0A,0x5C,0x04
#define HID_USAGE_SENSOR_DATA_MOTION_ANGULAR_POSITION_Z_AXIS               0x0A,0x5D,0x04
#define HID_USAGE_SENSOR_DATA_MOTION_SPEED                                 0x0A,0x5E,0x04
#define HID_USAGE_SENSOR_DATA_MOTION_INTENSITY                             0x0A,0x5F,0x04

//data type orientation
//data field usages (input report)
#define HID_USAGE_SENSOR_DATA_ORIENTATION                                  0x0A,0x70,0x04
#define HID_USAGE_SENSOR_DATA_ORIENTATION_MAGNETIC_HEADING                 0x0A,0x71,0x04
#define HID_USAGE_SENSOR_DATA_ORIENTATION_MAGNETIC_HEADING_X               0x0A,0x72,0x04
#define HID_USAGE_SENSOR_DATA_ORIENTATION_MAGNETIC_HEADING_Y               0x0A,0x73,0x04
#define HID_USAGE_SENSOR_DATA_ORIENTATION_MAGNETIC_HEADING_Z               0x0A,0x74,0x04
#define HID_USAGE_SENSOR_DATA_ORIENTATION_COMPENSATED_MAGNETIC_NORTH       0x0A,0x75,0x04
#define HID_USAGE_SENSOR_DATA_ORIENTATION_COMPENSATED_TRUE_NORTH           0x0A,0x76,0x04
#define HID_USAGE_SENSOR_DATA_ORIENTATION_MAGNETIC_NORTH                   0x0A,0x77,0x04
#define HID_USAGE_SENSOR_DATA_ORIENTATION_TRUE_NORTH                       0x0A,0x78,0x04
#define HID_USAGE_SENSOR_DATA_ORIENTATION_DISTANCE                         0x0A,0x79,0x04
#define HID_USAGE_SENSOR_DATA_ORIENTATION_DISTANCE_X                       0x0A,0x7A,0x04
#define HID_USAGE_SENSOR_DATA_ORIENTATION_DISTANCE_Y                       0x0A,0x7B,0x04
#define HID_USAGE_SENSOR_DATA_ORIENTATION_DISTANCE_Z                       0x0A,0x7C,0x04
#define HID_USAGE_SENSOR_DATA_ORIENTATION_DISTANCE_OUT_OF_RANGE            0x0A,0x7D,0x04
#define HID_USAGE_SENSOR_DATA_ORIENTATION_TILT                             0x0A,0x7E,0x04
#define HID_USAGE_SENSOR_DATA_ORIENTATION_TILT_X                           0x0A,0x7F,0x04
#define HID_USAGE_SENSOR_DATA_ORIENTATION_TILT_Y                           0x0A,0x80,0x04
#define HID_USAGE_SENSOR_DATA_ORIENTATION_TILT_Z                           0x0A,0x81,0x04
#define HID_USAGE_SENSOR_DATA_ORIENTATION_ROTATION_MATRIX                  0x0A,0x82,0x04
#define HID_USAGE_SENSOR_DATA_ORIENTATION_QUATERNION                       0x0A,0x83,0x04
#define HID_USAGE_SENSOR_DATA_ORIENTATION_MAGNETIC_FLUX                    0x0A,0x84,0x04
#define HID_USAGE_SENSOR_DATA_ORIENTATION_MAGNETIC_FLUX_X_AXIS             0x0A,0x85,0x04
#define HID_USAGE_SENSOR_DATA_ORIENTATION_MAGNETIC_FLUX_Y_AXIS             0x0A,0x86,0x04
#define HID_USAGE_SENSOR_DATA_ORIENTATION_MAGNETIC_FLUX_Z_AXIS             0x0A,0x87,0x04

//data type mechanical
//data field usages (input report)
#define HID_USAGE_SENSOR_DATA_MECHANICAL                                   0x0A,0x90,0x04
#define HID_USAGE_SENSOR_DATA_MECHANICAL_BOOLEAN_SWITCH_STATE              0x0A,0x91,0x04
#define HID_USAGE_SENSOR_DATA_MECHANICAL_BOOLEAN_SWITCH_ARRAY_STATES       0x0A,0x92,0x04
#define HID_USAGE_SENSOR_DATA_MECHANICAL_MULTIVALUE_SWITCH_VALUE           0x0A,0x93,0x04
#define HID_USAGE_SENSOR_DATA_MECHANICAL_FORCE                             0x0A,0x94,0x04
#define HID_USAGE_SENSOR_DATA_MECHANICAL_ABSOLUTE_PRESSURE                 0x0A,0x95,0x04
#define HID_USAGE_SENSOR_DATA_MECHANICAL_GAUGE_PRESSURE                    0x0A,0x96,0x04
#define HID_USAGE_SENSOR_DATA_MECHANICAL_STRAIN                            0x0A,0x97,0x04
#define HID_USAGE_SENSOR_DATA_MECHANICAL_WEIGHT                            0x0A,0x98,0x04
//property usages (get/set feature report)
#define HID_USAGE_SENSOR_PROPERTY_MECHANICAL                               0x0A,0xA0,0x04
#define HID_USAGE_SENSOR_PROPERTY_MECHANICAL_VIBRATION_STATE               0x0A,0xA1,0x04
#define HID_USAGE_SENSOR_DATA_MECHANICAL_VIBRATION_SPEED_FORWARD           0x0A,0xA2,0x04
#define HID_USAGE_SENSOR_DATA_MECHANICAL_VIBRATION_SPEED_BACKWARD          0x0A,0xA3,0x04

//data type biometric
//data field usages (input report)
#define HID_USAGE_SENSOR_DATA_BIOMETRIC                                    0x0A,0xB0,0x04
#define HID_USAGE_SENSOR_DATA_BIOMETRIC_HUMAN_PRESENCE                     0x0A,0xB1,0x04
#define HID_USAGE_SENSOR_DATA_BIOMETRIC_HUMAN_PROXIMITY_RANGE              0x0A,0xB2,0x04
#define HID_USAGE_SENSOR_DATA_BIOMETRIC_HUMAN_PROXIMITY_OUT_OF_RANGE       0x0A,0xB3,0x04
#define HID_USAGE_SENSOR_DATA_BIOMETRIC_HUMAN_TOUCH_STATE                  0x0A,0xB4,0x04

//data type light sensor
//data field usages (input report)
#define HID_USAGE_SENSOR_DATA_LIGHT                                        0x0A,0xD0,0x04
#define HID_USAGE_SENSOR_DATA_LIGHT_ILLUMINANCE                            0x0A,0xD1,0x04
#define HID_USAGE_SENSOR_DATA_LIGHT_COLOR_TEMPERATURE                      0x0A,0xD2,0x04
#define HID_USAGE_SENSOR_DATA_LIGHT_CHROMATICITY                           0x0A,0xD3,0x04
#define HID_USAGE_SENSOR_DATA_LIGHT_CHROMATICITY_X                         0x0A,0xD4,0x04
#define HID_USAGE_SENSOR_DATA_LIGHT_CHROMATICITY_Y                         0x0A,0xD5,0x04
#define HID_USAGE_SENSOR_DATA_LIGHT_CONSUMER_IR_SENTENCE_RECEIVE           0x0A,0xD6,0x04
//property usages (get/set feature report)
#define HID_USAGE_SENSOR_PROPERTY_LIGHT                                    0x0A,0xE0,0x04
#define HID_USAGE_SENSOR_PROPERTY_LIGHT_CONSUMER_IR_SENTENCE_SEND          0x0A,0xE1,0x04

//data type scanner
//data field usages (input report)
#define HID_USAGE_SENSOR_DATA_SCANNER                                      0x0A,0xF0,0x04
#define HID_USAGE_SENSOR_DATA_SCANNER_RFID_TAG                             0x0A,0xF1,0x04
```

```
#define HID_USAGE_SENSOR_DATA_SCANNER_NFC_SENTENCE_RECEIVE                    0x0A,0xF2,0x04
//property usages (get/set feature report)
#define HID_USAGE_SENSOR_PROPERTY_SCANNER                                     0x0A,0xF8,0x04
#define HID_USAGE_SENSOR_PROPERTY_SCANNER_NFC_SENTENCE_SEND                   0x0A,0xF9,0x04

//data type electrical
//data field usages (input report)
#define HID_USAGE_SENSOR_DATA_ELECTRICAL                                      0x0A,0x00,0x05
#define HID_USAGE_SENSOR_DATA_ELECTRICAL_CAPACITANCE                          0x0A,0x01,0x05
#define HID_USAGE_SENSOR_DATA_ELECTRICAL_CURRENT                              0x0A,0x02,0x05
#define HID_USAGE_SENSOR_DATA_ELECTRICAL_POWER                                0x0A,0x03,0x05
#define HID_USAGE_SENSOR_DATA_ELECTRICAL_INDUCTANCE                           0x0A,0x04,0x05
#define HID_USAGE_SENSOR_DATA_ELECTRICAL_RESISTANCE                           0x0A,0x05,0x05
#define HID_USAGE_SENSOR_DATA_ELECTRICAL_VOLTAGE                              0x0A,0x06,0x05
#define HID_USAGE_SENSOR_DATA_ELECTRICAL_FREQUENCY                            0x0A,0x07,0x05
#define HID_USAGE_SENSOR_DATA_ELECTRICAL_PERIOD                               0x0A,0x08,0x05
#define HID_USAGE_SENSOR_DATA_ELECTRICAL_PERCENT_OF_RANGE                     0x0A,0x09,0x05

//data type time
//data field usages (input report)
#define HID_USAGE_SENSOR_DATA_TIME                                            0x0A,0x20,0x05
#define HID_USAGE_SENSOR_DATA_TIME_YEAR                                       0x0A,0x21,0x05
#define HID_USAGE_SENSOR_DATA_TIME_MONTH                                      0x0A,0x22,0x05
#define HID_USAGE_SENSOR_DATA_TIME_DAY                                        0x0A,0x23,0x05
#define HID_USAGE_SENSOR_DATA_TIME_DAY_OF_WEEK                                0x0A,0x24,0x05
#define HID_USAGE_SENSOR_DATA_TIME_HOUR                                       0x0A,0x25,0x05
#define HID_USAGE_SENSOR_DATA_TIME_MINUTE                                     0x0A,0x26,0x05
#define HID_USAGE_SENSOR_DATA_TIME_SECOND                                     0x0A,0x27,0x05
#define HID_USAGE_SENSOR_DATA_TIME_MILLISECOND                                0x0A,0x28,0x05
#define HID_USAGE_SENSOR_DATA_TIME_TIMESTAMP                                  0x0A,0x29,0x05
#define HID_USAGE_SENSOR_DATA_TIME_JULIAN_DAY_OF_YEAR                         0x0A,0x2A,0x05
//property usages (get/set feature report)
#define HID_USAGE_SENSOR_PROPERTY_TIME                                        0x0A,0x30,0x05
#define HID_USAGE_SENSOR_PROPERTY_TIME_TIME_ZONE_OFFSET_FROM_UTC              0x0A,0x31,0x05
#define HID_USAGE_SENSOR_PROPERTY_TIME_TIME_ZONE_NAME                         0x0A,0x32,0x05
#define HID_USAGE_SENSOR_PROPERTY_TIME_DAYLIGHT_SAVINGS_TIME_OBSERVED         0x0A,0x33,0x05
#define HID_USAGE_SENSOR_PROPERTY_TIME_TIME_TRIM_ADJUSTMENT                   0x0A,0x34,0x05
#define HID_USAGE_SENSOR_PROPERTY_TIME_ARM_ALARM                              0x0A,0x35,0x05

//data type custom
//data field usages (input report)
#define HID_USAGE_SENSOR_DATA_CUSTOM                                          0x0A,0x40,0x05
#define HID_USAGE_SENSOR_DATA_CUSTOM_USAGE                                    0x0A,0x41,0x05
#define HID_USAGE_SENSOR_DATA_CUSTOM_BOOLEAN_ARRAY                            0x0A,0x42,0x05
#define HID_USAGE_SENSOR_DATA_CUSTOM_VALUE                                    0x0A,0x43,0x05
#define HID_USAGE_SENSOR_DATA_CUSTOM_VALUE_1                                  0x0A,0x44,0x05
#define HID_USAGE_SENSOR_DATA_CUSTOM_VALUE_2                                  0x0A,0x45,0x05
#define HID_USAGE_SENSOR_DATA_CUSTOM_VALUE_3                                  0x0A,0x46,0x05
#define HID_USAGE_SENSOR_DATA_CUSTOM_VALUE_4                                  0x0A,0x47,0x05
#define HID_USAGE_SENSOR_DATA_CUSTOM_VALUE_5                                  0x0A,0x48,0x05
#define HID_USAGE_SENSOR_DATA_CUSTOM_VALUE_6                                  0x0A,0x49,0x05

//data type generic
//data field usages (input report)
#define HID_USAGE_SENSOR_DATA_GENERIC                                         0x0A,0x60,0x05
#define HID_USAGE_SENSOR_DATA_GENERIC_GUID_OR_PROPERTYKEY                     0x0A,0x61,0x05
#define HID_USAGE_SENSOR_DATA_GENERIC_CATEGORY_GUID                           0x0A,0x62,0x05
#define HID_USAGE_SENSOR_DATA_GENERIC_TYPE_GUID                               0x0A,0x63,0x05
#define HID_USAGE_SENSOR_DATA_GENERIC_EVENT_PROPERTYKEY                       0x0A,0x64,0x05
#define HID_USAGE_SENSOR_DATA_GENERIC_PROPERTY_PROPERTYKEY                    0x0A,0x65,0x05
#define HID_USAGE_SENSOR_DATA_GENERIC_DATAFIELD_PROPERTYKEY                   0x0A,0x66,0x05
#define HID_USAGE_SENSOR_DATA_GENERIC_EVENT                                   0x0A,0x67,0x05
#define HID_USAGE_SENSOR_DATA_GENERIC_PROPERTY                                0x0A,0x68,0x05
#define HID_USAGE_SENSOR_DATA_GENERIC_DATAFIELD                               0x0A,0x69,0x05
#define HID_USAGE_SENSOR_DATA_ENUMERATOR_TABLE_ROW_INDEX                      0x0A,0x6A,0x05
#define HID_USAGE_SENSOR_DATA_ENUMERATOR_TABLE_ROW_COUNT                      0x0A,0x6B,0x05
#define HID_USAGE_SENSOR_DATA_GENERIC_GUID_OR_PROPERTYKEY_KIND                0x0A,0x6C,0x05 // NAry
//begin GorPK kind selectors
#define HID_USAGE_SENSOR_GORPK_KIND_CATEGORY                                  0x0A,0xD0,0x08 // Sel
#define HID_USAGE_SENSOR_GORPK_KIND_TYPE                                      0x0A,0xD1,0x08 // Sel
#define HID_USAGE_SENSOR_GORPK_KIND_EVENT                                     0x0A,0xD2,0x08 // Sel
#define HID_USAGE_SENSOR_GORPK_KIND_PROPERTY                                  0x0A,0xD3,0x08 // Sel
#define HID_USAGE_SENSOR_GORPK_KIND_DATAFIELD                                 0x0A,0xD4,0x08 // Sel
//end GorPK kind selectors
#define HID_USAGE_SENSOR_DATA_GENERIC_GUID                                    0x0A,0x6D,0x05
#define HID_USAGE_SENSOR_DATA_GENERIC_PROPERTYKEY                             0x0A,0x6E,0x05
#define HID_USAGE_SENSOR_DATA_GENERIC_TOP_LEVEL_COLLECTION_ID                 0x0A,0x6F,0x05
#define HID_USAGE_SENSOR_DATA_GENERIC_REPORT_ID                               0x0A,0x70,0x05
#define HID_USAGE_SENSOR_DATA_GENERIC_REPORT_ITEM_POSITION_INDEX              0x0A,0x71,0x05
#define HID_USAGE_SENSOR_DATA_GENERIC_FIRMWARE_VARTYPE                        0x0A,0x72,0x05 // NAry
//begin firmware vartype selectors
#define HID_USAGE_SENSOR_FIRMWARE_VARTYPE_VT_NULL                             0x0A,0x00,0x09 // Sel
#define HID_USAGE_SENSOR_FIRMWARE_VARTYPE_VT_BOOL                             0x0A,0x01,0x09 // Sel
#define HID_USAGE_SENSOR_FIRMWARE_VARTYPE_VT_UI1                              0x0A,0x02,0x09 // Sel
#define HID_USAGE_SENSOR_FIRMWARE_VARTYPE_VT_I1                               0x0A,0x03,0x09 // Sel
#define HID_USAGE_SENSOR_FIRMWARE_VARTYPE_VT_UI2                              0x0A,0x04,0x09 // Sel
#define HID_USAGE_SENSOR_FIRMWARE_VARTYPE_VT_I2                               0x0A,0x05,0x09 // Sel
#define HID_USAGE_SENSOR_FIRMWARE_VARTYPE_VT_UI4                              0x0A,0x06,0x09 // Sel
#define HID_USAGE_SENSOR_FIRMWARE_VARTYPE_VT_I4                               0x0A,0x07,0x09 // Sel
#define HID_USAGE_SENSOR_FIRMWARE_VARTYPE_VT_UI8                              0x0A,0x08,0x09 // Sel
#define HID_USAGE_SENSOR_FIRMWARE_VARTYPE_VT_I8                               0x0A,0x09,0x09 // Sel
#define HID_USAGE_SENSOR_FIRMWARE_VARTYPE_VT_R4                               0x0A,0x0A,0x09 // Sel
#define HID_USAGE_SENSOR_FIRMWARE_VARTYPE_VT_R8                               0x0A,0x0B,0x09 // Sel
#define HID_USAGE_SENSOR_FIRMWARE_VARTYPE_VT_WSTR                             0x0A,0x0C,0x09 // Sel
#define HID_USAGE_SENSOR_FIRMWARE_VARTYPE_VT_STR                              0x0A,0x0D,0x09 // Sel
```

```
#define HID_USAGE_SENSOR_FIRMWARE_VARTYPE_VT_CLSID                    0x0A,0x0E,0x09 // Sel
#define HID_USAGE_SENSOR_FIRMWARE_VARTYPE_VT_VECTOR_VT_UI1            0x0A,0x0F,0x09 // Sel
#define HID_USAGE_SENSOR_FIRMWARE_VARTYPE_VT_F16E0                    0x0A,0x10,0x09 // Sel
#define HID_USAGE_SENSOR_FIRMWARE_VARTYPE_VT_F16E1                    0x0A,0x11,0x09 // Sel
#define HID_USAGE_SENSOR_FIRMWARE_VARTYPE_VT_F16E2                    0x0A,0x12,0x09 // Sel
#define HID_USAGE_SENSOR_FIRMWARE_VARTYPE_VT_F16E3                    0x0A,0x13,0x09 // Sel
#define HID_USAGE_SENSOR_FIRMWARE_VARTYPE_VT_F16E4                    0x0A,0x14,0x09 // Sel
#define HID_USAGE_SENSOR_FIRMWARE_VARTYPE_VT_F16E5                    0x0A,0x15,0x09 // Sel
#define HID_USAGE_SENSOR_FIRMWARE_VARTYPE_VT_F16E6                    0x0A,0x16,0x09 // Sel
#define HID_USAGE_SENSOR_FIRMWARE_VARTYPE_VT_F16E7                    0x0A,0x17,0x09 // Sel
#define HID_USAGE_SENSOR_FIRMWARE_VARTYPE_VT_F16E8                    0x0A,0x18,0x09 // Sel
#define HID_USAGE_SENSOR_FIRMWARE_VARTYPE_VT_F16E9                    0x0A,0x19,0x09 // Sel
#define HID_USAGE_SENSOR_FIRMWARE_VARTYPE_VT_F16EA                    0x0A,0x1A,0x09 // Sel
#define HID_USAGE_SENSOR_FIRMWARE_VARTYPE_VT_F16EB                    0x0A,0x1B,0x09 // Sel
#define HID_USAGE_SENSOR_FIRMWARE_VARTYPE_VT_F16EC                    0x0A,0x1C,0x09 // Sel
#define HID_USAGE_SENSOR_FIRMWARE_VARTYPE_VT_F16ED                    0x0A,0x1D,0x09 // Sel
#define HID_USAGE_SENSOR_FIRMWARE_VARTYPE_VT_F16EE                    0x0A,0x1E,0x09 // Sel
#define HID_USAGE_SENSOR_FIRMWARE_VARTYPE_VT_F16EF                    0x0A,0x1F,0x09 // Sel
#define HID_USAGE_SENSOR_FIRMWARE_VARTYPE_VT_F32E0                    0x0A,0x20,0x09 // Sel
#define HID_USAGE_SENSOR_FIRMWARE_VARTYPE_VT_F32E1                    0x0A,0x21,0x09 // Sel
#define HID_USAGE_SENSOR_FIRMWARE_VARTYPE_VT_F32E2                    0x0A,0x22,0x09 // Sel
#define HID_USAGE_SENSOR_FIRMWARE_VARTYPE_VT_F32E3                    0x0A,0x23,0x09 // Sel
#define HID_USAGE_SENSOR_FIRMWARE_VARTYPE_VT_F32E4                    0x0A,0x24,0x09 // Sel
#define HID_USAGE_SENSOR_FIRMWARE_VARTYPE_VT_F32E5                    0x0A,0x25,0x09 // Sel
#define HID_USAGE_SENSOR_FIRMWARE_VARTYPE_VT_F32E6                    0x0A,0x26,0x09 // Sel
#define HID_USAGE_SENSOR_FIRMWARE_VARTYPE_VT_F32E7                    0x0A,0x27,0x09 // Sel
#define HID_USAGE_SENSOR_FIRMWARE_VARTYPE_VT_F32E8                    0x0A,0x28,0x09 // Sel
#define HID_USAGE_SENSOR_FIRMWARE_VARTYPE_VT_F32E9                    0x0A,0x29,0x09 // Sel
#define HID_USAGE_SENSOR_FIRMWARE_VARTYPE_VT_F32EA                    0x0A,0x2A,0x09 // Sel
#define HID_USAGE_SENSOR_FIRMWARE_VARTYPE_VT_F32EB                    0x0A,0x2B,0x09 // Sel
#define HID_USAGE_SENSOR_FIRMWARE_VARTYPE_VT_F32EC                    0x0A,0x2C,0x09 // Sel
#define HID_USAGE_SENSOR_FIRMWARE_VARTYPE_VT_F32ED                    0x0A,0x2D,0x09 // Sel
#define HID_USAGE_SENSOR_FIRMWARE_VARTYPE_VT_F32EE                    0x0A,0x2E,0x09 // Sel
#define HID_USAGE_SENSOR_FIRMWARE_VARTYPE_VT_F32EF                    0x0A,0x2F,0x09 // Sel
//end firmware vartype selectors
#define HID_USAGE_SENSOR_DATA_GENERIC_UNIT_OF_MEASURE                 0x0A,0x73,0x05 // NAry
//begin unit of measure selectors
#define HID_USAGE_SENSOR_GENERIC_UNIT_NOT_SPECIFIED                   0x0A,0x40,0x09 // Sel
#define HID_USAGE_SENSOR_GENERIC_UNIT_LUX                             0x0A,0x41,0x09 // Sel
#define HID_USAGE_SENSOR_GENERIC_UNIT_DEGREES_KELVIN                  0x0A,0x42,0x09 // Sel
#define HID_USAGE_SENSOR_GENERIC_UNIT_DEGREES_CELSIUS                 0x0A,0x43,0x09 // Sel
#define HID_USAGE_SENSOR_GENERIC_UNIT_PASCAL                          0x0A,0x44,0x09 // Sel
#define HID_USAGE_SENSOR_GENERIC_UNIT_NEWTON                          0x0A,0x45,0x09 // Sel
#define HID_USAGE_SENSOR_GENERIC_UNIT_METERS_PER_SECOND               0x0A,0x46,0x09 // Sel
#define HID_USAGE_SENSOR_GENERIC_UNIT_KILOGRAM                        0x0A,0x47,0x09 // Sel
#define HID_USAGE_SENSOR_GENERIC_UNIT_METER                           0x0A,0x48,0x09 // Sel
#define HID_USAGE_SENSOR_GENERIC_UNIT_METERS_PER_SEC_SQRD             0x0A,0x49,0x09 // Sel
#define HID_USAGE_SENSOR_GENERIC_UNIT_FARAD                           0x0A,0x4A,0x09 // Sel
#define HID_USAGE_SENSOR_GENERIC_UNIT_AMPERE                          0x0A,0x4B,0x09 // Sel
#define HID_USAGE_SENSOR_GENERIC_UNIT_WATT                            0x0A,0x4C,0x09 // Sel
#define HID_USAGE_SENSOR_GENERIC_UNIT_HENRY                           0x0A,0x4D,0x09 // Sel
#define HID_USAGE_SENSOR_GENERIC_UNIT_OHM                             0x0A,0x4E,0x09 // Sel
#define HID_USAGE_SENSOR_GENERIC_UNIT_VOLT                            0x0A,0x4F,0x09 // Sel
#define HID_USAGE_SENSOR_GENERIC_UNIT_HERTZ                           0x0A,0x50,0x09 // Sel
#define HID_USAGE_SENSOR_GENERIC_UNIT_BAR                             0x0A,0x51,0x09 // Sel
#define HID_USAGE_SENSOR_GENERIC_UNIT_DEGREES_ANTI_CLOCKWISE          0x0A,0x52,0x09 // Sel
#define HID_USAGE_SENSOR_GENERIC_UNIT_DEGREES_CLOCKWISE               0x0A,0x53,0x09 // Sel
#define HID_USAGE_SENSOR_GENERIC_UNIT_DEGREES                         0x0A,0x54,0x09 // Sel
#define HID_USAGE_SENSOR_GENERIC_UNIT_DEGREES_PER_SECOND              0x0A,0x55,0x09 // Sel
#define HID_USAGE_SENSOR_GENERIC_UNIT_DEGREES_PER_SEC_SQRD            0x0A,0x56,0x09 // Sel
#define HID_USAGE_SENSOR_GENERIC_UNIT_KNOT                            0x0A,0x57,0x09 // Sel
#define HID_USAGE_SENSOR_GENERIC_UNIT_PERCENT                         0x0A,0x58,0x09 // Sel
#define HID_USAGE_SENSOR_GENERIC_UNIT_SECOND                          0x0A,0x59,0x09 // Sel
#define HID_USAGE_SENSOR_GENERIC_UNIT_MILLISECOND                     0x0A,0x5A,0x09 // Sel
#define HID_USAGE_SENSOR_GENERIC_UNIT_G                               0x0A,0x5B,0x09 // Sel
#define HID_USAGE_SENSOR_GENERIC_UNIT_BYTES                           0x0A,0x5C,0x09 // Sel
#define HID_USAGE_SENSOR_GENERIC_UNIT_MILLIGAUSS                      0x0A,0x5D,0x09 // Sel
#define HID_USAGE_SENSOR_GENERIC_UNIT_BITS                            0x0A,0x5E,0x09 // Sel
//end unit of measure selectors
#define HID_USAGE_SENSOR_DATA_GENERIC_UNIT_EXPONENT                   0x0A,0x74,0x05 // NAry
//begin unit exponent selectors
#define HID_USAGE_SENSOR_GENERIC_EXPONENT_0                           0x0A,0x70,0x09 // Sel
#define HID_USAGE_SENSOR_GENERIC_EXPONENT_1                           0x0A,0x71,0x09 // Sel
#define HID_USAGE_SENSOR_GENERIC_EXPONENT_2                           0x0A,0x72,0x09 // Sel
#define HID_USAGE_SENSOR_GENERIC_EXPONENT_3                           0x0A,0x73,0x09 // Sel
#define HID_USAGE_SENSOR_GENERIC_EXPONENT_4                           0x0A,0x74,0x09 // Sel
#define HID_USAGE_SENSOR_GENERIC_EXPONENT_5                           0x0A,0x75,0x09 // Sel
#define HID_USAGE_SENSOR_GENERIC_EXPONENT_6                           0x0A,0x76,0x09 // Sel
#define HID_USAGE_SENSOR_GENERIC_EXPONENT_7                           0x0A,0x77,0x09 // Sel
#define HID_USAGE_SENSOR_GENERIC_EXPONENT_8                           0x0A,0x78,0x09 // Sel
#define HID_USAGE_SENSOR_GENERIC_EXPONENT_9                           0x0A,0x79,0x09 // Sel
#define HID_USAGE_SENSOR_GENERIC_EXPONENT_A                           0x0A,0x7A,0x09 // Sel
#define HID_USAGE_SENSOR_GENERIC_EXPONENT_B                           0x0A,0x7B,0x09 // Sel
#define HID_USAGE_SENSOR_GENERIC_EXPONENT_C                           0x0A,0x7C,0x09 // Sel
#define HID_USAGE_SENSOR_GENERIC_EXPONENT_D                           0x0A,0x7D,0x09 // Sel
#define HID_USAGE_SENSOR_GENERIC_EXPONENT_E                           0x0A,0x7E,0x09 // Sel
#define HID_USAGE_SENSOR_GENERIC_EXPONENT_F                           0x0A,0x7F,0x09 // Sel
//end unit exponent selectors
#define HID_USAGE_SENSOR_DATA_GENERIC_REPORT_SIZE                     0x0A,0x75,0x05
#define HID_USAGE_SENSOR_DATA_GENERIC_REPORT_COUNT                    0x0A,0x76,0x05
//property usages (get/set feature report)
#define HID_USAGE_SENSOR_PROPERTY_GENERIC                             0x0A,0x80,0x05
#define HID_USAGE_SENSOR_PROPERTY_ENUMERATOR_TABLE_ROW_INDEX          0x0A,0x81,0x05
#define HID_USAGE_SENSOR_PROPERTY_ENUMERATOR_TABLE_ROW_COUNT          0x0A,0x82,0x05
```

```
////////////////////////////////////////////////////////////////////////////////
//
// Other HID definitions
//
////////////////////////////////////////////////////////////////////////////////

//NOTE: These definitions are designed to permit compiling the HID report descriptors
// with somewhat self-explanatory information to help readability and reduce errors

//input,output,feature flags
#define Data_Arr_Abs                                                    0x00
#define Const_Arr_Abs                                                   0x01
#define Data_Var_Abs                                                    0x02
#define Const_Var_Abs                                                   0x03
#define Data_Var_Rel                                                    0x06
//collection flags
#define Physical                                                        0x00
#define Application                                                     0x01
#define Logical                                                         0x02
#define NamedArray                                                      0x04
#define UsageSwitch                                                     0x05
//other
#define Undefined                                                       0x00

#define HID_USAGE_PAGE(a)                                               0x05,a
#define HID_USAGE(a)                                                    0x09,a
#define HID_USAGE16(a,b)                                                0x0A,a,b
#define HID_USAGE_SENSOR_DATA(a,b)                                      a|b     //This or-s the mod into usage
#define HID_COLLECTION(a)                                               0xA1,a
#define HID_REPORT_ID(a)                                                0x85,a
#define HID_REPORT_SIZE(a)                                              0x75,a
#define HID_REPORT_COUNT(a)                                             0x95,a
#define HID_USAGE_MIN_8(a)                                              0x19,a
#define HID_USAGE_MIN_16(a,b)                                           0x1A,a,b
#define HID_USAGE_MAX_8(a)                                              0x29,a
#define HID_USAGE_MAX_16(a,b)                                           0x2A,a,b
#define HID_LOGICAL_MIN_8(a)                                            0x15,a
#define HID_LOGICAL_MIN_16(a,b)                                         0x16,a,b
#define HID_LOGICAL_MIN_32(a,b,c,d)                                     0x17,a,b,c,d
#define HID_LOGICAL_MAX_8(a)                                            0x25,a
#define HID_LOGICAL_MAX_16(a,b)                                         0x26,a,b
#define HID_LOGICAL_MAX_32(a,b,c,d)                                     0x27,a,b,c,d
#define HID_UNIT_EXPONENT(a)                                            0x55,a
#define HID_INPUT(a)                                                    0x81,a
#define HID_OUTPUT(a)                                                   0x91,a
#define HID_FEATURE(a)                                                  0xB1,a
#define HID_END_COLLECTION                                              0xC0

#endif
```

4.2 Special Constructions

4.2.1 Values, Types, and Unit Exponents

The HID Report Descriptors in this section use the following definitions for values, units and unit exponents.

The value communicated as part of a Report Descriptor is in terms of the Report Size and Report Count attributes, combined with the Logical Minimum, Logical Maximum, and Units for data values associated with that Report Item.

The value is treated in one of three ways:

- As a bitfield
- As a signed or unsigned integer value
- As a float value

Bitfield

A value is identified as a bitfield when the Report Size field = 1. In this section, this is expressed as HID_REPORT_SIZE(1). In this case, Logical Maximum, Logical Minimum, Units and Units Exponent are not used.

Unsigned Integer

A value is identified as an unsigned integer when the ReportSize field = 8, 16 or 32 while the Units Exponent value = 0. In this section, this is expressed as HID_REPORT_SIZE(8), HID_REPORT_SIZE(16), or HID_REPORT_SIZE(32) respectively. Logical Minimum and Logical Maximum must both be positive values. Units can be specified or remain unspecified. Units Exponent must be = 0.

Signed Integer

A value is identified as an signed integer when the ReportSize field = 8, 16 or 32 while the Units Exponent value = 0. In this section, this is expressed as HID_REPORT_SIZE(8), HID_REPORT_SIZE(16), or HID_REPORT_SIZE(32) respectively. Logical Minimum must be a negative value and Logical Maximum must be a positive value. Units can be specified or remain unspecified. Units Exponent must be = 0.

Float Value

Essentially, a float is expressed as a combination of a mantissa carried in the value field, and the exponent expressed as power of 10 carried in the Unit Exponent field. A value is identified as a float value when the ReportSize field = 16 or 32 while the Units Exponent value is not 0. In this section, this is expressed as HID_REPORT_SIZE(16) or HID_REPORT_SIZE(32) respectively. Logical Minimum must be a negative value and Logical Maximum must be a positive value. Units can be specified or remain unspecified. Units Exponent must not be = 0. The Unit Exponent field is translated into powers of 10 as specified by the following table.

Value	Exponent	Power of Ten
0x00	1x10E0	1
0x01	1X10E1	10
0x02	1x10E2	100
0x03	1x10E3	1 000
0x04	1x10E4	10 000
0x05	1x10E5	100 000
0x06	1x10E6	1 000 000
0x07	1x10E7	10 000 000
0x08	1x10E-8	0.00 000 001
0x09	1x10E-7	0.0 000 001
0x0A	1x10E-6	0.000 001
0x0B	1x10E-5	0.00 001
0x0C	1x10E-4	0.0 001
0x0D	1x10E-3	0.001
0x0E	1x10E-2	0.01
0x0F	1x10E-1	0.1

Table 18. HID Unit Exponent encoding and meanings

These Unit Exponent field usages are not unique to this specification, but are the same as the standard HID definitions.

4.2.2 Extended Properties

The HID Report Descriptors illustrations in Section 4.3 are meant to be examples and not prescriptive. A large number of sensor Properties (transferred in Feature Reports) can be described from within this specification, but few are actually shown in the examples.[4]

Typically, the examples include the following sensor properties:

```
.
.
.
HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE,  // NAry
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_8(5),
HID_REPORT_SIZE(8),
HID_REPORT_COUNT(1),
HID_COLLECTION(Logical),
          HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_NO_EVENTS,
          HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS,
          HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_THRESHOLD_EVENTS,
          HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_NO_EVENTS_WAKE,
          HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS_WAKE,
          HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_THRESHOLD_EVENTS_WAKE,
          HID_FEATURE(Data_Arr_Abs),
HID_END_COLLECTION,
HID_USAGE_SENSOR_PROPERTY_SENSOR_STATUS,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_32(0xFF,0xFF,0xFF,0xFF),
HID_REPORT_SIZE(32),
HID_REPORT_COUNT(1)
HID_FEATURE(Data_Var_Abs), // up to VT_UI4 worth of status info
HID_USAGE_SENSOR_PROPERTY_REPORT_INTERVAL,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_32(0xFF,0xFF,0xFF,0xFF),
HID_REPORT_SIZE(32),
HID_REPORT_COUNT(1),
// HID_USAGE_SENSOR_UNITS_MILLISECOND,
HID_UNIT_EXPONENT(0),
HID_FEATURE(Data_Var_Abs),
HID_USAGE_SENSOR_PROPERTY_SENSOR_CONNECTION_TYPE,  // NAry
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_8(2),
HID_REPORT_SIZE(8),
HID_REPORT_COUNT(1),
HID_COLLECTION(Logical),
          HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_INTEGRATED,
          HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_ATTACHED,
          HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_EXTERNAL,
          HID_FEATURE(Const_Arr_Abs),
HID_END_COLLECTION,
.
.
.
```

The examples also typically include the following per-datafield properties, like these taken from the **Barometer** report descriptor:

```
.
.
.
HID_USAGE_SENSOR_PROPERTY_CHANGE_SENSITIVITY_ABS,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_16(0xFF,0xFF),
HID_REPORT_SIZE(16),
HID_REPORT_COUNT(1),
// HID_USAGE_SENSOR_UNITS_BAR,
HID_UNIT_EXPONENT(0x0E), // scale default unit “bar” to provide 2 digits past the decimal point
HID_FEATURE(Data_Var_Abs),
HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_ENVIRONMENTAL_ATMOSPHERIC_PRESSURE,HID_USAGE_SENSOR_DATA_MOD_MAX),
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_16(0xFF,0xFF),
HID_REPORT_SIZE(16),
HID_REPORT_COUNT(1),
// HID_USAGE_SENSOR_UNITS_BAR,
HID_UNIT_EXPONENT(0x0E), // scale default unit “bar” to provide 2 digits past the decimal point
HID_FEATURE(Data_Var_Abs),
```

[4] In this and other HID Report Descriptor examples, key Usages are shown in blue color merely for ease of reading. The color does not have any other special significance.

```
HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_ENVIRONMENTAL_ATMOSPHERIC_PRESSURE,HID_USAGE_SENSOR_DATA_MOD_MIN),
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_16(0xFF,0xFF),
HID_REPORT_SIZE(16),
HID_REPORT_COUNT(1),
// HID_USAGE_SENSOR_UNITS_BAR,
HID_UNIT_EXPONENT(0x0E), // scale default unit “bar” to provide 2 digits past the decimal point
HID_FEATURE(Data_Var_Abs),
.
.
.
```

A further discussion of per-datafield properties is left to Section 4.2.3; this section will focus on the ‘extended’ Properties not used as explicit examples in Section 4.3.

For reference, the complete set of sensor Properties is repeated below for convenience:

```
#define HID_USAGE_SENSOR_PROPERTY_FRIENDLY_NAME                          0x0A,0x01,0x03
#define HID_USAGE_SENSOR_PROPERTY_PERSISTENT_UNIQUE_ID                   0x0A,0x02,0x03
#define HID_USAGE_SENSOR_PROPERTY_SENSOR_STATUS                          0x0A,0x03,0x03
#define HID_USAGE_SENSOR_PROPERTY_MINIMUM_REPORT_INTERVAL                0x0A,0x04,0x03
#define HID_USAGE_SENSOR_PROPERTY_SENSOR_MANUFACTURER                    0x0A,0x05,0x03
#define HID_USAGE_SENSOR_PROPERTY_SENSOR_MODEL                           0x0A,0x06,0x03
#define HID_USAGE_SENSOR_PROPERTY_SENSOR_SERIAL_NUMBER                   0x0A,0x07,0x03
#define HID_USAGE_SENSOR_PROPERTY_SENSOR_DESCRIPTION                     0x0A,0x08,0x03
#define HID_USAGE_SENSOR_PROPERTY_SENSOR_CONNECTION_TYPE                 0x0A,0x09,0x03 // NAry
#define HID_USAGE_SENSOR_PROPERTY_SENSOR_DEVICE_PATH                     0x0A,0x0A,0x03
#define HID_USAGE_SENSOR_PROPERTY_HARDWARE_REVISION                      0x0A,0x0B,0x03
#define HID_USAGE_SENSOR_PROPERTY_FIRMWARE_VERSION                       0x0A,0x0C,0x03
#define HID_USAGE_SENSOR_PROPERTY_RELEASE_DATE                           0x0A,0x0D,0x03
#define HID_USAGE_SENSOR_PROPERTY_REPORT_INTERVAL                        0x0A,0x0E,0x03
#define HID_USAGE_SENSOR_PROPERTY_CHANGE_SENSITIVITY_ABS                 0x0A,0x0F,0x03
#define HID_USAGE_SENSOR_PROPERTY_CHANGE_SENSITIVITY_RANGE_PCT           0x0A,0x10,0x03
#define HID_USAGE_SENSOR_PROPERTY_CHANGE_SENSITIVITY_REL_PCT             0x0A,0x11,0x03
#define HID_USAGE_SENSOR_PROPERTY_ACCURACY                               0x0A,0x12,0x03
#define HID_USAGE_SENSOR_PROPERTY_RESOLUTION                             0x0A,0x13,0x03
#define HID_USAGE_SENSOR_PROPERTY_RANGE_MAXIMUM                          0x0A,0x14,0x03
#define HID_USAGE_SENSOR_PROPERTY_RANGE_MINIMUM                          0x0A,0x15,0x03
#define HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE                        0x0A,0x16,0x03 // NAry
#define HID_USAGE_SENSOR_PROPERTY_SAMPLING_RATE                          0x0A,0x17,0x03
#define HID_USAGE_SENSOR_PROPERTY_RESPONSE_CURVE                         0x0A,0x18,0x03
#define HID_USAGE_SENSOR_PROPERTY_POWER_STATE                            0x0A,0x19,0x03 // NAry
```

The following extract from a hypothetical HID Report Descriptor shows how to represent each of these. Note that in the case of string descriptors such as FRIENDLY_NAME and PERSISTENT_UNIQUE_ID the report count should be large enough to contain the expected value (16-bits for each wide character, plus 16-bits for a wide NULL termination) but need be not larger (16 is used here for reference, long enough to hold a 15 wide-character string):

```
.
.
.
HID_USAGE_SENSOR_PROPERTY_FRIENDLY_NAME,
HID_REPORT_SIZE(16),
HID_REPORT_COUNT(16),
HID_FEATURE(Const_Arr_Abs),

HID_USAGE_SENSOR_PROPERTY_PERSISTENT_UNIQUE_ID,
HID_REPORT_SIZE(16),
HID_REPORT_COUNT(16),
HID_FEATURE(Data_Arr_Abs),

HID_USAGE_SENSOR_PROPERTY_SENSOR_STATUS,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_32(0xFF,0xFF,0xFF,0xFF),
HID_REPORT_SIZE(32),
HID_REPORT_COUNT(1),
HID_FEATURE(Const_Var_Abs), // up to VT_UI4 worth of status info

HID_USAGE_SENSOR_PROPERTY_MINIMUM_REPORT_INTERVAL,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_32(0xFF,0xFF,0xFF,0xFF),
HID_REPORT_SIZE(32),
HID_REPORT_COUNT(1),
// HID_USAGE_SENSOR_UNITS_MILLISECOND,
HID_UNIT_EXPONENT(0),
HID_FEATURE(Const_Var_Abs),

HID_USAGE_SENSOR_PROPERTY_SENSOR_MANUFACTURER,
HID_REPORT_SIZE(16),
HID_REPORT_COUNT(16),
HID_FEATURE(Const_Arr_Abs),

HID_USAGE_SENSOR_PROPERTY_SENSOR_MODEL,
```

```
HID_REPORT_SIZE(16),
HID_REPORT_COUNT(16),
HID_FEATURE(Data_Var_Abs),

HID_USAGE_SENSOR_PROPERTY_SENSOR_SERIAL_NUMBER,
HID_REPORT_SIZE(16),
HID_REPORT_COUNT(16),
HID_FEATURE(Const_Arr_Abs),

HID_USAGE_SENSOR_PROPERTY_SENSOR_DESCRIPTION,
HID_REPORT_SIZE(16),
HID_REPORT_COUNT(16),
HID_FEATURE(Const_Arr_Abs),

HID_USAGE_SENSOR_PROPERTY_SENSOR_DEVICE_PATH,
HID_REPORT_SIZE(16),
HID_REPORT_COUNT(16),
HID_FEATURE(Const_Arr_Abs),

HID_USAGE_SENSOR_PROPERTY_HARDWARE_REVISION,
HID_REPORT_SIZE(16),
HID_REPORT_COUNT(16),
HID_FEATURE(Const_Arr_Abs),

HID_USAGE_SENSOR_PROPERTY_FIRMWARE_VERSION,
HID_REPORT_SIZE(16),
HID_REPORT_COUNT(16),
HID_FEATURE(Const_Arr_Abs),

HID_USAGE_SENSOR_PROPERTY_RELEASE_DATE,
HID_REPORT_SIZE(16),
HID_REPORT_COUNT(16),
HID_FEATURE(Const_Arr_Abs),

HID_USAGE_SENSOR_PROPERTY_REPORT_INTERVAL,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_32(0xFF,0xFF,0xFF,0xFF),
HID_REPORT_SIZE(32),
HID_REPORT_COUNT(1),
// HID_USAGE_SENSOR_UNITS_MILLISECOND,
HID_UNIT_EXPONENT(0),
HID_FEATURE(Data_Var_Abs),

HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE, // NAry
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_8(5),
HID_REPORT_SIZE(8),
HID_REPORT_COUNT(1),
HID_COLLECTION(Logical),
          HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_NO_EVENTS,
          HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS,
          HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_THRESHOLD_EVENTS,
          HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_NO_EVENTS_WAKE,
          HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS_WAKE,
          HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_THRESHOLD_EVENTS_WAKE,
          HID_FEATURE(Data_Arr_Abs),
HID_END_COLLECTION,

HID_USAGE_SENSOR_PROPERTY_SAMPLING_RATE,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_32(0xFF,0xFF,0xFF,0xFF),
HID_REPORT_SIZE(32),
HID_REPORT_COUNT(1),
// HID_USAGE_SENSOR_UNITS_MILLISECOND,
HID_UNIT_EXPONENT(0),
HID_FEATURE(Data_Var_Abs),

HID_USAGE_SENSOR_PROPERTY_RESPONSE_CURVE,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_16(255),
HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
HID_REPORT_SIZE(16),
HID_REPORT_COUNT(10), //as required for n pair of values
HID_USAGE_SENSOR_UNITS_NOT_SPECIFIED, //define as required to match application
HID_UNIT_EXPONENT(0x0E), // scale default unit to provide 2 digits past the decimal point
HID_FEATURE(Data_Arr_Abs),

HID_USAGE_SENSOR_PROPERTY_POWER_STATE, // NAry
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_8(5),
HID_REPORT_SIZE(8),
HID_REPORT_COUNT(1),
HID_COLLECTION(Logical),
          HID_USAGE_SENSOR_PROPERTY_POWER_STATE_UNDEFINED,
          HID_USAGE_SENSOR_PROPERTY_POWER_STATE_D0_FULL_POWER,
          HID_USAGE_SENSOR_PROPERTY_POWER_STATE_D1_LOW_POWER,
          HID_USAGE_SENSOR_PROPERTY_POWER_STATE_D2_STANDBY_WITH_WAKE,
          HID_USAGE_SENSOR_PROPERTY_POWER_STATE_D3_SLEEP_WITH_WAKE,
          HID_USAGE_SENSOR_PROPERTY_POWER_STATE_D4_POWER_OFF,
          HID_FEATURE(Data_Arr_Abs),
HID_END_COLLECTION,
.
.
.
```

It is up to the device to expose these as required by the particular application in which the device is used.

Note that many of these are strings, and their over-use can result in the need for very large bufffers to handle the information represented by these report descriptors; these very large buffers may be a problem for devices with very small amounts of internal memory.

4.2.3 Modifiers: Per-datafield Properties

A number of *Properties* (transferred in Feature Reports) that can be applied to *Data Fields* (transferred in Input Reports) are on a per-datafield basis. This presents some options in how these per-datafield *Properties* can be expressed using the definitions in this document.

One way to do so assumes there is only a single type of Data Field, and that the Property applies to all Data Fields of that type.

```
.
.
.
HID_USAGE_SENSOR_PROPERTY_CHANGE_SENSITIVITY_ABS,
HID_USAGE_SENSOR_PROPERTY_MAXIMUM,
HID_USAGE_SENSOR_PROPERTY_MINIMUM,
HID_USAGE_SENSOR_PROPERTY_ACCURACY,
HID_USAGE_SENSOR_PROPERTY_RESOLUTION,
.
.
.
```

Even though the Data Field is not stated, it is assumed that there is only one type supported and that the Property specified applies in the same way to all examples of that type. For example, if this were a single Data Field for a thermometer:

```
HID_USAGE_SENSOR_DATA_EVNIRONMENTAL_TEMPERATURE,
```

then the Properties specified would apply only to that Data Field. If instead this were a tuple of Data Fields for an accelerometer:

```
HID_USAGE_SENSOR_DATA_MOTION_ACCELERATION_X,
HID_USAGE_SENSOR_DATA_MOTION_ACCELERATION_Y,
HID_USAGE_SENSOR_DATA_MOTION_ACCELERATION_Z,
```

Then the Properties specified would apply to all the Data Fields in the same way.

A more explicit construction has been provided that is semantically equivalent. Data Fields are expressed by means of a Data Field type and a Modifier Usage Switch.

The per-datafield properties expressed as Data Field Modifiers are defined elsewhere in the document. Those defined in this document are repeated below for convenience:

```
//data type usages modifiers
//NOTE: the usage tables actually define these as two bytes, but in order
//to get the define macros to work so these are 'or-ed' these are defined
//here as only one byte.
#define HID_USAGE_SENSOR_DATA_MOD_NONE                                  0x00 //US
#define HID_USAGE_SENSOR_DATA_MOD_CHANGE_SENSITIVITY_ABS                0x10 //US
#define HID_USAGE_SENSOR_DATA_MOD_MAX                                   0x20 //US
#define HID_USAGE_SENSOR_DATA_MOD_MIN                                   0x30 //US
#define HID_USAGE_SENSOR_DATA_MOD_ACCURACY                              0x40 //US
#define HID_USAGE_SENSOR_DATA_MOD_RESOLUTION                            0x50 //US
#define HID_USAGE_SENSOR_DATA_MOD_THRESHOLD_HIGH                        0x60 //US
#define HID_USAGE_SENSOR_DATA_MOD_THRESHOLD_LOW                         0x70 //US
#define HID_USAGE_SENSOR_DATA_MOD_CALIBRATION_OFFSET                    0x80 //US
#define HID_USAGE_SENSOR_DATA_MOD_CALIBRATION_MULTIPLIER                0x90 //US
#define HID_USAGE_SENSOR_DATA_MOD_REPORT_INTERVAL                       0xA0 //US
#define HID_USAGE_SENSOR_DATA_MOD_FREQUENCY_MAX                         0xB0 //US
#define HID_USAGE_SENSOR_DATA_MOD_PERIOD_MAX                            0xC0 //US
#define HID_USAGE_SENSOR_DATA_MOD_CHANGE_SENSITIVITY_RANGE_PCT          0xD0 //US
#define HID_USAGE_SENSOR_DATA_MOD_CHANGE_SENSITIVITY_REL_PCT            0xE0 //US
```

Any of these Modifiers can be applied to any Data Field. Below is an example extracted from a HID Report Descriptor that again uses the single Data Field thermometer example:

```
.
.
.
HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_ENVIRONMENTAL_TEMPERATURE,
          HID_USAGE_SENSOR_DATA_MOD_CHANGE_SENSITIVITY_ABS),
HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_ENVIRONMENTAL_TEMPERATURE,HID_USAGE_SENSOR_DATA_MOD_MAX),
HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_ENVIRONMENTAL_TEMPERATURE,HID_USAGE_SENSOR_DATA_MOD_MIN),
HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_ENVIRONMENTAL_TEMPERATURE,HID_USAGE_SENSOR_DATA_MOD_ACCURACY),
HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_ENVIRONMENTAL_TEMPERATURE,HID_USAGE_SENSOR_DATA_MOD_PRECISION),
.
.
.
```

Provision has been made for this sytax to apply to those cases where there are multiple Data Fields defined for the sensor. In each case where multiple Data Fields are defines, a definition has been created that refers to all of them collectively. Using again the accelerometer as an example, the collective and individual Data Field defintions are below:

```
#define HID_USAGE_SENSOR_DATA_MOTION_ACCELERATION                    0x0A,0x52,0x04
#define HID_USAGE_SENSOR_DATA_MOTION_ACCELERATION_X_AXIS             0x0A,0x53,0x04
#define HID_USAGE_SENSOR_DATA_MOTION_ACCELERATION_Y_AXIS             0x0A,0x54,0x04
#define HID_USAGE_SENSOR_DATA_MOTION_ACCELERATION_Z_AXIS             0x0A,0x55,0x04
```

Applying the list of Properties to the collective version of the Data Field would be done as follows:

```
.
.
.
HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_MOTION_ACCELERATION,HID_USAGE_SENSOR_DATA_MOD_CHANGE_SENSITIVITY_ABS),
HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_MOTION_ACCELERATION,HID_USAGE_SENSOR_DATA_MOD_MAX),
HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_MOTION_ACCELERATION,HID_USAGE_SENSOR_DATA_MOD_MIN),
HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_MOTION_ACCELERATION,HID_USAGE_SENSOR_DATA_MOD_ACCURACY),
HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_MOTION_ACCELERATION,HID_USAGE_SENSOR_DATA_MOD_PRECISION),
.
.
.
```

Note that in each case the Data Field to which the Modifier applies is specified, and that in each case the Data Field specified is for the collective version of the Data Field. This is mostly equivalent to the following definitions presented previously and repeated here for convenience:

```
.
.
.
HID_USAGE_SENSOR_PROPERTY_CHANGE_SENSITIVITY_ABS,
HID_USAGE_SENSOR_PROPERTY_MAXIMUM,
HID_USAGE_SENSOR_PROPERTY_MINIMUM,
HID_USAGE_SENSOR_PROPERTY_ACCURACY,
HID_USAGE_SENSOR_PROPERTY_RESOLUTION,
.
.
.
```

In the case of the collective Data Field specification, this will only apply to Data Fields of that type. In the case of the HID_USAGE_SENSOR_PROPERTY_xxx construction, this would apply to all Data Fields even if they are not of the same type.

In the case of specifying Properties that are applied per-datafield with the expectation that the Property may change depending on the Data Field, the following homogenous constructing again using the accelerometer follows:

```
.
.
.
HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_MOTION_ACCELERATION_X_AXIS,
          HID_USAGE_SENSOR_DATA_MOD_CHANGE_SENSITIVITY_ABS),
HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_MOTION_ACCELERATION_Y_AXIS,
           HID_USAGE_SENSOR_DATA_MOD_CHANGE_SENSITIVITY_ABS),
HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_MOTION_ACCELERATION_Z_AXIS,
          HID_USAGE_SENSOR_DATA_MOD_CHANGE_SENSITIVITY_ABS),
```

```
HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_MOTION_ACCELERATION_X_AXIS,HID_USAGE_SENSOR_DATA_MOD_MAX),
HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_MOTION_ACCELERATION_Y_AXIS,HID_USAGE_SENSOR_DATA_MOD_MAX),
HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_MOTION_ACCELERATION_Z_AXIS,HID_USAGE_SENSOR_DATA_MOD_MAX),
HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_MOTION_ACCELERATION_X_AXIS,HID_USAGE_SENSOR_DATA_MOD_MIN),
HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_MOTION_ACCELERATION_Y_AXIS,HID_USAGE_SENSOR_DATA_MOD_MIN),
HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_MOTION_ACCELERATION_Z_AXIS,HID_USAGE_SENSOR_DATA_MOD_MIN),
HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_MOTION_ACCELERATION_X_AXIS,HID_USAGE_SENSOR_DATA_MOD_ACCURACY),
HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_MOTION_ACCELERATION_Y_AXIS,HID_USAGE_SENSOR_DATA_MOD_ACCURACY),
HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_MOTION_ACCELERATION_Z_AXIS,HID_USAGE_SENSOR_DATA_MOD_ACCURACY),
HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_MOTION_ACCELERATION_X_AXIS,HID_USAGE_SENSOR_DATA_MOD_RESOLUTION),
HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_MOTION_ACCELERATION_Y_AXIS,HID_USAGE_SENSOR_DATA_MOD_RESOLUTION),
HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_MOTION_ACCELERATION_Z_AXIS,HID_USAGE_SENSOR_DATA_MOD_RESOLUTION),
.
.
.
```

Note that in each case the Data Field is the specific Data Field and not the collective version, and that the Property modifier only applies to that Data Field. This specificity is the most desireable way to express per-datafield properties, though this comes at some cost to the device that must support these Report Descriptors.

A heterogenous example, from a hypothetical 1D accelerometer combined with a thermometer follows:

```
.
.
.
HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_ENVIRONMENTAL_TEMPERATURE,
          HID_USAGE_SENSOR_DATA_MOD_CHANGE_SENSITIVITY_ABS),
HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_MOTION_ACCELERATION_X_AXIS,
          HID_USAGE_SENSOR_DATA_MOD_CHANGE_SENSITIVITY_ABS),
HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_ENVIRONMENTAL_TEMPERATURE,HID_USAGE_SENSOR_DATA_MOD_MAX),
HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_MOTION_ACCELERATION_X_AXIS,HID_USAGE_SENSOR_DATA_MOD_MAX),
HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_ENVIRONMENTAL_TEMPERATURE,HID_USAGE_SENSOR_DATA_MOD_MIN),
HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_MOTION_ACCELERATION_X_AXIS,HID_USAGE_SENSOR_DATA_MOD_MIN),
HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_ENVIRONMENTAL_TEMPERATURE,HID_USAGE_SENSOR_DATA_MOD_ACCURACY),
HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_MOTION_ACCELERATION_X_AXIS,HID_USAGE_SENSOR_DATA_MOD_ACCURACY),
HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_ENVIRONMENTAL_TEMPERATURE,HID_USAGE_SENSOR_DATA_MOD_PRECISION),
HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_MOTION_ACCELERATION_X_AXIS,HID_USAGE_SENSOR_DATA_MOD_RESOLUTION),
.
.
.
```

There is no requirement that per-datafield Properties be supported at all for any Data Field; it follows, too, that these can be mixed and matched to express Properties that are important to the device implementer.

4.2.4 Event Thresholds

Modifiers (Section 4.2.3) can also be used to define Properties that express thresholds for Data Fields:

```
#define HID_USAGE_SENSOR_DATA_MOD_CHANGE_SENSITIVITY_ABS          0x10 //US
#define HID_USAGE_SENSOR_DATA_MOD_MAX                             0x20 //US
#define HID_USAGE_SENSOR_DATA_MOD_MIN                             0x30 //US
#define HID_USAGE_SENSOR_DATA_MOD_THRESHOLD_HIGH                  0x60 //US
#define HID_USAGE_SENSOR_DATA_MOD_THRESHOLD_LOW                   0x70 //US
#define HID_USAGE_SENSOR_DATA_MOD_FREQUENCY_MAX                   0xB0 //US
#define HID_USAGE_SENSOR_DATA_MOD_PERIOD_MAX                      0xC0 //US
#define HID_USAGE_SENSOR_DATA_MOD_CHANGE_SENSITIVITY_RANGE_PCT    0xD0 //US
#define HID_USAGE_SENSOR_DATA_MOD_CHANGE_SENSITIVITY_REL_PCT      0xE0 //US
```

When the “current reading” of a data value exceeds the bounds of a defined threshold, a “candidate event” is in effect.

Example:

```
#define HID_USAGE_SENSOR_DATA_ENVIRONMENTAL_TEMPERATURE           0x0A,0x34,0x04
```

Temperature is a Data Field. Bounds for that Data Field can be expressed as Properties by “OR-ing” the Modifier Usage with the Data Field usage.

Data Field	Data Field	Modifer	Modifier Usage	“OR-ed” Usage	Semantics

	Usage				
TEMPERATURE	0434	NONE	0	0434	TEMPERATURE
TEMPERATURE	0434	CHANGE SENSITIVITY ABS	1	1434	SENSITIVITY_ABS (TEMPERATURE)
TEMPERATURE	0434	MAX	2	2434	MAX(TEMPERATURE)
TEMPERATURE	0434	MIN	3	3434	MIN(TEMPERATURE)
TEMPERATURE	0434	THRESHOLD HIGH	6	6434	THRESHOLD_HIGH(TEMPERATURE)
TEMPERATURE	0434	THRESHOLD LOW	7	7434	THRESHOLD_LOW(TEMPERATURE)
TEMPERATURE	0434	CHANGE SENSITIVITY RANGE PERCENT	D	D434	SENSITIVITY_RANGE_PCT (TEMPERATURE)
TEMPERATURE	0434	CHANGE SENSITIVITY RELATIVE PERCENT	E	E434	SENSITIVITY_REL_PCT (TEMPERATURE)

Table 19. Modifier Usage example

If MAX(TEMPERATURE) was defined to be +35.0, and the current temperature rose from a value below that until the point that it crossed it in an upward direction, then a “Max Temperature Exceeded” event is in effect.

Whether or not the event is actually reported depends upon the setting of the Reporting State property:

```
//begin reporting state selectors
#define HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_NO_EVENTS                    0x0A,0x40,0x08 // Sel
#define HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS                   0x0A,0x41,0x08 // Sel
#define HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_THRESHOLD_EVENTS             0x0A,0x42,0x08 // Sel
#define HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_NO_EVENTS_WAKE               0x0A,0x43,0x08 // Sel
#define HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS_WAKE              0x0A,0x44,0x08 // Sel
#define HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_THRESHOLD_EVENTS_WAKE        0x0A,0x45,0x08 // Sel
//end reporting state selectors
```

- If the Reporting State selector is NO_EVENTS, then the event is not reported, it is discarded.
- If the Reporting State selector is ALL_EVENTS, then then the event is always reported.
- If the Reporting State selector is THRESHOLD_EVENTS, then the event is reported when it is a threshold exceeded event.
- If the Reporting State selector is one ending in _WAKE, then the sensor is asked to wake the Host CPU if it is asleep, and then send the event.

The actual event ID that is reported depends upon the threshold event type that occurred:

```
#define HID_USAGE_SENSOR_EVENT_UNKNOWN                      0x00
#define HID_USAGE_SENSOR_EVENT_STATE_CHANGED                0x01
#define HID_USAGE_SENSOR_EVENT_PROPERTY_CHANGED             0x02
#define HID_USAGE_SENSOR_EVENT_DATA_UPDATED                 0x03
#define HID_USAGE_SENSOR_EVENT_POLL_RESPONSE                0x04
#define HID_USAGE_SENSOR_EVENT_CHANGE_SENSITIVITY           0x05
#define HID_USAGE_SENSOR_EVENT_MAX_REACHED                  0x06
#define HID_USAGE_SENSOR_EVENT_MIN_REACHED                  0x07
#define HID_USAGE_SENSOR_EVENT_HIGH_THRESHOLD_CROSS_ABOVE   0x08
#define HID_USAGE_SENSOR_EVENT_HIGH_THRESHOLD_CROSS_BELOW   0x09
#define HID_USAGE_SENSOR_EVENT_LOW_THRESHOLD_CROSS_ABOVE    0x0A
#define HID_USAGE_SENSOR_EVENT_LOW_THRESHOLD_CROSS_BELOW    0x0B
#define HID_USAGE_SENSOR_EVENT_ZERO_THRESHOLD_CROSS_ABOVE   0x0C
#define HID_USAGE_SENSOR_EVENT_ZERO_THRESHOLD_CROSS_BELOW   0x0D
#define HID_USAGE_SENSOR_EVENT_PERIOD_EXCEEDED              0x0E
#define HID_USAGE_SENSOR_EVENT_FREQUENCY_EXCEEDED           0x0F
#define HID_USAGE_SENSOR_EVENT_COMPLEX_TRIGGER              0x10
```

Examples:

- The MAX_REACHED event would be reported if the TEMPERATURE rose to the MAX(TEMPERATURE) threshold or higher.
- The MIN_REACHED event would be reported if the TEMPERATURE dropped to the MIN(TEMPERATURE) or lower.
- The HIGH_THRESHOLD_CROSS_ABOVE event would be reported if the TEMPERATURE rose from below the “upper threshold” [which could be defined by THRESHOLD_HIGH(TEMPERATURE), SENSITIVITY_RANGE_PCT(TEMPERATURE), or SENSITIVITY_REL_PCT(TEMPERATURE)] and crossed it in an upward direction.

4.2.5 Sensor Collections

This simple example illustrates a hypothetical device containing two sensors: an accelerometer and an ambient light sensor.

Each sensor is described by its own HID collection. In this example, each sensor has a single Input Report and a single Feature Report. Report ID 0x01 is used for the accelerometer sensor, and Report ID 0x02 is used for the ambient light sensor.

Using this technique, a single sensor device board can support multiple different sensors, and present these as individually identifiable sensors to the Operating System. The platform driver can use this to instantiate a separate “logical sensor object” for each described HID collection.

The first example shows a single Top Level Collection with the individual sensors represented as nested sub-collections:

```
// For reference: Complete HID report descriptor

// Two sensor example: 3D Accelerometer & AmbientLight
const unsigned char nest_col_report_descriptor[] = {
          HID_USAGE_PAGE_SENSOR,
          HID_USAGE_SENSOR_TYPE_COLLECTION,
          HID_COLLECTION(Application),  // Top Level Collection for holding 2 sensors as nested collections

          HID_REPORT_ID(1),
          HID_USAGE_PAGE_SENSOR,
          HID_USAGE_SENSOR_TYPE_MOTION_ACCELEROMETER_3D,
          HID_COLLECTION(Physical),     // first nested sub-collection, for accelerometer

          //feature reports (xmit/receive)
          HID_USAGE_PAGE_SENSOR,
          HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE,
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_8(5),
          HID_REPORT_SIZE(8),
          HID_REPORT_COUNT(1),
          HID_COLLECTION(Logical),
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_NO_EVENTS,
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS,
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_THRESHOLD_EVENTS,
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_NO_EVENTS_WAKE,
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS_WAKE,
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_THRESHOLD_EVENTS_WAKE,
                    HID_FEATURE(Data_Arr_Abs),
          HID_END_COLLECTION,
          HID_USAGE_SENSOR_PROPERTY_SENSOR_STATUS,
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_32(0xFF,0xFF,0xFF,0xFF),
          HID_REPORT_SIZE(32),
          HID_REPORT_COUNT(1),
          HID_FEATURE(Data_Var_Abs), // up to VT_UI4 worth of status info
          HID_USAGE_SENSOR_PROPERTY_REPORT_INTERVAL,
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_32(0xFF,0xFF,0xFF,0xFF),
          HID_REPORT_SIZE(32),
          HID_REPORT_COUNT(1),
          // HID_USAGE_SENSOR_UNITS_MILLISECOND,
          HID_UNIT_EXPONENT(0),
          HID_FEATURE(Data_Var_Abs),
          HID_USAGE_SENSOR_PROPERTY_SENSOR_CONNECTION_TYPE,  // NAry
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_8(2),
          HID_REPORT_SIZE(8),
          HID_REPORT_COUNT(1),
          HID_COLLECTION(Logical),
                    HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_INTEGRATED,
                    HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_ATTACHED,
                    HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_EXTERNAL,
                    HID_FEATURE(Const_Arr_Abs),
          HID_END_COLLECTION,
          HID_USAGE_SENSOR_PROPERTY_CHANGE_SENSITIVITY_ABS,
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_16(0xFF,0xFF),
          HID_REPORT_SIZE(16),
          HID_REPORT_COUNT(1),
          // HID_USAGE_SENSOR_UNITS_G,
          HID_UNIT_EXPONENT(0x0E), // scale default unit “G” to provide 2 digits past the decimal point
          HID_FEATURE(Data_Var_Abs),
          HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_MOTION_ACCELERATION,HID_USAGE_SENSOR_DATA_MOD_MAX),
          HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
          HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
          HID_REPORT_SIZE(16),
```

```
HID_REPORT_COUNT(1),
// HID_USAGE_SENSOR_UNITS_G,
HID_UNIT_EXPONENT(0x0E), // scale default unit “G” to provide 2 digits past the decimal point
HID_FEATURE(Data_Var_Abs),
HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_MOTION_ACCELERATION,HID_USAGE_SENSOR_DATA_MOD_MIN),
HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
HID_REPORT_SIZE(16),
HID_REPORT_COUNT(1),
// HID_USAGE_SENSOR_UNITS_G,
HID_UNIT_EXPONENT(0x0E), // scale default unit “G” to provide 2 digits past the decimal point
HID_FEATURE(Data_Var_Abs),

//input reports (transmit)
HID_USAGE_PAGE_SENSOR,
HID_USAGE_SENSOR_STATE,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_8(6),
HID_REPORT_SIZE(8),
HID_REPORT_COUNT(1),
HID_COLLECTION(Logical),
          HID_USAGE_SENSOR_STATE_UNKNOWN,
          HID_USAGE_SENSOR_STATE_READY,
          HID_USAGE_SENSOR_STATE_NOT_AVAILABLE,
          HID_USAGE_SENSOR_STATE_NO_DATA,
          HID_USAGE_SENSOR_STATE_INITIALIZING,
          HID_USAGE_SENSOR_STATE_ACCESS_DENIED,
          HID_USAGE_SENSOR_STATE_ERROR,
          HID_INPUT(Const_Arr_Abs),
HID_END_COLLECTION,
HID_USAGE_SENSOR_EVENT,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_8(16),
HID_REPORT_SIZE(8),
HID_REPORT_COUNT(1),
HID_COLLECTION(Logical),
          HID_USAGE_SENSOR_EVENT_UNKNOWN,
          HID_USAGE_SENSOR_EVENT_STATE_CHANGED,
          HID_USAGE_SENSOR_EVENT_PROPERTY_CHANGED,
          HID_USAGE_SENSOR_EVENT_DATA_UPDATED,
          HID_USAGE_SENSOR_EVENT_POLL_RESPONSE,
          HID_USAGE_SENSOR_EVENT_CHANGE_SENSITIVITY,
          HID_USAGE_SENSOR_EVENT_MAX_REACHED,
          HID_USAGE_SENSOR_EVENT_MIN_REACHED,
          HID_USAGE_SENSOR_EVENT_HIGH_THRESHOLD_CROSS_UPWARD,
          HID_USAGE_SENSOR_EVENT_HIGH_THRESHOLD_CROSS_DOWNWARD,
          HID_USAGE_SENSOR_EVENT_LOW_THRESHOLD_CROSS_UPWARD,
          HID_USAGE_SENSOR_EVENT_LOW_THRESHOLD_CROSS_DOWNWARD,
          HID_USAGE_SENSOR_EVENT_ZERO_THRESHOLD_CROSS_UPWARD,
          HID_USAGE_SENSOR_EVENT_ZERO_THRESHOLD_CROSS_DOWNWARD,
          HID_USAGE_SENSOR_EVENT_PERIOD_EXCEEDED,
          HID_USAGE_SENSOR_EVENT_FREQUENCY_EXCEEDED,
          HID_USAGE_SENSOR_EVENT_COMPLEX_TRIGGER,
          HID_INPUT(Const_Arr_Abs),
HID_END_COLLECTION,
HID_USAGE_SENSOR_DATA_MOTION_ACCELERATION_X_AXIS,
HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
HID_REPORT_SIZE(16),
HID_REPORT_COUNT(1),
// HID_USAGE_SENSOR_UNITS_G,
HID_UNIT_EXPONENT(0x0E), // scale default unit “G” to provide 2 digits past the decimal point
HID_INPUT(Const_Var_Abs),
HID_USAGE_SENSOR_DATA_MOTION_ACCELERATION_Y_AXIS,
HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
HID_REPORT_SIZE(16),
HID_REPORT_COUNT(1),
// HID_USAGE_SENSOR_UNITS_G,
HID_UNIT_EXPONENT(0x0E), // scale default unit “G” to provide 2 digits past the decimal point
HID_INPUT(Const_Var_Abs),
HID_USAGE_SENSOR_DATA_MOTION_ACCELERATION_Z_AXIS,
HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
HID_REPORT_SIZE(16),
HID_REPORT_COUNT(1),
// HID_USAGE_SENSOR_UNITS_G,
HID_UNIT_EXPONENT(0x0E), // scale default unit “G” to provide 2 digits past the decimal point
HID_INPUT(Const_Var_Abs),

HID_END_COLLECTION, // end of accelerometer nested sub-collection

HID_REPORT_ID(2),
HID_USAGE_PAGE_SENSOR,          // USAGE_PAGE (Sensor)
HID_USAGE_SENSOR_TYPE_LIGHT_AMBIENTLIGHT, // USAGE (AmbientLight)
HID_COLLECTION(Physical),     // second nested sub-collection, for ALS

//feature reports (xmit/receive)
HID_USAGE_PAGE_SENSOR,
HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_8(5),
HID_REPORT_SIZE(8),
HID_REPORT_COUNT(1),
HID_COLLECTION(Logical),
```

```
          HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_NO_EVENTS,
          HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS,
          HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_THRESHOLD_EVENTS,
          HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_NO_EVENTS_WAKE,
          HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS_WAKE,
          HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_THRESHOLD_EVENTS_WAKE,
          HID_FEATURE(Data_Arr_Abs),
HID_END_COLLECTION,
HID_USAGE_SENSOR_PROPERTY_SENSOR_STATUS,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_32(0xFF,0xFF,0xFF,0xFF),
HID_REPORT_SIZE(32),
HID_REPORT_COUNT(1),
HID_FEATURE(Data_Var_Abs), // up to VT_UI4 worth of status info
HID_USAGE_SENSOR_PROPERTY_REPORT_INTERVAL,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_32(0xFF,0xFF,0xFF,0xFF),
HID_REPORT_SIZE(32),
HID_REPORT_COUNT(1),
// HID_USAGE_SENSOR_UNITS_MILLISECOND,
HID_UNIT_EXPONENT(0),
HID_FEATURE(Data_Var_Abs),
HID_USAGE_SENSOR_PROPERTY_SENSOR_CONNECTION_TYPE,  // NAry
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_8(2),
HID_REPORT_SIZE(8),
HID_REPORT_COUNT(1),
HID_COLLECTION(Logical),
          HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_INTEGRATED,
          HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_ATTACHED,
          HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_EXTERNAL,
          HID_FEATURE(Const_Arr_Abs),
HID_END_COLLECTION,
HID_USAGE_SENSOR_PROPERTY_CHANGE_SENSITIVITY_REL_PCT,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_16(0x10,0x27), // 10000 = 0.00 to 100.00 percent with 2 digits past decimal point
HID_REPORT_SIZE(16),
HID_REPORT_COUNT(1),
// HID_USAGE_SENSOR_UNITS_PERCENT,
HID_UNIT_EXPONENT(0x0E), // scale default unit to provide 2 digits past the decimal point
HID_UNIT_EXPONENT(2),
HID_FEATURE(Data_Var_Abs),
HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_LIGHT_ILLUMINANCE,HID_USAGE_SENSOR_DATA_MOD_MAX),
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_16(0xFF,0xFF),
HID_REPORT_SIZE(16),
HID_REPORT_COUNT(1),
HID_USAGE_SENSOR_UNITS_LUX,
HID_UNIT_EXPONENT(0x0F), // scale unit to provide 1 digit past the decimal point
HID_FEATURE(Data_Var_Abs),
HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_LIGHT_ILLUMINANCE,HID_USAGE_SENSOR_DATA_MOD_MIN),
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_16(0xFF,0xFF),
HID_REPORT_SIZE(16),
HID_REPORT_COUNT(1),
HID_USAGE_SENSOR_UNITS_LUX,
HID_UNIT_EXPONENT(0x0F), // scale unit to provide 1 digit past the decimal point
HID_FEATURE(Data_Var_Abs),

//input reports (transmit)
HID_USAGE_PAGE_SENSOR,
HID_USAGE_SENSOR_STATE,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_8(6),
HID_REPORT_SIZE(8),
HID_REPORT_COUNT(1),
HID_COLLECTION(Logical),
          HID_USAGE_SENSOR_STATE_UNKNOWN,
          HID_USAGE_SENSOR_STATE_READY,
          HID_USAGE_SENSOR_STATE_NOT_AVAILABLE,
          HID_USAGE_SENSOR_STATE_NO_DATA,
          HID_USAGE_SENSOR_STATE_INITIALIZING,
          HID_USAGE_SENSOR_STATE_ACCESS_DENIED,
          HID_USAGE_SENSOR_STATE_ERROR,
          HID_INPUT(Const_Arr_Abs),
HID_END_COLLECTION,
HID_USAGE_SENSOR_EVENT,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_8(16),
HID_REPORT_SIZE(8),
HID_REPORT_COUNT(1),
HID_COLLECTION(Logical),
          HID_USAGE_SENSOR_EVENT_UNKNOWN,
          HID_USAGE_SENSOR_EVENT_STATE_CHANGED,
          HID_USAGE_SENSOR_EVENT_PROPERTY_CHANGED,
          HID_USAGE_SENSOR_EVENT_DATA_UPDATED,
          HID_USAGE_SENSOR_EVENT_POLL_RESPONSE,
          HID_USAGE_SENSOR_EVENT_CHANGE_SENSITIVITY,
          HID_USAGE_SENSOR_EVENT_MAX_REACHED,
          HID_USAGE_SENSOR_EVENT_MIN_REACHED,
          HID_USAGE_SENSOR_EVENT_HIGH_THRESHOLD_CROSS_UPWARD,
          HID_USAGE_SENSOR_EVENT_HIGH_THRESHOLD_CROSS_DOWNWARD,
          HID_USAGE_SENSOR_EVENT_LOW_THRESHOLD_CROSS_UPWARD,
          HID_USAGE_SENSOR_EVENT_LOW_THRESHOLD_CROSS_DOWNWARD,
          HID_USAGE_SENSOR_EVENT_ZERO_THRESHOLD_CROSS_UPWARD,
```

```
                    HID_USAGE_SENSOR_EVENT_ZERO_THRESHOLD_CROSS_DOWNWARD,
                    HID_USAGE_SENSOR_EVENT_PERIOD_EXCEEDED,
                    HID_USAGE_SENSOR_EVENT_FREQUENCY_EXCEEDED,
                    HID_USAGE_SENSOR_EVENT_COMPLEX_TRIGGER,
                    HID_INPUT(Const_Arr_Abs),
          HID_END_COLLECTION,
          HID_USAGE_SENSOR_DATA_LIGHT_ILLUMINANCE,
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_16(0xFF,0xFF),
          HID_USAGE_SENSOR_UNITS_LUX,
          HID_UNIT_EXPONENT(0x0F), // scale unit to provide 1 digit past the decimal point
          HID_REPORT_SIZE(16),
          HID_REPORT_COUNT(1),
          HID_INPUT(Const_Var_Abs),
          HID_USAGE_SENSOR_DATA_LIGHT_COLOR_TEMPERATURE,
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_16(0xFF,0xFF),
          HID_USAGE_SENSOR_UNITS_KELVIN,
          HID_UNIT_EXPONENT(0),
          HID_REPORT_SIZE(16),
          HID_REPORT_COUNT(1),
          HID_INPUT(Const_Var_Abs),
          HID_USAGE_SENSOR_DATA_LIGHT_CHROMATICITY_X,
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_16(0xFF,0xFF),
          HID_USAGE_SENSOR_UNITS_NOT_SPECIFIED,
          HID_UNIT_EXPONENT(0x0C), // scale unit to provide 4 digits past the decimal point
          HID_REPORT_SIZE(16),
          HID_REPORT_COUNT(1),
          HID_INPUT(Const_Var_Abs),
          HID_USAGE_SENSOR_DATA_LIGHT_CHROMATICITY_Y,
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_16(65535),
          HID_USAGE_SENSOR_UNITS_NOT_SPECIFIED,
          HID_UNIT_EXPONENT(0x0C), // scale unit to provide 4 digits past the decimal point
          HID_REPORT_SIZE(16),
          HID_REPORT_COUNT(1),
          HID_INPUT(Const_Var_Abs),

          HID_END_COLLECTION,           // end of ALS nested sub-collection

          HID_END_COLLECTION
};
```

The second example shows two Top Level Collections, one for each individual sensor:

```
// For reference: Complete HID report descriptor

// Two sensor example: 3D Accelerometer & AmbientLight
const unsigned char non_nest_col_report_descriptor[] = {
          HID_USAGE_PAGE_SENSOR,
          HID_USAGE_SENSOR_TYPE_MOTION_ACCELEROMETER_3D,
          HID_REPORT_ID(1),
          HID_COLLECTION(Application),  // first TLC, for accelerometer

          //feature reports (xmit/receive)
          HID_USAGE_PAGE_SENSOR,
          HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE,
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_8(5),
          HID_REPORT_SIZE(8),
          HID_REPORT_COUNT(1),
          HID_COLLECTION(Logical),
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_NO_EVENTS,
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS,
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_THRESHOLD_EVENTS,
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_NO_EVENTS_WAKE,
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS_WAKE,
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_THRESHOLD_EVENTS_WAKE,
                    HID_FEATURE(Data_Arr_Abs),
          HID_END_COLLECTION,
          HID_USAGE_SENSOR_PROPERTY_SENSOR_STATUS,
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_32(0xFF,0xFF,0xFF,0xFF),
          HID_REPORT_SIZE(32),
          HID_REPORT_COUNT(1),
          HID_FEATURE(Data_Var_Abs), // up to VT_UI4 worth of status info
          HID_USAGE_SENSOR_PROPERTY_REPORT_INTERVAL,
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_32(0xFF,0xFF,0xFF,0xFF),
          HID_REPORT_SIZE(32),
          HID_REPORT_COUNT(1),
          // HID_USAGE_SENSOR_UNITS_MILLISECOND,
          HID_UNIT_EXPONENT(0),
          HID_FEATURE(Data_Var_Abs),
          HID_USAGE_SENSOR_PROPERTY_SENSOR_CONNECTION_TYPE,  // NAry
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_8(2),
          HID_REPORT_SIZE(8),
          HID_REPORT_COUNT(1),
          HID_COLLECTION(Logical),
                    HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_INTEGRATED,
```

```
          HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_ATTACHED,
          HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_EXTERNAL,
          HID_FEATURE(Const_Arr_Abs),
HID_END_COLLECTION,
HID_USAGE_SENSOR_PROPERTY_CHANGE_SENSITIVITY_ABS,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_16(0xFF,0xFF),
HID_REPORT_SIZE(16),
HID_REPORT_COUNT(1),
// HID_USAGE_SENSOR_UNITS_G,
HID_UNIT_EXPONENT(0x0E), // scale default unit “G” to provide 2 digits past the decimal point
HID_FEATURE(Data_Var_Abs),
HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_MOTION_ACCELERATION,HID_USAGE_SENSOR_DATA_MOD_MAX),
HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
HID_REPORT_SIZE(16),
HID_REPORT_COUNT(1),
// HID_USAGE_SENSOR_UNITS_G,
HID_UNIT_EXPONENT(0x0E), // scale default unit “G” to provide 2 digits past the decimal point
HID_FEATURE(Data_Var_Abs),
HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_MOTION_ACCELERATION,HID_USAGE_SENSOR_DATA_MOD_MIN),
HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
HID_REPORT_SIZE(16),
HID_REPORT_COUNT(1),
// HID_USAGE_SENSOR_UNITS_G,
HID_UNIT_EXPONENT(0x0E), // scale default unit “G” to provide 2 digits past the decimal point
HID_FEATURE(Data_Var_Abs),

//input reports (transmit)
HID_USAGE_PAGE_SENSOR,
HID_USAGE_SENSOR_STATE,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_8(6),
HID_REPORT_SIZE(8),
HID_REPORT_COUNT(1),
HID_COLLECTION(Logical),
          HID_USAGE_SENSOR_STATE_UNKNOWN,
          HID_USAGE_SENSOR_STATE_READY,
          HID_USAGE_SENSOR_STATE_NOT_AVAILABLE,
          HID_USAGE_SENSOR_STATE_NO_DATA,
          HID_USAGE_SENSOR_STATE_INITIALIZING,
          HID_USAGE_SENSOR_STATE_ACCESS_DENIED,
          HID_USAGE_SENSOR_STATE_ERROR,
          HID_INPUT(Const_Arr_Abs),
HID_END_COLLECTION,
HID_USAGE_SENSOR_EVENT,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_8(16),
HID_REPORT_SIZE(8),
HID_REPORT_COUNT(1),
HID_COLLECTION(Logical),
          HID_USAGE_SENSOR_EVENT_UNKNOWN,
          HID_USAGE_SENSOR_EVENT_STATE_CHANGED,
          HID_USAGE_SENSOR_EVENT_PROPERTY_CHANGED,
          HID_USAGE_SENSOR_EVENT_DATA_UPDATED,
          HID_USAGE_SENSOR_EVENT_POLL_RESPONSE,
          HID_USAGE_SENSOR_EVENT_CHANGE_SENSITIVITY,
          HID_USAGE_SENSOR_EVENT_MAX_REACHED,
          HID_USAGE_SENSOR_EVENT_MIN_REACHED,
          HID_USAGE_SENSOR_EVENT_HIGH_THRESHOLD_CROSS_UPWARD,
          HID_USAGE_SENSOR_EVENT_HIGH_THRESHOLD_CROSS_DOWNWARD,
          HID_USAGE_SENSOR_EVENT_LOW_THRESHOLD_CROSS_UPWARD,
          HID_USAGE_SENSOR_EVENT_LOW_THRESHOLD_CROSS_DOWNWARD,
          HID_USAGE_SENSOR_EVENT_ZERO_THRESHOLD_CROSS_UPWARD,
          HID_USAGE_SENSOR_EVENT_ZERO_THRESHOLD_CROSS_DOWNWARD,
          HID_USAGE_SENSOR_EVENT_PERIOD_EXCEEDED,
          HID_USAGE_SENSOR_EVENT_FREQUENCY_EXCEEDED,
          HID_USAGE_SENSOR_EVENT_COMPLEX_TRIGGER,
          HID_INPUT(Const_Arr_Abs),
HID_END_COLLECTION,
HID_USAGE_SENSOR_DATA_MOTION_ACCELERATION_X_AXIS,
HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
HID_REPORT_SIZE(16),
HID_REPORT_COUNT(1),
// HID_USAGE_SENSOR_UNITS_G,
HID_UNIT_EXPONENT(0x0E), // scale default unit “G” to provide 2 digits past the decimal point
HID_INPUT(Const_Var_Abs),
HID_USAGE_SENSOR_DATA_MOTION_ACCELERATION_Y_AXIS,
HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
HID_REPORT_SIZE(16),
HID_REPORT_COUNT(1),
// HID_USAGE_SENSOR_UNITS_G,
HID_UNIT_EXPONENT(0x0E), // scale default unit “G” to provide 2 digits past the decimal point
HID_INPUT(Const_Var_Abs),
HID_USAGE_SENSOR_DATA_MOTION_ACCELERATION_Z_AXIS,
HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
HID_REPORT_SIZE(16),
HID_REPORT_COUNT(1),
// HID_USAGE_SENSOR_UNITS_G,
HID_UNIT_EXPONENT(0x0E), // scale default unit “G” to provide 2 digits past the decimal point
HID_INPUT(Const_Var_Abs),
```

```
HID_END_COLLECTION, // end of accelerometer TLC

HID_USAGE_PAGE_SENSOR,         // USAGE_PAGE (Sensor)
HID_USAGE_SENSOR_TYPE_LIGHT_AMBIENTLIGHT, // USAGE (AmbientLight)
HID_REPORT_ID(2),
HID_COLLECTION(Application),  // second TLC, for ALS

//feature reports (xmit/receive)
HID_USAGE_PAGE_SENSOR,
HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_8(5),
HID_REPORT_SIZE(8),
HID_REPORT_COUNT(1),
HID_COLLECTION(Logical),
          HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_NO_EVENTS,
          HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS,
          HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_THRESHOLD_EVENTS,
          HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_NO_EVENTS_WAKE,
          HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS_WAKE,
          HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_THRESHOLD_EVENTS_WAKE,
          HID_FEATURE(Data_Arr_Abs),
HID_END_COLLECTION,
HID_USAGE_SENSOR_PROPERTY_SENSOR_STATUS,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_32(0xFF,0xFF,0xFF,0xFF),
HID_REPORT_SIZE(32),
HID_REPORT_COUNT(1),
HID_FEATURE(Data_Var_Abs), // up to VT_UI4 worth of status info
HID_USAGE_SENSOR_PROPERTY_REPORT_INTERVAL,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_32(0xFF,0xFF,0xFF,0xFF),
HID_REPORT_SIZE(32),
HID_REPORT_COUNT(1),
// HID_USAGE_SENSOR_UNITS_MILLISECOND,
HID_UNIT_EXPONENT(0),
HID_FEATURE(Data_Var_Abs),
HID_USAGE_SENSOR_PROPERTY_SENSOR_CONNECTION_TYPE,  // NAry
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_8(2),
HID_REPORT_SIZE(8),
HID_REPORT_COUNT(1),
HID_COLLECTION(Logical),
          HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_INTEGRATED,
          HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_ATTACHED,
          HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_EXTERNAL,
          HID_FEATURE(Const_Arr_Abs),
HID_END_COLLECTION,
HID_USAGE_SENSOR_PROPERTY_CHANGE_SENSITIVITY_REL_PCT,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_16(0x10,0x27), // 10000 = 0.00 to 100.00 percent with 2 digits past decimal point
HID_REPORT_SIZE(16),
HID_REPORT_COUNT(1),
// HID_USAGE_SENSOR_UNITS_PERCENT,
HID_UNIT_EXPONENT(0x0E), // scale unit to provide 2 digits past the decimal point
HID_FEATURE(Data_Var_Abs),
HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_LIGHT_ILLUMINANCE,HID_USAGE_SENSOR_DATA_MOD_MAX),
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_16(0xFF,0xFF),
HID_REPORT_SIZE(16),
HID_REPORT_COUNT(1),
HID_USAGE_SENSOR_UNITS_LUX,
HID_UNIT_EXPONENT(0x0F), // scale unit to provide 1 digit past the decimal point
HID_FEATURE(Data_Var_Abs),
HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_LIGHT_ILLUMINANCE,HID_USAGE_SENSOR_DATA_MOD_MIN),
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_16(0xFF,0xFF),
HID_REPORT_SIZE(16),
HID_REPORT_COUNT(1),
HID_USAGE_SENSOR_UNITS_LUX,
HID_UNIT_EXPONENT(0x0F), // scale unit to provide 1 digit past the decimal point
HID_FEATURE(Data_Var_Abs),

//input reports (transmit)
HID_USAGE_PAGE_SENSOR,
HID_USAGE_SENSOR_STATE,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_8(6),
HID_REPORT_SIZE(8),
HID_REPORT_COUNT(1),
HID_COLLECTION(Logical),
          HID_USAGE_SENSOR_STATE_UNKNOWN,
          HID_USAGE_SENSOR_STATE_READY,
          HID_USAGE_SENSOR_STATE_NOT_AVAILABLE,
          HID_USAGE_SENSOR_STATE_NO_DATA,
          HID_USAGE_SENSOR_STATE_INITIALIZING,
          HID_USAGE_SENSOR_STATE_ACCESS_DENIED,
          HID_USAGE_SENSOR_STATE_ERROR,
          HID_INPUT(Const_Arr_Abs),
HID_END_COLLECTION,
HID_USAGE_SENSOR_EVENT,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_8(16),
HID_REPORT_SIZE(8),
```

```
        HID_REPORT_COUNT(1),
        HID_COLLECTION(Logical),
                HID_USAGE_SENSOR_EVENT_UNKNOWN,
                HID_USAGE_SENSOR_EVENT_STATE_CHANGED,
                HID_USAGE_SENSOR_EVENT_PROPERTY_CHANGED,
                HID_USAGE_SENSOR_EVENT_DATA_UPDATED,
                HID_USAGE_SENSOR_EVENT_POLL_RESPONSE,
                HID_USAGE_SENSOR_EVENT_CHANGE_SENSITIVITY,
                HID_USAGE_SENSOR_EVENT_MAX_REACHED,
                HID_USAGE_SENSOR_EVENT_MIN_REACHED,
                HID_USAGE_SENSOR_EVENT_HIGH_THRESHOLD_CROSS_UPWARD,
                HID_USAGE_SENSOR_EVENT_HIGH_THRESHOLD_CROSS_DOWNWARD,
                HID_USAGE_SENSOR_EVENT_LOW_THRESHOLD_CROSS_UPWARD,
                HID_USAGE_SENSOR_EVENT_LOW_THRESHOLD_CROSS_DOWNWARD,
                HID_USAGE_SENSOR_EVENT_ZERO_THRESHOLD_CROSS_UPWARD,
                HID_USAGE_SENSOR_EVENT_ZERO_THRESHOLD_CROSS_DOWNWARD,
                HID_USAGE_SENSOR_EVENT_PERIOD_EXCEEDED,
                HID_USAGE_SENSOR_EVENT_FREQUENCY_EXCEEDED,
                HID_USAGE_SENSOR_EVENT_COMPLEX_TRIGGER,
                HID_INPUT(Const_Arr_Abs),
        HID_END_COLLECTION,
        HID_USAGE_SENSOR_DATA_LIGHT_ILLUMINANCE,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_16(0xFF,0xFF),
        HID_USAGE_SENSOR_UNITS_LUX,
        HID_UNIT_EXPONENT(0x0F), // scale unit to provide 1 digit past the decimal point
        HID_REPORT_SIZE(16),
        HID_REPORT_COUNT(1),
        HID_INPUT(Const_Var_Abs),
        HID_USAGE_SENSOR_DATA_LIGHT_COLOR_TEMPERATURE,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_16(0xFF,0xFF),
        HID_USAGE_SENSOR_UNITS_KELVIN,
        HID_UNIT_EXPONENT(0),
        HID_REPORT_SIZE(16),
        HID_REPORT_COUNT(1),
        HID_INPUT(Const_Var_Abs),
        HID_USAGE_SENSOR_DATA_LIGHT_CHROMATICITY_X,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_16(0xFF,0xFF),
        HID_USAGE_SENSOR_UNITS_NOT_SPECIFIED,
        HID_UNIT_EXPONENT(0x0C), // scale unit to provide 4 digits past the decimal point
        HID_REPORT_SIZE(16),
        HID_REPORT_COUNT(1),
        HID_INPUT(Const_Var_Abs),
        HID_USAGE_SENSOR_DATA_LIGHT_CHROMATICITY_Y,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_16(65535),
        HID_USAGE_SENSOR_UNITS_NOT_SPECIFIED,
        HID_UNIT_EXPONENT(0x0C), // scale unit to provide 4 digits past the decimal point
        HID_REPORT_SIZE(16),
        HID_REPORT_COUNT(1),
        HID_INPUT(Const_Var_Abs),

        HID_END_COLLECTION,           // end of ALS TLC
};
```

4.2.6 Custom Sensor

The Custom sensor provides a simple means by which platform driver support can be extended dynamically without changes to the platform driver. The Custom sensor achieves this by encapsulating datafields within known structures that are defined within the driver.

Use of a Custom sensor may be desireable in several circumstances:

1) The vendor may wish to extend the use of an existing device driver (one that must already support the Custom sensor definition)
2) The vendor may wish to obfuscate the data being communicated by a sensor; without knowing the mapping between the data and its encapsulated form, it is difficult for an application to determine what data is being communcated by the sensor.

In order for an application to be able to make sense of these encapsulated datafields it must have fore knowledge of the mapping between unencapsulated datafields and and their subsequent presentation by the driver. Providing this mapping is the responsibility of the sensor vendor.

Provision is made for communicating three types of data:

1) The HID Sensor Usage, which can be placed in the `HID_USAGE_SENSOR_DATA_CUSTOM_USAGE` field. This can provide a hint to the application about the type of sensor that has been encapsulated
2) An entry in a field of BOOLEAN values, `HID_USAGE_SENSOR_DATA_CUSTOM_BOOLEAN_ARRAY`. This field can be used if a large number of BOOLEAN values are supported by the sensor
3) An entry in one of six (6) datafields, labeled `HID_USAGE_SENSOR_DATA_CUSTOM_VALUE_`*n* (where *n* is a number from 1 through 6.) This field provides for the use of the UnitExp usage. If the UnitExp usage = ‘0’ the encapsulated value is assumed to be an integer value; if the UnitExp usage is anything other than ‘0’ the encapsulated value is assumed to be the mantissa of a fixed-point number the exponent of which is contained in the UnitExp field.

Following is a Custom Sensor Report Descriptor that illustrates these concepts. If we apply this example, the encapsulated fields would be populated as follows for a Speedometer sensor (HID Usage = `HID_USAGE_SENSOR_TYPE_MOTION_SPEEDOMETER`).

```
const unsigned char cus_report_descriptor[] = {
        HID_USAGE_PAGE_SENSOR,         // USAGE_PAGE (Sensor)
        HID_USAGE_SENSOR_TYPE_SIMPLE_CUSTOM, // USAGE (Simple Custom)
        HID_COLLECTION(Physical),

        //feature reports (xmit/receive)
        HID_USAGE_PAGE_SENSOR,
        HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_8(5),
        HID_REPORT_SIZE(8),
        HID_REPORT_COUNT(1),
        HID_COLLECTION(Logical),
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_NO_EVENTS,
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS,
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_THRESHOLD_EVENTS,
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_NO_EVENTS_WAKE,
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS_WAKE,
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_THRESHOLD_EVENTS_WAKE,
                HID_FEATURE(Data_Arr_Abs),
        HID_END_COLLECTION,
        HID_USAGE_SENSOR_PROPERTY_SENSOR_STATUS,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_32(0xFF,0xFF,0xFF,0xFF),
        HID_REPORT_SIZE(32),
        HID_REPORT_COUNT(1),
        HID_FEATURE(Data_Var_Abs), // up to VT_UI4 worth of status info
        HID_USAGE_SENSOR_PROPERTY_REPORT_INTERVAL,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_32(0xFF,0xFF,0xFF,0xFF),
        HID_REPORT_SIZE(32),
        HID_REPORT_COUNT(1),
        // HID_USAGE_SENSOR_UNITS_MILLISECOND,
        HID_UNIT_EXPONENT(0),
        HID_FEATURE(Data_Var_Abs),
        HID_USAGE_SENSOR_PROPERTY_SENSOR_CONNECTION_TYPE,  // NAry
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_8(2),
        HID_REPORT_SIZE(8),
        HID_REPORT_COUNT(1),
        HID_COLLECTION(Logical),
                HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_INTEGRATED,
                HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_ATTACHED,
                HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_EXTERNAL,
                HID_FEATURE(Const_Arr_Abs),
        HID_END_COLLECTION,
        HID_USAGE_SENSOR_PROPERTY_CHANGE_SENSITIVITY_ABS,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_16(0xFF,0xFF),
        HID_REPORT_SIZE(16),
        HID_REPORT_COUNT(1),
        HID_USAGE_SENSOR_UNITS_NOT_SPECIFIED,
        HID_UNIT_EXPONENT(0x0E), // scale unit to provide 2 digits past the decimal point
        HID_FEATURE(Data_Var_Abs),
        HID_USAGE_SENSOR_PROPERTY_RANGE_MAXIMUM,
        HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
        HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
        HID_REPORT_SIZE(16),
        HID_REPORT_COUNT(1),
        HID_USAGE_SENSOR_UNITS_NOT_SPECIFIED,
        HID_UNIT_EXPONENT(0x0E), // scale unit to provide 2 digits past the decimal point
        HID_FEATURE(Data_Var_Abs),
        HID_USAGE_SENSOR_PROPERTY_RANGE_MINIMUM,
        HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
        HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
        HID_REPORT_SIZE(16),
```

```
        HID_REPORT_COUNT(1),
        HID_USAGE_SENSOR_UNITS_NOT_SPECIFIED,
        HID_UNIT_EXPONENT(0x0E), // scale unit to provide 2 digits past the decimal point
        HID_FEATURE(Data_Var_Abs),

        //input reports (transmit)
        HID_USAGE_PAGE_SENSOR,
        HID_USAGE_SENSOR_STATE,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_8(6),
        HID_REPORT_SIZE(8),
        HID_REPORT_COUNT(1),
        HID_COLLECTION(Logical),
                HID_USAGE_SENSOR_STATE_UNKNOWN,
                HID_USAGE_SENSOR_STATE_READY,
                HID_USAGE_SENSOR_STATE_NOT_AVAILABLE,
                HID_USAGE_SENSOR_STATE_NO_DATA,
                HID_USAGE_SENSOR_STATE_INITIALIZING,
                HID_USAGE_SENSOR_STATE_ACCESS_DENIED,
                HID_USAGE_SENSOR_STATE_ERROR,
                HID_INPUT(Const_Arr_Abs),
        HID_END_COLLECTION,
        HID_USAGE_SENSOR_EVENT,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_8(16),
        HID_REPORT_SIZE(8),
        HID_REPORT_COUNT(1),
        HID_COLLECTION(Logical),
                HID_USAGE_SENSOR_EVENT_UNKNOWN,
                HID_USAGE_SENSOR_EVENT_STATE_CHANGED,
                HID_USAGE_SENSOR_EVENT_PROPERTY_CHANGED,
                HID_USAGE_SENSOR_EVENT_DATA_UPDATED,
                HID_USAGE_SENSOR_EVENT_POLL_RESPONSE,
                HID_USAGE_SENSOR_EVENT_CHANGE_SENSITIVITY,
                HID_USAGE_SENSOR_EVENT_MAX_REACHED,
                HID_USAGE_SENSOR_EVENT_MIN_REACHED,
                HID_USAGE_SENSOR_EVENT_HIGH_THRESHOLD_CROSS_UPWARD,
                HID_USAGE_SENSOR_EVENT_HIGH_THRESHOLD_CROSS_DOWNWARD,
                HID_USAGE_SENSOR_EVENT_LOW_THRESHOLD_CROSS_UPWARD,
                HID_USAGE_SENSOR_EVENT_LOW_THRESHOLD_CROSS_DOWNWARD,
                HID_USAGE_SENSOR_EVENT_ZERO_THRESHOLD_CROSS_UPWARD,
                HID_USAGE_SENSOR_EVENT_ZERO_THRESHOLD_CROSS_DOWNWARD,
                HID_USAGE_SENSOR_EVENT_PERIOD_EXCEEDED,
                HID_USAGE_SENSOR_EVENT_FREQUENCY_EXCEEDED,
                HID_USAGE_SENSOR_EVENT_COMPLEX_TRIGGER,
                HID_INPUT(Const_Arr_Abs),
        HID_END_COLLECTION,
        HID_USAGE_SENSOR_DATA_CUSTOM_USAGE,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_16(0xFF,0xFF),
        HID_REPORT_SIZE(16),
        HID_REPORT_COUNT(1),
        HID_INPUT(Const_Var_Abs), // = HID_USAGE_SENSOR_TYPE_MOTION_SPEEDOMETER
        HID_USAGE_SENSOR_DATA_CUSTOM_VALUE_1,
        HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
        HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
        HID_REPORT_SIZE(16),
        HID_REPORT_COUNT(1),
        HID_USAGE_SENSOR_UNITS_NOT_SPECIFIED,
        HID_UNIT_EXPONENT(0x0E), // scale unit to provide 2 digits past the decimal point
        HID_INPUT(Const_Var_Abs), // = HID_USAGE_SENSOR_DATA_MOTION_SPEED value

        HID_END_COLLECTION
};
```

Following is a complete report descriptor that illustrates fields not used in the above example:

```
// Complete HID report descriptor

const unsigned char cus_report_descriptor[] = {
        HID_USAGE_PAGE_SENSOR,          // USAGE_PAGE (Sensor)
        HID_USAGE_SENSOR_TYPE_SIMPLE_CUSTOM, // USAGE (Simple Custom)
        HID_COLLECTION(Physical),

        //feature reports (xmit/receive)
        HID_USAGE_PAGE_SENSOR,
        HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_8(5),
        HID_REPORT_SIZE(8),
        HID_REPORT_COUNT(1),
        HID_COLLECTION(Logical),
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_NO_EVENTS,
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS,
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_THRESHOLD_EVENTS,
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_NO_EVENTS_WAKE,
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS_WAKE,
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_THRESHOLD_EVENTS_WAKE,
                HID_FEATURE(Data_Arr_Abs),
        HID_END_COLLECTION,
        HID_USAGE_SENSOR_PROPERTY_SENSOR_STATUS,
        HID_LOGICAL_MIN_8(0),
```

```
HID_LOGICAL_MAX_32(0xFF,0xFF,0xFF,0xFF),
HID_REPORT_SIZE(32),
HID_REPORT_COUNT(1),
HID_FEATURE(Data_Var_Abs), // up to VT_UI4 worth of status info
HID_USAGE_SENSOR_PROPERTY_REPORT_INTERVAL,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_32(0xFF,0xFF,0xFF,0xFF),
HID_REPORT_SIZE(32),
HID_REPORT_COUNT(1),
// HID_USAGE_SENSOR_UNITS_MILLISECOND,
HID_UNIT_EXPONENT(0),
HID_FEATURE(Data_Var_Abs),
HID_USAGE_SENSOR_PROPERTY_SENSOR_CONNECTION_TYPE,  // NAry
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_8(2),
HID_REPORT_SIZE(8),
HID_REPORT_COUNT(1),
HID_COLLECTION(Logical),
          HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_INTEGRATED,
          HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_ATTACHED,
          HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_EXTERNAL,
          HID_FEATURE(Const_Arr_Abs),
HID_END_COLLECTION,
HID_USAGE_SENSOR_PROPERTY_CHANGE_SENSITIVITY_ABS,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_16(0xFF,0xFF),
HID_REPORT_SIZE(16),
HID_REPORT_COUNT(1),
HID_USAGE_SENSOR_UNITS_NOT_SPECIFIED,
HID_UNIT_EXPONENT(0x0E), // scale unit to provide 2 digits past the decimal point
HID_FEATURE(Data_Var_Abs),
HID_USAGE_SENSOR_PROPERTY_RANGE_MAXIMUM,
HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
HID_REPORT_SIZE(16),
HID_REPORT_COUNT(1),
HID_USAGE_SENSOR_UNITS_NOT_SPECIFIED,
HID_UNIT_EXPONENT(0x0E), // scale unit to provide 2 digits past the decimal point
HID_FEATURE(Data_Var_Abs),
HID_USAGE_SENSOR_PROPERTY_RANGE_MINIMUM,
HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
HID_REPORT_SIZE(16),
HID_REPORT_COUNT(1),
HID_USAGE_SENSOR_UNITS_NOT_SPECIFIED,
HID_UNIT_EXPONENT(0x0E), // scale unit to provide 2 digits past the decimal point
HID_FEATURE(Data_Var_Abs),

//input reports (transmit)
HID_USAGE_PAGE_SENSOR,
HID_USAGE_SENSOR_STATE,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_8(6),
HID_REPORT_SIZE(8),
HID_REPORT_COUNT(1),
HID_COLLECTION(Logical),
          HID_USAGE_SENSOR_STATE_UNKNOWN,
          HID_USAGE_SENSOR_STATE_READY,
          HID_USAGE_SENSOR_STATE_NOT_AVAILABLE,
          HID_USAGE_SENSOR_STATE_NO_DATA,
          HID_USAGE_SENSOR_STATE_INITIALIZING,
          HID_USAGE_SENSOR_STATE_ACCESS_DENIED,
          HID_USAGE_SENSOR_STATE_ERROR,
          HID_INPUT(Const_Arr_Abs),
HID_END_COLLECTION,
HID_USAGE_SENSOR_EVENT,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_8(16),
HID_REPORT_SIZE(8),
HID_REPORT_COUNT(1),
HID_COLLECTION(Logical),
          HID_USAGE_SENSOR_EVENT_UNKNOWN,
          HID_USAGE_SENSOR_EVENT_STATE_CHANGED,
          HID_USAGE_SENSOR_EVENT_PROPERTY_CHANGED,
          HID_USAGE_SENSOR_EVENT_DATA_UPDATED,
          HID_USAGE_SENSOR_EVENT_POLL_RESPONSE,
          HID_USAGE_SENSOR_EVENT_CHANGE_SENSITIVITY,
          HID_USAGE_SENSOR_EVENT_MAX_REACHED,
          HID_USAGE_SENSOR_EVENT_MIN_REACHED,
          HID_USAGE_SENSOR_EVENT_HIGH_THRESHOLD_CROSS_UPWARD,
          HID_USAGE_SENSOR_EVENT_HIGH_THRESHOLD_CROSS_DOWNWARD,
          HID_USAGE_SENSOR_EVENT_LOW_THRESHOLD_CROSS_UPWARD,
          HID_USAGE_SENSOR_EVENT_LOW_THRESHOLD_CROSS_DOWNWARD,
          HID_USAGE_SENSOR_EVENT_ZERO_THRESHOLD_CROSS_UPWARD,
          HID_USAGE_SENSOR_EVENT_ZERO_THRESHOLD_CROSS_DOWNWARD,
          HID_USAGE_SENSOR_EVENT_PERIOD_EXCEEDED,
          HID_USAGE_SENSOR_EVENT_FREQUENCY_EXCEEDED,
          HID_USAGE_SENSOR_EVENT_COMPLEX_TRIGGER,
          HID_INPUT(Const_Arr_Abs),
HID_END_COLLECTION,
HID_USAGE_SENSOR_DATA_CUSTOM_USAGE,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_16(0xFF,0xFF),
HID_REPORT_SIZE(16),
HID_REPORT_COUNT(1),
```

```
        HID_INPUT(Const_Var_Abs),
        HID_USAGE_SENSOR_DATA_CUSTOM_BOOLEAN_ARRAY,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_16(0xFF,0xFF),
        HID_REPORT_SIZE(16),
        HID_REPORT_COUNT(1),
        HID_INPUT(Const_Var_Abs),
        HID_USAGE_SENSOR_DATA_CUSTOM_VALUE_1,
        HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
        HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
        HID_REPORT_SIZE(16),
        HID_REPORT_COUNT(1),
        HID_USAGE_SENSOR_UNITS_NOT_SPECIFIED,
        HID_UNIT_EXPONENT(0x0E), // scale unit to provide 2 digits past the decimal point
        HID_INPUT(Const_Var_Abs),
        HID_USAGE_SENSOR_DATA_CUSTOM_VALUE_2,
        HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
        HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
        HID_REPORT_SIZE(16),
        HID_REPORT_COUNT(1),
        HID_USAGE_SENSOR_UNITS_NOT_SPECIFIED,
        HID_UNIT_EXPONENT(0x0E), // scale unit to provide 2 digits past the decimal point
        HID_INPUT(Const_Var_Abs),
        HID_USAGE_SENSOR_DATA_CUSTOM_VALUE_3,
        HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
        HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
        HID_REPORT_SIZE(16),
        HID_REPORT_COUNT(1),
        HID_USAGE_SENSOR_UNITS_NOT_SPECIFIED,
        HID_UNIT_EXPONENT(0x0E), // scale unit to provide 2 digits past the decimal point
        HID_INPUT(Const_Var_Abs),
        HID_USAGE_SENSOR_DATA_CUSTOM_VALUE_4,
        HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
        HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
        HID_REPORT_SIZE(16),
        HID_REPORT_COUNT(1),
        HID_USAGE_SENSOR_UNITS_NOT_SPECIFIED,
        HID_UNIT_EXPONENT(0x0E), // scale unit to provide 2 digits past the decimal point
        HID_INPUT(Const_Var_Abs),
        HID_USAGE_SENSOR_DATA_CUSTOM_VALUE_5,
        HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
        HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
        HID_REPORT_SIZE(16),
        HID_REPORT_COUNT(1),
        HID_USAGE_SENSOR_UNITS_NOT_SPECIFIED,
        HID_UNIT_EXPONENT(0x0E), // scale unit to provide 2 digits past the decimal point
        HID_INPUT(Const_Var_Abs),
        HID_USAGE_SENSOR_DATA_CUSTOM_VALUE_6,
        HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
        HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
        HID_REPORT_SIZE(16),
        HID_REPORT_COUNT(1),
        HID_USAGE_SENSOR_UNITS_NOT_SPECIFIED,
        HID_UNIT_EXPONENT(0x0E), // scale unit to provide 2 digits past the decimal point
        HID_INPUT(Const_Var_Abs),

        HID_END_COLLECTION
};
```

4.2.7 Generic Sensor

The Generic sensor provides a comprehensive means by which platform driver support can be extended dynamically without changes to the platform driver. The Generic sensor achieves this by directly exposing the sensor category, type and supported events, properties and datafields by means of GUIDs and PROPERTYKEYs, which can consumed directly by the Operating System's sensor driver without having to parse and translate HID Usages.

A Generic sensor is different from a Custom sensor in that, where the Custom sensor presents itself at the platform level as a Custom sensor with certain properties, the Generic sensor presents itself as the Category and Type of sensor defined in the Category and Type fields. A further distinction is that, where the Custom sensor must encapsulate datafields in one of the defined Custom datafields, the Generic sensor can represent any supported event, property or datafield defined in the Event, Property and Datafield fields.

Use of a Generic sensor may be desireable in several circumstances:

1) The vendor may wish create a new category or type of sensor not previously anticipated by the platform driver;

2) The vendor may wish to add new events, properties or datafields to an existing sensor that is already supported as a Generic sensor;
3) The vendor may wish to obfuscate the data being communicated by a sensor and tunnel it through the platform driver up to the application as “opaque data”. Without knowing the mapping between the GUIDs and PROPERTYKEY representations of the sensor category, and type and supported events, properties and datafields, it is difficult for an application to determine what data is being communcated by the sensor.

In order for an application to be able to make sense of the exposed sensor category and type and supported events, properties and datafields it must have fore knowledge of the mapping between the exposed properties and datafields and and their subsequent presentation by the platform driver. Providing this mapping is the responsibility of the sensor vendor, ideally in conjuction with the platform vendor.

Provision is made for communicating six types of data:

1) The Sensor Category (a GUID), which is placed in the `HID_USAGE_SENSOR_DATA_GENERIC_CATEGORY_GUID` field;
2) The Sensor Type (a GUID) which is placed in the `HID_USAGE_SENSOR_GENERIC_DATA_TYPE_GUID` field;
3) One or more Sensor Events (each a PROPERTYKEY) which is placed in the `HID_USAGE_SENSOR_DATA_GENERIC_EVENT_PROPERTYKEY` field;
4) The `HID_USAGE_SENSOR_DATA_GENERIC_GUID_OR_PROPERTYKEY` field, which provides identification and type information regarding the Property or Data Field;
5) Sensor Properties:
    a. Each is identified by a PROPERTYKEY which is placed in the `HID_USAGE_SENSOR_DATA_GENERIC_PROPERTY_PROPERTYKEY` or `HID_USAGE_SENSOR_DATA_GENERIC_GUID_OR_PROPERTYKEY` field. This field provides for the use of the UnitExp usage. If the UnitExp usage = ‘0’ the encapsulated value is assumed to be an integer value; if the UnitExp usage is anything other than ‘0’ the encapsulated value is assumed to be the mantissa of a fixed-point number the exponent of which is contained in the UnitExp field;
    b. Each Property’s data is placed in the `HID_USAGE_SENSOR_DATA_GENERIC_PROPERTY` field;
6) One or more Sensor Datafields:
    a. Each is identified by a PROPERTYKEY which is placed in the `HID_USAGE_DATA_SENSOR_GENERIC_DATAFIELD_PROPERTYKEY` or `HID_USAGE_SENSOR_DATA_GENERIC_GUID_OR_PROPERTYKEY` field. This field provides for the use of the UnitExp usage. If the UnitExp usage = ‘0’ the encapsulated value is assumed to be an integer value; if the UnitExp usage is anything other than ‘0’ the encapsulated value is assumed to be the mantissa of a fixed-point number the exponent of which is contained in the UnitExp field;
    b. Each Data Fields’ data is placed in the `HID_USAGE_SENSOR_DATA_GENERIC_DATAFIELD` field;

These six types of data may be mixed with the non-generic HID Usages defined in this document.

GUIDs and PROPERTYKEYS are defined as follows:

```
typedef struct _GUID
{
          unsigned long       Data1;
          unsigned short      Data2;
          unsigned short      Data3;
          unsigned char       Data4[ 8 ];
} GUID;

typedef struct _PROPERTYKEY {
          GUID                fmtid;
```

```
          unsigned long       pid;
} PROPERTYKEY;
```

Actual data can be described in a single structure called a VARIANT type, as follows:

```
#define VARIANT_BOOL          unsigned char // use LSb of byte
#define HIDFLOAT16            unsigned short // 16-bit mantissa
#define HIDFLOAT32            unsigned long // 32-bit mantissa
typedef union _flattened_variant_u {
          VARIANT_BOOL        boolVal;
          unsigned char       bVal;
          unsigned short      uiVal;
          short               iVal;
          unsigned long       ulVal;
          long                lVal;
          unsigned long long  uhVal;
          long long           hVal;
          float               fltVal;
          double              dblVal;
          wchar_t             wszVal[ WIDE_STRING_SIZE_MAX ];
          char                szVal[ STRING_SIZE_MAX ];
          FLAT_VARIANTVEC     vecVal;
          HIDFLOAT16          f16Val;
          HIDFLOAT32          f32Val;
          GUID                guidVal;
} FLAT_VARIANT_UNION;
```

Special mention needs to be made about the VARIANTVEC type, which is used to store compex “struct” data types, such as used for:

- HID_USAGE_SENSOR_PROPERTY_RESPONSE_CURVE
- HID_USAGE_SENSOR_DATA_ORIENTATION_ROTATION_MATRIX
- HID_USAGE_SENSOR_DATA_ORIENTATION_QUATERNION
- HID_USAGE_SENSOR_DATA_GENERIC_GUID_OR_PROPERTYKEY
- Any other vendor-opaque data

The definition of the VARIANTVEC structure is:

```
typedef struct _flattened_variant_vector_t {
          // the number valid BYTEs in paubElems
          unsigned long       caubElems;
          // array of BYTEs holding the data
          unsigned char       paubElems[ VECTOR_BYTE_SIZE_MAX ];
} FLAT_VARIANTVEC;
```

The struct data is type-cast over the top of the paubElems field, and the length (in bytes) of the struct is stored in the caubElems field.

Here is an example for SENSOR_DATA_TYPE_LIGHT_RESPONSE_CURVE, which MSDN says each value element must be 32-bits (size of VT_UI4), allowing 14 of them to fit into a USB Interrupt Packet:

```
struct _light_response_curve {
          // LCD Display should be set to this percentage when...
          int DisplayBrightnessPercent : 8; // 0% to 100%
          // ...Illuminance Lux up to 64K is at this value
          int CorrespondingLuxValue: 24; // 0 to 100,000 Lux
} LightResponseCurve[ 14 ];

FLAT_VARIANT_UNION fvu;
fvu.vecVal.caubElems = (unsigned long)sizeof(LightResponseCurve);
struct _light_response_curve *pLRC = (struct _light_response_curve *)fvu.vecVal.paubElems;
pLRC->DisplayBrightnessPercent = 10;
pLRC->CorrespondingLuxValue = 10;
pLRC++;
pLRC->DisplayBrightnessPercent = 50;
pLRC->CorrespondingLuxValue = 100;
pLRC++;
pLRC->DisplayBrightnessPercent = 75;
pLRC->CorrespondingLuxValue = 10000;
.
.
.
```

Here is an example for `SENSOR_DATA_TYPE_ORIENTATION_ROTATION_MATRIX:`

```
struct _rotation_matrix {
          float               fM11; // matrix[1][1]
          float               fM12; // matrix[1][2]
          float               fM13; // matrix[1][3]
          float               fM21; // matrix[2][1]
          float               fM22; // matrix[2][2]
          float               fM23; // matrix[2][3]
          float               fM31; // matrix[3][1]
          float               fM32; // matrix[3][2]
          float               fM33; // matrix[3][3]
} RotationMatrix;

FLAT_VARIANT_UNION fvu;
fvu.vecVal.caubElems = (unsigned long)sizeof(RotationMatrix);
struct _rotation_matrix *pRM = (struct _rotation_matrix *)fvu.vecVal.paubElems;
pRM->fM11 = 1.0;
pRM->fM12 = 0.0;
pRM->fM13 = 0.0;
.
.
.
```

Here is an example for `SENSOR_DATA_TYPE_ORIENTATION_QUATERNION:`

```
struct _quaternion {
          float               fW; // real axis component
          float               fX; // imaginary “i” axis component
          float               fY; // imaginary “j” axis component
          float               fZ; // imaginary “k” axis component
} Quaternion;

FLAT_VARIANT_UNION fvu;
fvu.vecVal.caubElems = (unsigned long)sizeof(Quaternion);
struct _quaternion *pQ = (struct _quaternion *)fvu.vecVal.paubElems;
pQ->fW = 1.0;
pQ->fX = 0.0;
pQ->fY = 0.0;
pQ->fZ = 0.0;
.
.
.
```

There are two ways to identify a Property or Data Field:

1. Using simply the PROPERTYKEY field;
2. Using a more expressive GUID_OR_PROPERTYKEY field.

For convenience, the PROPERTYKEY and the VARIANT value can be combined together as follows:

```
typedef struct _HID_SENSOR_PROPERTYKEY_VALUE_PAIR {
          PROPERTYKEY         key;
          FLAT_VARIANT_UNION  fvu;
} HID_SENSOR_PROPERTYKEY_VALUE_PAIR;
```

Or the GUID_OR_PROPERTYKEY and the VARIANT value can be combined together as follows:

```
typedef struct _HID_SENSOR_GorPK_VALUE_PAIR {
          GUID_OR_PROPERTYKEY GorPK;
          FLAT_VARIANT_UNION  fvu;
} HID_SENSOR_GorPK_VALUE_PAIR;
```

In order to disambiguate which of the data types in the FLAT_VARIANT_UNION are actually being used:

- The platform driver must be able to deduce the correct type “expected” for the specified PROPERTYKEY;
- The GUID_OR_PROPERTYKEY field has included type information that the platform driver can use to extract the data and map it to the “expected” data type for the specified PROPERTYKEY.

The GUID_OR_PROPERTYKEY field is defined as follows:

```
enum GorPK_KIND {
          category_guid       = 1,
          type_guid,
          event_propertykey,
          property_propertykey,
          datafield_propertykey
};

typedef struct _GUID_OR_PROPERTYKEY {
          GorPK_KIND          Kind;
          unsigned char       TopLevelCollection;
          unsigned char       ReportId;
          unsigned char       PackingPosition;
          FIRMWARE_VARTYPE    Vartype;
          unsigned char       Modifier;
          unsigned char       UnitOfMeasure;
          unsigned char       UnitsExponent;
          unsigned char       ReportSize;
          unsigned char       ReportCount;
          GUID                fmtid;
          unsigned long       pid;
} GUID_OR_PROPERTYKEY;
```

Many of the members of the GUID_OR_PROPERTYKEY define attributes of the Report Item that would otherwise have to be specified in (and parsed out of) the HID Report Descriptor:

- TopLevelCollection: which HID top level collection the Item is a part of;
- ReportId: which HID report the Item is a part of;
- PackingPosition: the sequence order of the Item in a Report when more than one Item is packed into a single Report;
- Vartype: the data type of the Item as intended by the sensor firmware;
- Modifier: the associated `HID_USAGE_SENSOR_DATA_MOD_`*`xxx`*;
- UnitOfMeasure: the associated `HID_USAGE_SENSOR_UNITS_`*`xxx`*;
- UnitExponent: the associated `HID_UNIT_EXPONENT`;
- ReportSize: the associated `HID_REPORT_SIZE`;
- ReportCount: the associated `HID_REPORT_COUNT`;
- fmtid: the associated GUID of the Sensor Category, Sensor Type, or Sensor Event; or the GUID portion of the associated PROPERTYKEY;
- pid: the PID portion of the associated PROPERTYKEY (for Category, Type, and Event – enter zero for the PID).

Following is a Generic Sensor Report Descriptor that illustrates these concepts. If we apply this example, the encapsulated fields would be populated as follows for a Speedometer sensor (HID Usage = `HID_USAGE_SENSOR_TYPE_MOTION_SPEEDOMETER`).

```
// Complete HID report descriptor

const unsigned char gen_report_descriptor[] = {
          HID_USAGE_PAGE_SENSOR,
          HID_USAGE_SENSOR_TYPE_OTHER_GENERIC,
          HID_COLLECTION(Physical),

          //feature reports (xmit/receive)
          HID_USAGE_PAGE_SENSOR,
          HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE,
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_8(5),
          HID_REPORT_SIZE(8),
          HID_REPORT_COUNT(1),
          HID_COLLECTION(Logical),
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_NO_EVENTS,
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS,
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_THRESHOLD_EVENTS,
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_NO_EVENTS_WAKE,
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS_WAKE,
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_THRESHOLD_EVENTS_WAKE,
                    HID_FEATURE(Data_Arr_Abs),
          HID_END_COLLECTION,
          HID_USAGE_SENSOR_PROPERTY_SENSOR_STATUS,
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_32(0xFF,0xFF,0xFF,0xFF),
          HID_REPORT_SIZE(32),
          HID_REPORT_COUNT(1),
          HID_FEATURE(Data_Var_Abs), // up to VT_UI4 worth of status info
          HID_USAGE_SENSOR_DATA_GENERIC_CATEGORY_GUID,
          HID_REPORT_SIZE(8),
```

```
        HID_REPORT_COUNT(16),
        HID_FEATURE(Const_Arr_Abs), //= Sensor Category Motion
        HID_USAGE_SENSOR_DATA_GENERIC_TYPE_GUID,
        HID_REPORT_SIZE(8),
        HID_REPORT_COUNT(16),
        HID_FEATURE(Const_Arr_Abs), //= Sensor Type Speedometer
        HID_USAGE_SENSOR_PROPERTY_REPORT_INTERVAL,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_32(0xFF,0xFF,0xFF,0xFF),
        HID_REPORT_SIZE(32),
        HID_REPORT_COUNT(1),
        // HID_USAGE_SENSOR_UNITS_MILLISECOND,
        HID_UNIT_EXPONENT(0),
        HID_FEATURE(Data_Var_Abs),
        HID_USAGE_SENSOR_PROPERTY_SENSOR_CONNECTION_TYPE,  // NAry
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_8(2),
        HID_REPORT_SIZE(8),
        HID_REPORT_COUNT(1),
        HID_COLLECTION(Logical),
                HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_INTEGRATED,
                HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_ATTACHED,
                HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_EXTERNAL,
                HID_FEATURE(Const_Arr_Abs),
        HID_END_COLLECTION,
        HID_USAGE_SENSOR_PROPERTY_CHANGE_SENSITIVITY_ABS,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_16(0xFF,0xFF),
        HID_REPORT_SIZE(16),
        HID_REPORT_COUNT(1),
        // HID_USAGE_SENSOR_UNITS_G,
        HID_UNIT_EXPONENT(0x0E), // scale default unit “G” to provide 2 digits past the decimal point
        HID_FEATURE(Data_Var_Abs),

        //input reports (transmit)
        HID_USAGE_PAGE_SENSOR,
        HID_USAGE_SENSOR_STATE,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_8(6),
        HID_REPORT_SIZE(8),
        HID_REPORT_COUNT(1),
        HID_COLLECTION(Logical),
                HID_USAGE_SENSOR_STATE_UNKNOWN,
                HID_USAGE_SENSOR_STATE_READY,
                HID_USAGE_SENSOR_STATE_NOT_AVAILABLE,
                HID_USAGE_SENSOR_STATE_NO_DATA,
                HID_USAGE_SENSOR_STATE_INITIALIZING,
                HID_USAGE_SENSOR_STATE_ACCESS_DENIED,
                HID_USAGE_SENSOR_STATE_ERROR,
                HID_INPUT(Const_Arr_Abs),
        HID_END_COLLECTION,
        HID_USAGE_SENSOR_EVENT,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_8(16),
        HID_REPORT_SIZE(8),
        HID_REPORT_COUNT(1),
        HID_COLLECTION(Logical),
                HID_USAGE_SENSOR_EVENT_UNKNOWN,
                HID_USAGE_SENSOR_EVENT_STATE_CHANGED,
                HID_USAGE_SENSOR_EVENT_PROPERTY_CHANGED,
                HID_USAGE_SENSOR_EVENT_DATA_UPDATED,
                HID_USAGE_SENSOR_EVENT_POLL_RESPONSE,
                HID_USAGE_SENSOR_EVENT_CHANGE_SENSITIVITY,
                HID_USAGE_SENSOR_EVENT_MAX_REACHED,
                HID_USAGE_SENSOR_EVENT_MIN_REACHED,
                HID_USAGE_SENSOR_EVENT_HIGH_THRESHOLD_CROSS_UPWARD,
                HID_USAGE_SENSOR_EVENT_HIGH_THRESHOLD_CROSS_DOWNWARD,
                HID_USAGE_SENSOR_EVENT_LOW_THRESHOLD_CROSS_UPWARD,
                HID_USAGE_SENSOR_EVENT_LOW_THRESHOLD_CROSS_DOWNWARD,
                HID_USAGE_SENSOR_EVENT_ZERO_THRESHOLD_CROSS_UPWARD,
                HID_USAGE_SENSOR_EVENT_ZERO_THRESHOLD_CROSS_DOWNWARD,
                HID_USAGE_SENSOR_EVENT_PERIOD_EXCEEDED,
                HID_USAGE_SENSOR_EVENT_FREQUENCY_EXCEEDED,
                HID_USAGE_SENSOR_EVENT_COMPLEX_TRIGGER,
                HID_INPUT(Const_Arr_Abs),
        HID_END_COLLECTION,
        HID_USAGE_SENSOR_DATA_GENERIC_DATAFIELD_PROPERTYKEY,  //datafield
        HID_REPORT_SIZE(8),
        HID_REPORT_COUNT(24),
        HID_INPUT(Const_Arr_Abs), //= Sensor Datafield Speed + value

        HID_END_COLLECTION
};
```

4.2.7.1 Generic Enumerator

In the preceding discussion of Generic Properties and Generic Data Fields, it is assumed that the “identifying information” (PROPERTYKEY or GUID_OR_PROPERTYKEY struct) are grouped “inline” and adjacent to the actual data value inside the Report.

Because these structures are large (a PROPERTYKEY is 20 bytes, the GUID_OR_PROPERTYKEY struct is 30 bytes), this leaves only a modest amount of available space in the Report for the data value itself (leaves 43 bytes when using a PROPERTYKEY, and leaves 33 bytes when using a GUID_OR_PROPERTYKEY struct).

In many cases, this may be acceptable. But this provides a practical limit of packing only a single Item in a Report. And some Items have a native size (narrow character strings, wide character strings, response curve structs, rotation matrix structs) that may no longer fit at all.

To mitigate this, a strategy called the Generic Enumerator is introduced, whereby all the bulky "indentifying information" (specifically GUID_OR_PROPERTYKEY structs) are taken out of their "inline" positions and grouped together in their own dedicated HID top level collection.

The HID Report Descriptor for this new top level collection is as follows:

```
// Complete HID report descriptor

const unsigned char enumerator_report_descriptor[] = {
          HID_USAGE_PAGE_SENSOR,
          HID_USAGE_SENSOR_TYPE_OTHER_GENERIC,
          HID_COLLECTION(Application),
          HID_REPORT_ID(1),

          // Report ID 0x01: ENUMERATOR INPUT report
          // Data Field 1: Enumerator Table Row Index.  Read-Only.
          // On a HID GET INPUT, auto-post-increments (use HID SET FEATURE to set the start row).
          HID_USAGE_SENSOR_DATA_ENUMERATOR_TABLE_ROW_INDEX,
          HID_LOGICAL_MIN_16(0x00, 0x00),
          HID_LOGICAL_MAX_16(0xff, 0xff),
          HID_REPORT_SIZE(16),
          HID_REPORT_COUNT(1),
          HID_INPUT(Const_Var_Abs),
          // Data Field 2: Enumerator Table Row Data.  Read-Only.
          // Contents defined by GUID_OR_PROPERTYKEY struct (which is 30 bytes long).
          HID_USAGE_SENSOR_DATA_GENERIC_GUID_OR_PROPERTYKEY,
          HID_LOGICAL_MIN_8(0x00),
          HID_LOGICAL_MAX_8(0xff),
          HID_REPORT_SIZE(8),
          HID_REPORT_COUNT(30),
          HID_INPUT(Const_Arr_Abs),

          // Report ID 0x01: ENUMERATOR FEATURE report
          // Property 1: Enumerator Table Row Index.  Read/Write.
          // On a HID SET FEATURE, used to set the start row.
          HID_USAGE_SENSOR_PROPERTY_ENUMERATOR_TABLE_ROW_INDEX,
          HID_LOGICAL_MIN_16(0x00, 0x00),
          HID_LOGICAL_MAX_16(0xff, 0xff),
          HID_REPORT_SIZE(16),
          HID_REPORT_COUNT(1),
          HID_FEATURE(Data_Var_Abs),
          // Property 2: Enumerator Table Row Count.  Read-Only.
          // On a HID GET FEATURE, used to get the total count of Table Rows.
          HID_USAGE_SENSOR_PROPERTY_ENUMERATOR_TABLE_ROW_COUNT,
          HID_LOGICAL_MIN_16(0x00, 0x00),
          HID_LOGICAL_MAX_16(0xff, 0xff),
          HID_REPORT_SIZE(16),
          HID_REPORT_COUNT(1),
          HID_FEATURE(Const_Var_Abs),

          HID_END_COLLECTION
};
```

The Generic Enumerator must be the first top level collection. The Report ID for the Input Report and Feature Report must be 0x01.

The platform driver uses these pre-defined values to "bootstrap" itself by:

- Performing a HID Get Feature request to read the `HID_USAGE_SENSOR_PROPERTY_ENUMERATOR_TABLE_ROW_COUNT`;
- Performing a HID Set Feature request to reset the `HID_USAGE_SENSOR_PROPERTY_ENUMERATOR_TABLE_ROW_INDEX` to zero.
- Performing a loop for `HID_USAGE_SENSOR_PROPERTY_ENUMERATOR_TABLE_ROW_COUNT` times of:

- Performing a HID Get Input Report request to retrieve a single “table row” consisting of the GUID_OR_PROPERTYKEY struct;
- After each HID Get Input Report request, the sensor firmware automatically increments the `HID_USAGE_SENSOR_DATA_ENUMERATOR_TABLE_ROW_INDEX.`

Once this operation is complete, the platform driver will now know:

- The top level collection number (from 2 to *n*) of every sensor represented by the device;
- The Sensor Category and Sensor Type of each sensor;
- The Sensor Events emitted by each sensor;
- All of the Data Fields reported by each sensor; and for each of those Data Fields, its Report ID, packing position, data type, size, unit of measure, and so on;
- All of the Properties supported by each sensor; and for each of those Properties, its Report ID, packing position, data type, size, unit of measure, and so on.

When the platform driver wishes to interpret any Data Field or Property, it has all the information it needs without requiring any of the identifying information to be “inline” in the Reports themselves. This frees up the *entire* packet (other than the single byte for the Report ID) for holding the data values.

Here is an example of a HID Report Descriptor illustrating this concept:

```
// Complete HID report descriptor

const unsigned char enum_sensor_report_descriptor[] = {
          HID_USAGE_PAGE_SENSOR,
          HID_USAGE_SENSOR_TYPE_COLLECTION
          HID_COLLECTION(Application),

          //   Global items
          HID_REPORT_SIZE(8),
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_8(255),

          //   Report ID 0x08: an Input Report with 37 bytes of Data Fields
          HID_REPORT_ID(0x08),
          HID_USAGE_SENSOR_DATA_GENERIC_DATAFIELD,
          HID_REPORT_COUNT(37),
          HID_INPUT(Const_Arr_Abs),

          //   Report ID 0x09: a Feature Report with 63 bytes of Read-Only Properties
          HID_REPORT_ID(0x09),
          HID_USAGE_SENSOR_DATA_GENERIC_PROPERTY,
          HID_REPORT_COUNT(63),
          HID_FEATURE(Const_Arr_Abs),

          //   Report ID 0x0a: a Feature Report with 42 bytes of Read/Write Properties
          HID_REPORT_ID(0x0a),
          HID_USAGE_SENSOR_DATA_GENERIC_PROPERTY,
          HID_REPORT_COUNT(42),
          HID_FEATURE(Data_Arr_Abs),

          HID_END_COLLECTION
};
```

Note from the above that the contents of the Input Reports and Feature Reports are completely opaque. The Generic Enumerator table rows must be consulted to make any sense out of the data values in the Reports. The advantage of this approach is that it allows complete “run time” flexibility in the semantics of the data being transferred.

4.3 Illustrative Sensor Report Descriptors

4.3.1 Biometric: Human Presence

```
// Complete HID report descriptor

//Human Presence
const unsigned char pres_report_descriptor[] = {
          HID_USAGE_PAGE_SENSOR,
          HID_USAGE_SENSOR_TYPE_BIOMETRIC_PRESENCE,
          HID_COLLECTION(Physical),
```

```
        //feature reports (xmit/receive)
        HID_USAGE_PAGE_SENSOR,
        HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_8(5),
        HID_REPORT_SIZE(8),
        HID_REPORT_COUNT(1),
        HID_COLLECTION(Logical),
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_NO_EVENTS,
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS,
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_THRESHOLD_EVENTS,
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_NO_EVENTS_WAKE,
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS_WAKE,
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_THRESHOLD_EVENTS_WAKE,
                HID_FEATURE(Data_Arr_Abs),
        HID_END_COLLECTION,
        HID_USAGE_SENSOR_PROPERTY_SENSOR_STATUS,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_32(0xFF,0xFF,0xFF,0xFF),
        HID_REPORT_SIZE(32),
        HID_REPORT_COUNT(1),
        HID_FEATURE(Data_Var_Abs), // up to VT_UI4 worth of status info
        HID_USAGE_SENSOR_PROPERTY_REPORT_INTERVAL,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_32(0xFF,0xFF,0xFF,0xFF),
        HID_REPORT_SIZE(32),
        HID_REPORT_COUNT(1),
        // HID_USAGE_SENSOR_UNITS_MILLISECOND,
        HID_UNIT_EXPONENT(0),
        HID_FEATURE(Data_Var_Abs),
        HID_USAGE_SENSOR_PROPERTY_SENSOR_CONNECTION_TYPE,  // NAry
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_8(2),
        HID_REPORT_SIZE(8),
        HID_REPORT_COUNT(1),
        HID_COLLECTION(Logical),
                HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_INTEGRATED,
                HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_ATTACHED,
                HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_EXTERNAL,
                HID_FEATURE(Const_Arr_Abs),
        HID_END_COLLECTION,

        //input reports (transmit)
        HID_USAGE_PAGE_SENSOR,
        HID_USAGE_SENSOR_STATE,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_8(6),
        HID_REPORT_SIZE(8),
        HID_REPORT_COUNT(1),
        HID_COLLECTION(Logical),
                HID_USAGE_SENSOR_STATE_UNKNOWN,
                HID_USAGE_SENSOR_STATE_READY,
                HID_USAGE_SENSOR_STATE_NOT_AVAILABLE,
                HID_USAGE_SENSOR_STATE_NO_DATA,
                HID_USAGE_SENSOR_STATE_INITIALIZING,
                HID_USAGE_SENSOR_STATE_ACCESS_DENIED,
                HID_USAGE_SENSOR_STATE_ERROR,
                HID_INPUT(Const_Arr_Abs),
        HID_END_COLLECTION,
        HID_USAGE_SENSOR_EVENT,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_8(16),
        HID_REPORT_SIZE(8),
        HID_REPORT_COUNT(1),
        HID_COLLECTION(Logical),
                HID_USAGE_SENSOR_EVENT_UNKNOWN,
                HID_USAGE_SENSOR_EVENT_STATE_CHANGED,
                HID_USAGE_SENSOR_EVENT_PROPERTY_CHANGED,
                HID_USAGE_SENSOR_EVENT_DATA_UPDATED,
                HID_USAGE_SENSOR_EVENT_POLL_RESPONSE,
                HID_USAGE_SENSOR_EVENT_CHANGE_SENSITIVITY,
                HID_USAGE_SENSOR_EVENT_MAX_REACHED,
                HID_USAGE_SENSOR_EVENT_MIN_REACHED,
                HID_USAGE_SENSOR_EVENT_HIGH_THRESHOLD_CROSS_UPWARD,
                HID_USAGE_SENSOR_EVENT_HIGH_THRESHOLD_CROSS_DOWNWARD,
                HID_USAGE_SENSOR_EVENT_LOW_THRESHOLD_CROSS_UPWARD,
                HID_USAGE_SENSOR_EVENT_LOW_THRESHOLD_CROSS_DOWNWARD,
                HID_USAGE_SENSOR_EVENT_ZERO_THRESHOLD_CROSS_UPWARD,
                HID_USAGE_SENSOR_EVENT_ZERO_THRESHOLD_CROSS_DOWNWARD,
                HID_USAGE_SENSOR_EVENT_PERIOD_EXCEEDED,
                HID_USAGE_SENSOR_EVENT_FREQUENCY_EXCEEDED,
                HID_USAGE_SENSOR_EVENT_COMPLEX_TRIGGER,
                HID_INPUT(Const_Arr_Abs),
        HID_END_COLLECTION,
        HID_USAGE_SENSOR_DATA_BIOMETRIC_HUMAN_PRESENCE,
        HID_LOGICAL_MIN_8(0), // False
        HID_LOGICAL_MAX_8(1), // True
        HID_REPORT_SIZE(8),
        HID_REPORT_COUNT(1),
        HID_INPUT(Const_Var_Abs),

        HID_END_COLLECTION
};
```

4.3.2 Biometric: Human Proximity

```
// For reference: Complete HID report descriptor

const unsigned char prox_report_descriptor[] = {
          HID_USAGE_PAGE_SENSOR,
          HID_USAGE_SENSOR_TYPE_BIOMETRIC_PROXIMITY,
          HID_COLLECTION(Physical),

          //feature reports (xmit/receive)
          HID_USAGE_PAGE_SENSOR,
          HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE,
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_8(5),
          HID_REPORT_SIZE(8),
          HID_REPORT_COUNT(1),
          HID_COLLECTION(Logical),
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_NO_EVENTS,
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS,
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_THRESHOLD_EVENTS,
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_NO_EVENTS_WAKE,
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS_WAKE,
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_THRESHOLD_EVENTS_WAKE,
                    HID_FEATURE(Data_Arr_Abs),
          HID_END_COLLECTION,
          HID_USAGE_SENSOR_PROPERTY_SENSOR_STATUS,
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_32(0xFF,0xFF,0xFF,0xFF),
          HID_REPORT_SIZE(32),
          HID_REPORT_COUNT(1),
          HID_FEATURE(Data_Var_Abs), // up to VT_UI4 worth of status info
          HID_USAGE_SENSOR_PROPERTY_REPORT_INTERVAL,
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_32(0xFF,0xFF,0xFF,0xFF),
          HID_REPORT_SIZE(32),
          HID_REPORT_COUNT(1),
          // HID_USAGE_SENSOR_UNITS_MILLISECOND,
          HID_UNIT_EXPONENT(0),
          HID_FEATURE(Data_Var_Abs),
          HID_USAGE_SENSOR_PROPERTY_SENSOR_CONNECTION_TYPE,  // NAry
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_8(2),
          HID_REPORT_SIZE(8),
          HID_REPORT_COUNT(1),
          HID_COLLECTION(Logical),
                    HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_INTEGRATED,
                    HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_ATTACHED,
                    HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_EXTERNAL,
                    HID_FEATURE(Const_Arr_Abs),
          HID_END_COLLECTION,
          HID_USAGE_SENSOR_PROPERTY_CHANGE_SENSITIVITY_ABS,
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_16(0xFF,0xFF),
          HID_REPORT_SIZE(16),
          HID_REPORT_COUNT(1),
          // HID_USAGE_SENSOR_UNITS_METER,
          HID_UNIT_EXPONENT(0x0D), // scale default unit “meter” to “centimeter” to provide 2 digits past decimal point
          HID_FEATURE(Data_Var_Abs),
          HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_BIOMETRIC_HUMAN_PROXIMITY_RANGE,HID_USAGE_SENSOR_DATA_MOD_MAX),
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_16(0xFF,0xFF),
          HID_REPORT_SIZE(16),
          HID_REPORT_COUNT(1),
          // HID_USAGE_SENSOR_UNITS_METER,
          HID_UNIT_EXPONENT(0x0D), // scale default unit “meter” to “centimeter” to provide 2 digits past decimal point
          HID_FEATURE(Data_Var_Abs),
          HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_BIOMETRIC_HUMAN_PROXIMITY_RANGE,HID_USAGE_SENSOR_DATA_MOD_MIN),
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_16(0xFF,0xFF),
          HID_REPORT_SIZE(16),
          HID_REPORT_COUNT(1),
          // HID_USAGE_SENSOR_UNITS_METER,
          HID_UNIT_EXPONENT(0x0D), // scale default unit “meter” to “centimeter” to provide 2 digits past decimal point
          HID_FEATURE(Data_Var_Abs),

          //input reports (transmit)
          HID_USAGE_PAGE_SENSOR,
          HID_USAGE_SENSOR_STATE,
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_8(6),
          HID_REPORT_SIZE(8),
          HID_REPORT_COUNT(1),
          HID_COLLECTION(Logical),
                    HID_USAGE_SENSOR_STATE_UNKNOWN,
                    HID_USAGE_SENSOR_STATE_READY,
                    HID_USAGE_SENSOR_STATE_NOT_AVAILABLE,
                    HID_USAGE_SENSOR_STATE_NO_DATA,
                    HID_USAGE_SENSOR_STATE_INITIALIZING,
                    HID_USAGE_SENSOR_STATE_ACCESS_DENIED,
                    HID_USAGE_SENSOR_STATE_ERROR,
                    HID_INPUT(Const_Arr_Abs),
          HID_END_COLLECTION,
          HID_USAGE_SENSOR_EVENT,
          HID_LOGICAL_MIN_8(0),
```

```
        HID_LOGICAL_MAX_8(16),
        HID_REPORT_SIZE(8),
        HID_REPORT_COUNT(1),
        HID_COLLECTION(Logical),
                HID_USAGE_SENSOR_EVENT_UNKNOWN,
                HID_USAGE_SENSOR_EVENT_STATE_CHANGED,
                HID_USAGE_SENSOR_EVENT_PROPERTY_CHANGED,
                HID_USAGE_SENSOR_EVENT_DATA_UPDATED,
                HID_USAGE_SENSOR_EVENT_POLL_RESPONSE,
                HID_USAGE_SENSOR_EVENT_CHANGE_SENSITIVITY,
                HID_USAGE_SENSOR_EVENT_MAX_REACHED,
                HID_USAGE_SENSOR_EVENT_MIN_REACHED,
                HID_USAGE_SENSOR_EVENT_HIGH_THRESHOLD_CROSS_UPWARD,
                HID_USAGE_SENSOR_EVENT_HIGH_THRESHOLD_CROSS_DOWNWARD,
                HID_USAGE_SENSOR_EVENT_LOW_THRESHOLD_CROSS_UPWARD,
                HID_USAGE_SENSOR_EVENT_LOW_THRESHOLD_CROSS_DOWNWARD,
                HID_USAGE_SENSOR_EVENT_ZERO_THRESHOLD_CROSS_UPWARD,
                HID_USAGE_SENSOR_EVENT_ZERO_THRESHOLD_CROSS_DOWNWARD,
                HID_USAGE_SENSOR_EVENT_PERIOD_EXCEEDED,
                HID_USAGE_SENSOR_EVENT_FREQUENCY_EXCEEDED,
                HID_USAGE_SENSOR_EVENT_COMPLEX_TRIGGER,
                HID_INPUT(Const_Arr_Abs),
        HID_END_COLLECTION,
        HID_USAGE_SENSOR_DATA_BIOMETRIC_HUMAN_PROXIMITY_OUT_OF_RANGE,
        HID_LOGICAL_MIN_8(0), // False
        HID_LOGICAL_MAX_8(1), // True
        HID_REPORT_SIZE(8),
        HID_REPORT_COUNT(1),
        HID_INPUT(Const_Var_Abs),
        HID_USAGE_SENSOR_DATA_BIOMETRIC_HUMAN_PROXIMITY_RANGE,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_16(0xFF,0xFF),
        HID_REPORT_SIZE(16),
        HID_REPORT_COUNT(1),
        // HID_USAGE_SENSOR_UNITS_METER,
        HID_UNIT_EXPONENT(0x0D), // scale default unit “meter” to “centimeter” to provide 2 digits past decimal point
        HID_INPUT(Const_Var_Abs),

        HID_END_COLLECTION
};
```

4.3.3 Biometric: Touch

This example describes a sensor that detects human touch for biometric purposes. It is not to be confused with a touch-screen that uses touch for graphical navigation control.

```
// For reference: Complete HID report descriptor

//Touch sensor
const unsigned char biotouch_report_descriptor[] = {
        HID_USAGE_PAGE_SENSOR,
        HID_USAGE_SENSOR_TYPE_BIOMETRIC_TOUCH,
        HID_COLLECTION(Physical),

        //feature reports (xmit/receive)
        HID_USAGE_PAGE_SENSOR,
        HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_8(5),
        HID_REPORT_SIZE(8),
        HID_REPORT_COUNT(1),
        HID_COLLECTION(Logical),
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_NO_EVENTS,
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS,
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_THRESHOLD_EVENTS,
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_NO_EVENTS_WAKE,
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS_WAKE,
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_THRESHOLD_EVENTS_WAKE,
                HID_FEATURE(Data_Arr_Abs),
        HID_END_COLLECTION,
        HID_USAGE_SENSOR_PROPERTY_SENSOR_STATUS,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_32(0xFF,0xFF,0xFF,0xFF),
        HID_REPORT_SIZE(32),
        HID_REPORT_COUNT(1),
        HID_FEATURE(Data_Var_Abs), // up to VT_UI4 worth of status info
        HID_USAGE_SENSOR_PROPERTY_REPORT_INTERVAL,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_32(0xFF,0xFF,0xFF,0xFF),
        HID_REPORT_SIZE(32),
        HID_REPORT_COUNT(1),
        // HID_USAGE_SENSOR_UNITS_MILLISECOND,
        HID_UNIT_EXPONENT(0),
        HID_FEATURE(Data_Var_Abs),
        HID_USAGE_SENSOR_PROPERTY_SENSOR_CONNECTION_TYPE,  // NAry
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_8(2),
        HID_REPORT_SIZE(8),
        HID_REPORT_COUNT(1),
        HID_COLLECTION(Logical),
```

```
                    HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_INTEGRATED,
                    HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_ATTACHED,
                    HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_EXTERNAL,
                    HID_FEATURE(Const_Arr_Abs),
          HID_END_COLLECTION,
          HID_USAGE_SENSOR_PROPERTY_CHANGE_SENSITIVITY_ABS,
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_16(0xFF,0xFF),
          HID_REPORT_SIZE(16),
          HID_REPORT_COUNT(1),
          HID_USAGE_SENSOR_UNITS_NOT_SPECIFIED,
          HID_UNIT_EXPONENT(0x0D), // scale default unit to provide 2 digits past decimal point
          HID_FEATURE(Data_Var_Abs),

          //input reports (transmit)
          HID_USAGE_PAGE_SENSOR,
          HID_USAGE_SENSOR_STATE,
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_8(6),
          HID_REPORT_SIZE(8),
          HID_REPORT_COUNT(1),
          HID_COLLECTION(Logical),
                    HID_USAGE_SENSOR_STATE_UNKNOWN,
                    HID_USAGE_SENSOR_STATE_READY,
                    HID_USAGE_SENSOR_STATE_NOT_AVAILABLE,
                    HID_USAGE_SENSOR_STATE_NO_DATA,
                    HID_USAGE_SENSOR_STATE_INITIALIZING,
                    HID_USAGE_SENSOR_STATE_ACCESS_DENIED,
                    HID_USAGE_SENSOR_STATE_ERROR,
                    HID_INPUT(Const_Arr_Abs),
          HID_END_COLLECTION,
          HID_USAGE_SENSOR_EVENT,
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_8(16),
          HID_REPORT_SIZE(8),
          HID_REPORT_COUNT(1),
          HID_COLLECTION(Logical),
                    HID_USAGE_SENSOR_EVENT_UNKNOWN,
                    HID_USAGE_SENSOR_EVENT_STATE_CHANGED,
                    HID_USAGE_SENSOR_EVENT_PROPERTY_CHANGED,
                    HID_USAGE_SENSOR_EVENT_DATA_UPDATED,
                    HID_USAGE_SENSOR_EVENT_POLL_RESPONSE,
                    HID_USAGE_SENSOR_EVENT_CHANGE_SENSITIVITY,
                    HID_USAGE_SENSOR_EVENT_MAX_REACHED,
                    HID_USAGE_SENSOR_EVENT_MIN_REACHED,
                    HID_USAGE_SENSOR_EVENT_HIGH_THRESHOLD_CROSS_UPWARD,
                    HID_USAGE_SENSOR_EVENT_HIGH_THRESHOLD_CROSS_DOWNWARD,
                    HID_USAGE_SENSOR_EVENT_LOW_THRESHOLD_CROSS_UPWARD,
                    HID_USAGE_SENSOR_EVENT_LOW_THRESHOLD_CROSS_DOWNWARD,
                    HID_USAGE_SENSOR_EVENT_ZERO_THRESHOLD_CROSS_UPWARD,
                    HID_USAGE_SENSOR_EVENT_ZERO_THRESHOLD_CROSS_DOWNWARD,
                    HID_USAGE_SENSOR_EVENT_PERIOD_EXCEEDED,
                    HID_USAGE_SENSOR_EVENT_FREQUENCY_EXCEEDED,
                    HID_USAGE_SENSOR_EVENT_COMPLEX_TRIGGER,
                    HID_INPUT(Const_Arr_Abs),
          HID_END_COLLECTION,
          HID_USAGE_SENSOR_DATA_BIOMETRIC_HUMAN_TOUCH_STATE,
          HID_LOGICAL_MIN_8(0), // False
          HID_LOGICAL_MAX_8(1), // True
          HID_REPORT_SIZE(8),
          HID_REPORT_COUNT(1),
          HID_INPUT(Const_Var_Abs),

          HID_END_COLLECTION
};
```

4.3.4 Electrical: Current

```
// Complete HID report descriptor

const unsigned char amp_report_descriptor[] = {
          HID_USAGE_PAGE_SENSOR,
          HID_USAGE_SENSOR_TYPE_ELECTRICAL_CURRENT,
          HID_COLLECTION(Physical),

          //feature reports (xmit/receive)
          HID_USAGE_PAGE_SENSOR,
          HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE,
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_8(5),
          HID_REPORT_SIZE(8),
          HID_REPORT_COUNT(1),
          HID_COLLECTION(Logical),
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_NO_EVENTS,
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS,
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_THRESHOLD_EVENTS,
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_NO_EVENTS_WAKE,
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS_WAKE,
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_THRESHOLD_EVENTS_WAKE,
                    HID_FEATURE(Data_Arr_Abs),
          HID_END_COLLECTION,
          HID_USAGE_SENSOR_PROPERTY_SENSOR_STATUS,
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_32(0xFF,0xFF,0xFF,0xFF),
```

```
HID_REPORT_SIZE(32),
HID_REPORT_COUNT(1),
HID_FEATURE(Data_Var_Abs), // up to VT_UI4 worth of status info
HID_USAGE_SENSOR_PROPERTY_REPORT_INTERVAL,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_32(0xFF,0xFF,0xFF,0xFF),
HID_REPORT_SIZE(32),
HID_REPORT_COUNT(1),
// HID_USAGE_SENSOR_UNITS_MILLISECOND,
HID_UNIT_EXPONENT(0),
HID_FEATURE(Data_Var_Abs),
HID_USAGE_SENSOR_PROPERTY_SENSOR_CONNECTION_TYPE,  // NAry
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_8(2),
HID_REPORT_SIZE(8),
HID_REPORT_COUNT(1),
HID_COLLECTION(Logical),
          HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_INTEGRATED,
          HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_ATTACHED,
          HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_EXTERNAL,
          HID_FEATURE(Const_Arr_Abs),
HID_END_COLLECTION,
HID_USAGE_SENSOR_PROPERTY_CHANGE_SENSITIVITY_ABS,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_16(0xFF,0xFF),
HID_REPORT_SIZE(16),
HID_REPORT_COUNT(1),
HID_USAGE_SENSOR_UNITS_AMPERE,
HID_UNIT_EXPONENT(0x0E), // scale default unit to provide 2 digits past the decimal point
HID_FEATURE(Data_Var_Abs),
HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_ELECTRICAL_CURRENT,HID_USAGE_SENSOR_DATA_MOD_MAX),
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_16(0xFF,0xFF),
HID_REPORT_SIZE(16),
HID_REPORT_COUNT(1),
HID_USAGE_SENSOR_UNITS_AMPERE,
HID_UNIT_EXPONENT(0x0E), // scale default unit to provide 2 digits past the decimal point
HID_FEATURE(Data_Var_Abs),
HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_ELECTRICAL_CURRENT,HID_USAGE_SENSOR_DATA_MOD_MIN),
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_16(0xFF,0xFF),
HID_REPORT_SIZE(16),
HID_REPORT_COUNT(1),
HID_USAGE_SENSOR_UNITS_AMPERE,
HID_UNIT_EXPONENT(0x0E), // scale default unit to provide 2 digits past the decimal point
HID_FEATURE(Data_Var_Abs),

//input reports (transmit)
HID_USAGE_PAGE_SENSOR,
HID_USAGE_SENSOR_STATE,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_8(6),
HID_REPORT_SIZE(8),
HID_REPORT_COUNT(1),
HID_COLLECTION(Logical),
          HID_USAGE_SENSOR_STATE_UNKNOWN,
          HID_USAGE_SENSOR_STATE_READY,
          HID_USAGE_SENSOR_STATE_NOT_AVAILABLE,
          HID_USAGE_SENSOR_STATE_NO_DATA,
          HID_USAGE_SENSOR_STATE_INITIALIZING,
          HID_USAGE_SENSOR_STATE_ACCESS_DENIED,
          HID_USAGE_SENSOR_STATE_ERROR,
          HID_INPUT(Const_Arr_Abs),
HID_END_COLLECTION,
HID_USAGE_SENSOR_EVENT,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_8(16),
HID_REPORT_SIZE(8),
HID_REPORT_COUNT(1),
HID_COLLECTION(Logical),
          HID_USAGE_SENSOR_EVENT_UNKNOWN,
          HID_USAGE_SENSOR_EVENT_STATE_CHANGED,
          HID_USAGE_SENSOR_EVENT_PROPERTY_CHANGED,
          HID_USAGE_SENSOR_EVENT_DATA_UPDATED,
          HID_USAGE_SENSOR_EVENT_POLL_RESPONSE,
          HID_USAGE_SENSOR_EVENT_CHANGE_SENSITIVITY,
          HID_USAGE_SENSOR_EVENT_MAX_REACHED,
          HID_USAGE_SENSOR_EVENT_MIN_REACHED,
          HID_USAGE_SENSOR_EVENT_HIGH_THRESHOLD_CROSS_UPWARD,
          HID_USAGE_SENSOR_EVENT_HIGH_THRESHOLD_CROSS_DOWNWARD,
          HID_USAGE_SENSOR_EVENT_LOW_THRESHOLD_CROSS_UPWARD,
          HID_USAGE_SENSOR_EVENT_LOW_THRESHOLD_CROSS_DOWNWARD,
          HID_USAGE_SENSOR_EVENT_ZERO_THRESHOLD_CROSS_UPWARD,
          HID_USAGE_SENSOR_EVENT_ZERO_THRESHOLD_CROSS_DOWNWARD,
          HID_USAGE_SENSOR_EVENT_PERIOD_EXCEEDED,
          HID_USAGE_SENSOR_EVENT_FREQUENCY_EXCEEDED,
          HID_USAGE_SENSOR_EVENT_COMPLEX_TRIGGER,
          HID_INPUT(Const_Arr_Abs),
HID_END_COLLECTION,
HID_USAGE_SENSOR_DATA_ELECTRICAL_CURRENT,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_16(0xFF,0xFF),
HID_REPORT_SIZE(16),
HID_REPORT_COUNT(1),
HID_USAGE_SENSOR_UNITS_AMPERE,
```

```
        HID_UNIT_EXPONENT(0x0E), // scale default unit to provide 2 digits past the decimal point
        HID_INPUT(Const_Var_Abs),

        HID_END_COLLECTION
};
```

4.3.5 Electrical: Power

```
// Complete HID report descriptor

const unsigned char watt_report_descriptor[] = {
        HID_USAGE_PAGE_SENSOR,
        HID_USAGE_SENSOR_TYPE_ELECTRICAL_POWER,
        HID_COLLECTION(Physical),

        //feature reports (xmit/receive)
        HID_USAGE_PAGE_SENSOR,
        HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_8(5),
        HID_REPORT_SIZE(8),
        HID_REPORT_COUNT(1),
        HID_COLLECTION(Logical),
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_NO_EVENTS,
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS,
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_THRESHOLD_EVENTS,
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_NO_EVENTS_WAKE,
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS_WAKE,
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_THRESHOLD_EVENTS_WAKE,
                HID_FEATURE(Data_Arr_Abs),
        HID_END_COLLECTION,
        HID_USAGE_SENSOR_PROPERTY_SENSOR_STATUS,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_32(0xFF,0xFF,0xFF,0xFF),
        HID_REPORT_SIZE(32),
        HID_REPORT_COUNT(1),
        HID_FEATURE(Data_Var_Abs), // up to VT_UI4 worth of status info
        HID_USAGE_SENSOR_PROPERTY_REPORT_INTERVAL,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_32(0xFF,0xFF,0xFF,0xFF),
        HID_REPORT_SIZE(32),
        HID_REPORT_COUNT(1),
        // HID_USAGE_SENSOR_UNITS_MILLISECOND,
        HID_UNIT_EXPONENT(0),
        HID_FEATURE(Data_Var_Abs),
        HID_USAGE_SENSOR_PROPERTY_SENSOR_CONNECTION_TYPE,  // NAry
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_8(2),
        HID_REPORT_SIZE(8),
        HID_REPORT_COUNT(1),
        HID_COLLECTION(Logical),
                HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_INTEGRATED,
                HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_ATTACHED,
                HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_EXTERNAL,
                HID_FEATURE(Const_Arr_Abs),
        HID_END_COLLECTION,
        HID_USAGE_SENSOR_PROPERTY_CHANGE_SENSITIVITY_ABS,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_16(0xFF,0xFF),
        HID_REPORT_SIZE(16),
        HID_REPORT_COUNT(1),
        HID_USAGE_SENSOR_UNITS_WATT,
        HID_UNIT_EXPONENT(0x0E), // scale default unit to provide 2 digits past the decimal point
        HID_FEATURE(Data_Var_Abs),
        HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_ELECTRICAL_POWER,HID_USAGE_SENSOR_DATA_MOD_MAX),
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_16(0xFF,0xFF),
        HID_REPORT_SIZE(16),
        HID_REPORT_COUNT(1),
        HID_USAGE_SENSOR_UNITS_WATT,
        HID_UNIT_EXPONENT(0x0E), // scale default unit to provide 2 digits past the decimal point
        HID_FEATURE(Data_Var_Abs),
        HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_ELECTRICAL_POWER,HID_USAGE_SENSOR_DATA_MOD_MIN),
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_16(0xFF,0xFF),
        HID_REPORT_SIZE(16),
        HID_REPORT_COUNT(1),
        HID_USAGE_SENSOR_UNITS_WATT,
        HID_UNIT_EXPONENT(0x0E), // scale default unit to provide 2 digits past the decimal point
        HID_FEATURE(Data_Var_Abs),

        //input reports (transmit)
        HID_USAGE_PAGE_SENSOR,
        HID_USAGE_SENSOR_STATE,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_8(6),
        HID_REPORT_SIZE(8),
        HID_REPORT_COUNT(1),
        HID_COLLECTION(Logical),
                HID_USAGE_SENSOR_STATE_UNKNOWN,
                HID_USAGE_SENSOR_STATE_READY,
                HID_USAGE_SENSOR_STATE_NOT_AVAILABLE,
                HID_USAGE_SENSOR_STATE_NO_DATA,
                HID_USAGE_SENSOR_STATE_INITIALIZING,
```

```
                    HID_USAGE_SENSOR_STATE_ACCESS_DENIED,
                    HID_USAGE_SENSOR_STATE_ERROR,
                    HID_INPUT(Const_Arr_Abs),
          HID_END_COLLECTION,
          HID_USAGE_SENSOR_EVENT,
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_8(16),
          HID_REPORT_SIZE(8),
          HID_REPORT_COUNT(1),
          HID_COLLECTION(Logical),
                    HID_USAGE_SENSOR_EVENT_UNKNOWN,
                    HID_USAGE_SENSOR_EVENT_STATE_CHANGED,
                    HID_USAGE_SENSOR_EVENT_PROPERTY_CHANGED,
                    HID_USAGE_SENSOR_EVENT_DATA_UPDATED,
                    HID_USAGE_SENSOR_EVENT_POLL_RESPONSE,
                    HID_USAGE_SENSOR_EVENT_CHANGE_SENSITIVITY,
                    HID_USAGE_SENSOR_EVENT_MAX_REACHED,
                    HID_USAGE_SENSOR_EVENT_MIN_REACHED,
                    HID_USAGE_SENSOR_EVENT_HIGH_THRESHOLD_CROSS_UPWARD,
                    HID_USAGE_SENSOR_EVENT_HIGH_THRESHOLD_CROSS_DOWNWARD,
                    HID_USAGE_SENSOR_EVENT_LOW_THRESHOLD_CROSS_UPWARD,
                    HID_USAGE_SENSOR_EVENT_LOW_THRESHOLD_CROSS_DOWNWARD,
                    HID_USAGE_SENSOR_EVENT_ZERO_THRESHOLD_CROSS_UPWARD,
                    HID_USAGE_SENSOR_EVENT_ZERO_THRESHOLD_CROSS_DOWNWARD,
                    HID_USAGE_SENSOR_EVENT_PERIOD_EXCEEDED,
                    HID_USAGE_SENSOR_EVENT_FREQUENCY_EXCEEDED,
                    HID_USAGE_SENSOR_EVENT_COMPLEX_TRIGGER,
                    HID_INPUT(Const_Arr_Abs),
          HID_END_COLLECTION,
          HID_USAGE_SENSOR_DATA_ELECTRICAL_POWER,
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_16(0xFF,0xFF),
          HID_REPORT_SIZE(16),
          HID_REPORT_COUNT(1),
          HID_USAGE_SENSOR_UNITS_WATT,
          HID_UNIT_EXPONENT(0x0E), // scale default unit to provide 2 digits past the decimal point
          HID_INPUT(Const_Var_Abs),

          HID_END_COLLECTION
};
```

4.3.6 Electrical: Voltage

```
// Complete HID report descriptor

const unsigned char volt_report_descriptor[] = {
          HID_USAGE_PAGE_SENSOR,
          HID_USAGE_SENSOR_TYPE_ELECTRICAL_VOLTAGE,
          HID_COLLECTION(Physical),

          //feature reports (xmit/receive)
          HID_USAGE_PAGE_SENSOR,
          HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE,
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_8(5),
          HID_REPORT_SIZE(8),
          HID_REPORT_COUNT(1),
          HID_COLLECTION(Logical),
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_NO_EVENTS,
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS,
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_THRESHOLD_EVENTS,
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_NO_EVENTS_WAKE,
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS_WAKE,
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_THRESHOLD_EVENTS_WAKE,
                    HID_FEATURE(Data_Arr_Abs),
          HID_END_COLLECTION,
          HID_USAGE_SENSOR_PROPERTY_SENSOR_STATUS,
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_32(0xFF,0xFF,0xFF,0xFF),
          HID_REPORT_SIZE(32),
          HID_REPORT_COUNT(1),
          HID_FEATURE(Data_Var_Abs), // up to VT_UI4 worth of status info
          HID_USAGE_SENSOR_PROPERTY_REPORT_INTERVAL,
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_32(0xFF,0xFF,0xFF,0xFF),
          HID_REPORT_SIZE(32),
          HID_REPORT_COUNT(1),
          // HID_USAGE_SENSOR_UNITS_MILLISECOND,
          HID_UNIT_EXPONENT(0),
          HID_FEATURE(Data_Var_Abs),
          HID_USAGE_SENSOR_PROPERTY_SENSOR_CONNECTION_TYPE,  // NAry
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_8(2),
          HID_REPORT_SIZE(8),
          HID_REPORT_COUNT(1),
          HID_COLLECTION(Logical),
                    HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_INTEGRATED,
                    HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_ATTACHED,
                    HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_EXTERNAL,
                    HID_FEATURE(Const_Arr_Abs),
          HID_END_COLLECTION,
          HID_USAGE_SENSOR_PROPERTY_CHANGE_SENSITIVITY_ABS,
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_16(0xFF,0xFF),
```

```
        HID_REPORT_SIZE(16),
        HID_REPORT_COUNT(1),
        HID_USAGE_SENSOR_UNITS_VOLT,
        HID_UNIT_EXPONENT(0x0E), // scale default unit to provide 2 digits past the decimal point
        HID_FEATURE(Data_Var_Abs),
        HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_ELECTRICAL_VOLTAGE,HID_USAGE_SENSOR_DATA_MOD_MAX),
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_16(0xFF,0xFF),
        HID_REPORT_SIZE(16),
        HID_REPORT_COUNT(1),
        HID_USAGE_SENSOR_UNITS_VOLT,
        HID_UNIT_EXPONENT(0x0E), // scale default unit to provide 2 digits past the decimal point
        HID_FEATURE(Data_Var_Abs),
        HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_ELECTRICAL_VOLTAGE,HID_USAGE_SENSOR_DATA_MOD_MIN),
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_16(0xFF,0xFF),
        HID_REPORT_SIZE(16),
        HID_REPORT_COUNT(1),
        HID_USAGE_SENSOR_UNITS_VOLT,
        HID_UNIT_EXPONENT(0x0E), // scale default unit to provide 2 digits past the decimal point
        HID_FEATURE(Data_Var_Abs),

        //input reports (transmit)
        HID_USAGE_PAGE_SENSOR,
        HID_USAGE_SENSOR_STATE,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_8(6),
        HID_REPORT_SIZE(8),
        HID_REPORT_COUNT(1),
        HID_COLLECTION(Logical),
                HID_USAGE_SENSOR_STATE_UNKNOWN,
                HID_USAGE_SENSOR_STATE_READY,
                HID_USAGE_SENSOR_STATE_NOT_AVAILABLE,
                HID_USAGE_SENSOR_STATE_NO_DATA,
                HID_USAGE_SENSOR_STATE_INITIALIZING,
                HID_USAGE_SENSOR_STATE_ACCESS_DENIED,
                HID_USAGE_SENSOR_STATE_ERROR,
                HID_INPUT(Const_Arr_Abs),
        HID_END_COLLECTION,
        HID_USAGE_SENSOR_EVENT,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_8(16),
        HID_REPORT_SIZE(8),
        HID_REPORT_COUNT(1),
        HID_COLLECTION(Logical),
                HID_USAGE_SENSOR_EVENT_UNKNOWN,
                HID_USAGE_SENSOR_EVENT_STATE_CHANGED,
                HID_USAGE_SENSOR_EVENT_PROPERTY_CHANGED,
                HID_USAGE_SENSOR_EVENT_DATA_UPDATED,
                HID_USAGE_SENSOR_EVENT_POLL_RESPONSE,
                HID_USAGE_SENSOR_EVENT_CHANGE_SENSITIVITY,
                HID_USAGE_SENSOR_EVENT_MAX_REACHED,
                HID_USAGE_SENSOR_EVENT_MIN_REACHED,
                HID_USAGE_SENSOR_EVENT_HIGH_THRESHOLD_CROSS_UPWARD,
                HID_USAGE_SENSOR_EVENT_HIGH_THRESHOLD_CROSS_DOWNWARD,
                HID_USAGE_SENSOR_EVENT_LOW_THRESHOLD_CROSS_UPWARD,
                HID_USAGE_SENSOR_EVENT_LOW_THRESHOLD_CROSS_DOWNWARD,
                HID_USAGE_SENSOR_EVENT_ZERO_THRESHOLD_CROSS_UPWARD,
                HID_USAGE_SENSOR_EVENT_ZERO_THRESHOLD_CROSS_DOWNWARD,
                HID_USAGE_SENSOR_EVENT_PERIOD_EXCEEDED,
                HID_USAGE_SENSOR_EVENT_FREQUENCY_EXCEEDED,
                HID_USAGE_SENSOR_EVENT_COMPLEX_TRIGGER,
                HID_INPUT(Const_Arr_Abs),
        HID_END_COLLECTION,
        HID_USAGE_SENSOR_DATA_ELECTRICAL_VOLTAGE,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_16(0xFF,0xFF),
        HID_REPORT_SIZE(16),
        HID_REPORT_COUNT(1),
        HID_USAGE_SENSOR_UNITS_VOLT,
        HID_UNIT_EXPONENT(0x0E), // scale default unit to provide 2 digits past the decimal point
        HID_INPUT(Const_Var_Abs),

        HID_END_COLLECTION
};
```

4.3.7 Electrical: Potentiometer

```
// Complete HID report descriptor

const unsigned char pot_report_descriptor[] = {
        HID_USAGE_PAGE_SENSOR,
        HID_USAGE_SENSOR_TYPE_ELECTRICAL_POTENTIOMETER,
        HID_COLLECTION(Physical),

        //feature reports (xmit/receive)
        HID_USAGE_PAGE_SENSOR,
        HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_8(5),
        HID_REPORT_SIZE(8),
        HID_REPORT_COUNT(1),
        HID_COLLECTION(Logical),
```

```
          HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_NO_EVENTS,
          HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS,
          HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_THRESHOLD_EVENTS,
          HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_NO_EVENTS_WAKE,
          HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS_WAKE,
          HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_THRESHOLD_EVENTS_WAKE,
          HID_FEATURE(Data_Arr_Abs),
HID_END_COLLECTION,
HID_USAGE_SENSOR_PROPERTY_SENSOR_STATUS,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_32(0xFF,0xFF,0xFF,0xFF),
HID_REPORT_SIZE(32),
HID_REPORT_COUNT(1),
HID_FEATURE(Data_Var_Abs), // up to VT_UI4 worth of status info
HID_USAGE_SENSOR_PROPERTY_REPORT_INTERVAL,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_32(0xFF,0xFF,0xFF,0xFF),
HID_REPORT_SIZE(32),
HID_REPORT_COUNT(1),
// HID_USAGE_SENSOR_UNITS_MILLISECOND,
HID_UNIT_EXPONENT(0),
HID_FEATURE(Data_Var_Abs),
HID_USAGE_SENSOR_PROPERTY_SENSOR_CONNECTION_TYPE,  // NAry
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_8(2),
HID_REPORT_SIZE(8),
HID_REPORT_COUNT(1),
HID_COLLECTION(Logical),
          HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_INTEGRATED,
          HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_ATTACHED,
          HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_EXTERNAL,
          HID_FEATURE(Const_Arr_Abs),
HID_END_COLLECTION,
HID_USAGE_SENSOR_PROPERTY_CHANGE_SENSITIVITY_ABS,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_16(0xFF,0xFF),
HID_REPORT_SIZE(16),
HID_REPORT_COUNT(1),
// HID_USAGE_SENSOR_UNITS_PERCENT,
HID_UNIT_EXPONENT(0x0E), // scale default unit to provide 2 digits past the decimal point
HID_FEATURE(Data_Var_Abs),
HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_ELECTRICAL_PERCENT_OF_RANGE,HID_USAGE_SENSOR_DATA_MOD_MAX),
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_16(0x10,0x27), // 10000 = 0.00 to 100.00 percent with 2 digits past decimal point
HID_REPORT_SIZE(16),
HID_REPORT_COUNT(1),
// HID_USAGE_SENSOR_UNITS_PERCENT,
HID_UNIT_EXPONENT(0x0E), // scale default unit to provide 2 digits past the decimal point
HID_FEATURE(Data_Var_Abs),
HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_ELECTRICAL_PERCENT_OF_RANGE,HID_USAGE_SENSOR_DATA_MOD_MIN),
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_16(0x10,0x27), // 10000 = 0.00 to 100.00 percent with 2 digits past decimal point
HID_REPORT_SIZE(16),
HID_REPORT_COUNT(1),
// HID_USAGE_SENSOR_UNITS_PERCENT,
HID_UNIT_EXPONENT(0x0E), // scale default unit to provide 2 digits past the decimal point
HID_FEATURE(Data_Var_Abs),

//input reports (transmit)
HID_USAGE_PAGE_SENSOR,
HID_USAGE_SENSOR_STATE,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_8(6),
HID_REPORT_SIZE(8),
HID_REPORT_COUNT(1),
HID_COLLECTION(Logical),
          HID_USAGE_SENSOR_STATE_UNKNOWN,
          HID_USAGE_SENSOR_STATE_READY,
          HID_USAGE_SENSOR_STATE_NOT_AVAILABLE,
          HID_USAGE_SENSOR_STATE_NO_DATA,
          HID_USAGE_SENSOR_STATE_INITIALIZING,
          HID_USAGE_SENSOR_STATE_ACCESS_DENIED,
          HID_USAGE_SENSOR_STATE_ERROR,
          HID_INPUT(Const_Arr_Abs),
HID_END_COLLECTION,
HID_USAGE_SENSOR_EVENT,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_8(16),
HID_REPORT_SIZE(8),
HID_REPORT_COUNT(1),
HID_COLLECTION(Logical),
          HID_USAGE_SENSOR_EVENT_UNKNOWN,
          HID_USAGE_SENSOR_EVENT_STATE_CHANGED,
          HID_USAGE_SENSOR_EVENT_PROPERTY_CHANGED,
          HID_USAGE_SENSOR_EVENT_DATA_UPDATED,
          HID_USAGE_SENSOR_EVENT_POLL_RESPONSE,
          HID_USAGE_SENSOR_EVENT_CHANGE_SENSITIVITY,
          HID_USAGE_SENSOR_EVENT_MAX_REACHED,
          HID_USAGE_SENSOR_EVENT_MIN_REACHED,
          HID_USAGE_SENSOR_EVENT_HIGH_THRESHOLD_CROSS_UPWARD,
          HID_USAGE_SENSOR_EVENT_HIGH_THRESHOLD_CROSS_DOWNWARD,
          HID_USAGE_SENSOR_EVENT_LOW_THRESHOLD_CROSS_UPWARD,
          HID_USAGE_SENSOR_EVENT_LOW_THRESHOLD_CROSS_DOWNWARD,
          HID_USAGE_SENSOR_EVENT_ZERO_THRESHOLD_CROSS_UPWARD,
          HID_USAGE_SENSOR_EVENT_ZERO_THRESHOLD_CROSS_DOWNWARD,
```

```
                    HID_USAGE_SENSOR_EVENT_PERIOD_EXCEEDED,
                    HID_USAGE_SENSOR_EVENT_FREQUENCY_EXCEEDED,
                    HID_USAGE_SENSOR_EVENT_COMPLEX_TRIGGER,
                    HID_INPUT(Const_Arr_Abs),
          HID_END_COLLECTION,
          HID_USAGE_SENSOR_DATA_ELECTRICAL_PERCENT_OF_RANGE,
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_16(0x10,0x27), // 10000 = 0.00 to 100.00 percent with 2 digits past decimal point
          HID_REPORT_SIZE(16),
          HID_REPORT_COUNT(1),
          // HID_USAGE_SENSOR_UNITS_PERCENT,
          HID_UNIT_EXPONENT(0x0E), // scale default unit “percent” to provide 2 digits past the decimal point
          HID_INPUT(Const_Var_Abs),

          HID_END_COLLECTION
};
```

4.3.8 Electrical: Frequency

```
// Complete HID report descriptor

const unsigned char hertz_report_descriptor[] = {
          HID_USAGE_PAGE_SENSOR,
          HID_USAGE_SENSOR_TYPE_ELECTRICAL_FREQUENCY,
          HID_COLLECTION(Physical),

          //feature reports (xmit/receive)
          HID_USAGE_PAGE_SENSOR,
          HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE,
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_8(5),
          HID_REPORT_SIZE(8),
          HID_REPORT_COUNT(1),
          HID_COLLECTION(Logical),
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_NO_EVENTS,
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS,
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_THRESHOLD_EVENTS,
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_NO_EVENTS_WAKE,
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS_WAKE,
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_THRESHOLD_EVENTS_WAKE,
                    HID_FEATURE(Data_Arr_Abs),
          HID_END_COLLECTION,
          HID_USAGE_SENSOR_PROPERTY_SENSOR_STATUS,
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_32(0xFF,0xFF,0xFF,0xFF),
          HID_REPORT_SIZE(32),
          HID_REPORT_COUNT(1),
          HID_FEATURE(Data_Var_Abs), // up to VT_UI4 worth of status info
          HID_USAGE_SENSOR_PROPERTY_REPORT_INTERVAL,
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_32(0xFF,0xFF,0xFF,0xFF),
          HID_REPORT_SIZE(32),
          HID_REPORT_COUNT(1),
          // HID_USAGE_SENSOR_UNITS_MILLISECOND,
          HID_UNIT_EXPONENT(0),
          HID_FEATURE(Data_Var_Abs),
          HID_USAGE_SENSOR_PROPERTY_SENSOR_CONNECTION_TYPE,  // NAry
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_8(2),
          HID_REPORT_SIZE(8),
          HID_REPORT_COUNT(1),
          HID_COLLECTION(Logical),
                    HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_INTEGRATED,
                    HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_ATTACHED,
                    HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_EXTERNAL,
                    HID_FEATURE(Const_Arr_Abs),
          HID_END_COLLECTION,
          HID_USAGE_SENSOR_PROPERTY_CHANGE_SENSITIVITY_ABS,
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_16(0xFF,0xFF),
          HID_REPORT_SIZE(16),
          HID_REPORT_COUNT(1),
          HID_USAGE_SENSOR_UNITS_HERTZ,
          HID_UNIT_EXPONENT(0x0E), // scale default unit to provide 2 digits past the decimal point
          HID_FEATURE(Data_Var_Abs),
          HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_ELECTRICAL_FREQUENCY,HID_USAGE_SENSOR_DATA_MOD_MAX),
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_16(0xFF,0xFF),
          HID_REPORT_SIZE(16),
          HID_REPORT_COUNT(1),
          HID_USAGE_SENSOR_UNITS_HERTZ,
          HID_UNIT_EXPONENT(0x0E), // scale default unit to provide 2 digits past the decimal point
          HID_FEATURE(Data_Var_Abs),
          HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_ELECTRICAL_FREQUENCY,HID_USAGE_SENSOR_DATA_MOD_MIN),
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_16(0xFF,0xFF),
          HID_REPORT_SIZE(16),
          HID_REPORT_COUNT(1),
          HID_USAGE_SENSOR_UNITS_HERTZ,
          HID_UNIT_EXPONENT(0x0E), // scale default unit to provide 2 digits past the decimal point
          HID_FEATURE(Data_Var_Abs),

          //input reports (transmit)
          HID_USAGE_PAGE_SENSOR,
```

```
        HID_USAGE_SENSOR_STATE,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_8(6),
        HID_REPORT_SIZE(8),
        HID_REPORT_COUNT(1),
        HID_COLLECTION(Logical),
                HID_USAGE_SENSOR_STATE_UNKNOWN,
                HID_USAGE_SENSOR_STATE_READY,
                HID_USAGE_SENSOR_STATE_NOT_AVAILABLE,
                HID_USAGE_SENSOR_STATE_NO_DATA,
                HID_USAGE_SENSOR_STATE_INITIALIZING,
                HID_USAGE_SENSOR_STATE_ACCESS_DENIED,
                HID_USAGE_SENSOR_STATE_ERROR,
                HID_INPUT(Const_Arr_Abs),
        HID_END_COLLECTION,
        HID_USAGE_SENSOR_EVENT,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_8(16),
        HID_REPORT_SIZE(8),
        HID_REPORT_COUNT(1),
        HID_COLLECTION(Logical),
                HID_USAGE_SENSOR_EVENT_UNKNOWN,
                HID_USAGE_SENSOR_EVENT_STATE_CHANGED,
                HID_USAGE_SENSOR_EVENT_PROPERTY_CHANGED,
                HID_USAGE_SENSOR_EVENT_DATA_UPDATED,
                HID_USAGE_SENSOR_EVENT_POLL_RESPONSE,
                HID_USAGE_SENSOR_EVENT_CHANGE_SENSITIVITY,
                HID_USAGE_SENSOR_EVENT_MAX_REACHED,
                HID_USAGE_SENSOR_EVENT_MIN_REACHED,
                HID_USAGE_SENSOR_EVENT_HIGH_THRESHOLD_CROSS_UPWARD,
                HID_USAGE_SENSOR_EVENT_HIGH_THRESHOLD_CROSS_DOWNWARD,
                HID_USAGE_SENSOR_EVENT_LOW_THRESHOLD_CROSS_UPWARD,
                HID_USAGE_SENSOR_EVENT_LOW_THRESHOLD_CROSS_DOWNWARD,
                HID_USAGE_SENSOR_EVENT_ZERO_THRESHOLD_CROSS_UPWARD,
                HID_USAGE_SENSOR_EVENT_ZERO_THRESHOLD_CROSS_DOWNWARD,
                HID_USAGE_SENSOR_EVENT_PERIOD_EXCEEDED,
                HID_USAGE_SENSOR_EVENT_FREQUENCY_EXCEEDED,
                HID_USAGE_SENSOR_EVENT_COMPLEX_TRIGGER,
                HID_INPUT(Const_Arr_Abs),
        HID_END_COLLECTION,
        HID_USAGE_SENSOR_DATA_ELECTRICAL_FREQUENCY,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_16(0xFF,0xFF),
        HID_REPORT_SIZE(16),
        HID_REPORT_COUNT(1),
        HID_USAGE_SENSOR_UNITS_HERTZ,
        HID_UNIT_EXPONENT(0x0E), // scale default unit to provide 2 digits past the decimal point
        HID_INPUT(Const_Var_Abs),

        HID_END_COLLECTION
};
```

4.3.9 Environmental: Atmospheric Pressure

```
// For reference: Complete HID report descriptor

const unsigned char bar_report_descriptor[] = {
        HID_USAGE_PAGE_SENSOR,
        HID_USAGE_SENSOR_TYPE_ENVIRONMENTAL_ATMOSPHERIC_PRESSURE,
        HID_COLLECTION(Physical),

        //feature reports (xmit/receive)
        HID_USAGE_PAGE_SENSOR,
        HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_8(5),
        HID_REPORT_SIZE(8),
        HID_REPORT_COUNT(1),
        HID_COLLECTION(Logical),
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_NO_EVENTS,
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS,
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_THRESHOLD_EVENTS,
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_NO_EVENTS_WAKE,
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS_WAKE,
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_THRESHOLD_EVENTS_WAKE,
                HID_FEATURE(Data_Arr_Abs),
        HID_END_COLLECTION,
        HID_USAGE_SENSOR_PROPERTY_SENSOR_STATUS,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_32(0xFF,0xFF,0xFF,0xFF),
        HID_REPORT_SIZE(32),
        HID_REPORT_COUNT(1),
        HID_FEATURE(Data_Var_Abs), // up to VT_UI4 worth of status info
        HID_USAGE_SENSOR_PROPERTY_REPORT_INTERVAL,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_32(0xFF,0xFF,0xFF,0xFF),
        HID_REPORT_SIZE(32),
        HID_REPORT_COUNT(1),
        // HID_USAGE_SENSOR_UNITS_MILLISECOND,
        HID_UNIT_EXPONENT(0),
        HID_FEATURE(Data_Var_Abs),
        HID_USAGE_SENSOR_PROPERTY_SENSOR_CONNECTION_TYPE,  // NAry
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_8(2),
```

```
        HID_REPORT_SIZE(8),
        HID_REPORT_COUNT(1),
        HID_COLLECTION(Logical),
                HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_INTEGRATED,
                HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_ATTACHED,
                HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_EXTERNAL,
                HID_FEATURE(Const_Arr_Abs),
        HID_END_COLLECTION,
        HID_USAGE_SENSOR_PROPERTY_CHANGE_SENSITIVITY_ABS,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_16(0xFF,0xFF),
        HID_REPORT_SIZE(16),
        HID_REPORT_COUNT(1),
        // HID_USAGE_SENSOR_UNITS_BAR,
        HID_UNIT_EXPONENT(0x0E), // scale default unit “Bar” to provide 2 digits past the decimal point
        HID_FEATURE(Data_Var_Abs),
        HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_ENVIRONMENTAL_ATMOSPHERIC_PRESSURE,HID_USAGE_SENSOR_DATA_MOD_MAX),
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_16(0xFF,0xFF),
        HID_REPORT_SIZE(16),
        HID_REPORT_COUNT(1),
        // HID_USAGE_SENSOR_UNITS_BAR,
        HID_UNIT_EXPONENT(0x0E), // scale default unit “Bar” to provide 2 digits past the decimal point
        HID_FEATURE(Data_Var_Abs),
        HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_ENVIRONMENTAL_ATMOSPHERIC_PRESSURE,
                HID_USAGE_SENSOR_DATA_MOD_MIN),
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_16(0xFF,0xFF),
        HID_REPORT_SIZE(16),
        HID_REPORT_COUNT(1),
        // HID_USAGE_SENSOR_UNITS_BAR,
        HID_UNIT_EXPONENT(0x0E), // scale default unit “Bar” to provide 2 digits past the decimal point
        HID_FEATURE(Data_Var_Abs),

        //input reports (transmit)
        HID_USAGE_PAGE_SENSOR,
        HID_USAGE_SENSOR_STATE,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_8(6),
        HID_REPORT_SIZE(8),
        HID_REPORT_COUNT(1),
        HID_COLLECTION(Logical),
                HID_USAGE_SENSOR_STATE_UNKNOWN,
                HID_USAGE_SENSOR_STATE_READY,
                HID_USAGE_SENSOR_STATE_NOT_AVAILABLE,
                HID_USAGE_SENSOR_STATE_NO_DATA,
                HID_USAGE_SENSOR_STATE_INITIALIZING,
                HID_USAGE_SENSOR_STATE_ACCESS_DENIED,
                HID_USAGE_SENSOR_STATE_ERROR,
                HID_INPUT(Const_Arr_Abs),
        HID_END_COLLECTION,
        HID_USAGE_SENSOR_EVENT,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_8(16),
        HID_REPORT_SIZE(8),
        HID_REPORT_COUNT(1),
        HID_COLLECTION(Logical),
                HID_USAGE_SENSOR_EVENT_UNKNOWN,
                HID_USAGE_SENSOR_EVENT_STATE_CHANGED,
                HID_USAGE_SENSOR_EVENT_PROPERTY_CHANGED,
                HID_USAGE_SENSOR_EVENT_DATA_UPDATED,
                HID_USAGE_SENSOR_EVENT_POLL_RESPONSE,
                HID_USAGE_SENSOR_EVENT_CHANGE_SENSITIVITY,
                HID_USAGE_SENSOR_EVENT_MAX_REACHED,
                HID_USAGE_SENSOR_EVENT_MIN_REACHED,
                HID_USAGE_SENSOR_EVENT_HIGH_THRESHOLD_CROSS_UPWARD,
                HID_USAGE_SENSOR_EVENT_HIGH_THRESHOLD_CROSS_DOWNWARD,
                HID_USAGE_SENSOR_EVENT_LOW_THRESHOLD_CROSS_UPWARD,
                HID_USAGE_SENSOR_EVENT_LOW_THRESHOLD_CROSS_DOWNWARD,
                HID_USAGE_SENSOR_EVENT_ZERO_THRESHOLD_CROSS_UPWARD,
                HID_USAGE_SENSOR_EVENT_ZERO_THRESHOLD_CROSS_DOWNWARD,
                HID_USAGE_SENSOR_EVENT_PERIOD_EXCEEDED,
                HID_USAGE_SENSOR_EVENT_FREQUENCY_EXCEEDED,
                HID_USAGE_SENSOR_EVENT_COMPLEX_TRIGGER,
                HID_INPUT(Const_Arr_Abs),
        HID_END_COLLECTION,
        HID_USAGE_SENSOR_DATA_ENVIRONMENTAL_ATMOSPHERIC_PRESSURE,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_16(0xFF,0xFF),
        HID_REPORT_SIZE(16),
        HID_REPORT_COUNT(1),
        // HID_USAGE_SENSOR_UNITS_BAR,
        HID_UNIT_EXPONENT(0x0E), // scale default unit “Bar” to provide 2 digits past the decimal point
        HID_INPUT(Const_Var_Abs),

        HID_END_COLLECTION
};
```

4.3.10 Environmental: Humidity

```
// For reference: Complete HID report descriptor

const unsigned char hyg_report_descriptor[] = {
        HID_USAGE_PAGE_SENSOR,
```

```
HID_USAGE_SENSOR_TYPE_ENVIRONMENTAL_HUMIDITY,
HID_COLLECTION(Physical),

//feature reports (xmit/receive)
HID_USAGE_PAGE_SENSOR,
HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_8(5),
HID_REPORT_SIZE(8),
HID_REPORT_COUNT(1),
HID_COLLECTION(Logical),
          HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_NO_EVENTS,
          HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS,
          HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_THRESHOLD_EVENTS,
          HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_NO_EVENTS_WAKE,
          HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS_WAKE,
          HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_THRESHOLD_EVENTS_WAKE,
          HID_FEATURE(Data_Arr_Abs),
HID_END_COLLECTION,
HID_USAGE_SENSOR_PROPERTY_SENSOR_STATUS,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_32(0xFF,0xFF,0xFF,0xFF),
HID_REPORT_SIZE(32),
HID_REPORT_COUNT(1),
HID_FEATURE(Data_Var_Abs), // up to VT_UI4 worth of status info
HID_USAGE_SENSOR_PROPERTY_REPORT_INTERVAL,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_32(0xFF,0xFF,0xFF,0xFF),
HID_REPORT_SIZE(32),
HID_REPORT_COUNT(1),
// HID_USAGE_SENSOR_UNITS_MILLISECOND,
HID_UNIT_EXPONENT(0),
HID_FEATURE(Data_Var_Abs),
HID_USAGE_SENSOR_PROPERTY_SENSOR_CONNECTION_TYPE,  // NAry
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_8(2),
HID_REPORT_SIZE(8),
HID_REPORT_COUNT(1),
HID_COLLECTION(Logical),
          HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_INTEGRATED,
          HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_ATTACHED,
          HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_EXTERNAL,
          HID_FEATURE(Const_Arr_Abs),
HID_END_COLLECTION,
HID_USAGE_SENSOR_PROPERTY_CHANGE_SENSITIVITY_ABS,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_16(0x10,0x27), // 10000 = 0.00 to 100.00 percent with 2 digits past decimal point
HID_REPORT_SIZE(16),
HID_REPORT_COUNT(1),
// HID_USAGE_SENSOR_UNITS_PERCENT,
HID_UNIT_EXPONENT(0x0E), // scale default unit to provide 2 digits past the decimal point
HID_FEATURE(Data_Var_Abs),
HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_ENVIRONMENTAL_RELATIVE_HUMIDITY,HID_USAGE_SENSOR_DATA_MOD_MAX),
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_16(0x10,0x27), // 10000 = 0.00 to 100.00 percent with 2 digits past decimal point
HID_REPORT_SIZE(16),
HID_REPORT_COUNT(1),
// HID_USAGE_SENSOR_UNITS_PERCENT,
HID_UNIT_EXPONENT(0x0E), // scale default unit to provide 2 digits past the decimal point
HID_FEATURE(Data_Var_Abs),
HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_ENVIRONMENTAL_RELATIVE_HUMIDITY,HID_USAGE_SENSOR_DATA_MOD_MIN),
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_16(0x10,0x27), // 10000 = 0.00 to 100.00 percent with 2 digits past decimal point
HID_REPORT_SIZE(16),
HID_REPORT_COUNT(1),
// HID_USAGE_SENSOR_UNITS_PERCENT,
HID_UNIT_EXPONENT(0x0E), // scale default unit to provide 2 digits past the decimal point
HID_FEATURE(Data_Var_Abs),

//input reports (transmit)
HID_USAGE_PAGE_SENSOR,
HID_USAGE_SENSOR_STATE,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_8(6),
HID_REPORT_SIZE(8),
HID_REPORT_COUNT(1),
HID_COLLECTION(Logical),
          HID_USAGE_SENSOR_STATE_UNKNOWN,
          HID_USAGE_SENSOR_STATE_READY,
          HID_USAGE_SENSOR_STATE_NOT_AVAILABLE,
          HID_USAGE_SENSOR_STATE_NO_DATA,
          HID_USAGE_SENSOR_STATE_INITIALIZING,
          HID_USAGE_SENSOR_STATE_ACCESS_DENIED,
          HID_USAGE_SENSOR_STATE_ERROR,
          HID_INPUT(Const_Arr_Abs),
HID_END_COLLECTION,
HID_USAGE_SENSOR_EVENT,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_8(16),
HID_REPORT_SIZE(8),
HID_REPORT_COUNT(1),
HID_COLLECTION(Logical),
          HID_USAGE_SENSOR_EVENT_UNKNOWN,
          HID_USAGE_SENSOR_EVENT_STATE_CHANGED,
          HID_USAGE_SENSOR_EVENT_PROPERTY_CHANGED,
```

```
                    HID_USAGE_SENSOR_EVENT_DATA_UPDATED,
                    HID_USAGE_SENSOR_EVENT_POLL_RESPONSE,
                    HID_USAGE_SENSOR_EVENT_CHANGE_SENSITIVITY,
                    HID_USAGE_SENSOR_EVENT_MAX_REACHED,
                    HID_USAGE_SENSOR_EVENT_MIN_REACHED,
                    HID_USAGE_SENSOR_EVENT_HIGH_THRESHOLD_CROSS_UPWARD,
                    HID_USAGE_SENSOR_EVENT_HIGH_THRESHOLD_CROSS_DOWNWARD,
                    HID_USAGE_SENSOR_EVENT_LOW_THRESHOLD_CROSS_UPWARD,
                    HID_USAGE_SENSOR_EVENT_LOW_THRESHOLD_CROSS_DOWNWARD,
                    HID_USAGE_SENSOR_EVENT_ZERO_THRESHOLD_CROSS_UPWARD,
                    HID_USAGE_SENSOR_EVENT_ZERO_THRESHOLD_CROSS_DOWNWARD,
                    HID_USAGE_SENSOR_EVENT_PERIOD_EXCEEDED,
                    HID_USAGE_SENSOR_EVENT_FREQUENCY_EXCEEDED,
                    HID_USAGE_SENSOR_EVENT_COMPLEX_TRIGGER,
                    HID_INPUT(Const_Arr_Abs),
          HID_END_COLLECTION,
          HID_USAGE_SENSOR_DATA_ENVIRONMENTAL_RELATIVE_HUMIDITY,
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_16(0x10,0x27), // 10000 = 0.00 to 100.00 percent with 2 digits past decimal point
          HID_REPORT_SIZE(16),
          HID_REPORT_COUNT(1),
          // HID_USAGE_SENSOR_UNITS_PERCENT,
          HID_UNIT_EXPONENT(0x0E), // scale default unit “percent” to provide 2 digits past the decimal point
          HID_INPUT(Const_Var_Abs),

          HID_END_COLLECTION
};
```

4.3.11 Environmental: Temperature

```
// For reference: Complete HID report descriptor

const unsigned char temp_report_descriptor[] = {
          HID_USAGE_PAGE_SENSOR,
          HID_USAGE_SENSOR_TYPE_ENVIRONMENTAL_TEMPERATURE,
          HID_COLLECTION(Physical),

          //feature reports (xmit/receive)
          HID_USAGE_PAGE_SENSOR,
          HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE,
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_8(5),
          HID_REPORT_SIZE(8),
          HID_REPORT_COUNT(1),
          HID_COLLECTION(Logical),
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_NO_EVENTS,
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS,
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_THRESHOLD_EVENTS,
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_NO_EVENTS_WAKE,
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS_WAKE,
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_THRESHOLD_EVENTS_WAKE,
                    HID_FEATURE(Data_Arr_Abs),
          HID_END_COLLECTION,
          HID_USAGE_SENSOR_PROPERTY_SENSOR_STATUS,
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_32(0xFF,0xFF,0xFF,0xFF),
          HID_REPORT_SIZE(32),
          HID_REPORT_COUNT(1),
          HID_FEATURE(Data_Var_Abs), // up to VT_UI4 worth of status info
          HID_USAGE_SENSOR_PROPERTY_REPORT_INTERVAL,
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_32(0xFF,0xFF,0xFF,0xFF),
          HID_REPORT_SIZE(32),
          HID_REPORT_COUNT(1),
          // HID_USAGE_SENSOR_UNITS_MILLISECOND,
          HID_UNIT_EXPONENT(0),
          HID_FEATURE(Data_Var_Abs),
          HID_USAGE_SENSOR_PROPERTY_SENSOR_CONNECTION_TYPE,  // NAry
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_8(2),
          HID_REPORT_SIZE(8),
          HID_REPORT_COUNT(1),
          HID_COLLECTION(Logical),
                    HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_INTEGRATED,
                    HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_ATTACHED,
                    HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_EXTERNAL,
                    HID_FEATURE(Const_Arr_Abs),
          HID_END_COLLECTION,
          HID_USAGE_SENSOR_PROPERTY_CHANGE_SENSITIVITY_ABS,
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_16(0xFF,0xFF),
          HID_REPORT_SIZE(16),
          HID_REPORT_COUNT(1),
          // HID_USAGE_SENSOR_UNITS_CELSIUS,
          HID_UNIT_EXPONENT(0x0E), // scale default unit “Celsius” to provide 2 digits past the decimal point
          HID_FEATURE(Data_Var_Abs),
          HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_ENVIRONMENTAL_TEMPERATURE,HID_USAGE_SENSOR_DATA_MOD_MAX),
          HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
          HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
          HID_REPORT_SIZE(16),
          HID_REPORT_COUNT(1),
          // HID_USAGE_SENSOR_UNITS_CELSIUS,
          HID_UNIT_EXPONENT(0x0E), // scale default unit “Celsius” to provide 2 digits past the decimal point
          HID_FEATURE(Data_Var_Abs),
```

```
        HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_ENVIRONMENTAL_TEMPERATURE,HID_USAGE_SENSOR_DATA_MOD_MIN),
        HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
        HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
        HID_REPORT_SIZE(16),
        HID_REPORT_COUNT(1),
        // HID_USAGE_SENSOR_UNITS_CELSIUS,
        HID_UNIT_EXPONENT(0x0E), // scale default unit "Celsius" to provide 2 digits past the decimal point
        HID_FEATURE(Data_Var_Abs),

        //input reports (transmit)
        HID_USAGE_PAGE_SENSOR,
        HID_USAGE_SENSOR_STATE,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_8(6),
        HID_REPORT_SIZE(8),
        HID_REPORT_COUNT(1),
        HID_COLLECTION(Logical),
                HID_USAGE_SENSOR_STATE_UNKNOWN,
                HID_USAGE_SENSOR_STATE_READY,
                HID_USAGE_SENSOR_STATE_NOT_AVAILABLE,
                HID_USAGE_SENSOR_STATE_NO_DATA,
                HID_USAGE_SENSOR_STATE_INITIALIZING,
                HID_USAGE_SENSOR_STATE_ACCESS_DENIED,
                HID_USAGE_SENSOR_STATE_ERROR,
                HID_INPUT(Const_Arr_Abs),
        HID_END_COLLECTION,
        HID_USAGE_SENSOR_EVENT,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_8(16),
        HID_REPORT_SIZE(8),
        HID_REPORT_COUNT(1),
        HID_COLLECTION(Logical),
                HID_USAGE_SENSOR_EVENT_UNKNOWN,
                HID_USAGE_SENSOR_EVENT_STATE_CHANGED,
                HID_USAGE_SENSOR_EVENT_PROPERTY_CHANGED,
                HID_USAGE_SENSOR_EVENT_DATA_UPDATED,
                HID_USAGE_SENSOR_EVENT_POLL_RESPONSE,
                HID_USAGE_SENSOR_EVENT_CHANGE_SENSITIVITY,
                HID_USAGE_SENSOR_EVENT_MAX_REACHED,
                HID_USAGE_SENSOR_EVENT_MIN_REACHED,
                HID_USAGE_SENSOR_EVENT_HIGH_THRESHOLD_CROSS_UPWARD,
                HID_USAGE_SENSOR_EVENT_HIGH_THRESHOLD_CROSS_DOWNWARD,
                HID_USAGE_SENSOR_EVENT_LOW_THRESHOLD_CROSS_UPWARD,
                HID_USAGE_SENSOR_EVENT_LOW_THRESHOLD_CROSS_DOWNWARD,
                HID_USAGE_SENSOR_EVENT_ZERO_THRESHOLD_CROSS_UPWARD,
                HID_USAGE_SENSOR_EVENT_ZERO_THRESHOLD_CROSS_DOWNWARD,
                HID_USAGE_SENSOR_EVENT_PERIOD_EXCEEDED,
                HID_USAGE_SENSOR_EVENT_FREQUENCY_EXCEEDED,
                HID_USAGE_SENSOR_EVENT_COMPLEX_TRIGGER,
                HID_INPUT(Const_Arr_Abs),
        HID_END_COLLECTION,
        HID_USAGE_SENSOR_DATA_ENVIRONMENTAL_TEMPERATURE,
        HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
        HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
        HID_REPORT_SIZE(16),
        HID_REPORT_COUNT(1),
        // HID_USAGE_SENSOR_UNITS_CELSIUS,
        HID_UNIT_EXPONENT(0x0E), // scale default unit "Celsius" to provide 2 digits past the decimal point
        HID_INPUT(Const_Var_Abs),

        HID_END_COLLECTION
};
```

4.3.12 Light: Ambient Light

```
// For reference: Complete HID report descriptor

const unsigned char als_report_descriptor[] = {
        HID_USAGE_PAGE_SENSOR,          // USAGE_PAGE (Sensor)
        HID_USAGE_SENSOR_TYPE_LIGHT_AMBIENTLIGHT, // USAGE (AmbientLight)
        HID_COLLECTION(Physical),

        //feature reports (xmit/receive)
        HID_USAGE_PAGE_SENSOR,
        HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_8(5),
        HID_REPORT_SIZE(8),
        HID_REPORT_COUNT(1),
        HID_COLLECTION(Logical),
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_NO_EVENTS,
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS,
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_THRESHOLD_EVENTS,
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_NO_EVENTS_WAKE,
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS_WAKE,
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_THRESHOLD_EVENTS_WAKE,
                HID_FEATURE(Data_Arr_Abs),
        HID_END_COLLECTION,
        HID_USAGE_SENSOR_PROPERTY_SENSOR_STATUS,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_32(0xFF,0xFF,0xFF,0xFF),
        HID_REPORT_SIZE(32),
        HID_REPORT_COUNT(1),
        HID_FEATURE(Data_Var_Abs), // up to VT_UI4 worth of status info
```

```
HID_USAGE_SENSOR_PROPERTY_REPORT_INTERVAL,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_32(0xFF,0xFF,0xFF,0xFF),
HID_REPORT_SIZE(32),
HID_REPORT_COUNT(1),
// HID_USAGE_SENSOR_UNITS_MILLISECOND,
HID_UNIT_EXPONENT(0),
HID_FEATURE(Data_Var_Abs),
HID_USAGE_SENSOR_PROPERTY_SENSOR_CONNECTION_TYPE,  // NAry
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_8(2),
HID_REPORT_SIZE(8),
HID_REPORT_COUNT(1),
HID_COLLECTION(Logical),
          HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_INTEGRATED,
          HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_ATTACHED,
          HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_EXTERNAL,
          HID_FEATURE(Const_Arr_Abs),
HID_END_COLLECTION,
HID_USAGE_SENSOR_PROPERTY_CHANGE_SENSITIVITY_REL_PCT,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_16(0x10,0x27), // 10000 = 0.00 to 100.00 percent with 2 digits past decimal point
HID_REPORT_SIZE(16),
HID_REPORT_COUNT(1),
// HID_USAGE_SENSOR_UNITS_PERCENT,
HID_UNIT_EXPONENT(0x0E),  // scale default unit to provide 2 digits past decimal point
HID_FEATURE(Data_Var_Abs),
HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_LIGHT_ILLUMINANCE,HID_USAGE_SENSOR_DATA_MOD_MAX),
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_16(0xFF,0xFF),
HID_REPORT_SIZE(16),
HID_REPORT_COUNT(1),
HID_USAGE_SENSOR_UNITS_LUX,
HID_UNIT_EXPONENT(0x0F), // scale default unit to provide 1 digit past decimal point
HID_FEATURE(Data_Var_Abs),
HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_LIGHT_ILLUMINANCE,HID_USAGE_SENSOR_DATA_MOD_MIN),
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_16(0xFF,0xFF),
HID_REPORT_SIZE(16),
HID_REPORT_COUNT(1),
HID_USAGE_SENSOR_UNITS_LUX,
HID_UNIT_EXPONENT(0x0F), // scale default unit to provide 1 digit past decimal point
HID_FEATURE(Data_Var_Abs),

//input reports (transmit)
HID_USAGE_PAGE_SENSOR,
HID_USAGE_SENSOR_STATE,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_8(6),
HID_REPORT_SIZE(8),
HID_REPORT_COUNT(1),
HID_COLLECTION(Logical),
          HID_USAGE_SENSOR_STATE_UNKNOWN,
          HID_USAGE_SENSOR_STATE_READY,
          HID_USAGE_SENSOR_STATE_NOT_AVAILABLE,
          HID_USAGE_SENSOR_STATE_NO_DATA,
          HID_USAGE_SENSOR_STATE_INITIALIZING,
          HID_USAGE_SENSOR_STATE_ACCESS_DENIED,
          HID_USAGE_SENSOR_STATE_ERROR,
          HID_INPUT(Const_Arr_Abs),
HID_END_COLLECTION,
HID_USAGE_SENSOR_EVENT,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_8(16),
HID_REPORT_SIZE(8),
HID_REPORT_COUNT(1),
HID_COLLECTION(Logical),
          HID_USAGE_SENSOR_EVENT_UNKNOWN,
          HID_USAGE_SENSOR_EVENT_STATE_CHANGED,
          HID_USAGE_SENSOR_EVENT_PROPERTY_CHANGED,
          HID_USAGE_SENSOR_EVENT_DATA_UPDATED,
          HID_USAGE_SENSOR_EVENT_POLL_RESPONSE,
          HID_USAGE_SENSOR_EVENT_CHANGE_SENSITIVITY,
          HID_USAGE_SENSOR_EVENT_MAX_REACHED,
          HID_USAGE_SENSOR_EVENT_MIN_REACHED,
          HID_USAGE_SENSOR_EVENT_HIGH_THRESHOLD_CROSS_UPWARD,
          HID_USAGE_SENSOR_EVENT_HIGH_THRESHOLD_CROSS_DOWNWARD,
          HID_USAGE_SENSOR_EVENT_LOW_THRESHOLD_CROSS_UPWARD,
          HID_USAGE_SENSOR_EVENT_LOW_THRESHOLD_CROSS_DOWNWARD,
          HID_USAGE_SENSOR_EVENT_ZERO_THRESHOLD_CROSS_UPWARD,
          HID_USAGE_SENSOR_EVENT_ZERO_THRESHOLD_CROSS_DOWNWARD,
          HID_USAGE_SENSOR_EVENT_PERIOD_EXCEEDED,
          HID_USAGE_SENSOR_EVENT_FREQUENCY_EXCEEDED,
          HID_USAGE_SENSOR_EVENT_COMPLEX_TRIGGER,
          HID_INPUT(Const_Arr_Abs),
HID_END_COLLECTION,
HID_USAGE_SENSOR_DATA_LIGHT_ILLUMINANCE,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_16(0xFF,0xFF),
HID_UNIT_EXPONENT(0x0F), // scale default unit to provide 1 digit past decimal point
HID_REPORT_SIZE(16),
HID_REPORT_COUNT(1),
HID_INPUT(Const_Var_Abs),
HID_USAGE_SENSOR_DATA_LIGHT_COLOR_TEMPERATURE,
HID_LOGICAL_MIN_8(0),
```

```
          HID_LOGICAL_MAX_16(0xFF,0xFF),
          HID_USAGE_SENSOR_UNITS_KELVIN,
          HID_UNIT_EXPONENT(0),
          HID_REPORT_SIZE(16),
          HID_REPORT_COUNT(1),
          HID_INPUT(Const_Var_Abs),
          HID_USAGE_SENSOR_DATA_LIGHT_CHROMATICITY_X,
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_16(0xFF,0xFF),
          HID_USAGE_SENSOR_UNITS_NOT_SPECIFIED,
          HID_UNIT_EXPONENT(0x0C), // scale default unit to provide 4 digits past decimal point
          HID_REPORT_SIZE(16),
          HID_REPORT_COUNT(1),
          HID_INPUT(Const_Var_Abs),
          HID_USAGE_SENSOR_DATA_LIGHT_CHROMATICITY_Y,
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_16(0xFF,0xFF),
          HID_USAGE_SENSOR_UNITS_NOT_SPECIFIED,
          HID_UNIT_EXPONENT(0x0C), // scale default unit to provide 4 digits past decimal point
          HID_REPORT_SIZE(16),
          HID_REPORT_COUNT(1),
          HID_INPUT(Const_Var_Abs),

          HID_END_COLLECTION
};
```

4.3.13 Location: GPS

The use of a Location category sensor may be typified by definitions for a Global Positioning Sytem type sensor. As an example, two examples are provided illustrating how a GPS sensor may be defined.

The first example assumes that NMEA sentences are produced by the sensor, and that those sentences are parsed within the driver that supports that sensor.

Notice that to accommodate the fact that NMEA sentences can be quite long, the sensor may break those sentences up into chunks that are sent in back-to-back Input Reports. The driver (and/or upper-layer software) would need to concatenate the NMEA sentence fragments from some number of Input reports in order to create a complete NMEA sentence. This is not difficult to do, because all NMEA sentences begin with the unique symbol ‘$’ and end with an ‘*’ followed by a 2-digit hex checksum and then a Carriage Return and Line Feed.

```
// Complete NMEA GPS HID report descriptor

const unsigned char nmea_report_descriptor[] = {
          HID_USAGE_PAGE_SENSOR,
          HID_USAGE_SENSOR_TYPE_LOCATION_GPS,
          HID_COLLECTION(Physical),

          //feature reports (xmit/receive)
          HID_USAGE_PAGE_SENSOR,
          HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE,
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_8(5),
          HID_REPORT_SIZE(8),
          HID_REPORT_COUNT(1),
          HID_COLLECTION(Logical),
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_NO_EVENTS,
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS,
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_THRESHOLD_EVENTS,
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_NO_EVENTS_WAKE,
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS_WAKE,
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_THRESHOLD_EVENTS_WAKE,
                    HID_FEATURE(Data_Arr_Abs),
          HID_END_COLLECTION,
          HID_USAGE_SENSOR_PROPERTY_SENSOR_STATUS,
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_32(0xFF,0xFF,0xFF,0xFF),
          HID_REPORT_SIZE(32),
          HID_REPORT_COUNT(1),
          HID_FEATURE(Data_Var_Abs), // up to VT_UI4 worth of status info
          HID_USAGE_SENSOR_PROPERTY_REPORT_INTERVAL,
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_32(0xFF,0xFF,0xFF,0xFF),
          HID_REPORT_SIZE(32),
          HID_REPORT_COUNT(1),
          // HID_USAGE_SENSOR_UNITS_MILLISECOND,
          HID_UNIT_EXPONENT(0),
          HID_FEATURE(Data_Var_Abs),
          HID_USAGE_SENSOR_PROPERTY_SENSOR_CONNECTION_TYPE,  // NAry
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_8(2),
          HID_REPORT_SIZE(8),
```

```
        HID_REPORT_COUNT(1),
        HID_COLLECTION(Logical),
                HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_INTEGRATED,
                HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_ATTACHED,
                HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_EXTERNAL,
                HID_FEATURE(Const_Arr_Abs),
        HID_END_COLLECTION,
        HID_USAGE_SENSOR_PROPERTY_CHANGE_SENSITIVITY_ABS,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_16(0xFF,0xFF),
        HID_REPORT_SIZE(16),
        HID_REPORT_COUNT(1),
        HID_USAGE_SENSOR_UNITS_HERTZ,
        HID_UNIT_EXPONENT(0x0E), // scale default unit to provide 2 digits past the decimal point
        HID_FEATURE(Data_Var_Abs),

        //input reports (transmit)
        HID_USAGE_PAGE_SENSOR,
        HID_USAGE_SENSOR_STATE,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_8(6),
        HID_REPORT_SIZE(8),
        HID_REPORT_COUNT(1),
        HID_COLLECTION(Logical),
                HID_USAGE_SENSOR_STATE_UNKNOWN,
                HID_USAGE_SENSOR_STATE_READY,
                HID_USAGE_SENSOR_STATE_NOT_AVAILABLE,
                HID_USAGE_SENSOR_STATE_NO_DATA,
                HID_USAGE_SENSOR_STATE_INITIALIZING,
                HID_USAGE_SENSOR_STATE_ACCESS_DENIED,
                HID_USAGE_SENSOR_STATE_ERROR,
                HID_INPUT(Const_Arr_Abs),
        HID_END_COLLECTION,
        HID_USAGE_SENSOR_EVENT,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_8(16),
        HID_REPORT_SIZE(8),
        HID_REPORT_COUNT(1),
        HID_COLLECTION(Logical),
                HID_USAGE_SENSOR_EVENT_UNKNOWN,
                HID_USAGE_SENSOR_EVENT_STATE_CHANGED,
                HID_USAGE_SENSOR_EVENT_PROPERTY_CHANGED,
                HID_USAGE_SENSOR_EVENT_DATA_UPDATED,
                HID_USAGE_SENSOR_EVENT_POLL_RESPONSE,
                HID_USAGE_SENSOR_EVENT_CHANGE_SENSITIVITY,
                HID_USAGE_SENSOR_EVENT_MAX_REACHED,
                HID_USAGE_SENSOR_EVENT_MIN_REACHED,
                HID_USAGE_SENSOR_EVENT_HIGH_THRESHOLD_CROSS_UPWARD,
                HID_USAGE_SENSOR_EVENT_HIGH_THRESHOLD_CROSS_DOWNWARD,
                HID_USAGE_SENSOR_EVENT_LOW_THRESHOLD_CROSS_UPWARD,
                HID_USAGE_SENSOR_EVENT_LOW_THRESHOLD_CROSS_DOWNWARD,
                HID_USAGE_SENSOR_EVENT_ZERO_THRESHOLD_CROSS_UPWARD,
                HID_USAGE_SENSOR_EVENT_ZERO_THRESHOLD_CROSS_DOWNWARD,
                HID_USAGE_SENSOR_EVENT_PERIOD_EXCEEDED,
                HID_USAGE_SENSOR_EVENT_FREQUENCY_EXCEEDED,
                HID_USAGE_SENSOR_EVENT_COMPLEX_TRIGGER,
                HID_INPUT(Const_Arr_Abs),
        HID_END_COLLECTION,
        HID_USAGE_SENSOR_DATA_LOCATION_NMEA_SENTENCE,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_8(255),
        HID_REPORT_SIZE(8), // narrow characters
        HID_REPORT_COUNT(60),
        HID_INPUT(Const_Arr_Abs),

        HID_END_COLLECTION
};
```

The second example assumes that the GPS sensor internally separates out the various fields that it reports to the driver. This method reports the same information to the driver without regard to how it is derived from the physical GPS receiver.

```
// Complete HID report descriptor

const unsigned char gps_report_descriptor[] = {
        HID_USAGE_PAGE_SENSOR,
        HID_USAGE_SENSOR_TYPE_LOCATION_GPS,
        HID_COLLECTION(Physical),

        //feature reports (xmit/receive)
        HID_USAGE_PAGE_SENSOR,
        HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_8(5),
        HID_REPORT_SIZE(8),
        HID_REPORT_COUNT(1),
        HID_COLLECTION(Logical),
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_NO_EVENTS,
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS,
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_THRESHOLD_EVENTS,
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_NO_EVENTS_WAKE,
```

```
          HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS_WAKE,
          HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_THRESHOLD_EVENTS_WAKE,
          HID_FEATURE(Data_Arr_Abs),
HID_END_COLLECTION,
HID_USAGE_SENSOR_PROPERTY_SENSOR_STATUS,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_32(0xFF,0xFF,0xFF,0xFF),
HID_REPORT_SIZE(32),
HID_REPORT_COUNT(1),
HID_FEATURE(Data_Var_Abs), // up to VT_UI4 worth of status info
HID_USAGE_SENSOR_PROPERTY_REPORT_INTERVAL,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_32(0xFF,0xFF,0xFF,0xFF),
HID_REPORT_SIZE(32),
HID_REPORT_COUNT(1),
// HID_USAGE_SENSOR_UNITS_MILLISECOND,
HID_UNIT_EXPONENT(0),
HID_FEATURE(Data_Var_Abs),
HID_USAGE_SENSOR_PROPERTY_SENSOR_CONNECTION_TYPE,  // NAry
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_8(2),
HID_REPORT_SIZE(8),
HID_REPORT_COUNT(1),
HID_COLLECTION(Logical),
          HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_INTEGRATED,
          HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_ATTACHED,
          HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_EXTERNAL,
          HID_FEATURE(Const_Arr_Abs),
HID_END_COLLECTION,
HID_USAGE_SENSOR_PROPERTY_CHANGE_SENSITIVITY_ABS,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_16(0xFF,0xFF),
HID_REPORT_SIZE(16),
HID_REPORT_COUNT(1),
// HID_USAGE_SENSOR_UNITS_METER,
HID_UNIT_EXPONENT(0x0E), // scale default unit “meter” to provide 2 digits past the decimal point
HID_FEATURE(Data_Var_Abs),

//input reports (transmit)
HID_USAGE_PAGE_SENSOR,
HID_USAGE_SENSOR_STATE,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_8(6),
HID_REPORT_SIZE(8),
HID_REPORT_COUNT(1),
HID_COLLECTION(Logical),
          HID_USAGE_SENSOR_STATE_UNKNOWN,
          HID_USAGE_SENSOR_STATE_READY,
          HID_USAGE_SENSOR_STATE_NOT_AVAILABLE,
          HID_USAGE_SENSOR_STATE_NO_DATA,
          HID_USAGE_SENSOR_STATE_INITIALIZING,
          HID_USAGE_SENSOR_STATE_ACCESS_DENIED,
          HID_USAGE_SENSOR_STATE_ERROR,
          HID_INPUT(Const_Arr_Abs),
HID_END_COLLECTION,
HID_USAGE_SENSOR_EVENT,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_8(16),
HID_REPORT_SIZE(8),
HID_REPORT_COUNT(1),
HID_COLLECTION(Logical),
          HID_USAGE_SENSOR_EVENT_UNKNOWN,
          HID_USAGE_SENSOR_EVENT_STATE_CHANGED,
          HID_USAGE_SENSOR_EVENT_PROPERTY_CHANGED,
          HID_USAGE_SENSOR_EVENT_DATA_UPDATED,
          HID_USAGE_SENSOR_EVENT_POLL_RESPONSE,
          HID_USAGE_SENSOR_EVENT_CHANGE_SENSITIVITY,
          HID_USAGE_SENSOR_EVENT_MAX_REACHED,
          HID_USAGE_SENSOR_EVENT_MIN_REACHED,
          HID_USAGE_SENSOR_EVENT_HIGH_THRESHOLD_CROSS_UPWARD,
          HID_USAGE_SENSOR_EVENT_HIGH_THRESHOLD_CROSS_DOWNWARD,
          HID_USAGE_SENSOR_EVENT_LOW_THRESHOLD_CROSS_UPWARD,
          HID_USAGE_SENSOR_EVENT_LOW_THRESHOLD_CROSS_DOWNWARD,
          HID_USAGE_SENSOR_EVENT_ZERO_THRESHOLD_CROSS_UPWARD,
          HID_USAGE_SENSOR_EVENT_ZERO_THRESHOLD_CROSS_DOWNWARD,
          HID_USAGE_SENSOR_EVENT_PERIOD_EXCEEDED,
          HID_USAGE_SENSOR_EVENT_FREQUENCY_EXCEEDED,
          HID_USAGE_SENSOR_EVENT_COMPLEX_TRIGGER,
          HID_INPUT(Const_Arr_Abs),
HID_END_COLLECTION,
HID_USAGE_SENSOR_DATA_LOCATION_LATITUDE,
HID_LOGICAL_MIN_32(0xFF,0xFF,0x01,0x00), // LOGICAL_MINIMUM (-2147483647)
HID_LOGICAL_MAX_32(0xFF,0x7F,0xFF,0xFF), // LOGICAL_MAXIMUM (2147483647)
HID_REPORT_SIZE(32),
HID_REPORT_COUNT(1),
HID_USAGE_SENSOR_UNITS_DEGREES,
HID_UNIT_EXPONENT(0x09), // scale unit to provide 7 digits past the decimal point
HID_INPUT(Const_Var_Abs),
HID_USAGE_SENSOR_DATA_LOCATION_LONGITUDE,
HID_LOGICAL_MIN_32(0xFF,0xFF,0x01,0x00), // LOGICAL_MINIMUM (-2147483647)
HID_LOGICAL_MAX_32(0xFF,0x7F,0xFF,0xFF), // LOGICAL_MAXIMUM (2147483647)
HID_REPORT_SIZE(32),
HID_REPORT_COUNT(1),
HID_USAGE_SENSOR_UNITS_DEGREES,
HID_UNIT_EXPONENT(0x09), // scale unit to provide 7 digits past the decimal point
```

```
        HID_INPUT(Const_Var_Abs),
        HID_USAGE_SENSOR_DATA_LOCATION_ERROR_RADIUS,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_32(0xFF,0xFF,0xFF,0xFF),
        HID_REPORT_SIZE(32),
        HID_REPORT_COUNT(1),
        // HID_USAGE_SENSOR_UNITS_METER,
        HID_UNIT_EXPONENT(0x09), // scale default unit “meter” to provide 7 digits past the decimal point
        HID_INPUT(Const_Var_Abs),
        HID_USAGE_SENSOR_DATA_LOCATION_ALTITUDE_SEALEVEL,
        HID_LOGICAL_MIN_32(0xFF,0xFF,0x01,0x00), // LOGICAL_MINIMUM (-2147483647)
        HID_LOGICAL_MAX_32(0xFF,0x7F,0xFF,0xFF), // LOGICAL_MAXIMUM (2147483647)
        HID_REPORT_SIZE(32),
        HID_REPORT_COUNT(1),
        HID_USAGE_SENSOR_UNITS_DEGREES,
        HID_UNIT_EXPONENT(0x0E), // scale unit to provide 2 digits past the decimal point
        HID_INPUT(Const_Var_Abs),
        HID_USAGE_SENSOR_DATA_LOCATION_ALTITUDE_SEALEVEL_ERROR,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_32(0xFF,0xFF,0xFF,0xFF),
        HID_REPORT_SIZE(32),
        HID_REPORT_COUNT(1),
        HID_USAGE_SENSOR_UNITS_DEGREES,
        HID_UNIT_EXPONENT(0x0E), // scale unit to provide 2 digits past the decimal point
        HID_INPUT(Const_Var_Abs),

        HID_END_COLLECTION
};
```

4.3.14 Mechanical: Switches

```
// For reference: Complete HID report descriptor

//Boolean Switch
const unsigned char swi_report_descriptor[] = {
        HID_USAGE_PAGE_SENSOR,
        HID_USAGE_SENSOR_TYPE_MECHANICAL_BOOLEAN_SWITCH,
        HID_COLLECTION(Physical),

        //feature reports (xmit/receive)
        HID_USAGE_PAGE_SENSOR,
        HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_8(5),
        HID_REPORT_SIZE(8),
        HID_REPORT_COUNT(1),
        HID_COLLECTION(Logical),
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_NO_EVENTS,
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS,
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_THRESHOLD_EVENTS,
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_NO_EVENTS_WAKE,
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS_WAKE,
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_THRESHOLD_EVENTS_WAKE,
                HID_FEATURE(Data_Arr_Abs),
        HID_END_COLLECTION,
        HID_USAGE_SENSOR_PROPERTY_SENSOR_STATUS,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_32(0xFF,0xFF,0xFF,0xFF),
        HID_REPORT_SIZE(32),
        HID_REPORT_COUNT(1),
        HID_FEATURE(Data_Var_Abs), // up to VT_UI4 worth of status info
        HID_USAGE_SENSOR_PROPERTY_REPORT_INTERVAL,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_32(0xFF,0xFF,0xFF,0xFF),
        HID_REPORT_SIZE(32),
        HID_REPORT_COUNT(1),
        // HID_USAGE_SENSOR_UNITS_MILLISECOND,
        HID_UNIT_EXPONENT(0),
        HID_FEATURE(Data_Var_Abs),
        HID_USAGE_SENSOR_PROPERTY_SENSOR_CONNECTION_TYPE,  // NAry
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_8(2),
        HID_REPORT_SIZE(8),
        HID_REPORT_COUNT(1),
        HID_COLLECTION(Logical),
                HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_INTEGRATED,
                HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_ATTACHED,
                HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_EXTERNAL,
                HID_FEATURE(Const_Arr_Abs),
        HID_END_COLLECTION,
        HID_USAGE_SENSOR_PROPERTY_CHANGE_SENSITIVITY_ABS,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_16(0xFF,0xFF),
        HID_REPORT_SIZE(16),
        HID_REPORT_COUNT(1),
        HID_USAGE_SENSOR_UNITS_NOT_SPECIFIED,
        HID_UNIT_EXPONENT(0),
        HID_FEATURE(Data_Var_Abs),

        //input reports (transmit)
        HID_USAGE_PAGE_SENSOR,
        HID_USAGE_SENSOR_STATE,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_8(6),
```

```
        HID_REPORT_SIZE(8),
        HID_REPORT_COUNT(1),
        HID_COLLECTION(Logical),
                HID_USAGE_SENSOR_STATE_UNKNOWN,
                HID_USAGE_SENSOR_STATE_READY,
                HID_USAGE_SENSOR_STATE_NOT_AVAILABLE,
                HID_USAGE_SENSOR_STATE_NO_DATA,
                HID_USAGE_SENSOR_STATE_INITIALIZING,
                HID_USAGE_SENSOR_STATE_ACCESS_DENIED,
                HID_USAGE_SENSOR_STATE_ERROR,
                HID_INPUT(Const_Arr_Abs),
        HID_END_COLLECTION,
        HID_USAGE_SENSOR_EVENT,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_8(16),
        HID_REPORT_SIZE(8),
        HID_REPORT_COUNT(1),
        HID_COLLECTION(Logical),
                HID_USAGE_SENSOR_EVENT_UNKNOWN,
                HID_USAGE_SENSOR_EVENT_STATE_CHANGED,
                HID_USAGE_SENSOR_EVENT_PROPERTY_CHANGED,
                HID_USAGE_SENSOR_EVENT_DATA_UPDATED,
                HID_USAGE_SENSOR_EVENT_POLL_RESPONSE,
                HID_USAGE_SENSOR_EVENT_CHANGE_SENSITIVITY,
                HID_USAGE_SENSOR_EVENT_MAX_REACHED,
                HID_USAGE_SENSOR_EVENT_MIN_REACHED,
                HID_USAGE_SENSOR_EVENT_HIGH_THRESHOLD_CROSS_UPWARD,
                HID_USAGE_SENSOR_EVENT_HIGH_THRESHOLD_CROSS_DOWNWARD,
                HID_USAGE_SENSOR_EVENT_LOW_THRESHOLD_CROSS_UPWARD,
                HID_USAGE_SENSOR_EVENT_LOW_THRESHOLD_CROSS_DOWNWARD,
                HID_USAGE_SENSOR_EVENT_ZERO_THRESHOLD_CROSS_UPWARD,
                HID_USAGE_SENSOR_EVENT_ZERO_THRESHOLD_CROSS_DOWNWARD,
                HID_USAGE_SENSOR_EVENT_PERIOD_EXCEEDED,
                HID_USAGE_SENSOR_EVENT_FREQUENCY_EXCEEDED,
                HID_USAGE_SENSOR_EVENT_COMPLEX_TRIGGER,
                HID_INPUT(Const_Arr_Abs),
        HID_END_COLLECTION,
        HID_USAGE_SENSOR_DATA_MECHANICAL_BOOLEAN_SWITCH_STATE,
        HID_LOGICAL_MIN_8(0), // Off
        HID_LOGICAL_MAX_8(1), // On
        HID_REPORT_SIZE(8),
        HID_REPORT_COUNT(1),
        HID_INPUT(Const_Var_Abs),

        HID_END_COLLECTION
};

//Multi-value Switch
const unsigned char swm_report_descriptor[] = {
        HID_USAGE_PAGE_SENSOR,
        HID_USAGE_SENSOR_TYPE_MECHANICAL_MULTIVALUE_SWITCH,
        HID_COLLECTION(Physical),

        //feature reports (xmit/receive)
        HID_USAGE_PAGE_SENSOR,
        HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_8(5),
        HID_REPORT_SIZE(8),
        HID_REPORT_COUNT(1),
        HID_COLLECTION(Logical),
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_NO_EVENTS,
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS,
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_THRESHOLD_EVENTS,
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_NO_EVENTS_WAKE,
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS_WAKE,
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_THRESHOLD_EVENTS_WAKE,
                HID_FEATURE(Data_Arr_Abs),
        HID_END_COLLECTION,
        HID_USAGE_SENSOR_PROPERTY_SENSOR_STATUS,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_32(0xFF,0xFF,0xFF,0xFF),
        HID_REPORT_SIZE(32),
        HID_REPORT_COUNT(1),
        HID_FEATURE(Data_Var_Abs), // up to VT_UI4 worth of status info
        HID_USAGE_SENSOR_PROPERTY_REPORT_INTERVAL,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_32(0xFF,0xFF,0xFF,0xFF),
        HID_REPORT_SIZE(32),
        HID_REPORT_COUNT(1),
        // HID_USAGE_SENSOR_UNITS_MILLISECOND,
        HID_UNIT_EXPONENT(0),
        HID_FEATURE(Data_Var_Abs),
        HID_USAGE_SENSOR_PROPERTY_SENSOR_CONNECTION_TYPE,  // NAry
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_8(2),
        HID_REPORT_SIZE(8),
        HID_REPORT_COUNT(1),
        HID_COLLECTION(Logical),
                HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_INTEGRATED,
                HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_ATTACHED,
                HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_EXTERNAL,
                HID_FEATURE(Const_Arr_Abs),
        HID_END_COLLECTION,
        HID_USAGE_SENSOR_PROPERTY_CHANGE_SENSITIVITY_ABS,
```

```
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_16(0xFF,0xFF),
        HID_REPORT_SIZE(16),
        HID_REPORT_COUNT(1),
        HID_USAGE_SENSOR_UNITS_NOT_SPECIFIED,
        HID_UNIT_EXPONENT(0),
        HID_FEATURE(Data_Var_Abs),

        //input reports (transmit)
        HID_USAGE_PAGE_SENSOR,
        HID_USAGE_SENSOR_STATE,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_8(6),
        HID_REPORT_SIZE(8),
        HID_REPORT_COUNT(1),
        HID_COLLECTION(Logical),
                HID_USAGE_SENSOR_STATE_UNKNOWN,
                HID_USAGE_SENSOR_STATE_READY,
                HID_USAGE_SENSOR_STATE_NOT_AVAILABLE,
                HID_USAGE_SENSOR_STATE_NO_DATA,
                HID_USAGE_SENSOR_STATE_INITIALIZING,
                HID_USAGE_SENSOR_STATE_ACCESS_DENIED,
                HID_USAGE_SENSOR_STATE_ERROR,
                HID_INPUT(Const_Arr_Abs),
        HID_END_COLLECTION,
        HID_USAGE_SENSOR_EVENT,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_8(16),
        HID_REPORT_SIZE(8),
        HID_REPORT_COUNT(1),
        HID_COLLECTION(Logical),
                HID_USAGE_SENSOR_EVENT_UNKNOWN,
                HID_USAGE_SENSOR_EVENT_STATE_CHANGED,
                HID_USAGE_SENSOR_EVENT_PROPERTY_CHANGED,
                HID_USAGE_SENSOR_EVENT_DATA_UPDATED,
                HID_USAGE_SENSOR_EVENT_POLL_RESPONSE,
                HID_USAGE_SENSOR_EVENT_CHANGE_SENSITIVITY,
                HID_USAGE_SENSOR_EVENT_MAX_REACHED,
                HID_USAGE_SENSOR_EVENT_MIN_REACHED,
                HID_USAGE_SENSOR_EVENT_HIGH_THRESHOLD_CROSS_UPWARD,
                HID_USAGE_SENSOR_EVENT_HIGH_THRESHOLD_CROSS_DOWNWARD,
                HID_USAGE_SENSOR_EVENT_LOW_THRESHOLD_CROSS_UPWARD,
                HID_USAGE_SENSOR_EVENT_LOW_THRESHOLD_CROSS_DOWNWARD,
                HID_USAGE_SENSOR_EVENT_ZERO_THRESHOLD_CROSS_UPWARD,
                HID_USAGE_SENSOR_EVENT_ZERO_THRESHOLD_CROSS_DOWNWARD,
                HID_USAGE_SENSOR_EVENT_PERIOD_EXCEEDED,
                HID_USAGE_SENSOR_EVENT_FREQUENCY_EXCEEDED,
                HID_USAGE_SENSOR_EVENT_COMPLEX_TRIGGER,
                HID_INPUT(Const_Arr_Abs),
        HID_END_COLLECTION,
        HID_USAGE_SENSOR_DATA_MECHANICAL_MULTIVALUE_SWITCH_VALUE,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_8(255),
        HID_REPORT_SIZE(8),
        HID_REPORT_COUNT(1),
        HID_INPUT(Const_Var_Abs),

        HID_END_COLLECTION
};

//Boolean switch array
const unsigned char swa_report_descriptor[] = {
        HID_USAGE_PAGE_SENSOR,
        HID_USAGE_SENSOR_TYPE_MECHANICAL_BOOLEAN_SWITCH_ARRAY,
        HID_COLLECTION(Physical),

        //feature reports (xmit/receive)
        HID_USAGE_PAGE_SENSOR,
        HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_8(5),
        HID_REPORT_SIZE(8),
        HID_REPORT_COUNT(1),
        HID_COLLECTION(Logical),
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_NO_EVENTS,
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS,
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_THRESHOLD_EVENTS,
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_NO_EVENTS_WAKE,
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS_WAKE,
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_THRESHOLD_EVENTS_WAKE,
                HID_FEATURE(Data_Arr_Abs),
        HID_END_COLLECTION,
        HID_USAGE_SENSOR_PROPERTY_SENSOR_STATUS,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_32(0xFF,0xFF,0xFF,0xFF),
        HID_REPORT_SIZE(32),
        HID_REPORT_COUNT(1),
        HID_FEATURE(Data_Var_Abs), // up to VT_UI4 worth of status info
        HID_USAGE_SENSOR_PROPERTY_REPORT_INTERVAL,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_32(0xFF,0xFF,0xFF,0xFF),
        HID_REPORT_SIZE(32),
        HID_REPORT_COUNT(1),
        // HID_USAGE_SENSOR_UNITS_MILLISECOND,
        HID_UNIT_EXPONENT(0),
```

```
        HID_FEATURE(Data_Var_Abs),
        HID_USAGE_SENSOR_PROPERTY_SENSOR_CONNECTION_TYPE,  // NAry
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_8(2),
        HID_REPORT_SIZE(8),
        HID_REPORT_COUNT(1),
        HID_COLLECTION(Logical),
                HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_INTEGRATED,
                HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_ATTACHED,
                HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_EXTERNAL,
                HID_FEATURE(Const_Arr_Abs),
        HID_END_COLLECTION,
        HID_USAGE_SENSOR_PROPERTY_CHANGE_SENSITIVITY_ABS,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_16(0xFF,0xFF),
        HID_REPORT_SIZE(16),
        HID_REPORT_COUNT(1),
        HID_USAGE_SENSOR_UNITS_NOT_SPECIFIED,
        HID_UNIT_EXPONENT(0),
        HID_FEATURE(Data_Var_Abs),

        //input reports (transmit)
        HID_USAGE_PAGE_SENSOR,
        HID_USAGE_SENSOR_STATE,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_8(6),
        HID_REPORT_SIZE(8),
        HID_REPORT_COUNT(1),
        HID_COLLECTION(Logical),
                HID_USAGE_SENSOR_STATE_UNKNOWN,
                HID_USAGE_SENSOR_STATE_READY,
                HID_USAGE_SENSOR_STATE_NOT_AVAILABLE,
                HID_USAGE_SENSOR_STATE_NO_DATA,
                HID_USAGE_SENSOR_STATE_INITIALIZING,
                HID_USAGE_SENSOR_STATE_ACCESS_DENIED,
                HID_USAGE_SENSOR_STATE_ERROR,
                HID_INPUT(Const_Arr_Abs),
        HID_END_COLLECTION,
        HID_USAGE_SENSOR_EVENT,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_8(16),
        HID_REPORT_SIZE(8),
        HID_REPORT_COUNT(1),
        HID_COLLECTION(Logical),
                HID_USAGE_SENSOR_EVENT_UNKNOWN,
                HID_USAGE_SENSOR_EVENT_STATE_CHANGED,
                HID_USAGE_SENSOR_EVENT_PROPERTY_CHANGED,
                HID_USAGE_SENSOR_EVENT_DATA_UPDATED,
                HID_USAGE_SENSOR_EVENT_POLL_RESPONSE,
                HID_USAGE_SENSOR_EVENT_CHANGE_SENSITIVITY,
                HID_USAGE_SENSOR_EVENT_MAX_REACHED,
                HID_USAGE_SENSOR_EVENT_MIN_REACHED,
                HID_USAGE_SENSOR_EVENT_HIGH_THRESHOLD_CROSS_UPWARD,
                HID_USAGE_SENSOR_EVENT_HIGH_THRESHOLD_CROSS_DOWNWARD,
                HID_USAGE_SENSOR_EVENT_LOW_THRESHOLD_CROSS_UPWARD,
                HID_USAGE_SENSOR_EVENT_LOW_THRESHOLD_CROSS_DOWNWARD,
                HID_USAGE_SENSOR_EVENT_ZERO_THRESHOLD_CROSS_UPWARD,
                HID_USAGE_SENSOR_EVENT_ZERO_THRESHOLD_CROSS_DOWNWARD,
                HID_USAGE_SENSOR_EVENT_PERIOD_EXCEEDED,
                HID_USAGE_SENSOR_EVENT_FREQUENCY_EXCEEDED,
                HID_USAGE_SENSOR_EVENT_COMPLEX_TRIGGER,
                HID_INPUT(Const_Arr_Abs),
        HID_END_COLLECTION,
        HID_USAGE_SENSOR_DATA_MECHANICAL_BOOLEAN_SWITCH_ARRAY_STATES,
        HID_LOGICAL_MIN_8(0), // Off
        HID_LOGICAL_MAX_8(1), // On
        HID_REPORT_SIZE(1),
        HID_REPORT_COUNT(8),
        HID_INPUT(Const_Arr_Abs),

        HID_END_COLLECTION
};
```

4.3.15 Motion: Accelerometer

```
// For reference: Complete HID report descriptor

// 1D Accelerometer
const unsigned char accel1_report_descriptor[] = {
        HID_USAGE_PAGE_SENSOR,
        HID_USAGE_SENSOR_TYPE_MOTION_ACCELEROMETER_1D,
        HID_COLLECTION(Physical),

        //feature reports (xmit/receive)
        HID_USAGE_PAGE_SENSOR,
        HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_8(5),
        HID_REPORT_SIZE(8),
        HID_REPORT_COUNT(1),
        HID_COLLECTION(Logical),
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_NO_EVENTS,
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS,
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_THRESHOLD_EVENTS,
```

```
          HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_NO_EVENTS_WAKE,
          HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS_WAKE,
          HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_THRESHOLD_EVENTS_WAKE,
          HID_FEATURE(Data_Arr_Abs),
HID_END_COLLECTION,
HID_USAGE_SENSOR_PROPERTY_SENSOR_STATUS,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_32(0xFF,0xFF,0xFF,0xFF),
HID_REPORT_SIZE(32),
HID_REPORT_COUNT(1),
HID_FEATURE(Data_Var_Abs), // up to VT_UI4 worth of status info
HID_USAGE_SENSOR_PROPERTY_REPORT_INTERVAL,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_32(0xFF,0xFF,0xFF,0xFF),
HID_REPORT_SIZE(32),
HID_REPORT_COUNT(1),
// HID_USAGE_SENSOR_UNITS_MILLISECOND,
HID_UNIT_EXPONENT(0),
HID_FEATURE(Data_Var_Abs),
HID_USAGE_SENSOR_PROPERTY_SENSOR_CONNECTION_TYPE,  // NAry
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_8(2),
HID_REPORT_SIZE(8),
HID_REPORT_COUNT(1),
HID_COLLECTION(Logical),
          HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_INTEGRATED,
          HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_ATTACHED,
          HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_EXTERNAL,
          HID_FEATURE(Const_Arr_Abs),
HID_END_COLLECTION,
HID_USAGE_SENSOR_PROPERTY_CHANGE_SENSITIVITY_ABS,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_16(0xFF,0xFF),
HID_REPORT_SIZE(16),
HID_REPORT_COUNT(1),
// HID_USAGE_SENSOR_UNITS_G,
HID_UNIT_EXPONENT(0x0E),  // scale default unit Gs to “centi-Gs” to provide 2 digits past Gs decimal point
HID_FEATURE(Data_Var_Abs),
HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_MOTION_ACCELERATION,HID_USAGE_SENSOR_DATA_MOD_MAX),
HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
HID_REPORT_SIZE(16),
HID_REPORT_COUNT(1),
// HID_USAGE_SENSOR_UNITS_G,
HID_UNIT_EXPONENT(0x0E),  // scale default unit Gs to “centi-Gs” to provide 2 digits past Gs decimal point
HID_FEATURE(Data_Var_Abs),
HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_MOTION_ACCELERATION,HID_USAGE_SENSOR_DATA_MOD_MIN),
HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
HID_REPORT_SIZE(16),
HID_REPORT_COUNT(1),
// HID_USAGE_SENSOR_UNITS_G,
HID_UNIT_EXPONENT(0x0E), // scale default unit Gs to “centi-Gs” to provide 2 digits past Gs decimal point
HID_FEATURE(Data_Var_Abs),

//input reports (transmit)
HID_USAGE_PAGE_SENSOR,
HID_USAGE_SENSOR_STATE,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_8(6),
HID_REPORT_SIZE(8),
HID_REPORT_COUNT(1),
HID_COLLECTION(Logical),
          HID_USAGE_SENSOR_STATE_UNKNOWN,
          HID_USAGE_SENSOR_STATE_READY,
          HID_USAGE_SENSOR_STATE_NOT_AVAILABLE,
          HID_USAGE_SENSOR_STATE_NO_DATA,
          HID_USAGE_SENSOR_STATE_INITIALIZING,
          HID_USAGE_SENSOR_STATE_ACCESS_DENIED,
          HID_USAGE_SENSOR_STATE_ERROR,
          HID_INPUT(Const_Arr_Abs),
HID_END_COLLECTION,
HID_USAGE_SENSOR_EVENT,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_8(16),
HID_REPORT_SIZE(8),
HID_REPORT_COUNT(1),
HID_COLLECTION(Logical),
          HID_USAGE_SENSOR_EVENT_UNKNOWN,
          HID_USAGE_SENSOR_EVENT_STATE_CHANGED,
          HID_USAGE_SENSOR_EVENT_PROPERTY_CHANGED,
          HID_USAGE_SENSOR_EVENT_DATA_UPDATED,
          HID_USAGE_SENSOR_EVENT_POLL_RESPONSE,
          HID_USAGE_SENSOR_EVENT_CHANGE_SENSITIVITY,
          HID_USAGE_SENSOR_EVENT_MAX_REACHED,
          HID_USAGE_SENSOR_EVENT_MIN_REACHED,
          HID_USAGE_SENSOR_EVENT_HIGH_THRESHOLD_CROSS_UPWARD,
          HID_USAGE_SENSOR_EVENT_HIGH_THRESHOLD_CROSS_DOWNWARD,
          HID_USAGE_SENSOR_EVENT_LOW_THRESHOLD_CROSS_UPWARD,
          HID_USAGE_SENSOR_EVENT_LOW_THRESHOLD_CROSS_DOWNWARD,
          HID_USAGE_SENSOR_EVENT_ZERO_THRESHOLD_CROSS_UPWARD,
          HID_USAGE_SENSOR_EVENT_ZERO_THRESHOLD_CROSS_DOWNWARD,
          HID_USAGE_SENSOR_EVENT_PERIOD_EXCEEDED,
          HID_USAGE_SENSOR_EVENT_FREQUENCY_EXCEEDED,
          HID_USAGE_SENSOR_EVENT_COMPLEX_TRIGGER,
```

```
                    HID_INPUT(Const_Arr_Abs),
          HID_END_COLLECTION,
          HID_USAGE_SENSOR_DATA_MOTION_ACCELERATION_X_AXIS,
          HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
          HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
          HID_REPORT_SIZE(16),
          HID_REPORT_COUNT(1),
          HID_UNIT_EXPONENT(0x0E), // scale default unit Gs to “centi-Gs” to provide 2 digits past Gs decimal point
          HID_INPUT(Const_Var_Abs),

          HID_END_COLLECTION
};

// 2D Accelerometer
const unsigned char accel2_report_descriptor[] = {
          HID_USAGE_PAGE_SENSOR,
          HID_USAGE_SENSOR_TYPE_MOTION_ACCELEROMETER_2D,
          HID_COLLECTION(Physical),

          //feature reports (xmit/receive)
          HID_USAGE_PAGE_SENSOR,
          HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE,
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_8(5),
          HID_REPORT_SIZE(8),
          HID_REPORT_COUNT(1),
          HID_COLLECTION(Logical),
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_NO_EVENTS,
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS,
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_THRESHOLD_EVENTS,
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_NO_EVENTS_WAKE,
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS_WAKE,
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_THRESHOLD_EVENTS_WAKE,
                    HID_FEATURE(Data_Arr_Abs),
          HID_END_COLLECTION,
          HID_USAGE_SENSOR_PROPERTY_SENSOR_STATUS,
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_32(0xFF,0xFF,0xFF,0xFF),
          HID_REPORT_SIZE(32),
          HID_REPORT_COUNT(1),
          HID_FEATURE(Data_Var_Abs), // up to VT_UI4 worth of status info
          HID_USAGE_SENSOR_PROPERTY_REPORT_INTERVAL,
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_32(0xFF,0xFF,0xFF,0xFF),
          HID_REPORT_SIZE(32),
          HID_REPORT_COUNT(1),
          // HID_USAGE_SENSOR_UNITS_MILLISECOND,
          HID_UNIT_EXPONENT(0),
          HID_FEATURE(Data_Var_Abs),
          HID_USAGE_SENSOR_PROPERTY_SENSOR_CONNECTION_TYPE,  // NAry
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_8(2),
          HID_REPORT_SIZE(8),
          HID_REPORT_COUNT(1),
          HID_COLLECTION(Logical),
                    HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_INTEGRATED,
                    HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_ATTACHED,
                    HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_EXTERNAL,
                    HID_FEATURE(Const_Arr_Abs),
          HID_END_COLLECTION,
          HID_USAGE_SENSOR_PROPERTY_CHANGE_SENSITIVITY_ABS,
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_16(0xFF,0xFF),
          HID_REPORT_SIZE(16),
          HID_REPORT_COUNT(1),
          // HID_USAGE_SENSOR_UNITS_G,
          HID_UNIT_EXPONENT(0x0E),  // scale default unit Gs to “centi-Gs” to provide 2 digits past Gs decimal point
          HID_FEATURE(Data_Var_Abs),
          HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_MOTION_ACCELERATION,HID_USAGE_SENSOR_DATA_MOD_MAX),
          HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
          HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
          HID_REPORT_SIZE(16),
          HID_REPORT_COUNT(1),
          // HID_USAGE_SENSOR_UNITS_G,
          HID_UNIT_EXPONENT(0x0E),  // scale default unit Gs to “centi-Gs” to provide 2 digits past Gs decimal point
          HID_FEATURE(Data_Var_Abs),
          HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_MOTION_ACCELERATION,HID_USAGE_SENSOR_DATA_MOD_MIN),
          HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
          HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
          HID_REPORT_SIZE(16),
          HID_REPORT_COUNT(1),
          // HID_USAGE_SENSOR_UNITS_G,
          HID_UNIT_EXPONENT(0x0E),  // scale default unit Gs to “centi-Gs” to provide 2 digits past Gs decimal point
          HID_FEATURE(Data_Var_Abs),

          //input reports (transmit)
          HID_USAGE_PAGE_SENSOR,
          HID_USAGE_SENSOR_STATE,
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_8(6),
          HID_REPORT_SIZE(8),
          HID_REPORT_COUNT(1),
          HID_COLLECTION(Logical),
                    HID_USAGE_SENSOR_STATE_UNKNOWN,
                    HID_USAGE_SENSOR_STATE_READY,
```

```
                    HID_USAGE_SENSOR_STATE_NOT_AVAILABLE,
                    HID_USAGE_SENSOR_STATE_NO_DATA,
                    HID_USAGE_SENSOR_STATE_INITIALIZING,
                    HID_USAGE_SENSOR_STATE_ACCESS_DENIED,
                    HID_USAGE_SENSOR_STATE_ERROR,
                    HID_INPUT(Const_Arr_Abs),
          HID_END_COLLECTION,
          HID_USAGE_SENSOR_EVENT,
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_8(16),
          HID_REPORT_SIZE(8),
          HID_REPORT_COUNT(1),
          HID_COLLECTION(Logical),
                    HID_USAGE_SENSOR_EVENT_UNKNOWN,
                    HID_USAGE_SENSOR_EVENT_STATE_CHANGED,
                    HID_USAGE_SENSOR_EVENT_PROPERTY_CHANGED,
                    HID_USAGE_SENSOR_EVENT_DATA_UPDATED,
                    HID_USAGE_SENSOR_EVENT_POLL_RESPONSE,
                    HID_USAGE_SENSOR_EVENT_CHANGE_SENSITIVITY,
                    HID_USAGE_SENSOR_EVENT_MAX_REACHED,
                    HID_USAGE_SENSOR_EVENT_MIN_REACHED,
                    HID_USAGE_SENSOR_EVENT_HIGH_THRESHOLD_CROSS_UPWARD,
                    HID_USAGE_SENSOR_EVENT_HIGH_THRESHOLD_CROSS_DOWNWARD,
                    HID_USAGE_SENSOR_EVENT_LOW_THRESHOLD_CROSS_UPWARD,
                    HID_USAGE_SENSOR_EVENT_LOW_THRESHOLD_CROSS_DOWNWARD,
                    HID_USAGE_SENSOR_EVENT_ZERO_THRESHOLD_CROSS_UPWARD,
                    HID_USAGE_SENSOR_EVENT_ZERO_THRESHOLD_CROSS_DOWNWARD,
                    HID_USAGE_SENSOR_EVENT_PERIOD_EXCEEDED,
                    HID_USAGE_SENSOR_EVENT_FREQUENCY_EXCEEDED,
                    HID_USAGE_SENSOR_EVENT_COMPLEX_TRIGGER,
                    HID_INPUT(Const_Arr_Abs),
          HID_END_COLLECTION,
          HID_USAGE_SENSOR_DATA_MOTION_ACCELERATION_X_AXIS,
          HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
          HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
          HID_REPORT_SIZE(16),
          HID_REPORT_COUNT(1),
          // HID_USAGE_SENSOR_UNITS_G,
          HID_UNIT_EXPONENT(0x0E),  // scale default unit Gs to “centi-Gs” to provide 2 digits past Gs decimal point
          HID_INPUT(Const_Var_Abs),
          HID_USAGE_SENSOR_DATA_MOTION_ACCELERATION_Y_AXIS,
          HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
          HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
          HID_REPORT_SIZE(16),
          HID_REPORT_COUNT(1),
          // HID_USAGE_SENSOR_UNITS_G,
          HID_UNIT_EXPONENT(0x0E),  // scale default unit Gs to “centi-Gs” to provide 2 digits past Gs decimal point
          HID_INPUT(Const_Var_Abs),

          HID_END_COLLECTION
};

// 3D Accelerometer
const unsigned char accel3_report_descriptor[] = {
          HID_USAGE_PAGE_SENSOR,
          HID_USAGE_SENSOR_TYPE_MOTION_ACCELEROMETER_3D,
          HID_COLLECTION(Physical),

          //feature reports (xmit/receive)
          HID_USAGE_PAGE_SENSOR,
          HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE,
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_8(5),
          HID_REPORT_SIZE(8),
          HID_REPORT_COUNT(1),
          HID_COLLECTION(Logical),
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_NO_EVENTS,
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS,
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_THRESHOLD_EVENTS,
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_NO_EVENTS_WAKE,
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS_WAKE,
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_THRESHOLD_EVENTS_WAKE,
                    HID_FEATURE(Data_Arr_Abs),
          HID_END_COLLECTION,
          HID_USAGE_SENSOR_PROPERTY_SENSOR_STATUS,
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_32(0xFF,0xFF,0xFF,0xFF),
          HID_REPORT_SIZE(32),
          HID_REPORT_COUNT(1),
          HID_FEATURE(Data_Var_Abs), // up to VT_UI4 worth of status info
          HID_USAGE_SENSOR_PROPERTY_REPORT_INTERVAL,
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_32(0xFF,0xFF,0xFF,0xFF),
          HID_REPORT_SIZE(32),
          HID_REPORT_COUNT(1),
          // HID_USAGE_SENSOR_UNITS_MILLISECOND,
          HID_UNIT_EXPONENT(0),
          HID_FEATURE(Data_Var_Abs),
          HID_USAGE_SENSOR_PROPERTY_SENSOR_CONNECTION_TYPE,  // NAry
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_8(2),
          HID_REPORT_SIZE(8),
          HID_REPORT_COUNT(1),
          HID_COLLECTION(Logical),
                    HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_INTEGRATED,
```

```
          HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_ATTACHED,
          HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_EXTERNAL,
          HID_FEATURE(Const_Arr_Abs),
HID_END_COLLECTION,
HID_USAGE_SENSOR_PROPERTY_CHANGE_SENSITIVITY_ABS,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_16(0xFF,0xFF),
HID_REPORT_SIZE(16),
HID_REPORT_COUNT(1),
// HID_USAGE_SENSOR_UNITS_G,
HID_UNIT_EXPONENT(0x0E),  // scale default unit Gs to “centi-Gs” to provide 2 digits past Gs decimal point
HID_FEATURE(Data_Var_Abs),
HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_MOTION_ACCELERATION,HID_USAGE_SENSOR_DATA_MOD_MAX),
HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
HID_REPORT_SIZE(16),
HID_REPORT_COUNT(1),
// HID_USAGE_SENSOR_UNITS_G,
HID_UNIT_EXPONENT(0x0E),  // scale default unit Gs to “centi-Gs” to provide 2 digits past Gs decimal point
HID_FEATURE(Data_Var_Abs),
HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_MOTION_ACCELERATION,HID_USAGE_SENSOR_DATA_MOD_MIN),
HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
HID_REPORT_SIZE(16),
HID_REPORT_COUNT(1),
// HID_USAGE_SENSOR_UNITS_G,
HID_UNIT_EXPONENT(0x0E),  // scale default unit Gs to “centi-Gs” to provide 2 digits past Gs decimal point
HID_FEATURE(Data_Var_Abs),

//input reports (transmit)
HID_USAGE_PAGE_SENSOR,
HID_USAGE_SENSOR_STATE,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_8(6),
HID_REPORT_SIZE(8),
HID_REPORT_COUNT(1),
HID_COLLECTION(Logical),
          HID_USAGE_SENSOR_STATE_UNKNOWN,
          HID_USAGE_SENSOR_STATE_READY,
          HID_USAGE_SENSOR_STATE_NOT_AVAILABLE,
          HID_USAGE_SENSOR_STATE_NO_DATA,
          HID_USAGE_SENSOR_STATE_INITIALIZING,
          HID_USAGE_SENSOR_STATE_ACCESS_DENIED,
          HID_USAGE_SENSOR_STATE_ERROR,
          HID_INPUT(Const_Arr_Abs),
HID_END_COLLECTION,
HID_USAGE_SENSOR_EVENT,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_8(16),
HID_REPORT_SIZE(8),
HID_REPORT_COUNT(1),
HID_COLLECTION(Logical),
          HID_USAGE_SENSOR_EVENT_UNKNOWN,
          HID_USAGE_SENSOR_EVENT_STATE_CHANGED,
          HID_USAGE_SENSOR_EVENT_PROPERTY_CHANGED,
          HID_USAGE_SENSOR_EVENT_DATA_UPDATED,
          HID_USAGE_SENSOR_EVENT_POLL_RESPONSE,
          HID_USAGE_SENSOR_EVENT_CHANGE_SENSITIVITY,
          HID_USAGE_SENSOR_EVENT_MAX_REACHED,
          HID_USAGE_SENSOR_EVENT_MIN_REACHED,
          HID_USAGE_SENSOR_EVENT_HIGH_THRESHOLD_CROSS_UPWARD,
          HID_USAGE_SENSOR_EVENT_HIGH_THRESHOLD_CROSS_DOWNWARD,
          HID_USAGE_SENSOR_EVENT_LOW_THRESHOLD_CROSS_UPWARD,
          HID_USAGE_SENSOR_EVENT_LOW_THRESHOLD_CROSS_DOWNWARD,
          HID_USAGE_SENSOR_EVENT_ZERO_THRESHOLD_CROSS_UPWARD,
          HID_USAGE_SENSOR_EVENT_ZERO_THRESHOLD_CROSS_DOWNWARD,
          HID_USAGE_SENSOR_EVENT_PERIOD_EXCEEDED,
          HID_USAGE_SENSOR_EVENT_FREQUENCY_EXCEEDED,
          HID_USAGE_SENSOR_EVENT_COMPLEX_TRIGGER,
          HID_INPUT(Const_Arr_Abs),
HID_END_COLLECTION,
HID_USAGE_SENSOR_DATA_MOTION_ACCELERATION_X_AXIS,
HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
HID_REPORT_SIZE(16),
HID_REPORT_COUNT(1),
// HID_USAGE_SENSOR_UNITS_G,
HID_UNIT_EXPONENT(0x0E),  // scale default unit Gs to “centi-Gs” to provide 2 digits past Gs decimal point
HID_INPUT(Const_Var_Abs),
HID_USAGE_SENSOR_DATA_MOTION_ACCELERATION_Y_AXIS,
HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
HID_REPORT_SIZE(16),
HID_REPORT_COUNT(1),
// HID_USAGE_SENSOR_UNITS_G,
HID_UNIT_EXPONENT(0x0E),  // scale default unit Gs to “centi-Gs” to provide 2 digits past Gs decimal point
HID_INPUT(Const_Var_Abs),
HID_USAGE_SENSOR_DATA_MOTION_ACCELERATION_Z_AXIS,
HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
HID_REPORT_SIZE(16),
HID_REPORT_COUNT(1),
// HID_USAGE_SENSOR_UNITS_G,
HID_UNIT_EXPONENT(0x0E),  // scale default unit Gs to “centi-Gs” to provide 2 digits past Gs decimal point
HID_INPUT(Const_Var_Abs),
```

```
        HID_END_COLLECTION
};
```

4.3.16 Motion: Gyrometer

```
// For reference: Complete HID report descriptor

// 1D Gyrometer
const unsigned char gyro1_report_descriptor[] = {
        HID_USAGE_PAGE_SENSOR,
        HID_USAGE_SENSOR_TYPE_MOTION_GYROMETER_1D,
        HID_COLLECTION(Physical),

        //feature reports (xmit/receive)
        HID_USAGE_PAGE_SENSOR,
        HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_8(5),
        HID_REPORT_SIZE(8),
        HID_REPORT_COUNT(1),
        HID_COLLECTION(Logical),
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_NO_EVENTS,
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS,
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_THRESHOLD_EVENTS,
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_NO_EVENTS_WAKE,
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS_WAKE,
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_THRESHOLD_EVENTS_WAKE,
                HID_FEATURE(Data_Arr_Abs),
        HID_END_COLLECTION,
        HID_USAGE_SENSOR_PROPERTY_SENSOR_STATUS,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_32(0xFF,0xFF,0xFF,0xFF),
        HID_REPORT_SIZE(32),
        HID_REPORT_COUNT(1),
        HID_FEATURE(Data_Var_Abs), // up to VT_UI4 worth of status info
        HID_USAGE_SENSOR_PROPERTY_REPORT_INTERVAL,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_32(0xFF,0xFF,0xFF,0xFF),
        HID_REPORT_SIZE(32),
        HID_REPORT_COUNT(1),
        // HID_USAGE_SENSOR_UNITS_MILLISECOND,
        HID_UNIT_EXPONENT(0),
        HID_FEATURE(Data_Var_Abs),
        HID_USAGE_SENSOR_PROPERTY_SENSOR_CONNECTION_TYPE,  // NAry
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_8(2),
        HID_REPORT_SIZE(8),
        HID_REPORT_COUNT(1),
        HID_COLLECTION(Logical),
                HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_INTEGRATED,
                HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_ATTACHED,
                HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_EXTERNAL,
                HID_FEATURE(Const_Arr_Abs),
        HID_END_COLLECTION,
        HID_USAGE_SENSOR_PROPERTY_CHANGE_SENSITIVITY_ABS,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_16(0xFF,0xFF),
        HID_REPORT_SIZE(16),
        HID_REPORT_COUNT(1),
        HID_USAGE_SENSOR_UNITS_DEGREES_PER_SECOND,
        HID_UNIT_EXPONENT(0x0E), // scale default unit to provide 2 digits past decimal point
        HID_FEATURE(Data_Var_Abs),
        HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_MOTION_ANGULAR_VELOCITY,HID_USAGE_SENSOR_DATA_MOD_MAX),
        HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
        HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
        HID_REPORT_SIZE(16),
        HID_REPORT_COUNT(1),
        HID_USAGE_SENSOR_UNITS_DEGREES_PER_SECOND,
        HID_UNIT_EXPONENT(0x0E), // scale default unit to provide 2 digits past decimal point
        HID_FEATURE(Data_Var_Abs),
        HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_MOTION_ANGULAR_VELOCITY,HID_USAGE_SENSOR_DATA_MOD_MIN),
        HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
        HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
        HID_REPORT_SIZE(16),
        HID_REPORT_COUNT(1),
        HID_USAGE_SENSOR_UNITS_DEGREES_PER_SECOND,
        HID_UNIT_EXPONENT(0x0E), // scale default unit to provide 2 digits past decimal point
        HID_FEATURE(Data_Var_Abs),

        //input reports (transmit)
        HID_USAGE_PAGE_SENSOR,
        HID_USAGE_SENSOR_STATE,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_8(6),
        HID_REPORT_SIZE(8),
        HID_REPORT_COUNT(1),
        HID_COLLECTION(Logical),
                HID_USAGE_SENSOR_STATE_UNKNOWN,
                HID_USAGE_SENSOR_STATE_READY,
                HID_USAGE_SENSOR_STATE_NOT_AVAILABLE,
                HID_USAGE_SENSOR_STATE_NO_DATA,
                HID_USAGE_SENSOR_STATE_INITIALIZING,
                HID_USAGE_SENSOR_STATE_ACCESS_DENIED,
```

```
                    HID_USAGE_SENSOR_STATE_ERROR,
                    HID_INPUT(Const_Arr_Abs),
          HID_END_COLLECTION,
          HID_USAGE_SENSOR_EVENT,
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_8(16),
          HID_REPORT_SIZE(8),
          HID_REPORT_COUNT(1),
          HID_COLLECTION(Logical),
                    HID_USAGE_SENSOR_EVENT_UNKNOWN,
                    HID_USAGE_SENSOR_EVENT_STATE_CHANGED,
                    HID_USAGE_SENSOR_EVENT_PROPERTY_CHANGED,
                    HID_USAGE_SENSOR_EVENT_DATA_UPDATED,
                    HID_USAGE_SENSOR_EVENT_POLL_RESPONSE,
                    HID_USAGE_SENSOR_EVENT_CHANGE_SENSITIVITY,
                    HID_USAGE_SENSOR_EVENT_MAX_REACHED,
                    HID_USAGE_SENSOR_EVENT_MIN_REACHED,
                    HID_USAGE_SENSOR_EVENT_HIGH_THRESHOLD_CROSS_UPWARD,
                    HID_USAGE_SENSOR_EVENT_HIGH_THRESHOLD_CROSS_DOWNWARD,
                    HID_USAGE_SENSOR_EVENT_LOW_THRESHOLD_CROSS_UPWARD,
                    HID_USAGE_SENSOR_EVENT_LOW_THRESHOLD_CROSS_DOWNWARD,
                    HID_USAGE_SENSOR_EVENT_ZERO_THRESHOLD_CROSS_UPWARD,
                    HID_USAGE_SENSOR_EVENT_ZERO_THRESHOLD_CROSS_DOWNWARD,
                    HID_USAGE_SENSOR_EVENT_PERIOD_EXCEEDED,
                    HID_USAGE_SENSOR_EVENT_FREQUENCY_EXCEEDED,
                    HID_USAGE_SENSOR_EVENT_COMPLEX_TRIGGER,
                    HID_INPUT(Const_Arr_Abs),
          HID_END_COLLECTION,
          HID_USAGE_SENSOR_DATA_MOTION_ANGULAR_VELOCITY_X_AXIS,
          HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
          HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
          HID_REPORT_SIZE(16),
          HID_REPORT_COUNT(1),
          HID_USAGE_SENSOR_UNITS_DEGREES_PER_SECOND,
          HID_UNIT_EXPONENT(0x0E), // scale default unit to provide 2 digits past decimal point
          HID_INPUT(Const_Var_Abs),

          HID_END_COLLECTION
};

// 2D Gyrometer
const unsigned char gyro2_report_descriptor[] = {
          HID_USAGE_PAGE_SENSOR,
          HID_USAGE_SENSOR_TYPE_MOTION_GYROMETER_2D,
          HID_COLLECTION(Physical),

          //feature reports (xmit/receive)
          HID_USAGE_PAGE_SENSOR,
          HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE,
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_8(5),
          HID_REPORT_SIZE(8),
          HID_REPORT_COUNT(1),
          HID_COLLECTION(Logical),
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_NO_EVENTS,
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS,
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_THRESHOLD_EVENTS,
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_NO_EVENTS_WAKE,
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS_WAKE,
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_THRESHOLD_EVENTS_WAKE,
                    HID_FEATURE(Data_Arr_Abs),
          HID_END_COLLECTION,
          HID_USAGE_SENSOR_PROPERTY_SENSOR_STATUS,
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_32(0xFF,0xFF,0xFF,0xFF),
          HID_REPORT_SIZE(32),
          HID_REPORT_COUNT(1),
          HID_FEATURE(Data_Var_Abs), // up to VT_UI4 worth of status info
          HID_USAGE_SENSOR_PROPERTY_REPORT_INTERVAL,
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_32(0xFF,0xFF,0xFF,0xFF),
          HID_REPORT_SIZE(32),
          HID_REPORT_COUNT(1),
          // HID_USAGE_SENSOR_UNITS_MILLISECOND,
          HID_UNIT_EXPONENT(0),
          HID_FEATURE(Data_Var_Abs),
          HID_USAGE_SENSOR_PROPERTY_SENSOR_CONNECTION_TYPE,  // NAry
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_8(2),
          HID_REPORT_SIZE(8),
          HID_REPORT_COUNT(1),
          HID_COLLECTION(Logical),
                    HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_INTEGRATED,
                    HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_ATTACHED,
                    HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_EXTERNAL,
                    HID_FEATURE(Const_Arr_Abs),
          HID_END_COLLECTION,
          HID_USAGE_SENSOR_PROPERTY_CHANGE_SENSITIVITY_ABS,
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_16(0xFF,0xFF),
          HID_REPORT_SIZE(16),
          HID_REPORT_COUNT(1),
          HID_USAGE_SENSOR_UNITS_DEGREES_PER_SECOND,
          HID_UNIT_EXPONENT(0x0E), // scale default unit to provide 2 digits past decimal point
          HID_FEATURE(Data_Var_Abs),
```

```
        HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_MOTION_ANGULAR_VELOCITY,HID_USAGE_SENSOR_DATA_MOD_MAX),
        HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
        HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
        HID_REPORT_SIZE(16),
        HID_REPORT_COUNT(1),
        HID_USAGE_SENSOR_UNITS_DEGREES_PER_SECOND,
        HID_UNIT_EXPONENT(0x0E), // scale default unit to provide 2 digits past decimal point
        HID_FEATURE(Data_Var_Abs),
        HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_MOTION_ANGULAR_VELOCITY,HID_USAGE_SENSOR_DATA_MOD_MIN),
        HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
        HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
        HID_REPORT_SIZE(16),
        HID_REPORT_COUNT(1),
        HID_USAGE_SENSOR_UNITS_DEGREES_PER_SECOND,
        HID_UNIT_EXPONENT(0x0E), // scale default unit to provide 2 digits past decimal point
        HID_FEATURE(Data_Var_Abs),

        //input reports (transmit)
        HID_USAGE_PAGE_SENSOR,
        HID_USAGE_SENSOR_STATE,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_8(6),
        HID_REPORT_SIZE(8),
        HID_REPORT_COUNT(1),
        HID_COLLECTION(Logical),
                HID_USAGE_SENSOR_STATE_UNKNOWN,
                HID_USAGE_SENSOR_STATE_READY,
                HID_USAGE_SENSOR_STATE_NOT_AVAILABLE,
                HID_USAGE_SENSOR_STATE_NO_DATA,
                HID_USAGE_SENSOR_STATE_INITIALIZING,
                HID_USAGE_SENSOR_STATE_ACCESS_DENIED,
                HID_USAGE_SENSOR_STATE_ERROR,
                HID_INPUT(Const_Arr_Abs),
        HID_END_COLLECTION,
        HID_USAGE_SENSOR_EVENT,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_8(16),
        HID_REPORT_SIZE(8),
        HID_REPORT_COUNT(1),
        HID_COLLECTION(Logical),
                HID_USAGE_SENSOR_EVENT_UNKNOWN,
                HID_USAGE_SENSOR_EVENT_STATE_CHANGED,
                HID_USAGE_SENSOR_EVENT_PROPERTY_CHANGED,
                HID_USAGE_SENSOR_EVENT_DATA_UPDATED,
                HID_USAGE_SENSOR_EVENT_POLL_RESPONSE,
                HID_USAGE_SENSOR_EVENT_CHANGE_SENSITIVITY,
                HID_USAGE_SENSOR_EVENT_MAX_REACHED,
                HID_USAGE_SENSOR_EVENT_MIN_REACHED,
                HID_USAGE_SENSOR_EVENT_HIGH_THRESHOLD_CROSS_UPWARD,
                HID_USAGE_SENSOR_EVENT_HIGH_THRESHOLD_CROSS_DOWNWARD,
                HID_USAGE_SENSOR_EVENT_LOW_THRESHOLD_CROSS_UPWARD,
                HID_USAGE_SENSOR_EVENT_LOW_THRESHOLD_CROSS_DOWNWARD,
                HID_USAGE_SENSOR_EVENT_ZERO_THRESHOLD_CROSS_UPWARD,
                HID_USAGE_SENSOR_EVENT_ZERO_THRESHOLD_CROSS_DOWNWARD,
                HID_USAGE_SENSOR_EVENT_PERIOD_EXCEEDED,
                HID_USAGE_SENSOR_EVENT_FREQUENCY_EXCEEDED,
                HID_USAGE_SENSOR_EVENT_COMPLEX_TRIGGER,
                HID_INPUT(Const_Arr_Abs),
        HID_END_COLLECTION,
        HID_USAGE_SENSOR_DATA_MOTION_ANGULAR_VELOCITY_X_AXIS,
        HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
        HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
        HID_REPORT_SIZE(16),
        HID_REPORT_COUNT(1),
        HID_USAGE_SENSOR_UNITS_DEGREES_PER_SECOND,
        HID_UNIT_EXPONENT(0x0E), // scale default unit to provide 2 digits past decimal point
        HID_INPUT(Const_Var_Abs),
        HID_USAGE_SENSOR_DATA_MOTION_ANGULAR_VELOCITY_Y_AXIS,
        HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
        HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
        HID_REPORT_SIZE(16),
        HID_REPORT_COUNT(1),
        HID_USAGE_SENSOR_UNITS_DEGREES_PER_SECOND,
        HID_UNIT_EXPONENT(0x0E), // scale default unit to provide 2 digits past decimal point
        HID_INPUT(Const_Var_Abs),

        HID_END_COLLECTION
};

// 3D Gyrometer
const unsigned char gyro3_report_descriptor[] = {
        HID_USAGE_PAGE_SENSOR,
        HID_USAGE_SENSOR_TYPE_MOTION_GYROMETER_3D,
        HID_COLLECTION(Physical),

        //feature reports (xmit/receive)
        HID_USAGE_PAGE_SENSOR,
        HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_8(5),
        HID_REPORT_SIZE(8),
        HID_REPORT_COUNT(1),
        HID_COLLECTION(Logical),
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_NO_EVENTS,
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS,
```

```
            HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_THRESHOLD_EVENTS,
            HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_NO_EVENTS_WAKE,
            HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS_WAKE,
            HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_THRESHOLD_EVENTS_WAKE,
            HID_FEATURE(Data_Arr_Abs),
HID_END_COLLECTION,
HID_USAGE_SENSOR_PROPERTY_SENSOR_STATUS,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_32(0xFF,0xFF,0xFF,0xFF),
HID_REPORT_SIZE(32),
HID_REPORT_COUNT(1),
HID_FEATURE(Data_Var_Abs), // up to VT_UI4 worth of status info
HID_USAGE_SENSOR_PROPERTY_REPORT_INTERVAL,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_32(0xFF,0xFF,0xFF,0xFF),
HID_REPORT_SIZE(32),
HID_REPORT_COUNT(1),
// HID_USAGE_SENSOR_UNITS_MILLISECOND,
HID_UNIT_EXPONENT(0),
HID_FEATURE(Data_Var_Abs),
HID_USAGE_SENSOR_PROPERTY_SENSOR_CONNECTION_TYPE,  // NAry
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_8(2),
HID_REPORT_SIZE(8),
HID_REPORT_COUNT(1),
HID_COLLECTION(Logical),
            HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_INTEGRATED,
            HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_ATTACHED,
            HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_EXTERNAL,
            HID_FEATURE(Const_Arr_Abs),
HID_END_COLLECTION,
HID_USAGE_SENSOR_PROPERTY_CHANGE_SENSITIVITY_ABS,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_16(0xFF,0xFF),
HID_REPORT_SIZE(16),
HID_REPORT_COUNT(1),
HID_USAGE_SENSOR_UNITS_DEGREES_PER_SECOND,
HID_UNIT_EXPONENT(0x0E), // scale default unit to provide 2 digits past decimal point
HID_FEATURE(Data_Var_Abs),
HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_MOTION_ANGULAR_VELOCITY,HID_USAGE_SENSOR_DATA_MOD_MAX),
HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
HID_REPORT_SIZE(16),
HID_REPORT_COUNT(1),
HID_USAGE_SENSOR_UNITS_DEGREES_PER_SECOND,
HID_UNIT_EXPONENT(0x0E), // scale default unit to provide 2 digits past decimal point
HID_FEATURE(Data_Var_Abs),
HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_MOTION_ANGULAR_VELOCITY,HID_USAGE_SENSOR_DATA_MOD_MIN),
HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
HID_REPORT_SIZE(16),
HID_REPORT_COUNT(1),
HID_USAGE_SENSOR_UNITS_DEGREES_PER_SECOND,
HID_UNIT_EXPONENT(0x0E), // scale default unit to provide 2 digits past decimal point
HID_FEATURE(Data_Var_Abs),

//input reports (transmit)
HID_USAGE_PAGE_SENSOR,
HID_USAGE_SENSOR_STATE,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_8(6),
HID_REPORT_SIZE(8),
HID_REPORT_COUNT(1),
HID_COLLECTION(Logical),
            HID_USAGE_SENSOR_STATE_UNKNOWN,
            HID_USAGE_SENSOR_STATE_READY,
            HID_USAGE_SENSOR_STATE_NOT_AVAILABLE,
            HID_USAGE_SENSOR_STATE_NO_DATA,
            HID_USAGE_SENSOR_STATE_INITIALIZING,
            HID_USAGE_SENSOR_STATE_ACCESS_DENIED,
            HID_USAGE_SENSOR_STATE_ERROR,
            HID_INPUT(Const_Arr_Abs),
HID_END_COLLECTION,
HID_USAGE_SENSOR_EVENT,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_8(16),
HID_REPORT_SIZE(8),
HID_REPORT_COUNT(1),
HID_COLLECTION(Logical),
            HID_USAGE_SENSOR_EVENT_UNKNOWN,
            HID_USAGE_SENSOR_EVENT_STATE_CHANGED,
            HID_USAGE_SENSOR_EVENT_PROPERTY_CHANGED,
            HID_USAGE_SENSOR_EVENT_DATA_UPDATED,
            HID_USAGE_SENSOR_EVENT_POLL_RESPONSE,
            HID_USAGE_SENSOR_EVENT_CHANGE_SENSITIVITY,
            HID_USAGE_SENSOR_EVENT_MAX_REACHED,
            HID_USAGE_SENSOR_EVENT_MIN_REACHED,
            HID_USAGE_SENSOR_EVENT_HIGH_THRESHOLD_CROSS_UPWARD,
            HID_USAGE_SENSOR_EVENT_HIGH_THRESHOLD_CROSS_DOWNWARD,
            HID_USAGE_SENSOR_EVENT_LOW_THRESHOLD_CROSS_UPWARD,
            HID_USAGE_SENSOR_EVENT_LOW_THRESHOLD_CROSS_DOWNWARD,
            HID_USAGE_SENSOR_EVENT_ZERO_THRESHOLD_CROSS_UPWARD,
            HID_USAGE_SENSOR_EVENT_ZERO_THRESHOLD_CROSS_DOWNWARD,
            HID_USAGE_SENSOR_EVENT_PERIOD_EXCEEDED,
            HID_USAGE_SENSOR_EVENT_FREQUENCY_EXCEEDED,
```

```
                    HID_USAGE_SENSOR_EVENT_COMPLEX_TRIGGER,
                    HID_INPUT(Const_Arr_Abs),
          HID_END_COLLECTION,
          HID_USAGE_SENSOR_DATA_MOTION_ANGULAR_VELOCITY_X_AXIS,
          HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
          HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
          HID_REPORT_SIZE(16),
          HID_REPORT_COUNT(1),
          HID_USAGE_SENSOR_UNITS_DEGREES_PER_SECOND,
          HID_UNIT_EXPONENT(0x0E), // scale default unit to provide 2 digits past decimal point
          HID_INPUT(Const_Var_Abs),
          HID_USAGE_SENSOR_DATA_MOTION_ANGULAR_VELOCITY_Y_AXIS,
          HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
          HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
          HID_REPORT_SIZE(16),
          HID_REPORT_COUNT(1),
          HID_USAGE_SENSOR_UNITS_DEGREES_PER_SECOND,
          HID_UNIT_EXPONENT(0x0E), // scale default unit to provide 2 digits past decimal point
          HID_INPUT(Const_Var_Abs),
          HID_USAGE_SENSOR_DATA_MOTION_ANGULAR_VELOCITY_Z_AXIS,
          HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
          HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
          HID_REPORT_SIZE(16),
          HID_REPORT_COUNT(1),
          HID_USAGE_SENSOR_UNITS_DEGREES_PER_SECOND,
          HID_UNIT_EXPONENT(0x0E), // scale default unit to provide 2 digits past decimal point
          HID_INPUT(Const_Var_Abs),

          HID_END_COLLECTION
};
```

4.3.17 Motion: Motion Detector

```
// For reference: Complete HID report descriptor

//Motion sensor
const unsigned char mot_report_descriptor[] = {
          HID_USAGE_PAGE_SENSOR,
          HID_USAGE_SENSOR_TYPE_MOTION_MOTION_DETECTOR,
          HID_COLLECTION(Physical),

          //feature reports (xmit/receive)
          HID_USAGE_PAGE_SENSOR,
          HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE,
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_8(5),
          HID_REPORT_SIZE(8),
          HID_REPORT_COUNT(1),
          HID_COLLECTION(Logical),
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_NO_EVENTS,
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS,
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_THRESHOLD_EVENTS,
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_NO_EVENTS_WAKE,
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS_WAKE,
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_THRESHOLD_EVENTS_WAKE,
                    HID_FEATURE(Data_Arr_Abs),
          HID_END_COLLECTION,
          HID_USAGE_SENSOR_PROPERTY_SENSOR_STATUS,
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_32(0xFF,0xFF,0xFF,0xFF),
          HID_REPORT_SIZE(32),
          HID_REPORT_COUNT(1),
          HID_FEATURE(Data_Var_Abs), // up to VT_UI4 worth of status info
          HID_USAGE_SENSOR_PROPERTY_REPORT_INTERVAL,
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_32(0xFF,0xFF,0xFF,0xFF),
          HID_REPORT_SIZE(32),
          HID_REPORT_COUNT(1),
          // HID_USAGE_SENSOR_UNITS_MILLISECOND,
          HID_UNIT_EXPONENT(0),
          HID_FEATURE(Data_Var_Abs),
          HID_USAGE_SENSOR_PROPERTY_SENSOR_CONNECTION_TYPE,  // NAry
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_8(2),
          HID_REPORT_SIZE(8),
          HID_REPORT_COUNT(1),
          HID_COLLECTION(Logical),
                    HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_INTEGRATED,
                    HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_ATTACHED,
                    HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_EXTERNAL,
                    HID_FEATURE(Const_Arr_Abs),
          HID_END_COLLECTION,
          HID_USAGE_SENSOR_PROPERTY_CHANGE_SENSITIVITY_ABS,
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_16(0xFF,0xFF),
          HID_REPORT_SIZE(16),
          HID_REPORT_COUNT(1),
          HID_USAGE_SENSOR_UNITS_NOT_SPECIFIED,
          HID_UNIT_EXPONENT(0x0E), // scale default unit to provide 2 digits past the decimal point
          HID_FEATURE(Data_Var_Abs),

          //input reports (transmit)
          HID_USAGE_PAGE_SENSOR,
          HID_USAGE_SENSOR_STATE,
```

```
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_8(6),
        HID_REPORT_SIZE(8),
        HID_REPORT_COUNT(1),
        HID_COLLECTION(Logical),
                HID_USAGE_SENSOR_STATE_UNKNOWN,
                HID_USAGE_SENSOR_STATE_READY,
                HID_USAGE_SENSOR_STATE_NOT_AVAILABLE,
                HID_USAGE_SENSOR_STATE_NO_DATA,
                HID_USAGE_SENSOR_STATE_INITIALIZING,
                HID_USAGE_SENSOR_STATE_ACCESS_DENIED,
                HID_USAGE_SENSOR_STATE_ERROR,
                HID_INPUT(Const_Arr_Abs),
        HID_END_COLLECTION,
        HID_USAGE_SENSOR_EVENT,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_8(16),
        HID_REPORT_SIZE(8),
        HID_REPORT_COUNT(1),
        HID_COLLECTION(Logical),
                HID_USAGE_SENSOR_EVENT_UNKNOWN,
                HID_USAGE_SENSOR_EVENT_STATE_CHANGED,
                HID_USAGE_SENSOR_EVENT_PROPERTY_CHANGED,
                HID_USAGE_SENSOR_EVENT_DATA_UPDATED,
                HID_USAGE_SENSOR_EVENT_POLL_RESPONSE,
                HID_USAGE_SENSOR_EVENT_CHANGE_SENSITIVITY,
                HID_USAGE_SENSOR_EVENT_MAX_REACHED,
                HID_USAGE_SENSOR_EVENT_MIN_REACHED,
                HID_USAGE_SENSOR_EVENT_HIGH_THRESHOLD_CROSS_UPWARD,
                HID_USAGE_SENSOR_EVENT_HIGH_THRESHOLD_CROSS_DOWNWARD,
                HID_USAGE_SENSOR_EVENT_LOW_THRESHOLD_CROSS_UPWARD,
                HID_USAGE_SENSOR_EVENT_LOW_THRESHOLD_CROSS_DOWNWARD,
                HID_USAGE_SENSOR_EVENT_ZERO_THRESHOLD_CROSS_UPWARD,
                HID_USAGE_SENSOR_EVENT_ZERO_THRESHOLD_CROSS_DOWNWARD,
                HID_USAGE_SENSOR_EVENT_PERIOD_EXCEEDED,
                HID_USAGE_SENSOR_EVENT_FREQUENCY_EXCEEDED,
                HID_USAGE_SENSOR_EVENT_COMPLEX_TRIGGER,
                HID_INPUT(Const_Arr_Abs),
        HID_END_COLLECTION,
        HID_USAGE_SENSOR_DATA_MOTION_STATE,
        HID_LOGICAL_MIN_8(0), // False = Still
        HID_LOGICAL_MAX_8(1), // True = In Motion
        HID_REPORT_SIZE(8),
        HID_REPORT_COUNT(1),
        HID_INPUT(Const_Var_Abs),
        HID_USAGE_SENSOR_DATA_MOTION_INTENSITY,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_8(100), // percent
        HID_REPORT_SIZE(8),
        HID_REPORT_COUNT(1),
        HID_INPUT(Const_Var_Abs),

        HID_END_COLLECTION
};
```

4.3.18 Orientation: Compass

```
// For reference: Complete HID report descriptor

// 1D Compass, like a “traditional” Boy Scouts compass
const unsigned char comp1_report_descriptor[] = {
        HID_USAGE_PAGE_SENSOR,
        HID_USAGE_SENSOR_TYPE_ORIENTATION_COMPASS_1D,
        HID_COLLECTION(Physical),

        //feature reports (xmit/receive)
        HID_USAGE_PAGE_SENSOR,
        HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_8(5),
        HID_REPORT_SIZE(8),
        HID_REPORT_COUNT(1),
        HID_COLLECTION(Logical),
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_NO_EVENTS,
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS,
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_THRESHOLD_EVENTS,
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_NO_EVENTS_WAKE,
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS_WAKE,
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_THRESHOLD_EVENTS_WAKE,
                HID_FEATURE(Data_Arr_Abs),
        HID_END_COLLECTION,
        HID_USAGE_SENSOR_PROPERTY_SENSOR_STATUS,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_32(0xFF,0xFF,0xFF,0xFF),
        HID_REPORT_SIZE(32),
        HID_REPORT_COUNT(1),
        HID_FEATURE(Data_Var_Abs), // up to VT_UI4 worth of status info
        HID_USAGE_SENSOR_PROPERTY_REPORT_INTERVAL,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_32(0xFF,0xFF,0xFF,0xFF),
        HID_REPORT_SIZE(32),
        HID_REPORT_COUNT(1),
        // HID_USAGE_SENSOR_UNITS_MILLISECOND,
        HID_UNIT_EXPONENT(0),
```

```
        HID_FEATURE(Data_Var_Abs),
        HID_USAGE_SENSOR_PROPERTY_SENSOR_CONNECTION_TYPE,  // NAry
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_8(2),
        HID_REPORT_SIZE(8),
        HID_REPORT_COUNT(1),
        HID_COLLECTION(Logical),
                HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_INTEGRATED,
                HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_ATTACHED,
                HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_EXTERNAL,
                HID_FEATURE(Const_Arr_Abs),
        HID_END_COLLECTION,
        HID_USAGE_SENSOR_PROPERTY_CHANGE_SENSITIVITY_ABS,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_16(0xFF,0xFF),
        HID_REPORT_SIZE(16),
        HID_REPORT_COUNT(1),
        HID_USAGE_SENSOR_UNITS_DEGREES,
        HID_UNIT_EXPONENT(0x0E), // scale default unit to provide 2 digits past decimal point
        HID_FEATURE(Data_Var_Abs),
        HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_ORIENTATION_MAGNETIC_HEADING,HID_USAGE_SENSOR_DATA_MOD_MAX),
        HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
        HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
        HID_REPORT_SIZE(16),
        HID_REPORT_COUNT(1),
        HID_USAGE_SENSOR_UNITS_DEGREES,
        HID_UNIT_EXPONENT(0x0F), // scale default unit to provide 1 digit past decimal point
        HID_FEATURE(Data_Var_Abs),
        HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_ORIENTATION_MAGNETIC_HEADING,HID_USAGE_SENSOR_DATA_MOD_MIN),
        HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
        HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
        HID_REPORT_SIZE(16),
        HID_REPORT_COUNT(1),
        HID_USAGE_SENSOR_UNITS_DEGREES,
        HID_UNIT_EXPONENT(0x0F), // scale default unit to provide 1 digit past decimal point
        HID_FEATURE(Data_Var_Abs),

        //input reports (transmit)
        HID_USAGE_PAGE_SENSOR,
        HID_USAGE_SENSOR_STATE,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_8(6),
        HID_REPORT_SIZE(8),
        HID_REPORT_COUNT(1),
        HID_COLLECTION(Logical),
                HID_USAGE_SENSOR_STATE_UNKNOWN,
                HID_USAGE_SENSOR_STATE_READY,
                HID_USAGE_SENSOR_STATE_NOT_AVAILABLE,
                HID_USAGE_SENSOR_STATE_NO_DATA,
                HID_USAGE_SENSOR_STATE_INITIALIZING,
                HID_USAGE_SENSOR_STATE_ACCESS_DENIED,
                HID_USAGE_SENSOR_STATE_ERROR,
                HID_INPUT(Const_Arr_Abs),
        HID_END_COLLECTION,
        HID_USAGE_SENSOR_EVENT,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_8(16),
        HID_REPORT_SIZE(8),
        HID_REPORT_COUNT(1),
        HID_COLLECTION(Logical),
                HID_USAGE_SENSOR_EVENT_UNKNOWN,
                HID_USAGE_SENSOR_EVENT_STATE_CHANGED,
                HID_USAGE_SENSOR_EVENT_PROPERTY_CHANGED,
                HID_USAGE_SENSOR_EVENT_DATA_UPDATED,
                HID_USAGE_SENSOR_EVENT_POLL_RESPONSE,
                HID_USAGE_SENSOR_EVENT_CHANGE_SENSITIVITY,
                HID_USAGE_SENSOR_EVENT_MAX_REACHED,
                HID_USAGE_SENSOR_EVENT_MIN_REACHED,
                HID_USAGE_SENSOR_EVENT_HIGH_THRESHOLD_CROSS_UPWARD,
                HID_USAGE_SENSOR_EVENT_HIGH_THRESHOLD_CROSS_DOWNWARD,
                HID_USAGE_SENSOR_EVENT_LOW_THRESHOLD_CROSS_UPWARD,
                HID_USAGE_SENSOR_EVENT_LOW_THRESHOLD_CROSS_DOWNWARD,
                HID_USAGE_SENSOR_EVENT_ZERO_THRESHOLD_CROSS_UPWARD,
                HID_USAGE_SENSOR_EVENT_ZERO_THRESHOLD_CROSS_DOWNWARD,
                HID_USAGE_SENSOR_EVENT_PERIOD_EXCEEDED,
                HID_USAGE_SENSOR_EVENT_FREQUENCY_EXCEEDED,
                HID_USAGE_SENSOR_EVENT_COMPLEX_TRIGGER,
                HID_INPUT(Const_Arr_Abs),
        HID_END_COLLECTION,
        HID_USAGE_SENSOR_DATA_ORIENTATION_HEADING_MAGNETIC_NORTH,
        HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
        HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
        HID_REPORT_SIZE(16),
        HID_REPORT_COUNT(1),
        HID_USAGE_SENSOR_UNITS_DEGREES,
        HID_UNIT_EXPONENT(0x0F), // scale default unit to provide 1 digit past decimal point
        HID_INPUT(Const_Var_Abs),

        HID_END_COLLECTION
};

// 3D Compass, a 3-axis flux magnetometer
const unsigned char comp3_report_descriptor[] = {
        HID_USAGE_PAGE_SENSOR,
        HID_USAGE_SENSOR_TYPE_ORIENTATION_COMPASS_3D,
```

```
HID_COLLECTION(Physical),

//feature reports (xmit/receive)
HID_USAGE_PAGE_SENSOR,
HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_8(5),
HID_REPORT_SIZE(8),
HID_REPORT_COUNT(1),
HID_COLLECTION(Logical),
          HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_NO_EVENTS,
          HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS,
          HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_THRESHOLD_EVENTS,
          HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_NO_EVENTS_WAKE,
          HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS_WAKE,
          HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_THRESHOLD_EVENTS_WAKE,
          HID_FEATURE(Data_Arr_Abs),
HID_END_COLLECTION,
HID_USAGE_SENSOR_PROPERTY_SENSOR_STATUS,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_32(0xFF,0xFF,0xFF,0xFF),
HID_REPORT_SIZE(32),
HID_REPORT_COUNT(1),
HID_FEATURE(Data_Var_Abs), // up to VT_UI4 worth of status info
HID_USAGE_SENSOR_PROPERTY_REPORT_INTERVAL,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_32(0xFF,0xFF,0xFF,0xFF),
HID_REPORT_SIZE(32),
HID_REPORT_COUNT(1),
// HID_USAGE_SENSOR_UNITS_MILLISECOND,
HID_UNIT_EXPONENT(0),
HID_FEATURE(Data_Var_Abs),
HID_USAGE_SENSOR_PROPERTY_SENSOR_CONNECTION_TYPE,  // NAry
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_8(2),
HID_REPORT_SIZE(8),
HID_REPORT_COUNT(1),
HID_COLLECTION(Logical),
          HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_INTEGRATED,
          HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_ATTACHED,
          HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_EXTERNAL,
          HID_FEATURE(Const_Arr_Abs),
HID_END_COLLECTION,
HID_USAGE_SENSOR_PROPERTY_CHANGE_SENSITIVITY_ABS,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_16(0xFF,0xFF),
HID_REPORT_SIZE(16),
HID_REPORT_COUNT(1),
HID_USAGE_SENSOR_UNITS_DEGREES,
HID_UNIT_EXPONENT(0x0E), // scale default unit to provide 2 digits past decimal point
HID_FEATURE(Data_Var_Abs),
HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_ORIENTATION_MAGNETIC_HEADING,HID_USAGE_SENSOR_DATA_MOD_MAX),
HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
HID_REPORT_SIZE(16),
HID_REPORT_COUNT(1),
HID_USAGE_SENSOR_UNITS_DEGREES,
HID_UNIT_EXPONENT(0x0F), // scale default unit to provide 1 digit past decimal point
HID_FEATURE(Data_Var_Abs),
HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_ORIENTATION_MAGNETIC_HEADING,HID_USAGE_SENSOR_DATA_MOD_MIN),
HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
HID_REPORT_SIZE(16),
HID_REPORT_COUNT(1),
HID_USAGE_SENSOR_UNITS_DEGREES,
HID_UNIT_EXPONENT(0x0F), // scale default unit to provide 1 digit past decimal point
HID_FEATURE(Data_Var_Abs),

//input reports (transmit)
HID_USAGE_PAGE_SENSOR,
HID_USAGE_SENSOR_STATE,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_8(6),
HID_REPORT_SIZE(8),
HID_REPORT_COUNT(1),
HID_COLLECTION(Logical),
          HID_USAGE_SENSOR_STATE_UNKNOWN,
          HID_USAGE_SENSOR_STATE_READY,
          HID_USAGE_SENSOR_STATE_NOT_AVAILABLE,
          HID_USAGE_SENSOR_STATE_NO_DATA,
          HID_USAGE_SENSOR_STATE_INITIALIZING,
          HID_USAGE_SENSOR_STATE_ACCESS_DENIED,
          HID_USAGE_SENSOR_STATE_ERROR,
          HID_INPUT(Const_Arr_Abs),
HID_END_COLLECTION,
HID_USAGE_SENSOR_EVENT,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_8(16),
HID_REPORT_SIZE(8),
HID_REPORT_COUNT(1),
HID_COLLECTION(Logical),
          HID_USAGE_SENSOR_EVENT_UNKNOWN,
          HID_USAGE_SENSOR_EVENT_STATE_CHANGED,
          HID_USAGE_SENSOR_EVENT_PROPERTY_CHANGED,
          HID_USAGE_SENSOR_EVENT_DATA_UPDATED,
```

```
                    HID_USAGE_SENSOR_EVENT_POLL_RESPONSE,
                    HID_USAGE_SENSOR_EVENT_CHANGE_SENSITIVITY,
                    HID_USAGE_SENSOR_EVENT_MAX_REACHED,
                    HID_USAGE_SENSOR_EVENT_MIN_REACHED,
                    HID_USAGE_SENSOR_EVENT_HIGH_THRESHOLD_CROSS_UPWARD,
                    HID_USAGE_SENSOR_EVENT_HIGH_THRESHOLD_CROSS_DOWNWARD,
                    HID_USAGE_SENSOR_EVENT_LOW_THRESHOLD_CROSS_UPWARD,
                    HID_USAGE_SENSOR_EVENT_LOW_THRESHOLD_CROSS_DOWNWARD,
                    HID_USAGE_SENSOR_EVENT_ZERO_THRESHOLD_CROSS_UPWARD,
                    HID_USAGE_SENSOR_EVENT_ZERO_THRESHOLD_CROSS_DOWNWARD,
                    HID_USAGE_SENSOR_EVENT_PERIOD_EXCEEDED,
                    HID_USAGE_SENSOR_EVENT_FREQUENCY_EXCEEDED,
                    HID_USAGE_SENSOR_EVENT_COMPLEX_TRIGGER,
                    HID_INPUT(Const_Arr_Abs),
          HID_END_COLLECTION,
          HID_USAGE_SENSOR_DATA_ORIENTATION_MAGNETIC_FLUX_X,
          HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
          HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
          HID_REPORT_SIZE(16),
          HID_REPORT_COUNT(1),
          HID_USAGE_SENSOR_UNITS_GAUSS,
          HID_UNIT_EXPONENT(0x0D), // scale default unit to “milliGauss”; provide 3 digits past decimal point
          HID_INPUT(Const_Var_Abs),
          HID_USAGE_SENSOR_DATA_ORIENTATION_MAGNETIC_FLUX_Y,
          HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
          HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
          HID_REPORT_SIZE(16),
          HID_REPORT_COUNT(1),
          HID_USAGE_SENSOR_UNITS_GAUSS,
          HID_UNIT_EXPONENT(0x0D), // scale default unit to “milliGauss”; provide 3 digits past decimal point
          HID_INPUT(Const_Var_Abs),
          HID_USAGE_SENSOR_DATA_ORIENTATION_MAGNETIC_FLUX_Z,
          HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
          HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
          HID_REPORT_SIZE(16),
          HID_REPORT_COUNT(1),
          HID_USAGE_SENSOR_UNITS_GAUSS,
          HID_UNIT_EXPONENT(0x0D), // scale default unit to “milliGauss”; provide 3 digits past decimal point
          HID_INPUT(Const_Var_Abs),
          HID_USAGE_SENSOR_DATA_ORIENTATION_COMPENSATED_MAGNETIC_NORTH,
          HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
          HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
          HID_REPORT_SIZE(16),
          HID_REPORT_COUNT(1),
          HID_USAGE_SENSOR_UNITS_DEGREES,
          HID_UNIT_EXPONENT(0x0F), // scale default unit to provide 1 digit past decimal point
          HID_INPUT(Const_Var_Abs),
          HID_USAGE_SENSOR_DATA_ORIENTATION_COMPENSATED_TRUE_NORTH,
          HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
          HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
          HID_REPORT_SIZE(16),
          HID_REPORT_COUNT(1),
          HID_USAGE_SENSOR_UNITS_DEGREES,
          HID_UNIT_EXPONENT(0x0F), // scale default unit to provide 1 digit past decimal point
          HID_INPUT(Const_Var_Abs),
          HID_USAGE_SENSOR_DATA_ORIENTATION_MAGNETIC_NORTH,
          HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
          HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
          HID_REPORT_SIZE(16),
          HID_REPORT_COUNT(1),
          HID_USAGE_SENSOR_UNITS_DEGREES,
          HID_UNIT_EXPONENT(0x0F), // scale default unit to provide 1 digit past decimal point
          HID_INPUT(Const_Var_Abs),
          HID_USAGE_SENSOR_DATA_ORIENTATION_TRUE_NORTH,
          HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
          HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
          HID_REPORT_SIZE(16),
          HID_REPORT_COUNT(1),
          HID_USAGE_SENSOR_UNITS_DEGREES,
          HID_UNIT_EXPONENT(0x0F), // scale default unit to provide 1 digit past decimal point
          HID_INPUT(Const_Var_Abs),

          HID_END_COLLECTION
};
```

4.3.19 Orientation: Inclinometer

```
// For reference: Complete HID report descriptor

// 1D Inclinometer
const unsigned char inc1_report_descriptor[] = {
          HID_USAGE_PAGE_SENSOR,
          HID_USAGE_SENSOR_TYPE_ORIENTATION_INCLINOMETER_1D,
          HID_COLLECTION(Physical),

          //feature reports (xmit/receive)
          HID_USAGE_PAGE_SENSOR,
          HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE,
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_8(5),
          HID_REPORT_SIZE(8),
          HID_REPORT_COUNT(1),
          HID_COLLECTION(Logical),
```

```
          HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_NO_EVENTS,
          HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS,
          HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_THRESHOLD_EVENTS,
          HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_NO_EVENTS_WAKE,
          HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS_WAKE,
          HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_THRESHOLD_EVENTS_WAKE,
          HID_FEATURE(Data_Arr_Abs),
HID_END_COLLECTION,
HID_USAGE_SENSOR_PROPERTY_SENSOR_STATUS,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_32(0xFF,0xFF,0xFF,0xFF),
HID_REPORT_SIZE(32),
HID_REPORT_COUNT(1),
HID_FEATURE(Data_Var_Abs), // up to VT_UI4 worth of status info
HID_USAGE_SENSOR_PROPERTY_REPORT_INTERVAL,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_32(0xFF,0xFF,0xFF,0xFF),
HID_REPORT_SIZE(32),
HID_REPORT_COUNT(1),
// HID_USAGE_SENSOR_UNITS_MILLISECOND,
HID_UNIT_EXPONENT(0),
HID_FEATURE(Data_Var_Abs),
HID_USAGE_SENSOR_PROPERTY_SENSOR_CONNECTION_TYPE,  // NAry
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_8(2),
HID_REPORT_SIZE(8),
HID_REPORT_COUNT(1),
HID_COLLECTION(Logical),
          HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_INTEGRATED,
          HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_ATTACHED,
          HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_EXTERNAL,
          HID_FEATURE(Const_Arr_Abs),
HID_END_COLLECTION,
HID_USAGE_SENSOR_PROPERTY_CHANGE_SENSITIVITY_ABS,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_16(0xFF,0xFF),
HID_REPORT_SIZE(16),
HID_REPORT_COUNT(1),
HID_USAGE_SENSOR_UNITS_DEGREES,
HID_UNIT_EXPONENT(0x0E), // scale default unit to provide 2 digits past decimal point
HID_FEATURE(Data_Var_Abs),
HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_ORIENTATION_TILT,HID_USAGE_SENSOR_DATA_MOD_MAX),
HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
HID_REPORT_SIZE(16),
HID_REPORT_COUNT(1),
HID_USAGE_SENSOR_UNITS_DEGREES,
HID_UNIT_EXPONENT(0x0E), // scale default unit to provide 2 digits past decimal point
HID_FEATURE(Data_Var_Abs),
HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_ORIENTATION_TILT,HID_USAGE_SENSOR_DATA_MOD_MIN),
HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
HID_REPORT_SIZE(16),
HID_REPORT_COUNT(1),
HID_USAGE_SENSOR_UNITS_DEGREES,
HID_UNIT_EXPONENT(0x0E), // scale default unit to provide 2 digits past decimal point
HID_FEATURE(Data_Var_Abs),

//input reports (transmit)
HID_USAGE_PAGE_SENSOR,
HID_USAGE_SENSOR_STATE,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_8(6),
HID_REPORT_SIZE(8),
HID_REPORT_COUNT(1),
HID_COLLECTION(Logical),
          HID_USAGE_SENSOR_STATE_UNKNOWN,
          HID_USAGE_SENSOR_STATE_READY,
          HID_USAGE_SENSOR_STATE_NOT_AVAILABLE,
          HID_USAGE_SENSOR_STATE_NO_DATA,
          HID_USAGE_SENSOR_STATE_INITIALIZING,
          HID_USAGE_SENSOR_STATE_ACCESS_DENIED,
          HID_USAGE_SENSOR_STATE_ERROR,
          HID_INPUT(Const_Arr_Abs),
HID_END_COLLECTION,
HID_USAGE_SENSOR_EVENT,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_8(16),
HID_REPORT_SIZE(8),
HID_REPORT_COUNT(1),
HID_COLLECTION(Logical),
          HID_USAGE_SENSOR_EVENT_UNKNOWN,
          HID_USAGE_SENSOR_EVENT_STATE_CHANGED,
          HID_USAGE_SENSOR_EVENT_PROPERTY_CHANGED,
          HID_USAGE_SENSOR_EVENT_DATA_UPDATED,
          HID_USAGE_SENSOR_EVENT_POLL_RESPONSE,
          HID_USAGE_SENSOR_EVENT_CHANGE_SENSITIVITY,
          HID_USAGE_SENSOR_EVENT_MAX_REACHED,
          HID_USAGE_SENSOR_EVENT_MIN_REACHED,
          HID_USAGE_SENSOR_EVENT_HIGH_THRESHOLD_CROSS_UPWARD,
          HID_USAGE_SENSOR_EVENT_HIGH_THRESHOLD_CROSS_DOWNWARD,
          HID_USAGE_SENSOR_EVENT_LOW_THRESHOLD_CROSS_UPWARD,
          HID_USAGE_SENSOR_EVENT_LOW_THRESHOLD_CROSS_DOWNWARD,
          HID_USAGE_SENSOR_EVENT_ZERO_THRESHOLD_CROSS_UPWARD,
          HID_USAGE_SENSOR_EVENT_ZERO_THRESHOLD_CROSS_DOWNWARD,
```

```
                    HID_USAGE_SENSOR_EVENT_PERIOD_EXCEEDED,
                    HID_USAGE_SENSOR_EVENT_FREQUENCY_EXCEEDED,
                    HID_USAGE_SENSOR_EVENT_COMPLEX_TRIGGER,
                    HID_INPUT(Const_Arr_Abs),
          HID_END_COLLECTION,
          HID_USAGE_SENSOR_DATA_ORIENTATION_TILT_X,
          HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
          HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
          HID_REPORT_SIZE(16),
          HID_REPORT_COUNT(1),
          HID_USAGE_SENSOR_UNITS_DEGREES,
          HID_UNIT_EXPONENT(0x0E), // scale default unit to provide 2 digits past decimal point
          HID_INPUT(Const_Var_Abs),

          HID_END_COLLECTION
};

// 2D Inclinometer
const unsigned char inc2_report_descriptor[] = {
          HID_USAGE_PAGE_SENSOR,
          HID_USAGE_SENSOR_TYPE_ORIENTATION_INCLINOMETER_2D,
          HID_COLLECTION(Physical),

          //feature reports (xmit/receive)
          HID_USAGE_PAGE_SENSOR,
          HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE,
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_8(5),
          HID_REPORT_SIZE(8),
          HID_REPORT_COUNT(1),
          HID_COLLECTION(Logical),
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_NO_EVENTS,
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS,
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_THRESHOLD_EVENTS,
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_NO_EVENTS_WAKE,
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS_WAKE,
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_THRESHOLD_EVENTS_WAKE,
                    HID_FEATURE(Data_Arr_Abs),
          HID_END_COLLECTION,
          HID_USAGE_SENSOR_PROPERTY_SENSOR_STATUS,
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_32(0xFF,0xFF,0xFF,0xFF),
          HID_REPORT_SIZE(32),
          HID_REPORT_COUNT(1),
          HID_FEATURE(Data_Var_Abs), // up to VT_UI4 worth of status info
          HID_USAGE_SENSOR_PROPERTY_REPORT_INTERVAL,
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_32(0xFF,0xFF,0xFF,0xFF),
          HID_REPORT_SIZE(32),
          HID_REPORT_COUNT(1),
          // HID_USAGE_SENSOR_UNITS_MILLISECOND,
          HID_UNIT_EXPONENT(0),
          HID_FEATURE(Data_Var_Abs),
          HID_USAGE_SENSOR_PROPERTY_SENSOR_CONNECTION_TYPE,  // NAry
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_8(2),
          HID_REPORT_SIZE(8),
          HID_REPORT_COUNT(1),
          HID_COLLECTION(Logical),
                    HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_INTEGRATED,
                    HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_ATTACHED,
                    HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_EXTERNAL,
                    HID_FEATURE(Const_Arr_Abs),
          HID_END_COLLECTION,
          HID_USAGE_SENSOR_PROPERTY_CHANGE_SENSITIVITY_ABS,
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_16(0xFF,0xFF),
          HID_REPORT_SIZE(16),
          HID_REPORT_COUNT(1),
          HID_USAGE_SENSOR_UNITS_DEGREES,
          HID_UNIT_EXPONENT(0x0E), // scale default unit to provide 2 digits past decimal point
          HID_FEATURE(Data_Var_Abs),
          HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_ORIENTATION_TILT,HID_USAGE_SENSOR_DATA_MOD_MAX),
          HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
          HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
          HID_REPORT_SIZE(16),
          HID_REPORT_COUNT(1),
          HID_USAGE_SENSOR_UNITS_DEGREES,
          HID_UNIT_EXPONENT(0x0E), // scale default unit to provide 2 digits past decimal point
          HID_FEATURE(Data_Var_Abs),
          HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_ORIENTATION_TILT,HID_USAGE_SENSOR_DATA_MOD_MIN),
          HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
          HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
          HID_REPORT_SIZE(16),
          HID_REPORT_COUNT(1),
          HID_USAGE_SENSOR_UNITS_DEGREES,
          HID_UNIT_EXPONENT(0x0E), // scale default unit to provide 2 digits past decimal point
          HID_FEATURE(Data_Var_Abs),

          //input reports (transmit)
          HID_USAGE_PAGE_SENSOR,
          HID_USAGE_SENSOR_STATE,
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_8(6),
          HID_REPORT_SIZE(8),
```

```
        HID_REPORT_COUNT(1),
        HID_COLLECTION(Logical),
                HID_USAGE_SENSOR_STATE_UNKNOWN,
                HID_USAGE_SENSOR_STATE_READY,
                HID_USAGE_SENSOR_STATE_NOT_AVAILABLE,
                HID_USAGE_SENSOR_STATE_NO_DATA,
                HID_USAGE_SENSOR_STATE_INITIALIZING,
                HID_USAGE_SENSOR_STATE_ACCESS_DENIED,
                HID_USAGE_SENSOR_STATE_ERROR,
                HID_INPUT(Const_Arr_Abs),
        HID_END_COLLECTION,
        HID_USAGE_SENSOR_EVENT,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_8(16),
        HID_REPORT_SIZE(8),
        HID_REPORT_COUNT(1),
        HID_COLLECTION(Logical),
                HID_USAGE_SENSOR_EVENT_UNKNOWN,
                HID_USAGE_SENSOR_EVENT_STATE_CHANGED,
                HID_USAGE_SENSOR_EVENT_PROPERTY_CHANGED,
                HID_USAGE_SENSOR_EVENT_DATA_UPDATED,
                HID_USAGE_SENSOR_EVENT_POLL_RESPONSE,
                HID_USAGE_SENSOR_EVENT_CHANGE_SENSITIVITY,
                HID_USAGE_SENSOR_EVENT_MAX_REACHED,
                HID_USAGE_SENSOR_EVENT_MIN_REACHED,
                HID_USAGE_SENSOR_EVENT_HIGH_THRESHOLD_CROSS_UPWARD,
                HID_USAGE_SENSOR_EVENT_HIGH_THRESHOLD_CROSS_DOWNWARD,
                HID_USAGE_SENSOR_EVENT_LOW_THRESHOLD_CROSS_UPWARD,
                HID_USAGE_SENSOR_EVENT_LOW_THRESHOLD_CROSS_DOWNWARD,
                HID_USAGE_SENSOR_EVENT_ZERO_THRESHOLD_CROSS_UPWARD,
                HID_USAGE_SENSOR_EVENT_ZERO_THRESHOLD_CROSS_DOWNWARD,
                HID_USAGE_SENSOR_EVENT_PERIOD_EXCEEDED,
                HID_USAGE_SENSOR_EVENT_FREQUENCY_EXCEEDED,
                HID_USAGE_SENSOR_EVENT_COMPLEX_TRIGGER,
                HID_INPUT(Const_Arr_Abs),
        HID_END_COLLECTION,
        HID_USAGE_SENSOR_DATA_ORIENTATION_TILT_X,
        HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
        HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
        HID_REPORT_SIZE(16),
        HID_REPORT_COUNT(1),
        HID_USAGE_SENSOR_UNITS_DEGREES,
        HID_UNIT_EXPONENT(0x0E), // scale default unit to provide 2 digits past decimal point
        HID_INPUT(Const_Var_Abs),
        HID_USAGE_SENSOR_DATA_ORIENTATION_TILT_Y,
        HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
        HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
        HID_REPORT_SIZE(16),
        HID_REPORT_COUNT(1),
        HID_USAGE_SENSOR_UNITS_DEGREES,
        HID_UNIT_EXPONENT(0x0E), // scale default unit to provide 2 digits past decimal point
        HID_INPUT(Const_Var_Abs),

        HID_END_COLLECTION
};

// 3D Inclinometer
const unsigned char inc3_report_descriptor[] = {
        HID_USAGE_PAGE_SENSOR,
        HID_USAGE_SENSOR_TYPE_ORIENTATION_INCLINOMETER_3D,
        HID_COLLECTION(Physical),

        //feature reports (xmit/receive)
        HID_USAGE_PAGE_SENSOR,
        HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_8(5),
        HID_REPORT_SIZE(8),
        HID_REPORT_COUNT(1),
        HID_COLLECTION(Logical),
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_NO_EVENTS,
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS,
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_THRESHOLD_EVENTS,
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_NO_EVENTS_WAKE,
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS_WAKE,
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_THRESHOLD_EVENTS_WAKE,
                HID_FEATURE(Data_Arr_Abs),
        HID_END_COLLECTION,
        HID_USAGE_SENSOR_PROPERTY_SENSOR_STATUS,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_32(0xFF,0xFF,0xFF,0xFF),
        HID_REPORT_SIZE(32),
        HID_REPORT_COUNT(1),
        HID_FEATURE(Data_Var_Abs), // up to VT_UI4 worth of status info
        HID_USAGE_SENSOR_PROPERTY_REPORT_INTERVAL,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_32(0xFF,0xFF,0xFF,0xFF),
        HID_REPORT_SIZE(32),
        HID_REPORT_COUNT(1),
        // HID_USAGE_SENSOR_UNITS_MILLISECOND,
        HID_UNIT_EXPONENT(0),
        HID_FEATURE(Data_Var_Abs),
        HID_USAGE_SENSOR_PROPERTY_SENSOR_CONNECTION_TYPE,  // NAry
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_8(2),
```

```
HID_REPORT_SIZE(8),
HID_REPORT_COUNT(1),
HID_COLLECTION(Logical),
          HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_INTEGRATED,
          HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_ATTACHED,
          HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_EXTERNAL,
          HID_FEATURE(Const_Arr_Abs),
HID_END_COLLECTION,
HID_USAGE_SENSOR_PROPERTY_CHANGE_SENSITIVITY_ABS,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_16(0xFF,0xFF),
HID_REPORT_SIZE(16),
HID_REPORT_COUNT(1),
HID_USAGE_SENSOR_UNITS_DEGREES,
HID_UNIT_EXPONENT(0x0E), // scale default unit to provide 2 digits past decimal point
HID_FEATURE(Data_Var_Abs),
HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_ORIENTATION_TILT,HID_USAGE_SENSOR_DATA_MOD_MAX),
HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
HID_REPORT_SIZE(16),
HID_REPORT_COUNT(1),
HID_USAGE_SENSOR_UNITS_DEGREES,
HID_UNIT_EXPONENT(0x0E), // scale default unit to provide 2 digits past decimal point
HID_FEATURE(Data_Var_Abs),
HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_ORIENTATION_TILT,HID_USAGE_SENSOR_DATA_MOD_MIN),
HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
HID_REPORT_SIZE(16),
HID_REPORT_COUNT(1),
HID_USAGE_SENSOR_UNITS_DEGREES,
HID_UNIT_EXPONENT(0x0E), // scale default unit to provide 2 digits past decimal point
HID_FEATURE(Data_Var_Abs),

//input reports (transmit)
HID_USAGE_PAGE_SENSOR,
HID_USAGE_SENSOR_STATE,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_8(6),
HID_REPORT_SIZE(8),
HID_REPORT_COUNT(1),
HID_COLLECTION(Logical),
          HID_USAGE_SENSOR_STATE_UNKNOWN,
          HID_USAGE_SENSOR_STATE_READY,
          HID_USAGE_SENSOR_STATE_NOT_AVAILABLE,
          HID_USAGE_SENSOR_STATE_NO_DATA,
          HID_USAGE_SENSOR_STATE_INITIALIZING,
          HID_USAGE_SENSOR_STATE_ACCESS_DENIED,
          HID_USAGE_SENSOR_STATE_ERROR,
          HID_INPUT(Const_Arr_Abs),
HID_END_COLLECTION,
HID_USAGE_SENSOR_EVENT,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_8(16),
HID_REPORT_SIZE(8),
HID_REPORT_COUNT(1),
HID_COLLECTION(Logical),
          HID_USAGE_SENSOR_EVENT_UNKNOWN,
          HID_USAGE_SENSOR_EVENT_STATE_CHANGED,
          HID_USAGE_SENSOR_EVENT_PROPERTY_CHANGED,
          HID_USAGE_SENSOR_EVENT_DATA_UPDATED,
          HID_USAGE_SENSOR_EVENT_POLL_RESPONSE,
          HID_USAGE_SENSOR_EVENT_CHANGE_SENSITIVITY,
          HID_USAGE_SENSOR_EVENT_MAX_REACHED,
          HID_USAGE_SENSOR_EVENT_MIN_REACHED,
          HID_USAGE_SENSOR_EVENT_HIGH_THRESHOLD_CROSS_UPWARD,
          HID_USAGE_SENSOR_EVENT_HIGH_THRESHOLD_CROSS_DOWNWARD,
          HID_USAGE_SENSOR_EVENT_LOW_THRESHOLD_CROSS_UPWARD,
          HID_USAGE_SENSOR_EVENT_LOW_THRESHOLD_CROSS_DOWNWARD,
          HID_USAGE_SENSOR_EVENT_ZERO_THRESHOLD_CROSS_UPWARD,
          HID_USAGE_SENSOR_EVENT_ZERO_THRESHOLD_CROSS_DOWNWARD,
          HID_USAGE_SENSOR_EVENT_PERIOD_EXCEEDED,
          HID_USAGE_SENSOR_EVENT_FREQUENCY_EXCEEDED,
          HID_USAGE_SENSOR_EVENT_COMPLEX_TRIGGER,
          HID_INPUT(Const_Arr_Abs),
HID_END_COLLECTION,
HID_USAGE_SENSOR_DATA_ORIENTATION_TILT_X,
HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
HID_REPORT_SIZE(16),
HID_REPORT_COUNT(1),
HID_USAGE_SENSOR_UNITS_DEGREES,
HID_UNIT_EXPONENT(0x0E), // scale default unit to provide 2 digits past decimal point
HID_INPUT(Const_Var_Abs),
HID_USAGE_SENSOR_DATA_ORIENTATION_TILT_Y,
HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
HID_REPORT_SIZE(16),
HID_REPORT_COUNT(1),
HID_USAGE_SENSOR_UNITS_DEGREES,
HID_UNIT_EXPONENT(0x0E), // scale default unit to provide 2 digits past decimal point
HID_INPUT(Const_Var_Abs),
HID_USAGE_SENSOR_DATA_ORIENTATION_TILT_Z,
HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
HID_REPORT_SIZE(16),
```

```
        HID_REPORT_COUNT(1),
        HID_USAGE_SENSOR_UNITS_DEGREES,
        HID_UNIT_EXPONENT(0x0E), // scale default unit to provide 2 digits past decimal point
        HID_INPUT(Const_Var_Abs),

        HID_END_COLLECTION
};
```

4.3.20 Orientation: Distance

```
// For reference: Complete HID report descriptor

//Distance 1D
const unsigned char dis1_report_descriptor[] = {
        HID_USAGE_PAGE_SENSOR,
        HID_USAGE_SENSOR_TYPE_ORIENTATION_DISTANCE_1D,
        HID_COLLECTION(Physical),

        //feature reports (xmit/receive)
        HID_USAGE_PAGE_SENSOR,
        HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_8(5),
        HID_REPORT_SIZE(8),
        HID_REPORT_COUNT(1),
        HID_COLLECTION(Logical),
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_NO_EVENTS,
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS,
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_THRESHOLD_EVENTS,
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_NO_EVENTS_WAKE,
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS_WAKE,
                HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_THRESHOLD_EVENTS_WAKE,
                HID_FEATURE(Data_Arr_Abs),
        HID_END_COLLECTION,
        HID_USAGE_SENSOR_PROPERTY_SENSOR_STATUS,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_32(0xFF,0xFF,0xFF,0xFF),
        HID_REPORT_SIZE(32),
        HID_REPORT_COUNT(1),
        HID_FEATURE(Data_Var_Abs), // up to VT_UI4 worth of status info
        HID_USAGE_SENSOR_PROPERTY_REPORT_INTERVAL,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_32(0xFF,0xFF,0xFF,0xFF),
        HID_REPORT_SIZE(32),
        HID_REPORT_COUNT(1),
        // HID_USAGE_SENSOR_UNITS_MILLISECOND,
        HID_UNIT_EXPONENT(0),
        HID_FEATURE(Data_Var_Abs),
        HID_USAGE_SENSOR_PROPERTY_SENSOR_CONNECTION_TYPE,  // NAry
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_8(2),
        HID_REPORT_SIZE(8),
        HID_REPORT_COUNT(1),
        HID_COLLECTION(Logical),
                HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_INTEGRATED,
                HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_ATTACHED,
                HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_EXTERNAL,
                HID_FEATURE(Const_Arr_Abs),
        HID_END_COLLECTION,
        HID_USAGE_SENSOR_PROPERTY_CHANGE_SENSITIVITY_ABS,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_16(0xFF,0xFF),
        HID_REPORT_SIZE(16),
        HID_REPORT_COUNT(1),
        // HID_USAGE_SENSOR_UNITS_METER,
        HID_UNIT_EXPONENT(0x0E), // scale default unit “meter” to provide 2 digits past the decimal point
        HID_FEATURE(Data_Var_Abs),
        HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_ORIENTATION_DISTANCE,HID_USAGE_SENSOR_DATA_MOD_MAX),
        HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
        HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
        HID_REPORT_SIZE(16),
        HID_REPORT_COUNT(1),
        // HID_USAGE_SENSOR_UNITS_METER,
        HID_UNIT_EXPONENT(0x0E), // scale default unit “meter” to provide 2 digits past the decimal point
        HID_FEATURE(Data_Var_Abs),
        HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_ORIENTATION_DISTANCE,HID_USAGE_SENSOR_DATA_MOD_MIN),
        HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
        HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
        HID_REPORT_SIZE(16),
        HID_REPORT_COUNT(1),
        // HID_USAGE_SENSOR_UNITS_METER,
        HID_UNIT_EXPONENT(0x0E), // scale default unit “meter” to provide 2 digits past the decimal point
        HID_FEATURE(Data_Var_Abs),

        //input reports (transmit)
        HID_USAGE_PAGE_SENSOR,
        HID_USAGE_SENSOR_STATE,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_8(6),
        HID_REPORT_SIZE(8),
        HID_REPORT_COUNT(1),
        HID_COLLECTION(Logical),
                HID_USAGE_SENSOR_STATE_UNKNOWN,
                HID_USAGE_SENSOR_STATE_READY,
```

```
                    HID_USAGE_SENSOR_STATE_NOT_AVAILABLE,
                    HID_USAGE_SENSOR_STATE_NO_DATA,
                    HID_USAGE_SENSOR_STATE_INITIALIZING,
                    HID_USAGE_SENSOR_STATE_ACCESS_DENIED,
                    HID_USAGE_SENSOR_STATE_ERROR,
                    HID_INPUT(Const_Arr_Abs),
          HID_END_COLLECTION,
          HID_USAGE_SENSOR_EVENT,
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_8(16),
          HID_REPORT_SIZE(8),
          HID_REPORT_COUNT(1),
          HID_COLLECTION(Logical),
                    HID_USAGE_SENSOR_EVENT_UNKNOWN,
                    HID_USAGE_SENSOR_EVENT_STATE_CHANGED,
                    HID_USAGE_SENSOR_EVENT_PROPERTY_CHANGED,
                    HID_USAGE_SENSOR_EVENT_DATA_UPDATED,
                    HID_USAGE_SENSOR_EVENT_POLL_RESPONSE,
                    HID_USAGE_SENSOR_EVENT_CHANGE_SENSITIVITY,
                    HID_USAGE_SENSOR_EVENT_MAX_REACHED,
                    HID_USAGE_SENSOR_EVENT_MIN_REACHED,
                    HID_USAGE_SENSOR_EVENT_HIGH_THRESHOLD_CROSS_UPWARD,
                    HID_USAGE_SENSOR_EVENT_HIGH_THRESHOLD_CROSS_DOWNWARD,
                    HID_USAGE_SENSOR_EVENT_LOW_THRESHOLD_CROSS_UPWARD,
                    HID_USAGE_SENSOR_EVENT_LOW_THRESHOLD_CROSS_DOWNWARD,
                    HID_USAGE_SENSOR_EVENT_ZERO_THRESHOLD_CROSS_UPWARD,
                    HID_USAGE_SENSOR_EVENT_ZERO_THRESHOLD_CROSS_DOWNWARD,
                    HID_USAGE_SENSOR_EVENT_PERIOD_EXCEEDED,
                    HID_USAGE_SENSOR_EVENT_FREQUENCY_EXCEEDED,
                    HID_USAGE_SENSOR_EVENT_COMPLEX_TRIGGER,
                    HID_INPUT(Const_Arr_Abs),
          HID_END_COLLECTION,
          HID_USAGE_SENSOR_DATA_ORIENTATION_DISTANCE_X,
          HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
          HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
          HID_REPORT_SIZE(16),
          HID_REPORT_COUNT(1),
          // HID_USAGE_SENSOR_UNITS_METER,
          HID_UNIT_EXPONENT(0x0E), // scale default unit “meter” to provide 2 digits past the decimal point
          HID_INPUT(Const_Var_Abs),

          HID_END_COLLECTION
};

//Distance 2D
const unsigned char dis2_report_descriptor[] = {
          HID_USAGE_PAGE_SENSOR,
          HID_USAGE_SENSOR_TYPE_ORIENTATION_DISTANCE_2D,
          HID_COLLECTION(Physical),

          //feature reports (xmit/receive)
          HID_USAGE_PAGE_SENSOR,
          HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE,
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_8(5),
          HID_REPORT_SIZE(8),
          HID_REPORT_COUNT(1),
          HID_COLLECTION(Logical),
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_NO_EVENTS,
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS,
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_THRESHOLD_EVENTS,
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_NO_EVENTS_WAKE,
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS_WAKE,
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_THRESHOLD_EVENTS_WAKE,
                    HID_FEATURE(Data_Arr_Abs),
          HID_END_COLLECTION,
          HID_USAGE_SENSOR_PROPERTY_SENSOR_STATUS,
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_32(0xFF,0xFF,0xFF,0xFF),
          HID_REPORT_SIZE(32),
          HID_REPORT_COUNT(1),
          HID_FEATURE(Data_Var_Abs), // up to VT_UI4 worth of status info
          HID_USAGE_SENSOR_PROPERTY_REPORT_INTERVAL,
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_32(0xFF,0xFF,0xFF,0xFF),
          HID_REPORT_SIZE(32),
          HID_REPORT_COUNT(1),
          // HID_USAGE_SENSOR_UNITS_MILLISECOND,
          HID_UNIT_EXPONENT(0),
          HID_FEATURE(Data_Var_Abs),
          HID_USAGE_SENSOR_PROPERTY_SENSOR_CONNECTION_TYPE,  // NAry
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_8(2),
          HID_REPORT_SIZE(8),
          HID_REPORT_COUNT(1),
          HID_COLLECTION(Logical),
                    HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_INTEGRATED,
                    HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_ATTACHED,
                    HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_EXTERNAL,
                    HID_FEATURE(Const_Arr_Abs),
          HID_END_COLLECTION,
          HID_USAGE_SENSOR_PROPERTY_CHANGE_SENSITIVITY_ABS,
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_16(0xFF,0xFF),
          HID_REPORT_SIZE(16),
```

```
        HID_REPORT_COUNT(1),
        // HID_USAGE_SENSOR_UNITS_METER,
        HID_UNIT_EXPONENT(0x0E), // scale default unit “meter” to provide 2 digits past the decimal point
        HID_FEATURE(Data_Var_Abs),
        HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_ORIENTATION_DISTANCE,HID_USAGE_SENSOR_DATA_MOD_MAX),
        HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
        HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
        HID_REPORT_SIZE(16),
        HID_REPORT_COUNT(1),
        // HID_USAGE_SENSOR_UNITS_METER,
        HID_UNIT_EXPONENT(0x0E), // scale default unit “meter” to provide 2 digits past the decimal point
        HID_FEATURE(Data_Var_Abs),
        HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_ORIENTATION_DISTANCE,HID_USAGE_SENSOR_DATA_MOD_MIN),
        HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
        HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
        HID_REPORT_SIZE(16),
        HID_REPORT_COUNT(1),
        // HID_USAGE_SENSOR_UNITS_METER,
        HID_UNIT_EXPONENT(0x0E), // scale default unit “meter” to provide 2 digits past the decimal point
        HID_FEATURE(Data_Var_Abs),

        //input reports (transmit)
        HID_USAGE_PAGE_SENSOR,
        HID_USAGE_SENSOR_STATE,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_8(6),
        HID_REPORT_SIZE(8),
        HID_REPORT_COUNT(1),
        HID_COLLECTION(Logical),
                HID_USAGE_SENSOR_STATE_UNKNOWN,
                HID_USAGE_SENSOR_STATE_READY,
                HID_USAGE_SENSOR_STATE_NOT_AVAILABLE,
                HID_USAGE_SENSOR_STATE_NO_DATA,
                HID_USAGE_SENSOR_STATE_INITIALIZING,
                HID_USAGE_SENSOR_STATE_ACCESS_DENIED,
                HID_USAGE_SENSOR_STATE_ERROR,
                HID_INPUT(Const_Arr_Abs),
        HID_END_COLLECTION,
        HID_USAGE_SENSOR_EVENT,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_8(16),
        HID_REPORT_SIZE(8),
        HID_REPORT_COUNT(1),
        HID_COLLECTION(Logical),
                HID_USAGE_SENSOR_EVENT_UNKNOWN,
                HID_USAGE_SENSOR_EVENT_STATE_CHANGED,
                HID_USAGE_SENSOR_EVENT_PROPERTY_CHANGED,
                HID_USAGE_SENSOR_EVENT_DATA_UPDATED,
                HID_USAGE_SENSOR_EVENT_POLL_RESPONSE,
                HID_USAGE_SENSOR_EVENT_CHANGE_SENSITIVITY,
                HID_USAGE_SENSOR_EVENT_MAX_REACHED,
                HID_USAGE_SENSOR_EVENT_MIN_REACHED,
                HID_USAGE_SENSOR_EVENT_HIGH_THRESHOLD_CROSS_UPWARD,
                HID_USAGE_SENSOR_EVENT_HIGH_THRESHOLD_CROSS_DOWNWARD,
                HID_USAGE_SENSOR_EVENT_LOW_THRESHOLD_CROSS_UPWARD,
                HID_USAGE_SENSOR_EVENT_LOW_THRESHOLD_CROSS_DOWNWARD,
                HID_USAGE_SENSOR_EVENT_ZERO_THRESHOLD_CROSS_UPWARD,
                HID_USAGE_SENSOR_EVENT_ZERO_THRESHOLD_CROSS_DOWNWARD,
                HID_USAGE_SENSOR_EVENT_PERIOD_EXCEEDED,
                HID_USAGE_SENSOR_EVENT_FREQUENCY_EXCEEDED,
                HID_USAGE_SENSOR_EVENT_COMPLEX_TRIGGER,
                HID_INPUT(Const_Arr_Abs),
        HID_END_COLLECTION,
        HID_USAGE_SENSOR_DATA_ORIENTATION_DISTANCE_X,
        HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
        HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
        HID_REPORT_SIZE(16),
        HID_REPORT_COUNT(1),
        // HID_USAGE_SENSOR_UNITS_METER,
        HID_UNIT_EXPONENT(0x0E), // scale default unit “meter” to provide 2 digits past the decimal point
        HID_INPUT(Const_Var_Abs),
        HID_USAGE_SENSOR_DATA_ORIENTATION_DISTANCE_Y,
        HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
        HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
        HID_REPORT_SIZE(16),
        HID_REPORT_COUNT(1),
        // HID_USAGE_SENSOR_UNITS_METER,
        HID_UNIT_EXPONENT(0x0E), // scale default unit “meter” to provide 2 digits past the decimal point
        HID_INPUT(Const_Var_Abs),

        HID_END_COLLECTION
};

//Distance 3D
const unsigned char dis3_report_descriptor[] = {
        HID_USAGE_PAGE_SENSOR,
        HID_USAGE_SENSOR_TYPE_ORIENTATION_DISTANCE_3D,
        HID_COLLECTION(Physical),

        //feature reports (xmit/receive)
        HID_USAGE_PAGE_SENSOR,
        HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE,
        HID_LOGICAL_MIN_8(0),
        HID_LOGICAL_MAX_8(5),
        HID_REPORT_SIZE(8),
```

```
HID_REPORT_COUNT(1),
HID_COLLECTION(Logical),
          HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_NO_EVENTS,
          HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS,
          HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_THRESHOLD_EVENTS,
          HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_NO_EVENTS_WAKE,
          HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS_WAKE,
          HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_THRESHOLD_EVENTS_WAKE,
          HID_FEATURE(Data_Arr_Abs),
HID_END_COLLECTION,
HID_USAGE_SENSOR_PROPERTY_SENSOR_STATUS,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_32(0xFF,0xFF,0xFF,0xFF),
HID_REPORT_SIZE(32),
HID_REPORT_COUNT(1),
HID_FEATURE(Data_Var_Abs), // up to VT_UI4 worth of status info
HID_USAGE_SENSOR_PROPERTY_REPORT_INTERVAL,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_32(0xFF,0xFF,0xFF,0xFF),
HID_REPORT_SIZE(32),
HID_REPORT_COUNT(1),
// HID_USAGE_SENSOR_UNITS_MILLISECOND,
HID_UNIT_EXPONENT(0),
HID_FEATURE(Data_Var_Abs),
HID_USAGE_SENSOR_PROPERTY_SENSOR_CONNECTION_TYPE,  // NAry
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_8(2),
HID_REPORT_SIZE(8),
HID_REPORT_COUNT(1),
HID_COLLECTION(Logical),
          HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_INTEGRATED,
          HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_ATTACHED,
          HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_EXTERNAL,
          HID_FEATURE(Const_Arr_Abs),
HID_END_COLLECTION,
HID_USAGE_SENSOR_PROPERTY_CHANGE_SENSITIVITY_ABS,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_16(0xFF,0xFF),
HID_REPORT_SIZE(16),
HID_REPORT_COUNT(1),
// HID_USAGE_SENSOR_UNITS_METER,
HID_UNIT_EXPONENT(0x0E), // scale default unit “meter” to provide 2 digits past the decimal point
HID_FEATURE(Data_Var_Abs),
HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_ORIENTATION_DISTANCE,HID_USAGE_SENSOR_DATA_MOD_MAX),
HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
HID_REPORT_SIZE(16),
HID_REPORT_COUNT(1),
// HID_USAGE_SENSOR_UNITS_METER,
HID_UNIT_EXPONENT(0x0E), // scale default unit “meter” to provide 2 digits past the decimal point
HID_FEATURE(Data_Var_Abs),
HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_ORIENTATION_DISTANCE,HID_USAGE_SENSOR_DATA_MOD_MIN),
HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
HID_REPORT_SIZE(16),
HID_REPORT_COUNT(1),
// HID_USAGE_SENSOR_UNITS_METER,
HID_UNIT_EXPONENT(0x0E), // scale default unit “meter” to provide 2 digits past the decimal point
HID_FEATURE(Data_Var_Abs),

//input reports (transmit)
HID_USAGE_PAGE_SENSOR,
HID_USAGE_SENSOR_STATE,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_8(6),
HID_REPORT_SIZE(8),
HID_REPORT_COUNT(1),
HID_COLLECTION(Logical),
          HID_USAGE_SENSOR_STATE_UNKNOWN,
          HID_USAGE_SENSOR_STATE_READY,
          HID_USAGE_SENSOR_STATE_NOT_AVAILABLE,
          HID_USAGE_SENSOR_STATE_NO_DATA,
          HID_USAGE_SENSOR_STATE_INITIALIZING,
          HID_USAGE_SENSOR_STATE_ACCESS_DENIED,
          HID_USAGE_SENSOR_STATE_ERROR,
          HID_INPUT(Const_Arr_Abs),
HID_END_COLLECTION,
HID_USAGE_SENSOR_EVENT,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_8(16),
HID_REPORT_SIZE(8),
HID_REPORT_COUNT(1),
HID_COLLECTION(Logical),
          HID_USAGE_SENSOR_EVENT_UNKNOWN,
          HID_USAGE_SENSOR_EVENT_STATE_CHANGED,
          HID_USAGE_SENSOR_EVENT_PROPERTY_CHANGED,
          HID_USAGE_SENSOR_EVENT_DATA_UPDATED,
          HID_USAGE_SENSOR_EVENT_POLL_RESPONSE,
          HID_USAGE_SENSOR_EVENT_CHANGE_SENSITIVITY,
          HID_USAGE_SENSOR_EVENT_MAX_REACHED,
          HID_USAGE_SENSOR_EVENT_MIN_REACHED,
          HID_USAGE_SENSOR_EVENT_HIGH_THRESHOLD_CROSS_UPWARD,
          HID_USAGE_SENSOR_EVENT_HIGH_THRESHOLD_CROSS_DOWNWARD,
          HID_USAGE_SENSOR_EVENT_LOW_THRESHOLD_CROSS_UPWARD,
          HID_USAGE_SENSOR_EVENT_LOW_THRESHOLD_CROSS_DOWNWARD,
```

```
                    HID_USAGE_SENSOR_EVENT_ZERO_THRESHOLD_CROSS_UPWARD,
                    HID_USAGE_SENSOR_EVENT_ZERO_THRESHOLD_CROSS_DOWNWARD,
                    HID_USAGE_SENSOR_EVENT_PERIOD_EXCEEDED,
                    HID_USAGE_SENSOR_EVENT_FREQUENCY_EXCEEDED,
                    HID_USAGE_SENSOR_EVENT_COMPLEX_TRIGGER,
                    HID_INPUT(Const_Arr_Abs),
          HID_END_COLLECTION,
          HID_USAGE_SENSOR_DATA_ORIENTATION_DISTANCE_X,
          HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
          HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
          HID_REPORT_SIZE(16),
          HID_REPORT_COUNT(1),
          // HID_USAGE_SENSOR_UNITS_METER,
          HID_UNIT_EXPONENT(0x0E), // scale default unit “meter” to provide 2 digits past the decimal point
          HID_INPUT(Const_Var_Abs),
          HID_USAGE_SENSOR_DATA_ORIENTATION_DISTANCE_Y,
          HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
          HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
          HID_REPORT_SIZE(16),
          HID_REPORT_COUNT(1),
          // HID_USAGE_SENSOR_UNITS_METER,
          HID_UNIT_EXPONENT(0x0E), // scale default unit “meter” to provide 2 digits past the decimal point
          HID_INPUT(Const_Var_Abs),
          HID_USAGE_SENSOR_DATA_ORIENTATION_DISTANCE_Z,
          HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
          HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
          HID_REPORT_SIZE(16),
          HID_REPORT_COUNT(1),
          // HID_USAGE_SENSOR_UNITS_METER,
          HID_UNIT_EXPONENT(0x0E), // scale default unit “meter” to provide 2 digits past the decimal point
          HID_INPUT(Const_Var_Abs),

          HID_END_COLLECTION
};
```

4.3.21 Orientation: Device Orientation

```
// For reference: Complete HID report descriptor

// Device Orientation sensor
const unsigned char devor_report_descriptor[] = {
          HID_USAGE_PAGE_SENSOR,
          HID_USAGE_SENSOR_TYPE_ORIENTATION_DEVICE_ORIENTATION,
          HID_COLLECTION(Physical),

          //feature reports (xmit/receive)
          HID_USAGE_PAGE_SENSOR,
          HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE,
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_8(5),
          HID_REPORT_SIZE(8),
          HID_REPORT_COUNT(1),
          HID_COLLECTION(Logical),
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_NO_EVENTS,
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS,
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_THRESHOLD_EVENTS,
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_NO_EVENTS_WAKE,
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS_WAKE,
                    HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_THRESHOLD_EVENTS_WAKE,
                    HID_FEATURE(Data_Arr_Abs),
          HID_END_COLLECTION,
          HID_USAGE_SENSOR_PROPERTY_REPORT_INTERVAL,
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_32(0xFF,0xFF,0xFF,0xFF),
          HID_REPORT_SIZE(32),
          HID_REPORT_COUNT(1),
          // HID_USAGE_SENSOR_UNITS_MILLISECOND,
          HID_UNIT_EXPONENT(0),
          HID_FEATURE(Data_Var_Abs),
          HID_USAGE_SENSOR_PROPERTY_SENSOR_CONNECTION_TYPE,  // NAry
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_8(2),
          HID_REPORT_SIZE(8),
          HID_REPORT_COUNT(1),
          HID_COLLECTION(Logical),
                    HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_INTEGRATED,
                    HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_ATTACHED,
                    HID_USAGE_SENSOR_PROPERTY_CONNECTION_TYPE_PC_EXTERNAL,
                    HID_FEATURE(Const_Arr_Abs),
          HID_END_COLLECTION,
          HID_USAGE_SENSOR_PROPERTY_CHANGE_SENSITIVITY_ABS,
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_16(0xFF,0xFF),
          HID_REPORT_SIZE(16),
          HID_REPORT_COUNT(1),
          // HID_USAGE_SENSOR_UNITS_METER,
          HID_UNIT_EXPONENT(0x0E), // scale default unit “meter” to provide 2 digits past the decimal point
          HID_FEATURE(Data_Var_Abs),
          HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_ORIENTATION_QUATERNION,
                    HID_USAGE_SENSOR_DATA_MOD_CHANGE_SENSITIVITY_ABS),
          HID_LOGICAL_MIN_8(0),
          HID_LOGICAL_MAX_16(0xFF,0xFF),
          HID_REPORT_SIZE(16),
          HID_REPORT_COUNT(1),
```

```
HID_USAGE_SENSOR_UNITS_NOT_SPECIFIED,
HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_ORIENTATION_QUATERNION,HID_USAGE_SENSOR_DATA_MOD_MAX),
HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
HID_REPORT_SIZE(16),
HID_REPORT_COUNT(1),
HID_USAGE_SENSOR_UNITS_NOT_SPECIFIED,
HID_UNIT_EXPONENT(0x01),
HID_FEATURE(Data_Var_Abs),
HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_ORIENTATION_QUATERNION,HID_USAGE_SENSOR_DATA_MOD_MIN),
HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
HID_REPORT_SIZE(16),
HID_REPORT_COUNT(1),
HID_USAGE_SENSOR_UNITS_NOT_SPECIFIED,
HID_UNIT_EXPONENT(0x01),
HID_FEATURE(Data_Var_Abs),
HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_ORIENTATION_ROTATION_MATRIX,
          HID_USAGE_SENSOR_DATA_MOD_CHANGE_SENSITIVITY_ABS),
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_16(0xFF,0xFF),
HID_REPORT_SIZE(16),
HID_REPORT_COUNT(1),
HID_USAGE_SENSOR_UNITS_NOT_SPECIFIED,
HID_UNIT_EXPONENT(0x0E),
HID_FEATURE(Data_Var_Abs),
HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_ORIENTATION_ROTATION_MATRIX,HID_USAGE_SENSOR_DATA_MOD_MAX),
HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
HID_REPORT_SIZE(16),
HID_REPORT_COUNT(1),
HID_USAGE_SENSOR_UNITS_NOT_SPECIFIED,
HID_UNIT_EXPONENT(0x0E),
HID_FEATURE(Data_Var_Abs),
HID_USAGE_SENSOR_DATA(HID_USAGE_SENSOR_DATA_ORIENTATION_ROTATION_MATRIX,HID_USAGE_SENSOR_DATA_MOD_MIN),
HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
HID_REPORT_SIZE(16),
HID_REPORT_COUNT(1),
HID_USAGE_SENSOR_UNITS_NOT_SPECIFIED,
HID_UNIT_EXPONENT(0x0E),
HID_FEATURE(Data_Var_Abs),

//input reports (transmit)
HID_USAGE_PAGE_SENSOR,
HID_USAGE_SENSOR_STATE,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_8(6),
HID_REPORT_SIZE(8),
HID_REPORT_COUNT(1),
HID_COLLECTION(Logical),
          HID_USAGE_SENSOR_STATE_UNKNOWN,
          HID_USAGE_SENSOR_STATE_READY,
          HID_USAGE_SENSOR_STATE_NOT_AVAILABLE,
          HID_USAGE_SENSOR_STATE_NO_DATA,
          HID_USAGE_SENSOR_STATE_INITIALIZING,
          HID_USAGE_SENSOR_STATE_ACCESS_DENIED,
          HID_USAGE_SENSOR_STATE_ERROR,
          HID_INPUT(Const_Arr_Abs),
HID_END_COLLECTION,
HID_USAGE_SENSOR_EVENT,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_8(16),
HID_REPORT_SIZE(8),
HID_REPORT_COUNT(1),
HID_COLLECTION(Logical),
          HID_USAGE_SENSOR_EVENT_UNKNOWN,
          HID_USAGE_SENSOR_EVENT_STATE_CHANGED,
          HID_USAGE_SENSOR_EVENT_PROPERTY_CHANGED,
          HID_USAGE_SENSOR_EVENT_DATA_UPDATED,
          HID_USAGE_SENSOR_EVENT_POLL_RESPONSE,
          HID_USAGE_SENSOR_EVENT_CHANGE_SENSITIVITY,
          HID_USAGE_SENSOR_EVENT_MAX_REACHED,
          HID_USAGE_SENSOR_EVENT_MIN_REACHED,
          HID_USAGE_SENSOR_EVENT_HIGH_THRESHOLD_CROSS_UPWARD,
          HID_USAGE_SENSOR_EVENT_HIGH_THRESHOLD_CROSS_DOWNWARD,
          HID_USAGE_SENSOR_EVENT_LOW_THRESHOLD_CROSS_UPWARD,
          HID_USAGE_SENSOR_EVENT_LOW_THRESHOLD_CROSS_DOWNWARD,
          HID_USAGE_SENSOR_EVENT_ZERO_THRESHOLD_CROSS_UPWARD,
          HID_USAGE_SENSOR_EVENT_ZERO_THRESHOLD_CROSS_DOWNWARD,
          HID_USAGE_SENSOR_EVENT_PERIOD_EXCEEDED,
          HID_USAGE_SENSOR_EVENT_FREQUENCY_EXCEEDED,
          HID_USAGE_SENSOR_EVENT_COMPLEX_TRIGGER,
          HID_INPUT(Const_Arr_Abs),
HID_END_COLLECTION,
HID_USAGE_SENSOR_DATA_ORIENTATION_QUATERNION,
HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
HID_REPORT_SIZE(16),
HID_REPORT_COUNT(4),
HID_USAGE_SENSOR_UNITS_NOT_SPECIFIED,
HID_UNIT_EXPONENT(0x0E),
HID_INPUT(Const_Arr_Abs),
HID_USAGE_SENSOR_DATA_ORIENTATION_ROTATION_MATRIX,
HID_LOGICAL_MIN_16(0x01,0x80), //    LOGICAL_MINIMUM (-32767)
```

```
        HID_LOGICAL_MAX_16(0xFF,0x7F), //    LOGICAL_MAXIMUM (32767)
        HID_REPORT_SIZE(16),
        HID_REPORT_COUNT(9),
        HID_USAGE_SENSOR_UNITS_NOT_SPECIFIED,
        HID_UNIT_EXPONENT(0x0F),
        HID_INPUT(Const_Arr_Abs),

        HID_END_COLLECTION
};
```

[End of document]