

Request #: HUTRR44
Title: HID Accuracy Proposal
Spec Release: HID 1.12
Received: 18 April 2013
Requester: Andrew Dye (must be authorized by MS)
Company: Microsoft Corporation
Phone: +1 (425) 4212463 X12463
Fax:
Email: andrewdy@microsoft.com

Current Status: Approved
Priority: Normal
Submitted: 19 April 2013
Voting Starts: 4 May 2013
Voting Ends: 12 May 2013
Approved: 15 May 2013
Required Voter: HID Chair
Nathan Sherman
nathans@microsoft.com
Required Voter: ST Microelectronics
Slim Jallouli
slim.jallouli@st.com
Required Voter: Intel Corporation
Steve McGowan
Steve.mcgowan@intel.com

Summary

This change defines new usages for magnetometer accuracy and associated low, medium, and high values. These allow a HID-connected magnetometer-derived sensor to specify the runtime accuracy based on current calibration data.

Background

Magnetometer sensors are susceptible to electromagnetic field interference from the surrounding environment or the computing device itself. Because orientation sensors such as compass, inclinometer, and device orientation rely on magnetometer data, as interference increases, sensor accuracy degrades. In many cases dynamic calibration is needed to account for the changing electromagnetic environment, which requires the device to be moved around all three device axis.

This change proposes a new magnetometer accuracy data field usage to be included in each orientation input report. Magnetometer accuracy will be one of low, medium, or high, indicating how closely the data represents the heading of a magnetically-calibrated device with respect to a horizontal plane. Here magnetometer accuracy relates to the component of the fused sensor data impacted by the magnetic field and not just the raw magnetic field vector. For example magnetometer accuracy for an inclinometer would describe the accuracy of the Z (yaw) component of the data, but not the X and Y components. Often computing an absolute accuracy in degrees is computationally expensive or impossible as data values vary within some confidence interval. This solution allows magnetometer

accuracy to be specified at a reasonable granularity: low, medium, and high. This proposal will only define the relative meaning of each of these values; an explicit definition is left to the implementation.

With this new usage, the consumer of the data is responsible for determining minimum acceptable accuracy and taking the appropriate action. There is no event defined indicating calibration is needed or complete, and there is no means to instruct the sensor to enter a special calibration mode. It is expected that the magnetometer accuracy value will change in successive data reports as calibration changes.

Proposal

Definitions

Add the following to section 1.0, table 1, page 15:

Usage ID	Usage Name	Usage Type
0488	Data Field: Magnetometer Accuracy	NArY
08E0	Magnetometer Accuracy: Low	Sel
08E1	Magnetometer Accuracy: Medium	Sel
08E2	Magnetometer Accuracy: High	Sel

Remove the following from section 1.0, table 1, page 15:

- Omit 0488 from the reserved range Data Field: Orientation Reserved

Add the following to section 1.13, page 38:

Magnetometer Accuracy	<p>NArY – indicates accuracy of the sensor data component impacted by the magnetic field</p> <ul style="list-style-type: none"> • Sel – Low. Sensor is providing a low level of magnetometer accuracy. It should be calibrated for reliable data. • Sel – Medium. Sensor is providing a medium level of magnetometer accuracy and may benefit from additional calibration data. • Sel – High. Sensor is fully calibrated and providing a high level of magnetometer accuracy.
-----------------------	--

Add the following to section 2.1, page 47

Define Magnetometer Accuracy in the Glossary:

Magnetometer Accuracy	<p>How closely an orientation sensor’s data fields represent the heading of a magnetically-calibrated device with respect to a horizontal plane. Magnetometer accuracy relates to the component of the fused sensor data impacted by the magnetic field and not just the raw magnetic field vector.</p>
-----------------------	---

Add the following to section 4.1, page 75

```
#define HID_USAGE_SENSOR_DATA_ORIENTATION_MAGNETOMETER_ACCURACY          0x0A,0x88,0x04 // NArY
// begin orientation magnetometer accuracy selectors
#define HID_USAGE_SENSOR_DATA_ORIENTATION_MAGNETOMETER_ACCURACY_LOW      0x0A,0xE0,0x08 // Sel
#define HID_USAGE_SENSOR_DATA_ORIENTATION_MAGNETOMETER_ACCURACY_MEDIUM  0x0A,0xE1,0x08 // Sel
```

```
#define HID_USAGE_SENSOR_DATA_ORIENTATION_MAGNETOMETER_ACCURACY_HIGH 0x0A,0xE2,0x08 // Sel
```

Report Descriptor Reference

Add the following to section 4.3.18, page 137-140. Changes in red.

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

    //feature reports (xmit/receive)
    ...

    //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_USAGE_SENSOR_DATA_ORIENTATION_MAGNETOMETER_ACCURACY,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_8(2),
HID_REPORT_SIZE(8),
HID_REPORT_COUNT(1),
HID_COLLECTION(Logical),
    HID_USAGE_SENSOR_DATA_ORIENTATION_MAGNETOMETER_ACCURACY_LOW,
    HID_USAGE_SENSOR_DATA_ORIENTATION_MAGNETOMETER_ACCURACY_MEDIUM,
    HID_USAGE_SENSOR_DATA_ORIENTATION_MAGNETOMETER_ACCURACY_HIGH,
    HID_INPUT(Data_Arr_Abs),
HID_END_COLLECTION,

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)
    ...

    //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_USAGE_SENSOR_DATA_ORIENTATION_MAGNETOMETER_ACCURACY,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_8(2),
HID_REPORT_SIZE(8),
HID_REPORT_COUNT(1),
HID_COLLECTION(Logical),
    HID_USAGE_SENSOR_DATA_ORIENTATION_MAGNETOMETER_ACCURACY_LOW,
    HID_USAGE_SENSOR_DATA_ORIENTATION_MAGNETOMETER_ACCURACY_MEDIUM,
    HID_USAGE_SENSOR_DATA_ORIENTATION_MAGNETOMETER_ACCURACY_HIGH,
    HID_INPUT(Data_Arr_Abs),
HID_END_COLLECTION,

HID_END_COLLECTION
};

```

Add the following to section 4.3.19, page 140-145. Changes in red.

```

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

    //feature reports (xmit/receive)
    ...

    //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_MAGNETOMETER_ACCURACY,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_8(2),
HID_REPORT_SIZE(8),
HID_REPORT_COUNT(1),
HID_COLLECTION(Logical),
    HID_USAGE_SENSOR_DATA_ORIENTATION_MAGNETOMETER_ACCURACY_LOW,
    HID_USAGE_SENSOR_DATA_ORIENTATION_MAGNETOMETER_ACCURACY_MEDIUM,
    HID_USAGE_SENSOR_DATA_ORIENTATION_MAGNETOMETER_ACCURACY_HIGH,
    HID_INPUT(Data_Arr_Abs),
HID_END_COLLECTION,

```

```

    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)
    ...

    //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_MAGNETOMETER_ACCURACY,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_8(2),
HID_REPORT_SIZE(8),
HID_REPORT_COUNT(1),
HID_COLLECTION(Logical),
    HID_USAGE_SENSOR_DATA_ORIENTATION_MAGNETOMETER_ACCURACY_LOW,
    HID_USAGE_SENSOR_DATA_ORIENTATION_MAGNETOMETER_ACCURACY_MEDIUM,
    HID_USAGE_SENSOR_DATA_ORIENTATION_MAGNETOMETER_ACCURACY_HIGH,
    HID_INPUT(Data_Arr_Abs),
HID_END_COLLECTION,

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)
...

//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_USAGE_SENSOR_DATA_ORIENTATION_MAGNETOMETER_ACCURACY,
HID_LOGICAL_MIN_8(0),
HID_LOGICAL_MAX_8(2),
HID_REPORT_SIZE(8),
HID_REPORT_COUNT(1),

```

```

HID_COLLECTION(Logical),
    HID_USAGE_SENSOR_DATA_ORIENTATION_MAGNETOMETER_ACCURACY_LOW,
    HID_USAGE_SENSOR_DATA_ORIENTATION_MAGNETOMETER_ACCURACY_MEDIUM,
    HID_USAGE_SENSOR_DATA_ORIENTATION_MAGNETOMETER_ACCURACY_HIGH,
    HID_INPUT(Data_Arr_Abs),
HID_END_COLLECTION,

HID_END_COLLECTION
};

```

Add the following to section 4.3.21, page 149-151. Changes in red.

```

// 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)
    ...

    //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_USAGE_SENSOR_DATA_ORIENTATION_MAGNETOMETER_ACCURACY,
    HID_LOGICAL_MIN_8(0),
    HID_LOGICAL_MAX_8(2),
    HID_REPORT_SIZE(8),
    HID_REPORT_COUNT(1),
    HID_COLLECTION(Logical),
        HID_USAGE_SENSOR_DATA_ORIENTATION_MAGNETOMETER_ACCURACY_LOW,
        HID_USAGE_SENSOR_DATA_ORIENTATION_MAGNETOMETER_ACCURACY_MEDIUM,
        HID_USAGE_SENSOR_DATA_ORIENTATION_MAGNETOMETER_ACCURACY_HIGH,
        HID_INPUT(Data_Arr_Abs),
    HID_END_COLLECTION,

    HID_END_COLLECTION
};

```