



USI STYLUS APPLICATION NOTE

July 27, 2016

Legal Notices and Disclaimers

THIS DOCUMENT IS PROVIDED TO YOU “AS IS” WITH NO WARRANTIES WHATSOEVER, WHETHER EXPRESS, IMPLIED, STATUTORY, AT COMMON LAW, OR OTHERWISE. USER ASSUMES THE FULL RISK OF USING THIS DOCUMENT. TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, UNIVERSAL STYLUS INITIATIVE, INC. AND ALL AUTHORS OF THIS DOCUMENT HEREBY DISCLAIM ANY AND ALL WARRANTIES AND CONDITIONS WITH RESPECT TO THIS DOCUMENT AND ITS CONTENT AND THE TRADEMARKS APPEARING IN THIS DOCUMENT, INCLUDING WITHOUT LIMITATION ANY AND ALL WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, SUITABILITY, RELIABILITY, ACCURACY, NON-INFRINGEMENT, VALIDITY OF RIGHTS, AND/OR TITLE. UNIVERSAL STYLUS INITIATIVE, INC. AND THE AUTHORS OF THIS DOCUMENT HEREBY DISCLAIM ANY AND ALL LIABILITY OF ANY KIND, INCLUDING WITHOUT LIMITATION, LIABILITY FOR INFRINGEMENT OF ANY PROPRIETARY RIGHTS AND LIABILITY FOR PRODUCT DEFECTS AND PRODUCTS LIABILITY CLAIMS, RELATING TO USE AND/OR IMPLEMENTATION OF INFORMATION IN THIS DOCUMENT. IN NO EVENT SHALL UNIVERSAL STYLUS INITIATIVE, INC. AND/OR THE AUTHORS OF THIS DOCUMENT BE LIABLE FOR ANY ACTUAL, DIRECT, INDIRECT, INCIDENTAL, CONSEQUENTIAL, SPECIAL, PUNITIVE, EXEMPLARY, OR ENHANCED DAMAGES ARISING FROM YOUR USE AND/OR IMPLEMENTATION OF INFORMATION IN THIS DOCUMENT, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES IN ADVANCE.

THE PROVISION OF THIS DOCUMENT TO YOU DOES NOT PROVIDE YOU, AND NOTHING IN THIS DOCUMENT SHALL BE DEEMED AS GRANTING YOU, WITH ANY LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO USE ANY CONTENT OR TRADEMARKS DESCRIBED OR CONTAINED IN THIS DOCUMENT OR TO ANY INTELLECTUAL PROPERTY RIGHT OWNED OR CONTROLLED BY UNIVERSAL STYLUS INITIATIVE, INC. OR THE AUTHORS OF THIS DOCUMENT. YOU MAY HAVE BEEN GIVEN PERMISSION IN A SEPARATE WRITTEN AGREEMENT WITH UNIVERSAL STYLUS INITIATIVE, INC. TO USE THIS DOCUMENT, AND YOUR USE IS SUBJECT TO SUCH AGREEMENT.

UNIVERSAL STYLUS INITIATIVE®, USI™, USI UNIVERSAL STYLUS INITIATIVE & Pen Design®, Pen Design™, and USI CERTIFIED™ are registered and unregistered trademarks, service marks, and certification marks of Universal Stylus Initiative, Inc. in the United States and other countries. Unauthorized use strictly prohibited.

© 2016 Universal Stylus Initiative, Inc. All rights reserved. Unauthorized use strictly prohibited.

Table of Contents

TABLE OF CONTENTS	3
1 INTRODUCTION	4
1.1 OBJECTIVE OF THIS DOCUMENT	4
1.2 DOCUMENT SCOPE	4
2 USAGE SCENARIOS.....	5
2.1 INTRODUCTION.....	5
2.2 SINGLE STYLUS WITH SINGLE DEVICE	5
2.3 SIMULTANEOUS STYLUS AND TOUCH	6
2.4 SINGLE STYLUS WITH MULTIPLE DEVICES	6
2.5 MULTIPLE STYLUSES WITH A SINGLE DEVICE	7
3 OS/APPLICATION INTERFACE TO A USI STYLUS.....	9
3.1 INTRODUCTION.....	9
3.2 DYNAMIC DISCOVERY	9
3.3 HID REPORTS	9
3.3.1 <i>Data Report</i>	9
3.3.2 <i>Summary of HID Usages in Data Reports</i>	13
3.3.3 <i>Status Report</i>	15
3.3.4 <i>Feature Reports</i>	15
4 DETAILED HID DESCRIPTORS	19
4.1 HID DESCRIPTOR FOR DATA REPORT	19
4.2 HID DESCRIPTOR FOR STATUS REPORT	22
4.3 HID DESCRIPTOR FOR FEATURE REPORTS	22
APPENDIX A – ORIENTATION DATA	27
APPENDIX B – PRESSURE LOG CURVE PROFILE.....	28
APPENDIX C – PREFERRED COLOR INDEX VALUES	29

1 Introduction

A stylus is an input device that works with a computer/tablet/phone screen to provide selection, pointing, drawing and writing capabilities.

The passive stylus is the simplest stylus design, with a tip made of a conductive material. It simply acts as a finger touch.

An active stylus, also known as an ‘electronic pen’, contains circuitry which allows it to produce a more natural and engaging user experience. Active styluses enable fine-grain writing and precision pointing by providing the touch controller a specific means to determine its location, responding to pressure, enhancing navigation while hovering above the screen and manipulating on-screen menus and controls through button presses.

In this document the general term ‘stylus’ refers to the active stylus and includes pen stylus designs.

The ‘USI Stylus’ is an interoperable active stylus that works seamlessly with multiple different touch capable devices whose touch controller (called “device” in this document) conforms to the USI requirements.

1.1 Objective of this document

This document is intended to be shared with an OS vendor and/or a software vendor with the intent to add capabilities in the software stack to support a USI stylus and all its features.

1.2 Document Scope

A USI stylus communicates with a USI controller on a target device. USI specification defines the interface between USI Stylus and a USI controller on the device. Furthermore, the specification also defines the HID interface for the Operating System and software components on the target device to interact with a USI stylus.

This document focuses on the software interactions with a USI stylus.

2 Usage Scenarios

2.1 Introduction

The USI specification supports a number of usage scenarios, including the operation of single and multiple styluses with single and multiple devices, sequentially or concurrently.

One example: a creative writing/sketching session. When the author picks up a stylus and moves it to the device, the actual writing (inking) session starts when the stylus approaches a device. When the stylus is operating on the device, the session is in progress. Of course, the author doesn't expect this progress to stop when he or she moves to another device. Instead, the immediate expectation is that progress will continue as the stylus moves to another device. The stylus needs to operate just as a pencil or a pen does – a pencil/pen doesn't suddenly stop making marks when the author picks up a new piece of paper or starts a sketch on an art pad. Once the author finishes writing and sketching, he or she might put the stylus away and end the creative session.

The exact definition of when an author's session ends is subjective and varies according to the author's preferences, intentions and general behavior.

Generally, a 10 to 15 minute timeout (no use of the stylus on the device surface) is the guideline of when the author's session ends. Sometimes an author has significant gaps in output at the beginning of the effort. However, once it starts, a 10 to 15 minute gap in usage is a viable guideline for the session having ended.

2.2 Single Stylus with Single Device

A single stylus operating with a single device is the most common usage scenario. The single device is a touch capable device, such as a PC, tablet or smart phone.

In a writing session, when a stylus approaches a screen, it shall be detected within a latency period that allows a seamless and fluid user experience. A stylus is detected when it is in close proximity to the screen. The user expects to see some sort of feedback on the screen (usually in the form of a hover cursor). If the stylus is used for writing or drawing, inking needs to start as soon as the stylus tip touches the screen. If the stylus is used for on-screen object selection, then inking needs to start immediately after an icon selection is successful.

A stylus can be used in a variety of ways. Some of the common uses include:

- Writing
 - A student writes a solution to a problem.
 - A business executive takes notes in a meeting.
 - A home user creates a to-do list, and periodically adds items to the list.
 - An author writes a chapter of a new book.
- Annotating documents
 - Documents are edited, transferred for comments, and transferred back for more work, all 'on the fly'.
- Drawing

- Detailed drawings and paintings are created by an artist, with the same variety of brushstrokes, colors, mixture palettes and effects available with physical media – and all in real time (no programming, no extraneous setup).
- Pointing and user interface navigation
 - Manipulating user interfaces that contain millions of small identification points (maps, photos, design graphics), a stylus is capable of much more precision input than fingertips can provide.
- Signature capture
 - Forms are presented in software for clear (legible) user annotations, initials and signatures.
- Gaming
 - More precise inputs than typically possible with mice and game controllers enable new types of games.
 - Some games use forms and precise responses to enable interactive interfaces with the players – such as Sudoku and crossword puzzles

2.3 Simultaneous Stylus and Touch

Most devices that support stylus input also respond to finger touch – often supporting multiple fingers at the same time. In a photo album, for instance, a user might shuffle, organize, move and delete pictures – all using multi-touch gestures (e.g., pinch-to-zoom, hand rotation, flick left, flick right, etc.) by both hands and up to 10 fingers. On some larger screens, such as a lay-flat portable all-in-one, multiple users might use both hands to operate on the device's touch screen.

The user expects that the device's multi-touch capabilities continue to operate when the stylus is being used.

For instance, while an artist is drawing with one hand, he or she might use the other to change colors or to perform gestures to zoom in/out. And the user's latency expectations for the drawing actions remain the same as if the other actions were not taking place.

Of course, during stylus operations, the user also expects to be able to rest his or her palm on the screen without the system sensing the palm as a touch. All palm touches need to be rejected.

2.4 Single Stylus with Multiple Devices

Multiple device usage is normal today. Many users regularly use desktop PCs, laptops, tablets, smartphones, as well as more dedicated devices, including 'smartTVs'/DVRs and automotive navigation/entertainment/communications systems. Users want/need to be able to carry a single stylus and use it with all of their devices. All while, the user expects a consistent experience – performance, latency and otherwise across all of the devices.

When a stylus is in close proximity to a device screen, a hover cursor appears. When the stylus is in contact with the screen, it inks and when it physically leaves the screen, a hover cursor re-appears briefly while the stylus is still in close proximity to the screen. Once it is out of range of that screen and in close proximity to another, a hover cursor appears on the second screen. Finally, when it is in contact with the second screen, the stylus inks on that. If the stylus is in range of both the first and second screens, the stylus continues to work on the screen on which it was last inking as long as it still is in proximity of the screen (as indicated by the hover cursor). If the hover cursor disappears, the stylus can be used with the other screen.

An example of a single stylus being used with multiple devices:

- The user starts by taking notes in a meeting with a tablet. The notes are shared in real time on all of the user's devices.
- The user edits those notes on a PC for a report.
- During a commuter train ride, the user reviews the report on a smartphone and makes minor corrections with the stylus.
- The user does a final review of the report on another PC and adds a signature before submitting the final report.

2.5 Multiple Styluses with a Single Device

Some devices are able to simultaneously support multiple styluses. While this capability might add complexity, it is needed to support expected near-term usages and advanced future scenarios. The latency and performance expectations remain the same as they are with a single stylus on a single device.

A few examples of the need for supporting multiple styluses:

- The user lost the stylus that came with the tablet, picks up another stylus and starts writing (this particular case does not require simultaneous operation).
- Two users, each with a stylus in hand, are collaborating on a presentation; they take turns marking and annotating the document on the same device
- Same as the two-user case above, except that the users want their annotations to be identified uniquely, so they can modify/erase their own content
- On a lay-flat all-in-one that has a multi-touch screen, two to four users play a game, simultaneously using their own styluses to write or draw on the screen.

Depending on its implementation, a device might support one or more styluses at the same time. The user needs a way to determine how many simultaneous styluses the device supports.

For example, if a device only supports four styluses at once, and a fifth stylus arrives, then the fifth will not be able to function. The device either gives an indication of why the fifth stylus does not work or supports user to queries about the device's capabilities.

When one of the four styluses goes out of range, the fifth stylus can be discovered and used.

If the device runs out of some of its other resources before it reaches the advertised limit of the number of styluses, then it provides a notification to the user that no more styluses can be supported. This might happen if one of the stylus is using advanced features that require the transfer of additional information / applications, and thus the device's resources become saturated. Example: an implementation advertises that four styluses are supported and four styluses are detected by the device, but one of the styluses uses special features that increase the communications payload so heavily that data link bandwidth runs out. In this case the implementation might choose only to support two or three styluses until the necessary resources are freed up. A warning message appears to notify the users.

When a device implementation supports multiple styluses, it needs to support at least two styluses, even if they are feature rich styluses that saturate the available resources. And, in that case, the devices needs to issue a warning message to notify the user(s) when additional styluses attempt to write on the device screen.

While one user is writing on a device with a USI Stylus, another user brings in another USI Stylus and makes some annotations on the notes written by the first user. Both USI Styluses work simultaneously, and show the written text and annotations in different (application specific) colors.

USI Styluses can come and go seamlessly on the device and the user's experience with the stylus that is inking is not impacted. Additional styluses continue to work until the device reaches a limit of what its implementation can support. A new stylus is never able to 'bump off' a stylus that is currently being used on the device. And the user is always able to find out how many simultaneous styluses an implementation can support.

3 OS/Application Interface to a USI Stylus

3.1 Introduction

Because of its broad availability, the Human Interface Device (HID) specification defined by the Universal Serial Bus (USB) Implementers' Forum is the software protocol that allows all of the USI stylus capabilities be exposed to the Operating System.

Only the HID descriptor, and not the device descriptor, is defined in the USI Technical Specification. The device descriptor depends on vendor values and hardware transport details that are determined by the implementation.

The API for applications to use the HID data is defined by the host operating system vendor. Some aspects of the data descriptor might not be supported by some operating systems and/or some applications.

3.2 Dynamic Discovery

HID discovery is static (it occurs only once, at controller device initialization), so HID usages cannot be updated to reflect the capabilities of a stylus when it is paired. To overcome this, the default descriptor includes a report for all possible HID usages for a USI compliant stylus. This allows a newly discovered stylus to provide a minimum set or a full range of all possible data, depending on the stylus' capabilities. Any HID usages not provided with the currently in-use stylus are filled with default values (as specified below).

3.3 HID Reports

HID reports are used for host communications. There are three sets of HID reports - data, status and feature. These are described in detail in the next three sections.

The Report ID is only an example of several possible IDs; it can be changed to another ID, as appropriate. While there is one report ID for data and one for status, there are multiple feature report IDs.

3.3.1 Data Report

The data report provides a maximal set of stylus data items. This data report is used for all stylus data transactions. Usages that are only known after a stylus has been detected are given default ranges and values. Values can be expected to be equally spaced.

3.3.1.1 Coordinates

(Usage(X) and Usage(Y))

X and Y represents the physical position of the stylus on the sensing surface. It is a reasonable expectation that logical maximum and minimums and physical maximums and minimums for X and Y coordinates are correctly filled in.

3.3.1.2 Transducer Index

(Usage(Transducer Index))

When multiple styluses are simultaneously tracked, the Transducer Index indicates which stylus produced the report. A unique Transducer Index is assigned to a stylus when it is detected and persists until it leaves range. The Transducer Index is arbitrary; its values may be reused.

The Logical Maximum of the Transducer Index usage in the HID descriptor is equal to the maximum number of simultaneous styluses supported by the controller. This still leaves one more index (zero) than the maximum number of styluses so that it is still possible to avoid immediately reusing the same index number.

3.3.1.3 Tip Pressure

(Usage(Tip Pressure))

The Tip Pressure value output by the stylus is a 12 bit unsigned value. The relationship between this value and physical force applied is shown in Appendix B – Pressure Log Curve Profile.

3.3.1.4 Barrel Pressure

(Usage(Barrel Pressure))

The Barrel Pressure value output by the stylus is an 8 bit unsigned value. If the stylus does not support barrel pressure, the value reported is set to maximum (255). By defaulting Barrel Pressure to maximum, the full effect of a brush dynamic is applied if barrel pressure is unsupported.

3.3.1.5 Tip Switch

(Usage(Tip Switch))

The Tip Switch indicates that force is applied to the stylus tip. If the tip pressure reported by the stylus is zero, then Tip Switch is set to zero. Otherwise the value of Tip Switch is set to 1.

If the stylus Eraser Ready bit is set, Tip Switch is not set (instead the Eraser bit is 1 when there is force applied to the tip).

3.3.1.6 Barrel Switches

(Usage(Barrel Switch) and Usage(Secondary Barrel Switch))

If one of the barrel switches is dedicated to an eraser affordance (that is, either a tail switch or a barrel button), it is not reported as a barrel switch by the stylus. Instead it is reported as an Eraser Ready bit by the stylus. The secondary barrel switch is a new HID function.

3.3.1.7 Invert

(Usage(Invert))

The stylus provides an Eraser Ready bit thru the Invert HID usage. This bit shall be set when the eraser affordance is in use, meaning either that the tail end of the stylus is closest to the surface (if there is an eraser switch and sensor in the tail) or that the barrel button dedicated to erasing has been pressed.

3.3.1.8 Eraser

(Usage(Eraser)):

Unless the Eraser Ready bit from the stylus is set, the Erase bit is zero.

If the Eraser Ready bit from the stylus is set and the Tip Pressure value reported by the stylus is zero then Eraser is set to zero. Otherwise Eraser is set to 1.

3.3.1.9 In Range

(Usage(In Range)):

A report with the In Range bit set is sent by the controller when a stylus is detected and its capabilities are known. All subsequent reports while receiving data from the stylus shall have In Range set. At least one report with In Range clear is sent when the stylus is no longer detected.

3.3.1.10 Orientation

(Usage(X Tilt), Usage(Y Tilt), and Usage(Twist)):

Any orientation values not supported by the stylus are reported as zero. Tilt values have a physical range from -90.00 to 90.00 degrees (measured between the stylus and screen surface), although they may be reported in radians. Twist values have a physical range from 0.00 to 360.00 degrees, although they may be reported in radians.

The physical and logical ranges are reported as two decimal places of resolution for degrees, or four decimal places for radians. If the stylus reports lower resolution values, these ranges are scaled to meet this requirement.

See Appendix A for more details on Orientation, Tilt and Twist.

3.3.1.11 IMU

Any IMU values not supported by the stylus are reported as zero.

The IMU uses the HID Sensor Usage Page.

See Appendix A for more details on IMU data.

3.3.1.11.1 Accelerometer

(Usage (Acceleration Axis X), Usage (Acceleration Axis Y), and Usage (Acceleration Axis Z))

Accelerometer values from the stylus are placed in the Acceleration usages. Note that since due to gravity there should always be a non-zero value at rest, a reading of 0,0,0 indicates that an accelerometer is not present.

3.3.1.11.2 Gyroscope

(Usage (Angular Velocity X Axis), Usage (Angular Velocity Y Axis), and Usage (Angular Velocity Z Axis))

Gyroscope values from the stylus are placed in the Angular Velocity usages.

3.3.1.11.3 Magnetometer

(Usage (Heading X Axis), Usage (Heading Y Axis), and Usage (Heading Z Axis))

Magnetometer values from the stylus are placed in the Heading usages. Note that since due to the earth's magnetic field there should always be a non-zero value, a reading of 0,0,0 indicates that a magnetometer is not present.

3.3.1.12 Charge Level

(Usage(Battery Strength)):

Charge level is reported in percentage of full capacity. The report is in the range of 0 to 100 percent. If the Stylus does not support reporting the charge level, and only supports reporting low-battery indications, then this value is reported as 1 (1%) when the battery is low and 100 (100%) when the battery is not low.

3.3.1.13 Serial Number

(Usage(Transducer Serial Number)):

The serial number is the 52 bit serial number from the stylus with the 12 bit Vendor ID from the stylus prepended to make a unique 64 bit value.

3.3.1.14 Preferred Color

(Usage(Preferred Color)): (New HID usage)

The preferred color from the stylus capabilities is placed in the Preferred Color. If there is no preferred color from the stylus, the field is filled in with 0xFF (no preference). Preferred color is a new HID function.

Applications are expected to parse this value and honor the preferred color reported by the stylus to change the inking color.

Appendix C – Preferred Color Index Values shows the association between the Preferred Color value and its named color.

3.3.1.15 Preferred Line Width

Usage(Preferred Line Width): (New HID usage 0x5E)

The preferred width from the stylus capabilities is placed in the Preferred Line Width. If there is no preferred line width from the stylus, the field is filled in with 0xFF (no preference). Preferred Line Width is a new HID usage. Line widths are from 1 to 254 in 0.1 mm increments. A width of zero indicates a thin line at any zoom factor. A width of 255 means no preferred width.

Applications are expected to parse this value and honor the preferred line width reported by the stylus to change the line stroke width.

3.3.1.16 Preferred Line Style

(Usage(Preferred Line Style)): (New HID usage 0x70)

The preferred type from the stylus capabilities is placed in the Preferred Line Style. If there is no preferred type from the stylus, the field shall be filled in with (Usage (No preference)) (New HID usage 0x77). Preferred Line Style is a new HID usage.

Styles may be one of the following:

- 1: Ink (Usage (Ink)) (New HID usage 0x72)
- 2: Pencil (Usage (Pencil)) (New HID usage 0x73)
- 3: Highlighter (Usage (Highlighter)) (New HID usage 0x74)
- 4: Chisel Marker (Usage (Chisel Marker)) (New HID usage 0x75)
- 5: Brush (Usage (Brush)) (New HID usage 0x76)
- 6: No Preference (Usage (No Preference)) (New HID usage 0x77)

Applications are expected to parse this value and honor the preferred line style reported by the stylus to change the brush stroke style.

3.3.1.17 Vendor Defined Data and Usage

(Usage Page(Vendor Defined Usage Page 1) Usage(Vendor Usage 2)):

One or two custom 16 bit words received from the stylus are placed into the vendor defined usage at the end of the report. Any unused words are set to zero.

An application should retrieve the value of vendor ID from the Serial Number, and then based on the vendor ID, use vendor specific methods to parse the information in this field.

3.3.2 Summary of HID Usages in Data Reports

All Styluses report the data using the complete HID data report as detailed in the previous section (Section 3.3.1). Some styluses may not support all the data elements in the HID Data report. In such cases, the report will have default values. The following table indicates which data elements an application can expect to be always present, and which data elements may not be supplied, and be filled with default values.

HID Usage	Required	Range	Default
Coordinates - Usage(X) and Usage(Y)	*	0 to tablet max	
Transducer Index	*	0 to max supported styluses	
Tip Pressure	*	0 to 4095	
Barrel Pressure		0 to 255	255
Tip Switch	*	0,1	
Barrel Switch	*	0,1	
Secondary Barrel Switch		0,1	0 (off)
Invert	*	0,1	
Eraser	*	0,1	
In Range	*	0,1	
X Tilt		-90.00° to +90.00° (can also be in radians)	0.00° (vertical)
Y Tilt		-90.00° to +90.00° (can also be in radians)	0.00° (vertical)
Twist		0.00° to 360.00° (can also be in radians)	0.00°
Acceleration Axis X		-2047 to +2047	0
Acceleration Axis Y		-2047 to +2047	0
Acceleration Axis Z		-2047 to +2047	0
Angular Velocity X		-2047 to +2047	0
Angular Velocity Y		-2047 to +2047	0
Angular Velocity Z		-2047 to +2047	0
Heading X		-2047 to +2047	0
Heading Y		-2047 to +2047	0
Heading Z		-2047 to +2047	0
Battery Strength	*	0% to 100% (can be 1%,100%)	
Transducer Serial Number	*	all	
Preferred Color		0 to 140, 255 (see Appendix C)	255 (no preference)
Preferred Line Width		0, 1 to 254, 255	255 (no preference)
Preferred Line Style		1 to 6 (see enumerations for styles)	6 (no preference)
Vendor Usage		vendor defined	0

3.3.3 Status Report

Usage (Digitizer Error): (New HID Usage 0x81 on the Digitizer usage page)

All controllers include the status report descriptor specified below to indicate error conditions.

The Status Report uses the HID Digitizer Error usage array (New usage 0x81 on the Digitizer usage page). One of the following error usages array index is sent:

- 1: Normal Operation
(Usage (Err Normal Status): (New HID Usage 0x82) - operating within specification (there is no requirement to indicate this condition). This might be sent to indicate an error has cleared.
- 2: Too many transducers
(Usage (Err Transducers Exceeded): (New HID Usage 0x83) – more styluses are in range than can be supported.
- 3: Unable to support all stylus features
(Usage (Err Full Trans Features Unavail): (New HID Usage 0x84) – usually this indicates the touch controller was unable to provide enough bandwidth to transfer all stylus data and could only support higher priority data.
- 4: The energy source is running low
(Usage (Err Charge Low): (New HID Usage 0x85) –this indicates the stylus' battery or super-cap needs to be recharged.

There is also a Transducer Index usage. This usage is filled in with the Transducer Index of a stylus reporting a low charge. Otherwise this usage is undefined.

It is expected that a user level application or the OS provides a way to display the messages to the user, if the user wishes to receive these notifications.

3.3.4 Feature Reports

Diagnostic and stylus information is available through a series of HID feature reports.

Report IDs in the sample descriptor are for illustration only and may be changed. Feature reports are discovered thru their HID usages.

HID Get Feature Report and HID Set Feature Report commands are issued to obtain the information.

3.3.4.1.1 Diagnostic Command

(Usage (Digitizer Diagnostic): (New HID Usage 0x80 on the Digitizer usage page)

The diagnostic command allows a host based test application to issue commands to a stylus and receive the results. The touch controller forwards the command to the stylus and returns the result.

The exact contents of the return value are defined by the USI Technical Specification.

This is a new HID usage.

3.3.4.1.2 Set Transducer Index Selector

Usage (Transducer Index):

The Get features below get a feature for a given transducer. Since HID does not allow any parameters for a get feature, the single byte value of the Set Transducer Index Selector informs the touch controller which transducer index the following get feature applies to. The requester can do the Set Transducer

Index Selector to a specific value and can do multiple subsequent Get feature requests. The transducer index in the get feature should be tested to insure it matches the requested index in Set Transducer Index Selector, as another process could potentially issue a Set Transducer Index Selector in the middle of a Set index Get feature sequence.

3.3.4.1.3 Get Stylus Firmware Info

Usage (Transducer Software Info): (New HID Usage 0x90)

The Get Stylus Firmware Information command returns the vendor, product, and firmware version information. This can be used to determine when a firmware update is needed.

The command contains a one byte Transducer Index, 16 bits of Vendor ID (Usage (Transducer Vendor ID)) (new HID usage 0x91), 64 bits of Product ID (Usage (Transducer Product ID)) (new HID usage 0x92), and a Software Version report (Usage (Software Version)) (new HID usage 0x2A on the Generic Device usage page) consisting of one byte each of Major (Usage (Major)) (new HID usage 0x2D) and Minor (Usage (Minor)) (new HID usage 0x2E) HID usages. Information is filled in for the transducer number given by the Set Transducer Index Selector feature.

These are new HID usages.

3.3.4.1.4 Get Stylus USI Version

Usage (Protocol Version): (New HID Usage 0x2B Generic Controls Page)

The Get Stylus USI Version command returns the USI version information.

The command contains a one byte Transducer Index and a Protocol Version report consisting of one byte each of Major (Usage (Major)) (usage 0x2A on the Generic Device usage page) and Minor (Usage (Minor)) (usage 0x2B) HID usages.

These are new HID usages.

3.3.4.1.5 Get/Set Preferred Color

(Usage (Protocol Version)): (New HID Usage 0x5C Digitizer Page)

The command contains a one byte Transducer Index HID usage, one byte Preferred Color usage (Usage (Preferred Color)) (usage 0x5C), and one bit Preferred Color is Locked usage (Usage (Preferred Color is Locked)) (usage 0x5D) with seven upper bits of padding.

On a HID GetFeature the Preferred Color and Preferred Color is Locked values are filled in for the transducer number indicated by the Set Transducer Index Selector feature.

On a HID SetFeature a command is issued for the transducer number indicated in the Transducer Index usage with the Preferred Color indicated in the Preferred Color usage. If the preferred color is not locked, this stores the Preferred Color in the stylus for future retrieval. The Preferred Color is Locked usage is ignored.

These are new HID usages.

3.3.4.1.6 Get/Set Preferred Line Width

(Usage (Preferred Line Width)): (New HID Usage 0x5E Digitizer Page)

The command contains a one byte Transducer Index HID usage, one byte Preferred Line Width usage (Usage (Preferred Line Width)) (usage 0x5E), and one bit Preferred Line Width is Locked usage (Usage Preferred Line Width is Locked)) (usage 0x5F) with seven upper bits of padding.

On a HID GetFeature the Preferred Line Width and Preferred Line Width is Locked values are filled in with the results for the transducer number indicated by the Set Transducer Index Selector feature.

On a HID SetFeature a command is issued for the transducer number indicated in the Transducer Index usage with the Preferred Width indicated in the Preferred Width usage. . If the preferred line width is not locked, this stores the Preferred Line Width in the stylus for future retrieval. The Preferred Width is Locked usage is ignored.

These are new HID usages.

3.3.4.1.7 Get/Set Preferred Line Style

(Usage (Preferred Line Style)): (New HID Usage 0x70 Digitizer Page)

The command contains a one byte Transducer Index HID usage, one byte Preferred Line Style usage (Usage (Preferred Line Style)) (usage 0x70), and one bit Preferred Line Style is Locked usage (Usage Preferred Line Style is Locked)) (usage 0x71) with seven upper bits of padding.

On a HID GetFeature the Preferred Line Style and Preferred Line Style is Locked values are filled in for the transducer number indicated by the Set Transducer Index Selector feature.

On a HID SetFeature a command is issued for the transducer number indicated in the Transducer Index usage with the Preferred Style indicated in the Preferred Style usage. . If the preferred style is not locked, this stores the Preferred Style in the stylus for future retrieval. The Preferred Style is Locked usage is ignored.

Styles may be one of the following:

- 1: Ink (Usage (Ink)) (New HID usage 0x72)
- 2: Pencil (Usage (Pencil)) (New HID usage 0x73)
- 3: Highlighter (Usage (Highlighter)) (New HID usage 0x74)
- 4: Chisel Marker (Usage (Chisel Marker)) (New HID usage 0x75)
- 5: Brush (Usage (Brush)) (New HID usage 0x76)
- 6: No Preference (Usage (No Preference)) (New HID usage 0x77)

These are new HID usages.

3.3.4.1.8 Get/Set Buttons

(Usage (Barrel Switch))

The command contains a one byte Transducer Index HID usage, and a three-byte array of button definitions. Each array element is tagged with one of the button usages as follows:

- Barrel Switch (Usage 0x44): The barrel switch closest to the stylus tip
- Secondary Barrel Switch (Usage 0x5A): The barrel switch second closest to the stylus tip.
- Eraser (Usage 0x45): The switch on the tail, or a barrel switch dedicated to an erase function.

Each of the three switches can take on one of the following switch function usages:

- Switch Unimplemented (new HID Usage 0xA4): This reports a switch as “missing”, such as the secondary barrel switch for a stylus with only one barrel switch. A switch cannot be Set to Unimplemented.
- Barrel Switch (Usage 0x44): The switch will act as if it were the barrel switch.
- Secondary Barrel Switch (Usage 0x5A): The switch will act as if it is the secondary barrel switch.
- Eraser (Usage 0x45): The switch will provide an Eraser Ready indicator.
- Switch Disabled (new HID Usage 0xA3): The switch exists but will not send any status to the host.

On a HID GetFeature the button values are filled in with the results for the transducer number indicated by the Set Transducer Index Selector feature.

On a HID SetFeature a command is issued for the transducer number indicated in the Transducer Index usage and all three buttons are set.

3.3.4.1.9 Get/Set Vendor Extension

(Usage (Vendor Usage 1)): (0x01 Vendor Define Page)

The command contains a one byte Transducer Index HID usage and a one word vendor specific usage.

On a HID GetFeature the word of vendor specific usage is filled in with the results for the transducer number indicated in the Set Transducer Index Selector feature.

On a HID SetFeature, a command is issued for the transducer number indicated in the Transducer Index usage with the word vendor specific usage placed in the VendorExtension field

4 Detailed HID Descriptors

4.1 HID Descriptor for Data Report

The following is the HID descriptor for the data report that includes all the usages supported by a USI Stylus. A USI controller supports this descriptor and sends the HID input reports that are conformant with this descriptor.

```

0x05, 0x0d, // USAGE_PAGE (Digitizers)
0x09, 0x02, // USAGE (Pen)
0xa1, 0x01, // COLLECTION (Application)
0x09, 0x20, //   USAGE (Stylus)
0xa1, 0x00, //   COLLECTION (Physical)
0x85, HID_REPORTID_TABLET, //     REPORT_ID ( )
0x05, 0x01, //     USAGE_PAGE (Generic Desktop)
0xa4, //     PUSH
0x09, 0x30, //     USAGE (X)
0x35, 0x00, //     PHYSICAL_MINIMUM (0)
0x46, 0x3a, 0x20, //     PHYSICAL_MAXIMUM (8250)
0x15, 0x00, //     LOGICAL_MINIMUM (0)
0x26, 0xf8, 0x52, //     LOGICAL_MAXIMUM (21240)
0x55, 0x0d, //     UNIT_EXPONENT (-3)
0x65, 0x13, //     UNIT (Eng Lin:Distance)
0x75, 0x10, //     REPORT_SIZE (16)
0x95, 0x01, //     REPORT_COUNT (1)
0x81, 0x02, //     INPUT (Data,Var,Abs)
0x09, 0x31, //     USAGE (Y)
0x46, 0x2c, 0x18, //     PHYSICAL_MAXIMUM (6188)
0x26, 0x6c, 0x3e, //     LOGICAL_MAXIMUM (15980)
0x81, 0x02, //     INPUT (Data,Var,Abs)
0xb4, //     POP
0x05, 0x0d, //     USAGE_PAGE (Digitizers)
0x09, 0x38, //     USAGE (Transducer Index)
0x95, 0x01, //     REPORT_COUNT (1)
0x75, 0x08, //     REPORT_SIZE (8)
0x15, 0x00, //     LOGICAL_MINIMUM (0)
0x25, MAX_SUPPORTED_STYLI, //     LOGICAL_MAXIMUM (2)
0x81, 0x02, //     INPUT (Data,Var,Abs)
0x09, 0x30, //     USAGE (Tip Pressure)
0x75, 0x10, //     REPORT_SIZE (16)
0x26, 0xff, 0x0f, //     LOGICAL_MAXIMUM (4095)
0x81, 0x02, //     INPUT (Data,Var,Abs)
0x09, 0x31, //     USAGE (Barrel Pressure)
0x81, 0x02, //     INPUT (Data,Var,Abs)
0x09, 0x42, //     USAGE (Tip Switch)
0x09, 0x44, //     USAGE (Barrel Switch)
0x09, 0x5a, //     USAGE (Secondary Barrel Switch)
0x09, 0x3c, //     USAGE (Invert)
0x09, 0x45, //     USAGE (Eraser)
0x09, 0x32, //     USAGE (In Range)
0x75, 0x01, //     REPORT_SIZE (1)
0x95, 0x06, //     REPORT_COUNT (6)

```

```

0x25, 0x01, // LOGICAL_MAXIMUM (1)
0x81, 0x02, // INPUT (Data,Var,Abs)
0x95, 0x02, // REPORT_COUNT (2)
0x81, 0x03, // INPUT (Cnst,Var,Abs)
0x09, 0x3d, // USAGE (X Tilt)
0x55, 0x0e, // UNIT_EXPONENT (-2)
0x65, 0x14, // UNIT (Eng Rot:Angular Pos)
0x36, 0xd8, 0xdc, // PHYSICAL_MINIMUM (-9000)
0x46, 0x28, 0x23, // PHYSICAL_MAXIMUM (9000)
0x16, 0xd8, 0xdc, // LOGICAL_MINIMUM (-9000)
0x26, 0x28, 0x23, // LOGICAL_MAXIMUM (9000)
0x95, 0x01, // REPORT_COUNT (1)
0x75, 0x10, // REPORT_SIZE (16)
0x81, 0x02, // INPUT (Data,Var,Abs)
0x09, 0x3e, // USAGE (Y Tilt)
0x81, 0x02, // INPUT (Data,Var,Abs)
0x09, 0x41, // USAGE (Twist)
0x15, 0x00, // LOGICAL_MINIMUM (0)
0x27, 0xa0, 0x8c, 0x00, 0x00, // LOGICAL_MAXIMUM (36000)
0x35, 0x00, // PHYSICAL_MINIMUM (0)
0x47, 0xa0, 0x8c, 0x00, 0x00, // PHYSICAL_MAXIMUM (36000)
0x81, 0x02, // INPUT (Data,Var,Abs)
0x05, 0x20, // USAGE_PAGE (Sensors)
0x0a, 0x53, 0x04, // USAGE (Data Field: Acceleration)
Axis X)
0x65, 0x00, // UNIT (None)
0x16, 0x01, 0xf8, // LOGICAL_MINIMUM (-2047)
0x26, 0xff, 0x07, // LOGICAL_MAXIMUM (2047)
0x75, 0x10, // REPORT_SIZE (16)
0x95, 0x01, // REPORT_COUNT (1)
0x81, 0x02, // INPUT (Data,Var,Abs)
0x0a, 0x54, 0x04, // USAGE (Data Field: Acceleration)
Axis Y)
0x81, 0x02, // INPUT (Data,Var,Abs)
0x0a, 0x55, 0x04, // USAGE (Data Field: Acceleration)
Axis Z)
0x81, 0x02, // INPUT (Data,Var,Abs)
0x0a, 0x57, 0x04, // USAGE (Data Field: Angular)
Velocity Axis X)
0x81, 0x02, // INPUT (Data,Var,Abs)
0x0a, 0x58, 0x04, // USAGE (Data Field: Angular)
Velocity Axis Y)
0x81, 0x02, // INPUT (Data,Var,Abs)
0x0a, 0x59, 0x04, // USAGE (Data Field: Angular)
Velocity Axis Z)
0x81, 0x02, // INPUT (Data,Var,Abs)
0x0a, 0x72, 0x04, // USAGE (Data Field: Heading X Axis)
0x81, 0x02, // INPUT (Data,Var,Abs)
0x0a, 0x73, 0x04, // USAGE (Data Field: Heading Y Axis)
0x81, 0x02, // INPUT (Data,Var,Abs)
0x0a, 0x74, 0x04, // USAGE (Data Field: Heading Z Axis)
0x81, 0x02, // INPUT (Data,Var,Abs)
0x05, 0x0d, // USAGE_PAGE (Digitizers)
0x09, 0x3b, // USAGE (Battery Strength)
0x15, 0x00, // LOGICAL_MINIMUM (0)

```

```

0x25, 0x64, // LOGICAL_MAXIMUM (100)
0x75, 0x08, // REPORT_SIZE (8)
0x81, 0x02, // INPUT (Data,Var,Abs)
0x09, 0x5b, // USAGE (Transducer Serial Number)
0x25, 0xff, // LOGICAL_MAXIMUM (-1)
0x75, 0x40, // REPORT_SIZE (64)
0x81, 0x02, // INPUT (Data,Var,Abs)
0x06, 0x00, 0xff, // USAGE_PAGE (Vendor 1)
0x09, 0x5b, // USAGE (Transducer Serial Number)
0x75, 0x20, // REPORT_SIZE (32)
0x81, 0x02, // INPUT (Data,Var,Abs)
0x05, 0x0d, // USAGE_PAGE (Digitizers)
0x09, 0x5c, // USAGE (Preferred Color)
0x26, 0xff, 0x00, // LOGICAL_MAXIMUM (255)
0x75, 0x08, // REPORT_SIZE (8)
0x81, 0x02, // INPUT (Data,Var,Abs)
0x09, 0x5e, // USAGE (Preferred Line Width)
0x81, 0x02, // INPUT (Data,Var,Abs)
0x09, 0x70, // USAGE (Preferred Line Style)
0xa1, 0x02, // COLLECTION (Logical)
0x15, 0x01, // LOGICAL_MINIMUM (0)
0x25, 0x06, // LOGICAL_MAXIMUM (5)
0x09, 0x72, // USAGE (Ink)
0x09, 0x73, // USAGE (Pencil)
0x09, 0x74, // USAGE (Highlighter)
0x09, 0x75, // USAGE (Chisel Marker)
0x09, 0x76, // USAGE (Brush)
0x09, 0x77, // USAGE (No preference)
0x81, 0x20, // INPUT (Data,Ary,Abs,NPrf)
0xc0, // END_COLLECTION
0x06, 0x00, 0xff, // USAGE_PAGE (Generic Desktop)
0x09, 0x01, // USAGE (Vendor Usage 1)
0x15, 0x00, // LOGICAL_MINIMUM (0)
0x27, 0xff, 0xff, 0x00, 0x00, // LOGICAL_MAXIMUM (65535)
0x75, 0x10, // REPORT_SIZE (16)
0x95, 0x01, // REPORT_COUNT (1)
0x81, 0x02, // INPUT (Data,Var,Abs)
0xc0, // END_COLLECTION

```

4.2 HID Descriptor for Status Report

The following is the HID descriptor for the status report that A USI controller supports for reporting status and error conditions.

```

0x05, 0x0d, // USAGE_PAGE (Digitizers)
0x85, HID_REPORTID_ERROR, // REPORT_ID ( )
0x09, 0x81, // USAGE (Digitizer Error)
0xa1, 0x02, // COLLECTION (Logical)
0x09, 0x38, // USAGE (Transducer Index)
0x75, 0x08, // REPORT_SIZE (8)
0x95, 0x01, // REPORT_COUNT (1)
0x15, 0x00, // LOGICAL_MINIMUM (0)
0x25, MAX_SUPPORTED_STYLI, // LOGICAL_MAXIMUM (2)
0x81, 0x02, // INPUT (Data,Var,Abs)
0x09, 0x81, // USAGE (Digitizer Error)
0x15, 0x01, // LOGICAL_MINIMUM (0)
0x25, 0x04, // LOGICAL_MAXIMUM (3)
0x09, 0x82, // USAGE (Err Normal Status)
0x09, 0x83, // USAGE (Err Transducers Exceeded)
0x09, 0x84, // USAGE (Err Full Trans Features
Unavail)
0x09, 0x85, // USAGE (Err Charge Low)
0x81, 0x20, // INPUT (Data,Ary,Abs,NPrf)
0xc0, // END_COLLECTION

```

4.3 HID Descriptor for Feature Reports

Following is the HID descriptor for the Get/Set Feature Reports.

```

// Get/Set Line Color
0x85, HID_REPORTID_GETSET_COLOR, // REPORT_ID ( )
0x09, 0x5c, // USAGE (Preferred Color)
0xa1, 0x02, // COLLECTION (Logical)
0x15, 0x00, // LOGICAL_MINIMUM (0)
0x26, 0xff, 0x00, // LOGICAL_MAXIMUM (2)
0x75, 0x08, // REPORT_SIZE (8)
0x95, 0x01, // REPORT_COUNT (1)
0x09, 0x38, // USAGE (Transducer Index)
0xb1, 0x02, // FEATURE (Data,Var,Abs)
0x09, 0x5c, // USAGE (Preferred Color)
0x26, 0xff, 0x00, // LOGICAL_MAXIMUM (255)
0xb1, 0x02, // FEATURE (Data,Var,Abs)
0x09, 0x5d, // USAGE (Preferred Color is Locked)
0x75, 0x01, // REPORT_SIZE (1)
0x95, 0x01, // REPORT_COUNT (1)
0x25, 0x01, // LOGICAL_MAXIMUM (1)
0xb1, 0x02, // FEATURE (Data,Var,Abs)
0x95, 0x07, // REPORT_COUNT (7)
0xb1, 0x03, // FEATURE (Cnst,Var,Abs)
0xc0, // END_COLLECTION

// Get/Set Line Width

```

```

0x85, HID_REPORTID_GETSET_WIDTH,//
0x09, 0x5e, //
0xa1, 0x02, //
0x09, 0x38, //
0x15, 0x00, //
0x25, MAX_SUPPORTED_STYLI, //
0x75, 0x08, //
0x95, 0x01, //
0xb1, 0x02, //
0x09, 0x5e, //
0x26, 0xff, 0x00, //
0xb1, 0x02, //
0x09, 0x5f, //
Locked)
0x75, 0x01, //
0x25, 0x01, //
0xb1, 0x02, //
0x75, 0x07, //
0xb1, 0x03, //
0xc0, //

// Get/Set Line Style
0x85, HID_REPORTID_GETSET_STYLE,//
0x09, 0x70, //
0xa1, 0x02, //
0x75, 0x08, //
0x95, 0x01, //
0x15, 0x00, //
0x25, MAX_SUPPORTED_STYLI, //
0x09, 0x38, //
0xb1, 0x02, //
0x09, 0x70, //
0xa1, 0x02, //
0x25, 0x06, //
0x09, 0x72, //
0x09, 0x73, //
0x09, 0x74, //
0x09, 0x75, //
0x09, 0x76, //
0x09, 0x77, //
0xb1, 0x20, //
0xc0, //
0x09, 0x71, //
Locked)
0x75, 0x01, //
0x25, 0x01, //
0xb1, 0x02, //
0x75, 0x07, //
0xb1, 0x03, //
0xc0, //

// Get/Set Diagnostic
0x85, HID_REPORTID_DIAGNOSE, //
0x09, 0x80, //
0x15, 0x00, //

```

```

REPORT_ID ( )
USAGE (Preferred Line Width)
COLLECTION (Logical)
    USAGE (Transducer Index)
    LOGICAL_MINIMUM (0)
    LOGICAL_MAXIMUM (2)
    REPORT_SIZE (8)
    REPORT_COUNT (1)
    FEATURE (Data,Var,Abs)
    USAGE (Preferred Line Width)
    LOGICAL_MAXIMUM (255)
    FEATURE (Data,Var,Abs)
    USAGE (Preferred Line Width is
REPORT_SIZE (1)
LOGICAL_MAXIMUM (1)
FEATURE (Data,Var,Abs)
REPORT_SIZE (7)
FEATURE (Cnst,Var,Abs)
END_COLLECTION

REPORT_ID ( )
USAGE (Preferred Line Style)
COLLECTION (Logical)
    REPORT_SIZE (8)
    REPORT_COUNT (1)
    LOGICAL_MINIMUM (0)
    LOGICAL_MAXIMUM (2)
    USAGE (Transducer Index)
    FEATURE (Data,Var,Abs)
    USAGE (Preferred Line Style)
    COLLECTION (Logical)
        LOGICAL_MAXIMUM (5)
        USAGE (Ink)
        USAGE (Pencil)
        USAGE (Highlighter)
        USAGE (Chisel Marker)
        USAGE (Brush)
        USAGE (No preference)
        FEATURE (Data,Ary,Abs,NPrf)
    END_COLLECTION
    USAGE (Preferred Line Style is
REPORT_SIZE (1)
LOGICAL_MAXIMUM (1)
FEATURE (Data,Var,Abs)
REPORT_SIZE (7)
FEATURE (Cnst,Var,Abs)
END_COLLECTION

REPORT_ID ( )
USAGE (Digitizer Diagnostic)
LOGICAL_MINIMUM (0)

```

```

0x25, 0xff, // LOGICAL_MAXIMUM (-1)
0x75, 0x40, // REPORT_SIZE (64)
0x95, 0x01, // REPORT_COUNT (1)
0xb1, 0x02, // FEATURE (Data,Var,Abs)

// Get/Set Buttons
0x85, HID_REPORTID_GETSET_BUTTONS, // REPORT_ID ( )
0x09, 0x44, // USAGE (Barrel Switch)
0xa1, 0x02, // COLLECTION (Logical)
0x09, 0x38, // USAGE (Transducer Index)
0x75, 0x08, // REPORT_SIZE (8)
0x95, 0x01, // REPORT_COUNT (1)
0x25, MAX_SUPPORTED_STYLI, // LOGICAL_MAXIMUM (2)
0xb1, 0x02, // FEATURE (Data,Var,Abs)
0x15, 0x01, // LOGICAL_MINIMUM (0)
0x25, 0x03, // LOGICAL_MAXIMUM (5)
0x09, 0x44, // USAGE (Barrel Switch)
0xa1, 0x02, // COLLECTION (Logical)
0x09, 0xa4, // USAGE (Switch Unimplemented)
0x09, 0x44, // USAGE (Barrel Switch)
0x09, 0x5a, // USAGE (Secondary Barrel Switch)
0x09, 0x45, // USAGE (Eraser)
0x09, 0xa3, // USAGE (Switch Disabled)
0xb1, 0x20, // FEATURE (Data,Ary,Abs,NPrf)
0xc0, // END_COLLECTION
0x09, 0x5a, // USAGE (Secondary Barrel Switch)
0xa1, 0x02, // COLLECTION (Logical)
0x09, 0xa4, // USAGE (Switch Unimplemented)
0x09, 0x44, // USAGE (Barrel Switch)
0x09, 0x5a, // USAGE (Secondary Barrel Switch)
0x09, 0x45, // USAGE (Eraser)
0x09, 0xa3, // USAGE (Switch Disabled)
0xb1, 0x20, // FEATURE (Data,Ary,Abs,NPrf)
0xc0, // END_COLLECTION
0x09, 0x45, // USAGE (Eraser)
0xa1, 0x02, // COLLECTION (Logical)
0x09, 0xa4, // USAGE (Switch Unimplemented)
0x09, 0x44, // USAGE (Barrel Switch)
0x09, 0x5a, // USAGE (Secondary Barrel Switch)
0x09, 0x45, // USAGE (Eraser)
0x09, 0xa3, // USAGE (Switch Disabled)
0xb1, 0x20, // FEATURE (Data,Ary,Abs,NPrf)
0xc0, // END_COLLECTION
0xc0, // END_COLLECTION

// Get Firmware Version
0x85, HID_REPORTID_GET_FIRMWARE, // REPORT_ID ( )
0x75, 0x08, // REPORT_SIZE (8)
0x95, 0x01, // REPORT_COUNT (1)
0x05, 0x0d, // USAGE_PAGE (Digitizers)
0x09, 0x90, // USAGE (Transducer Software Info.)
0xa1, 0x02, // COLLECTION (Logical)
0x09, 0x38, // USAGE (Transducer Index)
0x25, MAX_SUPPORTED_STYLI, // LOGICAL_MAXIMUM (2)
0xb1, 0x02, // FEATURE (Data,Var,Abs)

```



```

0x09, 0x91, // USAGE (Transducer Vendor ID)
0x75, 0x10, // REPORT_SIZE (16)
0x26, 0xff, 0x0f, // LOGICAL_MAXIMUM (4095)
0xb1, 0x02, // FEATURE (Data,Var,Abs)
0x09, 0x92, // USAGE (Transducer Product ID)
0x75, 0x40, // REPORT_SIZE (64)
0x25, 0xff, // LOGICAL_MAXIMUM (-1)
0xb1, 0x02, // FEATURE (Data,Var,Abs)
0x05, 0x06, // USAGE_PAGE (Generic Device)
0x09, 0x2a, // USAGE (Software Version)
0x75, 0x08, // REPORT_SIZE (8)
0x26, 0xff, 0x00, // LOGICAL_MAXIMUM (255)
0xa1, 0x02, // COLLECTION (Logical)
0x09, 0x2d, // USAGE (Major)
0xb1, 0x02, // FEATURE (Data,Var,Abs)
0x09, 0x2e, // USAGE (Minor)
0xb1, 0x02, // FEATURE (Data,Var,Abs)
0xc0, // END_COLLECTION
0xc0, // END_COLLECTION

// Get USI Version
0x85, HID_REPORTID_GET_PROTOCOL, // REPORT_ID ( )
0x05, 0x06, // USAGE_PAGE (Generic Device)
0x09, 0x2b, // USAGE (Protocol Version)
0xa1, 0x02, // COLLECTION (Logical)
0x05, 0x0d, // USAGE_PAGE (Digitizers)
0x25, MAX_SUPPORTED_STYLI, // LOGICAL_MAXIMUM (2)
0x09, 0x38, // USAGE (Transducer Index)
0xb1, 0x02, // FEATURE (Data,Var,Abs)
0x05, 0x06, // USAGE_PAGE (Generic Device)
0x09, 0x2b, // USAGE (Protocol Version)
0xa1, 0x02, // COLLECTION (Logical)
0x09, 0x2d, // USAGE (Major)
0x26, 0xff, 0x00, // LOGICAL_MAXIMUM (255)
0xb1, 0x02, // FEATURE (Data,Var,Abs)
0x09, 0x2e, // USAGE (Minor)
0xb1, 0x02, // FEATURE (Data,Var,Abs)
0xc0, // END_COLLECTION
0xc0, // END_COLLECTION

// Get/Set Vendor Specific
0x85, HID_REPORTID_GETSET_VENDOR, // REPORT_ID ( )
0x06, 0x00, 0xff, // USAGE_PAGE (Generic Desktop)
0x09, 0x01, // USAGE (Vendor Usage 1)
0xa1, 0x02, // COLLECTION (Logical)
0x05, 0x0d, // USAGE_PAGE (Digitizers)
0x09, 0x38, // USAGE (Transducer Index)
0x75, 0x08, // REPORT_SIZE (8)
0x95, 0x01, // REPORT_COUNT (1)
0x25, MAX_SUPPORTED_STYLI, // LOGICAL_MAXIMUM (2)
0xb1, 0x02, // FEATURE (Data,Var,Abs)
0x06, 0x00, 0xff, // USAGE_PAGE (Generic Desktop)
0x09, 0x01, // USAGE (Vendor Usage 1)
0x75, 0x10, // REPORT_SIZE (16)
0x27, 0xff, 0xff, 0x00, 0x00, // LOGICAL_MAXIMUM (65535)

```

```

0xb1, 0x02,          //      FEATURE (Data,Var,Abs)
0xc0,                //      END_COLLECTION

// Set Select Transducer Index
0x85, HID_REPORTID_SET_TRANSDUCER, // REPORT_ID ( )
0x05, 0x0d,          //      USAGE_PAGE (Digitizers)
0x09, 0x38,          //      USAGE (Transducer Index)
0x75, 0x08,          //      REPORT_SIZE (8)
0x95, 0x01,          //      REPORT_COUNT (1)
0x15, 0x00,          //      LOGICAL_MINIMUM (0)
0x25, MAX_SUPPORTED_STYLI, //      LOGICAL_MAXIMUM (2)
0xb1, 0x02,          //      FEATURE (Data,Var,Abs)

0xc0                // END_COLLECTION

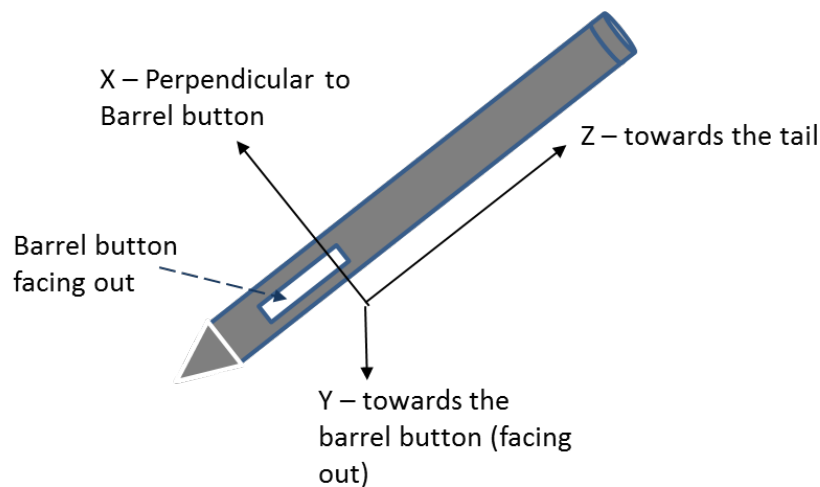
```

Appendix A – Orientation Data

The orientation data provided by a USI Stylus may contain tilt, twist, and IMU data.

Orientation data is represented as follows:

- X tilt and Y tilt:**
 Tilt is reported as degrees relative to the device surface. Zero degrees is vertical to the device surface. -90.00 degrees X tilt is to the left. 90.00 degrees X tilt is to the right. -90.00 degrees Y tilt is away from the user. 90.00 degrees Y tilt is toward the user. The resolution is approximately 0.05 degrees. If tilt is not supported the tilt is reported as 0,0 (vertical to the surface).
- Twist:**
 Twist is reported as degrees between 0.00 and 360.00 with an approximate resolution of 0.09 degrees. If twist is not supported it is reported as a zero degrees.
- IMU:**
 Inertial measurement units report orientation relative to the earth. For consistency, positive Z is toward the tail, positive Y is toward the barrel buttons, and positive X is tangent to the barrel buttons in a clockwise direction.



Complete IMU data (all sensors for all axis) is measured as simultaneously as possible.

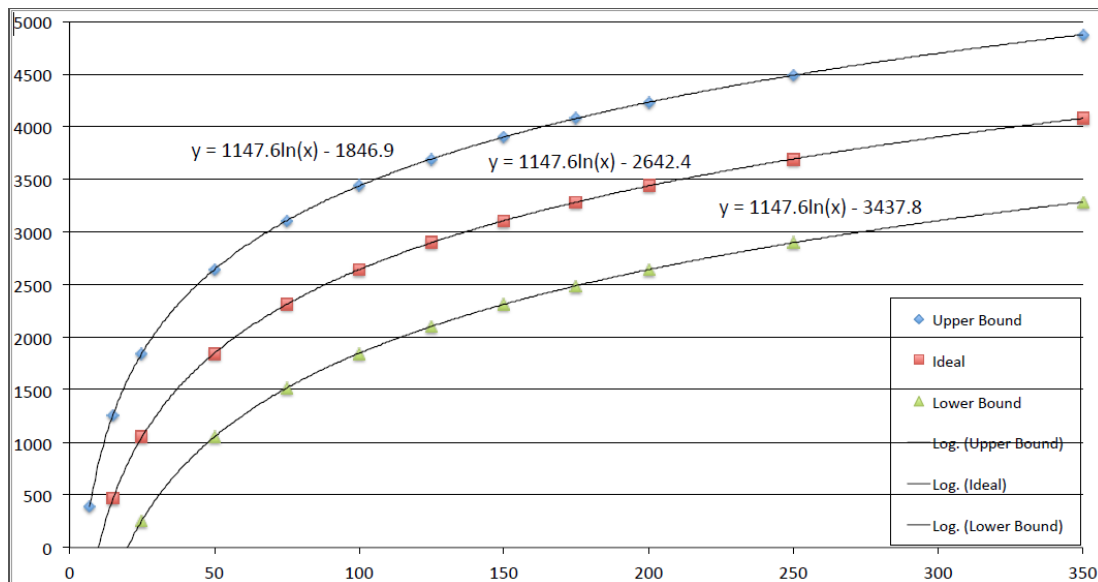
Values are reported as signed 12 bit numbers. The units are as follows

- Accelerometer: reported in mG (milli G)
- Gyroscope: reported in dps (degrees per second)
- Magnetometer: reported in milli Gauss

Appendix B – Pressure Log Curve Profile

The following graphs represent the log curve profiles that are used by the stylus to convert the raw gram force value (X-axis) to the unsigned value (Y-axis) reported by the Stylus.

The graph shows an ideal curve (in middle) and the upper bounds and the lower bounds of the curves. The actual curve used by the Stylus is within these bounds.



The constants used in creating the above profile are stated below.

Upper Bound : $y = 1147.6 \ln(x) - 1846.9$

Ideal : $y = 1147.6 \ln(x) - 2642.4$

Lower Bound : $y = 1147.6 \ln(x) - 3437.8$

The reported levels of pressure falls within the tolerance described above.

A logarithmic fit to the data has an r^2 goodness of fit of 90% or greater.

A linear fit to the data has an r^2 goodness of fit of less than the logarithmic goodness of fit.

Appendix C – Preferred Color Index Values

The following table defines the 8-bit index value to be used for the PreferredColor. The indices correspond to the keywords defined in <http://www.w3.org/TR/SVG/types.html#ColorKeywords> and cover major colors.

Table C-1 8-bit Index values for PreferredColor

Index	Name	RGBValues
0	AliceBlue	#F0F8FF
1	AntiqueWhite	#FAEBD7
2	Aqua	#00FFFF
3	Aquamarine	#7FFFD4
4	Azure	#F0FFFF
5	Beige	#F5F5DC
6	Bisque	#FFE4C4
7	Black	#000000
8	BlanchedAlmond	#FFEBCD
9	Blue	#0000FF
10	BlueViolet	#8A2BE2
11	Brown	#A52A2A
12	BurlyWood	#DEB887
13	CadetBlue	#5F9EA0
14	Chartreuse	#7FFF00
15	Chocolate	#D2691E
16	Coral	#FF7F50
17	CornflowerBlue	#6495ED
18	Cornsilk	#FFF8DC
19	Crimson	#DC143C
20	Cyan	#00FFFF

Index	Name	RGBValues
21	DarkBlue	#00008B
22	DarkCyan	#008B8B
23	DarkGoldenRod	#B8860B
24	DarkGray	#A9A9A9
25	DarkGreen	#006400
26	DarkKhaki	#BDB76B
27	DarkMagenta	#8B008B
28	DarkOliveGreen	#556B2F
29	DarkOrange	#FF8C00
30	DarkOrchid	#9932CC
31	DarkRed	#8B0000
32	DarkSalmon	#E9967A
33	DarkSeaGreen	#8FBC8F
34	DarkSlateBlue	#483D8B
35	DarkSlateGray	#2F4F4F
36	DarkTurquoise	#00CED1
37	DarkViolet	#9400D3
38	DeepPink	#FF1493
39	DeepSkyBlue	#00BFFF
40	DimGray	#696969
41	DodgerBlue	#1E90FF
42	FireBrick	#B22222
43	FloralWhite	#FFFAF0
44	ForestGreen	#228B22
45	Fuchsia	#FF00FF
46	Gainsboro	#DCDCDC
47	GhostWhite	#F8F8FF
48	Gold	#FFD700
49	GoldenRod	#DAA520
50	Gray	#808080
51	Green	#008000

Index	Name	RGBValues
52	GreenYellow	#ADFF2F
53	HoneyDew	#F0FFF0
54	HotPink	#FF69B4
55	IndianRed	#CD5C5C
56	Indigo	#4B0082
57	Ivory	#FFFFFF0
58	Khaki	#F0E68C
59	Lavender	#E6E6FA
60	LavenderBlush	#FFF0F5
61	LawnGreen	#7CFC00
62	LemonChiffon	#FFFACD
63	LightBlue	#ADD8E6
64	LightCoral	#F08080
65	LightCyan	#E0FFFF
66	LightGoldenRodYellow	#FAFAD2
67	LightGray	#D3D3D3
68	LightGreen	#90EE90
69	LightPink	#FFB6C1
70	LightSalmon	#FFA07A
71	LightSeaGreen	#20B2AA
72	LightSkyBlue	#87CEFA
73	LightSlateGray	#778899
74	LightSteelBlue	#B0C4DE
75	LightYellow	#FFFFE0
76	Lime	#00FF00
77	LimeGreen	#32CD32
78	Linen	#FAF0E6
79	Magenta	#FF00FF
80	Maroon	#800000
81	MediumAquaMarine	#66CDAA
82	MediumBlue	#0000CD

Index	Name	RGBValues
83	MediumOrchid	#BA55D3
84	MediumPurple	#9370DB
85	MediumSeaGreen	#3CB371
86	MediumSlateBlue	#7B68EE
87	MediumSpringGreen	#00FA9A
88	MediumTurquoise	#48D1CC
89	MediumVioletRed	#C71585
90	MidnightBlue	#191970
91	MintCream	#F5FFFA
92	MistyRose	#FFE4E1
93	Moccasin	#FFE4B5
94	NavajoWhite	#FFDEAD
95	Navy	#000080
96	OldLace	#FDF5E6
97	Olive	#808000
98	OliveDrab	#6B8E23
99	Orange	#FFA500
100	OrangeRed	#FF4500
101	Orchid	#DA70D6
102	PaleGoldenRod	#EEE8AA
103	PaleGreen	#98FB98
104	PaleTurquoise	#AFEEEE
105	PaleVioletRed	#DB7093
106	PapayaWhip	#FFEFD5
107	PeachPuff	#FFDAB9
108	Peru	#CD853F
109	Pink	#FFC0CB
110	Plum	#DDA0DD
111	PowderBlue	#B0E0E6
112	Purple	#800080
113	RebeccaPurple	#663399

Index	Name	RGBValues
114	Red	#FF0000
115	RosyBrown	#BC8F8F
116	RoyalBlue	#4169E1
117	SaddleBrown	#8B4513
118	Salmon	#FA8072
119	SandyBrown	#F4A460
120	SeaGreen	#2E8B57
121	SeaShell	#FFF5EE
122	Sienna	#A0522D
123	Silver	#C0C0C0
124	SkyBlue	#87CEEB
125	SlateBlue	#6A5ACD
126	SlateGray	#708090
127	Snow	#FFFAFA
128	SpringGreen	#00FF7F
129	SteelBlue	#4682B4
130	Tan	#D2B48C
131	Teal	#008080
132	Thistle	#D8BFD8
133	Tomato	#FF6347
134	Turquoise	#40E0D0
135	Violet	#EE82EE
136	Wheat	#F5DEB3
137	White	#FFFFFF
138	WhiteSmoke	#F5F5F5
139	Yellow	#FFFF00
140	YellowGreen	#9ACD32
141-254	Reserved	
255	No Preferred Color	