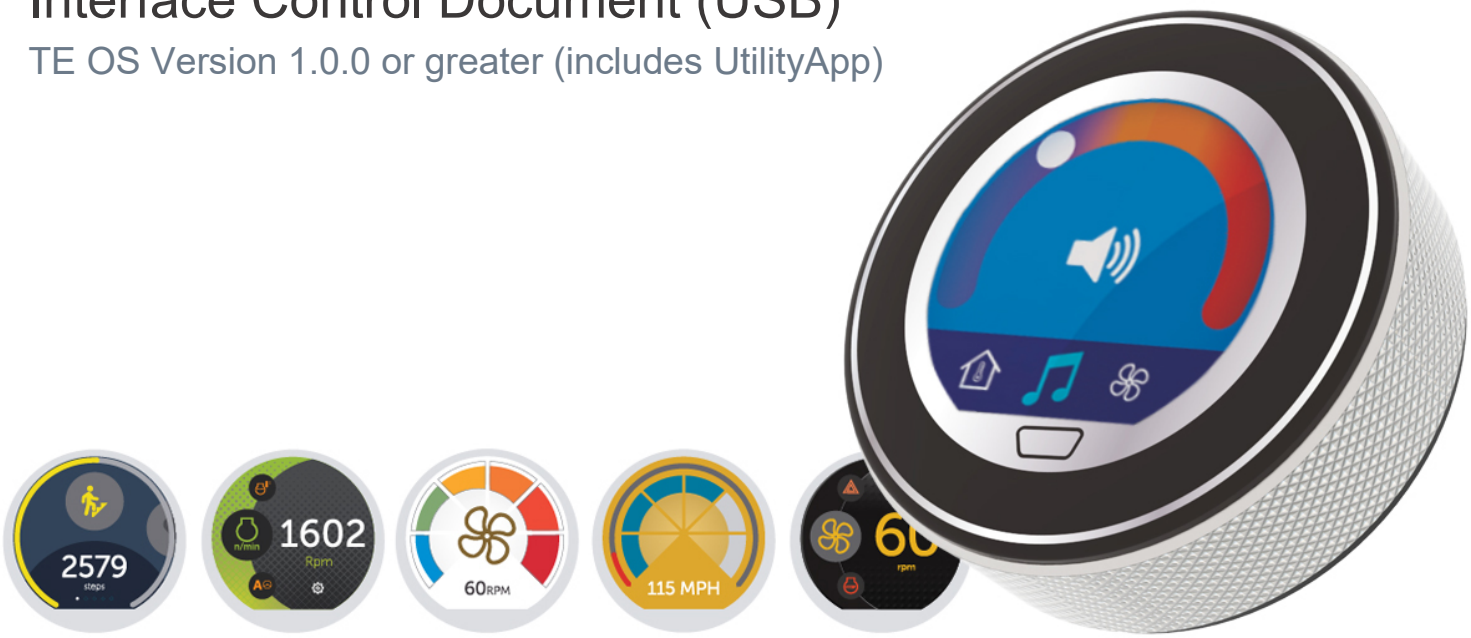
Grayhill

Intuitive Human Interface Solutions

Touch Encoder

Interface Control Document (USB)

TE OS Version 1.0.0 or greater (includes UtilityApp)



Revision History

Revision	Date	Description
A	3/27/2018	Original Release
B	7/12/2018	Changed pinout on M12 to reflect production pinout
C	8/24/2018	Added Programming Harness Information
D	10/25/2018	Updated Run/Programming Mode Schematics
E	12/7/2018	Updated Touch Encoder Image
F	4/23/2019	Added Command, Multi-value, and Display Code Information

Table of Contents

1. Overview

The USB interface of the Touch Encoder product is designed to conform to the USB 2.0 specification and, specifically, the USB HID and the USB Mass Storage Device protocols. Both protocols allows system integrators to retrieve the data from, or send data to, the device using the ubiquitous USB HID/Mass Storage Device support in the host OS. For more information about the USB standard or the USB HID/Mass Storage Device support in your specific host OS, please visit the USB web site (usb.org) or contact the OS vendor.

The Touch Encoder device connects to the host as a composite USB device with two Generic HID interfaces and one USB Mass Storage Device interface. The HID interfaces consist of several top-level collections (TLC) to virtually separate different device functions. Each report type generated by either the host (OUT) or the device (IN) will have a unique report ID to denote which interface and TLC it is associated with.

The Generic HID interfaces are used to send data to the host application. These interfaces also allow the host application to make changes to the configuration of the device, including screen/widget settings.

The Mass Storage Device interface is used to support the upload of firmware updates or configuration settings to the device.

In order to identify the Touch Encoder within the OS, use the following USB Vendor ID (VID), Product ID (PID) combination:

VID:	0x1658	Grayhill, Inc.
	0x0060	Touch Encoder, 2nd generation Touch Encoder

2. Generic HID Interfaces

The Touch Encoder device's Generic HID Interfaces are designed to interface with the generic HID support in the host OS. This means that the host OS does not consume the data itself, but that the data needs to be retrieved (IN report) or transmitted (OUT report) by an application or a driver running on the host device.

When new information is available from the device, a new report of the appropriate type is generated. Each report type can be generated by the device at a maximum rate of every 10 ms, although it is possible that the device generates mixed-type reports at a faster rate.

The different types of USB reports for this interface are explained below.

3. USB Reports

3.1 IN: Events Data Report (Interface #1, Collection #1, Report ID 1)

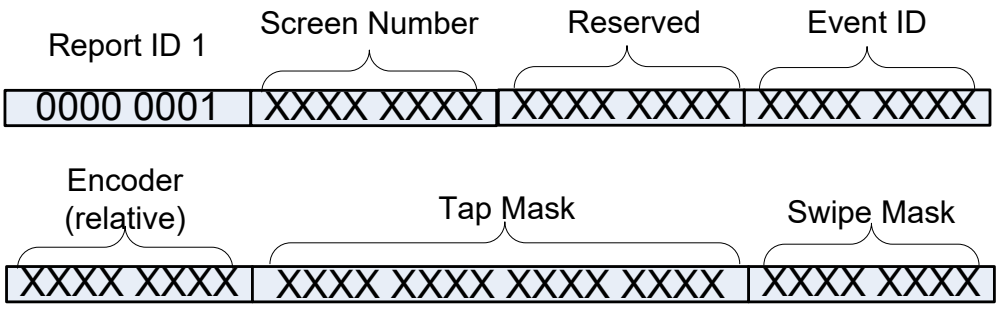


Figure 1 – Events Data Report

The Events Report is the lowest level report generated by the Touch Encoder device. Even though the Touch Encoder is designed to automatically respond to events occurring on the device, it will use this type of report to keep the host application informed about those events as well. In addition to the Events Data report, the device will also send a Widgets Data (IN) report to the host when an event causes a new widget/screen to be displayed on the device or causes one of the widget Values to change. This method of sending a widget report as well as the underlying event report is implemented in order to allow the host to closely monitor the activity of the device, and for the host to respond appropriately should a data mismatch occur.

The Events report is 8 bytes long and contains the report ID byte followed by the current event data. The Event data consists of a Screen Number byte, a reserved byte, an Event ID byte, an Encoder byte, a Tap Mask word and a Swipe byte.

The Encoder byte is a signed 8-bit value containing the relative change in encoder position since the last event report. It uses a special signing, like that used in the Touch Encoder CAN protocol, where the 0 value is 0x80. For example, if the encoder was turned two detents in the positive (CW) direction since the last report, the Encoder byte would be 0x82. Similarly, if the encoder was turned three detents in the negative (CCW) direction since the last report, the Encoder byte would be 0x7D (Grayhilll's notation for -3). In case the encoder was not turned but the Events report was transmitted because another event value changed, then the encoder byte would remain 0x80.

The Tap Mask word is a 16-bit value containing the tap information currently available. The most significant bit of the 16-bit value shows whether or not a tap occurred since the last Event report. The remaining 15 bits contain a bit mask of the tap zones the tap occurred in. The bit mask is defined as follows:

Bit0	Zone 0
Bit1	Zone 1
Bit2	Zone 2
Bit3	Zone 3
Bit4	Zone 4
Bit5	Zone 5
Bit6	Zone 6
Bit7	Zone 7
Bit8	Zone 8
Bit9	Zone 9
Bit10	Zone 10
Bit11	Zone 11
Bit12	Zone 12
Bit13	Zone 13
Bit14	Zone 14

The Swipe Mask byte is an 8-byte value containing the swipe information currently available. The most significant bit of this byte shows whether or not a swipe occurred since the last Event report. The least significant 4 bits contain a bit mask of the direction in which the swipe occurred. The bit mask is defined as follows:

Bit0	Up
Bit1	Down
Bit2	Left
Bit3	Right

3.2 OUT: Command Report (Interface #1, Collection #2, Report ID 2)

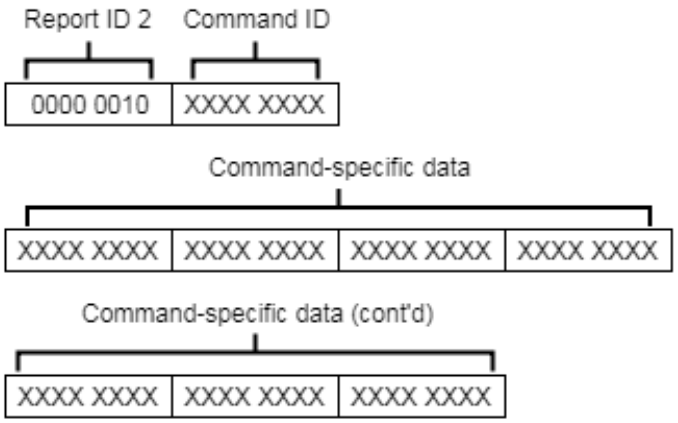


Figure 2 – Command Report

The Command Report is the lowest level control available to the USB host controller. It allows modification of system-level attributes, such as the display's backlight level.

The Command Report is 9 bytes long, with the first byte being the Report ID, the second byte being the Command ID, and the remaining 7 bytes being reserved for any Command-specific data.

The 8 bytes following the Report ID are meant to emulate the 8 byte Command messages in the Touch Encoder's CAN J1939 protocol, simplifying documentation and implementation for both the host device and the Touch Encoder.

3.2.1 Backlight Control

The backlight control command report uses a Command ID of 0x80 (128) and the second Command-specific data byte as the new percentage of backlight brightness.

The permitted range for the new percentage (data byte 2) is from 0x00 (0%) to 0x64 (100%).

As an example, the report to change the backlight to 100% would look like:

Interface #1, Collection #2, OUT: [02 xx 64 xx xx xx xx xx xx]

(where 'xx' bytes can be any value)

Similarly, the report to change the backlight to 30% would look like:

Interface #1, Collection #2, OUT: [02 xx 1E xx xx xx xx xx xx]

3.3 IN: Widget Data Report (Interface #2, Collection #1, Report ID 3)

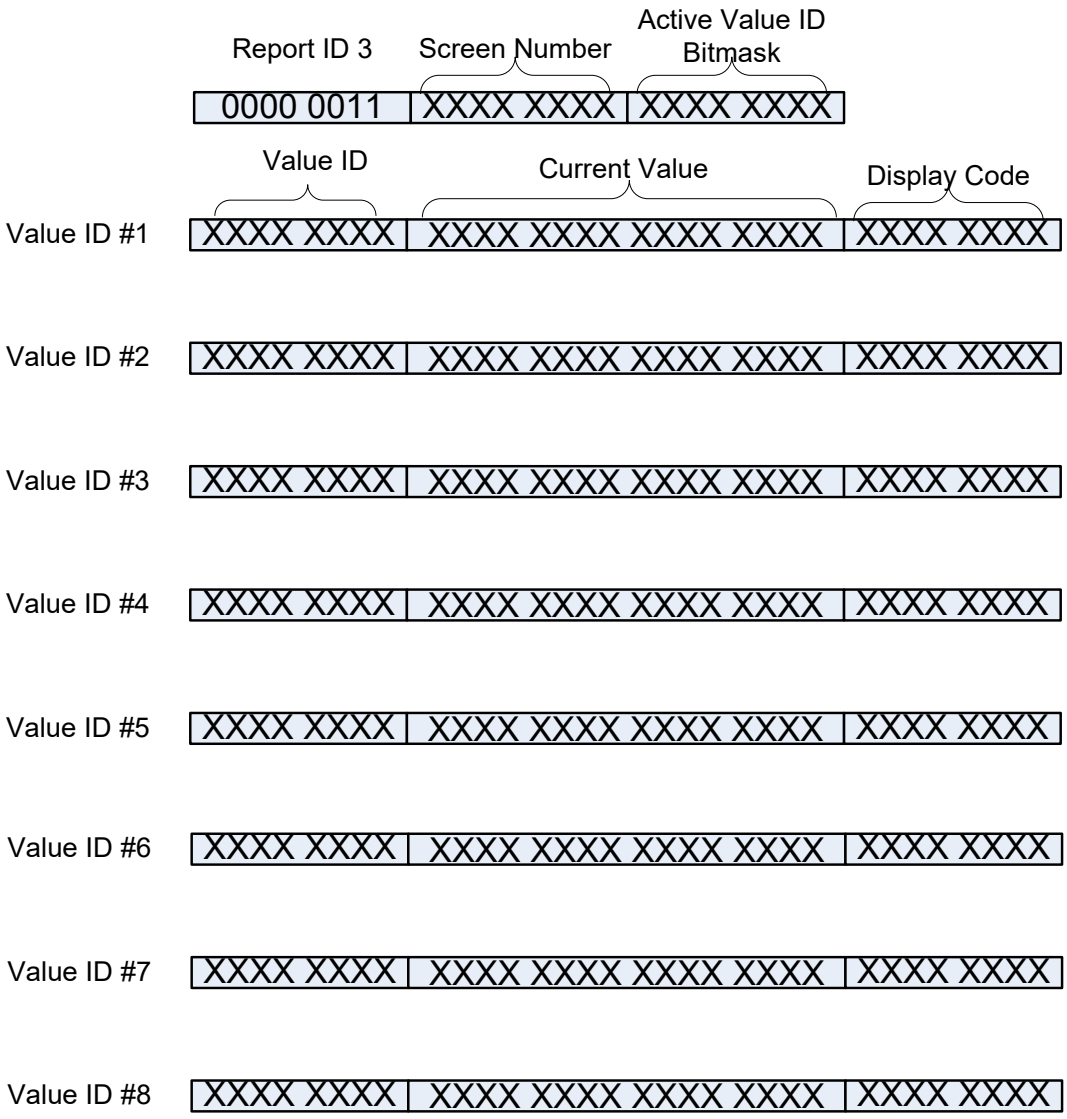


Figure 3 – Widget Data Report (IN)

This type of report is sent automatically by the device to the host whenever an event on the device causes a new screen to be displayed or causes one of the current screen's "Value ID" values to change. This type of report is used to keep the host informed about the widgets and "in-sync" with the device.

The Screen Number byte is an 8-bit value containing the screen number currently being displayed on the device.

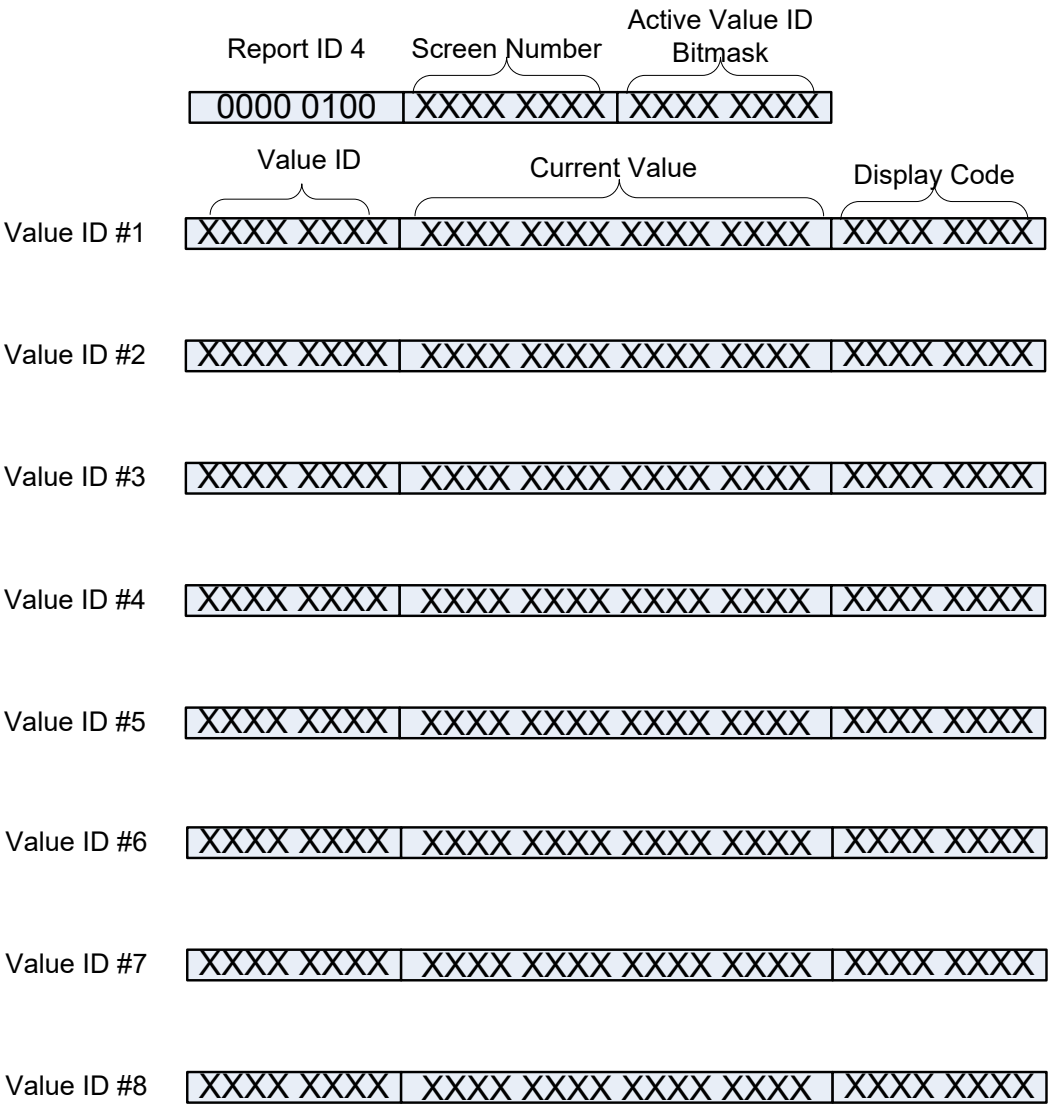
The Active Value ID Bitmask byte is a bitmask of the individual Value IDs that are currently active on the screen. There are 8 Value IDs available per screen ranging from 0x01 to 0x80. For each of the Value IDs listed in the Active Value ID Bitmask, one of the following Value ID Information Sections is populated.

The Value ID Information Section contains current value information for each of the up to eight active Value IDs on the current screen. Each field in this section contains a Value ID byte, a Current Value associated with that Value ID and a Display Code.

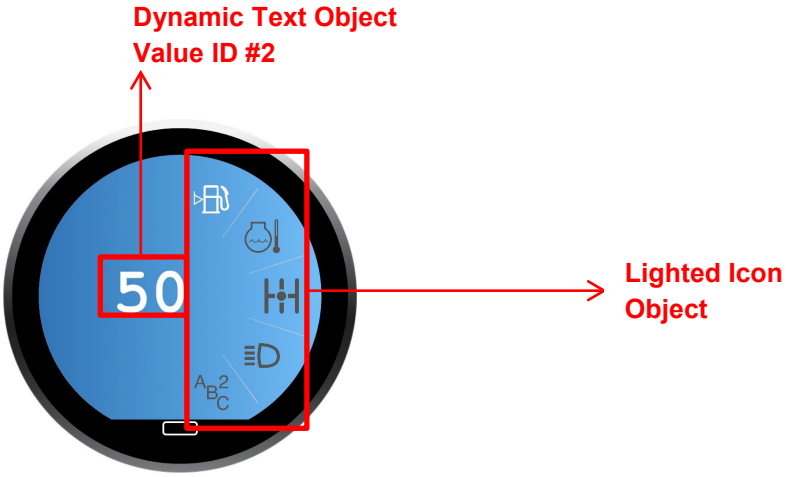
The Current Value field is constrained by the initial conditions, minimum and maximum values, and the step constraints defined during the design stage in the IDE. However, the host has the ability to overwrite or initialize the Current Value field.

The Display Code field contains a code specifying the format to apply to the Current Value before it is displayed on the widget.

3.4 OUT: Force Widget Data Report (Interface #2, Collection #2, Report ID 4



3.4.1 Multi-Value Data Example



Example: The figure above displays an example of a multi-value widget.

The dynamic text object is designated at Value ID #2. The lighted icon object is designated as Value ID #3.

Below is the sequence of messages to turn on the top lighted icon and change the dynamic text to 100 (note that the lighted icon object has an offset of 0x8000).

Interface #2, Collection #2, OUT: [04 03 06
xx xx xx xx
02 64 00 00
04 01 80 00
xx xx xx xx
xx xx xx xx
xx xx xx xx
xx xx xx xx]

(where 'xx' bytes can be any value)

3.4.2 Display Code and Decimal Code

The Display Code can allow the Touch Encoder to display decimal values within text boxes, while still using the 16-bit integer values for the respective Value ID's.

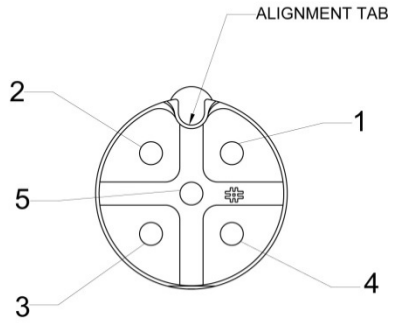
To do this, we use the top 4 bits of the Display Code as a signed 4-bit integer (which we call the Decimal Code). This integer is used as an exponent value with a base of 10, like in scientific notation.

As a quick note, this Decimal Code is only relevant for Value ID's which are used by text boxes and the remaining 4 bits are currently reserved for future uses.

Display Code Byte

Display Code	Input	Output
0x1X	Integer x 10	e.g. 10=100
0x2X	Integer x 100	e.g. 10=1000
0xEX	Integer ÷ 100	e.g. 10=0.10
0xFX	Integer ÷ 10	e.g. 10=1.0

3.5 Physical Layer



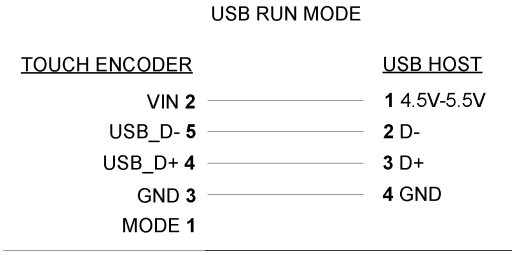
	USB	CAN
1	MODE	MODE
2	VIN	VIN
3	GND	GND
4	USB_D+	CAN_H
5	USB_D-	CAN_L

For Touch Encoder serial numbers less than A100000 please contact Grayhill for pinout detail.

If MODE Pin is floating at power up, the Touch Encoder will assume run mode operation. If Mode pin is connected to GND externally at startup, the Touch Encoder will assume programming mode. The Touch Encoder will download updates from a USB mass storage device (if connected) and update the Touch Encoder accordingly.

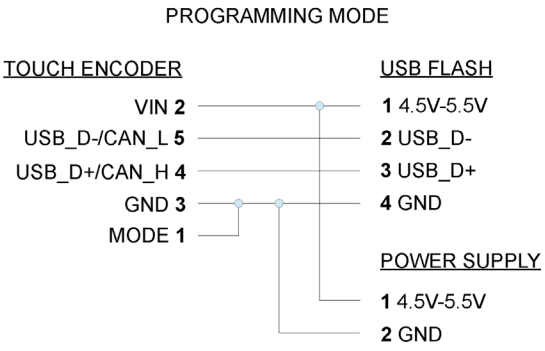
Run Mode:

See previous page:
Connector Pinouts



Programming Mode:

See previous page:
Connector Pinouts



Recommended Flash Drives:
Verbatim 49304
SanDisk SDCZ60-016G-B35

Note: Load files to FLASH drive and plug into socket. Touch Encoder only samples MODE pin during power up, so be sure this occurs after connecting harness and FLASH drive.

4. Mouse HID Interface (Coming Soon)

The Mouse HID interface of the Touch Encoder is designed to interface directly with the mouse driver of the host OS. The interface generates USB reports that contain relative motion, as well as left, and right mouse button click data. Since only one type of report is used in this interface, no Report ID is included in Mouse HID reports.

The firmware on the Touch Encoder is responsible for processing the hardware button data and the touch data reported by the touch controller, and converting this data into the mouse data format. This processing includes converting individual touch points (as reported by the touch controller) to relative mouse motion data, smoothing the motion data to reduce noise while keeping the processing latency as low as possible, and calculating the duration of individual touches to determine if a tap or other single-touch gesture occurred.

The supported single-touch gestures are as follows:

Tap – touch did not move significantly and was shorter than approximately 360 milliseconds in duration

Drag Enable – touch did not move significantly and was longer than 1 second in but shorter than 2 seconds in duration

Right-Click – touch did not move significantly and was longer than 2 seconds in duration

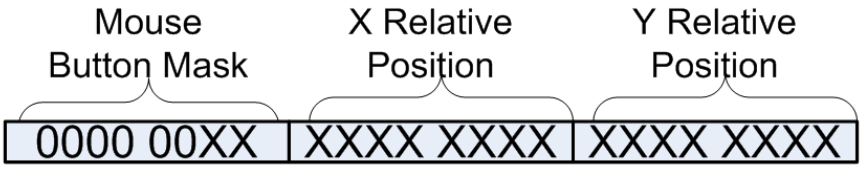
The “Tap” gesture is realized by sending a single report with the left-click button active, which also allows the use of a double-tap to generate a double-click.

If the “Drag Enable” gesture mentioned above is followed by (significant) relative motion of the touch on the touch pad surface, then the firmware will send persistent left-click reports for as long as the touch is active on the pad. This allows for items to be “pinned” by performing the gesture and then moving them across the screen. The dragging stops when the touch is lifted off the screen.

The “Right-Click” gesture mentioned above causes the device to send a single report with the right-click button active. This results in a right-click menu being opened. The user can then use simple touch motion to move the mouse cursor to the desired menu selection.

5. USB Reports

5.1 IN: Mouse Report (Interface #2)



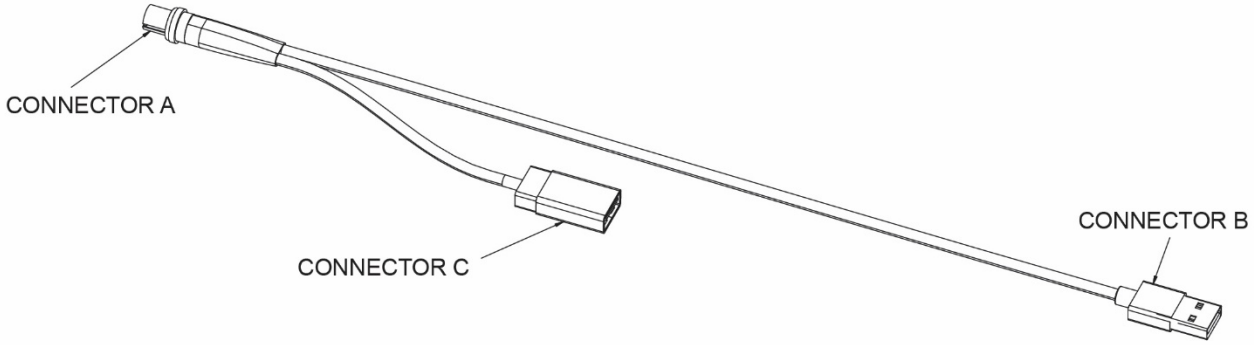
This type of report is 3 bytes long and contains 1 byte of button mask data, followed by 1 byte of relative X position data and 1 byte of relative Y position data. The button mask indicates whether the left (bit 0) or right (bit 1) buttons are either active (1) or inactive (0). The button bits reflect the output of the single-touch gesture recognition mentioned above.

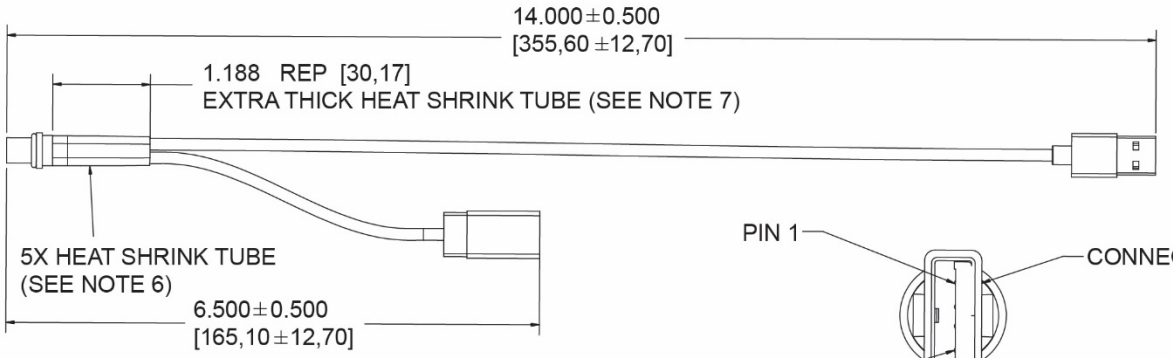
While touches are being sensed on the touch surface, this report is generated every 20 ms.

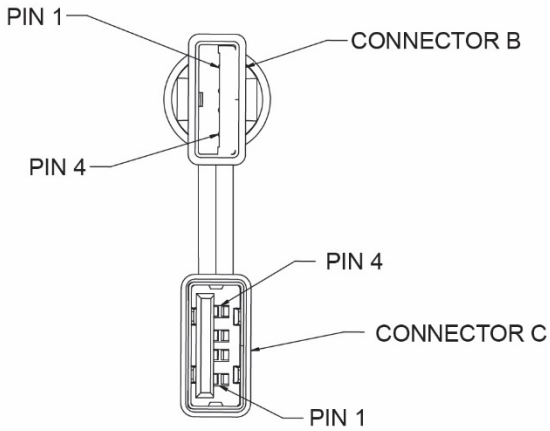
6. Appendix

6.1 Programming Harness

GRAYHILL P/N T18908 - USED FOR PROGRAMMING MODE ONLY



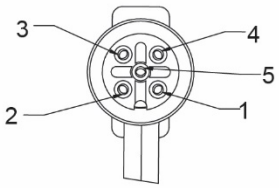




NOTES:

1. CONNECTOR A IS NORCOMP INC. P/N: 858-005-203RSU4 WITH METAL NUT REMOVED.
2. WIRE WITH USB CONNECTORS B & C IS TRIPP LITE P/N: U024-003 OR EQUIVALENT
3. CONNECTOR A PINS TO CONNECT TO CORRESPONDING CONNECTOR B & C TERMINALS ON THE CONNECTOR (SEE TABLE)..
4. WIRE LENGTH SHOULD BE MEASURED WITH WIRES STRAIGHT.
5. WIRE PULLOUT FORCE MUST BE 2 LBS. MINIMUM.
6. 5X HEAT SHRINK TUBING TO BE USED OVER CONNECTOR A CONNECTIONS.
7. EXTRA THICK HEAT SHRINK TUBING TO BE USED OVER CONNECTOR A CONNECTIONS.

WIRE #	CONNECTOR A PIN	CONNECTOR B PIN	CONNECTOR C PIN	COLOR	DESCRIPTION
1	1	N/A	4	BLACK	MODE
2	2	1	1	RED	Vin
3	3	4	4	BLACK	GND
4	4	N/A	3	GREEN	USB+
5	5	N/A	2	WHITE	USB-



Connector A Pinout

Grayhill
Intuitive Human Interface Solutions

561 Hillgrove Avenue
La Grange, IL 60525
web: www.grayhill.com
e-mail: te@grayhill.com
phone: +1 (708) 354-1040

About Grayhill

Grayhill, Inc. is a privately held firm which designs and manufacturers intuitive human interface solutions that make life simpler, safer and more efficient. Standard products include optical and Hall Effect encoders, discrete and Hall Effect joysticks, rotary switches, keypads, and pushbuttons; all with finely tuned haptics. Grayhill specializes in creating ergonomic panels and product shells that integrate various interface technologies, including displays, our components, and gesture recognizing multi-touch technology. With headquarters in La Grange, Illinois, and multiple state-of-the-art facilities around the world, Grayhill's team has the full engineering, product development and manufacturing expertise to deliver both standard and customized products quickly and cost-effectively. To learn more about Grayhill's products and capabilities, visit www.grayhill.com.