3M™ Multi-Touch
PX4nn ASIC Controller
Reference Guide

For USB: PX441, PX451, PX461, PX471, PX4611/PRX4401
For Dual-Mode: PX443, PX453, PX463, PX473

Read and understand all safety information contained in this document before using this product.
The information in this document is subject to change without notice. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of 3M Touch Systems, Inc. 3M may have patents or pending patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. The furnishing of this document does not give you license to these patents, trademarks, copyrights, or other intellectual property except as expressly provided in any written license agreement from 3M Touch Systems, Inc.

The information provided in this document is intended as a guide only. For the latest detailed engineering specifications, please contact your 3M Touch Systems, Inc. Application Engineer. 3M Touch Systems, Inc. is committed to continually improving product designs. As a result, product specifications may be subject to change without notification.

"RoHS 2011/65/EU" means that the product or part does not contain any of the substances in excess of the maximum concentration values ("MCVs") in EU RoHS Directive 2011/65/EU, unless the substance is in an application that is exempt under EU RoHS. The MCVs are by weight in homogeneous materials. This information represents 3M’s knowledge and belief, which may be based in whole or in part on information provided by third party suppliers to 3M.

NOTICE: Given the variety of factors that can affect the use and performance of a 3M Touch Systems, Inc. product (the “Product”), including that solid state equipment has operation characteristics different from electromechanical equipment, some of which factors are uniquely within User’s knowledge and control, it is essential that User evaluate the 3M Touch Systems, Inc. Product and software to determine whether it is suitable for User’s particular purpose and suitable for User’s method of application. 3M Touch Systems, Inc. statements, engineering/technical information, and recommendations are provided for User’s convenience, but their accuracy or completeness is not warranted. 3M Touch Systems, Inc. products and software are not specifically designed for use in medical devices as defined by United States federal law. 3M Touch Systems, Inc. products and software should not be used in such applications without 3M Touch Systems, Inc. express written consent. User should contact its sales representative if User’s opportunity involves a medical device application.

IMPORTANT NOTICE TO PURCHASER: Specifications are subject to change without notice. These 3M Touch Systems, Inc. Products and software are warranted to meet their published specifications from the date of shipment and for the period stated in the specification. 3M Touch Systems, Inc. makes no additional warranties, express or implied, including but not limited to any implied warranties of merchantability or fitness for a particular purpose. User is responsible for determining whether the 3M Touch Systems, Inc. Products and software are fit for User’s particular purpose and suitable for its method of production, including intellectual property liability for User's application. If the Product, software or software media is proven not to have met 3M Touch Systems, Inc. warranty, then 3M Touch Systems, Inc. sole obligation and User’s and Purchaser’s exclusive remedy, will be, at 3M Touch Systems, Inc. option, to repair or replace that Product quantity or software media or to refund its purchase price. 3M Touch Systems, Inc. has no obligation under 3M Touch Systems, Inc. warranty for any Product, software or software media that has been modified or damaged through misuse, accident, neglect, or subsequent manufacturing operations or assemblies by anyone other than 3M Touch Systems, Inc. 3M Touch Systems, Inc. shall not be liable in any action against it in any way related to the Products or software for any loss or damages, whether non-specified direct, indirect, special, incidental or consequential (including downtime, loss of profits or goodwill) regardless of the legal theory asserted.

© 3M 2013 All rights reserved.

Document Title: 3M™ Multi-Touch Controller PX4nn Reference Guide
Document Number: TSD-40847, Version C

3M, the 3M logo, MicroTouch, and the MicroTouch logo are either registered trademarks or trademarks of 3M in the United States and/or other countries.

Windows and/or other Microsoft products referenced herein are either registered trademarks or trademarks of Microsoft Corporation in the U.S. and/or other countries.

All other trademarks are the property of their respective owners.
Contents

Chapter 1
Introduction
What You Need to Know ................................................................. 5
Important Safety Information ........................................................... 6
Touch Sensor Care and Cleaning ....................................................... 7
3M Touch Systems Support Services ............................................... 8
Contact 3M Touch Systems ......................................................... 8

Chapter 2
Integrating the 3M™ Multi-Touch PX4nn Controller
Overview of the PX4nn Controllers .................................................. 9
Handling and ESD Protection ......................................................... 10
Establishing the Data Connection .................................................... 10
2-Pin Connector for Power – Dual Mode Self-Powered Only ................. 11
Sensor Connection ........................................................................ 12
Supplying Power to the USB and Dual-Mode Controller .................... 13
Mounting the Sensor ..................................................................... 13
Windows® 7 USB Compatibility ..................................................... 14
All Other Platforms ....................................................................... 14
Multi-Touch Application Support ..................................................... 14
Installing 3M™ MicroTouch™ Software .......................................... 14
Video Alignment .......................................................................... 15

Chapter 3
3M™ PX4nn USB Controller Communications
Overview of USB Firmware Communications ................................. 17
Communication Basics .................................................................... 17
Receiving Reports from the Controller ............................................ 18
Command Set ................................................................................ 18
Set Feature - Calibration ............................................................... 18
Get Feature - GetStatus ................................................................. 20
Chapter 4

3M™ PX4nn Serial Controller Communications

Overview of Firmware Commands ................................................................. 25
Controller Default Settings ........................................................................... 25
Communicating with the Controller ............................................................. 27
Firmware Commands .................................................................................... 28
Calibrate ......................................................................................................... 29
Guidelines for Calibrate Command .............................................................. 30
Mode Down Up .............................................................................................. 31
Mode Inactive ................................................................................................. 31
Mode Stream ................................................................................................. 31
Null Command ............................................................................................... 31
Output Identity ............................................................................................... 32
Restore Defaults ............................................................................................ 32
Soft Reset ....................................................................................................... 33
Unit Type ......................................................................................................... 33

Appendix A

PX4nn Controller Specifications

PX441 USB Controller .................................................................................... 37
PX443 Dual-Mode Controller ......................................................................... 38
PX451 USB Controller .................................................................................... 39
PX453 Dual-Mode Controller ......................................................................... 40
PX461 USB Controller .................................................................................... 41
PX463 Dual-Mode Controller ......................................................................... 42
PX471 USB Controller .................................................................................... 43
PX473 Dual-Mode Controller ......................................................................... 44
CHAPTER 1

Introduction

3M Touch Systems offers several multi-touch ASIC controllers designed for reliability and easy installation. Each controller provides superior performance and delivers excellent stability, sensitivity, accuracy, and fast response. These controllers are available in either USB or dual mode designed to work with sensors ranging from 7 inches to 32 inches. Controllers 33” to 47” are USB only.

This reference guide, designed for developers of touch systems, provides installation and configuration information for the 3M™ Multi-Touch PX4nn controller. This document includes information on integrating the PX4nn controller into your design, communicating with the controller, installing the MT 7 software user interface, and troubleshooting setup problems. It also includes a complete description of the firmware commands and controller specifications.

<table>
<thead>
<tr>
<th>Sensor Size</th>
<th>USB</th>
<th>Dual Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>7” to 15”</td>
<td>PX441</td>
<td>PX443</td>
</tr>
<tr>
<td>16” to 23”</td>
<td>PX451</td>
<td>PX453</td>
</tr>
<tr>
<td>24” to 26”</td>
<td>PX461</td>
<td>PX463</td>
</tr>
<tr>
<td>27” to 32”</td>
<td>PX471</td>
<td>PX473</td>
</tr>
<tr>
<td>33” to 47”</td>
<td>PX4611/PX4401</td>
<td>N/A</td>
</tr>
</tbody>
</table>

3M Touch Systems is committed to being a premier supplier in touch systems throughout the world. As a 3M Touch Systems customer, you are aware that we have strong internal programs that meet or exceed environmental regulations of our customers and the regions in which we conduct business.

What You Need to Know

This document assumes you are familiar with firmware commands and how to use them. Executing some commands may alter the performance of your touch product. You should be aware of the results of using these commands before executing them.
Important Safety Information

Read, understand and follow all safety information before using this product. Follow all instructions marked on the product and described in this document. Pay close attention to the following installation warnings and safety precautions.

### Intended Use

The PX4nn multi-touch controller was designed to enable projected capacitive touch in conjunction with other 3M™ Multi-Touch sensor products and was tested to replace an existing multi-touch controller. The PX4nn multi-touch controller is intended for internal mounting only and is not suitable for use in hazardous locations.

### Explanation of Signal Word Consequences

**DANGER:** Indicates a potentially hazardous situation, which, if not avoided, will result in death or serious injury and/or property damage.

**WARNING:** Indicates a potentially hazardous situation, which, if not avoided, could result in death or serious injury and/or property damage.

**CAUTION:** Indicates a potentially hazardous situation, which, if not avoided, may result in minor or moderate injury and/or property damage.

**CAUTION:** Indicates a potentially hazardous situation, which, if not avoided, may result in property damage.

---

**DANGER**

To reduce the risk of fire and/or explosion which will result in death or serious injury and/or property damage:
Do not install or use this product in a hazardous location.

---

**WARNING**

To avoid the risk of electric shock which could result in death or serious injury and/or property damage:

- You must be a qualified technician with experience in assembling and disassembling different types of displays. You must know the specifics of your display and have access to its documentation. There may be hazardous voltages present in the display. If you do not understand display electronics, you may injure yourself, damage the sensor, or damage the touch controller.
- Prior to working on the display, power off and unplug the system power cord and observe all Warnings and Cautions.
- Do not use a damaged power supply.
- Do not use a power cord that is frayed or otherwise damaged.
- Plug power cord into appropriate grounded power source.

To reduce the risk of fire and/or explosion which could result in death or serious injury and/or property damage:
Do not use this product in any outdoor environment unless NEMA standards (or similar standards such as IP rating) are followed.
WARNING

To avoid the risk of electric shock which could result in death or serious injury and/or property damage:
- Do not use a damaged power supply.
- Do not use a power cord that is frayed or otherwise damaged.

CAUTION

To reduce the risks associated with improper disposal, which if not avoided may result in minor or moderate injury and/or property damage from ground water contamination:
Dispose of components in accordance with federal, state and local regulations.

To reduce the risk of possible environmental contamination which may result in minor or moderate injury and/or property damage:
Dispose of the display in accordance with federal, state and local regulations.

To reduce the risk of the potentially hazardous situations associated with the use of isopropyl alcohol which may result in minor or moderate injury or property damage:
Follow all instructions and recommendations in the manufacturer's Material Safety Data Sheet and product label.

Touch Sensor Care and Cleaning

The sensor requires very little maintenance. 3M Touch Systems recommends that you periodically clean the glass sensor surface.

CAUTION

To avoid the potentially hazardous situations associated with the use of isopropyl alcohol which may result in minor or moderate injury or property damage:
Follow all instructions and recommendations in the manufacturer's Material Safety Data Sheet and product label.

NOTICE:

To avoid cosmetic damage to the touch sensor and scratches on the black border:
Do not clean the backside of the sensor; the surface can be easily scratched. If you need to remove any debris from the sensor, use a filtered de-ionized air source.

Typically, an isopropyl alcohol and water solution ratio of 50:50 is the best cleaning agent for your sensor. Do not use straight isopropyl alcohol. Be sure to follow solvent manufacturer's precautions and directions for use when using any solvents.
- It is important to avoid using any corrosive or caustic chemicals on the sensor.
- Always dampen the cloth and then clean the sensor. Be sure to spray the cleaning liquid onto the cloth, not the sensor, so that drips do not seep inside the display or stain the bezel.
- Apply the cleaner with a soft, lint-free cloth. Avoid using gritty cloths.
- Always handle the sensor with care. Do not pull on or stress flex tails.
• Use the recommended cleaner and a soft, lint free cloth to clean the sensor. Make sure the glass is clean and dry before you attach the sensor.
• Clean only the front surface of the sensor. Only remove the protective liner from the touch sensor just prior to placing it on the LCD.

Note: Do not clean the backside of the sensor; the surface can be easily scratched. If you need to remove any debris from the sensor, use a filtered de-ionized air source. If you do happen to scratch the black border, we recommend using a Snowman® brand black paint marker to touch up any small scratches.

3M Touch Systems Support Services

3M Touch Systems provides extensive support services through our website and technical support organization. Visit the 3M Touch Systems website at http://www.3M.com/touch, where you can download touch software and drivers, obtain regularly updated technical documentation on 3M Touch Systems products, and learn more about our company.

Whenever you contact Technical Support, please provide the following information:
• Touch display size, part number and serial number
• Current driver version
• Operating system used
• Information on additional peripherals

Technical Support is available Monday through Friday 8:30 a.m. to 5:30 p.m. with limited call back service after 5:30 p.m. until 8:00 p.m. US Eastern Standard Time – 9 a.m. to 5 p.m. throughout Europe.

You can contact 3M Touch Systems Technical Support (US only -- Eastern Standard Time) by calling the hot line, sending email or a fax.
• Technical Support Hot Line: 978-659-9200
• Technical Support Fax: 978-659-9461
• Toll Free: 1-866-407-6666 (Option 3)
• Email: US-TS-techsupport@mmm.com

Contact 3M Touch Systems

Contact information for all offices can be found on our website at:
http://www.3M.com/touch/
CHAPTER 2

Integrating the 3M™ Multi-Touch PX4nn Controller

The 3M™ Multi-Touch PX4nn controllers provide a functional equivalent touch controller with wide dynamic range, increased noise immunity, wide operating temperature stability, reprogrammability using software utilities and improved capability in ungrounded environments.

The firmware for the PX4nn controller is optimized for projected capacitive sensors integrated in the latest flat panel displays. The performance is controlled by firmware and can be customized for user applications.

This chapter covers the following PX4nn controller specifications:
- Cable connections
- Mounting requirements
- Power requirements and options

Overview of the PX4nn Controllers

Note: For complete specifications for the PX4nn controller, refer to the appendix at the end of this manual.

The PX4nn controller has a built-in Universal Serial Bus (USB) full speed interface. A full speed USB interface has a data rate of 12 Mb/s.

To integrate and test the PX4nn controller, you need the following items:
- A 3M™ Multi-Touch projected capacitive sensor
- A method of establishing the data communication between the controller and your system.
- A method of powering the controller:
- The USB controller will operate with the standard USB +5V bus power.
• The Dual-Mode controller requires separate 5V input and is self-powered. It can operate either as a USB or Serial RS-232 input controller. If both USB and serial are connected, the default is USB. Otherwise the controller will use whichever USB or serial input is connected.

• PX4611/PRX4401 require a separate 5V input which can supply at least 1A current.

• A standard Microsoft HID driver or a custom touch driver and a calibration program. You can use 3M™ MicroTouch MT 7 software which includes the necessary touch driver and utilities software.

The controller is optimized for Windows 7 certification.

Handling and ESD Protection

When mounting the sensor and controller, use normal precautions for handling electrostatic sensitive devices. The PX4nn controller has internal protection to ±4KV for ESD discharges to the controller or touch sensor surface that may occur during normal assembly operations. Refer to the appendix for further specifications for integrated ESD capability.

Establishing the Data Connection

The PX400 series controller can operate in either USB or serial mode – no need for different controllers. However, you cannot operate in both modes at the same time. If you try to connect both USB and serial cables, the controller will default to USB mode.

PX4611/PRX4401 operate in USB only

USB Connection

In USB mode, the controller uses a 3M Touch Systems USB communication cable (P/N 7319420) PC 99 compatible or equivalent interconnects. One end of this cable plugs into the USB connector (P1) on the PX400 series controller (P2 for PX4611 only). The other end has a Type-A connector, and plugs into a USB port on your PC.

When creating a custom cable, use the Molex 51004-0500 mating connector. The following table describes the interconnections of the 3M Touch Systems USB cable.

Table 1. USB Cable for PX400 Series Controllers

<table>
<thead>
<tr>
<th>PC Side (USB Type A)</th>
<th>Wire Color</th>
<th>Controller Side (5-Pin Molex)</th>
<th>Pin</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pin</td>
<td>USB Assigned</td>
<td></td>
<td>Color</td>
<td>Pin</td>
</tr>
<tr>
<td>1</td>
<td>+5Vdc (VBUS)</td>
<td>Red</td>
<td>1</td>
<td>+5Vdc VBUS power</td>
</tr>
<tr>
<td>2</td>
<td>Data (DN)</td>
<td>Gray</td>
<td>2</td>
<td>Data (DN) differential pair</td>
</tr>
<tr>
<td>3</td>
<td>Data (DP)</td>
<td>Green</td>
<td>3</td>
<td>Data (DP) differential pair</td>
</tr>
<tr>
<td>4</td>
<td>0V</td>
<td>Black</td>
<td>4</td>
<td>Power return</td>
</tr>
<tr>
<td>5</td>
<td>Cable Shield Shell</td>
<td>Charcoal</td>
<td>Gray</td>
<td>5</td>
</tr>
</tbody>
</table>
Serial Connection

For serial mode, the PX400 series controller uses a 3M Touch Systems RS-232 plug and play cable (P/N 7319630) or an equivalent interconnects.

One end of this cable plugs into the RS-232 connector (P3) on the PX400 series controller. The other end, which has a 9-pin D connector, can plug directly into a serial port on your PC extender card. The following table describes the interconnections for the 3M Touch Systems RS-232 cable.

When creating a custom cable, use the Molex 51004-0800 mating connector. The following table describes the interconnections of the 3M Touch Systems serial cable.

Table 2. Serial Cable for PX400 Series Controllers

<table>
<thead>
<tr>
<th>Pin</th>
<th>PC Side (9-Pin D)</th>
<th>Controller Side (8-Pin Molex)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RS-232 Assigned</td>
<td>Jumpered to:</td>
</tr>
<tr>
<td>1</td>
<td>Data Carrier Detect (DCD)</td>
<td>4 and 6 DTR and DSR</td>
</tr>
<tr>
<td>2</td>
<td>Receive Data (RXD)</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>Transmit Data (TXD)</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>Data Terminal Ready (DTR)</td>
<td>1 and 6 DCD and DSR</td>
</tr>
<tr>
<td>5</td>
<td>Signal Ground</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>Data Set Ready (DSR)</td>
<td>1 and 4 DCD and DSR</td>
</tr>
<tr>
<td>7</td>
<td>Request to Send (RTS)</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>Clear to Send (CTS)</td>
<td>4</td>
</tr>
<tr>
<td>9</td>
<td>Not Used</td>
<td>9</td>
</tr>
<tr>
<td>Sleeve</td>
<td>5V (not used)</td>
<td>6</td>
</tr>
<tr>
<td>Pin</td>
<td>Ground</td>
<td>7</td>
</tr>
</tbody>
</table>

2-Pin Connector for Power – Dual Mode Self-Powered Only

A 2-pin connector (pin 1 is 5V, pin 2 is ground) is used for 5V power input. Use the Molex 51004-0200 mating connector for 2.00 mm pitch or Molex 22-01-2025 mating connector for 0.100-inch pitch.

2-Pin Connector for Power – PX4611/PRX4401

A 2-pin connector (pin 1 is ground, pin 2 is ground) is used for 5V power input. Use the Molex 51004-0200 mating connector for 2.00 mm pitch or Molex 22-01-2025 mating connector for 0.100-inch pitch.
Sensor Connection

Each controller has a unique set of sensor ZIF mating connectors. The schematics define the number of pins and the BOM defines the vendor part number for each. The sensor flex tails should be plugged directly into the controller connectors. The integration guide defines the normal connections.

The sensors flex ZIF connector is always contact up.

**Note:** ZIF connectors can be fragile. Do not force these connectors open. These ZIF connectors are not removable; they must be opened/released to connect or disconnect a cable from them.
Mounting the Controller

The controller should be mounted internally and positioned to mate with the sensor flex tails without placing strain on the connections. Choose a convenient spot away from high-voltage and high power cables and noisy electronics. The mounting screws should be connected to the chassis ground. It is recommended that the AC power use a chassis ground connection for best operation.

Refer to the illustrations in Appendix A for details on controller sizes and mounting holes.

Supplying Power to the USB and Dual-Mode Controller

The USB controller is designed to use USB bus power (that is, tap power from the USB port).

The source must deliver the current defined in the controller specification in Appendix A, with a maximum ripple and noise of 50mV peak-to-peak. The current is either supplied by the Vbus in bus powered devices or the 5V input on self-powered devices. Typical currents are defined below:

<table>
<thead>
<tr>
<th>Interface</th>
<th>Max Current, 3Sigma, 5VDC Input</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dual Mode</td>
<td>402 mA 6 ASIC 366 mA 5 ASIC 314 mA 4 ASIC 238 mA 14 ASIC -----------</td>
</tr>
<tr>
<td>USB</td>
<td>384 mA 362 mA 310 mA 238 mA 780 mA</td>
</tr>
</tbody>
</table>

⚠️ CAUTION

To avoid possible damage to the controller, you must provide a path for electrostatic discharge. The controller-mounting holes should be used to connect to chassis safety ground and must be attached by the shortest possible route to a good earth return (chassis) in all applications.

Mounting the Sensor

There are several methods for mounting the sensor depending on your application. If you need instructions or recommendations from 3M Touch Systems on how to incorporate a sensor into your design, refer to the 3M™ Multi-Touch System PCT2000PX Integration Guide (TSD-39954). All 3M Touch Systems documentation is available from the corporate website at www.3Mtouch.com.
Windows® 7 USB Compatibility

3M multi-touch PCT technology works seamlessly with the Windows® 7 operating system. The Multi-Touch system supports USB HID for direct communication. The Multi-Touch PCT2000PX Series System leverages all the multi-touch functionality that is native to Windows® 7. Plug the display in to a system running Windows® 7 and you’ll quickly enter the world of true multi-touch functionality.

The Multi-Touch PCT2000PX Series System is compatible with Windows® 7 and is Windows® 7AQ certified for full multi-touch support. Refer to Microsoft's Windows® 7 documentation for additional information.

All Other Platforms

When using a Windows® 7 serial connection or for Windows® XP, Vista or Linux® operating systems (either USB or serial), 3M provides MicroTouch MT 7 Software for multi-touch drivers. This guide includes information installing the touch controller driver and setting the video alignment.

In the Appendix of this guide, you will find the communication protocols necessary to talk directly with the system electronics. This enables software developers using other operating systems such as Microsoft® Windows® Vista or Linux® to write their own drivers and optimize their applications.

Multi-Touch Application Support

Remember not all applications are multi-touch ready – Multi-touch behavior is a function of YOUR application. Check with your application vendor to determine if your software has multi-touch capability.

Installing 3M™ MicroTouch™ Software

Remember that Windows® 7 does not require any additional software to enable multi-touch functionality.

However, for Windows® XP, Vista or Linux® operating systems, 3M™ MicroTouch™ Software enables your Multi-Touch PCT2000PX Series System to work with your computer. 3M™ MicroTouch™ Software includes a control panel for setting your touch sensor preferences and a diagnostic utility. If you are experiencing problems with the touch sensor, you can use the diagnostic utilities provided to test the system.

For more information on installing this software and using the control panel, refer to the 3M™ MicroTouch™ Software User Guide (TSD-25695) on the corporate website at www.3m.com/touch.
Video Alignment

The Multi-Touch PCT2000PX Series System does not require video alignment if you are able to accurately touch icons on the sensor. If after integrating the system you cannot do this, the touch sensor’s active area may not be correctly aligned to the underlying video. To compensate for any variability in touch sensor placement during integration, you should perform a video alignment of the sensor to the display to ensure touch accuracy.

There are three ways to perform a video alignment. Regardless of the operating system, after you connect your touch display:

1. If you are using Windows®, you may calibrate by accessing the Software Diagnostic Utility (SDU) found on our website.

   Launch the SDU. Select the Tools menu and highlight Calibration. The screen below appears and you should touch the 2 targets as they appear. Press Escape to cancel Calibration.
2. If you are using 3M™ MicroTouch™ MT 7 Software, launch the MT 7 Control Panel and follow the instructions on the Main tab. You'll be asked to touch 3 targets.

3. If you are writing your own drivers, you should provide your own video alignment tool.

   Use the Paint program (Start → Programs → Accessories → Paint) to determine if you have multi-touch operation. Retest the accuracy after you perform a calibration.
CHAPTER 3

3M™ PX4nn USB Controller Communications

This chapter is intended for software developers only and discusses the fundamentals of communicating with the 3M™ PX4nn controller. The firmware commands, which are usually issued by a driver or utility program on the host system, control the operation of the controller. This chapter lists the recommended firmware commands and describes how to use each of these commands.

Overview of USB Firmware Communications

Developers may use this information when writing touch applications, developing custom drivers or touch configurations, or testing their touch systems. Developers can issue commands to initialize the controller, select operating modes, and execute diagnostic functions.

Note: This document assumes you are familiar with USB standards and modes of communication with USB devices, as well as firmware commands and how to use them. Executing some commands may alter the performance of your sensor and render it inoperable. You should be aware of the results before executing any firmware commands.

To optimize the performance of the PX4nn controller and simplify the development of custom drivers, 3M Touch Systems recommends you use the commands listed in this chapter for current development.

Communication Basics

This section provides information on sending firmware commands to the controller and interpreting the responses that the controller returns. The default operation of the PX4nn controller is USB Rev 2.0 full speed.

The USB command set is implemented by using vendor requests and vendor reports, i.e., vendor specific transactions. The controller issues some reports without prompting the computer. The computer can also send requests to the controller to change how it operates or receives information about the controller. The controller issues a synchronous report in response to some of these requests.

You need to know product ID and the vendor ID to write your own driver. These values are required for identifying the controller and can be found in your product specification.
Receiving Reports from the Controller

The controller sends a variety of reports to the computer. The first byte of each report is the Report ID that defines the structure and content of the report. The controller sends some reports as a direct response to a computer request (synchronous). The controller will also send some reports as the result of an external event, such as a touch (asynchronous).

Command Set

The USB command set is implemented by using HID Get Feature and Set Feature commands. The various requests and reports are grouped together by report size under a common feature ID. The following table summarizes the available HID class requests.

Table 1. HID Class Requests Summary

<table>
<thead>
<tr>
<th>HID Report Name</th>
<th>Command Name</th>
<th>bmRequestType</th>
<th>bRequest</th>
<th>Feature Report ID</th>
<th>Report Subtype</th>
<th>Data Stage Bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Get Feature</td>
<td>GetStatus</td>
<td>0xA1 (D2H)</td>
<td>0x01</td>
<td>0x06</td>
<td>--</td>
<td>8</td>
</tr>
<tr>
<td>Set Feature</td>
<td>Calibrate</td>
<td>0x21 (H2D)</td>
<td>0x09</td>
<td>0x03</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>Set Feature</td>
<td>Reset</td>
<td>0x21 (H2D)</td>
<td>0x09</td>
<td>0x03</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>Set Feature</td>
<td>Restore Defaults</td>
<td>0x21 (H2D)</td>
<td>0x09</td>
<td>0x03</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>Get Feature</td>
<td>GetMaxCount</td>
<td>0xA1 (D2H)</td>
<td>0x01</td>
<td>0x12</td>
<td>--</td>
<td>2</td>
</tr>
</tbody>
</table>

Set Feature - Calibration

This is a command to do a Calibrate Extended style calibration. The controller will auto-orient on this 2 point calibration.

Table 2. Calibration Setup Stage

<table>
<thead>
<tr>
<th>Offset</th>
<th>Field</th>
<th>Size</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>bmRequestType</td>
<td>1</td>
<td>0x21</td>
<td>Class,H2D,Interface</td>
</tr>
<tr>
<td>1</td>
<td>bRequest</td>
<td>1</td>
<td>0x09</td>
<td>Set Report</td>
</tr>
<tr>
<td>2</td>
<td>wValue</td>
<td>2</td>
<td>0x0303</td>
<td>Msb 03 = Feature Report ID</td>
</tr>
<tr>
<td>4</td>
<td>wIndex</td>
<td>2</td>
<td>0</td>
<td>Always 0</td>
</tr>
<tr>
<td>6</td>
<td>wLength</td>
<td>2</td>
<td>8</td>
<td>Always 8</td>
</tr>
</tbody>
</table>
Table 3. Data Stage

<table>
<thead>
<tr>
<th>Offset</th>
<th>Field</th>
<th>Size</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Report ID</td>
<td>1</td>
<td>0x03</td>
<td>Feature report ID</td>
</tr>
<tr>
<td>1</td>
<td>Report Subtype</td>
<td>1</td>
<td>0x04</td>
<td>Indicates a calibration request</td>
</tr>
<tr>
<td>2</td>
<td>bCalType</td>
<td>1</td>
<td>0xXX</td>
<td>0x01 = Extended cal (CX)</td>
</tr>
<tr>
<td>3--7</td>
<td>Not used</td>
<td>5</td>
<td>0</td>
<td>Not used</td>
</tr>
</tbody>
</table>

The device stalls endpoint 0 if the command cannot be processed successfully. The request cannot be processed if an invalid calibration type is given in the wValue field. The request will also fail if the 2 calibration points do not fall within certain bounds established by the firmware. These bounds require that the 2 calibration points be in opposite quadrants of the sensor.

The host should issue a GetStatus request to determine the status of this request. The status report includes a command status byte which will be set as shown below.

Table 4. Calibration Response

<table>
<thead>
<tr>
<th>Command Status Byte</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Calibration Failed</td>
</tr>
<tr>
<td>1</td>
<td>Controller is waiting for a touch in the lower left corner. Calibration software paints a target in the lower left corner.</td>
</tr>
<tr>
<td>2</td>
<td>Controller is waiting for a touch in the upper right corner. Calibration software paints a target in the upper right corner.</td>
</tr>
<tr>
<td>3</td>
<td>Calibration completed successfully.</td>
</tr>
</tbody>
</table>

The controller does not timeout waiting for touch. Use the Soft Reset command to abort the calibration.

Determining Target Areas

The default calibration targets (points) are located 12.5% (1/8) inward from the corners of the video image. For example, suppose the resolution of your Windows-based display is 1680 x 1050. The Calibrate Extended command calculates the amount to move inward as follows:

- Amount to move inward in the X direction: 1680 x 1/8 = 210
- Amount to move inward in the Y direction: 1050 x 1/8 = 131

The Calibrate Extended command then positions the first calibration target inward from the lower left corner (0,1049) and the second calibration target inward from the upper right corner (1679,0). The following illustration shows how the calibration targets are calculated for a Windows-based system. Your operating system may be different.

The illustration below shows the coordinates of the calibration targets and display corners. The corners show the video coordinates in parentheses and the touch screen coordinates in brackets.
Note: Other screen resolutions will scale proportionally. The touch coordinates will not change.

Get Feature - GetStatus

This is a request to send information that indicates the status of the controller. Among the uses for this request are determining whether there were any power on check errors and determining whether the last request was completed successfully.

Table 5. Controller Status Setup Stage

<table>
<thead>
<tr>
<th>Offset</th>
<th>Field</th>
<th>Size</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>bmRequestType</td>
<td>1</td>
<td>0xA1</td>
<td>Class,D2H,Interface</td>
</tr>
<tr>
<td>1</td>
<td>bRequest</td>
<td>1</td>
<td>0x01</td>
<td>Get Report</td>
</tr>
<tr>
<td>2</td>
<td>wValue</td>
<td>2</td>
<td>0x0306</td>
<td>msb=03=Feature</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>lsb=06=Feature Report ID</td>
</tr>
<tr>
<td>4</td>
<td>wIndex</td>
<td>2</td>
<td>0</td>
<td>Always 0</td>
</tr>
<tr>
<td>6</td>
<td>wLength</td>
<td>2</td>
<td>8</td>
<td>Always 8</td>
</tr>
</tbody>
</table>

Table 6. Controller Status Data Stage (controller response)

<table>
<thead>
<tr>
<th>Offset</th>
<th>Field</th>
<th>Size</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Report ID</td>
<td>1</td>
<td>0x06</td>
<td>Feature Report ID</td>
</tr>
<tr>
<td>1</td>
<td>POC Status</td>
<td>1</td>
<td>0xx</td>
<td>Power On Check Status</td>
</tr>
<tr>
<td>2</td>
<td>Cmd Status</td>
<td>1</td>
<td>0xx</td>
<td>Status of last command</td>
</tr>
<tr>
<td>3</td>
<td>Touch Status</td>
<td>1</td>
<td>0</td>
<td>Not used</td>
</tr>
<tr>
<td>4</td>
<td>Not Used</td>
<td>1</td>
<td>0xx</td>
<td>0x00 = Async touch output off</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0x01 = Async touch output on</td>
</tr>
<tr>
<td>5-7</td>
<td>Not used</td>
<td>3</td>
<td>0</td>
<td>Not used</td>
</tr>
</tbody>
</table>

POC Status – The status of the Power-on Checks. Various controller systems are checked at power-up. If any failures in these systems are detected, a POC flag is set. The POC status field reports the state of these flags.
Table 7. Power On Check Bit Fields

<table>
<thead>
<tr>
<th>Bit Number</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Not used</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>ROM_ERROR</td>
<td>Code area checksum error</td>
</tr>
<tr>
<td>2</td>
<td>PWM Error</td>
<td>Touch screen not connected or potential problem.</td>
</tr>
<tr>
<td>3</td>
<td>NOV_ERROR</td>
<td>Parameter Block1 checksum error</td>
</tr>
<tr>
<td>4</td>
<td>HDW_ERROR</td>
<td>Problem with ADCs</td>
</tr>
<tr>
<td>5</td>
<td>Not used</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Not used</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Not used</td>
<td></td>
</tr>
</tbody>
</table>

Last Command Request Status – This field is used to determine whether the last request was processed successfully. It is also used to track the progress of a multi-stage request, such as 2 point calibration. The Status Request does not affect the contents of this field, i.e., successful or unsuccessful processing of a previous status request does not cause the command status field to be updated.

Table 8. Valid Command Status Field Entries

<table>
<thead>
<tr>
<th>Response</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Failure in command processing</td>
</tr>
<tr>
<td>1</td>
<td>Command being processed</td>
</tr>
<tr>
<td>2</td>
<td>Stage 1 processing complete (for multi-stage commands)</td>
</tr>
<tr>
<td>3</td>
<td>Command complete</td>
</tr>
<tr>
<td>4</td>
<td>Soft Reset Occurred</td>
</tr>
<tr>
<td>5</td>
<td>Hard Reset Occurred</td>
</tr>
<tr>
<td>6 -- 7</td>
<td>Not used</td>
</tr>
</tbody>
</table>

Get Feature - GetMaxCount

This is a request to send information that indicates the maximum number of simultaneous touches supported by the controller.

Table 9. Setup Stage

<table>
<thead>
<tr>
<th>Offset</th>
<th>Field</th>
<th>Size</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>bmRequestType</td>
<td>1</td>
<td>0xA1</td>
<td>Class,D2H,Interface</td>
</tr>
<tr>
<td>1</td>
<td>bRequest</td>
<td>1</td>
<td>0x01</td>
<td>Get Report</td>
</tr>
<tr>
<td>2</td>
<td>wValue</td>
<td>2</td>
<td>0x0312</td>
<td>msb=03=Feature</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>lsb=12 Feature Report ID</td>
</tr>
<tr>
<td>4</td>
<td>wIndex</td>
<td>2</td>
<td>0</td>
<td>Always 0</td>
</tr>
<tr>
<td>6</td>
<td>wLength</td>
<td>2</td>
<td>2</td>
<td>Always 2</td>
</tr>
</tbody>
</table>
Table 10. Data Stage (controller response)

<table>
<thead>
<tr>
<th>Offset</th>
<th>Field</th>
<th>Size</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Report ID</td>
<td>1</td>
<td>0x12</td>
<td>Feature Report ID</td>
</tr>
<tr>
<td>1</td>
<td>Max Count</td>
<td>1</td>
<td>0x1NN</td>
<td>Number of actual fingers supported</td>
</tr>
</tbody>
</table>

Note: The number of actual contacts reported may exceed this number.

Set Feature - Reset

This is a request to perform a controller reset. Soft resets are automatic after any block parameter changes.

Table 11. Reset - Setup Stage

<table>
<thead>
<tr>
<th>Offset</th>
<th>Field</th>
<th>Size</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>bmRequestType</td>
<td>1</td>
<td>0x21</td>
<td>Class,H2D,Interface</td>
</tr>
<tr>
<td>1</td>
<td>bRequest</td>
<td>1</td>
<td>0x09</td>
<td>Set Report</td>
</tr>
<tr>
<td>2</td>
<td>wValue</td>
<td>2</td>
<td>0x0303</td>
<td>msb=03=Feature</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>lsb=03=Feature Report ID</td>
</tr>
<tr>
<td>4</td>
<td>wIndex</td>
<td>2</td>
<td>0</td>
<td>Always 0</td>
</tr>
<tr>
<td>6</td>
<td>wLength</td>
<td>2</td>
<td>8</td>
<td>Always 8</td>
</tr>
</tbody>
</table>

Table 12. Reset – Data Stage

<table>
<thead>
<tr>
<th>Offset</th>
<th>Field</th>
<th>Size</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Report ID</td>
<td>1</td>
<td>0x03</td>
<td>Feature Report ID</td>
</tr>
<tr>
<td>1</td>
<td>Report Subtype</td>
<td>1</td>
<td>0x07</td>
<td>Indicates a reset request</td>
</tr>
<tr>
<td>2</td>
<td>bResetType</td>
<td>1</td>
<td>0x01</td>
<td>Soft Reset</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0x02</td>
<td>Hard Reset</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0x03</td>
<td>Reboot ROM</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0x05</td>
<td>Reboot EEPROM</td>
</tr>
<tr>
<td>3--7</td>
<td></td>
<td>5</td>
<td>0</td>
<td>Not used</td>
</tr>
</tbody>
</table>

A Hard Reset will cause the controller to re-enumerate.

In the case of a Soft Reset, after sending the command, the controller will acknowledge (ACK) the transfer, but the command will not yet be completed. Before sending any other commands, the host should poll with GetStatus until the command status field returns “Soft Reset Occurred,” “Command Complete,” or “Fail”. Any timeout for this status polling should be 2 seconds minimum.
Set Feature – Restore Defaults

This is a request to restore parameter defaults.
If you did not connect the sensors tails correctly, you may not get the full advantage of your touch screen size. Typically, on initial power-up, the controller will automatically detect the correct size of the touch sensor. It will not support the touch sensor until you reboot the controller.

Typically, the controller will enumerate on start-up with the correct size of the touch sensor. If this does not happen, the HID descriptor is filled with zeroes. You can issue a Restore Defaults command, reboot your controller and perform a 2-point calibration to determine the correct size of the touch sensor. If you reboot again, after the 2-point calibration, the controller enumerates and the HID descriptor will contain accurate calculated dimensions.

Table 13. Restore Defaults -- Setup Stage

<table>
<thead>
<tr>
<th>Offset</th>
<th>Field</th>
<th>Size</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>bmRequestType</td>
<td>1</td>
<td>0x21</td>
<td>Class,H2D,Interface</td>
</tr>
<tr>
<td>1</td>
<td>bRequest</td>
<td>1</td>
<td>0x09</td>
<td>Set Report</td>
</tr>
<tr>
<td>2</td>
<td>wValue</td>
<td>2</td>
<td>0x0303</td>
<td>msb=03=Feature lsb=03= Feature Report ID</td>
</tr>
<tr>
<td>4</td>
<td>wIndex</td>
<td>2</td>
<td>0</td>
<td>Always 0</td>
</tr>
<tr>
<td>6</td>
<td>wLength</td>
<td>2</td>
<td>8</td>
<td>Always 8</td>
</tr>
</tbody>
</table>

Table 14. Restore Defaults - Data Stage

<table>
<thead>
<tr>
<th>Offset</th>
<th>Field</th>
<th>Size</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Report ID</td>
<td>1</td>
<td>0x03</td>
<td>Feature Report ID</td>
</tr>
<tr>
<td>1</td>
<td>Report Subtype</td>
<td>1</td>
<td>0x08</td>
<td>Restore Defaults</td>
</tr>
<tr>
<td>2-7</td>
<td></td>
<td>6</td>
<td>0</td>
<td>Not used</td>
</tr>
</tbody>
</table>

After sending this command, the controller will acknowledge (ACK) the transfer, but the command will not yet be completed. Before sending any other commands, the host should poll with GetStatus until the command status field returns “Command Complete” or “Fail”. Any timeout for this status polling should be 2 seconds minimum.

Asynchronous Reports

Coordinate Data – MultiTouch Digitizer mode Async Report 0x13

This is used to transfer the coordinate data to the host. This report, when activated, is sent to the host whenever new data is available or scheduled for transmission. It is an asynchronous report that is activated by default at power up.
The report shown supports up to 6 simultaneous touches. Note that if more than 6 touches were to be supported, Report 0x13 would be sent as many times as necessary to accommodate the number of fingers touching. Only the first Report 0x13 will have a non-zero actual count. Each valid touch is marked with an ID number that remains the same from touchdown through liftoff. The ID number can be any value from 0 to 255. Ignore all other data within a touch report structure with a status marked “not valid”.

The coordinate system is Upper Left origin.

**Table 15. Coordinate Data Report 0x13 --Data Stage**

<table>
<thead>
<tr>
<th>Offset</th>
<th>Field</th>
<th>Size</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Report ID</td>
<td>1</td>
<td>0x13</td>
<td>Report ID</td>
</tr>
<tr>
<td>1</td>
<td>Touch Report</td>
<td>10</td>
<td>See table 16</td>
<td>Touch Report Structure</td>
</tr>
<tr>
<td>11</td>
<td>Touch Report</td>
<td>10</td>
<td>See table 16</td>
<td>Touch Report Structure</td>
</tr>
<tr>
<td>21</td>
<td>Touch Report</td>
<td>10</td>
<td>See table 16</td>
<td>Touch Report Structure</td>
</tr>
<tr>
<td>31</td>
<td>Touch Report</td>
<td>10</td>
<td>See table 16</td>
<td>Touch Report Structure</td>
</tr>
<tr>
<td>41</td>
<td>Touch Report</td>
<td>10</td>
<td>See table 16</td>
<td>Touch Report Structure</td>
</tr>
<tr>
<td>51</td>
<td>Touch Report</td>
<td>10</td>
<td>See table 16</td>
<td>Touch Report Structure</td>
</tr>
<tr>
<td>61</td>
<td>Actual Count</td>
<td>1</td>
<td>1 to max</td>
<td>Number of valid touch reports (no more than declared max touch reports)</td>
</tr>
<tr>
<td>62</td>
<td>Not used</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>63</td>
<td>Not used</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

**Table 16. Touch Report Structure**

<table>
<thead>
<tr>
<th>Offset</th>
<th>Field</th>
<th>Size</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Status</td>
<td>6</td>
<td>0xXX</td>
<td>0x00 Report not valid 0x04 Not touching 0x07 Touching</td>
</tr>
<tr>
<td>1</td>
<td>Touch ID</td>
<td>1</td>
<td>0-255</td>
<td>Not used</td>
</tr>
<tr>
<td>2</td>
<td>X lsb</td>
<td>1</td>
<td>0xXX</td>
<td>X (0-7FFF)</td>
</tr>
<tr>
<td>3</td>
<td>X msb</td>
<td>1</td>
<td>0xXX</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Y lsb</td>
<td>1</td>
<td>0xXX</td>
<td>Y (0-7FFF)</td>
</tr>
<tr>
<td>5</td>
<td>Y msb</td>
<td>1</td>
<td>0xXX</td>
<td></td>
</tr>
</tbody>
</table>
CHAPTER 4

3M™ PX4nn Serial Controller Communications

This chapter is intended for software developers only and discusses the fundamentals of communicating with the 3M™ PX4nn serial controller. The firmware commands, which are usually issued by a driver or utility program on the host system, control the operation of the controller; however developers can enter these commands directly. This chapter:

- Describes the controller default settings.
- Lists the recommended firmware commands for current development.
- Describes how to use each of these commands.
- References additional commands developers may need to use.

The description of each command includes the command syntax, the default value, how the command works, and the expected response from the controller.

Overview of Firmware Commands

Developers may use these commands when writing touch applications, developing custom drivers or touch configurations, or testing their touch systems. Developers can issue commands to initialize the controller, select operating modes, and execute diagnostic functions.

**Note:** This document assumes you are familiar with standards and modes of communication with serial devices, as well as firmware commands and how to use them. Executing some commands may alter the performance of your sensor and render it inoperable. You should be aware of the results before executing any firmware commands.

To optimize the performance of the 3M™ PX4nn controller and simplify the development of custom drivers, 3M Touch Systems recommends you use the commands listed in this chapter for current development.

Controller Default Settings

**Communication Parameters**

The operation of the PX4nn serial controllers is N81 (no parity, 8 data bits, and 1 stop bits) at 115,200 baud (nonadjustable).
Data Format

Data format refers to the type of packet the controller uses to send the X/Y touch coordinates to the host system. Format Tablet Multi-touch is the default format for the 3M™ PX4nn serial controller. In Format Tablet Multi-touch, the controller sends 6 bytes per point and provides the most rapid response time to a touch.

Note that 3M MicroTouch Software automatically sets communications parameters to N81.

The format is a six byte packet that includes one status byte, four bytes of binary X, Y position data and a touch ID. The X, Y coordinates are 14 bits, providing a range of 0 to 16,383. The data is sent in the following format:

Table 17. Touch Data Format Settings

<table>
<thead>
<tr>
<th>Data Sequence</th>
<th>Bits</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>7</td>
</tr>
<tr>
<td>Status - Byte 1</td>
<td>1</td>
</tr>
<tr>
<td>X - Byte 2</td>
<td>0</td>
</tr>
<tr>
<td>X - Byte 3</td>
<td>0</td>
</tr>
<tr>
<td>Y - Byte 4</td>
<td>0</td>
</tr>
<tr>
<td>Y - Byte 5</td>
<td>0</td>
</tr>
<tr>
<td>ID – Byte 6</td>
<td>0</td>
</tr>
</tbody>
</table>

Note 1: P is the proximity bit. It is set to 1 when there is a touch, 0 for a liftoff.

The ID is associated with a touch stream from one finger. The ID, in the range of 0-127, is assigned on a rotating basis at touchdown. A finger touch stream consists of a touchdown, maybe some continuing touch points, ending with a liftoff. The ID is released on liftoff, and becomes available to be reassigned.

When multiple fingers are down, the packets from the touch streams are sent interleaved. There is no order imposed on the interleave.

Operating Mode

The operating mode specifies the conditions under which the controller sends the X/Y touch coordinates (input data packet) to the host system.

Mode Stream is the default operating mode for the 3M™ PX4nn serial controller. In Mode Stream, the controller sends a continuous stream of data packets when the sensor is touched. The controller sends the data as long as a touch continues on the sensor.

Because Mode Stream sends touch data continually, it is the most versatile mode, and it provides the best response time and overall feel.

3M Touch Systems recommends that the touch driver generate an interrupt as each packet in the data stream arrives. Because touchdown and liftoff events are specially coded, your software can generate mouse events that correspond to what the user is doing. This enables instant feedback and prevents data loss.
Communicating with the Controller

This section provides information on sending firmware commands to the controller and interpreting the responses that the controller returns.

The commands listed in Table 18 are those that 3M Touch Systems currently uses for development. 3M Touch Systems recommends that you use only these commands for 3M™ PX4nn serial controller development.

Commands to the controller are sent on the signal Receive Data (RXD) line as a serial data stream. For each command it receives, the controller sends a response to the host on the signal Transmit Data (TXD) line also as a serial data stream.

Sending Commands to the Controller

When you send a command to the controller, you must use the correct command format. The general format of a command is as follows:

<Header>Command<Terminator>

Note: The following descriptions of header, command, and terminator, use 3M Touch Systems terminal emulator key sequences. You may need to enter the sequence in a different format, depending on your emulator.

The header is the first character in the command string and is the ASCII start-of-header control character SOH. The hexadecimal code for the ASCII SOH control character is 01. To start the command sequence, use the key combination Ctrl A (^A). If you are working with an IBM PC compatible system, the Ctrl A key combination immediately returns an ASCII ☺ character.

The command, which always follows the header, consists of ASCII uppercase letters and numbers only (printable characters).

The terminator is the last character of each command string and is an ASCII carriage return CR. An ASCII CR control character is 0D hexadecimal. To enter a carriage return, ending the command sequence, use Enter or the key combination Ctrl M (^M).

This chapter lists each command as a string of ASCII control characters and printable characters consisting of a header, the command, and a terminator as follows:

<SOH>Command<CR>

Receiving Responses from the Controller

After executing a command, the controller returns a response or acknowledgment to the host system. Each controller response consists of a header, the command response, and a terminator in the following format:

<Header>Command Response<Terminator>

Note: The following descriptions of header, response, and terminator, use 3M Touch Systems terminal emulator key sequences. The format of controller responses varies depending on the terminal emulation mode you are using.
The header is the first character in the response string and is the ASCII start-of-header control character SOH. The hexadecimal code for the ASCII SOH control character is 01. If you are working with an IBM PC compatible system in terminal mode, the SOH control character returns a ☺ character to the sensor.

The command response, which always follows the header, is a range of ASCII characters depending on the type of command sent. Responses can be in many forms. For example, one standard response is 0 (ASCII character ‘zero’ or 30 hexadecimal). This response indicates a successful command completion for most commands, while it indicates a failed completion for other commands. Refer to the firmware command section for a description of what the response indicates for each particular command.

Another standard response is 1 (ASCII character ‘one’ or 31 hexadecimal). In most cases, this response indicates the command failed. The controller received an invalid command that it could not execute.

Some possible reasons for a command failure include:
- The command was not formatted correctly.
- The system parameters were not set up to allow command execution.
- The controller does not support the command.

The terminator is the last character of each response string and is an ASCII carriage return CR. The hexadecimal code for the ASCII CR control character is 0D hexadecimal. The value returned in the response will be the ASCII control character for a carriage return, displayed on the screen as the cursor moving to the next line.

In this chapter, responses are shown as a string of ASCII characters consisting of a header, the response, and a terminator as follows:

<SOH>Response<CR>

Controller Initialization

To initialize the 3M™ PX4nn serial controller for new development, 3M Touch Systems recommends that the host system issue a Reset command whenever the host system is powered on and is attempting to establish communication with the controller.

Firmware Commands

Developers may use this information when writing touch applications, developing custom drivers or touch configurations, or testing their touch systems. Developers can use firmware commands to initialize the controller, select operating modes, specify data formats, and execute diagnostic functions.

Caution: This document assumes you are familiar with firmware commands and how to use them. Executing some commands may alter the performance of your touch system and render it inoperable. You should be aware of the results before executing any firmware commands.

To optimize the performance of the touch controller and simplify the development of custom drivers, 3M Touch Systems recommends you use the commands listed in Table 18 for current development. Using these commands ensures compatibility with all 3M™ controllers.
Table 18. Firmware Commands for PX4nn Serial Controller Development

<table>
<thead>
<tr>
<th>Command</th>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Calibrate</td>
<td>CX</td>
<td>Initiates an interactive 2-point calibration.</td>
</tr>
<tr>
<td>Mode Down Up</td>
<td>MDU</td>
<td>Reports touchdowns and liftoffs only.</td>
</tr>
<tr>
<td>Mode Inactive</td>
<td>MI</td>
<td>Turns off touch reports.</td>
</tr>
<tr>
<td>Mode Stream</td>
<td>MS</td>
<td>Default mode. Streaming touch reports.</td>
</tr>
<tr>
<td>Null Command</td>
<td>Z</td>
<td>Returns an ACK. No effect.</td>
</tr>
<tr>
<td>Output Identity</td>
<td>OI</td>
<td>Returns Identity string: ADxxxx</td>
</tr>
<tr>
<td>Soft Reset</td>
<td>R</td>
<td>Resets certain modes and gets a new baseline.</td>
</tr>
<tr>
<td>Restore Defaults</td>
<td>RD</td>
<td>Restores default settings.</td>
</tr>
<tr>
<td>Unit Type</td>
<td>UT</td>
<td>Returns the Status string: AD****ss.</td>
</tr>
</tbody>
</table>

Calibrate

Syntax:  

<SOH>CX<CR>

Response:  

<SOH>1<CR>  Valid point response.

A Valid Point Response (‘1’) signifies that a touch coordinate was detected and is in range of the expected screen target area. Two valid point responses indicate successful calibration.

<SOH>0<CR>  Invalid point response

An Invalid Point Response (‘0’) is returned if the touch coordinate is out of range. If either calibration point is invalid, the touch coordinates are discarded and the calibration points remain unchanged from their previous values.

<SOH>2<CR>  Invalid point response

An Invalid Point Response (‘2’) is returned if the user did not touch the target long enough to provide an accurate point.

Description:  

Initiates an interactive 2-point calibration.

During the calibration process, you define the active area of the touch sensor by mapping locations to an absolute X/Y coordinate system. You touch two target areas on the sensor. Touching the target areas sends the X/Y coordinates for those touch points to the controller. The controller calculates all other touch points based on these two points.

The Calibrate command sets the calibration targets (points) 12½ percent inward from the corner of the video image. Setting the targets inward makes the calibration process easier and more accurate.

It also establishes screen orientation.
Calibrate Procedure

To use the CX command:

Enter the Calibrate (CX) command. The controller sends an initial acknowledgment of 
<SOH>0<CR>.

1. Touch the sensor at a lower left target, which is located 12.5% (1/8) in from the corner of the video image. The controller returns an acknowledgment of 
<SOH>1<CR>. This is a positive response. If you receive a negative response 
<SOH>0<CR>, you can cancel the calibration by issuing a Reset command and then start over again.

2. Touch the sensor at an upper right target, which is located 12.5% (1/8) in from the corner of the video image. The controller returns an acknowledgment of 
<SOH>1<CR>. This is a positive response. If you receive a negative response, you must start over again.

Touching the two valid calibration points result in a successful calibration. If either calibration point is invalid, the calibration fails. The PX4nn controller restores the previous calibration values. If Calibrate fails, repeat the CX process.

Guidelines for Calibrate Command

Here are several guidelines for using the Calibrate commands:

- The controller uses the data immediately before liftoff to register a calibration touch. Therefore, you can touch the coordinate target, hold for a few seconds, and then lift off. Instructing users to touch this way results in a more accurate calibration.

- The controller stores the data in non-volatile memory (NOVRAM). Therefore, you do not have to calibrate the sensor each time you power on the system. You should, however, recalibrate the touch sensor any time the video display changes size or resolution.

- You can cancel calibration at any time during this sequence by issuing a Reset command.

Determining Target Areas

The default calibration targets (points) are located 12.5% (1/8) inward from the corners of the video image. For example, the resolution of your Windows-based display is 1680 x 1050. The Calibrate Extended command calculates the amount to move inward as follows:

- Amount to move inward in the X direction: 1680 x 1/8 = 210
- Amount to move inward in the Y direction: 1050 x 1/8 = 131

The Calibrate Extended command then positions the first calibration target inward from the lower left corner (0,1049) and the second calibration target inward from the upper right corner (1679,0). The following illustration shows how the calibration targets are calculated for a Windows-based system. Your operating system may be different.
The illustration below shows the coordinates of the calibration targets and display corners. The corners show the video coordinates in parentheses and the touch screen coordinates in brackets. Note that the touch screen coordinates for the serial interface have their origin in the lower-left corner.

Note: Other screen resolutions will scale proportionally. The touch coordinates will not change.

### Mode Down Up

**Syntax:**  
\(<SOH>\text{MDU}<CR>\)

**Response:**  
\(<SOH>\text{0}<CR>\) Positive response.

**Description:** Limits touch reports to touchdown and liftoff events only.

### Mode Inactive

**Syntax:**  
\(<SOH>\text{MI}<CR>\)

**Response:**  
\(<SOH>\text{0}<CR>\) Positive response.

**Description:** Turns off touch reports.

### Mode Stream

**Syntax:**  
\(<SOH>\text{MS}<CR>\)

**Response:**  
\(<SOH>\text{0}<CR>\) Positive response.

The controller will respond with acknowledge (ACK) for compatibility. This format is the only format supported by this controller.

**Description:** This is the default mode of operation at power up. Touch reports include touchdown and liftoff events as well as continuing touches.

### Null Command

**Syntax:**  
\(<SOH>\text{Z}<CR>\)

**Response:**  
\(<SOH>\text{0}<CR>\) Positive response.
Description: Queries the controller and waits for a response.

Use Z to determine that you are communicating with the controller or to make sure that a utility is communicating with the controller. Using this command does not affect the controller’s current operating parameters.

Output Identity
Syntax: <SOH>OI<CR>
Response: <SOH>ADXxxx<CR>
where:
AD = Two ASCII characters that describe the type of 3M Touch Systems controller.
Xxxx = This is a unique identifier not necessarily the actual firmware revision number.

Description: Returns a 6-character identifier, which describes the controller type and the firmware version number.

3M Touch Systems recommends that the host system issue a Reset command whenever the host system is powered on and is attempting to establish communication with the controller.

The amount of time needed to execute a Reset command is typically 500 milliseconds with a defined maximum of 2000 milliseconds. Therefore, the application program should wait and be sure it receives the command response before issuing another command to the controller following the reset.

Restore Defaults
Syntax: <SOH>RD<CR>
Response: <SOH>0<CR> Positive response.

Description: Returns to the factory default operating parameters. The Restore Defaults command copies the 3M Touch Systems factory default parameters from ROM to the non-volatile memory (NOVRAM).

If you did not connect the sensors tails correctly, you may not get the full advantage of your touch screen size. Typically, on initial power-up, the controller will automatically detect the correct size of the touch sensor. It will not support the touch sensor until you reboot the controller.

The following table lists the factory defaults for the PX4nn controller. The Restore Defaults command is useful in situations where inadvertent commands to the controller have rendered the sensor inoperative.
Table 19. PX4nn Factory Default Settings

<table>
<thead>
<tr>
<th>Operating Parameter</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baud Rate</td>
<td>115,200</td>
</tr>
<tr>
<td>Serial Communication Settings</td>
<td>N, 8, 1</td>
</tr>
<tr>
<td>Data Format</td>
<td>Format Tablet Multitouch</td>
</tr>
<tr>
<td>Operating Mode</td>
<td>Mode Stream</td>
</tr>
<tr>
<td>Return to Factory Calibration</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**Note**: After you issue a Restore Defaults command, you must recalibrate your sensor.

**Soft Reset**

Syntax: \(<SOH>\)R<CR>

Response: \(<SOH>0<CR>\) Positive response.

Description: Initializes the hardware and the firmware, causes the controller to stop sending data, and recalculates the environmental conditions (for example, stray and offset values). The Reset command also cancels the Calibrate command and returns the controller to normal operation.

**Unit Type**

Syntax: \(<SOH>UT<CR>\)

Description: Responds with an 8-character identity string. This string identifies the type of controller currently attached to the system, lists the features supported by the controller, and outputs the status of the controller hardware (self-test code).

Returns an identification code up to 8 ASCII characters in the following format:

\(<SOH>TtFFFfFs<CR>\)

where:

- \(Tt\) = Two ASCII characters that identify the controller type. AD = Indicates the PX4nn series of controllers
- \(FFFf\) = Four ASCII characters that indicate the features supported by the controller. **** = Indicates no additional features configured
- \(Ss\) = Two ASCII characters that provide status information about the controller hardware. The two characters represent one byte. Each character is in the range 0 to 9 and A to F. Table 2 defines the meaning of each bit in the status byte. Each bit can be set to 1 or 0, where 1 = an error and 0 = no error. So a response of:

\(00\) = No diagnostic errors (normal response)

The PX4nn running with no errors returns the following string:

\(<SOH>AD****00<CR>\)

The format includes two bitmapped ASCII hex status bytes. A “00” indicates no errors.
<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Not used</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>ROM_ERROR</td>
<td>Code area checksum error</td>
</tr>
<tr>
<td>2</td>
<td>PWM Error</td>
<td>Touch screen not connected or potential problem.</td>
</tr>
<tr>
<td>3</td>
<td>NOV_ERROR</td>
<td>Parameter Block1 checksum error</td>
</tr>
<tr>
<td>4</td>
<td>HDW_ERROR</td>
<td>Problem with ADCs</td>
</tr>
<tr>
<td>5</td>
<td>Not used</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Not used</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Not used</td>
<td></td>
</tr>
</tbody>
</table>
APPENDIX A

PX4nn Controller Specifications

Regulatory Requirements

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Class</th>
<th>Compliance</th>
</tr>
</thead>
<tbody>
<tr>
<td>CE</td>
<td></td>
<td>Compliant</td>
</tr>
<tr>
<td>Radiated Emissions – EN 55022:1998</td>
<td>Class B*</td>
<td>Compliant</td>
</tr>
<tr>
<td>AC Mains Conducted Emissions – EN 55022:1998</td>
<td>Class B</td>
<td>Compliant</td>
</tr>
<tr>
<td>Telco Lines Conducted Emissions</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>RFI – EN 61000-4-3 / ENV 50140</td>
<td>Class A</td>
<td>Compliant</td>
</tr>
<tr>
<td>CRFI – EN 61000-4-6</td>
<td>Cable &lt; 3 meters long</td>
<td>N/A</td>
</tr>
<tr>
<td>EFT (Burst Immunity) – EN 61000-4-4</td>
<td>Class B</td>
<td>Compliant</td>
</tr>
<tr>
<td>ESD Susceptibility – IEC 61000-4-2</td>
<td>Class 1</td>
<td>Compliant</td>
</tr>
<tr>
<td>Surge – EN 61000-4-5</td>
<td>Class B</td>
<td>Compliant</td>
</tr>
<tr>
<td>Harmonics – EN 61000-3-2</td>
<td>Class A</td>
<td>Compliant</td>
</tr>
<tr>
<td>Flicker – EN 61000-3-3</td>
<td></td>
<td>Compliant</td>
</tr>
<tr>
<td>Power Frequency Magnetic Field – EN 61000-4-8</td>
<td>Class A</td>
<td>Compliant</td>
</tr>
<tr>
<td>Voltage Dips – EN 61000-4-11</td>
<td>Class B</td>
<td>&lt; 5% V</td>
</tr>
<tr>
<td></td>
<td>Class C</td>
<td>&lt; 70% V</td>
</tr>
<tr>
<td>Voltage Intermittence – EN 61000-4-11</td>
<td>Class C</td>
<td>Compliant</td>
</tr>
<tr>
<td>FCC Class B / CISPR22 Class B</td>
<td>Class B</td>
<td>Compliant</td>
</tr>
<tr>
<td>VCCI Class B ITE Emissions (Japan)</td>
<td>Class B</td>
<td>Compliant</td>
</tr>
<tr>
<td>AS/NZS 3548:1995/CISPR 22 Class B ITE Emissions (Aus.)</td>
<td>Class B</td>
<td>Compliant</td>
</tr>
<tr>
<td>UL 60950/EN 60950/IEC 60950</td>
<td></td>
<td>Compliant</td>
</tr>
</tbody>
</table>

*EMC performance is dependent upon proper integration. Refer to the 3M Multi-Touch System PCT2000PX Series Integration Guide (TSD-39954) for detailed information.

Ambient Operating and Storage Environmental Conditions

All Humidity is Non-Condensing

- Operating Temperature Range: - 40 °C to +70 °C
- Operating Humidity Range: < 36 °C 0-90% RH
  ≥ 36 °C (see Figure below)
- Storage Temperature Range: - 50 °C to +85 °C
Performance and Reliability

**Touch Response Time**

<10 ms for up to 5 simultaneous finger inputs or <10 ms for up to 10 simultaneous finger inputs when integrated with a 3M PCT sensor

**Touch Resolution**

Maximum addressable coordinates generated by the controller 16K x 16K

**ESD Susceptibility**

±8 kV Contact Discharge* – Class 2 per section 9 of IEC 61000-4-2  Compliant

1 false touch allowed

±15 kV Air Discharge* – Class 1 per section 9 of IEC 61000-4-2  Compliant

Normal Operation – No false touches

* ESD discharges to a 3M Touch Systems touch screen connected to the controller

MTBF (by MIL Std. 217F Calculation)  > 700,000 Hours

Touch System Parameters

**Accuracy vs. Dynamic Temperature Change**

Maintains 99% Accuracy

(Tested at 0 deg. C to 60 deg. C with a 0.5 deg. C/minute temperature ramp)

**Touch Screen Compatibility**

3M Touch Systems Projected Capacitive Touch Screens
Communications Protocol
Either USB or dual mode

Warranty Period
See Standard Terms and Conditions, unless otherwise specifically agreed to by 3M Touch Systems in writing.

PX441 USB Controller

Dimensions
2.75 in. x 2.75 in. x 0.374 in.  (69.85 mm x 69.85 mm x 9.51 mm)

Power
USB Vbus = 5 VDC (238 mA max)
PX443 Dual-Mode Controller

Dimensions
2.75 in. x 2.75 in. x 0.374 in.  (69.85 mm x 69.85 mm x 9.51 mm)

Power
5 VDC (238 mA max)
PX451 USB Controller

Dimensions
2.75 in. x 3.25 in. x 0.374 in.  (69.85 mm x 82.55 mm x 9.51 mm)

Power
USB Vbus = 5 VDC (310 mA max)
PX453 Dual-Mode Controller

Dimensions
2.75 in. x 3.25 in. x 0.374 in. (69.85 mm x 82.55 mm x 9.51 mm)

Power
5 VDC (314 mA max)
PX461 USB Controller

Dimensions
2.75 in. x 3.25 in. x 0.374 in.  (69.85 mm x 82.55 mm x 9.51 mm)

Power
USB Vbus = 5 VDC (362 mA max)
PX463 Dual-Mode Controller

Dimensions
2.75 in. x 3.25 in. x 0.374 in. (69.85 mm x 82.55 mm x 9.51 mm)

Power
5 VDC (366 mA max)
PX471 USB Controller

**Dimensions**

3.325 in. x 3.45 in. x 0.374 in.  (84.46 mm x 87.63 mm x 9.51 mm)

**Power**

USB Vbus = 5 VDC (384 mA max)
PX473 Dual-Mode Controller

Dimensions
3.325 in. x 3.45 in. x .374 in. (84.46 mm x 87.63 mm x 9.51 mm)

Power
5 VDC (402 mA max)
PX4611 USB Controller (33” – 47”)

Dimensions
9.90 in. x 2.50 in. x .395 in.  (251.46 mm x 63.50 mm x 10.03 mm)

Power
5 VDC (780 mA max)
PRX4401 Satellite board (33” – 47”)

Dimensions
4.33 in. x 2.165 in. x .39 in. (110.0 mm x 55.0 mm x 9.91 mm)

Power
5 VDC (780 mA max)