3M™ Multi-Touch ASIC Controller PX5nnn Dual Mode Reference Guide

PX5232

Please read, understand and follow all safety information contained in the 3M™ Multi-Touch PCT System Integration Guide found at 3M.com/Touch prior to the use of this device. Retain the Integration Guide for future reference.

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 direct, indirect, special, incidental or consequential (including downtime, loss of profits or goodwill) regardless of the legal theory asserted.**

Document Title*: 3M™ Multi-Touch Controller PX5nnn Dual Mode Reference Guide*
Document Number: TSD-52007, Revision A

Contents

Contents

Chapter 1

Introduction

What You Need to Know 5
Important Safety Information 5
3M Touch Systems Support Services 6
Contact 3M Touch Systems 7

Chapter 2

Integrating the 3M™ Multi-Touch Controller *PX5nnn*

Overview of the 3M *PX5nnn* Controllers 8
Handling and ESD Protection 9
Establishing the Data Connection 9
Sensor Connection 10
Supplying Power to the Controller 11
Mounting the Sensor 12
Windows® Compatibility 12
All Other Platforms 12
Multi-Touch Application Support 12
Video Alignment 13

Chapter 3

3M™ *PX5*nnn Controller USB Communications

Overview of USB Firmware Communications 15
Communication Basics 15
Receiving Reports from the Controller 16
Command Set 16
Set Feature – Calibration 16
Get Feature – Get Status 18
Get Feature – Get Max Count 19
Set Feature – Reset 20
Set Feature – Restore Defaults 21
Get Feature – Get Controller ID 21
Asynchronous Reports 23

Chapter 4

3M™ PX5nnn Serial Controller Communications

Overview of Firmware Commands 25
Controller Default Settings 26
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
Controller Drawings 34

CHAPTER 1

Introduction

3M Touch Systems offers several multi-touch 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 to work with sensors ranging from 16 inches to 24 inches.

This reference guide, designed for developers of touch systems, provides installation and configuration information for the 3M Multi-Touch Controller *PX5nnn*. This document includes information on integrating the 3M *PX5nnn* controller into your design, communicating with the controller and troubleshooting setup problems. It also includes a complete description of the firmware commands and controller specifications.

Table 1. 3M™ Multi-Touch Controller Descriptions

Sensor Size	Model Number
16” to 24”	PX5232

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

Please read, understand and follow all safety information marked on the product and contained in the 3M™ Multi-Touch PCT System Integration Guide found at 3M.com/Touch prior to the use of this device.

Intended Use
The 3M™ Projected Capacitive Touch (PCT) Systems are designed for adding touch input functionality to an existing display. These kits are intended for professional integration and use in an indoor environment. They are not designed or tested for use in hazardous locations. Use in any other application has not been evaluated by 3M and may lead to an unsafe condition.

Disposal

Dispose components in accordance with all applicable local and governmental regulations.

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 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 or by sending an email.

- 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: www.3m.com/touch.

CHAPTER 2

Integrating the 3M™ Multi-Touch Controller *PX5nnn*

The 3M *PX5nnn* 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 3M *PX5nnn* 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 3M *PX5nnn* controller specifications:

- Cable connections
- Mounting requirements
- Power requirements and options

Overview of the 3M *PX5nnn* Controllers

The 3M *PX5nnn* controller has a built-in Universal Serial Bus (USB) full speed interface and a plug and play RS232 interface. A full speed USB interface has a data rate of 12 Mb/s.

To integrate and test the 3M *PX5nnn* 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
- Many operating systems, including Microsoft® Windows® and Linux®, naturally provide support to the touch controllers – No additional software is needed. Contact 3M Touch Systems if you need support.

- 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.

Handling and ESD Protection

When mounting the sensor and controller, use normal precautions for handling electrostatic sensitive devices. The 3M Multi-Touch Controller *PX5nnn* has internal protection to ±4KV for ESD discharges to the controller or touch sensor surface that may occur during normal assembly operations.

Establishing the Data Connection

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 on the PX5000 series controller. The other end has a Type-A connector, and plugs into a USB port on your PC.

When creating a custom cable, use the components found in Table 4 (Page 11). The following table describes the interconnections of the 3M Touch Systems USB cable.

Table 2. USB Cable for 3M™ Multi-Touch Controllers PX5000 Series

PC Side (USB Type A)		Wire	Controller Side (5-Pin Molex)	
Pin	**USB Assigned**	**Color**	**Pin**	**Description**
1	+5Vdc (VBUS)	Red	1	+5Vdc VBUS power
2	Data (DN)	Gray	2	Data (DN) differential pair
3	Data (DP)	Green	3	Data (DP) differential pair
4	0V	Black	4	Power return
5	Cable Shield Shell	Charcoal Gray	5	Outer cable shield around signal and power lines. Chassis (earth) ground

Serial Connection

For serial mode, the PX5nnn 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 on the PX5000 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 components found in Table 4 (Page 11). The following table describes the interconnections of the 3M Touch Systems serial cable.

Table 3. Serial Cable for PX5nnn Series Controllers

PC Side (9-Pin D)			Controller Side (8-Pin Molex)	
Pin	**RS-232 Assigned**	**Jumpered to:**	**Pin**	**Description**
1	Data Carrier Detect (DCD)	4 and 6 DTR and DSR	8	DCD, DTR, DSR
2	Receive Data (RXD)		2	Transmit Data (TXD)
3	Transmit Data (TXD)		3	Receive Data (RXD)
4	Data Terminal Ready (DTR)	1 and 6 DCD and DSR	8	DCD, DTR, DSR
5	Signal Ground		5	Power supply ground
6	Data Set Ready (DSR)	1 and 4 DCD and DSR	8	DCD, DTR, DSR
7	Request to Send (RTS)		1	Request to Send (RTS)
8	Clear to Send (CTS)		4	Clear to Send (CTS)
9	Not Used			Do not ground
Sleeve	5V (not used)		6	DC power jack (Not Used)
Pin	Ground		7	Cable shield connected to ground. DC power jack ground

Sensor Connection

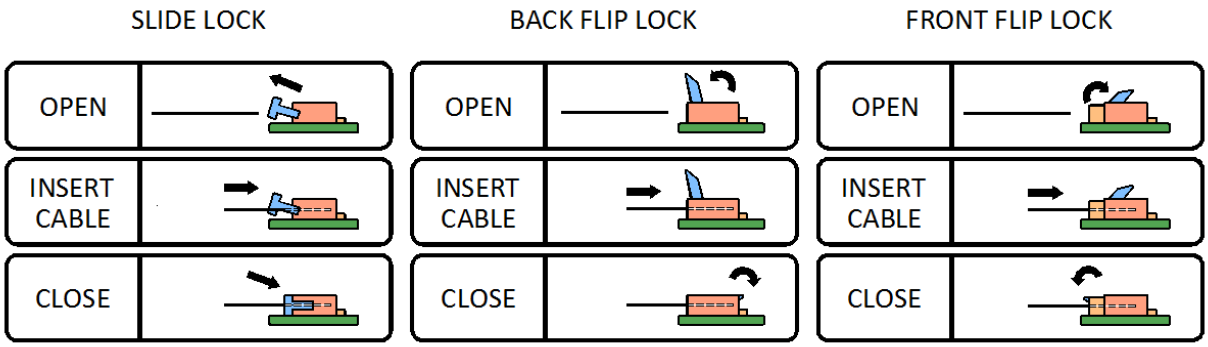
Each controller has a unique set of sensor ZIF mating connectors. The schematics define the number of pins and the BOM defines the vendor part number for each. The sensor flex tails should be plugged directly into the controller connectors.

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.

Take the tail straight from the sensor and carefully align with the ZIF connector.

Connector Locking Styles

Controllers may be built using one or more of the three Connector Locking Stytles shown below.



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.

Supplying Power to the Controller

A 2-pin connector (pin 1 is 5V, pin 2 is ground) is used for 5V power input. When creating a power cable, use the components found in Table 4.

Table 4.

Connector		Molex	Adam Tech	JWT
P1 USB	Housing	51004-0500	2CH-F-05	A2004H00-5P
	Crimp	50011-8100	2CTF-R	A2004TOP-2
P4 SERIAL	Housing	51004-0800	2CH-F-08	A2004H00-8P
	Crimp	50011-8100	2CTF-R	A2004TOP-2
P3 POWER	Housing	51004-0200	2CH-F-02	A2004H00-2P
	Crimp	50011-8100	2CTF-R	A2004TOP-2

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 PCT System Integration Guide (TSD-48194). All 3M Touch Systems documentation is available from the corporate website at www.3m.com/touch.

Windows® Compatibility

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

All Other Platforms

3M Multi-Touch PCT technology also works seamlessly with the Linux® operating system, kernel 3.5 or later.

When using any other operating system, you may need additional drivers and 3M provides the 3M Microtouch™ MT 7 Software for multi-touch drivers for some of these operating systems. Refer to our website www.3m.com/touch – for options.

When using a Windows® 7 serial connection, 3M provides MicroTouch MT 7 Software for multi-touch drivers. This guide includes information installing the touch controller driver and setting the video alignment.

This guide contains all the communication protocols necessary to talk directly with the system electronics. This enables software developers using other operating systems such as Microsoft® Windows® 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.

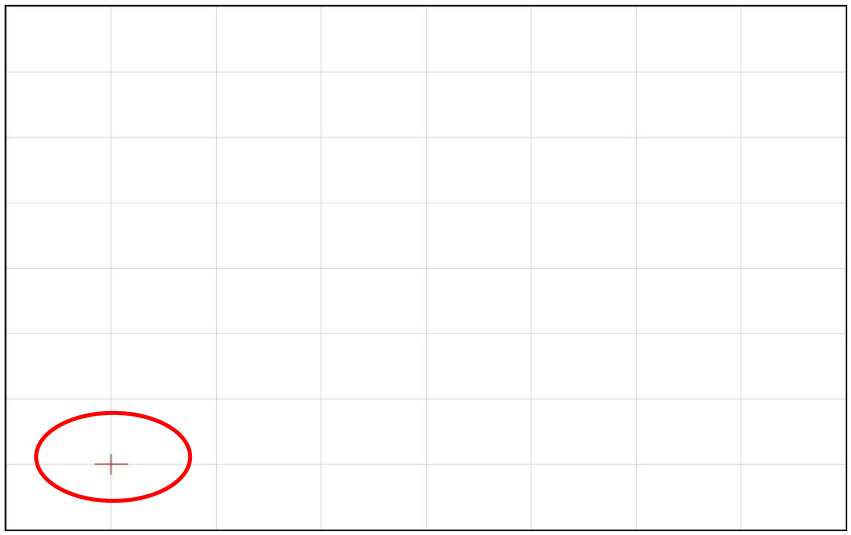
Video Alignment

The 3M Multi-Touch PCT PX 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.

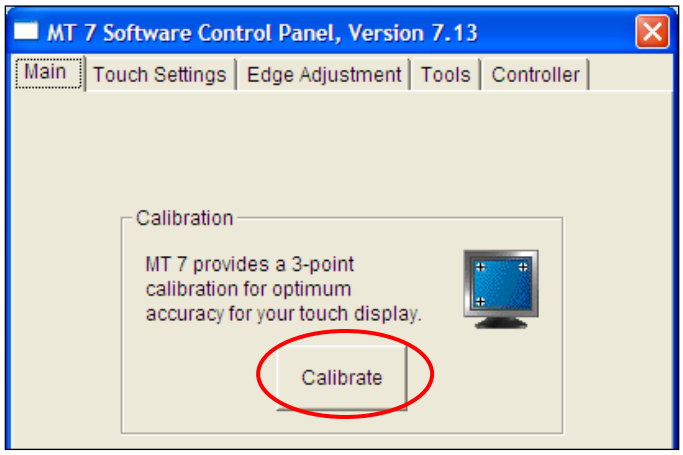
There are three ways to perform a video alignment. Regardless of the operating system, after you connect your touch display:

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.

CHAPTER 3

3M™ *PX5*nnn Controller USB Communications

This chapter is intended for software developers only and discusses the fundamentals of communicating with the 3M™ *PX5nnn* USB 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 *PX5nnn* 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 *PX5nnn* 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 spec # TSD-48146.

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 5. HID Class Requests Summary

HID Report	Command Name	bmRequest Type	bRequest	Feature Report ID	Report Subtype	Data Stage Bytes
Get Feature	GetStatus	0xA1 (D2H)	0x01	0x06	-	8
Set Feature	Calibrate	0x21 (H2D)	0x09	0x03	4	8
Set Feature	Reset	0x21 (H2D)	0x09	0x03	7	8
Set Feature	Restore Defaults	0x21 (H2D)	0x09	0x03	8	8
Get Feature	GetMaxCount	0xA1 (D2H)	0x01	0x12	-	2
Get Feature	GetControllerID	0xA1 (D2H)	0x01	0x04	-	24

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 6. Calibration Setup Stage

Offset	Field	Size	Value	Description
0	bmRequestType	1	0x21	Class,H2D,Interface
1	bRequest	1	0x09	Set Report
2	wValue	2	0x0303	Msb 03 = Feature Lsb 03 = Feature Report ID
4	wIndex	2	0	Always 0
6	wLength	2	8	Always 8

Table 7. Calibration Data Stage

Offset	Field	Size	Value	Description
0	Report ID	1	0x03	Feature report ID
1	Report Subtype	1	0x04	Indicates a calibration request
2	bCalType	1	0xXX	0x01 = Extended cal (CX)
3-7	Not used	5	0	Not used

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 Get Status request to determine the status of this request. The status report includes a command status byte which will be set as shown below.

Table 8. Calibration Response

Command Status Byte	Description
0	Calibration Failed
1	Controller is waiting for a touch in the lower left corner. Calibration software paints a target in the lower left corner.
2	Controller is waiting for a touch in the upper right corner. Calibration software paints a target in the upper right corner.
3	Calibration completed successfully.

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

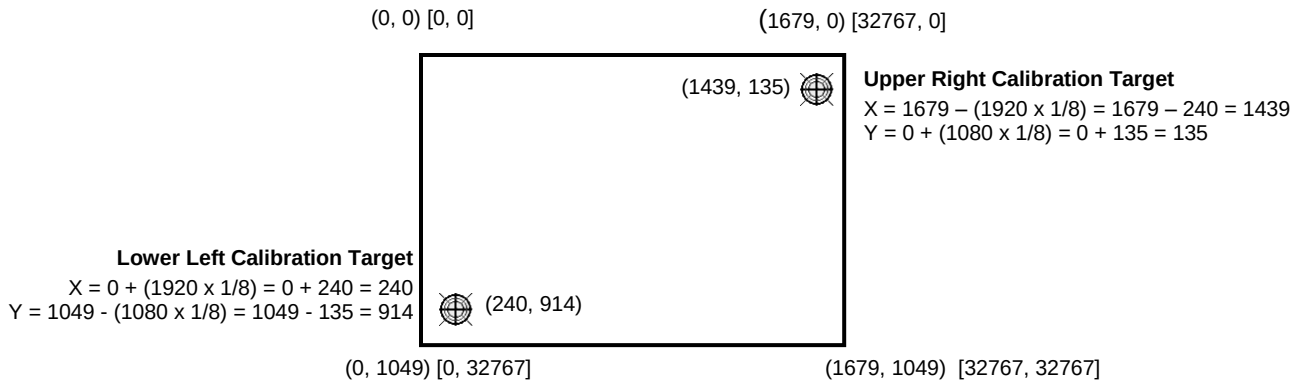
Determining Target Areas

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 – Get Status

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 9. Get Status Setup Stage

Offset	Field	Size	Value	Description
0	bmRequestType	1	0xA1	Class,D2H,Interface
1	bRequest	1	0x01	Get Report
2	wValue	2	0x0306	msb=03=Feature lsb=06= Feature Report ID
4	wIndex	2	0	Always 0
6	wLength	2	8	Always 8

Table 10. Get Status Data Stage

Offset	Field	Size	Value	Description
0	Report ID	1	0x06	Feature Report ID
1	POC Status	1	0xXX	Power On Check Status
2	Cmd Status	1	0xXX	Status of last command
3	Touch Status	1	0	Not used
4	Not Used	1	0xXX	0x00 = Async touch output off 0x01 = Async touch output on
5-7	Not used	3	0	Not used

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 11. Power On Check Bit Fields

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

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 12. Valid Command Status Field Entries

Response	Description
0	Failure in command processing
1	Command being processed
2	Stage 1 processing complete (for multi-stage commands)
3	Command complete
4	Soft Reset Occurred
5	Hard Reset Occurred
6-7	Not used

Get Feature – Get Max Count

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

Table 13. Get Max Count Setup Stage

Offset	Field	Size	Value	Description
0	bmRequestType	1	0xA1	Class,D2H,Interface
1	bRequest	1	0x01	Get Report
2	wValue	2	0x0312	msb=03=Feature lsb=12 Feature Report ID
4	wIndex	2	0	Always 0
6	wLength	2	2	Always 2

Table 14. Get Max Count Data Stage

Offset	Field	Size	Value	Description
0	Report ID	1	0x12	Feature Report ID
1	Max Count	1	0xNN	Number of actual fingers supported (NN= maximum number of "Actual Counts" in the touch report)

Set Feature – Reset

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

Table 15. Reset Setup Stage

Offset	Field	Size	Value	Description
0	bmRequestType	1	0x21	Class,H2D,Interface
1	bRequest	1	0x09	Set Report
2	wValue	2	0x0303	msb=03=Feature lsb=03= Feature Report ID
4	wIndex	2	0	Always 0
6	wLength	2	8	Always 8

Table 16. Reset Data Stage

Offset	Field	Size	Value	Description
0	Report ID	1	0x03	Feature Report ID
1	Report Subtype	1	0x07	Indicates a reset request
2	bResetType	1	0x01 0x02 0x03 0x05	Soft Reset Hard Reset Reboot ROM Reboot EEP
3-7		5	0	Not used

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 17. Restore Defaults Setup Stage

Offset	Field	Size	Value	Description
0	bmRequestType	1	0x21	Class,H2D,Interface
1	bRequest	1	0x09	Set Report
2	wValue	2	0x0303	msb=03=Feature lsb=03= Feature Report ID
4	wIndex	2	0	Always 0
6	wLength	2	8	Always 8

Table 18. Restore Defaults Data Stage

Offset	Field	Size	Value	Description
0	Report ID	1	0x03	Feature Report ID
1	Report Subtype	1	0x08	Restore Defaults
2-7		6	0	Not used

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.

Get Feature – Get Controller ID

This is a request to send various pieces of information, including the controller type, firmware revision level, and the block revision levels.

Table 19. Get Controller ID Setup Stage

Offset	Field	Size	Value	Description
0	bmRequestType	1	0xA1	Class,D2H,Interface
1	bRequest	1	0x01	Get Report
2	wValue	2	0x0304	msb=03=Feature lsb=04= Feature Report ID
4	wIndex	2	0x0000	Always 0
6	wLength	2	0x0018	Always 24

Table 20. Get Controller ID Data Stage

Offset	Field	Size	Value	Description
0	Report ID	1	0x04	Feature Report ID
1	Ctlr type lsb	1	0x44	Indicates the type of controller BCD ‘D1’
2	Ctlr type msb	1	0x31	
3	FWmajor revision	1	0xXX	Bootloader firmware revision (BCD encoding)
4	FWminor revision	1	0xXX	
5	Features	1	0xXX	“Special Features”
6	Code ChkSum lsb	1	0xXX	Application code checksum lsb
7	Code ChkSum msb	1	0xXX	Application code checksum msb
8	MaxParamWrite lsb	1	0x40	Max block data bytes for a set/get param request
9	MaxParamWritemsb	1	0x00	
10	Block1 Rev	1	0x41	‘A’
11	Not used	1	0x00	Not used
12	Not used	1	0x00	Not used
13	Not Used	1	0x00	Not used
14	Block5 Rev	1	A	
15	Block6 Rev	1	A	
16	Block7 Rev	1	A	
17	Block8 Rev	1	A	
18	Not Used	1	0x00	Not used
19	Reserved	1	0xXX	
20	Touch Packet	1	0xXX	0=10 , 2=4, 1=6 touches
21	Reserved	1	0xXX	
22	Boot ChkSum lsb	1	0xXX	Bootloader code checksum lsb
23	Boot ChkSum msb	1	0xXX	Bootloader code checksum msb

Max Parameter Write – The largest transfer that can take place. This only affects the Get and Set Parameter requests.

Data Stage offset 5 Special Features will indicate if Wake On Touch and / or UEFI Loaded Firmware is present. e.g.

Wake on Touch = (Special Features & 0x10)

UEFI Firmware = (Special Features & 0x80)

Asynchronous Reports

Depending on the firmware, one of these two reports will be sent when using this feature.

These are used to transfer the coordinate data to the host. One of these reports, depending of the particular firmware used, is sent to the host whenever new data is available or scheduled for transmission.

Of the two reports, report 0x13 can hold up to 6 simultaneous touches while report 0x17 supports up to 10 simultaneous touches. Note that if there are more than a report can hold, multiple reports are sent as many times as necessary to accommodate the number of fingers touching. Only the first report of a set 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's origin is in the upper left corner of the touch screen, consistent with most operating systems' coordinate systems.

Table 21. Coordinate Data Report 0x13 Data Stage

Offset	Field	Size	Value	Description
0	Report ID	1	0x13	Report ID
1	Touch Report	10	See Table 22	Touch Report Structure
11	Touch Report	10	See Table 22	Touch Report Structure
21	Touch Report	10	See Table 22	Touch Report Structure
31	Touch Report	10	See Table 22	Touch Report Structure
41	Touch Report	10	See Table 22	Touch Report Structure
51	Touch Report	10	See Table 22	Touch Report Structure
61	Actual Count	1	1 to max	Number of valid touch reports (no more than declared max touch reports)
62	Not used	1	0	
63	Not used	1	0	

Table 22. Touch Report Structure for Report 0x13

Offset	Field	Size	Value	Description
0	Status	6	0xXX	0x00 Report not valid 0x04 Not touching 0x07 Touching
1	Touch ID	1	0-255	Not used
2	X lsb	1	0xXX	X (0-7FFF)
3	X msb	1	0xXX	
4	Y lsb	1	0xXX	Y (0-7FFF)
5	Y msb	1	0xXX	

Table 23. Coordinate Data Report 0x17 Data Stage

Offset	Field	Size	Value	Description
0	Report ID	1	0x17	Report ID
1	Touch Report	6	See Table 24	touch report structure
7	Touch Report	6	See Table 24	touch report structure
13	Touch Report	6	See Table 24	touch report structure
19	Touch Report	6	See Table 24	touch report structure
25	Touch Report	6	See Table 24	touch report structure
31	Touch Report	6	See Table 24	touch report structure
37	Touch Report	6	See Table 24	touch report structure
43	Touch Report	6	See Table 24	touch report structure
49	Touch Report	6	See Table 24	touch report structure
55	Touch Report	6	See Table 24	touch report structure
61	Actual Count	1	1 to MaxCount	Number of valid touch reports (no more than declared max touch reports)
62	Scan Time	2	XXXX	Scan Time

Table 24. Touch Report Structure for Report 0x17

Offset	Field	Size	Value	Description
0	Status	1	0xXX	0x00 Report not valid 0x04 Not touching 0x07 Touching
1	Touch ID	1	0-255	Touch thread ID number
2	X lsb	1	0xXX	X (0-7FFF)
3	X msb	1	0xXX	
4	Y lsb	1	0xXX	Y (0-7FFF)
5	Y msb	1	0xXX	

CHAPTER 4

3M™ PX5nnn Serial Controller Communications

This chapter is intended for software developers only and discusses the fundamentals of communicating with the 3M™ PX5nnn 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™ PX5nnn 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 PX5nnn 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™ PX5nnn 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 25. Touch Data Format Settings

Data Sequence	Bits							
	7	***6***	***5***	***4***	***3***	***2***	***1***	***0***
Status - Byte 1	1	P[1]	0	0	0	0	0	0
X - Byte 2	0	X6	X5	X4	X3	X2	X1	X0
X - Byte 3	0	X13	X12	X11	X10	X9	X8	X7
Y - Byte 4	0	Y6	Y5	Y4	Y3	Y2	Y1	Y0
Y - Byte 5	0	Y13	Y12	Y11	Y10	Y9	Y8	Y7
ID – Byte 6	0	N6	N5	N4	N3	N2	N1	N0

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™ PX5nnn 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™ PX5nnn 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™ PX5nnn 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 19 for current development. Using these commands ensures compatibility with all 3M™ controllers.

Note: When you enter commands in terminal mode, precede each command with <CTRL> A to enter the start of header.

Table 26. Firmware Commands for PX5nnn Serial Controller Development

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

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 PX5nnn 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.

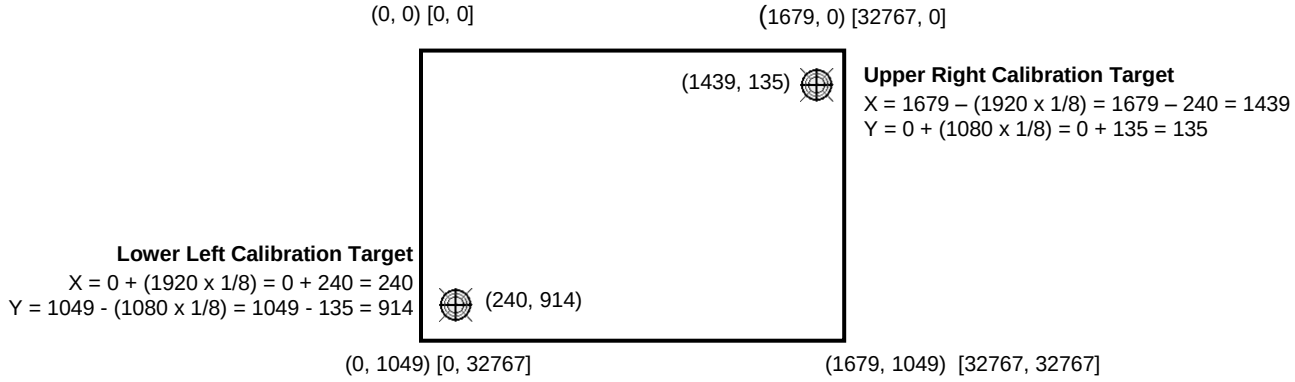
Determining Target Areas

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>**MDU**<CR>

Response: <SOH>**0**<CR> Positive response.

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

Mode Inactive

Syntax: <SOH>**MI**<CR>

Response: <SOH>**0**<CR> Positive response.

Description: Turns off touch reports.

Mode Stream

Syntax: <SOH>**MS**<CR>

Response: <SOH>**0**<CR> Positive response.

Description: The controller will respond with acknowledge (ACK) for compatibility. This format is the only format supported by this controller.
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>**Z**<CR>

Response: <SOH>**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 PX5nnn controller. The Restore Defaults command is useful in situations where inadvertent commands to the controller have rendered the sensor inoperative.

Table 27. PX5nnn Factory Default Settings

Operating Parameter	Default
Baud Rate	115,200
Serial Communication Settings	N, 8, 1
Data Format	Format Tablet Multitouch
Operating Mode	Mode Stream
Return to Factory Calibration	Yes

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 (a self-test code).

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

<SOH>***TtFfffSs***<CR>

where:

Tt = Two ASCII characters that identify the controller type.
AD Indicates the PX5nnn 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 PX5nnn 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 28. Bit Meanings in Status Byte

Bit	Description	Notes
0	Not used	
1	ROM_ERROR	Code area checksum error
2	PWM Error	Touch screen not connected or potential problem.
3	NOV_ERROR	Parameter Block1 checksum error
4	HDW_ERROR	Problem with ADCs
5	Not used	
6	Not used	
7	Not used	

Controller Drawings

Request drawings from your 3M Touch Systems representative.