mXT640UD-CCUBHA1 1.0 Family

Protocol Guide

DEVICES

mXT640UD-CCUBHA1

mXT448UD-CCUBHA1

© 2021 Microchip Technology Inc. and its subsidiaries

TABLE OF CONTENTS

To Our Valued Customers 4
1.0 Overview 5
1.1 Introduction 5
1.2 Memory Map Structure 5
1.3 Information Block 6
1.4 ID Information 6
1.5 Object Table 7
1.6 Objects 9
1.7 Configuration Checks 11
1.8 Byte Order 11
1.9 Delta Values 11
1.10 Configuration and Tuning 12
1.11 Software Tools 12
2.0 Debug Objects 13
2.1 Introduction 13
2.2 Diagnostic Debug T37 Object 14
3.0 General Objects 21
3.1 Introduction 21
3.2 Message Processor T5 Object 22
3.3 Command Processor T6 Object 24
3.4 Power Configuration T7 Object 29
3.5 Acquisition Configuration T8 Object 34
4.0 Touch Objects 43
4.1 Introduction 43
4.2 Key Array T15 Object 44
4.3 Multiple Touch Touchscreen T100 Object 53
5.0 Signal Processing Objects 93
5.1 Introduction 93
5.2 Key Thresholds T14 Object 94
5.3 Key ID Configuration T16 Object 95
5.4 One-touch Gesture Processor T24 Object 96
5.5 Two-touch Gesture Processor T27 Object 105
5.6 Grip Suppression T40 Object 108
5.7 Touch Suppression T42 Object 112
5.8 Passive Stylus T47 Object 117
5.9 Shieldless T56 Object 126
5.10 Lens Bending T65 Object 128
5.11 Noise Suppression T72 Object 134
5.12 Glove Detection T78 Object 152
5.13 Retransmission Compensation T80 Object 156
5.14 Self Capacitance Noise Suppression T108 Object 163
5.15 Self Capacitance Grip Suppression T112 Object 174
5.16 Ignore Nodes T141 Object 177

Under NDA

© 2021 Microchip Technology Inc. and its subsidiaries

6.0 Support Objects 178
6.1 Introduction 178
6.2 Self Test Control T10 Object 180
6.3 Self Test Pin Faults T11 Object 191
6.4 Self Test Signal Limits T12 Object 194
6.5 Communications Configuration T18 Object 198
6.6 GPIO Configuration T19 Object 201
6.7 User Data T38 Object 204
6.8 CTE Configuration T46 Object 205
6.9 Timer T61 Object 213
6.10 Dynamic Configuration Controller T70 Object 215
6.11 Dynamic Configuration Container T71 Object 224
6.12 Touch Event Trigger T79 Object 225
6.13 Auxiliary Touch Configuration T104 Object 227
6.14 Self Capacitance Global Configuration T109 Object 231
6.15 Self Capacitance Tuning Parameters T110 Object 235
6.16 Self Capacitance Configuration T111 Object 236
6.17 Self Capacitance Measurement Configuration T113 Object 248
6.18 Data Container T117 Object 251
6.19 Data Container Controller T118 Object 253
6.20 Self Capacitance Voltage Modulation T133 Object 256
6.21 Message Count T144 Object 259
6.22 Ignore Nodes Controller T145 Object 260
Appendix A. Measurement Processing 262
A.1 Measurement Processing Objects 262
A.2 Touch Detection 263
A.3 Touch Classification 263
Appendix B. Checksum Calculation 265
B.1 24-bit CRC 265
B.2 8-bit CRC 267
Appendix C. Bootloading Procedure 273
C.1 Bootloader ID 273
C.2 I^2C address 273
Appendix D. Locating Configuration Errors 274
Appendix E. Future Information Block T254 Object 275
E.1 Introduction 275
E.2 Information Block T254 275
Appendix F. Associated Documents 277
Appendix G. Revision History 278

TO OUR VALUED CUSTOMERS

It is our intention to provide our valued customers with the best documentation possible to ensure successful use of your Microchip products. To this end, we will continue to improve our publications to better suit your needs. Our publications will be refined and enhanced as new volumes and updates are introduced.

If you have any questions or comments regarding this publication, please contact the Marketing Communications Department via E-mail at **docerrors@microchip.com**. We welcome your feedback.

Most Current Document

To obtain the most up-to-date version of this document, please contact your Microchip representative.

You can determine the version of a document by examining its literature number found on the bottom outside corner of any page. The last character of the literature number is the version number, (for example, DS30000000A is version A of document DS30000000).

Errata

An errata sheet, describing minor operational differences from the protocol guide and recommended workarounds, may exist for current devices. As device/documentation issues become known to us, we will publish an errata sheet. The errata will specify the revision of silicon and revision of document to which it applies.

To determine if an errata sheet exists for a particular device, please contact your Microchip representative.

When contacting Microchip, please specify which device, revision of silicon and data sheet (include literature number) you are using.

Customer Notification System

Register on our web site at **http://www.microchip.com** to receive the most current information on all of our products.

Under NDA
© 2021 Microchip Technology Inc. and its subsidiaries

1.0 OVERVIEW

1.1 Introduction

The maXTouch Object Protocol provides a single common interface across the Microchip maXTouch controllers. This allows the different features in each controller to be configured in a consistent manner. This makes the future expansion of features and simple product upgrades possible, whilst allowing backwards compatibility for the host driver and application code.

This document describes in detail the Object Protocol for the following devices:

- mXT640UD-CCUBHA1 1.0.AA
- mXT448UD-CCUBHA1 1.0.AA

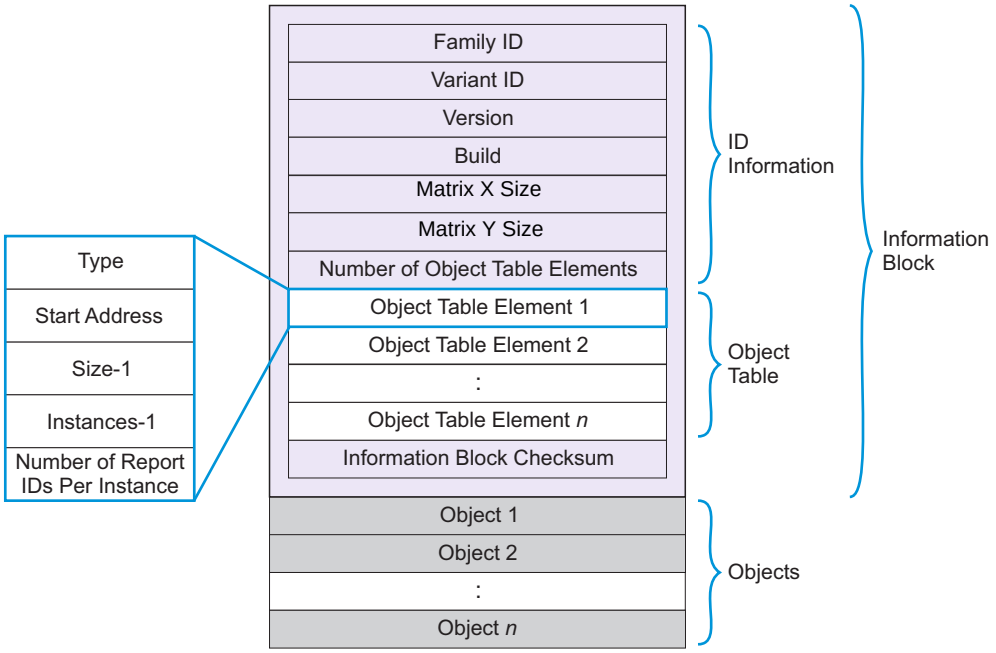
1.2 Memory Map Structure

The protocol is designed to control the processing chain in a modular manner. This is achieved by breaking the features of the device into objects that can be controlled individually. Each object represents a certain feature or function of the device, such as a **touchscreen**. Where appropriate, objects can be disabled or enabled as needed.

Each object has its own configuration memory. The objects are stacked together to produce an object-based memory map. A generalized structure of this memory map is shown in Figure 1-1.

There are some special objects that can use their memory space for a unique purpose. An example is the Command Processor T6 object, which executes a command when a certain value is written into its memory space.

FIGURE 1-1: GENERIC MEMORY MAP STRUCTURE



From Figure 1-1 it can be seen that the memory map contains two main sections:

- An Information Block. This documents which objects are contained in the memory map for the device (see Section 1.3 “Information Block”). It is further sub-divided into the ID Information Block (see Section 1.4 “ID Information”) and the Object Table (see Section 1.5 “Object Table”).
- The objects themselves (see Section 1.6 “Objects”).

1.3 Information Block

The Information Block allows the host to read information about the layout of objects in the memory map. It contains a list of all the objects in the memory map. This is used by the host driver to determine which objects exist, where they are located in the memory map and their sizes. The host driver can therefore read the device Information Block and gather enough information to be able to communicate with the device.

The Information Block is positioned at the start of the device memory map at address zero. This allows it to be read easily by the host as the first operation.

The Information Block contains the following three sections:

- **ID Information Fields** – These include:
  - Standard ID fields that make up the unique identifier for the device
  - The size of the sensor matrix the device supports
  - The number of objects in the Object Table
- **Object Table** – This acts as an “index” to the objects in the memory map. Note that one of the objects may be an Information Block object that holds additional Object Table entries (see Appendix E “Future Information Block T254 Object”).
- **Information Block Checksum** – This allows the host to check that the Information Block has been read correctly over the communications interface.

Table 1-1 shows the contents of the Information Block.

TABLE 1-1: INFORMATION BLOCK LAYOUT

Byte	Description of Field	Section
0	Family ID	ID Information
1	Variant ID	
2	Version	
3	Build	
4	Matrix X Size	
5	Matrix Y Size	
6	Number of elements in the Object Table	
7 – 12	Object Table element 1 (6 bytes)	Object Table
13 – 18	Object Table element 2 (6 bytes)	...
...	...	
...	Last Object Table element (6 bytes)	
(*end*-2) – *end*	24-bit checksum (3 bytes)	Checksum Field

1.4 ID Information

The first four bytes of the Information Block are used to identify the device and its version, as in Table 1-2.

TABLE 1-2: DEVICE IDENTIFIER FIELDS

Field	Description
Family ID	A unique identifier that indicates the device family. The family ID for the mXT640UD-CCUBHA1 Family is 0xA6.
Variant ID	A unique identifier that indicates the device variant. The variant IDs for the mXT640UD-CCUBHA1 Family are: • mXT640UD-CCUBHA1: 0x15 • mXT448UD-CCUBHA1: 0x16
Version	The current major and minor firmware version of the device. The upper nibble contains the major version and the lower nibble contains the minor version. For example, firmware version 1.0 is stored as 0x10.

Under NDA

© 2021 Microchip Technology Inc. and its subsidiaries

TABLE 1-2: DEVICE IDENTIFIER FIELDS (CONTINUED)

Field	Description
Build	The firmware build number.

1.5 Object Table

1.5.1 INTRODUCTION

The Object Table is held within the Information Block. The Object Table contains information on all the objects held within the memory map. It indicates which objects are present and their addresses.

Each element in the Object Table has the fields listed in Table 1-3.

TABLE 1-3: FORMAT OF AN OBJECT TABLE ELEMENT

Byte	Description of Field
0	Type
1	Start position LSByte
2	Start position MSByte
3	Size – 1
4	Instances – 1
5	Number of Report IDs per instance

IMPORTANT! Future versions of the device may include the Information Block T254 object. This object will contain additional Object Table entries. For future compatibility, any current driver code should be written to allow for the presence of the Information Block T254 object when it is eventually used. The host driver code must parse the Object Table, as at present, and also process the Information Block T254 object, if this is found to be present on the device. See Appendix E “Future Information Block T254 Object” for more information.

1.5.2 TYPES

Each type of object has a unique type code to identify it. This is the number after the “T” suffix at the end of the object’s name, as given in the object descriptions. For example, the type code for the Command Processor T6 (GEN_COMMANDPROCESSOR_**T6**) is **6**.

1.5.3 START POSITION

Bytes 1 and 2 of the Object Table element holds the start location of the object in the memory map (LSByte and MSByte respectively).

The driver code should ALWAYS read these bytes to find out where in the memory map the object is located and use this address to communicate with the object. The driver code should never use hard-coded addresses for the objects, as these may change with firmware updates.

This means that driver code can be written without making assumptions about the addresses of the objects. This ensures that the code is “future-proof” and will work correctly following firmware updates to the device. It also makes it possible to write common driver code for communication with any Microchip device that uses this object-based protocol approach.

1.5.4 SIZE

Byte 3 of the Object Table element holds the size in bytes (minus 1) of the object in the memory map. This is stored as Size–1, so it is effectively the offset to the end of the object.

1.5.5 NUMBER OF INSTANCES

Byte 4 of the Object Table element holds the number of instances of the object in the memory map, minus 1. The number of instances can be calculated by adding 1 to this number (see Table 1-4 on page 9). The different instances of an object are arranged sequentially in the memory map.

1.5.6 REPORT IDS

If an object sends messages, it is necessary to identify the messages from the object so that they can be correctly interpreted. A report ID is therefore used to identify the source object of a message returned in the Message Processor T5 object (see Section 3.2 “Message Processor T5 Object”).

Report IDs are numbered uniquely and sequentially in the order in which the objects are listed in the Object Table, allowing for the appropriate number of instances for each object. Note the following:

- A report ID of zero is a reserved value for use by Microchip. Report IDs from a user’s perspective therefore effectively start from 1.
- A report ID value of 255 is reserved to indicate an “invalid message” response.
- If an object has report IDs allocated, each instance of the object will have its own block of report IDs.
- In the case of the Multiple Touch Touchscreen T100 object, there is 1 report ID allocated for the screen status message, 1 reserved report ID, and 16 report IDs for the touch status messages.

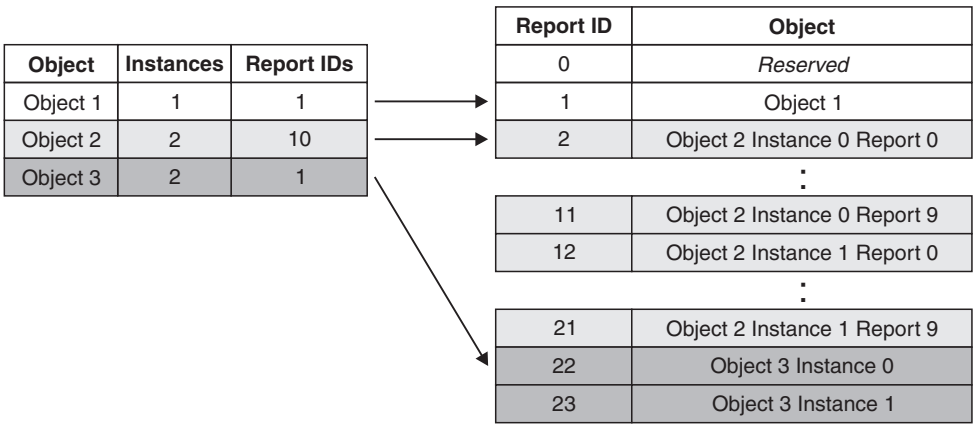
Figure 1-2 shows an example of how the number of instances and the Report IDs per instance determine the report IDs for a set of objects.

The driver code should build up its own in-memory table of object types and associated report IDs during its initialization. It can do this by parsing the object structure given in the Object Table. This in-memory table can then be used to interpret the messages returned by the device.

A typical algorithm to process the report IDs is as follows:

1. For each element in the Object Table:
   a) Read byte 4 to retrieve the number of instances (remember to add 1 to the value retrieved).
   b) Read byte 5 to retrieve the number of report IDs per instance.
   c) Multiply the figures retrieved in steps a. and b. together, and then add this number of “object type/report ID” pairings to the table being built. The report IDs should have sequential values starting with 1 (the zero value is reserved for use by Microchip).
2. Check for the presence of an Information Block T254 object (See Appendix E “Future Information Block T254 Object”). If the object is present on the device, repeat Step 1 for each element held within the Information Block T254 object, continuing to allocate the new report IDs to those already allocated.

FIGURE 1-2: EXAMPLE ASSIGNMENT OF REPORT IDS



The number of report IDs for each object is shown in Table 1-4.

Under NDA

© 2021 Microchip Technology Inc. and its subsidiaries

1.6 Objects

1.6.1 CLASSES OF OBJECTS

The mXT640UD-CCUBHA1 Family contains the following classes of objects:

- **Debug objects** – provide a raw data output method for development and testing. See Section 2.0 “Debug Objects”.
- **General objects** – required for global configuration, transmitting messages and receiving commands. See Section 3.0 “General Objects”.
- **Touch objects** – operate on measured signals from the touch sensor and report touch data. See Section 4.0 “Touch Objects”.
- **Signal processing objects** – process data from other objects (typically signal filtering operations). See Section 5.0 “Signal Processing Objects”.
- **Support objects** – provide additional functionality on the device. See Section 6.0 “Support Objects”.

1.6.2 OBJECT INSTANCES

TABLE 1-4: OBJECTS ON THE MXT640UD-CCUBHA1 FAMILY

Object	Number of Instances	Report IDs per Instance	Reference
Debug Objects			
Diagnostic Debug T37	1	0	Section 2.2 “Diagnostic Debug T37 Object”
General Objects			
Message Processor T5	1	0	Section 3.2 “Message Processor T5 Object”
Command Processor T6	1	1	Section 3.3 “Command Processor T6 Object”
Power Configuration T7	1	0	Section 3.4 “Power Configuration T7 Object”
Acquisition Configuration T8	1	0	Section 3.5 “Acquisition Configuration T8 Object”
Touch Objects			
Key Array T15	1	1	Section 4.2 “Key Array T15 Object”
Multiple Touch Touchscreen T100	1	18	Section 4.3 “Multiple Touch Touchscreen T100 Object”
Signal Processing Objects			
Key Thresholds T14	1	0	Section 5.2 “Key Thresholds T14 Object”
Key ID Configuration T16	1	0	Section 5.3 “Key ID Configuration T16 Object”
One-touch Gesture Processor T24	1	4	Section 5.4 “One-touch Gesture Processor T24 Object”
Two-touch Gesture Processor T27	1	1	Section 5.5 “Two-touch Gesture Processor T27 Object”
Grip Suppression T40	1	0	Section 5.6 “Grip Suppression T40 Object”
Touch Suppression T42	1	0	Section 5.7 “Touch Suppression T42 Object”
Passive Stylus T47	1	0	Section 5.8 “Passive Stylus T47 Object”
Shieldless T56	1	1	Section 5.9 “Shieldless T56 Object”
Lens Bending T65	3	1	Section 5.10 “Lens Bending T65 Object”
Noise Suppression T72	1	1	Section 5.11 “Noise Suppression T72 Object”
Glove Detection T78	1	0	Section 5.12 “Glove Detection T78 Object”
Retransmission Compensation T80	1	1	Section 5.13 “Retransmission Compensation T80 Object”
Self Capacitance Noise Suppression T108	1	1	Section 5.14 “Self Capacitance Noise Suppression T108 Object”

TABLE 1-4: OBJECTS ON THE MXT640UD-CCUBHA1 FAMILY (CONTINUED)

Object	Number of Instances	Report IDs per Instance	Reference
Self Capacitance Grip Suppression T112	1	1	Section 5.15 “Self Capacitance Grip Suppression T112 Object”
Ignore Nodes T141	32	0	Section 5.16 “Ignore Nodes T141 Object”
Support Objects			
Self Test Control T10	1	1	Section 6.2 “Self Test Control T10 Object”
Self Test Pin Faults T11	1	0	Section 6.3 “Self Test Pin Faults T11 Object”
Self Test Signal Limits T12	2	0	Section 6.4 “Self Test Signal Limits T12 Object”
Communications Configuration T18	2	0	Section 6.5 “Communications Configuration T18 Object”
GPIO Configuration T19	1	1	Section 6.6 “GPIO Configuration T19 Object”
User Data T38	1	0	Section 6.7 “User Data T38 Object”
CTE Configuration T46	1	1	Section 6.8 “CTE Configuration T46 Object”
Timer T61	6	1	Section 6.9 “Timer T61 Object”
Dynamic Configuration Controller T70	20	1	Section 6.10 “Dynamic Configuration Controller T70 Object”
Dynamic Configuration Container T71	1	0	Section 6.11 “Dynamic Configuration Container T71 Object”
Touch Event Trigger T79	3	0	Section 6.12 “Touch Event Trigger T79 Object”
Auxiliary Touch Configuration T104	1	0	Section 6.13 “Auxiliary Touch Configuration T104 Object”
Self Capacitance Global Configuration T109	1	1	Section 6.14 “Self Capacitance Global Configuration T109 Object”
Self Capacitance Tuning Parameters T110	12	0	Section 6.15 “Self Capacitance Tuning Parameters T110 Object”
Self Capacitance Configuration T111	2	0	Section 6.16 “Self Capacitance Configuration T111 Object”
Self Capacitance Measurement Configuration T113	1	0	Section 6.17 “Self Capacitance Measurement Configuration T113 Object”
Data Container T117	6	0	Section 6.18 “Data Container T117 Object”
Data Container Controller T118	1	1	Section 6.19 “Data Container Controller T118 Object”
Self Capacitance Voltage Modulation T133	2	0	Section 6.20 “Self Capacitance Voltage Modulation T133 Object”
Message Count T144	1	0	Section 6.21 “Message Count T144 Object”
Ignore Nodes Controller T145	1	0	Section 6.22 “Ignore Nodes Controller T145 Object”

1.6.3 CONFIGURATION VALUES

The objects are designed such that a value of zero in their fields is a “safe” value. Typically a value of zero means that a default value is used, and this will be indicated within the object’s field descriptions in this protocol guide.

An object must be configured as required with non-zero values before use. Any unused settings can be left at their default zero values. The settings should also be written to the non-volatile memory using the Command Processor T6 object (see Section 3.3 “Command Processor T6 Object”).

The field descriptions in this protocol guide specify the recommended ranges for values. Any values outside the specified range for a field should be treated as reserved values and should not be used. Note that in many cases the device will not allow values outside the specified ranges.

Under NDA

© 2021 Microchip Technology Inc. and its subsidiaries

1.6.4 COMPATIBILITY OF OBJECT VERSIONS

The Object Protocol described in this document is specific to a particular firmware version of the device and may contain fields that are not present in the memory map as it is implemented for another firmware version.

Over time, newer versions of the objects in the Object Protocol may gain additional fields to implement new features, and new fields are added at the end of an object to allow for this situation. This preserves the order of the fields between old and new object versions. A driver designed to work with an older version of the device can safely set any unknown fields located at the end of the object to zero. The device will then behave in the same manner as the older version of the device without the field.

IMPORTANT!	The host driver must always use the Object Table to locate the address of each object and the object's current size. It must also zero any fields that it does not intend to use. This ensures that the host driver code is compatible with this object expansion scheme.

1.7 Configuration Checks

The device prevents invalid configurations by performing a sequence of checks. These checks are performed on power-up. They are also performed whenever certain configuration settings are updated. These are typically settings that force a recalculation, a reinitialization or a recalibration when they are changed.

If an error is found during the configuration check, the device pauses until the configuration error is resolved. The device also flags the error to the host by setting the CFGERR bit of the Command Processor T6 message data (see Section 3.3.3 "Messages"). This message will be sent to the host every 200 ms until the error is corrected.

Backup requests are allowed when an error has been found during a configuration check. This allows a setting to be corrected and backed up to the non-volatile memory (NVM). The device can then be reset for the setting to take effect, if the setting requires this. The device will not operate until all errors have been corrected.

Possible causes of configuration errors are:

- Touch objects with overlapping channels or channels outside the maximum matrix dimensions
- Touch objects under the minimum X and Y size, or a Key Array T15 object with a channel size greater than 32
- Field settings outside the permitted range
- A bad checksum for the configuration settings held in the non-volatile memory

If a configuration error is encountered during product development, it is sufficient for the designer to check the settings used and correct them. In a working product, the device driver should resend the configuration settings and request a backup. The device driver should be written to handle this.

To find the source of the configuration error, first disable all touch, signal processing and support objects. Then re-enable each object in turn until the configuration error is found. Once the error has been corrected, the other objects can be re-enabled and processing can continue as normal. See Appendix D "Locating Configuration Errors" for a flow chart showing this method for finding configuration errors.

Notice should be taken of any recommendations in this document concerning configuration checks for the offending object.

1.8 Byte Order

The memory map uses a "little-endian" configuration for its bytes, meaning that all multibyte fields lead with the least significant byte (LSByte). On the rare occasions that the Object-based Protocol uses "big endian" (MSByte first) configuration instead, this is clearly marked.

1.9 Delta Values

During measurement, the device uses raw 16-bit values for the deltas. These are converted by the Object-based Protocol to 8-bit deltas wherever possible in many of the controls and in the diagnostic debug data. This saves space and, in the case of debug data, allows for significant time savings in data transfer.

For Mutual Capacitance and Self Capacitance Touch deltas, the conversion between 16-bit deltas and 8-bit deltas is as follows:

16-bit delta = 8-bit delta × 8

8-bit delta = 16-bit delta / 8

Note that this conversion occurs irrespective of any additional scaling that may occur due to the delta shifting mechanism described in Section 1.9.1 “Delta Scaling”.

1.9.1 DELTA SCALING

In addition to the conversion between 16-bit and 8-bit deltas, the mXT640UD-CCUBHA1 Family can be configured (using CTE Configuration T46 “CFG Field” DELTASHIFT) to scale down deltas in the debug data (excluding single-ended self-capacitance deltas). This reduces the magnitude of deltas which would otherwise be beyond the 8-bit range. In particular, it allows thresholds to be configured as 8-bit values. This feature is useful, for example, with high gains in P2P measurements, or on sensors that produce very large moisture antitouch deltas.

Note that the magnitude of the measured signals is not affected by this shift mechanism, so care must be taken to ensure that the signals are within acceptable limits. This can be done by inspecting the signals, for example by using the Hardware Debug Interface (controlled by Command Processor T6), or by monitoring the results from a Self Test Control T10 Signal Limit test for exceeded thresholds. It is also possible to view self capacitance signals using the Diagnostic Debug T37 object.

An example of using delta shifting is shown in Example 1-1.

EXAMPLE 1-1: DELTA SHIFTING FOR THICK GLOVES

To support thick gloves, gains are chosen which result in touch peak 16-bit deltas of 4000.
A suitable touch threshold is chosen at 3200 on the 16-bit deltas.

By setting DELTASHIFT = 1 (÷2 scaling), the required threshold can be specified as 200, which fits in an 8-bit Object-based Protocol parameter field. Without this scaling, the threshold setting would need to be 400, which is outside the 255 limit for an 8-bit field.

1.10 Configuration and Tuning

This Protocol Guide provides reference information on the various configuration parameters for the mXT640UD-CCUBHA1 Family.

For further guidance on configuring and tuning the mXT640UD-CCUBHA1 Family, refer to the *maXTouch U Series Tuning Guide*.

1.11 Software Tools

The primary interface with the mXT640UD-CCUBHA1 Family device is through the host driver and application code within the user’s product.

An alternative interface is also available by using maXTouch Studio. This provides a graphical IDE within which the configuration and tuning of a maXTouch device can take place so that the performance of the device can be assessed. Note that maXTouch Studio Lite is supplied with the maXTouch evaluation kits.

Various plug-in tools are also available to help with configuring and tuning the various objects within the device and can be integrated into the maXTouch Studio development environment. Contact your Microchip representative for more information.

Under NDA

© 2021 Microchip Technology Inc. and its subsidiaries

2.0 DEBUG OBJECTS

2.1 Introduction

Debug objects contain raw data for development and testing purposes. Table 2-1 lists the debug objects on the mXT640UD-CCUBHA1 Family.

TABLE 2-1: DEBUG OBJECTS

Object	Description
Diagnostic Debug T37 (DEBUG_DIAGNOSTIC_T37)	Allows access to diagnostic debug data to aid development. See Section 2.2 “Diagnostic Debug T37 Object”.

2.2 Diagnostic Debug T37 Object

2.2.1 INTRODUCTION

The Diagnostic Debug T37 object holds the diagnostic debug data. It reports both system data and object-specific data from individual objects. See Section 2.2.1.1 “System Diagnostic Debug Modes” for more information on the system diagnostic debug modes. The object-specific diagnostic debug modes are described elsewhere in this document in the sections for the specific objects.

2.2.1.1 System Diagnostic Debug Modes

The Diagnostic Debug T37 object works by organizing the data in pages. The value of the PAGE field determines which page of data is currently available, and the object's DATA field holds the data for that page. The pages can be navigated by writing Page Up/Page Down commands to the DIAGNOSTIC field in the Command Processor T6 object (see “DIAGNOSTIC Field”).

The mode in which the data is displayed is determined by the command written to the DIAGNOSTIC field of the Command Processor T6 object (see “DIAGNOSTIC Field”).

A mode or page change command is processed only once per measurement cycle. Note that no command processing is done if the device is in deep sleep mode.

The system diagnostic debug modes are shown in Table 2-2. The object-specific diagnostic debug modes are described within the individual object sections.

TABLE 2-2: SYSTEM DIAGNOSTIC DEBUG DATA MODES

T6 Command	Debug Mode	Description
0x01	Page Up	Command Processor T6 debug data page navigation
0x02	Page Down	Command Processor T6 debug data page navigation
0x10	Mutual Capacitance Delta Mode	The Diagnostic Debug T37 object holds signal deltas. These are signed (two's complement) 16-bit mutual capacitance delta values for all nodes. Note that these are the raw deltas. The firmware divides these by 8 before use elsewhere (such as for threshold comparisons).
0x11	Mutual Capacitance Reference Mode	The Diagnostic Debug T37 object holds reference values. These are signed 16-bit capacitance reference values for all nodes.
0x38	DC Data Mode	The Diagnostic Debug T37 object holds the DC level estimates for Self Capacitance Touch data as signed 16-bit values. Note that data values will be set to zero if Self Capacitance Touch is not enabled.
0x80	Device Information Mode	The Diagnostic Debug T37 object holds information on the device; such as, the revision identifier. See Section 2.2.1.7 “Device Information Mode”.
0xF5	Self Capacitance Signal Mode	The Diagnostic Debug T37 object holds the signed 16-bit self capacitance raw signal values.
0xF7	Self Capacitance Delta Mode	The Diagnostic Debug T37 object holds the self capacitance deltas. These are signed (two's complement) 16-bit self capacitance delta values for all nodes.
0xF8	Self Capacitance Reference Mode	The Diagnostic Debug T37 object holds the signed 16-bit self capacitance reference values.
Other value	Object-specific debug mode	See individual objects for details of the debug data available.

Note: Changing the mode resets the page number to zero.

Under NDA
© 2021 Microchip Technology Inc. and its subsidiaries

2.2.1.2 Delta Mode (Mutual Capacitance)

Figure 2-1 shows the organization of the data in the pages for the mutual capacitance delta mode. The deltas are output as signed (two's complement) 16-bit numbers.

The nodes are numbered along the device X lines, that is in the order: element 0 = X0Y0, 1 = X0Y1, 2 = X0Y2 and so on.

NOTE	On the mXT448UD-CCUBHA1 variant device, although up to 640 nodes are available on the matrix, only a maximum of 448 nodes can be used. Therefore, some of the bytes of data will represent unused nodes.

The page number and data index for a given mutual capacitance node's data can be calculated as follows:

Page number = FLOOR (node_number / nodes_per_page)

Data index = bytes_per_node × (node_number MOD nodes_per_page)

where:

- bytes_per_node is 2.
- nodes_per_page is the page size divided by bytes_per_node. In Figure 2-1 this is (128 / 2).

NOTE	If Dual X mode is active, the data for the highest X line measured with Dual X (that is, highest X line used by Multiple Touch Touchscreen T100) is invalid and should be ignored.

FIGURE 2-1: DIAGNOSTIC DEBUG T37 FIELDS – DELTA MODE (MUTUAL CAPACITANCE)

Page	Data Index	Data[]
Page 0	0 – 127	Deltas for nodes 0 to 63
		:
		:
Page 9	1152 – 1279	Deltas for nodes 576 to 639

Assumes page size = 128

2.2.1.3 Reference Mode (Mutual Capacitance)

Figure 2-2 shows the organization of the data in the pages for the mutual capacitance reference mode. The references are output as signed 16-bit numbers.

The nodes are numbered along the device X lines, that is in the order: element 0 = X0Y0, 1 = X0Y1, 2 = X0Y2 and so on.

NOTE	On the mXT448UD-CCUBHA1 variant device, although up to 640 nodes are available on the matrix, only a maximum of 448 nodes can be used. Therefore, some of the bytes of data will represent unused nodes.

The page number and data index for a given mutual capacitance node's data can be calculated as follows:

Page number = FLOOR (node_number / nodes_per_page)

Data index = bytes_per_node × (node_number MOD nodes_per_page)

where:

- bytes_per_node is 2.
- nodes_per_page is the page size divided by bytes_per_node. In Figure 2-2 this is (128 / 2).

NOTE	If Dual X mode is active, the data for the highest X line measured with Dual X (that is, the highest X line used by Multiple Touch Touchscreen T100) is invalid and should be ignored.

FIGURE 2-2: DIAGNOSTIC DEBUG T37 FIELDS – REFERENCE MODE (MUTUAL CAPACITANCE)

Page	Data Index	Data[]
Page 0	0 – 127	References for nodes 0 to 63
		:
		:
Page 9	1152 – 1279	References for nodes 576 to 639

Assumes page size = 128

2.2.1.4 Delta Mode (Self Capacitance)

Figure 2-3 shows the organization of the data in the pages for the self capacitance delta mode. The deltas are output as signed 16-bit numbers.

NOTE	On the mXT448UD-CCUBHA1 variant device, although up to 640 nodes are available on the matrix, only a maximum of 448 nodes can be used. Therefore, some of the bytes of data will represent unused nodes.

FIGURE 2-3: DIAGNOSTIC DEBUG T37 FIELDS – DELTA MODE (SELF CAPACITANCE)

Page	Data Index	Data[]	
Page 0	0 – 39	Touch deltas for Y0 to Y19	
	40 – 103	Touch deltas for X0 to X31	
	104 – 127	Reserved	
Page 1		Reserved; page unused	
Page 2	0 – 39	Single-ended deltas for Y0 to Y19	When Self Capacitance Measurement Configuration T113 CTRL ALTAXISEN = 1
	40 – 103	Single-ended deltas for X0 to X31	When Self Capacitance Measurement Configuration T113 CTRL ALTAXISEN = 0

Assumes page size = 128

Under NDA

© 2021 Microchip Technology Inc. and its subsidiaries

2.2.1.5 Reference Mode (Self Capacitance)

Figure 2-4 shows the organization of the data in the pages for the self capacitance reference mode. The self capacitance touch and single-ended self capacitance references are output as signed 16-bit numbers.

NOTE	On the mXT448UD-CCUBHA1 variant device, although up to 640 nodes are available on the matrix, only a maximum of 448 nodes can be used. Therefore, some of the bytes of data will represent unused nodes.

FIGURE 2-4: DIAGNOSTIC DEBUG T37 FIELDS – REFERENCE MODE (SELF CAPACITANCE)

Page	Data Index	Data[]	
Page 0	0 – 39	Touch references for Y0 to Y19	
	40 – 71	Touch references for even X lines	
	72 – 79	Reserved	
	80 – 111	Touch references for odd X lines	
	112 – 127	Reserved	
Page 1		Reserved; page unused	
Page 2	0 – 39	Single-ended references for Y0 to Y19	When Self Capacitance Measurement Configuration T113 CTRL ALTAXISEN = 1
	40 – 71	Single-ended references for even X lines	When Self Capacitance Measurement Configuration T113 CTRL ALTAXISEN = 0
	72 – 79	Reserved	
	80 – 111	Single-ended references for odd X lines	
	112 – 127	Reserved	

Assumes page size = 128

2.2.1.6 Signal Mode (Self Capacitance)

Figure 2-5 shows the organization of the data in the pages for the self capacitance signal mode. The self capacitance touch and single-ended self capacitance signals are output as signed 16-bit numbers.

NOTE	On the mXT448UD-CCUBHA1 variant device, although up to 640 nodes are available on the matrix, only a maximum of 448 nodes can be used. Therefore, some of the bytes of data will represent unused nodes.

NOTE	If the Driven Shield Signal Limit tests are enabled (see Section 6.4 “Self Test Signal Limits T12 Object”), the Driven Shield test measurements are output at the end of the debug data.

FIGURE 2-5: DIAGNOSTIC DEBUG T37 FIELDS – SIGNAL MODE (SELF CAPACITANCE)

Page	Data Index	Data[]	
Page 0	0 – 39	Touch signals for Y0 to Y19	
	40 – 71	Touch signals for even X lines	
	72 – 79	Reserved	
	80 – 111	Touch signals for odd X lines	
	112 – 127	Reserved	
Page 1		Reserved; page unused	
Page 2	0 – 39	Single-ended signals for Y0 to Y19	When Self Capacitance Measurement Configuration T113 CTRL ALTAXISEN = 1
	40 – 71	Single-ended signals for even X lines	When Self Capacitance Measurement Configuration T113 CTRL ALTAXISEN = 0
	72 – 79	Reserved	
	80 – 111	Single-ended signals for odd X lines	
	112 – 127	Reserved	
Page 3		Reserved; page unused	
Page 4	0 – 1	Raw signal for Driven Shield test measurements – DS0	
	2 – 127	Reserved	

Assumes page size = 128

Under NDA

© 2021 Microchip Technology Inc. and its subsidiaries

2.2.1.7 Device Information Mode

Figure 2-6 shows the organization of the data in the pages for the device information mode.

NOTE The format of the device information mode may vary from device to device.

The following information can be read:

- The Revision ID for the device silicon can be read from the upper 4 bits of Byte 19 on Page 0. On the mXT640UD-CCUBHA1 Family the revision ID is 0x1 (Revision B).

FIGURE 2-6: DIAGNOSTIC DEBUG T37 FIELDS – DEVICE INFORMATION MODE

Page	Data Index	Data[]
Page 0	0 – 18	Reserved
	19	Revision ID (upper 4 bits)
	20 – 127	Reserved

Assumes page size = 128

2.2.1.8 DC Data Mode

Figure 2-7 shows the organization of the data for the DC Data Mode.

FIGURE 2-7: DIAGNOSTIC DEBUG T37 FIELDS – DC DATA MODE

Page	Data Index	Data[]
Page 0	0 – 1	Self Capacitance Touch (SCT) DC Estimate seeded by Single-ended Self Capacitance (SESC) DC Estimate for SCT
	2 – 3	SESC DC Estimate for SCT after filtering and removal of the SESC Reference
	4 – 5	SESC DC Reference for SCT
	6 – 7	Raw SESC DC Estimate for SCT
	8 – 9	The larger of the SCT DC Estimates for X and Y axes
	10 – 11	SCT DC Estimate for the axis processed by Self Capacitance Measurement Configuration T113
	12 – 127	Reserved

Assumes page size = 128

2.2.2 CONFIGURATION

TABLE 2-3: CONFIGURATION FOR DIAGNOSTIC DEBUG T37 (DEBUG_DIAGNOSTIC_T37)

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	MODE	Current mode							
1	PAGE	Current page							
2	DATA[0]	First byte of data on current page							
…									
128	DATA[127]	Last byte of data on current page							

MODE Field

This field contains an indication of the current mode for the data in the DATA[] field. It has the same value as the mode change commands (but not the page change commands) in the DIAGNOSTIC field of the Command Processor T6 object (see “DIAGNOSTIC Field”).

PAGE Field

This field contains the current page number for the data held in the DATA[] field, as controlled by the page change commands in the DIAGNOSTIC field of the Command Processor T6 object (see “DIAGNOSTIC Field”).

DATA[] Fields

This field contains the current page of data.

Under NDA
© 2021 Microchip Technology Inc. and its subsidiaries

3.0 GENERAL OBJECTS

3.1 Introduction

General objects provide global configuration, such as receiving commands and transmitting messages. Table 3-1 lists the general objects on the mXT640UD-CCUBHA1 Family.

TABLE 3-1: GENERAL OBJECTS

Object	Description
Message Processor T5 (GEN_MESSAGEPROCESSOR_T5)	Handles the transmission of messages. This object holds a message in its memory space for the host to read. See Section 3.2 “Message Processor T5 Object”.
Command Processor T6 (GEN_COMMANDPROCESSOR_T6)	Performs a command when written to. Commands include reset, calibrate and backup settings. See Section 3.3 “Command Processor T6 Object”.
Power Configuration T7 (GEN_POWERCONFIG_T7)	Controls the sleep mode of the device. Power consumption can be lowered by controlling the acquisition frequency and the sleep time between acquisitions. See Section 3.4 “Power Configuration T7 Object”.
Acquisition Configuration T8 (GEN_ACQUISITIONCONFIG_T8)	Controls how the device takes each capacitive measurement. See Section 3.5 “Acquisition Configuration T8 Object”.

3.2 Message Processor T5 Object

3.2.1 INTRODUCTION

The purpose of the Message Processor T5 object is to relay the latest status information to the host. This object contains the message data from those objects in the memory map that generate messages (for example, the touch objects and the Command Processor T6 object). A message is generated whenever the status of an object has changed. For this to happen, the object report enable bit must be set.

The system provides an interrupt style mechanism (on both the I^2C and the SPI interface) using the $\overline{\text{CHG_I2C}}$ and $\overline{\text{CHG_SPI}}$ lines to indicate to the host that there is at least one message to read. See Section 3.2.1.1 “Reading Messages with the CHG_I2C and CHG_SPI Lines” for more details.

3.2.1.1 Reading Messages with the $\overline{\text{CHG_I2C}}$ and $\overline{\text{CHG_SPI}}$ Lines

There are two Change interrupt lines on the mXT640UD-CCUBHA1 Family: $\overline{\text{CHG_I2C}}$ for the primary communications interface, and $\overline{\text{CHG_SPI}}$ for the secondary communications interface. These provide two independent interrupt-style interfaces so that one or more hosts can receive messages from the device, as determined by the requirements of the user’s product.

The Message Processor T5 object routes the messages to the primary or secondary interface, as necessary, so that each interface will receive its own messages. In addition, the Message Count T144 object maintains a message count and checksum for each interface independently. Note that the Communications Configuration T18 object can be used to determine which messages are to be routed to which interface (see Section 6.5 “Communications Configuration T18 Object”).

If the device has data to send, it asserts the $\overline{\text{CHG_I2C}}$ or $\overline{\text{CHG_SPI}}$ line (active low), as appropriate, to indicate to the host on that interface that there is a message (note that the $\overline{\text{CHG_SPI}}$ line line is also asserted during SPI transactions to show that the device is ready to send a response). The host should then read the message and use the REPORTID field to determine from which object the message originated.

The host should *always* use the $\overline{\text{CHG_I2C}}$ or $\overline{\text{CHG_SPI}}$ line as an indication that a message is available to read in the Message Processor T5 object. The host should not read the Message Processor T5 object at any other time, such as to poll the device for messages. If the Message Processor T5 object is read when the $\overline{\text{CHG_I2C}}$ or $\overline{\text{CHG_SPI}}$ line is not asserted, an “invalid message” report ID is returned in the Message Processor T5 object.

Multiple messages can easily be read in a continuous read operation using direct memory access (DMA) and message pointer wrapping is implemented to allow this. The Message Count T144 object (see Section 6.21 “Message Count T144 Object”) provides a count of pending messages so that the host driver code knows how many messages to read.

Message pointer wrapping occurs if the address pointer is pointing at either the start of the Message Count T144 object or the start of the Message Processor T5 object (that is, the report ID of the next available message).

NOTE	The Message Count T144 object occurs immediately before the Message Processor T5 object in the memory map so that the message wrapping mechanism will work.

3.2.2 CONFIGURATION

TABLE 3-2: CONFIGURATION FOR MESSAGE PROCESSOR T5 (GEN_MESSAGEPROCESSOR_T5)

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	REPORTID	Report ID for source object							
1 – *n*	MESSAGE	Message data from source object							
n+1	SEQNUM	Sequence number							
n+2	CHECKSUM	Checksum							

Note: *n* = size of largest message possible

NOTE	The size of the MESSAGE field is set to the length of largest message possible. It is dependent on the objects present in the device at a particular revision. The MESSAGE field size should be calculated by subtracting 3 from the size of the Message Processor T5 object retrieved from the Object Table entry (see Section 1.5 “Object Table”). Any unused bytes in a particular message should be treated as reserved bytes.

Under NDA

© 2021 Microchip Technology Inc. and its subsidiaries

REPORTID Field

This field contains the report ID for the message. Messages contain report IDs to allow the host to identify the type of message and its originator. Report IDs are assigned to any object that can send messages. See Section 1.5.6 “Report IDs” for more information on the assignment of report IDs.

MESSAGE Field

This field contains the message data for the object generating the message.

The size of the MESSAGE field is fixed to the size of the message data for the largest object. For compatibility with future firmware updates, this should *always* be calculated by subtracting 3 from the size of the object recorded in the Object Table entry for the Message Processor T5 object (see Section 1.5 “Object Table”).

For information on the contents of the MESSAGE field, see the descriptions for each object elsewhere in this protocol guide.

SEQNUM Field

This field contains the Sequence Number for the message. The sequence number is 0 for the first message to be read by the host after reset. It is then incremented for each subsequent message. When it reaches 255, it wraps around to restart at 0.

NOTE	The mXT640UD-CCUBHA1 Family supports 2 active communications interfaces, so the Message Processor T5 object maintains a sequence number for each interface independently. The Message Processor T5 object routes the messages to the appropriate interface, thus ensuring that each interface will receive its own messages.

CHECKSUM Field

This field contains the 8-bit checksum for the Message Processor T5 object (that is, for the REPORTID, MESSAGE and SEQNUM fields).

See Appendix B “Checksum Calculation” for details on how to calculate the checksum.

3.3 Command Processor T6 Object

3.3.1 INTRODUCTION

The Command Processor T6 object allows commands to be sent to the device. This is done by writing an appropriate value to one of its fields.

3.3.2 CONFIGURATION

TABLE 3-3: CONFIGURATION FOR COMMAND PROCESSOR T6 (GEN_COMMANDPROCESSOR_T6)

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	RESET	Reset							
1	BACKUPNV	Backup settings							
2	CALIBRATE	Calibrate							
3	REPORTALL	Report current status							
4	DEBUGCTRL	Reserved		SIGSEN	REFSEN	DELTASEN	DELTAS8EN	TOUCHES	STATUS
5	DIAGNOSTIC	Diagnostic debug command							
6	DEBUGCTRL2	DBGOBJMODEEN	Reserved						

RESET Field

This field forces a reset of the device if a non-zero value is written. If 0xA5 is written to this field, the device resets itself and enters into bootloader mode.

Write value: Non-zero (normal) or 0xA5 (bootloader)

BACKUPNV Field

This field supports the commands in Table 3-4:

TABLE 3-4: BACKUPNV COMMANDS

Command	Description
0x55	Backs up the settings to the Non-volatile Memory (NVM). When the Dynamic Configuration Controller T70 object is enabled, this command clears the lock on events and resumes the Dynamic Configuration Controller T70 object.
0x33	Stops the Dynamic Configuration Controller T70 object and restores the settings from the NVM.
0x44	Restores the settings from the NVM without stopping the Dynamic Configuration Controller T70 object.
All other values	No effect

Once the device has processed this command it generates a status message containing the new NVM checksum.

CALIBRATE Field

This field performs a global recalibration on all channels (both mutual capacitance and self capacitance). If all the channels are disabled, no message is generated.

If the device is in Deep Sleep mode (see Section 3.4.1 “Introduction”), a message is generated when the device wakes from sleep.

This field is cleared once the command has been processed.

Write value: Non-zero

Under NDA

© 2021 Microchip Technology Inc. and its subsidiaries

REPORTALL Field

This field forces all objects that generate messages to report their current status:

- For optional objects, this applies only if they have their report enable bit set and are currently enabled.
- For objects that are always present and generate messages setting this bit will always cause a status message to be reported.

This field is cleared once the command has been processed.

Write value: Non-zero

DEBUGCTRL Field

This field generates device debugging data using the hardware debug interface.

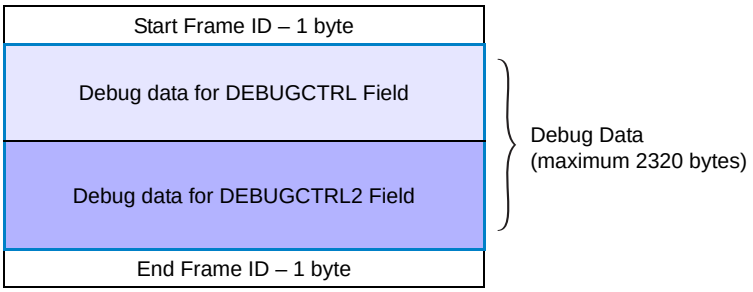
Refer to QTAN0050, *Using the maXTouch Debug Port*, for more details on the hardware debug interface and the format of the debug data.

NOTE	To use the hardware debug interface, the circuit has to be designed accordingly. Refer to the relevant device *Datasheet* for more information.

The output buffer can hold a maximum of 2322 bytes of data on the mXT640UD-CCUBHA1 Family (including two 1-byte Frame IDs).

The data is output in the order of the controls below. If data is also output using the DEBUGCTRL2 field (see "DEBUGCTRL2 Field"), that data is placed in the buffer after the data generated by the DEBUGCTRL field, subject to space being available. Note that two Frame IDs are also output as the first and last bytes of data in the buffer (see Figure 3-1).

FIGURE 3-1: DEBUG DATA BUFFER



If too much debug data has been requested, it will result in partial debug data. Furthermore, as there will be no room for the End Frame ID, there is no match for the Start Frame ID and the data will be considered corrupt.

STATUS: Enables the output of the Command Processor T6 general status byte.

TOUCHES: Enables the output of the touchscreen data (for example, XY positions, states, and so on) for each supported touch.

DELTAS8EN: Enables the output of mutual capacitance signed 8-bit delta values for all nodes.

DELTASEN: Enables the output of mutual capacitance signed 16-bit delta values for all nodes. The format of the data is as for Diagnostic Debug T37 debug data (see Section 2.2.1.2 "Delta Mode (Mutual Capacitance)").

REFSEN: Enables the output of mutual capacitance signed reference values for all nodes. The format of the data is as for Diagnostic Debug T37 debug data (see Section 2.2.1.3 "Reference Mode (Mutual Capacitance)").

SIGSEN: Enables the output of mutual capacitance signed signal values for all nodes.

NOTE	If Dual X mode is active, the data for the highest X line measured with Dual X (that is, the highest X line used) is invalid and should be ignored.

DIAGNOSTIC Field

This field allows commands to be written to control the use of the Diagnostic Debug T37 object (see Section 2.2.1 "Introduction"). Specifically, it allows the pages of diagnostic debug data to be navigated and sets the data mode. This field is cleared once the command has been processed. The host must poll this field to determine that a page change has been actioned and that the requested data is now valid.

The valid commands are listed in Table 3-5.

TABLE 3-5: DIAGNOSTIC DEBUG COMMANDS

Command	Name	Description
0x01	Page Up	Increment page number.
0x02	Page Down	Decrement page number.
Other Value	Debug data mode	The Diagnostic Debug T37 object holds the debug data for a specific debug mode. See Section 2.2 "Diagnostic Debug T37 Object" for details of the system debug modes and the commands to access them. See individual objects for details of the object-specific debug modes and the commands to access them.

Note: Changing the mode resets the page number to zero.

DEBUGCTRL2 Field

DBGOBJMODEEN: Enables object specific debugging data using the hardware debug interface. If this bit is set to 1, other objects can output their specific debug information as required, subject to space being available in the output buffer (see "DEBUGCTRL Field" for details).

Refer to QTAN0050, *Using the maXTouch Debug Port*, for more details on the hardware debug interface and the format of the debug data.

> **NOTE** To use the hardware debug interface, the circuit has to be designed accordingly. Refer to the relevant device *Datasheet* for more information.

The data is output in the order shown in Table 3-6 and is dependent on the data for the particular object being enabled. If an object has its debugging data enabled, the data is output at the location shown. If the debugging data is not enabled for an object, the data for that object is skipped and the data for the next object is output (if enabled).

> **NOTE** All deltas except Single-ended Self Capacitance deltas may be right-shifted (scaled down) by the CTE Configuration T46 DELTASHIFT control (see Section 1.9.1 "Delta Scaling"). All references and signals, however, remain unscaled to allow the true signals in the CTE to be visualized.

TABLE 3-6: ORDER OF DEBUGGING DATA

Object	Debugging Data	Size (Bytes)	Enabled by...
Acquisition Configuration T8	Calibration recovery tuning data. See Section 3.5.4.1 "Calibration Recovery Tuning Data Mode" for format	23	"CFG Field" CALRECTUN
Multiple Touch Touchscreen T100	Mutual capacitance data for all configured nodes: (2)		"CFG2 Field":
	• 8-bit signed deltas	DBGXSIZE × DBGYSIZE	DBGDELTAS8EN
	• 16-bit signed deltas	2 × DBGXSIZE × DBGYSIZE	DBGDELTASEN
	• 16-bit signed references	2 × DBGXSIZE × DBGYSIZE	DBGREFSEN
	• 16-bit signed signals	2 × DBGXSIZE × DBGYSIZE	DBGSIGSEN

Under NDA

© 2021 Microchip Technology Inc. and its subsidiaries

TABLE 3-6: ORDER OF DEBUGGING DATA (Continued)

Object	Debugging Data	Size (Bytes)	Enabled by...
Self Capacitance Configuration T111	Self Capacitance Deltas: • Raw 16-bit signed Touch deltas • 16-bit signed Touch deltas • 16-bit signed Single-ended deltas	 120 120 120	Instance 0 "DBGCTRL Field": RAWDATAEN DELTASEN Instance 1 "DBGCTRL Field": DELTASEN
	Self Capacitance References: • Raw 16-bit signed Touch references • 16-bit signed Touch references • 16-bit signed Single-ended references	 120 120 120	Instance 0 "DBGCTRL Field": RAWDATAEN REFSEN Instance 1 "DBGCTRL Field": REFSEN
	Self Capacitance Signals: • Raw 16-bit signed Touch signals • 16-bit signed Touch signals • 16-bit signed Single-ended signals	 120 120 120	Instance 0 "DBGCTRL Field": RAWDATAEN SIGSEN Instance 1 "DBGCTRL Field": SIGSEN
Self Capacitance Configuration T111	DC debug data: • Self Capacitance Touch measurements	 12	 Instance 0 "DBGCTRL Field": DCEN
Self Capacitance Grip Suppression T112	Grip suppression debug data	23	"CTRL Field" DBGEN
Glove Detection T78	Discriminant for Instance 0	1	"CTRL Field" DBGDISCRIMEN
Key Array T15	Debugging data for all defined keys (zero-padded to 64 bytes): • 16-bit signed deltas • 16-bit signed references • 16-bit signed signals	 64 64 64	 "CFG Field"DBGDELTASEN "CFG Field"DBGREFSEN "CFG Field"DBGSIGSEN

Note 1: Data is output only if debug data is enabled within the object, otherwise the object's data is skipped.
2: If Dual X mode is in use and the debug data includes the top X line covered by Multiple Touch Touchscreen T100, the data for the highest X line is invalid and should be ignored.

3.3.3 MESSAGES

The message data for the Command Processor T6 object is shown in Table 3-7.

TABLE 3-7: MESSAGE DATA FOR COMMAND PROCESSOR T6 (GEN_COMMANDPROCESSOR_T6)

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	STATUS	RESET	OFL	SIGERR	CAL	CFGERR	COMSERR	Reserved	
2 – 4	CHECKSUM	Configuration settings checksum							

STATUS Field

Reports the current status and flags errors. A bit is set to indicate the corresponding status/error. Note that there may be more than one status/error reported.

CFGERR, CAL, SIGERR and OFL report ongoing status and error conditions, so once these status/error conditions have terminated, a further message is sent with the appropriate bit cleared.

COMSERR and RESET are one-off reports indicating already terminated conditions. These error conditions do not generate a further message with a cleared bit.

COMSERR: There is an error with the communications checksum. This error bit is set when there has been a checksum error on the bytes that have been written to the device.

NOTE	If there is a checksum error after a write, the device will ignore the data. Nevertheless, the internal (expected) sequence number will still be incremented.

CFGERR: There is a configuration error in one or more of the enabled objects. The device pauses its processing and generates a status message every 200 ms.

See Section 1.7 “Configuration Checks” for more information on configuration checks.

NOTE	The device will stop scanning for touches while the error persists. It is possible to make configuration changes and to execute a backup command while the device is in this error state.

CAL: The device is calibrating. Note that CAL messages may not be generated if other objects (for example, Acquisition Configuration T8) disable these.

SIGERR: There was an error in the acquisition. This error should not normally be seen.

OFL: The acquisition and processing cycle length (see “IDLEACQINT and ACTVACQINT Fields” and “IDLEACQINT and ACTVACQINT Fields”) has overflowed the desired power mode interval. The OFL flag is not updated in Free-run or Deep Sleep modes. This message can be suppressed by setting the OVFRPTSUP bit in the Power Configuration T7 CFG field (see Section 3.4.2 “Configuration”).

RESET: The device has reset.

CHECKSUM Field

Reports the checksum of the configuration settings held in the non-volatile memory. See Appendix B “Checksum Calculation” for details on how to calculate the checksum.

Under NDA

© 2021 Microchip Technology Inc. and its subsidiaries

3.4 Power Configuration T7 Object

3.4.1 INTRODUCTION

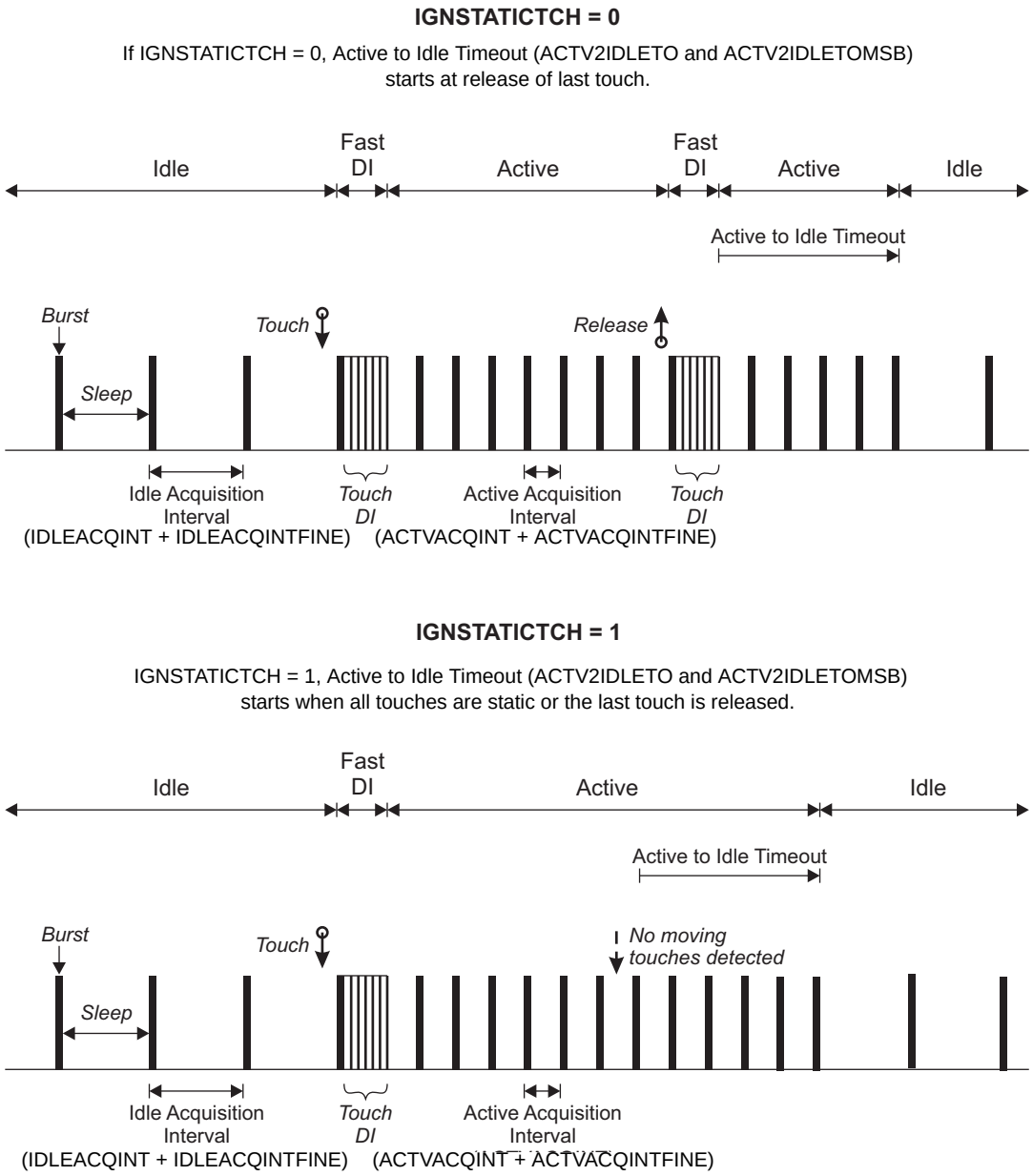
The Power Configuration T7 object controls the active and idle (sleep) times of the device. Power consumption can be lowered by controlling the acquisition frequency and sleep times between measurements.

The device operates in two modes: active (touch detected) and idle (no touches detected).

The normal state for the device is idle mode. In this mode the device operates in a series of long burst cycles. Each cycle consists of a short burst (during which measurements are taken to detect a possible touch) followed by an inactive sleep period.

Figure 3-2 shows the power modes for a Multiple Touch Touchscreen T100 object.

FIGURE 3-2: POWER MODE FIELDS – MULTIPLE TOUCH TOUCHSCREEN T100 OBJECT



When the user touches a touchscreen, the device enters a Fast Detect Integration (Fast DI) mode (see the TCHDI fields in the touch objects for an explanation of detect integration). This consists of a series of quick, short bursts to confirm that a change in the touch state has indeed occurred. The number of bursts is determined by the TCHDI fields in the touch objects. If it is a genuine touch, the device enters active mode. In this mode the device operates in a series of burst cycles that intersperse measurement bursts with very short sleep periods. These sleep periods are typically shorter than those in idle mode.

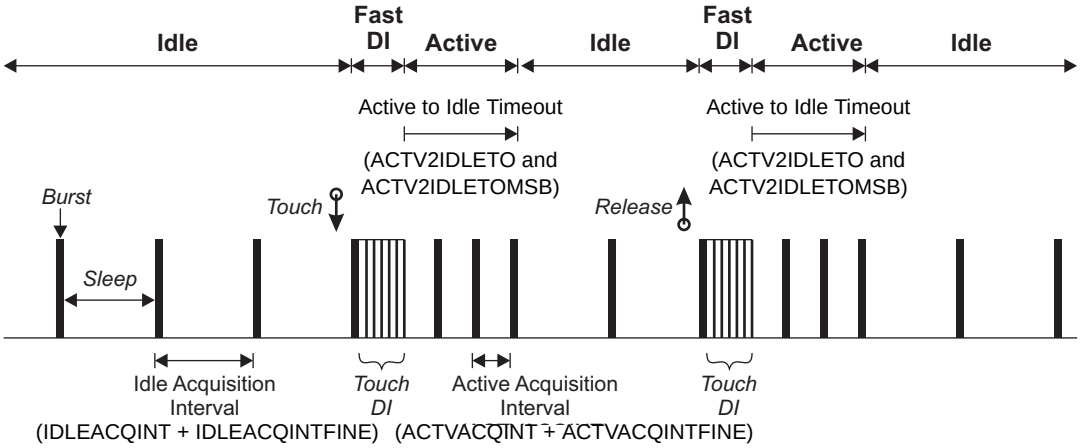
If IGNSTATICTCH = 0, when the user releases the touch, the device again enters a short Fast DI mode to confirm that a change in the touch state has occurred. If there are other touches still on the screen, the device will stay in active mode. If this was the last touch, the device will stay in active mode for a short timeout period (Active to Idle Timeout), and drop to idle mode after this timeout has expired.

If IGNSTATICTCH = 1, when all touches have been static for a short timeout period (Active to Idle Timeout), the device will switch over to idle mode. If the timeout is not reset by a moving or a new touch then, when the user releases the static touch, the device again enters a short Fast DI mode to confirm that a change in the touch state has occurred, immediately followed by the idle mode.

For a Key Array T15 object, the active to idle timeout (and the subsequent Idle mode) applies on a touchdown as well as a release, as in Figure 3-3.

NOTE	The changes to the cycle time happen regardless of whether the touch object is reporting or not.

FIGURE 3-3: POWER MODE FIELDS – KEY ARRAY T15 OBJECT



The Power Configuration T7 object also controls whether the acquisition and subsequent processing are pipelined or not. The normal sequence of events during an acquisition cycle is that data is acquired, the data is processed and then the device sleeps for the remainder of the cycle. When data acquiring is pipelined with the data processing, the processing is postponed until such time as the next data acquisition is performed. This means that the processing of one acquisition interval is done in parallel with the acquisition of data for the next acquisition interval. This can significantly increase the throughput of data when in Free-run mode.

Pipelining of the active and idle modes is controlled separately by the ACTVPIPEEN and IDLEPIPEEN bits in the CFG field.

Under NDA
© 2021 Microchip Technology Inc. and its subsidiaries

3.4.2 CONFIGURATION

TABLE 3-8: CONFIGURATION FOR POWER CONFIGURATION T7 (GEN_POWERCONFIG_T7)

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	IDLEACQINT	Idle acquisition interval							
1	ACTVACQINT	Active acquisition interval							
2	ACTV2IDLETO	Active to idle time out							
3	CFG	INITACTV	OVFRPTSUP	ACTV2IDLETOMSB				ACTVPIPEEN	IDLEPIPEEN
4	CFG2	Reserved					POWMONMODE	DISPOWMON	IGNSTATICTCH
5	IDLEACQINTFINE	Fine idle acquisition interval							
6	ACTVACQINTFINE	Fine active acquisition interval							

IDLEACQINT and ACTVACQINT Fields

The length of the idle and active burst cycles is determined by the Idle Acquisition Interval (IDLEACQINT) and the Active Acquisition Interval (ACTVACQINT) fields respectively in combination with the IDLEACQINTFINE and ACTVACQINTFINE fields respectively (see "IDLEACQINTFINE and ACTVACQINTFINE Fields"). The IDLEACQINT/ACTVACQINT field specifies the millisecond part of the interval and the IDLEACQINTFINE/ACTVACQINTFINE field specifies the microsecond part. See Figure 3-2 and Figure 3-3).

A setting of 255 in IDLEACQINT/ACTVACQINT forces the device to enter Free-run mode the next time that the appropriate mode (idle or active) is entered. In Free-run mode the device does not sleep between acquisitions. This gives the fastest response time at the expense of increased power consumption.

A setting of zero forces the device to enter Deep Sleep mode the next time that the appropriate mode (idle or active) is entered. The device remains in Deep Sleep mode until the IDLEACQINT or ACTVACQINT setting is restored. Deep Sleep mode is used to minimize power consumption if the device does not need to be sensing. If Deep Sleep mode is requested, it is advisable to set both IDLEACQINT and ACTVACQINT to zero to avoid indeterminate behavior if one mode is still active (note that, in this case, IDLEACQINTFINE and ACTVACQINTFINE will be ignored). The status flags in the Command Processor T6 (see Section 3.3.2 "Configuration") are not updated when the device is in Deep Sleep mode.

Other values for IDLEACQINT or ACTVACQINT specify the millisecond part of the Idle or Active Acquisition Interval, which is added to IDLEACQINTFINE or ACTVACQINTFINE, as appropriate, to form the actual Idle or Active Acquisition Interval that is used. A high value causes more sleep time between acquisitions. This results in lower power consumption but a slower response time.

The Idle and Active Acquisition intervals must not be set to less than the actual burst time. The device is also designed to sleep as much as possible in order to conserve power. The Idle Acquisition Interval should therefore be set longer than Active Acquisition Interval. Under some circumstances it is desirable to set Idle Acquisition Interval lower than Active Acquisition Interval. For example, this might be necessary to minimize the difference between the best-case and the worst-case touchdown latency.

Range: 0 (Deep Sleep), 1 to 254, 255 (Free-run)

IDLEACQINT Typical: 32

ACTVACQINT Typical: 10

ACTV2IDLETO Field and CFG ACTV2IDLETOMSB Bits

The device automatically goes into idle mode whenever possible after each matrix scan to conserve power, unless a touch object is being touched (see Figure 3-2 on page 29).

The device does not go into idle mode immediately. Instead there is a timeout period. The device runs in active mode during this timeout period to allow further touches to keep the device active. The timeout period is determined by the Active to Idle Timeout setting. This is a 12-bit value with the 4 MSBits in the ACTV2IDLETOMSB bits of the CFG field and the 8 LSBits in the ACTV2IDLETO field.

Under normal operation, the device enters idle mode after the expiry of the Active to Idle Timeout and then remains in idle mode until the next touch is detected. If there is more than one touch present, the Active to Idle Timeout applies only after the last touch has been released. This means that once the device has been awakened by a change, the touch response time is fast for as long as the sensor remains in use. Once channel activity lapses for a period longer than the Active to Idle Timeout, the device returns to idle mode.

The Active to Idle Timeout is specified in 200 ms increments, where 0 means 1 cycle.

Range: 0 (1 cycle), 1 to 4095

Typical: 50

CFG Field

IDLEPIPEEN: Enables pipelining in idle mode. Pipelining is enabled if set to 1 and disabled if set to 0.

ACTVPIPEEN: Enables pipelining in active mode. Pipelining is enabled if set to 1 and disabled if set to 0.

ACTV2IDLETOMSB: Specifies the four MSBits of the Active to Idle Timeout. See "ACTV2IDLETO Field and CFG ACTV2IDLETOMSB Bits" for details].

OVFRPTSUP: If this bit is set, overflow messages are suppressed (and a pending overflow message will be cleared).

INITACTV: Selects whether the chip starts in active mode or idle mode after a power-up or reset. If set to 1 then the chip starts in active mode and remains in active mode until the Active to Idle Timeout expires (as set by the ACTV2IDLETO Field and the CFG ACTV2IDLETOMSB bits). If INITACTV is set to 0, then the device starts in idle mode.

CFG2 Field

IGNSTATICTCH: Under normal operation (that is, when this bit is set to 0), the Active to Idle Timeout is refreshed while any touch is active (either moving or static). This means that a static touch will keep the object in active mode and the Active to Idle Timeout expires only after the last touch has been released (see Figure 3-2 on page 29).

If this bit is set to 1, then static touches do not cause the Active to Idle Timeout to reset and the timeout expires once all tracked touches are either static or released (see Figure 3-2 on page 29).

This feature trades off subsequent touch up/down latency (but not first touch down latency) against power saving.

DISPOWMON: Disables Hardware Power Monitoring.

NOTE	Any change to this control will be effective only after a backup and reset.

By default (DISPOWMON = 0) hardware power monitoring is enabled. If this control is set to 1, hardware power monitoring is disabled. This can help reduce current consumption for low power designs. However, disabling power monitoring will significantly reduce the robustness of the device to withstand ESD and power rail brown out events.

Note that the Dynamic Configuration Controller T70 Power OK event (Event Enumeration 310) will not be available if hardware power monitoring has been disabled.

IMPORTANT!	Disabling the power monitors (DISPOWMON = 1) has an impact on Self Test Control T10 processing. In particular: • The Self Test Control T10 Power tests require the power monitors to be enabled and may not operate correctly if the power monitors are disabled. The power monitors should therefore be left enabled (DISPOWMON = 0) if power self tests are required for compliance with UL/IEC 60730 Class B. • If the hardware power monitors are disabled, Self Test Control T10 Signal Limit tests may be compromised because measurements might be corrupted in the event of a power loss that the system is unable to detect. If the Signal Limit tests report a failure, it should be verified that a power failure was not the cause of the failure. This can be done, for example, by repeating the Signal Limit tests. Alternatively, it is recommended that on-demand Self Test Control T10 Power test is run before running any Signal Limit tests.

POWMONMODE: Sets the Power Monitor mode for use when hardware power monitoring is enabled (that is, DISPOWMON = 0).

The default mode (POWMONMODE = 0) is Power Monitoring Sampling mode. This mode causes the power monitors to perform power checks at 1 KHz frequency, and it is recommended that this mode is selected to minimize power consumption.

Any change to this control will be effective only after a Backup and Reset.

Under NDA
© 2021 Microchip Technology Inc. and its subsidiaries

POWMONMODE Range: See Table 3-9

TABLE 3-9: POWMONMODE

Value	Mode
0 (default)	Power Monitoring Sampling mode
1	Power Monitoring Continuous mode

IDLEACQINTFINE and ACTVACQINTFINE Fields

These fields fine tune the Idle and Active Acquisition Interval. They specify an additional interval (in units of 10 µs) that is added to the values specified by the IDLEACQINT and ACTVACQINT fields to achieve a more exact refresh rate.

Range: 0 to 255 (effective range: 0 to 99)

3.5 Acquisition Configuration T8 Object

3.5.1 INTRODUCTION

The Acquisition Configuration T8 object controls how the device takes mutual capacitive measurements.

3.5.1.1 Calibration Recovery Process

Figure 3-4 and subsequent figures show an overview of the recalibration processes on the mXT640UD-CCUBHA1 Family. The mutual capacitance recovery process is controlled by the Acquisition Configuration T8 object and the self capacitance recovery process is controlled by the Self Capacitance Configuration T111 object (see Section 6.16 “Self Capacitance Configuration T111 Object”).

FIGURE 3-4: RECALIBRATION PROCESSES

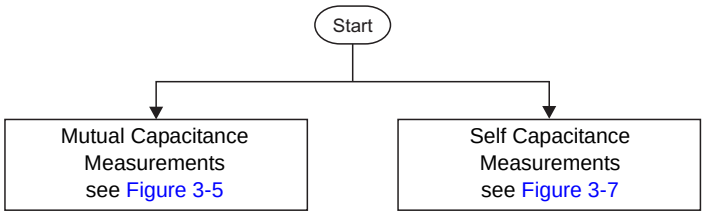
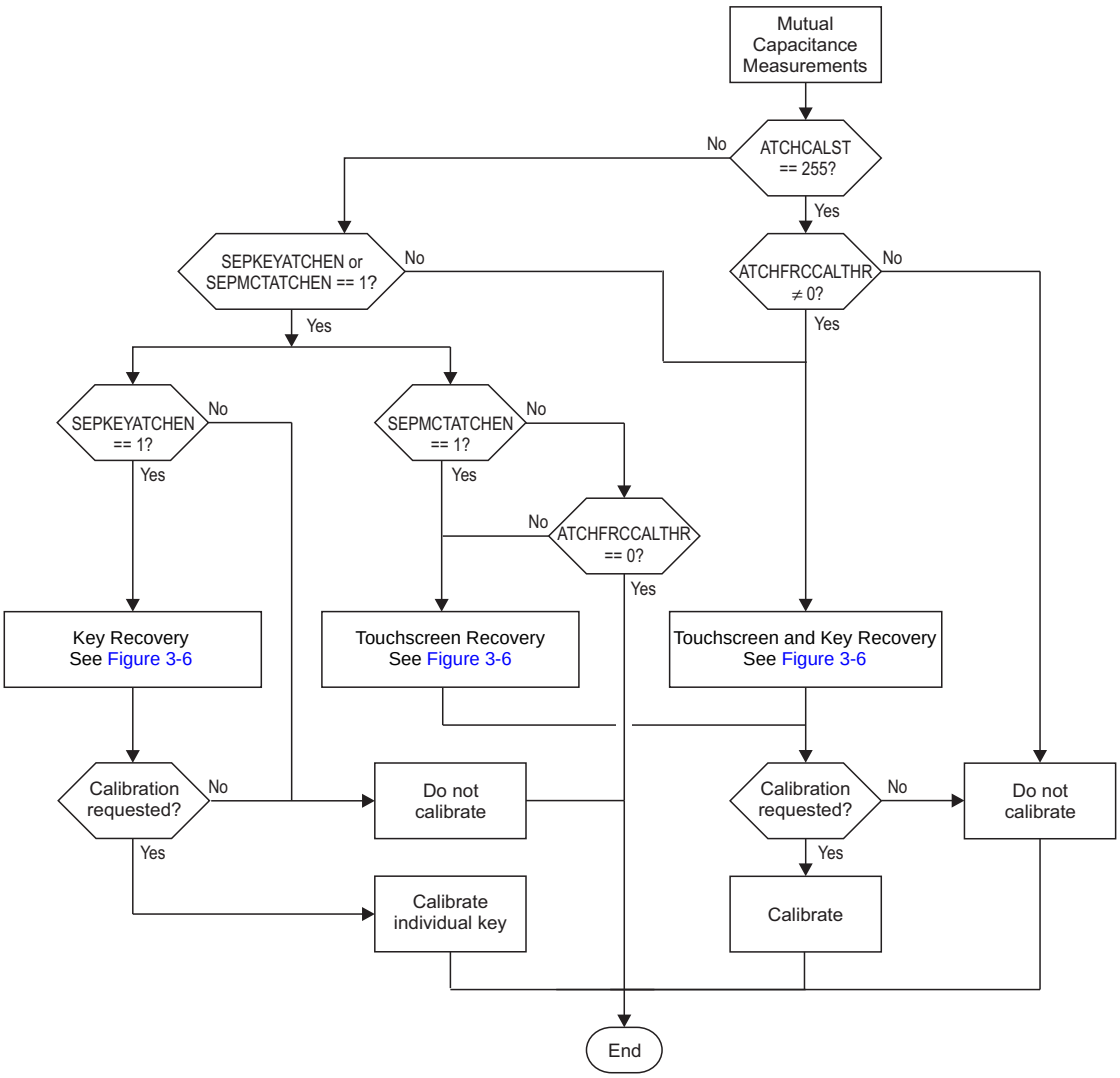
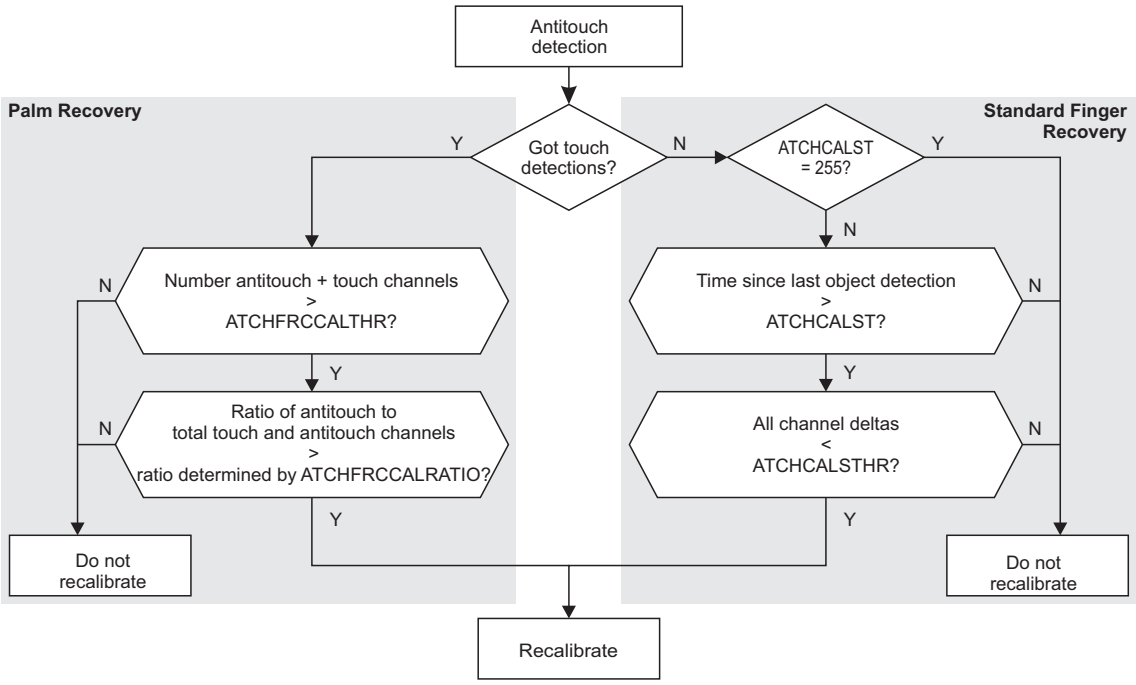


FIGURE 3-5: RECALIBRATION PROCESS – MUTUAL CAPACITANCE



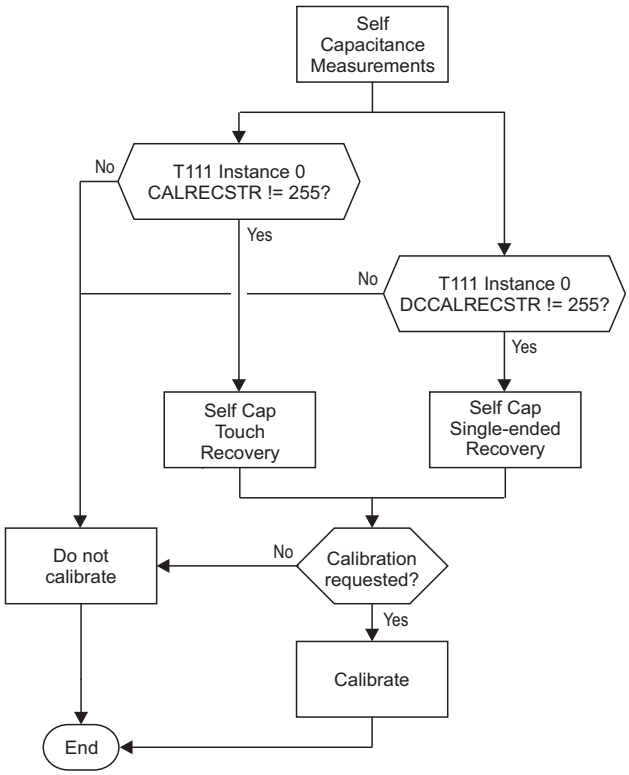
Under NDA
© 2021 Microchip Technology Inc. and its subsidiaries

FIGURE 3-6: RECALIBRATION PROCESS – TOUCH RECOVERY



Note: The antitouch detection and the number of antitouch channels are affected by the Multiple Touch Touchscreen T100 CFG1 ATCHTHRSEL control (see "CFG1 Field").

FIGURE 3-7: RECALIBRATION PROCESS – SELF CAPACITANCE



The standard finger mutual capacitance recovery process (see Figure 3-6) allows the sensor to recover when a finger is present on the sensor during calibration and is then subsequently removed. This process attempts to calibrate the sensor when only antitouches are detected on the sensor. The ATCHCALST and ATCHCALSTHR fields allow this process to be blocked under certain circumstances (see "ATCHCALST Field" and "ATCHCALSTHR Field").

In addition, the Acquisition Configuration T8 object provides further control over how individual keys are calibrated. See "CFG Field" for more information on the CFG field SEPKEYATCHEN and SEPMCTATCHEN bits.

A palm touch during calibration results in a complex pattern of both touch and antitouch detections. The standard finger recovery recalibration process, however, is blocked by any channels in touch, so a recalibration would never occur. The mutual capacitance palm recovery process therefore allows the sensor to recover when a palm is present on the sensor during calibration and is then subsequently removed. Palm recovery process is controlled by the ATCHFRCCALTHR and ATCHFRCCALRATIO fields (see "ATCHFRCCALTHR and ATCHFRCCALRATIO Fields").

3.5.2 CONFIGURATION

TABLE 3-10: CONFIGURATION FOR ACQUISITION CONFIGURATION T8 (GEN_ACQUISITIONCONFIG_T8)

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	CHRGTIME	Charge-transfer dwell time							
1	Reserved	Reserved							
2	TCHDRIFT	Touch drift							
3	DRIFTST	Drift suspend time							
4	TCHAUTOCAL	Touch automatic calibration							
5	SYNC	Reserved					DISRISE	Reserved	ENABLE
6	ATCHCALST	Antitouch calibration suspend time							
7	ATCHCALSTHR	Antitouch calibration suspend threshold							
8	ATCHFRCCALTHR	Antitouch force calibration threshold							
9	ATCHFRCCALRATIO	Antitouch force calibration ratio							
10	MEASALLOW	Reserved				SELFPROX	Reserved	SELFTCH	MUTUALTCH
11	Reserved	Reserved							
12	Reserved	Reserved							
13	Reserved	Reserved							
14	CFG	CALRECTUN	Reserved				SEPMCTATCHEN	SEPKEYATCHEN	DISCALRPT

CHRGTIME Field

This setting controls the charge-transfer dwell time. It is specified in units of 41.67 ns (24 MHz clock cycles). Higher charge times result in a slower scan time. The maximum for this field is 255 (10.625 µs); however, for practical purposes 5 µs should be considered the recommended maximum. The default value of 0 means 48 (2 µs).

Range: 0 (48), 1 to 255

TCHDRIFT Field

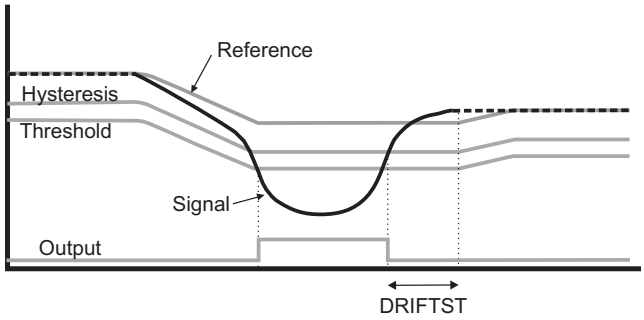
The Touch Drift (TCHDRIFT) setting controls the drift interval for mutual capacitance measurements on the touchscreen array. The drift intervals for any other measurement types (if present) are configured in other objects.

Signals can drift because of changes in the nature of the components and materials over time and temperature. It is crucial that such drift is compensated for, otherwise false detections and sensitivity shifts can occur.

Drift compensation (see Figure 3-8) is performed by making the reference level track the raw signal at a slow rate, but only while there is no touch detection in effect. The rate of adjustment must be slow, otherwise legitimate detections could be ignored. The TCHDRIFT field can be configured in increments of 200 ms, where a value of 0 means 10 (2 seconds).

Under NDA
© 2021 Microchip Technology Inc. and its subsidiaries

FIGURE 3-8: THRESHOLDS AND DRIFT COMPENSATION



The device compensates for drift by using a slew-rate limited change to the reference level. The threshold and hysteresis values are tied to this reference. Any changes to TCHDRIFT affect the rate. The drift compensation can drift at a rate of up to ±16 16-bit delta values once every (TCHDRIFT × 200) ms.

Once the touch signal has crossed the threshold for the current touch classification (see Section A.3 “Touch Classification”) on the appropriate touch object, the drift compensation mechanism is suspended.

Range: 0 (10), 1 to 254, 255 (disabled)

Typical: 20

DRIFTST Field

The Drift Suspend Time (DRIFTST) setting for mutual capacitance measurements controls the time from a touch release on a touch object until the drift process is re-enabled. DRIFTST is used to restrict drift on all channels while one or more mutual-capacitance channels are activated. It defines the length of time the drift is halted after a touch detection.

This feature is particularly useful in preventing an actual touch – or simply a hovering finger – from causing untouched channels to drift. Without this feature, a sensitivity shift could be created that would ultimately inhibit any further touch detection.

DRIFTST can be configured to a value of between 0 and 255 in increments of 200 ms, where a value of zero means 20 (4 seconds). This gives a range of 200 ms to 51s.

Range: 0 (20), 1 to 255

TCHAUTOCAL Field

A prolonged (usually unintentional) contact from a foreign object may result in a mutual capacitance touch detection for a prolonged interval. It is desirable to perform a recalibration in order to restore a touch object’s function. This is usually done after a time delay of some seconds.

The Touch Automatic Calibration (TCHAUTOCAL) setting controls the length of time a touch is held until it is considered false and an automatic recalibration is performed to compensate.

The TCHAUTOCAL timer monitors touch detections. If a detection event exceeds the timer setting, an automatic recalibration occurs. After the recalibration has taken place, normal functionality resumes, even if the touch object is still being contacted by the foreign object. This feature is enabled globally, but the exact mechanism depends on the object being touched:

- For a Multiple Touch Touchscreen T100 object (see Section 4.3 “Multiple Touch Touchscreen T100 Object”) there is a counter per touch, incremented while the touches remain stationary. If a touch is stationary throughout the TCHAUTOCAL period, an automatic recalibration occurs. A touch is considered stationary if it moves less than 1 node (in either axis) within the TCHAUTOCAL period. Note that the entire sensor matrix is calibrated, and not just the area occupied by the Multiple Touch Touchscreen T100 object that triggered the recalibration.
- For a Key Array T15 object (see Section 4.2 “Key Array T15 Object”) there is a counter per key, incremented while a touch remains on the key. An automatic recalibration occurs if the touch is still present at the end of the TCHAUTOCAL period. Note that a touch detection within a key does not clear the reset counters for any other keys, or the reset counters for any other touch objects.

NOTE	Recalibration of a Key Array T15 does not calibrate the entire sensor matrix, only that part occupied by touched keys. Recalibration of the Multiple Touch Touchscreen T100, however, does calibrate the entire sensor matrix.

TCHAUTOCAL can be disabled by setting it to zero (infinite timeout). In this case the object never automatically recalibrates during a continuous detection (but the host could still command it). TCHAUTOCAL above 0 is configured in 200 ms increments.

NOTE	TCHAUTOCAL should not be used if Retransmission Compensation T80 is enabled.

Range: 0 (infinite), 1 to 255

SYNC Field

IMPORTANT!	Frame Synchronization functionality on the mXT640UD-CCUBHA1 Family is not considered UL/IEC 60730 Class B compliant. Therefore, the device should not be configured to use Frame Synchronization in a UL/IEC 60730 Class B product.

NOTE	The SYNC line can be used for either frame synchronization or pulse synchronization, but not both. If the SYNC line is used for frame synchronization functionality, then it cannot be used for pulse synchronization. See Section 6.8.1.1 “Pulse Synchronization” and Section 6.16.1.3 “Pulse Synchronization” for more information on pulse synchronization.

This field enables frame synchronization. This forces the device to synchronize with the SYNC pin when performing a measurement. In this case, the device waits for an edge on the SYNC pin before performing a measurement. If there is no edge, the device will not acquire. This feature overrules the settings in the Power Configuration T7 object (see Section 3.4 “Power Configuration T7 Object”). If this feature is disabled, the device will run as configured by the Power Configuration T7 object.

If synchronization edges occur more frequently than the acquisition cycles, the device triggers on the next synchronization edge that occurs after the current acquisition cycle.

Frame synchronization can be enabled and disabled at runtime. If the device is sleeping between acquisition cycles, this takes effect immediately. If an acquisition cycle is ongoing, the frame synchronization will be toggled on or off at the end of the current acquisition cycle. Any synchronization edge that occurs during the current cycle, will be acted on or ignored according to the current synchronization state. Similarly, messages may still be generated for this cycle according to the current synchronization state.

NOTE	The frame synchronization function shares a pin with the GPIO0 function. If the frame synchronization function is enabled, then this takes precedence over any specified GPIO0 function in the GPIO Configuration T19 object (see Section 6.6 “GPIO Configuration T19 Object”).

The use of the SYNC pin is controlled by two bits. The ENABLE bit enables the use of the SYNC pin, with frame synchronization occurring on the rising edge only by default. The DISRISE bit disables synchronization with the rising edges and switches the synchronization to be on the falling edge only. This means that to enable the SYNC pin for use, and synchronize with the rising edge, the ENABLE bit is set to 1 and the DISRISE bit is left at the default zero value.

ENABLE: The SYNC pin is enabled if set to 1, and disabled if set to 0.

DISRISE: Disables frame synchronization with the rising edge (and enables synchronization with the falling edge). Synchronization with the rising edge is disabled if DISRISE is set to 1, and enabled if set to 0.

The values for these controls are listed in Table 3-11.

Under NDA
© 2021 Microchip Technology Inc. and its subsidiaries

TABLE 3-11: SYNC FIELD VALUES

Bit		
DISRISE	**ENABLE**	**Effect**
X	0	Disabled (default); DISRISE has no effect
0	1	Enabled: synchronized with rising edge
1	1	Enabled: rising edge disabled; synchronized with falling edge

Note: X = indeterminate value (0 or 1)

ATCHCALST Field

The ATCHCALST field sets the antitouch calibration suspend time for use in the touch recovery process (see Section 3.5.1.1 “Calibration Recovery Process”). This is the time from the last touch release to when a standard finger recovery recalibration can occur. ATCHCALST can be configured to a value of between 0 and 254 (0 and 50.8s), in increments of 200 ms. A setting of zero allows the recalibration logic to work immediately after the last finger is removed from the sensor. A value of 255 disables the mutual capacitance finger recovery process.

If the device is configured to use mutual capacitance measurements only, it is recommended to set ATCHCALST to zero, as other values can affect the ability of the device to recover from a bad calibration.

Note that the antitouch calibration suspend time should not be set to a period longer than the drift suspend time (DRIFTST).

NOTE	If the device is configured to use mutual capacitance measurements only (that is, MEASALLOW MUTUALTCH = 1), the Antitouch Calibration Suspend Time feature should not be used if the Retransmission Compensation T80 object is enabled. ATCHCALST should therefore be set to 255 if the Retransmission Compensation T80 object is enabled.

Range: 0 to 254, 255 (disabled)

Typical: 0 (immediate)

ATCHCALSTHR Field

The ATCHCALSTHR field sets the antitouch calibration suspend threshold. Any channel with a touch delta above this threshold will suspend all standard finger recovery recalibrations. If this field is set to zero, the standard finger recovery calibration is never blocked except by the ATCHCALST period.

Note that if the ATCHCALST field is set to a value of 255, the mutual capacitance finger recovery process is disabled and the ATCHCALSTHR field has no effect.

See also Section 3.5.1.1 “Calibration Recovery Process”.

Range: 0 to 255

Typical: 0 (never suspend)

ATCHFRCCALTHR and ATCHFRCCALRATIO Fields

These two fields control the mutual capacitance palm recovery process that allows the sensor to recover when a palm is present on the sensor during calibration and is then subsequently removed.

Note that this field controls the palm recovery process for mutual capacitance measurements; for the calibration recovery process for self capacitance measurements, see Section 6.16 “Self Capacitance Configuration T111 Object”

A palm touch during calibration results in a complex pattern of both touch and antitouch detections. A touch channel is one that is above +TCHTHR (touch threshold); an antitouch channel is one that is below either –TCHTHR or –INTTHR, as determined by the ATCHTHRSEL control in the Multiple Touch Touchscreen T100 “CFG1 Field”.

The standard finger recovery recalibration process (see “ATCHCALSTHR Field” and “ATCHCALST Field”), however, is blocked by any channels in touch, so a recalibration would never occur. These two fields ensure that this block is overridden and that a palm recovery calibration occurs (see Figure 3-6 on page 35).

The ATCHFRCCALTHR field sets the threshold (in channels) to allow a palm recovery calibration. If the number of channels in touch or antitouch is greater than or equal to ATCHFRCCALTHR, then the ATCHFRCCALRATIO field is used to determine if a recalibration should occur.

Setting the ATCHFRCCALTHR field to zero disables both that field and the ATCHFRCCALRATIO field and a recalibration will not occur.

The ATCHFRCCALRATIO field determines the ratio of antitouch channels to total touch and antitouch channels that must be met for a recalibration to occur. This field takes a signed value that represents the desired ratio (see Table 3-12). Note that negative values should be avoided as they risk rogue calibrations when a palm is placed over the sensor.

TABLE 3-12: TYPICAL VALUES FOR ATCHFRCCALRATIO

Value	Meaning
-128 to 0	Reserved; only use a value of zero or less if advised to do so by Microchip.
+1 to +127	Antitouch channels must be greater than 50 percent total (touch + antitouch) channels (where 127 = 100 percent total channels)

ATCHFRCCALTHR Range: 0 (mutual capacitance palm recovery disabled), 1 to 255

ATCHFRCCALTHR Typical: 50

ATCHFRCCALRATIO Range: –128 to +127

ATCHFRCCALRATIO Typical: 25 (60 percent)

MEASALLOW Field

This field defines which measurement types are allowed in the user's product. One or more types of touches can be set using the bits in this field. A value of 0 means 1; that is, mutual capacitance touches are always allowed.

MUTUALTCH: If this bit is set to 1, mutual capacitance touches are allowed.

NOTE	Mutual capacitance touches are always allowed, so setting this bit to 0 has the same effect as setting it to 1.

SELFTCH: If this bit is set to 1, self capacitance touches are allowed.

SELFPROX: If this bit is set to 1, self capacitance single-ended measurements are allowed for use by the Self Capacitance Measurement Configuration T113 object. Note that if SELFPROX is enabled, then the SELFTCH bit must also be enabled.

CFG Field

DISCALRPT: Disables reporting of Acquisition Configuration T8 auto calibration messages. If this bit is set to 1, generation of a Command Processor T6 message on a recalibration event that is generated by the touch or antitouch mechanisms of the Acquisition Configuration T8 object will be suppressed. The recalibration itself will still happen, but it will not be flagged.

SEPKEYATCHEN: Enables Separate Key Antitouch Calibration. The effect of this field depends on ATCHCALST (see "ATCHCALST Field"). If ATCHCALST is set to a value less than 255, the effect of SEPKEYATCHEN is as shown in Table 3-13. If ATCHCALST is set to 255, mutual capacitance finger recovery is disabled and SEPKEYATCHEN has no effect (although palm recovery still applies).

SEPMCTATCHEN: Enables Separate Mutual Capacitance Antitouch Calibration. The effect of this field depends on ATCHCALST (see "ATCHCALST Field"). If ATCHCALST is set to a value less than 255, the effect of SEPMCTATCHEN is as shown in Table 3-13. If ATCHCALST is set to 255, mutual capacitance finger recovery is disabled and SEPMCTATCHEN has no effect (although palm recovery still applies).

TABLE 3-13: SEPMCTATCHEN AND SEPKEYATCHEN (ATCHCALST < 255)

SEPMCTATCHEN	SEPKEYATCHEN	Effect on Touchscreen	Effect on Keys †
0	0	Mutual calibration takes touchscreen and keys into account equally.	Keys are calibrated with the touchscreen only. *

Under NDA
© 2021 Microchip Technology Inc. and its subsidiaries

TABLE 3-13: SEPMCTATCHEN AND SEPKEYATCHEN (ATCHCALST < 255) (CONTINUED)

SEPMCTATCHEN	SEPKEYATCHEN	Effect on Touchscreen	Effect on Keys †
0	1	Mutual capacitance touchscreen finger recovery is disabled.	Keys calibrate individually without causing a touchscreen calibration.
1	0	Mutual touchscreen calibration calibrates the entire sensor matrix, but touched keys are ignored even though keys are calibrated with the touchscreen.	Mutual capacitance key recovery is disabled.
1	1	Mutual capacitance touchscreen calibration calibrates the entire sensor matrix, but touched keys are ignored even though keys are calibrated with the touchscreen.	Keys calibrate individually without causing a touchscreen calibration.

† Applies to the Key Array T15 object only.
* This configuration option equates to the behavior on previous maXTouch devices.

CALRECTUN: Enables Calibration Recovery Tuning Data. If this bit is set to 1, tuning data for all calibration recovery mechanisms is output via the Diagnostic Debug T37 and Command Processor T6 objects (see Section 2.2 “Diagnostic Debug T37 Object” and “DIAGNOSTIC Field”). The tuning and debug data are described in Section 3.5.4 “Diagnostic Debug Data”.

3.5.3 CONFIGURATION CHECKS

The Acquisition Configuration T8 object causes a configuration check to be performed in the following circumstances:

- When certain fields are changed

In addition, some fields may cause an automatic recalibration to be performed.

A configuration check may determine that a configuration error has occurred (for example, if a setting is set outside of its allowed range or a conflict has occurred between two settings). This is signaled to the host (see Section 3.3.3 “Messages”). The device halts until the error has been corrected. To fix the error, check that all the object settings are within their allowed limits, as stated in the field descriptions.

TABLE 3-14: CONFIGURATION CHECKS FOR ACQUISITION CONFIGURATION T8 (GEN_ACQUISITIONCONFIG_T8)

Field	Changing The Field Causes...		Effect of Configuration Checks On Field
	Configuration Check	Automatic Recalibration	
CHRGTIME	No	Yes	None
TCHDRIFT	No	No	None
DRIFTST	No	No	None
TCHAUTOCAL	No	No	None
SYNC	Yes	No	None
ATCHCALST	No	No	None
ATCHCALSTHR	No	No	None
ATCHFRCCALTHR	No	No	None
ATCHFRCCALRATIO	No	No	None
MEASALLOW	Yes	Yes	None
CFG	No	No	None

3.5.4 DIAGNOSTIC DEBUG DATA

The diagnostic debug data modes for the Acquisition Configuration T8 object is shown in Table 3-15. The diagnostic debug data can be viewed in the Diagnostic Debug T37 object by using the appropriate Command Processor T6 command, also listed in Table 3-15.

The diagnostic debug modes are described in the following sections.

TABLE 3-15: DIAGNOSTIC DEBUG COMMANDS

T6 Command	Name	Description
0x33	Calibration Recovery Tuning Data Mode	The Diagnostic Debug T37 object holds the Calibration Recovery Tuning Data. This is the tuning data for any enabled calibration recovery mechanisms. Note that the CFG CALRECTUN bit must be set to 1 for this to happen (see "CFG Field"). See Section 3.5.4.1 "Calibration Recovery Tuning Data Mode".

3.5.4.1 Calibration Recovery Tuning Data Mode

Figure 3-9 shows the format of the data in Calibration Recovery Tuning Data Mode. This reports the tuning data for any enabled calibration recovery mechanisms.

FIGURE 3-9: FORMAT OF THE CALIBRATION RECOVERY TUNING DATA

	Data Index	Data[]	
Page 0	0	Acquisition Configuration T8 ATCHCALST counter	Mutual Capacitance Data
	1	Peak touch delta	
	2	Number of nodes in touch	
	3		
	4	Number of nodes in antitouch	
	5		
	6	Calibration recovery counter	
	7	Calibration recovery state (1)	
	8	Calibration recovery metric	Self Capacitance Touch Data
	9		
	10	Calibration recovery counter	
	11	Calibration recovery state (1)	
	12	Acquisition Configuration T8 ATCHCALST counter	Separate Key Antitouch Calibration Data
	13	Calibration recovery counter	
	14	Calibration recovery state (1)	
	15	Calibration recovery metric	Self Capacitance Touch DC Data
	16		
	17	Calibration recovery counter	
	18	Calibration recovery state (1)	
	19 – 127	Reserved	

Assumes page size = 128

Note 1: Calibration recovery state has one of the following values:
0 = Idle
1 = Counting
2 = Triggered

Under NDA

© 2021 Microchip Technology Inc. and its subsidiaries

4.0 TOUCH OBJECTS

4.1 Introduction

Touch objects operate on measured signals from the touch sensor and report touch data. For example, a Multiple Touch Touchscreen T100 object reports XY touch positions. Table 4-1 lists the touch objects on the mXT640UD-CCUBHA1 Family.

TABLE 4-1: TOUCH OBJECTS

Object	Description
Key Array T15 (TOUCH_KEYARRAY_T15)	Defines a rectangular array of keys. A Key Array T15 object reports simple on/off touch information. See Section 4.2 “Key Array T15 Object”.
Multiple Touch Touchscreen T100 (TOUCH_MULTITOUCHSCREEN_T100)	Creates a touchscreen that supports the tracking of more than one touch. See Section 4.3 “Multiple Touch Touchscreen T100 Object”.

NOTE On the mXT448UD-CCUBHA1 variant device, although up to 640 nodes are available on the matrix, only a maximum of 448 nodes can be used on this device across all enabled touch object instances.

4.2 Key Array T15 Object

4.2.1 INTRODUCTION

4.2.1.1 Standard Keys

The Key Array T15 object is used to configure a rectangular array of XY nodes on the mutual capacitance sensor for use as keys. The Key Array T15 object can be used to configure an array of up to 32 keys.

Refer to the relevant device *Datasheet* for details on the touchscreen layout rules and how the positioning of the Key Array T15 object relates to the positions of the other touch objects.

4.2.1.2 Generic Keys

The Key Array T15 object can be configured to use the Generic Key X and Y lines instead of the Mutual Capacitance X and Y lines. The Generic Key lines effectively provide an additional 3 mutual capacitance nodes that can be used to form a Key Array only.

For the purposes of configuration:

- The single Generic Key X line is treated as being logically located at X32, that is following the standard X lines.
- The 3 Generic Key Y lines are treated as being logically located at Y0 to Y2, that is within the bounds of the standard Y lines. The generic key Y lines thus logically overlap with the standard Y lines.
- The Key Array cannot make use of both Generic Key lines and standard lines, so the maximum size of the Key Array is 3 keys.

Because the generic keys use a single X line, the key numbers are assigned along the Y axis.

Refer to relevant device *Datasheet* for more information on the layout of Key Arrays incorporating the Generic Keys.

Note that using the Generic Keys may add extra noise line measurements, which will impact power consumption and timings. It is therefore recommended that, where spare mutual capacitance sense lines are available, the sense lines are used to form a standard Key Array in preference to using the Generic Key lines.

4.2.1.3 Key IDs

For reporting purposes, the key IDs reported in the Key Array T15 message (see "KEYSTATE Field") correspond to the order of the keys in the defined key array. The keys are numbered along the X lines in Y–X order, as defined by the following equation:

$$\text{Key ID} = (\text{Xline} - \text{XORIGIN}) + ((\text{Yline} - \text{YORIGIN}) \times \text{XSIZE})$$

For example, for a standard key array of 4 X by 3 Y lines, the key IDs are allocated as in Table 4-2.

Note that, because the generic keys use a single X line, the generic keys are effectively numbered along the Y axis (see Table 4-3).

TABLE 4-2: EXAMPLE KEY NUMBERS – STANDARD KEYS 4 X BY 3 Y LINES

		X Lines			
		X*n*	**X*n*+1**	**X*n*+2**	**X*n*+3**
Y Lines	**Y*m***	Key 0	Key 1	Key 2	Key 3
	Y*m*+1	Key 4	Key 5	Key 6	Key 7
	Y*m*+2	Key 8	Key 9	Key 10	Key 11

TABLE 4-3: EXAMPLE KEY NUMBERS – GENERIC KEYS 1 X BY 3 Y LINES

		X Lines
		GKEYX0
Y Lines	**GKEYY0**	Key 0
	GKEYY1	Key 1
	GKEYY2	Key 2

Under NDA

© 2021 Microchip Technology Inc. and its subsidiaries

By default, the key IDs correspond to the key numbers of the defined key array. The Key ID Configuration T16 object, if enabled, allows an alternative mapping between the keys and the key IDs to be configured. This allows the Key Array T15 object to report its key states according to the key IDs defined by the Key ID Configuration T16 object. This may be useful, for example, if the user requires standard key IDs to be used across various products.

4.2.1.4 Touch Threshold

The Key Array T15 object has a Touch Threshold (TCHTHR) that defines how large a node's touch delta (that is, Reference minus the signal) must be to qualify as a potential touch detection. The Reference is determined during calibration and adjusted using drift compensation. The Reference is determined by the Key Array T15 DRIFT and DRIFTST fields.

The final detection confirmation uses the Touch Detect Integration as described in the TCHDI field. Larger values for the threshold desensitize nodes since the signal must change more in order to exceed the threshold level. Conversely, lower thresholds make nodes more sensitive.

The setting for TCHTHR for each node depends on the amount of signal swing that occurs when a node is touched. Thicker panels or smaller electrode geometries reduce node gain, that is signal swing from touch. In this case smaller TCHTHR values are required to detect touch.

A Touch Hysteresis (TCHHYST) can also be set for all keys in the key array. Note, however, that this hysteresis is capped at run time to 25% of the current touch threshold for all keys.

It is also possible to configure separate (per-key) thresholds and/or hystereses using Key Thresholds T14 EXTRATHR and/or EXTRAHYST (see Key Thresholds T14 “EXTRATHR[] Fields” and “EXTRAHYST[] Fields”).

Other objects can adjust the minimum threshold in specific circumstances. These objects are:

- Noise Suppression T72 – Supplies a minimum touch threshold that desensitizes the Key Array T15 to avoid reporting any detected noise that would be reported as touches (see Section 5.11 “Noise Suppression T72 Object”).
- Glove Detection T78 – Adjusts the threshold to allow for glove touches (see Section 5.12 “Glove Detection T78 Object”) unless CTRL IGNGLOVEMODE is set (see “CTRL Field”).

4.2.1.5 Adjacent Key Suppression Technology

Adjacent Key Suppression (AKS) technology is a patented method used to detect which touch object (Multiple Touch Touchscreen T100 or Key Array T15) is touched, and to suppress touches on the other touch objects, when touch objects are located close together.

The device has two levels of AKS:

- The first level works between the touch objects (Multiple Touch Touchscreen T100 and Key Array T15). The touch objects are assigned to AKS groups. If a touch occurs within one of the touch objects in a group, then touches within other objects inside that group are suppressed. For example, if a touchscreen and a Key Array are placed in the same AKS group, then a touch in the touchscreen will suppress touches in the Key Array, and vice versa.

  A touch in a group of AKS objects is indicated only on the object with the largest signal. This is assumed to be the intended object. Once an object in an AKS group is in detect, there can be no further detections within that group until the object is released.

  Objects can be in more than one AKS group.
- The second level of AKS is internal AKS within an individual Key Array object (note that internal AKS is not present on other types of touch objects, only a Key Array T15). If internal AKS is enabled, then when one key is touched, touches on all the other keys within the Key Array are suppressed.

AKS technology works best when it operates in conjunction with a detect integration setting of several acquisition cycles.

NOTE	If a touch is in detect and then AKS is enabled, that touch will not be forced out of detect. It will not go out of detect until the touch is released. AKS will then operate normally. This applies to both levels of AKS.

4.2.2 CONFIGURATION

TABLE 4-4: CONFIGURATION FOR KEY ARRAY T15 (TOUCH_KEYARRAY_T15)

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	CTRL	INTAKSEN	Reserved		IGNGLOVEMODE	TSCONTSUP	Reserved	RPTEN	ENABLE
1	XORIGIN	X line start position of object							
2	YORIGIN	Y line start position of object							
3	XSIZE	Number of X lines the object occupies							
4	YSIZE	Number of Y lines the object occupies							
5	AKSCFG	Group8	Group7	Group6	Group5	Group4	Group3	Group2	Group1
6	GAIN	Measurement gain							
7	TCHTHR	Touch threshold							
8	TCHDI	DISTCHDIDOWN	Reserved	TCHDI					
9	TCHHYST	Reserved		Touch hysteresis					
10	CFG	Reserved					DBGSIGSEN	DBGREFSEN	DBGDELTASEN

CTRL Field

ENABLE: Enables the use of the Key Array T15 object. The object is enabled if set to 1, and disabled if set to 0.

RPTEN: Allows the object to send status messages to the host through the Message Processor T5 object. Reporting is enabled if set to 1, and disabled if set to 0.

TSCONTSUP: If this bit is set to 1, the Touchscreen Contact Suppression algorithm enabled. When this feature is enabled, key reporting for Key Array T15 will be suppressed if there is a contacting touch on any touchscreen when the keys are going into detect. Key reporting will remain suppressed until all keys are released.

In designs where keys are situated near to the touchscreen without a bezel separation, there is a possibility that the user's touch might unintentionally slide onto a key. Even with AKS enabled, the key can still report, once the touch is no longer detected on the touchscreen. In this case, enabling the Touchscreen Contact Suppression can help suppress the key.

Note that for the Touchscreen Contact Suppression to function, the external AKS (see "AKSCFG Field") between the key array and the touchscreen has to be disabled (that is, they cannot be in the same AKS group).

IGNGLOVEMODE: Ignores Glove Mode. This controls any threshold adjustment relating to gloves. If this bit is set to 0, Glove Detection T78 may lower the Key Array T15 TCHTHR so that gloves may be detected. If this bit is set to 1 (recommended), Glove Detection T78 no longer has any control over the Key Array T15 TCHTHR, but the user may set Key Array T15 TCHTHR to detect gloves explicitly.

Microchip recommends that IGNGLOVEMODE is always set to 1.

INTAKSEN: Enables internal Adjacent Key Suppression® (AKS®) between keys in the array (see Section 4.2.1.5 "Adjacent Key Suppression Technology"). If internal AKS is enabled, then when one key is touched, touches on all the other keys within the Key Array are suppressed. Set this bit to 1 to enable internal AKS, and to 0 to disable internal AKS.

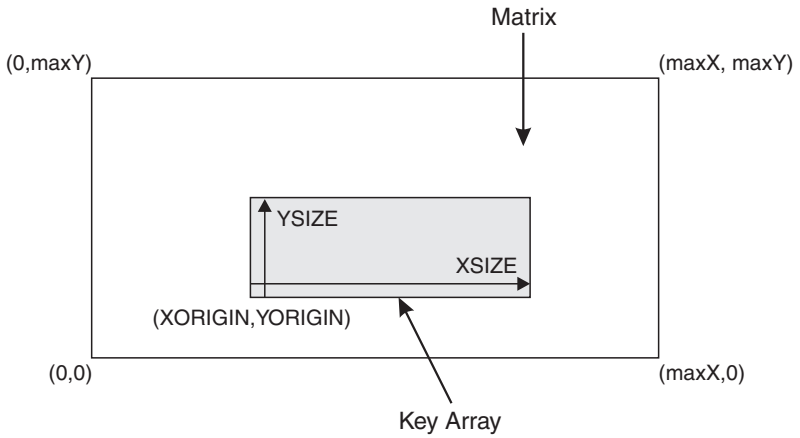
XORIGIN, YORIGIN, XSIZE and YSIZE Fields

These fields specify the size and position of the key Array on the actual sensor matrix (see Figure 4-1). The XORIGIN and YORIGIN fields specify the origin in X and Y lines. The XSIZE and YSIZE fields specify the size in X/Y nodes (keys).

Under NDA

© 2021 Microchip Technology Inc. and its subsidiaries

FIGURE 4-1: KEY ARRAY T15 ORIGIN AND SIZE FIELDS



The minimum size for an enabled key array is 1 X by 1 Y line. The maximum size for a key array is such that when XSIZE and YSIZE are multiplied together they obey the rule: XSIZE × YSIZE $\leq$ 32 nodes. Setting XSIZE and YSIZE to values that create more than 32 keys causes a configuration error.

NOTE	A standard key array must not share X and Y lines with other touch objects (specifically Multiple Touch Touchscreen T100). In addition, the key array must use X and Y lines numbered above those used by the enabled Multiple Touch Touchscreen T100 object. Note that this restriction does not apply to a key array that is implemented using the generic keys as they do not use the physical X and Y lines on the main sensor. Refer to the relevant device *Datasheet* for full details on the layout of the sensor.

XORIGIN Range for Standard Key Array: (T100.XORIGIN + T100.XSIZE) to (maxX – T15.XSIZE + 1)

YORIGIN Range for Standard Key Array: (T100.YORIGIN + T100.YSIZE) to (maxY – T15.YSIZE + 1)

XORIGIN Range for Generic Key Array: 32 only

YORIGIN Range for Generic Key Array: 0 to 2

XSIZE Range for Standard Key Array: 1 to (32 / YSIZE)

YSIZE Range for Standard Key Array: 1 to (32 / XSIZE)

XSIZE Range for Generic Key Array: 1 only

YSIZE Range for Generic Key Array: 1 to 3

AKSCFG Field

This field configures Adjacent Key Suppression (AKS) between this Key Array T15 object and any other touch objects (see Section 4.2.1.5 “Adjacent Key Suppression Technology”).

Group1 to 8: These bits form a bit field that specifies which AKS groups this Key Array is within. The default value of 0 means that the object is in no AKS groups and the AKS feature is disabled for the Key Array.

Range: 0 to 255

GAIN Field

This field sets the gain of the analog circuits in front of the analog to digital converter (ADC).

Range: 0 to 22

TCHTHR Field

This field specifies the Touch Threshold in 8-bit deltas (see Section 4.2.1 “Introduction” for details).

NOTE

- The Touch Hysteresis specified in the TCHHYST field also has an effect on the Touch Threshold (See “TCHHYST Field”) and should therefore be allowed for in the TCHTHR setting.
- A per-key threshold and/or hysteresis may be configured using Key Thresholds T14 EXTRATHR and/or EXTRAHYST (see Key Thresholds T14 “EXTRATHR[] Fields” and “EXTRAHYST[] Fields”).

Range: 0 to 255

TCHDI Field

To suppress false detections caused by spurious events like electrical noise, the device incorporates a touch detection integrator (DI) counter mechanism to provide signal filtering. A per-key counter is incremented each cycle that a touch is detected. When this counter reaches a preset limit the touch is finally declared to be present. If on any acquisition a delta is not seen to exceed the threshold level, the counter is cleared and the process has to start from the beginning.

The number of cycles for a touch to go into detect depends on the setting of DISTCHDIDOWN:

- If DISTCHDIDOWN = 0, it takes TCHDI + 1 cycles from touchdown to when the first touch is actually reported via the $\overline{\text{CHG_I2C}}$ or $\overline{\text{CHG_SPI}}$ line, with a minimum time of 2 cycles.

  However, if DISTCHDIDOWN is set to 0 in the Key Array T15 object, but is set to 1 in any other touch object instance with a DISTCHDIDOWN control (for example, the Multiple Touch Touchscreen T100 object), it takes only TCHDI cycles, with a minimum time of 1 cycle.

  Once a touch is detected, the limit is changed to TCHDI cycles.
- If DISTCHDIDOWN = 1, it takes 1 cycle from touchdown to when the first touch is actually reported (that is, TCHDI is ignored).

IMPORTANT! Touch DI must be enabled for UL/IEC 60730 Class B compliance. Therefore the DISTCHDIDOWN must be set to 0 (that is, touch DI must not be disabled).

An opposite process is applied when a key leaves detection. The counter is decremented each cycle that the delta does not exceed the threshold level, and incremented again if it does exceed the threshold. When the counter reaches zero, the touch is finally declared to be out of detect. For TCHDI values of 0 and 1, it takes 1 cycle for a touch to go out of detect; for TCHDI values of 2 to 63 it takes (TCHDI + 1) cycles.

The effects of various TCHDI values, and the DISTCHDIDOWN control, are summarized in Table 4-5 and Table 4-6.

TABLE 4-5: TCHDI – DISTCHDIDOWN = 0

Condition	Number of Cycles				Notes
	TCHDI = 0	TCHDI = 1	TCHDI = 2	TCHDI ≥ 3 (*n*)	
First touch down (from Idle)	2	2	3	*n*+1	DISTCHDIDOWN = 0 in all other objects/instances
	1	1	2	n	DISTCHDIDOWN = 1 in at least one other object/instance
Next touch down (from Active)	1	1	2	n	
Touch release	1	1	3	*n*+1	

TABLE 4-6: TCHDI – DISTCHDIDOWN = 1

Condition	Number of Cycles			
	TCHDI = 0	TCHDI = 1	TCHDI = 2	TCHDI ≥ 3 (*n*)
Touch down	1	1	1	1
Touch release	1	1	3	*n*+1

Under NDA

© 2021 Microchip Technology Inc. and its subsidiaries

TCHDI: Specifies the base number of DI cycles.

The range for TCHDI is 0 to 63, where 0 is the same as 1.

TCHDI Range: 0 (1), 1 to 63

DISTCHDIDOWN: Disables Detect Integration (DI) Touchdown. When this control is set to 1, the DI process is disabled for touchdown and the value set in the TCHDI control is ignored. The TCHDI control is, however, still used for the DI process on touch release.

NOTE	By setting this control to 1, first touch latency will be reduced. This can, however, have a detrimental effect on the reliability of touchdown detection for all touch object instances other than Multiple Touch Touchscreen T100. Microchip therefore recommends setting DISTCHDIDOWN to 0.

TCHHYST Field

This field sets the Touch Hysteresis (in 8-bit deltas). This is the level of hysteresis applied to touch detections for all keys in the key array. For a touch to enter detection the touch delta must be greater than the touch threshold (TCHTHR). For a touch to leave detection the touch delta must be less than (TCHTHR – TCHHYST). This field allows the Key Array T15 to be tuned so that a hovering finger does not cause detection to flicker on and off.

Note that if the Noise Suppression T72 object supplies a minimum touch threshold that is greater than TCHTHR – TCHHYST, then a touch's delta must be greater than the minimum touch threshold plus TCHHYST to enter detection, and less than the minimum touch threshold to leave detection.

TCHHYST is capped internally to (TCHTHR / 4). A hysteresis greater than 25% of the Touch Threshold is therefore not possible.

NOTE	A per-key hysteresis may be configured using Key Thresholds T14 EXTRAHYST (see "EXTRAHYST[] Fields").

Range: 0 (2), 1 to 63

CFG Field

This field allows different debug data types to be enabled or disabled for use with the hardware debug interface

DBGDELTASEN: If this bit is set to 1, the key deltas are output by the hardware debug interface. Note that the Command Processor T6 DEBUGCTRL2 DBGOBJMODEEN bit must be set to 1 for this to work (see "DEBUGCTRL2 Field").

DBGREFSEN: If this bit is set to 1, the key references are output by the hardware debug interface. Note that the Command Processor T6 DEBUGCTRL2 DBGOBJMODEEN bit must be set to 1 for this to work (see "DEBUGCTRL2 Field").

DBGSIGSEN: If this bit is set to 1, the key signals are output by the hardware debug interface. Note that the Command Processor T6 DEBUGCTRL2 DBGOBJMODEEN bit must be set to 1 for this to work (see "DEBUGCTRL2 Field").

4.2.3 CONFIGURATION CHECKS

The Key Array T15 object causes a configuration check to be performed in the following circumstances:

- When the object is enabled (that is, the ENABLE bit is set in the CTRL field)
- If the object is enabled, when certain fields are changed

In addition, some fields will cause an automatic recalibration to be performed.

A configuration check may determine that a configuration error has occurred (for example, if a setting is set outside of its allowed range or a conflict has occurred between two settings). This is signaled to the host (see Section 3.3 "Command Processor T6 Object"). The device halts until the error has been corrected.

In addition to the configuration errors caused by the field listed in Table 4-7, the following may also cause an error:

- If the defined key array includes both standard matrix keys and generic keys.
- If the Key Arrays uses Generic Keys, XORIGIN or YORGIN is not within the Generic Key region.
- If the Key Arrays uses Generic Keys, a Generic Key pin has an alternative function and that function is being used.

To fix the error, the object settings should be checked to verify that they are all within their allowed limits, as described in the field descriptions. In particular, the following should be checked:

- There are no more than 32 keys defined, that is: XSIZE x YSIZE ≤ 32.
- The key array does not share X and Y lines with other touch objects (specifically Multiple Touch Touchscreen T100). Specifically, the key array must use X and Y lines numbered above those used by the enabled Multiple Touch Touchscreen T100 object.
- The key array does not straddle both standard sense lines and generic key sense lines.
- If the Generic Key pins (GKEYX0,GKEYY0 to GKEYY2) have alternative functions, the pins are not being used for their alternative functions.

TABLE 4-7: CONFIGURATION CHECKS FOR KEY ARRAY T15 (TOUCH_KEYARRAY_T15)

Field	Changing The Field Causes...		Effect of Configuration Checks On Field
	Configuration Check	Automatic Recalibration	
CTRL	Yes (1)	Yes (1) (2)	None
XORIGIN	Yes	Yes	Error if out of range
YORIGIN	Yes	Yes	Error if out of range
XSIZE	Yes	Yes	Error if out of range
YSIZE	Yes	Yes	Error if out of range
AKSCFG	No	No	None
GAIN	No	Yes	None
TCHTHR	No	No	None
TCHDI	No	No	None
TCHHYST	No	No	None
CFG	No	No	None

Note 1: If the ENABLE bit is toggled on or off.
2: If the ENABLE bit is toggled on or off. Note that if the ENABLE bit is cleared and *all of* the touch objects are now disabled, no calibration takes place.

4.2.4 MESSAGES

A Key Array T15 object reports on/off touch information in its message data. The message data for a Key Array T15 object is shown in Table 4-8.

Note that Key Array T15 messages will be suppressed if Touchscreen Contact Suppression is enabled, and the conditions are met (see TSCONTSUP bit in the "CTRL Field").

TABLE 4-8: MESSAGE DATA FOR KEY ARRAY T15 (TOUCH_KEYARRAY_T15)

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	STATUS	DETECT	Reserved						
2	KEYSTATE	KEY7	KEY6	KEY5	KEY4	KEY3	KEY2	KEY1	KEY0
3		KEY15	KEY14	KEY13	KEY12	KEY11	KEY10	KEY9	KEY8
4		KEY23	KEY22	KEY21	KEY20	KEY19	KEY18	KEY17	KEY16
5		KEY31	KEY30	KEY29	KEY28	KEY27	KEY26	KEY25	KEY24

STATUS Field

Reports the current status of the object.

DETECT: Set if any key is in a touched state.

Under NDA

© 2021 Microchip Technology Inc. and its subsidiaries

KEYSTATE Field

KEY0 to KEY31: Report the states of each key in the Key Array, one bit per reported Key ID, which by default equals the key number (see Section 4.2.1.3 “Key IDs”). For example, by default, Key0 has a key ID of 0, which is reported in bit 0 of byte 2. The key state is reported as 0 if the key is untouched, and 1 if the key is touched.

If the Key ID Configuration T16 object is enabled for use (see Section 5.3 “Key ID Configuration T16 Object”), then the key states are reported according to the key IDs configured in the associated Key ID Configuration T16 object. For example, if Key0 is configured in Key ID Configuration T16 to report as Key ID 8, Key0 will be reported in bit 0 of byte 3 (and not in bit 0 of byte 2).

4.2.5 DIAGNOSTIC DEBUG DATA

The diagnostic debug data modes for the Key Array T15 object is shown in Table 4-9. The diagnostic debug data can be viewed in the Diagnostic Debug T37 object by using the appropriate Command Processor T6 command, also listed in Table 4-9.

> The debug data is output according to the physical key number, and not according to any mapping of key IDs defined by the Key ID Configuration T16 object. It is therefore the user’s responsibility to maintain a record of the Key ID Configuration T16 Key ID mapping information while inspecting the debug data for any Key Array T15 keys. Alternatively, the Key ID Configuration T16 should be disabled while inspecting the debug data.

The diagnostic debug modes are described in the following sections.

TABLE 4-9: DIAGNOSTIC DEBUG COMMANDS

T6 Command	Name	Description
0x17	Key Array Delta mode	The Diagnostic Debug T37 object holds the 16-bit deltas for each of the enabled keys on the device. See Section 4.2.5.1 “Key Delta Mode”.
0x18	Key Array Reference mode	The Diagnostic Debug T37 object holds the 16-bit references for each of the enabled keys on the device. See Section 4.2.5.2 “Key Reference Mode”.
0x19	Key Array Signal mode	The Diagnostic Debug T37 object holds the 16-bit signals for each of the enabled keys on the device. See Section 4.2.5.3 “Key Signal Mode”.

4.2.5.1 Key Delta Mode

Figure 4-2 shows the organization of the data in the pages for the key delta mode. The bytes give the 16-bit deltas for the enabled keys in the Key Array (XSIZE × YSIZE). The keys are numbered in node order, as described in Section 4.2.1 “Introduction”.

FIGURE 4-2: DIAGNOSTIC DEBUG T37 DATA – KEY DELTA MODE

Page	Data Index	Data[]
Page 0	0 – ((2 × *n*)-1)	Deltas for enabled keys in Key Array T15
	(2 × *n*) – 127	Reserved

NOTE: *n* = number of enabled keys (XSIZE × YSIZE)

Assumes page size = 128

4.2.5.2 Key Reference Mode

Figure 4-3 shows the organization of the data in the pages for the key reference mode. The bytes give the 16-bit references for the enabled keys in the Key Array (XSIZE × YSIZE). The keys are numbered in node order, as described in Section 4.2.1 “Introduction”.

FIGURE 4-3: DIAGNOSTIC DEBUG T37 DATA – KEY REFERENCE MODE

Page	Data Index	Data[]
Page 0	0 – ((2 × *n*)-1)	References for enabled keys in Key Array T15
	(2 × *n*) – 127	Reserved

NOTE: *n* = number of enabled keys (XSIZE × YSIZE)

Assumes page size = 128

4.2.5.3 Key Signal Mode

Figure 4-4 shows the organization of the data in the pages for the key signal mode. The bytes give the 16-bit signals for the enabled keys in the Key Array (XSIZE × YSIZE). The keys are numbered in node order, as described in Section 4.2.1 “Introduction”.

FIGURE 4-4: DIAGNOSTIC DEBUG T37 DATA – KEY SIGNAL MODE

Page	Data Index	Data[]
Page 0	0 – ((2 × *n*)-1)	Signals for enabled keys in Key Array T15
	(2 × *n*) – 127	Reserved

NOTE: *n* = number of enabled keys (XSIZE × YSIZE)

Assumes page size = 128

Under NDA

© 2021 Microchip Technology Inc. and its subsidiaries

4.3 Multiple Touch Touchscreen T100 Object

4.3.1 INTRODUCTION

The Multiple Touch Touchscreen T100 object is used to configure a multiple-touch touchscreen on the sensor matrix.

The Multiple Touch Touchscreen T100 object reports on all measurements made in the region of the sensor matrix that it occupies, including those detected by other linked objects (see Section 4.3.4 “Messages”).

NOTE	To be able to track and report touches in self capacitance measurements, the Multiple Touch Touchscreen T100 object needs to have data for both axes (see Self Capacitance Global Configuration T109 “CTRL Field” and Appendix A “Measurement Processing”).

4.3.1.1 Touch Threshold

Multiple Touch Touchscreen T100 object has a Touch Threshold (TCHTHR) that defines how large a node's touch delta (Reference minus the signal) must be to qualify as a potential touch detection (in 8-bit deltas). The reference level is determined during calibration and adjusted using drift compensation. The final detection confirmation uses the Touch Detect Integration as described in “TCHDIDOWN Field”. Larger values for the threshold desensitize nodes, since the signal must change more in order to exceed the threshold level. Conversely, lower thresholds make nodes more sensitive.

The setting for TCHTHR for each node depends on the amount of signal swing that occurs when a node is touched. Thicker panels or smaller electrode geometries reduce node gain, that is signal swing from touch. In this case smaller TCHTHR values are required to detect touch.

A Touch Hysteresis is provided by the TCHHYST field. For a touch to enter detection the touch delta must be greater or equal to the touch threshold (TCHTHR). For a touch to leave detection the touch delta must be less than (TCHTHR – TCHHYST).

In addition to the Touch Threshold there is also an Internal Touch Threshold that operates in a similar manner to control internally tracked touch detections. By default, it is set to (TCHTHR – TCHHYST)/2. The Internal Touch Threshold also has a hysteresis (INTTHRHYST). For a touch, therefore, to enter detection the touch delta must be greater or equal to INTTHR and for a touch to leave detection the touch delta must be less than (INTTHR – INTTHRHYST).

Other objects can adjust the minimum touch threshold in specific circumstances. These objects are:

- Noise Suppression T72 – Supplies a minimum touch threshold that desensitizes the Multiple Touch Touchscreen T100 to avoid reporting any detected noise that would be reported as touches (see Section 5.11 “Noise Suppression T72 Object”). This affects both the Touch Threshold and the Internal Touch Threshold.

  Note that if the Noise Suppression T72 object supplies a minimum touch threshold that is greater than TCHTHR – TCHHYST, then a touch delta must be greater than the minimum touch threshold plus TCHHYST to enter detection, and less than the minimum touch threshold to leave detection.

  Similarly, if the Noise Suppression T72 object supplies a minimum touch threshold that is greater than INTTHR – INTTHRHYST, then a touch’s delta must be greater than the minimum touch threshold plus INTTHRHYST to enter detection, and less than the minimum touch threshold to leave detection.
- Glove Detection T78 – Can adjust the threshold to allow for glove touches (see Section 5.12 “Glove Detection T78 Object”).

4.3.1.2 Edge Correction

The Multiple Touch Touchscreen T100 object allows edge correction to be applied for improved linearity at the edge of the screen on certain touchscreen designs. The following edge correction functionality is provided (see Table 4-10):

- **Uniform edge correction** – The edge correction settings are the same for both the high and low sense lines on the X or Y axis.
- **Separate edge correction** – The edge correction settings are different for both the high and low sense lines on the X or Y axis. This allows separate control of the edge correction that is applied to the high and low sense line edges. Sensors with different non-linear responses at the high and low sense lines of a particular axis can therefore be corrected for.
- **Dual X** – In addition to the uniform and separate edge correction modes, the X axis can have additional settings that apply when Dual X measurements are being made.

The chosen edge correction can also be overridden by other objects for certain types of touches (although edge correction for normal finger touches is still controlled by Multiple Touch Touchscreen T100):

- Passive Stylus T47 – provides edge correction for passive stylus touches (see Section 5.8.2 “Configuration”).

Edge correction requires two parameters to be specified: the Distance fields (see Table 4-10), which specify the width of the region at the edge of the sensor where edge corrections are applied, and the Gradient controls, which adjust the “slope” of the corrections within the edge regions specified by the Distance fields.

The edge correction controls operate on the physical ITO matrix of the touchscreen.

The edge correction controls are not affected by the logical orientation set by the orientation controls in the CFG1 field.

The types of edge correction can be set independently on the X and Y axes. For example, it is possible to have uniform edge correction on the Y axis and separate edge correction on the X axis.

See the following sections for more information on the Multiple Touch Touchscreen T100 fields that control edge correction:

“XEDGECFG, DXXEDGECFG and YEDGECFG Fields”
“XEDGEDIST, DXXEDGEDIST and YEDGEDIST Fields”
“XEDGECFGHI, DXXEDGECFGHI and YEDGECFGHI Fields”
“XEDGEDISTHI, DXXEDGEDISTHI and YEDGEDISTHI Fields”

Detailed information on tuning the edge correction can be found in the following application note:

- Application Note AN2659: *Edge Correction on a maXTouch Touch Sensor: an Empirical Method*

TABLE 4-10: EDGE CORRECTION FIELDS

Setting	Fields	Effect	
Uniform X Edge Correction XEDGECFGHI ENABLE = 0	XEDGECFG XEDGEDIST	Set the edge correction for both high and low X lines.	Low and high sense lines have same edge correction
Uniform Dual X Edge Correction DXXEDGECFGHI ENABLE = 0	DXXEDGECFG DXXEDGEDIST	Set the edge correction for both high and low X lines for Dual X measurements.	
Uniform Y Edge Correction YEDGECFGHI ENABLE = 0	YEDGECFG YEDGEDIST	Set the edge correction for both high and low Y lines.	
Separate X Edge Correction XEDGECFGHI ENABLE = 1	XEDGECFGHI XEDGEDISTHI	Set the edge correction for high X lines.	Low and high sense lines have different edge correction
	XEDGECFG XEDGEDIST	Set the edge correction for low X lines.	
Separate Dual X Edge Correction DXXEDGECFGHI ENABLE = 1	DXXEDGECFGHI DXXEDGEDISTHI	Set the edge correction for high X lines for Dual X measurements.	
	DXXEDGECFG DXXEDGEDIST	Set the edge correction for low X lines for Dual X measurements.	
Separate Y Edge Correction YEDGECFGHI ENABLE = 1	YEDGECFGHI YEDGEDISTHI	Set the edge correction for high Y lines.	
	YEDGECFG YEDGEDIST	Set the edge correction for low Y lines.	

Note: Other objects may supply edge correction for specific types of touches, in which case they will take precedence over Multiple Touch Touchscreen T100 edge correction settings.

Under NDA

© 2021 Microchip Technology Inc. and its subsidiaries

4.3.2 CONFIGURATION

TABLE 4-11: CONFIGURATION FOR MULTIPLE TOUCH TOUCHSCREEN T100 (TOUCH_MULTITOUCHSCREEN_T100)

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	CTRL	SCANEN	Reserved				DISSCRMSG0	RPTEN	ENABLE
1	CFG1	INVERTX	INVERTY	SWITCHXY	DISLOCK	ATCHTHRSEL	Reserved		RPTEACHCYCLE
2	SCRAUX	Reserved				INTHRAREA	ATCHAREA	TCHAREA	NUMRPTTCH
3	TCHAUX	Reserved		AREAHW	PEAK	HW	AREA	AMPL	VECT
4	TCHEVENTCFG	Reserved		DISMOVE	DISSUP	DISUNSUP	DISUP	DISDOWN	Reserved
5	AKSCFG	GROUP8	GROUP7	GROUP6	GROUP5	GROUP4	GROUP3	GROUP2	GROUP1
6	NUMTCH	Reserved			Number of reported touches				
7	XYCFG	YSPAN	Reserved			XSPAN	Reserved		
8	XORIGIN	X line start position of object							
9	XSIZE	Number of X lines the object occupies							
10	XPITCH	X line pitch							
11	XLOCLIP	X low clipping boundary width							
12	XHICLIP	X high clipping boundary width							
13	XRANGE	X resolution LSByte							
14		X resolution MSByte							
15	XEDGECFG	Reserved		CORRECTIONGRADIENT					
16	XEDGEDIST	X edge correction distance							
17	DXXEDGECFG	Reserved		CORRECTIONGRADIENT					
18	DXXEDGEDIST	Dual X X edge correction distance							
19	YORIGIN	Y line start position of object							
20	YSIZE	Number of Y lines the object occupies							
21	YPITCH	Y line pitch							
22	YLOCLIP	Y low clipping boundary width							
23	YHICLIP	Y high clipping boundary width							
24	YRANGE	Y resolution LSByte							
25		Y resolution MSByte							
26	YEDGECFG	Reserved		CORRECTIONGRADIENT					
27	YEDGEDIST	Y edge correction distance							
28	GAIN	Gain							
29	DXGAIN	Dual X Gain							
30	TCHTHR	Touch threshold							
31	TCHHYST	Touch threshold hysteresis							
32	INTTHR	Internal threshold							
33	NOISESF	Noise Scaling Factor							
34	CUTOFFTHR	Cut-off threshold							
35	MRGTHR	Merge threshold							
36	MRGTHRADJSTR	Merge Threshold Adjustment Strength							
37	MRGHYST	Merge hysteresis							
38	DXTHRSF	Dual X Threshold scaling factor							
39	TCHDIDOWN	DISTCHDIDOWN	Reserved	TCHDIDOWN					
40	TCHDIUP	Reserved		TCHDIUP					
41	NEXTTCHDI	Reserved		NEXTTCHDI					
42	CALCFG	Reserved							P2PMUTUALEN
43	JUMPLIMIT	Jump limit							
44	MOVFILTER	DISABLE	MEDOFF	Reserved		SPEEDRESP			
45	MOVSMOOTH	Movement smoothing							

TABLE 4-11: CONFIGURATION FOR MULTIPLE TOUCH TOUCHSCREEN T100 (TOUCH_MULTITOUCHSCREEN_T100) (CONTINUED)

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
46	MOVPRED	Movement prediction							
47	MOVHYSTI	Movement hysteresis, initial LSByte							
48		Reserved				Movement hysteresis, initial msbits			
49	MOVHYSTN	Movement hysteresis, next LSByte							
50		Reserved				Movement hysteresis, next msbits			
51	AMPLHYST	Touch amplitude reporting hysteresis							
52	SCRAREAHYST	Screen area reporting hysteresis							
53	INTTHRHYST	Internal tracking threshold hysteresis							
54	XEDGECFGHI	ENABLE	Reserved	CORRECTIONGRADIENT					
55	XEDGEDISTHI	X high edge correction distance							
56	DXXEDGECFGHI	ENABLE	Reserved	CORRECTIONGRADIENT					
57	DXXEDGEDISTHI	Dual X X high edge correction distance							
58	YEDGECFGHI	ENABLE	Reserved	CORRECTIONGRADIENT					
59	YEDGEDISTHI	Y high edge correction distance							
60	CFG2	DBGSIGSEN	DBGREFSEN	DBGDELTASEN	DBGDELTAS8EN	DISESBFSREJ	CONFTHR		
61	MOVHYSTCFG	Reserved	MINSPEED			SPEEDRSP			
62	AMPLCOEFF	Coefficient for scaled touch amplitude							
63	AMPLOFFSET	Offset for scaled touch amplitude							
64	JUMPLIMITMOV	JumpLimit for Moving Touches							
65	JLMMOVTHR	JLMMOVTHR LSByte							
66		Reserved				JLMMOVTHR MSbits			
67	JLMMOVINTTHR	Jump Limit Movement Integration Threshold							
68	EDGESNPCFGIN	ENABLE	RATIO						
69	EDGESNPCFGOUT	ENABLE	RATIO						
70	XBFSREJRANGE	Reserved		XBFSREJRANGE					
71	YBFSREJRANGE	Reserved		YBFSREJRANGE					
72	DBGXOFFSET	Debug region X offset							
73	DBGXSIZE	Debug region X size							
74	DBGYOFFSET	Debug region Y offset							
75	DBGYSIZE	Debug region Y size							

CTRL Field

ENABLE: Enables the use of the Multiple Touch Touchscreen T100 object. The object is enabled if set to 1, and disabled if set to 0. The object does not scan for touches if it is disabled, in order to conserve power.

RPTEN: Allows the object to send status messages to the host through the Message Processor T5 object. Reporting is enabled if set to 1, and disabled if set to 0. Events must be enabled for this bit to have an effect.

DISSCRMSG0: Disables the generation of screen messages in the first report ID.

SCANEN: Enables close scanning for touch detections. The device normally has a very efficient mechanism for determining touch separation (that is, distinguishing between multiple fingers). This is sufficient for most purposes. Under some (very unusual) circumstances, however, it is possible for it to fail to distinguish between multiple touches. Setting this bit to 1 causes the device to perform a close scan for touch detections once every 200 ms. Multiple touches are correctly detected, but the checks still occur infrequently enough to prevent excessive power consumption. This scan is done only when the touchscreen is touched.

Under NDA

© 2021 Microchip Technology Inc. and its subsidiaries

CFG1 Field

RPTEACHCYCLE: Setting this bit to 1 causes the Multiple Touch Touchscreen T100 object to generate a touch status message at the end of each sensor scan cycle for each touch that is reporting, irrespective of whether any events have happened. The message will have the same touch co-ordinates as the preceding touch status message. Setting this bit to 0 gives the default message generation behavior described in Section 4.3.4.3 "Subsequent Report IDs – Touch Status Messages".

ATCHTHRSEL: Selects whether the Antitouch Threshold for Multiple Touch Touchscreen T100 is based on the TCHTHR or INTTHR field. If ATCHTHRSEL is set to 0, the antitouch threshold will be –(TCHTHR–TCHHYST); if ATCHTHRSEL is set to 1, then the antitouch threshold will be –(INTTHR–INTTHRHYST).

This control affects the following features:

- The SCRAUX ATCHAREA data reported in the AUXDATA[] message fields (see "SCRAUX Field" and "AUXDATA[] Fields")
- The Acquisition Configuration T8 mutual capacitance calibration recovery (see Acquisition Configuration T8 Section 3.5.1 "Introduction" and configuration fields)
- The Retransmission Compensation T80 Working Threshold, as determined by the MOISTTHR and MOISTDEGTHR fields (see Section 5.13.1.3 "Working Threshold")

DISLOCK: Disables the coordinate position locking that is applied to touches inside a clipping boundary (see "XRANGE/YRANGE, XLOCLIP/YLOCLIP and XHICLIP/YHICLIP Fields" for more information). Set this field to 1 to disable locking, and to 0 to enable locking.

SWITCHXY: Switches the X and Y positions; that is, the screen is flipped about the diagonal from (0, 0) to (XRANGE, YRANGE).

INVERTY: Inverts Y coordinates; that is: Ynewval = (YRANGE – Y), where Ynewval is the new Y coordinate after the invert operation.

INVERTX: Inverts X coordinates; that is: Xnewval = (XRANGE – X), where Xnewval is the new X coordinate after the invert operation.

Notes:

- The orientation controls are applied to the touchscreen after the resolution has been applied and the position determined (see Figure 4-16 on page 70).
- An INVERTX and/or INVERTY operation takes place before a SWITCHXY.

The effect of these three bits is shown in Table 4-12.

TABLE 4-12: CFG1 ORIENTATION SETTINGS

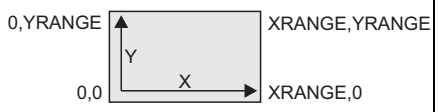
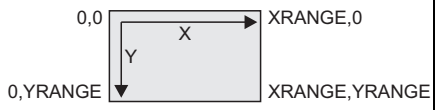
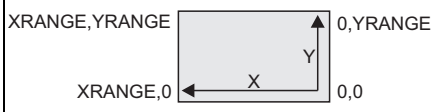
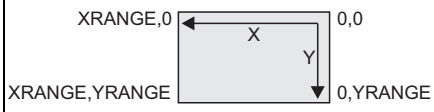
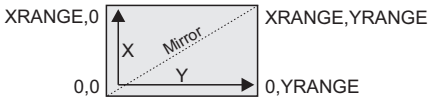
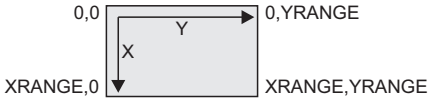
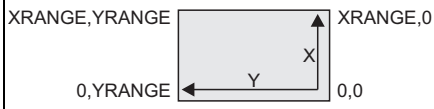
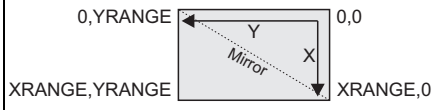
Bits			Touchscreen Coordinates	Touchscreen Orientation
7 (INVERTX)	6 (INVERTY)	5 (SWITCHXY)		
0	0	0		Normal orientation
0	1	0		Invert on Y axis
1	0	0		Invert on X axis

TABLE 4-12: CFG1 ORIENTATION SETTINGS (CONTINUED)

Bits			Touchscreen Coordinates	Touchscreen Orientation
7 (INVERTX)	6 (INVERTY)	5 (SWITCHXY)		
1	1	0		Rotated 180°
0	0	1		Diagonal mirror along axis from (0, 0) to (XRANGE, YRANGE)
0	1	1		Rotated 90° clockwise
1	0	1		Rotated 90° counterclockwise
1	1	1		Diagonal mirror along axis from (0, YRANGE) to (XRANGE, 0)

Note: XRANGE and YRANGE refer to the maximum X and Y screen positions; that is, they relate to the touchscreen resolution and not to the device X and Y lines.

SCRAUX Field

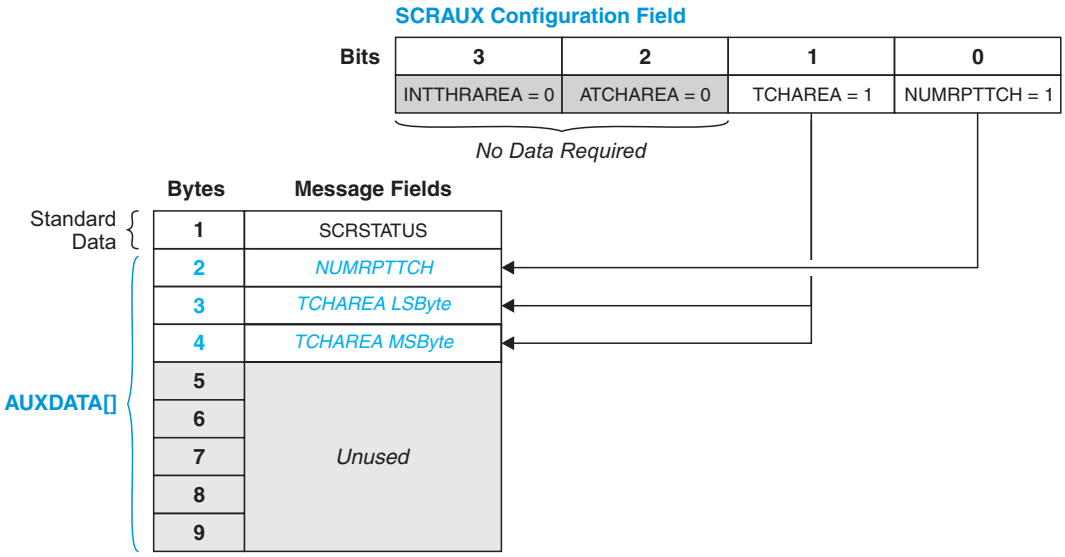
This field configures the report output in the AUXDATA[] fields for the screen status messages (see Section 4.3.4.1 “First Report ID – Screen Status Messages”). See also “SCRAREAHYST Field”.

The AUXDATA[] message bytes are filled in the same order as the SCRAUX bits, depending on which bits are enabled. For example, if only the NUMRPTTCH and TCHAREA bits are set in the SCRAUX configuration field, then only AUXDATA[0] to AUXDATA[2] are used (that is, byte for NUMRPTTCH followed by two bytes for TCHAREA). This is summarized in Figure 4-5.

Under NDA

© 2021 Microchip Technology Inc. and its subsidiaries

FIGURE 4-5: EXAMPLE SCREEN STATUS MESSAGE



NUMRPTTCH: If this bit is set to 1, the number of reportable touches that are being tracked on the touchscreen will be reported in the AUXDATA[] fields. The AUXDATA[] Fields for NUMRPTTCH are shown in Table 4-13.

TABLE 4-13: AUXDATA[] FIELDS FOR NUMRPTTCH

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
n	NUMRPTTCH	Number of Reportable Touches							

The NUMRPTTCH message field indicates the number of reportable touches currently on the screen. A screen status message is generated when this value changes.

TCHAREA: If this bit is set to 1, the number of nodes ≥ (TCHTHR – TCHHYST) on the touchscreen will be reported in the AUXDATA[] fields. The AUXDATA[] Fields for TCHAREA are shown in Table 4-14.

TABLE 4-14: AUXDATA[] FIELDS FOR TCHAREA

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
n	TCHAREA	Touch area LSByte							
n + 1		Touch area MSByte							

The TCHAREA message field indicates the total number of nodes with deltas ≥ (TCHTHR – TCHHYST) on the touchscreen. A screen status message is generated when this value changes by more than SCRAREAHYST.

The data reported in TCHAREA is based on data after processing by the Retransmission Compensation T80 object if the Retransmission Compensation T80 object is enabled.

ATCHAREA: If this bit is set to 1, the number of nodes below the antitouch threshold on the touchscreen will be reported in the AUXDATA[] fields (see ATCHTHRSEL in the "CFG1 Field"). The AUXDATA[] Fields for ATCHAREA are shown in Table 4-15.

TABLE 4-15: AUXDATA[] FIELDS FOR ATCHAREA

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
n	ATCHAREA	Antitouch area LSByte							
n + 1		Antitouch Area MSByte							

The ATCHAREA message field indicates the total number of nodes below the antitouch threshold on the touchscreen. A screen status message is generated when this value changes by more than SCRAREAHYST.

INTTHRAREA: If this bit is set to 1, the number of nodes above (INTTHR –INTTHRHYST) on the touchscreen will be reported in the AUXDATA[] fields. The AUXDATA[] Fields for INTTHRAREA are shown in Table 4-16.

TABLE 4-16: AUXDATA[] FIELDS FOR INTTHRAREA

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
n	INTTHRAREA	Internal tracking threshold area LSByte							
n + 1		Internal tracking threshold area MSByte							

The INTTHRAREA message field indicates the total number of nodes above (INTTHR –INTTHRHYST) on the touchscreen. A screen status message is generated when this value changes by more than SCRAREAHYST.

The data reported in INTTCHAREA is based on data after processing by the Retransmission Compensation T80 object if the Retransmission Compensation T80 object is enabled.

TCHAUX Field

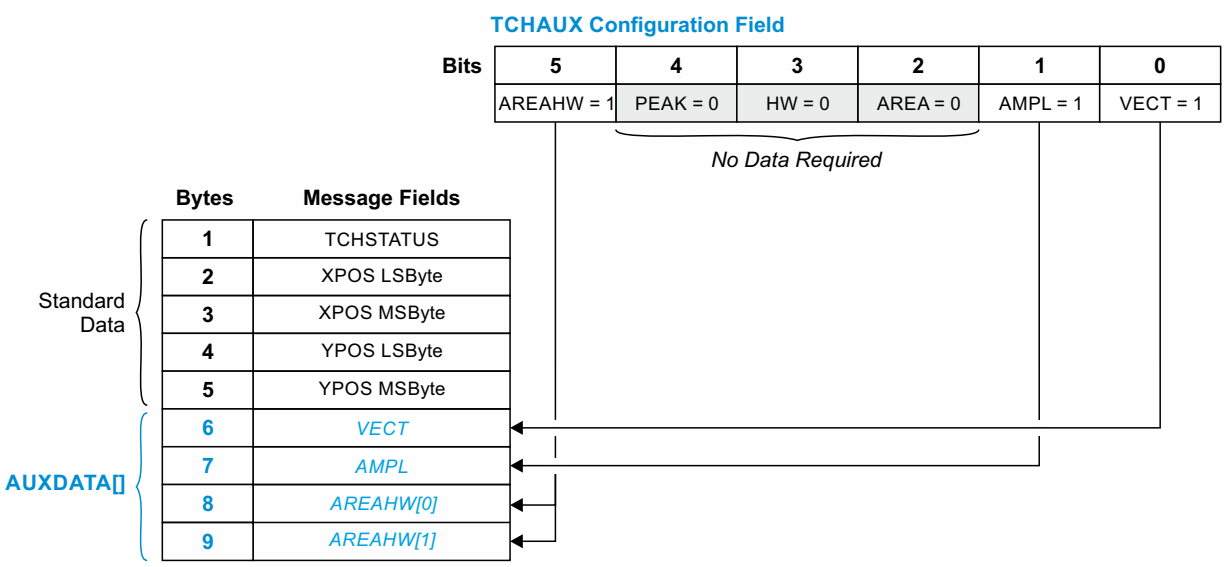
This field configures the report output in the AUXDATA[] fields for the touch status messages (see Section 4.3.4.3 “Subsequent Report IDs – Touch Status Messages”).

The AUXDATA[] bytes are filled in the same order as the TCHAUX bits (starting from bit 0), depending on which bits are enabled.

Note that the Multiple Touch Touchscreen T100 auxiliary data for touch messages are only updated when a touch is detected. This means that touch release messages will contain, for example, peak deltas that are above threshold.

For example, if the VECT, AMPL and AREAHW bits are set to 1 in the TCHAUX field, then the AUXDATA[] bytes will contain one byte for the VECT data, followed by one byte for AMPL data and finally two bytes for the AREAHW data. This is summarized in Figure 4-6. Note that the intermediate bits (AREA, HW and PEAK) are set to 0 in this example. The AREA, HW and PEAK data is therefore skipped in the AUXDATA[] bytes.

FIGURE 4-6: EXAMPLE TOUCH STATUS MESSAGE

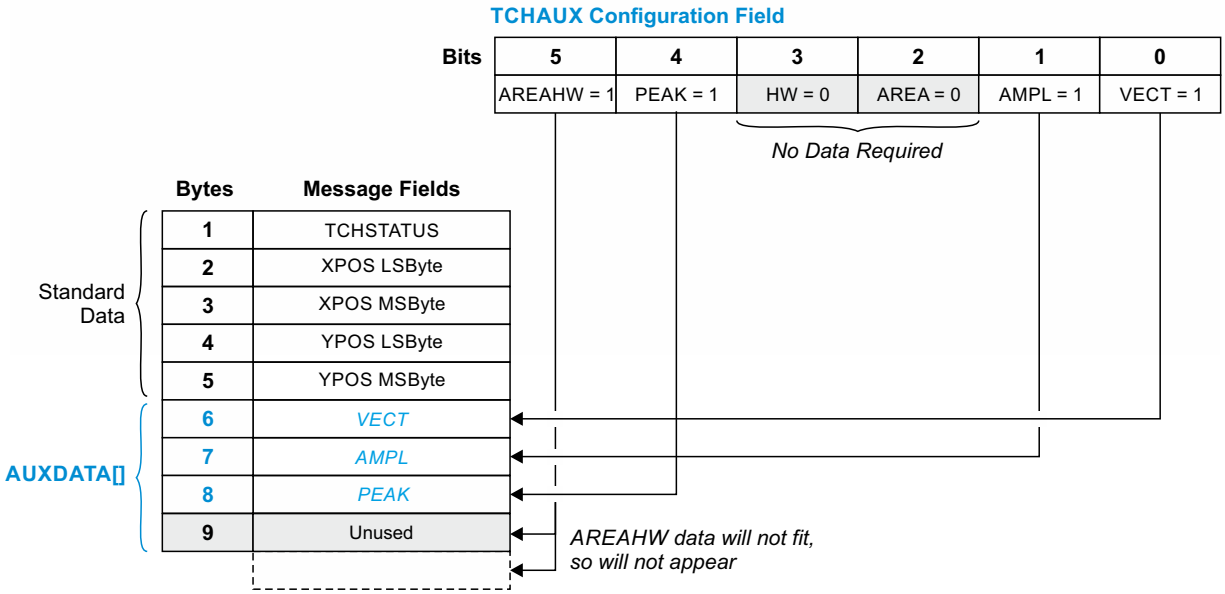


The AUXDATA[] bytes can hold a maximum of four bytes of data, so if more than four bytes of data is selected in TCHAUX, the data for the most significant TCHAUX bits will not appear. For example, if PEAK is also selected, in addition to the bits selected in Figure 4-6, the resulting data is as shown in Figure 4-7. In this case, there is not enough room in the AUXDATA[] bytes for the AREAHW data so it does not appear.

Under NDA

© 2021 Microchip Technology Inc. and its subsidiaries

FIGURE 4-7: EXAMPLE TOUCH STATUS MESSAGE – DATA DOES NOT FIT



VECT: If this bit is set to 1, the vector calculation will be performed on each touch and reported in the AUXDATA[] fields. If this bit set to 0, the calculation will not be performed allowing processing time to be minimized.

The AUXDATA[] Fields for VECT are shown in Table 4-17.

TABLE 4-17: AUXDATA[] FIELDS FOR VECT

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
n	VECT	VCOMP1				VCOMP0			

The VECT message field gives an indication of the orientation of the touch and a confidence level as a vector made up of two signed (two's complement) 4-bit components:

- VCOMP1: ratio of (+X,+Y) to (–X,–Y), representing a diagonal (value: –7 to +7)
- VCOMP0: ratio of X to Y (value: –7 to +7)

The magnitude of the vector represents the confidence. The vector uses the following calculations:

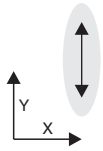
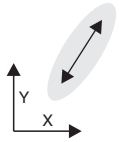
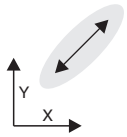
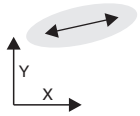
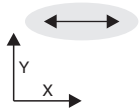
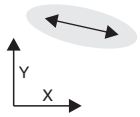
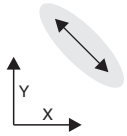
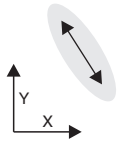
$$Angle = \tan^{-1}\left(\frac{vcomp_1}{vcomp_0}\right)/2$$

$$Magnitude = \sqrt{vcomp_1^{\ 2} + vcomp_0^{\ 2}}$$

NOTE	The touch vector is affected by the orientation of the screen (see "CFG1 Field"). In the example touch directions shown in Table 4-18 the normal orientation is assumed.

Changes to this field do not generate a touch status message; the vector is reported only when another message-triggering event occurs.

TABLE 4-18: TOUCH DIRECTION (NORMAL ORIENTATION ASSUMED)

TCHVECTOR Field		Touch Direction
VCOMP1	VCOMP0	
0	Positive	
Negative	Positive	
Negative	0	
Negative	Negative	
0	Negative	
Positive	Negative	
Positive	0	
Positive	Positive	

AMPL: If this bit is set to 1, the amplitude calculation will be performed on each touch and reported in the AUXDATA[] fields. If this bit set to 0, the calculation will not be performed allowing processing time to be minimized. See also “AMPLHYST Field”. The AUXDATA[] Fields for AMPL are shown in Table 4-19.

Under NDA

© 2021 Microchip Technology Inc. and its subsidiaries

TABLE 4-19: AUXDATA[] FIELDS FOR AMPL

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
n	AMPL	Touch amplitude							

The AMPL message field reports the touch amplitude (saturated to 8 bits). This is a value that is proportional to the signal delta of the nodes within the touch. This can be used, for example, to extrapolate the size or pressure of the user's finger touch. Touch status messages are generated when the delta change in the value is greater than the Amplitude Hysteresis (see "AMPLHYST Field"). The value of the reported touch amplitude can be scaled by the AMPLCOEFF and AMPLOFFSET fields, which are applied before AMPLHYST (see "AMPLCOEFF Field" and "AMPLOFFSET Field").

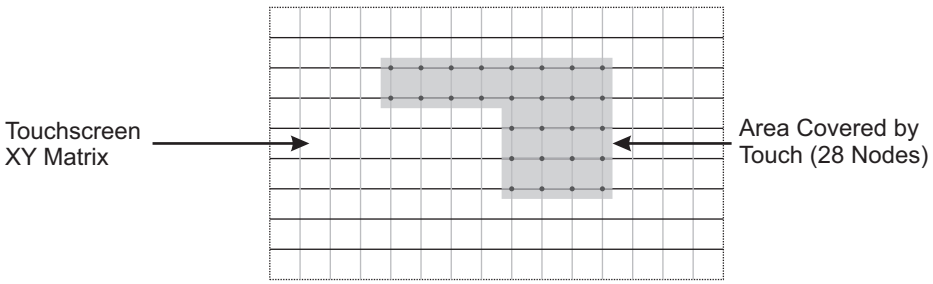
AREA: If this bit is set to 1, the area calculation will be performed on each touch and reported in the AUXDATA[] fields. If this bit set to 0, the calculation will not be performed allowing processing time to be minimized. The AUXDATA[] Fields for AREA are shown in Table 4-20.

TABLE 4-20: AUXDATA[] FIELDS FOR AREA

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
n	AREA	Area							

The AREA message field reports the size of the touch area in terms of the number of nodes that are covered by the touch and are above (TCHTHR–TCHHYST). For example, the area covered by the touch on the touchscreen in Figure 4-8 is 28 nodes.

FIGURE 4-8: TOUCH AREA



Changes to this field do not generate a touch status message; the touch area is reported only when another message-triggering event occurs.

HW: If this bit is set to 1, the height/width calculation will be performed on each touch and reported using the AUXDATA[] fields. Note that the calculation of the height and width uses the XPITCH and YPITCH values and so will be accurate only if these are set correctly for the sensor. If this bit set to 0, the calculation may not be performed, allowing processing time to be minimized.

The AUXDATA[] Fields for HW are shown in Table 4-21.

TABLE 4-21: AUXDATA[] FIELDS FOR HW

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
n	HEIGHT	Touch height							
n + 1	WIDTH	Touch width							

The HEIGHT field reports the height of the touch in mm.

The WIDTH field reports the width of the touch in mm.

PEAK: If this bit is set to 1, the peak calculation will be performed on each touch and reported in the AUXDATA[] fields. If this bit set to 0, the calculation will not be performed allowing processing time to be minimized. The AUXDATA[] Fields for PEAK are shown in Table 4-22.

TABLE 4-22: AUXDATA[] FIELDS FOR PEAK

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
n	PEAK	Peak delta							

The PEAK message field reports the peak delta of the touch in 8-bit delta units (saturated to 8 bits). Changes to this field do not generate a touch status message; the peak delta is reported only when another message-triggering event occurs.

AREAHW: If this bit is set to 1, the area and height/width calculations will be performed on each touch and the data reported in the AUXDATA[] fields. If this bit set to 0, the calculations may not be performed (allowing processing time to be minimized) and the data will not be reported.

The AREAHW bit can be used to allow more pieces of auxiliary data to be reported at once (for example, area, height and width can be reported at the same time as VECT and AMPL). There is no point in setting AREAHW at the same time as AREA or HW as the same information will be reported in two different ways.

The AUXDATA[] Fields for AREAHW are shown in Table 4-23.

TABLE 4-23: AUXDATA[] FIELDS FOR AREAHW

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
n	AREAHW[0]	Reserved	EXP		AREA				
n + 1	AREAHW[1]	WIDTH				HEIGHT			

The AREAHW[0] EXP is an exponent (shift) which applies to AREAHW[0] AREA, AREAHW[1] HEIGHT and AREAHW=[1] WIDTH. For example, EXP can be used to extract the touch area from AREAHW[0], as follows:

Touch Area = AREAHW[0] AREA << AREAHW[0] EXP

After the exponent has been applied, the units of area are nodes and the units of height and width are mm.

If the exponent reaches its maximum value then the reported values in AREAHW[0] AREA, AREAHW[1] HEIGHT and AREAHW[1] WIDTH will be saturated to their respective bit widths.

See AREA on page 63 and HW on page 63 for further information about the reported data.

Changes to the AREAHW bit do not generate a touch status message; the AREAHW data is reported only when another message-triggering event occurs.

TCHEVENTCFG Field

This field provides secondary configuration options to configure the touch events that are able to cause touch status messages to be generated (see Section 4.3.4.3 “Subsequent Report IDs – Touch Status Messages”).

DISDOWN: If this bit is set to 1, the touch DOWN event will not generate a touch message.

DISUP: If this bit is set to 1, the touch UP event will not generate a touch message.

DISUNSUP: If this bit is set to 1, the touch UNSUP event will not generate a touch message.

DISSUP: If this bit is set to 1, the touch SUP event will not generate a touch message.

DISMOVE: If this bit is set to 1, the touch MOVE event will not generate a touch message.

AKSCFG Field

This field configures Adjacent Key Suppression (AKS) between this Multiple Touch Touchscreen T100 instance and any other touch present on the device.

AKS technology is a patented method used to detect which instance of a touch object is touched when objects are located close together. The touch object instances are assigned to AKS groups (objects can be in more than one AKS group). If a touch occurs within one of the touch objects in a group, then touches within other objects inside that group are suppressed and there can be no further detections within that group until that object instance is released. Note that the touch is indicated on the object with the largest signal and this is assumed to be the intended object instance. See also Section 4.2.1.5 “Adjacent Key Suppression Technology”.

GROUP1 to GROUP8: These bits form a bit field that specifies which AKS groups this Multiple Touch Touchscreen T100 object is within. The default value of 0 means that the object is in no AKS groups and the AKS feature is disabled for the Multiple Touch Touchscreen T100.

NUMTCH Field

The Number of Touches (NUMTCH) setting indicates the number of touches to be reported. Report IDs are not removed (see Section 1.5.6 “Report IDs”). The upper limit for this field is the maximum number of touches (16 on the mXT640UD-CCUBHA1 Family).

Under NDA
© 2021 Microchip Technology Inc. and its subsidiaries

Touch IDs are allocated sequentially as touches are detected and any Touch IDs below NUMTCH are reported. This means that Touch IDs ≥ NUMTCH are not reported, even if the number of touches currently on the touchscreen does not exceed NUMTCH.

Range: 0 to 16

XYCFG Field

This field configures settings related to the physical layout of a touchscreen sensor in both axes.

XSPAN and YSPAN: These bits enable the use of an “extra electrode” in the creation of the touch coordinates. XSPAN/YSPAN should be set to 1 if the first and last pixels of the LCD are outside the center point of the first and last sense electrodes.

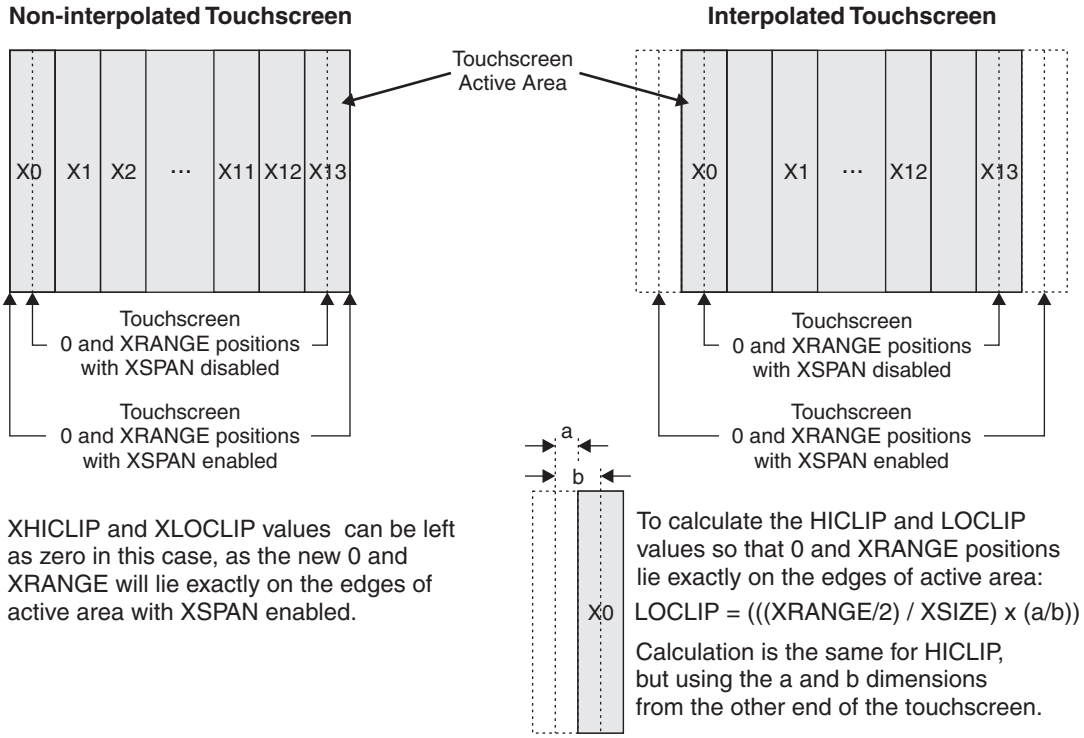
The XSPAN control applies to the low X and high X edges of the touchscreen, and the YSPAN control applies to the low Y and high Y edges of the touchscreen.

At the edges of a touchscreen sensor pattern there should ideally be a half-width electrode. Depending on how the screen has been designed, there may actually be an electrode that is somewhere between 50 percent and 100 percent width at the edge. This can cause discrepancies between the physical touch position and the reported touch position. XSPAN/YSPAN helps correct these discrepancies.

The XSPAN/YSPAN field is used in combination with XLOCLIP/YLOCLIP and XHICLIP/YHICLIP settings. This allows the device to correct for accuracy or linearity problems resulting from the practicalities of the ITO design.

XSPAN/YSPAN moves the theoretical 0 and XRANGE/YRANGE position outwards from the screen center by one half electrode width each. The XLOCLIP/YLOCLIP and XHICLIP/YHICLIP controls can then be used to move the theoretical 0 and XRANGE/YRANGE position back in towards the center in finer steps.Figure 4-9 shows an example of how to correct for two touchscreen designs with an X size of 14. Note that although Figure 4-9 shows the XSPAN control for the X lines, the same principles apply when the YSPAN bit is used with the Y lines.

FIGURE 4-9: TOUCHSCREEN CORRECTION USING XSPAN CONTROL



XSIZE and YSIZE Fields

The XSIZE and YSIZE fields specify the size of the touchscreen on the actual matrix in terms of the number of X and Y lines that the touchscreen occupies.

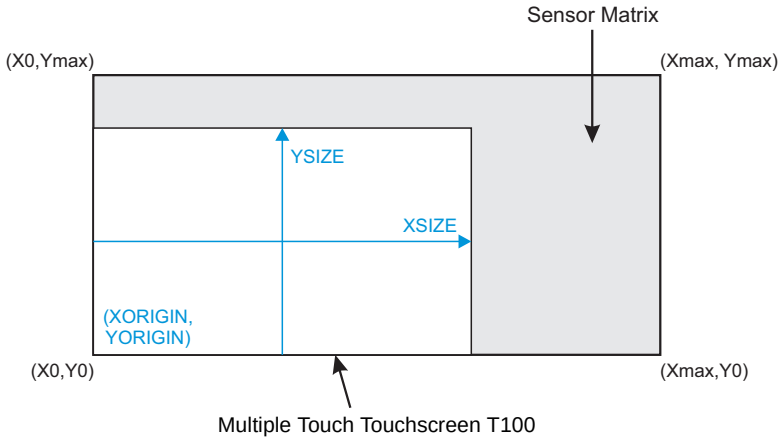
The position of the touchscreen is determined by the XORIGIN and YORIGIN fields (see "XORIGIN and YORIGIN Fields").

For a touchscreen the XSIZE and YSIZE fields should be set as follows (see Figure 4-10):

- XSIZE has a minimum value of 3. If Dual X Drive is enabled for use in the Noise Suppression T72 object, the minimum XSIZE is increased to 4. If self capacitance measurements are enabled in the Acquisition Configuration T8 object on a device where maximum X > maximum Y, the minimum is increased to 6.
- YSIZE has a minimum value of 3.
- The maximum size for a Multiple Touch Touchscreen T100 depends on the number of X and Y lines on a particular device. On the mXT640UD-CCUBHA1 Family the maximum XSIZE is 32 and the maximum YSIZE is 20.

Refer to the relevant device *Datasheet* for more information on the layout and configuration of the touchscreen sensor.

FIGURE 4-10: SIZE AND ORIGIN FIELDS



Note 1: XORIGIN and YORIGIN must be 0.
2: Xmax and Ymax equal the highest X and Y lines (that is, X31 and Y19 on the mXT640UD-CCUBHA1 Family).

XSIZE Range: 3 to 32 or
4 to 32 if Dual X Drive enabled or
6 to 32 if self capacitance measurements enabled

YSIZE Range: 3 to 20

XORIGIN and YORIGIN Fields

The XORIGIN and YORIGIN fields specify the position of the touchscreen on the actual matrix in terms of the index of the X or Y line of the origin (see Figure 4-10). Note that the size of the touchscreen is specified by the XSIZE and YSIZE fields.

XORIGIN Range: 0 only

YORIGIN Range: 0 only

Under NDA
© 2021 Microchip Technology Inc. and its subsidiaries

XPITCH and YPITCH Fields

These fields specify the physical pitch of the X and Y lines on the touchscreen; that is, the spacing between the centers of the X or Y lines. These settings are specified in units of 0.1 mm, where a setting of zero means 5 mm. These fields allow the touch suppression algorithms to be tuned to non-standard pitch sizes.

Range: 0 (5 mm), 1 to 255

XRANGE/YRANGE, XLOCLIP/YLOCLIP and XHICLIP/YHICLIP Fields

These six fields control the resolution, and thus the reported position, of the touchscreen.

NOTE	The XRANGE/YRANGE, XLOCLIP/YLOCLIP and XHICLIP/YHICLIP fields operate on the base touchscreen layout, and are not affected by the logical orientation set by the orientation controls in the CFG1 field.

The XRANGE and YRANGE fields set the output resolution for the reported position.

The minimum resolution is:

(R × N) / 256

The maximum resolution is:

(R × N) – 2 *(capped to 65534)*

where:

For XRANGE

R = 1024 – XLOCLIP – XHICLIP

N = XSIZE – 1 + XSPAN_ENABLED

For YRANGE:

R = 1024 – YLOCLIP – YHICLIP

N = YSIZE – 1 + YSPAN_ENABLED

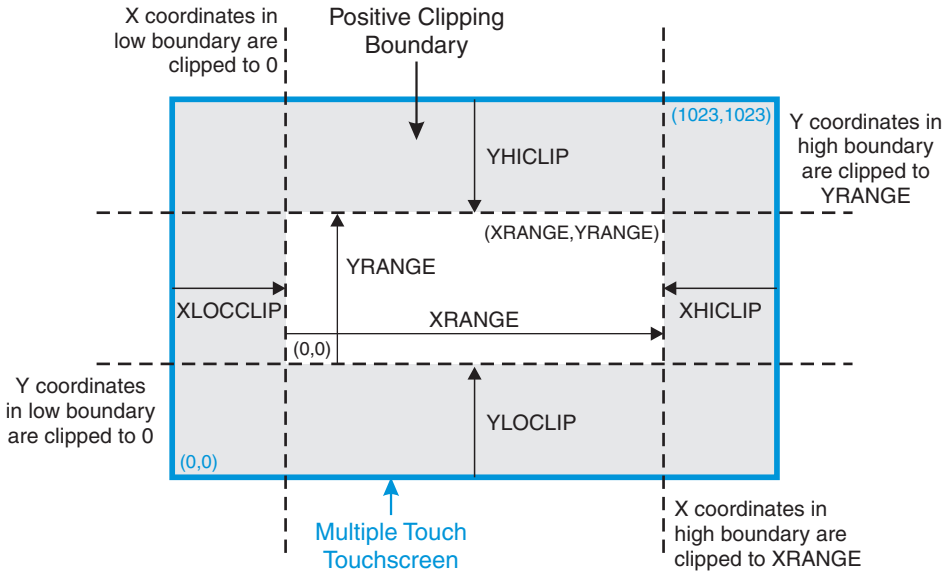
A setting of 0 (default) sets the resolution to 1024. These fields allow a position on the touchscreen to match the resolution of an LCD resolution. For example, to set up a touchscreen to match an LCD with a resolution of 800 x 600, XRANGE and YRANGE would be set to 799 and 599 respectively.

NOTE	The maximum allowed value for XRANGE/YRANGE is 65534.

The XLOCLIP/YLOCLIP and XHICLIP/YHICLIP fields set up a clipping boundary (illustrated by the dark gray regions in Figure 4-11 and Figure 4-12). Any touch position within this boundary has its X or Y coordinate clipped to the minimum or maximum value. These fields take a signed (two's complement) 8-bit value, allowing settings in the range –128 to +127. These fields work at 10-bit resolution (that is, a touchscreen range of 1023). This means that the maximum clipping values allow a clipping boundary one eighth of the screen width or height.

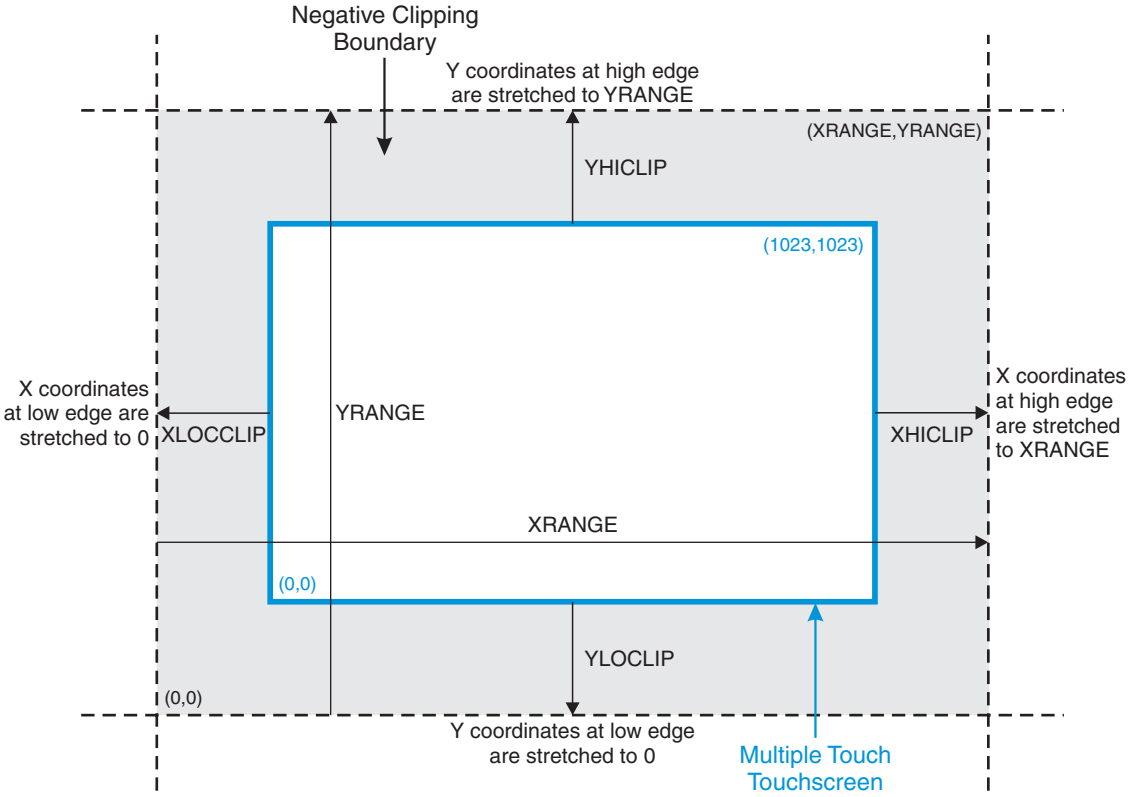
Positive values increase the size of the clipping boundary, moving the 0 and XRANGE/YRANGE reported positions further inside the screen (see Figure 4-11). If the touch enters the clipping boundary, the coordinates are locked at the point at which the touch entered the region. The coordinates are updated only when the touch leaves the clipping boundary. This coordinate locking can be disabled using the DISLOCK field.

FIGURE 4-11: POSITIVE CLIPPING BOUNDARY



Negative values stretch the 0 and XRANGE/YRANGE positions outside of the screen (see Figure 4-12). This limits the reported minimum and maximum value.

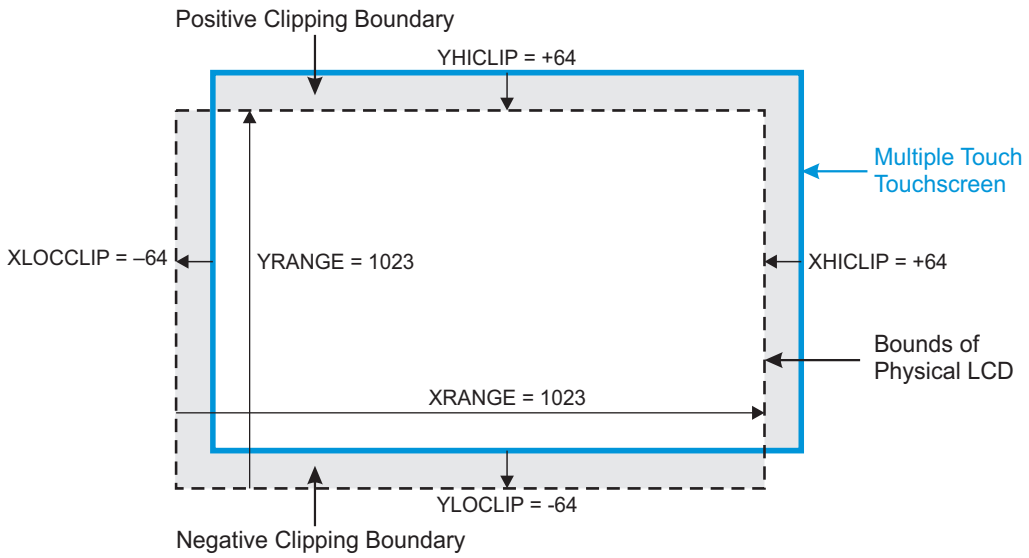
FIGURE 4-12: NEGATIVE CLIPPING BOUNDARY



Adjusting the clipping boundary modifies the accuracy of the reported positions. For example, if the touchscreen is smaller than the physical LCD or is not aligned with the LCD, a combination of positive and negative clipping boundaries help align the reported zero point with the zero point of the LCD. An example of this is shown in Figure 4-13.

Under NDA
© 2021 Microchip Technology Inc. and its subsidiaries

FIGURE 4-13: EXAMPLE RESOLUTION FIELDS



This example results in the coordinates shown in Figure 4-14. The coordinate mapping is shown in Figure 4-15.

FIGURE 4-14: EXAMPLE RESOLUTION FIELDS – RESULTING COORDINATES

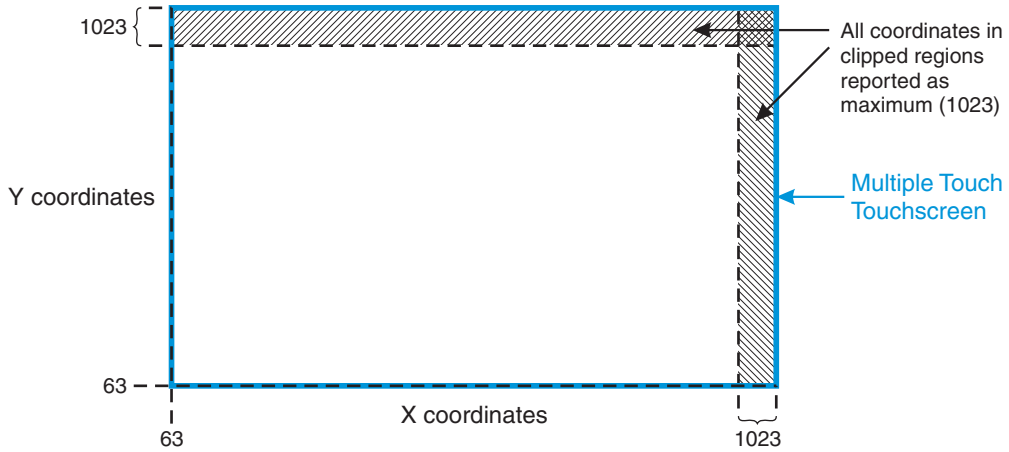
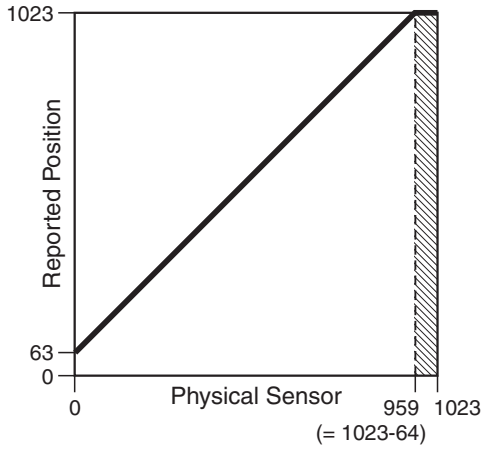
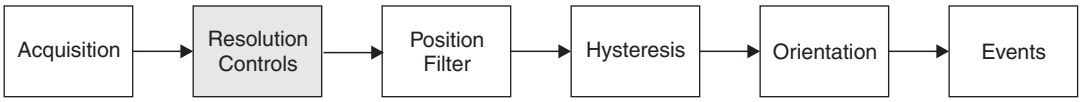


FIGURE 4-15: EXAMPLE RESOLUTION FIELDS – MAPPING OF REPORTED POSITIONS



These six resolution controls operate first on the positional data received from the touchscreen. The other touchscreen controls will be relative to the resolution set. For example, a hysteresis setting of 2 will have a different physical meaning on the same screen if the resolution is 200 or 2000. Figure 4-16 shows the processing order of all the touchscreen settings and shows how the resolution controls affect the other settings.

FIGURE 4-16: ORDER OF TOUCHSCREEN SETTINGS



XRANGE/YRANGE Range: 0 (1023; 10-bit resolution) to minimum and maximum as given on page 67

XLOCLIP/YLOCLIP/XHICLIP/YHICLIP Range: –128 to +127

XEDGECFG, DXXEDGECFG and YEDGECFG Fields

The XEDGECFG and YEDGECFG fields work with the XEDGEDIST and YEDGEDIST fields respectively to perform edge correction. They also work with the XEDGECFGHI/YEDGECFGHI and XEDGEDISTHI/YEDGEDISTHI fields, if separate edge correction is required (see "XEDGECFGHI, DXXEDGECFGHI and YEDGECFGHI Fields" and "XEDGEDISTHI, DXXEDGEDISTHI and YEDGEDISTHI Fields").

The DXXEDGECFG field overrides XEDGECFG when Dual X measurements are being performed. It also overrides XEDGECFGHI if XEDGECFGHI ENABLE = 1 and DXXEDGECFGHI ENABLE = 0 (that is, when separate edge correction is required for non Dual X measurements, but not for Dual X measurements).

The XEDGECFG, YEDGECFG and DXXEDGECFG fields operate on the physical ITO matrix of the touchscreen. The XEDGECFG, YEDGECFG and DXXEDGECFG fields are not affected by the logical orientation set by the orientation controls in the CFG1 field.

See Section 4.3.1.2 "Edge Correction" for more information on how the various Multiple Touch Touchscreen T100 fields provide edge correction and which other objects contribute to edge correction functionality.

CORRECTIONGRADIENT: Controls the correction gradient applied when the touch is between the edge of the sensor and the XEDGEDIST/YEDGEDIST coordinate when calculating the reported coordinate. It is used for correcting the nonlinearity seen at the edges of the sensor. A setting of 0 disables edge correction. A value of 1 to 63 sets the gradient value.

Range: 0 (disable), 1 to 63

Typical: 9

XEDGEDIST, DXXEDGEDIST and YEDGEDIST Fields

The XEDGEDIST and YEDGEDIST fields work with the XEDGECFG and YEDGECFG fields to perform edge correction. These fields specify the onset point for when the edge correction calculation is applied. The value is specified as if the screen was in 10-bit mode and should be set to half of the typical touch diameter in pixels. This improves linearity at the edge of the screen on certain touchscreen designs.

The DXXEDGEDIST field overrides XEDGEDIST when Dual X measurements are being performed. It also overrides XEDGEDISTHI if XEDGECFGHI ENABLE = 1 and DXXEDGECFGHI ENABLE = 0 (that is, when separate edge correction is required for non Dual X measurements, but not for Dual X measurements).

These fields also work with the XEDGECFGHI/YEDGECFGHI and XEDGEDISTHI/YEDGEDISTHI fields, if separate edge correction is required (see "XEDGECFGHI, DXXEDGECFGHI and YEDGECFGHI Fields" and "XEDGEDISTHI, DXXEDGEDISTHI and YEDGEDISTHI Fields").

The fields can also be overridden by other objects for certain types of touches.

The XEDGEDIST, YEDGEDIST and DXXEDGEDIST fields operate on the physical ITO matrix of the touchscreen. The XEDGEDIST, YEDGEDIST and DXXEDGEDIST fields are not affected by the logical orientation set by the orientation controls in the CFG1 field.

See Section 4.3.1.2 "Edge Correction" for more information on how the various Multiple Touch Touchscreen T100 fields provide edge correction and which other objects contribute to edge correction functionality.

For low-end positions, the corrected position is calculated as follows (for X):

position – (((XEDGEDIST – position) × XEDGECFG_CORRECTIONGRADIENT) / 16)

Under NDA
© 2021 Microchip Technology Inc. and its subsidiaries

OR (for Y):

position – (((YEDGEDIST – position) × YEDGECFG_CORRECTIONGRADIENT) / 16)

For high-end positions, the corrected position is calculated as follows (for X):

position + (((position – (1023 – XEDGEDIST)) × XEDGECFG_CORRECTIONGRADIENT) / 16)

OR (for Y):

position + (((position – (1023 – YEDGEDIST)) × YEDGECFG_CORRECTIONGRADIENT) / 16)

NOTE	These fields work at 10-bit resolution (that is, a touchscreen with a range of 1023).

Range: 0 to 255

Typical: Half expected touch diameter

GAIN Field

This field sets the gain of the analog circuits in front of the analog to digital converter (ADC) for mutual capacitance measurements.

The effective system gain is a complex combination of the GAIN value, the device gain calibration value and the operating conditions (for example, Cx value, use of XVDD or AVDD, single or dual X line pulsing, and so on). A Gain Calculator, where available, should therefore be used to explore different scenarios and determine a suitable gain value. Contact your Microchip representative for more information.

Range: 0 to 22

DXGAIN Field

This field specifies the Gain to be applied when operating in Dual X mode.

A value of 255 in the field turns on auto correction of the gain in Dual X mode. In Dual X two X lines are pulsed at the same time so there will be double the amount of charge is present. The purpose of the auto correction is to halve the system gain so it is at the correct level.

Range: 0 to 22, 255 (auto correction)

TCHTHR Field

This field specifies the Touch Threshold value in units of 8-bit deltas. This defines how much a node's touch delta must be to qualify as a potential touch detection. See Section 4.3.1 “Introduction” for more details.

Range: 0 to 255

Typical: 30 to 80

TCHHYST Field

This field controls the level of Touch Hysteresis applied to touch detections. See Section 4.3.1 “Introduction” for more details.

TCHHYST is capped internally to (TCHTHR × 0.75). A hysteresis greater than 75% of the Touch Threshold (TCHTHR) is therefore not possible. Note that, under normal circumstances, it is not recommended to set the hysteresis to more than 25% of the Touch Threshold, as this can cause potential problems during touch lift off, double taps, and noise handling.

A setting of 0 means that there is no hysteresis.

Range: 0 (off), 1 to (TCHTHR × 0.75)

INTTHR Field

This field configures the Internal Touch Tracking Threshold. This is the delta level (in 8-bit deltas) above which touches are internally tracked. The default value of 0 means (TCHTHR-TCHHYST)/2. See also “INTTHRHYST Field”.

See Section 4.3.1 “Introduction” for more details.

Range: 0 ((TCHTHR – TCHHYST)/2), 1 to (TCHTHR – TCHHYST)

NOISESF Field

This field specifies the Noise Scaling Factor to apply to the Noise Suppression T72 minimum threshold when searching for touches at the Internal Touch Tracking Threshold (INTTHR).

This feature is present to allow large touches to not be broken up by the Noise Suppression T72 minimum threshold processing in noisy conditions (that is, noise causing a dip in the middle of a touch). In this situation MRGTHR should also be increased to ensure that the large touches are not split up into multiple touches.

Note that NOISESF is not applied when determining whether a touch should be tracked or not. That is, the touch's peak delta must be above the Noise Suppression T72 minimum threshold to be tracked. It is only applied when determining whether a "region of touch" is actually one or multiple touches once the touchscreen has decided to track it.

A setting of 0 turns the feature off. This gives the behavior of using the full Noise Suppression T72 minimum threshold when searching for touches at the internal touch tracking threshold.

Settings of 1 to 255 result in a scaling factor of (NOISESF/256) being applied to the Noise Suppression T72 minimum threshold when searching for touches at the internal touch tracking threshold.

Range: 0 (no scaling factor), 1 to 255

CUTOFFTHR Field

This field specifies the Cut-off Threshold. This is the threshold on top of (TCHTHR – TCHHYST) above which touch signals are cut off. In other words the threshold is forced to be equal to (TCHTHR – TCHYST + CUTOFFTHR). The purpose of this feature is to help resolve position jitter on a big touch under extremely noisy conditions.

Note that the cut-off threshold is for mutual capacitance processing and is applied only to the mutual capacitance deltas. When using self capacitance touch as the primary measurement, the touches are tracked using self capacitance touch processing and thresholds.

The value is specified in 8-bit deltas, where a setting of 0 (default) means no cut-off threshold is required.

Range: 0 (OFF), 1 to 255

MRGTHR Field

The Merge Threshold (MRGTHR) setting allows the host to tune the amount of unevenness in a measured area that the device still considers as one individual touch. Beyond this threshold the device considers there to be multiple touches present on the sensor matrix.

This value may be used to allow the grouping of a large and uneven touch (such as a user's palm) into a single tracked touch. Larger values allow more uneven touches to be tracked as a single touch, but reduce the minimum separation of two true touches. This field is specified in units of signal delta.

Range: 0 to 255

Typical: 5

MRGTHRADJSTR

This field allows the MRGTHR to be increased dynamically when the capacitive coupling of the touchscreen is poor. The value controls the aggressiveness of the increase in merge threshold based on the estimated coupling level. This feature is useful for touchscreens under floating conditions, when large single touches may be broken up due to retransmission effects.

A larger value for MRGTHRADJSTR means that the merge threshold will increase faster as more antitouch becomes present within a touch region. The value of MRGTHRADJSTR is approximately equal to the applied MRGTHR when the touch/antitouch ratio is 1.

Note that if MRGTHRADJSTR is less than or equal to MRGTHR, this field has no effect.

Range: 0 to 255

Under NDA
© 2021 Microchip Technology Inc. and its subsidiaries

MRGHYST Field

The Merge Hysteresis (MRGHYST) setting is used in conjunction with the MRGTHR field. It provides a hysteresis to ensure that once two touches have merged (and a single touch is reported), the distance needed for separation before two touches are reported again is slightly larger than for convergence. This can help to stop oscillations between one and two touches being reported at the boundary of convergence. This field is specified in units of signal delta.

Range: 0 to 255

Typical: 5

DXTHRSF Field

This field specifies the Dual X Threshold Scaling Factor; that is, how much thresholds should be scaled by when dual X measurements are performed. The Threshold Scaling Factor is specified in units of 1/128, where the default value of 0 means 128.

Range: 0 (128), 1 to 255

TCHDIDOWN Field

This field configures the Touch Detect Integration (DI) Down setting. This, together with the NEXTTCHDI field (see "NEXTTCHDI Field"), is used to provide a debounce filter that is applied to each touch as it is being detected. This helps to suppress false detections caused by spurious events like electrical noise.

A per-touch counter is incremented each cycle that a touch is detected. When this counter reaches a preset limit the touch is finally declared to be present. If on any acquisition a delta is not seen to exceed the threshold level, the counter is cleared and the process has to start from the beginning. It takes TCHDIDOWN + 1 cycles from touchdown to when the first touch is actually reported via the $\overline{\text{CHG_I2C}}$ or $\overline{\text{CHG_SPI}}$ line pin, with a minimum time of 2 cycles. Once there is a touch detected, the limit is changed to (TCHDIDOWN +1 + NEXTTCHDI). This allows a shorter detection integration for the first touch detection.

When the DISTCHDIDOWN control bit is set to 1, the value of the TCHDIDOWN control is ignored and the DI process for a touchdown is not performed. The DISTCHDIDOWN control has no effect on the behavior of all the other Multiple Touch Touchscreen T100 DI related controls (for example, NEXTTCHDI).

TCHDIDOWN: Configures the Touch Detect Integration Down setting. The value is specified in fast acquisition cycles (with no sleep in between).

TCHDIDOWN Range: 0 to 63

DISTCHDIDOWN: Disables Detect Integration Touchdown. When this control is set to 1, the DI process is disabled for a touchdown and the value set in the TCHDIDOWN control is ignored.

IMPORTANT!	Touch DI must be enabled for UL/IEC 60730 Class B compliance. Therefore the DISTCHDIDOWN must be set to 0 (that is, touch DI must not be disabled).

By setting this control to 1, first touch latency will be reduced. This can, however, have a detrimental effect on the reliability of touchdown detection for all touch object instances other than Multiple Touch Touchscreen T100.

When this control is set to 1, it will also have an effect on the DI count used by all instances of other touch objects (for example, Key Array T15). Note, however, that Multiple Touch Touchscreen T100 touchdown DI behavior does not depend upon the state of the DISTCHDIDOWN control in any other touch object.

Microchip recommends setting DISTCHDIDOWN to 0.

NOTE	For self capacitance measurements, a mutual capacitance measurement is also performed to confirm the touchdown before reporting. This means that, if the DI process is disabled with Self Capacitance Touch set as the default Idle/Active measurement type, an extra cycle will still be seen. Disabling the DI process, therefore, will reduce touchdown latency to a single cycle only if the Mutual Capacitance measurement type is set as the default Idle measurement type (see Section A.2.1 "Default Measurements").

TCHDIUP Field

TCHDIUP: Configures the Touch Detect Integration Up setting. This, together with the NEXTTCHDI field (see "NEXTTCHDI Field"), is used to provide a debounce filter that is applied to each touch on release.

A per-touch counter is decremented each cycle that the delta does not exceed the threshold level. If the delta does exceed the threshold, it is incremented again. When the counter reaches zero, the touch is finally declared to be out of detect.

If there is only one touch in detect, it takes TCHDIUP + 1 cycle for the touch to cease to be reported. If there are two or more touches in detect, however, it takes (TCHDIUP + 1 + NEXTTCHDI) cycles for the touches to cease to be reported, with a minimum time of 1 cycle. This allows a shorter release integration if there is only one touch detection.

The value is specified in fast acquisition cycles (with no sleep in between).

TCHDIUP Range: 0 to 63

NEXTTCHDI Field

This field configures the Next Touch Detect Integration. This field, together with the TCHDIDOWN or TCHDIUP field, provides Touch Detect Integration. See "TCHDIDOWN Field" and "TCHDIUP Field" for more detail.

The value is specified in fast acquisition cycles (with no sleep in between).

Range: 0 to 63

CALCFG Field

P2PMUTUALEN: Enables P2P Mutual capacitance measurements. If this bit is set to 1, mutual capacitance measurements for this touchscreen will be made using a P2P set-up. Otherwise, standard mutual capacitance measurements will be employed.

NOTE	The Lens Bending T65 instance associated with mutual capacitance measurements for the Multiple Touch Touchscreen T100 object must be enabled if this control is set to 1.

JUMPLIMIT Field

This field specifies a limit on the allowed change in touch speed per cycle. If this limit is exceeded, the touch is classified as a new touch and all touchdown logic takes place on the new touch.

Note that low values can hinder fast movements, such as flicks.

The range for this field is 0 to 255, where 0 is off (no limit) and 1 to 255 is the allowed change in speed in units of the reported touchscreen resolution per millisecond.

Range: 0 (off, no limit), 1 to 255

MOVFILTER Field

This field controls the position filter that is used for filtering the reported touch position.

SPEEDRESP: Controls the system response to touch speed. Increasing this value results in the system response increasing with touch speed and less lag.

Figure 4-17 shows how varying the SPEEDRESP setting affects the level of smoothing applied. Increasing the value of SPEEDRESP results in the application of less smoothing for the same estimated touch velocity.

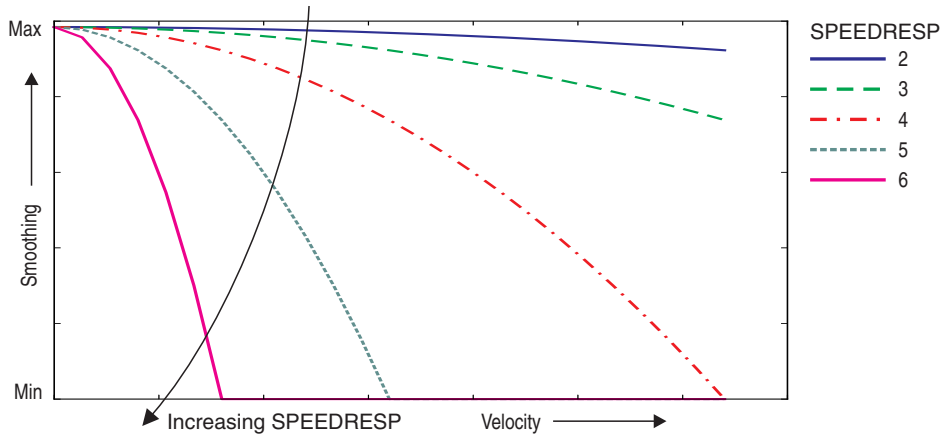
Note that Figure 4-17 is for a particular value of MOVSMOOTH. MOVSMOOTH sets the maximum amount of smoothing to apply (at very low speeds), while SPEEDRESP controls how quickly that smoothing falls off as movement speed increases.

The range for SPEEDRESP is 0 to 8, where a value of 0 means 4.

SPEEDRESP Range: 0 (4), 1 to 8.

Under NDA
© 2021 Microchip Technology Inc. and its subsidiaries

FIGURE 4-17: VARYING SPEEDRESP



MEDOFF: Disables the use of a median filter for pre-processing the position information. The median filter introduces a small delay to the system. If the system noise levels are low and the minimum delay is required, the median filter should be disabled.

DISABLE: Disables the movement filter.

MOVSMOOTH Field

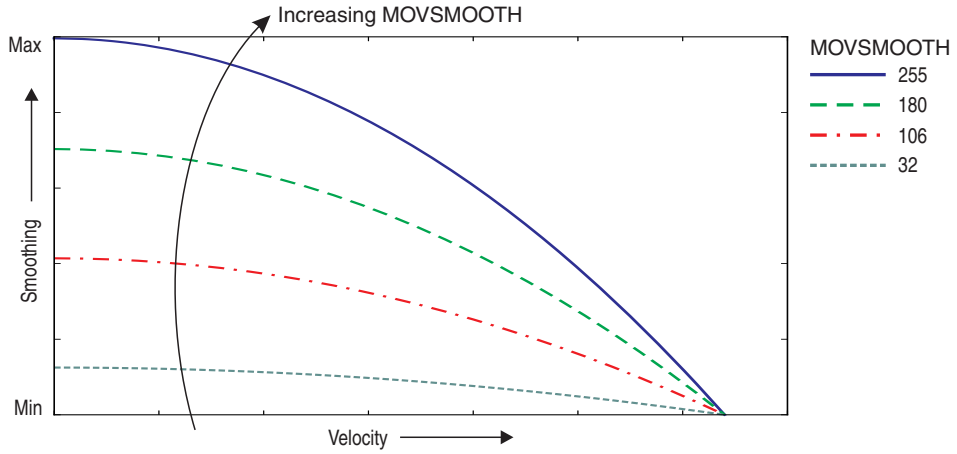
The Smooth Control limits the maximum amount of filtering that can be applied to the touch screen position (that is, the level of smoothing). Increasing the setting results in an increase in the maximum filtering. If the resolution of the touchscreen is modified this value may need adjusting to achieve the same level of performance.

Figure 4-18 shows the variation in the level of position smoothing applied with variation in speed for different settings of MOVSMOOTH.

Range: 0 (193),1 to 255

Typical: 65 to 193

FIGURE 4-18: VARYING MOVSMOOTH



MOVPRED Field

This field determines the level of prediction that the algorithm applies by controlling the amount of history used to estimate the position.

Increasing the MOVPRED setting increases the level of prediction at a given speed, as more data is used to predict the position.

The amount of prediction employed is dependent on the speed of travel. As the touch speed reduces the level of prediction is also reduced.

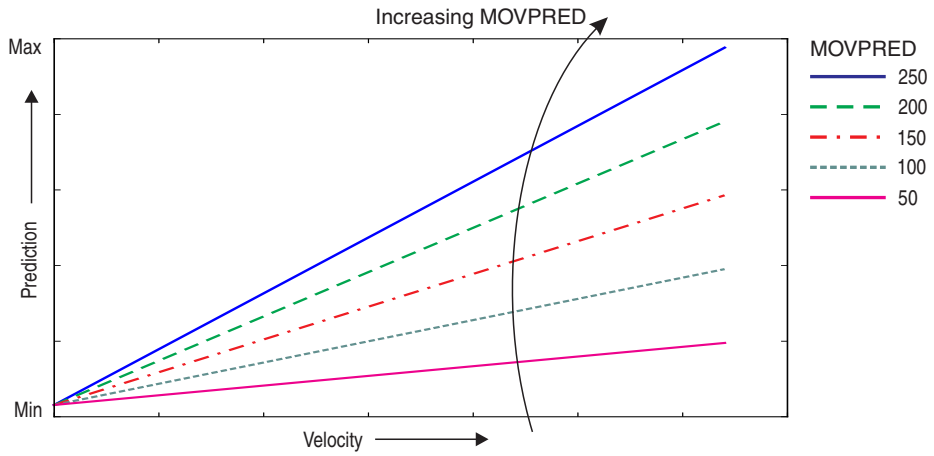
Note that setting too high a value may result in system overshooting a position if the touch changes direction abruptly. Values outside the typical range are not recommended.

Figure 4-19 shows the effect of varying the MOVPRED setting on the level of prediction applied dependent on the touch velocity.

Range: 0 (129), 1 to 255

Typical: 65 to 161

FIGURE 4-19: VARYING MOVPRED



MOVHYSTI and MOVHYSTN Fields

These fields are applied to the reported touch positions to provide simple filtering.

When the user first touches the touchscreen (that is, when the user's finger has just touched the touchscreen), the Initial Movement Hysteresis (MOVHYSTI) setting is applied to the reported position. Once the user's finger starts to move, the MOVHYSTI must be exceeded In a positive or negative X or Y direction (that is, in one of four directions) for position updates to be reported. This allows the host to differentiate between an intended drag and a simple press.

Once the MOVHYSTI limit has been exceeded, the Next Movement Hysteresis (MOVHYSTN) setting is applied to the calculated positions. The MOVHYSTN is applied separately to the X and Y axes and only in the opposite direction to the most recently reported movement in each axis. This allows small movements in the direction of travel to be reported while small backwards movements are not reported. A simple jitter filter, therefore, can be implemented to prevent unwanted movement reports due to small movement changes (jitter) as the position moves back and forth (providing that the movement is less than the hysteresis applied).

The MOVHYSTN setting is internally reduced while the touch is moving. This avoids "steps" when drawing, whilst still allowing jitter to be suppressed when the touch is stationary. The effect of this control is reduced when the touch is in motion. The extent of this reduction is determined by the MOVFILTER SPEEDRESP control (which can be overridden by MOVHYSTCFG SPEEDRESP) and the MOVHYSTCFG MINSPEED control. If the movement position filter is turned off, there will be no internal reduction effect on moving touches.

If MOVHYSTI is not required, set it to the same value as the MOVHYSTN setting.

The units for these fields are the least significant bits of position.

Range: 0 to 4095

MOVHYSTN Typical: 0 to 10

MOVHYSTI Typical: 50

Under NDA
© 2021 Microchip Technology Inc. and its subsidiaries

AMPLHYST Field

This field configures the hysteresis applied to the AMPL touch status reports (if enabled) that are sent to the host. This avoids excessive messaging to the host (see "TCHAUX Field"). The units are in 8-bit deltas.

Range: 0 to 255

SCRAREAHYST Field

This field configures the hysteresis (in nodes) applied to the TCHAREA, ATCHAREA and INTTHRAREA screen status reports (if enabled) that are sent to the host to avoid excessive messaging to the host (see "SCRAUX Field").

Range: 0 to 255

INTTHRHYST Field

This field controls the Internal Touch Tracking Threshold Hysteresis that is applied to the internally tracked touch detections. See Section 4.3.1 "Introduction" for more details.

INTTHRHYST is capped internally to (INTTHR × 0.75). A hysteresis greater than 75% of the Internal Touch Tracking Threshold (INTTHR) is therefore not possible. Note that under normal circumstances, it is not recommended to set the hysteresis to more than 25% of the Internal Touch Tracking Threshold, as this can cause potential problems during touch lift off, double taps, and noise handling.

A setting of 0 means that there is no hysteresis.

Range: 0 (off), 1 to (INTTHR × 0.75)

XEDGECFGHI, DXXEDGECFGHI and YEDGECFGHI Fields

The XEDGECFGHI and YEDGECFGHI fields work with the XEDGEDISTHI and YEDGEDISTHI fields respectively to perform edge correction on the high sense lines.

The DXXEDGECFGHI field overrides either XEDGECFGHI or XEDGECFG when Dual X measurements are being performed (depending on whether uniform or separate edge correction is chosen by the XEDGECFGHI ENABLE bit).

The XEDGECFGHI, YEDGECFGHI and DXXEDGECFGHI fields operate on the physical ITO matrix of the touchscreen. The XEDGECFGHI, YEDGECFGHI and DXXEDGECFGHI fields are not affected by the logical orientation set by the orientation controls in the CFG1 field.

The XEDGECFGHI, DXXEDGECFGHI and YEDGECFGHI fields can also be overridden by other objects for certain types of touches.

See Section 4.3.1.2 "Edge Correction" for more information on how the various Multiple Touch Touchscreen T100 fields provide edge correction and which other objects contribute to edge correction functionality.

CORRECTIONGRADIENT: Controls the correction gradient applied when the touch is between the edge of the sensor and the XEDGEDISTHI/YEDGEDISTHI coordinate when calculating the reported coordinate. It is used for correcting the nonlinearity seen at the edges of the sensor. A setting of 0 disables edge correction. A value of 1 to 63 sets the gradient value.

CORRECTIONGRADIENT Range: 0 (disable), 1 to 63

CORRECTIONGRADIENT Typical: 9

ENABLE: Enables separate edge correction.

If this bit is set to 1, XEDGECFGHI, DXXEDGECFGHI or YEDGECFGHI, as appropriate, can be used specify the separate edge correction for the high sense line edge of the Multiple Touch Touchscreen T100 X or Y axis, as appropriate (along with XEDGEDISTHI, DXXEDGEDISTHI or YEDGEDISTHI). This allows separate control of the edge correction that is applied to the top and bottom edges, so that sensors with different non-linear responses at the top and bottom of the X or Y axis can be corrected for.

If this bit is set to 0, uniform edge correction is used, as determined by the XEDGECFG, DXXEDGECFG and YEDGECFG fields.

XEDGEDISTHI, DXXEDGEDISTHI and YEDGEDISTHI Fields

The XEDGEDISTHI, DXXEDGEDISTHI and YEDGEDISTHI fields work with the XEDGECFGHI, DXEDGECFGHI and YEDGECFGHI fields to perform separate edge correction. These fields specify the onset point for when the edge correction calculation is applied. The value is specified as if the screen was in 10-bit mode and should be set to half of the typical touch diameter in pixels. This improves linearity at the edge of the screen on certain touchscreen designs.

The DXXEDGEDISTHI field overrides either XEDGEDISTHI or XEDGEDIST when Dual X measurements are being performed (depending on whether uniform or separate edge correction is chosen by the XEDGECFGHI ENABLE bit; see "XEDGECFGHI, DXXEDGECFGHI and YEDGECFGHI Fields"). That is, it overrides XEDGEDIST if XEDGECFGHI ENABLE is set to 0, and it overrides XEDGEDISTHI if XEDGECFGHI ENABLE is set to 1.

The XEDGEDISTHI, YEDGEDISTHI and DXXEDGEDISTHI fields operate on the physical ITO matrix of the touchscreen. The XEDGEDISTHI, YEDGEDISTHI and DXXEDGEDISTHI fields are not affected by the logical orientation set by the orientation controls in the CFG1 field.

See Section 4.3.1.2 "Edge Correction" for more information on how the various Multiple Touch Touchscreen T100 fields provide edge correction and which other objects contribute to edge correction functionality.

With separate edge correction, the corrected position for high-end positions is calculated as follows (for X):

position + (((position – (1023 – XEDGEDISTHI)) × XEDGECFGHI_CORRECTIONGRADIENT) / 16)

OR (for Y):

position + (((position – (1023 – YEDGEDISTHI)) × YEDGECFGHI_CORRECTIONGRADIENT) / 16)

Note that these fields work at 10-bit resolution (that is, a touchscreen range of 1023). See "XEDGEDIST, DXXEDGEDIST and YEDGEDIST Fields" for the calculated correction on low-end positions).

Range: 0 to 255

Typical: Half expected touch diameter

CFG2 Field

Configuration Controls

CONFTHR: Specifies a debounce filter for the TCHSTATUS TYPE of a reporting touch changing to FINGER (see "TCHSTATUS Field"). This delays a contacting touch class from being reported as a finger under very noisy conditions, which would then block other class transitions (such as PASSIVE STYLUS).

The threshold is specified in acquisition cycles. Higher values will introduce higher delays between the transitions to finger touch type and may lead to missing touches during finger taps.

CONFTHR Range: 0 to 7

DISESBFSREJ: Disables Bad First Sample Rejection logic for the edge snapping algorithm. Bad first sample rejection increases the effective TCHDIDOWN setting to at least 2 for those touches that appear to have a bad first sample.

If this control is set to 0, bad first sample rejection logic for edge snapping is enabled. If this control is set to 1, bad first sample rejection logic for edge snapping is disabled.

If bad first sample rejection is enabled (DISESBFSREJ = 0), the region of the rejection logic can be limited so that it occurs only along the edges or in the corners of the sensor. See "XBFSREJRANGE and YBFSREJRANGE Fields" for more details.

Note that bad first sample rejection logic takes effect only if touchdown edge snapping is enabled (that is, EDGESNPCFGIN ENABLE is set to 1). See "EDGESNPCFGIN Field" for more details on touchdown edge snapping.

Debug Controls

The following controls allow different debug data types to be enabled or disabled for use with the hardware debug interface. If any of the controls are set to 1, the appropriate debug data is output for the debug region within this Multiple Touch Touchscreen T100 instance. See "DBGXOFFSET and DBGYOFFSET Fields" and "DBGXSIZE and DBGYSIZE Fields") for details on how to define the debug data region.

If debug data for the entire sensor matrix extent is required, the Command Processor T6 debug data output feature can be used instead (see "DEBUGCTRL Field").

Under NDA
© 2021 Microchip Technology Inc. and its subsidiaries

Note that the Command Processor T6 DEBUGCTRL2 DBGOBJMODEEN bit must be set to 1 for the following controls to work. See "DEBUGCTRL2 Field" for more information on the debug data for the hardware debug interface.

DBGDELTAS8EN: Enables the output of signed 8-bit mutual capacitance deltas debug data for this touchscreen instance. The output of the debug data is enabled if this control is set to 1 and disabled if it is set to 0.

DBGDELTASEN: Enables the output of signed 16-bit mutual capacitance deltas debug data for this touchscreen instance. The output of the debug data is enabled if this control is set to 1 and disabled if it is set to 0.

DBGREFSEN: Enables the output of signed 16-bit mutual capacitance references debug data for this touchscreen instance. The output of the debug data is enabled if this control is set to 1 and disabled if it is set to 0.

DBGSIGSEN: Enables the output of signed 16-bit mutual capacitance signals debug data for this touchscreen instance. The output of the debug data is enabled if this control is set to 1 and disabled if it is set to 0.

NOTE	If Dual X mode is active, the data for the highest X line measured with Dual X (that is, the highest X line used by Multiple Touch Touchscreen T100) is invalid and should be ignored.

MOVHYSTCFG Field

This field provides additional control over the scaling of Next Movement Hysteresis (see "MOVHYSTI and MOVHYSTN Fields") with touch speed. SPEEDRESP controls the rate of scaling and MINSPEED controls the speed at which the scaling starts.

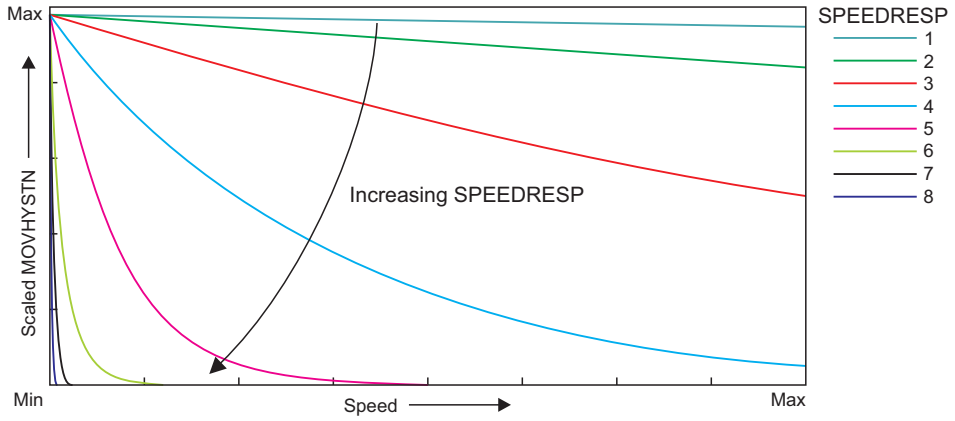
SPEEDRESP: Configures the Speed Response. This controls the rate at which movement hysteresis is reduced with increasing touch speed after the minimum speed (specified by MINSPEED) has been reached. Increasing the value of SPEEDRESP increases the rate at which Movement Hysteresis is reduced with increasing speed (see Figure 4-20).

By using this control with MOVHYSTN, the touchscreen can be configured to apply a large movement hysteresis to slow moving touches but a small movement hysteresis to faster moving touches.

If the movement filter is disabled this control has no effect.

The value for SPEEDRESP is in the range 0 to 8, where the default value of 0 means use MOVFILTER SPEEDRESP (see "MOVFILTER Field"). Values above 8 are capped to 8.

FIGURE 4-20: EFFECT OF SPEEDRESP ON MOVEMENT HYSTERESIS



SPEEDRESP Range: 0 (Use MOVFILTER SPEEDRESP), 1 to 8

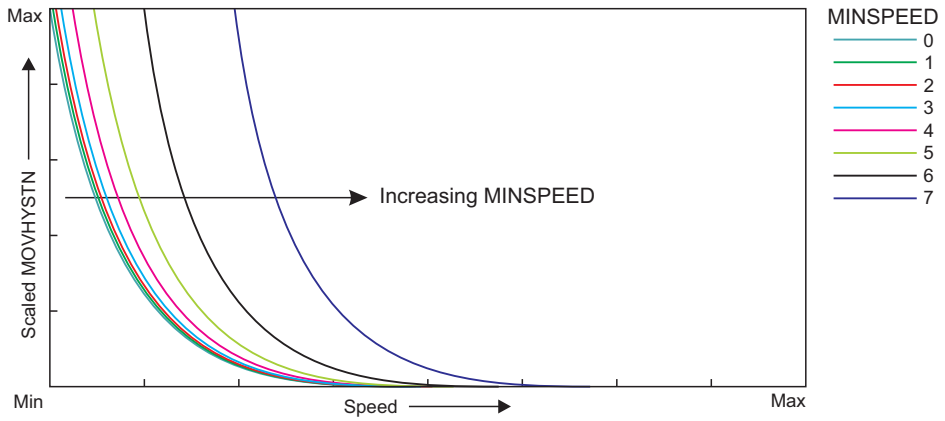
SPEEDRESP Typical: 4

MINSPEED: This control adjusts the minimum touch speed that must be reached before movement hysteresis scaling occurs. Once the minimum speed has been reached, any further increase in touch speed will result in the Movement Hysteresis being reduced at the rate determined by SPEEDRESP (see Figure 4-21). Increasing the value of MINSPEED increases the minimum touch speed in an exponential fashion, such that increasing MINSPEED by 1 doubles the required minimum touch speed.

This control allows the system to be configured so that the full value of MOVHYSTN is used even when position jitter prevents the touch speed reaching 0, while retaining the rate of MOVHYSTN reduction with speed.

If the movement filter is disabled this control has no effect.

FIGURE 4-21: EFFECT OF MINSPEED ON MOVEMENT HYSTERESIS



MINSPEED Range: 0 (Disabled), 1 to 7

MINSPEED Typical: 2

AMPLCOEFF Field

This field specifies the Scaled Amplitude Coefficient. Together with the AMPLOFFSET field, this configures the scaling applied to the AMPL touch status reports (if enabled) that are sent to the host (see "TCHAUX Field"). The scaling is as follows:

$$AMPL_{scaled} = (AMPLCOEFF \times AMPL_{original}) + AMPLOFFSET$$

where $AMPL_{scaled}$ has a minimum value of 1 and is saturated to a maximum of 255.

AMPLCOEFF is the multiplier for the amplitude and is a 5.3 format scale factor, giving a range of 0.125 to 31.875 in units of 0.125 (1/8). The default value of 0 means 8 (= 1.0).

The scaling has no effect on internal touch processing.

Range: 0 (8), 1 to 255

AMPLOFFSET Field

This field specifies the Scaled Amplitude Offset. This is the offset for the touch amplitude. See "AMPLCOEFF Field" for more details.

Range: –128 to 127

JUMPLIMITMOV Field

This field configures the Jump Limit for Moving Touches.

This is the limit on the allowed change in touch speed per cycle of moving touches (that is, touches that have exceeded JLMMOVTHR; see "JLMMOVTHR Field"). This field operates in a similar way to the JUMPLIMIT field (see "JUMPLIMIT Field"), but it is only applied to touches the movement of which has exceeded JLMMOVTHR.

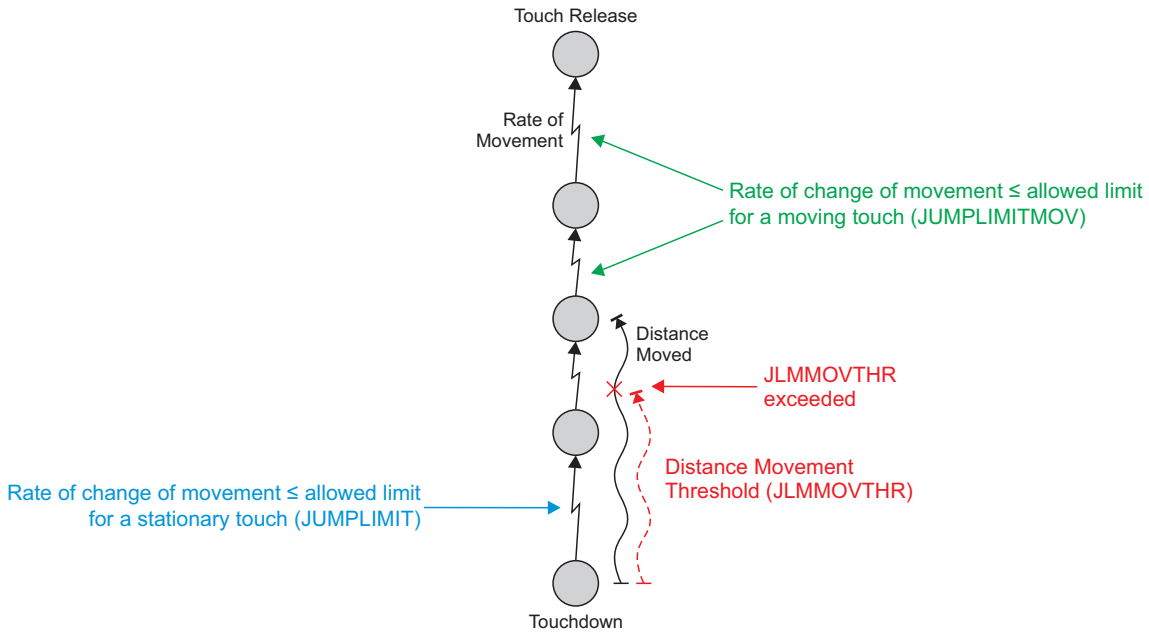
JUMPLIMIT and JLMMOVTHR must both be configured to non zero values for the JUMPLIMITMOV field to have an effect.

The range for this field is 0 to 255, where 0 means no limit and 1 to 255 is the allowed change in speed. The value is specified in units of the reported touchscreen resolution per millisecond.

Range: 0 (no limit), 1 to 255

Under NDA
© 2021 Microchip Technology Inc. and its subsidiaries

FIGURE 4-22: JUMP LIMIT THRESHOLDS FOR MOVING TOUCHES



JLMMOVTHR Field

This field configures the Movement Threshold for the Jump Limit for Moving Touches. This is the minimum distance a touch must move before JUMPLIMITMOV is used instead of JUMPLIMIT (see "JUMPLIMIT Field"). Note that the distance is along the actual path of the movement, and not a straight line.

The range for this field is 0 to 4095, where 0 disables the use of JUMPLIMITMOV and 1 to 4095 specifies a distance in reported position units.

Range: 0 (disabled), 1 to 4095

JLMMOVINTTHR Field

This field configures the Movement Integration Threshold for the Jump Limit for Moving Touches. This is the minimum touch movement that can count toward the total movement against which JLMMOVTHR is compared. By setting this control to a value larger than the stationary jitter radius, this value can be used to prevent the jitter on taps and stationary touches from being accumulated and causing JLMMOVTHR to be exceeded.

The range for this field is 0 to 255, specified in reported position units.

Range: 0 to 255

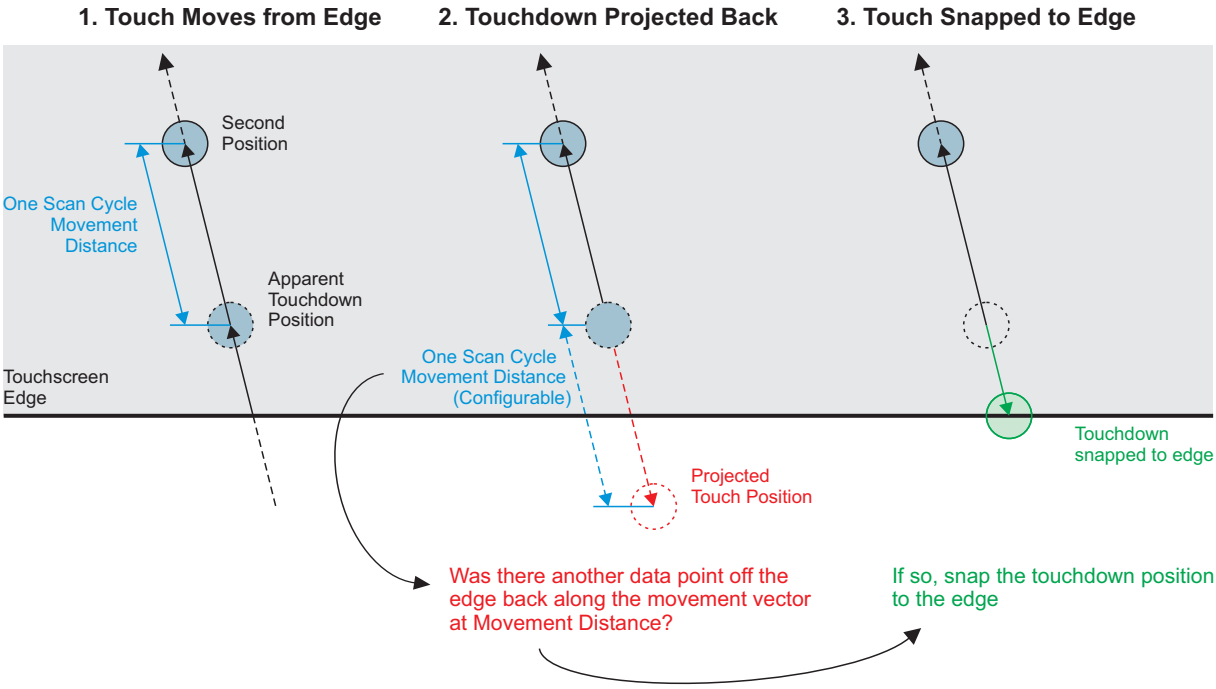
EDGESNPCFGIN Field

The EDGESNPCFGIN field configures the touchdown Edge Snap algorithm for when a touch comes into the touchscreen. This allows a swipe that starts off the edge of the active touchscreen area to be reported as entering the touchscreen at the edge. Such swipes are commonly used by an operating system to perform special functions, such as bringing up hidden menus. Typically, the operating system recognizes this swipe gesture only if the first reported touchdown point is at, or very close to, the edge of the sensor and so the Edge Snap algorithm ensures that this is the case.

The Edge Snap algorithm calculates the movement speed and direction of a swipe and then extrapolates its position in the cycle before it was first detected. If the swipe has crossed the edge of the sensor, the reported touchdown position can then be snapped to the edge of the sensor.

Figure 4-23 shows an example of how the Edge Snap algorithm operates for an edge swipe in.

FIGURE 4-23: EDGE SNAP IN – EXAMPLE



See also the DISESBFSREJ control in the "CFG2 Field".

ENABLE: Enables Edge Snap on touch swipes into the Touchscreen. If this bit is set to 1, the edge snapping algorithm is enabled.

RATIO: Specifies the Edge Snap Ratio. This allows rescaling of the Edge Snap Ratio that is used to control the sensitivity of the edge snapping algorithm. Reducing the Edge Snap Ratio has the effect of reducing sensitivity and will make an edge snap less likely to occur. For example, this reduces the chance of the user accidentally triggering a pull-in side menu when just touching near the edge of the touchscreen.

The Edge Snap Ratio is expressed in $^{1}/_{64}$ units, where the default value of 0 means 64 and gives normal sensitivity. For example, if RATIO = 80, the rescaled ratio will be 1.25. Values between 1 and 15 are saturated to 16.

RATIO Range: 0 (64), 16 to 127

EDGESNPCFGOUT Field

The EDGESNPCFGOUT field configures the release Edge Snap algorithm for when a touch goes out of the touchscreen.

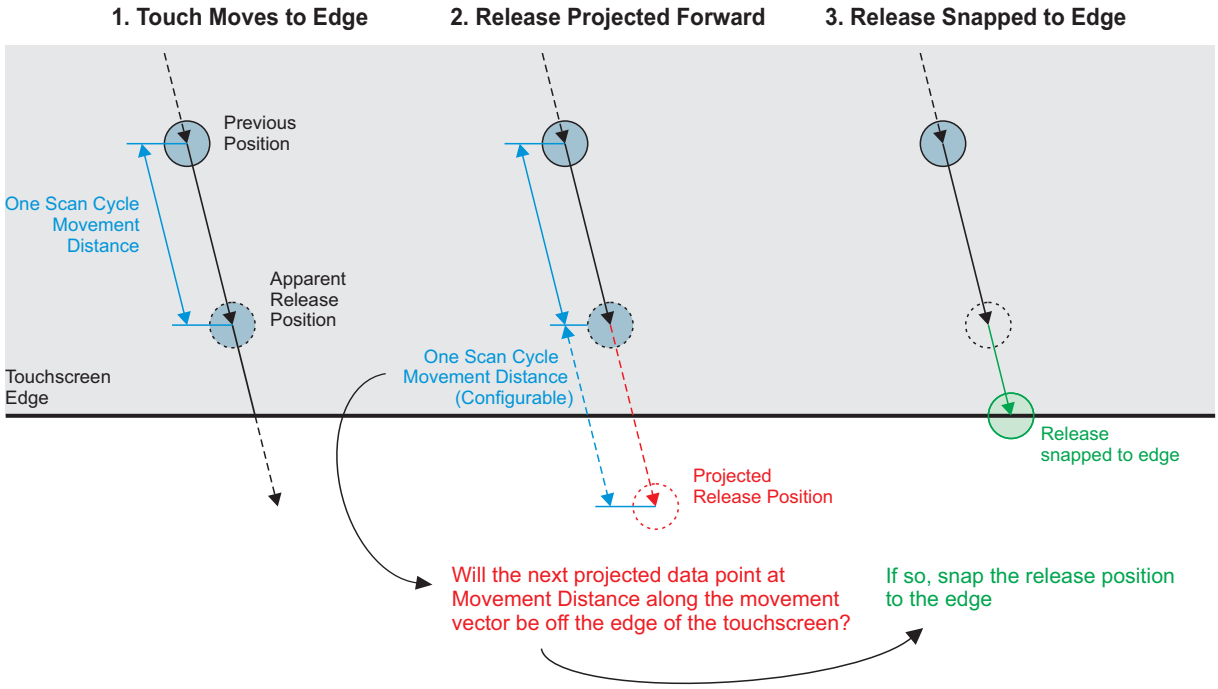
This allows a swipe that ends off the edge of the active touchscreen area to be reported as exiting the touchscreen at the edge. Such swipes are commonly used by an operating system to perform special functions, such as bringing up hidden menus. Typically, the operating system recognizes this swipe gesture only if the final reported release point is at, or very close to, the edge of the sensor and so the Edge Snap algorithm ensures that this is the case.

The Edge Snap algorithm calculates the movement speed and direction of a swipe and then extrapolates its position in the cycle after it was finally released. If the swipe will cross the edge of the sensor, the reported release position can then be snapped to the edge of the sensor.

Figure 4-24 shows an example of how the Edge Snap algorithm operates for an edge swipe out.

Under NDA
© 2021 Microchip Technology Inc. and its subsidiaries

FIGURE 4-24: EDGE SNAP OUT – EXAMPLE



ENABLE: Enables Edge Snap on touch swipes out of the Touchscreen. If this bit is set to 1, the edge snapping algorithm is enabled.

RATIO: Specifies the Edge Snap Ratio. This allows rescaling of the Edge Snap Ratio that is used to control the sensitivity of the edge snapping algorithm. Reducing the Edge Snap Ratio has the effect of reducing sensitivity and will make an edge snap less likely to occur. For example, this reduces the chance of the user accidentally triggering a pull-in side menu when just touching near the edge of the touchscreen.

The Edge Snap Ratio is expressed in $^{1}/_{64}$ units, where the default value of 0 means 64 and gives normal sensitivity. For example, if RATIO = 80, the rescaled ratio will be 1.25. Values between 1 and 15 are saturated to 16.

RATIO Range: 0 (64), 16 to 127

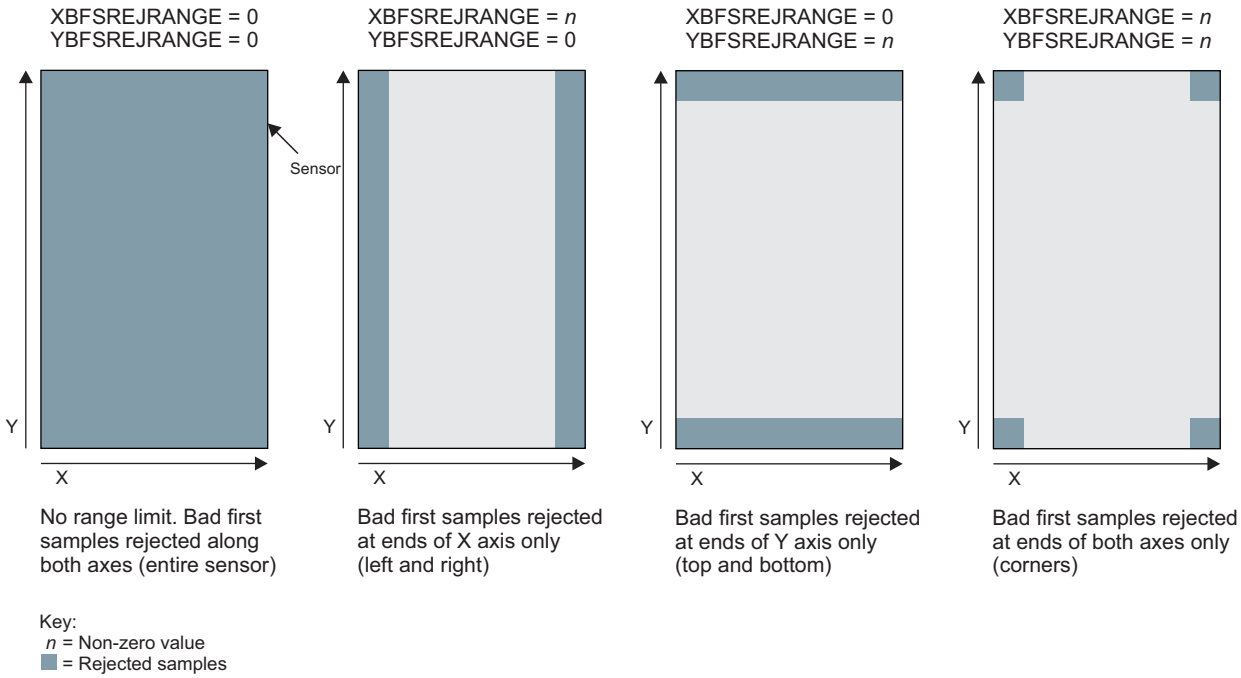
XBFSREJRANGE and YBFSREJRANGE Fields

These fields limit the region of the Bad First Sample Rejection (see the DISESBFSREJ control in the "CFG2 Field"). The XBFSREJRANGE field limits the bad first sample rejection along the X axis and the YBFSREJRANGE limits the bad first sample rejection along the Y axis. Depending on whether one or both fields are set, the bad first samples are rejected along an edge or in a corner only. If neither field specifies a limit, then bad first samples are rejected across the entire sensor (that is, along both axes with no limits). See Figure 4-25 for details.

XBFSREJRANGE and YBFSREJRANGE are expressed in units of 1/128 of axis length from each end of the axis. For example, if XBFSREJRANGE = 32, the range will be 32/128 (0.25) of the length of the X axis from each end of the X axis. If XBFSREJRANGE or YBFSREJRANGE is set to zero, there is no limit along the corresponding axis.

Range: 0 (not limited), 1 to 63

FIGURE 4-25: XBFSREJRANGE AND YBFSREJRANGE FIELDS



DBGXOFFSET and DBGYOFFSET Fields

These fields specify the start of a Debug Region within this Multiple Touch Touchscreen T100 instance as X and Y line offsets from the origin of this instance. This limits the output of sensor debug measurement data (over the hardware debug interface) from this Multiple Touch Touchscreen T100 instance to this region. The size of the region is specified by the DBGXSIZE and DBGYSIZE fields.

The type of data that is output for the defined region is determined by the debug controls within the CFG2 field (see "CFG2 Field" for details).

If the starting position offset exceeds the Multiple Touch Touchscreen T100 extents (that is, the starting point is off the touchscreen), then no debug data will be sent. However, the region may extend off the touchscreen and the debug data will be clipped to the Multiple Touch Touchscreen T100 extents.

DBGXOFFSET Range: 0 to (XSIZE–1)

DBGYOFFSET Range: 0 to (YSIZE–1)

DBGXSIZE and DBGYSIZE Fields

These fields specify the size of a Debug Region (in terms of X and Y lines) within this Multiple Touch Touchscreen T100 instance so that the output of sensor debug measurement data (over the hardware debug interface) can be limited to this region. The origin of this region is specified by the DBGXOFFSET and DBGYOFFSET fields.

The type of data that is output for the defined region is determined by the debug controls within the CFG2 field (see "CFG2 Field" for details).

The range of these two fields is 0 to XSIZE/YSIZE, where the default of 0 means use XSIZE or YSIZE as appropriate. If DBGXSIZE or DBGYSIZE exceeds the Multiple Touch Touchscreen T100 extent when added to DBGXOFFSET or DBGYOFFSET, the debug data region is clipped to the Multiple Touch Touchscreen T100 extent on that axis.

DBGXSIZE Range: 0 (XSIZE), 1 to XSIZE

DBGYSIZE Range: 0 (YSIZE), 1 to YSIZE

NOTE	If Dual X mode is active and the Debug Region includes the top X line covered by, the data for the top X line is invalid and should be ignored.

Under NDA
© 2021 Microchip Technology Inc. and its subsidiaries

4.3.3 CONFIGURATION CHECKS

The Multiple Touch Touchscreen T100 object causes a configuration check to be performed in the following circumstances:

- When the object is enabled (that is, the ENABLE bit is set in the CTRL field)
- If the object is enabled, when certain fields are changed

In addition, some fields will cause an automatic recalibration to be performed.

A configuration check may determine that a configuration error has occurred (for example, if a setting is set outside of its allowed range or a conflict has occurred between two settings). This is signaled to the host (see Section 3.3 "Command Processor T6 Object"). The device halts until the error has been corrected. To fix the error, check that all the object settings are within their allowed limits, as stated in the field descriptions.

The XORIGIN/YORIGIN and XSIZE/YSIZE settings should be checked to ensure that:

- Part of the object is not placed off the edge of the sensor matrix. The matrix size is specified in the Information Block (see Section 1.3 "Information Block")
- The object does not occupy the same X or Y line as another enabled touch object
- The XORIGIN and YORIGIN fields are set correctly (see "XORIGIN and YORIGIN Fields")
- XSIZE and YSIZE are set correctly (see "XSIZE and YSIZE Fields")

TABLE 4-24: CONFIGURATION CHECKS FOR MULTIPLE TOUCH TOUCHSCREEN T100 (TOUCH_MULTITOUCHSCREEN_T100)

Field	Changing The Field Causes...		Effect of Configuration Checks On Field
	Configuration Check	Automatic Recalibration	
CTRL	Yes (1)	Yes (1) (2)	None
CFG1	No	No	None
SCRAUX	No	No	None
TCHAUX	No	No	None
TCHEVENTCFG	No	No	None
AKSCFG	No	No	None
NUMTCH	No	No	None
XYCFG	Yes	No	None
XORIGIN	Yes	Yes	Error if out of range
XSIZE	Yes	Yes	Error if out of range
XPITCH	Yes	No	None
XLOCLIP	Yes	No	Error if out of range
XHICLIP	Yes	No	Error if out of range
XRANGE	Yes	No	Error if out of range
XEDGECFG	No	No	None
XEDGEDIST	No	No	None
DXXEDGECFG	No	No	None
DXXEDGEDIST	No	No	None
YORIGIN	Yes	Yes	Error if out of range
YSIZE	Yes	Yes	Error if out of range
YPITCH	Yes	No	None
YLOCLIP	Yes	No	Error if out of range
YHICLIP	Yes	No	Error if out of range
YRANGE	Yes	No	Error if out of range
YEDGECFG	No	No	None

TABLE 4-24: CONFIGURATION CHECKS FOR MULTIPLE TOUCH TOUCHSCREEN T100 (TOUCH_MULTITOUCHSCREEN_T100) (CONTINUED)

Field	Changing The Field Causes...		Effect of Configuration Checks On Field
	Configuration Check	Automatic Recalibration	
YEDGEDIST	No	No	None
GAIN	Yes	Yes	None
DXGAIN	Yes	Yes	None
TCHTHR	Yes	No	None
TCHHYST	Yes	No	None
INTTHR	Yes	No	None
NOISESF	No	No	None
CUTOFFTHR	No	No	None
MRGTHR	Yes	No	None
MRGTHRADJSTR	Yes	No	None
MRGHYST	No	No	None
DXTHRSF	No	No	None
TCHDIDOWN	No	No	None
TCHDIUP	No	No	None
NEXTTCHDI	No	No	None
CALCFG	No	Yes	None
JUMPLIMIT	Yes	No	None
MOVFILTER	Yes	No	None
MOVSMOOTH	Yes	No	None
MOVPRED	Yes	No	None
MOVHYSTI	No	No	None
MOVHYSTN	No	No	None
AMPLHYST	No	No	None
SCRAREAHYST	No	No	None
INTTHRHYST	Yes	No	None
XEDGECFGHI	No	No	None
XEDGEDISTHI	No	No	None
DXXEDGECFGHI	No	No	None
DXXEDGEDISTHI	No	No	None
YEDGECFGHI	No	No	None
YEDGEDISTHI	No	No	None
CFG2	No	No	None
MOVHYSTCFG	No	No	None
AMPLCOEFF	No	No	None
AMPLOFFSET	No	No	None
JUMPLIMITMOV	Yes	No	None
JLMMOVTHR	No	No	None
JLMMOVINTTHR	No	No	None
EDGESNPCFGIN	No	No	None

Under NDA
© 2021 Microchip Technology Inc. and its subsidiaries

TABLE 4-24: CONFIGURATION CHECKS FOR MULTIPLE TOUCH TOUCHSCREEN T100 (TOUCH_MULTITOUCHSCREEN_T100) (CONTINUED)

Field	Changing The Field Causes...		Effect of Configuration Checks On Field
	Configuration Check	Automatic Recalibration	
EDGESNPCFGOUT	No	No	None
XBFSREJRANGE	No	No	None
YBFSREJRANGE	No	No	None
DBGXOFFSET	No	No	None
DBGXSIZE	No	No	None
DBGYOFFSET	No	No	None
DBGYSIZE	No	No	None

Note 1: If the ENABLE bit is toggled on or off.
2: If the ENABLE bit is cleared and *all of* the touch objects are now disabled, no calibration takes place.

4.3.4 MESSAGES

The Multiple Touch Touchscreen T100 object reports the following screen and touch status information:

- Screen status information, such as the number of nodes affected and the number of reported touches
- Finger touches detected by the Multiple Touch Touchscreen T100 object
- Grip suppression by the Grip Suppression T40 object
- Screen suppression by the Touch Suppression T42 object
- Passive stylus touches detected by the Passive Stylus T47 object
- Glove touches detected by the Glove Detection T78 object

NOTE To be able to track and report touches in self capacitance measurements, the Multiple Touch Touchscreen T100 object needs to have data for both axes (see Self Capacitance Global Configuration T109 “CTRL Field” and Appendix A “Measurement Processing”).

There are three types of message reported from a Multiple Touch Touchscreen T100. The first report ID from a Multiple Touch Touchscreen T100 is for the global screen status messages that indicate the state of the touchscreen. The second report ID is reserved. Subsequent report IDs are for each of the reportable tracked touches.

4.3.4.1 First Report ID – Screen Status Messages

The first report ID from the Multiple Touch Touchscreen T100 object is used to output screen status messages.

Table 4-25 shows the message data for the first report ID.

TABLE 4-25: MESSAGE DATA FOR MULTIPLE TOUCH TOUCHSCREEN T100 (TOUCH_MULTITOUCHSCREEN_T100)

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	SCRSTATUS	DETECT	SUP	Reserved					
2–9	AUXDATA[]	Auxiliary data							

A screen status message is triggered when one of the reported data fields reaches a particular trigger value, as summarized in Table 4-26.

TABLE 4-26: MESSAGE TRIGGERS – SCREEN STATUS MESSAGES

Data Field		Effect on Message Generation
SCRSTATUS Bits	AUXDATA[] Fields	
SUP		Message generated if value changes (that is, if full screen suppression is in effect or no longer in effect)
DETECT		Message generated if value changes (that is, if there are any touches reporting or not).

TABLE 4-26: MESSAGE TRIGGERS – SCREEN STATUS MESSAGES

Data Field		Effect on Message Generation
SCRSTATUS Bits	AUXDATA[] Fields	
	NUMRPTTCH	Message generated if value changes
	TCHAREA	Message generated if value > SCRAREAHYST
	ATCHAREA	Message generated if value > SCRAREAHYST
	INTTHRAREA	Message generated if value > SCRAREAHYST

SCRSTATUS Field

This field indicates the status of the touchscreen.

DETECT: This bit is set to indicate that there is at least one reporting touch on the touchscreen.

SUP: This bit is set to indicate that full screen suppression is in effect.

AUXDATA[] Fields

These fields report user-configurable information about the screen. The items to report, and thereby the content of the AUXDATA[] fields, is determined by which bits are enabled in the SCRAUX configuration field (see "SCRAUX Field").

4.3.4.2 Second Report ID – Reserved

The second report ID is reserved for future use.

TABLE 4-27: MESSAGE DATA FOR MULTIPLE TOUCH TOUCHSCREEN T100 (TOUCH_MULTITOUCHSCREEN_T100)

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1 – 9	Reserved	Reserved							

4.3.4.3 Subsequent Report IDs – Touch Status Messages

Subsequent report IDs are used to output the touch status messages. There are 16 report IDs, one for each of the 16 touches. Table 4-28 shows the message data for the Multiple Touch Touchscreen T100 object for these report IDs.

TABLE 4-28: MESSAGE DATA FOR MULTIPLE TOUCH TOUCHSCREEN T100 (TOUCH_MULTITOUCHSCREEN_T100)

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	TCHSTATUS	DETECT	TYPE			EVENT			
2	XPOS	X position LSByte							
3		X position MSByte							
4	YPOS	Y position LSByte							
5		Y position MSByte							
6 – 9	AUXDATA[]	Auxiliary data							

A touch status message is triggered when one of the reported data fields reaches a particular trigger value or the touch type changes, as summarized in Table 4-29.

TABLE 4-29: MESSAGE TRIGGERS – TOUCH STATUS MESSAGES

Data Field		Effect on Message Generation
TCHSTATUS Bits	AUXDATA[] Fields	
EVENT		Message generated if there is an event
TYPE		Message generated if TYPE changes
	VECT	No effect on message generation
	AMPL	Message generated if delta change > AMPLHYST

Under NDA
© 2021 Microchip Technology Inc. and its subsidiaries

TABLE 4-29: MESSAGE TRIGGERS – TOUCH STATUS MESSAGES

Data Field		
TCHSTATUS Bits	**AUXDATA[] Fields**	**Effect on Message Generation**
	AREA	No effect on message generation
	HW	No effect on message generation
	PEAK	No effect on message generation
	AREAHW	No effect on message generation

NOTE

- The RPTEACHCYCLE bit in the CFG1 field can be used to generate a touch status message for each touch that is reporting, irrespective of whether any triggering events have happened. Note that this applies to all types of touches. See "CFG1 Field" for more information.
- Only a single touch will be reported if a touch is present in a touchscreen area that the Retransmission Compensation T80 object has detected as having moisture.

TCHSTATUS Field

This field indicates the status of the touch.

EVENT: These bits indicate the event that has just occurred for this finger touch (see Table 4-30).

TABLE 4-30: EVENT CODES

Event	Name	Description
0	NO EVENT	No specific event has occurred
1	MOVE	The touch position has changed
2	UNSUP	The touch has just been unsuppressed by the touch suppression features of other objects
3	SUP	The touch has been suppressed by the touch suppression features of other objects
4	DOWN	The touch has just come within range of the sensor
5	UP	The touch has just left the range of the sensor
6	UNSUPSUP	Both UNSUP and SUP events have occurred (in either order)
7	UNSUPUP	Both UNSUP and UP events have occurred (in either order)
8	DOWNSUP	Both DOWN and SUP events have occurred (in either order)
9	DOWNUP	Both DOWN and UP events have occurred (in either order)

TYPE: These bits indicates the type of touch that is being reported (see Table 4-31). A touch status message is generated if the type changes.

TABLE 4-31: TOUCH TYPES

Type	Name	Description
0	RESERVED	Reserved for future use.
1	FINGER	The touch is considered to be a finger that is contacting the screen.
2	PASSIVE STYLUS	The touch is a passive stylus.
5	GLOVE	The touch is a glove touch. Note that touches will be reported as GLOVE events only if the CTRL GLOVERPTEN bit in the corresponding instance of the Glove Detection T78 object is set to 1; otherwise they will be reported as FINGER events.

TABLE 4-31: TOUCH TYPES (CONTINUED)

Type	Name	Description
6	LARGE TOUCH	The touch is a suppressed large touch. When CFG SUPTCHRPTEN bit is set in the Touch Suppression T42 object, a touch that would normally be suppressed is reported. When SUPTCHRPTEN is cleared, the suppression method returns to normal behavior.

DETECT: This bit is set to 1 to indicate that the touch is reporting and present within the range of the sensor.

XPOS Field

This field reports the X position.

YPOS Field

This field reports the Y position.

AUXDATA[] Fields

These fields hold the user configurable (type specific) information about the touch, as determined by the TYPE code in the TCHSTATUS field.

NOTE	AUXDATA[] may contain zero value fields and in some cases these may be treated as special values within the product's operating system. It may be necessary, therefore, for the product's driver code to replace certain zero values with non zero values (for example, if the value is used for major axis length in Android devices).

If the touch is a finger touch, the content of these fields is determined by which bits are enabled in the TCHAUX configuration field. The AUXDATA[] bytes are filled in the same order as the configuration TCHAUX bits, depending on which bits are enabled. See "TCHAUX Field") for more details.

4.3.5 DIAGNOSTIC DEBUG DATA

The diagnostic debug data modes for the Multiple Touch Touchscreen T100 object is shown in Table 4-32. The diagnostic debug data can be viewed in the Diagnostic Debug T37 object by using the appropriate Command Processor T6 command, also listed in Table 4-32.

The debug modes are described in the following sections.

TABLE 4-32: DIAGNOSTIC DEBUG COMMANDS

Command	Name	Description
0x36	Touch Status Data Mode	The Diagnostic Debug T37 object holds the status data associated with each touch. See Section 4.3.5.1 "Touch Status Data Mode".
0xF4	Touchscreen Mode	The Diagnostic Debug T37 object holds the current configuration data for the Multiple Touch Touchscreen T100 object. Specifically, it gives the current touch threshold in use as an 8-bit value. See Section 4.3.5.2 "Touchscreen Mode".

Under NDA
© 2021 Microchip Technology Inc. and its subsidiaries

4.3.5.1 Touch Status Data Mode

Figure 4-26 shows the organization of the data in the pages for the Touch Status Data mode.

FIGURE 4-26: DIAGNOSTIC DEBUG T37 FIELDS – TOUCH STATUS DATA MODE

Page	Data Index	Data[]
Page 0	0 – 18	Touch status for Touch 0
	19 – 37	Touch status for Touch 1
	38 – 56	Touch status for Touch 2
	57 – 75	Touch status for Touch 3
	76 – 94	Touch status for Touch 4
	95 – 113	Touch status for Touch 5
	114 – 127	Touch status for Touch 6 (bytes 0 to 13)
Page 1	128 – 132	Touch status for Touch 6 (bytes 14 to 18)
	133 – 151	Touch status for Touch 7
	152 – 170	Touch status for Touch 8
	171 – 189	Touch status for Touch 9
	190 – 208	Touch status for Touch 10
	209 – 227	Touch status for Touch 11
	228 – 246	Touch status for Touch 12
	247 – 255	Touch status for Touch 13 (bytes 0 to 8)
Page 2	256 – 265	Touch status for Touch 13 (bytes 9 to 18)
	266 – 284	Touch status for Touch 14
	285 – 303	Touch status for Touch 15
	304 – 383	Reserved

Assumes page size = 128

In Figure 4-26 the touch status data for each touch consists of the following:

Byte	Data
0	TCHSTATUS from Multiple Touch Touchscreen T100 messages (see “TCHSTATUS Field”)
1	X filtered position LSByte
2	X filtered position MSByte
3	Y filtered position LSByte
4	Y filtered position MSByte
5	X unfiltered position whole part LSByte
6	X unfiltered position whole part MSByte
7	Y unfiltered position whole part LSByte
8	Y unfiltered position whole part MSByte
9	Reserved
10	Reserved
11	Reserved
12	Reserved
13	X unfiltered position fractional part
14	Y unfiltered position fractional part
15 – 18	Reserved

4.3.5.2 Touchscreen Mode

Figure 4-27 shows the organization of the data in the pages for the Touchscreen mode. This shows the current configuration data for the Multiple Touch Touchscreen T100 object.

FIGURE 4-27: DIAGNOSTIC DEBUG T37 DATA – TOUCHSCREEN MODE

Page	Data Index	Data[]
Page 0	0	Touch Threshold for Multiple Touch Touchscreen T100
	1 – 6	Reserved
	7	Coupling Level (from Retransmission Compensation T80)
	8 – 127	Reserved

Assumes page size = 128

Under NDA
© 2021 Microchip Technology Inc. and its subsidiaries

5.0 SIGNAL PROCESSING OBJECTS

5.1 Introduction

Signal processing objects process the data from other objects, for example to provide filtering operations. Table 5-1 lists the signal processing objects on the mXT640UD-CCUBHA1 Family.

TABLE 5-1: SIGNAL PROCESSING OBJECTS

Object	Description
Key Thresholds T14 (PROCI_KEYTHRESHOLD_T14)	Allows different thresholds to be specified for each key in a Key Array. See Section 5.2 “Key Thresholds T14 Object”.
Key ID Configuration T16 (PROCI_KEYIDCONFIG_T16)	Controls the reporting of Key Array T15 keys. See Section 5.3 “Key ID Configuration T16 Object”.
One-touch Gesture Processor T24 (PROCI_ONETOUCHGESTUREPROCESSOR_T24)	Operates on the data from a Touchscreen object. A One-touch Gesture Processor T24 converts touches into one-touch finger gestures (for example, taps, double taps and drags). See Section 5.4 “One-touch Gesture Processor T24 Object”.
Two-touch Gesture Processor T27 (PROCI_TWOTOUCHGESTUREPROCESSOR_T27)	Operates on the data from a One-touch Gesture Processor T24 object. A Two-touch Gesture Processor T27 converts touches into two-touch finger gestures (for example, pinches, stretches and rotates). See Section 5.5 “Two-touch Gesture Processor T27 Object”.
Grip Suppression T40 (PROCI_GRIPSUPPRESSION_T40)	Suppresses false detections caused, for example, by the user gripping the edge of a touchscreen. See Section 5.6 “Grip Suppression T40 Object”.
Touch Suppression T42 (PROCI_TOUCHSUPPRESSION_T42)	Suppresses false detections caused by unintentional large touches by the user. See Section 5.7 “Touch Suppression T42 Object”.
Passive Stylus T47 (PROCI_STYLUS_T47)	Processes passive stylus input. See Section 5.8 “Passive Stylus T47 Object”.
Shieldless T56 (PROCI_SHIELDLESS_T56)	Allows a sensor to use true single-layer co-planar construction. See Section 5.9 “Shieldless T56 Object”.
Lens Bending T65 (PROCI_LENSBENDING_T65)	Compensates for lens deformation (lens bending) by attempting to eliminate the disturbance signal from the reported deltas. See Section 5.10 “Lens Bending T65 Object”.
Noise Suppression T72 (PROCG_NOISESUPPRESSION_T72)	Performs various noise reduction techniques during sensor signal acquisition. See Section 5.11 “Noise Suppression T72 Object”.
Glove Detection T78 (PROCI_GLOVEDETECTION_T78)	Allows for the reporting of glove touches. See Section 5.12 “Glove Detection T78 Object”.
Retransmission Compensation T80 (PROCI_RETRANSMISSIONCOMPENSATION_T80)	Limits the negative effects on touch signals caused by poor device coupling to ground or moisture on the sensor. See Section 5.13 “Retransmission Compensation T80 Object”.
Self Capacitance Noise Suppression T108 (PROCG_NOISESUPSELFCAP_T108)	Suppresses the effects of external noise within the context of self capacitance touch measurements. See Section 5.14 “Self Capacitance Noise Suppression T108 Object”.
Self Capacitance Grip Suppression T112 (PROCI_SELFCAPGRIPSUPPRESSION_T112)	Allows touches to be reported from the self capacitance measurements while the device is being gripped. See Section 5.15 “Self Capacitance Grip Suppression T112 Object”.
Ignore Nodes T141 (PROCG_IGNORENODES_T141)	Defines a set of sensor nodes that are to be excluded from normal processing. See Section 5.16 “Ignore Nodes T141 Object”.

5.2 Key Thresholds T14 Object

5.2.1 INTRODUCTION

The Key Thresholds T14 object allows a different threshold and hysteresis to be specified for each key in the linked Key Array T15 object.

Variation in flexible PCB (FPCB) key geometry and/or key location means that a wide range of deltas can be seen over the whole key array for the same touch. As a result, if one touch threshold is used for all keys, the keys would appear to have different sensitivities. Usually this variation is not great and is not very noticeable. There are some cases, however, where the touch signal can vary by as much as 50% from key to key, which then becomes noticeable.

By using a different threshold and hysteresis for each key, it is possible to individually tune each key and remove any issues that could be seen with false touches, poor sensitivity and so on.

- If the Key ID Configuration T16 object is enabled for use, the Key Thresholds T14 key thresholds will be applied to the actual Key Array T15 key IDs rather than the remapped keys, as the threshold settings are related to the key measurement data obtained from physical key locations.

5.2.2 CONFIGURATION

TABLE 5-2: CONFIGURATION FOR KEY THRESHOLDS T14 (PROCI_KEYTHRESHOLD_T14)

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Reserved	Reserved							
1	EXTRATHR[0]	Extra Threshold for Key 0							
...									
32	EXTRATHR[31]	Extra Threshold for Key 31							
33	EXTRAHYST[0]	Extra Hysteresis for Key 0							
...									
64	EXTRAHYST[31]	Extra Hysteresis for Key 31							

EXTRATHR[] Fields

These fields specify the extra threshold that is applied to the Key Array T15 TCHTHR to obtain the threshold for each key (see “TCHTHR Field”). Increasing the extra threshold will add to the base Key Array T15 threshold, thus making the key less sensitive. The Key Array T15 TCHTHR value therefore becomes the lowest TCHTHR required.

Range: 0 (use Key Array T15 TCHTHR value) to 255

EXTRAHYST[] Fields

These fields specify the Extra Hysteresis that is applied to the Key Array T15 TCHHYST to obtain the hysteresis for each key (see “TCHHYST Field”). Note that the combined hysteresis is still limited to 25% of the per-key threshold. Increasing the extra hysteresis will add to the hysteresis, thus making it easier to keep the key in detect. The Key Array T15 TCHHYST value therefore becomes the lowest TCHHYST required.

Range: 0 to 255

Under NDA

© 2021 Microchip Technology Inc. and its subsidiaries

5.3 Key ID Configuration T16 Object

5.3.1 INTRODUCTION

The Key ID Configuration T16 object controls the reporting of Key Array T15 keys (see Section 4.2 “Key Array T15 Object”).

By default, the key IDs correspond to the key numbers of the defined key array. The Key ID Configuration T16 object, if enabled, allows an alternative mapping between the keys and the key IDs to be configured. This allows the Key Array T15 object to report its key states according to the key IDs defined by the Key ID Configuration T16 object. This may be useful, for example, if the user requires standard key IDs to be used across various products.

Note that the Key ID Configuration T16 object only affects the reporting of the keys, and not their physical operation. This means that:

- Key Thresholds T14 key thresholds will be applied to the actual Key Array T15 key IDs rather than the remapped keys (if applicable), as the threshold settings are related to the key measurement data obtained from physical key locations.
- If a GPIO Configuration T19 trigger is used, this will still be detected on the base key ID and not on a remapped key ID. For example, if a touch on Key 0 is detected (regardless of its configured key ID), this will correspond to the actual key 0 in the key array and not to a key that is remapped to Key ID 0.
- The Diagnostic Debug T37 object will show the debug data following the sequential order of the actual Key Array T15 key IDs and not the remapped key order as per Key ID Configuration T16 (see Section 4.2.5 “Diagnostic Debug Data”). This also applies to the Key Array T15 data output over the Hardware Debug Interface (see Command Processor T6 “DEBUGCTRL2 Field”).

On the mXT640UD-CCUBHA1 Family there is a single instance of the Key ID Configuration T16 object, linked to the Key Array T15 object. Note that the Key Array T15 object can represent either standard keys or generic keys (see Key Array T15 Section 4.2.1 “Introduction”).

5.3.2 CONFIGURATION

TABLE 5-3: CONFIGURATION FOR KEY ID CONFIGURATION T16 (PROCI_KEYIDCONFIG_T16)

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	CTRL	Reserved							ENABLE
1	KEYID[0]	Reported Key ID for Key 0							
:	:	:							
32	KEYID[31]	Reported Key ID for Key 31							
Note:	If the linked Key Array T15 object represents generic keys, only up to 3 key IDs can be mapped (that is, key IDs 0 to 2).								

CTRL Field

ENABLE: Enables the use of the Key ID Configuration T16 object. The object is enabled if set to 1, and disabled if set to 0.

KEYID[] Fields

These fields are used to configure Reported Key IDs for the individual keys defined by the Key Array T15 object. Each KEYID entry configures the Key ID for the Key Array T15 key number that corresponds to the index of the KEYID[] array. For example, KEYID[2] configures the Key ID for Key 2 in the Key Array.

Key IDs can be specified in the range 0 to 31.

Note that, if the Key ID Configuration T16 object is enabled for use, all the Key IDs for use by the Key Array must be defined, even if they are unchanged from their default values (there is no default setting). Care should also be taken to avoid multiple physical keys reporting the same Key ID.

Range: 0 to 31

5.4 One-touch Gesture Processor T24 Object

5.4.1 INTRODUCTION

A One-touch Gesture Processor T24 object configures the on-chip gesture processing for one-touch gestures (such as taps, double taps, presses, flicks and drags).

The device supports the one-touch gestures listed in Table 5-4.

TABLE 5-4: ONE-TOUCH GESTURES

Gesture	Description
Press	A press occurs when the user touches and holds the touch surface. No significant movement takes place while the user's finger is on the touch surface. This could be used to select a number from a displayed numeric keypad. The same mechanism could be used to autorepeat the selected number if the user continues to press on the displayed number.
Release	A release occurs when the user removes their finger from the touch surface.
Tap	A tap occurs when the user quickly touches and releases the touchscreen. No significant movement takes place while the user's finger is on the touchscreen. It is characterized by a short touch duration. This could be used to activate a hyperlink on a displayed web page.
Double Tap	A double tap occurs when the user quickly touches and releases the touchscreen twice in quick succession. No significant movement takes place while the user's finger is on the touchscreen, or between successive touches. It is characterized by short touch durations, and a short gap between the first release and the second touch. This could be used to select a word in a displayed document.
Flick	A flick occurs when the user quickly touches the touchscreen, moves their finger a short distance across the surface and releases touch. It is characterized by a short touch duration. This could be used to display the next image in a sequence of images.
Throw	A throw occurs when the user touches the touchscreen, moves their finger across the surface, and releases the touch in a quick flick-type motion. Conceptually, it can be considered like a drag motion followed by a flick. It is characterized by a large movement across the touchscreen and a rapid acceleration of the touch just before the release.
Drag	A drag occurs when the user touches the touchscreen, moves their finger across the surface, and releases touch. It is characterized by a large movement across the touchscreen. Depending on the application, multiple drag events may be generated as the user moves their finger. This could be used to select a sentence in a displayed document.
Tap-and-press	A tap-and-press occurs when the user generates both a tap gesture and a press gesture in quick succession.
Tap-and-touch	A tap-and-touch occurs when the user generates a tap and then places their finger back on the screen in quick succession. The event is generated as soon as the second touch-down is validated. This gesture is useful for implementing a “knock-on” feature which allows the user’s product to be woken up by a tap-and-touch gesture on the touchscreen.

The host can combine one-touch gestures to create two-touch and multitouch gestures, either within the host operating system or in the driver code. For example, two simultaneous tap gestures on the touchscreen can be combined to form a two-touch tap. The mXT640UD-CCUBHA1 Family also provides on-chip touchscreen gesture processing for three specific two-touch gestures: pinch, rotate and stretch (see the Two-touch Gesture Processor T27 in Section 5.5 “Two-touch Gesture Processor T27 Object”).

5.4.1.1 Touchscreen Units

Many of the One-touch Gesture Processor T24 configuration parameters are specified in units of touchscreen distance, which is dependent on Multiple Touch Touchscreen T100 XRANGE and YRANGE settings.

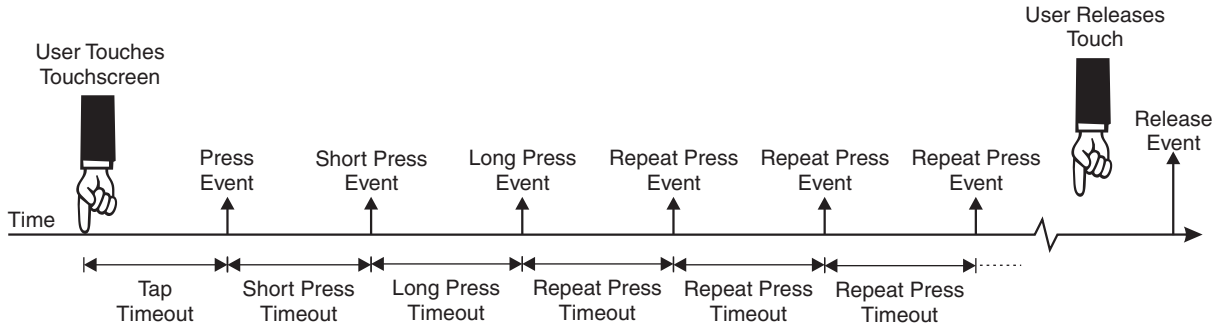
The touchscreen units can be different for the X and Y axes. For example, if XRANGE is twice as large as YRANGE, the touchscreen distance in the X direction is twice as large as in the Y direction. In this case, the circle shown in Figure 5-6 on page 102 would actually be more elliptical in shape.

Under NDA
© 2021 Microchip Technology Inc. and its subsidiaries

5.4.1.2 Press and Tap Gestures

Press and Tap gestures are created from a series of press events. These are generated when the touchscreen is touched, the touch is held, and then the touch is released. Figure 5-1 shows the most general case.

FIGURE 5-1: SEQUENCE OF PRESS EVENTS



In this case:

1. The user touches the touchscreen.
2. If the touch is held for longer than the Tap Timeout, a Press event is generated.
3. If the user continues to hold the touch, and the Short Press Timeout is exceeded, a short press event is generated.
4. If the user continues to hold the touch, and the Long Press Timeout is exceeded, a long press event is generated.
5. If the user continues to hold the touch, and the Repeat Press Timeout is exceeded, a repeat press event is generated.
6. If the user continues to hold the touch, further repeat press events are generated at regular intervals whenever the Repeat Press Timeout is exceeded.
7. When the user releases the touchscreen, a release event is generated.

Note that in Figure 5-1 all three press events are enabled. The One-touch Gesture Processor T24 object, however, allows you to enable or disable the generation of the short, long and repeat press events, as required (see "PROCESS Field"). All touches always generate release events.

5.4.1.3 Double Tap, Tap-and-Touch and Tap-and-Press Gestures

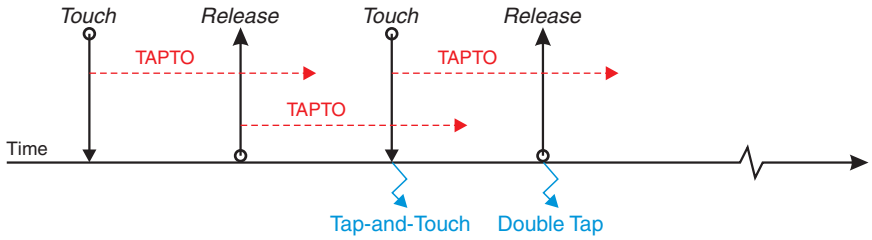
Double Tap, Tap-and-Touch and Tap-and-Press Gestures are created from two successive touch events. The second touch is reported as a Double Tap, Tap-and-Touch or Tap-and-Press event depending on the timing of the second touch. The generation of all three gestures is enabled by the PROCESS DBLTAPEN control (see "PROCESS Field").

Figure 5-2 shows how the three events are distinguished from each other:

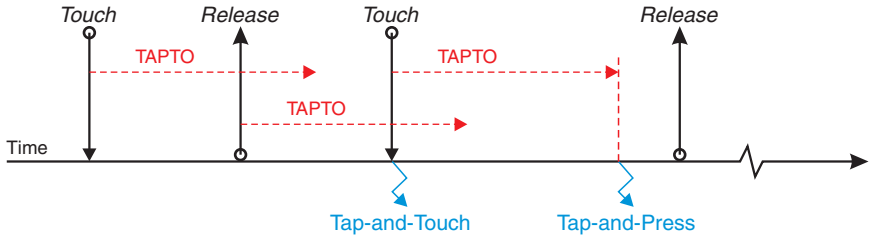
- A Tap-and-touch event is generated at the point of touchdown, if a second touch event occurs before TAPTO expires after a valid Tap. Note that this occurs whether or not the Tap event is enabled.
- A Double Tap event is generated only if a second valid Tap occurs. In this case the second touchdown must be before TAPTO expires after the first touch release, and the second release must occur before TAPTO expires after the second touchdown.
- A Tap-and-press is generated only if there is a second touchdown before TAPTO expires after the first release, and the second press is held for longer than TAPTO before being released. The Tap-and-press event is generated in this case at the point when the TAPTO timer expires.

FIGURE 5-2: SEQUENCE OF EVENTS WITH TAP-AND-TOUCH EVENT

Tap-and-Touch with Double Tap



Tap-and-Touch with Tap-and-Press



5.4.2 CONFIGURATION

TABLE 5-5: CONFIGURATION FOR ONE-TOUCH GESTURE PROCESSOR T24 (PROCI_ONETOUCHGESTUREPROCESSOR_T24)

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	CTRL	Reserved			DISTSF	DISRPTTAPOTH	DISRPTTAPMOV	RPTEN	ENABLE
1	NUMGEST	Maximum number of reported gestures							
2	GESTEN	LPRESS	SPRESS	DRAG	FLICK	DBLTAP	TAP	RELEASE	PRESS
3		Reserved					TAPTCH	THROW	RPRESS
4	PROCESS	Reserved		THROWEN	FLICKEN	DBLTAPEN	REPEN	LONGEN	SHORTEN
5	TAPTO	Reserved	Tap Timeout						
6	FLICKTO	Reserved	Flick Timeout						
7	DRAGTO	Reserved	Drag Timeout						
8	SPRESSTO	Reserved	Short Press Timeout						
9	LPRESSTO	Reserved	Long Press Timeout						
10	REPPRESSTO	Reserved	Repeat Press Timeout						
11	FLICKTHR	Flick Threshold LSByte							
12		Flick Threshold MSByte							
13	DRAGTHR	Drag Threshold LSByte							
14		Drag Threshold MSByte							
15	TAPTHR	Tap Threshold LSByte							
16		Tap Threshold MSByte							
17	THROWTHR	Throw Threshold LSByte							
18		Throw Threshold MSByte							

CTRL Field

ENABLE: Enables the use of the One-touch Gesture Processor T24 object. The object is enabled if set to 1, and disabled if set to 0.

RPTEN: Allows the object to send status messages to the host through the Message Processor T5 object. Reporting is enabled if set to 1, and disabled if set to 0.

Under NDA
© 2021 Microchip Technology Inc. and its subsidiaries

DISRPTTAPMOV: Suppresses the reporting of moving taps. If this bit is set to 1, reporting of all tap related (tap, tap-and-press, tap-and-touch, double tap) gesture events is disabled if the touch is moving. This check is performed on each gesture independently.

DISRPTTAPOTH: Suppresses the reporting of Tap related gestures. If this bit is set to 1, reporting of all tap related (tap, tap-and-press, tap-and-touch, double tap) gesture events is disabled if there is a touch down/up event registered against any other touch that does not have gesture processing enabled on it via the NUMGEST field.

NOTE	This feature should be used only when there is only one gesture process enabled via the NUMGEST configuration field (NUMGEST = 1).

DISTSF: Specifies the Distance Scaling Factor for the reported DIST value in the One-touch Gesture Processor T24 message. If this bit is set to 0, no downscaling is applied to the reported DIST value. If this bit is set to 1, the DIST message field is scaled to half its value before being output.

NUMGEST Field

This field specifies the maximum number of touches that are to be reported. For example, if this field is set to 1, only the first touch is reported as a gesture and all other touches do not generate messages.

NOTE	If Two-touch Gesture Processor T27 is enabled for use, NUMGEST must be ≥ 2 for the Two-touch Gesture Processor T27 object to report two-touch gesture messages.

This field does not remove report IDs. Report IDs are fixed to the maximum number of touch gestures.

Range: 1 to 4

GESTEN Field

Enables one-touch gestures and reporting.

PRESS: Press event reporting is enabled if set to 1, and disabled if set to 0.

RELEASE: Release event reporting is enabled if set to 1, and disabled if set to 0.

TAP: Tap event reporting is enabled if set to 1, and disabled if set to 0.

DBLTAP: Double Tap event reporting is enabled if set to 1, and disabled if set to 0.

FLICK: Flick event reporting is enabled if set to 1, and disabled if set to 0.

DRAG: Drag event reporting is enabled if set to 1, and disabled if set to 0.

SPRESS: Short Press event reporting is enabled if set to 1, and disabled if set to 0.

LPRESS: Long Press event reporting is enabled if set to 1, and disabled if set to 0.

RPRESS: Repeat Press event reporting is enabled if set to 1, and disabled if set to 0.

THROW: Throw event reporting is enabled if set to 1, and disabled if set to 0.

TAPTCH: Tap-and-touch event reporting is enabled if set to 1, and disabled if set to 0.

PROCESS Field

This field enables and disables the press events and some of the one-touch gestures that can be processed by the device.

SHORTEN: Enables short presses to be processed by the device. Short presses are enabled if set to 1, and disabled if set to 0.

LONGEN: Enables long presses to be processed by the device. Long presses are enabled if set to 1, and disabled if set to 0.

REPEN: Enables repeat presses to be processed by the device. Repeat presses are enabled if set to 1, and disabled if set to 0.

DBLTAPEN: Enables double tap, tap-and-touch and tap-and-press gestures to be processed by the device. Double taps are enabled if set to 1, and disabled if set to 0.

FLICKEN: Enables flicks to be processed by the device. Flicks presses are enabled if set to 1, and disabled if set to 0.

THROWEN: Enables throws to be processed by the device. Throws are enabled if set to 1, and disabled if set to 0.

TAPTO Field

This field specifies the timeout for a tap gesture. This is the maximum duration of a tap. A tap gesture is when the user touches and releases a touch, without moving their finger. If this sequence takes less time than the tap timeout, a tap event is generated. The timeout is specified in units of 4 ms. A value of zero means 75 (300 ms). The timeout should not be set shorter than the active cycle period.

This value also controls the generation of double tap events. A double tap event consists of a press, release, press, and final release. If the time between each of these actions is less than the tap timeout, a double tap event is generated.

The user must move their finger less than the Tap Threshold (TAPTHR) during a tap to generate a tap or double tap gesture. Also, the distance between consecutive taps must be less than the Tap Threshold to generate a double tap gesture.

If the timeout expires and the touch is still present, the press processing commences.

Range: 0 (75 = 300 ms), 1 to 127

FLICKTO Field

This field specifies the timeout for a flick gesture. This is the maximum duration of a flick gesture. A flick gesture is when the user presses the touchscreen, moves their finger, and then releases the touchscreen. A flick event is generated if a touch moves more than the flick threshold (defined by the FLICKTHR field), in less time than the tap timeout (TAPTO), and is then released within the FLICKTO. The flick timeout should not be set shorter than the active cycle period.

The flick timeout is specified in units of 4 ms. A value of zero means 75 (300 ms).

See also “FLICKTHR and TAPTHR Fields”.

Range: 0 (75 = 300 ms), 1 to 127

DRAGTO Field

This field specifies the timeout for detecting a new drag event. This is the maximum time between two drag events. A drag gesture is when the user presses the touchscreen and moves their finger. This generates a drag event. If the user moves their finger again within the drag timeout period, another drag event is generated. If the user’s finger remains stationary for longer than the drag timeout period, drag processing stops. In this case, any enabled press events start to be generated. The drag timeout is specified in units of 4 ms. A value of zero means 75 (300 ms). The timeout should not be set shorter than the active cycle period. See also “DRAGTHR Field”.

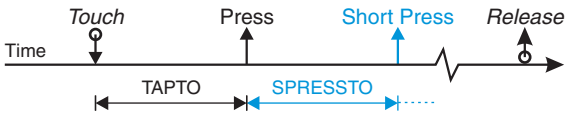
Range: 0 (75 = 300 ms), 1 to 127

SPRESSTO Field

This field specifies the Short Press Timeout. This is the time between the start of press processing and the generation of a short press event (see Figure 5-3). The Short Press Timeout is used only if short press processing is enabled using the “SHORTEN” bit (see “PROCESS Field”).

FIGURE 5-3: SHORT PRESS TIMEOUT

SHORTEN Set



The short timeout is specified in units of 4 ms. A value of zero means 75 (300 ms). The timeout should not be set shorter than the active cycle period.

Range: 0 (75 = 300 ms), 1 to 127

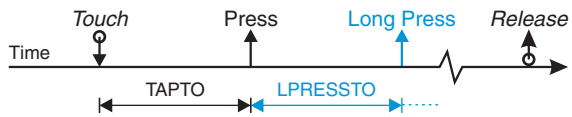
Under NDA
© 2021 Microchip Technology Inc. and its subsidiaries

LPRESSTO Field

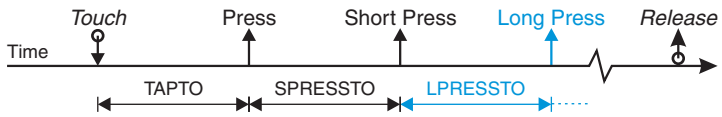
This field specifies the Long Press Timeout. This is the time taken for the generation of a long press event. The start point for the timeout depends on whether or not short events are also enabled. If short presses are enabled, the timeout is from the short press event; otherwise it is from the start of press processing (see Figure 5-4). The Long Press Timeout is used only if long press processing is enabled using the “LONGEN” bit (see “PROCESS Field”).

FIGURE 5-4: LONG PRESS TIMEOUT

LONGEN Only Set



LONGEN and SHORTEN Set



The long timeout is specified in units of 4 ms. A value of zero means 75 (300 ms). The timeout should not be set shorter than the active cycle period.

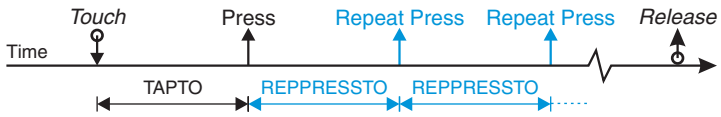
Range: 0 (75 = 300 ms), 1 to 127

REPPRESSTO Field

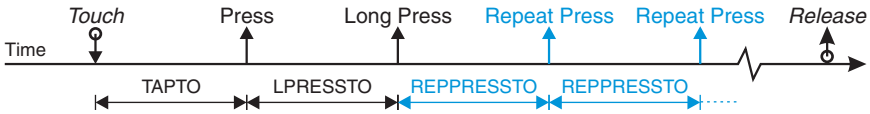
This field specifies the Repeat Press Timeout. This is the time taken for the generation of a repeat press event. The start point for the timeout depends on whether short press and long press events are also enabled (see Figure 5-5). The repeat timeout also specifies the time between the generation of consecutive repeat press events. The Repeat Press Timeout is used only if repeat press processing is enabled using the “REPEN” bit (see “PROCESS Field”).

FIGURE 5-5: REPEAT PRESS TIMEOUT

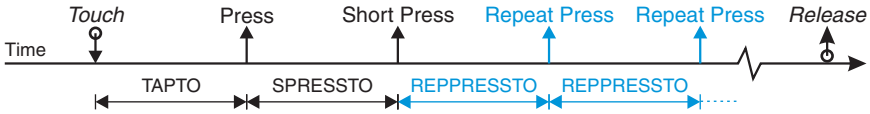
REPEN Only Set



REPEN and LONGEN Set



REPEN and SHORTEN Set



REPEN, LONGEN and SHORTEN Set

The repeat timeout is specified in units of 4 ms. A value of zero means 75 (300 ms). The timeout should not be set shorter than the active cycle period.

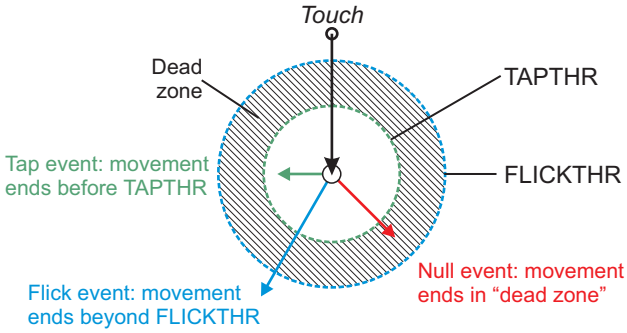
Range: 0 (75 = 300 ms), 1 to 127

FLICKTHR and TAPTHR Fields

These fields dictate how a quick touch-and-release action is to be interpreted. This action can be interpreted as either a tap or a flick gesture. These two gestures are essentially the same. The only difference is the amount of movement that occurs between the touch and the release. With a flick, the user's finger moves a significant distance before the touch is release. With a tap there is minimal or no movement.

If the movement is less than the Tap Threshold (TAPTHR), the gesture is interpreted as a tap. If the movement is greater than or equal to the Flick Threshold (FLICKTHR), the gesture is interpreted as a flick. If the movement is in the "dead zone" between the two thresholds, no gesture is generated for the touch. This is summarized in Figure 5-6.

FIGURE 5-6: FLICK AND TAP THRESHOLDS



The thresholds are specified in units of touchscreen distance.

It may be useful to set both the TAPTHR and FLICKTHR values to the same values initially. This means there is no "dead zone" and so either a tap or flick gesture is always reported. If the design needs to be fine tuned, the two controls can later be set to different values to create a "dead zone". Note that, in this case, FLICKTHR must be greater than or equal to TAPTHR.

See also "TAPTO Field" and "FLICKTO Field".

Range: 0 to 65535

DRAGTHR Field

This value specifies how far (units of touchscreen distance) the user must move their finger across the touchscreen to generate a drag event. See also "DRAGTO Field". The drag threshold is specified in units of touchscreen distance.

Range: 0 to 65535

THROWTHR Field

This field specifies the speed on the drag or release from the drag that is needed to generate a throw gesture. The speed of the user's finger must be greater than or equal to THROWTHR for a throw gesture to be generated. The speed is specified as:

32 × touchscreen_distance_per_ms

A minimum time of 4 ms for the release is assumed.

Range: 0 to 65535

Under NDA
© 2021 Microchip Technology Inc. and its subsidiaries

5.4.3 MESSAGES

The message data for a One-touch Gesture Processor T24 object is shown in Table 5-6.

TABLE 5-6: MESSAGE DATA FOR ONE-TOUCH GESTURE PROCESSOR T24 (PROCI_ONETOUCHGESTUREPROCESSOR_T24)

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	STATUS	Reserved				Event			
2	XPOSMSB	X position bits 4 to 11							
3	YPOSMSB	Y position bits 4 to 11							
4	XYPOSLSB	X position bits 0 to 3				Y position bits 0 to 3			
5	DIR	Gesture direction							
6	DIST	Gesture distance LSByte							
7		Gesture distance MSByte							
8	XYPOSMSB	X position bits 12 to 15				Y position bits 12 to 15			

STATUS Field

This field reports which single-touch event has occurred. Table 5-7 gives the possible values and their meanings.

TABLE 5-7: EVENTS

Event	Description
0	Reserved
1	Press
2	Release
3	Tap
4	Double-tap
5	Flick
6	Drag
7	Short Press
8	Long Press
9	Repeat Press
10	Tap-and-press
11	Throw
12	Tap-and-touch
13 to 15	Reserved

XPOSMSB, YPOSMSB, XYPOSLSB and XYPOSMSB Fields

These fields report the X and Y position of the gesture. XYPOSLSB contains bits 0 to 3 of the position, XPOSMSB/YPOSMSB contains bits 4 to 11 of the position and XYPOSMSB contains bits 12 to 15 of the position. Thus, the position is constructed as follows:

X position = ((XYPOSMSB & 0xF0) << 8) | (XPOSMSB << 4) | ((XYPOSLSB & 0xF0) >> 4)

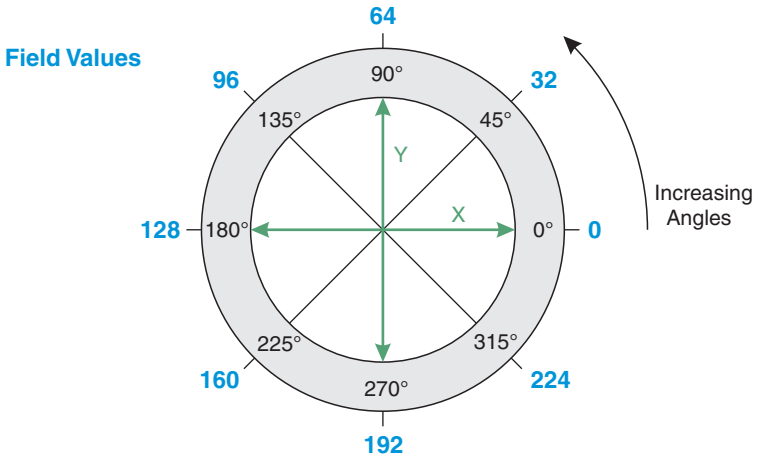
Y position = ((XYPOSMSB & 0x0F) << 12) | (YPOSMSB << 4) | (XYPOSLSB & 0x0F)

The position is reported in 16-bit format within the range specified by the Multiple Touch Touchscreen T100 XRANGE and YRANGE settings.

DIR Field

The DIR field contains the direction of the flick or throw when a flick or throw event is generated. The direction is specified in 1/256 of 360°, as in Figure 5-7. Note that if CTRL DISTSF is set to 1, the direction will be scaled down by half. In other words, the reported distance is equal to DIST × 2^DISTSF.

FIGURE 5-7: DIRECTION AND ANGLE VALUES



Note: The X and Y positions are before any processing of the Multiple Touch Touchscreen T100 ORIENT field.

DIST Field

The DIST field contains the distance of the flick or throw when a flick or throw event is generated.

Under NDA
© 2021 Microchip Technology Inc. and its subsidiaries

5.5 Two-touch Gesture Processor T27 Object

5.5.1 INTRODUCTION

NOTE	This object operates only if the linked One-touch Gesture Processor T24 object has also been set up correctly and enabled for use (see Section 5.4 “One-touch Gesture Processor T24 Object”). In particular, One-touch Gesture Processor T24 NUMGEST must be ≥ 2 for the Two-touch Gesture Processor T27 object to report two-touch gesture messages.

The Two-touch Gesture Processor T27 object configures the on-chip touchscreen gesture processing for the two-touch gestures: pinch, rotate and stretch (see Table 5-8). Other two-touch gestures can be generated by the host from one-touch gestures (see Section 5.4.1 “Introduction”).

TABLE 5-8: TWO-TOUCH GESTURES

Gesture	Description
Pinch	A pinch occurs when the user touches and holds the touch surface with two fingers and brings them together. It is not necessary for both fingers to move. One finger can remain stationary while the other moves towards it. It is the relative (decreasing) distance between the fingers that determines the pinch gesture.
Stretch	A stretch is the inverse of a pinch. The user touches and holds the touch surface with two fingers and moves them apart. It is not necessary for both fingers to move. One finger can remain stationary while the other moves away from it. It is the relative (increasing) distance between the fingers that determines the pinch gesture.
Rotate	A rotate occurs when the user touches and holds the touch surface with two fingers and then moves them in a circle around each other. It is not necessary for both fingers to move. One finger can remain stationary while the other moves around it. It is the relative angle between the fingers that determines the rotate gesture.

The two-touch gestures in Table 5-8 can be combined. For example, combining a rotate gesture with a stretch or pinch gesture will form an increasing or decreasing spiral gesture.

5.5.1.1 Touchscreen Units

Many of the Two-touch Gesture Processor T27, and related One-touch Gesture Processor T24, configuration parameters are specified in units of touchscreen distance, which is dependent on Multiple Touch Touchscreen T100 XRANGE and YRANGE settings.

This may affect the perceived aspect ratio. For example, if XRANGE is twice as large as YRANGE, the touchscreen distance in the X direction is twice as large as in the Y direction. In this case, the circle shown in Figure 5-6 on page 102 would actually be more elliptical in shape.

5.5.2 CONFIGURATION

TABLE 5-9: CONFIGURATION FOR TWO-TOUCH GESTURE PROCESSOR T27 (PROCI_TWOTOUCHGESTUREPROCESSOR_T27)

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	CTRL	Reserved			DISTSF	Reserved		RPTEN	ENABLE
1	NUMGEST	Maximum number of reported gestures							
2	Reserved	Reserved							
3	GESTEN	STRETCH	ROTATE	PINCH	Reserved				
4	ROTATETHR	Reserved	Rotate Threshold						
5	ZOOMTHR	Zoom Threshold LSByte							
6		Zoom Threshold MSByte							

CTRL Field

ENABLE: Enables the use of the Two-touch Gesture Processor T27 object. The object is enabled if set to 1, and disabled if set to 0.

RPTEN: Allows this object to send status messages to the host through the Message Processor T5 object. Reporting is enabled if set to 1, and disabled if set to 0.

DISTSF: Specifies the Distance Scaling Factor for the reported SEP value in the Two-touch Gesture Processor T27 message. If this bit is set to 0, no downscaling is applied to the reported SEP value. If this bit is set to 1, the SEP message field is scaled to half its value before being output.

NUMGEST Field

This field specifies the maximum number of two-touch gestures to be reported. Note that each gesture is made up of a pair of touches so the number of gestures specified represents half the number of touches. For example, if this field is set to 1, the first and second touch are paired as a gesture and are reported; all other touches are ignored by the processor.

Note that this field does not remove report IDs. Report IDs are fixed to the maximum number of touch gestures.

Range: 1

GESTEN Field

This field enables two-touch gestures and reporting.

PINCH: Pinch event reporting is enabled if set to 1, and disabled if set to 0.

ROTATE: Rotate event reporting is enabled if set to 1, and disabled if set to 0.

STRETCH: Stretch event reporting is enabled if set to 1, and disabled if set to 0.

ROTATETHR Field

This field specifies the minimum change in angle between two points for generating two-touch rotate gestures. The rotate threshold is specified in 1/256 of 360° (see "DIR Field" for details) with a maximum allowed angle of less than 180°. The reported angle is affected by the order in which the two touches are applied.

Range: 0 to 127

ZOOMTHR Field

This field specifies the minimum change in distance (in units of touchscreen distance) required between two simultaneous touches to generate a stretch or pinch gesture.

Note that the units of touchscreen distance are dependent on Multiple Touch Touchscreen T100 XRANGE and YRANGE settings and this can be different for the X and Y axes. For example, if XRANGE is twice as large as YRANGE, the touchscreen distance in the X direction is twice as large as in the Y direction.

Range: 0 to 65535

5.5.3 MESSAGES

The message data for a Two-touch Gesture Processor T27 object is shown in Table 5-10.

TABLE 5-10: MESSAGE DATA FOR TWO-TOUCH GESTURE PROCESSOR T27 (PROCI_TWOTOUCHGESTUREPROCESSOR_T27)

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	STATUS	STRETCH	ROTATE	PINCH	ROTATEDIR	Reserved			
2	XPOSMSB	Center of gesture X position bits 4 to 11							
3	YPOSMSB	Center of gesture Y position bits 4 to 11							
4	XYPOSLSB	X position bits 0 to 3				Y position bits 0 to 3			
5	ANGLE	Gesture angle							
6	SEP	Gesture separation LSByte							
7		Gesture separation MSByte							
8	XYPOSMSB	X position bits 12 to 15				Y position bits 12 to 15			

STATUS Field

This field reports which two-touch events have occurred. A two-touch gesture may consist of more than one two-touch gestures. For example, a rotate may be combined with a stretch to create a spiraling zoom effect.

Under NDA

© 2021 Microchip Technology Inc. and its subsidiaries

ROTATEDIR: Indicates the direction of the rotate event: 0 = rotation in the direction of increasing angles, 1 = rotation in the direction of decreasing angles (see "DIR Field" for details). This bit should be ignored if the ROTATE bit is not set.

PINCH: If set to 1, a pinch event has occurred.

ROTATE: If set to 1, a rotate event has occurred.

STRETCH: If set to 1, a stretch event has occurred.

XPOSMSB, YPOSMSB, XYPOSLSB and XYPOSMSB Fields

These fields report the X and Y position of the center of the gesture. XYPOSLSB contains bits 0 to 3 of the position, XPOSMSB/YPOSMSB contains bits 4 to 11 of the position and XYPOSMSB contains bits 12 to 15 of the position. Thus, the position is constructed as follows:

X position = ((XYPOSMSB & 0xF0) << 8) | (XPOSMSB << 4) | ((XYPOSLSB & 0xF0) >> 4)

Y position = ((XYPOSMSB & 0x0F) << 12) | (YPOSMSB << 4) | (XYPOSLSB & 0x0F)

The position is reported in 16-bit format within the range specified by the Multiple Touch Touchscreen T100 XRANGE and YRANGE settings.

ANGLE Field

The ANGLE field reports the angle between two touches in a two-touch gesture. The angle between the two touches, and its direction, is determined by the relative X and Y positions of the second touch from the first touch. The angle is reported in 1/256 of 360° (see "DIR Field" for details).

SEP Field

The SEP field reports the separation (distance) between the two touches in a two-touch gesture as a 16-bit number. Note that if CTRL DISTSF is set to 1, the separation will be scaled down by half. In other words, the reported separation is equal to SEP × 2^DISTSF.

This value could be used by the host, for example, as an indication of the speed of a rotate or stretch gesture.

5.6 Grip Suppression T40 Object

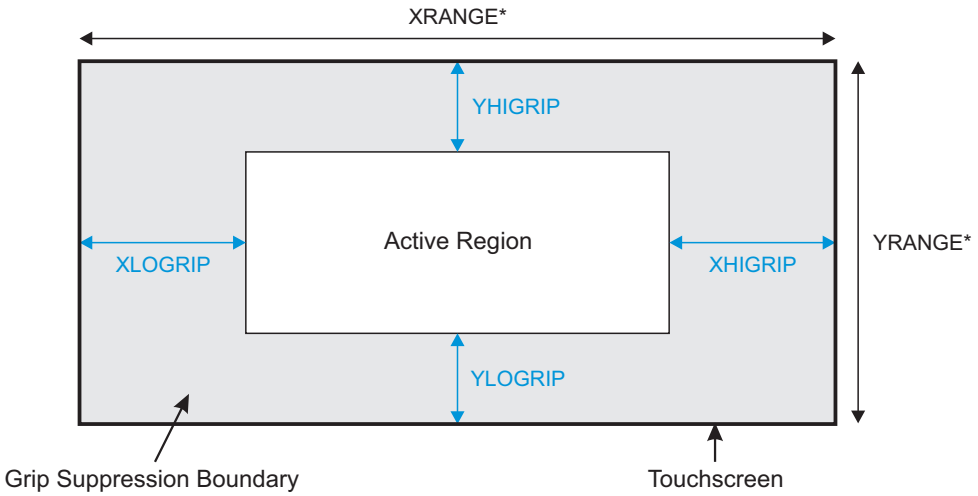
5.6.1 INTRODUCTION

The Grip Suppression T40 object suppresses false detections around the edge of the touchscreen, such as those caused by the user gripping the touchscreen on a handheld device. Specifically, this object defines a grip suppression boundary around the edge of the Multiple Touch Touchscreen T100 within which grip suppression may occur.The Grip Suppression T40 object processes the measurement data received from the Multiple Touch Touchscreen T100 object.

5.6.1.1 Grip Suppression Boundary

Figure 5-8 shows the settings that define the grip suppression boundary for a touchscreen.

FIGURE 5-8: GRIP SUPPRESSION BOUNDARY FIELDS



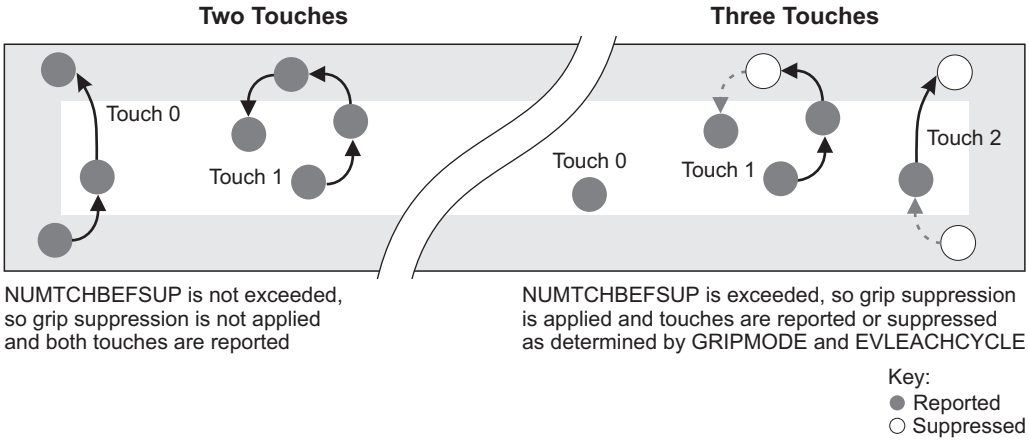
* XRANGE and YRANGE fields in a Multiple Touch Touchscreen T100

5.6.1.2 Number of Touches

Grip suppression takes place only when more than a certain number of touches are present, as specified by the NUMTCHBEFSUP field. When the number of touches on the sensor is less than or equal to the value of NUMTCHBEFSUP, grip suppression is not performed. When more than NUMTCHBEFSUP touches present, then all touches will be tested to see if they should be suppressed or not according to the grip mode and evaluation mode settings.

For example, Figure 5-9 shows the effect of specifying NUMTCHBEFSUP = 2 (that is, grip suppression is applied if three or more touches are present on the screen).

FIGURE 5-9: EXAMPLE: NUMTCHBEFSUP = 2



Under NDA
© 2021 Microchip Technology Inc. and its subsidiaries

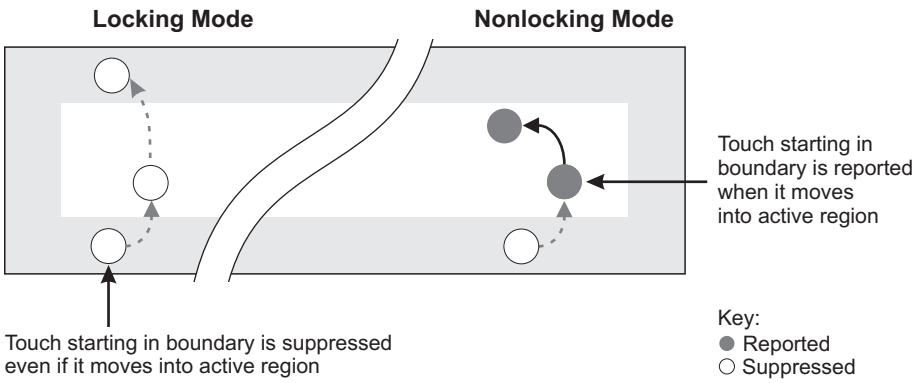
Note that the default value of zero effectively means that grip suppression will be evaluated as soon as a touch is placed on the screen (that is, at least one touch must be present for grip suppression to be performed).

5.6.1.3 Grip Mode

The grip mode (as defined by the GRIPMODE bit in the CTRL field) defines what happens to touches moving from the grip suppression boundary to the active region. Specifically, it defines whether a suppressed touch is locked (that is, remains suppressed) or becomes reported (see Section 5.6.1.3 “Grip Mode”):

- In locking mode, once a touch in the grip suppression boundary has been suppressed, it remains suppressed even if the touch subsequently moves into the active region.
- In non-locking mode, a touch that has been suppressed while in the grip suppression boundary becomes reported if it moves into the active region.

FIGURE 5-10: GRIP MODE



5.6.1.4 Evaluation Mode

The EVLEACHCYCLE control in the CTRL field determines when the touch is evaluated so that it can be suppressed or reported, as appropriate, as the touch is moved.

Touches can be evaluated for grip suppression as follows:

- Evaluated per touchdown – a touch is evaluated for grip suppression at the start of the touch (that is, on touchdown).
- Evaluated per cycle – a touch is evaluated for grip suppression on every cycle. This means that the state of the touch can change cycle by cycle during the touch depending on its position.

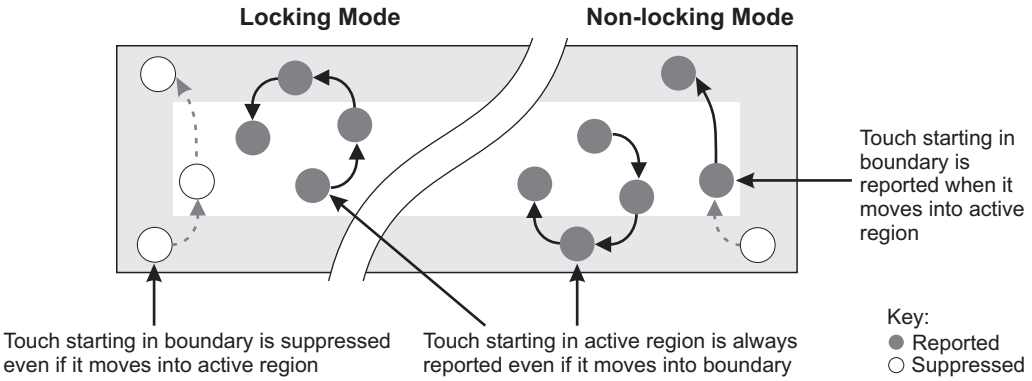
Between them, the EVLEACHCYCLE control and the GRIPMODE control (see Section 5.6.1.3 “Grip Mode”) create four combinations of behavior. Roughly speaking, GRIPMODE controls whether to report (unsuppress) a touch and EVLEACHCYCLE controls whether to suppress a touch.

If per touchdown evaluation is chosen (EVLEACHCYCLE = 0):

- In locking grip mode (GRIPMODE = 0), a touch that starts inside the grip suppression boundary is suppressed, and remains suppressed even if it subsequently moves into the active region.
- In non-locking grip mode (GRIPMODE = 1), a touch that starts in the grip suppression boundary is initially suppressed but becomes reported when it moves into the active region.
- In both grip modes, touches that start in the active region are always reported.

See Figure 0-1 for a further explanation.

Table 0-1. PER TOUCHDOWN EVALUATION

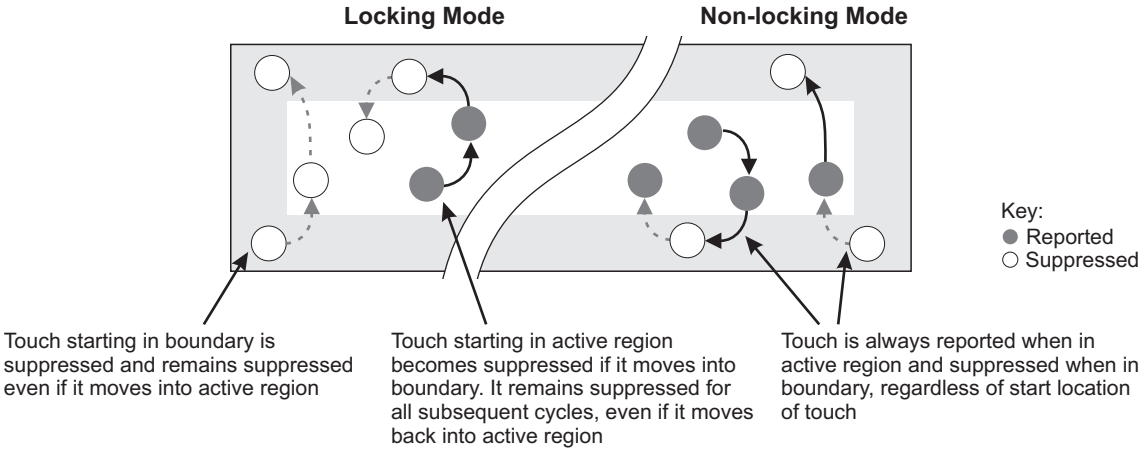


If per cycle evaluation is chosen (EVLEACHCYCLE = 1):

- In locking grip mode (GRIPMODE = 0), if a touch starts in the grip suppression boundary, or moves into the grip suppression boundary, it becomes suppressed and remains suppressed even if it subsequently moves into the active region.
- In non-locking grip mode (GRIPMODE = 1), the touch is suppressed when in the grip suppression boundary and reported when in the active region. Thus, in non-locking mode, the state of the touch will change cycle by cycle as it moves between the active region and the grip suppression boundary, regardless of where the touch started.

See Figure 5-11 for a further explanation.

FIGURE 5-11: PER CYCLE EVALUATION



5.6.2 CONFIGURATION

TABLE 5-11: CONFIGURATION FOR GRIP SUPPRESSION T40 (PROCI_GRIPSUPPRESSION_T40)

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	CTRL	Reserved		EVLEACHCYCLE	GRIPMODE	Reserved			ENABLE
1	XLOGRIP	Grip suppression X low boundary							
2	XHIGRIP	Grip suppression X high boundary							
3	YLOGRIP	Grip suppression Y low boundary							
4	YHIGRIP	Grip suppression Y high boundary							
5	Reserved	Reserved							
6	NUMTCHBEFSUP	Reserved		NUMTCHBEFSUP					

Under NDA

© 2021 Microchip Technology Inc. and its subsidiaries

CTRL Field

This field enables or disables grip suppression.

ENABLE: Enables the use of the Grip Suppression T40 object. If this bit is set to 1, the object is enabled and grip suppression takes place when appropriate. If this bit is set to 0, the object is disabled and grip suppression never occurs.

GRIPMODE: Sets the grip mode. Set this bit to 1 for non-locking mode, and 0 for locking mode. See Section 5.6.1.3 “Grip Mode” for more details.

EVLEACHCYCLE: Sets the evaluation mode. If this bit is set to 1, the system evaluates whether to suppress or report each touch within the grip boundary on every cycle. If this bit is set to 0, the system evaluates whether to suppress or report each touch within the grip boundary on touchdown and thereafter, as the touch moves, the subsequent behavior is determined by the grip mode alone. See Section 5.6.1.4 “Evaluation Mode” for more details.

Note that, in both cases, the current number of touches on the screen must exceed NUMTCHBEFSUP for grip suppression to be performed.

XLOGRIP, XHIGRIP, YLOGRIP and YHIGRIP Fields

These fields specify a grip suppression boundary around the edge of the Multiple Touch Touchscreen T100 that this object processes. Inside this boundary is the active region of the Touchscreen. XLOGRIP and XHIGRIP define the width of the boundaries on the low and high X edges respectively and YLOGRIP and YHIGRIP define the width of the boundaries on the low and high Y edges respectively. See Figure 5-8 on page 108 for more information on how the controls define the grip suppression boundary.

NOTE	The boundary settings are applied before the Multiple Touch Touchscreen T100 orientation controls (see “CFG1 Field”).

XLOGRIP/XHIGRIP Range: 0 to 255

YLOGRIP/YHIGRIP Range: 0 to 255

NUMTCHBEFSUP Field

NUMTCHBEFSUP: Specifies the Number of Touches Before Grip Suppression is activated. When the number of touches on the sensor is less than or equal to the value of NUMTCHBEFSUP, grip suppression is not performed. When the total number of touches on the screen exceeds NUMTCHBEFSUP, then the touches are evaluated to see if they should be reported or suppressed, according to the GRIPMODE and EVLEACHCYCLE controls in the CTRL field (see “CTRL Field”).

Note that the default value of zero means that at least one touch must be present for grip suppression to be performed (that is, grip suppression will be evaluated as soon as a touch is placed on the screen). Similarly, setting the value to 16 (maximum number of reported touches) or above, results in grip suppression never been applied.

See Section 5.6.1.2 “Number of Touches” for more details.

NUMTCHBEFSUP Range: 0 to 15 (number of touches to be exceeded),
16 to 63 (grip suppression never occurs)

5.6.3 MESSAGES

The Grip Suppression T40 object does not send messages directly. Instead the Multiple Touch Touchscreen T100 reports changes in the suppression state in its touch status message for a touch (see Section 4.3.4 “Messages” for an explanation of the SUP and UNSUP events).

The linked Multiple Touch Touchscreen T100 object reports an UNSUP event in the touch status message for a touch in the following circumstance:

- In non-locking mode, if the touch becomes unsuppressed (reported) after having previously been suppressed.

The linked Multiple Touch Touchscreen T100 object reports a SUP event in the touch status message for a touch in the following circumstances:

- If per cycle evaluation is chosen (EVLEACHCYCLE = 1), when the touch that moves from the active region into the grip suppression boundary
- If per cycle evaluation is chosen (EVLEACHCYCLE = 1), if the touch that is in the grip suppression boundary when NUMTCHBEFSUP becomes satisfied (that is, another touch is put down)

5.7 Touch Suppression T42 Object

5.7.1 INTRODUCTION

The Touch Suppression T42 object suppresses false detections from unintentional touches on the touchscreen. It processes the measurement data received from the Multiple Touch Touchscreen T100 object.

The Touch Suppression T42 object provides the following types of touch suppression:

- **Large Object Touch Suppression** – Detects unintentional touches from a large body area, such as from a palm. If an unintentional touch is detected, and both distance touch suppression and the reporting of suppressed large object touches are not in use, the entire Multiple Touch Touchscreen is suppressed. Otherwise, the large touch is suppressed, with further suppression determined by the behavior of the additional distance touch suppression. The suppression is released only when all the touches are released.
- **Maximum Touch Suppression** – Suppresses all touches if more than a specified number of touches has been detected. Any touches that are already detected will also be suppressed. The touches remain suppressed until all the touches are released or a recalibration occurs.
- **Distance Touch Suppression** – Operates in conjunction with large object suppression to suppress false touches if they are less than a specified distance from a suppressed touch. Any touches greater than this distance from a suppressed touch remain unsuppressed. Large object touch suppression needs to be enabled for distance touch suppression to operate.
- **Maximum Screen Area Suppression** – Triggers a full screen suppression when at least a specified number of nodes are in detect on the touchscreen. Note that number of nodes in detect are not necessarily part of a single touch.
- **Edge Touch Suppression** – Operates in conjunction with large object suppression to suppress false touches if they are on the same edge as a suppressed large object. Any touches not covering the edge sensor line that a suppressed touch also covers, remain unsuppressed. Note that large object touch suppression also needs to be enabled for edge touch suppression to operate.

Note that large object touch suppression (without distance touch suppression or suppressed large object touches reporting), and maximum touch suppression and maximum screen area suppression all lock the Touchscreen and with it any Adjacent Key Suppression locks that are currently applied.

5.7.2 CONFIGURATION

TABLE 5-12: CONFIGURATION FOR TOUCH SUPPRESSION T42 (PROCI_TOUCHSUPPRESSION_T42)

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	CTRL	Reserved	EDGESUP	SUPDISTEN	DISTLOCK	DISLOBJ	SHAPEEN	Reserved	ENABLE
1	Reserved	Reserved							
2	MAXAPPRAREA	Maximum approach area threshold							
3	MAXTCHAREA	Maximum touch area threshold							
4	SUPSTRENGTH	Suppression aggressiveness							
5	SUPEXTTO	Suppression extension timeout							
6	MAXNUMTCHS	Maximum touches							
7	SHAPESTRENGTH	Shaped-based aggressiveness							
8	SUPDIST	Suppression distance							
9	DISTHYST	Suppression distance hysteresis							
10	MAXSCRNAREA	Maximum screen area							
11	CFG	Reserved					SUPTCHRPTEN	RELAXDIAGSUP	RELAXCLOSESUP
12	Reserved	Reserved							
13	EDGESUPSTRENGTH	Edge suppression strength							

CTRL Field

This field enables or disables large touch suppression.

ENABLE: Enables the use of the Touch Suppression T42 object. The object is enabled if set to 1, and disabled if set to 0.

Under NDA

© 2021 Microchip Technology Inc. and its subsidiaries

SHAPEEN: Enables or disables shape-based suppression. This is a second level of rejection logic that is based upon the touch shape. Shape-based suppression is enabled if this bit is set to 1, and disabled if this bit is set to 0.

DISLOBJ: Disables the large object touch suppression. This allows the maximum touch suppression to be used exclusively. Large object touch suppression is disabled if this bit is set to 1, and enabled if it is set to 0 (the default).

DISTLOCK: Locks a distance touch suppression. That is, once a touch has been suppressed by distance touch suppression, it remains suppressed even if it moves outside the suppression boundary. Distance touch suppression is locked if this bit is set to 1, and unlocked if it is set to 0.

SUPDISTEN: Enables distance touch suppression, as defined by the SUPDIST field. Distance touch suppression is enabled if set to 1, and disabled if set to 0.

EDGESUP: Enables edge suppression. This operates in addition to large object touch suppression, such that if a large touch is suppressed on an edge X or Y line then all other touches on that line are also suppressed. An edge line is the lowest or highest X or Y line at the edge of the sensor (for example, X0, X31, Y0 or Y19).

If this bit is set to 1, then if a touch is suppressed while touching the edge of the touchscreen, all other touches that are touching that edge are also suppressed, irrespective of their size.

If this bit is set to 0, then if a touch is suppressed while touching the edge of the touchscreen, no other touches that are touching that edge are suppressed (unless they are so large that they themselves are suppressed).

For Example:

Touch 1 is a suppressed large touch that overlaps X0

Touch 2 is a normal finger touch that would not normally be suppressed but which also overlaps X0

Suppression of these touches occurs as follows:

- If EDGESUP = 0: only Touch 1 is suppressed; Touch 2 is not suppressed
- If EDGESUP = 1: both Touch 1 and Touch 2 are suppressed

Note that EDGESUP relates only to the edge at which a suppressed touch is present. In the example above, if Touch 1 does not move and continues to be suppressed, but Touch 2 is removed and touches down on the sensor on another edge line (X31, Y0 or Y19), then Touch 2 is not suppressed even if EDGESUP is set to 1.

MAXAPPRAREA Field

This field specifies the Maximum Approach Area Threshold. This is specified as the number of channels above the threshold based on Multiple Touch Touchscreen T100 (INTTHR – INTTHRHYST). If the number of channels exceeds this threshold, the touch is automatically treated as a large unintentional touch and the touch is suppressed without any further checks or suppression calculations.

The maximum Maximum Approach Area Threshold is specified as the number of channels, where a value of 0 means 40 channels.

Range: 0 (40), 1 to 255

MAXTCHAREA Field

This field specifies the Maximum Touch Area Threshold. This is specified as the number of channels above the Touch Threshold (TCHTHR – TCHHYST) specified in the Multiple Touch Touchscreen T100 object (see "TCHTHR Field"). If the number of channels in a touch that are above (TCHTHR – TCHHYST) exceeds this threshold, the touch is automatically treated as a large unintentional touch. In this case the touch is suppressed without any further checks or suppression calculations.

The touch area is specified as the number of channels, where a value of 0 means 35 channels.

Range: 0 (35), 1 to 255

SUPSTRENGTH Field

This field specifies the Suppression Strength. This controls the aggressiveness of the Touch Suppression T42 object (see Table 5-13). A higher value reduces the aggressiveness and allows larger touches to be reported, and a lower value increases the aggressiveness and allows only small touches to be reported.

A value of 0 means 128; that is, normal aggressiveness.

A value of 255 disables both edge suppression and large object suppression. It does not, however, affect Maximum Touch Suppression (as defined by MAXNUMTCHS) so large touches in the center of the touchscreen can still be suppressed.

Range: See Table 5-13

TABLE 5-13: SUPSTRENGTH

Value	Meaning
0	Treated as a value 128: normal aggressiveness (default)
1 to 127	Increases aggressiveness (allows only small touches to be reported)
128	Normal aggressiveness
129 to 254	Reduces aggressiveness (allows larger touches to be reported)
255	Disables both Edge Touch Suppression and Large Object Touch Suppression (Maximum Touch Suppression is unaffected)

SUPEXTTO Field

Suppression extension timeout may extend the time needed to make a decision as to whether a touch is to be suppressed or not. The timeout applies only if the Touch Suppression T42 object cannot determine whether a touch is an actual touch or a rogue touch and requires more evidence to reach the proper decision.

The process will keep extending the DI, up to the value in the SUPEXTTO field, if it cannot determine whether a touch is an actual touch or a rogue touch until this timer expires. When this timer expires, the touch is suppressed.

The SUPEXTTO field is specified as the number of cycles, where a value of 0 disables the timeout and the determination process never expires. The effect of this is to allow the device to sleep after the normal DI period and then continue the decision process.

Range: 0 (never expires), 1 to 255

MAXNUMTCHS Field

This field specifies the maximum number of touches that will be reported before touches are suppressed by the maximum touch suppression. If more than MAXNUMTCHS touches are detected, the touchscreen suppresses all touches. This includes those touches already detected. The touches remain suppressed until all touches are released or a recalibration occurs.

A value of 0 ensures that all touches will be reported.

Range: 0 to 15

SHAPESTRENGTH Field

This field specifies the shape-based suppression strength. It is valid only if the second-level shape-based suppression is enabled (that is, the SHAPEEN bit of the CTRL field is set) and controls the shape-based classifier.

The range for SHAPESTRENGTH is 0 to 31, where the default value of 0 is treated as 10.

Values less than 10 result in more aggressive touch suppression; values greater than 10 result in less aggressive touch suppression.

Increasing the shape strength value above the default value of 10 allows more elongated or curved touches to be reported. Decreasing the shape strength value below the default results in the suppression of more touches so that only particularly square-shaped or circle-shaped touches are reported.

For most practical use cases, a SHAPESTRENGTH value between 5 and 20 is adequate.

Range: 0 (10), 1 to 31

SUPDIST Field

This field specifies the suppression distance for distance touch suppression. This feature suppresses false touches if they are less than SUPDIST distance from a touch that is suppressed by the large object suppression. Any touches greater than this distance from a suppressed touch remain unsuppressed.

Under NDA
© 2021 Microchip Technology Inc. and its subsidiaries

Distance touch suppression is enabled by the SUPDISTEN bit in the CTRL field.

The distance is specified as the number of nodes, where a value of 0 means a distance of 5 nodes.

Range: 0 (5), 1 to 255

DISTHYST Field

This field specifies the hysteresis for distance touch suppression. The hysteresis is specified as the number of nodes.

Range: 0 to 255

MAXSCRNAREA Field

This field is used to determine total number of nodes in detect on the touchscreen that will trigger a full screen suppression when a palm is present. The number of nodes in detect is MAXSCRNAREA × 4. Note that number of nodes in detect are not necessarily part of a single touch.

A value of 0 (default) disables this feature. This feature is not disabled by DISLOBJ.

Range: 0 (Off), 1 to 255

Typical: 32

CFG Field

RELAXCLOSESUP: When enabled, alters the suppression algorithm to treat groups of medium-sized touches differently. If groups of close touches are triggering the suppression mechanism incorrectly, enabling this mode may improve performance.

RELAXDIAGSUP: When enabled, alters the second level shape-based suppression algorithm to treat diagonal "close" touches (close in terms of simple search) differently. As such, this only applies when shape-based suppression is enabled. If lines of small touches are triggering the shape-based suppression mechanism incorrectly, enabling this mode may improve performance.

SUPTCHRPTEN: Enables the reporting of touches that are suppressed by the large object touch suppression. When this feature is enabled, suppressed touches are reported via the Multiple Touch Touchscreen T100 object with a specific indication. Suppressed touch reporting is enabled if this bit is set to 1, and disabled if set to 0.

EDGESUPSTRENGTH Field

This field specifies the Edge Suppression Strength. This controls the aggressiveness of touch suppression at the edges of the sensor.

A higher value reduces the aggressiveness and allows larger touches to be reported at the edges. A lower value increases the aggressiveness and allows only small touches to be reported at the edges. The default value of 0 means 73, which is the normal aggressiveness value. Values above 127 are capped to 127. A value of 255 will disable edge classification.

Range: 0 (73), 1 to 127, 255 (disable edge classification)

5.7.3 CONFIGURATION CHECKS

The Touch Suppression T42 object causes a configuration check to be performed in the following circumstances:

- If the object is enabled, when certain fields are changed

In addition, some fields will cause an automatic recalibration to be performed.

A configuration check may determine that a configuration error has occurred (for example, if a setting is set outside of its allowed range or a conflict has occurred between two settings). This is signaled to the host (see Section 3.3 “Command Processor T6 Object”). The device halts until the error has been corrected. To fix the error, check that all the object settings are within their allowed limits, as stated in the field descriptions.

TABLE 5-14: CONFIGURATION CHECKS FOR TOUCH SUPPRESSION T42 (PROCI_TOUCHSUPPRESSION_T42)

Field	Changing The Field Causes...		Effect of Configuration Checks On Field
	Configuration Check	Automatic Recalibration	
CTRL	Yes	No	None
MAXAPPRAREA	Yes	No	None
MAXTCHAREA	Yes	No	None
SUPSTRENGTH	Yes	No	None
SUPEXTTO	No	No	None
MAXNUMTCHS	No	No	None
SHAPESTRENGTH	Yes	No	None
SUPDIST	No	No	None
DISTHYST	No	No	None
MAXSCRNAREA	No	No	None
CFG	No	No	None
EDGESUPSTRENGTH	Yes	No	None

5.7.4 MESSAGES

No messages are generated directly for the Touch Suppression T42 object. Screen suppression messages are reported through the linked Multiple Touch Touchscreen T100 object (see Section 4.3.4 “Messages”).

Under NDA
© 2021 Microchip Technology Inc. and its subsidiaries

5.8 Passive Stylus T47 Object

5.8.1 INTRODUCTION

A Passive Stylus T47 object processes the measurement data received from a particular instance of a Multiple Touch Touchscreen T100 object. It allows the detection of touches that would otherwise be considered too small for the touchscreen. Additionally, there are controls to distinguish a stylus touch from an unwanted approaching finger.

Passive stylus touches are reported by the linked Multiple Touch Touchscreen T100 object with a touch area of 0 (see Section 4.3.4 “Messages”). Note that if a stylus touchdown occurs at the edge of the touchscreen, it will be tracked but not reported until it moves away from the edge unless the EDGEDETEN bit is set in the CTRL byte. This is to ensure that lightly touching fingertips on the edge of the touchscreen do not report as passive stylus touches.

Stylus logic works best if the Touch Threshold (TCHTHR) field in the Multiple Touch Touchscreen T100 object is set to a high value (see “TCHTHR Field”).

If the Noise Suppression T72 object supplies a minimum touch threshold, then the Passive Stylus T47 object will use this information to avoid incorrectly reporting noise as passive stylus touches.

Even if self capacitance is used as the primary measurement mechanism, passive stylus touches will still cause mutual capacitance scans to be requested on every cycle (see Appendix A “Measurement Processing”).

5.8.1.1 Edge Correction

The Passive Stylus T47 object allows edge correction to be applied for improved linearity at the edge of the screen on certain touchscreen designs for passive stylus touches (see Table 5-15).

See Section 4.3.1.2 “Edge Correction” for more information on edge correction and which other objects also contribute to edge correction functionality.

The following sections describe the fields that control edge correction for passive stylus touches:

“XEDGECTRL and YEDGECTRL Fields”
“XEDGEDIST and YEDGEDIST Fields”
“XEDGECTRLHI, DXXEDGECTRLHI and YEDGECTRLHI Fields”
“XEDGEDISTHI, DXXEDGEDISTHI and YEDGEDISTHI Fields”

TABLE 5-15: EDGE CORRECTION FIELDS

Setting	Fields	Effect	
No Specific X Edge Correction XEDGECTRL ENABLE = 0 XEDGECTRLHI ENABLE = 0	Passive Stylus T47 provides no specific edge correction for passive stylus touches. Edge correction (if required) is controlled by Multiple Touch Touchscreen T100.		No specific edge correction for passive stylus touches
No Specific Dual X Edge Correction DXXEDGECTRL ENABLE = 0 DXXEDGECTRLHI ENABLE = 0			
No Specific Y Edge Correction YEDGECTRL ENABLE = 0 YEDGECTRLHI ENABLE = 0			
Uniform X Edge Correction XEDGECTRL ENABLE = 1 XEDGECTRLHI ENABLE = 0	XEDGECTRL XEDGEDIST	Set the edge correction for both high and low X lines.	Low and high sense lines have same edge correction for passive stylus touches
Uniform Dual X Edge Correction DXXEDGECTRL ENABLE = 1 DXXEDGECTRLHI ENABLE = 0	DXXEDGECTRL DXXEDGEDIST	Set the edge correction for both high and low X lines for Dual X measurements.	
Uniform Y Edge Correction YEDGECTRL ENABLE = 1 YEDGECTRLHI ENABLE = 0	YEDGECTRL YEDGEDIST	Set the edge correction for both high and low Y lines.	

TABLE 5-15: EDGE CORRECTION FIELDS (CONTINUED)

Setting	Fields	Effect
Separate X Edge Correction XEDGECTRL ENABLE = 1 XEDGECTRLHI ENABLE = 1	XEDGECTRLHI XEDGEDISTHI	Set the edge correction for high X lines.
	XEDGECTRL XEDGEDIST	Set the edge correction for low X lines.
Separate Dual X Edge Correction DXXEDGECTRL ENABLE = 1 DXXEDGECTRLHI ENABLE = 1	DXXEDGECTRLHI DXXEDGEDISTHI	Set the edge correction for high X lines for Dual X measurements.
	DXXEDGECTRL DXXEDGEDIST	Set the edge correction for low X lines for Dual X measurements.
Separate Y Edge Correction YEDGECTRL ENABLE = 1 YEDGECTRLHI ENABLE = 1	YEDGECTRLHI YEDGEDISTHI	Set the edge correction for high Y lines.
	YEDGECTRL YEDGEDIST	Set the edge correction for low Y lines.

Low and high sense lines have different edge correction for passive stylus touches

5.8.2 CONFIGURATION

TABLE 5-16: CONFIGURATION FOR PASSIVE STYLUS T47 (PROCI_STYLUS_T47)

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	CTRL	Reserved				EDGEDETEN	Reserved		ENABLE
1	Reserved	Reserved							
2	CONTMAX	Maximum contact diameter							
3	STABILITY	Stability							
4	MAXTCHAREA	Maximum touch area							
5	AMPLTHR	Maximum touch amplitude							
6	STYSHAPE	Stylus shape adjustment							
7	HOVERSUP	Hovering finger suppression							
8	CONFTHR	Confidence threshold							
9	SYNCSPERX	ADC sets per X							
10	XPOSADJ	X position adjustment							
11	YPOSADJ	Y position adjustment							
12	CFG	Reserved		SUPSTY	Reserved				
13 – 19	Reserved	Reserved							
20	SUPSTYTO	Suppression timeout							
21	MAXNUMSTY	Maximum passive stylus touches							
22	XEDGECTRL	ENABLE	Reserved	CORRECTIONGRADIENT					
23	XEDGEDIST	X edge correction distance							
24	YEDGECTRL	ENABLE	Reserved	CORRECTIONGRADIENT					
25	YEDGEDIST	Y edge correction distance							
26	SUPTO	Touch suppression timeout							
27	SUPCLASSMODE	Suppression class mode							
28	DXXEDGECTRL	ENABLE	Reserved	CORRECTIONGRADIENT					
29	DXXEDGEDIST	Dual X X edge correction distance							
30	XEDGECTRLHI	ENABLE	Reserved	CORRECTIONGRADIENT					
31	XEDGEDISTHI	X high edge correction distance							
32	DXXEDGECTRLHI	ENABLE	Reserved	CORRECTIONGRADIENT					
33	DXXEDGEDISTHI	Dual X X high edge correction distance							
34	YEDGECTRLHI	ENABLE	Reserved	CORRECTIONGRADIENT					

Under NDA

© 2021 Microchip Technology Inc. and its subsidiaries

TABLE 5-16: CONFIGURATION FOR PASSIVE STYLUS T47 (PROCI_STYLUS_T47) (CONTINUED)

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
35	YEDGEDISTHI	Y high edge correction distance							
36	CFG2	Reserved						SEPMOVFILTEN	SPATFILTEN
37	MOVFILTER	Reserved				SPEEDRESP			
38	MOVSMOOTH	Movement smoothing							
39	MOVPRED	Movement prediction							
40	SATBXLO	Saturation Boundary X Low							
41	SATBXHI	Saturation Boundary X High							
42	SATBYLO	Saturation Boundary Y Low							
43	SATBYHI	Saturation Boundary Y High							
44	SATBDXXLO	Saturation Boundary DualX X Low							
45	SATBDXXHI	Saturation Boundary DualX X High							
46	MOVHYSTCFG	Reserved	MINSPEED			SPEEDRSP			

CTRL Field

This field enables or disables the Passive Stylus T47 object.

ENABLE: Enables the use of this Passive Stylus T47 object. The object is enabled if set to 1, and disabled if set to 0.

EDGEDETEN: Setting this bit allows the Passive Stylus T47 object to report passive stylus touches that touch down on the edges of the touchscreen. If this bit is set to 0 (disabled), the stylus touch will be tracked by the linked Multiple Touch Touchscreen T100 object, but not reported until it moves away from the edge.

CONTMAX Field

This field specifies the maximum contact diameter of the stylus that will be used. Note that this is the contact diameter of the tip and not the diameter of the body of the stylus. Objects larger than this size will not be tracked as passive stylus touches and instead will be treated as finger touches.

This setting should be adjusted to the lowest possible value that allows the desired maximum stylus size to be detected when the stylus is touched to the screen. Note, however, that if CONTMAX is too large, finger touches will not be detected; a touch is reported as a finger only if its diameter is larger than CONTMAX.

The maximum contact diameter is specified in units of 0.1 mm, where a value of 0 means 72 (7.2 mm).

Range: 0 (72 = 7.2 mm), 1 to 255 (in 0.1 mm increments)

STABILITY Field

This field is used to stabilize the stylus detection. Increasing the stability prevents small styluses from dropping out of detection. This causes the algorithm to continue reporting a stylus touch even if it temporarily becomes too small. In normal circumstances this setting should not need adjusting. The default value of 0 means that the setting is disabled.

Range: 0 (disabled), 1 to 255

MAXTCHAREA Field

This field specifies the Maximum Touch Area in the linked Multiple Touch Touchscreen T100 object that a touch can have and still be considered a stylus touch. The touch area is specified in channels. The default value of 0 means that only touches that are not detected by the Multiple Touch Touchscreen T100 object are considered as passive stylus touches.

Range: 0 to 255

AMPLTHR Field

This field specifies the Touch Amplitude Threshold. Any touches with an amplitude smaller than this value will be considered a stylus touch. Any touch with an amplitude larger than this will be reported as a finger, unless the touch has already been reported as a stylus and the amplitude has since grown, which brings other factors into play. The amplitude is reported by the AMPL touch status message field in the Multiple Touch Touchscreen T100 object (see Section 4.3.4 “Messages”). The default value of 0 means a Touch Amplitude of 40.

Range: 0 (40), 1 to 255

STYSHAPE Field

This field controls how the hovering finger suppression treats the stylus. This is used to adjust for the specific parameters of the touchscreen, such as signal strength and pitch. Increasing STYSHAPE makes touches look “bigger” or “flatter” to the hovering finger suppression algorithm; decreasing STYSHAPE makes the algorithm see touches as “smaller” or “more pointed”. A value of 0 means 16.

Range: 0 (16), 1 to 255

HOVERSUP Field

This field controls the aggressiveness of the hovering finger suppression logic. Higher settings will make this logic less aggressive. In this case a hovering finger is more likely to be tracked as a stylus touch. A value of 0 means 150 and a value of 255 disables this feature.

Range: 0 (150), 1 to 254, 255 (disable)

CONFTHR Field

This field controls the Confidence Threshold. It is specified in units of measurement cycles + 1. Note that these cycles are fast DI cycles and overlap with the Multiple Touch Touchscreen T100 TCHDI fast DI cycles.

Higher settings will reduce the speed of the touchdown detection of a stylus touch but will make a stylus touch more likely to be identified correctly. The default value of 0 means there is no delay in detecting passive stylus touches.

Range: 0 (no delay), 1 to 255 (measurement cycles + 1)

SYNCSPERX Field

This field specifies the minimum ADC sets per X line to be used when passive stylus touches are being detected (see Section 6.8.1 “Introduction”). The actual ADCs per X line value used is the maximum of this SYNCSPERX field (if in detect) and the equivalent values in other objects (see Section 6.8.1 “Introduction” for details).

If SYNCSPERX is set to 0, this field has no effect.

Range: 0 (none), 1 to 255 (number of Sync Groups)

XPOSADJ, YPOSADJ Fields

These controls adjust the calculated stylus position to compensate for underestimation or overestimation of the distance from a node due to the small contact area of the stylus.

Larger magnitude values will make larger adjustments to the non-linearity. Positive values push stylus positions away from the electrodes. Negative values pull stylus positions towards the electrodes.

Range: 0 (no adjustment), –128 to 127

CFG Field

SUPSTY: When this bit is set to 1, passive stylus touches are suppressed if any other (non-stylus) touches are in detect. The suppression ends when all touches have been released, including any suppressed passive stylus touches.

SUPSTYTO Field

This field sets up a timeout to continue to suppress passive stylus touches (suppressed via CFG field SUPSTY bit) until a specified time after a non-stylus touch has been removed. The duration of the timeout is specified in units of 200 ms.

Under NDA
© 2021 Microchip Technology Inc. and its subsidiaries

Range: 0 to 255 (51 seconds)

MAXNUMSTY Field

This field specifies the maximum number of passive stylus touches that will be reported before all touches are suppressed. If more than MAXNUMSTY are detected, the touchscreen suppresses all touches. This includes those touches already detected. The touches remain suppressed until all touches are released or a recalibration occurs.

This field may be useful in helping to suppress spurious touches that could be interpreted as passive stylus touches, such as from a headphone cable accidentally brushing the touchscreen.

A value of 0 ensures that all passive stylus touches will be reported (that is, no suppression)

Range: 0 (no suppression), 1 to 15 (maximum number of touches minus 1)

XEDGECTRL and YEDGECTRL Fields

These fields, together with the XEDGEDIST and YEDGEDIST fields, override the Multiple Touch Touchscreen T100 edge correction controls to provide specific edge correction for passive stylus touches, as determined by ENABLE bits. Specifically, XEDGECTRL and YEDGECTRL override the Multiple Touch Touchscreen T100 XEDGECFG and YEDGECFG fields (see "XEDGECFG, DXXEDGECFG and YEDGECFG Fields"). Note that these fields affect passive stylus touches only.

See Section 5.8.1.1 "Edge Correction" for more information on edge correction for passive stylus touches.

XEDGEGRADIENT/YEDGEGRADIENT: The gradient to apply.

XEDGEGRADIENT/YEDGEGRADIENT Range: 0 to 63.

ENABLE: If this bit is set to 1, enables specific edge correction for passive stylus touches on the X or Y axis, as appropriate (See Section 5.8.1.1 "Edge Correction"). This is either uniform edge correction or separate edge correction, depending on the setting for XEDGECTRLHI/YEDGECTRLHI ENABLE (see "XEDGECTRLHI, DXXEDGECTRLHI and YEDGECTRLHI Fields").

If this bit is set to 0, specific edge correction settings for passive stylus touches are not used and the Multiple Touch Touchscreen T100 edge correction will apply to passive stylus touches.

XEDGEDIST and YEDGEDIST Fields

NOTE	These fields affect passive stylus touches only; other touch types are unaffected.

These fields override the XEDGEDIST/YEDGEDIST fields of the Multiple Touch Touchscreen T100 edge correction controls to provide specific edge correction for passive stylus touches, as determined by XEDGECTRL and YEDGECTRL ENABLE bits. They specify the distance to start applying the gradient, in units of 0 to 1023 (not in units of the current resolution).

See Section 5.8.1.1 "Edge Correction" for more information on edge correction for passive stylus touches.

Range: 0 to 255

SUPTO Field

NOTE	This field should not be confused with SUPSTYTO (see "SUPSTYTO Field").

This field configures the Suppression Time Out. This is the time to wait, once any of the touch classes defined by SUPCLASSMODE is seen, from when the last touch is lifted off the screen before allowing reporting of touches other than passive stylus touches. The timeout period is reset if a second set of touches lands in the time period between the lift-off of the first set of touches and the end of the waiting period.

The Suppression Time Out is specified in units of 200 ms.

Range: 0 (off), 1 to 255 (51 seconds)

SUPCLASSMODE Field

This field specifies the Suppression Class Mode; that is, the touch classes that trigger the behavior of SUPTO. This field takes one of the values in Table 5-17.

TABLE 5-17: SUPPRESSION CLASS MODES

Code	Mode
0	Off
1	Stylus and Palm Touch
2	Palm Touch
3	Stylus
4	Stylus or Palm Touch

DXXEDGECTRL Field

NOTE This field affects passive stylus touches only; other touch types are unaffected.

This field, together with the DXXEDGEDIST field, overrides the Multiple Touch Touchscreen T100 edge correction controls to provide specific edge correction for passive stylus touches for Dual X measurements, as determined by the ENABLE bit. Specifically, DXXEDGECTRL overrides the Multiple Touch Touchscreen T100 DXXEDGECFG field (see "XEDGECFG, DXXEDGECFG and YEDGECFG Fields").

See Section 5.8.1.1 "Edge Correction" for more information on edge correction for passive stylus touches.

CORRECTIONGRADIENT: The gradient to apply.

CORRECTIONGRADIENT Range: 0 to 63.

ENABLE: If this bit is set to 1, enables specific edge correction for passive stylus touches when Dual X measurements are being taken (See Section 5.8.1.1 "Edge Correction"). This is either uniform edge correction or separate edge correction, depending on the setting for DXXEDGECTRLHI ENABLE (see "XEDGECTRLHI, DXXEDGECTRLHI and YEDGECTRLHI Fields").

If this bit is set to 0, specific edge correction settings for passive stylus touches are not used and the Multiple Touch Touchscreen T100 edge correction will apply to passive stylus touches in Dual X mode.

DXXEDGEDIST Fields

NOTE This field affects passive stylus touches only; other touch types are unaffected.

This field overrides the Multiple Touch Touchscreen T100 DXXEDGEDIST field to provide specific edge correction for passive stylus touches when Dual X measurements are being taken, as determined by the DXXEDGECTRL ENABLE bit. Specifically, the DXXEDGEDIST field specifies the distance to start applying the gradient, in units of 0 to 1023 (not in units of the current resolution).

See Section 5.8.1.1 "Edge Correction" for more information on edge correction for passive stylus touches.

Range: 0 to 255

XEDGECTRLHI, DXXEDGECTRLHI and YEDGECTRLHI Fields

NOTE This field affects passive stylus touches only; other touch types are unaffected.

These fields, together with the XEDGEDISTHI, DXXEDGEDISTHI and YEDGEDISTHI fields, provide separate high edge correction for passive stylus touches only. The ENABLE bit for the appropriate field must be set to 1 for this feature to work. Note that the ENABLE bit of the XEDGECTRL/DXXEDGECTRL/YEDGECTRL field must also be set to 1 in order for these fields to have an effect.

These fields allow separate control of the edge correction that is applied to the high and low sense line edges, so that sensors with different non-linear responses at the high and low sense lines of the X and Y axes can be corrected for.

See Section 5.8.1.1 "Edge Correction" for more information on edge correction for passive stylus touches.

CORRECTIONGRADIENT: The gradient to apply.

CORRECTIONGRADIENT Range: 0 to 63.

ENABLE: If set to 1, CORRECTIONGRADIENT and XEDGEDISTHI/DXXEDGEDISTHI/YEDGEDISTHI provide separate high edge correction for passive stylus touches only.

Under NDA

© 2021 Microchip Technology Inc. and its subsidiaries

XEDGEDISTHI, DXXEDGEDISTHI and YEDGEDISTHI Fields

NOTE	This field affects passive stylus touches only; other touch types are unaffected.

These fields provide separate edge correction for passive stylus touches only, as determined by the XEDGECTRLHI/DXXEDGECTRLHI/YEDGECTRLHI ENABLE bit. Specifically, these fields specify the distance to start applying the gradient, in units of 0 to 1023 (not in units of the current resolution).

See Section 5.8.1.1 “Edge Correction” for more information on edge correction for passive stylus touches.

Range: 0 to 255.

CFG2 Field

SPATFILTEN: If this bit is set to 1, the deltas measured for the stylus will be post processed. This can improve linearity for some styluses.

SEPMOVFILTEN: Enables Separate Movement Filter. If this bit is set to 1, the movement filter settings defined in the MOVFILTER, MOVSMOOTH and MOVPRED fields are used for all passive stylus touches instead of the equivalent fields defined in Multiple Touch Touchscreen T100.

MOVFILTER, MOVSMOOTH and MOVPRED Fields

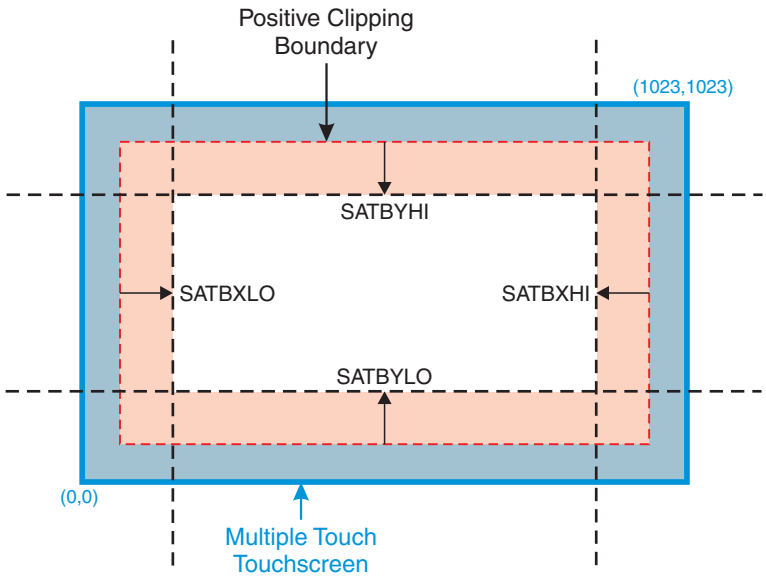
These fields provide separate movement filter control for passive stylus touches. They have the same meaning as the Multiple Touch Touchscreen T100 equivalents, but note that use of the median filter and disabling the filter are still controlled by the Multiple Touch Touchscreen T100 object.

If the movement filter is enabled by the Multiple Touch Touchscreen T100 object, and the CFG2 SEPMOVFILTEN bit is set (see “CFG2 Field”), then these MOVFILTER SPEEDRESP, MOVSMOOTH and MOVPRED settings are used instead of the equivalent Multiple Touch Touchscreen T100 settings. Note that this applies to passive stylus touches only; other touches will be controlled by the Multiple Touch Touchscreen T100 movement filter settings.

SATBXLO, SATBXHI, SATBYLO and SATBYHI Fields

These fields define a Saturation Boundary beyond which the passive stylus reported position will be saturated to the minimum or maximum value. The boundary will lie inside any positive clipping region (if enabled) or the touchscreen edge (see Figure 5-12).

FIGURE 5-12: SATURATION BOUNDARY



Note that positions will not be locked within the passive stylus boundary. That is, the position will always be unlocked until the passive stylus enters the clipping region (which may lock depending on the state of Multiple Touch Touchscreen T100 CFG1 DISLOCK) or leaves the screen. See "XRANGE/YRANGE, XLOCLIP/YLOCLIP and XHICLIP/YHICLIP Fields" for more information on the screen clipping region.

Range: 0 to 255 (10-bit reporting units)

SATBDXXLO and SATBDXXHI Fields

Long Name: DualX X Low/High

These fields define alternative Saturation Boundary values for Dual X mode. If Dual X is in operation, and these values are non-zero, then these values are used for the saturation boundary. Otherwise, the SATBXLO and SATBXHI values are used.

Range: 0 (use SATBXLO and SATBXHI values), 1 to 255 (10-bit reporting units)

MOVHYSTCFG Field

This field overrides the Multiple Touch Touchscreen T100 MOVHYSTCFG field for passive stylus touches, but only if CFG2 SEPMOVFILTEN bit is also set to 1.

See "MOVHYSTCFG Field" for more information on this field.

Range: See "MOVHYSTCFG Field"

5.8.3 CONFIGURATION CHECKS

A Passive Stylus T47 object causes a configuration check to be performed in the following circumstances:

- If the object is enabled, when certain fields are changed

A configuration check may determine that a configuration error has occurred (for example, if a setting is set outside of its allowed range or a conflict has occurred between two settings). This is signaled to the host (see Section 3.3.3 "Messages"). The device halts until the error has been corrected. To fix the error, check that all the object settings are within their allowed limits, as stated in the field descriptions.

TABLE 5-18: CONFIGURATION CHECKS FOR PASSIVE STYLUS T47 (PROCI_STYLUS_T47)

Field	Changing The Field Causes...		Effect of Configuration Checks On Field
	Configuration Check	Automatic Recalibration	
CTRL	Yes	Yes	None
CONTMAX	Yes	No	None
STABILITY	Yes	No	None
MAXTCHAREA	No	No	None
AMPLTHR	No	No	None
STYSHAPE	No	No	None
HOVERSUP	No	No	None
CONFTHR	No	No	None
SYNCSPERX	No	No	None
XPOSADJ	No	No	None
YPOSADJ	No	No	None
CFG	No	No	None
SUPSTYTO	No	No	None
MAXNUMSTY	No	No	None
XEDGECTRL	No	No	None
XEDGEDIST	No	No	None
YEDGECTRL	No	No	None

Under NDA

© 2021 Microchip Technology Inc. and its subsidiaries

TABLE 5-18: CONFIGURATION CHECKS FOR PASSIVE STYLUS T47 (PROCI_STYLUS_T47) (CONTINUED)

Field	Changing The Field Causes...		Effect of Configuration Checks On Field
	Configuration Check	Automatic Recalibration	
YEDGEDIST	No	No	None
SUPTO	No	No	None
SUPCLASSMODE	No	No	None
DXXEDGECTRL	No	No	None
DXXEDGEDIST	No	No	None
XEDGECTRLHI	No	No	None
XEDGEDISTHI	No	No	None
DXXEDGECTRLHI	No	No	None
DXXEDGEDISTHI	No	No	None
YEDGECTRLHI	No	No	None
YEDGEDISTHI	No	No	None
CFG2	Yes	No	None
MOVFILTER	Yes	No	None
MOVSMOOTH	Yes	No	None
MOVPRED	Yes	No	None
SATBXLO	No	No	None
SATBXHI	No	No	None
SATBYLO	No	No	None
SATBYHI	No	No	None
SATBDXXLO	No	No	None
SATBDXXHI	No	No	None
MOVHYSTCFG	No	No	None

5.8.4 MESSAGES

No messages are generated directly for the Passive Stylus T47 object. Passive stylus touches are reported by the linked Multiple Touch Touchscreen T100 object with a touch area of 0 (see Section 4.3.4 “Messages”).

5.9 Shieldless T56 Object

5.9.1 INTRODUCTION

The Shieldless T56 object allows the use of shieldless touch sensor stacks. Specifically, the Shieldless T56 object uses Optimal Integration to adjust the integration window and timing to maximize the signal-to-noise ratio from the sensor.

Note that display noise suppression for shieldless designs is provided by the filtering features of the Lens Bending T65 object (See Section 5.10 “Lens Bending T65 Object”).

5.9.2 CONFIGURATION

TABLE 5-19: CONFIGURATION FOR SHIELDLESS T56 (PROCI_SHIELDLESS_T56)

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	CTRL	Reserved						RPTEN	ENABLE
1	Reserved	Reserved							
2	OPTINT	Reserved							OPTINTEN
3	INTTIME	Optimal integration time							
4	INTDELAY[0]	Optimal integration delay for X0							
...									
35	INTDELAY[31]	Optimal integration delay for X31							

CTRL Field

ENABLE: Enables the use of this Shieldless T56 object. The object is enabled if set to 1, and disabled if set to 0.

RPTEN: Allows the object to send status messages to the host through the Message Processor T5 object. Reporting is enabled if set to 1, and disabled if set to 0.

OPTINT Field

This field sets the options for an Optimal Integration.

OPTINTEN: If this bit is set to 1, a Delay Compensation per X line is applied and the specified optimal integration time (INTIME) is used. If this bit is set to 0, Delay Compensation is not applied per X line, and the Acquisition Configuration T8 object’s Integration Window is applied.

INTTIME Field

This field sets the Optimal Integration Time. A non-zero value specifies the Optimal Integration Time plus 1. The units for this field are 24 MHz clock cycles (41.67 ns).

Range: 0 (no time), 1 to 255

INTDELAY[] Fields

These fields set the Optimal Integration Delays for each of the X lines when Optimal Integration is enabled. A non-zero value specifies the Optimal Integration Delay plus 1 for X line *n*. The units for this field are 24 MHz clock cycles (41.67 ns).

Range: 0 (no delay), 1 to 255

5.9.3 CONFIGURATION CHECKS

The Shieldless T56 object does not perform configuration checks. However, changing the values of the fields in the Shieldless T56 object will cause an automatic recalibration to be performed (see Table 5-20).

TABLE 5-20: CONFIGURATION CHECKS FOR SHIELDLESS T56 (PROCI_SHIELDLESS_T56)

Field	Changing The Field Causes...		Effect of Configuration Checks On Field
	Configuration Check	Automatic Recalibration	
CTRL	No	Yes (1)	None

Under NDA

© 2021 Microchip Technology Inc. and its subsidiaries

TABLE 5-20: CONFIGURATION CHECKS FOR SHIELDLESS T56 (PROCI_SHIELDLESS_T56)

Field	Changing The Field Causes...		Effect of Configuration Checks On Field
	Configuration Check	Automatic Recalibration	
OPTINT	No	Yes (2)	None
INTTIME	No	Yes	None
INTDELAY[]	No	Yes	None

Note 1: If the ENABLE bit is toggled on or off.
2: If the OPTINTEN bit is toggled on or off.

5.9.4 MESSAGES

The message data for the Shieldless T56 object is shown in Table 5-21.

TABLE 5-21: MESSAGE DATA FOR SHIELDLESS T56 (PROCI_SHIELDLESS_T56)

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	STATUS	Reserved							UNCAL

STATUS Field

This field reports the current state of the Shieldless T56 object.

UNCAL: Indicates that optimal integration has been enabled but no on-chip calibrated value has been set for the integration time.

NOTE On-chip calibration is not possible on the mXT640UD-CCUBHA1 Family. However, UNCAL bit set always set to 1 in any generated message, even though on-chip calibration is not supported.

5.10 Lens Bending T65 Object

5.10.1 INTRODUCTION

The Lens Bending T65 object implements the lens bending correction algorithm. When the sensor suffers from the screen deformation (that is, lens bending) the delta values obtained by the normal measurement procedure are corrupted by the low frequency disturbance component (that is, the sensor measurement's bend). The amount of bending depends on the following:

- The mechanical and electrical characteristics of the sensor
- The amount and location of the force applied by the user touch to the sensor

The Lens Bending algorithm aims to eliminate this disturbance signal from the reported deltas, and measure the bend component at the same time.

On the mXT640UD-CCUBHA1 Family there are 3 instances of the Lens Bending T65 object. Each instance corrects for the effects of lens distortion from a different measurement source. See Table 5-22 for details.

TABLE 5-22: LENS BENDING T65 INSTANCES

Instance	Purpose
Instance 0	Corrects the mutual capacitance measurements.
Instance 1	Corrects the self capacitance touch Y line measurements.
Instance 2	Corrects the self capacitance touch X line measurements.

5.10.2 CONFIGURATION

TABLE 5-23: CONFIGURATION FOR LENS BENDING T65 (PROCI_LENSBENDING_T65)

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	CTRL	DISHIST	Reserved		DISRELEASE	DISPRESS	Reserved	RPTEN	ENABLE
1	GRADTHR	Gradient Threshold							
2	YLONOISEMUL	Y low noise multiplier LSByte							
3		Y low noise multiplier MSByte							
4	YLONOISEDIV	Y low noise divisor LSByte							
5		Y low noise divisor MSByte							
6	YHINOISEMUL	Y high noise multiplier LSByte							
7		Y high noise multiplier MSByte							
8	YHINOISEDIV	Y high noise divisor LSByte							
9		Y high noise divisor MSByte							
10	LPFILTCOEF	Low pass filter coefficient							
11	FORCESCALE	Force scale LSByte							
12		Force scale MSByte							
13	FORCETHR	Force threshold							
14	FORCETHRHYST	Force threshold hysteresis							
15	FORCEDI	Force DI							
16	FORCEHYST	Force update hysteresis							
17	ATCHRATIO	DSRATIO				Reserved	ATCHRATIO		
18 – 19	Reserved	Reserved							
20	EXFRCTHR	Excessive force threshold							
21	EXFRCTHRHYST	Excessive force threshold hysteresis							
22	EXFRCTO	Excessive force timeout							

CTRL Field

ENABLE: Enables or disables the use of this instance of the Lens Bending T65 object. The instance is enabled if set to 1, and disabled if set to 0.

Under NDA

© 2021 Microchip Technology Inc. and its subsidiaries

RPTEN: Allows this object to send messages to the host through the Message Processor T5 object. Specifically it enables reporting of the estimated amount of deformation. Reporting is enabled if set to 1, and disabled if set to 0.

DISPRESS: If set to 1, disables press messages related to the amount of deformation.

DISRELEASE: If set to 1, disables release messages related to the amount of deformation.

DISHIST: Disables inter-line history. The inter-line history feature stores the results of the calculations on one line so that they can be used to affect the next. If this bit is set to 1, the use of this inter-line history is disabled.

NOTE	For an Lens Bending T65 instance that processes only one line of data (that is X or Y lines only), no inter-line history should be used. The DISHIST bit should therefore be set to 1 to disable the history.

GRADTHR Field

This field specifies the maximum amount of bend gradient present in the measured deltas. Sensor measurements with gradient smaller than GRADTHR are considered part of the bend and excluded from further processing. Sensor measurements with a gradient equal or greater than the GRADTHR are considered part of the touch and preserved.

Note that if the DSRATIO setting is changed (see "ATCHRATIO Field"), the gradient threshold should be adjusted accordingly.

The range of values for GRADTHR setting is 0 to 255 in units of 8-bit deltas. The default value of zero means 15.

Range: 0 (15), 1 to 255

YLONOISEMUL, YLONOISEDIV, YHINOISEMUL and YHINOISEDIV Fields

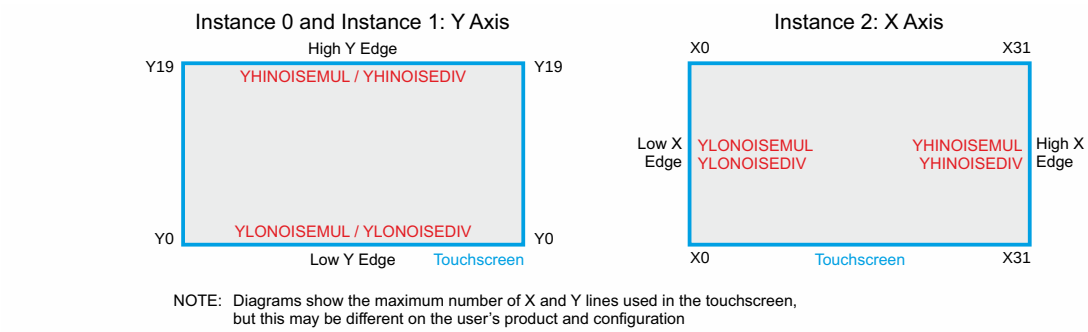
NOTE	On Lens Bending T65 Instance 2 for Self Capacitance Touch X axis, these fields apply to the edge X lines (even though their names suggest they apply to Y lines).

These fields allow the noise that is picked up at the edge of the screen to be scaled when the amount of noise at the edge is different to the rest of the screen.

These fields apply to the Y or X lines as follows (see Figure 5-13):

- YLONOISEMUL and YLONOISEDIV determine the scaling applied to the low Y edge (Instance 0 and Instance 1) or low X edge (Instance 2) of the screen.
- YHINOISEMUL and YHINOISEDIV determine the scaling applied to the high Y edge (Instance 0 and Instance 1) or high X edge (Instance 2) of the screen.

FIGURE 5-13: NOISE SCALING CONTROLS



These four fields are specified using 8.8 format, where the upper 8 bits represent the integer part of the number and the lower 8 bits represent the fractional part of the number in units of 1/256. For example, a value of 1.128 represents an integer of 1 and a decimal fraction of 0.5. The default value of 0 means 256 (1.0); that is multiply by 1.

Range: 0 (256), 1 to 65535

LPFILTCOEF Field

This field specifies the low pass filter coefficient used to separate touches and bend. Increasing the value of LPFILTCOEF makes the algorithm less selective between touch and bend, but enables the algorithm to be applied with fast-changing bend.

Higher values of LPFILTCOEF should be used with higher values of the GRADTHR parameter.

The range of values is 0 to 15. The default value of zero means 5.

Range: 0 (5), 1 to 15

FORCESCALE Field

This field scales the size of the reported deformation (bend size). The value is represented in 8.8 format, where a default of 0 means 256. (see the YLONOISEMUL, YLONOISEDIV, YHINOISEMUL and YHINOISEDIV fields for details).

Range: 0 (256), 1 to 65535

FORCETHR Field

This field configures the Force Threshold above which a force event is reported. If the force is greater than or equal to this threshold, then a force event is reported. In addition, when the force transitions from below this threshold to above this threshold, a press event is reported. The default value of 0 means 64.

Range: 0 (64), 1 to 255

FORCETHRHYST Field

This field configures the Force Threshold Hysteresis below which a release event is reported. If the force is transitions below FORCETHR – FORCEHYST, then a release event is reported and force events stop being reported.

Range: 0 to 255

FORCEDI Field

This field specifies the Detect Integration for the deformation press and release events. To suppress false detections caused by spurious events like electrical noise, the device incorporates a detection integrator (FORCEDI) counter mechanism to provide signal filtering. A counter is incremented each measurement cycle that a force is detected. When this counter reaches the FORCEDI setting limit the force is finally declared to be present. If on any acquisition a force is not seen to exceed FORCETHR the counter is cleared and the process has to start from the beginning. It takes FORCEDI cycles from when the force is detected to when the force is reported by the $\overline{\text{CHG_I2C}}$ or $\overline{\text{CHG_SPI}}$ line, with a minimum of 2 for the first touch, or a minimum of 1 for subsequent touches.

NOTE	The detection integration counter associated with FORCEDI is processed only when the associated measurement for the Lens Bending T65 instance is made (see Table 5-22 on page 128). This may result in many more cycles being performed before FORCEDI elapses if the configuration is such that the associated measurement is not made on every cycle.

An opposite process is applied when a force leaves detection. The counter is decremented each cycle that the force is below FORCETHR – FORCETHRHYST, and incremented again when the force exceeds FORCETHR. When the counter reaches zero, the force is finally declared to be out of detect. For FORCEDI values of 0 and 1, it takes 1 cycle from when the force is no longer detected to it being reported by the $\overline{\text{CHG_I2C}}$ or $\overline{\text{CHG_SPI}}$ line pin; for FORCEDI values of 2 to 255, it takes (FORCEDI + 1) cycles.

The effects of various FORCEDI values are summarized in Table 5-24.

TABLE 5-24: FORCEDI

Condition	Number of Cycles			
	FORCEDI = 0	**FORCEDI = 1**	**FORCEDI = 2**	**FORCEDI ≥ 3 (*n*)**
First touch down	2	2	3	*n*+1
Next Touch down	1	1	2	n
Touch release	1	1	3	*n*+1

Under NDA

© 2021 Microchip Technology Inc. and its subsidiaries

Range: 0 (1), 1 to 255

FORCEHYST Field

This field configures the Force Hysteresis for the reporting threshold (that is, the amount of change necessary for the force to be reported). A value of 255 disables the force value reporting.

Range: 0 to 254, 255 (disable reporting)

ATCHRATIO Field

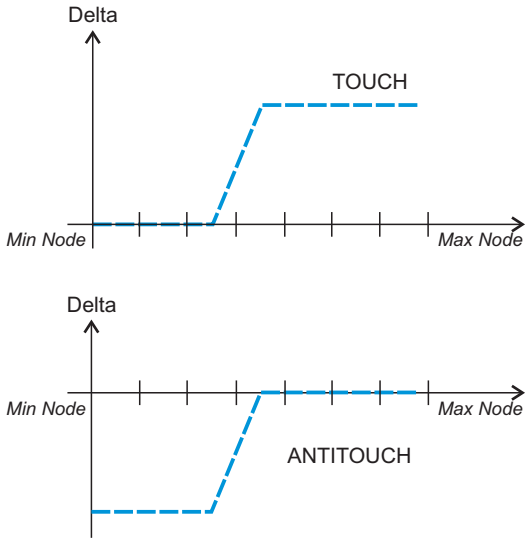
ATCHRATIO: Specifies the ratio between the positive and negative slew rate limit. This parameter is used to bias the touch estimate towards a touch where the touch covers more than 50% of the nodes on the line being measured. Due to the presence of LCD noise, a touch covering ≥ 50% of the nodes on the line being measured can be interpreted either as a touch or as an antitouch as shown in Figure 5-14.

This ambiguity in interpretation of the measured delta is resolved using ATCHRATIO. A higher ATCHRATIO biases the lens bending algorithm towards a touch.

If the value specified for the ATCHRATIO field exceeds the maximum value allowed, it is internally capped to the maximum (4).

ATCHRATIO Range: 0 (2), 1 to 4

FIGURE 5-14: LARGE TOUCH INTERPRETATION AS TOUCH OR ANTITOUCH DUE TO LCD NOISE



DSRATIO: Defines the Downsampling Ratio. This is how many lines should be used in the anti touch algorithm. For example, if DSRATIO is set to 2, then every other line is analyzed, if it is set to 3 then every third line is analyzed, and so on. This reduces the processing time for lens bending on larger screens. The range for DSRATIO is 0 to 4, where the default value of 0 means 2. In addition, a value of 15 means disable downsampling.

If the value specified for the DSRATIO field exceeds the maximum value allowed, it is set to the default value of 2.

DSRATIO Range: 0 (2), 1 to 4, 15 (downsampling disabled)

EXFRCTHR Field

This field specifies the Excessive Force Threshold. The effect of this field depends on the object instance and the type of data:

- For a Lens Bending T65 instance that handles self capacitance measurement type data, if the reported deformation (bend size) on the current primary measurement type exceeds this threshold, then the device will start to use the mutual capacitance measurement type. Note that the deformation that is checked against this threshold is the one being reported (that is, the one being re-scaled by the FORCESCALE control).
- For a Lens Bending T65 instance that handles mutual capacitance measurement type data, if the reported deformation on the current primary measurement type exceeds this threshold, then the device will generate a positive edge event for handling via the Dynamic Configuration Controller T70 object.

The range for EXFRCTHR is 0 to 255 (in units of force), where a value of 0 disables this feature.

Range: 0 (disabled), 1 to 255

EXFRCTHRHYST Field

This field specifies the Excessive Force Threshold Hysteresis. The effect of this field depends on the object instance and the type of data:

- For a Lens Bending T65 instance that handles self capacitance measurement type data, if the reported deformation (bend size) on the current primary measurement type is below (EXFRCTHR – EXFRCTHRHYST), then the system will stop using the mutual capacitance measurement type once the timeout has expired (see "EXFRCTO Field"), but only if the mutual capacitance measurement type is already running. Note that the deformation that is checked against this threshold is the one being reported (that is, the one being re-scaled by the FORCESCALE field).
- For a Lens Bending T65 instance that handles mutual capacitance measurement type data, if the reported deformation (bend size) on the current primary measurement type is below (EXFRCTHR – EXFRCTHRHYST), then the device will generate a negative edge event for handling via the Dynamic Configuration Controller T70 object, but only once the timeout has expired (see "EXFRCTO Field").

Range: 0 to (EXFRCTHR – 1)

EXFRCTO Field

This field specifies the Excessive Force Timeout. The effect of this field depends on the object instance and the type of data:

- For a Lens Bending T65 instance that handles self capacitance measurement type data, this timeout determines how long the system will use the mutual capacitance measurement type after the reported deformation (bend size) has dropped below (EXFRCTHR – EXFRCTHRHYST). Note that if the reported deformation (bend size) goes above (EXFRCTHR – EXFRCTHRHYST) during the timeout period, the timer is reset.
- For a Lens Bending T65 instance that handles mutual capacitance measurement type data, this timeout determines if a negative edge event is generated for handling via the Dynamic Configuration Controller T70 object.

The timeout is specified in units of 200 ms, where the default value of 0 means no timeout.

Range: 0 (no timeout), 1 to 255

5.10.3 CONFIGURATION CHECKS

An instance of the Lens Bending T65 object causes a configuration check to be performed in the following circumstances:

- If the instance of the object is enabled, when certain fields are changed

A configuration check may determine that a configuration error has occurred (for example, if a setting is set outside of its allowed range or a conflict has occurred between two settings). This is signaled to the host (see Section 3.3.3 "Messages". The device halts until the error has been corrected. To fix the error, check that all the object settings are within their allowed limits, as stated in the field descriptions.

TABLE 5-25: CONFIGURATION CHECKS FOR LENS BENDING T65 (PROCI_LENSBENDING_T65)

Field	Changing The Field Causes...		Effect of Configuration Checks On Field
	Configuration Check	Automatic Recalibration	
CTRL	Yes (1)	Yes	None

Under NDA
© 2021 Microchip Technology Inc. and its subsidiaries

TABLE 5-25: CONFIGURATION CHECKS FOR LENS BENDING T65 (PROCI_LENSBENDING_T65) (CONTINUED)

Field	Changing The Field Causes...		Effect of Configuration Checks On Field
	Configuration Check	Automatic Recalibration	
GRADTHR	No	No	None
YLONOISEMUL	No	Yes	None
YLONOISEDIV	No	Yes	None
YHINOISEMUL	No	Yes	None
YHINOISEDIV	No	Yes	None
LPFILTCOEF	No	No	None
FORCESCALE	No	No	None
FORCETHR	No	No	None
FORCETHRHYST	No	No	None
FORCEDI	No	No	None
FORCEHYST	No	No	None
ATCHRATIO	No	No	None
EXFRCTHR	No	No	None
EXFRCTHRHYST	No	No	None
EXFRCTO	No	No	None

Note 1: If the ENABLE bit is toggled on or off.

5.10.4 MESSAGES

The message data for the Lens Bending T65 object is shown in Table 5-26.

TABLE 5-26: MESSAGE DATA FOR LENS BENDING T65 (PROCI_LENSBENDING_T65)

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	STATUS	Detect	Press	Release	Reserved				
2	FORCE	Force Level							

STATUS Field

DETECT: This bit indicates the current force state. If the current force level is above FORCETHR, this bit is set to 1; if current force level is below (FORCETHR – FORCETHRHYST), this bit is set to 0.

PRESS: Indicates a press event. This bit is set if DETECT changes from 0 to 1; that is, the level of deformation was less than (FORCETHR – FORCETHRHYST) and has now changed to greater than FORCETHR, and has been so for FORCEDI cycles.

RELEASE: Indicates a release event (DETECT = 0). This bit is set if DETECT changes from 1 to 0; that is, the level of deformation was above FORCETHR and has now changed to less than (FORCETHR – FORCETHRHYST), and has been so for FORCEDI cycles.

FORCE Field

This field reports the current calculated force. This is rescaled by FORCESCALE and updated once it changes by at least FORCEHYST.

5.11 Noise Suppression T72 Object

5.11.1 INTRODUCTION

The Noise Suppression T72 object provides a noise suppression algorithm to suppress the effects of external noise.

5.11.1.1 Noise Notification Mechanisms

The Noise Suppression T72 object needs to be made aware of when a noise source appears and disappears (for example, when a charger is turned on or off) or when the current noise level rises to a level beyond which the current acquisition parameters can cope. The following notification mechanisms are available:

- **Software trigger** – The host can set the NOISEON bit in the CFG1 field when a noise source appears. For example, this bit should be set when a noisy charger is plugged into the user's product. Note that the alternative use of the NOISEON bit in the CALCFG1 field is not recommended.

IMPORTANT!	Enabling and disabling the Noise Suppression T72 object dynamically using the CTRL ENABLE bit in response to noise is not recommended, as this can cause unstable behavior. The correct procedure is to enable the object at startup and then use the notification trigger whenever noise is detected.

5.11.1.2 Noise Measurement

The Noise Suppression T72 object schedules the acquisition of numerous noise measurements so that suitable actions can be taken to best minimize the effects of such noise (see Section 5.11.1.3 “Transitions Between Noise States” and Section 5.11.1.4 “Frequency Hopping”).

The NLGAINDUALX and NLGAINSINGX fields determine the gains to be used for the noise lines measurements.

Suggested initial settings for the Noise Suppression T72 object are given in Table 5-27.

TABLE 5-27: SUGGESTED INITIAL SETTINGS – NOISE MEASUREMENT

Field	Suggested Initial Setting
NLGAINDUALX	Field needs tuning; depends on whether Dual X is running or not. Start with Multiple Touch Touchscreen T100 Dual X gain.
NLGAINSINGX	Field needs tuning; depends on whether Dual X is running or not. Start with Multiple Touch Touchscreen T100 Single X gain.

5.11.1.3 Transitions Between Noise States

As the noise suppression algorithm is notified of increasing or decreasing levels of noise, it transitions between the STABLE, NOISY and VERY_NOISY states (see Figure 5-15). In each state it can dynamically load a new set of acquisition parameters to cope with the current level of noise. This provides a framework within which the behavior of the Noise Suppression T72 object can be modified to provide better noise handling at the expense of touch performance and/or touch message rate. The thresholds that determine the transitions between the states are specified by the NOISLOWNLTHR, VNOILOWNLTHR, STABHIGHNLTHR and NOISHIGHNLTHR fields.

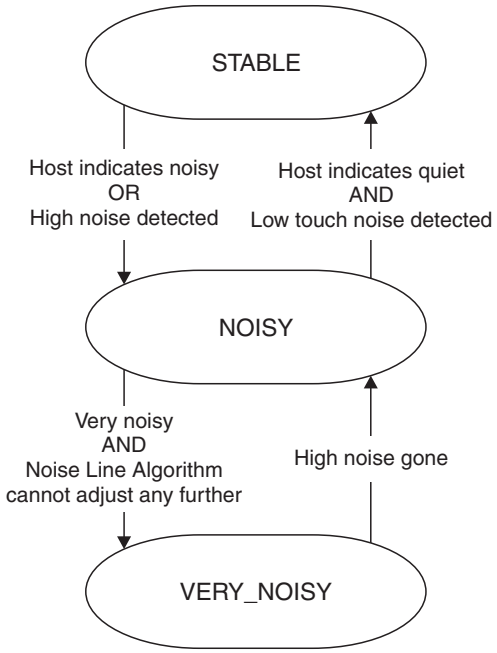
The Noise Suppression T72 object permits noise state promotions and demotions in one of two circumstances:

- If the measured noise exceeds STABHIGHNLTHR or NOISHIGHNLTHR (when in STABLE or NOISY state respectively), or falls below VNOILOWNLTHR or NOISLOWNLTHR (when in VERY_NOISY or NOISY state respectively).
- If the measured noise exceeds the maximum value of NLTHR, and all available hops have been taken (see “HOPCNT Field” and “HOPCNTPER Field”).

The DISNLTHRSCHG bit in CFG3 field (see “CFG3 Field”) determines whether one or both circumstances are used. When DISNLTHRSCHG is set to 0, both circumstances will cause a noise state promotion or demotion. When DISNLTHRSCHG is set to 1, such promotions and demotions occur in the first circumstance only. In this case, the NLTHR functionality specifically manages frequency hopping only, making its configuration much simpler.

Under NDA
© 2021 Microchip Technology Inc. and its subsidiaries

FIGURE 5-15: NOISE SUPPRESSION STATES



On entry to a new state, the Noise Suppression T72 object loads the appropriate set of control, frequency and ADCs per X parameters, and the touchscreen is recalibrated, if needed, to use the new settings.

Settings for any object on the device can also be loaded using the event-handling capabilities of the Dynamic Configuration Controller T70 object (see Section 6.10 “Dynamic Configuration Controller T70 Object”). The current state is sent as an event to the Dynamic Configuration Controller T70 object on entry to a new state. The Dynamic Configuration Controller T70 object can then respond to this event and load new parameters. For example, by responding to a transition from the NOISY to VERY_NOISY state, more aggressive settings for the Multiple Touch Touchscreen T100 object can be loaded.

5.11.1.4 Frequency Hopping

The noise suppression algorithm can change the burst frequency if the current burst frequency becomes too noisy. A set of five candidate frequencies can be defined for each state and the Noise Suppression T72 will use one of these frequencies when in operation. At suitable intervals between touch measurement cycles, the Noise Suppression T72 object perform background noise scans for each of the five frequency presets to ensure that the optimal frequency is always selected. The sets of candidate frequencies are configured using the STABFREQ, NOISFREQ and VNOIFREQ fields.

The Noise Suppression T72 object provides two frequency hopping mechanisms:

- **Internal Analysis of the Noise Lines Threshold** – For frequency hopping to occur, the measured noise must meet one of the following conditions:
  - The measured noise is greater than the maximum NLTHR (as defined by MINNLTHR + INCNLTHR; see “MINNLTHR Field” and “INCNLTHR Field”); and the sensor is in either an in-touch or no-touch state
  - The measured noise is less than the maximum NLTHR and greater than the current NLTHR / 2; the sensor is in an in-touch state for a specified hop evaluation period (see “HOPEVALTO Field”); and the allowed hops have not all been taken (see “HOPCNT Field” and “HOPCNTPER Field”).
- **Continuous frequency hopping mechanism** – Frequency hops occur whenever suitable candidate frequencies are determined by comparing the measured noise against the NOTCHMINDIFF, TCHMINDIFF, NOTCHMINHOP and TCHMINHOP fields. In this case, the decision to hop is free of any restrictions placed by the complex comparisons to NLTHR, and is made irrespective of the magnitude of measured noise at a given juncture.

The DISNLTHRHOP bit in the CFG3 field determines which mechanism is used (see “CFG3 Field”).

Suggested initial settings for frequency hopping are given in Table 5-28.

TABLE 5-28: SUGGESTED INITIAL SETTINGS – FREQUENCY HOPPING

Field	Suggested Initial Setting
CFG3	216 (OPTBGSCANEN = 1, TCHHOFFEN [1] = 1, DISNLTHRHOP = 1, DISNLTHRSCHG = 1)
BGSCAN	11 (SUPFRQHOPEN=1, SUPPCOEFF=3)

1. For details regarding TCHHOFFEN, see Section 5.11.1.7 “Background Scanning and Touch Hold-off Qualification”

The subsequent suggested initial settings for the Noise Suppression T72 object depend on the specific configuration of DISNLTHRHOP and DISNLTHRSCHG, as shown in Table 5-29.

TABLE 5-29: SUGGESTED INITIAL SETTINGS – FREQUENCY HOPPING

Field	DISNLTHRHOP = 0 (DISNLTHRSCHG = X)	DISNLTHRHOP = 1	
		DISNLTHRSCHG = 0	DISNLTHRSCHG = 1
HOPST	5	0 (default)	0 (default)
HOPCNT	5 (default)	255 [1]	255 [1]
HOPCNTPER	10 (default)	1 [1]	1 [1]
HOPEVALTO	5	5	Not used
MINNLTHR	Field needs tuning [2]	Field needs tuning [2]	Not used
INCNLTHR	MINNLTHR	MINNLTHR	Not used
FALLNLTHR	Field needs tuning	Field needs tuning	Not used
STABFREQ[]	Fields need tuning	Fields need tuning	Fields need tuning
NOISFREQ[]	Fields need tuning	Fields need tuning	Fields need tuning
VNOIFREQ[]	Fields need tuning	Fields need tuning	Fields need tuning
NOTCHMINDIFF	0 (default)	Field needs tuning	Field needs tuning
TCHMINDIFF	0 (default)	Field needs tuning	Field needs tuning
NOTCHMINHOP	0 (default)	Field needs tuning	Field needs tuning
TCHMINHOP	0 (default)	0 (default)	Field needs tuning

1. Setting prevents unnecessary noise state promotions.
2. Depends on whether Dual X is running or not. Start with Multiple Touch Touchscreen T100 TCHTHR / 2.

5.11.1.5 Touch Processing in the Presence of Noise

Touch processing will be blocked under certain circumstances, due to the potential presence of high noise:

- During frequency hopping. Touch processing is blocked for the cycle in which the frequency hop occurs. This behavior can be disabled by setting the CFG2 NOBLKTCH control to 1.
- When high noise is detected on the lines. This behavior can be disabled by setting the BLKNLTHR field to 0 (disabled).
- When a state change occurs as the result of detecting a high noise level. This behavior cannot be disabled.

The BLKTCH field in the Noise Suppression T72 messages will be set when touch processing is blocked (see “STATUS1 Field”).

5.11.1.6 Minimum Touch Threshold

The Noise Suppression T72 object can provide a Minimum Touch Threshold below which the touch objects should consider all delta data to be caused by noise. This is controlled by the MINTHRADJ field (MINTHRFACTOR and MINTHRFRAC bits) and the NLTHRMARGIN field. See the other objects in this protocol guide for details about how they respond to a minimum threshold being supplied by the Noise Suppression T72 object. In general, they will desensitize to avoid interpreting deltas caused by noise as touches.

Under NDA
© 2021 Microchip Technology Inc. and its subsidiaries

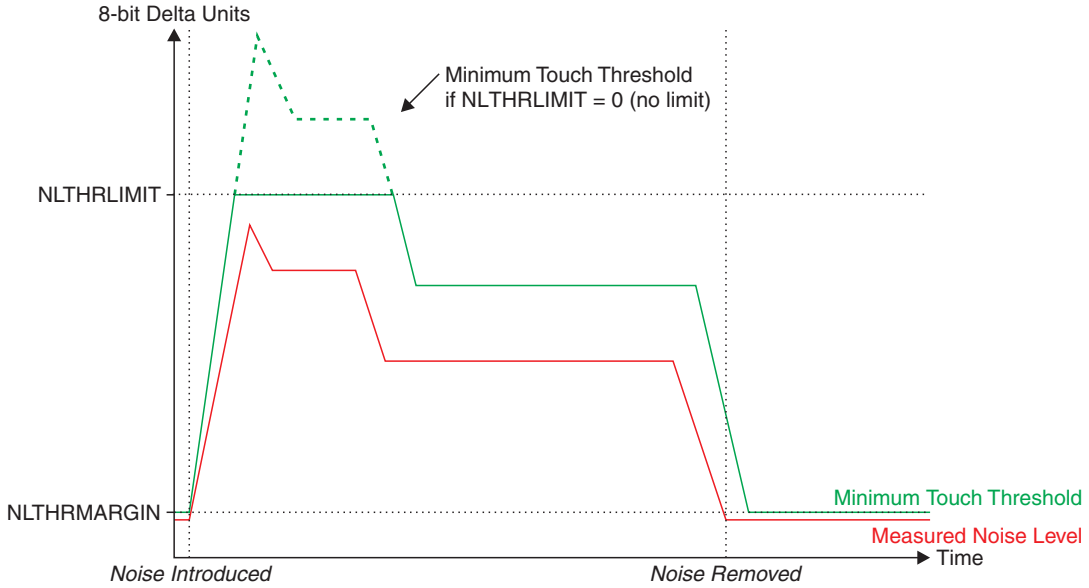
The Minimum Touch Threshold output by the Noise Suppression T72 object is a gain-adjusted value derived as follows:

$$\text{Minimum Touch Threshold} = \left(\frac{NL \times ((MINTHRFACTOR \times 16) + MINTHRFRAC)}{16}\right) + NLTHRMARGIN$$

where NL is the measured noise level.

Note that the calculated minimum touch threshold is scaled when applied to another object, based upon the ratio of NLGAIN / *other-object-gain*. This means that the Minimum Touch Threshold might not be denominated in the same units as the Touch Thresholds in other touch objects if the NLGAIN setting is different to the GAIN setting of other touch objects.

FIGURE 5-16: MINIMUM TOUCH THRESHOLD



Minimum Touch Threshold = (Measured Noise Level x ((MINTHRFACTOR x 16) + MINTHRFRAC) / 16) + NLTHRMARGIN

Suggested initial minimum touch threshold settings for the Noise Suppression T72 object are given in Table 5-30.

TABLE 5-30: SUGGESTED INITIAL SETTINGS – MINIMUM TOUCH THRESHOLD

Field	Suggested Initial Setting
NLTHRMARGIN	2
MINTHRADJ	19 (MINTHRFACTOR = 1, MINTHRFRAC = 3)
NLTHRLIMIT	Field needs tuning

5.11.1.7 Background Scanning and Touch Hold-off Qualification

For the Noise Suppression T72 object to perform effective frequency hopping, continuous noise measurements are made for each of the candidate presets. This allows for the selection of the optimal settings that can best avoid the effects of noise.

The Noise Suppression T72 object maintains two sets of IIR filter measurements (carousels), each set associated with a specific touch or no-touch state.

A pre-qualification process is applied to background noise measurements. This collates the measurements until all the measurements are deemed to have been collated within a period of consistent touch-state reporting. Once this is achieved the carousel filters updated and any subsequent frequency hopping determination is made. If a touch state transition occurs during a measurement cycle, the noise measurements made during the transition are rejected and a new carousel collation begins. This qualification process is referred to as Touch Hold-off.

The Touch Hold-off mechanism can improve the frequency hopping candidate selection process, as the filter carousel content is more accurate when repeated touch transitions take place. Without this process, mis-reports from false touches and/or momentary loss of touch determination can reduce the effectiveness of any frequency hopping.

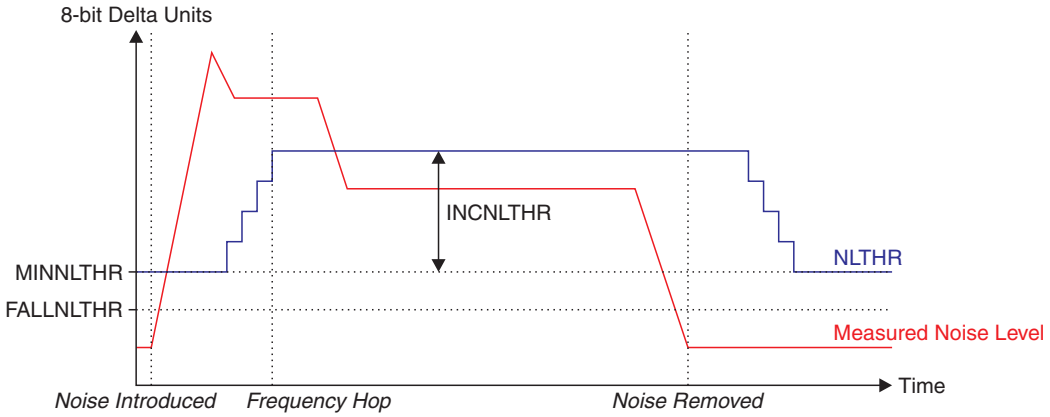
Touch Hold-off is selectable using the TCHHOFFEN bit in the CFG3 field (see "CFG3 Field").

5.11.1.8 Noise Lines Threshold

The Noise Suppression T72 object maintains an internal Noise Lines Threshold (NLTHR) against which the measured noise can be compared. This allows an appropriate action to be taken, such as transitioning to a different noise state or choosing a new burst frequency (see Section 5.11.1.4 "Frequency Hopping").

The INCNLTHR, FALLNLTHR and MINNLTHR fields are used to determine how the NLTHR is increased and decreased (see Figure 5-17).

FIGURE 5-17: NOISE LINES THRESHOLD



Suggested initial settings for the Noise Suppression T72 object are given in Table 5-31.

TABLE 5-31: SUGGESTED INITIAL SETTINGS – NOISE LINES THRESHOLD

Field	Suggested Initial Setting
MINNLTHR	Field needs tuning; Depends on whether Dual X is running or not. Start with Multiple Touch Touchscreen T100 TCHTHR / 2
INCNLTHR	MINNLTHR
FALLNLTHR	Field needs tuning

5.11.1.9 Adjusting the Number of ADCs per X

Each of the five candidate frequencies has a corresponding explicit no touch and touch ADCs Per X settings, which are applied whenever frequency hopping occurs. This facilitates the following capabilities:

- In normal conditions, the combination of frequency and ADCs Per X can be controlled to ensure message report rate requirements can be met.
- In very high noise conditions, a larger number of ADCs Per X could be used to improve noise performance at the expense of message rate and power consumption.
- In low noise conditions, or with no touch detected, a smaller number of ADCs Per X could be used to help reduce power consumption.

Under NDA
© 2021 Microchip Technology Inc. and its subsidiaries

5.11.1.10 Dual X Drive

The Noise Suppression T72 object provides an alternative burst mode on the X lines known as Dual X Drive.

With Dual X Drive, X pulses occur on pairs of X lines instead of single X lines. When Dual x is enabled, the noise suppression algorithm can use Dual X Drive to generate acquisition pulses on pairs of X lines instead of single X lines (X0 and X1, then X1 and X2, and so on). Note that Dual X Drive occurs only on those X lines that are covered by an enabled Multiple Touch Touchscreen T100 object.

Dual X mode is useful when finger touches will cover more than one X line on a closely spaced X sensor matrix as it improves the signal-to-noise ratio (SNR). Note, however, that it will affect accuracy and linearity on the X edge. Furthermore, the enlarged touch means that the two-touch separation is increased and the classification of stylus touches is made more difficult.

Dual X operation is selected according to the DUALXMODE bit in the STABCTRL, NOISCTRL or VNOICTRL field (as appropriate).

5.11.1.11 Resetting the Noise Suppression Algorithm

The noise suppression algorithm is reset whenever any of the following occurs:

- The ENABLE bit in the CTRL field is changed (note that it is not recommended to do this dynamically)
- The CALCFG1 field is changed

5.11.1.12 Event Triggers

The Noise Suppression T72 object does not directly request a change in the acquisition parameters to the measurement layer (such as accessed frequency, number of ADCs per X, detection thresholds, and so on). Instead, the measurement parameters can be altered indirectly by the means of the events mechanism provided by the Dynamic Configuration Controller T70 object. In this case, as soon as a new noise state is entered, the Noise Suppression T72 object will request the triggering of an event to the Dynamic Configuration Controller T70 object.

5.11.1.13 General Initial Settings

Suggested initial settings for fields that have not been listed in earlier sections are given in Table 5-32 on page 139.

TABLE 5-32: SUGGESTED INITIAL SETTINGS

Field	Suggested Initial Setting
CTRL	ENABLE = 1 Other bits as required by user
CALCFG1	0
CFG1	0 when a potential noise source is not present (for example, a charger is turned off) 1 (NOISEON = 1) when a potential noise source is notified by the host (for example, a charger is turned on)
CFG2	0
BLKNLTHR	0
STABCTRL	0
STABTCHAPX[]	Fields need tuning
STABNOTCHAPX[]	Fields need tuning
STABHIGHNLTHR	Field needs tuning
NOISCTRL	9 (AUTO = 1, DUALXMODE = 1)
NOISCHAPX[]	Fields need tuning
NOISNOTCHAPX[]	Fields need tuning
NOISLOWNLTHR	Field needs tuning
NOISHIGHNLTHR	Field needs tuning
NOISCNT	3 (NOTCH = 3)
VNIOCTRL	9 (AUTO = 1, DUALXMODE = 1)

TABLE 5-32: SUGGESTED INITIAL SETTINGS (CONTINUED)

Field	Suggested Initial Setting
VNOIFREQ[]	Fields need tuning
VNOITCHAPX[]	Fields need tuning
VNOINOTCHAPX[]	Fields need tuning
VNOILOWNLTHR	Field needs tuning
VNOICNT	3 (NOTCH = 3)

5.11.2 CONFIGURATION

TABLE 5-33: CONFIGURATION FOR NOISE SUPPRESSION T72 (PROCG_NOISESUPPRESSION_T72)

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	CTRL	Reserved	RPTDUALX	RPTNLTHR	RPTNOISELVL	RPTSTATE	RPTACQ	RPTEN	ENABLE
1	CALCFG1	Reserved						VNOISEON	NOISEON
2	CFG1	Reserved						VNOISEON	NOISEON
3	CFG2	NOTCHNOISEON	Reserved						NOBLKTCH
4	Reserved	Reserved							
5	HOPCNT	Hop Count							
6	HOPCNTPER	Hop Count Period							
7	HOPEVALTO	Hop Evaluation Timeout							
8	HOPST	Hop Suspend Time							
9	NLGAINDUALX	Dual X Gain							
10	MINNLTHR	Minimum Noise Lines Threshold							
11	INCNLTHR	Increase Noise Lines Threshold							
12	FALLNLTHR	Falling Noise Level Threshold							
13	NLTHRMARGIN	Minimum Noise Lines Threshold Margin							
14	MINTHRADJ	MINTHRFACTOR				MINTHRFRAC			
15	NLTHRLIMIT	Minimum Touch Threshold limit							
16	BGSCAN	RATIO				SUPPFRQHOPEN	SUPPCOEFF		
17	NLGAINSINGX	GAIN							
18	BLKNLTHR	Noise line limit for blocking touch processing							
19	CFG3	OPTBGSCANEN	TCHHOFFEN	Reserved	DISNLTHRSCHG	DISNLTHRHOP	NUMPRESETS		
20	STABCTRL	CAL	Reserved			DUALXMODE	Reserved		NOTCHTHR
21	STABFREQ[]	STABLE State Frequency Setting 0							
22		STABLE State Frequency Setting 1							
23		STABLE State Frequency Setting 2							
24		STABLE State Frequency Setting 3							
25		STABLE State Frequency Setting 4							
26	STABTCHAPX[]	STABLE State Touch ADCs per X Setting 0							
27		STABLE State Touch ADCs per X Setting 1							
28		STABLE State Touch ADCs per X Setting 2							
29		STABLE State Touch ADCs per X Setting 3							
30		STABLE State Touch ADCs per X Setting 4							
31	STABNOTCHAPX[]	STABLE State No Touch ADCs per X Setting 0							
32		STABLE State No Touch ADCs per X Setting 1							
33		STABLE State No Touch ADCs per X Setting 2							
34		STABLE State No Touch ADCs per X Setting 3							
35		STABLE State No Touch ADCs per X Setting 4							
36	Reserved	Reserved							

Under NDA

© 2021 Microchip Technology Inc. and its subsidiaries

TABLE 5-33: CONFIGURATION FOR NOISE SUPPRESSION T72 (PROCG_NOISESUPPRESSION_T72) (CONTINUED)

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
37	Reserved	Reserved							
38	STABHIGHNLTHR	STABLE State High noise detection level							
39	Reserved	Reserved							
40	NOISCTRL	CAL	Reserved			DUALXMODE	Reserved		AUTO
41	NOISFREQ[]	NOISY State Frequency Setting 0							
42		NOISY State Frequency Setting 1							
43		NOISY State Frequency Setting 2							
44		NOISY State Frequency Setting 3							
45		NOISY State Frequency Setting 4							
46	NOISTCHAPX[]	NOISY State Touch ADCs per X Setting 0							
47		NOISY State Touch ADCs per X Setting 1							
48		NOISY State Touch ADCs per X Setting 2							
49		NOISY State Touch ADCs per X Setting 3							
50		NOISY State Touch ADCs per X Setting 4							
51	NOISNOTCHAPX[]	NOISY State No Touch ADCs per X Setting 0							
52		NOISY State No Touch ADCs per X Setting 1							
53		NOISY State No Touch ADCs per X Setting 2							
54		NOISY State No Touch ADCs per X Setting 3							
55		NOISY State No Touch ADCs per X Setting 4							
56	Reserved	Reserved							
57	NOISLOWNLTHR	NOISY State Low noise detection level							
58	NOISHIGHNLTHR	NOISY State High noise detection level							
59	NOISCNT	LOWNOISTCH				NOTCH			
60	VNOICTRL	CAL	Reserved			DUALXMODE	Reserved		AUTO
61	VNOIFREQ[]	VERY_NOISY State Frequency Setting 0							
62		VERY_NOISY State Frequency Setting 1							
63		VERY_NOISY State Frequency Setting 2							
64		VERY_NOISY State Frequency Setting 3							
65		VERY_NOISY State Frequency Setting 4							
66	VNOITCHAPX[]	VERY_NOISY State Touch ADCs per X Setting 0							
67		VERY_NOISY State Touch ADCs per X Setting 1							
68		VERY_NOISY State Touch ADCs per X Setting 2							
69		VERY_NOISY State Touch ADCs per X Setting 3							
70		VERY_NOISY State Touch ADCs per X Setting 4							
71	VNOINOTCHAPX[]	VERY_NOISY State No Touch ADCs per X Setting 0							
72		VERY_NOISY State No Touch ADCs per X Setting 1							
73		VERY_NOISY State No Touch ADCs per X Setting 2							
74		VERY_NOISY State No Touch ADCs per X Setting 3							
75		VERY_NOISY State No Touch ADCs per X Setting 4							
76	Reserved	Reserved							
77	VNOILOWNLTHR	VERY_NOISY State Low noise detection level							
78	Reserved	Reserved							
79	VNOICNT	LOWNOISTCH				NOTCH			
80	Reserved	Reserved							
81	NOTCHMINDIFF	Minimum measured filter range for frequency hopping – no touch							
82	TCHMINDIFF	Minimum measured filter range for frequency hopping – in touch							
83	NOTCHMINHOP	Minimum quieter hop filter range for frequency hopping – no touch							
84	TCHMINHOP	Minimum quieter hop filter range for frequency hopping – in touch							

TABLE 5-33: CONFIGURATION FOR NOISE SUPPRESSION T72 (PROCG_NOISESUPPRESSION_T72) (CONTINUED)

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
85	BGNLRATIOSTAB	TCH				NOTCH			
86	BGNLRATIONOIS	TCH				NOTCH			
87	BGNLRATIOVNOI	TCH				NOTCH			
88	NLRATIO	NOTSTAB				STAB			

CTRL Field

This is the main control field.

ENABLE: Enables the Noise Suppression T72 object. The object is enabled if set to 1, and disabled if set to 0.

RPTEN: Allows this object to send messages to the host through the Message Processor T5 object. Reporting is enabled if set to 1, and disabled if set to 0.

RPTACQ: Allows changes to the frequency index to trigger the reporting of messages to the host. Reporting is enabled if set to 1, and disabled if set to 0.

RPTSTATE: Allows changes in noise state to trigger reporting messages to the host. Reporting is enabled if set to 1, and disabled if set to 0.

RPTNOISELVL: Allows noise level changes to trigger reporting messages to the host. Reporting is enabled if set to 1, and disabled if set to 0.

RPTNLTHR: Allows noise lines threshold changes to trigger reporting messages to the host. Reporting is enabled if set to 1, and disabled if set to 0.

RPTDUALX: Allows changes between Dual X and non Dual X states to trigger reporting messages to the host. Reporting is enabled if set to 1, and disabled if set to 0.

CALCFG1 Field

This field repeats bits from CFG1 field so that changing these bits causes a touchscreen recalibration.

NOTE	Use of this field is not recommended.

NOISEON: See “CFG1 Field”.

VNOISEON: See “CFG1 Field”.

CFG1 Field

This field repeats bits from CALCFG1 field so that these bits can be changed without causing a touchscreen recalibration.

NOISEON: This bit can be set to 1 by the host to indicate that a noise-source is activated for the product. The object state-machine will change to the NOISY state if it was previously in the STABLE state.

VNOISEON: Manually forces a change to the VERY_NOISY state for use during setup or development. If this bit is set to 1, the object state machine changes to the VERY_NOISY state if it was previously in the NOISY or STABLE state.

CFG2 Field

NOBLKTCH: Disables the blocking of touch processing. By default (NOBLKTCH = 0), touch processing is blocked for the cycle in which a frequency hop occurs. When this bit is set to 1, the block is disabled and touch processing occurs. See also Section 5.11.1.5 “Touch Processing in the Presence of Noise”.

NOTCHNOISEON: Enables demotion of noise-level from either NOISY or VERY_NOISY states when clamped to one of these noise-levels by either the NOISEON or VNOISEON bits (CALCFG1 or CFG1 field) respectively. Consequently, this field alters the default behavior when noise-level promotion is enforced by configuration settings established by the host device.

Such a demotion can only occur during continuous touch, whereby noise-levels are deemed acceptably low. Essentially this bit allows the device to assume that noise is present until a touch is introduced for a sufficient time to determine its demotion to a quieter noise state.

Under NDA

© 2021 Microchip Technology Inc. and its subsidiaries

If such a demotion occurs, then as soon as touch is removed, the noise-level will revert to the original state, as mandated by the configuration bits specified above.

HOPCNT Field

This field specifies the Hop Count. This limits the number of allowed frequency jumps within the Hop Count Period before the object takes counter-measures to improve performance (for example, changing the noise state if all other required conditions are met). The default value of 0 means 5 hops.

Range: 0 (5), 1 to 255

HOPCNTPER Field

This field specifies the Hop Count Period. This specifies the time period during which the number of frequency jumps is compared to the HOPCNT limit. The Hop Count Period is specified in units of 100 acquisition cycles, where the default value of 0 means 10 (1000 acquisition cycles).

Range: 0 (10), 1 to 255

HOPEVALTO Field

This field specifies the Hop Evaluation Timeout. This setting provides the opportunity to hop to a quieter frequency when the currently selected frequency is consistently noisy (but does not suffer from high noise to initiate a frequency hop). The Hop Evaluation Timeout is specified in units of 10 acquisition cycles, where the default value of 0 means 10 (100 acquisition cycles).

Range: 0 (10), 1 to 255

HOPST Field

This field specifies the Hop Suspend Time. This is the time period from when the last frequency hop occurred to when a new frequency hop is allowed. The Hop Suspend Time is specified in units of 10 acquisition cycles, where the default value of 0 indicates that no suspending restriction is placed between hops. This indicates that such hops can take place whenever the IIR filter carousel analysis deems that a more suitable candidate is available.

Range: 0 (no suspending restriction), 1 to 255

NLGAINDUALX Field

This field specifies the Noise Lines Gain for Dual X. Note that this is the gain for use when making noise line measurements; the gain for touch measurements is set by the Multiple Touch Touchscreen T100 object.

GAIN: The noise line gain to be used.

GAIN Range: 0 to 22

MINNLTHR Field

This field specifies the Minimum Noise Lines Threshold (in 8-bit deltas). The noise suppression algorithm sometimes compares the measured noise level to this field, or to a proportion of this field, in its decision making process (see Figure 5-17 on page 138). The default value of 0 means 32.

Range: 0 (32), 1 to 255

INCNLTHR Field

NOTE	The host should not change this field dynamically at runtime without resetting the noise suppression algorithm.

This field specifies the Increment for the Noise Lines Threshold (in 8-bit deltas). This is how high the noise suppression algorithm can increase the reported noise lines threshold (see Figure 5-17 on page 138).

Range: 0 to 255

FALLNLTHR Field

This field specifies the Falling Noise Lines Threshold (in 8-bit deltas). When the measured noise level becomes less than or equal to FALLNLTHR, the noise suppression algorithm is allowed to decrease the noise lines threshold (see Figure 5-17 on page 138).

Range: 0 to 255

NLTHRMARGIN Field

NOTE	The host should not change this field dynamically at runtime without resetting the noise suppression algorithm.

This field specifies the Noise Lines Threshold Margin (in 8-bit deltas). This is the amount by which the Minimum Touch Threshold can be increased, compared to the reported noise lines threshold, when the noise suppression algorithm is enforcing a Minimum Touch Threshold (see Figure 5-16 on page 137). Note that the gain setting specified by the NLGAIN field is applied. This means that the Minimum Touch Threshold might not be denominated in the same units as the Touch Thresholds in other touch objects if the NLGAIN setting is different to the GAIN setting of other touch objects.

Range: 0 to 255

MINTHRADJ Field

This field specifies the Minimum Threshold Adjustment. This is used to determine the Minimum Touch Threshold (see Figure 5-16 on page 137). The Minimum Touch Threshold is calculated from the measured noise level using the following formula:

$$\text{Minimum Touch Threshold} = \left(\frac{NL \times ((MINTHRFACTOR \times 16) + MINTHRFRAC)}{16}\right) + NLTHRMARGIN$$

where NL is the measured noise level

MINTHRFRAC: configures the Minimum Touch Threshold Fractional component for use in the minimum touch threshold calculation. This effectively is 1/16 of the scaling applied to measured noise to calculate the Minimum Touch Threshold.

MINTHRFRAC Range: 0 to 15

MINTHRFACTOR: Configures the Minimum Touch Threshold Factor component for use in the minimum touch threshold calculation.

MINTHRFACTOR Range: 0 to 15

NLTHRLIMIT Field

This field specifies the Noise Lines Threshold Limit. This is the maximum allowed value for the calculated Minimum Touch Threshold (see "MINTHRADJ Field"). A zero value means that the calculated Minimum Touch Threshold value is not constrained. The effect of this field is shown in Figure 5-16 on page 137.

Range: 0 (unconstrained), 1 to 255

BGSCAN Field

SUPPCOEFF: Specifies the supplementary filter coefficient. This value is used to scale the supplementary background-scan noise-line delta filter. Larger values for this coefficient result in a more aggressive response to noise measurements. This value is ignored when BGSCAN SUPPFRQHOPEN is set to 0. The range is 1 to 7, where a value of 1 yields the fastest response and 7 yields the slowest. The default value of 0 means 5.

SUPPCOEFF Range: 0 (5), 1 to 7

SUPPFRQHOPEN: Enables supplementary filter frequency hopping. When this bit is set to 1, the supplementary background-scan noise-line delta filter will be utilized. This is used to improve decision performance when frequency hopping.

RATIO: Specifies the in-touch background scan rate. This is the rate at which background scan measurements are inserted into the acquisition cycle during a continuous touch. The range for this control is 0 to 15, where the default value of 0 means 10.

Under NDA

© 2021 Microchip Technology Inc. and its subsidiaries

NOTE	This field may be overridden by the TCH control in the BGNLRATIOSTAB, BGNLRATIONOIS or BGNLRATIOVNOI fields (see "BGNLRATIOSTAB, BGNLRATIONOIS and BGNLRATIOVNOI Fields").

RATIO Range: 0 (10), 1 to 15

NLGAINSINGX Field

This field specifies the Noise Lines Gain for when Dual X is not used. Note that this is the gain for use when making noise line measurements; the gain for touch measurements is set by the Multiple Touch Touchscreen T100 object.

GAIN: The noise line gain to be used.

Range: 0 to 22

BLKNLTHR Field

This field specifies the Block Noise Lines Threshold (in 8-bit deltas). Touch processing is blocked whenever the measured unfiltered noise level exceeds the limit defined by this value. The default value of 0 means this feature is disabled.

Range: 0 (disabled), 1 to 255

CFG3 Field

NUMPRESETS: Specifies the Number of Frequency Presets to use for background scans. Up to five preset frequencies are usable for background scans. Note that if NUMPRESETS is set to 1, no background scanning is performed and frequency hopping is disabled.

Range: 0 (5), 1 (no background scans), 2 to 5

DISNLTHRHOP: Disables internal analysis of the Noise Lines Threshold as a means to determine frequency hopping. If this bit is set to 0 (default), frequency hopping is determined by all the conditions listed in Section 5.11.1.4 "Frequency Hopping". If this bit is set to 1, Continuous Frequency Hopping is enabled. In this case, frequency hops can occur only as a consequence of comparing the measured noise against the NOTCHMINDIFF, TCHMINDIFF, NOTCHMINHOP and TCHMINHOP fields.

DISNLTHRSCHG: Disables the internally-managed NLTHR functionality from being used to invoke noise state promotions (see Section 5.11.1.3 "Transitions Between Noise States").

When this bit is set to 1, transitions between states occur only according to the first circumstance listed in Section 5.11.1.3 "Transitions Between Noise States" (that is, when the thresholds are passed). In this case (and dependent on DISNLTHRHOP), the NLTHR functionality may only be used to manage frequency hopping, making configuration much simpler.

TCHHOFFEN: Enables Touch Hold-off (see Section 5.11.1.7 "Background Scanning and Touch Hold-off Qualification"). If this bit is set 1, the IIR filters are held off from being updated until touch status (in-touch or no-touch) is consistent over time. This option improves filter quality during touch and no-touch transitions, but does so at the potential expense of frequency-hopping responsiveness.

OPTBGSCANEN: Indicates that first-touch response is to be optimized for noise detection.

If this bit is set to 1, the background scan rate adopted during the active to idle timeout period will be the in-touch background scan rate (as configured in the BGSCAN field RATIO control). This ensures the optimal acquisition duration.

If this bit is set to 0, the background-scan rate adopted during the active to idle timeout period will be that which is used during no-touch detection. This ensures the best noise determination.

STABCTRL, NOISCTRL and VNOICTRL Fields

These fields are used to configure specific operation settings for the appropriate noise states.

NOTCHTHR (STABCTRL only): Indicates how the Minimum Touch Threshold is to be determined in the STABLE state. If this bit is set to 1, the Minimum Touch Threshold is set to zero. If this bit is set to 0, the Minimum Touch Threshold is calculated.

AUTO (NOISCTRL and VNOICTRL only): Controls the transitions into and out of the NOISY/VERY_NOISY state. Specifically, this bit selects whether transitions between the states can be made based on detected noise levels. If this bit is set to 1, the transitions are influenced by the detected noise levels. If this bit is set to 0, the detected noise levels are ignored.

When this bit is set to 1, the transition up from the STABLE to NOISY state, or the NOISY to VERY_NOISY state, is made using one of the following conditions:

- The number of frequency hops requested > HOPCNT within HOPCNTPER period, and the measured noise exceeds the maximum value of NLTHR (see Section 5.11.1.3 “Transitions Between Noise States”)

Or

- Detected noise ≥ STABHIGHNLTHR / NOISHIGHNLTHR

The transition down from the VERY_NOISY to the NOISY state occurs if:

- Detected noise ≤ VNOILOWNLTHR for a given number of cycles.
  The number of cycles is determined by LOWNOISTCH, if a touch has been detected, and by NOTCH, if no touch has been detected.

Similarly, the transition down from the NOISY to the STABLE state occurs if:

- Detected noise ≤ NOISLOWNLTHR for a given number of cycles.
  The number of cycles is determined by LOWNOISTCH, if a touch has been detected, and by NOTCH, if no touch has been detected.

Note: LOWNOISTCH and NOTCH in the above are in the NOISCNT and VNOICNT fields.

DUALXMODE: Used to specify the conditions for Dual X operation in the appropriate noise states. If this bit is set to 1, Dual X operation is enabled. If this bit is set to 0, Dual X operation is disabled.

CAL: If this bit is set to 1, it forces a calibration on entry to the state. Note, however, that if this bit is set to 0, it may be overridden and a calibration still performed under certain circumstances (regardless of the setting of this bit).

NOTE	Use of this control is not recommended.

STABFREQ[], NOISFREQ[] and VNOIFREQ[] Fields

These fields specify the set of five candidate frequencies that the noise suppression algorithm can select when operating in a particular state.

It is recommended that the candidate frequencies are calculated using a suitable tool, such as the Basic Noise Suppression tool in maXTouch Studio.

Range: 0 to 255

STABTCHAPX[], NOISTCHAPX[] and VNOITCHAPX[] Fields

These fields specify the set of five candidate ADCs per X settings that the noise suppression algorithm can select when operating in the appropriate noise states when touch has been detected (see Section 5.11.1 “Introduction”).

These candidate ADCs per X settings may be used instead of the configured Active Syncs per X setting, as determined by the CTE Configuration T46 (see Section 6.8.1 “Introduction” for more information).

Range: 0 (8), 1 to 255

STABNOTCHAPX[], NOISNOTCHAPX[] and VNOINOTCHAPX[] Fields

These fields specify the set of five candidate ADCs per X settings that the noise suppression algorithm can select when operating in the appropriate noise states when no touch has been detected (see Section 5.11.1 “Introduction”).

These candidate ADCs per X settings may be used instead of the configured Idle Syncs per X setting, as determined by the CTE Configuration T46 (see Section 6.8.1 “Introduction” for more information).

Range: 0 (8), 1 to 255

Under NDA
© 2021 Microchip Technology Inc. and its subsidiaries

NOISLOWNLTHR and VNOILOWNLTHR Fields

These fields define the noise threshold (in 8-bit deltas) below which the noise is identified as too low for the current state. When the noise level falls below VNOILOWNLTHR, the noise suppression algorithm enters NOISY state and when the noise level falls below NOISLOWNLTHR, the noise suppression algorithm enters STABLE state.

The range for these fields is 0 to 255, where a value of 0 sets the threshold to the falling noise lines threshold.

Range: 0 (falling noise lines threshold), 1 to 255

STABHIGHNLTHR and NOISHIGHNLTHR Fields

These fields define the noise threshold (in 8-bit deltas) above which the noise is identified as too high for the current state. When the noise level rises above STABHIGHNLTHR, the noise suppression algorithm enters the NOISY state and when the noise level rises above NOISHIGHNLTHR, the noise suppression algorithm enters the VERY_NOISY state.

The range for these fields is 0 to 255, where a value of 0 sets the threshold to 75% of the noise lines threshold.

Range: 0 (75% of noise line threshold), 1 to 255

NOISCNT and VNOICNT Fields

These fields provide real-time counters that are used in the NOISY and VERY_NOISY states to allow demotion to the NOISY or STABLE state, as appropriate.

NOTCH: Specifies the number of “no touch” cycles used to identify a low noise condition. This value is specified in units of 100 cycles, where a value of 0 means 1.

NOTCH Range: 0 (1), 1 to 15

LOWNOISTCH: Specifies the number of “low noise touched” cycles used to identify a low noise condition. This field is specified in units of 10 cycles.

LOWNOISTCH Range: 0 (do not wait for “low noise touched”),
1 to 15

NOTCHMINDIFF Field

This field specifies the No Touch Minimum Differential (in 8-bit deltas). This is the absolute extent between the noisiest and quietest value for the IIR filters, when no touch is being detected, below which the algorithm deems that the data does not yield a clear candidate for frequency hopping selection.

If the range is less than this value, no frequency hopping will occur.

When the range is greater than or equal to the value, the frequency selection algorithm will make use of filters to determine a frequency hopping candidate.

A value of 0 means that effectively no range qualification takes place.

Effectively, the higher the value specified, the more distinct the noise measured for the preset candidates has to be before frequency hopping can occur.

In practical terms, this field is used to reduce excessive hopping between two similarly quiet candidates. This is potentially problematic if the object is also using HOPST and HOPCNT to manage hopping, whereby excessive hops that yield little or no gain would be counter-productive.

Range: 0 (no differential restriction), 1 to 255

TCHMINDIFF Field

This field specifies the Touch Minimum Differential (in 8-bit deltas). This is the absolute extent between the noisiest and quietest value for the IIR filters, when a touch is being detected, below which the algorithm deems that the data does not yield a clear candidate for frequency hopping selection.

If this range is less than the specified value, no frequency hopping will occur.

When the range is greater than or equal to the specified value, the frequency selection algorithm will make use of filters to determine a frequency hopping candidate.

A value of 0 means that effectively no range qualification takes place.

Effectively, the higher the value specified, the more distinct the noise measured for the preset candidates has to be before frequency hopping can occur.

In practical terms, this field is used to ensure that no hopping will occur until the IIR filters indicate that there is enough of a range for a suitable candidate to be selected. This prevents superfluous frequency hopping from taking place in low noise conditions.

If no differential restriction is applied, then frequency hopping will occur even if measured noise is very similar for all the preset candidates. This may not be beneficial if the nature of the noise is highly transient.

Range: 0 (no differential restriction), 1 to 255

NOTCHMINHOP Field

This field specifies the No Touch Minimum Hop (in 8-bit deltas). This is the reduction required between the measured noise associated with the currently selected frequency and the measured noise of the proposed (quieter) frequency hopping candidate, when no touch is being detected, that is permitted for the hop to occur.

If this filter differential is less than the specified value, no frequency hopping will occur.

When the filter differential is greater than or equal to the specified value, the selected quieter candidate is deemed to yield a more suitable configuration, and so a frequency hop will occur in accordance with other hopping configuration limitations.

A value of 0 means that effectively no range qualification takes place.

In practical terms, this is used to reduce excessive frequency hopping between two similarly quiet candidates. This is potentially problematic if the object is also using HOPST and HOPCNT to manage hopping (see "HOPST Field" and "HOPCNT Field"), whereby excessive hops that yield little or no gain would be counter-productive.

Range: 0 (no differential restriction), 1 to 255

TCHMINHOP Field

This field specifies the Touch Minimum Hop (in 8-bit deltas). This is the minimum reduction required between the measured noise associates with the currently utilized preset configuration, and the quieter measured noise of the proposed frequency-hop candidate, when a touch is being detected, that is permitted for the hop to occur.

If this filter differential is less than the specified value, no frequency hopping will occur.

When the filter differential is greater than or equal to the specified value, the selected quieter candidate is deemed to yield a more suitable configuration, and so a frequency hop will occur.

In practical terms, this field is used to reduce excessive frequency hopping between two similarly quiet candidates. This is potentially problematic if the object is also using HOPST and HOPCNT to manage hopping (see "HOPST Field" and "HOPCNT Field"), whereby excessive hops that yield little or no gain would be counter-productive.

Range: 0 (no differential restriction), 1 to 255

BGNLRATIOSTAB, BGNLRATIONOIS and BGNLRATIOVNOI Fields

These fields specify the Background Noise Line Ratio configuration in the STABLE, NOISY and VERY_NOISY states.

NOTCH: Configures the background noise line ratio in the untouched state. The ratio is specified in units of 4 cycles, where the default value of 0 means perform a scan every cycle.

NOTCH Range: 0 (every cycle), 1 to 15

TCH: Configures the background noise line ratio in the touched state. A non-zero setting overrides the BGSCAN RATIO setting (see "BGSCAN Field"). The ratio is specified in units of 2 cycles, where the default value of 0 means use the BGSCAN RATIO setting.

TCH Range: 0 (use BGSCAN RATIO), 1 to 15

NLRATIO Field

This field specifies the Noise Line Ratio (that is, the noise line scan rates) for the STABLE state and non STABLE states (that is, NOISY and VERY_NOISY states). This allows the noise line measurements to be performed less frequently on an untouched system when mutual measurements are not being performed. Note that noise lines will run on every cycle for which a mutual measurement is performed.

Under NDA
© 2021 Microchip Technology Inc. and its subsidiaries

STAB: Configures the noise line ratio in the STABLE state. The ratio is specified in units of 4 cycles, where the default value of 0 means perform a scan every cycle.

STAB Range: 0 (every cycle), 1 to 15

NOTSTAB: Configures the noise line ratio in all other states (that is, NOISY and VERY_NOISY states). The ratio is specified in units of 2 cycles, where the default value of 0 means perform a scan every cycle.

NOTSTAB Range: 0 (every cycle), 1 to 15

5.11.3 CONFIGURATION CHECKS

The Noise Suppression T72 object causes a configuration check to be performed in the following circumstances:

- When the object is enabled (that is, the ENABLE bit is set in the CTRL field)
- When certain fields, including the CTRL field, are changed

In addition, some fields will cause an automatic recalibration to be performed.

A configuration check may determine that a configuration error has occurred (for example, if a setting is set outside of its allowed range or a conflict has occurred between two settings). This is signaled to the host (see Section 3.3.3 “Messages”), and the device halts until the error has been corrected. To fix the error, the object settings should be checked to verify that they are all within their allowed limits, as stated in the field descriptions.

If a configuration check occurs, some of the Noise Suppression T72 object’s processing is reset. Note that this happens whenever any object causes a configuration check, not just the Noise Suppression T72 object.

TABLE 5-34: CONFIGURATION CHECKS FOR NOISE SUPPRESSION T72 (PROCG_NOISESUPPRESSION_T72)

Field	Changing The Field Causes...		Effect of Configuration Checks On Field
	Configuration Check	Automatic Recalibration	
CTRL	Yes(1)	Yes(1)	None
CALCFG1	Yes	Yes	None
CFG1	Yes	No	None
CFG2	Yes	No	None
HOPCNT	Yes	No	None
HOPCNTPER	Yes	No	None
HOPEVALTO	Yes	No	None
HOPST	Yes	No	None
MINNLTHR	Yes	No	None
INCNLTHR	No	No	None
FALLNLTHR	Yes	No	None
NLTHRMARGIN	No	No	None
MINTHRADJ	No	No	None
NLTHRLIMIT	No	No	None
BGSCAN	Yes	No	None
NLGAINSINGX	Yes	Yes	None
BLKNLTHR	No	No	None
CFG3	Yes	No	None
STABCTRL	Yes	No	None
STABFREQ[]	No	No	None
STABTCHAPX[]	No	No	None
STABNOTCHAPX[]	No	No	None
STABHIGHNLTHR	No	No	None
NOISCTRL	Yes	No	None

TABLE 5-34: CONFIGURATION CHECKS FOR NOISE SUPPRESSION T72 (PROCG_NOISESUPPRESSION_T72) (CONTINUED)

Field	Changing The Field Causes...		Effect of Configuration Checks On Field
	Configuration Check	Automatic Recalibration	
NOISFREQ[]	No	No	None
NOISTCHAPX[]	No	No	None
NOISNOTCHAPX[]	No	No	None
NOISLOWNLTHR	No	No	None
NOISHIGHNLTHR	No	No	None
NOISCNT	No	No	None
VNOICTRL	Yes	No	None
VNOIFREQ[]	No	No	None
VNOITCHAPX[]	No	No	None
VNOINOTCHAPX[]	No	No	None
NOTCHMINDIFF	No	No	None
TCHMINDIFF	No	No	None
NOTCHMINHOP	No	No	None
TCHMINHOP	No	No	None
BGNLRATIOSTAB	No	No	None
BGNLRATIONOIS	No	No	None
BGNLRATIOVNOI	No	No	None
NLRATIO	No	No	None

Note 1: If the ENABLE bit is toggled on or off.

5.11.4 MESSAGES

The message data for the Noise Suppression T72 object is shown in Table 5-35.

TABLE 5-35: MESSAGE DATA FOR NOISE SUPPRESSION T72 (PROCG_NOISESUPPRESSION_T72)

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	STATUS1	Reserved	DUALXCHG	NLTHRCHG	NOISELVLCHG	STATECHG	ACQCHG	TCH	BLKTCH
2	STATUS2	ACQINDEX			Reserved	DUALX	STATE		
3	MINTCHTHR	Current Minimum Touch Threshold							
4	PKNOISELVL	Peak measured noise level							
5	NOISELVL	Current measured noise level							
6	NLTHR	Current noise lines threshold							

STATUS1 Field

This field reports the current status of the noise suppression algorithm.

BLKTCH: Indicates noise suppression algorithm has generated a request to block touch processing if the algorithm is unable to suppress the effects of noise. See also Section 5.11.1.5 “Touch Processing in the Presence of Noise”.

Note that a Noise Suppression T72 message is not reported for every changing value of the BLKTCH bit. It is possible for the blocking of touch processing to be toggled multiple times between consecutive Noise Suppression T72 messages. This could cause Multiple Touch Touchscreen T100 messages to be reported even though the BLKTCH bit is seen to be set in the Noise Suppression T72 message.

TCH: Indicates that a touch is present. This bit is set to 1 if a touch is present, and is set to 0 if no touch is present.

Under NDA

© 2021 Microchip Technology Inc. and its subsidiaries

ACQCHG: Indicates that the noise suppression algorithm has changed the selected frequency (that is, a frequency hop has occurred) at least once since the previous message. This bit is set to 1 if the selected frequency has changed, and set to 0 otherwise.

STATECHG: Indicates that the noise suppression algorithm has changed its state at least once since the previous message. This bit is set to 1 if the state has changed, and 0 otherwise.

The details of the state change are reported in the STATE field (see below).

NOISELVLCHG: Indicates that the measured noise level has changed at least once since the previous message. This bit is set to 1 if the measured noise level has changed, and 0 otherwise.

NLTHRCHG: Indicates that the noise level threshold has changed at least once since the previous message. This bit is set to 1 if the noise level threshold has changed, and 0 otherwise.

DUALXCHG: Indicates the Dual X operational state has changed at least once since the previous message. This bit is set to 1 if the Dual X operational state has changed, and 0 otherwise.

STATUS2 Field

This field reports the current status of the noise suppression algorithm.

STATE: Indicates the current noise suppression algorithm state selected (see Table 5-36).

TABLE 5-36: NOISE SUPPRESSION ALGORITHM STATES

Value	State
0 to 1	Reserved
2	STABLE
3	NOISY
4	VERY_NOISY
5 to 7	Reserved

DUALX: Indicates that Dual X is operational.

ACQINDEX: Indicates the index of the acquisition parameter set in use for the current noise state. That is, an index into the candidate frequency table (STABFREQ[], NOISFREQ[] or VNOIFREQ[]) and the associated ADC tables (STABTCHAPX[], STABNOTCHAPX[], NOISTCHAPX[], NOISNOTCHAPX[], VNOITCHAPX[] and VNOINOTCHAPX[]).

MINTCHTHR Field

This field reports the current Minimum Touch Threshold.

PKNOISELVL Field

This field reports the peak measured noise level since the last message was dispatched. This noise level is the value of measured noise that is assessed against:

- STABHIGHNLTHR and NOISHIGHNLTHR, to invoke noise state promotion
- NOISLOWNLTHR and VNOILOWNLTHR to invoke noise state demotion
- BLKNLTHR to invoke blocking of touch processing under high noise levels

NOISELVL Field

This field reports the current measured noise level. A higher number means a higher measured noise level, a lower number means a lower measured noise level, and a zero means the noise level is not measured.

NLTHR Field

This field reports the current noise lines threshold.

5.12 Glove Detection T78 Object

5.12.1 INTRODUCTION

The Glove Detection T78 object processes the measurement data received from the Multiple Touch Touchscreen T100 object. It allows touches that are above Multiple Touch Touchscreen T100 INTTHR to be classified as potential gloved touches. Any such touch can be reported in the Multiple Touch Touchscreen T100 messages either as a GLOVE event or as a normal finger touch, as selected by the GLOVERPTEN bit in the CTRL field.

When the Glove Detection T78 object is enabled, it has two modes of operation:

- Normal Mode – The Glove Detection T78 object applies vigorous glove classification to small signal touches to minimize the effect of unintentional hovering finger reporting. The EDGESUP and LOCKEDGESUP bits in the CTRL field and the MINAREA, CONFTHR and MINDIST configuration fields are applied only when in Normal Mode. Once a gloved touch is found, the Glove Detection T78 object enters Glove Confidence Mode.
- Glove Confidence Mode – The Glove Detection T78 object expects the user to be wearing gloves so the classification process is much less stringent.

  Additionally, when Glove Detection T78 object is in this mode, the Key Array T15 object will also respond to small signal touches unless the respective Key Array T15 IGNGLOVEMODE is set (see Key Array T15 “CTRL Field”), in which case the key array needs to be configured to respond to small signal touches.

The Glove Detection T78 object remains in Glove Confidence Mode until any of the following are true:

- The timer controlled by GLOVEMODETO expires.
- The system is reset
- A finger touch is present on the touchscreen. That is, the touch has been above Multiple Touch Touchscreen T100 TCHTHR but has not been below (TCHTHR – TCHHYST).

Care must be taken when using the Passive Stylus T47 object with Glove Detection T78 as the Passive Stylus T47 object also reports small signal touches. Both objects must be correctly configured in order to distinguish stylus touches from glove touches. For example, Passive Stylus T47 CONTMAX and Glove Detection T78 MINAREA will require careful tuning.

If the system supports self capacitance measurements (see Section 3.5.2 “Configuration” and Appendix A “Measurement Processing”), then the glove touch signals which would not usually be detected by Multiple Touch Touchscreen T100 may be reported using the self capacitance touch measurement by setting SCTGLOVETHR. Touches must be above SCTGLOVETHR in both Normal Mode and Glove Confidence Mode. In Normal Mode all usual classification stages must be passed for self capacitance glove touches, with the exception of DISCRIMTHR.

Hysteresis is provided for the SCTGLOVETHR field. For a glove to enter detection the touch delta must be greater than or equal to SCTGLOVETHR. For a glove to leave detection the touch delta must be less than (SCTGLOVETHR – SCTGLOVEHYST).

Note that when glove detections are being tracked in the Multiple Touch Touchscreen T100 object, mutual capacitance scans are performed each cycle, even if the device is using self capacitance measurements as the primary measurement mechanism (see Appendix A “Measurement Processing”).

5.12.2 CONFIGURATION

TABLE 5-37: CONFIGURATION FOR GLOVE DETECTION T78 (PROCI_GLOVEDETECTION_T78)

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	CTRL	GLOVERPTEN	DBGDISCRIMEN	Reserved		LOCKEDGEDSUP	EDGESUP	Reserved	ENABLE
1	MINAREA	Reserved			Minimum contact diameter				
2	CONFTHR	Reserved		Confidence threshold classification cycles					
3	MINDIST	Minimum distance before classification							
4	GLOVEMODETO	Glove confidence mode timeout							
5	SUPTO	Suppression timeout							
6	SYNCSPERX	ADC sets per X							
7	Reserved	Reserved							
8	DISCRIMTHR	Glove discriminant threshold							

Under NDA

© 2021 Microchip Technology Inc. and its subsidiaries

TABLE 5-37: CONFIGURATION FOR GLOVE DETECTION T78 (PROCI_GLOVEDETECTION_T78) (CONTINUED)

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
9	SCTGLOVETHR	Self capacitance touch glove threshold							
10	SCTGLOVEHYST	Reserved	Self-capacitance touch glove hysteresis						
11	SCTMINAREA	Reserved			SCTMINAREA				

CTRL Field

ENABLE: Enables the use of the Glove Detection T78 object. The object is enabled if set to 1, and disabled if set to 0.

EDGESUP: Enables the edge suppression mechanism. When this bit is set to 1, small signal touches that enter the touchscreen from an edge sensor line cannot be classified as glove touches until they move away from the edge.

LOCKEDGESUP: Enables the edge suppression locking mode. When this bit is set to 1, small signal touches that were suppressed by EDGESUP remain suppressed as they move away from the edge.

DBGDISCRIMEN: Enables the outputting of the glove discriminant. Only the discriminant for a single glove touch can be output using this control. When more than one touch is present, the discriminant that is output via this control could correspond to any touch.

NOTE	This bit is for use only during design, under the guidance of Microchip field engineers, to help tune the product for characterization. It should not be used in production. This control should be left at its default value of 0 unless advised otherwise by Microchip.

GLOVERPTEN: If this bit is set to 1, glove touches are reported as a GLOVE event in the Multiple Touch Touchscreen T100 messages. Otherwise, if this bit is set to 0, glove touches are reported as FINGER events. See Section 4.3.4 “Messages” for more information on touch reporting.

MINAREA Field

This field specifies the Minimum Contact Area in nodes for a small signal touch before it can be considered as a glove touch. The contact area is defined as the number of nodes above Multiple Touch Touchscreen T100 (INTTHR – INTTHRHYST). The MINAREA field can help to distinguish between glove touches and hovering fingers, especially when used in conjunction with CONFTHR.

Range: 0 to 31

CONFTHR Field

This field specifies the Confidence Threshold. This is the number of cycles a touch must consistently be considered as a glove touch before it can be classified.

Range: 0 to 63

MINDIST Field

This field specifies the Minimum Distance a small signal touch must move from its touchdown position before it can be classified as a glove touch. The units for this field is $^{1}/_{10}$ of a node, such that 255 represents 25.5 nodes.

Range: 0 to 255

GLOVEMODETO Field

This field specifies the Glove Confidence Mode Timeout. This is the time that the system remains in Glove Confidence Mode after a gloved touch is removed from the screen. The timer is reset if a finger touch is present on the sensor before the timeout has expired. The units for this field is 200 ms.

A value of 0 means an infinite timeout. In this case, the system remains in Glove Confidence Mode indefinitely until an action takes it out of Glove Confidence Mode (that is, the system is reset or a finger touch is present).

A value of 255 means an instant timeout such that the system leaves Glove Confidence Mode immediately after a gloved touch is removed from the touchscreen.

Range: 0 (Infinite), 1 to 254, 255 (Instant)

SUPTO Field

This field specifies a Suppression Timeout. This is the length of time after finger touches are present on the screen before small signal touches can be classified as glove touches. This field is useful, for example, in preventing hovering finger touches from being classified as gloved touches and reported during fast finger taps on the screen while typing. When this setting is enabled, glove touches cannot be classified while fingers are on the touchscreen. The unit for this field is 200 ms, where a value of 0 disables the Suppression Timeout.

Range: 0 (Off) to 255

SYNCSPERX Field

This field specifies the minimum ADC sets per X line to be used when glove touches are being detected subject to other considerations. The actual ADCs per X line value used is the maximum of this SYNCSPERX field (if in detect) and the equivalent values in other objects (see Section 6.8.1 “Introduction” for details).

Range: 1 to 255

DISCRIMTHR Field

This field specifies the Glove Discriminant Threshold. This enables the use of a discriminant based classification threshold for potential glove touches. Note that the value specified by this field applies only to glove touches detected using the mutual capacitance measurement.

A higher value in this field will make it harder for thin or small glove touches to classify, but will decrease the likelihood of false hovering finger detections. A lower value will give less improvement for hovering finger rejection, but will not affect the classification of thinner gloves.

Range: 0 (Off), 1 to 255

SCTGLOVETHR Field

This field specifies the Self Capacitance Touch Glove Threshold in 8 bit delta counts.

If the system supports self capacitance measurements (see “MEASALLOW Field” and Appendix A “Measurement Processing”), it is possible for touches that are not detected by the mutual capacitance measurement to be eligible for glove detection if the average of their X and Y peak delta ≥ SCTGLOVETHR.

Note that EDGESUP, LOCKEDEDGESUP, MINAREA, CONFTHR and MINDIST still apply to eligible self capacitance touches.

If this field has a value less than Auxiliary Touch Configuration T104 INTTHR, then it will have no effect.

Using SCTGLOVETHR to detect thick gloves will allow only a single thick glove to report due to the nature of self capacitance measurements. In order to allow multitouch glove operation, gloves must have deltas above Multiple Touch Touchscreen T100 INTTHR.

Range: 0 (Off), 1 to 255

SCTGLOVEHYST Field

This field specifies the Self Capacitance Touch Glove Hysteresis in 8-bit delta counts. This is the level of hysteresis applied to self capacitance glove detections. See Section 5.12.1 “Introduction” for more details.

The value is limited to (SCTGLOVETHR / 2); see “SCTGLOVETHR Field”. If (SCTGLOVETHR – SCTGLOVEHYST) is less than Auxiliary Touch Configuration T104 (INTTHR – INTHRHYST), then the glove touch will leave detect at Auxiliary Touch Configuration T104 (INTTHR – INTHRHYST). See Section 5.12.1 “Introduction” for more details.

Range: 0 to (SCTGLOVETHR / 2)

SCTMINAREA Field

This field specifies the Self Capacitance Touch Minimum Contact Area. This is the minimum contact area in nodes for self capacitance glove touches before they can be considered as glove touches.

The contact area is defined as the number of X lines above Auxiliary Touch Configuration T104 (XINTTHR – XINTHYST) plus the number of Y lines above Auxiliary Touch Configuration T104 (YINTTHR – YINTHYST).

Range: 0 to 31

Under NDA
© 2021 Microchip Technology Inc. and its subsidiaries

5.12.3 CONFIGURATION CHECKS

The Glove Detection T78 object causes a configuration check to be performed in the following circumstances:

- If the object is enabled, when certain fields are changed

A configuration check may determine that a configuration error has occurred (for example, if a setting is set outside of its allowed range or a conflict has occurred between two settings). This is signaled to the host (see Section 3.3 “Command Processor T6 Object”). The device halts until the error has been corrected. To fix the error, check that all the object settings are within their allowed limits, as stated in the field descriptions.

TABLE 5-38: CONFIGURATION CHECKS FOR GLOVE DETECTION T78 (PROCI_GLOVEDETECTION_T78)

Field	Changing The Field Causes...		Effect of Configuration Checks On Field
	Configuration Check	Automatic Recalibration	
CTRL	Yes	No	None
MINAREA	No	No	None
CONFTHR	No	No	None
MINDIST	Yes	No	None
GLOVEMODETO	No	No	None
SUPTO	No	No	None
SYNCSPERX	No	No	None
DSCRIMTHR	No	No	None
SCTGLOVETHR	No	No	None
SCTGLOVEHYST	No	No	None
SCTMINAREA	No	No	None

5.12.4 MESSAGES

No messages are generated directly for the Glove Detection T78 object. Glove touches are reported through the reporting mechanisms of the linked Multiple Touch Touchscreen T100 object (see Section 4.3.4 “Messages”).

5.13 Retransmission Compensation T80 Object

5.13.1 INTRODUCTION

The Retransmission Compensation T80 object limits the undesirable effects on the mutual capacitance touch signals caused by poor device coupling to ground, such as poor sensitivity and touch break-up. The user can configure this object to allow the linked Multiple Touch Touchscreen T100 object to compensate for signal degradation due to these effects. If self-capacitance measurements are also scheduled, the Retransmission Compensation T80 object will use the resultant data to enhance the compensation process.

Note that this object will only process signals that are above the low tracking threshold of the linked Multiple Touch Touchscreen T100. This threshold should be configured accordingly.

5.13.1.1 Moisture Compensation

The Retransmission Compensation T80 object is also capable of compensating for water presence on the sensor if self capacitance measurements are scheduled. In this case, both mutual capacitance and self capacitance measurements are used to detect moisture. Once moisture is touched, the system goes into single touch mode (that is, self capacitance measurements are used to detect single touches in the presence of moisture). A recalibration can also be performed in self capacitance mode once calibrated-in moisture is detected, if this is configured. See “MOISTCFG Field” for more information.

The TCHIMEASMOV control in the MOISTCFG3 field (see “MOISTCFG3 Field”) allows for the execution of additional touch dilation to account for inter-measurement movement (that is, potential variation in position between the different types of measurements performed within a single cycle). This feature targets Passive Stylus applications that generally exhibit low coupling noise.

The MOISTVLDTHRSF field (see “MOISTVLDTHRSF Field”) configures a self capacitance dynamic threshold that is aimed at finger touches under very noisy conditions. When dynamic thresholding is enabled, touch dilation is effectively reduced.

The behaviors associated with TCHIMEASMOV and MOISTVLDTHRSF are independent and can counteract one another to some degree if these controls are used together. Care should be taken to apply them appropriately, based upon touch type and noise state.

If either Self Capacitance Global Configuration T109 DISMEASX or DISMEASY are set to 1 (that is, a particular axis has been disabled) due to one axis of self capacitance data not being measurable, then moisture compensation will not run. Note, however, that retransmission compensation will use mutual capacitance data only.

5.13.1.2 Moisture Degraded Mode

Moisture Degraded Mode disables close scanning for touch detections in order to prevent small amounts of moisture around touches being reported as separate touches. It also suspends drift. As a result:

- Moisture around touches cannot be split from the real touch and is counted as part of the touch.
- Minimum touch separation is increased.
- Finger touches will still be detected on the dry part of the sensor, but other touches (for example, glove, passive stylus) will be detected only if DISINVTCHREM = 1 and no touches breach the self capacitance touch INTTHR–INTTHRHYST in either axis.

The system can be configured to enter Moisture Degraded Mode using the MOISTINVATCHTHR and CTRL DISINVTCHREM controls (see “MOISTINVATCHTHR Field” and “CTRL Field”). By setting these controls, Moisture Degraded Mode will be entered in either of the following two cases:

- The number of degraded nodes is at least the MOISTINVATCHTHR value
  (Note: the number of degraded nodes is defined as the number of nodes more negative than the degraded delta threshold).

OR

- All of the following conditions are satisfied:
  - DISINVTCHREM = 1

  AND
  - Self capacitance touch is below (INTTHR–INTTHRHYST) in both axes (that is, there are deemed to be no touches)

Under NDA
© 2021 Microchip Technology Inc. and its subsidiaries

AND

- Mutual capacitance negative peak delta is less (further from zero) than –2×(INTTHR–INTTHRHYST)

AND

- Mutual capacitance negative peak delta is not on an edge of the touchscreen

5.13.1.3 Working Threshold

The Moisture Threshold (MOISTTHR) and Moisture Degraded Threshold (MOISTDEGTHR) have internal limits to keep the threshold out of the noise. The MOISTTHR and MOISTDEGTHR fields therefore specify base thresholds that are then manipulated internally to determine the actual working thresholds. The process is as follows:

1. Multiply the threshold by the Multiple Touch Touchscreen T100 Dual X threshold scaling factor, if running in Dual X mode.
2. Multiply the threshold by the Retransmission Compensation T80 coupling level.
3. Use Noise Suppression T72 Minimum Touch Threshold instead if this is further from zero.

If the working threshold is closer to zero than the Multiple Touch Touchscreen T100 antitouch threshold, then there is a time penalty on a moist sensor depending on the number of X lines affected by moisture. Touch latency will be increased, and the report rate might be reduced.

The time penalty is heavily dependent on the configuration as well as the area of moisture. When tuning the configuration, the report rate should be measured with a representative (for the intended application) area of moisture on the sensor and a decision made on whether the time penalty is acceptable.

For more information on the Multiple Touch Touchscreen T100 antitouch threshold, see the ATCHTHRSEL control in the Multiple Touch Touchscreen T100 "CFG1 Field").

5.13.2 CONFIGURATION

TABLE 5-39: CONFIGURATION FOR RETRANSMISSION COMPENSATION T80 (PROCI_RETRANSMISSIONCOMPENSATION_T80)

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	CTRL	Reserved		DISATCHCHECK	DISINVTCHREM	COMPENHEN	DISGAINLIMIT	RPTEN	ENABLE
1	COMPGAIN	Compensation gain							
2	TARGETDELTA	Target delta							
3	COMPTHR	Compensation threshold							
4	ATCHTHR	Antitouch threshold							
5	MOISTCFG	MOISTCALEN	MOISTEN	MOISTDI					
6	Reserved	Reserved							
7	MOISTTHR	Moisture threshold							
8	MOISTINVATCHTHR	Moisture invalid antitouch threshold							
9	MOISTCFG2	MOISTCALMODE	MOISTFIREN	Reserved		MOISTDETINT			
10	COMPSTRTHR	Compensation strength threshold							
11	COMPCFG	Reserved				BNDGBOXTHR			
12	MOISTVLDTHRSF	Moisture validation threshold scale factor							
13	MOISTCFG3	Reserved					TCHIMEASMOV		
14	MOISTDEGTHR	Moisture Degraded Threshold							

CTRL Field

ENABLE: Enables or disables the use of the Retransmission Compensation T80 object. The object is enabled if set to 1, and disabled if set to 0.

RPTEN: Allows the object to send status messages to the host through the Message Processor T5 object. Reporting is enabled if set to 1, and disabled if set to 0.

DISGAINLIMIT: Disables gain limiting based on TARGETDELTA when using mutual data for compensation.

NOTE	This bit applies only when using mutual capacitance only retransmission compensation.

COMPENHEN: Enables Compensation Enhancement.

NOTE	This bit applies only when using mutual capacitance only retransmission compensation.

If this bit is set to 1, retransmission compensation that is using mutual data is enhanced such that compensated touches are made larger (at the possible expense of pinch separation). When this control is enabled to improve large touch performance, COMPGAIN should be reduced to the minimum value required to achieve the desired performance.

DISINVTCHREM: Disables Invalid Touch Removal. The Retransmission Compensation T80 object identifies regions of potential moisture on a touchscreen from the self capacitance and mutual capacitance measurements. By default the Retransmission Compensation T80 object removes invalid nodes (that is nodes identified as having moisture present). If DISINVTCHREM is set to 1, however, invalid touches are not removed. This allows small touches to be detected in the presence of moisture. See Section 5.13.1.2 “Moisture Degraded Mode” for details on how DISINVTCHREM determines Moisture Degraded Mode.

DISATCHCHECK: If this bit is set to 1, it disables a check for the presence of antitouch to indicate that a touch should be recovered by retransmission compensation (see “ATCHTHR Field”).

COMPGAIN Field

When mutual capacitance measurements only are allowed, this field specifies the Compensation Gain. This is the strength of the applied compensation when the device is poorly coupled to the user. This field should be tuned based on the screen type, panel thickness, and so on.

If self capacitance measurements are enabled, this field controls whether or not retransmission compensation is applied. If this field is set to 0 (default), no retransmission compensation is applied and the Retransmission Compensation T80 object provides moisture compensation only. If this field is set to a non zero value, retransmission compensation is applied according to TARGETDELTA (a non-zero value effectively has no meaning).

Range (mutual capacitance only): 0 to 255

Range (self capacitance enabled): 0 (no compensation applied),
>0 (compensation applied according to TARGETDELTA)

Typical (mutual capacitance only): 40

TARGETDELTA Field

This field specifies the Target Delta in 8-bit deltas. This is the desired mutual capacitance touch delta for a fully grounded touch. It is used to calculate the coupling level for the mutual capacitance compensation algorithm. It is also used to limit the gain for both the mutual and self capacitance versions of the algorithm.

Range: 0 to 255

Typical: 125

COMPTHR Field

This field specifies the Compensation Threshold in 8-bit deltas. If the antitouch exceeds ATCHTHR and the compensated delta count of a node reaches the Compensation Threshold, then compensation is applied to that node. When this field is set to 0, there is no Compensation Threshold criteria so all compensated nodes are modified.

COMPTHR should be set to a value that is less than TARGETDELTA (typically 50% to 80% of TARGETDELTA).

Range: 0 (disabled), 1 to 255

Typical: 50% to 80% of TARGETDELTA

ATCHTHR Field

This field specifies the Antitouch Threshold in 8-bit deltas. This threshold specifies the negative peak antitouch in mutual capacitance deltas that must be breached for retransmission compensation to take effect. When this control is disabled, retransmission compensation will run regardless of antitouch presence (as long as other criteria for compensation are met).

Under NDA
© 2021 Microchip Technology Inc. and its subsidiaries

NOTE	A retransmitting touch typically incurs some antitouch deltas, whereas a well-coupled, low-signal touch will not. The objective of the Retransmission Compensation T80 object, therefore, is to compensate retransmitting touches but not well-coupled, low-signal touches.

A typical setting for this control is just below the negative noise floor (that is, equivalent of INTTHR of the associated Multiple Touch Touchscreen T100), but this field may require some tuning depending on the application and the other processing objects used. A value of 0 disables this feature.

Range: 0 (disabled), 1 to 255

MOISTCFG Field

MOISTDI: Specifies the Moisture Detection Integration. This is the number of extra mutual capacitance frames requested to confirm the persistent invalid touch present before calling calibration.

This feature is useful for a slow moving stylus, as this may drop below the detection threshold of the associated object (Auxiliary Touch Configuration T104 in this case) under some circumstances and trigger a recalibration.

It is recommended that this functionality is enabled by setting MOISTDI to a value. If MOISTDI is left at the default value of 0 (disabled), calibrated-in moisture may not be discriminated from noise and moisture compensation will be degraded.

Note that MOISTCALEN must be enabled for this feature to have an effect.

MOISTDI Range: 0 (disabled), 1 to 63

MOISTEN: Enables moisture compensation. If this bit is set to 1, the Retransmission Compensation T80 object removes noise due to moisture so that the moisture is not detected as touches.

Note that this feature will run only if self capacitance measurement is configured and enabled (see "MEASALLOW Field" and Section 6.13 "Auxiliary Touch Configuration T104 Object").

MOISTCALEN: Enables recalibration when persistent invalid touches are present (that is, in regions where moisture has been calibrated in).

Note that this feature will run only if self capacitance measurement is configured and enabled (see "MEASALLOW Field" and Section 6.13 "Auxiliary Touch Configuration T104 Object").

MOISTTHR Field

This field configures the Moisture Threshold in 8-bit (negative) delta counts. This threshold defines the amount of antitouch that must be present before nodes are marked as being moisture. Note that there is an internal limit to keep the working threshold out of the noise. See Section 5.13.1.3 "Working Threshold" for details.

Range: 0 to 255

MOISTINVATCHTHR Field

This field configures the Moisture Invalid Antitouch Threshold. This is the number of invalid antitouch nodes that can be seen before the touchscreen is marked as degraded.

If the number of invalid antitouch nodes is greater than or equal to MOISTINVATCHTHR, the touchscreen will report only touches above one of the following thresholds (depending on the value of MOISTDEGTHR):

- If MOISTDEGTHR = 0, the threshold is the Multiple Touch Touchscreen T100 antitouch threshold. See the ATCHTHRSEL control in the Multiple Touch Touchscreen T100 "CFG1 Field") for more information on the Multiple Touch Touchscreen T100 antitouch threshold.
- If MOISTDEGTHR > 0, the threshold is the value of MOISTDEGTHR.

If the threshold is met, and the touchscreen is marked as degraded, a DEGACTV message is sent (see "STATUS Field"). See also Section 5.13.1.3 "Working Threshold".

A value of 0 disables this feature.

Range: 0 (disabled), 1 to 255

MOISTCFG2 Field

MOISTDETINT: Configures the Moisture Detection Interval. This is the interval between the periodic requests for mutual scans when self capacitance measurements are the primary measurement type. The interval is specified in units of 200 ms.

Range: 0 (disable), 1 to 15

Typical: 5

MOISTFIREN: Enables FIR Filtering on SCT Deltas.

If this bit is set to 1, FIR filtering with fixed coefficients is applied to the SCT delta values of both axes prior to further processing. This feature is particularly useful with a passive stylus, when the loss of tracking is observed because of the moisture compensation.

If this bit is set to 0, raw SCT deltas will be used for normal processing.

MOISTCALMODE: Configures the Moisture Calibration Mode. This controls the calibration behavior for moisture.

If this bit is set to 0, then calibration is triggered only when calibrated-in moisture has been removed.

If this bit is set to 1, then calibration is triggered when moisture is either added or removed.

COMPSTRTHR Field

This field defines the Compensation Strength Threshold. This is the strength of the self capacitance deltas that, when reached, causes touch compensation to be applied in the case where there is not enough antitouches present in the mutual data (that is, ATCHTHR is not exceeded; see “ATCHTHR Field”). The strength of the touch is evaluated as the peak self capacitance 8-bit delta divided by the number of nodes above threshold.

This field is provided in order to allow touches to be reported, even though retransmission effects are substantially reducing the level of antitouches present. This is caused when touches are placed on the same line of the axis, or fingers are placed over the positions of antitouch associated with retransmission effects (such as touches on all four corners of a square).

This field also prevents hovering fingers being compensated. This value should be increased when the system is compensating unwanted hovering fingers; it should be decreased when small touches on the same X or Y line need to be compensated.

Range: 0 (disable), 1 to 255

Typical: 20

COMPCFG Field

This field configures retransmission compensation.

BNDGBOXTHR: Specifies the Bounding Box Threshold. This is the minimum area (in nodes) that must be reached before retransmission compensation is applied. The area is defined as the total rectangular extent of all deltas exceeding the lower touch threshold for the measurement type (mutual or self capacitance) that is being used to do the compensation (see Section 4.3 “Multiple Touch Touchscreen T100 Object” and Section 6.13 “Auxiliary Touch Configuration T104 Object” for more information on thresholds). If the area is equal to or greater than BNDGBOXTHR, retransmission compensation is applied.

For areas below BNDGBOXTHR, compensation is not applied, in order to avoid the incorrect boosting of low amplitude touch deltas (such as, stylus touches or noise spikes). If stylus support is not required, then the minimum area limit can be relaxed from the default value so that small or light touches can be boosted.

Range: 0 (10), 1 to 15

MOISTVLDTHRSF Field

This field specifies the Moisture Validation Threshold Scale Factor. This is used to modify the threshold that is used to validate mutual capacitance deltas as part of moisture processing. Mutual capacitance data is checked against the self capacitance data for invalid nodes. By default, this happens when the Auxiliary Touch Configuration T104 INTTHR is reached, but the MOISTVLDTHRSF field allows a portion of the peak positive touch delta to be used instead. Use of this field can only increase the threshold, not reduce it. Care is needed to use this field appropriately, as it would be possible to raise the threshold above smaller touches if a larger touch is also present. Generally this field should be triggered in noisy conditions for finger touches (which couple in much of the noise).

Under NDA
© 2021 Microchip Technology Inc. and its subsidiaries

Note that MOISTFIREN must be enabled for this feature to work. DISINVTCHREM should also be enabled to maintain the reporting of small touches (for example, passive stylus touches) in the presence of larger touches.

The value for this fields is specified in units of 1/256 of the peak positive touch delta, where the default value of 0 means the feature is disabled.

Range: 0 (disabled), 1 to 255

MOISTCFG3 Field

TCHIMEASMOV: Specifies the Inter-Measurement Movement for Touches. This prevents fast-moving touches from potentially being marked as moisture due to the timing between different types of touch measurements on a single cycle. The value should be set to at least the number of lines over which a touch can move between the different types of measurement. Larger values might be required for larger touchscreens.

Note that TCHIMEASMOV controls dilation for small touches only (that is, self capacitance delta values below Auxiliary Touch Configuration T104 TCHTHR). For touches above Auxiliary Touch Configuration T104 TCHTHR no dilation is applied.

The minimum effective value for this control is 2, which accounts for a potential positional difference of at least one node between the different types of measurements within a single cycle. The value is specified as the number of lines plus 1, where the default value of 0 means 2 (that is, a single line).

Range: 0 (2 = 1 line), 1 to 7

MOISTDEGTHR Field

This field configures the Moisture Degraded Threshold used to determine Moisture Degraded Mode (see Section 5.13.1.2 “Moisture Degraded Mode”.

If MOISTDEGTHR is non zero, it configures the (negative) delta threshold used to count antitouch (invalid) nodes for determining Moisture Degraded Mode. See also “MOISTINVATCHTHR Field”, which defines the number of antitouch nodes required to enter Moisture Degraded Mode.

If MOISTDEGTHR is zero (default), the Multiple Touch Touchscreen T100 antitouch threshold will be used instead. See the ATCHTHRSEL control in the Multiple Touch Touchscreen T100 “CFG1 Field”) for more information on the Multiple Touch Touchscreen T100 antitouch threshold.

Note that there is an internal limit to keep the working threshold out of the noise. See Section 5.13.1.3 “Working Threshold” for details.

Range: 0 (use Multiple Touch Touchscreen T100 antitouch threshold), 1 to 255

5.13.3 CONFIGURATION CHECKS

The Retransmission Compensation T80 object causes a configuration check to be performed in the following circumstances:

- If the object is enabled, when certain fields are changed.

A configuration check may determine that a configuration error has occurred (for example, if a setting is set outside of its allowed range or a conflict has occurred between two settings). This is signaled to the host (see Section 3.3.3 “Messages”). The Retransmission Compensation T80 halts until the error has been corrected. To fix the error, check that all the object settings are within their allowed limits, as stated in the field descriptions.

TABLE 5-40: CONFIGURATION CHECKS FOR RETRANSMISSION COMPENSATION T80 (PROCI_RETRANSMISSIONCOMPENSATION_T80)

Field	Changing The Field Causes...		Effect of Configuration Checks On Field
	Configuration Check	Automatic Recalibration	
CTRL	Yes	No	None
COMPGAIN	No	No	None
TARGETDELTA	No	No	None
COMPTHR	No	No	None
ATCHTHR	No	No	None
MOISTCFG	No	No	None

TABLE 5-40: CONFIGURATION CHECKS FOR RETRANSMISSION COMPENSATION T80 (PROCI_RETRANSMISSIONCOMPENSATION_T80) (CONTINUED)

Field	Changing The Field Causes...		Effect of Configuration Checks On Field
	Configuration Check	Automatic Recalibration	
MOISTTHR	Yes	No	None; but see Section 5.13.1.3 “Working Threshold” for behavior
MOISTINVATCHTHR	No for	No	None
MOISTCFG2	No	No	None
COMPSTRTHR	No	No	None
COMPCFG	No	No	None
MOISTVLDTHRSF	No	No	None
MOISTCFG3	No	No	None
MOISTDEGTHR	No	No	None

5.13.4 MESSAGES

The message data for the Retransmission Compensation T80 object is shown in Table 5-41.

TABLE 5-41: MESSAGE DATA FOR RETRANSMISSION COMPENSATION T80 (PROCI_RETRANSMISSIONCOMPENSATION_T80)

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	STATUS	COMPACTV	Reserved			DEGACTV	TCHMOISTDET	MOISTDET	TCHDET

STATUS Field

This field reports the on a number of states. A new message is generated whenever there is a change of state.

TCHDET: Indicates that retransmission/moisture processing has detected a touch.

MOISTDET: Indicates that moisture processing has detected moisture.

TCHMOISTDET: Indicates that moisture processing has detected a touch in moisture.

DEGACTV: Indicates that moisture processing is running in degraded mode (see “MOISTINVATCHTHR Field” and “MOISTDEGTHR Field”).

COMPACTV: Indicates that retransmission compensation is actively reconstructing touches. Note, however, that although COMPACTV may be set, touches may not be reconstructed due to other configuration factors (for example, COMPTHR is set too high).

Under NDA
© 2021 Microchip Technology Inc. and its subsidiaries

5.14 Self Capacitance Noise Suppression T108 Object

5.14.1 INTRODUCTION

The Self Capacitance Noise Suppression T108 object provides a noise suppression algorithm to suppress the effects of external noise.

Similar in both design and configuration to the Noise Suppression T72 object, the Self Capacitance Noise Suppression T108 object is the noise suppression interface for self capacitance touch measurements.

Furthermore, to utilize the Self Capacitance Noise Suppression T108 object, the Noise Suppression T72 object must also be enabled and configured, as both objects are essential to managing noise within self capacitance touch measurements.

Such noise can take many forms, including:

- **Internal noise** – due to electrical interferences inherent in the product's architecture (such as LCD display emissions)
- **Environmental noise** – due to exposure to noise within the user's living environment (such as from other electrical equipment in near proximity)
- **Charger noise** – due to noise induced by peripheral equipment used to recharge the product's battery.

5.14.1.1 Noise Notification Mechanisms

The Self Capacitance Noise Suppression T108 object needs to be made aware of when a noise source appears and disappears (for example, when a charger is turned on or off) or when the current noise level rises to a level beyond which the current acquisition parameters can cope. The following notification mechanisms are available:

- **Software trigger** – The host can set the NOISEON bit in the CALCFG1 field when a noise source appears. For example, this bit should be set when a noisy charger is plugged into the user's product. Note that the alternative use of the NOISEON bit in the CFG1 field is not recommended.
- **Automatic transition** – The noise suppression algorithm automatically transitions between the noise states as the noise level exceeds or drops below certain thresholds. These thresholds are specified by the STABHINLTDTHR, NOISLONLTDTHR, NOISHINLTDTHR and VNOILONLTDTHR fields.

IMPORTANT!	Enabling and disabling the Self Capacitance Noise Suppression T108 object dynamically using the CTRL ENABLE bit in response to noise is not recommended, as this can cause unstable behavior. The correct procedure is to enable the object at startup and then use one of the notification methods above whenever noise is detected.

5.14.1.2 Transitions Between Noise States

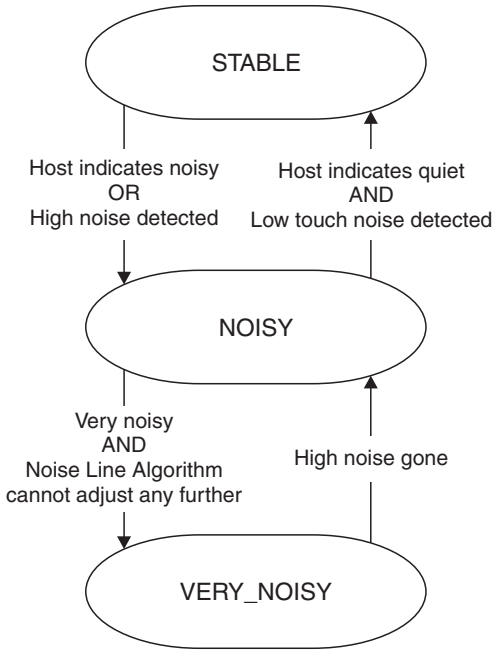
The Self Capacitance Noise Suppression T108 can also dynamically switch between three wholly independent configurations presets, known as noise states. The three noise states are referred to as STABLE, NOISY and VERY_NOISY, and are intended to be configured to counter the effects of increased measured noise. Typically noise states offer differing touch measurement configuration, which improves noise immunity at the expense of acquisition power consumption touch.

Noise state transition as a consequence of increased noise (that is, from STABLE to NOISY, or NOISY to VERY_NOISY) is known as promotion, which typically occurs when noise escalates beyond the effectiveness of frequency hopping (see Section 5.14.1.3 "Frequency Hopping").

Conversely, noise state transition as a consequence of decreased noise (that is, from VERY_NOISY to NOISY, or NOISY to STABLE) is known as demotion, which occurs when noise recedes to more satisfactory levels.

Figure 5-18 shows how the noise suppression algorithm transitions between the STABLE, NOISY and VERY_NOISY noise states as it notified of increasing or decreasing levels of noise.

FIGURE 5-18: NOISE SUPPRESSION STATES



In each state the Self Capacitance Noise Suppression T108 object can dynamically load a new set of acquisition parameters to cope with the current level of noise. This provides a framework within which the behavior of the Self Capacitance Noise Suppression T108 can be modified to provide better noise handling at the expense of touch performance and/or touch message rate.

On entry to a new noise state, the Self Capacitance Noise Suppression T108 object loads the appropriate set of control, frequency and ADCs per X parameters and the touchscreen is recalibrated, if needed, to use the new settings.

5.14.1.3 Frequency Hopping

The Self Capacitance Noise Suppression T108 object works much like the Noise Suppression T72 object. It uses an automatic algorithmic method whereby measurements on special “noise lines" allow an appropriate action to be taken, such as choosing a new frequency.

For touch measurement, there are up to 5 configurable frequency presets, but the actual number that are used is configurable (see NUMPRESETS control in the “CFG3 Field”). Each preset is accompanied by ADCs Per X configurations, for both touch and no-touch environments. The acquisition index (ACQIDX) addresses the preset in the frequencies array that is currently being used for the line bursts as part of touch measurement.

At any given juncture, one of these presets is used to drive touch measurements and the index of the selected preset is reported in the Self Capacitance Noise Suppression T108 messages.

The algorithm for noise suppression works by analyzing the maximum absolute values of the noise levels (known as noise lines tallest deltas – NLTD) to determine the nature of the noise detected and the consequential best course of action. This data is processed using IIR filters, so that inherent fluctuations in noise measurements can be countered.

Furthermore, continuous noise measurements are also performed automatically, and these are completely separate from the touch requirement. This carousel of measurements, from which a set of IIR filters are updated is referred to as background scanning.

The subsequent determination of the optimal frequency configuration to minimize the effects of noise at a given moment, and the transition to a different preset configuration for touch measurement, is known as frequency hopping.

5.14.1.4 Background Scanning and Touch Hold-off Qualification

For the Self Capacitance Noise Suppression T108 object to perform effective frequency hopping, continuous noise measurements are made for each of the candidate presets. This allows for the selection of the optimal settings that can best avoid the effects of noise.

The Self Capacitance Noise Suppression T108 object maintains two sets of IIR filter measurements (carousels), each set associated with a specific touch or no-touch state.

Under NDA
© 2021 Microchip Technology Inc. and its subsidiaries

A pre-qualification process is applied to background noise measurements. This collates the measurements until all the measurements are deemed to have been collated within a period of consistent touch-state reporting. Once this is achieved the carousel filters updated and any subsequent frequency hopping determination is made. If a touch state transition occurs during a measurement cycle, all the noise measurements are rejected and a new carousel collation begins. This process is referred to as Touch Hold-off.

The Touch Hold-off mechanism can improve the frequency hopping candidate selection process, as the filter carousel content is more accurate when repeated touch transitions take place. Without this process, mis-reports from false touches and/or momentary loss of touch determination can reduce the effectiveness of any frequency hopping.

Touch Hold-off is selectable using the TCHHOFFEN bit in the CFG3 field (see "CFG3 Field").

5.14.1.5 Suggested Initial Settings

Suggested initial settings for the Self Capacitance Noise Suppression T108 object are given in Table 5-42.

TABLE 5-42: SUGGESTED INITIAL SETTINGS

Field	Suggested Initial Setting
CTRL	ENABLE = 1 Other bits as required by user
CALCFG1	0 when a potential noise source is not present (for example, a charger is turned off) 1 (NOISEON = 1) when a potential noise source is notified by the host (for example, a charger is turned on)
CFG1	0
CFG2	0
CFG3	64 (TCHHOFFEN = 1)
NLGAIN	Field needs tuning
IIRCOEFF	0
MINNLTDDIFF	6 Field benefits from tuning
MINNLTDHOP	3 Field benefits from tuning
HOPST	0
STABCTRL	0
STABFREQ[]	Fields need tuning
STABTCHAPX[]	Fields need tuning
STABNOTCHAPX[]	Fields need tuning
STABHINLTDTHR	Field needs tuning
NOISCTRL	1 (AUTO = 1)
NOISFREQ[]	Fields need tuning
NOISCHAPX[]	Fields need tuning
NOISNOTCHAPX[]	Fields need tuning
NOISLONLTDTHR	Field needs tuning
NOISHINLTDTHR	Field needs tuning
NOISCNT	3 (NOTCH = 3)
VNOICTRL	1 (AUTO = 1)
VNOIFREQ[]	Fields need tuning
VNOITCHAPX[]	Fields need tuning
VNOINOTCHAPX[]	Fields need tuning

TABLE 5-42: SUGGESTED INITIAL SETTINGS (CONTINUED)

Field	Suggested Initial Setting
VNOILONLTDTHR	Field needs tuning
VNOICNT	3 (NOTCH = 3)
BLKNLTDTHR	Field needs tuning

5.14.2 CONFIGURATION

TABLE 5-43: CONFIGURATION FOR SELF CAPACITANCE NOISE SUPPRESSION T108 (PROCG_NOISESUPSELFCAP_T108)

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	CTRL	Reserved			RPTLVLEN	RPTSTATEEN	RPTACQIDXEN	RPTEN	ENABLE
1	CALCFG1	Reserved						VNOISEON	NOISEON
2	CFG1	Reserved						VNOISEON	NOISEON
3	CFG2	NOTCHNOISEON	Reserved						DISBLKTCH
4	CFG3	Reserved	TCHHOFFEN	Reserved			NUMPRESETS		
5	NLGAIN	Noise line gain							
6	Reserved	Reserved							
7	IIRCOEFF	Reserved					NLTD		
8	MINNLTDDIFF	Minimum measured filter range for NLTD-based frequency hopping algorithm							
9	MINNLTDHOP	Minimum filter change for NLTD-based frequency hopping algorithm							
10	Reserved	Reserved							
11	Reserved	Reserved							
12	HOPST	Hop Suspend Time							
13	STABCTRL	CAL	Reserved						
14	STABFREQ[]	STABLE State Frequency Setting 0							
15		STABLE State Frequency Setting 1							
16		STABLE State Frequency Setting 2							
17		STABLE State Frequency Setting 3							
18		STABLE State Frequency Setting 4							
19	STABTCHAPX[]	STABLE State Touch ADCs per X Setting 0							
20		STABLE State Touch ADCs per X Setting 1							
21		STABLE State Touch ADCs per X Setting 2							
22		STABLE State Touch ADCs per X Setting 3							
23		STABLE State Touch ADCs per X Setting 4							
24	STABNOTCHAPX[]	STABLE State No Touch ADCs per X Setting 0							
25		STABLE State No Touch ADCs per X Setting 1							
26		STABLE State No Touch ADCs per X Setting 2							
27		STABLE State No Touch ADCs per X Setting 3							
28		STABLE State No Touch ADCs per X Setting 4							
29 – 30	Reserved	Reserved							
31	STABHINLTDTHR	STABLE State high NLTD noise level							
32	Reserved	Reserved							
33	NOISCTRL	CAL	Reserved						AUTO
34	NOISFREQ[]	NOISY State Frequency Setting 0							
35		NOISY State Frequency Setting 1							
36		NOISY State Frequency Setting 2							
37		NOISY State Frequency Setting 3							
38		NOISY State Frequency Setting 4							

Under NDA
© 2021 Microchip Technology Inc. and its subsidiaries

TABLE 5-43: CONFIGURATION FOR SELF CAPACITANCE NOISE SUPPRESSION T108 (PROCG_NOISESUPSELFCAP_T108) (CONTINUED)

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
39	NOISTCHAPX[]	NOISY State Touch ADCs per X Setting 0							
40		NOISY State Touch ADCs per X Setting 1							
41		NOISY State Touch ADCs per X Setting 2							
42		NOISY State Touch ADCs per X Setting 3							
43		NOISY State Touch ADCs per X Setting 4							
44	NOISNOTCHAPX[]	NOISY State No Touch ADCs per X Setting 0							
45		NOISY State No Touch ADCs per X Setting 1							
46		NOISY State No Touch ADCs per X Setting 2							
47		NOISY State No Touch ADCs per X Setting 3							
48		NOISY State No Touch ADCs per X Setting 4							
49	Reserved	Reserved							
50	NOISLONLTDTHR	NOISY State low NLTD noise level							
51	NOISHINLTDTHR	NOISY State high NLTD noise level							
52	NOISCNT	LOWNOISTCH				NOTCH			
53	VNOICTRL	CAL	Reserved						AUTO
54	VNOIFREQ[]	VERY_NOISY State Frequency Setting 0							
55		VERY_NOISY State Frequency Setting 1							
56		VERY_NOISY State Frequency Setting 2							
57		VERY_NOISY State Frequency Setting 3							
58		VERY_NOISY State Frequency Setting 4							
59	VNOITCHAPX[]	VERY_NOISY State Touch ADCs per X Setting 0							
60		VERY_NOISY State Touch ADCs per X Setting 1							
61		VERY_NOISY State Touch ADCs per X Setting 2							
62		VERY_NOISY State Touch ADCs per X Setting 3							
63		VERY_NOISY State Touch ADCs per X Setting 4							
64	VNOINOTCHAPX[]	VERY_NOISY State No Touch ADCs per X Setting 0							
65		VERY_NOISY State No Touch ADCs per X Setting 1							
66		VERY_NOISY State No Touch ADCs per X Setting 2							
67		VERY_NOISY State No Touch ADCs per X Setting 3							
68		VERY_NOISY State No Touch ADCs per X Setting 4							
69	Reserved	Reserved							
70	VNOILONLTDTHR	VERY_NOISY State Low noise detection level							
71	Reserved	Reserved							
72	VNOICNT	LOWNOISTCH				NOTCH			
73	BLKNLTDTHR	Maximum NLTD threshold before blocking touch processing							
74	Reserved	Reserved							

CTRL Field

This is the main control field.

ENABLE: Enables the Self Capacitance Noise Suppression T108 object. The object is enabled if set to 1, and disabled if set to 0.

RPTEN: Allows this object to send messages to the host through the Message Processor T5 object. Reporting is enabled if set to 1, and disabled if set to 0.

RPTACQIDXEN: Allows selected frequency or ADCs per X changes to trigger reporting messages to the host. Reporting is enabled if set to 1, and disabled if set to 0.

RPTSTATEEN: Allows changes in noise state or blocking of touch processing to trigger reporting messages to the host. Reporting is enabled if set to 1, and disabled if set to 0.

RPTLVLEN: Allows noise level changes to trigger reporting messages to the host. Reporting is enabled if set to 1, and disabled if set to 0.

CALCFG1 Field

NOTE	Writing this field will cause a recalibration to occur.

NOISEON: This bit can be set to 1 by the host to indicate that a noise source is activated for the product (for example, a charger is turned on). The object state-machine will change to the NOISY state if it was previously in the STABLE state

VNOISEON: Manually forces a change to the VERY_NOISY state for use during setup or development. If this bit is set to 1, the object state machine changes to the VERY_NOISY state if it was previously in the NOISY or STABLE state.

CFG1 Field

NOTE	Use of this field is not recommended.

This field repeats bits from CALCFG1 field so that these bits can be changed without causing a touchscreen recalibration.

NOISEON: See "CALCFG1 Field".

VNOISEON: See "CALCFG1 Field".

CFG2 Field

DISBLKTCH: This bit can be set to 1 by the host to disable the blocking of touch processing when noise state promotion occurs. When a noise state transition occurs, it is often in response to a sudden escalation of measured noise, which may cause touch processing to perform ineffectively. The default behavior, when this bit is set to 0, is to block touch processing briefly as a safeguard in these circumstances. If this bit is set to 1, then no intervention with touch processing is made.

NOTCHNOISEON: Enables demotion of noise-level from either NOISY or VERY_NOISY states when clamped to one of these noise-levels by either the NOISEON or VNOISEON bits (CALCFG1 or CFG1 field) respectively. Consequently, this field alters the default behavior when noise-level promotion is enforced by configuration settings established by the host device.

Such a demotion can only occur during continuous touch, whereby noise-levels are deemed acceptably low. Essentially this bit allows the device to assume that noise is present until a touch is introduced for a sufficient time to determine its demotion to a quieter noise state.

If such a demotion occurs, then as soon as touch is removed, the noise-level will revert to the original state, as mandated by the configuration bits specified above.

CFG3 Field

NUMPRESETS: Specifies the number of frequency presets to use for touch measurement. Up to five preset frequencies are available, as defined in "STABFREQ[], NOISFREQ[] and VNOIFREQ[] Fields", "STABFREQ[], NOISFREQ[] and VNOIFREQ[] Fields", and "STABFREQ[], NOISFREQ[] and VNOIFREQ[] Fields". The default of 0 means that the maximum number of 5 presets will be used.

Range: 0 (5), 1 to 5

TCHHOFFEN: Enables Touch Hold-off (see Section 5.14.1.4 "Background Scanning and Touch Hold-off Qualification"). If this bit is set 1, the IIR filters are held off from being updated until touch status (in-touch or no-touch) is consistent over time. If a touch state transition occurs during a measurement cycle, all the noise measurements are rejected, and a new carousel collation begins.

This option improves filter quality during touch and no-touch transitions, but does so at the potential expense of frequency-hopping responsiveness.

NLGAIN Field

This field specifies the Noise Lines Gain to be used. Note that this is the gain for use when making noise line measurements; the gain for self capacitance touch measurements is set by the Auxiliary Touch Configuration T104 object.

Under NDA
© 2021 Microchip Technology Inc. and its subsidiaries

The value for the noise line gain is based on the mutual capacitance gain value, so the chosen value should be Multiple Touch Touchscreen T100 "GAIN Field".

Range: 0 to 22

Typical: Multiple Touch Touchscreen T100 GAIN

IIRCOEFF Field

This field specifies the coefficients applied to IIR filters that affect their response to changes in noise.

NLTD: Specifies the Coefficient for the IIR filtering of noise lines tallest deltas (NLTD) values. The range is 1 to 7, where a value of 1 is the fastest and 7 is the slowest. The default value of 0 means 5.

NLTD Range: 0 (5), 1 to 7

MINNLTDDIFF Field

This field specifies the minimum range (in 8-bit deltas) for the NLTD IIR filters below which the algorithm deems that noise measurements cannot yield a suitable candidate for frequency hopping. The minimum range is defined as the absolute extent between the noisiest and quietest filters associated with the presets currently available, as configured for the current noise state.

When the measured range is greater than or equal to this difference threshold, the frequency selection algorithm will make use of the NLTD IIR filters to determine a candidate preset for touch measurements.

This field (which is used in conjunction with the "MINNLTDHOP Field") is used to prevent excessive frequency hopping under very low noise environments, or when there is no clear candidate, based on IIR filter analysis.

A value of 255 effectively means that the NLTD data can never yield a suitable candidate for frequency hopping.

Range: 0 to 255

MINNLTDHOP Field

This field specifies the minimum filter change (in 8-bit deltas) for NLTD IIR filters below which it is deemed that noise measurements do not indicate a suitable frequency hopping candidate (that is, less effected by noise).

This filter change is defined as the difference between the NLTD filter value of the currently selected acquisition index and that of the quietest alternative NLTD filter. If the difference in the two filters exceeds this specified minimum, then a suitable candidate for frequency hopping is deemed to have been found.

This field is only considered if the criterion defined by MINNLTDDIFF is met first (see "MINNLTDDIFF Field").

Range: 0 to 255

HOPST Field

This field specifies the Hop Suspend Time. This is the time period from when the last frequency hop occurred to when a new frequency hop is allowed. The Hop Suspend Time is specified in units of 10 acquisition cycles. A value of zero effectively results in no hop timing restrictions.

Range: 0 to 255

STABCTRL, NOISCTRL and VNOICTRL Fields

These fields are used to configure specific operation settings for the appropriate noise states.

AUTO (NOISCTRL and VNOICTRL only): Controls the transitions into and out of the NOISY/VERY_NOISY state. Specifically, this bit selects whether transitions between the states can be made based on measured noise levels. If this bit is set to 1, the transitions are influenced by the detected noise levels. If this bit is set to 0, the detected noise levels are ignored. In the case of the VERY_NOISY state (VNOICTRL field), this bit effectively enables the use of the VERY_NOISY state.

When this bit is set to 1, the transition from the STABLE to NOISY state is made using the following condition:

- Detected noise ≥ STABHINLTDTHR

Similarly, the transition from the NOISY to VERY_NOISY state is made using the following condition:

- Detected noise ≥ NOISHINLTDTHR

The transition down from the VERY_NOISY to the NOISY state occurs if:

- Detected noise ≤ VNOILONLTDTHR for a given number of cycles.

  The number of cycles is determined by LOWNOISTCH, if a touch has been detected, and by NOTCH, if no touch has been detected.

Similarly, the transition down from the NOISY to the STABLE state occurs if:

- Detected noise ≤ NOISLONLTDTHR for a given number of cycles.

  The number of cycles is determined by LOWNOISTCH, if a touch has been detected, and by NOTCH, if no touch has been detected.

Note: LOWNOISTCH and NOTCH in the above are in the NOISCNT and VNOICNT fields.

CAL: If this bit is set to 1, it forces a calibration on entry to the state.

STABFREQ[], NOISFREQ[] and VNOIFREQ[] Fields

These fields specify the set of five candidate frequencies that the noise suppression algorithm can select when operating in a particular state.

The number of candidate frequencies in use is determined by NUMPRESETS in "CFG3 Field". If NUMPRESETS is less than 5, then the latter presets within the arrays are ignored.

Range: 0 to 255

STABTCHAPX[], NOISTCHAPX[] and VNOITCHAPX[] Fields

These fields specify the set of five candidate ADCs per X settings that the noise suppression algorithm can select when operating in the appropriate noise states when touch has been detected (see Section 5.14.1 "Introduction").

These candidate ADCs per X settings may be used instead of the configured Active Syncs per Line setting as determined by the Self Capacitance Configuration T111 object (see Section 6.16.1 "Introduction" for more information).

The number of candidate ADCs per X settings in use is determined by NUMPRESETS in "CFG3 Field". If NUMPRESETS is less than 5, then the latter presets within the arrays are ignored.

Range: 0 (8), 1 to 255

STABNOTCHAPX[], NOISNOTCHAPX[] and VNOINOTCHAPX[] Fields

These fields specify the set of five candidate ADCs per X settings that the noise suppression algorithm can select when operating in the appropriate noise states when no touch has been detected (see Section 5.14.1 "Introduction").

These candidate ADCs per X settings may be used instead of the configured Idle Syncs per Line setting as determined by the Self Capacitance Configuration T111 object (see Section 6.16.1 "Introduction" for more information).

The number of candidate no touch ADCs per X settings in use is determined by NUMPRESETS in "CFG3 Field". If NUMPRESETS is less than 5, then the latter presets within the arrays are ignored.

Range: 0 (8), 1 to 255

NOISLONLTDTHR and VNOILONLTDTHR Fields

These fields define the noise threshold (in 8-bit deltas) below which the noise is identified as too low for the current state. When the noise level is less than VNOILONLTDTHR, the noise suppression algorithm enters NOISY state and when the noise level is less than NOISLONLTDTHR, the noise suppression algorithm enters STABLE state.

The range for these fields is 0 to 255

Range: 0 to 255

STABHINLTDTHR and NOISHINLTDTHR Fields

These fields define the noise threshold (in 8-bit deltas) above which the noise is identified as too high for the current state. When the noise level is greater or equal to STABHINLTDTHR, the noise suppression algorithm enters the NOISY state and when the noise level is greater or equal to NOISHINLTDTHR, the noise suppression algorithm enters the VERY_NOISY state.

Under NDA
© 2021 Microchip Technology Inc. and its subsidiaries

The range for these fields is 0 to 255, where a value of 0 indicates that this threshold is disabled and therefore cannot impose a noise state promotion accordingly.

Range: 0 (no threshold applied), 1 to 255

NOISCNT and VNOICNT Fields

These fields provide real-time counters that are used in the NOISY and VERY_NOISY states.

NOTCH: Specifies the number of “no touch” cycles used to identify a low noise condition. This value is specified in units of 100 cycles, where a value of 0 means 1 (100 cycles).

NOTCH Range: 0 (1 = 100 cycles), 1 to 15

LOWNOISTCH: Specifies the number of “low noise touched” cycles used to identify a low noise condition. This field is specified in units of 10 cycles.

LOWNOISTCH Range: 0 (do not wait for “low noise touched”), 1 to 15

BLKNLTDTHR Field

This field specifies the Block NLTD Threshold. This is the threshold (in 8-bit deltas) at which measured noise is deemed excessively high, resulting in the blocking of touch processing. The default value of 0 indicates that no blocking threshold is applied.

Range: 0 (no blocking threshold applied), 1 to 255

5.14.3 CONFIGURATION CHECKS

The Self Capacitance Noise Suppression T108 object causes a configuration check to be performed in the following circumstances:

- When certain fields are changed

In addition, some fields will cause an automatic recalibration to be performed.

A configuration check may determine that a configuration error has occurred (for example, if a setting is set outside of its allowed range or a conflict has occurred between two settings). This is signaled to the host (see Section 3.3 “Command Processor T6 Object”). The device halts until the error has been corrected. To fix the error, check that all the object settings are within their allowed limits, as stated in the field descriptions.

Note that while the object is disabled, potential configuration errors are allowed.

TABLE 5-44: CONFIGURATION CHECKS FOR SELF CAPACITANCE NOISE SUPPRESSION T108 (PROCG_NOISESUPSELFCAP_T108)

Field	Changing The Field Causes...		Effect of Configuration Checks On Field
	Configuration Check	Automatic Recalibration	
CTRL	Yes (1)	Yes (1)	None
CALCFG1	Yes	Yes	May cause a noise state transition, so filters may require resetting.
CFG1	Yes	No	May cause a noise state transition, so filters may require resetting.
CFG2	Yes	No	May cause a noise state transition, so filters may require resetting.
CFG3	Yes	No	May cause a noise state transition, so filters may require resetting.
NLGAIN	Yes	Yes	Affects noise acquisition measurements, so filters will require resetting
IIRCOEFF	Yes	No	Filters require resetting
MINNLTDDIFF	No	No	None
MINNLTDHOP	No	No	None
HOPST	Yes	No	None

TABLE 5-44: CONFIGURATION CHECKS FOR SELF CAPACITANCE NOISE SUPPRESSION T108 (PROCG_NOISESUPSELFCAP_T108) (CONTINUED)

Field	Changing The Field Causes...		Effect of Configuration Checks On Field
	Configuration Check	Automatic Recalibration	
STABCTRL	Yes	No	May cause a noise state transition, so filters may require resetting.
STABFREQ[]	No	No	None
STABTCHAPX[]	No	No	None
STABNOTCHAPX[]	No	No	None
STABHINLTDTHR	No	No	None
NOISCTRL	Yes	No	May cause a noise state transition, so filters may require resetting.
NOISFREQ[]	No	No	None
NOISTCHAPX[]	No	No	None
NOISNOTCHAPX[]	No	No	None
NOISLONLTDTHR	No	No	None
NOISHINLTDTHR	No	No	None
NOISCNT	No	No	None
VNOICTRL	Yes	No	May cause a noise state transition, so filters may require resetting.
VNOIFREQ[]	No	No	None
VNOITCHAPX[]	No	No	None
VNOINOTCHAPX[]	No	No	None
VNOILONLTDTHR	No	No	None
VNOICNT	No	No	None
BLKNLTDTHR	No	No	None

5.14.4 MESSAGES

The message data for the Self Capacitance Noise Suppression T108 object is shown in Table 5-45.

TABLE 5-45: MESSAGE DATA FOR SELF CAPACITANCE NOISE SUPPRESSION T108 (PROCG_NOISESUPSELFCAP_T108)

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	STATUS1	Reserved			LVLCHG	STATECHG	ACQIDXCHG	TCH	BLKTCH
2	STATUS2	ACQIDX			Reserved			STATE	
3	SUSHOPS	The number of suspended hops since last change							
4	PKNLTDLVL	The last measured NLTD noise level							
5	Reserved	Reserved							
6	PKNLTDDIFF	The peak measured filter range for NLTD based frequency hopping algorithm							
7	PKNLTDHOP	The peak measured quieter-hop filter change for NLTD based frequency hopping algorithm							

STATUS1 Field

This field reports the current status of the noise suppression algorithm.

BLKTCH: Indicates noise suppression algorithm has generated a request to block touch processing. This can occur under excessive measured noise-levels (as configured using the "BLKNLTDTHR Field") or briefly when noise state promotion occurs (if DISBLKTCH is set to 0; see "CFG2 Field").

TCH: Indicates that a touch is present. This bit is set to 1 if a touch is present, and is set to 0 if no touch is present.

Under NDA
© 2021 Microchip Technology Inc. and its subsidiaries

ACQIDXCHG: Indicates that the noise suppression algorithm has changed the number of ADCs per X at least once since the previous message (that is, there has been a frequency hop). This bit is set to 1 if the number of ADCs per X has changed, and 0 otherwise.

The index of the frequency preset being used is reflected in the STATUS2 ACQIDX bit (see "STATUS2 Field").

STATECHG: Indicates that the noise suppression algorithm has changed its state at least once since the previous message. This bit is set to 1 if the state has changed, and 0 otherwise.

The details of the state change are reported in the STATUS2 message field (see "STATUS2 Field").

LVLCHG: Indicates that the measured noise level has changed at least once since the previous message. This bit is set to 1 if the measured noise level has changed, and 0 otherwise.

The peak measured NLTD noise level is reported in the PKNLTDLVL message field (see "PKNLTDLVL Field").

STATUS2 Field

This field reports the current status of the noise suppression algorithm.

STATE: Indicates the current noise suppression algorithm state selected (see Table 5-46).

TABLE 5-46: NOISE SUPPRESSION ALGORITHM STATES

Value	State
0	STABLE
1	NOISY
2	VERY_NOISY
3	Reserved

ACQINDEX: Indicates the index of the current ADCs per X settings and frequency settings in use (that is, an index into the ADCs per X settings and frequency byte arrays between bytes 14 and 28, 34 and 48, and 54 and 68; see Table 5-43 on page 166).

SUSHOPS Field

This field reports the number of suspended hops since the last change.

This field indicates the number of instances when a quieter preset was determined, but ignored because of a requirement to suspend frequency-hopping, as mandated by the HOPST configuration field (see "HOPST Field").

This gives a useful indication of the variance in the measured noise between hops.

PKNLTDLVL Field

This field reports the peak measured NLTD noise level, since the last message was dispatched.

This noise level is the value of measured noise that is assessed against:

- STABHINLTDTHR and NOISHINLTDTHR, to invoke noise-state promotion
- NOISLONLTDTHR and VNOILONLTDTHR to invoke noise-state demotion
- BLKNLTDTHR to invoke blocking of touch processing under high noise levels

PKNLTDDIFF Field

This field reports the peak filter range limit for NLTD noise measurements, since the last message was dispatched.

This range is the largest measured difference between IIR filters, and is assessed against:

- MINNLTDDIFF to assess candidate presets and invoke frequency hopping.

PKNLTDHOP Field

This field reports the peak measured quieter IIR filter difference for the NLTD noise measurements, since the last message was dispatched.

This range is the largest measured difference between the IIR filter associated with the current presets being used for touch measurements and the IIR filter with the lowest noise level, and is assessed against:

- MINNLTDHOP to assess the candidate preset and invoke frequency hopping.

5.15 Self Capacitance Grip Suppression T112 Object

5.15.1 INTRODUCTION

The Self Capacitance Grip Suppression T112 object allows touches to be reported from the self capacitance measurements while at the same time rejecting accidental touches around the edge of the screen, such as from the user's gripping hand on a hand-held device.

When the user's hand accidentally touches the edge of the touchscreen, it can influence the self capacitance measurements. The influence of the accidental touch can exceed that of a small contact area touch, due to the large area in close proximity to the sensor. The Self Capacitance Grip Suppression T112 object removes the influence of the user's hand, allowing the continued, and correct, reporting of contacting touches.

5.15.1.1 Shape Thresholds

For self capacitance touch grip suppression, the shape thresholds are used to distinguish between a grip and a touch further into the sensor. Smaller numbers are more aggressive as a touch will produce a larger value the closer it is to the edge of the sensor. A touch near the edge of the sensor will typically produce a similar value to a grip.

5.15.2 CONFIGURATION

NOTE In the description that follows, the terms X axis and Y axis refer to the physical axes of the device. The terms High Edge and Low Edge refer to the edges of the sensor with the highest and lowest numbered X or Y lines respectively.

TABLE 5-47: CONFIGURATION FOR SELF CAPACITANCE GRIP SUPPRESSION T112 (PROCI_SELFCAPGRIPSUPPRESSION_T112)

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	CTRL	DBGEN	Reserved					RPTEN	ENABLE
1	CHGSHPTHRX	Grip shape change threshold for X axis							
2	CHGSHPTHRY	Grip shape change threshold for Y axis							
3	APPRSHPTHRX	Approaching grip shape threshold for X axis							
4	APPRSHPTHRY	Approaching grip shape threshold for Y axis							

CTRL Field

ENABLE: Enables the use of this object. The object is enabled if set to 1, and disabled if set to 0.

RPTEN: Allows the object to send status messages to the host through the Message Processor T5 object. Reporting is enabled if set to 1, and disabled if set to 0.

DBGEN: If this bit is set to 1, debug data for this object is output using the hardware interface. Note that the Command Processor T6 DEBUGCTRL2 DBGOBJMODEEN bit must be set to 1 for this to work (see "DEBUGCTRL2 Field").

CHGSHPTHRX Field

This field configures the X axis Grip Shape Change Threshold. This is the shape threshold used to suppress a grip change at the edge of the X axis.

Values are in the range 0 and 32 to 255, where 0 means disable grip suppression in the X axis.

See Section 5.15.1.1 "Shape Thresholds" for further information on shape thresholds.

Range: 0 (disabled), 32 to 255

Typical: 56

CHGSHPTHRY Field

This field configures the Y axis Grip Shape Change Threshold. This is the shape threshold used to suppress a grip change at the edge of the Y axis.

Values are in the range 0 and 32 to 255, where 0 means disable grip suppression in the Y axis.

Under NDA
© 2021 Microchip Technology Inc. and its subsidiaries

See Section 5.15.1.1 “Shape Thresholds” for further information on shape thresholds.

Range: 0 (disabled), 32 to 255

Typical: 56

APPRSHPTHRX Field

This field configures the X axis Approaching Grip Shape Threshold. This is the shape threshold used to detect an approaching grip change at the edge of the X axis. This value should be smaller than CHGSHPTHRX.

Values are in the range 0 and 32 to 255, where the default value of 0 means use the following:

32 + 0.5 × (CHGSHPTHRX – 32)

See Section 5.15.1.1 “Shape Thresholds” for further information on shape thresholds.

Range: 0 (default as above), 32 to 255

Typical: 40

APPRSHPTHRY Field

This field configures the Y axis Approaching Grip Shape Threshold. This is the shape threshold used to detect an approaching grip change at the edge of Y axis. This value should be smaller than CHGSHPTHRY.

Values are in the range 0 and 32 to 255, where the default value of 0 means use the following:

32 + 0.5 × (CHGSHPTHRY – 32)

See Section 5.15.1.1 “Shape Thresholds” for further information on shape thresholds.

Range: 0 (default as above), 32 to 255

Typical: 40

5.15.3 CONFIGURATION CHECKS

The Self Capacitance Grip Suppression T112 object causes a configuration check to be performed in the following circumstances:

- When the object is enabled (that is, the ENABLE bit is set to 1 in the CTRL field)
- If the object is enabled, when certain fields are changed

A configuration check may determine that a configuration error has occurred (for example, if a setting is set outside of its allowed range or a conflict has occurred between two settings). This is signaled to the host (see Section 3.3 “Command Processor T6 Object”). The device halts until the error has been corrected. To fix the error, check that all the object settings are within their allowed limits, as stated in the field descriptions.

TABLE 5-48: CONFIGURATION CHECKS FOR SELF CAPACITANCE GRIP SUPPRESSION T112 (PROCI_SELFCAPGRIPSUPPRESSION_T112)

Field	Changing The Field Causes...		Effect of Configuration Checks On Field
	Configuration Check	Automatic Recalibration	
CTRL	Yes (1)	No	None
CHGSHPTHRX	No	No	None
CHGSHPTHRY	No	No	None
APPRSHPTHRX	No	No	None
APPRSHPTHRY	No	No	None

Note 1: If the ENABLE bit is toggled on or off.

5.15.4 MESSAGES

The messages produced by the Self Capacitance Grip Suppression T112 object are intended to provide real time information to aid with object configuration and are not intended to be used at run time.

TABLE 5-49: MESSAGE DATA FOR SELF CAPACITANCE GRIP SUPPRESSION T112 (PROCI_SELFCAPGRIPSUPPRESSION_T112)

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	STATUS	Reserved					CONFMODE	APPRSUP	CHGSUP
2	GRIPDETECT	APPRYHI	APPRYLO	APPRXHI	APPRXLO	CHGYHI	CHGYLO	CHGXHI	CHGXLO

STATUS Field

CHGSUP: Indicates that a grip change is being suppressed.

APPRSUP: Indicates that an approaching grip is being suppressed.

CONFMODE: Indicates that a grip has been detected and the device is evaluating whether to permanently suppress the grip.

GRIPDETECT Field

This field indicates when a Grip Change or an Approaching Grip has been detected.

CHGXLO: Set to 1 if a grip change has been detected on the X Axis, Low Edge.

CHGXHI: Set to 1 if a grip change has been detected on the X Axis, High Edge.

CHGYLO: Set to 1 if a grip change has been detected on the Y Axis, Low Edge.

CHGYHI: Set to 1 if a grip change has been detected on the Y Axis, High Edge.

APPRXLO: Set to 1 if an approaching grip has been detected on the X Axis, Low Edge.

APPRXHI: Set to 1 if an approaching grip has been detected on the X Axis, High Edge.

APPRYLO: Set to 1 if an approaching grip has been detected on the Y Axis, Low Edge.

APPRYHI: Set to 1 if an approaching grip has been detected on the Y Axis, High Edge.

Under NDA
© 2021 Microchip Technology Inc. and its subsidiaries

5.16 Ignore Nodes T141 Object

5.16.1 INTRODUCTION

The Ignore Nodes T141 object allows for a defined set of sensor nodes to be excluded from various processes as determined by the Ignore Nodes Controller T145 object. This may be useful when parts of the touchscreen are not enabled for touch, such as with a free-form (non rectangular) screen shape, or when there are mechanical widgets present on the touchscreen.

There are currently two processes that Ignore Nodes T141 can be applied to:

- Touch Processing
- Self Test Control T10 Signal Limit Test

See Ignore Nodes Controller T145 "CTRL Field" for more details.

Each instance of the Ignore Nodes T141 object relates to a single X line and provides a node mask that indicates which nodes on that X line should be ignored. The object instance number therefore corresponds to the X line index on which the mask is applied (that is, Instance 0 relates to X0, Instance 1 relates to X1, and so on).

On the mXT640UD-CCUBHA1 Family there are therefore 32 instances of the Ignore Nodes T141 object, one for each X line (X0 to X31). Note that if less than 32 X lines are configured for use, some instances will be unused.

5.16.2 CONFIGURATION

CONFIGURATION FOR IGNORE NODES T141 (PROCG_IGNORENODES_T141)

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	YMASK	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0
1		Y15	Y14	Y13	Y12	Y11	Y10	Y9	Y8
2		Reserved				Y19	Y18	Y17	Y16

YMASK Field

This field contains the Y line Mask. This controls which Y lines on a given X line (that is, which nodes) are to be ignored in processes that are controlled by the Ignore Nodes Controller T145 object. There are 20 bits in YMASK, one bit for each Y line (Y0 to Y19). Note that if less than 20 Y lines are configured for use, some bits in YMASK will be unused and should be treated as reserved bits.

For example, if Ignore Nodes T141 Instance 4 (that is, the instance representing X4) has YMASK Y10 set to 1, then it indicates that the node at (X4, Y10) is to be ignored in processes that are controlled by the Ignore Nodes Controller T145 object.

If a YMASK bit is set to 0, then normal processing will be applied to data from any processes at the corresponding node.

Yn: Indicates whether the corresponding ultrawide touchscreen node on the X line for this Ignore Nodes T141 instance is to be ignored or not.

6.0 SUPPORT OBJECTS

6.1 Introduction

Support objects provide additional functionality on the device. Table 6-1 lists the support objects on the mXT640UD-CCUBHA1 Family.

TABLE 6-1: SUPPORT OBJECTS

Object	Description
Self Test Control T10 (SPT_SELFTESTCONTROL_T10)	Controls the self-test routines to find faults on the device. See Section 6.2 “Self Test Control T10 Object”.
Self Test Pin Faults T11 (SPT_SELFTESTPINFAULT_T11)	Specifies the configuration settings for the Pin Fault self tests. See Section 6.3 “Self Test Pin Faults T11 Object”.
Self Test Signal Limits T12 (SPT_SELFTESTSIGLIMIT_T12)	Specifies the configuration settings for the Signal Limit self tests. See Section 6.4 “Self Test Signal Limits T12 Object”.
Communications Configuration T18 (SPT_COMMSCONFIG_T18)	Configures additional communications behavior for the device. See Section 6.5 “Communications Configuration T18 Object”.
GPIO Configuration T19 (SPT_GPIOPWM_T19)	Allows the host controller to configure and use the general purpose I/O pins on the device. See Section 6.6 “GPIO Configuration T19 Object”.
User Data T38 (SPT_USERDATA_T38)	Provides a data storage area for user data. See Section 6.7 “User Data T38 Object”.
CTE Configuration T46 (SPT_CTECONFIG_T46)	Controls the capacitive touch engine for the device. See Section 6.8 “CTE Configuration T46 Object”.
Timer T61 (SPT_TIMER_T61)	Provides control of a timer. See Section 6.9 “Timer T61 Object”.
Dynamic Configuration Controller T70 (SPT_DYNAMICCONFIGURATIONCONTROLLER_T70)	Allows rules to be defined that respond to system events. See Section 6.10 “Dynamic Configuration Controller T70 Object”.
Dynamic Configuration Container T71 (SPT_DYNAMICCONFIGURATIONCONTAINER_T71)	Allows the storage of user configuration on the device that can be selected at runtime based on rules defined in the Dynamic Configuration Controller T70 object. See Section 6.11 “Dynamic Configuration Container T71 Object”.
Touch Event Trigger T79 (SPT_TOUCHEVENTTRIGGER_T79)	Configures touch triggers for use with the event handler. See Section 6.12 “Touch Event Trigger T79 Object”.
Auxiliary Touch Configuration T104 (SPT_AUXTOUCHCONFIG_T104)	Allows the setting of self capacitance gain and thresholds for a particular measurement to generate auxiliary touch data for use by other objects. See Section 6.13 “Auxiliary Touch Configuration T104 Object”.
Self Capacitance Global Configuration T109 (SPT_SELFCAPGLOBALCONFIG_T109)	Provides configuration for self capacitance measurements employed on the device. See Section 6.14 “Self Capacitance Global Configuration T109 Object”.
Self Capacitance Tuning Parameters T110 (SPT_SELFCAPTUNINGPARAMS_T110)	Provides configuration space for a generic set of settings for self capacitance measurements. See Section 6.15 “Self Capacitance Tuning Parameters T110 Object”.
Self Capacitance Configuration T111 (SPT_SELFCAPCONFIG_T111)	Provides configuration for self capacitance measurements employed on the device. See Section 6.16 “Self Capacitance Configuration T111 Object”.
Self Capacitance Measurement Configuration T113 (SPT_SELFCAPMEASURECONFIG_T113)	Configures self capacitance measurements to generate data for use by other objects. See Section 6.17 “Self Capacitance Measurement Configuration T113 Object”.
Data Container T117 (SPT_DATACONTAINER_T117)	Provides a mechanism for retrieving specific data held in the device's internal memory. See Section 6.18 “Data Container T117 Object”.
Data Container Controller T118 (SPT_DATACONTAINERCTRL_T118)	Provides direct access to internal data in memory for use with the Data Container T117 objects. See Section 6.19 “Data Container Controller T118 Object”.

Under NDA

© 2021 Microchip Technology Inc. and its subsidiaries

TABLE 6-1: SUPPORT OBJECTS (CONTINUED)

Object	Description
Self Capacitance Voltage Modulation T133 (SPT_SELFCAPVOLTAGEMOD_T133)	Controls the voltage modulation on self capacitance scans. See Section 6.20 “Self Capacitance Voltage Modulation T133 Object”.
Message Count T144 (SPT_MESSAGECOUNT_T144)	Provides a count of pending messages. See Section 6.21 “Message Count T144 Object”.
Ignore Nodes Controller T145 (SPT_IGNORENODESCONTROL_T145)	Specifies how ignored nodes configured in Ignore Nodes T141 are applied to various measurement processes on the device. See Section 6.22 “Ignore Nodes Controller T145 Object”.

6.2 Self Test Control T10 Object

6.2.1 INTRODUCTION

The Self Test Control T10 object controls the self-test routines in the device to find faults in the device.

The following objects are associated with the Self Test Control T10 object to provide any configuration data that is used by the tests:

- Self Test Pin Faults T11 – provides configuration data for the Pin Fault tests (see Section 6.3 “Self Test Pin Faults T11 Object”)
- Self Test Signal Limits T12 – provides configuration data for the Signal Limit tests (see Section 6.4 “Self Test Signal Limits T12 Object”)

The Self Test Control T10 object controls the following types of tests:

- **Pre-Operation Self Test (POST)** – A set of safety-related tests run at start-up. See Section 6.2.1.1 “Pre-Operation Self Test (POST)” for details.
- **Built-In Self Test (BIST)** – A set of safety-related tests run in the background when time permits. See Section 6.2.1.2 “Built-In Self Test (BIST)” for details.
- **On Demand** – Some tests can be run on demand using the CMD field (see “CMD Field”) and the results reported immediately to the host. See Section 6.2.1.3 “On Demand Tests” for details.

NOTE	A GPIO output can be used to indicate a successful BIST test in addition to Self Test Control T10 messages. This provides heartbeat (“alive”) functionality and is toggled at the same frequency as the BIST message. A suitable GPIO pin must be set up as an output (see GPIO Configuration T19 “TRIGGER[] Fields”). In order to comply with IEC60730 Class B requirements, the host must provide suitable verification that the chosen GPIO output is working correctly.

6.2.1.1 Pre-Operation Self Test (POST)

Pre-Operation Self Test (POST) testing can be scheduled to run at system start up. See Section 6.2.1.4 “Self Test Groups” for a list of available tests.

Most of the tests are completed before the reset message is sent to the host and touch processing starts, but some tests require touch processing to be running before they can be carried out. These tests will be completed as soon as possible once touch processing is running.

Once all the POST tests have been finished, the Self Test Control T10 object sends a message to the host to indicate that all the POST tests have completed successfully. If a failure occurs during a test, this will be reported immediately to the host through a Self Test Control T10 test failure message. Some critical failures, however, cannot trigger Self Test Control T10 test failure message, as the device is unable to function correctly (for example, a CPU or memory failure). In these cases, the firmware will reset the device and the resulting communications failure will indicate to the host that a failure has occurred.

NOTE	If the Multiple Touch Touchscreen T100 object is disabled during start-up, it may result in some of the dependent POST tests remaining stuck. It is recommended, therefore, to ensure that the Multiple Touch Touchscreen T100 object is always enabled.

6.2.1.2 Built-In Self Test (BIST)

Built-In Self Test (BIST) testing is scheduled to run in the background at regular intervals when time permits. See Section 6.2.1.4 “Self Test Groups” for a list of available tests.

BIST testing does not start until POST testing is complete, and is temporarily suspended while an on-demand test request is serviced. A new cycle of BIST testing is scheduled regularly at the rate defined by TESTINTERVAL (see “TESTINTERVAL Field”) and will run in the time available after touch and other higher priority processing.

If there are no failures, a message is sent to the host to indicate that all the BIST tests in the current test cycle have passed and that the product is working correctly. This success message is sent at the rate defined by RPTINTERVAL (see “RPTINTERVAL Field”) while there are no test failures. If an "alive" output has also been configured in the GPIO Configuration T19 object, then the output will also change state (toggle) each time the BIST Tests Passed message is sent.

Under NDA
© 2021 Microchip Technology Inc. and its subsidiaries

If a failure is detected during the tests, the BIST tests passed message will stop and will not start again until a BIST test cycle has completed without any failures.

If a BIST test cycle has not completed within the time defined by OVERRUNTIME (see "OVERRUNTIME Field"), then a BIST Overrun message is sent to the host. Note that all the tests are still run to completion and will report failures or the BIST tests passed message, as appropriate.

Critical test failures will result in the device being reset. Other failures will each be reported to the host through a Self Test Control T10 test failure message, which is sent as soon as the failure is detected. For a permanent failure, the message will continue to be reported every BIST test cycle, and therefore repeated at the rate defined by TESTINTERVAL.

The time taken to run the BIST tests will depend on how much time is available after high priority touch processing has run compared with the cycle time defined by Power Configuration T7 IDLEACQINT and ACTVACQINT (see "IDLEACQINT and ACTVACQINT Fields"). BIST processing will fill any time remaining in the T7 cycle period after the high priority touch processing has finished.

Each time the BIST processing runs, it will run for a minimum time defined by MINTSTTIMEPERCYC (see "MINTSTTIMEPERCYC Field"). If there is less than the minimum time available before the high priority touch processing expects to run again, then the high priority touch processing will be delayed until the minimum period is complete.

If critical touch processing needs to be run urgently (for example, DI cycles), this takes priority and the minimum BIST time (MINTSTTIMEPERCYC) is ignored, so that no BIST testing is performed during that time slot. However, the minimum can only be ignored for a certain number of consecutive touch processing cycles, as defined by MAXHOLDOFF (see "MAXHOLDOFF Field"). Once the MAXHOLDOFF count has been reached, the minimum BIST time will be applied regardless of the touch processing requirements, and touch processing will be delayed. Note that if MAXHOLDOFF = 0, there is no Hold-off period and the minimum BIST time is applied.

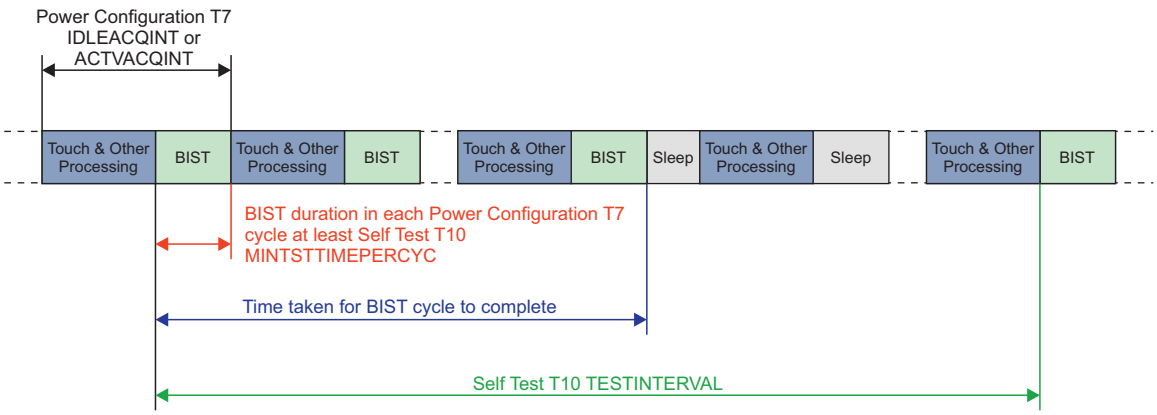
The interplay between the MINTSTTIMEPERCYC and MAXHOLDOFF fields therefore allows the BIST test scheduling to be finely tuned. Recommended initial settings are MINTSTTIMEPERCYC ≥ 40 and MAXHOLDOFF = Multiple Touch Touchscreen T100 TCHDIDOWN + 2.

IMPORTANT!	The MINTSTTIMEPERCYC and MAXHOLDOFF configuration fields need to be carefully tuned. Care must be taken to ensure that sufficient time is allocated to the BIST processing so that the completion of a BIST test cycle can be guaranteed to be within a BIST test interval. Apparent failures such as "keep alive" messages not being generated may be symptoms of the BIST time parameters being set too aggressively. If a Signal Limit or CTE BIST test (only) is unavailable to run, the test is skipped for the current test cycle and the test is retried in the next test cycle.

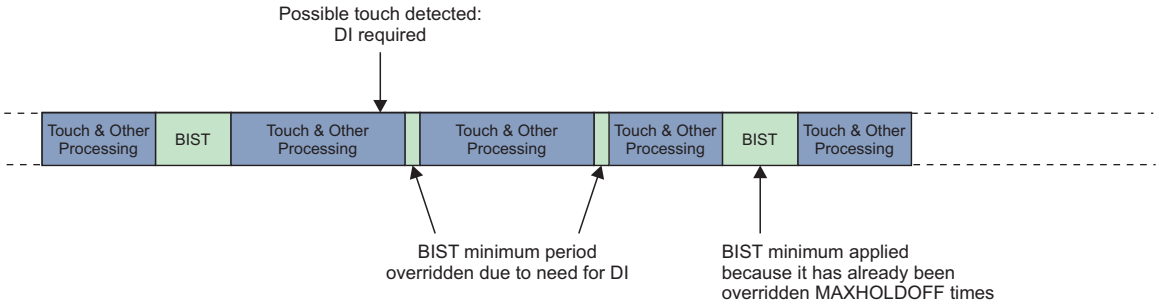
The interleaving of BIST and higher priority processing is shown in Figure 6-1. This shows the scheduling in a normal idle or active touch processing cycle, and the effect of DI cycles and the operation of MAXHOLDOFF.

FIGURE 6-1: BIST PROCESSING SCHEDULING

Normal Touch Processing Cycle



Touch DI with Self Test T10 MAXHOLDOFF = 2



6.2.1.3 On Demand Tests

Some tests can be run on demand and the results reported immediately to the host. These tests can be requested once POST testing has finished and they take priority over touch processing. BIST is suspended while the on-demand test is carried out and resumes again once the on-demand test has finished.

An on-demand test can be carried out by writing the appropriate command value to the CMD field (see "CMD Field").

See Section 6.2.1.4 "Self Test Groups" for details of which tests can be run in this way.

If one or more on-demand tests are run (either individually or by using the "Run all on-demand tests" command; see "CMD Field"), one or more of the following messages will be sent to the host:

- **Test Unavailable** – Sent to the host when the application reports that the requested test is unavailable. If more than one test is unavailable, there will be a message for each unavailable test.
- **Test Failed** – Sent to the host when the application reports that a requested test has failed. Only one such message will be sent: any "Run all on-demand tests" command is terminated as soon as a failure occurs.
- **All Tests Passed** – One message only is sent to the host after all the tests have finished, if at least one test reported passed.

Note that, in the case of the "Run all on-demand tests" command, if a Test Unavailable message is sent, this applies to only one individual test. The other tests will still continue to run and report as required, including further Test Unavailable messages, a Test Failed message, or an All Tests Passed message.

NOTE	The Signal Limit test is the only on-demand test that can be reported as unavailable.

Under NDA
© 2021 Microchip Technology Inc. and its subsidiaries

6.2.1.4 Self Test Groups

Table 6-2 lists the self tests available on the mXT640UD-CCUBHA1 Family.

TABLE 6-2: SELF TEST GROUPS

Self Test Group	Can be Configured to Run as…			How Failure Reported
	Pre-Operation Self Test (POST)	Built-In Self Test (BIST)	On Demand Test	
Self Tests for Internal System Operation				
CPU	Automatically run at start-up	Yes	–	Forced reset
Internal Interrupts	Yes	Yes	–	Forced reset
Clock	Yes	Yes	–	Depends on test. Forced reset or Fail Message (Test Code: 2)
Flash Memory	Yes	Yes	–	Depends on test. Forced reset or Fail Message (Test Code: 3)
RAM	CTE RAM: Yes AVR RAM: Automatically tested at start-up	Yes	–	Depends on test. Forced reset or Fail Message (Test Code: 4)
Power	Yes	Yes	Yes	Depends on test. Forced reset or Fail Message (Test Code: 7) (1)
Self Tests for CTE and Touch System Operation				
CTE (Capacitive Touch Engine)	Yes (2)	Yes (2)	–	Fail Message (Test Code: 5)
Pin Faults	Yes	Yes	Yes	Fail Message (Test Code: 8)
Signal Limits	Yes (2)	Yes (2)	Yes	Fail Message (Test Code: 6)

Note 1: If a power supplies fails during power on, the device may remain in bootloader mode instead.

2: If the Driven Shield Signal Limit test is enabled in Self Test Signal Limits T12, both the CTE and the Signal Limit tests include additional tests on the Driven Shield (DS0 pin). See Section 6.4.1.2 “Driven Shield Signal Limits Test” for more details.

The self tests are grouped into the following test groups, as listed below. The test groups can be run as POST and/or BIST tests. Note, however, that the CPU test group POST processing is compulsory and is always run automatically at start-up. In addition, the Power, Pin Fault and Signal Limit test groups can also be run using direct commands (see “CMD Field”.

SELF TESTS FOR INTERNAL SYSTEM OPERATION

- **CPU** – Tests the CPU registers.
- **Internal Interrupts** – Tests the interrupt latency and handlers in the device.
- **Clock** – Tests the clocks and timers.
- **Flash Memory** – Tests the validity of the data in flash memory.
- **RAM** – Tests the RAM.
- **Power** – Tests that analog (AVdd), XVdd, and digital I/O (VddIO) power supplies are within their expected ranges. It also tests the brown-out (BOD) detectors. The Power tests can be enabled for both POST and BIST processing, if enabled by the appropriate PWREN bit.

  Note that the possibility of receiving a POST power fail message is small because, if there is a power fail at start up, the bootloader will not allow the application (and thereby the POST tests) to run. A POST power failure is reported only if it occurs after the application starts and before the POST power test runs.

If Self Test Control T10 is enabled, and reporting is also enabled, any spurious voltage drop of the power supplies that is detected by the system at runtime will cause the Self Test Control T10 object to generate an automatic "Power Fault" message. For this to happen, however, BIST must be enabled for power tests (BISTCFG PWREN = 1).

> **NOTE** Power Configuration T7 must be configured to allow continuous power monitoring in order for this test to operate (that is, Power Configuration T7 CFG DISPOWMON = 0; see Power Configuration T7 "CFG Field"). Note also that the Power Monitor checks the power supply every cycle, regardless of BIST scheduling.

SELF TESTS FOR CTE AND TOUCH SYSTEM OPERATION

- **CTE** – Tests the Capacitive Touch Engine (CTE).

  If the Driven Shield Signal Limit test is enabled (see Self Test Signal Limits T12 "CTRL Field"), the CTE tests also validate the CTE measurements performed on the DS0 pin. This allows open pin faults on the DS0 pin to be detected by the signal Limit tests (see below).
- **Pin Fault** – Tests the sense pins (X0 to X31 and Y0 to Y19), and the Generic Key pins (GKEYX0 and GKEYY0 to GKEYY2), on the device. This test module also tests for shorts on the driven shield pin (DS0). This allows low-resistance shorts (line-to-line, line-to-power and line-to-GND) to be detected, as well as resistive line-to-line shorts. The pin fault tests are configured using the Self Test Pin Faults T11 object (see Section 6.3 "Self Test Pin Faults T11 Object".
- **Signal Limit** – Tests the signal limits for all measurement types. These test the signals from each of the touch objects on the device. These tests are run for each enabled touch object rather than for the entire touchscreen sensor matrix. Note that Multiple Touch Touchscreen T100 must be enabled for the tests to run successfully. The user must specify the expected minimum and maximum level for each touch object under test. The signal limit tests are configured using the Self Test Signal Limits T12 object (see Section 6.4 "Self Test Signal Limits T12 Object").

  If the Driven Shield Signal Limits test is enabled (see Self Test Signal Limits T12 "CTRL Field"), the Signal Limit tests also test the DS0 pin. This allows open circuit faults on the DS0 pin to be detected.

  Note that Self Capacitance Global Configuration T109 CTRL DISMEASX must also be set to 0 if the Driven Shield Signal Limits test is enabled.

> **NOTE** For free-form (non rectangular) screens, it is possible to use the Ignore Nodes T141 and Ignore Nodes Controller T145 objects to indicate that certain nodes are to be ignored by the signal limit test. See Section 5.16 "Ignore Nodes T141 Object" and Section 6.22 "Ignore Nodes Controller T145 Object" for more details.

6.2.2 CONFIGURATION

TABLE 6-3: CONFIGURATION FOR SELF TEST CONTROL T10 (SPT_SELFTESTCONTROL_T10)

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	CTRL	POSTEN	BISTEN	Reserved				RPTEN	ENABLE
1	CMD	Test command							
2	POSTCFG	PWREN	SIGLIMEN	CTEEN	RAMEN	FLASHEN	CLOCKEN	INTSEN	Reserved
3		Reserved							PINFLTEN
4	BISTCFG	PWREN	SIGLIMEN	CTEEN	RAMEN	FLASHEN	CLOCKEN	INTSEN	CPUEN
5		Reserved							PINFLTEN
6	TESTINTERVAL	BIST interval							
7	MINTSTTIMEPERCYC	Minimum test duration per BIST processing cycle							
8	MAXHOLDOFF	Maximum BIST hold-off count							
9	OVERRUNTIME	BIST overrun							
10	RPTINTERVAL	BIST report interval							

CTRL Field

ENABLE: Enables the use of the Self Test Control T10 object. The object is enabled if set to 1, and disabled if set to 0.

Under NDA

© 2021 Microchip Technology Inc. and its subsidiaries

RPTEN: Allows the object to send status messages to the host through the Message Processor T5 object. Reporting is enabled if set to 1, and disabled if set to 0.

BISTEN: Enables Built-In Self Test (BIST) processing. The specific tests to be run during BIST processing must be enabled using the BISTCFG field (see "POSTCFG and BISTCFG Fields").

POSTEN: Enables Pre-Operation Self Test (POST) processing. The specific tests to be run during POST processing must be enabled using the POSTCFG field (see "POSTCFG and BISTCFG Fields").

CMD Field

This field is used to send test commands to the device. Valid test commands are listed in Table 6-4.

TABLE 6-4: TEST COMMANDS

Command	Description
0x00	No command
0x31	Run the Power test
0X32	Run the Pin Fault test
0x33	Run the Signal Limit test
0x3E	Run all the tests in the order above
All other values	Reserved; no effect

When the CMD field is set to the code of the test to be run, the Self Test Control T10 object runs the test once and then stops. At the end of the test, the CMD field is cleared to zero. If reporting is enabled (see the "CTRL Field" field), the Self Test Control T10 object also sends a report message with the result of the test (see Section 6.2.3 "Messages").

POSTCFG and BISTCFG Fields

These fields enable the various types of test available during POST or BIST testing. A test is enabled if a control is set to 1 and disabled if the control is set to 0.

CPUEN: Enables CPU register BIST tests.

Note that the CPU register POST tests are compulsory and therefore there is no corresponding control to enable or disable it.

INTSEN: Enables Interrupt POST or BIST tests.

CLOCKEN: Enables Clock POST or BIST tests.

FLASHEN: Enables Flash Memory POST or BIST tests.

RAMEN: Enables RAM POST or BIST tests.

CTEEN: Enables CTE POST or BIST tests.

SIGLIMEN: Enables Signal Limit POST or BIST tests.

Note that if the Driven Shield Signal Limit test is enabled (see Self Test Signal Limits T12 "CTRL Field"), the Signal Limit tests also include tests for signal limits on the DS0 pin.

PWREN: Enables Power POST or BIST tests. Power Configuration T7 must be configured to allow continuous power monitoring in order for this test to operate (that is, Power Configuration T7 CFG DISPOWMON = 0; see Power Configuration T7 "CFG Field").

PINFLTEN: Enables Pin Fault POST or BIST tests.

TESTINTERVAL Field

This field specifies the BIST Test Interval. This is the interval at which the BIST test cycles are started. All enabled BIST tests are run in each BIST test cycle.

The interval is specified in units of 50 ms, meaning a maximum time of 12750 ms can be specified. The default value of 0 means start a BIST test cycle every measurement processing cycle (as defined by Power Configuration T7 IDLEACQINT/ACTVACQINT; see "IDLEACQINT and ACTVACQINT Fields").

Range: 0 (every processing cycle), 1 to 255

MINTSTTIMEPERCYC Field

This field specifies the minimum allocated test time (in units of 20 µs) per BIST test cycle. This is the minimum amount of time allocated to run the BIST tests in each normal measurement processing cycle. Where possible, the BIST test processing will make use of any available spare time before other processing is scheduled.

This field should be used in conjunction with the MAXHOLDOFF field. See Section 6.2.1.2 “Built-In Self Test (BIST)” for more information.

The Minimum Cycle Duration is specified in units of 20 µs, allowing for a minimum duration up to 5.1 ms.

Range: 0 to 255

Typical: ≥ 40

MAXHOLDOFF Field

This field specifies the Maximum Hold-off Count for BIST processing. See Section 6.2.1.2 “Built-In Self Test (BIST)” for more information on how this field is used, and its association with the MINTSTTIMEPERCYC field.

The MAXHOLDOFF field needs to be carefully tuned in conjunction with MINTSTTIMEPERCYC to ensure that sufficient time is allocated to the BIST processing. Recommended initial settings are MINTSTTIMEPERCYC ≥ 40 and MAXHOLDOFF = Multiple Touch Touchscreen T100 TCHDIDOWN + 2.

The default value of 0 means no Maximum Hold-off Count, resulting in no Hold-off period.

Range: 0 (no maximum hold off count), 1 to 255

Typical: Multiple Touch Touchscreen T100 TCHDIDOWN + 2

OVERRUNTIME Field

This field defines the BIST Overrun Time. This is the maximum time allowed for a BIST test cycle. If this time is exceeded, a message reporting a BIST cycle overrun is sent to the host.

The BIST Overrun Time is defined in units of 50 ms, thus allowing a maximum time of 12.75 seconds. The default value of 0 means use the time defined for the BIST test interval (see “TESTINTERVAL Field”).

Range: 0 (use TESTINTERVAL), 1 to 255

RPTINTERVAL Field

This field defines the Report Interval. This is the time between sending each BIST test OK message. A GPIO pin can also be toggled (that is, if it is high then set it low, and if it is low then set it high) at the same frequency if a GPIO Configuration T19 GPIO trigger is configured (see GPIO Configuration T19 “TRIGGER[] Fields”).

See Section 6.2.1.2 “Built-In Self Test (BIST)” for a detailed description. The report interval is specified in units of 50 ms, meaning a maximum time of 12.75 seconds can be specified. The default value of 0 means every processing cycle (see Power Configuration T7 “IDLEACQINT and ACTVACQINT Fields”).

Range: 0 (every processing cycle), 1 to 255

6.2.3 MESSAGES

The Self Test Control T10 object reports the success or failure of POST, BIST and on-demand tests in its message data. In addition, if a BIST test cycle has not completed within the specified maximum time, the Self Test Control T10 object sends a BIST test cycle overrun message to the host. See “OVERRUNTIME Field” for configuration details.

Some failures, however, cannot trigger a Message Processor T5 message, as the device is unable to function correctly (for example, a CPU or memory failure). In these cases, the firmware will reset the device and the resulting communications failure will indicate to the host that a failure has occurred.

In addition to regular BIST reporting, a GPIO pin can be configured to send a regular signals to the host to indicate that communications are working. See “TRIGGER[] Fields” for more details.

The message data for the Self Test Control T10 object is shown in Table 6-5.

TABLE 6-5: MESSAGE DATA FOR SELF TEST CONTROL T10 (SPT_SELFTESTCONTROL_T10)

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	STATUS	Result code							
2 – 6	INFO	Result data							

Under NDA
© 2021 Microchip Technology Inc. and its subsidiaries

STATUS Field

This field contains a result code that indicates the success or failure of the test. Valid codes are given in Table 6-6.

TABLE 6-6: STATUS FIELD

Result Code	Meaning	INFO[] Field Data
0x11	All POST tests have completed successfully	Not used; the INFO field consists of reserved bytes only
0x12	A POST test has failed	See Section "INFO Field – Test Failed"
0x21	All BIST tests have completed successfully	Not used; the INFO field consists of reserved bytes only
0x22	A BIST test has failed	See Section "INFO Field – Test Failed"
0x23	BIST test cycle overrun	Not used; the INFO field consists of reserved bytes only
0x31	All on-demand tests have passed	Not used; the INFO field consists of reserved bytes only
0x32	An on demand test has failed	See Section "INFO Field – Test Failed"
0x33	The requested On-demand test is unavailable	See Section "INFO Field – Unavailable On-demand Test"
0x3F	The test code supplied in the CMD field is not associated with a valid test	See Section "INFO Field – Invalid Command Code"

INFO Field

This field contains the result data of the test. The actual data depends on which test was run, as detailed below. If a test does not generate data (see Table 6-6), the INFO field consists solely of reserved bytes.

INFO Field – Unavailable On-demand Test

If the result indicates that an on-demand test is unavailable, the INFO field has the data format shown in Table 6-7.

TABLE 6-7: TEST RESULT DATA FOR AN *Unavailable On-demand Test*

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
2	TEST_GROUP	Test group							

TEST_GROUP Field

This field indicates the test group for which this is the report message.

NOTE	Currently only the Signal Limit test is detected as being unavailable.

Range: See Table 6-8

TABLE 6-8: TEST_GROUP FIELD

Code	Test Group
6	Signal Limit tests
Other	Reserved

INFO Field – Invalid Command Code

If the result was an invalid command code, the INFO field has the data format shown in Table 6-9.

TABLE 6-9: TEST RESULT DATA FOR AN UNKNOWN COMMAND CODE

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
2	CMD	Invalid command code							

CMD Field

This field indicates the invalid command code.

INFO Field – Test Failed

If a POST, BIST or On-demand test fails, the INFO field has the data format shown in Table 6-10.

TABLE 6-10: TEST RESULT DATA FOR A POST/BIST TEST FAILURE

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
2	TEST_GROUP	Test group							
3	DATA[]	Result data							
4		Result data							
5		Result data							
6		Result data							

TEST_GROUP Field

This field indicates the test group for which this is the report message.

Range: See Table 6-8

TABLE 6-11: TEST_GROUP FIELD

Code	Test Group
2	Clock related tests
3	Flash Memory tests
4	RAM memory tests
5	CTE tests
6	Signal Limit tests
7	Power-related tests
8	Pin Fault tests
Other	Reserved

DATA[] Field

This field contains the result data of the test. The actual data depends on which POST/BIST test was run, as detailed below.

DATA Field – Clock Error
DATA Field – Flash Memory Error
DATA Field – RAM Error
DATA Field – CTE Error
DATA Field – Power Error

The DATA field consists of reserved bytes only. Any non-zero data is for use by Microchip engineers only.

DATA Field – Signal Limit Error

If the result was a signal limit error, the DATA field has the data format shown in Table 6-12. This is reported as soon as possible after the test has failed.

TABLE 6-12: TEST RESULT DATA FOR A SIGNAL LIMIT ERROR

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
3	TYPE_NUM	Object type number							
4	TYPE_INSTANCE	Object type instance							
5	Reserved	Reserved							
6	Reserved	Reserved							

TYPE_NUM Field

The type number of the object for which a signal limit error was found (see Section 1.5 “Object Table”).

Under NDA

© 2021 Microchip Technology Inc. and its subsidiaries

TYPE_INSTANCE Field

The instance number of the object for which a signal limit error was found.

For example, if the touchscreen sensor signal limit test fails, TYPE_NUM and TYPE_INSTANCE will indicate Multiple Touch Touchscreen T100 Instance 0. Similarly, if a Driven Shield Signal Limits measurement fails on DS0, TYPE_NUM and TYPE_INSTANCE will indicate Self Test Signal Limits T12 Instance 0.

DATA Field – Pin Fault

If the result was a pin fault, the DATA field has the data format shown in Table 6-13.

NOTE	The Pin Fault test reports the first failed pin detected (that is, the lowest index pin with a fault). Therefore, if more than one pin has failed, subsequent failures will not be reported until the fault on the first detected pin has been resolved.

TABLE 6-13: TEST RESULT DATA FOR A PIN FAULT

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
3	SEQ_NUM	Test sequence number							
4	X_PIN	Failing X pin indication							
5	Y_PIN	Failing Y pin indication							
6	DS_PIN	Failing Driven Shield pin indication							

SEQ_NUM Field

The test sequence number of the test in which a fault is found. The sequence numbers are listed in Table 6-14. See Section 6.3.1.1 “Pin Fault Tests” for full details of the sequence patterns.

TABLE 6-14: SEQUENCE NUMBERS FOR PIN FAULT TEST

Code	Description
0x01	**Driven Ground** Detects shorts to a power rail
0x02	**Driven High** Detects shorts to Ground
0x03	**Walking 1** Uses the pull-up resistors to detect resistive shorts
0x04	**Walking 0** Uses the pull-up resistors to detect resistive shorts

X_PIN Field

This field indicates that a failure on an X sense pin or the GKEYX0 pin has been detected.

The value of X_PIN is 1 to 32, which represents the number of the failed X pin plus 1. For example, if X5 has failed, X_PIN will read 6. If more than one pin fails, the test reports the first failed pin detected.

Failures on the Generic Key X pin is reported following the sense pins. Thus, GKEYX0 is reported as a value of 33.

Y_PIN Field

This field indicates that a failure on a Y sense pin or a GKEYY pin has been detected.

The value of Y_PIN is 1 to 20, which represents the number of the failed Y pin plus 1. For example, if Y3 has failed, Y_PIN will read 4. If more than one pin fails, the test reports the first failed pin detected.

Failures on the Generic Key Y pins are reported following the sense pins. Thus, GKEYY0 is reported as a value of 21, and so on.

DS_PIN Field

This field indicates that a failure has been detected on the driven shield pin. The possible values are shown in Table 6-15.

Range: See Table 6-15

TABLE 6-15: DS_PIN VALUES

Value	Description
0	The driven shield has passed
1	DS0 has failed

Under NDA
© 2021 Microchip Technology Inc. and its subsidiaries

6.3 Self Test Pin Faults T11 Object

6.3.1 INTRODUCTION

The Self Test Pin Faults T11 object specifies the configuration settings for the Pin Fault self tests. Specifically, it determines how the pin fault testing is carried out (see "PINDWELLUS Field" and "PINTHR Field").

This object is used in conjunction with the Self Test Control T10 object, which controls the tests and reports the results (see Section 6.2 "Self Test Control T10 Object").

6.3.1.1 Pin Fault Tests

The Pin Fault self tests perform tests on the following pins:

- All X pins (X0 to X31)
- All Y pins (Y0 to Y19)
- All Generic Key pins (GKEYX0, and GKEYY0 to GKEYY2)
- The driven shield pin (DS0)

NOTE	All the relevant pins on the device are tested, even if they have not been defined for use by any enabled touch object.

The pin fault tests allow low-resistance shorts (line-to-line, line-to-power and line-to-GND) to be detected, as well as resistive line-to-line shorts. The tests are performed by setting the pins high or low in particular sequences, as shown in Table 6-16.

TABLE 6-16: PIN FAULT TEST SEQUENCES

Sequence (Code)	Purpose	Test Operation
Driven Ground (0x01)	Detects shorts to a power rail (that is, they fail high). Note that the detection of X/Y shorts is dependent on the supply voltage levels.	All pins are driven low to Ground.
Driven High (0x02)	Detects shorts to Ground (that is, they fail low). (1)	All pins are pulled high.
Walking 1 (0x03)	Similar to the Driven Ground sequence, but uses the pull-up resistors to detect resistive shorts. (1)	All the pins are set to zero. Then, starting with the first pin, each pin in turn is pulled high (set to 1), leaving all the other pins set to zero. The effect of this operation is that a 1 bit walks along the pins.
Walking 0 (0x04)	Similar to the Driven High sequence, but uses the pull-up resistors to detect resistive shorts. (1)	All the pins are set to 1. Then, starting with the first pin, each pin in turn is cleared (set to 0), leaving all the other pins set to 1. The effect of this operation is that a "0" bit "walks" along the pins.

Note 1: Test can detect resistive shorts, but only up to ~200 kΩ for the X/Y sense pins, generic key pins, and driven shield pin.

6.3.2 CONFIGURATION

TABLE 6-17: CONFIGURATION FOR SELF TEST PIN FAULTS T11 (SPT_SELFTESTPINFAULT_T11)

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Reserved	Reserved							
1	Reserved	Reserved							
2	PINDWELLUS	Pin dwell time							
3	PINTHR	Pin threshold							

PINDWELLUS Field

This field configures the pin dwell time (in µs) for the pin fault test, where 0 means use a special case of 4 × Acquisition Configuration T8 CHRGTIME (in µs). The value for this field is determined using the Microchip Sensor Design Calculator (contact your Microchip representative for more information).

If the pin fault test is enabled for POST operation (see Self Test Control T10 "POSTCFG and BISTCFG Fields"), particular care should be taken to ensure that the value of this field is set to allow enough time for the X drive lines to slew to the correct level before testing.

Range: 0 (4 x Acquisition Configuration T8 CHRGTIME), 1 to 254

PINTHR Field

This field specifies the Pin Thresholds used to detect shorts in the pin fault tests. Increasing the value of this threshold increases the resistance that will be detected as a short.

An exact value of PINTHR can be selected using the following formula:

$$PINTHR = \text{floor}\left(\left(\frac{R_{THR}}{R_P + R_{THR}} - 0.5\right) \times 512\right) + 1$$

Where:

The typical value for R_P is 16×10^3

The worst case value for R_P is 13×10^3

R_{THR} is the resistance below which it is desired that a short is guaranteed to be detected.

The default value of 225 equates to a threshold of 200 kΩ, assuming the worse case value for R_P, and this is the recommended level. Attempting to detect short resistances larger than 200 kΩ is not recommended unless advised by Microchip.

In order to assist in tuning of this threshold, tuning data is provided by the Diagnostic Debug T37 and Command Processor T6 objects (see Section 2.2 "Diagnostic Debug T37 Object" and Section 3.3 "Command Processor T6 Object"). The tuning output provides minimum and maximum pin values detected during the last run of the pin fault test. Pin values greater than or equal to PINTHR result in a pass.

Range: 0 (225 = 200 kΩ), 1 to 255

6.3.3 MESSAGES

The results of the pin fault tests are reported through the Self Test Control T10 object. See Section 6.2.3 "Messages" for details.

6.3.4 DIAGNOSTIC DEBUG DATA

The diagnostic debug data modes for the Self Test Pin Faults T11 object is shown in Table 6-18. The diagnostic debug data can be viewed in the Diagnostic Debug T37 object by using the appropriate Command Processor T6 command, also listed in Table 6-18.

The diagnostic debug modes are described in the following sections.

TABLE 6-18: DIAGNOSTIC DEBUG COMMANDS

T6 Command	Name	Description
0x35	Pin Fault Self Test Mode	The Diagnostic Debug T37 object holds the tuning data for the Self Test Control T10 pin fault tests. See Section 6.3.4.1 "Pin Fault Self Test Mode".

Under NDA
© 2021 Microchip Technology Inc. and its subsidiaries

6.3.4.1 Pin Fault Self Test Mode

Figure 6-2 shows the format of the data in Pin Fault Self Test Mode. This reports the tuning data reported by the Self Test Control T10 object for the pin fault test.

FIGURE 6-2: DIAGNOSTIC DEBUG T37 FIELDS – PIN FAULT SELF TEST MODE

Page	Data Index	Data[]
Page 0	0	Minimum measured pin value for a pin that is expected to be pulled low
	1	Maximum measured pin value for a pin that is expected to be pulled low
	2	Minimum measured pin value for a pin that is expected to be pulled high
	3	Maximum measured pin value for a pin that is expected to be pulled high
	4 – 127	Reserved

Assumes page size = 128

6.4 Self Test Signal Limits T12 Object

6.4.1 INTRODUCTION

The Self Test Signal Limits T12 object specifies the configuration settings for the Signal Limit self tests. Specifically, it determines how the signal limit testing is carried out.

The Signal limit tests test the signals from the touch object associated with this instance of the Self Test Signal Limits T12 object. The user must specify the expected minimum and maximum level for the associated touch object under test, using the SIGLIMUP, SIGLIMLO and SIGLIMRANGE fields.

This object is used in conjunction with the Self Test Control T10 object, which controls the tests and reports the results (see Section 6.2 “Self Test Control T10 Object”). Note that Multiple Touch Touchscreen T100 must be enabled for the tests to run successfully.

On the mXT640UD-CCUBHA1 Family there are 2 instances of the Self Test Signal Limits T12 object, one for each touch object on the device. Note that the Signal Limit self tests are run for each enabled touch object rather than for the entire touchscreen sensor matrix.

TABLE 6-19: SELF TEST SIGNAL LIMITS T12 INSTANCES

Instance	Purpose
Instance 0	Configures Signal Limit self tests for Multiple Touch Touchscreen T100. This instance also configures the Driven Shield Signal Limit tests for DS0.
Instance 1	Configures Signal Limit self tests for Key Array T15.

NOTE For free-form (non rectangular) screens, it is possible to use the Ignore Nodes T141 and Ignore Nodes Controller T145 objects to indicate that certain nodes are to be ignored by the Signal Limit self tests. See Section 5.16 “Ignore Nodes T141 Object” and Section 6.22 “Ignore Nodes Controller T145 Object” for more details.

6.4.1.1 Signal Limit Tests

For a Self Test Signal Limits T12 instance that tests the signal limits, the SIGLIMLO and SIGLIMUP fields specify the lower and upper signal limits for the signal limit tests. When the test is run, the mutual capacitance signals must fall between SIGLIMLO and SIGLIMUP for the test to pass. Any signal values outside this range will generate a signal limit error in the test.

The SIGLIMRANGE field specifies the signal range limit used by the signal limit tests for the associated touch object under test. When the test is run, the difference between the largest and smallest mutual capacitance signal must be less than the value specified in SIGLIMRANGE for the test to pass.

For a touchscreen operating with P2P Mutual Capacitance measurements enabled (see Multiple Touch Touchscreen T100 “CALCFG Field”), it is necessary to specify separate single-ended mutual capacitance measurement gains for the Signal Limit tests. See “GAIN Field” and “DXGAIN Field” for more details.

NOTE The Self Test Signal Limits T12 object should have an appropriate configuration for signal limit tests, especially the gains for signal limit testing, in order to avoid other self test failures occurring even if signal limit testing is not enabled.

Under NDA
© 2021 Microchip Technology Inc. and its subsidiaries

6.4.1.2 Driven Shield Signal Limits Test

If the Driven Shield Signal Limits test is enabled (see "CTRL Field"), the Signal Limit tests also test the signal limits on the DS0 pin. This allows open circuit faults on the DS0 pin to be detected. The DSGAIN field must be set for this test to operate correctly (see "DSGAIN Field").

Note that Self Capacitance Global Configuration T109 CTRL DISMEASX must also be set to 0 if the Driven Shield Signal Limits test is enabled.

6.4.2 CONFIGURATION

TABLE 6-20: CONFIGURATION FOR SELF TEST SIGNAL LIMITS T12 – MULTIPLE TOUCH TOUCHSCREEN T100 INSTANCES (SPT_SELFTESTSIGLIMIT_T12)

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	CTRL	DSTESTEN	Reserved						
1	GAIN	Gain							
2	DXGAIN	Dual X gain							
3	SIGLIMLO	Signal lower limit LSByte							
4		Signal lower limit MSByte							
5	SIGLIMUP	Signal upper limit LSByte							
6		Signal upper limit MSByte							
7	SIGLIMRANGE	Signal range limit LSByte							
8		Signal range limit MSByte							
9	DSGAIN	Driven shield gain							

TABLE 6-21: CONFIGURATION FOR SELF TEST SIGNAL LIMITS T12 – KEY ARRAY T15 INSTANCES (SPT_SELFTESTSIGLIMIT_T12)

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Reserved	Reserved							
1	Reserved	Reserved							
2	Reserved	Reserved							
3	SIGLIMLO	Signal lower limit LSByte							
4		Signal lower limit MSByte							
5	SIGLIMUP	Signal upper limit LSByte							
6		Signal upper limit MSByte							
7	SIGLIMRANGE	Signal range limit LSByte							
8		Signal range limit MSByte							
9	Reserved	Reserved							

CTRL Field

DSTESTEN: Enables the inclusion of the Driven Shield Signal Limits tests for DS0, as part of the standard Signal Limit tests. This control is set to 1 to enable the Driven Shield Signal Limits tests, and set to 0 to disable the tests.

NOTE

- The Auxiliary Touch Configuration T104 object must be enabled for use, and Self Capacitance Global Configuration T109 CTRL DISMEASX must be set to 0 (that is, the X axis measurement must not be disabled), for the Driven Shield Signal Limit tests to operate.
- If the Driven Shield Signal Limits Test is enabled for use (DSTESTEN = 1), low frequency compensation cannot be used and must therefore be disabled by setting Self Capacitance Configuration T111 CTRL LFCOMPEN to 0 (see Self Capacitance Configuration T111 "CTRL Field").

NOTE This control is present on Self Test Signal Limits T12 Instance 0 only, and operates on DS0.

GAIN Field

NOTE This field is used by Self Test Signal Limits T12 Instance 0 only (that is, the instance linked to Multiple Touch Touchscreen T100): this field is not used by the Self Test Signal Limits T12 instance linked to Key Array T15.

This field specifies the Single-ended Mutual Capacitance Measurement Gain for the Signal Limit tests. If P2P Mutual Capacitance measurements are enabled (see Multiple Touch Touchscreen T100 "CALCFG Field"), the Signal Limit tests require the running of additional Single-ended Mutual Capacitance measurements. These measurements will use a different gain setting when running in Single X mode to that of the normal Mutual Capacitance measurements (which use the Multiple Touch Touchscreen T100 Single X gain setting). This field specifies this additional Single X gain setting.

If P2P Mutual Capacitance measurements are not enabled (Multiple Touch Touchscreen T100 CALCFG P2PMUTUALEN = 0), this field should be set to the same value as Multiple Touch Touchscreen T100 GAIN.

Range: 0 to 22

DXGAIN Field

NOTE This field is used by Self Test Signal Limits T12 Instance 0 only (that is, the instance linked to Multiple Touch Touchscreen T100): this field is not used by the Self Test Signal Limits T12 instance linked to Key Array T15.

This field specifies the Single-ended Dual X Mutual Capacitance Measurement Gain for the Signal Limit Test. If P2P Mutual Capacitance measurements are enabled (see Multiple Touch Touchscreen T100 "CALCFG Field"), the Signal Limit Tests require the running of additional Single-ended Mutual Capacitance measurements. These measurements will use a different gain setting when running in Dual X mode to that of the normal Mutual Capacitance measurements (which use the Multiple Touch Touchscreen T100 Dual X gain setting). This field specifies this additional Dual X gain setting.

If P2P Mutual Capacitance measurements are not enabled (Multiple Touch Touchscreen T100 CALCFG P2PMUTUALEN = 0), this field should be set to the same value as Multiple Touch Touchscreen T100 DXGAIN.

A value of 255 in the field turns on auto correction of the gain in Dual X mode. In Dual X two X lines are pulsed at the same time so there will be double the amount of charge present. The purpose of the auto correction is to halve the system gain so it is at the correct level.

Range: 0 to 22, 255 (auto correction)

SIGLIMLO and SIGLIMUP Fields

These fields specify the signal limits used by the signal limit tests for the touch object associated with this instance of the Self Test Signal Limits T12 object (see Section 6.4.1.1 "Signal Limit Tests").

The values of SIGLIMLO and SIGLIMUP should be chosen depending on whether the unit is a design prototype or a production item.

Under NDA

© 2021 Microchip Technology Inc. and its subsidiaries

For production use, SIGLIMLO should never be lower than 1200 and SIGLIMUP should never be higher than 14600. The exact values, however, are determined by analyzing the data from actual sensor samples. Using the full range can cause sensors that have too much signal to pass the test.

Range: See Table 6-22

TABLE 6-22: RECOMMENDED VALUES FOR SIGLIMLO[] AND SIGLIMUP[]

Touch Object	Limits		Recommended Design Values		Recommended Production Values	
	SIGLIMLO	SIGLIMUP	SIGLIMLO	SIGLIMUP	SIGLIMLO	SIGLIMUP
Key Array T15	1200	14600	6800	10200	Determined on a case-by-case basis	
Multiple Touch Touchscreen T100	1200	14600	6800	10200	Determined on a case-by-case basis	

Note: SIGLIMUP must be ≥ SIGLIMLO

SIGLIMRANGE Field

This field specifies the signal range limit used by the signal limit tests for the touch object associated with this instance of the Self Test Signal Limits T12 object (see Section 6.4.1.1 “Signal Limit Tests”).

Range: 1 to 13500

DSGAIN Field

NOTE This field is present on Self Test Signal Limits T12 Instance 0 only, and operates on DS0.

This field configures the Gain for the Driven Shield Signal Limits test measurements for DS0.

Range: 0 to 29, 128 to 157 (Gain × 2 mode)

This field must be tuned such that the driven shield signal limit value remains saturated throughout normal operation. See Figure 2-5 on page 18 for details on how to read the value. Note that when the signal goes above the saturation level (that is, above a value of -16384), the Self Test Control T10 object reports a DS signal limit failure (that is, a Signal Limit Error with a TYPE_INSTANCE of 12; see “DATA Field – Signal Limit Error” on page 188).

6.4.3 MESSAGES

The results of the signal limit tests are reported through the Self Test Control T10 object. See Section 6.2.3 “Messages” for details.

6.5 Communications Configuration T18 Object

6.5.1 INTRODUCTION

The Communications Configuration T18 object specifies additional communications behavior for the device.

On the mXT640UD-CCUBHA1 Family there are 2 instances of the Communications Configuration T18 object (see Table 6-23 for details).

TABLE 6-23: COMMUNICATIONS CONFIGURATION T18 INSTANCES

Instance	Purpose
Instance 0	Primary communications interface Note that selection of this interface, and its address (if appropriate), is configured by hardware and not by fields in the Communications Configuration T18 object. Refer to the relevant device *Datasheet* for details.
Instance 1	Secondary communications interface This instance is configured by the CFG field.

Refer to the relevant device *Datasheet* for more information on the primary and secondary interfaces.

6.5.2 CONFIGURATION

6.5.2.1 Instance 0 – Primary Communications Interface

TABLE 6-24: CONFIGURATION FOR COMMUNICATIONS CONFIGURATION T18 (SPT_COMMSCONFIG_T18)

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	CTRL	Reserved	RETRIGEN	Reserved			MODE	Reserved	
1	COMMAND	$\overline{\text{CHG_I2C}}$ line command code							
2	Reserved	Reserved							
3	Reserved	Reserved							
4	DISRPTMASK[0]	T24 ReportID[2]	T24 ReportID[1]	T24 ReportID[0]	T19 ReportID[0]	T15 ReportID[0]	T10 ReportID[0]	T118 ReportID[0]	T6 ReportID[0]
5	DISRPTMASK[1]	T61[3] ReportID[0]	T61[2] ReportID[0]	T61[1] ReportID[0]	T61[0] ReportID[0]	T56 ReportID[0]	T46 ReportID[0]	T27 ReportID[0]	T24 ReportID[3]
6	DISRPTMASK[2]	T70[2] ReportID[0]	T70[1] ReportID[0]	T70[0] ReportID[0]	T65[2] ReportID[0]	T65[1] ReportID[0]	T65[0] ReportID[0]	T61[5] ReportID[0]	T61[4] ReportID[0]
7	DISRPTMASK[3]	T70[10] ReportID[0]	T70[9] ReportID[0]	T70[8] ReportID[0]	T70[7] ReportID[0]	T70[6] ReportID[0]	T70[5] ReportID[0]	T70[4] ReportID[0]	T70[3] ReportID[0]
8	DISRPTMASK[4]	T70[18] ReportID[0]	T70[17] ReportID[0]	T70[16] ReportID[0]	T70[15] ReportID[0]	T70[14] ReportID[0]	T70[13] ReportID[0]	T70[12] ReportID[0]	T70[11] ReportID[0]
9	DISRPTMASK[5]	T100 ReportID[4]	T100 ReportID[3]	T100 ReportID[2]	Reserved	T100 ReportID[0]	T80 ReportID[0]	T72 ReportID[0]	T70[19] ReportID[0]
10	DISRPTMASK[6]	T100 ReportID[12]	T100 ReportID[11]	T100 ReportID[10]	T100 ReportID[9]	T100 ReportID[8]	T100 ReportID[7]	T100 ReportID[6]	T100 ReportID[5]
11	DISRPTMASK[7]	T112 ReportID[0]	T109 ReportID[0]	T108 ReportID[0]	T100 ReportID[17]	T100 ReportID[16]	T100 ReportID[15]	T100 ReportID[14]	T100 ReportID[13]

Under NDA
© 2021 Microchip Technology Inc. and its subsidiaries

6.5.2.2 Instance 1 – Secondary Communications Interface

TABLE 6-25: CONFIGURATION FOR COMMUNICATIONS CONFIGURATION T18 (SPT_COMMSCONFIG_T18)

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	CTRL	Reserved	RETRIGEN	Reserved					ENABLE
1	Reserved	Reserved							
2	CFG	Reserved				COMMSTYPE			
3	Reserved	Reserved							
4	DISRPTMASK[0]	T24 ReportID[2]	T24 ReportID[1]	T24 ReportID[0]	T19 ReportID[0]	T15 ReportID[0]	T10 ReportID[0]	T118 ReportID[0]	T6 ReportID[0]
5	DISRPTMASK[1]	T61[3] ReportID[0]	T61[2] ReportID[0]	T61[1] ReportID[0]	T61[0] ReportID[0]	T56 ReportID[0]	T46 ReportID[0]	T27 ReportID[0]	T24 ReportID[3]
6	DISRPTMASK[2]	T70[2] ReportID[0]	T70[1] ReportID[0]	T70[0] ReportID[0]	T65[2] ReportID[0]	T65[1] ReportID[0]	T65[0] ReportID[0]	T61[5] ReportID[0]	T61[4] ReportID[0]
7	DISRPTMASK[3]	T70[10] ReportID[0]	T70[9] ReportID[0]	T70[8] ReportID[0]	T70[7] ReportID[0]	T70[6] ReportID[0]	T70[5] ReportID[0]	T70[4] ReportID[0]	T70[3] ReportID[0]
8	DISRPTMASK[4]	T70[18] ReportID[0]	T70[17] ReportID[0]	T70[16] ReportID[0]	T70[15] ReportID[0]	T70[14] ReportID[0]	T70[13] ReportID[0]	T70[12] ReportID[0]	T70[11] ReportID[0]
9	DISRPTMASK[5]	T100 ReportID[4]	T100 ReportID[3]	T100 ReportID[2]	Reserved	T100 ReportID[0]	T80 ReportID[0]	T72 ReportID[0]	T70[19] ReportID[0]
10	DISRPTMASK[6]	T100 ReportID[12]	T100 ReportID[11]	T100 ReportID[10]	T100 ReportID[9]	T100 ReportID[8]	T100 ReportID[7]	T100 ReportID[6]	T100 ReportID[5]
11	DISRPTMASK[7]	T112 ReportID[0]	T109 ReportID[0]	T108 ReportID[0]	T100 ReportID[17]	T100 ReportID[16]	T100 ReportID[15]	T100 ReportID[14]	T100 ReportID[13]

CTRL Field

ENABLE: Enables this instance of the Communications Configuration T18 object. Enabling Instance 1 effectively enables a secondary communications interface and associates it with this Communications Configuration T18 instance

> **NOTE** This control is applicable to Instance 1 only: Instance 0 (that is, the primary communications interface) is always enabled (and cannot be disabled).
>
> This control must be set at reset time to take effect.

MODE: Selects the $\overline{\text{CHG_I2C}}$ line mode for I^2C communications over the I^2C interface. If this bit is set to 0 (the default), the $\overline{\text{CHG_I2C}}$ line operates in Mode 0. If this bit is set to 1, the $\overline{\text{CHG_I2C}}$ line operates in Mode 1. Refer to the relevant device *Datasheet* for more information on the modes for the $\overline{\text{CHG_I2C}}$ line.

> **NOTE** This control has an effect only on an instance of Communications Configuration T18 used to define I^2C communications; it has no effect on an instance of Communications Configuration T18 used to define SPI communications.

RETRIGEN: Enables $\overline{\text{CHG_I2C}}$ or $\overline{\text{CHG_SPI}}$ line host retriggering mode. This provides extra edge-based interrupts to the host on the $\overline{\text{CHG_I2C}}$ or $\overline{\text{CHG_SPI}}$ line (as appropriate for this instance). This mode is enabled if this bit is set to 1 and disabled if set to 0. If this mode is enabled, the $\overline{\text{CHG_I2C}}$ or $\overline{\text{CHG_SPI}}$ line is deasserted once every acquisition cycle for 50 µs and then re-asserted. This creates edges on the $\overline{\text{CHG_I2C}}$ or $\overline{\text{CHG_SPI}}$ line for the host in case it missed the $\overline{\text{CHG_I2C}}$ or $\overline{\text{CHG_SPI}}$ line being asserted.

This re-triggering mechanism is performed once every acquisition cycle, if the time elapsed since the last time the $\overline{\text{CHG_I2C}}$ line was asserted is greater than 10 ms (if the I^2C interface is used for host communications), or if the time elapsed since the last time the $\overline{\text{CHG_SPI}}$ line was asserted is greater than 3 ms (if the SPI interface is used for host communications). This means that, if the configured acquisition interval is less than 10 ms or 3 ms (as appropriate), one or more extra acquisition cycles will be necessary to lapse for the $\overline{\text{CHG_I2C}}$ or $\overline{\text{CHG_SPI}}$ line to be re-triggered, as determined by Power Configuration T7 IDLEACQINT and ACTVACQINT (see "IDLEACQINT and ACTVACQINT Fields").

Note that if the I^2C interface is used, and command code 2 or 3 is set in the COMMAND field, then the command will override the RETRIGEN mode setting.

COMMAND Field

The COMMAND field provides direct control over the $\overline{\text{CHG_I2C}}$ line for the I^2C interface. This overrides the operation of the $\overline{\text{CHG_I2C}}$ line controlled by the MODE bit in the CTRL field. Entering one of the command codes listed in Table 6-26 alters the state of the $\overline{\text{CHG_I2C}}$ line.

NOTE	This control has an effect only if the I^2C interface is used for host communications; it has no effect if the SPI interface is used for host communications.

TABLE 6-26: COMMAND CODES

Command	Description
0	No command (default)
1	Return the $\overline{\text{CHG_I2C}}$ line to normal operation, as determined by the MODE bit of the CTRL field
2	Force the $\overline{\text{CHG_I2C}}$ line high (inactive)
3	Force the $\overline{\text{CHG_I2C}}$ line low (active)

CFG Field

COMMSTYPE: On Instance 1, specifies the communications interface type to be associated with this instance. In other words, it specifies which secondary communications interface is to be used.

NOTE	This control is applicable to Instance 1 only. The communications interface type for instance 0 is preset by the hardware. This control must be set at reset time to take effect.

COMMSTYPE Range: See Table 6-27

TABLE 6-27: COMMSTYPE

Command	Description
0	Reserved
1	Use the SPI Interface as the secondary communications interface
2 to 15	Reserved

DISRPTMASK[] Fields

These fields form an array of bitmaps that indicate which report IDs are to be ignored (disabled) when outputting messages, thus allowing for different reporting behavior on both interfaces.

By default if an object that generates messages has its CTRL RPTEN bit set, it will output messages for all report IDs over all active interfaces. These fields allow reporting to be disabled for the interface associated with this instance on a report ID basis (thus vetoing the object's CTRL RPTEN bit).

NOTE	The appropriate object's CTRL RPTEN bit must be set in order for a message to be sent for a particular report ID: if an object's CTRL RPTEN is set to 0 (disabled), then a message will never be sent, even if the appropriate bit in the DISRPTMASK is set to 0 (enabled).

See Section 1.5.6 "Report IDs" for more information on report IDs.

Under NDA
© 2021 Microchip Technology Inc. and its subsidiaries

6.6 GPIO Configuration T19 Object

6.6.1 INTRODUCTION

The GPIO Configuration T19 object sets up the general-purpose I/O pins. The pins can be set as input or output pins.

Note that the reporting of signals on the input pins occurs only at the appropriate point in the acquisition cycle. This means that the input signal must not occur faster than the Idle or Active Acquisition Interval specified in the Power Configuration T7 object (see Section 3.4 “Power Configuration T7 Object”). Similar criteria apply if an output bit is changed. This operation does not take place instantaneously and enough time must be allowed for it to happen.

IMPORTANT!	If the GPIO pins are used on a UL/IEC 60730 Class B compliant product, it is the responsibility of the host system design to ensure that GPIO Configuration T19 GPIO functionality meets UL/IEC 60730 Class B requirements. This applies in particular to the use of GPIO output pins.

- The GPIO0 function shares a pin with the SYNC function. If the SYNC function is enabled, then this takes precedence over any specified GPIO0 function.
- The hardware debug functionality is shared with some of the GPIO pins. If the hardware debug interface is in use, then these pins cannot be used as GPIO pins. Refer to the relevant device *Datasheet* for details on the hardware debug interface.

The GPIO pins can be used to trigger events, either by transitions from high to low, or vice versa, during use, or by setting the pin high at startup. Note that for the event to be generated, the GPIO pin needs to be set as an output and not overridden by an alternative function.

6.6.2 CONFIGURATION

TABLE 6-28: CONFIGURATION FOR GPIO CONFIGURATION T19 (SPT_GPIOPWM_T19)

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	CTRL	Reserved					FORCERPT	RPTEN	ENABLE
1	REPORTMASK (1)	Report mask bit field							
2	DIR (1)	Pin direction bit field							
3	INTPULLUP (1)	Internal pull-ups bit field							
4	OUT (1)	Output value bit field							
5	WAKE (2)	Wake up on bit change bit field							
6	Reserved	Reserved							
7	Reserved	Reserved							
8	Reserved	Reserved							
9	Reserved	Reserved							
10	Reserved	Reserved							
11	Reserved	Reserved							
12	TRIGGER[0]	Trigger for GPIO0							
13	Reserved	Reserved							
14	Reserved	Reserved							
15	Reserved	Reserved							

Note 1: These fields are bit fields with each bit representing a GPIO pin, as in Table 6-29

2: This field is a bit field with each bit representing a GPIO pin, as in Table 6-30

The bit field format for the REPORTMASK, DIR, INTPULLUP, OUT fields is shown in Table 6-29.

TABLE 6-29: BIT FIELD FORMAT – REPORTMASK, DIR, INTPULLUP, OUT

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Bytes 1 to 4 above		Reserved					GPIO2	GPIO1	GPIO0

The bit field for the WAKE field is shown in Table 6-30. Note that the WAKE functionality can be used only with certain GPIO pins.

TABLE 6-30: BIT FIELD FORMAT – WAKE

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
5	WAKE	Reserved							GPIO0

CTRL Field

ENABLE: Enables or disables the use of the GPIO Configuration T19 object. The object is enabled if set to 1, and disabled if set to 0.

RPTEN: Enables or disables the GPIO Configuration T19 object to report messages. If this bit is set, the device sends messages on bit changes. Reporting is enabled if set to 1, and disabled if set to 0.

FORCERPT: Writing a 1 to this bit forces the GPIO Configuration T19 object to send a message with its current status. The object then clears this bit. The RPTEN bit must be enabled for this to work. See Section 6.6.3 "Messages" for information on the message data.

REPORTMASK Field

Specifies a bit mask that suppresses the sending of messages on bit changes. Set these bits to 1 if the corresponding GPIO pins should not generate a message on a bit change. The pins must be set as input pins in the DIR field. The default value of 0 causes all bit changes to generate messages.

DIR Field

Sets the direction of the pins: 1 = GPIO*n* is an output pin; 0 = GPIO*n* is an input pin.

The default value of 0 sets a pin as an input pin, in which case if the pin is unused, it should be connected as described in the relevant device *Datasheet*.

NOTE	If the pins are set as inputs (or are left in their default state) and the object is subsequently enabled, the input pins are assumed to have a low initial state.

INTPULLUP Field

Enables the internal pull-up resistors for any input pins. If these bits are set to 1, the internal pull-up resistors for the corresponding pins are enabled. The pins must be set as input pins in the DIR field. If this field is set to 0 (default), all the internal pull-up resistors for the pins are disabled.

OUT Field

Sets the state of the output GPIO pins. Setting these bits to 1 sets the pins high. Clearing these bits to 0 sets the pins low. The pins must already be set as output pins in the DIR field.

WAKE Field

NOTE	On the mXT640UD-CCUBHA1 Family, the wake feature is not supported for GPIO1 or GPIO2.

Causes the device to wake up on bit changes if it is asleep. If these bits are set to 1, the device will wake up on a bit change on the IO pins. When the bit change occurs, the device wakes up, reports the GPIO message and then returns to sleep. The pins must already be set as input pins in the DIR field. If this field is set to 0 (default), the device will remain asleep on bit changes.

TRIGGER[] Fields

NOTE	On the mXT640UD-CCUBHA1 Family, trigger functionality can be enabled on GPIO0 only.

These fields select a trigger source for the corresponding GPIO pin (see Table 6-31 and Table 6-32). The trigger inverts the appropriate GPIO pin if the source object is active. The pin must be set as an output pin in the DIR field.

Note that on entering Deep Sleep mode, any triggered pins stick at their current triggered state until the device wakes again.

Under NDA

© 2021 Microchip Technology Inc. and its subsidiaries

TABLE 6-31: TRIGGER GPIO PINS

Field	GPIO Pin
TRIGGER[0]	Selects a trigger source for GPIO0 pin

Range: See Table 6-32

TABLE 6-32: TRIGGER[] FIELD VALUES

Value	Description
0 (or not listed below)	Trigger disabled.
4	Triggered whenever any touch is detected on Key 0 of the Key Array T15 object. Note that the trigger is related to the actual key and is not affected by any Key IDs configured by the Key ID Configuration T16 object.
7	Triggered whenever the Self Test Control T10 object sends an All BIST Tests Passed message (see Section 6.2.1.2 “Built-In Self Test (BIST)”). This can be used to provide a heartbeat (“alive”) signal for functional safety purposes.

6.6.3 MESSAGES

The message data for the GPIO Configuration T19 object is shown in Table 6-33.

TABLE 6-33: MESSAGE DATA FOR GPIO CONFIGURATION T19 (SPT_GPIOPWM_T19)

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	STATUS	Input pin states							

STATUS: Gives the current state of all GPIO pins that are set as inputs. Any GPIO pin that is set as an output reports as a zero. This field is a bit field with each bit representing a GPIO pin. See Table 6-29 on page 201 for the format of this field.

6.7 User Data T38 Object

6.7.1 INTRODUCTION

The User Data T38 object provides a small area on the device for the user to store their own custom data, such as the user's own version information. The size of this area varies from device to device. On the mXT640UD-CCUBHA1 Family the size of this area is 64 bytes.

Any data stored in this object will be backed up to the non-volatile memory when the Command Processor T6 object processes a backup command (see Section 3.3 "Command Processor T6 Object").

NOTE	The data is not included in the checksum reported by the Command Processor T6 after a backup has been processed.

6.7.2 CONFIGURATION

TABLE 6-34: CONFIGURATION FOR SPT_USERDATA_T38

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0 – 63	DATA[]	User data							

DATA[] Field

This field contains the user's data.

NOTE	The User Data T38 object does not provide any mechanism for checking the integrity of the data held within the object. It is therefore the responsibility of the host to perform any such checks for compliance with UL/IEC 60730 Class B.

Under NDA

© 2021 Microchip Technology Inc. and its subsidiaries

6.8 CTE Configuration T46 Object

6.8.1 INTRODUCTION

The CTE Configuration T46 object controls how the capacitive touch engine (CTE) in the device samples the mutual capacitive signals. This object also controls pulse synchronization during mutual capacitance measurements.

The X (drive) lines are sampled one at a time, with at least 2 analog-to-digital conversions (ADCs) performed on each X line in turn. The number of ADCs per X line is configured by a combination of the ACTVSYNCSPERX / IDLESYNCSPERX and ADCSPERSYNC fields.

The number of ADCs per X is a nominal value only. Other objects can suggest other values for the ADCs per X setting under specific circumstances, such as when external noise is detected or when a stylus is in use. The mXT640UD-CCUBHA1 Family will use the highest required setting for ADCs per X, as determined by the various objects. The objects that can suggest a maximum ADCs per X settings include:

- Passive Stylus T47 for use with passive stylus touches
- Glove Detection T78 for use with glove touches
- Noise Suppression T72 for use when external noise has been detected

NOTE	The minimum Active/Idle ADCs per X is saturated to 2. Therefore, if less than 2 ADCs per X is configured, 2 ADCs per X will be used.

The number of pulses for each ADC can also be configured. The amount of signal measured depends on the number of pulses per ADC. The more pulses per ADC, the higher the amount of charge that is measured. The number of pulses per ADC is determined by the PULSESPERADC field (see "PULSESPERADC Field").

Therefore, if ADCSPERSYNC = 0 or 1, the total number of pulses per X line is equivalent to:

(PULSESPERADC+1) × (ACTVSYNCSPERX or IDLESYNCSPERX)

Alternatively, if ADCSPERSYNC ≥ 2, the total number of pulses per X line is equivalent to:

(PULSESPERADC+1) × (ADCSPERSYNC–1) × (ACTVSYNCSPERX or IDLESYNCSPERX)

6.8.1.1 Pulse Synchronization

IMPORTANT!	Pulse Synchronization functionality on the mXT640UD-CCUBHA1 Family is not considered UL/IEC 60730 Class B compliant. Therefore, the device should not be configured to use Pulse Synchronization in a UL/IEC 60730 Class B product.

NOTE	The SYNC line can be used for either pulse synchronization or frame synchronization (see "SYNC Field"), but not both. If the SYNC line is used for pulse synchronization functionality, then it cannot be used for frame synchronization.

The ADCSPERSYNC field determines how the ADCs are synchronized to the SYNC line, if at all:

- If the ADCSPERSYNC field is set to zero, then no pulse synchronization occurs for mutual capacitance measurements.
- If the ADCSPERSYNC field is set to 1, then pulse synchronization occurs for mutual capacitance measurements. In this case, the rising and falling X edges of each ADC are synchronized to the falling edge of the SYNC signal.
- If the ADCSPERSYNC field is set to a value of 2 or above, the ADCs are arranged in "Sync Groups". The number of ADCs per Sync Group is (ADCSPERSYNC – 1). In this case, the ACTVSYNCSPERX and IDLESYNCSPERX fields determine the number of Sync Groups per X line. Thus, the total number of ADCs per X is equivalent to:

  (ADCSPERSYNC – 1) × ACTVSYNCSPERX or IDLESYNCSPERX

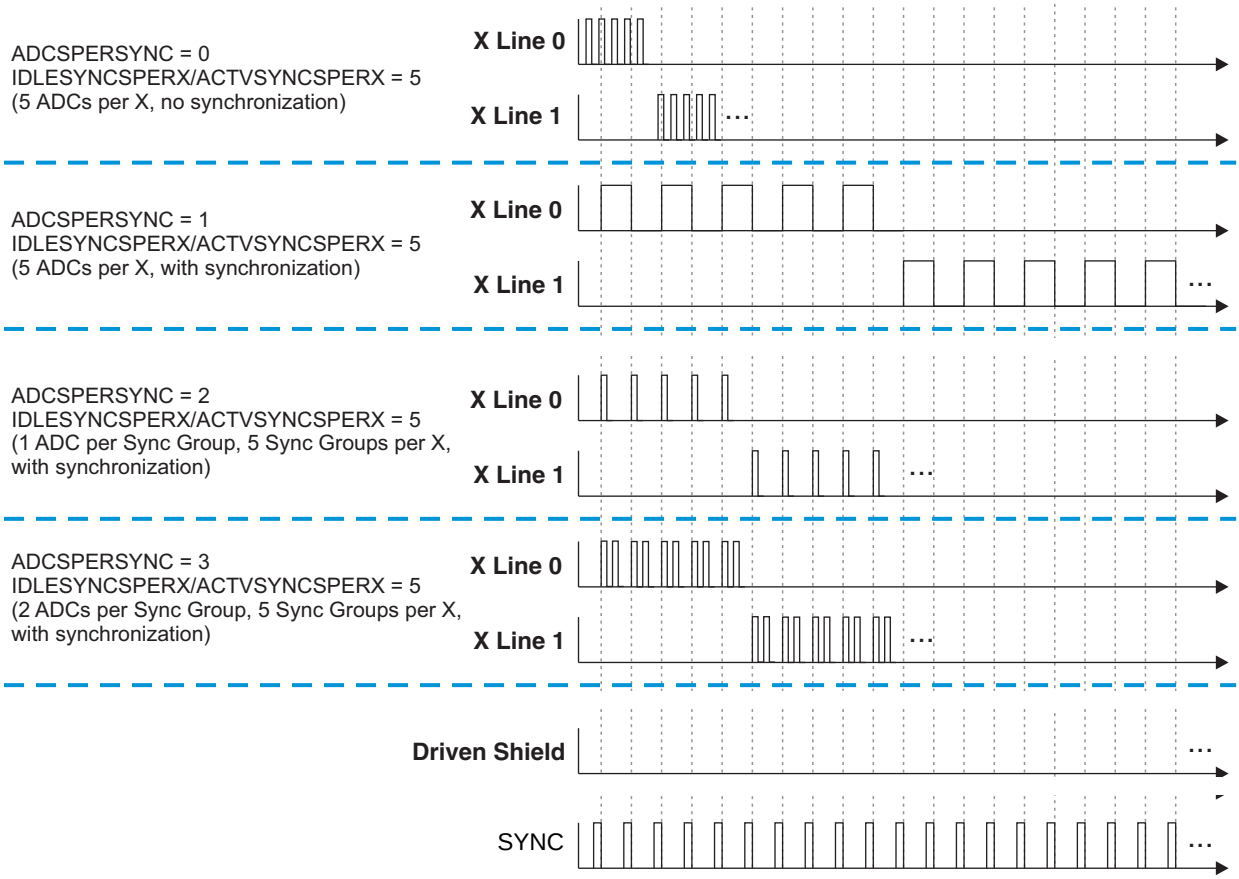
  with a minimum value of 2.

Note the following:

- Pulse synchronization is enabled for each measurement type independently. To enable pulse synchronization on self capacitance measurement types, see Section 6.16 "Self Capacitance Configuration T111 Object"
- If pulse synchronization is turned on (that is, ADCSPERSYNC is non-zero), GPIO0 cannot be configured by the GPIO Configuration T19 object. GPIO0 should be set as an input, or the GPIO Configuration T19 object disabled, for the pulse synchronization to work.

If there is a requirement to toggle pulse synchronization on and off at run time, the device must be configured and backed up with pulse synchronization enabled in either CTE Configuration T46 or Self Capacitance Configuration T111. This ensures that the device will start up with pulse synchronization enabled for at least one measurement type and the SYNC pin is latched for use. This prevents the pin from being used for other uses, such as configured as GPIO0 by the GPIO Configuration T19 object.

FIGURE 6-3: CTE CONFIGURATION T46 PULSE SYNCHRONIZATION



If the system has been configured to use pulse synchronization and there is a failure to detect a SYNC signal, the default behavior of the system is to continue measurements unsynchronized using user-configured timing. If this behavior is not required, CFG SYNCLOSSMODE can be set to 1 to pause measurements altogether while pulse synchronization is absent. When a SYNC signal is subsequently detected, pulse synchronization will be re-enabled and the measurements will continue with synchronized ADCs. Note that CTE Configuration T46 CFG SYNCLOSSMODE has an effect on all mutual and self capacitance measurements.

6.8.2 SPREAD SPECTRUM

NOTE	Use of Spread Spectrum Variance is incompatible with Pulse Synchronization functionality. If Pulse Synchronization functionality is enabled at the same time as SPREADSPECVAR (see "ADCSPERSYNC Field"), then Pulse Synchronization will take precedence.

The X drive burst pulses can have a spread spectrum algorithm applied that applies a decreasing frequency (spread spectrum variance) to the bursts (see "SPREADSPECVAR Field"). The Spread Spectrum algorithm calculates a set of 16 decreasing frequencies, starting with the base frequency (that is, frequency 1 is equal to the base frequency), and these 16 frequencies are then applied to the X drive burst pulses in the following order:

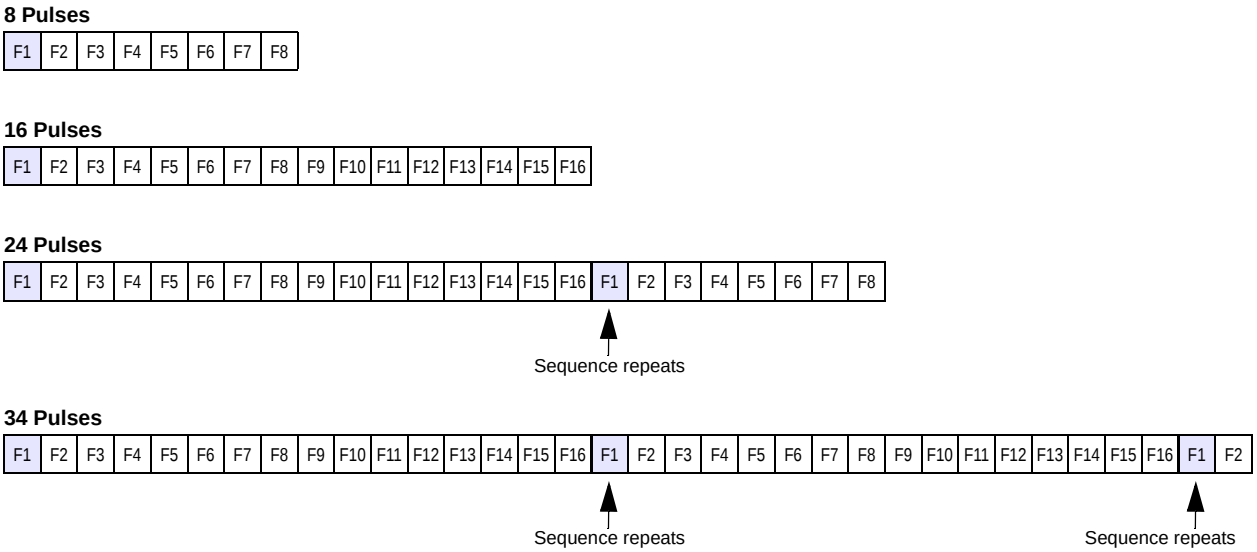
Frequency 1 … Frequency *n*

where *n* is the number of pulses.

Under NDA
© 2021 Microchip Technology Inc. and its subsidiaries

If the number of pulses per X line is less than 16, then not all 16 frequencies will be used and the effective spread spectrum variance is reduced.

FIGURE 6-4: SPREAD SPECTRUM VARIANCE – EXAMPLES



F*n* = Calculated frequency number

6.8.3 CONFIGURATION

TABLE 6-35: CONFIGURATION FOR CTE CONFIGURATION T46 (SPT_CTECONFIG_T46)

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Reserved	Reserved							
1	Reserved	Reserved							
2	IDLESYNCSPERX	Number of ADCs per X when idle							
3	ACTVSYNCSPERX	Number of ADCs per X when active							
4	ADCSPERSYNC	Reserved		ADCSPERSYNC					
5	PULSESPERADC	Number of pulses for each ADC							
6	XSLEW	X pulse slew rate							
7	SYNCDELAY	Delay from SYNC falling edge LSByte							
8		Delay from SYNC falling edge MSByte							
9	XVOLTAGE	XVdd voltage mode							
10	Reserved	Reserved							
11	INRUSHCFG	Reserved					YRESISTOR		
12	Reserved	Reserved							
13	Reserved	Reserved							
14	Reserved	Reserved							
15	Reserved	Reserved							
16	Reserved	Reserved							
17	Reserved	Reserved							
18	CFG	Reserved				DELTASHIFT		SYNCLOSSMODE	DISTEMPCOMP
19	SYNCPERIOD	Synchronization period (LSByte)							
20		Synchronization period (MSByte)							

TABLE 6-35: CONFIGURATION FOR CTE CONFIGURATION T46 (SPT_CTECONFIG_T46) (CONTINUED)

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
21	SPREADSPECVAR	Reserved	Spread spectrum variance						
22	YPHASEDELAY	Y line phase delay (LSByte)							
23		Y line phase delay (MSByte)							

IDLESYNCSPERX Field

This field sets the number of ADCs per X in Idle mode. The range for this field is 1 to 255, where a value of 0 means 8.

See Section 6.8.1 “Introduction” for more information on how this field helps determine the actual number of ADCs per X.

Range: 0 (8), 1 to 255

ACTVSYNCSPERX Field

This field determines the number of ADCs per X in Active mode. The range for this field is 1 to 255, where a value of 0 means 8.

See Section 6.8.1 “Introduction” for more information on how this field helps determine the actual number of ADCs per X.

Range: 0 (8), 1 to 255

ADCSPERSYNC Field

NOTE	If synchronization is turned on (that is, ADCSPERSYNC is non-zero), GPIO0 cannot be configured by the GPIO Configuration T19 object. GPIO0 must be set as an input, or the GPIO Configuration T19 object disabled, for the CTE Configuration T46 burst pulse synchronization to work.

ADCSPERSYNC: Determines whether or not the ADC conversions are synchronized to the SYNC pin for mutual capacitance measurements. To enable pulse synchronization on self capacitance measurement types, see Section 6.16 “Self Capacitance Configuration T111 Object”.

A value of 0 means no synchronization is required (that is, mutual capacitance measurements are not synchronized to the input signal on the SYNC pin).

A value of 1 synchronizes the rising and falling X edges of each ADC to the falling edge of the SYNC signal (see Figure 6-3 on page 206 for an example).

A value equal to or greater than 2 determines the number (value minus 1) of ADCs per Sync Group.

Range: 0 (no synchronization),
1 (ADCs synchronized to SYNC)
2 to 63

PULSESPERADC Field

This field determines the number of pulses for each ADC conversion. This is specified as the number of pulses minus 1. The range for this field is 0 to 3, which represents 1 to 4 pulses.

Range: 0 to 3

XSLEW Field

This field sets the X slew rate. The valid values for this field are given in Table 6-36.

TABLE 6-36: XSLEW SETTINGS

Setting	Slew Rate
0	No slew limit
1	Non-calibrated setting

Under NDA

© 2021 Microchip Technology Inc. and its subsidiaries

SYNCDELAY Field

This 16-bit field specifies the delay from when the SYNC falling edge occurs to when the X edge is generated. The delay is specified in units of the CTE clock period (20 ns; 50 MHz), where a value of 0 means no delay.

Range: 0 (no delay), 1 to 65534

XVOLTAGE Field

This field specifies the voltage for XVdd. Refer to the relevant device *Datasheet* for information on configuring the hardware to operate correctly at the selected XVdd voltage level.

NOTE	The device should be backed up and reset for this setting to take effect.

On the mXT640UD-CCUBHA1 Family the XVOLTAGE field has the format shown in Table 6-37.

TABLE 6-37: XVOLTAGE FIELD

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
9	XVOLTAGE	Reserved							MODE

The possible values for the XVdd Voltage are listed in Table 6-38.

NOTE	Care should be taken when changing the voltage settings to ensure that it matches the hardware configuration (that is, the capacitor between EXTCAP0 and EXTCAP3). If the XVOLTAGE field is configured for High voltage (XVOLTAGE = 0) but the hardware is configured for Low voltage (that is, the capacitor between EXTCAP0 and EXTCAP3 is removed), the system will trigger SIGERR messages until the error condition is removed. Refer to the relevant device *Datasheet* for more information on the use of the EXTCAP0 and EXTCAP3 pins.

Range: See Table 6-38

TABLE 6-38: XVOLTAGE SETTINGS

Setting	Mode
0	Use High voltage; that is, use the voltage tripler (default)
1	Use Low voltage; that is, use the voltage doubler

INRUSHCFG Field

This field controls the selection of the Inrush Current Resistors for mutual touch measurements.

YRESISTOR: Selects the Y line inrush resistors for all the Y lines.

YRESISTOR Range: See Table 6-39.

TABLE 6-39: YRESISTOR VALUES

YRESISTOR Setting	Resistor Value
0	50 Ω (default)
1	71 Ω
2	117 Ω
3	180 Ω
4	276 Ω
5	424 Ω
6	651 Ω
7	1 kΩ

CFG Field

NOTE	The CFG field control applies to all mutual and self capacitance measurement types in the device.

DISTEMPCOMP: Disables Temperature Compensation. When this control is set to 1, temperature compensation for both mutual capacitance and self capacitance measurements is disabled. It is not recommended to set this control to 1 unless advised to do so by Microchip.

Note that temperature compensation is also disabled when pulse synchronization is active but SYNCPERIOD is 0 (see "SYNCPERIOD Field").

SYNCLOSSMODE: Specifies the Synchronization Loss Mode. This controls the system behavior when the SYNC pulse is lost for pulse synchronized measurements.

If this bit is set to 0, all measurements will continue to measure in an unsynchronized manner with timing based upon configuration settings.

If this bit is set to 1, mutual and self capacitance measurements will cease to measure until the pulse synchronization signal returns.

DELTASHIFT: Specifies the Delta Shift that is used to scale (shift right) all 8-bit and 16-bit deltas, except single-ended self capacitance deltas, in any debug data. The delta scaling also allows for thresholds to be specified as 8-bit values for configuration purposes. See Section 1.9.1 "Delta Scaling" for details.

By default (DELTASHIFT = 0) no shift occurs (that is, there is no scaling applied), although the device still converts between 8-bit and 16-bit deltas where necessary. By setting this control to a value greater than 0, a greater usable range for all 8-bit threshold controls can be used.

IMPORTANT!	Modifying the delta scaling may affect the configuration values used for various threshold and gain values. The device may therefore need retuning if DELTASHIFT is altered.

Range: See Table 6-40

TABLE 6-40: DELTASHIFT CONTROL

DELTASHIFT Value	Scaling	Maximum Thresholdable Unscaled 16-bit Delta
0 (default)	÷ 1	2040
1	÷ 2	4080
2	÷ 4	8160
3	÷ 8	16320

SYNCPERIOD Field

NOTE	The SYNCPERIOD field control applies to all mutual and self capacitance measurement types in the device that use the Pulse Synchronization functionality.

This control sets the nominal SYNC clock period. This setting is required for both mutual capacitance and self capacitance touch temperature compensation if the pulse synchronization feature is enabled (see "ADCSPERSYNC Field").

The period is specified in units of the CTE clock period (20 ns; 50 MHz), where a value of 0 is not recommended unless advised by Microchip.

Range: 0 to 65535

SPREADSPECVAR Field

NOTE	Use of Spread Spectrum Variance is incompatible with Pulse Synchronization functionality. If Pulse Synchronization functionality is enabled at the same time as SPREADSPECVAR (see "ADCSPERSYNC Field"), then Pulse Synchronization will take precedence.

This field specifies the Spread Spectrum Variance. This is the overall total percentage variance when applying spread spectrum to mutual capacitance measurements and determines the amount the frequency decreases over each set of bursts (see Section 6.8.2 "Spread Spectrum").

Under NDA
© 2021 Microchip Technology Inc. and its subsidiaries

The Spread Spectrum Variance is specified as a sliding range from 0 to 127, which represents a range of 0 to 20% percent frequency decrease (which corresponds to a 25% increase in the period). The default value of 0 means that spread spectrum functionality is disabled.

Range: 0 (disabled), 1 to 127

YPHASEDELAY Field

This field specifies the Y Line Phase Delay. This is the delay between each Y line setup (biasing) before measurement. This field applies to all mutual and self capacitance measurements.

The delay is specified in units of the CTE clock period (20 ns; 50 MHz).

NOTE	The transition from a mutual capacitance measurement to a self capacitance measurement does not use Y line phasing if self capacitance pre-charge is disabled (see Self Capacitance Global Configuration T109 "CTRL Field" DISPRECHARGE control). In this case, settings configured for Voltage Modulation are used instead (see Section 6.20 "Self Capacitance Voltage Modulation T133 Object"). Note that the phase delay is still present at this transition, as X lines are also phased at this point.

Range: 0 to 65535

6.8.4 CONFIGURATION CHECKS

The CTE Configuration T46 object causes a configuration check to be performed in the following circumstances:

- When certain fields are changed

A configuration check may determine that a configuration error has occurred (for example, if a setting is set outside of its allowed range or a conflict has occurred between two settings). This is signaled to the host (see Section 3.3.3 "Messages"), and the device halts until the error has been corrected. To fix the error, the object settings should be checked to verify that they are all within their allowed limits, as stated in the field descriptions.

TABLE 6-41: CONFIGURATION CHECKS FOR CTE CONFIGURATION T46 (SPT_CTECONFIG_T46)

Field	Changing The Field Causes...		Effect of Configuration Checks On Field
	Configuration Check	Automatic Recalibration	
CTRL	No	No	None
MODE	Yes	No	None
IDLESYNCSPERX	Yes	No	None
ACTVSYNCSPERX	Yes	No	None
ADCSPERSYNC	No	No	None
PULSESPERADC	Yes	Yes	Error if out of range
XSLEW	No	Yes	None
SYNCDELAY	No	No	None
XVOLTAGE	No	No	None
INRUSHCFG	No	Yes	None
CFG	No	No	None
SYNCPERIOD	No	No	None
SPREADSPECVAR	No	No	None
YPHASEDELAY	No	No	None

6.8.5 MESSAGES

The message data for the CTE Configuration T46 object is shown in Table 6-42.

TABLE 6-42: MESSAGE DATA FOR CTE CONFIGURATION T46 (SPT_CTECONFIG_T46)

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	STATUS	Reserved							CHKERR

STATUS Field

CHKERR: Not used in this product; this field should always report a zero value.

Under NDA

© 2021 Microchip Technology Inc. and its subsidiaries

6.9 Timer T61 Object

6.9.1 INTRODUCTION

The Timer T61 object provides a timer that can be set up to trigger either a single message after a delay or a stream of regular messages. When the timer expires, a message is sent in the appropriate manner. The timer has two modes, set by the MODE field:

- **Single trigger mode:** Triggers a single message after a specified period.
- **Repeat trigger mode:** Triggers repeat messages at a specified interval.

There are 6 instances of the Timer T61 object on the mXT640UD-CCUBHA1 Family.

6.9.2 CONFIGURATION

TABLE 6-43: CONFIGURATION FOR TIMER T61 (SPT_TIMER_T61)

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	CTRL	Reserved						RPTEN	ENABLE
1	CMD	Timer command							
2	MODE	Reserved							MODE
3	PERIOD	Timer period LSByte							
4		Timer period MSByte							

CTRL Field

ENABLE: Enables or disables the use of the Timer T61 object. The object is enabled if set to 1, and disabled if set to 0.

RPTEN: Enables or disables the Timer T61 object to report messages. Reporting is enabled if set to 1, and disabled if set to 0.

CMD Field

This field allows command to be given to the timer. See Table 6-44 for a list of available commands.

TABLE 6-44: TIMER COMMANDS

Command	Description
0	No action
1	Start the timer
2	Stop the timer
3	Force the timer to send a message
4 to 255	Reserved

This field is cleared to zero once the command has been processed. Note that the host should write a command to this field only if the field is zero (that is, the previous command has been processed), otherwise the previous command will be overwritten.

MODE Field

MODE: Configures the timer mode; see Table 6-45.

TABLE 6-45: TIMER MODES

Value	Mode
0	The timer operates in single trigger mode
1	The timer operates in repeat trigger mode

PERIOD Field

This field specifies the timer period in 1 ms units, where a value of 0 means 1000 ms.

Range: 0 (1000), 1 to 65535

6.9.3 MESSAGES

The message data for the Timer T61 object is shown in Table 6-46.

TABLE 6-46: MESSAGE DATA FOR TIMER T61 (SPT_TIMER_T61)

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	STATUS	ELAPSED	START	STOP	FORCERPT	Reserved			RUNNING
2	CNTVAL	Current timer count value LSByte							
3		Current timer count value MSByte							

STATUS Field

RUNNING: The timer is running.

FORCEREPT: The timer has been forced to report via the CMD configuration field.

STOP: The timer has stopped.

START: The timer has started to run.

ELAPSED: The specified period in the PERIOD configuration field has elapsed.

CNTVAL Field

This field reports the current timer count value.

Under NDA

© 2021 Microchip Technology Inc. and its subsidiaries

6.10 Dynamic Configuration Controller T70 Object

6.10.1 INTRODUCTION

A Dynamic Configuration Controller T70 object provides a flexible way for the host to change configurations during run-time in response to a trigger. The configurations are defined for a specific event and when that event is triggered, the associated configuration settings are changed. There are 20 instances of the Dynamic Configuration Controller T70 object on the mXT640UD-CCUBHA1 Family.

Events can take two forms:

- Double edge event

  A double edge event is an event with an on and off state (such as a touch) and can therefore be considered to have a rising edge and a falling edge. The rising edge is the positive state change at the start of the event and the falling edge is the negative state change at the end of the event.

  For example, a noise detection event typically has two edges: the first (rising) edge is a change from a “no noise” to a “noise detected” state, and the second (falling) edge is the change back from a “noise detected” to a “no noise” state.

  Host defined events have the behavior of double edge events.
- Single edge event

  A single edge event is an event with a positive state change only.

  For example, a calibration command is a single edge event.

A Dynamic Configuration Controller T70 object can be configured to respond to the following:

- A single edge event
- A single edge (rising or falling) of a double edge event
- Both edges of a double edge event

The ENUMID of the EVENT field is used to specify which event to trigger on.

Alternative sets of configuration settings can be read from either the Non-volatile memory (NVM) or from the Dynamic Configuration Container T71 object (see Section 6.11 “Dynamic Configuration Container T71 Object”), as selected by the ALTSRC bit of the CTRL field. If the settings are to be read from the Dynamic Configuration Container T71 object, the SRCOFFSET field specifies the offset to use.

Note that, in the case of the Command Processor T6, the bytes that contain the RESET and BACKUPNV fields cannot be loaded with configuration data. That is, If the OBJTYPE is 6 (Command Processor T6), then DSTOFFSET must be at least 2 with an appropriate LENGTH, otherwise a configuration error is generated. All other objects in the device, including the Dynamic Configuration Controller T70 itself, can be loaded with new configuration data.

The BACKUPNV Field of the Command Processor T6 object supports commands that help avoid the issue of a Dynamic Configuration Controller T70 object modifying settings before they are backed up. The host can also restore a configuration from the NVM using the Command Processor T6 object. The restore function does not require a reset. See Section 3.3 “Command Processor T6 Object” for more details.

To set up and use a Dynamic Configuration Controller T70 object:

1. Stop the Dynamic Configuration Controller T70 object using the BACKUPNV field in the Command Processor T6 object (See Section 3.3.2 “Configuration”).
2. Write the intended settings to the device and back up the configuration to NVM.

Figure 6-5 shows an example of using a Dynamic Configuration Controller T70 object to set up a double edge trigger for a host event (EVENT HOSTDEF = 1) that loads alternative Multiple Touch Touchscreen T100 settings.

In this case, CTRL ALTSRC is set to 0 so that the configuration data is read from the Dynamic Configuration Container T71 object on the rising edge (event start) and from NVM on the falling edge (event end). The configuration data consists of 25 bytes of the Multiple Touch Touchscreen T100 settings, starting from byte 28 (GAIN).

FIGURE 6-5: EXAMPLE HOST EVENT – LOADING ALTERNATIVE MULTIPLE TOUCH TOUCHSCREEN T100 SETTINGS

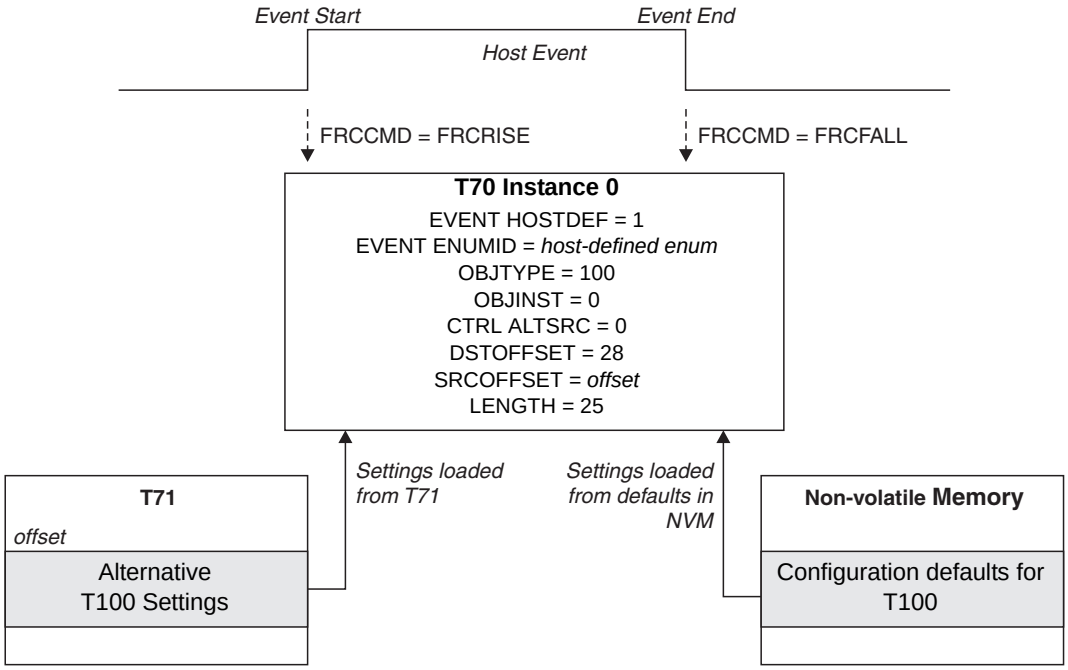
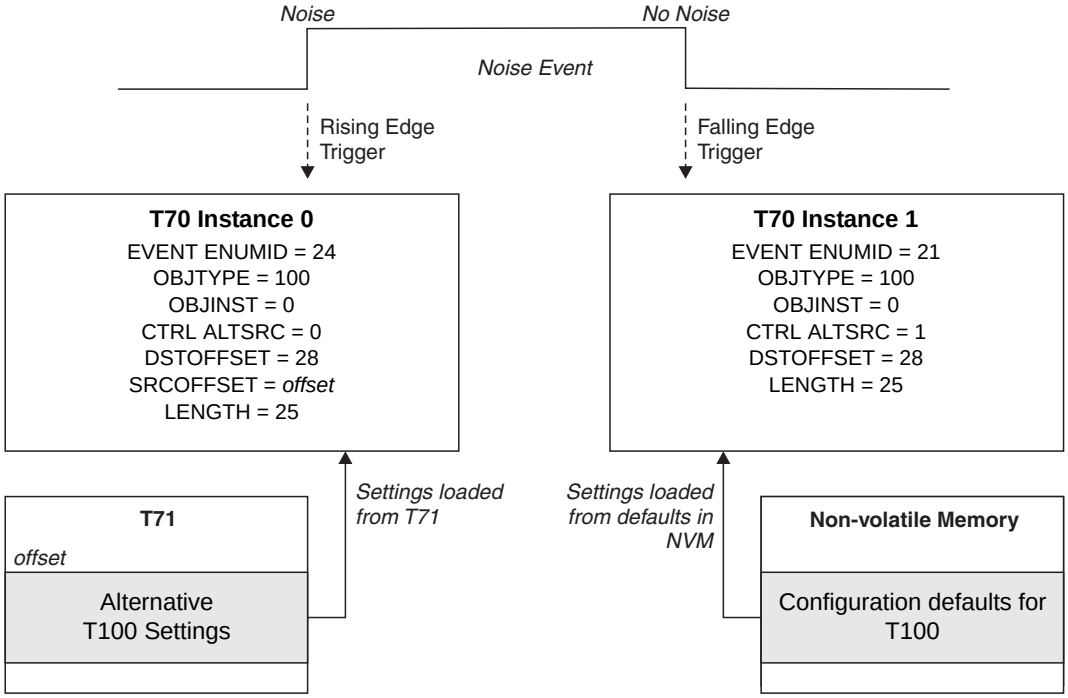


Figure 6-6 shows an example of using two separate Dynamic Configuration Controller T70 object instances to set up two triggers for noise events: one on the rising edge (EVENT ENUMID = 24) on the transition from the STABLE to NOISY state, and one on the falling edge (EVENT ENUMID = 21) on the transition from the NOISY to STABLE state.

FIGURE 6-6: EXAMPLE SINGLE EDGE EVENTS – LOADING MULTIPLE TOUCH TOUCHSCREEN T100 SETTINGS IN RESPONSE TO NOISE



Under NDA
© 2021 Microchip Technology Inc. and its subsidiaries

In this case, alternative settings are loaded for Multiple Touch Touchscreen T100 at the start of the noise event and NVM settings reloaded at the end of the noise event. As with Figure 6-5, the configuration data consists of 25 bytes of the Multiple Touch Touchscreen T100 settings, starting from byte 28 (GAIN).

6.10.2 CONFIGURATION

TABLE 6-47: CONFIGURATION FOR DYNAMIC CONFIGURATION CONTROLLER T70 (SPT_DYNAMICCONFIGURATIONCONTROLLER_T70)

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	CTRL	Reserved			FRCCMD		ALTSRC	RPTEN	ENABLE
1	EVENT	ENUMID (LSByte)							
2		HOSTDEF	ENUMID (MSbits)						
3	OBJTYPE	Object type element							
4	Reserved	Reserved							
5	OBJINST	Object instance							
6	DSTOFFSET	Destination offset							
7	SRCOFFSET	Source offset LSByte							
8		Source offset MSByte							
9	LENGTH	Length offset							

CTRL Field

ENABLE: Enables or disables the use of the Dynamic Configuration Controller T70 object. The object is enabled if set to 1, and disabled if set to 0.

RPTEN: Enables or disables the Dynamic Configuration Controller T70 object to report messages. Reporting is enabled if set to 1, and disabled when set to 0.

ALTSRC: This bit allows the host to select the location of the source data (see Table 6-48). The source data is selected from either the Dynamic Configuration Container T71 object with an associated offset, as defined in the SRCOFFSET field, or from the corresponding backed-up area in NVM.

TABLE 6-48: ALTSRC BIT – CONFIGURATION DATA SOURCES

Event Type	ALTSRC = 1	ALTSRC = 0
Single Edge Triggered Event	Configuration data read from NVM.	Configuration data read from Dynamic Configuration Container T71
Double Edge Triggered Event	Configuration data read from NVM on rising edge and from Dynamic Configuration Container T71 on falling edge	Configuration data read from Dynamic Configuration Container T71 on rising edge and from NVM on falling edge

FRCCMD: The FRCCMD control is used to allow the host to trigger the event configured in the EVENT Field. The FRCRISE and FRCFALL commands should be used accordingly to trigger on the corresponding edge of the configured event.

NOTE	If more than one instance of the Dynamic Configuration Controller T70 are configured to respond to the same event, using the FRCCMD control to force an event to be triggered on one instance will force all configured instances to respond as well.

Table 6-49 gives the values for the FRCCMD control.

TABLE 6-49: FRCCMD COMMAND CODES

FRCCMD	Command	Description
00	None	No trigger
01	FRCRISE	Triggers the event on a rising edge.
10	FRCFALL	Triggers the event on a falling edge.
11	Reserved	–

EVENT Field

HOSTDEF: This field allows the host to specify the type of event: a device defined enumeration is specified if set to 0, and a host defined enumeration is specified if set to 1. This field must be used along with the FRCCMD control to trigger the host defined events. Host defined events have the behavior of double-edged events.

ENUMID: This field configures the event enumeration ID that is used to trigger an event for a particular instance of the Dynamic Configuration Controller T70 object. Table 6-50 lists the standard ENUMIDs available on the mXT640UD-CCUBHA1 Family. All other values have an undefined behavior unless the HOSTDEF bit is set to 1, in which case the ENUMID is a host-defined value.

TABLE 6-50: EVENT ENUMERATIONS

ENUMID	Description	Event Type
0	**None** No action.	–
1	**User Calibration Command** A calibration is requested by the user through the Command Processor T6 object. See also event enumerations 2 and 41.	Single Edge
2	**User or System Calibration Command** A calibration has occurred. The calibration may be invoked by the system or by a user request through the Command Processor T6 object. See also event enumerations 1 and 41.	Single Edge
10	**Touch Detect (High)** Multiple Touch Touchscreen T100 goes from being not touched to being touched.	Single Edge
11	**Touch Detect Release (Low)** Multiple Touch Touchscreen T100 goes from being touched to not touched.	Single Edge
12	**Touch Detect Change (High or Low)** Multiple Touch Touchscreen T100 either goes from being not touched to touched, or from being touched to not touched.	Double Edge
13	**Timer 0 Elapsed** Instance 0 of the Timer T61 object has elapsed. See also event enumerations 14 and 28 to 31	Single Edge
14	**Timer 1 Elapsed** Instance 1 of the Timer T61 object has elapsed. See also event enumerations 13 and 28 to 31	Single Edge
18	**Touchscreen Suppression On (High)** Touchscreen suppression has started.	Single Edge
19	**Touchscreen Suppression Off (Low)** Touchscreen suppression has stopped.	Single Edge
20	**Touchscreen Suppression Change (High or Low)** Touchscreen suppression has started or stopped.	Double Edge
21	**Noise Suppression STABLE State, Dual X Off** Noise Suppression T72 has changed to the STABLE state with Dual X disabled.	Single Edge
22	**Noise Suppression STABLE State, Dual X On** Noise Suppression T72 has changed to the STABLE state with Dual X enabled.	Single Edge
23	**Noise Suppression NOISY State, Dual X Off** Noise Suppression T72 has changed to the NOISY state with Dual X disabled.	Single Edge
24	**Noise Suppression NOISY State, Dual X On** Noise Suppression T72 has changed to the NOISY state with Dual X enabled.	Single Edge
25	**Noise Suppression VERY_NOISY State, Dual X Off** Noise Suppression T72 has changed to the VERY_NOISY state with Dual X disabled.	Single Edge
26	**Noise Suppression VERY_NOISY state, Dual X On** Noise Suppression T72 has changed to the VERY_NOISY state with Dual X enabled.	Single Edge
27	**Acquisition Antitouch Calibration Request** Acquisition Configuration T8 has requested a calibration due to an antitouch timeout.	Single Edge

Under NDA
© 2021 Microchip Technology Inc. and its subsidiaries

TABLE 6-50: EVENT ENUMERATIONS (CONTINUED)

ENUMID	Description	Event Type
28	**Timer 2 Elapsed** Instance 2 of the Timer T61 object has elapsed. See also event enumerations 13, 14 and 29 to 31	Single Edge
29	**Timer 3 Elapsed** Instance 3 of the Timer T61 object has elapsed. See also event enumerations 13, 14, 28, 30 and 31	Single Edge
30	**Timer 4 Elapsed** Instance 4 of the Timer T61 object has elapsed. See also event enumerations 13, 14, 28, 29 and 31	Single Edge
31	**Timer 5 Elapsed** Instance 5 of the Timer T61 object has elapsed. See also event enumerations 13, 14, and 28 to 30	Single Edge
38	**Touch Class Presence Detect (High) – Instance 0** Touch Event Trigger T79 Instance 0 has triggered a touch class presence event. See also event enumerations 97 to 102.	Single Edge
39	**Touch Class Presence Release (Low) – Instance 0** Touch Event Trigger T79 Instance 0 has triggered a touch class presence release event. See also event enumerations 97 to 102.	Single Edge
40	**Touch Class Presence Change (High or Low) – Instance 0** Touch Event Trigger T79 Instance 0 has triggered a touch class presence or touch class presence release event. See also event enumerations 97 to 102.	Double Edge
41	**System Calibration Command** A system calibration has occurred (that is, a calibration that has been invoked by the system not the user). See also event enumerations 1 and 2.	Single Edge
42	**GPIO0 High** GPIO0 has changed from low to high, or has been set to high on startup.	Single Edge
43	**GPIO0 Low** GPIO0 has changed from high to low.	Single Edge
44	**GPIO0 changed (High or Low)** GPIO0 has changed either from high to low or from low to high, or has been set to high on startup.	Double Edge
45	**GPIO1 High** GPIO1 has changed from low to high, or has been set to high on startup.	Single Edge
46	**GPIO1 Low** GPIO1 has changed from high to low.	Single Edge
47	**GPIO1 changed (High or Low)** GPIO1 has changed either from high to low or from low to high, or has been set to high on startup.	Double Edge
48	**GPIO2 High** GPIO2 has changed from low to high, or has been set to high on startup.	Single Edge
49	**GPIO2 Low** GPIO2 has changed from high to low.	Single Edge
50	**GPIO2 changed (High or Low)** GPIO2 has changed either from high to low or from low to high, or has been set to high on startup.	Double Edge
97	**Touch Class Presence Detect (High) – Instance 1** Touch Event Trigger T79 Instance 1 has triggered a touch class presence event. See also event enumerations See also event enumerations 38 to 40 and 100 to 102.	Single Edge

TABLE 6-50: EVENT ENUMERATIONS (CONTINUED)

ENUMID	Description	Event Type
98	**Touch Class Presence Release (Low) – Instance 1** Touch Event Trigger T79 Instance 1 has triggered a touch class presence release event. See also event enumerations See also event enumerations 38 to 40 and 100 to 102.	Single Edge
99	**Touch Class Presence Change (High or Low) – Instance 1** Touch Event Trigger T79 Instance 1 has triggered a touch class presence or touch class presence release event. See also event enumerations See also event enumerations 38 to 40 and 100 to 102.	Double Edge
100	**Touch Class Presence Detect (High) – Instance 2** Touch Event Trigger T79 Instance 2 has triggered a touch class presence event. See also event enumerations See also event enumerations 38 to 40, 97 to 99.	Single Edge
101	**Touch Class Presence Release (Low) – Instance 2** Touch Event Trigger T79 Instance 2 has triggered a touch class presence release event. See also event enumerations See also event enumerations 38 to 40, 97 to 99.	Single Edge
102	**Touch Class Presence Change (High or Low) – Instance 2** Touch Event Trigger T79 Instance 2 has triggered a touch class presence or touch class presence release event. See also event enumerations 38 to 40, 97 to 99.	Double Edge
111	**Moisture Processing Touch Detect (High)** Retransmission Compensation T80 moisture processing has detected a touch.	Single Edge
112	**Moisture Processing Touch Release (Low)** Retransmission Compensation T80 moisture processing has detected a touch release.	Single Edge
113	**Moisture Processing Touch Detect Change (High or Low)** Retransmission Compensation T80 moisture processing has detected a touch or a touch release.	Double Edge
114	**Moisture Processing Moisture Detect (High)** Retransmission Compensation T80 moisture processing has detected the presence of moisture.	Single Edge
115	**Moisture Processing Moisture Detect Stop (Low)** Retransmission Compensation T80 moisture processing no longer detects the presence of moisture.	Single Edge
116	**Moisture Processing Moisture Detect Change (High or Low)** Retransmission Compensation T80 moisture processing has changed to detecting the presence of moisture or no longer detects moisture.	Double Edge
117	**Moisture Processing Touch Detect in Moisture (High)** Retransmission Compensation T80 moisture processing has detected a touch in moisture.	Single Edge
118	**Moisture Processing Touch Release in Moisture (Low)** Retransmission Compensation T80 moisture processing has detected a touch release in moisture.	Single Edge
119	**Moisture Processing Touch in Moisture Detect Change (High or Low)** Retransmission Compensation T80 moisture processing has detected a touch or a touch release in moisture.	Double Edge
120	**Moisture Processing Degraded Mode (High)** Retransmission Compensation T80 moisture processing has changed from normal mode to degraded mode.	Single Edge
121	**Moisture Processing Degraded Mode (Low)** Retransmission Compensation T80 moisture processing has changed from degraded mode to normal mode.	Single Edge
122	**Moisture Processing Degraded Mode Change (High or Low)** Retransmission Compensation T80 moisture processing has changed either from normal mode to degraded mode or from degraded mode to normal mode.	Double Edge

Under NDA
© 2021 Microchip Technology Inc. and its subsidiaries

TABLE 6-50: EVENT ENUMERATIONS (CONTINUED)

ENUMID	Description	Event Type
132	**Retransmission Compensation (High)** Retransmission Compensation T80 retransmission compensation is actively reconstructing touches.	Single Edge
133	**Retransmission Compensation (Low)** Retransmission Compensation T80 retransmission compensation is no longer actively reconstructing touches.	Single Edge
134	**Retransmission Compensation (High or Low)** Retransmission Compensation T80 retransmission compensation has changed to either actively reconstructing touches or to no longer actively reconstructing touches.	Double Edge
159	**Lens Bending T65 Excessive Force Detect (High) – Instance 0** The force measured in Lens Bending T65 Instance 0 has grown above the active excessive force threshold.	Single Edge
160	**Lens Bending T65 Excessive Force Release (Low) – Instance 0** Force measured in Lens Bending T65 Instance 0 has dropped below the active excessive force threshold.	Single Edge
161	**Lens Bending T65 Excessive Force Change (High or Low) – Instance 0** Force measured in Lens Bending T65 Instance 0 has grown above or has dropped below the active excessive force threshold.	Double Edge
162	**Lens Bending T65 Excessive Force Detect (High) – Instance 1** Force measured in Lens Bending T65 Instance 1 has grown above the active excessive force threshold.	Single Edge
163	**Lens Bending T65 Excessive Force Release (Low) – Instance 1** Force measured in Lens Bending T65 Instance 1 has dropped below the active excessive force threshold.	Single Edge
164	**Lens Bending T65 Excessive Force Change (High or Low) – Instance 1** Force measured in Lens Bending T65 Instance 1 has grown above or has dropped below the active excessive force threshold.	Double Edge
165	**Lens Bending T65 Excessive Force Detect (High) – Instance 2** The force measured in Lens Bending T65 Instance 2 has grown above the active excessive force threshold.	Single Edge
166	**Lens Bending T65 Excessive Force Release (Low) – Instance 2** Force measured in Lens Bending T65 Instance 2 has dropped below the active excessive force threshold.	Single Edge
167	**Lens Bending T65 Excessive Force Change (High or Low) – Instance 2** Force measured in Lens Bending T65 Instance 2 has grown above or has dropped below the active excessive force threshold.	Double Edge
195	**Pulse Synchronization Lost (High)** Pulse sync has been lost	Single Edge
196	**Pulse Synchronization Regained (Low)** Pulse sync has been regained (note that pulse synchronization has to be lost first before it can be regained)	Single Edge
197	**Pulse Synchronization Lost or Regained (High or Low)** Pulse sync has been lost or regained (note that pulse synchronization has to be lost first before it can be regained)	Double Edge
198	**System Reset Occurred** A system reset has occurred	Single Edge
199	**First Scan Completed** The first scan after a system reset has completed.	Single Edge
310	**Power OK (High)** The power state has changed to the OK state. Note that this event is triggered only if Power Configuration T7 CFG2 DISPOWMON = 0 (see Power Configuration T7 “CFG2 Field”)	Single Edge

TABLE 6-50: EVENT ENUMERATIONS (CONTINUED)

ENUMID	Description	Event Type
329	**Key Entering Detect (High) – Key Array T15** Key Array T15 has detected a touch on any key in the key array, assuming normal reporting rules (for example, AKS rules may cause a key to be ignored).	Single Edge
330	**Key Leaving Detect (Low)– Key Array T15** Key Array T15 has detected a release on any key in the key array.	Single Edge
331	**Key Entering or Leaving Detect (High or Low) – Key Array T15** Key Array T15 has detected a touch or release on any key in the key array.	Double Edge
335	**Key Array T15 Transitioned Into Detect (High)** Key Array T15 has detected the first key touch in the key array, assuming normal reporting rules (for example, AKS rules may cause a key to be ignored). Subsequent key touches will not trigger this event again until all touches have been released, but they can trigger Event IDs 329 and 331.	Single Edge
336	**Key Array T15 Transitioned Out of Detect (Low)** Key Array T15 has released the last key touch in the key array.	Single Edge
337	**Key Array T15 Transitioned Into or Out of Detect (High or Low)** Key Array T15 has detected the first key touch in the key array or has released the last key touch in the key array.	Double Edge
All other values	Reserved; undefined behavior may occur	–

OBJTYPE Field

This field specifies the Object Type of the object for which the configuration settings should be replaced when a specified event is triggered. This is the number after the “T” suffix at the end of the object’s name.

Range: Depends on the objects available (see Section 1.6.2 “Object Instances”)

OBJINST FIELD

This field specifies the Object Instance for which the configuration settings should be replaced when a specified event is triggered.

Range: 0 to number of object instances minus 1 (see Section 1.6.2 “Object Instances”)

DSTOFFSET

This field specifies the starting Destination Offset of the first byte in the configuration area for the object for which the settings should be replaced. For example, to start at the GAIN field of the Multiple Touch Touchscreen T100 object, DSTOFFSET would be 28.

Range: 0 to object size minus 1

SRCOFFSET Field

This field specifies the starting Source Offset in the Dynamic Configuration Container T71 object from which to copy the source data. The maximum offset is Dynamic Configuration Container T71 size – 1 (see Section 6.11 “Dynamic Configuration Container T71 Object”).

Range: 0 to Dynamic Configuration Container T71 size – 1

LENGTH Field

This field specifies the number of bytes to be written minus 1 (that is, the length of the data as an offset to the last byte).

Range: 0 to (Destination Object Size – Destination Object Offset) – 1

6.10.3 CONFIGURATION CHECKS

A Dynamic Configuration Controller T70 object causes a configuration check in the following circumstances:

- When the object is enabled
- If the object is enabled, when certain fields are changed

Under NDA
© 2021 Microchip Technology Inc. and its subsidiaries

A configuration check may determine that a configuration error has occurred (for example, a setting is outside of its allowed range). This is signaled to the host and the device is halted until the error has been corrected. To fix an error, check that all object settings are within their allowed limits, as stated in the field descriptions.

Note that a configuration error is generated in the following situation:

- If the OBJTYPE is equal to 6 (Command Processor T6) and DSTOFFSET is less than 2. That is, if the destination object is the Command Processor T6, and the destination is either the RESET or BACKUPNV fields, a configuration error results.

TABLE 6-51: CONFIGURATION CHECKS FOR DYNAMIC CONFIGURATION CONTROLLER T70 (SPT_DYNAMICCONFIGURATIONCONTROLLER_T70)

Field	Changing The Field Causes...		Effect of Configuration Checks On Field
	Configuration Check	Automatic Recalibration	
CTRL	Yes	No	None
EVENT	Yes	No	None
OBJTYPE	Yes	No	Error if specified object type is not supported.
OBJINST	Yes	No	Error if specified object instance is out of range.
DSTOFFSET	Yes	No	Error if specified destination offset is out of range of the specific object configuration area, or OBJTYPE = 6 and DSTOFFSET < 2.
SRCOFFSET	Yes	No	Error if specified address is out of range as defined in the Dynamic Configuration Container T71 object.
LENGTH	Yes	No	Error if specified LENGTH with DSTOFFSET is out of range, or LENGTH with SRCOFFSET is out of range.

6.10.4 MESSAGES

The message data for a Dynamic Configuration Controller T70 object is shown in Table 6-52.

TABLE 6-52: MESSAGE DATA FOR DYNAMIC CONFIGURATION CONTROLLER T70 (SPT_DYNAMICCONFIGURATIONCONTROLLER_T70)

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	STATUS	Reserved						EVENTEDGE	HANDLED

STATUS Field

HANDLED: Indicates that an event has occurred and has been handled.

EVENTEDGE: Indicates a single-edge triggered event or the rising edge of a double-edge triggered event when set to 1, or indicates a falling edge of a double-edge triggered event when set to 0.

6.11 Dynamic Configuration Container T71 Object

6.11.1 INTRODUCTION

The Dynamic Configuration Container T71 object provides an area on the device for the user to store their own sets of configuration data that can be copied to other objects during run-time. The configurations are copied based on different events, as defined in the Dynamic Configuration Controller T70 object.

6.11.2 CONFIGURATION

TABLE 6-53: CONFIGURATION FOR DYNAMIC CONFIGURATION CONTAINER T71 (SPT_DYNAMICCONFIGURATIONCONTAINER_T71)

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0 – (*n*–1)	DATA[]	Configuration Data							

Note: *n* is the number of bytes for the Dynamic Configuration Container T71 object (*n* = 200 in mXT640UD-CCUBHA1 Family)

DATA[] Field

This field contains the configuration data. See Section 6.10 “Dynamic Configuration Controller T70 Object” for details on how this data is used.

Under NDA
© 2021 Microchip Technology Inc. and its subsidiaries

6.12 Touch Event Trigger T79 Object

6.12.1 INTRODUCTION

Each instance of the Touch Event Trigger T79 object configures a touch class trigger for use with a Dynamic Configuration Controller T70 object. It operates on the data received from the following objects:

- Multiple Touch Touchscreen T100
- Touch Suppression T42
- Passive Stylus T47
- Glove Detection T78

A Touch Event Trigger T79 object provides an interface that can be configured to trigger the following events based on the touch class:

- A positive edge Dynamic Configuration Controller T70 event on the detection of the touch class specified in the TOUCHCLASS field.
- A negative edge Dynamic Configuration Controller T70 event when the touch class specified in the TOUCHCLASS field is released and the timeout specified in the TCHRELTO field has elapsed.

On the mXT640UD-CCUBHA1 Family there are 3 instances of the Touch Event Trigger T79 object. These operate on touch data from the Multiple Touch Touchscreen T100 object, or an associated signal processing object, as configured by the TOUCHCLASS field.

6.12.2 CONFIGURATION

TABLE 6-54: CONFIGURATION FOR TOUCH EVENT TRIGGER T79 (SPT_TOUCHEVENTTRIGGER_T79)

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	CTRL	Reserved							ENABLE
1	TOUCHCLASS	Touch class trigger							
2	TCHRELTO	Touch release timeout							
3	NUMTCHTHR	Reserved			NUMTCHTHR				

CTRL Field

ENABLE: Enables or disables the use of this instance of the Touch Event Trigger T79 object. The instance is enabled if set to 1, and disabled if set to 0.

TOUCHCLASS Field

This field specifies the touch class that triggers the Dynamic Configuration Controller T70 events on detection or release. The Touch Event Trigger T79 object supports the touch classes listed in Table 6-55.

TABLE 6-55: TOUCH CLASSES

Code	Mode
1	Palm Touch (other touch classes might also be present)
4	Passive Stylus and Palm Touch
5	Passive Stylus
6	Passive Stylus or Palm Touch
10	Any Touch (including all types of touches)
12	Glove Touch (other touch classes might also be present)
13	Finger Touch (other touch classes might also be present)
14	Finger Touch Only (no other touch class present)
All other values	Reserved; undefined behavior may occur

TCHRELTO Field

This field specifies Touch Release Timeout. This is the timeout after which a Dynamic Configuration Controller T70 negative edge event is triggered on a touch release that has the touch class specified in TCHCLASS. The Touch Release Timeout is specified in units of 200 ms.

Range: 0 to 255

NUMTCHTHR Field

NUMTCHTHR: Specifies the Number of Touches Threshold. This is the minimum number of any touches, of which one must satisfy the selected TOUCHCLASS for this instance of the Touch Event Trigger T79 object, that will trigger the associated Dynamic Configuration Controller T70 event (see Table 6-55). The range for this field is 0 to the maximum number of touches (16 on mXT640UD-CCUBHA1 Family), where the default value of 0 means 1.

NUMTCHTHR Range: 0 (1), 1 to 16

Under NDA
© 2021 Microchip Technology Inc. and its subsidiaries

6.13 Auxiliary Touch Configuration T104 Object

6.13.1 INTRODUCTION

An Auxiliary Touch Configuration T104 object allows the setting of gain and thresholds for self capacitance measurements to generate auxiliary touch data for use by other objects.

Specifically the Auxiliary Touch Configuration T104 object generates self capacitance touch data, which is then used by the Retransmission Compensation T80 object to help with retransmission errors, water immunity and gloved touches.

See Appendix A "Measurement Processing" for more details on how the Auxiliary Touch Configuration T104 object operates with other objects.

6.13.2 CONFIGURATION

TABLE 6-56: CONFIGURATION FOR AUXILIARY TOUCH CONFIGURATION T104 (SPT_AUXTOUCHCONFIG_T104)

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	CTRL	Reserved		DEFMEASACTV	DEFMEASIDLE	Reserved			ENABLE
1	XGAIN	X axis gain							
2	XTCHTHR	X axis touch threshold							
3	XTCHHYST	X axis touch hysteresis							
4	XINTTHR	X axis internal threshold							
5	XINTHYST	X axis internal hysteresis							
6	YGAIN	Y axis gain							
7	YTCHTHR	Y axis touch threshold							
8	YTCHHYST	Y axis touch hysteresis							
9	YINTTHR	Y axis internal threshold							
10	YINTHYST	Y axis internal hysteresis							

CTRL Field

ENABLE: Enables or disables the use of the Auxiliary Touch Configuration T104 object. The object is enabled if set to 1, and disabled if set to 0.

The MEASALLOW field in the Acquisition Configuration T8 object must be set appropriately to allow self capacitance touches in order for the Auxiliary Touch Configuration T104 to operate correctly.

> **NOTE** A tune should be performed using the Self Capacitance Global Configuration T109 object after enabling the Auxiliary Touch Configuration T104 object or modifying its configuration (see "CMD Field").

DEFMEASIDLE: If this bit is set to 1, self capacitance measurements are included in the Default Measurement types in Idle mode (see Section A.2.1 "Default Measurements").

DEFMEASACTV: If this bit is set to 1, self capacitance measurements are included in the Default Measurement types in Active mode.

XGAIN Field

This field sets the X Axis Gain. This is the gain of the analog circuits in front of the analog to digital converter (ADC) for X axis self capacitance measurements. The default range is 0 to 29.

The default range should be sufficient for most applications. However, using Gain × 2 mode may help improve moisture immunity performance. To enable Gain × 2 mode, set XGAIN bit 7 to 1. This will add additional settings in the range to 128 to 157.

For more information on choosing a suitable gain value, refer to *maXTouch U Series Tuning Guide*.

Range: 0 to 29, 128 to 157 (Gain × 2 mode)

XTCHTHR Field

This field specifies the X Axis Touch Threshold in 8-bit deltas. This is the threshold above which X axis nodes are marked as above threshold or in detect.

Range: 0 to 255

Typical: 80

XTCHHYST Field

This field specifies the X Axis Touch Hysteresis in 8-bit deltas. This is used to specify a difference in threshold which would cause nodes in detect (based upon XTCHTHR) to go out of detect.

XTCHHYST is capped internally to (XTCHTHR × 0.75). A hysteresis greater than 75% of the X Axis Touch Threshold is therefore not possible. Note that, under normal circumstances, it is not recommended to set the hysteresis to more than 25% of the X Axis Touch Threshold (XTCHTHR), as this can cause potential problems during touch lift off, double taps, and noise handling.

A value of 0 means there is no hysteresis.

Range: 0 (no hysteresis), 1 to XTCHTHR × 0.75

Typical: 20

XINTTHR Field

This field specifies the X Axis Internal Threshold in 8-bit deltas. This is the threshold above which X axis nodes are marked as above the internal threshold or in detect at a lower threshold.

Range: 0 to 255

Typical: 20

XINTHYST Field

This field specifies the X Axis Internal Hysteresis in 8-bit deltas. This is used to specify a difference in threshold which would cause nodes in detect (based upon XINTTHR) to go out of detect.

XINTHYST is capped internally to (XINTTHR × 0.75). A hysteresis greater than 75% of the X Axis Internal Threshold is therefore not possible. Note that, under normal circumstances, it is not recommended to set the hysteresis to more than 25% of the X Axis Internal Threshold (XINTTHR), as this can cause potential problems during touch lift off, double taps, and noise handling.

A value of 0 means there is no hysteresis.

Range: 0 (no hysteresis), 1 to XINTTHR × 0.75

Typical: 5

YGAIN Field

This field sets the Y Axis Gain. This is the gain of the analog circuits in front of the analog to digital converter (ADC) for Y axis self capacitance measurements. The default range is 0 to 29.

The default range should be sufficient for most applications. However, using Gain × 2 mode may help improve moisture immunity performance. To enable Gain × 2 mode, set YGAIN bit 7 to 1. This will add additional settings in the range to 128 to 157.

For more information on choosing a suitable gain value, refer to *maXTouch U Series Tuning Guide*.

Range: 0 to 29, 128 to 157 (Gain × 2 mode)

YTCHTHR Field

This field specifies the Y Axis Touch Threshold in 8-bit deltas. This is the threshold above which Y axis nodes are marked as above threshold or in detect.

Range: 0 to 255

Typical: 80

Under NDA
© 2021 Microchip Technology Inc. and its subsidiaries

YTCHHYST Field

This field specifies the Y Axis Touch Hysteresis in 8-bit deltas. This is used to specify a difference in threshold which would cause nodes in detect (based upon YTCHTHR) to go out of detect.

YTCHHYST is capped internally to (YTCHTHR × 0.75). A hysteresis greater than 75% of the Y Axis Touch Threshold is therefore not possible. Note that, under normal circumstances, it is not recommended to set the hysteresis to more than 25% of the Y Axis Touch Threshold (YTCHTHR), as this can cause potential problems during touch lift off, double taps, and noise handling.

A value of 0 means there is no hysteresis.

Range: 0 (no hysteresis), 1 to YTCHTHR × 0.75

Typical: 20

YINTTHR Field

This field specifies the Y Axis Internal Threshold in 8-bit deltas. This is the threshold above which Y axis nodes are marked as above the internal threshold or in detect at a lower threshold.

Range: 0 to 255

Typical: 20

YINTHYST Field

This field specifies the Y axis Internal Hysteresis in 8-bit deltas. This is used to specify a difference in threshold which would cause nodes in detect (based upon YINTTHR) to go out of detect.

YINTHYST is capped internally to (YINTTHR × 0.75). A hysteresis greater than 75% of the Y Axis Internal Threshold is therefore not possible. Note that, under normal circumstances, it is not recommended to set the hysteresis to more than 25% of the Y Axis Internal Threshold (YINTTHR), as this can cause potential problems during touch lift off, double taps, and noise handling.

A value of 0 means there is no hysteresis.

Range: 0 (no hysteresis), 1 to YINTTHR × 0.75

Typical: 5

6.13.3 CONFIGURATION CHECKS

The Auxiliary Touch Configuration T104 object causes a configuration check to be performed in the following circumstances:

- When the object is enabled (that is, the ENABLE bit is set in the CTRL field)
- If the object is enabled, when certain fields are changed

In addition, some fields may cause an automatic recalibration to be performed.

A configuration check may determine that a configuration error has occurred (for example, if a setting is set outside of its allowed range or a conflict has occurred between two settings). This is signaled to the host (see Section 3.3 “Command Processor T6 Object”). The device halts until the error has been corrected. To fix the error, check that all the object settings are within their allowed limits, as stated in the field descriptions.

TABLE 6-57: CONFIGURATION CHECKS FOR AUXILIARY TOUCH CONFIGURATION T104 (SPT_AUXTOUCHCONFIG_T104)

Field	Changing The Field Causes...		Effect of Configuration Checks On Field
	Configuration Check	Automatic Recalibration	
CTRL	Yes	Yes (1)	None
XGAIN	Yes	Yes	None
XTCHTHR	Yes	No	None
XTCHHYST	Yes	No	None
XINTTHR	Yes	No	None
XINTHYST	Yes	No	None

TABLE 6-57: CONFIGURATION CHECKS FOR AUXILIARY TOUCH CONFIGURATION T104 (SPT_AUXTOUCHCONFIG_T104)

Field	Changing The Field Causes...		Effect of Configuration Checks On Field
	Configuration Check	Automatic Recalibration	
YGAIN	Yes	Yes	None
YTCHTHR	Yes	No	None
YTCHHYST	Yes	No	None
YINTTHR	Yes	No	None
YINTHYST	Yes	No	None

Note 1: If the ENABLE bit is toggled on or off.

Under NDA
© 2021 Microchip Technology Inc. and its subsidiaries

6.14 Self Capacitance Global Configuration T109 Object

6.14.1 INTRODUCTION

The Self Capacitance Global Configuration T109 object provides configuration for self capacitance measurements employed on the device.

6.14.2 CONFIGURATION

TABLE 6-58: CONFIGURATION FOR SELF CAPACITANCE GLOBAL CONFIGURATION T109 (SPT_SELFCAPGLOBALCONFIG_T109)

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	CTRL	DISMEASY	DISMEASX	Reserved		DISPRECHARGE	SNGLENDEN	RPTEN	Reserved
1	Reserved	Reserved							
2	CMDONRESET	Command performed on reset							
3	CMD	Command							
4	Reserved	Reserved							
5	Reserved	Reserved							
6	Reserved	Reserved							
7	Reserved	Reserved							
8	TUNECFG	Reserved					EXTRATUNITER		
9	TUNPARAM[0]	Host side tuning parameters 0 LSByte							
10		Host side tuning parameters 0 MSByte							
11	TUNPARAM[1]	Host side tuning parameters 1 LSByte							
12		Host side tuning parameters 1 MSByte							
13	TUNPARAM[2]	Host side tuning parameters 2 LSByte							
14		Host side tuning parameters 2 MSByte							
15	TUNPARAM[3]	Host side tuning parameters 3 LSByte							
16		Host side tuning parameters 3 MSByte							

CTRL Field

RPTEN: Allows the object to send status messages to the host through the Message Processor T5 object. Reporting is enabled if set to 1, and disabled if set to 0.

SNGLENDEN: Enables Single Ended Measurement.

> **NOTE** This control is for use only during design, under the guidance of Microchip field engineers, to help tune the product for characterization. It should not be used in production. This control should be left at its default value of 0, unless advised otherwise by Microchip.

DISPRECHARGE: Disables Pre-charge. If this bit is set to 1, it disables the pre-charge to AVDD (rising edge) or GND (falling edge) before the integration phase of a self capacitance measurement pulse. This applies to all self capacitance measurements.

DISMEASX and DISMEASY: Disable Self Capacitance Measurements on the X or Y axis. If one of these bits is set to 1, all self capacitance measurements on the corresponding axis are disabled. If one of these bits is set to 0, all self capacitance measurements on the corresponding axis can be made. It is not recommended to disable measurements on both axes (that is DISMEASX = 1 and DISMEASY = 1); one axis should always remain enabled.

> **NOTE** DISMEASX must be set to 0 (that is, the X axis measurement must not be disabled) if the Driven Shield Signal Limit test is in use (see Self Test Signal Limits T12 “CTRL Field”).

Note that the ability to disable self capacitance measurements on the X or Y axis depends on other configuration settings:

- Acquisition Configuration T8: MEASALLOW (see Acquisition Configuration T8 “MEASALLOW Field”)
- Auxiliary Touch Configuration T104: CTRL ENABLE (see Auxiliary Touch Configuration T104 “CTRL Field”)
- Self Capacitance Measurement Configuration T113: CTRL ENABLE and CTRL ALTAXISEN (see Self Capacitance Measurement Configuration T113 “CTRL Field”)

In particular, setting DISMEASX and DISMEASY should be done in conjunction with the Self Capacitance Measurement Configuration T113 CTRL ALTAXISEN control.

CMDONRESET Field

This field schedules the given command on reset. See Table 6-59 for command values.

Range: See Table 6-59

CMD Field

This field schedules the given command.

The command field is cleared after the command has been actioned.

Range: See Table 6-59

TABLE 6-59: COMMANDS

Value	Command	Description
0	None	None.
1	Tune	Tunes the current self capacitance tuning parameters. The new settings are then used and the system performs a calibration. Self capacitance touch must be enabled for a tune to work (see Section 3.5 “Acquisition Configuration T8 Object” and Section 6.13 “Auxiliary Touch Configuration T104 Object”). In addition, appropriate measurements must be enabled for any associated parameters to be tuned (for example, in the Acquisition Configuration T8 and Self Capacitance Measurement Configuration T113 objects).
4	Store To Configuration RAM	Stores the current self capacitance tuning parameters into the Self Capacitance Tuning Parameters T110 object.
5	Apply From Configuration RAM	Applies the current self capacitance tuning parameters from the Self Capacitance Tuning Parameters T110 object and causes a system calibration.
7	Refresh Configuration RAM	Refreshes the Self Capacitance Tuning Parameters T110 object from NVM backup. Note that this area of NVM is inside the configuration backup region.
All other values	Reserved	

Note: NVM (non volatile memory) is outside the configuration backup region.

TUNECFG Field

This field provides additional tuning configurations.

EXTRATUNITER: Configures Extra Tuning Iterations. The range for this control is 0 to 7, where 0 means that no extra tuning iterations are required.

Range: 0 (disabled), 1 to 7

TUNPARAM[] Fields

These fields provide host side tuning. On the mXT640UD-CCUBHA1 Family the TUNPARAM[] fields have the format shown in Table 6-60.

TABLE 6-60: FORMAT OF TUNPARAM[] FIELDS

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
9	TUNPARAM[0]	VREFLEVELSEL		Reserved	IBUFPWR				
10		Reserved					IAMPBIAS		

Under NDA
© 2021 Microchip Technology Inc. and its subsidiaries

TABLE 6-60: FORMAT OF TUNPARAM[] FIELDS (CONTINUED)

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
11	TUNPARAM[1]	Reserved							
12		Reserved							
13	TUNPARAM[2]	Reserved							
14		Reserved							
15	TUNPARAM[3]	Reserved							
16		Reserved							

TUNPARAM[0] Field

IBUFPWR: Configures the Input Buffer Power. A higher value indicates a higher buffer strength.

IBUFPWR Range: 0 (5), 1 to 16

VREFLEVELSEL: Specifies the Voltage Reference Level Selection.

VREFLEVELSEL Range: See Table 6-61

TABLE 6-61: VREFLEVELSEL

Value	Voltage Reference in Voltage Modulation Mode
0 (default)	(AVdd / 2) ± 23% AVdd
1	(AVdd / 2) ± 25% AVdd
2	(AVdd / 2) ± 27% AVdd
3	(AVdd / 2) ± 30% AVdd

IAMPBIAS: Configures the Input Amplifier Bias. A higher value indicates a higher bias.

IAMPBIAS Range: 0 (7), 1 to 7

6.14.3 CONFIGURATION CHECKS

A Self Capacitance Global Configuration T109 object causes a configuration check to be performed in the following circumstances:

- When certain fields are changed

In addition, some fields may cause an automatic recalibration to be performed.

A configuration check may determine that a configuration error has occurred (for example, if a setting is set outside of its allowed range or a conflict has occurred between two settings). This is signaled to the host (see Section 3.3 “Command Processor T6 Object”). The device halts until the error has been corrected. To fix the error, check that all the object settings are within their allowed limits, as stated in the field descriptions. While the object is disabled, however, potential configuration errors are allowed.

TABLE 6-62: CONFIGURATION CHECKS FOR SELF CAPACITANCE GLOBAL CONFIGURATION T109 (SPT_SELFCAPGLOBALCONFIG_T109)

Field	Changing The Field Causes...		Effect of Configuration Checks On Field
	Configuration Check	Automatic Recalibration	
CTRL	No	Yes	None
CMDONRESET	No	No	None
CMD	No	No	None
TUNECFG	No	No	None
TUNPARAM[]	No	Yes	None

6.14.4 MESSAGES

TABLE 6-63: MESSAGE DATA FOR SELF CAPACITANCE GLOBAL CONFIGURATION T109 (SPT_SELFCAPGLOBALCONFIG_T109)

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	CMD	Command processed							
2	ERRORCODE	Error code							

CMD Field

This field reports the command that was processed and has caused this message (see "CMDONRESET Field" and "CMD Field").

ERRORCODE Field

This field reports the success/error code. An error code of 0 means that there is no error (success). Otherwise, a non-zero indicates the error resulting from the command that caused this message (see Table 6-64).

TABLE 6-64: ERROR CODES

Command	Error Codes	Order (1)
1 – Tune	0x04 – Multiple Touch Touchscreen T100 and/or Auxiliary Touch Configuration T104 not enabled or self capacitance touch measurements not allowed in Acquisition Configuration T8.	1
	0x01 – Self capacitance touch measurements tuning initialization failed because the signals saturated. Indicates a problem with the sensor design.	2
	0x02 – Self capacitance measurements tuning process could not complete successfully due to excessive loading of the sensor used. May indicate a problem with the sensor.	3
	0x05 – Self Capacitance Measurement Configuration T113 not enabled or single-ended self capacitance measurements not allowed in Acquisition Configuration T8. Tuning of single-ended self capacitance measurements has been excluded.	4
	0x03 – Single-ended self capacitance measurements tuning process was successful but single-ended self capacitance tuning was out of range. Further tuning may be possible. Refer to the *maXTouch U Series Tuning Guide* for more details.	5
	0 – Tuning successful	6

Note 1: Tune command error codes are listed in order of precedence: high (1) to low (6). Error codes for other commands have no order of precedence.

The error codes are generated in the order shown in Table 6-64 as the tuning process proceeds. This means that higher precedence errors may mask lower order errors. For example, an error code of 0x04 may mask one or more of the lower order errors (0x01, 0x02, 0x05 and/or 0x03 in order).

Under NDA
© 2021 Microchip Technology Inc. and its subsidiaries

6.15 Self Capacitance Tuning Parameters T110 Object

6.15.1 INTRODUCTION

NOTE	This object is for use only under the guidance of Microchip field engineers. This object should be left with its default values unless advised otherwise by Microchip.

The Self Capacitance Tuning Parameters T110 object provides configuration space for a generic set of settings for self capacitance measurements employed on the device. It has a size equivalent to 40 bytes on the mXT640UD-CCUBHA1 Family.

On the mXT640UD-CCUBHA1 Family there are 12 instances of the Self Capacitance Tuning Parameters T110 object (see Table 6-65).

TABLE 6-65: SELF CAPACITANCE TUNING PARAMETERS T110 INSTANCES

Instance	Purpose
Instance 0	Self capacitance touch parameters for Tuning Phase 0 – All Y lines
Instance 1	Self capacitance touch parameters for Tuning Phase 0 – All even-numbered X lines
Instance 2	Self capacitance touch parameters for Tuning Phase 0 – All odd-numbered X lines
Instance 3	Self capacitance touch parameters for Tuning Phase 1 – All Y lines (1)
Instance 4	Self capacitance touch parameters for Tuning Phase 1 – All even-numbered X lines (1)
Instance 5	Self capacitance touch parameters for Tuning Phase 1 – All odd-numbered X lines (1)
Instance 6	Self capacitance touch parameters for Tuning Phase 2 – All Y lines (1)
Instance 7	Self capacitance touch parameters for Tuning Phase 2 – All even-numbered X lines (1)
Instance 8	Self capacitance touch parameters for Tuning Phase 2 – All odd-numbered X lines (1)
Instance 9	Self capacitance single-ended parameters – All Y lines
Instance 10	Self capacitance single-ended parameters – All even-numbered X lines
Instance 11	Self capacitance single-ended parameters – All odd-numbered X lines

Note 1: If Self Capacitance Configuration T111 Instance 0 CTRL LFCOMPEN = 0, then the phase 0 parameters are copied to phases 1 and 2 after applying the single-ended part of the tuning process.

6.15.2 CONFIGURATION

TABLE 6-66: CONFIGURATION FOR SELF CAPACITANCE TUNING PARAMETERS T110 (SPT_SELFCAPTUNINGPARAMS_T110)

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	PARAMS[0]	Setting 0 LSByte							
1		Setting 0 MSByte							
...									
38	PARAMS[19]	Setting 19 LSByte							
39		Setting 19 MSByte							

Note: PARAMS[] stores settings for the lines for this instance. Unused bytes will be reserved.

PARAMS[] Fields

These fields specify the generic settings.

Range: 0 to 65535

6.16 Self Capacitance Configuration T111 Object

6.16.1 INTRODUCTION

The Self Capacitance Configuration T111 object provides configuration for self capacitance measurements (see Appendix A "Measurement Processing"), including pulse synchronization during self capacitance measurements.

On the mXT640UD-CCUBHA1 Family there are 2 instances of the Self Capacitance Configuration T111 object (see Table 6-67 for details).

TABLE 6-67: SELF CAPACITANCE CONFIGURATION T111 INSTANCES

Instance	Purpose
Instance 0	Used to configure self capacitance touch measurements. Settings for this instance use a subset of the configuration fields (see Table 6-68 on page 239).
Instance 1	Used to configure self capacitance single-ended measurements. Settings for this instance use a subset of the configuration fields (see Table 6-69 on page 240).

Note: Each instance configures a particular measurement type and the settings for that instance are applied to all measurements of that type for use by other objects. Therefore, the controls for each instance need to be tuned to values that are appropriate across all touch objects that use those measurements.

Self capacitance measurements are required for single touch measurements when self capacitance measurements are selected as the primary measurement type. Self capacitance data is also used to augment the mutual capacitance data to help cope with retransmission and moisture issues (see Section 5.13 "Retransmission Compensation T80 Object").

See Appendix A "Measurement Processing" for more details on how the Self Capacitance Configuration T111 object operates with other objects.

6.16.1.1 Self Capacitance Pulses

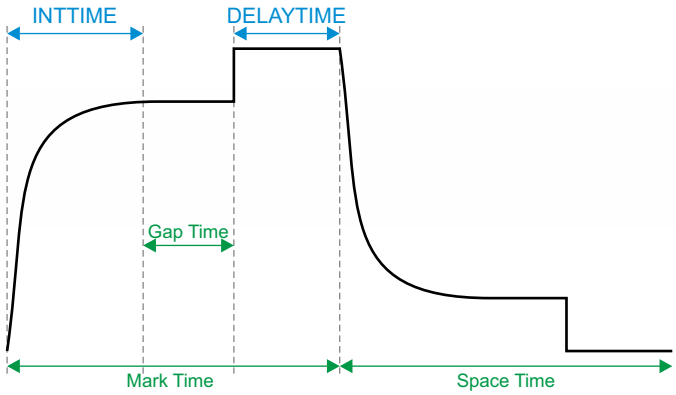
The pulse for a self capacitance measurement is outlined in Figure 6-7.

With reference to Figure 6-7, the Integration Time is specified by INTTIME (see "INTTIME Field"). The Delay Time is the additional time by which a pulse is extended after the integration time has completed, and is specified by DELAYTIME (see "DELAYTIME Field"). The Gap Time is the extra time provided by the associated Self Capacitance Noise Suppression T108 object.

These three parameters are used to derive the Mark Time and the Space Time. The Mark Time is the sum of the Integration Time, the Gap Time, and the Delay Time. The Space Time is equal to the Mark Time.

When there is no pulse synchronization, the length of the pulse is simply the Mark Time plus the Space Time. When pulse synchronization is applied, however, the timing of the pulses is determined by the frequency of the synchronization pulses. Consequently, the observed Delay Time is extended to the next synchronization pulse (see Figure 6-8).

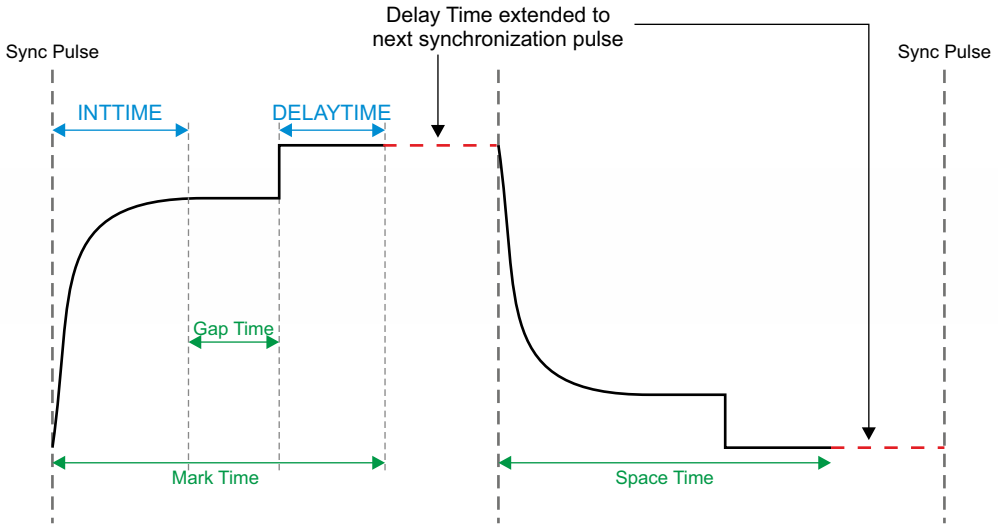
FIGURE 6-7: SELF CAPACITANCE MEASUREMENT PULSE – NO PULSE SYNCHRONIZATION



NOTE: The pulse shape is shown for guidance only. The shape may vary between devices, although the timing principles are the same.

Under NDA
© 2021 Microchip Technology Inc. and its subsidiaries

FIGURE 6-8: SELF CAPACITANCE MEASUREMENT PULSE – WITH PULSE SYNCHRONIZATION



6.16.1.2 Analog-to-Digital Conversions

The matrix axes are sampled one at a time (depending on the hardware there may be more than one burst per axis), with at least 2 analog-to-digital conversions (ADCs) performed on each axis in turn. Note that on the mXT640UD-CCUBHA1 Family, the X axis is scanned twice: first the odd numbered X lines are scanned and then the even numbered X lines.

The number of ADCs per X line is specified by the ACTVSYNCSPERL/IDLESYNCSPERL and ADCSPERSYNC fields.

The number of ADCs per line is a nominal value only. Other objects can influence the ADCs per line setting under specific circumstances. Specifically, if Self Capacitance Noise Suppression T108 is enabled, ADCs per line values from this object (selected based upon state) will be used for Self Capacitance Touch measurements instead of the T111 value. Note that Single-ended Self Capacitance measurements always use the Self Capacitance Configuration T111 value and are not subject to the Self Capacitance Noise Suppression T108 values.

NOTE	The minimum Active/Idle ADCs per line is saturated to 2. Therefore, if less than 2 ADCs per line is configured, 2 ADCs per line will be used.

6.16.1.3 Pulse Synchronization

IMPORTANT!	Pulse Synchronization functionality on the mXT640UD-CCUBHA1 Family is not considered UL/IEC 60730 Class B compliant. Therefore, the device should not be configured to use Pulse Synchronization in a UL/IEC 60730 Class B product.

NOTE	The SYNC line can be used for either pulse synchronization or frame synchronization (see "SYNC Field"), but not both. If the SYNC line is used for pulse synchronization functionality, then it cannot be used for frame synchronization.

The ADCSPERSYNC field determines how the ADCs are synchronized to the SYNC line (see Figure 6-9 on page 239):

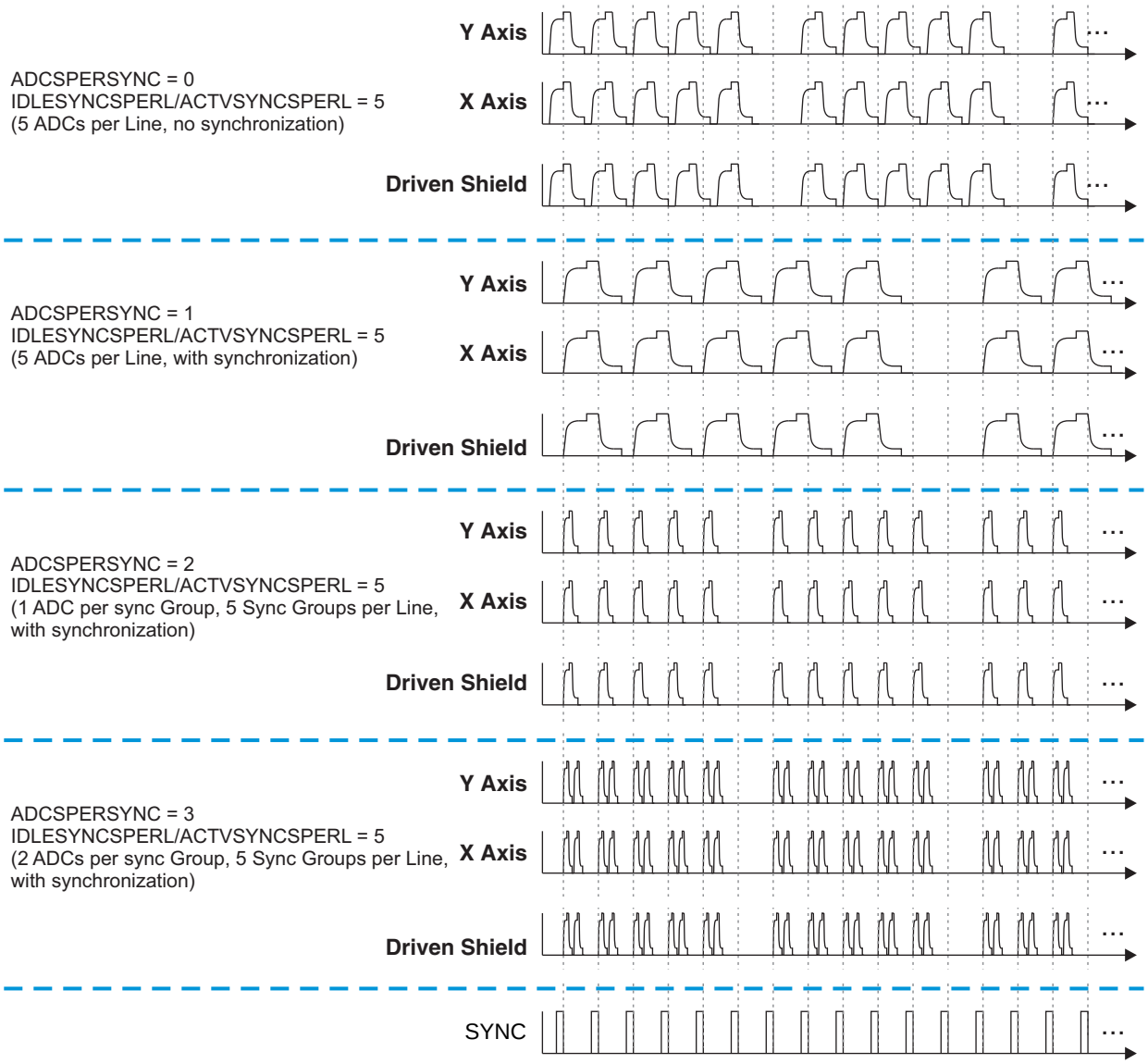
- If the ADCSPERSYNC field is set to zero, then no pulse synchronization occurs.
- If the ADCSPERSYNC field is set to 1, then pulse synchronization occurs. In this case, the rising and falling edges of each ADC are synchronized to the falling edge of the SYNC signal.
- If the ADCSPERSYNC field is set to a value of 2 or above, the ADCs are arranged in "Sync Groups". The number of ADCs per Sync Group is (ADCSPERSYNC – 1). In this case, the ACTVSYNCSPERL and IDLESYNCSPERL fields determine the number of Sync Groups per X line. Thus, the total number of ADCs per X is equivalent to:
- (ADCSPERSYNC – 1) × ACTVSYNCSPERL or IDLESYNCSPERL
- with a minimum value of 2.

Note the following:

- Pulse synchronization for Self capacitance measurements is enabled for each measurement type independently, according to which Self Capacitance Configuration T111 instances are configured. To enable pulse synchronization on mutual capacitance measurements, see Section 6.8 “CTE Configuration T46 Object”.
- If pulse synchronization is turned on (that is, ADCSPERSYNC is non-zero), GPIO0 cannot be configured by the GPIO Configuration T19 object. GPIO0 must be set as an input, or the GPIO Configuration T19 object disabled, for the pulse synchronization to work.
- If there is a requirement to toggle pulse synchronization on and off at run time, the device must be configured and backed up with pulse synchronization enabled in either Self Capacitance Configuration T111 or CTE Configuration T46. This ensures that the device will start up with pulse synchronization enabled for at least one measurement type and the SYNC pin is latched for use. This prevents the pin from being used for other uses.
- If the system has been configured to use pulse synchronization and there is a failure to detect a SYNC signal, the default behavior of the system is to continue measurements unsynchronized using user-configured timing. If this behavior is not required, CTE Configuration T46 CFG SYNCLOSSMODE can be set to 1 to pause measurements altogether while pulse synchronization is absent (see CTE Configuration T46 “CFG Field”). When a SYNC signal is subsequently detected, pulse synchronization will be re-enabled and the measurements will continue with synchronized ADCs. Note that CTE Configuration T46 CFG SYNCLOSSMODE has an effect on all mutual and self capacitance measurements.

Under NDA
© 2021 Microchip Technology Inc. and its subsidiaries

FIGURE 6-9: SELF CAPACITANCE CONFIGURATION T111 PULSE SYNCHRONIZATION



Notes:

- Both axes are burst simultaneously but measurements are on one axis only
- The pulse shape is shown for guidance only. The shape may vary between devices, although the timing principles are the same.

6.16.2 CONFIGURATION

6.16.2.1 Instance 0 – Self Capacitance Touch Measurements

TABLE 6-68: CONFIGURATION FOR SELF CAPACITANCE CONFIGURATION T111 (SPT_SELFCAPCONFIG_T111)

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	CTRL	DISDCCORR	Reserved				LFCOMPEN	Reserved	
1	DBGCTRL	Reserved			DCEN	RAWDATAEN	SIGSEN	REFSEN	DELTASEN
2	INTTIME	Integration time							
3	DELAYTIME	Delay time							
4	IDLESYNCSPERL	Number of ADCs per sensor line when idle							

TABLE 6-68: CONFIGURATION FOR SELF CAPACITANCE CONFIGURATION T111 (SPT_SELFCAPCONFIG_T111) (CONTINUED)

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
5	ACTVSYNCSPERL	Number of ADCs per sensor line when active							
6	DRIFT	Drift interval							
7	DRIFTST	Drift suspend time							
8	Reserved	Reserved							
9	CALRECSTR	Calibration recovery strength							
10	Reserved	Reserved							
11	Reserved	Reserved							
12	Reserved	Reserved							
13	Reserved	Reserved							
14	Reserved	Reserved							
15	INRUSHCFG	Reserved	XRESISTOR			Reserved	YRESISTOR		
16	ALTINTTIMEX	Alternative X axis measurement integration time							
17	ALTDELAYTIMEX	Alternative X axis measurement delay time							
18	DCDRIFT	DC drift interval							
19	Reserved	Reserved							
20	DCFILTER	Reserved		SLEWACC	SLEWEN	IIRCOEF			IIREN
21	DCCALRECSTR	DC Calibration recovery strength							
22	DCCALERRRATIO	DC Calibration recovery error ratio							
23	SYNCDELAY	Delay from SYNC falling edge LSByte							
24		Delay from SYNC falling edge MSByte							
25	ADCSPERSYNC	Reserved		ADCSPERSYNC					
26	DCGAINSF	DC gain scaling factor							
27	DCTHRX	X axis DC Self capacitance threshold							
28	DCTHRY	Y axis DC Self capacitance threshold							
29	DCIDLESLEWMIN	Idle Mode Minimum DC Slew Rate							
30	DCPROCTHRX	DC Level Processing Threshold for X							
31	DCPROCTHRY	DC Level Processing Threshold for Y							

6.16.2.2 Instance 1 – Self Capacitance Single-ended Measurements

TABLE 6-69: CONFIGURATION FOR SELF CAPACITANCE CONFIGURATION T111 (SPT_SELFCAPCONFIG_T111)

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Reserved	Reserved							
1	DBGCTRL	Reserved					SIGSEN	REFSEN	DELTASEN
2	INTTIME	Integration time							
3	DELAYTIME	Delay time							
4	IDLESYNCSPERL	Number of ADCs per sensor line when idle							
5	ACTVSYNCSPERL	Number of ADCs per sensor line when active							
6	Reserved	Reserved							
7	Reserved	Reserved							
8	Reserved	Reserved							
9	Reserved	Reserved							
10	Reserved	Reserved							
11	Reserved	Reserved							
12	Reserved	Reserved							
13	Reserved	Reserved							
14	Reserved	Reserved							

Under NDA
© 2021 Microchip Technology Inc. and its subsidiaries

TABLE 6-69: CONFIGURATION FOR SELF CAPACITANCE CONFIGURATION T111 (SPT_SELFCAPCONFIG_T111) (CONTINUED)

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
15	INRUSHCFG	Reserved	XRESISTOR			Reserved	YRESISTOR		
16	Reserved	Reserved							
17	Reserved	Reserved							
18	Reserved	Reserved							
19	Reserved	Reserved							
20	Reserved	Reserved							
21	Reserved	Reserved							
22	Reserved	Reserved							
23	SYNCDELAY	Delay from SYNC falling edge LSByte							
24		Delay from SYNC falling edge MSByte							
25	ADCSPERSYNC	Reserved		ADCSPERSYNC					
26	Reserved	Reserved							
27	Reserved	Reserved							
28	Reserved	Reserved							
29	Reserved	Reserved							
30	Reserved	Reserved							
31	Reserved	Reserved							

CTRL Field

LFCOMPEN: (Touch data instance only) Enables Low Frequency Compensation.

If this bit is set to 1, low frequency compensation is performed on the measurement. In this case, the Self Capacitance Global Configuration T109 tuning procedure will perform 3 phases of tuning for the self capacitance differential (assuming that there is enough storage space within Self Capacitance Tuning Parameters T110). Note the following:

- The number of ADCs per line should be set to a multiple of 3 (see "IDLESYNCSPERL Field" and "ACTVSYNCSPERL Field"), but the device will constrain this internally if the ADCs per line is not set appropriately.
- An additional sample is added per scan, so the amount of pulses seen is:

  ((ADCs_per_line / 3) × 3) + 1

If this bit is set to 0, low frequency compensation is not applied to the measurement and the tuning procedure will run as normal. In this case, the Self Capacitance Global Configuration T109 tuning procedure will perform only a single phase of tuning.

NOTE	If the Driven Shield Signal Limits Test is enabled for use (Self Test Signal Limits T12 CTRL DSTESTEN = 1), low frequency compensation cannot be used. Low frequency compensation must therefore be disabled by setting LFCOMPEN to 0 if the Driven Shield Signal Limits Test is enabled.

DISDCCORR: (Touch data instance only) If this bit is set to 0, the self capacitance single-ended measurement data is used to correct the DC level of the self capacitance touch data in the case when it appears that the self capacitance touch DC level is incorrectly too high and might therefore create false touches. If this bit is set to 1, this DC correction is disabled. Disabling DC Correction should only be used in cases where the self capacitance single-ended data is unreliable for this purpose.

NOTE	To correct the DC level of the self capacitance touch data, one of the following two options should be used: • Use the Self Capacitance Single-ended measurement data by setting DISDCCORR = 0 • Use the Lens Bending T65 Lens Bending Self Capacitance Instances (that is, enable and configure Lens Bending T65 Instances 1 and 2) Note that these two options should not be used together as the combined behavior is undefined. In addition, if neither option is used, the DC level will be poor.

DBGCTRL Field

This field allows different debug data types to be enabled or disabled for use with the hardware debug interface. Note that the Command Processor T6 DEBUGCTRL2 DBGOBJMODEEN bit must be set to 1 for this to work. See "DEBUGCTRL2 Field" for more information on the debug data for the hardware debug interface.

DELTASEN: (Self Capacitance Touch and Single-ended data instances only) If this bit is set to 1, self capacitance deltas are output by the hardware debug interface.

REFSEN: (Self Capacitance Touch and Single-ended data instances only) If this bit is set to 1, self capacitance references are output by the hardware debug interface.

SIGSEN: (Self Capacitance Touch and Single-ended data instances only) If this bit is set to 1, self capacitance signals are output by the hardware debug interface.

RAWDATAEN: (Self Capacitance Touch data instance only) If this bit is set to 1, raw data values are output by the hardware debug interface in addition to the processed data.

DCEN: (Self Capacitance Touch data instance only) If this bit is set to 1, DC debug data is output by the hardware debug interface.

INTTIME Field

This field specifies the Integration Time in 24 MHz clock cycles (41.67 ns), where the default value of 0 means 36 (1.5 µs).

Range: 0 (36), 1 to 255

DELAYTIME Field

This field specifies the Delay Time in 24 MHz clock cycles (41.67 ns). This is the length of additional time that a pulse is extended by after the integration time has completed. The default value of 0 means 36 (1.5 µs).

Range: 0 (36), 1 to 255

IDLESYNCSPERL Field

This field determines the number of ADCs per Sensor Line in Idle mode. The range for this field is 0 to 255, where a value of 0 means 8.

See Section 6.16.1 "Introduction" for more information on how this field helps determine the actual number of ADCs per Line

Range: 0 (8), 1 to 255

ACTVSYNCSPERL Field

This field determines the number of ADCs per Sensor Line in Active mode. The range for this field is 0 to 255, where a value of 0 means 8.

See Section 6.16.1 "Introduction" for more information on how this field helps determine the actual number of ADCs per Line

Range: 0 (8), 1 to 255

DRIFT Field

NOTE	This field is used only by Self Capacitance Configuration T111 instance 0 for self capacitance touch data.

The default behavior of this field is to provide the interval (specified in 200 ms increments) between drift operations for self capacitance bassed references. A drift operation samples the deltas after a cycle, and "moves" the references (per node) towards those deltas. This allows for small variations in operating conditions, such as temperature changes. The amount to drift is specific to the measurement type, but typically a drift "moves" the references by 4 to 8 16-bit delta counts.

The range is 0 to 255 in 200 ms increments, where the default value of 0 means 5 (1 second). A value of 255 (disabled) is not recommended for this field.

Range: 0 (5), 1 to 254, 255 (disabled)

Under NDA
© 2021 Microchip Technology Inc. and its subsidiaries

DRIFTST Field

NOTE	This field is used only by Self Capacitance Configuration T111 instance 0 for self capacitance touch data.

This field specifies the Drift Suspend Time. This is the duration for which drift is suspended after any touches have been removed. See the mutual capacitance handling of the "DRIFTST Field" for more information on the Drift Suspend Time.

DRIFTST is configured in units of 200 ms, where a value of zero means 10 (2 seconds).

Range: 0 (10), 1 to 255

CALRECSTR Field

This field configures the Calibration Recovery Strength.

The Calibration Recovery process causes an automatic calibration when the system decides that a calibrated in touch has been removed from the sensor. This field controls how aggressive that process is for self-capacitance touch data. Low values will increase the probability of a system calibration; high values will decrease this probability. A value of 255 disables this feature, and the default value of 0 means 30.

Note that the tuning data can be output through the Diagnostic Debug T37 object using the Command Processor T6 DIAGNOSTIC command (see Section 2.2 "Diagnostic Debug T37 Object" and "DIAGNOSTIC Field"). Note that this must be enabled using the Acquisition Configuration T8 CFG CALRECTUN control (see "CFG Field").

Range: 0 (30), 1 to 254, 255 (disabled)

INRUSHCFG Field

YRESISTOR/XRESISTOR: Controls the selection of the inrush current resistors for the measurement type associated with the object instance. The value specified controls the selection for all X or Y line measurements (as appropriate). The value specified controls the selection for all X or Y lines (as appropriate).

YRESISTOR/XRESISTOR Range: See Table 6-70.

TABLE 6-70: YRESISTOR AND XRESISTOR VALUES

YRESISTOR/XRESISTOR Setting	Resistor Value
0	50Ω (default)
1	71Ω
2	117Ω
3	180Ω
4	276Ω
5	424Ω
6	651Ω
7	1KΩ

ALTINTTIMEX Field

NOTE	This field is used only by Self Capacitance Configuration T111 instance 0 for self capacitance touch data.

This field specifies the Alternative X Axis Measurement Integration Time in 24 MHz clock cycles (41.67 ns). This allows different integration times to be specified for the self capacitance touch and hover X and Y measurements. Note that this field does not apply to self capacitance single-ended measurements.

If this field is set to zero, the INTTIME setting is used for both X and Y measurements (see "INTTIME Field").

If this field is set to a non-zero value then the specified value is used for X measurements, and the INTTIME field setting is used for the Y measurement.

Note that, if ALTINTTIMEX is being used, it is important to make sure that the DELAYTIME and ALTDELAYTIMEX are adjusted such that the following condition is met (see "DELAYTIME Field" and "ALTDELAYTIMEX Field"):

INTTIME + DELAYTIME = ALTINTTIMEX + ALTDELAYTIMEX

Otherwise external noise handling performance will be impacted.

Range: See Table 6-71

TABLE 6-71: ALTINTTIMEX VALUES

Value	X Measurements	Y Measurements
0	The INTTIME setting is used for both X and Y measurements	
1 to 255	ALTINTTIMEX specifies an alternative integration time for X measurements	The INTTIME setting is used for Y measurements.

ALTDELAYTIMEX Field

NOTE This field is used only by Self Capacitance Configuration T111 instance 0 for self capacitance touch data.

This field specifies the Alternative X Axis Measurement Delay Time in 24 MHz clock cycles (41.67 ns). This allows different delay times to be specified for the self capacitance touch X and Y measurements. Note that this field does not apply to self capacitance single-ended measurements.

If this field is set to zero, the DELAYTIME setting is used for both X and Y measurements (see "DELAYTIME Field").

If this field is set to a non-zero value then the specified value is used for X measurements, and the DELAYTIME field setting is used for the Y measurement.

Note that, if an alternative integration time is being used (see "ALTINTTIMEX Field"), ALTDELAYTIME value must be set such that the two burst frequencies match.

Range: 0 (DELAYTIME used for both X and Y measurements),
1 to 255

DCDRIFT Field

NOTE This field is used only by Self Capacitance Configuration T111 instance 0 for self capacitance touch data.

This field controls the DC Drift Interval for DC correction. The drift intervals for other measurement type instances are stored in other objects.

Signals can drift because of changes in the nature of the components and materials over time and temperature. It is crucial that such drift is compensated for, otherwise false detections and sensitivity shifts can occur.

Drift compensation is performed by adjusting the reference level. The rate of adjustment is performed slowly, otherwise the incorrect DC levels may introduce erroneous touch detections and calibrations. The DCDRIFT field is configured in increments of 200 ms, where the default value of 0 means 5 (1 second). A value of 255 disables drift.

Range: 0 (5), 1 to 254, 255 (disabled)

DCFILTER Field

NOTE This field is used only by Self Capacitance Configuration T111 instance 0 for self capacitance touch data.

This field provides the DC Filter controls.

IIREN: Enables the IIR Filter. The filter is enabled if this bit is set to 1, and disabled if it is set to 0.

IIRCOEF: Specifies the DC IIR Filter Coefficient. The units are 2s exponent of filter coefficient.

IIRCOEF Range: 0 to 7

IIRCOEF Typical: 1

SLEWEN: Enables the Slew Rate Filter. The slew rate filter is enabled if this bit is set to 1, and disabled if it is set to 0.

Under NDA
© 2021 Microchip Technology Inc. and its subsidiaries

SLEWACC: Enables the Slew Rate Filter Accelerator. The accelerator is enabled if this bit is set to 1, and disabled if it is set to 0.

DCCALRECSTR Field

NOTE	This field is used only by Self Capacitance Configuration T111 instance 0 for self capacitance touch data.

This field configures the DC Calibration Recovery Strength.

The DC Calibration Recovery process causes an automatic calibration when the system decides that the overall touch level has become negative. This field controls how aggressive that process is for the self capacitance DC level. Low values will increase the probability of a system calibration; high values will decrease this probability. A value of 255 disables this feature. The default value of 0 means 30.

Note that the tuning data can be output through the Diagnostic Debug T37 object using the Command Processor T6 DIAGNOSTIC command (see Section 2.2 “Diagnostic Debug T37 Object” and “DIAGNOSTIC Field”). Note that this must be enabled using the Acquisition Configuration T8 CFG CALRECTUN control (see “CFG Field”).

NOTE	If DC calibration is enabled (that is, not equal to 255), then the self capacitance touch measurement must be allowed using Acquisition Configuration T8 MEASALLOW (see “MEASALLOW Field”).

Range: 0 (30), 1 to 254, 255 (disabled)

DCCALERRRATIO Field

NOTE	This field is used only by Self Capacitance Configuration T111 instance 0 for self capacitance touch data.

This field configures the DC Calibration Error Ratio. This is the proportion (in units of 1/256) of the difference between the DC level estimates for the single-ended and auxiliary touch measurements. This difference is compared against the calibration recovery strength in order to trigger a calibration.

Range: 0 to 255

SYNCDELAY Field

This 16-bit field specifies the delay from when the SYNC edge occurs to when the positive/negative edge of the self capacitance pulse is generated.

The delay is specified in units of the CTE clock period (20 ns; 50 MHz), where a value of 0 means no additional delay.

Range: 0 (no additional delay), 1 to 65534

ADCSPERSYNC Field

NOTE	If synchronization is turned on (that is, ADCSPERSYNC is non-zero), GPIO0 cannot be configured by the GPIO Configuration T19 object. GPIO0 must be set as an input, or the GPIO Configuration T19 object disabled, for the CTE Configuration T46 pulse synchronization to work.

ADCSPERSYNC: Determines whether or not the ADC conversions are synchronized to the SYNC pin for self capacitance measurements. To enable pulse synchronization on mutual capacitance measurements, see Section 6.8 “CTE Configuration T46 Object”.

A value of 0 means no synchronization is required (that is, it is not synchronized to the input signal on the SYNC pin).

A value of 1 synchronizes the rising and falling edges of each ADC to the falling edge of the SYNC signal (see Figure 6-9 on page 239 for an example).

A value greater than 1 determines the number (value minus 1) of ADCs per Sync Group.

Note that if there is a requirement to toggle pulse synchronization on and off at run time, the device must be configured and backed up with pulse synchronization enabled in either Self Capacitance Configuration T111 or CTE Configuration T46. This ensures that the device will start up with pulse synchronization enabled for at least one measurement type and the SYNC pin is latched for use. This prevents the pin from being used for other uses.

Range: 0 (no synchronization),
1 (ADCs synchronized to SYNC)
2 to 63

DCGAINSF Field

NOTE	This field is used only by Self Capacitance Configuration T111 instance 0 for self capacitance touch data.

This field configures the DC Gain Scaling Factor. This is the scaling factor used in gain calculations associated with the system-provided DC level. The value is used to scale the magnitude of the input DC level so that it better matches the estimates based upon other self capacitance touch measurements.

A smaller value should be used if the DC level exceeds the estimate from other self capacitance measurements; a larger value should be used if the DC level is less than the estimate from other self capacitance measurements. It may be beneficial to re-tune this value if the gain of the self capacitance measurements is changed.

The value for this field is a signed 8-bit number, where –128 represents a scaling factor of 50%, 0 represents a scaling factor of 100% and +127 represents a scaling factor of 150%.

Range: –128 to +127

DCTHRX Field

NOTE	This field is used only by the Self Capacitance Configuration T111 instance 0 for self capacitance touch data.

This field configures the Self Capacitance DC Threshold for X axis measurements against which the DC estimate is compared for leaving the low power idle state.

Range: 0 to 255

DCTHRY Field

NOTE	This field is used only by the Self Capacitance Configuration T111 instance 0 for self capacitance touch data.

This field configures the Self Capacitance DC Threshold for Y axis measurements against which the DC estimate is compared for leaving the low power idle state.

Range: 0 to 255

DCIDLESLEWMIN Field

NOTE	This field is used only by the Self Capacitance Configuration T111 instance 0 for self capacitance touch data.

This field configures a minimum slew rate applied to the DC estimate for the measurement controlled by this Self Capacitance Configuration T111 instance. The slew rate can reduce to one count per acquisition cycle in quiet conditions and thus will increase first touch latency greatly when the idle default measurement is self capacitance single ended (see the DEFMEASIDLE control in the Self Capacitance Measurement Configuration T113 “CTRL Field”). The slew rate must allow enough change in the estimate to allow the DC threshold to be reached. A value of approximately (3 × DCTHRX) or (3 × DCTHRY), as appropriate, should result in a first touch latency very similar to that when the default idle mode is MUTUALTCH or SELFTCH.

Range: 0 (1), 1 to 255

Under NDA
© 2021 Microchip Technology Inc. and its subsidiaries

DCPROCTHRX and DCPROCTHRY Fields

NOTE These fields are used only by Self Capacitance Configuration T111 instance 0 for self capacitance touch data.

These fields set the DC Level Processing Threshold for the X and Y lines. This is the threshold used for processing DC level estimates for the X and Y lines (for example, for use by the Self Capacitance Measurement Configuration T113 algorithms). If the threshold is greater than or equal to DCPROCTHRX/DCPROCTHRY, the DC level processing is done. The threshold is specified in 8-bit delta values, where the default value of 0 effectively means always perform the processing.

Range: 0 to 255

6.16.3 CONFIGURATION CHECKS

An instance of the Self Capacitance Configuration T111 object may cause an automatic recalibration to be performed when certain fields are changed.

TABLE 6-72: CONFIGURATION CHECKS FOR SELF CAPACITANCE CONFIGURATION T111 (SPT_SELFCAPCONFIG_T111)

Field	Changing The Field Causes...		Effect of Configuration Checks On Field
	Configuration Check	Automatic Recalibration	
CTRL	No	Yes	None
DBGCTRL	No	No	None
INTTIME	No	Yes	None
DELAYTIME	No	Yes	None
IDLESYNCSPERL	No	No	None
ACTVSYNCSPERL	No	No	None
DRIFT	No	No	None
DRIFTST	No	No	None
CALRECSTR	No	No	None
INRUSHCFG	No	Yes	None
ALTINTTIMEX	No	Yes	None
ALTDELAYTIMEX	No	Yes	None
DCDRIFT	No	No	None
DCFILTER	No	No	None
DCCALRECSTR	No	No	None
DCCALERRRATIO	No	No	None
SYNCDELAY	No	No	None
ADCSPERSYNC	No	No	None
DCGAINSF	No	No	None
DCTHRX	No	No	None
DCTHRY	No	No	None
DCIDLESLEWMIN	No	No	None
DCPROCTHRX	No	No	None
DCPROCTHRY	No	No	None

6.17 Self Capacitance Measurement Configuration T113 Object

6.17.1 INTRODUCTION

A Self Capacitance Measurement Configuration T113 object is used to set the gain for a particular self capacitance single-ended measurement to generate data for use by other objects.

The current use of this object on the mXT640UD-CCUBHA1 Family is to make it possible for the Acquisition Configuration T8 object to trigger a recalibration when an antitouch is detected. To this end, the Self Capacitance Measurement Configuration T113 object computes the self capacitance single-ended measurement, and the Acquisition Configuration T8 object then triggers the recalibration.

Note that the SELFPROX bit in the Acquisition Configuration T8 MEASALLOW field must also be set for the self capacitance single-ended measurements to be used (see "MEASALLOW Field").

NOTE Self capacitance measurements must also be enabled for use in order for the self capacitance single-ended measurements to operate correctly. Therefore, the Auxiliary Touch Configuration T104 object must also be enabled for use.

6.17.2 CONFIGURATION

TABLE 6-73: CONFIGURATION FOR SELF CAPACITANCE MEASUREMENT CONFIGURATION T113 (SPT_SELFCAPMEASURECONFIG_T113)

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	CTRL	Reserved			DEFMEASIDLE	Reserved	ALTAXISEN	Reserved	ENABLE
1	XGAIN	Gain for the X axis							
2	YGAIN	Gain for the Y axis							
3	Reserved	Reserved							
4	Reserved	Reserved							
5	NODEAREASFCM	Per node X/Y scaling factor – Calculated to Measured axis							
6	NODEAREASFMC	Per node X/Y scaling factor – Measured to Calculated axis							

CTRL Field

ENABLE: Enables the use of the Self Capacitance Measurement Configuration T113 object. The object is enabled if set to 1, and disabled if set to 0.

NOTE A tune should be performed using the Self Capacitance Global Configuration T109 object after enabling the Self Capacitance Measurement Configuration T113 object (see "CMD Field").

ALTAXISEN: Specifies on which axis of electrodes to perform the single-ended self capacitance measurement.

If this bit is set to 0, the measurement is performed on the X axis (that is, the axis that has the larger number of lines).

If this bit is set to 1, the measurement is performed on the Y axis (that is, the axis that has the smaller number of lines).

Note that setting ALTAXISEN should be done in conjunction with the Self Capacitance Global Configuration T109 CTRL DISMEASX and DISMEASY controls if they are set.

DEFMEASIDLE: Setting this bit to 1 ensures that self capacitance single-ended measurements are included in the Default Measurement Types in Idle mode for use with Low Power Idle mode (see Section A.2.1 "Default Measurements").

Note that, regardless of the state of this bit, self capacitance single-ended measurements are always included in the Default Measurement Types for other processes (for example, Acquisition Configuration T8 recalibration) if the Self Capacitance Measurement Configuration T113 object is enabled.

Under NDA
© 2021 Microchip Technology Inc. and its subsidiaries

XGAIN and YGAIN Fields

These fields set the X or Y axis gain, as appropriate, for the self capacitance single-ended measurement.

The default range is 0 to 29 and this should be sufficient for most applications. However, using Gain × 2 mode may help improve moisture immunity performance. To enable Gain × 2 mode, set XGAIN/YGAIN bit 7 to 1. This will add additional settings in the range to 128 to 157.

For more information on choosing a suitable gain value, refer to *maXTouch U Series Tuning Guide*.

Range: 0 to 29, 128 to 157 (Gain × 2 mode)

NODEAREASFCM and NODEAREASFMC Fields

These fields specify the Node Area Scaling Factors for use with DC level estimation algorithm.

To save processing time, the DC estimation algorithm measures one axis to determine its DC level and then uses the results to calculate the DC level for the other axis. By default, the axis with the greater number of nodes is measured (that is, the X axis on the mXT640UD-CCUBHA1 Family), and the other (shorter) axis is calculated from this (that is, the Y axis). This can be changed, however, by setting the CTRL ALTAXISEN control to 1 (see "CTRL Field").

The scaling factors must be defined if the X and Y node areas differ, otherwise the calculated DC estimation will be incorrect. The NODEAREASFCM and NODEAREASFMC fields therefore specify the scaling factors to be used by the DC estimation algorithm in the form of two ratios:

- NODEAREASFCM specifies the ratio of the calculated axis area to the measured axis area; that is: ($Area_C$ / $Area_M$)
- NODEAREASFMC specifies the reciprocal of NODEAREASFCM. This is the ratio of the measured axis area to the calculated axis area; that is: ($Area_M$ / $Area_C$).

where $Area_C$ is the calculated axis and $Area_M$ is the measured axis.

NOTE	The NODEAREASFCM and NODEAREASFMC fields are not independent of each other: they should be configured as a pair, with NODEAREASFMC set to the reciprocal value of NODEAREASFCM.

The actual calculation of the ratios depends on the setting of the CTRL ALTAXISEN control (see Table 6-74.

TABLE 6-74: NODEAREASF CORRECTION FACTOR

	Ratio Calculation	
ALTAXISEN Setting	**NODEAREASFCM**	**NODEAREASFMC**
0	$Area_Y$ / $Area_X$	$Area_X$ / $Area_Y$
1	$Area_X$ / $Area_Y$	$Area_Y$ / $Area_X$

Note: $Area_X$ = Total area of X electrodes, $Area_Y$ = Total area of Y electrodes

The ratio specified in the NODEAREASFCM and NODEAREASFMC fields is specified in units of 1/128 minus 1, where the default value of 0 represents a value of 127 (that is, 128/128 = 1). For example:

63 = (64 / 128) = 0.5
127 = (128 / 128) = 1
255 = (256 / 128) = 2

NOTE	Values of 1 to 62 cannot be used, as the two fields are reciprocals of each other and the reciprocal values above 255 are not possible.

The maXTouch Studio tool can be used to calculate the values for these fields.

Range: 0 (127), 63 to 255

6.17.3 CONFIGURATION CHECKS

The Self Capacitance Measurement Configuration T113 object causes a configuration check to be performed in the following circumstances:

- When the object is enabled (that is, the ENABLE bit is set in the CTRL field)
- If the object is enabled, when certain fields are changed

In addition, some fields will cause an automatic recalibration to be performed.

A configuration check may determine that a configuration error has occurred (for example, if a setting is set outside of its allowed range or a conflict has occurred between two settings). This is signaled to the host (see Section 3.3 "Command Processor T6 Object"). The device halts until the error has been corrected. To fix the error, check that all the object settings are within their allowed limits, as stated in the field descriptions.

TABLE 6-75: CONFIGURATION CHECKS FOR SELF CAPACITANCE MEASUREMENT CONFIGURATION T113 (SPT_SELFCAPMEASURECONFIG_T113)

Field	Changing The Field Causes...		Effect of Configuration Checks On Field
	Configuration Check	Automatic Recalibration	
CTRL	Yes (1)	Yes	None
XGAIN	Yes	Yes	None
YGAIN	Yes	Yes	None
NODEAREASFCM	No	No	None
NODEAREASFMC	No	No	None

Note 1: If the ENABLE bit is toggled on or off.

Under NDA
© 2021 Microchip Technology Inc. and its subsidiaries

6.18 Data Container T117 Object

6.18.1 INTRODUCTION

The Data Container T117 objects are a set of read-only objects that together provide a mechanism for retrieving specific data from the device's internal memory. On the mXT640UD-CCUBHA1 Family, there are 6 instances of the Data Container T117 object and each instance holds 215 bytes, making 1290 bytes of data space in total. The multiple instances are concatenated so that the data is held in a contiguous manner across all the instances.

The data type is determined by the configuration of the Data Container Controller T118 object (see Section 6.19 "Data Container Controller T118 Object").

IMPORTANT!	The CRC included by header type 128 is an 8-bit CRC only and therefore may not detect all the errors in the data that could be detected by a larger CRC. For this reason, the Data Container T117 and Data Container Controller T118 objects are not recommended for use in a UL/IEC 60730 Class B compliant product.

6.18.2 CONFIGURATION

TABLE 6-76: CONFIGURATION FOR DATA CONTAINER T117 (SPT_DATACONTAINER_T117)

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	DATA[0]	Data byte 0							
…									
214	DATA[214]	Data byte 214							

DATA[] Fields

These fields contain the bytes of data, as determined by the Data Container Controller T118 object HEADERTYPE and DATATYPE fields (see "HEADERTYPE Field" and "DATATYPE Field").

IMPORTANT!	On the mXT640UD-CCUBHA1 Family, if the DATA[] fields contain delta data, the deltas are output in big endian format (that is, MSByte first). All other data is output as standard little endian format (LSByte first). On some other devices, however, the deltas may be output in standard little endian format (that is, LSByte first). Any generic host driver code will need to take this into account.

The data will extend across the DATA[] fields in the multiple instances of the Data Container T117 object in a contiguous manner. Any unused bytes are reserved.

NOTE	On the mXT640UD-CCUBHA1 Family, there are a total of 1290 data bytes (split across 6 instances of 215 bytes each) and only those bytes needed to hold the data for the chosen Data Container Controller T118 Data Type are used. Any unused bytes in the DATA[] fields above the header and data are reserved and should not be processed by the host. In addition, although the DATA[] fields include data for 640 possible nodes, on the mXT448UD-CCUBHA1 variant device, only a maximum of 448 nodes can be used on this device, so some of the bytes of data will represent unused nodes. These unused bytes may contain random data and should also be considered reserved.

For example, If the Data Container Controller T118 DATATYPE is 16 and the HEADERTYPE is 1, the DATA[] fields will contain a 1-byte header, followed by the 16-bit deltas for all the nodes in the sensor matrix. An example is shown in Figure 6-10.

FIGURE 6-10: EXAMPLE OF DATA[] FIELDS – MUTUAL CAPACITANCE DELTAS

Object Instance	Data Index	Data[] Field	Data Content	
	0	DATA[0]	Header: Data Type = 0x10	Header
	1	DATA[1]	Delta for Node 0 MSByte	Data (Deltas)
Instance 0	2	DATA[2]	Delta for Node 0 LSByte	
	…	…	…	
	214	DATA[214]	…	
	215	DATA[0]	…	
Instance 1	…	…	…	
	429	DATA[214]	…	
	…	…	…	
	1075	DATA[0]	…	
	…	…	…	
	1279	DATA[204]	Delta for Node 639 LSByte	
Instance 5	1280	DATA[205]	Delta for Node 639 LSByte	
	1281	DATA[206]	Reserved	
	…	…	…	
	1289	DATA[214]	Reserved	

Assumes object size = 215

Note: The example shows nodes on the mXT640UD-CCUBHA1; on the mXT448UD-CCUBHA1 variant device there will be fewer nodes.

Under NDA

© 2021 Microchip Technology Inc. and its subsidiaries

6.19 Data Container Controller T118 Object

6.19.1 INTRODUCTION

The Data Container Controller T118 object provides configuration for the Data Container T117 object. The Data Container Controller T118 object makes internal data, such as deltas and references, available. The data type is selected in the Data Container Controller T118 object. There is no data processing, as the Data Container Controller T118 object simply gives direct access to internal data in memory. The internal data pointer is updated each acquisition during touch processing.

IMPORTANT!	The CRC included by header type 128 is an 8-bit CRC only and therefore may not detect all the errors in the data that could be detected by a larger CRC. For this reason, the Data Container T117 and Data Container Controller T118 objects are not recommended for use in a UL/IEC 60730 Class B compliant product.

The following data types are available (see Table 6-78 for more information):

- Mutual Capacitance Deltas for all touchscreen nodes
- Mutual Capacitance References for all touchscreen nodes

A header that provides additional bytes at the start of data can also be selected (see the "HEADERTYPE Field"). The header is processed each acquisition during touch processing.

The data should not be accessed while it is being reloaded within the device. A message can be enabled to indicate when the data is safe to access and the header has been processed.

NOTE	Data Container T117 and Data Container Controller T118 can generate a large volume of message traffic when enabled. This can interfere with systems that read the configuration at start-up (such as maXTouch Studio) or systems that check or upload the configuration as an anti-corruption check. It may be helpful to defer enabling Data Container Controller T118 until any start-up bus traffic is completed. This can be done by using a timer, triggered from Dynamic Configuration Controller T70 event ID 199.

6.19.2 CONFIGURATION

TABLE 6-77: CONFIGURATION FOR DATA CONTAINER CONTROLLER T118 (SPT_DATACONTAINERCTRL_T118)

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	CTRL	Reserved						RPTEN	ENABLE
1	DATATYPE	Data type							
2	HEADERTYPE	Header type							

CTRL Field

ENABLE: Enables or disables the use of this object. The object is enabled if this bit is set to 1, and disabled if this bit is set to 0.

RPTEN: Enables or disables this object to report messages. Reporting is enabled if this bit is set to 1, and disabled if this bit is set to 0.

DATATYPE Field

This field selects the type of data to be used in the Data Container T117. See "DATA[] Fields" for an example of how DATATYPE helps populates the Data Container T117 DATA[] fields.

IMPORTANT!	On the mXT640UD-CCUBHA1 Family, if the Data Container T117 DATA[] fields contain delta data, the deltas are output in big endian format (that is, MSByte first). All other data is output in standard little endian format (LSByte first).

Range: See Table 6-78

TABLE 6-78: DATATYPE VALUES

Value	Data Type	Size (Bytes)
0x00	None; all zero	0
0x10	Mutual capacitance deltas 16-bit signed (two's complement) mutual capacitance delta values for all touchscreen nodes. These are the raw deltas. Note that these will be output in big endian format (that is, MSByte first).	2 per node
0x11	Mutual capacitance references 16-bit signed Mutual Capacitance touch Reference values (mutual capacitance data) for all touchscreen nodes.	2 per node
All other values	Reserved. Any reads of this object will return zeros.	–

HEADERTYPE Field

This field configures the format of the header bytes written to the Data Container T117 object. See "DATA[] Fields" for an example of how HEADERTYPE helps populates the Data Container T117 DATA[] fields.

Range: See Table 6-79

TABLE 6-79: HEADERTYPE VALUES

Value	Description
0	No header is output
1	A 1-byte header is output that contains the value of the DATATYPE field. See Table 6-80.
128	A 5-byte header is output that contains CRC, timestamp and cycle counter. See Table 6-81.

TABLE 6-80: HEADER FORMAT FOR HEADERTYPE = 1

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	DATATYPE	Data type							

TABLE 6-81: HEADER FORMAT FOR HEADERTYPE = 128

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	CRC	CRC							
1	TIMESTAMP	Timestamp LSByte							
2		Timestamp MSByte							
3	CYCCNT	Cycle counter LSByte							
4		Cycle counter MSByte							

DATATYPE Field

This field outputs the data type (see Table 6-78). Invalid data types are output as zero.

CRC Field

This 8-bit cyclic redundancy check (CRC) can be used to check the integrity of data. It is calculated over the header and the nodes specified in Multiple Touch Touchscreen T100 (see Section B.2 "8-bit CRC").

TIMESTAMP Field

This 16-bit field outputs the timestamp in units of 16.384 KHz clock cycles (61.04 ms increments).

CYCCNT Field

This 16-bit field outputs a Cycle Counter. The counter is incremented on each acquisition.

Under NDA
© 2021 Microchip Technology Inc. and its subsidiaries

6.19.3 MESSAGES

The message data for a Data Container Controller T118 object is shown in Table 6-82.

TABLE 6-82: MESSAGE DATA FOR DATA CONTAINER CONTROLLER T118 (SPT_DATACONTAINERCTRL_T118)

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	DATAREADY	Data type of data that is ready to send							

DATAREADY Field

This field outputs the data type that will be output to the Data Container T117 object. See Table 6-78 for the possible data types.

6.20 Self Capacitance Voltage Modulation T133 Object

6.20.1 INTRODUCTION

The Self Capacitance Voltage Modulation T133 object can be used to configure the shape of the transition waveform for a self capacitance measurement.

The shape of the transition waveform for a self capacitance measurement is formed under the control of a Voltage Modulated Drive pulse. The Off and On times for the drive pulses are stored in a Drive Pulse table (see "DRVPULSE[] Fields") and these are used to shape the transition waveform.

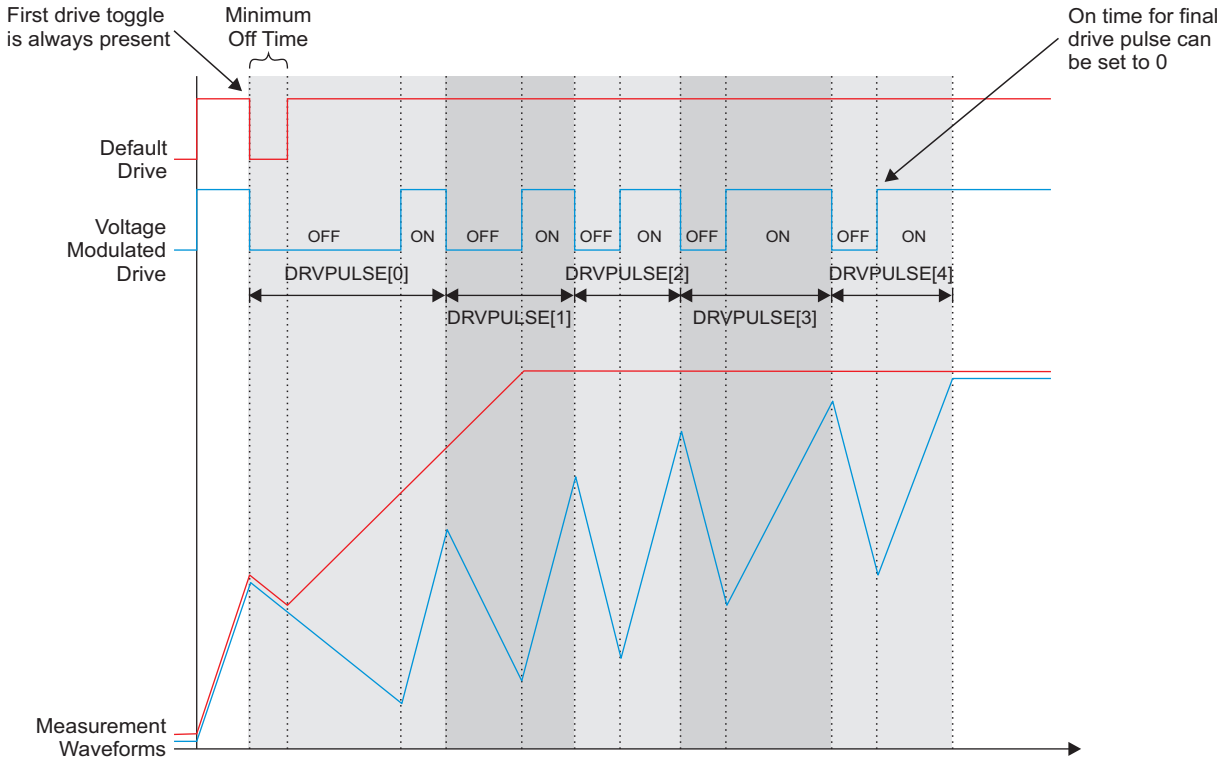
On the mXT640UD-CCUBHA1 Family there are 2 instances of the Self Capacitance Voltage Modulation T133 object, as follows:

- Instance 0 specifies the On and Off times for the positive pulse transitions on the rising edge (see Figure 6-11)
- Instance 1 specifies the On and Off times for the negative pulse transitions on the falling edge (see Figure 6-12)
- This allows for more control in matching the positive and negative transition waveforms.

6.20.1.1 Voltage Modulation

Figure 6-11 shows the effect of Self Capacitance Voltage Modulation T133 Instance 0 on the rising edge of the measurement waveform.

FIGURE 6-11: VOLTAGE MODULATION – RISING EDGE



In figure Figure 6-11:

- The red lines show the default drive pulse (upper) and measurement waveform (lower).
- The blue lines show the voltage modulated drive pulse (upper) and measurement waveform (lower).

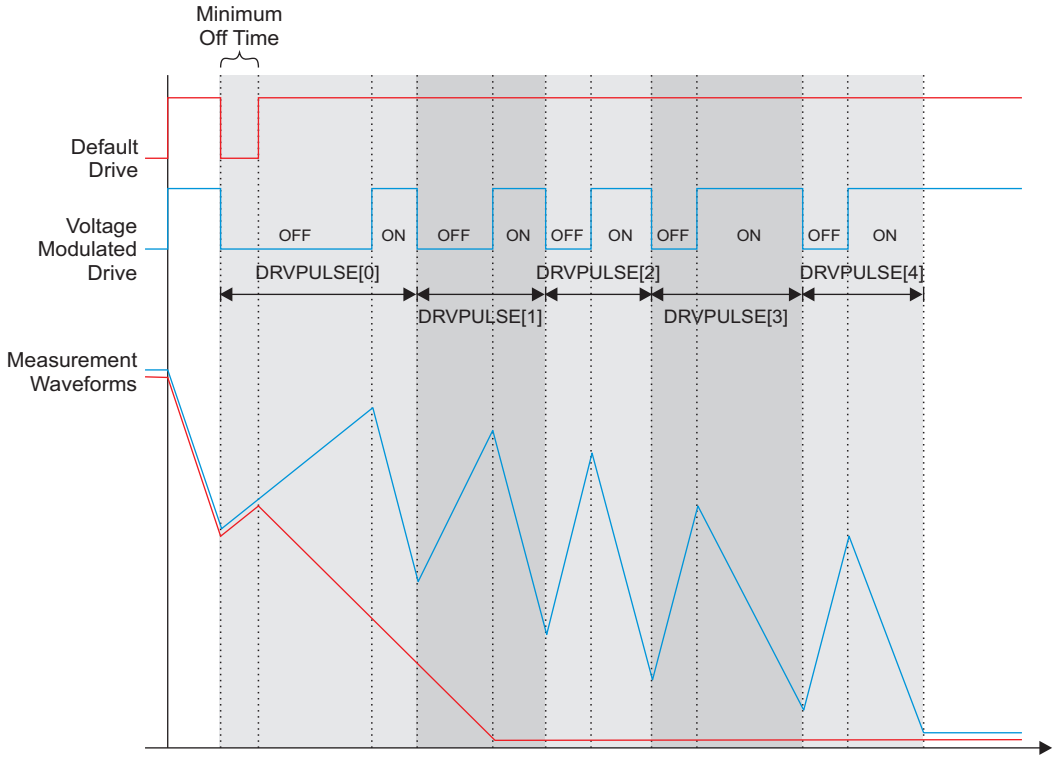
Note that both schemes have an automatic drive toggle at the start of the measurement.

In the case of Voltage Modulation, each drive pulse consists of a drive off time and a drive on time. A number of consecutive drive pulses may be used to shape the measurement waveform (up to a maximum of 20), as specified in the NUMDRVPULSES field. Note that the last drive pulse does not need to specify an on time and can be left at 0 (the minimum delay) to leave the drive on. In addition, there is a minimum off time, so if the Self Capacitance Voltage Modulation T133 object specifies a shorter off time, this is extended to the minimum off time.

Under NDA
© 2021 Microchip Technology Inc. and its subsidiaries

For a falling measurement waveform, as configured by Self Capacitance Voltage Modulation T133 Instance 1, the logic is reversed, with the drive on time pushing the measurement waveform downwards. This is shown in Figure 6-12.

FIGURE 6-12: VOLTAGE MODULATION – FALLING EDGE



NOTE	If self capacitance measurements have pre-charge disabled, the second half (rounded up) of the negative pulse transition settings are also applied to the Y lines when transitioning from a mutual capacitance measurement to a self capacitance measurement. That is, the following DRVPULSE[] settings are applied: ((ROUND_UP(NUMDRVPULSES / 2)) – 1) to (NUMDRVPULSES – 1) See the Self Capacitance Global Configuration T109 "CTRL Field" DISPRECHARGE control for details on disabling pre-charge.

6.20.2 CONFIGURATION

TABLE 6-83: CONFIGURATION FOR SELF CAPACITANCE VOLTAGE MODULATION T133 (SPT_SELFCAPVOLTAGEMOD_T133)

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	CTRL	Reserved							ENABLE
1	CFG	Reserved		NUMDRVPULSES					
2	DRVPULSE[0]	OFFTIME							
3		ONTIME							
...									
40	DRVPULSE[19]	OFFTIME							
41		ONTIME							

CTRL Field

ENABLE: Enables the use of this instance of the Self Capacitance Voltage Modulation T133 object. The object is enabled if set to 1, and disabled if set to 0.

CFG Field

NUMDRVPULSES: Specifies the number of Drive Pulses to be used for self capacitance Voltage Modulation, up to a maximum of 20 (see "DRVPULSE[] Fields").

Each Drive Pulse has an Off time and On time for each stage of the modulation. Note that the state of the Drive will be On at the end of each Drive Pulse. Therefore, the last Drive Pulse in use can have an On time of 0, after which the drive will be left on anyway.

Note that Drive On is in an upwards direction for a rising edge (Instance 0) and a downwards direction for a falling edge (Instance 1) (see Figure 6-11 and Figure 6-12).

A value of 0 disables the functionality.

Range: 0 (disabled), 1 to 20

DRVPULSE[] Fields

These fields specify the Drive Pulses for the voltage modulation.

OFFTIME and ONTIME: Specify the Off and On times for the drive pulse for this stage of the overall voltage modulation. The off or on time is specified in units of the CTE clock period (20 ns; 50 MHz).

Note that the device requires a few additional CTE cycles for essential processing, so the actual time will be slightly lengthened to allow for this. See the minimum off time in Figure 6-11 and Figure 6-12).

NOTE	It is the user's responsibility to increase the integration time for each self capacitance instance in use in line with the increase in timing introduced by the Voltage Modulation drive pulses (see "INTTIME Field").

OFFTIME/ONTIME Range: 0 to 255

6.20.3 CONFIGURATION CHECKS:

The Self Capacitance Voltage Modulation T133 performs no configuration checks, although an automatic recalibration is performed whenever a field is changed.

TABLE 6-84: CONFIGURATION CHECKS FOR SELF CAPACITANCE VOLTAGE MODULATION T133 (SPT_SELFCAPVOLTAGEMOD_T133)

Field	Changing The Field Causes...		Effect of Configuration Checks On Field
	Configuration Check	Automatic Recalibration	
CTRL	No	Yes	None (calibration only)
CFG	No	Yes	None (calibration only)
DRVPULSE[]	No	Yes	None (calibration only)

Under NDA
© 2021 Microchip Technology Inc. and its subsidiaries

6.21 Message Count T144 Object

6.21.1 INTRODUCTION

This object contains a count of the number of pending messages. This enables the host to use a direct memory access (DMA) transfer to read the messages from the Message Processor T5 object. See Section 3.2.1 “Introduction” for more information.

NOTE The mXT640UD-CCUBHA1 Family supports 2 active communications interfaces, so the Message Count T144 object maintains a message count and checksum for each interface independently. The Message Processor T5 object routes the messages to the appropriate interface, thus ensuring that each interface will receive its own messages.

6.21.2 CONFIGURATION

TABLE 6-85: CONFIGURATION FOR MESSAGE COUNT T144 (SPT_MESSAGECOUNT_T144)

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	COUNT	Message count							
1	CHECKSUM	Checksum							

COUNT Field

This field specifies the number of pending messages that are ready to be sent.

CHECKSUM Field

This field contains the 8-bit checksum for the Message Count T144 object.

See Appendix B “Checksum Calculation” for details on how to calculate the checksum.

6.22 Ignore Nodes Controller T145 Object

6.22.1 INTRODUCTION

The Ignore Nodes Controller T145 object is used to specify how ignored nodes configured in Ignore Nodes T141 (see Section 5.16 “Ignore Nodes T141 Object”) are applied to various measurement processes on the device. This may be useful for ignoring parts of the touchscreen that are not enabled for touch, such as with a free-form (non rectangular) screen shape, or when there are mechanical widgets present on the touchscreen.

There are currently two processes that Ignore Nodes T141 can be applied to:

- Touch Processing – Determined by CTRL IGNTCHPROCEN
- Self Test Control T10 Signal Limit Test – Determined by CTRL IGNSIGLIMEN

6.22.2 CONFIGURATION

TABLE 6-86: CONFIGURATION FOR IGNORE NODES CONTROLLER T145 (SPT_IGNORENODESCONTROL_T145)

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	CTRL	Reserved				IGNSIGLIMEN	IGNTCHPROCEN	Reserved	

CTRL Field

IGNTCHPROCEN: When this bit is set to 1, any nodes configured by the Ignore Nodes T141 ignore node masks are not processed by any touchscreen touch or anti-touch processes. However, any mutual capacitance measurement data from nodes indicated by the Ignore Nodes T141 object will still be subject to processing by the following objects:

- Key Array T15 if configured as a Standard Key Array (note that Generic Key nodes are not bound by Ignore Nodes T141 ignore node masks)
- Lens Bending T65
- Noise Suppression T72

but will not be processed by the following:

- Multiple Touch Touchscreen T100 and therefore any associated objects that process Multiple Touch Touchscreen T100 touches (for example, Touch Suppression T42 Maximum Screen Area Suppression)
- Acquisition Configuration T8 calibration recovery

Care must be taken when using the Ignore Nodes T141 object. For a sensor design where ignored nodes are not physically present on the array (for example, a circular sensor design), Self Capacitance Touch measurements may be used, as long as the sensor tunes correctly (that is, the difference in load capacitance is within the Self Capacitance Global Configuration T109 tuning range of the device, or physical capacitors have been added).

For designs where the nodes are still physically present, but it is desired to suppress touch reporting or signal limit testing only in those areas (by setting YMASK bits in one or more instances of the Ignore Nodes T141 object for use on the Multiple Touch Touchscreen T100 object), the associated instances of the following objects must be disabled:

- Retransmission Compensation T80
- Auxiliary Touch Configuration T104
- Self Capacitance Measurement Configuration T113

If Dual X is active, a node will be ignored in any measurement that includes that node if the node has been configured as an ignored node by Ignore Nodes T141. This applies to any of the processes listed above on objects that have the Dual X measurement capability (for example, Touchscreens, but not Key Arrays).

IGNSIGLIMEN: When this bit is set to 1, any nodes configured by the Ignore Nodes T141 ignore node masks are not tested by the Self Test Control T10 Mutual Capacitance Signal Limit Tests.

Note the following:

- The Ignore Nodes T141 instance representing the first X line used by a touchscreen (that is, X0) should contain at least one YMASK bit that is set to 0. The reason is that this X line should have at least one normal (not ignored) node to ensure that the minimum and maximum values can be calculated for comparisons during the tests.

Under NDA
© 2021 Microchip Technology Inc. and its subsidiaries

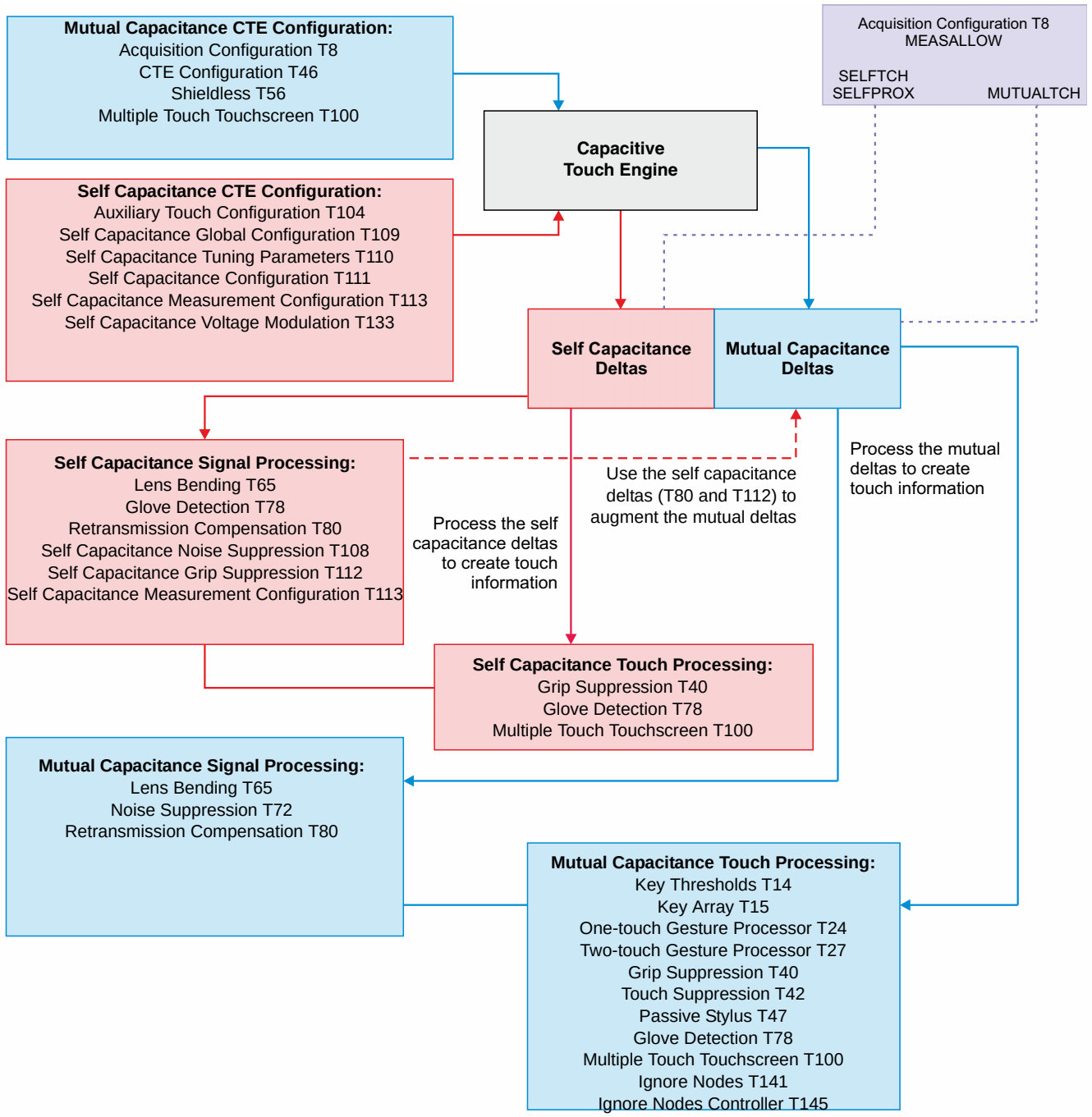
- If all the nodes for a touch object are configured as ignored nodes in Ignore Nodes T141, then the Self Test Control T10 Signal Limit Tests will fail for that touch object.

APPENDIX A: MEASUREMENT PROCESSING

A.1 Measurement Processing Objects

Figure A-1 shows the different measurement types on the mXT640UD-CCUBHA1 Family and the objects used to configure and process these.

FIGURE A-1: MEASUREMENT PROCESSING



Under NDA
© 2021 Microchip Technology Inc. and its subsidiaries

A.2 Touch Detection

The mXT640UD-CCUBHA1 Family allows for both mutual and self capacitance measurements, with the self capacitance measurements being used to augment the mutual capacitance measurements to produce reliable touch information.

When self capacitance measurements are enabled, touch classification is achieved using both mutual and self capacitance touch data. This has the advantage that both types of measurement systems can work together to detect touches under a wide variety of circumstances.

Mutual capacitance touch data is used wherever possible to classify touches as this has greater granularity than self capacitance measurements and provides positional information on touches. For this reason, multiple touches can only be determined by mutual capacitance touch data. In Self Capacitance Touch Default mode, if the self capacitance touch processing detects multiple touches, touchscreen processing is skipped until mutual capacitance touch data is available.

Self capacitance measurements and P2P mutual capacitance measurements allow for the detection of single touches in extreme cases, such as single thick glove touches, when mutual capacitance touch detection alone may miss touches.

A.2.1 DEFAULT MEASUREMENTS

The system may be configured for different types of default measurements in idle and active modes.

For example, the device may be configured for Mutual Capacitance Touch Default Idle and Self Capacitance Touch Default Active (or any other combination; see Table A-1). Note that other types of scans (such as P2P mutual capacitance scans and other types of self capacitance scans) may also be made depending on configuration.

TABLE A-1: DEFAULT MEASUREMENT TYPES IN IDLE AND ACTIVE MODES

Mode	Measurement Type	Set By...
Idle	Mutual Capacitance Touch	Automatically set as the default unless another measurement type is chosen as the default
	Self Capacitance Touch	Auxiliary Touch Configuration T104 CTRL DEFMEASIDLE control
	Self Capacitance Single-ended	Automatically measured if Self Capacitance Measurement Configuration T113 is enabled. Explicitly used for Low Power Idle mode if CTRL DEFMEASIDLE = 1.
Active	Mutual Capacitance Touch	Automatically set as the default unless another measurement type is chosen as the default
	Self Capacitance Touch	Auxiliary Touch Configuration T104 CTRL DEFMEASACTV control

A.3 Touch Classification

A.3.1 FINGER TOUCHES

Finger touches are classified when the following conditions are met:

- Mutual capacitance touch delta ≥ Multiple Touch Touchscreen T100 TCHTHR,
  or
  mutual capacitance touch deltas < Multiple Touch Touchscreen T100 TCHTHR
  but has been compensated by Retransmission Compensation T80

and (if Self Capacitance Touch Default Active mode)

- Self capacitance touch deltas ≥ Auxiliary Touch Configuration T104 XTCHTHR/YTCHTHR

A.3.2 GLOVE TOUCHES

Normal thin glove touches are classified using mutual capacitance data when the following conditions are met:

- Mutual capacitance touch deltas ≥ Multiple Touch Touchscreen T100 INTTHR
  and
  mutual capacitance touch deltas < Multiple Touch Touchscreen T100 TCHTHR (out of detect)

and

- The conditions set in Glove Detection T78 are met for glove touches

and

- (If Mutual Capacitance touch deltas are not present on a cycle)
  Self capacitance touch deltas ≥ Auxiliary Touch Configuration T104 XINTTHR/YINTTHR

Note that multiple touch glove touches are supported (mutual measurements only).

Thick glove touches are classified using self capacitance data when the following conditions are met:

- Self capacitance touch deltas ≥ Glove Detection T78 SCTGLOVETHR
  and meet Glove Detection T78 criteria

and

- (If Mutual Capacitance touch deltas are present on a cycle)
  Mutual capacitance touch delta < Multiple Touch Touchscreen T100 INTTHR

A.3.3 PASSIVE STYLUS TOUCHES

Stylus touches are classified when the following conditions are met:

- Mutual capacitance touch deltas ≥ Multiple Touch Touchscreen T100 INTTHR
  and
  mutual capacitance touch deltas < Multiple Touch Touchscreen T100 TCHTHR

and

- The conditions set in the Passive Stylus T47 object are met for stylus touches

and (if Self Capacitance Touch Default Active mode)

- Self capacitance touch deltas ≥ Auxiliary Touch Configuration T104 XINTTHR/YINTTHR

A.3.4 SUPPRESSED LARGE TOUCHES

Suppressed large touches are classified when the following conditions are met:

- Touch Suppression T42 has detected a large touch for suppression

and

- Touch Suppression T42 CFG SUPTCHRPTEN bit is set to 1 to allow the reporting of suppressed large touches

Under NDA
© 2021 Microchip Technology Inc. and its subsidiaries

APPENDIX B: CHECKSUM CALCULATION

B.1 24-bit CRC

B.1.1 INTRODUCTION

The device uses a 24-bit cyclic redundancy check (CRC) in the following places:

- To check the integrity of the main Information Block for the device
- To check the integrity of the Information Block T254 object
- To check the integrity of the configuration settings held in the non-volatile memory

These CRC checksums allow the host to be confident that the memory map layout has been read over the communications bus correctly.

The CRC polynomial for a 24-bit CRC is 0x80001B and the initial CRC value is 0x0000.

NOTE	The C code in this appendix uses the type declarations `uint8_t`, `uint16_t` and `uint32_t` for 8-bit, 16-bit and 32-bit unsigned integers respectively.

B.1.2 24-BIT CRC ALGORITHM

Each checksum is generated by running an algorithm which takes two new bytes of data and combines them with the current checksum to produce a new checksum value.

The sample code below shows how to calculate the checksum for each 16-bit word in a byte stream.

```
uint32_t crc24(uint32_t crc, uint8_t firstbyte, uint8_t secondbyte)
{

    static const uint32_t CRCPOLY = 0x80001B;
    uint32_t result;
    uint16_t data_word;

    data_word = (uint16_t)firstbyte | (uint16_t)((uint16_t)secondbyte << 8u);
    result = (uint32_t)data_word ^ (uint32_t)(crc << 1u);

    if(result & 0x1000000) // If bit 25 is set
    {
        result ^= CRCPOLY;// XOR result with CRCPOLY
    }

    return result;
}
```

The checksum routine is called iteratively to calculate the CRC two bytes (16-bit word) at a time. This means that two bytes must be read from the device before performing each iteration of the checksum. Therefore, if an odd number of bytes is read from the device, the host's checksum code should add a zero byte to the end of the byte stream to make the sequence even. For example, if the following stream of 7 bytes is received from the device:

byte1 — byte2 — byte3 — byte4 — byte5 — byte6 — byte7

The checksum could be calculated as follows:

```
uint32_t CRC = 0;
CRC = crc24(CRC, byte1, byte2)
CRC = crc24(CRC, byte3, byte4)
CRC = crc24(CRC, byte5, byte6)
CRC = crc24(CRC, byte7, 0)                      /* <- zero added for the last checksum */

CRC = CRC & 0x00FFFFFF;                         /* <- mask the 32-bit result to 24-bit */
```

An alternative method of calling the CRC calculation function is to use a loop, as shown in Section B.1.3 "Information Block Checksum".

Table B-1 shows an example block of 32 bytes and the CRCs these generate. The bytes consist of 31 data bytes plus an additional zero byte to even up the count. You can use the data in this table to verify the validity of any coded CRC routine.

TABLE B-1: EXAMPLE CRC CALCULATIONS FOR 24-BIT CRC

First Byte	Second Byte	CRC Calculations	
0x00	0xFF	0x00FF00	Intermediate CRC (partial calculation)
0x11	0xEE	0x011011	
0x22	0xDD	0x02FD00	
0x33	0xCC	0x053633	
0x44	0xBB	0x0AD722	
0x55	0xAA	0x150411	
0x66	0x99	0x2A9144	
0x77	0x88	0x55AAFF	
0x88	0x77	0xAB2276	
0x99	0x66	0xD6226E	
0xAA	0x55	0x2C116D	
0xBB	0x44	0x586661	
0xCC	0x33	0xB0FF0E	
0xDD	0x22	0xE1DCDA	
0xEE	0x11	0x43A841	
0xFF	0x00	0x87507D	Expected CRC (final calculation)

B.1.3 INFORMATION BLOCK CHECKSUM

The checksum for the Information Block should be calculated by the host on start-up when the Information Block is first read. This should be compared to the checksum value at the end of the Information Block. If there is a mismatch, an error has occurred.

The following code shows how to use the example `crc24()` function given in Section B.1.2 “24-bit CRC Algorithm” to calculate the CRC checksum for the Information Block.

```
uint32_t crc = 0;                                /* Calculated CRC */
uint16_t crc_area_size;                          /* Size of data for CRC calculation */
uint8_t *mem;                                    /* Data buffer */
uint8_t i;
uint8_t status;

/* 7 bytes of version data, 6 * NUM_OF_OBJECTS bytes of object table. */
crc_area_size = ID_INFORMATION_SIZE +
            info_block->info_id.num_declared_objects *
            OBJECT_TABLE_ELEMENT_SIZE;

mem = (uint8_t *) malloc(crc_area_size);
if (mem == NULL)
{
    /* Handle error */
}

/*
 * Read the data using a function written for this purpose
 * Here, it is assumed that the function is named read_mem()
 * and that it returns a status code with the value READ_MEM_OK.
 * It is assumed to have the following prototype:
 * uint8_t read_mem( uint16_t Address, uint8_t ByteCount, uint8_t *Data )
*/
status = read_mem(0, crc_area_size, mem);

if (status != READ_MEM_OK)
{
    /* Handle error */
}
```


Under NDA

© 2021 Microchip Technology Inc. and its subsidiaries

```
/*
 * Call the CRC function crc24() iteratively to calculate the CRC,
 * passing it two bytes at a time.
*/
i = 0;
while (i < (crc_area_size - 1))
{
    crc = crc24(crc, *(mem + i), *(mem + i + 1));
    i += 2;
}

/*
 * Call the crc24() with the final byte,
 * plus an extra 0 value byte to make the sequence even.
*/
crc = crc24(crc, *(mem + i), 0);

free(mem);

/* Final result */
crc = (crc & 0x00FFFFFF);                        /* <- mask the 32-bit result to 24-bit */
```

B.1.4 CONFIGURATION CHECKSUM

The configuration checksum checks the integrity of the memory map whenever a write to the device non-volatile memory takes place. Typically this is when the device is initially set in the factory or during subsequent firmware upgrades. The host should perform a CRC on the entire contents of the memory map from the address of the Key Thresholds T14 object onwards. This CRC should then be compared with the expected checksum.

B.2 8-bit CRC

B.2.1 INTRODUCTION

In order to comply with UL/IEC 60730 Class B, an 8-bit CRC must be used in the following places:

- When reading message data from the device, to check the integrity of data in messages from the Message Processor T5 object (see Section B.2.4 “Reading Message Processor T5 Messages with an 8-bit CRC”).
- When reading the Message Count T144 message count during DMA access.
- When performing a block read of Data Container T117 data, the Data Container Controller T118 object can request an 8-bit CRC for certain types of data. See Data Container Controller T118 “HEADERTYPE Field” for more information.
- When writing data to the device, to check that data has been transmitted by the host over the communications bus correctly (see Section B.2.3 “Writing to the device with an 8-bit CRC”).

The CRC polynomial for an 8-bit CRC is 0x8C and the initial CRC value is 0x00.

B.2.2 8-BIT CRC ALGORITHM

The sample code below shows how to calculate the 8-bit checksum.

```
uint8_t crc8(uint8_t crc, uint8_t data)
{
    static const uint8_t CRCPOLY = 0x8C;

    for (uint8_t index = 0; index < 8; index++)
    {
        uint8_t fb = (crc ^ data) & 0x01;
        data >>= 1;
        crc >>= 1;
        if (fb) {
            crc ^= CRCPOLY;
        }
    }

    return crc;
}
```

B.2.3 WRITING TO THE DEVICE WITH AN 8-BIT CRC

When writing to the device, an 8-bit CRC must be added to all data transmissions from the host. The CRC is calculated on all the bytes sent, including the address bytes, but excluding the SLA+W byte.

B.2.3.1 Example Write Operation with an 8-bit CRC

The examples in Figure B-1 and Figure B-2 set the address pointer to 0x1234.

FIGURE B-1: EXAMPLE WRITE WITH CHECKSUM – I²C INTERFACE

		Address								
		LSByte	MSByte	Data	Data	Data	Data	Seq No.	Data CRC	
START	SLA+W	0x34	0x12	0x96	0x9B	0xA0	0xA5	0x65	0x36	STOP

FIGURE B-2: EXAMPLE WRITE WITH CHECKSUM – SPI INTERFACE

Write Data

		Address		Length							
	Op Code	LSByte	MSByte	LSByte	MSByte	Hdr CRC	Data	Data	Data	Data	Data CRC
MOSI	0x01	0x34	0x12	0x05	0x00	0x71	0x96	0x9B	0xA0	0xA5	0x7F
MISO	X	X	X	X	X	X	X	X	X	X	X

Response

MOSI	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF

		Address		Length		
	Op Code	LSByte	MSByte	LSByte	MSByte	Hdr CRC
MISO	0x81	0x34	0x12	0x05	0x00	0xBA

An extra byte that holds the Data checksum is sent as the last byte. The device compares this checksum with its own checksum calculated from the data received. If the two checksums do not match, the Command Processor T6 object in the device generates a COMSERR message (see Section 3.3.3 “Messages”).

NOTE	The order of messages is not guaranteed so messages may be out of order. Other intervening messages may be received before the COMSERR message.

Under NDA

© 2021 Microchip Technology Inc. and its subsidiaries

The following tables show the sequence of the bytes in these examples and the intermediate calculations of the CRC.

TABLE B-2: EXAMPLE CRC CALCULATIONS: WRITE DATA – I²C INTERFACE

Bytes Sent by Host		CRC Calculations	
0x34	Address LSByte	0xDF	Intermediate CRCs (partial calculations)
0x12	Address MSByte	0x37	
0x96	Data	0xF1	
0x9B		0x1B	
0xA0		0x12	
0xA5		0xB1	
0x65	Sequence Number	0x36*	Expected CRC (final calculation)
0x36*	Actual Data CRC		

* These two values should match

TABLE B-3: EXAMPLE CRC CALCULATIONS: WRITE DATA – SPI INTERFACE

Bytes Sent by Host		CRC Calculations	
0x01	Op Code	0x5E	
0x34	Address LSByte	0x1B	Intermediate CRC (partial calculation)
0x12	Address MSByte	0x9C	
0x05	Length LSByte	0x8D	
0x00	Length MSByte	0x71*	Expected CRC (final calculation)
0X71*	Actual Header CRC		
0x96	Data	0xCC	Intermediate CRC (partial calculation)
0x9B		0x58	
0xA0		0xB6	
0xA5		0x7F*	Expected CRC (final calculation)
0x7F*	Actual Data CRC		

* These two values should match

TABLE B-4: EXAMPLE CRC CALCULATIONS: WRITE RESPONSE – SPI INTERFACE

Bytes Sent by Device		CRC Calculations in Host	
0x81	Op Code	0xD2	Intermediate CRC (partial calculation)
0x34	Address LSByte	0x34	
0x12	Address MSByte	0xFE	
0x05	Length LSByte	0x54	
0x00	Length MSByte	0xBA*	Expected CRC (final calculation)
0xBA*	Actual Header CRC		

* These two values should match

B.2.4 READING MESSAGE PROCESSOR T5 MESSAGES WITH AN 8-BIT CRC

In order to comply with UL/IEC 60730 Class B, an 8-bit CRC is added to reads of the messages from the Message Processor T5 object. The checksum is returned as the last byte of the Message Processor T5 object message data.

B.2.4.1 Example MESSAGE PROCESSOR T5 Read Operation with an 8-bit CRC

The examples in Figure B-3 and Figure B-4 read the message data from the Message Processor T5 object in the device. The examples consist of two operations: the first operation is a write of the address to the device and the second operation is a read of the message data.

FIGURE B-3: EXAMPLE MESSAGE PROCESSOR T5 READ WITH CHECKSUM – I²C INTERFACE

Set Address Pointer (= address of Message Processor T5)

		Address				
		LSByte	MSByte	Seq No.	Data CRC	
START	SLA+W	0x34	0x12	0x65	0x67	STOP

Read Data (= Message Processor T5 message data)

		Report ID	Data	Data	Data	Data	Data	Data	Data	Data	Data	Seq No.	Data CRC	
START	SLA+R	0x01	0x9B	0xA0	0xA5	0xAA	0xAF	0xB4	0xB9	0xCC	0xDE	0xD9	0x74	STOP

FIGURE B-4: EXAMPLE MESSAGE PROCESSOR T5 READ WITH CHECKSUM – SPI INTERFACE

Set Address Pointer (= address of Message Processor T5)

		Address		Length		
	Op Code	LSByte	MSByte	LSByte	MSByte	Hdr CRC
MOSI	0x02	0x34	0x12	0x0C	0x00	0x8D
MISO	X	X	X	X	X	X

Read Data (= Message Processor T5 message data)

MOSI	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF

0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF

		Address		Length		
	Op Code	LSByte	MSByte	LSByte	MSByte	Hdr CRC
MISO	0x82	0x34	0x12	0x0C	0x00	0x46

Report ID	Data	Data	Data	Data	Data	Data	Data	Data	Data	Seq No.	Data CRC
0x01	0x9B	0xA0	0xA5	0xAA	0xAF	0xB4	0xB9	0xCC	0xDE	0xD9	0x74

The following tables show the sequence of the initial write bytes with the intermediate calculations of the CRC.

TABLE B-5: EXAMPLE CRC CALCULATIONS: SET ADDRESS POINTER – I²C INTERFACE

Bytes Sent by Host		CRC Calculations	
0x34	Address LSByte	0xDF	Intermediate CRC (partial calculation)
0x12	Address MSByte	0x37	
0x65	Sequence Number	0x67*	Expected CRC (final calculation)
0x67*	Actual Data CRC		

* These two values should match

Under NDA

© 2021 Microchip Technology Inc. and its subsidiaries

TABLE B-6: EXAMPLE CRC CALCULATIONS: SET ADDRESS POINTER – SPI INTERFACE

Bytes Sent by Host (MOSI)		CRC Calculations	
0x02	Op Code	0xBC	
0x34	Address LSByte	0x4E	Intermediate CRC (partial calculation)
0x12	Address MSByte	0x78	
0x0C	Length LSByte	0x99	
0x00	Length MSByte	0x8D*	Expected CRC (final calculation)
0x8D*	Actual Header CRC		

* These two values should match

TABLE B-7: EXAMPLE CRC CALCULATIONS: READ RESPONSE HEADER – SPI INTERFACE

Bytes Sent by Device (MISO)		CRC Calculations	
0x82	Op Code	0x30	Intermediate CRC (partial calculation)
0x34	Address LSByte	0x61	
0x12	Address MSByte	0x1A	
0x0C	Length LSByte	0x40	
0x00	Length MSByte	0x46*	Expected CRC (final calculation)
0x46*	Actual Header CRC		

* These two values should match

Table B-8 gives the sequence of the read bytes with the intermediate calculations of the CRC.

TABLE B-8: EXAMPLE CRC CALCULATIONS: MESSAGE DATA

Bytes Sent by Device		CRC Calculations	
0x01	Report ID	0x5E	Intermediate CRCs (partial calculations)
0x9B	Data	0xF5	
0xA0		0xE4	
0xA5		0x18	
0xAA		0x8E	
0xAF		0x7D	
0xB4		0x56	
0xB9		0xA8	
0xCC		0x04	
0xDE		0x29	
0xD9	Sequence number	0x74*	Expected CRC (final calculation)
0x74*	Actual Data CRC		

* These two values should match

The following code shows how to use the example `crc8()` function given in Section B.2.2 “8-bit CRC Algorithm” to calculate the CRC checksum for the data in the Message Processor T5 object.

```
uint8_t crc = 0;
uint8_t data_in;

for(i=0; i<MESSAGEPROCESSOR_SIZE; i++)
{
    data_in = read_byte();
    crc = crc8(crc, data_in);
}

if(crc == 0)
{
    /* CRC is OK - do something appropriate */
    crc_pass();
}
else
{
    /* CRC failed - handle error */
    crc_fail();
}
```


Under NDA

© 2021 Microchip Technology Inc. and its subsidiaries

APPENDIX C: BOOTLOADING PROCEDURE

The maXTouch devices have a common bootloading procedure that allows them to be forced into a special application update mode of operation.

For more information on the bootloading procedure, refer to the following application note:

- Application Note: MXTAN0216 – *Bootloading Procedure for maXTouch Devices*

C.1 Bootloader ID

The bootloader ID for the mXT640UD-CCUBHA1 Family is 0xB9. The mXT640UD-CCUBHA1 Family bootloader uses extended ID mode.

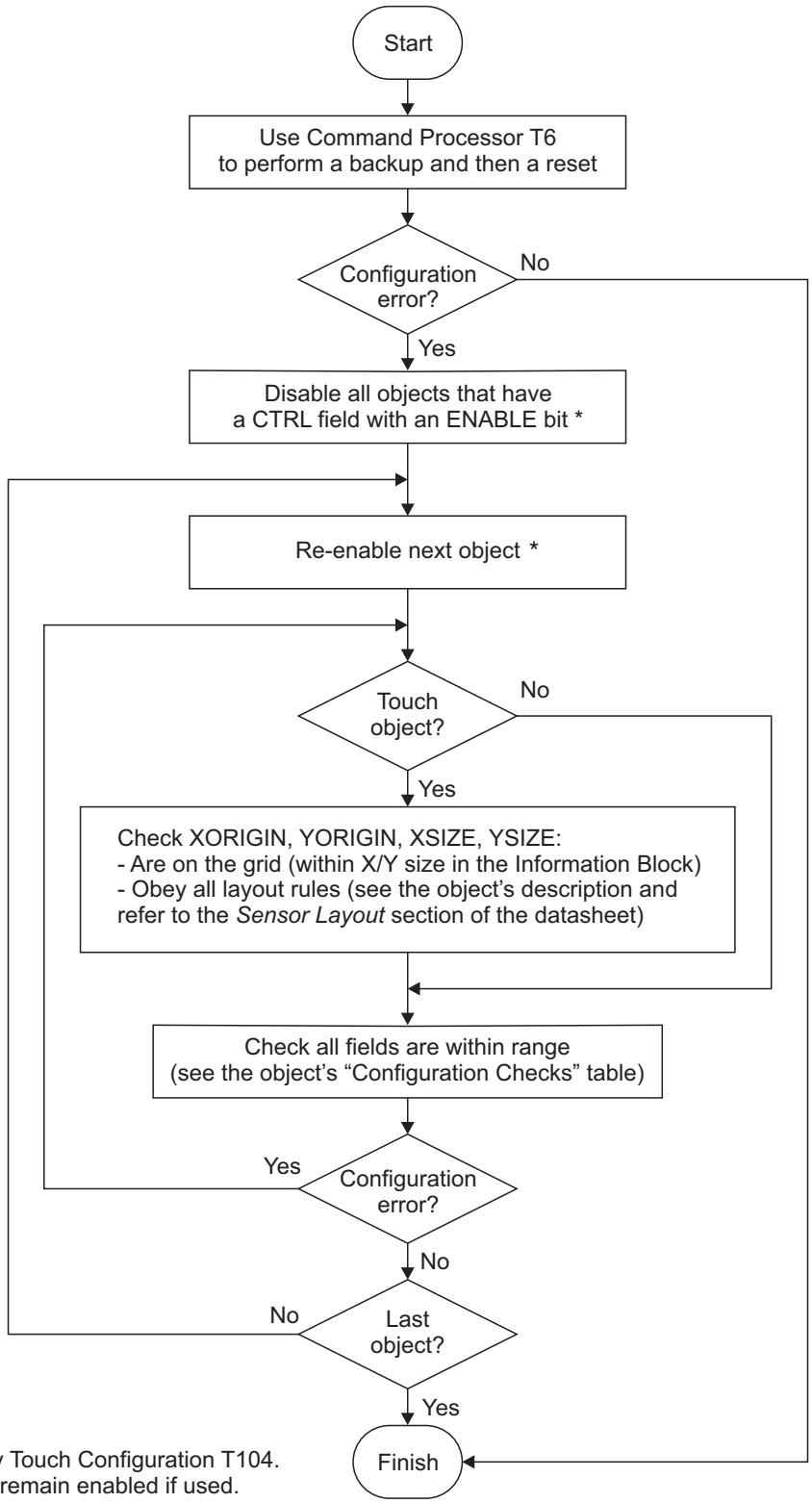
C.2 I^2C address

The I^2C address for bootloader mode on the mXT640UD-CCUBHA1 Family is 0x27.

APPENDIX D: LOCATING CONFIGURATION ERRORS

Figure D-1 shows the algorithm for locating configuration errors. See Section 1.7 “Configuration Checks” for more information on the configuration checks performed by the device.

FIGURE D-1: ALGORITHM FOR LOCATING CONFIGURATION ERRORS



* Except Auxiliary Touch Configuration T104. This object must remain enabled if used.

Under NDA

© 2021 Microchip Technology Inc. and its subsidiaries

APPENDIX E: FUTURE INFORMATION BLOCK T254 OBJECT

E.1 Introduction

Currently the Object Table entries are held within the Information Block for the device. The current format, however, limits the object type codes to 8 bits. It is anticipated that at some time in the future new objects will be issued with 16-bit type codes (that is, codes above 255). The Information Block T254 object will therefore be added to the protocol to accommodate Object Table elements with 16-bit type codes. This means that if the Information Block T254 object is present on a device, it must be checked for any further Object Table entries.

NOTE	It is important for future compatibility that any *current* driver code is written to allow for the presence of the Information Block T254 object when it is eventually used. The host driver code must parse the Object Table, as at present, and also process the Information Block T254 object if it is found to be present on the device.

This appendix gives the format of the future Information Block T254 object so that the driver code can be written accordingly.

E.2 Information Block T254

This object acts as an extension to the Information Block at the start of the device's memory map and holds additional Object Table entries. Specifically, it hold entries for those objects that have 16-bit type codes (that is, type codes above 255).

TABLE 6-87: CONFIGURATION FOR INFORMATION BLOCK T254 (GEN_INFOBLOCK16BIT_T254)

Byte	Field	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0 – 6	Object Table	Object Table element 1 (7 bytes)							
7 – 13	Object Table	Object Table element 2 (7 bytes)							
...									
*m**7 – (*m**7)+6	Object Table	Last Object Table element (7 bytes)							
(*m*+1)*7 – (*m*+1)*7+2	Checksum	24-bit checksum (3 bytes)							

Object Table Element Fields

These 7-byte fields hold the additional elements in the Object Table. The format of each element is similar to that of the Object Table elements held directly in the main Object Table for the device (see Section 1.5 "Object Table"), except that the type code occupies two bytes instead of one. The format of each element is given in Table E-1.

TABLE E-1: OBJECT TABLE ELEMENT FORMAT

Byte	Description
0	Type code LSByte
1	Type code MSByte
2	Start position LSByte
3	Start position MSByte
4	Size – 1
5	Instances – 1
6	Number of Report IDs per instance

Type code (2 bytes): This 16-bit field holds the unique type code for the object to identify it. This is the number after the "_T" suffix at the end of the object's internal name as given in the object descriptions. For example, the type code for the Command Processor T6 is 6 (from GEN_COMMANDPROCESSOR_T6).

Start position (2 bytes): This 16-bit field holds the start location of the object in the memory map.

Size – 1 (1 byte): This field holds the size (minus 1) of the object in the memory map. This is stored as Size-1, so it is effectively the offset to the end of the object.

Instances – 1 (1 byte): This holds the number of instances (minus 1) of the object in the memory map. The number of instances can be calculated by adding 1 to this number (see Section 1.6.2 “Object Instances”). The different instances of an object are arranged consecutively in the memory map.

Number of Report IDs per instance (1 byte): This field holds the number of report IDs for each instance of the object. See Section 1.5.6 “Report IDs”for more information on report IDs.

Checksum Field

A checksum of the contents of the Information Block T254 object. This allows the host to check that the Object Table elements stored in the Information Block T254 object have been read correctly over the communications interface. See Appendix B “Checksum Calculation” for details on calculating the checksum.

Under NDA
© 2021 Microchip Technology Inc. and its subsidiaries

APPENDIX F: ASSOCIATED DOCUMENTS

NOTE Some of the documents listed below are available under NDA only.

Microchip maXTouch Documents

The following documents are available on the Microchip website, except where indicated.

Product Documentation

- *mXT640UD-CCUBHA1 1.0 Datasheet*
- *mXT448UD-CCUBHA1 1.0 Datasheet*

Touchscreen Design and PCB/FPCB Layout Guidelines

- Application Note: MXTAN0208 – *Design Guide for PCB Layouts for maXTouch Touch Controllers*
- Application Note: QTAN0080 – *Touchscreens Sensor Design Guide*
- Application Note: AN2683 – *Edge Wiring for Self Capacitance maXTouch Touchscreens*

Configuring and Tuning the Device

- *maXTouch U Series Tuning Guide* (distributed with Microchip approval only)
- Application Note: AN2659 – *Edge correction on a maXTouch Touch Sensor: an Empirical Method*

Miscellaneous

- Application Note: QTAN0050 – *Using the maXTouch Debug Port*
- Application Note: MXTAN2016 – *Bootloading Procedure for maXTouch Devices*

Tools

- *maXTouch Studio User Guide* (accessible as on-line help from within maXTouch Studio)

Device Drivers

- Application Note: MXTAN0213 – *Interfacing with maXTouch Touchscreen Controllers*
- Application Note: AN3084 – *Using the maXTouch Linux Driver*
- Application note: MXT0201 – *Object Server Configuration File Format*

External Documents

The following documents are not supplied by Microchip. To obtain any of the following documents, please contact the relevant organization.

Standards

- IEC 60730-1, *Automatic electrical controls – Part 1: General requirement*, Edition 5.1, 2015-12
- UL 60730-1, *Automatic Electrical Controls – Part 1: General Requirements*, Edition 5, 2016-08-03

Communication Interfaces

- UM10204, *I^2C bus specification and user manual*, Rev. 6 — 4 April 2014
  Available from NXP
- AN991/D, *Using the Serial Peripheral Interface to Communicate Between Multiple Microcomputers*, Rev. 1 January 2002
  Available from NXP

APPENDIX G: REVISION HISTORY

Revision A (September 2021)

Initial edition for firmware revision 1.0.AA – Release

Under NDA

© 2021 Microchip Technology Inc. and its subsidiaries

INDEX

A

Acquisition Configuration T8 object 34
calibration recovery process 34
calibration recovery tuning data mode 42
configuration checks 41
configuration controls 36, 42
ATCHCALST field 39
ATCHCALSTHR field 39
ATCHFRCCALRATIO field 39
ATCHFRCCALTHR field 39
CFG field 40
CHRGTIME field 36
DRIFTST field 37
MEASALLOW field 40
SYNC field 38
TCHAUTOCAL field 37
TCHDRIFT field 36
diagnostic debug data 42
mutual capacitance recovery process 35
self capacitance measurements
calibration and touch recovery process 243
Active to idle timeout 29
ADCs, number of
CTE Configuration T46 object 205, 208
Noise Suppression T72 object 138
Self Capacitance Configuration T111 object 237, 242, 245
Adjacent key suppression technology 45
AKS. See *Adjacent key suppression*
Alive signal 180
Antitouch calibration 40
Auxiliary Touch Configuration T104 object 227
configuration checks 229
configuration controls 227
CTRL field 227
XGAIN field 227
XINTHYST field 228
XINTTHR field 228
XTCHHYST field 228
XTCHTHR field 228
YGAIN field 228
YINTHYST field 229
YINTTHR field 229
YTCHHYST field 229
YTCHTHR field 228

B

Background scanning and touch hold-off qualification
Noise Suppression T72 object 137
Self Capacitance Noise Suppression T108 object 164
BIST self tests 180, 185
overrun message 186
report period 180
reporting test results 186
Byte order 11

C

Calibration recovery process 34
Calibration recovery tuning data mode 42
Charge time 36
Checksum
24-bit 265
8-bit 267
CHG_I2C line 22
host retriggering mode 199
mode selection 199
CHG_SPI line 22
host retriggering mode 199
Classes of objects 9
Command Processor T6 object 24
configuration controls 24
BACKUPNV field 24
CALIBRATE field 24
DEBUGCTRL field 25
DEBUGCTRL2 field 26
DIAGNOSTIC field 26
REPORTALL field 25
RESET field 24
message data 27
CHECKSUM field 28
STATUS field 27
Communications Configuration T18 object 198
configuration controls 198
CFG field 200
COMMAND field 200
CTRL field 199
DISRPTMASK[] field 200
instances 198
primary communications interface 198
secondary communications interface 199
Configuration and tuning 12
Configuration checks 11
Configuration values 10
CTE Configuration T46 object 205
ADCs per X 208
configuration checks 211
configuration controls 207
ACTVSYNCSPERX field 205, 208
ADCSPERSYNC field 205, 208
CFG field 210
IDLESYNCSPERX field 205, 208
INRUSHCFG field 209
PULSESPERADC field 205, 208
SPREADSPECVAR field 210
SYNCDELAY field 209
SYNCPERIOD field 210
XSLEW field 208
XVOLTAGE field 209
YPHASEDELAY field 211
event triggers 221
lost pulse synchronization signal 206
message data 212
STATUS field 212
number of ADCs 205
pulse synchronization 205
spread spectrum 206
Custom data 204
Customer Change Notification Service 287
Customer Notification Service 287
Customer Support 287

D

Data Container Controller T118 object 253
configuration controls 253
CRC field 254
CTRL field 253
CYCCNT field 254
DATATYPE field 253, 254
HEADERTYPE field 254

TIMESTAMP field 254
message data..... 255
DATAREADY field 255
Data Container T117 object 251
configuration controls..... 251
DATA[] field..... 251
Debug data
See *Diagnostic Debug T37 object*
Deltas 11
mutual capacitance debug data 15, 25, 26
self capacitance debug data 16, 27
self capacitance single-ended debug data..... 27
shifting..... 12
Diagnostic Debug T37 object 14
configuration controls..... 19
DATA[] field..... 20
MODE field..... 19
PAGE field 20
DC data mode..... 19
delta mode (mutual capacitance)..... 15
delta mode (self capacitance) 16
device information mode..... 19
key array delta mode..... 51
key array reference mode 51
key array signal mode 51, 52
reference mode (mutual capacitance)..... 16
reference mode (self capacitance)..... 17
signal mode (self capacitance)..... 18
system diagnostic debug modes..... 14
Direct memory access (DMA) transfer 259
Distance touch suppression 112
Drift compensation 36
Driven shield
open circuit..... 195
signal limits test..... 195
Dual X drive..... 139
Dynamic Configuration Container T71 object 224
configuration controls..... 224
DATA[] field..... 224
Dynamic Configuration Controller T70 object 215
configuration checks 222
configuration controls..... 217
CTRL field 217
DSTOFFSET field 222
EVENT field 218
LENGTH field..... 222
OBJINST field 222
OBJTYPE field 222
SRCOFFSET field..... 222
event triggers 215
message data..... 223
STATUS field 223

E

Edge correction
Multiple Touch Touchscreen T100 object 53
Passive Stylus T47 object..... 117
Edge noise scaling 129
Edge snapping
Multiple Touch Touchscreen T100 object 81, 82
Edge touch suppression..... 112
Evaluation grip suppression mode 109
Event triggers 215
calibration..... 218
Dynamic Configuration Controller T70 object 215
GPIO pin 219
Lens bending force 221
moisture processing 220
noise suppression.....218
Noise Suppression T72 object.....139
power OK.....221
pulse synchronization 221
touch class.....219, 220, 225
touch detection 218
touchscreen suppression.....218

F

Fast detect integration (Fast DI) 30
Finger touches 263
Force on touchscreen, correcting for 128
Frame synchronization 38
Frequency hopping
Noise Suppression T72 object.....135
Self Capacitance Noise Suppression T108 object 164
touch processing 136

G

Gap time 236
Generic keys.....44
Gestures
one-touch gestures.....96
two-touch gestures 105
Glove Detection T78 object 152
configuration checks.....155
configuration controls 152
CONFTHR field 153
CTRL field.....153
DISCRIMTHR field 154
GLOVEMODETO field.....153
MINAREA field.....153
MINDIST field 153
SCTGLOVEHYST field.....154
SCTGLOVETHR field.....154
SCTMINAREA field 154
SUPTO field.....154
SYNCSPERX field.....154
glove confidence mode.....152
glove normal mode 152
message data 155
Glove touches.....152, 264
GPIO Configuration T19 object.....201
configuration controls 201
CTRL field.....202
DIR field.....202
INTPULLUP field 202
OUT field 202
REPORTMASK field.....202
TRIGGER[] field.....202
WAKE field 202
event triggers.....219
message data 203
GPIO pin event triggers 219
Grip suppression
self capacitance measurements 174
Grip Suppression T40 object 108
configuration controls 110
CTRL field.....111
NUMTCHBEFSUP field.....111
XHIGRIP field 111
XLOGRIP field.....111
YHIGRIP field 111
YLOGRIP field.....111

Under NDA

© 2021 Microchip Technology Inc. and its subsidiaries

evaluation mode ... 109
grip mode ... 109
grip suppression boundary ... 108
message data ... 111
number of touches ... 108

H

Hardware debug interface ... 201
Heartbeat signal ... 180
Host communications ... 198, 199

I

I²C communications
8-bit CRC ... 268, 270
CHG_I2C line ... 22
ID information ... 6
Ignore Nodes Controller T145 object ... 260
configuration controls ... 260
CTRL field ... 260
Ignore Nodes T141 object ... 177
configuration controls ... 177
YMASK field ... 177
Information block ... 6
Information Block T254 object ... 275
configuration controls
Checksum field ... 276
Object Table Element Fields ... 275
Internet Address ... 287

K

Key Array T15 object ... 44
configuration checks ... 49
configuration controls ... 46
AKSCFG field ... 47
CFG field ... 49
CTRL field ... 46
GAIN field ... 47
TCHDI field ... 48
TCHHYST field ... 49
TCHTHR field ... 48
XORIGIN field ... 46
XSIZE field ... 46
YORIGIN field ... 46
YSIZE field ... 46
debug data ... 51
key delta mode ... 51
key reference mode ... 51
key signal mode ... 51
generic keys ... 44
message data ... 50
KEYSTATE field ... 51
STATUS field ... 50
touch threshold ... 45
Key ID Configuration T16 object ... 95
configuration controls ... 95
CTRL field ... 95
KEYID[] field ... 95
Key Thresholds T14 object ... 94
configuration controls ... 94
EXTRAHYST[] field ... 94
EXTRATHR[] field ... 94

L

Large touch suppression ... 112, 264
Lens bending correction ... 128
Lens Bending T65 object ... 128
configuration checks ... 132
configuration controls ... 128
ATCHRATIO field ... 131
CTRL field ... 128
EXFRCTHR field ... 131
EXFRCTHRHYST field ... 132
EXFRCTO field ... 132
FORCEDI field ... 130
FORCEHYST field ... 131
FORCESCALE field ... 130
FORCETHR field ... 130
FORCETHRHYST field ... 130
GRADTHR field ... 129
LPFILTCOEF field ... 130
YHINOISEDIV field ... 129
YHINOISEMUL field ... 129
YLONOISEDIV field ... 129
YLONOISEMUL field ... 129
edge noise scaling ... 129
event triggers ... 221
message data ... 133
FORCE field ... 133
STATUS field ... 133
noise suppression for shieldless sensor designs ... 128
Low power mode ... 263

M

Mark time ... 236
Maximum screen area suppression ... 112
Maximum touch suppression ... 112
Measurement processing ... 262
finger touches ... 263
glove touches ... 152, 264
in active and idle modes ... 263
measurement processing objects ... 262
objects used ... 262
passive stylus touches ... 117, 264
suppressed large touch ... 264
touch classification ... 263
touch detection ... 263
Measurement types ... 262
selecting ... 40
Memory map structure ... 5
Message Count T144 object ... 259
configuration controls ... 259
CHECKSUM field ... 259
COUNT field ... 259
Message Processor T5 object ... 22
configuration controls ... 22
CHECKSUM field ... 23
MESSAGE field ... 23
REPORTID field ... 23
SEQNUM field ... 23
Microchip Internet Web Site ... 287
Minimum touch threshold ... 136
Moisture compensation ... 156
Moisture degraded mode ... 156
Multiple Touch Touchscreen T100 object ... 53
configuration checks ... 85
configuration controls ... 55
AKSCFG field ... 64
AMPLCOEFF field ... 80
AMPLHYST field ... 77
AMPLOFFSET field ... 80
CALCFG field ... 74
CFG1 field ... 57

CFG2 field 78
CTRL field 56
CUTOFFTHR field 72
DBGXOFFSET field 84
DBGXSIZE field 84
DBGYOFFSET field 84
DBGYSIZE field 84
DXGAIN field 71
DXTHRSF field 73
DXXEDGECFG field 70
DXXEDGECFGHI field 77
DXXEDGEDIST field 70
DXXEDGEDISTHI field 78
EDGESNPCFGIN field 81
EDGESNPCFGOUT field 82
GAIN field 71
INTTHR field 71
INTTHRHYST field 77
JLMMOVINTTHR field 81
JLMMOVTHR field 81
JUMPLIMIT field 74
JUMPLIMITMOV field 80
MOVFILTER field 74
MOVHYSTCFG field 79
MOVHYSTI field 76
MOVHYSTN field 76
MOVPRED field 75
MOVSMOOTH field 75
MRGHYST field 73
MRGTHR field 72
NEXTTCHDI field 74
NOISESF field 72
NUMTCH field 64
SCRAREAHYST field 77
SCRAUX field 58
TCHAUX field 60
TCHDIDOWN field 73
TCHDIUP field 73
TCHEVENTCFG field 64
TCHHYST field 71
TCHTHR field 71
XBFSREJRANGE field 83
XEDGECFG field 70
XEDGECFGHI field 77
XEDGEDIST field 70
XEDGEDISTHI field 78
XHICLIP field 67
XLOCLIP field 67
XORIGIN field 66
XPITCH field 67
XRANGE field 67
XSIZE field 66
XYCFG field 65
YBFSREJRANGE field 83
YEDGECFG field 70
YEDGECFGHI field 77
YEDGEDIST field 70
YEDGEDISTHI field 78
YHICLIP field 67
YLOCLIP field 67
YORIGIN field 66
YPITCH field 67
YRANGE field 67
YSIZE field 66
debug data 90
touch status data mode 91
touchscreen mode 92
edge correction 53
edge snapping 81, 82
event triggers 218
message data 87
AUXDATA[] field 88, 90
SCRSTATUS field 88
TCHSTATUS field 89
XPOS field 90
YPOS field 90
SUP message 111
touch threshold 53
UNSUP message 111
Mutual capacitance measurements
gain, setting 47, 71
noise suppression 134–151
selecting 40
thresholds, setting 45, 53
touch recovery process 35

N

Noise lines threshold 138
Noise measurement 134
Noise notification mechanisms
Noise Suppression T72 object 134
Self Capacitance Noise Suppression T108 object 163
Noise suppression
event triggers 218
mutual capacitance 134–151
self capacitance 163–173
shieldless sensor designs 128
Noise Suppression T72 object 134
ADCs per X 138
background scanning and touch hold-off qualification 137
configuration checks 149
configuration controls 140
BGNLRATIONOIS field 148
BGNLRATIOSTAB field 148
BGNLRATIOVNOI field 148
BGSCAN field 144
BLKNLTHR field 145
CALCFG1 field 142
CFG1 field 142
CFG2 field 142
CFG3 field 145
CTRL field 142
FALLNLTHR field 144
HOPCNT field 143
HOPCNTPER field 143
HOPEVALTO field 143
HOPST field 143
INCNLTHR field 143
MINNLTHR field 143
MINTHRADJ field 144
NLGAINDUALX field 143
NLGAINSINGX field 145
NLRATIO field 148
NLTHRLIMIT field 144
NLTHRMARGIN field 144
NOISCNT field 147
NOISCTRL field 145
NOISFREQ[] field 146
NOISHIGHNLTHR field 147
NOISLOWNLTHR field 147
NOISNOTCHAPX[] field 146
NOISTCHAPX[] field 146

Under NDA

© 2021 Microchip Technology Inc. and its subsidiaries

NOTCHMINDIFF field ... 147
NOTCHMINHOP field ... 148
STABCTRL field ... 145
STABFREQ[] field ... 146
STABHIGHNLTHR field ... 147
STABNOTCHAPX[] field ... 146
STABTCHAPX[] field ... 146
TCHMINDIFF field ... 147
TCHMINHOP field ... 148
VNOICNT field ... 147
VNOICTRL field ... 145
VNOIFREQ[] field ... 146
VNOILOWNLTHR field ... 147
VNOINOTCHAPX[] field ... 146
VNOITCHAPX[] field ... 146
dual X drive ... 139
event triggers ... 139, 218
frequency hopping ... 135
general initial settings ... 139
message data ... 150
MINTCHTHR field ... 151
NLTHR field ... 151
NOISELVL field ... 151
PKNOISELVL field ... 151
STATUS1 field ... 150
STATUS2 field ... 151
minimum touch threshold ... 136
noise lines threshold ... 138
noise measurement ... 134
noise notification mechanisms ... 134
resetting the noise suppression algorithm ... 139
touch processing ... 136
transitions between noise states ... 134
Number of instances ... 7
Number of touches
grip suppression ... 108
Multiple Touch Touchscreen T100 object ... 64

O

Object Protocol ... 5
byte order ... 11
classes of objects ... 9
compatibility of object versions ... 11
configuration checks ... 11
configuration values ... 10
delta values ... 11
ID information ... 6
information block ... 6
memory map structure ... 5
number of instances ... 7
object instances ... 9
object table ... 7
object types ... 7
objects ... 9
report IDs ... 8, 200
Object table ... 7
Object types ... 7
Objects ... 9
Acquisition Configuration T8 ... 34
Auxiliary Touch Configuration T104 ... 227
Command Processor T6 ... 24
Communications Configuration T18 ... 198
compatibility of object versions ... 11
configuration and tuning ... 12
configuration checks ... 11
configuration values ... 10
CTE Configuration T46 ... 205
Data Container Controller T118 ... 253
Data Container T117 ... 251
Diagnostic Debug T37 ... 14
Dynamic Configuration Container T71 ... 224
Dynamic Configuration Controller T70 ... 215
Glove Detection T78 ... 152
GPIO Configuration T19 ... 201
Grip Suppression T40 ... 108
Ignore Nodes Controller T145 ... 260
Ignore Nodes T141 ... 177
Information Block T254 object ... 275
instances ... 9
Key Array T15 ... 44
Key ID Configuration T16 ... 95
Key Thresholds T14 ... 94
Lens Bending T65 ... 128
Message Count T144 ... 259
Message Processor T5 ... 22
Multiple Touch Touchscreen T100 ... 53
Noise Suppression T72 ... 134
One-touch Gesture Processor T24 ... 96
Passive Stylus T47 ... 117
Power Configuration T7 ... 29
Retransmission Compensation T80 ... 156
Self Capacitance Configuration T111 ... 236
Self Capacitance Global Configuration T109 ... 231
Self Capacitance Grip Suppression T112 ... 174
Self Capacitance Measurement Configuration T113 ... 248
Self Capacitance Noise Suppression T108 ... 163
Self Capacitance Tuning Parameters T110 ... 235
Self Capacitance Voltage Modulation T133 ... 256
Self Test Control T10 ... 180
Self Test Pin Faults T11 ... 191
Self Test Signal Limits T12 ... 194
Shieldless T56 ... 126
Timer T61 ... 213
Touch Event Trigger T79 ... 225
Touch Suppression T42 ... 112
Two-touch Gesture Processor T27 ... 105
User Data T38 ... 204
On-demand self tests ... 180, 182, 185
One-touch Gesture Processor T24 object ... 96
configuration controls ... 98
CTRL field ... 98
DRAGTHR field ... 102
DRAGTO field ... 100
FLICKTHR field ... 102
FLICKTO field ... 100
GESTEN field ... 99
LPRESSTO field ... 101
NUMGEST field ... 99
PROCESS field ... 99
REPPRESSTO field ... 101
SPRESSTO field ... 100
TAPTHR field ... 102
TAPTO field ... 100
THROWTHR field ... 102
message data ... 103
DIR field ... 103
DIST field ... 104
STATUS field ... 103
XPOSMSB field ... 103
XYPOSLSB field ... 103
XYPOSMSB field ... 103
YPOSMSB field ... 103

P

Passive Stylus T47 object 117
configuration checks 124
configuration controls 118
AMPLTHR field 120
CFG field 120
CFG2 field 123
CONFTHR field 120
CONTMAX field 119
CTRL field 119
DXXEDGECTRL field 122
DXXEDGECTRLHI field 122
DXXEDGEDIST field 122
DXXEDGEDISTHI field 123
HOVERSUP field 120
MAXNUMSTY field 121
MAXTCHAREA field 119
MOVFILTER field 123
MOVHYSTCFG field 124
MOVPRED field 123
MOVSMOOTH field 123
SATBDXXHI field 124
SATBDXXLO field 124
SATBXHI field 123
SATBXLO field 123
SATBYHI field 123
SATBYLO field 123
STABILITY field 119
STYSHAPE field 120
SUPCLASSMODE field 121
SUPSTYTO field 120
SUPTO field 121
SYNCSPERX field 120
XEDGECTRL field 121
XEDGECTRLHI field 122
XEDGEDIST field 121
XEDGEDISTHI field 123
XPOSADJ field 120
YEDGECTRL field 121
YEDGECTRLHI field 122
YEDGEDIST field 121
YEDGEDISTHI field 123
YPOSADJ field 120
edge correction 117
message data 125
Passive stylus touches 264
Pin fault self tests 184, 191
debug data 192, 193
message data 189
POST self tests 180, 185
reporting test results 186
Power Configuration T7 object 29
active to idle timeout 29
configuration controls 31
ACTV2IDLETO field 31
ACTVACQINT field 31
ACTVACQINTFINE field 33
CFG field 32
CFG2 field 32
IDLEACQINT field 31
IDLEACQINTFINE field 33
power modes 29, 31
Power self tests 183
Primary communications interface 198
Pulse synchronization
CTE Configuration T46 object 205
event triggers 221
lost pulse synchronization signal 206, 238
Self Capacitance Configuration T111 object 237
Pulses per ADC 205

R

References
mutual capacitance debug data 16, 25, 26
self capacitance debug data 17, 27
self capacitance single-ended debug data 27
Report IDs 8
disabling in message output 200
Retransmission compensation 156
Retransmission Compensation T80
moisture degraded mode 156
Retransmission Compensation T80 object 156
configuration checks 161
configuration controls 157
ATCHTHR field 158
COMPCFG field 160
COMPGAIN field 158
COMPSTRTHR field 160
COMPTHR field 158
CTRL field 157
MOISTCFG field 159
MOISTCFG2 field 160
MOISTCFG3 field 161
MOISTDEGTHR field 161
MOISTINVATCHTHR field 159
MOISTTHR field 159
MOISTVLDTHRSF field 160
TARGETDELTA field 158
event triggers 220
message data 162
STATUS field 162
Revision ID 19

S

Secondary communications interface 199
Self Capacitance Configuration T111 object 236
configuration checks 247
configuration controls 239
ACTVSYNCSPERL field 242
ADCSPERSYNC field 237, 245
ALTDELAYTIMEX field 244
ALTINTTIMEX field 243
CALRECSTR field 243
CTRL field 241
DBGCTRL field 242
DCCALERRRATIO field 245
DCCALRECSTR field 245
DCDRIFT field 244
DCFILTER field 244
DCGAINSF field 246
DCIDLESLEWMIN field 246
DCPROCTHRX field 247
DCPROCTHRY field 247
DCTHRX field 246
DCTHRY field 246
DELAYTIME field 242
DRIFT field 242
DRIFTST field 243
IDLESYNCSPERL field 242
INRUSHCFG field 243
INTTIME field 242

Under NDA

© 2021 Microchip Technology Inc. and its subsidiaries

SYNCDELAY field ... 245
instances ... 236
lost pulse synchronization signal ... 238
pulse synchronization ... 237
self capacitance measurements ... 239
self capacitance single-ended measurements ... 240
Self Capacitance Global Configuration T109 object ... 231
configuration checks ... 233
configuration controls ... 231
CMD field ... 232
CMDONRESET field ... 232
CTRL field ... 231
TUNECFG field ... 232
TUNPARAM[] field ... 232
TUNPARAM[0] field ... 233
message data ... 234
CMD field ... 234
ERRORCODE field ... 234
Self Capacitance Grip Suppression T112 object ... 174
configuration checks ... 175
configuration controls ... 174
APPRSHPTHRX field ... 175
APPRSHPTHRY field ... 175
CHGSHPTHRX field ... 174
CHGSHPTHRY field ... 174
CTRL field ... 174
message data ... 176
GRIPDETECT field ... 176
STATUS field ... 176
shape thresholds ... 174
Self Capacitance Measurement Configuration T113 object ... 248
configuration checks ... 250
configuration controls ... 248
CTRL field ... 248
NODEAREASFCM field ... 249
NODEAREASFMC field ... 249
XGAIN field ... 249
YGAIN field ... 249
Self capacitance measurements
configuration ... 239
gain, setting ... 227, 248
global configuration ... 231
glove touches ... 152
grip suppression ... 174
noise suppression ... 163–173
selecting ... 40
thresholds, setting ... 227
Self Capacitance Noise Suppression T108 object ... 163
background scanning and touch hold-off qualification ... 164
configuration checks ... 171
configuration controls ... 166
BLKNLTDTHR field ... 171
CALCFG1 field ... 168
CFG1 field ... 168
CFG2 field ... 168
CFG3 field ... 168
CTRL field ... 167
HOPST field ... 169
IIRCOEFF field ... 169
MINNLTDDIFF field ... 169
MINNLTDHOP field ... 169
NLGAIN field ... 168
NOISCNT field ... 171
NOISCTRL field ... 169
NOISFREQ[] field ... 170
NOISHINLTDTHR field ... 170
NOISLONLTDTHR field ... 170
NOISNOTCHAPX[] field ... 170
NOISTCHAPX[] field ... 170
STABCTRL field ... 169
STABFREQ[] field ... 170
STABHINLTDTHR field ... 170
STABNOTCHAPX[] field ... 170
STABTCHAPX[] field ... 170
VNOICNT field ... 171
VNOICTRL field ... 169
VNOIFREQ[] field ... 170
VNOILONLTDTHR field ... 170
VNOINOTCHAPX[] field ... 170
VNOITCHAPX[] field ... 170
frequency hopping ... 164
message data ... 172
PKNLTDDIFF field ... 173
PKNLTDHOP field ... 173
PKNLTDLVL field ... 173
STATUS1 field ... 172
STATUS2 field ... 173
SUSHOPS field ... 173
noise notification mechanisms ... 163
suggested initial settings ... 165
transitions between noise states ... 163
Self capacitance pulse shape ... 236
Self capacitance single-ended measurements
configuration ... 240
gain, setting ... 248
See also *Self capacitance measurements*
selecting ... 40
Self Capacitance Tuning Parameters T110 object ... 235
configuration controls ... 235
PARAMS[] field ... 235
instances ... 235
Self Capacitance Voltage Modulation T133 object ... 256
configuration checks ... 258
configuration controls ... 257
CFG field ... 258
CTRL field ... 257
DRVPULSE[] field ... 258
Self Test Control T10 object ... 180
configuration controls ... 184
BISTCFG field ... 185
CMD field ... 185
CTRL field ... 184
MAXHOLDOFF field ... 186
MINTSTTIMEPERCYC field ... 186
OVERRUNTIME field ... 186
POSTCFG field ... 185
RPTINTERVAL field ... 186
TESTINTERVAL field ... 185
message data ... 186
CMD field ... 187
DATA[] field ... 188
DS_PIN field ... 190
INFO field ... 187, 188
SEQ_NUM field ... 189
STATUS field ... 187
TEST field ... 187, 188
TYPE_INSTANCE field ... 189
TYPE_NUM field ... 188
X_PIN field ... 190
Y_PIN field ... 190
Self Test Pin Faults T11 object ... 191
configuration controls ... 191

PINDWELLUS field 192
PINTHR field 192
debug data 192
message data 192
Self Test Signal Limits T12 object 194
configuration controls 195
CTRL field 196
DSGAIN field 197
DXGAIN field 196
GAIN field 196
SIGLIMLO field 196
SIGLIMRANGE field 197
SIGLIMUP field 196
message data 197
Self tests 183
BIST tests 180, 185
CTE and touch system 184
debug data 192
internal system 183
On-demand tests 180, 182, 185
pin fault tests 184, 189, 191
POST tests 180, 185
power tests 183
reporting test results 186
signal limit tests 184, 188, 194
Shape thresholds 174
Shieldless T56 object 126
configuration checks 126
configuration controls 126
CTRL field 126
INTDELAY[] field 126
INTTIME field 126
OPTINT field 126
message data 127
STATUS field 127
Signal limit self tests 194
message data 188
Signals
mutual capacitance debug data 25, 26
self capacitance debug data 18, 27
self capacitance single-ended debug data 27
Software tools 12
Space time 236
SPI communications
8-bit CRC 268, 270
CHG_SPI line 22
Spread spectrum 206
SUP message 111
SYNC pin 38, 201, 205
Synchronization
frame synchronization 38
pulse synchronization 205, 237

T

TCHDI fields 30
Timer T61 object 213
configuration controls 213
CMD field 213
CTRL field 213
MODE field 213
PERIOD field 214
message data 214
CNTVAL field 214
STATUS field 214
Timers 213
Touch class event triggers 219, 220, 225
Touch classification 263
Touch detection 263
Touch Event Trigger T79 object 225
configuration controls 225
CTRL field 225
NUMTCHTHR field 226
TCHRELTO field 226
TOUCHCLASS field 225
event triggers 219, 220, 225
Touch Suppression T42 object 112
configuration checks 115
configuration controls 112
CFG field 115
CTRL field 112
DISTHYST field 115
EDGESUPSTRENGTH field 115
MAXAPPRAREA field 113
MAXNUMTCHS field 114
MAXSCRNAREA field 115
MAXTCHAREA field 113
SHAPESTRENGTH field 114
SUPDIST field 114
SUPEXTTO field 114
SUPSTRENGTH field 113
distance touch suppression 112
edge touch suppression 112
event triggers 218
large touch suppression 112
maximum screen area suppression 112
maximum touch suppression 112
message data 116
Touch threshold
Key Array T15 object 45
Multiple Touch Touchscreen T100 object 53
Touchscreen suppression
event triggers 218
Transitions between noise states
Noise Suppression T72 object 134
Self Capacitance Noise Suppression T108 object 163
Two-touch Gesture Processor T27 object 105
configuration controls 105
CTRL field 105
GESTEN field 106
NUMGEST field 106
ROTATETHR field 106
ZOOMTHR field 106
message data 106
ANGLE field 107
SEP field 107
STATUS field 106
XPOSMSB field 107
XYPOSLSB field 107
XYPOSMSB field 107
YPOSMSB field 107

U

UL/IEC 60730 Class B compliance
safety tests 183
UNSUP message 111
User Data T38 object 204
configuration controls 204
DATA[] field 204

W

WWW Address 287

Under NDA

© 2021 Microchip Technology Inc. and its subsidiaries

THE MICROCHIP WEB SITE

Microchip provides online support via our WWW site at www.microchip.com. This web site is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the web site contains the following information:

- **Product Support** – Datasheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQ), technical support requests, online discussion groups, Microchip consultant program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

THE MAXTOUCH WEB SITE

Information on Microchip's maXTouch product line can be accessed via the microchip web site at www.microchip.com, This information is also available for direct access via a short-cut at www.maxtouch.com. The maXTouch web pages contain the following information:

- **Product Information** – Product specifications, brochures, datasheets, protocol guides
- **Tools and Software** – Evaluation kits, maXTouch Studio, software libraries for individual maXTouch touch controllers
- **Training and Support** – Generic application notes and training material for the maXTouch product range

CUSTOMER CHANGE NOTIFICATION SERVICE

Microchip's customer notification service helps keep customers current on Microchip products. Subscribers will receive e-mail notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, access the Microchip web site at www.microchip.com. Under "Support", click on "Customer Change Notification" and follow the registration instructions.

CUSTOMER SUPPORT

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support

Customers should contact their distributor, representative or Field Application Engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

Technical support is available through the web site at: http://microchip.com/support

NOTES:

Under NDA

© 2021 Microchip Technology Inc. and its subsidiaries

Note the following details of the code protection feature on Microchip products:

- Microchip products meet the specifications contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is secure when used in the intended manner, within operating specifications, and under normal conditions.
- Microchip values and aggressively protects its intellectual property rights. Attempts to breach the code protection features of Microchip products is strictly prohibited and may violate the Digital Millennium Copyright Act.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of its code. Code protection does not mean that we are guaranteeing the product is “unbreakable”. Code protection is constantly evolving. Microchip is committed to continuously improving the code protection features of our products.

This publication and the information herein may be used only with Microchip products, including to design, test, and integrate Microchip products with your application. Use of this information in any other manner violates these terms. Information regarding device applications is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. Contact your local Microchip sales office for additional support or, obtain additional support at https://www.microchip.com/en-us/support/design-help/client-supportservices.

THIS INFORMATION IS PROVIDED BY MICROCHIP "AS IS". MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE, OR WARRANTIES RELATED TO ITS CONDITION, QUALITY, OR PERFORMANCE.

IN NO EVENT WILL MICROCHIP BE LIABLE FOR ANY INDIRECT, SPECIAL, PUNITIVE, INCIDENTAL, OR CONSEQUENTIAL LOSS, DAMAGE, COST, OR EXPENSE OF ANY KIND WHATSOEVER RELATED TO THE INFORMATION OR ITS USE, HOWEVER CAUSED, EVEN IF MICROCHIP HAS BEEN ADVISED OF THE POSSIBILITY OR THE DAMAGES ARE FORESEEABLE. TO THE FULLEST EXTENT ALLOWED BY LAW, MICROCHIP'S TOTAL LIABILITY ON ALL CLAIMS IN ANY WAY RELATED TO THE INFORMATION OR ITS USE WILL NOT EXCEED THE AMOUNT OF FEES, IF ANY, THAT YOU HAVE PAID DIRECTLY TO MICROCHIP FOR THE INFORMATION. Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights unless otherwise stated.

For information regarding Microchip’s Quality Management Systems, please visit www.microchip.com/quality.

Trademarks

The Microchip name and logo, the Microchip logo, Adaptec, AnyRate, AVR, AVR logo, AVR Freaks, BesTime, BitCloud, CryptoMemory, CryptoRF, dsPIC, flexPWR, HELDO, IGLOO, JukeBlox, KeeLoq, Kleer, LANCheck, LinkMD, maXStylus, maXTouch, MediaLB, megaAVR, Microsemi, Microsemi logo, MOST, MOST logo, MPLAB, OptoLyzer, PIC, picoPower, PICSTART, PIC32 logo, PolarFire, Prochip Designer, QTouch, SAM-BA, SenGenuity, SpyNIC, SST, SST Logo, SuperFlash, Symmetricom, SyncServer, Tachyon, TimeSource, tinyAVR, UNI/O, Vectron, and XMEGA are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

AgileSwitch, APT, ClockWorks, The Embedded Control Solutions Company, EtherSynch, Flashtec, Hyper Speed Control, HyperLight Load, IntelliMOS, Libero, motorBench, mTouch, Powermite 3, Precision Edge, ProASIC, ProASIC Plus, ProASIC Plus logo, Quiet-Wire, SmartFusion, SyncWorld, Temux, TimeCesium, TimeHub, TimePictra, TimeProvider, TrueTime, WinPath, and ZL are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Adjacent Key Suppression, AKS, Analog-for-the-Digital Age, Any Capacitor, AnyIn, AnyOut, Augmented Switching, BlueSky, BodyCom, CodeGuard, CryptoAuthentication, CryptoAutomotive, CryptoCompanion, CryptoController, dsPICDEM, dsPICDEM.net, Dynamic Average Matching, DAM, ECAN, Espresso T1S, EtherGREEN, GridTime, IdealBridge, In-Circuit Serial Programming, ICSP, INICnet, Intelligent Paralleling, Inter-Chip Connectivity, JitterBlocker, Knob-on-Display, maxCrypto, maxView, memBrain, Mindi, MiWi, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, MultiTRAK, NetDetach, NVM Express, NVMe, Omniscient Code Generation, PICDEM, PICDEM.net, PICkit, PICtail, PowerSmart, PureSilicon, QMatrix, REAL ICE, Ripple Blocker, RTAX, RTG4, SAM-ICE, Serial Quad I/O, simpleMAP, SimpliPHY, SmartBuffer, SmartHLS, SMART-I.S., storClad, SQI, SuperSwitcher, SuperSwitcher II, Switchtec, SynchroPHY, Total Endurance, TSHARC, USBCheck, VariSense, VectorBlox, VeriPHY, ViewSpan, WiperLock, XpressConnect, and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

The Adaptec logo, Frequency on Demand, Silicon Storage Technology, Symmcom, and Trusted Time are registered trademarks of Microchip Technology Inc. in other countries.

GestIC is a registered trademark of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.

All other trademarks mentioned herein are property of their respective companies.

© 2021, Microchip Technology Incorporated and its subsidiaries.

All Rights Reserved.

ISBN: 978-1-5224-8903-0

Worldwide Sales and Service

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
http://www.microchip.com/support
Web Address:
www.microchip.com

Atlanta
Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

Austin, TX
Tel: 512-257-3370

Boston
Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

Chicago
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

Dallas
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

Detroit
Novi, MI
Tel: 248-848-4000

Houston, TX
Tel: 281-894-5983

Indianapolis
Noblesville, IN
Tel: 317-773-8323
Fax: 317-773-5453
Tel: 317-536-2380

Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608
Tel: 951-273-7800

Raleigh, NC
Tel: 919-844-7510

New York, NY
Tel: 631-435-6000

San Jose, CA
Tel: 408-735-9110
Tel: 408-436-4270

Canada - Toronto
Tel: 905-695-1980
Fax: 905-695-2078

ASIA/PACIFIC

Australia - Sydney
Tel: 61-2-9868-6733

China - Beijing
Tel: 86-10-8569-7000

China - Chengdu
Tel: 86-28-8665-5511

China - Chongqing
Tel: 86-23-8980-9588

China - Dongguan
Tel: 86-769-8702-9880

China - Guangzhou
Tel: 86-20-8755-8029

China - Hangzhou
Tel: 86-571-8792-8115

China - Hong Kong SAR
Tel: 852-2943-5100

China - Nanjing
Tel: 86-25-8473-2460

China - Qingdao
Tel: 86-532-8502-7355

China - Shanghai
Tel: 86-21-3326-8000

China - Shenyang
Tel: 86-24-2334-2829

China - Shenzhen
Tel: 86-755-8864-2200

China - Suzhou
Tel: 86-186-6233-1526

China - Wuhan
Tel: 86-27-5980-5300

China - Xian
Tel: 86-29-8833-7252

China - Xiamen
Tel: 86-592-2388138

China - Zhuhai
Tel: 86-756-3210040

ASIA/PACIFIC

India - Bangalore
Tel: 91-80-3090-4444

India - New Delhi
Tel: 91-11-4160-8631

India - Pune
Tel: 91-20-4121-0141

Japan - Osaka
Tel: 81-6-6152-7160

Japan - Tokyo
Tel: 81-3-6880- 3770

Korea - Daegu
Tel: 82-53-744-4301

Korea - Seoul
Tel: 82-2-554-7200

Malaysia - Kuala Lumpur
Tel: 60-3-7651-7906

Malaysia - Penang
Tel: 60-4-227-8870

Philippines - Manila
Tel: 63-2-634-9065

Singapore
Tel: 65-6334-8870

Taiwan - Hsin Chu
Tel: 886-3-577-8366

Taiwan - Kaohsiung
Tel: 886-7-213-7830

Taiwan - Taipei
Tel: 886-2-2508-8600

Thailand - Bangkok
Tel: 66-2-694-1351

Vietnam - Ho Chi Minh
Tel: 84-28-5448-2100

EUROPE

Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

Denmark - Copenhagen
Tel: 45-4485-5910
Fax: 45-4485-2829

Finland - Espoo
Tel: 358-9-4520-820

France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany - Garching
Tel: 49-8931-9700

Germany - Haan
Tel: 49-2129-3766400

Germany - Heilbronn
Tel: 49-7131-72400

Germany - Karlsruhe
Tel: 49-721-625370

Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Germany - Rosenheim
Tel: 49-8031-354-560

Israel - Ra'anana
Tel: 972-9-744-7705

Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

Italy - Padova
Tel: 39-049-7625286

Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

Norway - Trondheim
Tel: 47-7288-4388

Poland - Warsaw
Tel: 48-22-3325737

Romania - Bucharest
Tel: 40-21-407-87-50

Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

Sweden - Gothenberg
Tel: 46-31-704-60-40

Sweden - Stockholm
Tel: 46-8-5090-4654

UK - Wokingham
Tel: 44-118-921-5800
Fax: 44-118-921-5820

Under NDA
© 2021 Microchip Technology Inc and its subsidiaries