MTCH2120 - Twelve-Channel Capacitive Touch Controller

MTCH2120

Description

The MTCH2120 Capacitive Touch Controller with I^{2}C (Inter-Integrated Circuit) interface enables the application to use water-tolerant touch buttons or proximity sensing. The device features 12 capacitive touch sensors with configurable Driven Shield capability. Touch performance parameters can be adjusted via the I^{2}C interface, enabling the configuration of the MTCH2120 operation to suit the intended use case. Additionally, the device includes Easy Tune enabled out-of-the-box usage even under changing noise conditions without the need for tuning - your fastest way to a touch button interface. To ensure reliable touch detection even in challenging noise environments, the MTCH2120 incorporates advanced algorithms that actively filter out noise from the signal.

The MTCH2120 provides touch/proximity detection sensors that can work through plastic, wood, or metal front panels with Microchip's proprietary Metal over Capacitive (MoC) technology. It supports a variety of conductive materials as sensors, such as copper pads on PCB, silver ink, PEDOT carbon printing on plastic film, and Indium Tin Oxide (ITO) pads.

Features

- Capacitive Touch Sensing
- High Signal Noise Ratio (SNR)
- Configurable via I^{2}C Bus
- Adjustable Sensitivity
- Multi-Stage Active Noise Suppression Filters
- Automatic Environmental Compensation
- Water Tolerant Touch
- EasyTune
- Low power with the option to combine sensors (lump mode)
- Configurable GPIO as an optional host interface
- Support a wide range of Sensor Shapes and Sizes
- Interrupt pin for Touch Status change
- Option to Disable or Suspend Sensor
- Option to save configuration
- Brown-Out Protection
- Operating Voltage Range:
  - 3 to 5V
- Operating Temperature:
  - -40°C to +105°C

Table of Contents

1. Pin Details

1.1 Pin Description

Table 1-1. Pin Description

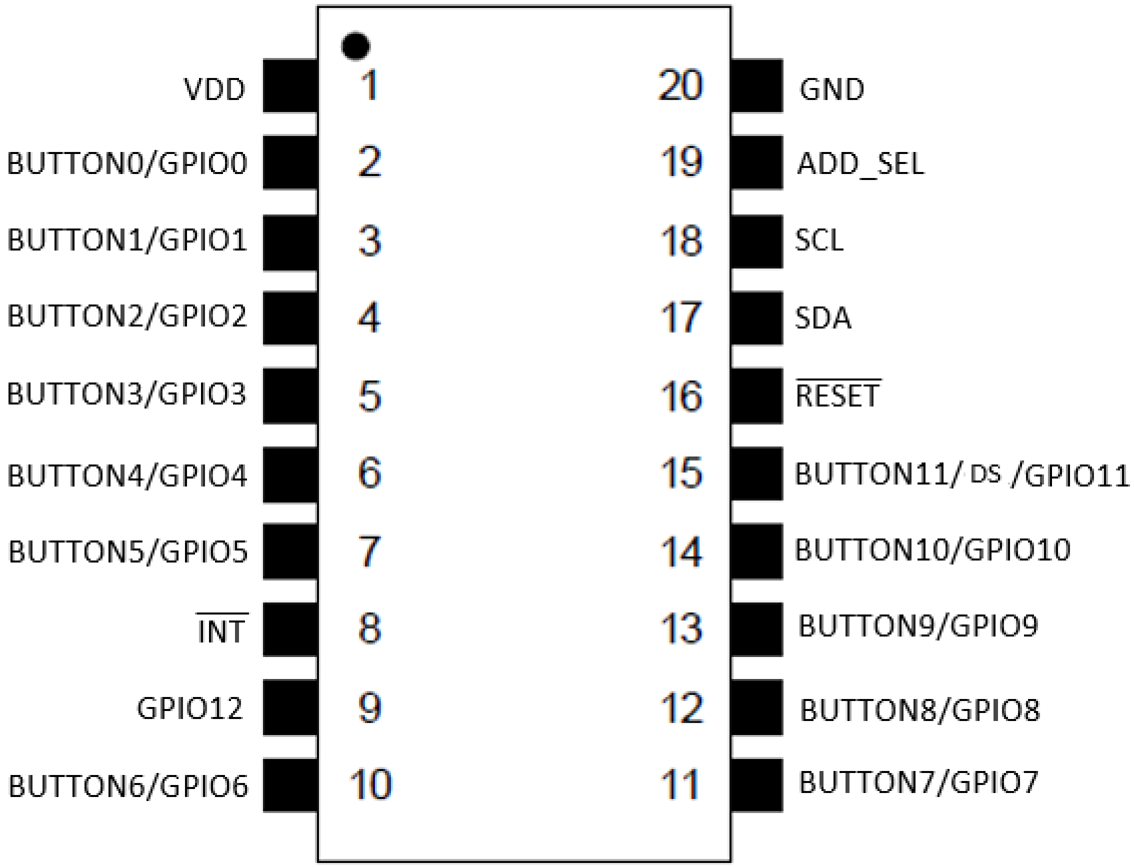
Name	20-Pin VQFN	20-Pin SOIC	Type	Comments	If Unused
SCL	1	18	I	I^{2}C Clock	NA
ADD_SEL	2	19	I	I^{2}C address select pin	Connect to GND
GND	3	20	P	Supply Ground	NA
V_{DD}	4	1	P	Power	NA
BUTTON0/GPIO0	5	2	I/O	Sensor Electrode for Button0 (or) GPIO0	Do not connect the pin. Configure the pin as a GPIO output with the state set to low.
BUTTON1/GPIO1	6	3	I/O	Sensor Electrode for Button1 (or) GPIO1	
BUTTON2/GPIO2	7	4	I/O	Sensor Electrode for Button2 (or) GPIO2	
BUTTON3/GPIO3	8	5	I/O	Sensor Electrode for Button3 (or) GPIO3	
BUTTON4/GPIO4	9	6	I/O	Sensor Electrode for Button4 (or) GPIO4	
BUTTON5/GPIO5	10	7	I/O	Sensor Electrode for Button5 (or) GPIO5	
$\overline{\text{INT}}$	11	8	O	Interrupt Pin	NA
GPIO12	12	9	I/O	Dedicated GPIO	Connect to V_{DD} or GND
BUTTON6/GPIO6	13	10	I/O	Sensor Electrode for Button6 (or) GPIO6	Do not connect the pin. Configure the pin as a GPIO output with the state set to low.
BUTTON7/GPIO7	14	11	I/O	Sensor Electrode for Button7 (or) GPIO7	
BUTTON8/GPIO8	15	12	I/O	Sensor Electrode for Button8 (or) GPIO8	
BUTTON9/GPIO9	16	13	I/O	Sensor Electrode for Button9 (or) GPIO9	
BUTTON10/GPIO10	17	14	I/O	Sensor Electrode for Button10 (or) GPIO10	
BUTTON11/DS/GPIO11	18	15	I/O	Sensor Electrode for Button11 (or) Driven Shield (or) GPIO11	
$\overline{\text{RESET}}$	19	16	I	Hardware Reset Pin	Connect to pull up
SDA	20	17	I/O	I^{2}C Data	NA

I – Input; O – Output; P – Power.

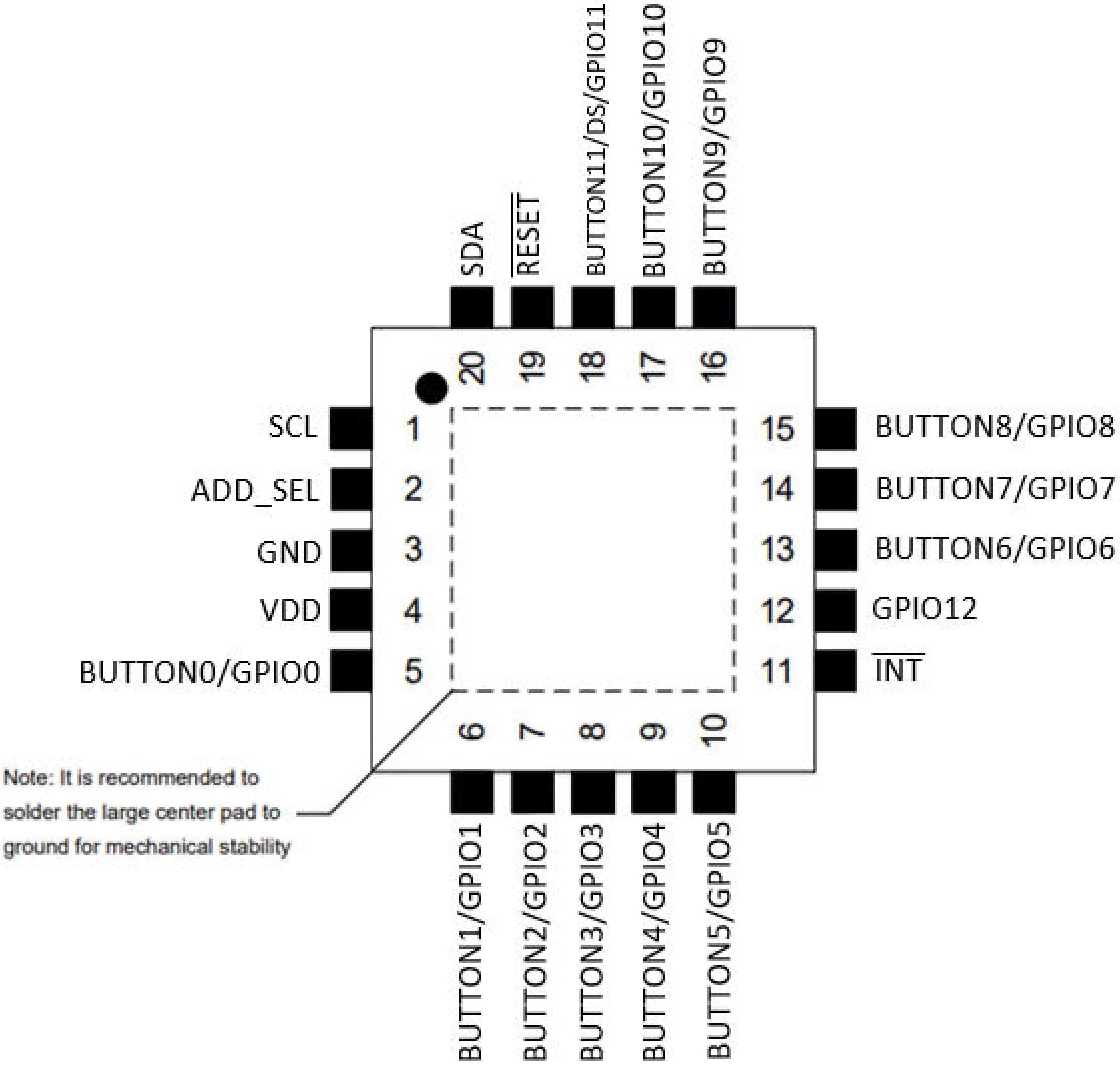
I/O – Input (or) Output

1.2 Pin Diagram

1.2.1 20-Pin SOIC

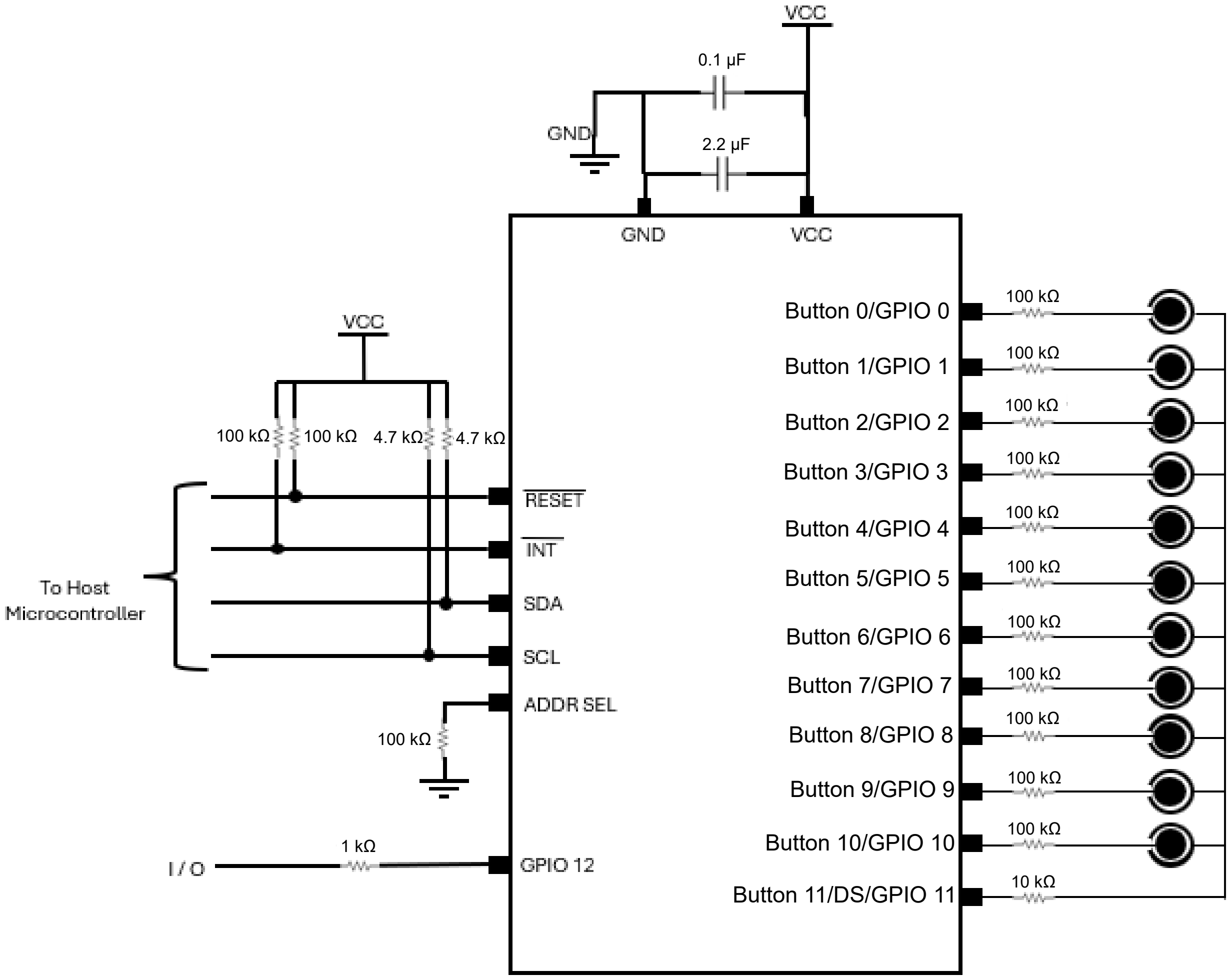


1.2.2 20-Pin VQFN

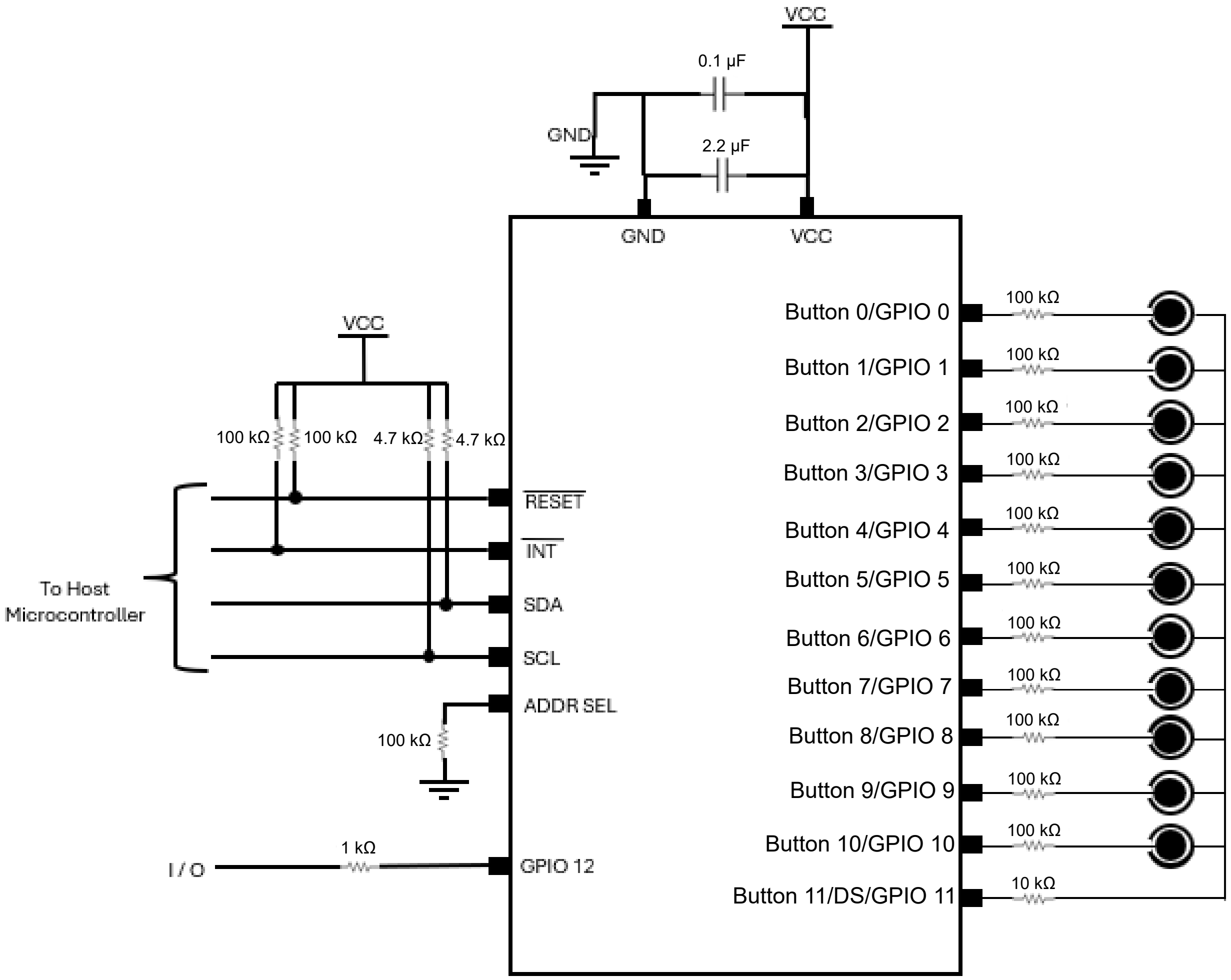


2. Example Circuit

11 Buttons with Driven Shield



12 Buttons with Driven Shield+



Note: Refer to section "Device Address" to configure the device address.

3. MTCH2120 Features

This section describes functionalities provided by the MTCH2120, detailing the touch operation and the configuration of features.

3.1 Touch Measurement Operation

This section explains the fundamental operation of touch acquisition.

Principle of Touch Measurement

The touch sensor electrode forms a parasitic capacitance (Cx) between the sensor pad and the ground. Cx increases when the sensor electrode is touched.

The device performs touch measurements by using the charge transfer principle. Touch measurement is performed by charging the internal compensation capacitor (CC) and sharing the charge accumulated in CC with Cx. A 10-bit Analog-to-Digital Converter (ADC) monitors the voltage at the mid-point between CC and Cx. Any change in Cx changes the mid-point voltage (measured by ADC). This measured data is called “Signal”.

The touch measurement relies on charge transfer. CC or Cx must be fully charged before charge sharing starts to avoid inconsistent results and to increase sensitivity and noise suppression. The charge time depends on the RC time constant of Cx and the series resistor. Use the Charge Share Delay, and Measurement Clock frequency parameters to increase charge time as required. The respective sections provide more information.

The charging and discharging involve toggling the touch pins. The series of pulses is called bursting pulses. The frequency of these pulses is the bursting frequency. The configuration of Measurement Clock frequency, Charge Share Delay and Frequency Hop will impact the burst frequency.

Calibration

The sensor capacitance without touch can vary based on environmental conditions like temperature or humidity. The system must calibrate at power-up.

The calibration process involves two steps.

1. The CC is adjusted to match the Cx. When CC is identical to Cx, the mid-point voltage will be VCC/2. Thus, the resulting signal value is usually a multiple of 512, depending on digital gain settings.
2. If the CC and Cx aren’t perfectly equal, the resulting signal value will not be 512. Thus, the system cannot always rely on 512 values to detect touch. The signal value measured at power-up (after matching CC and Cx) is used for subsequent measurement. This value is called “Reference”.

Touch Detection

After calibration, touch measurement is performed periodically, defined via the Measurement Period parameter. After each measurement, the Delta (Delta = Signal – Reference) value is computed. If the Delta is higher than the threshold, Detect integration is performed to confirm a touch. Once the sensor goes in to detect and if the Delta drops below the threshold (actually, threshold – hysteresis), the detect integration performs to confirm the touch removal. Refer to sections Threshold and Hysteresis for more information.

Drift

Touch buttons are usually not touched over an extended period, while the touch event happens quickly. For example, a light switch is touched a few times a day for a very short time while being able to operate around the clock.

During round-the-clock system operation, changes in environmental conditions, such as Temperature and/or Humidity, are likely to happen. These changes impact the touch sensor electrode's capacitance.

Complex applications may have buttons that are physically closer to each other. Due to proximity, activation (touch) of one button might impact on the signal of adjacent buttons. In such scenarios, to avoid unintended environmental drift, drift is stopped on all the buttons when any of the buttons are activated.

Refer to the Drift Operation section for more information.

Touch Measurement Flow

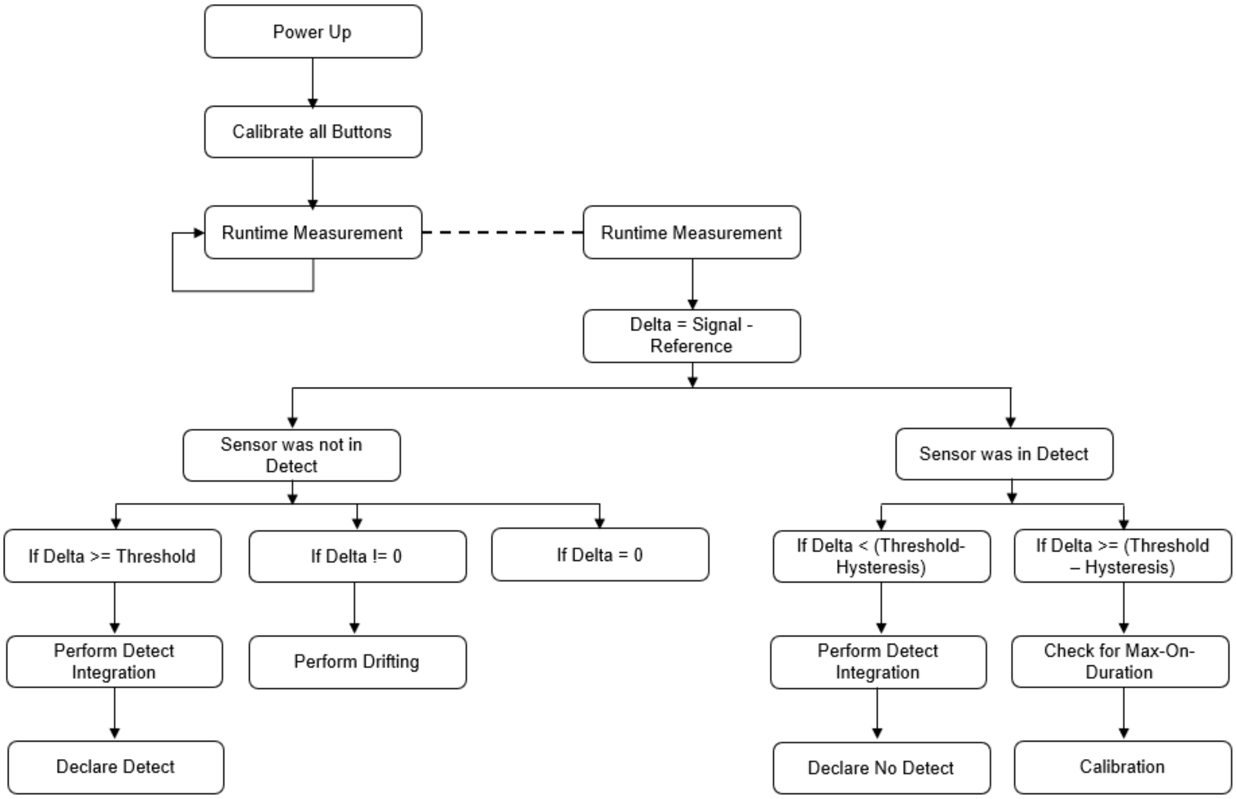


Figure 3-1. Simplified Measurement Flow

3.2 Touch Sensor - BUTTONx

BUTTONx is the touch sensor input pins, which must be connected to the sensor electrodes via a series resistor to reduce the EMI (Electromagnetic interference) and EMC (Electromagnetic compatibility). The series resistor can be 1-100 kΩ,, depending on the sensor capacitance and desired level of EMC performance. For further details, refer to Microchip Application Note (AN2934) Capacitive Touch Sensor Design.

3.2.1 Operation

MTCH2120 supports up to 12 buttons (Button 0 to 11) and up to four lumped buttons (Button 12 to 15) (Refer to section Lump.) Buttons are measured sequentially, from the lower to the higher index (BUTTON0 through BUTTON15).

Enable, Disable and Suspend

MTCH2120 provides an option to enable, disable, or suspend a button. By default, buttons 0 to 11 are enabled, and buttons 12 to 15 are disabled.

The Disable option can be used to disable buttons for designs that require only a sub-set number of buttons, thus improving the response time and power consumption.

The Suspend option helps temporarily remove the button from scanning. Suspend can be set or reset during run time, thus providing complete control to the application which needs dynamic button configuration.

If a button is disabled/suspended, for example - BUTTON6, then after scanning BUTTON5, BUTTON7 will be scanned.

Note: If, by mistake, all the buttons are suspended, then a watchdog reset will be triggered. Otherwise, if a mistake occurs when the watchdog is disabled, provide a hardware reset to recover the device.

Driven Shield

BUTTON11 can be configured as a Button or dedicated Driven Shield (Refer to SHIELD – Driven Shield). In case BUTTON11 is used as a dedicated driven shield line, the maximum number of touch buttons becomes 11.

3.2.2 Initialization

In the default configuration, buttons 0 to 11 are enabled. A device reset is required to reflect register modifications. Refer to Configuration Summary for more details.

Enable/Disable

Buttons can be enabled (or disabled) by writing a `1` (or `0`) to the Enable (EN) bit in the Sensor Control (SENCTRL) register. This configuration update will not reflect immediately. It will reflect after the device reset (it could be either a software or hardware reset).

Steps to Enable/Disable Buttons

1. Write `1` or `0` bit in SENCTRL[n].EN register to enable or disable the button.
2. Write `1` in DEVCTRL.SAVE register to save the configuration.
3. Write `1` in DEVCTRL.RESET to reset the device.

Unused Button Pins

To minimize power consumption:

- Unused buttons should be disabled. Refer to **Steps to Enable/Disable buttons**
- The pin should be left unconnected

Configuring the unused button's pin as GPIO output-LOW improves the noise immunity. Refer to GPIO Host Communication sections to configuring the pin as output-LOW.

Suspend

Buttons can be suspended by writing a `1` to the Suspend (SUSPEND) bit in the Sensor Control (SENCTRL) register.
Note: If the SUSPEND configuration is stored in memory, this bit is ignored during power up. All the enabled sensors will be measured immediately after power-up. If the Host wants to suspend a sensor, it must give a command again through the Inter-Integrated Circuit (I^2C).

Calibrate

Calibration is a process of compensating for the environmental conditions like humidity or temperature that impact sensor capacitance. During power-up, MTCH2120 performs calibration. After power-up calibration, further changes in environmental conditions are taken care of by the drifting feature explained in the Drift Operation section. Additionally, it is possible to calibrate a particular sensor or all sensors through an I^2C interface. For example, it is recommended to recalibrate a sensor if it is suspended for some period in which it may have experienced environmental changes.

Calibration for a button can be triggered by `1` to the Calibrate (CAL) bit in the Sensor Control (SENCTRL) register. And the calibration will happen in the next measurement cycle.

3.3 Sensor Control Parameters

3.3.1 Oversampling (Filter Level)

Oversampling defines the number of samples taken for each measurement. Higher oversample settings result in more samples for each measurement, which averages the noise. Measurement will take a longer time, which will impact response time. Start with the default value and increase or decrease depending on noise levels. The default value is 16.

Note: The minimum recommended ratio of Filter Level to digital Gain is eight. For example, if digital Gain is configured as two, then oversampling should be configured to at least 16.

Steps to Configure Oversampling

The oversampling value for Button N can be updated by writing in the Node Oversampling (FILTERLEVEL[n]) register. Refer to Configuration Summary for more details.

3.3.2 Gain

The Gain defines the digital Gain applied to the measured data. The accumulated result from an oversampled measurement is scaled down based on the gain. Standard setting and usage target is Gain = 1. The system must be tuned so the touch delta is between 40-60 counts. The Gain, in addition to the Threshold setting, can be used to adjust the sensitivity of each button.
Note: The minimum recommended ratio of Filter Level to digital Gain is eight. For example, if digital Gain is configured as two, then oversampling must be configured to 16.

Steps to Configure GAIN

Writing in the Node Gain (GAIN[n]) register will update the Gain value for Button N.. Refer to Configuration Summary for more details.

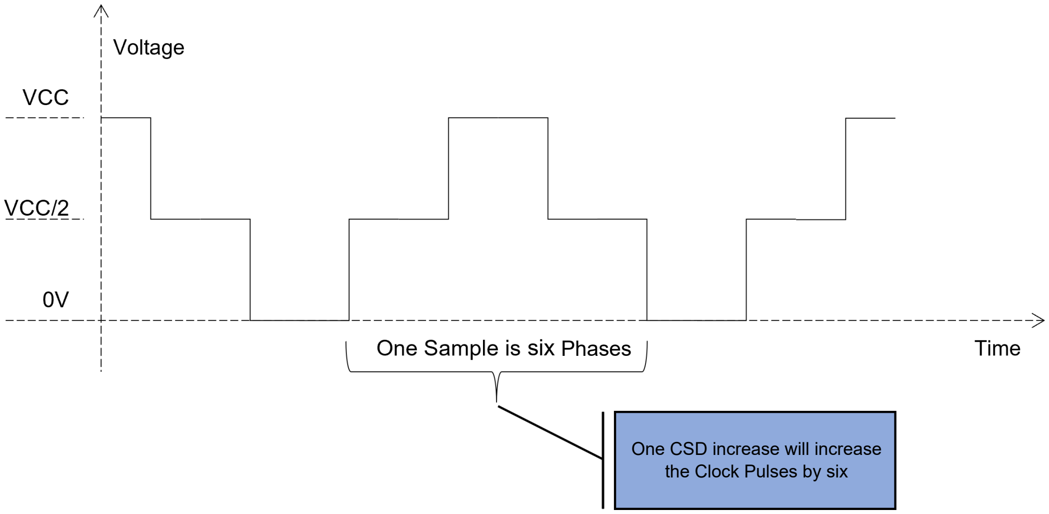
3.3.3 CSD - Charge Share Delay

The sensors must be fully charged and discharged for each measurement to get consistent results. The RC time constant may vary based on the sensor capacitance and series resistor option. CSD must be selected optimally ensure that complete charging and discharging happen.

Operation

As seen in the section Touch Measurement Operation, touch measurement is performed by sharing charge between CC and Cx. The pulse for each sample is divided into 6-phases. Increasing CSD by one count increases the measurement time of one sample by six clock pulses. Thus, increasing the CSD will increase the charging time. Refer to the section Steps To Configure Measurement Clock Frequency and CSD for more information on tuning CSD. Using higher CSD will have an impact on Response Time.

The CSD can be configured per channel by writing the value in the CSD [n] register. Refer to the Configuration Summary sections for more information.



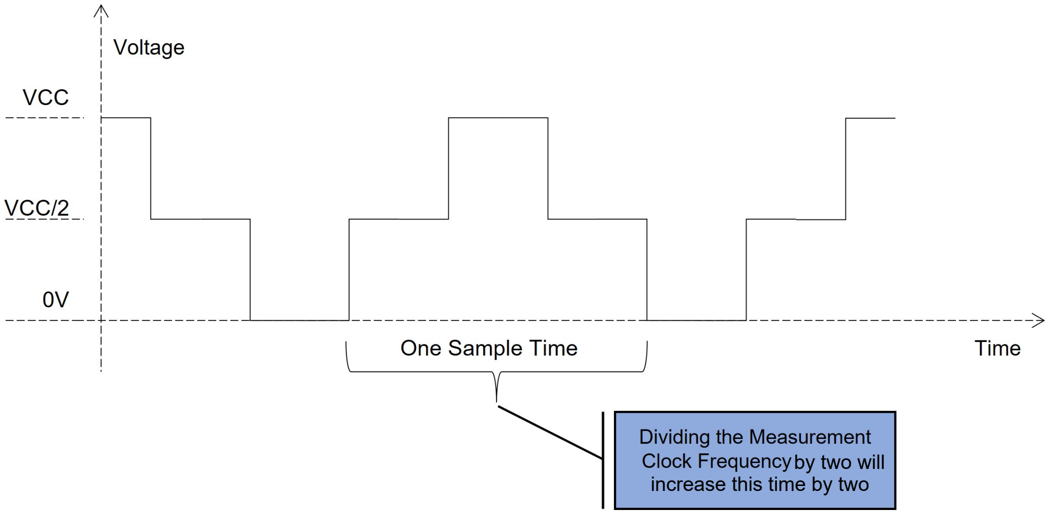
Steps to Configure CSD

1. Write the value in CSD[n] register to configure the CSD.
2. Save (Optional)
   - This save option will help to retain the values after the device is reset (or) rebooted. Otherwise, this step is optional.

3.3.4 Measurement Clock Frequency

The device performs touch measurement using the clock defined by Measurement Clock Frequency. Reducing this clock increases the measurement time. You may use this parameter to ensure that the sensor electrodes are fully charged when using CSD is exhausted. It is possible to use 1 MHz or 500 kHz as touch measurement frequency. Refer to the section Steps To Configure Measurement Clock Frequency and CSD for more information on tuning MCLKFREQ.

The measurement clock frequency can be configured per channel by writing the value in the MCLKFREQ[n] register. Refer to the Configuration Summary sections for more information.


Voltage
VCC
VCC/2
0V
One Sample Time
Time
Dividing the Measurement Clock Frequency by two will increase this time by two

Steps to Configure Measurement Clock Frequency

1. Write the value in the MCLKFREQ[n] register to configure Measurement Clock Frequency.
2. Save (Optional)
   - This save option will help to retain the values after the device is reset (or) rebooted. Otherwise, this step is optional.

3.3.5 Steps To Configure Measurement Clock Frequency and CSD

Following are the steps for tuning CSD:

1. Start with a higher CSD value, e.g. 30, and observe the touch delta.
2. Observe the Delta with CSD-values 25, 20, 15, 10, 5 and zero.
3. From these values, find the CSD for which the delta is not significantly less compared to the delta with a CSD of 30.
4. Use that CSD value.

In this example, the highest delta value of 40 is found with a CSD value of 15. Increasing CSD beyond 15 does not help to increase the delta. So, it is better to use CSD = 15. It is possible to finetune CSD by comparing the delta with CSD 14, 13, 12, and 11.

Use higher values than the values resulting from the tune to take tolerance into account. In the case of CSD, use a 20% higher value. For example: If the tuned CSD value is 10, use 12.

For most applications, the maximum 1 MHz measurement clock frequency works well along with fine tuning of CSD. Only when using maximum CSD at max Touch Acquisition Frequency doesn't lead to maximum delta, reducing the measurement clock frequency can be tried.

3.3.6 Threshold

Defines the detected Threshold. When a button is touched, the delta for that button increases. The button will report as touched when the touch delta value is higher than the button’s Threshold. Configuring the Threshold to 50~70% of the touch delta is recommended. It is good to start with the default value and adjust by monitoring the touch delta value. Threshold, in addition to the Gain setting, can be used to set the sensitivity of each button.

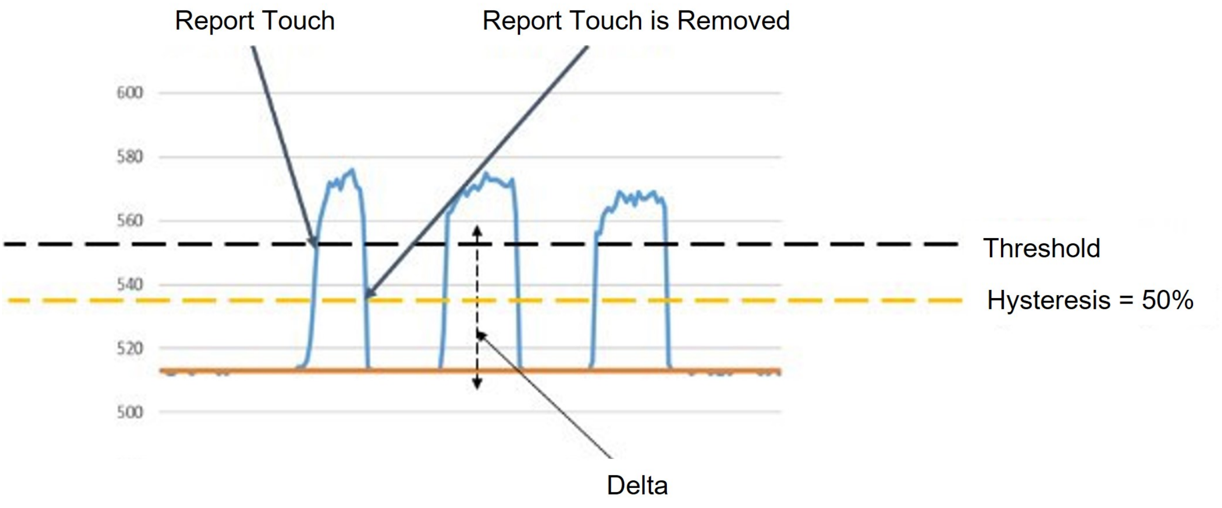
Steps to Configure Threshold

1. The Threshold value for Button[n] can be updated by writing in Channel Threshold (THRESHOLD[n]) register.
   Note: Prior to modifying the register, ensure that the Easy Tune function is disabled. Refer to the “Easy Tune” feature for more information.
2. Saving the Settings (optional):
   - Writing `1` in DEVCTRL.SAVE to save the configuration.

3.3.7 Hysteresis

Under severe noise conditions, the delta value can hover around the sensor threshold and cause the sensor to dither in and out of touch. To avoid this, once a sensor goes into the Detect state, the Threshold for the sensor is reduced by the hysteresis value, thus providing a configurable margin for the noise. Hysteresis values are derived from the Threshold is defined as a percentage of the Threshold. In other words - buttons go into touch with a higher threshold than they go out of touch.

It is not enough for the delta to go below the Threshold for the sensor to report out-of-touch. It has to go below the hysteresis level to declare no touch.



Steps to Configure Hysteresis

The hysteresis value for Button N can be updated by writing in the Channel Hysteresis (HYSTERESIS[n]) register. Refer to the Configuration Summary for more details.

3.3.8 Adjacent Key Suppression™ (AKS™)

AKS can be used when touching multiple buttons is NOT allowed or not desired in a system. When buttons are physically close to each other, there is a possibility that touching one button will cause a rise in the touch delta value on the adjacent, button which may also be caused by pressing between buttons with no or only a minor gap between them. Sometimes, the delta raised in other buttons may cross the threshold and lead to a false report of touch.

To control such behavior, buttons that are physically closer to each other can be configured in the same AKS group. Only the first button (which goes in to detect) will be reported as touched in a given AKS group. All other button states will be suppressed, even if their delta crosses the threshold value.

This device supports up to seven AKS groups.

Steps to Configure AKS

The AKS value for Button N can be updated by writing in the Channel AKS Group (AKSGRP [n]) register. Refer to the Configuration Summary for more details.

3.4 Global Control Parameters

The global parameter applies to all buttons.

3.4.1 Measurement Period

The measurement period is the combination of measurement time and sleep time.

Measurement time is the total time required to acquire (including oversampling) and process touch signals of all active buttons. Sleep time occurs between the measurements. The measurement period can be configured up to 250 milliseconds (ms).

Figure 3-2. Active Measurement Period

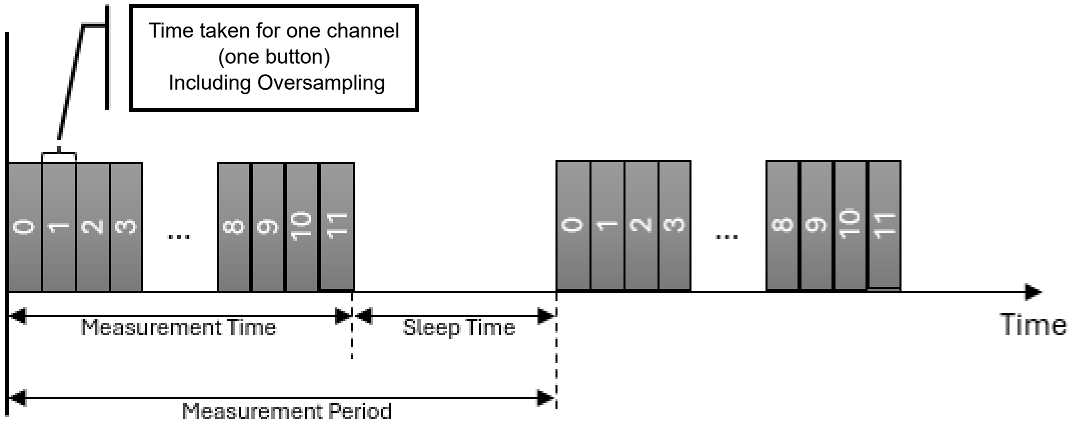
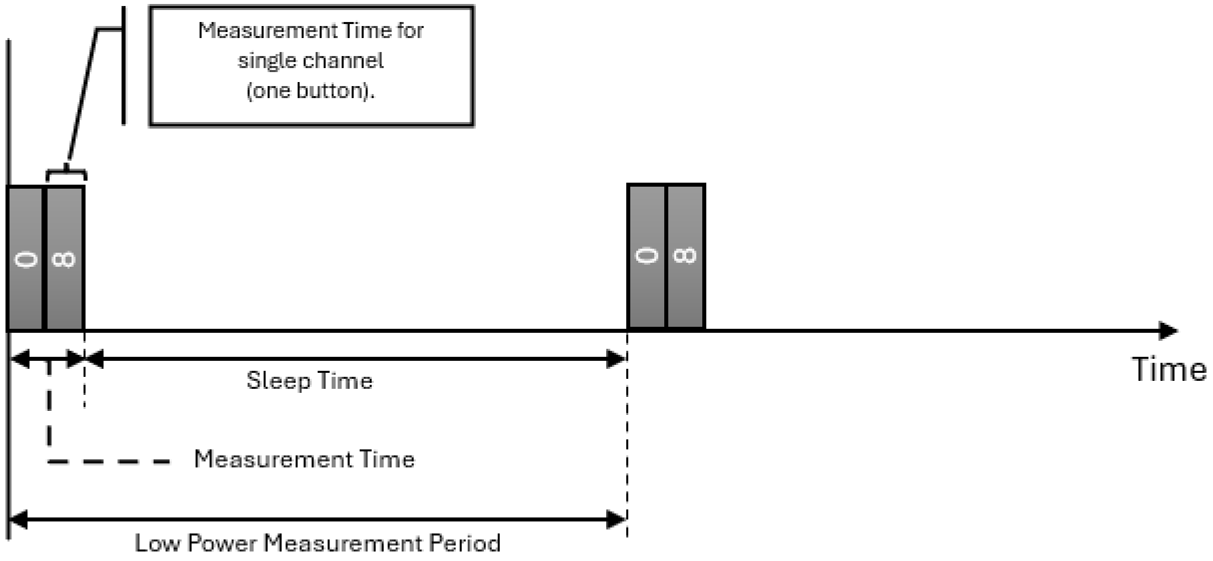


Figure 3-3. Low-Power Measurement Period



To clarify, note that buttons 0 and 8 are low-power buttons (meaning buttons configured to be scanned in low-power mode).

Steps to Configure Measurement Period

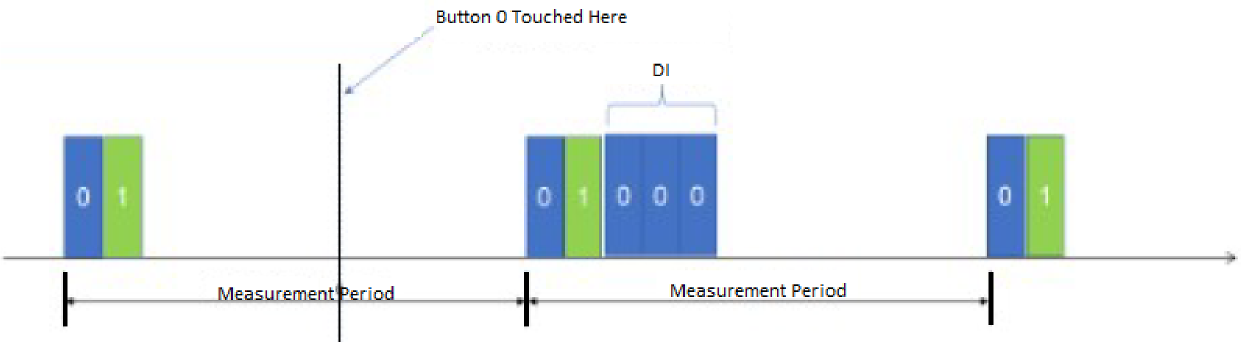
1. **The Active Measurement Period** value for Buttons can be updated by writing in the Touch Measurement Period (MP) register.
2. **Low-Power Measurement Period** value for Buttons can be updated by writing in the Low-Power Measurement Period (LPMP) register.
3. Refer to the Configuration Summary for more details.

3.4.2 Detect Integration (DI)

Detect integration is a simple yet very effective technique to tackle high noise. DI helps to confirm that the variations in touch measurements are caused by a valid finger touch made by doing confirmation scan(s) (repeated measurements). The touch is reported only when the touch measurement data is consistent. These additional confirmation scans are performed back-to-back to preserve touch interface snappiness.

Increasing the DI increases the number of confirmation scans and improves the robustness of touch detection. However, a higher DI can also lead to increased response time, making it crucial to select an optimal DI level that meets noise requirements. DI is also applied to verify when a touch is removed.

Here, only two buttons are enabled, with DI configured as three. Once a touch is detected, button 0 will do three more measurements to confirm the touch. Refer to the below image.



Steps to Configure Detect Integration (DI)

Writing the value in the DI register can update the DI value for Buttons. Refer to the Configuration Summary for more details.

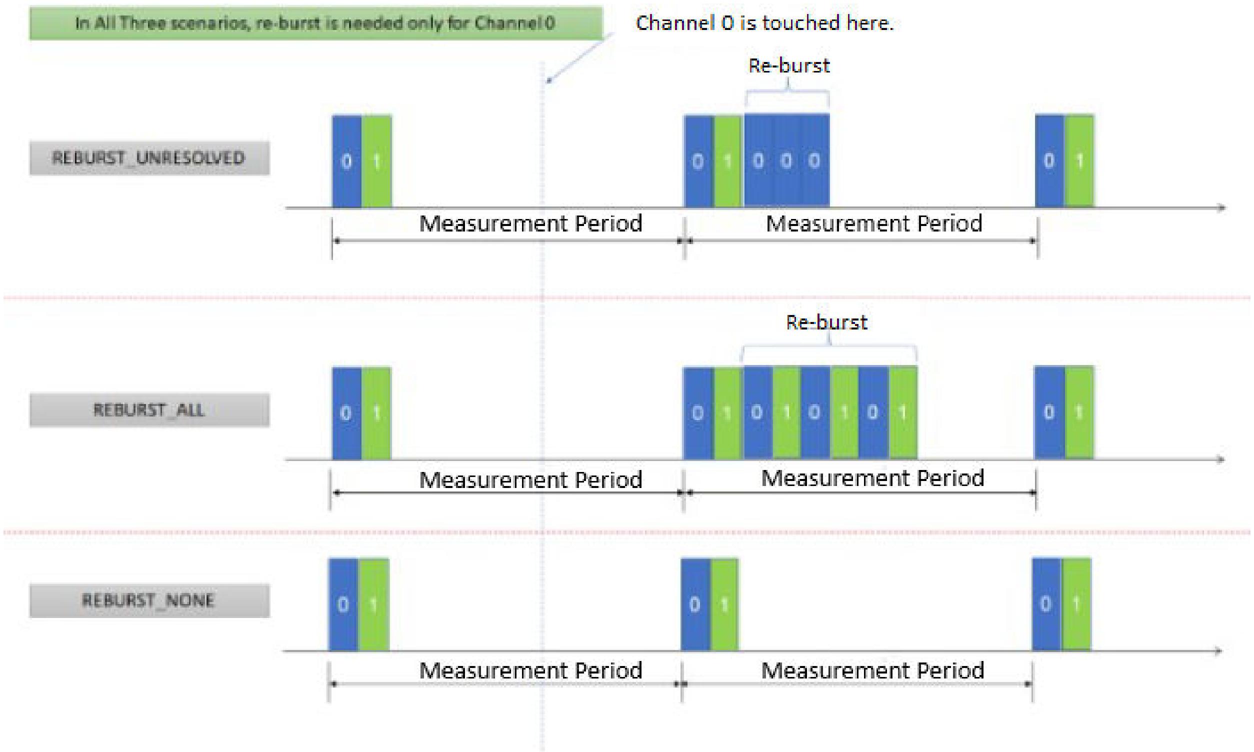
3.4.3 Reburst Modes

Under various conditions (like Calibration, Detect Integration, and Recalibration), additional touch measurements will be performed to confirm an activity. This parameter defines how the re-bursting (additional touch measurements) happens.

Table 3-1.

Mode	Descriptions
None	No further assessments have been performed
Reburst Unresolved	Additional measurements will be performed only for touch-detected buttons
Reburst All	Additional measurements will be performed for all active buttons

The following figures show how each reburst is done.



To clarify, note that only two buttons are active, while the remaining buttons are disabled.

Steps to Configure Reburst Modes

Writing the value in the DI register can update the DI value for Buttons. writing in Sensor Reburst Mode (REBURSTMODE) register. Refer to the Configuration Summary for more details.

3.4.4 Drift Operation

Drifting aligns the Reference with the Signal value by adjusting the Reference value incrementally at the configured interval.

3.4.4.1 Touch Drift Rate

When the signal value is greater than the reference, the Touch Drift Rate defines the rate at which the reference value is adjusted towards the change in signal value, and the Touch Drift value is scales by 200 ms. For example, if the Touch Drift Rate is 10, the drift period is 5 x 200 ms = 1s.Every second, the reference value will increase by one until it matches the signal value. When Drift Gain is enabled, the reference value is incremented by Gain for every second.

Steps to Configure Touch Drift Rate

The Touch Drift value for Buttons can be updated by writing in the Sensor Touch Drift Rate (DCHDR) register. Refer to the Configuration Summary for more details.

3.4.4.2 Anti-Touch Drift Rate

This parameter works similarly to Touch Drift but in the opposite direction.

When the signal value is less than the reference, Anti-Touch Drift Rate defines the rate at which the reference value is adjusted towards the change in signal value, and the Anti-Touch Drift Rate value scales by 200 ms.

If the Anti-Touch Drift Rate is 10, the drift period is 10 x 200 ms = 2s. Every two seconds, the reference value will be decremented by 1 until it equals the signal value. When Drift Gain is enabled, the reference value is decremented by Gain for every two seconds. For example, if Gain is 2 for a particular channel, that channel's Reference will be adjusted by two counts every 2s.

Steps to Configure Anti-Touch Drift Rate

The Anti-Touch Drift value for Buttons can be updated by writing in Sensor Anti-Touch Drift Rate (ANTITCHDR) register. Refer to theConfiguration Summary for more details.

3.4.4.3 Drift Hold Time

When a button is in the detect state, the drifting on all other buttons is stopped. When removing the finger (when the button goes out of the detect state), drifting is restarted. It is essential to pause the drifting restart for a couple of seconds, as there is a possibility that the finger is still around the buttons, and an immediate resume of drifting might cause false drifting.

The Drift Hold Time defines the time the drifting pauses after removing the finger, by doing so, this value is scaled by 200 ms. E.g., if the Drift Hold time is 20, then for 20 x 200 ms = 4s, the drifting is not resumed after removing a finger. For example, if Gain is 2 for a particular channel, then that channel's Reference will be adjusted by two counts every 4s.

Steps to Configure Drift Hold Time

The Drift Hold Time value for Buttons can be updated by writing in Sensor Drift Hold Time (DHT) register. Refer to the Configuration Summary for more details.

3.4.4.4 Drift Gain

The Drift Gain feature is relevant for Touch Drift and Anti-Touch Drift modes. Drift Gain defines the speed at which the device follows environmental changes. By default, the Drift Gain is enabled, allowing the ability to track the environmental conditions per the Gain settings. For example, if Gain is 2 for a particular channel, then that channel's Reference will be adjusted by two counts.

Steps to set Drift Gain

Drift Gain can be set by writing a `1` to the Drift Gain (DRIFTGAIN) bit in the Device Control (DEVCTRL) register.

3.4.4.5 Drift Low-Power Lump Buttons

This feature is applicable only when LUMP and Low-Power feature are enabled.

Drift settings apply to all buttons, including virtual buttons like a Lump. In this example, we save power, and two buttons are configured as a Lump and Lump as a low-power button. Since only the Lump is configured as a low-power button, measurement and drifting will be done only for the Lump buttons. The individual sensors that are part of the Lump will neither be measured nor drifted. When one of the two buttons (= the LUMP) is touched, the MTCH2120 will report via I^2C the lump as touched. Lets further assume in this example that the low-power Lump was used to (re)activate a device - now all buttons individually need to be scanned.

The Drift Low-Power Lump Buttons feature allows drifting of the individual buttons which are configured as a part of the low-power Lump buttons. Thus, no calibration is required for the real buttons when the device exits low-power mode since the signal and reference values are balanced for the actual buttons.

This feature can be enabled or disabled by writing bit in DEVCTRL.DLPLB. It is recommended to enable this option when lump is configured as low-power button.

Note: A drift assessment occurs every ten measurements.

Table 3-2. Example

Button	Low-Power Sensor (SENCTRL[n].LP)	Enable Low-Power (DEVCTRL.LP)	DRIFT (Buttons)	
			DEVCTRL.DLPLB = 0	DEVCTRL.DLPLB = 1
Button 0	1	1	0	0
Lump 1	1	1	12	12, 5, 6
Lump 0, Lump 1	1	1	12, 13	5, 6, 7, 8, 12, 13

...........continued

Button	Low-Power Sensor (SENCTRL[n].LP)	Enable Low-Power (DEVCTRL.LP)	DRIFT (Buttons)	
			DEVCTRL.DLPLB = 0	DEVCTRL.DLPLB = 1
Button 0, Lump 0	1	1	0, 12	0, 5, 6, 12

Notes:

1. Buttons 5 and 6 are part of lump 0.
2. Buttons 7 and 8 are part of lump 1.
3. Consider Lump 0 as button 12 and Lump 1 as button 13.

Steps to Enable Drift Low-Power Lump Buttons

1. Configure Lump. Refer to the Steps to Configure Lump.
2. Configure Lump as a low-power button and enable the low-power feature. Refer to the Steps to Configure Low-Power.
3. Saving the Settings (optional). Refer to the Save section for more information.

3.4.5 Anti-Touch Recalibration

The continuous negative delta will be observed when the system is calibrated with a finger/object placed on the button and such finger/object is removed. This scenario is monitored on the MTCH2120 board (no host intervention required), and recalibration is performed to resume the ordinary operation. Usually, the Anti-Touch recalibration will happen when the delta is less than the Anti-Touch recalibration threshold (ANTITCHRECALTHRESH) value for the Anti-Touch-Integration (ANTITCHINT) times.

Negative delta due to noise (an occasional spike) doesn't trigger recalibration.

Steps to Configure Anti-Touch Recalibration

The Anti-Touch Recalibration value for Buttons can be updated by writing in the Sensor Anti-Touch Recalibration Threshold (ANTITCHRECALTHRESH) register, and the Anti-touch Integration value can be updated by writing in the ANTITCHINT register. Refer to the Configuration Summary for more details.

3.4.6 Max-On Duration (MOD)

Max-On-Duration (MOD) defines the maximum button ON-time before a re-calibration performs. This setting has a fallback if a button gets "stuck in touch." You may configure MOD in a system where a prolonged touch is invalid. If a button is detected for more than MOD, the button will be calibrated (resulting in a revert to a no-detect state). If MOD is zero, it is seen as infinite, allowing buttons to be pushed and reported whenever activated, and the selected MOD value is scaled by 200 ms.

For example, if the MOD is configured as ten, then 10 * 200 = 2s. If a button is in detect for more than 2s, the button will go back to no-detect state.

Steps to Configure MOD

The MOD value for Buttons can be updated by writing in the Sensor Max-On-Time (MAXONTIME) register. Refer to the Configuration Summary for more details.

3.5 SHIELD – Driven Shield

The MTCH2120 features a Driven shield and Driven shield+ options to enhance touch sensitivity and enable a water-tolerant touch. A driven shield signal can be provided on a dedicated pin and/or on all not-scanned buttons (called Driven Shield+). Driven Shield+ functionality is very effective in multi-button designs, shielding trace lines, etc. The shielding signal is provided on pin BUTTON11 when the dedicated shield is enabled.

Driven Shield benefits:

- Reducing capacitive sensor load (e.g., enabling to the reduction of CSD, scan time, and power consumption in consequence)
- Establishing water tolerance. Recommend using Driven Shield+ so that the buttons not currently measured are driven as a shield, helping to avoid false detection due to the water bridge between the multiple buttons.
- If used as an alternative to a passive ground shield, unlike a ground shield, a driven shield will neither reduce sensitivity nor increase sensor capacitance

Refer to the Microchip Capacitive Touch Design Guide – AN2934 for further details and layout considerations for Driven Shield and Driven Shield Plus usage.

Steps to Configure Driven Shield

- Disable Button 11 by writing `0` to the Sensor Enable (EN) bit in the Sensor Control (SENCTRL) register

  Note: The Driven Shield is integrated with the Button 11 pin. Therefore Button 11 must be disabled.
- Enable the Driven Shield by writing `1` to the Driven Shield (DS) bit in the Device Control (DEVCTRL) register

Steps to Configure Driven Shield Plus

- The Enable Driven Shield Plus by writing `1` to the Driven Shield Plus (DSP) bit in the Device Control (DEVCTRL) register

Note:
Enabling or disabling the Shield feature by the Host requires a restart after the configuration update. Refer to theConfiguration Summary for more details.

3.6 Lump

The Lump feature allows multiple buttons to be combined (or Lumped) to form a single button. The Lump feature is useful to:

- Reduce the number of sensors by combining multiple buttons, which helps in power consumption
- Form a large touch surface using small buttons, thus increasing the sensitivity, which helps to realize proximity sensing. Proximity sensing requires a large electrode area to detect an approaching hand. Lumping the keypad's corner keys can help to detect approaching hand and wake the system up. In the keypad example, the low-power Lump (created out of existing keys of the keypad) is the only entity scanned in sleep, waking the system up by user proximity, exemplary back-lighting the keypad. After waking, all keypad buttons are scanned.

3.6.1 Operation

Touch detection on a Lump sensor is similar to an ordinary touch button. MTCH2120 provides greater flexibility to users when forming Lump sensors as follows.

- Up to four Lump sensors can be formed
- The Lump sensor can be formed by combining any number of buttons. Refer to note.
- The Same button can be part of more than one Lump button.

Note: The capacitance of the lump button is typically equal to the sum of the individual button’s capacitance. Lumping too many buttons may result in higher sensor capacitance, which may not be handled optimally by MTCH2120 - the usage of driven shield+ is advised and very beneficial.

The Lump sensor can be set up as a low-power sensor in scenarios such as a system where two buttons must be scanned in low-power mode. Instead of configuring two individual buttons as low-power sensors, the buttons can be lumped, and the lumped sensor can be configured as a low-power sensor. It is also possible to configure all 12 buttons in four different lump groups and configure the four Lumps as a low-power sensors.

3.6.2 Steps to Configure Lump

1. Enable the Lump button by writing `1` in SENCTRL[n].EN register.
2. Update the bit mask for the appropriate button in the Lump Configuration (LUMPCONF) register to assign button n to the corresponding Lump group.

Note:
The Lump configuration will be configured upon resetting the device. Refer to the Configuration Summary for more details.

3.7 Low-Power Mode

Low-power mode helps to control the power consumption of MTCH2120. The host presets the reduced button number in low-power mode and/or a different touch measurement interval. In low-power mode, the touch measurements occur at the interval defined in Low-Power Measurement Period (LPMP) register.

By optimally configuring the buttons as Lump sensors and configuring the Lump buttons as low-power buttons, the power consumption can be reduced effectively and controlled. For more details on Lump sensors, refer to Lump.

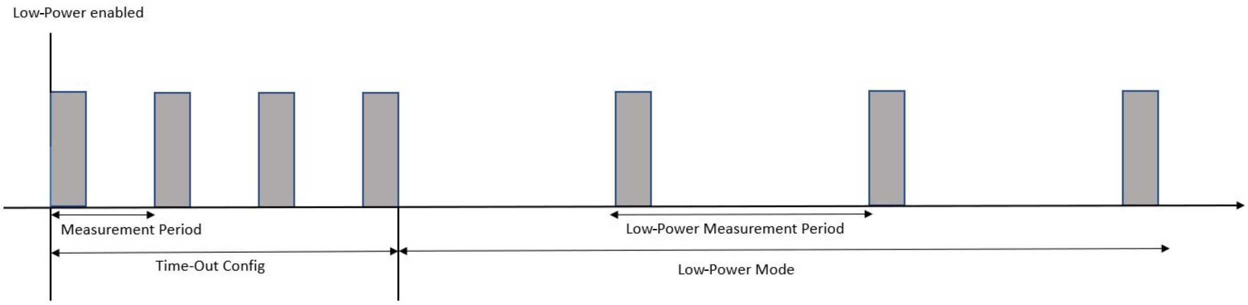
3.7.1 Operation

Low-Power Feature Disabled

When the low-power feature is disabled, MTCH2120 stays in Active mode, where all the enabled buttons are scanned at intervals defined by Touch Measurement Period MP.

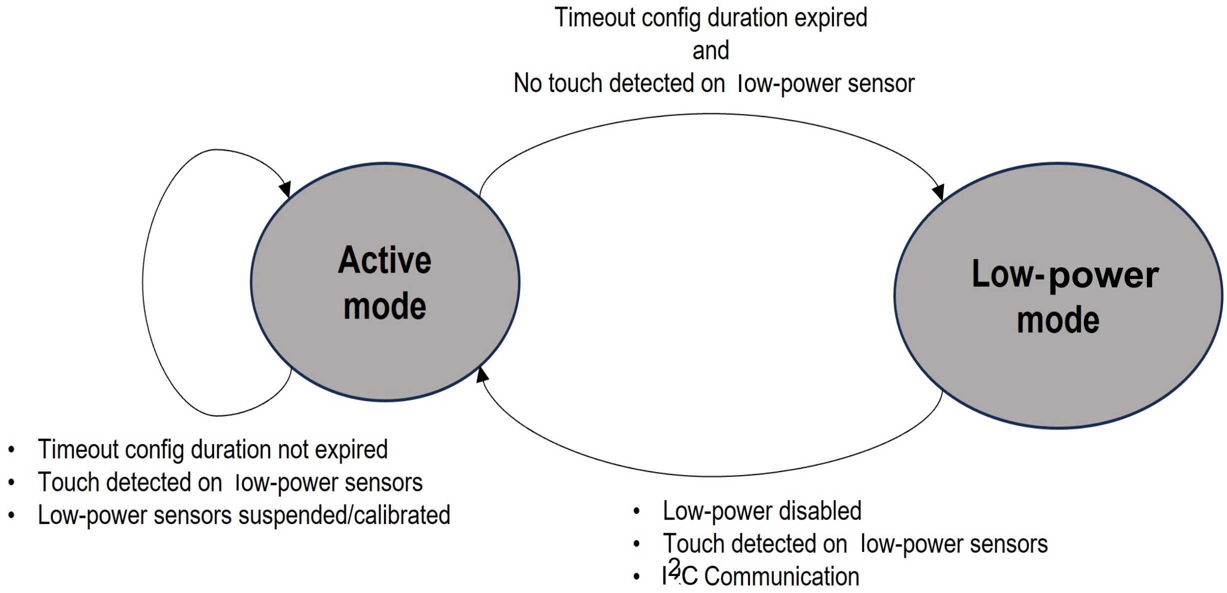
Low-Power Feature Enabled

When the low-power feature is enabled, and no buttons are touched for the Time-Out Configuration (Refer to the Time-Out Config (TIMEOUTCONF) register) period, MTCH2120 will enter the low-power mode. In low-power mode, buttons with the Low-Power Sensor [LP] bit set to `0` in the Sensor Control (SENCTRL) register are suspended from scanning. In low-power mode, only the buttons with the Low-Power Sensor [LP] bit set to `1` in the Sensor Control (SENCTRL) register are enabled for scanning. These low-power buttons are scanned at the measurement period defined by low-power measurement period (Refer to the Low-Power Measurement Period (LPMP) register). The suspended buttons are resumed and calibrated when a touch is detected on low-power buttons. The system returns to active mode.



Note: The Low-Power mode must be disabled by the Host within the period configured in the Time-Out Config (TIMEOUTCONF) register so MTCH2120 stays in the active mode. If the Host doesn't disable the Low Power mode, the device will reenter low-power mode automatically after a Time-Out period of no touch is detected.

While in Low-Power mode, the Host may switch to regular operation (active mode) at any time with immediate effect. The Low-Power mode can be used as a power button to unlock other keys or as a proximity trigger to detect an approaching finger towards the panel to activate the touch-sensor panel. The figure shows the operation sequence taking place between active and Low-Power mode when Low-Power mode is enabled.


Timeout config duration expired
and
No touch detected on low-power sensor
Active mode
Low-power mode
• Timeout config duration not expired
• Touch detected on low-power sensors
• Low-power sensors suspended/calibrated
• Low-power disabled
• Touch detected on low-power sensors
• I2C Communication

Power consumption depends on the number of low-power buttons enabled and the low-power measurement period. The exemplary use case is the power buttons on a device: Only the power buttons will turn on the device. After the device is turned on, all the buttons must be active and scanned.

Note: Then the host sends any I^{2}C command to the device, it will exit the low-power mode.

Note: Auto Threshold and Frequency Hopping with Auto-Tune features are disabled when the device enters the low-power mode.

3.7.2 Steps to Configure Low Power

1. Configure the Button n as a low-power button by writing `1` to the Low-Power Sensor (LP) bit in the Sensor Control (SENCTRL) register.

   Note: If one or more of the buttons are not set to be low-power buttons, the device will not go into low-power mode.
2. Enable Low-Power mode by writing `1` to the Low Power (LP) bit in the Device Control (DEVCTRL) register.
   Note: Calibration and Suspension of Button n should not be enabled (Refer to the Sensor Control (SENCTRL) register).

Refer to the Configuration Summary for more details.

3.8 Easy Tune

Easy Tune is enabled by default. The Easy Tune option saves development time by automatically adjusting the threshold based on the observed noise level, ensuring operations under changing noise environments while maximizing sensitivity if no or low noise is present. The Easy Tune feature controls threshold and oversampling(filter level) settings. Easy Tune automatically applies settings to satisfy the fundamental touch tuning recommendation from the Tuning Guide for Touch Sensors. With increasing sensitivity, increasing the oversampling is recommended to maintain noise robustness. If the user didn't adjust the oversampling accordingly, Easy Tune automatically adjusts to the recommended settings for robust operation. The minimum amount of recommended oversampling is applied with Easy Tune activated, or the user setting is used if selected higher than the minimum recommendation.

The threshold value will be reset to the previously saved value when this feature is disabled during runtime.

Example: The threshold for all buttons was set to 30 while the Easy Tune was disabled. This configuration has been stored in the device's memory. Upon enabling the Easy Tune feature, the threshold value will be dynamically adjusted according to the ambient noise level. If the feature is subsequently disabled, the threshold will reset to the previously stored value of 30 for all buttons.

3.8.1 Operation

When Easy Tune is enabled, the thresholds are adjusted automatically based on the noise level on the touch sensors. The threshold is adjusted based on the current noise floor ranging ±50% of the threshold configured using the Channel Threshold (THRESHOLD) register.

For example, if the Channel Threshold (THRESHOLD) register sets the threshold to 50, the touch delta on the given sensor will be nearly 100 counts. The Easy Tune feature would vary the threshold between 25 and 75 counts for this configuration. Under no noise, the threshold value will be set to 25. As noise levels increase, this feature increases the threshold in steps of 10 counts for a maximum of 75. As noise levels reduce, the threshold value is reduced in 5 count steps.

While Easy Tune is enabled, the oversampling/digital gain ratio adjusts automatically for all channels, while a ratio of 8 is maintained. For example, if the oversampling is 16 and the gain is four, the oversampling is adjusted to 32.

Notes:

- The decision to enable or disable the Easy Tune feature can be determined during the development. Ideally,disabling the feature at runtime is not recommended. But if such a situation arises, all the buttons must be calibrated if Easy Tune feature is disabled at runtime.
- The channel threshold (THRESHOLD) register will be reset to the previously saved value when enabling the Easy Tune feature at runtime

3.8.2 Steps to Configure Easy Tune

1. Configure the DEVCTRL.ET register to enable or disable this feature.
2. Saving the settings (optional):
   – Writing `1` in DEVCTRL.SAVE to save the configuration.

3.9 Frequency Hopping with Autotune

This very effective noise avoidance feature is, by default, enabled.

Frequency Hopping with Autotune helps to ensure robust touch sensing under various electrical noise conditions. The noise on touch measurement data increases rapidly if the touch measurement frequency matches the frequency of the noise source. Predicting the noise frequency in varying conditions of product usage at the end customer is challenging. Frequency Hopping with Autotune enables touch to remain flexible and active to avoid noise after your product is shipped. Frequency Hopping with Autotune varies the touch measurement frequency. The built-in advanced algorithms continuously monitor incoming noise and dynamically adjust the touch measurement frequency. In short: The system actively avoids noise.

3.9.1 Frequency Hop

Operation

The Frequency Hop feature varies the touch measurement frequency between consecutive measurements. Response time slightly increases with Frequency Hop enabled. A typical increase in response time would be one Measurement Period. Refer to the Response Time for more information.

The three frequencies to acquire touch are pre-defined by the host. Refer to the Hop Frequency (FREQ) register. The MTCH2120 user has complete control over the used frequencies. During run-time, the host can change these frequencies.

Note: In Low-Power mode, Frequency Hoping (and, if recommended, oversampling along with it) is deactivated.

Initialization

The Frequency Hop feature is enabled by writing `1` to FREQHOP bit the Device Control (DEVCTRL) register. The three frequencies must be defined in FREQ[n] registers.

Enabling or disabling these features by the Host does not require a reset after the configuration update. Refer to the Configuration Summary for more details.

Steps to Configure Frequency Hop

1. Different frequencies can be selected by writing in the FREQ[n] register. Frequencies will be taken into effect from the following measurement. Refer to the Frequency Hop for details on the possible values and offset address.
   Note: The Autotune feature must be disabled before writing the register. See section 3.8.2.
2. The Frequency Hop feature is enabled by writing `1` in DEVCTRL.FREQHOP register.
3. Save the Settings (optional):
   Note: Use the "save" command to apply the updated configuration after resetting the device. Otherwise, this step is optional.
   - Write `1` in DEVCTRL.SAVE to save the configuration

3.9.2 Frequency Hop with Autotune

Operation

Frequency Hop with Autotune is a recommended setting for running a robust touch implementation. It operates autonomously and offers the flexibility required to counteract noise. The Autotune module is a superset of the Frequency Hop. It adds noise monitoring and Autotuning of the touch acquisition frequency according. The MTCH2120 starts performing touch measurement on three frequencies as default or defined by the Host. The noise on each frequency is monitored. If the noise level of a frequency is higher than the Noise Threshold (Refer Noise Threshold (NOISETHRESH) register) for Noise Integration (Refer Noise Integration (NOISEINT) register) times, then that particular frequency will be replaced by another frequency from a pool of frequencies (Frequency 0 to 15).

Initialization

Autotune is enabled by writing `1` to Autotune (AT) bits in the Device Control (DEVCTRL) register.

Enabling or disabling these features by the Host does not require reset after the configuration update. Refer to the Configuration Summary for more details.
Notes:

- Enable the Frequency Hop feature to enable Autotune
- The three frequencies value will be reset to the previously saved value when the Autotune is disabled
  - **Example:** The three touch acquisition frequencies were set to 0, 3 and 7 while the Autotune feature was disabled. This configuration is stored in the device's memory. Upon activation of the Autotune feature, the three frequencies will be dynamically adjusted according to the ambient noise level. If the feature is subsequently disabled, the three frequencies restore to the original values 0, 3 and 7.

Steps to Configure Autotune

1. Write value `1` in the DEVCTRL.AT register to enable this feature.

- Ensure that, DEVCTRL.FREQHOP must be enabled.

2. Saving the Settings (optional):
   - Write `1` in the DEVCTRL.SAVE to save the configuration
   - When this feature is enabled, the frequency (FREQ[x]) register will be reset to the previously saved value

3.10 GPIO Host Communication

MTCH2120 provides an option to configure up to 13 GPIOs. Out the 13 pins, 12 are multiplexed with button functionality, one pin is dedicated GPIO. Users can activate or deactivate the GPIO capabilities on a per-pin basis.

Operation

On power up, MTCH2120 configures the selected pins as GPIO based on the GPIO.PIN register. If a GPIO is enabled, the IN/OUT in the GPIO is configured as Input or Output based on the IN/OUT register values. The device may require up to 2 milliseconds to update the GPIO.IN and GPIO.OUT registers.

The GPIO capabilities for twelve pins are multiplexed with button functions. The users must disable the button operations before the pins are configured as GPIO. GPIOs will not function correctly if the Button is not disabled.

Note: The driven shield option must be disabled before using Button 11 as GPIO.

Initialization

After the Reset, the GPIO configuration will load from the Non-Volatile Memory (NVM). Then, save the GPIO configuration.

Steps to Configure a Button Pin as a GPIO Pin

1. Disable the Button n by writing bit `0` in SENCTRLn.EN register.
2. Configure the Button n pin as GPIO by writing `1` in GPIO.PIN register.
3. The GPIO.PIN and SENCTRLn.EN instructions will be configured during the initial setup of the device. Thus, it is crucial to save the configuration and perform a device reset (which might include either a hardware or software reset).

Steps to Configure the GPIO Pin as Output

1. Enable the output driver for pin n by writing `1` to bit n in the GPIO.DIR register.
   - Ensure the button option is disabled and the GPIO pin n is enabled by reading the corresponding bit from the SENCTRLn.EN and GPIO.PIN registers
2. Set the output driver for pin n to high or low level respectively by writing `1` or `0` to bit n in the GPIO.OUT register.

Steps to Configure the GPIO Pin n as Input

1. Enable the input driver for pin n by writing `0` to bit n in the GPIO.DIR register.
   - Ensure the button option is disabled and the GPIO pin n is enabled by reading the corresponding bit from the SENCTRLn.EN and GPIO.PIN registers
2. Read the input of pin n by reading bit n in the GPIO.IN register.

Note: Save the Settings. If the updated configuration of the GPIO.IN and GPIO.OUT registers are used after a device reset, the “save” command must be utilized. Otherwise, this step is optional.

3.11 Save

The Save function allows users to preserve their configurations in the device's non-volatile memory. The target is to have the complete device configuration ready for all coming start-ups w/o the necessity to rewrite the configuration by the host.

3.11.1 Initialize

Write `1` to DEVCTRL.SAVE to save the configuration. Promptly, this bit will reset to `0`.

In this scenario, the DEVCTRL.RESET value will not be considered during the save operation.

3.11.2 Operation

3.11.2.1 Save Configuration

This device features integrated None Volatile Memory NVM for storing the device configuration. Out of the box, the device is pre-configured with factory settings. However, users are encouraged to modify and overwrite these settings, with the changes preserved in the NVM. Starting from the next power-up, the MTCH2120 will execute according to these settings - no need for the host to rewrite the setting at every startup via the I^{2}C bus. Each time a save operation is initiated, the system will calculate a 16bit-CRC for the configuration to verify its integrity and prevent corruption.

If the DEVSTA.SAVE bit reads as `1`, it signifies a failure in the save operation (for additional information, refer to the Device Status section). It is imperative for users to check this bit after executing each save command. The bit will reset only on a successful save or when the device performs a reset.

Notes:

1. The FREQ[n] and THRESHOLD[n] register values will be reset to previously stored values when the Autotune and Easytune functions are enabled. Refer to the Autotune and Easytune sections for more information.
2. After issuing a Save command, the host application can monitor the CRC register to understand whether the CRC computed by the host matches to confirm a successful save operation. The host can use this CRC value later to see whether the device operates at the required configuration.

3.11.2.2 Load Configuration

Configurations will be retrieved from the NVM for every device reset. These configurations undergo a validation process using a computed 16bit-CRC to ensure their integrity. If the loaded configuration is corrupted, the device will revert to the manufacturer's default configuration settings, and the default configurations will be stored in the NVM.

It is important to note that if the DEVSTA.LOAD bit is read as `1`, signifying a failure in the load operation. Users must check this bit that follows every device reset. The bit will be reset to `0` only after a successful load operation.

3.12 Restore Default Configuration

The device is initially set with the default configuration provided by Microchip®. To restore the device to this default state, the users must set the DEVCTRL.SMCFG bit to '`1`'. Upon doing so, the bit will automatically reset to '`0`'. It is imperative for users to verify the integrity of the configuration by assessing the 16bit-CRC to ensure that it has been loaded correctly.

Notes:

1. The existing saved configuration will be cleared.
2. The suspended and disabled sensor register values will be cleared immediately, but the measurements will be performed after device reset.

Refer the following manufacturer configuration table for further information.

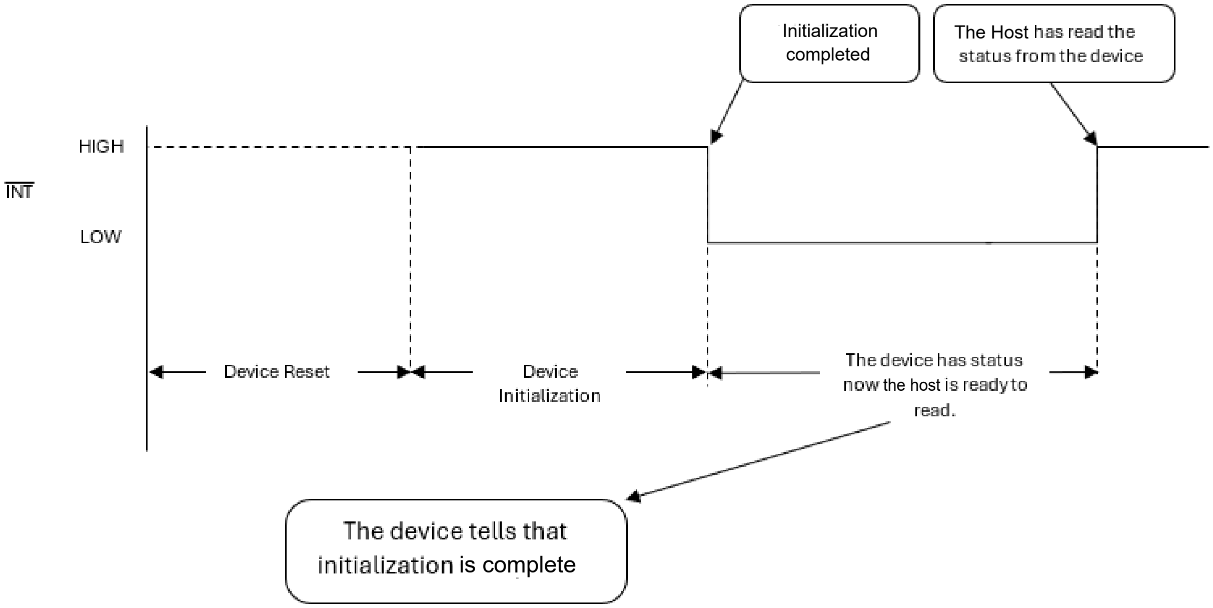
Register	Button[n]	Value
SENCTRL	BUTTON 0 to 11	0x01
	BUTTON 12 to 15	0x00
CSD	BUTTON 0 to 15	0x00
MCLKFREQ	BUTTON 0 to 15	1 MHz
FILTERLEVEL	BUTTON 0 to 15	FILTER_LEVEL_16
THRESHOLD	BUTTON 0 to 15	0x14
GAIN	BUTTON 0 to 15	0x01
HYSTERESIS	BUTTON 0 to 15	25%
AKSGRP	BUTTON 0 to 15	NO_AKS_GROUP
MP	-	30 ms
LPMP	-	100 ms
TIMEOUTCONF	-	5 sec
REBURSTMODE	-	REBURST_UNRESOLVED
DI	-	0x02
ANTITCHINT	-	0x05
MOD	-	0x00
DHT	-	4 sec
TCHDR	-	4 sec
ANTITCHDR	-	1 sec
ANTITCHRECALTHRESH	-	50%
NOISETHRESH	-	0x0F
NOISEINT	-	0x03
FREQ[0]	-	0x00
FREQ[1]	-	0x03
FREQ[2]	-	0x07
DEVCTRL	-	0x07D0
LUMPCONF[0]	-	0x0000
LUMPCONF[1]	-	0x0000
LUMPCONF[2]	-	0x0000
LUMPCONF[3]	-	0x0000
PIN	-	0x0000
DIR	-	0x0000
OUT	-	0x0000
IN	-	0x0000
CRC	-	0x1040

Note: The Cyclic Redundancy Check (CRC) values for the mentioned configuration; however, due to the activation of the Autotune and Easytune functionalities, modifications will be made to the channel threshold and frequency parameters.

3.13 $\overline{\text{INT}}$ - Interrupt

3.13.1 Interrupt Pin State on Device Initialization

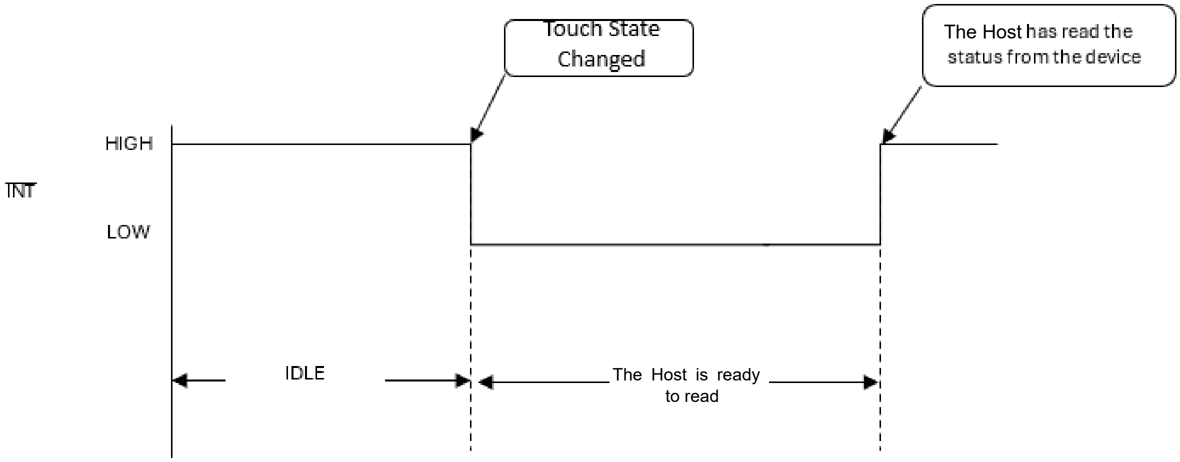
The interrupt pin will transition to a low state upon completion of the device's initialization. This state will be sustained until the host reads either the button status (BTTNSTA) or the sensor state (SENSTATE[n]).


Initialization completed
The Host has read the status from the device
HIGH
INT
LOW
Device Reset
Device Initialization
The device has status now the host is ready to read.
The device tells that initialization is complete

3.13.2 Interrupt Pin State on Touch State Changed

The interrupt pin will transition to a low state when the button state is from NO-DETECT to DETECT (or) DETECT to NO-DETECT. This state will be sustained until the host reads either the button status (BTTNSTA) or the sensor state (SENSTATE[n]).

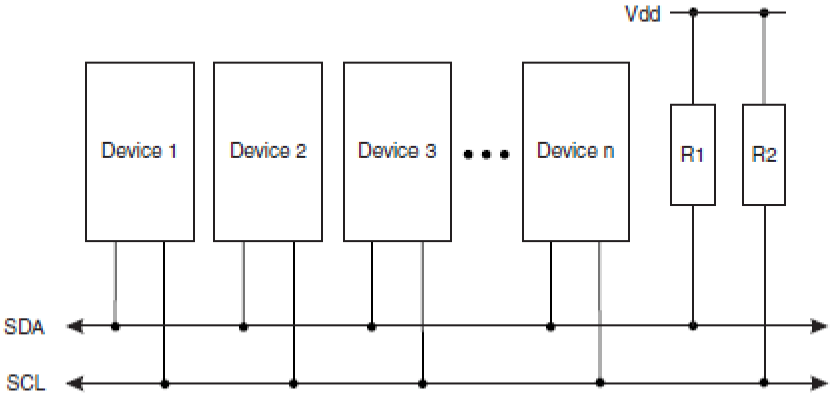
Figure 3-4.


Touch State Changed
The Host has read the status from the device
HIGH
INT
LOW
IDLE
The Host is ready to read

3.14 I^2C - Communication

Interface Bus

The device communicates with the host over an I^2C bus. Devices are connected to the I^2C bus, as shown in the figure below. Both bus lines are connected to V_{DD} via pull-up resistors. The bus drivers of all I^2C devices must be open-drain type, which implements a wired-AND function that allows all devices to drive the bus, one at a time. A low level on the bus is generated when a device outputs a zero.



General I²C Bus

The I²C provides a simple, bidirectional, two-wire communication bus consisting of:

- Serial Data Line (SDA) for packet transfer
- Serial Clock Line (SCL) for the bus clock

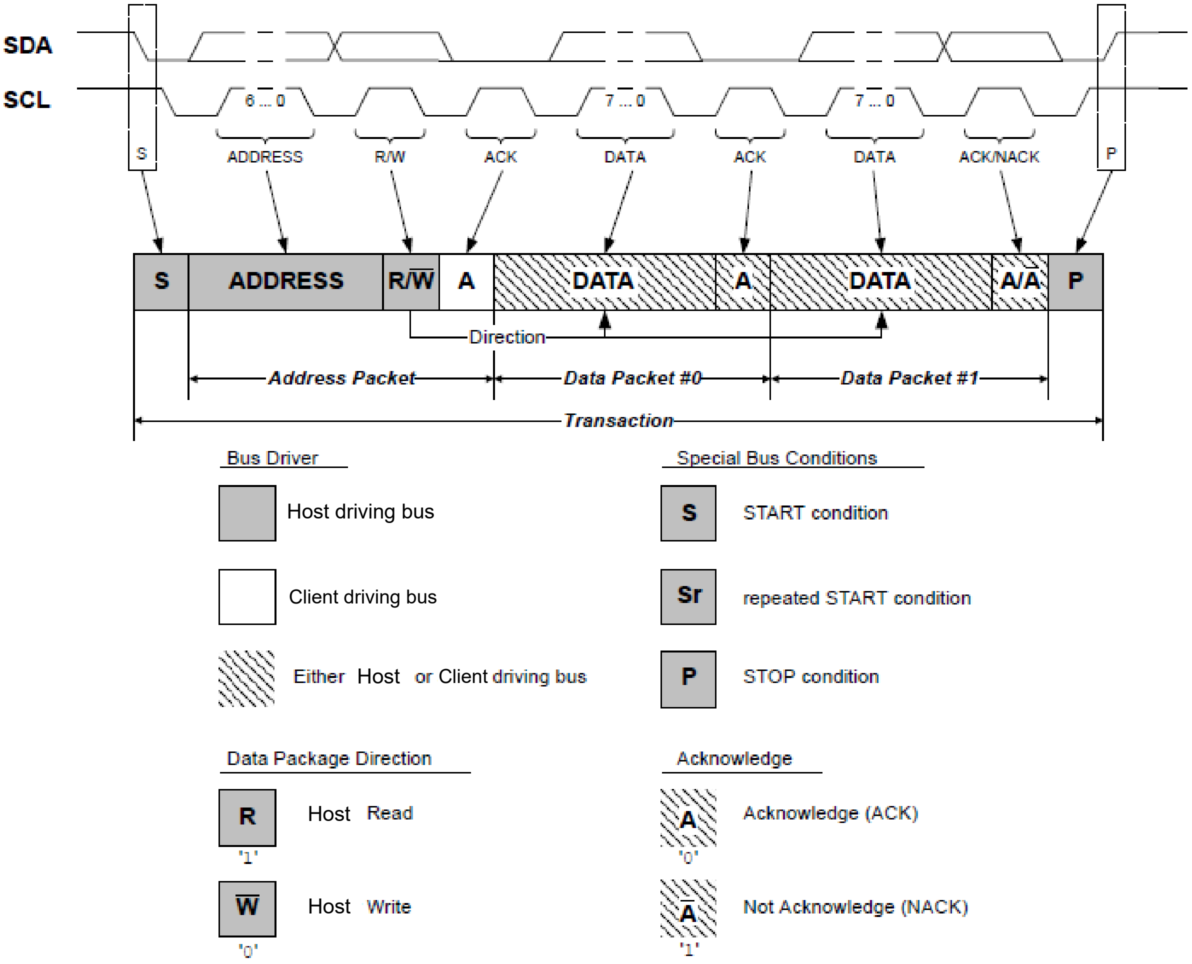
The two lines are open-collector lines (wired-AND).

The I²C bus topology is a simple and efficient method of interconnecting multiple devices on a serial bus. A device connected to the bus can be a host or a client. Only host devices can control the bus and the bus communication. A unique address is assigned to each client device connected to the bus, and the host will use it to control the client and initiate a transaction.

A host indicates the start of a transaction by issuing a Start condition (S) on the bus. The host provides the clock signal for the transaction. An address packet with a 7-bit client address (ADDRESS) and a direction bit, representing whether the host wishes to read or write data (R/W), are then sent.

The addressed I²C client will then acknowledge (ACK) the address, and data packet transactions can begin. Every 9-bit data packet consists of eight data bits followed by a 1-bit reply indicating whether the data were acknowledged by the receiver.

After all the data packets (DATA) are transferred, the host issues a Stop condition (P) on the bus to end the transaction.



3.14.1 Protocol

MTCH2120's registers can be accessed using 16-bit addressing.

Bit		7 6 5 4 3 2 1	0		15 14 13 12 11 10 9 8		23 22 21 20 19 18 17 16	
	S	ADDRESS	R/$\overline{W}$	A	MEMORY ADDRESS[MSB]	A	MEMORY ADDRESS[LSB]	A

Bit	30 29 28 27 26 25 24		
	DATA	N	P

□ Direction: host → client ■ Direction: client → host

START (S)

After a bus Idle state, a high-to-low transition of the SDAx line while the clock (SCLx) is high determines a Start condition. A Start condition must precede all data transfers.

DEVICE-ADDRESS

Initiate communication by transmitting the 7-bit device address along the start signal to select the client device. Upon successful selection, the client device will acknowledge by sending an acknowledgment bit to the host, indicating its readiness to receive a command.

MEMORY ADDRESS

This 16-bit selection determines which address the host wanted to read/write data from/to the device.

DATA

The state of the SDAx line represents valid data when, after a Start condition, the SDAx line is stable for the duration of the high period of the clock signal. There is one bit of data per SCLx clock.

ACKNOWLEDGE (A) OR NOT ACKNOWLEDGE (N)

All data byte transmissions must be Acknowledged (ACK) or Not Acknowledged (NACK) by the receiver. The receiver will pull the SDAx line low for an ACK or release the SDAx line for a NACK. The Acknowledge is a one-bit period using one SCLx clock.

STOP DATA TRANSFER (P)

A low-to-high transition of the SDAx line while the clock (SCLx) is high determines a Stop condition. All data transfers must end with a Stop condition.

3.14.1.1 Device Address

The device features an address selection pin that determines the address by reading the voltage level on the ADDR_SEL pin. The address configuration is established during the device's reset process and cannot be modified during runtime.

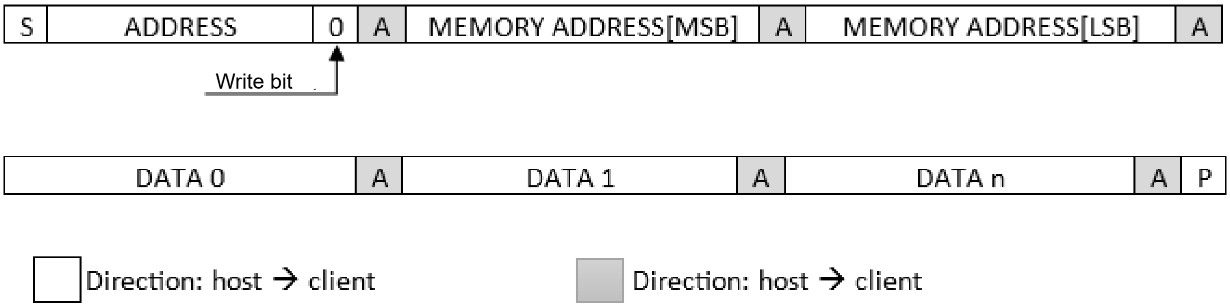
Table 3-3. ADDR_SEL

ADDR_SEL	Address
0	0x20
1	0x21

3.14.2 Write

To initiate communication, the host device must transmit a 16-bit memory address, which specifies the target location for data storage within the client device. Following the transmission of the memory address, it is imperative to await the acknowledgment signal ("ack") from the client device. Upon receipt of the "ack," the primary device may proceed with the sequential data transmission. It is crucial for the host device to vigilantly monitor the acknowledgment signal after each data transmission to ensure successful communication.

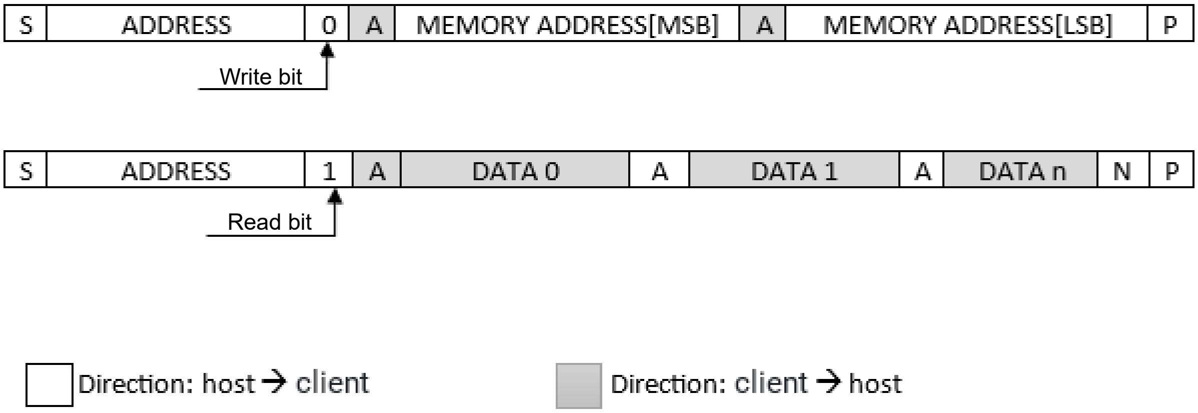
Figure 3-5.



Note: The "repeated start (Sr)" condition is ineffective for writing data to two distinct addresses; instead, the stop (P) bit should be utilized.

3.14.3 Read

The table provided delineates the data transmitted by the client device (MTCH2120). At the commencement of the communication process, the host device must input a 16-bit memory address to specify the precise memory location from which data retrieval is required. After the memory address input, the host device should await the acknowledgment signal (“ack”) from the client device. Upon receipt of the “ack”, the host device should reinitiate communication. This is followed by the host device sequentially reading data from the client device. It is essential for the host device to issue an 'ack' after each data read to maintain the flow of communication. The data transmission from the client device will cease when the host device sends a 'not acknowledged' signal (“nak”).



Users may utilize the "Repeated Start (R) bit" as an alternative to the stop (P) and start (S) bits to seamlessly continue communication.

3.14.4 Host Interface Support

Microchip provides host interface drivers for various 8- and 32-bit MCUs through MPLAB® Code Configurator (MCC) platform. The host MCU can read and write configuration from the device when using this host interface. Optionally, the tune data read from MCTH2120 can also be sent to the MPLAB Data Visualizer Touch Plugin. Refer to the link for more information.

3.15 BOD - Brown-Out Detector

This device has an on-chip Brown-out Detection circuit for monitoring the Operating Voltage (VCC) level during operation. By comparing the VCC to a fixed trigger level, it can determine if the device needs to be put into reset mode to prevent erratic operation.

This device operates from 3V to 5V. The BOD will reset the device if the VCC voltage falls below 3V, and the status will be updated in DEVSTA.BOD register.

This feature can be enabled or disabled by writing the value in the DEVCTRL.BOD register.
Note: The BOD monitors the VCC every 1msec. So, if voltage suddenly drops below the BOD level for a fraction of the time, then reset may not happen. Care should be taken in the system design that sudden voltage drops do not occur by adding adequate capacitors.

3.16 WDT - Watchdog Timer

The Watchdog Timer is a system function for monitoring the device operation. It allows the system to recover from deadlock situation by issuing a reset and the status will be updated in DEVSTA.WDT

This feature can be enabled or disabled by wirting the value in DEVCTRL.WDT.

Steps to Configure WDT

1. Writing a bit in the DEVCTRL.WDT register to enable or disable this feature.
2. Save the configuration by writing bit `1` in DEVCTRL.SAVE register.
3. Reset the device by writing bit `1` in DEVCTRL.RESET register.

Note: Disabling the WDT is strongly discouraged.

3.17 CRC - Cyclic Redundancy Check

This device is equipped with a 16-bit CRC-CCITT algorithm to ensure the integrity of loaded configurations and provides a way to confirm that the user stored configuration was loaded. By monitoring the CRC register, the host application can understand whether the loaded configuration is intended for their application.

Note: The CRC value will be calculated during the saving, loading, and loading processes of the user's configuration.

Sample Program

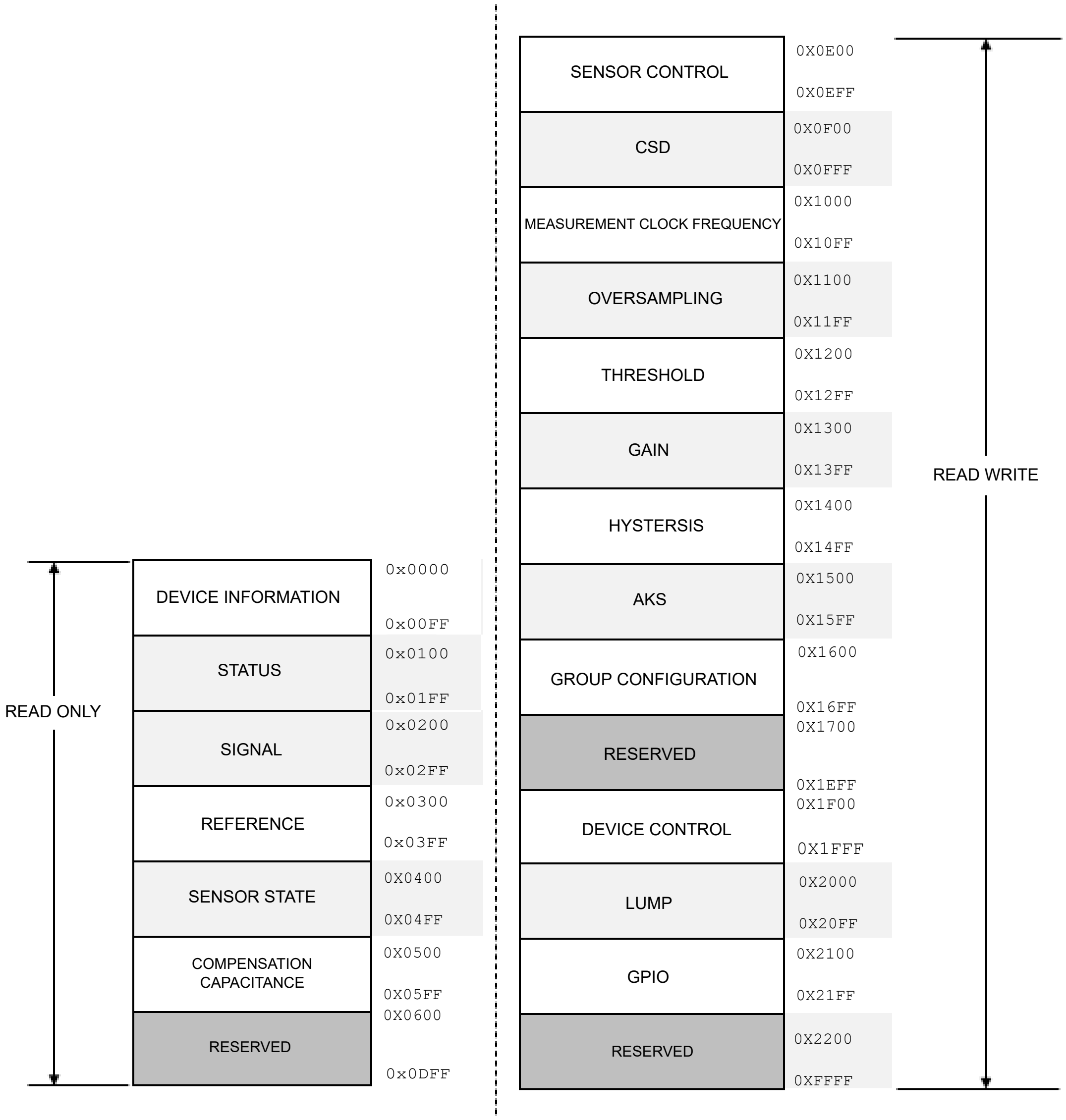
```
// Function to calculate the CRC-16 of a data array
uint16_t crc16_ccitt(const uint8_t *value, uint16_t length)
{
    // Initial value
    uint16_t crc = 0xFFFFu;
    for (uint16_t i = 0u; i < length; i++)
    {
        // XOR byte into most sig. byte of crc
        crc ^= (uint16_t)(((uint16_t)value[i]) << 8u);

        // Loop over each bit
        for (uint8_t j = 0u; j < 8u; j++)
        {
            // If the uppermost bit is 1...
            if ((crc & 0x8000u) != 0u)
            {
                // ...shift left and XOR with the polynomial
                crc = ((crc << ((uint16_t)1u)) ^ CRC16_POLY);
            }
            else
            {
                // Just shift left
                crc = (crc << 1u);
            }
        }
    }
    return crc; // Final CRC value
}
```

The following values have been used to compute the CRC.

Initial Value	0xFFFF
CRC16_POLY	0x1021

4. Memory Map



The Host communicates with the device using I^{2}C communication.

The Host possesses read-only access to the following parameters:

- Device Information:
  - Device Identification
  - Device Version
  - Device Features
  - Cyclic Redundancy Check
- Status:
  - Device Status
  - Button Status
- Signal

- Reference
- Sensor State
- Compensation Capacitance

These parameters can be Read and Written by the Host:

- Sensor Control
- CSD
- Measurement Clock Frequency
- Oversampling
- Threshold
- Gain
- Hysteresis
- AKS Group
- Global Configuration
- Device Control
- Lump Configuration
- GPIO

4.1 Configuration Summary

The following table summarizes whether a particular configuration update requires a reset.

Feature/Configuration	Reset required
Sensor Enable/Disable	Yes
Shield Selection (NONE/DS/ DS+)	Yes
Lump Configuration	Yes
GPIO Pin Enable/Disable	Yes
Load Configuration	Yes
WDT Enable/Disable	Yes
Frequency Hop Enable/Disable	No
Easy-Tune Enable/Disable	No
Low power Enable/Disable	No
Update GPIO DIR	No
Update GPIO OUT	No
BOD Enable/Disable during sleep	No
Save Configuration	No
Load Manufacturer Configuration	No
Software Reset	No
Calibrate individual sensor	No
Calibrate all the sensors	No
Suspend Sensor	No
Other parameters	No

Steps to be Followed to Update Configuration which Requires Reset After Update

1. Update Read/Write parameters in the Memory Block.
2. Save the Configuration by writing `1` to the Save Configuration to NVM (SAVE) bit in the Device Control (DEVCTRL) register, saving the updated configuration in NVM before the reset occurs. Refer to the Save section for more details.

3. Reset the device by writing `1` to the Device Reset (RESET) bit in the Device Control (DEVCTRL) register. Alternatively, initiate a hardware reset by lowering the voltage level on the RESET pin to reset the device.

Steps to be Followed to Update Configuration which does not Requires Reset After Update

1. Update Read/Write parameters in the Memory Block.
2. Save Configuration by writing `1` to the Save Configuration to NVM (SAVE) bit in the Device Control (DEVCTRL) register, saving the updated configuration in NVM before the reset occurs. Refer to the Save section for more details.
   Note: Utilize the "save" command if using the updated configuration after a device reset.
3. Updated values will be taken into effect from the following measurement.

4.2 Register Description

4.2.1 Memory Map Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	DEVID	7:0	DEVID[7:0]							
0x01	VER	7:0	MAJVER[3:0]				MINVER[3:0]			
0x02	DFT	7:0								KEY
		15:8				SHIELD[1:0]		LUMP	LP	FREQHOP
		23:16				ED[1:0]				SAVE
0x05	CRC	7:0	CRC[7:0]							
		15:8	CRC[15:8]							
0x07 ... 0xFF	Reserved									
0x0100	DEVSTA	7:0				SWRST	WDT	EXTRST	BOD	POR
		15:8						LOAD	SAVE	MEM
0x0102	BTNSTA	7:0	BTTNSTA[7:0]							
		15:8	BTTNSTA[15:8]							
0x0104 ... 0x01FF	Reserved									
0x0200	SIGNAL0	7:0	SIGNAL[7:0]							
		15:8	SIGNAL[15:8]							
...										
0x021E	SIGNAL15	7:0	SIGNAL[7:0]							
		15:8	SIGNAL[15:8]							
0x0220 ... 0x02FF	Reserved									
0x0300	REFERENCE0	7:0	REFERENCE[7:0]							
		15:8	REFERENCE[15:8]							
...										
0x031E	REFERENCE15	7:0	REFERENCE[7:0]							
		15:8	REFERENCE[15:8]							
0x0320 ... 0x03FF	Reserved									
0x0400	SENSTATE0	7:0	STATE[7:0]							
...										
0x040F	SENSTATE15	7:0	STATE[7:0]							
0x0410 ... 0x04FF	Reserved									
0x0500	CC0	7:0	CC[7:0]							
		15:8	CC[15:8]							
...										
0x051E	CC15	7:0	CC[7:0]							
		15:8	CC[15:8]							
0x0520 ... 0x0DFF	Reserved									
0x0E00	SENCTRL0	7:0	LP					SUSPEND	CAL	EN
...										
0x0E0F	SENCTRL15	7:0	LP					SUSPEND	CAL	EN
0x0E10 ... 0x0EFF	Reserved									
0x0F00	CSD0	7:0	CSD[7:0]							
...										
0x0F0F	CSD15	7:0	CSD[7:0]							
0x0F10 ... 0x0FFF	Reserved									
0x1000	MCLKFREQ0	7:0						MCLKFREQ[2:0]		

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
...										
0x100F	MCLKFREQ15	7:0						MCLKFREQ[2:0]		
0x1010 ... 0x10FF	Reserved									
0x1100	FILTERLEVEL0	7:0						FILTERLEVEL[2:0]		
...										
0x110F	FILTERLEVEL15	7:0						FILTERLEVEL[2:0]		
0x1110 ... 0x11FF	Reserved									
0x1200	THRESHOLD0	7:0	THRESHOLD[7:0]							
...										
0x120F	THRESHOLD15	7:0	THRESHOLD[7:0]							
0x1210 ... 0x12FF	Reserved									
0x1300	GAIN0	7:0						GAIN[2:0]		
...										
0x130F	GAIN15	7:0						GAIN[2:0]		
0x1310 ... 0x13FF	Reserved									
0x1400	HYSTERESIS0	7:0							HYSTERESIS[1:0]	
...										
0x140F	HYSTERESIS15	7:0							HYSTERESIS[1:0]	
0x1410 ... 0x14FF	Reserved									
0x1500	AKSGRP0	7:0						AKSGRP[2:0]		
...										
0x150F	AKSGRP15	7:0						AKSGRP[2:0]		
0x1510 ... 0x15FF	Reserved									
0x1600	MP	7:0	MP[7:0]							
0x1601	Reserved									
0x1602	LPMP	7:0	LPMP[7:0]							
0x1603	Reserved									
0x1604	TIMEOUTCONF	7:0	TIMEOUTCONF[7:0]							
		15:8	TIMEOUTCONF[15:8]							
0x1606	REBURSTMODE	7:0							REBURSTMODE[1:0]	
0x1607	DI	7:0	TCHDI[7:0]							
0x1608	ANTITCHINT	7:0	ANTITCHINT[7:0]							
0x1609	MOD	7:0	MOD[7:0]							
0x160A	DHT	7:0	DHT[7:0]							
0x160B	TCHDR	7:0	TCHDR[7:0]							
0x160C	ANTITCHDR	7:0	ANTITCHDR[7:0]							
0x160D	ANTITCHRECALTHRESH	7:0						ANTITCHRECALTHRESH[2:0]		
0x160E	NOISETHRESH	7:0	NOISETHRESH[7:0]							
		15:8	NOISETHRESH[15:8]							
0x1610	NOISEINT	7:0	NOISEINT[7:0]							
0x1611	FREQ0	7:0					FREQ[3:0]			
0x1612	FREQ1	7:0					FREQ[3:0]			
0x1613	FREQ2	7:0					FREQ[3:0]			
0x1614 ... 0x1EFF	Reserved									
0x1F00	DEVCTRL	7:0	AT	FREQHOP	DRIFTGAIN	DSP	DS	DLPLB	LP	CAL
		15:8			RESET	SAVE	SMCFG	BOD	WDT	ET

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x1F02 ... 0x1FFF	Reserved									
0x2000	LUMPCONF0	7:0	LUMPCONF[7:0]							
		15:8					LUMPCONF[11:8]			
0x2002 ... 0x2007	Reserved									
0x2008	LUMPCONF1	7:0	LUMPCONF[7:0]							
		15:8					LUMPCONF[11:8]			
0x200A ... 0x200F	Reserved									
0x2010	LUMPCONF2	7:0	LUMPCONF[7:0]							
		15:8					LUMPCONF[11:8]			
0x2012 ... 0x2017	Reserved									
0x2018	LUMPCONF3	7:0	LUMPCONF[7:0]							
		15:8					LUMPCONF[11:8]			
0x201A ... 0x20FF	Reserved									
0x2100	PIN	7:0	PIN[7:0]							
		15:8				PIN[12:8]				
0x2102 ... 0x2107	Reserved									
0x2108	DIR	7:0	DIR[7:0]							
		15:8				DIR[12:8]				
0x210A ... 0x210F	Reserved									
0x2110	OUT	7:0	OUT[7:0]							
		15:8				OUT[12:8]				
0x2112 ... 0x2117	Reserved									
0x2118	IN	7:0	IN[7:0]							
		15:8				IN[12:8]				

4.2.2 Device Identification Number

Name: DEVID
Offset: 0x0000
Reset: 0x0B
Property: R

Unique identification number of MTCH2120 device.

Bit	7	6	5	4	3	2	1	0
	DEVID[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	1	0	1	1

Bits 7:0 – DEVID[7:0]

Value
0x0B

4.2.3 Device Version

Name: VER
Offset: 0x0001
Reset: 0x10
Property: R

Bit	7	6	5	4	3	2	1	0
	MAJVER[3:0]				MINVER[3:0]			
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	1	0	0	0	0

Bits 7:4 – MAJVER[3:0] Major Version Number

Major Version Number
1

Bits 3:0 – MINVER[3:0] Minor Version Number

Minor Version Number
0

4.2.4 Device Features.

Name: DFT
Offset: 0x0002
Reset: 0x15
Property: R

This device supports the following features.

Bit	23	22	21	20	19	18	17	16
				ED[1:0]				SAVE
Access				R	R			R
Reset				1	0			1

Bit	15	14	13	12	11	10	9	8
				SHIELD[1:0]		LUMP	LP	FREQHOP
Access				R	R	R	R	R
Reset				1	1	1	1	1

Bit	7	6	5	4	3	2	1	0
								KEY
Access								R
Reset								1

Bits 20:19 – ED[1:0] This device supports the Error Detection feature

Value	Algorithm
2	16-bit CRC CCIT Algorithm is used to validate the configurations

Bit 16 – SAVE This device supports save option

Bits 12:11 – SHIELD[1:0] This device supports Driven Shield+ and Dedicated Driven Shield

Bit 10 – LUMP This device supports the Lump feature

Bit 9 – LP This device supports the Low Power feature

Bit 8 – FREQHOP This device supports the Frequency Hop feature

Bit 0 – KEY This device supports the button feature

4.2.5 Cyclic Redundancy Check

Name: CRC
Offset: 0x0005
Reset: 0xXX
Property: R

The 16-bit CRC-CCITT algorithm has been integrated to validate the configuration.

Bit	15	14	13	12	11	10	9	8
	CRC[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	x	x	x	x	x	x	x	x

Bit	7	6	5	4	3	2	1	0
	CRC[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	x	x	x	x	x	x	x	x

Bits 15:0 – CRC[15:0] The CRC value of the saved configuration

4.2.6 Device Status

Name: DEVSTA
Offset: 0x0100
Reset: 0xXX
Property: R

Bit	15	14	13	12	11	10	9	8
						LOAD	SAVE	MEM
Access						R	R	R
Reset						x	x	x

Bit	7	6	5	4	3	2	1	0
				SWRST	WDT	EXTRST	BOD	POR
Access				R	R	R	R	R
Reset				x	x	x	x	x

Bit 10 – LOAD Load Failure
This bit is set if it fails to load the configuration.

Bit 9 – SAVE Save Failure
This bit is set if it fails to save the configuration.

Bit 8 – MEM Memory Corruption
This bit is set if no configuration is loaded.

Bit 4 – SWRST Software Reset
This bit is set if a Software Reset occurs.

Bit 3 – WDT Watchdog Timer
This bit is set if a Watchdog Reset occurs.

Bit 2 – EXTRST External Reset
This bit is set if an External Reset occurs.

Bit 1 – BOD Brown-out Detection
This bit is set if a Brown-out Detection (BOR) occurs.

Bit 0 – POR Power-on Reset
This bit is set if a Power-on Reset (POR) occurs.
After a POR, only the POR flag is set and all the other flags are cleared. No other flags can be set before a full system boot is run after the POR.

4.2.7 Button Status

Name: BTNSTA
Offset: 0x0102
Reset: 0x0
Property: R

Bit	15	14	13	12	11	10	9	8
	BTTNSTA[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	BTTNSTA[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – BTTNSTA[15:0] Button Status Mask
This bit field shows the state of the 12 Buttons (Bit 0 to 11) and four Lumps (Bit 12 to 15).

Value	Description
1	Touch Detected
0	No Detect

4.2.8 Signal

Name: SIGNAL
Offset: 0x0200 + n*0x02 [n=0..15]
Reset: 0x0
Property: R

Bit	15	14	13	12	11	10	9	8
	SIGNAL[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	SIGNAL[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – SIGNAL[15:0] Channel-n signal value

4.2.9 Reference

Name: REFERENCE
Offset: 0x0300 + n*0x02 [n=0..15]
Reset: 0x0
Property: R

Bit	15	14	13	12	11	10	9	8
	REFERENCE[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	REFERENCE[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – REFERENCE[15:0] Channel-n reference value

4.2.10 Sensor State

Name: SENSTATE
Offset: 0x0400 + n*0x01 [n=0..15]
Reset: 0x0
Property: R

Bit	7	6	5	4	3	2	1	0
	STATE[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – STATE[7:0] Sensor state values are as follows:

Value	Sensor State
0x00	The sensor has been disabled and will not perform measurements
0x01	The sensor is in an initialized state and has yet to complete the calibration
0x02	The sensor is currently being calibrated
0x03	Touch not detected
0x04	A confirmation scan is being performed to confirm a touch
0x85	No touch is detected
0x86	A confirmation scan is being performed to confirm a touch removal
0x07	Away-from-touch recalibration is being performed for the sensor
0x08	The sensor is in a suspended state
0x09	The calibration process failed for the sensor
Note: Bit 7 is set in each state where the sensor is "In Detect".	

4.2.11 Compensation Capacitance

Name:	CC
Offset:	0x0500 + n*0x02 [n=0..15]
Reset:	0xXX
Property:	R

Bit	15	14	13	12	11	10	9	8
	CC[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	x	x	x	x	x	x	x	x

Bit	7	6	5	4	3	2	1	0
	CC[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	x	x	x	x	x	x	x	x

Bits 15:0 – CC[15:0] Channel-n CC value

4.2.12 Sensor Control

Name: SENCTRL
Offset: 0x0E00 + n*0x01 [n=0..15]
Reset: Loaded from NVM
Property: R/W

Bit	7	6	5	4	3	2	1	0
	LP					SUSPEND	CAL	EN
Access	R/W					R/W	R/W	R/W
Reset	x					x	x	x

Bit 7 – LP Low-Power Sensor
Writing '1' to this bit configures Sensor-n as a low-power sensor.

Bit 2 – SUSPEND Sensor Suspension
Writing '1' to this bit suspends Sensor-n.

Bit 1 – CAL Sensor Calibration
Writing '1' to this bit calibrates Sensor-n.

Bit 0 – EN Sensor Enable
Writing '1' to this bit enables Sensor-n.

4.2.13 CSD

Name: CSD
Offset: 0x0F00 + n*0x01 [n=0..15]
Reset: Loaded from NVM
Property: R/W

Bit	7	6	5	4	3	2	1	0
	CSD[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	x	x	x	x	x	x	x	x

Bits 7:0 – CSD[7:0] Channel-n Node CSD

CSD
0 to 28

4.2.14 Measurement Clock Frequency

Name: MCLKFREQ
Offset: 0x1000 + n*0x01 [n=0..15]
Reset: Loaded from NVM
Property: R/W

Bit	7	6	5	4	3	2	1	0
						MCLKFREQ[2:0]		
Access						R/W	R/W	R/W
Reset						x	x	x

Bits 2:0 – MCLKFREQ[2:0] Channel-n Measurement Frequencies

Values	Frequency
2	1 MHz
3	500 kHz

4.2.15 Oversampling

Name: FILTERLEVEL
Offset: 0x1100 + n*0x01 [n=0..15]
Reset: Loaded from NVM
Property: R/W

Bit	7	6	5	4	3	2	1	0
						FILTERLEVEL[2:0]		
Access						R/W	R/W	R/W
Reset						x	x	x

Bits 2:0 – FILTERLEVEL[2:0] Channel-n Node filterlevel

Values	Filterlevel
0	1
1	2
2	4
3	8
4	16
5	32
6	64
7	Reserved

4.2.16 Threshold

Name:	THRESHOLD
Offset:	0x1200 + n*0x01 [n=0..15]
Reset:	Loaded from NVM
Property:	R/W

Bit	7	6	5	4	3	2	1	0
	THRESHOLD[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	x	x	x	x	x	x	x	x

Bits 7:0 – THRESHOLD[7:0] Channel-n Threshold

4.2.17 Gain

Name: GAIN
Offset: 0x1300 + n*0x01 [n=0..15]
Reset: Loaded from NVM
Property: R/W

Bit	7	6	5	4	3	2	1	0
						GAIN[2:0]		
Access						R/W	R/W	R/W
Reset						x	x	x

Bits 2:0 – GAIN[2:0] Channel-n Node Gain

Values	Node Gain
0	1
1	2
2	4
3	8

4.2.18 Hysteresis

Name: HYSTERESIS
Offset: 0x1400 + n*0x01 [n=0..15]
Reset: Loaded from NVM
Property: R/W

Bit	7	6	5	4	3	2	1	0
							HYSTERESIS[1:0]	
Access							R/W	R/W
Reset							x	x

Bits 1:0 – HYSTERESIS[1:0] Channel-n Hysteresis

Value	Hysteresis
0	50%
1	25%
2	12.5%
3	6.25%

4.2.19 AKS Group

Name: AKSGRP
Offset: 0x1500 + n*0x01 [n=0..15]
Reset: Loaded from NVM
Property: R/W

Bit	7	6	5	4	3	2	1	0
						AKSGRP[2:0]		
Access						R/W	R/W	R/W
Reset						x	x	x

Bits 2:0 – AKSGRP[2:0] Channel-n AKS group

Values	Description
0	Channel-n does not belong to any AKS group
1	Channel-n belongs to the AKS group1
2	Channel-n belongs to the AKS group 2
3	Channel-n belongs to the AKS group 3
4	Channel-n belongs to the AKS group 4
5	Channel-n belongs to the AKS group 5
6	Channel-n belongs to the AKS group 6
7	Channel-n belongs to the AKS group 7

4.2.20 Touch Sensor Configuration

Global configuration parameters.

Configuration parameters that are applied to all the enabled sensors.

4.2.20.1 Touch Measurement Period

Name:	MP
Offset:	0x1600
Reset:	Loaded from NVM
Property:	R/W

Bit	7	6	5	4	3	2	1	0
	MP[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	x	x	x	x	x	x	x	x

Bits 7:0 – MP[7:0] Touch Measurement Period
The time between the start of one measurement and the following, including Measurement time and Sleep time.

Range
0-250 (measured in ms)

Note: If the configured Touch Measurement Period is less than the minimum required Measurement Period for all the enabled channels, then the minimum Measurement Period will be applied automatically.

4.2.20.2 Low-Power Measurement Period

Name:	LPMP
Offset:	0x1602
Reset:	Loaded from NVM
Property:	R/W

Bit	7	6	5	4	3	2	1	0
	LPMP[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	x	x	x	x	x	x	x	x

Bits 7:0 – LPMP[7:0] Low-Power Measurement Period

The Touch Measurement Period when the Low-Power feature is enabled.

Range
0-250 (measured in ms)

Note: If the configured Touch Measurement Period is less than the minimum required Measurement Period for all the enabled channels, then the minimum Measurement Period will be applied automatically.

4.2.20.3 Time-Out Config

Name:	TIMEOUTCONF
Offset:	0x1604
Reset:	Loaded from NVM
Property:	R/W

Bit	15	14	13	12	11	10	9	8
	TIMEOUTCONF[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	x	x	x	x	x	x	x	x

Bit	7	6	5	4	3	2	1	0
	TIMEOUTCONF[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	x	x	x	x	x	x	x	x

Bits 15:0 – TIMEOUTCONF[15:0] Time-Out Config

The required duration to activate the low-power feature after the low-power feature is enabled through software.

Range
0-65535 (measured in ms)

4.2.20.4 Sensor Reburst Mode

Name: REBURSTMODE
Offset: 0x1606
Reset: Loaded from NVM
Property: R/W

Bit	7	6	5	4	3	2	1	0
							REBURSTMODE[1:0]	
Access							R/W	R/W
Reset							x	x

Bits 1:0 – REBURSTMODE[1:0] Sensor Reburst Mode

Operates in three modes.

Value	Description
0	REBURST_NONE
1	REBURST_UNRESOLVED
2	REBURST_ALL

4.2.20.5 Detect Integration

Name:	DI
Offset:	0x1607
Reset:	Loaded from NVM
Property:	R/W

Bit	7	6	5	4	3	2	1	0
	TCHDI[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	x	x	x	x	x	x	x	x

Bits 7:0 – TCHDI[7:0] DI

The number of repeated measurements required to confirm both a touch and an out-of-touch detection.

Range
0-255

4.2.20.6 Sensor Anti-Touch-Integration

Name:	ANTITCHINT
Offset:	0x1608
Reset:	Loaded from NVM
Property:	R/W

Bit	7	6	5	4	3	2	1	0
	ANTITCHINT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	x	x	x	x	x	x	x	x

Bits 7:0 – ANTITCHINT[7:0] Sensor Anti-Touch-Integration

The number of repeated measurements required to confirm anti-touch recalibration.

Range
0 (Disabled)
1-255

4.2.20.7 Change as Max-On-Duration (MOD)

Name:	MOD
Offset:	0x1609
Reset:	Loaded from NVM
Property:	R/W

Bit	7	6	5	4	3	2	1	0
	MOD[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	x	x	x	x	x	x	x	x

Bits 7:0 – MOD[7:0] Max-On-Duration in milliseconds

The maximum duration for which a sensor can be in detect is defined by MOD. If a sensor is in detect for more than this period, the sensor will be calibrated.

Value	MOD (ms)
0 (Disabled)	0
1-255	Value * 200

4.2.20.8 Sensor Drift Hold Time

Name:	DHT
Offset:	0x160A
Reset:	Loaded from NVM
Property:	R/W

Bit	7	6	5	4	3	2	1	0
	DHT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	x	x	x	x	x	x	x	x

Bits 7:0 – DHT[7:0] Sensor Drift Hold Time

The number of timer periods to stop drifting after the touch event.

Value	Period (ms)
0 (Disabled)	0
1-255	Value * 200

4.2.20.9 Sensor Touch Drift Rate

Name:	TCHDR
Offset:	0x160B
Reset:	Loaded from NVM
Property:	R/W

Bit	7	6	5	4	3	2	1	0
	TCHDR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	x	x	x	x	x	x	x	x

Bits 7:0 – TCHDR[7:0] Sensor Touch Drift Rate

The number of timer periods to count down between towards touch drifts.

Value	Period (ms)
0 (Disabled)	0
1-to-255	Value * 200

4.2.20.10 Sensor Anti-Touch Drift Rate

Name: ANTITCHDR
Offset: 0x160C
Reset: Loaded from NVM
Property: R/W

Bit	7	6	5	4	3	2	1	0
	ANTITCHDR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	x	x	x	x	x	x	x	x

Bits 7:0 – ANTITCHDR[7:0] Sensor Anti-Touch Drift Rate
The number of timer periods to countdown between away-from-touch drifts.

Value	Period (ms)
0 (Disabled)	0
1-255	Value * 200

4.2.20.11 Sensor Anti-Touch Recalibration Threshold

Name: ANTITCHRECALTHRESH
Offset: 0x160D
Reset: Loaded from NVM
Property: R/W

Bit	7	6	5	4	3	2	1	0
						ANTITCHRECALTHRESH[2:0]		
Access						R/W	R/W	R/W
Reset						x	x	x

Bits 2:0 – ANTITCHRECALTHRESH[2:0] Sensor Anti-Touch Recalibration Threshold
A scale-down of the touch threshold to set the anti-touch threshold. It can be read and updated through I^{2}C communication.

Value	Percentage
0	100%
1	50%
2	25%
3	12.5%
4	6.25%
5	Reserved
6	Reserved
7	Reserved

4.2.20.12 Noise Threshold

Name:	NOISETHRESH
Offset:	0x160E
Reset:	Loaded from NVM
Property:	R/W

Bit	15	14	13	12	11	10	9	8
	NOISETHRESH[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	x	x	x	x	x	x	x	x

Bit	7	6	5	4	3	2	1	0
	NOISETHRESH[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	x	x	x	x	x	x	x	x

Bits 15:0 – NOISETHRESH[15:0] Noise Threshold

If the noise level is more than the noise threshold for the noise integration number of measurement cycles, then the noisy frequency will be replaced with another frequency.

Range
0-65535

4.2.20.13 Noise Integration

Name:	NOISEINT
Offset:	0x1610
Reset:	Loaded from NVM
Property:	R/W

Bit	7	6	5	4	3	2	1	0
	NOISEINT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	x	x	x	x	x	x	x	x

Bits 7:0 – NOISEINT[7:0] Noise Integration

The noise integration number of measurement cycles to replace the noisy frequency.

Range
0-255

4.2.20.14 Hop Frequency

Name: FREQ
Offset: 0x1611 + n*0x01 [n=0..2]
Reset: Loaded from NVM
Property: R/W

Bit	7	6	5	4	3	2	1	0
					FREQ[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					x	x	x	x

Bits 3:0 – FREQ[3:0] Hop Frequency

Hopping is done between ‘n’ frequencies, where each frequency supports up to 16 PTC Sampling Delay values.

Range
0-15

4.2.21 Device Control

Name: DEVCTRL
Offset: 0x1F00
Reset: Loaded from NVM
Property: R/W

Bit	15	14	13	12	11	10	9	8
			RESET	SAVE	SMCFG	BOD	WDT	ET
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	x	x	x

Bit	7	6	5	4	3	2	1	0
	AT	FREQHOP	DRIFTGAIN	DSP	DS	DLPLB	LP	CAL
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	x	x	0	x	x	x	x	0

Bit 13 – RESET Device Reset
Write '1' to this bit to reset the device

Bit 12 – SAVE Save Configuration to NVM
Write '1' to this bit to save the current configuration into the device.

Bit 11 – SMCFG Save Manufacturer Configuration
Write '1' to save the factory configuration.

Bit 10 – BOD Brown-Out-Detect
Writing '1' to this bit enables BOD.

Bit 9 – WDT Watchdog Timer
Writing '1' to this bit enables WDT.

Bit 8 – ET Easy Tune
Writing '1' to this bit enables the Easy Tune feature.

Bit 7 – AT Autotune
Writing '1' to this bit enables the Auto-tuning in the Frequency Hopping feature.

Bit 6 – FREQHOP Frequency Hopping
Writing '1' to this bit enables the Frequency Hopping feature.

Bit 5 – DRIFTGAIN Drift Gain
Writing '1' to this bit enables the drifting based on the Node Gain value.

Bit 4 – DSP Driven Shield Plus
Writing '1' to this bit enables the Driven Shield Plus feature.

Bit 3 – DS Driven Shield
Writing '1' to this bit enables the Driven Shield feature.

Bit 2 – DLPLB Drift Low Power Lump Buttons
Writing '1' to this bit enables the drift of the buttons configured into the low-power lump.

Bit 1 – LP Low-Power

Writing a '1' to this bit puts the device into Low-Power mode.

Bit 0 – CAL Triggers calibration for all the sensors

Writing '1' to this bit triggers calibration for all the sensors.

4.2.22 Device Control Bits Priority

The device is designed to prohibit simultaneous writing of the SMCFG, SAVE, RESET bits. By their assigned priority, other bits will be reset to zero (Refer to the Priority table).

For example, when these three bits are set to '1' concurrently, the SMCFG command will prioritize and execute. The remaining two bits will be disregarded and not executed.

Table 4-1. Priority Table

Register	Priority Number
SMCFG	0
SAVE	1
RESET	2

Note: Priority Number '0' is the highest priority.

4.2.23 Lump Configuration

Name: LUMPCONF
Offset: 0x2000 + n*0x08 [n=0..3]
Reset: Loaded from NVM
Property: R/W

Bit	15	14	13	12	11	10	9	8
					LUMPCONF[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					x	x	x	x

Bit	7	6	5	4	3	2	1	0
	LUMPCONF[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	x	x	x	x	x	x	x	x

Bits 11:0 – LUMPCONF[11:0] Enabled sensors are lumped together in Lump n
Mask to enable sensors that must be lumped together in Lump n.

4.2.24 GPIO

4.2.24.1 GPIO Pins

Name: PIN
Offset: 0x2100
Reset: Loaded from NVM
Property: R/W

Bit	15	14	13	12	11	10	9	8
				PIN[12:8]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				x	x	x	x	x

Bit	7	6	5	4	3	2	1	0
	PIN[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	x	x	x	x	x	x	x	x

Bits 12:0 – PIN[12:0] GPIO pin mask
Enable/Disable the GPIOn pin.

Values	Description
0	GPIOn is disabled
1	GPIOn is enabled

4.2.24.2 GPIO Direction

Name:	DIR
Offset:	0x2108
Reset:	Loaded from NVM
Property:	R/W

Bit	15	14	13	12	11	10	9	8
				DIR[12:8]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				x	x	x	x	x

Bit	7	6	5	4	3	2	1	0
	DIR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	x	x	x	x	x	x	x	x

Bits 12:0 – DIR[12:0] GPIO direction

This bit field controls the direction for each GPIOn pin.

Value	Description
0	Configure the GPIOn pin as an output
1	Configure the GPIOn pin as an input

4.2.24.3 GPIO Output Value

Name:	OUT
Offset:	0x2110
Reset:	Loaded from NVM
Property:	R/W

Bit	15	14	13	12	11	10	9	8
				OUT[12:8]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				x	x	x	x	x

Bit	7	6	5	4	3	2	1	0
	OUT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	x	x	x	x	x	x	x	x

Bits 12:0 – OUT[12:0] Output Buffer

This bit field controls the output driver level for each GPIOn pin.

This configuration works only when the output driver (GPIOn.DIR) is configured for the corresponding pin, and this pin must be enabled (GPIOn.EN).

Value	Description
0	The GPIOn output is driven low
1	The GPIOn output is driven high

4.2.24.4 GPIO Input Value

Name:	IN
Offset:	0x2118
Reset:	0x00
Property:	R

Bit	15	14	13	12	11	10	9	8
				IN[12:8]				
Access				R	R	R	R	R
Reset				0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	IN[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 12:0 – IN[12:0] Input Value

This bit field shows the state of the GPIOn pins.

This configuration works only when the output driver (GPIOn.DIR) is configured for the corresponding pin, and this pin must be enabled (GPIOn.EN).

Value	Description
0	The voltage level on GPIOn is low
1	The voltage level on GPIOn is high

5. Response Time

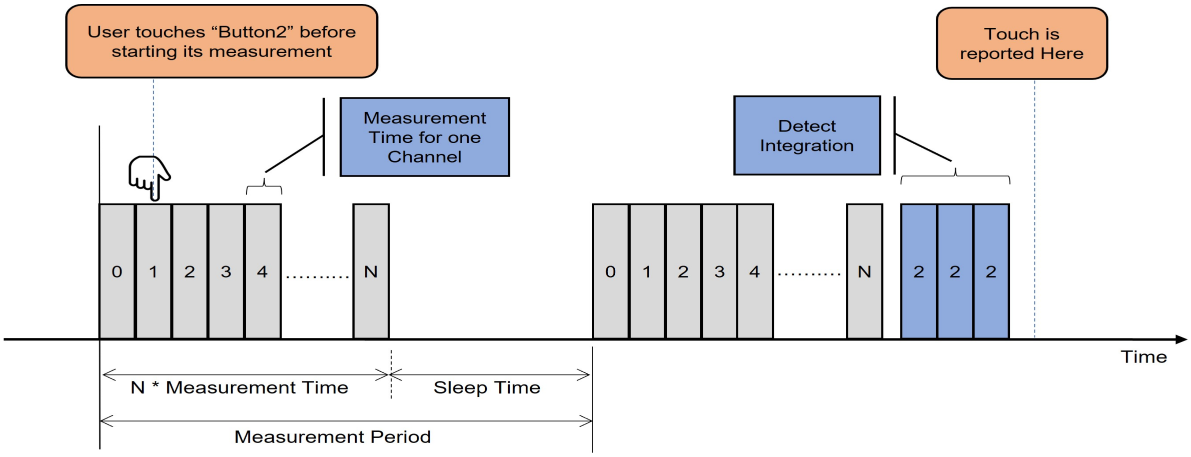
The response time refers to the time between the physical touch of a button and a touch reported to the host.

Response times change with the following configurations:

- Measurement Period
- Frequency Hop
- CSD
- Measurement Clock Frequency
- Detect Integration (DI)
- Response Time
- Oversampling (Filter Level)

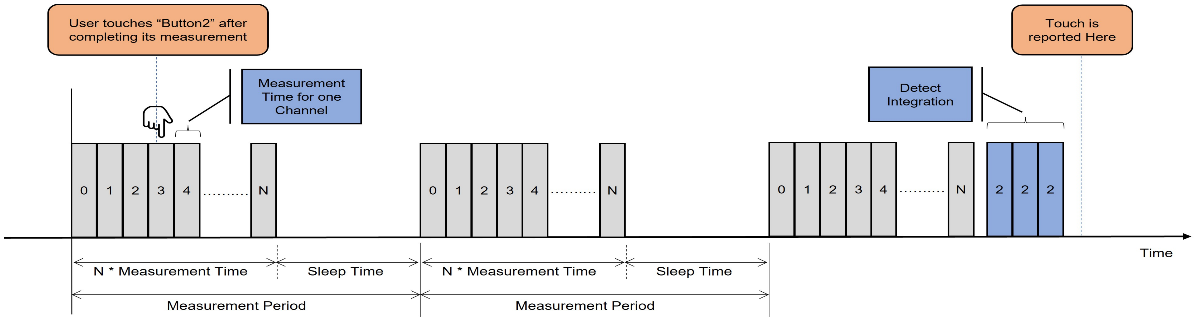
Refer to the following topics for more information.

Best-Case Response Time



The best-case response time is calculated when the button is touched before its measurement. The formula to calculate the best-case response time is present in the Formula chapter.

Worst-Case Response Time



The worst-case response time is calculated when the button is touched just after it was measured. The formula to calculate the worst-case response time is present in the Formula chapter.

5.1 Formula

The response time will be affected primarily by measurement time, Reburst mode, and frequency hopping.

Table 5-1. Formula to Calculate Best-Case Response Time

Frequency Hop	Formula
Disabled	Measurement Time of a single channel × Number of active sensors + Reburst Time
Enabled	Measurement Period + Measurement Time of a single channel × Number of active sensors + Reburst Time

Table 5-2. Formula to Calculate Worst-Case Response Time

Frequency Hop	Formula
Disabled	Measurement Period + Measurement Time of a single channel × Number of active sensors + Reburst Time
Enabled	2 × Measurement Period + Measurement Time of a single channel × Number of active sensors + Reburst Time

The measurement Time and Reburst Time used in the formulas is calculated as follows.

Measurement Time for a Single Channel:

(30 + 6 × CSD + Hop Freqency) x (1 / Measurement Clock Frequency) × FILTERLEVEL

Table 5-3. Reburst Time

Reburst Mode	Reburst Time
Reburst All	DI × Number of active sensors × Measurement Time of a single channel
Reburst Unresolved	DI × Number of sensors touched × Measurement Time of a single channel
Reburst None	DI × Measurement Period

5.2 Examples

This section provides calculated response times for various scenarios and configurations. Use the formula to compute response time for other configurations.

5.2.1 Example to Calculate Measurement Time of a Single Channel

Compute the measurement time before the response time. The following section shows how to calculate measurement time for a single channel.

CSD	Frequency	Measurement Clock Frequency (MHz)	Filterlevel	Measurement Time of Single Channel (μs)
0	15	1	16	720
4	0	0.5	8	864
4	15	0.5	8	1104
8	15	0.5	8	1488

Example 1:

- CSD: 0
- Frequency: 15
- Measurement Clock Frequency: 0.5 MHz
- Filterlevel: 16
- Measurement Time of single channel = (30 + 6 × 0 + 15) x (1 / 0.5) × 16 μs = 1440 μs

Example 2:

Note: The following values have been used to calculate the Measurement Time of a single sensor.

- CSD: 0

- Frequency: 15
- Measurement Clock Frequency: 1 MHz
- Filterlevel: 16
- Measurement Time of single channel = (30 + 6 × 0 + 15) × (1 / 1) × 16 µs = 720 µs

5.2.2 Example to Calculate Response Time

The response time is split into three values in the following tables: Consider "Min." as the best-case response time and "Max." as the worst-case response time.

Reburst Unresolved

Table 5-4. Frequency Hop Enabled

Measurement Period (ms)	Measurement Time of Single Channel (ms)	DI	Number of Active Sensors	Number of Sensors Touched	Response Time (ms)		
					Min.	Avg.	Max.
50	0.72	2	12	1	60	85	110
100	0.72	2	12	2	112	162	212
200	0.72	2	12	3	213	313	413
250	0.72	2	12	4	264	389	514

Table 5-5. Frequency Hop Disabled

Measurement Period (ms)	Measurement Time of Single Channel (ms)	DI	Number of Active Sensors	Number of Sensors Touched	Response Time (ms)		
					Min.	Avg.	Max.
50	0.72	2	12	1	10	35	60
100	0.72	2	12	2	12	62	112
200	0.72	2	12	3	13	113	213
250	0.72	2	12	4	14	139	264

Reburst All

Table 5-6. Frequency Hop Enabled

Measurement Period (ms)	Measurement Time of Single Channel (ms)	DI	Number of Active Sensors	Number of Sensors Touched	Response Time (ms)		
					Min.	Avg.	Max.
50	0.72	2	12	1	76	101	126
100	0.72	2	12	2	126	176	226
200	0.72	2	12	3	226	326	426
250	0.72	2	12	4	276	401	526

Table 5-7. Frequency Hop Disabled

Measurement Period (ms)	Measurement Time of Single Channel (ms)	DI	Number of Active Sensors	Number of Sensors Touched	Response Time (ms)		
					Min.	Avg.	Max.
50	0.72	2	12	1	26	51	76
100	0.72	2	12	2	26	76	126
200	0.72	2	12	3	26	126	226
250	0.72	2	12	4	26	151	276

Reburst None

Table 5-8. Frequency Hop Enabled

Measurement Period (ms)	Measurement Time of Single Channel (ms)	DI	Number of Active Sensors	Number of Sensors Touched	Response Time (ms)		
					Min.	Avg.	Max.
50	0.72	2	12	1	159	184	209

...........continued

Measurement Period (ms)	Measurement Time of Single Channel (ms)	DI	Number of Active Sensors	Number of Sensors Touched	Response Time (ms)		
					Min.	Avg.	Max.
100	0.72	2	12	2	309	359	409
200	0.72	2	12	3	609	709	809
250	0.72	2	12	4	759	884	1009

Table 5-9. Frequency Hop Disabled

Measurement Period (ms)	Measurement Time of Single Channel (ms)	DI	Number of Active Sensors	Number of Sensors Touched	Response Time (ms)		
					Min.	Avg.	Max.
50	0.72	2	12	1	109	134	159
100	0.72	2	12	2	209	259	309
200	0.72	2	12	3	409	509	609
250	0.72	2	12	4	509	634	759

6. Power Consumption

Low Power

The following tables show the power consumption of 12 buttons, single button, and lump while all other buttons are suspended for 3.3V and 5V V_{DD}.

Twelve Buttons

Voltage	Measurement Period	I_{DD} (µA)			
		Oversample			
		x8	x16	x32	x64
3.3V	100	93	187	329	440
	200	58	96	122	308
	250	38	59	98	178
5V	100	151	227	378	680
	200	77	116	192	345
	250	62	93	153	274

Single Button

Voltage	Measurement Period	I_{DD} (µA)			
		Oversample			
		x8	x16	x32	x64
3.3V	100	14.5	21	26.7	45.3
	200	8.1	10	14.4	23.6
	250	6.4	8.8	11.8	19.2
5V	100	21.7	29.5	44.4	74.3
	200	12.5	18.8	29	36.6
	250	10.4	15.4	23.1	29.8

Lump Without Drift

Table 6-1. Button 5 and Button 6 Are Part of Lump 0, Which Is Configured as Low-Power Button

Voltage	Measurement Period	I_{DD} (µA)			
		Oversample			
		DEVCTRL.DLPLB = 0			
		x8	x16	x32	x64
3.3V	100	14.5	21	26.7	45.3
	200	8.1	10	14.4	23.6
	250	6.4	8.8	11.8	19.2
5V	100	21.4	28.6	42.3	69.5
	200	12.3	18.8	29	36.6
	250	10.4	15.4	23.1	29.8

Lump With Drift

Table 6-2. Button 5 and Button 6 Are Part of Lump 0, Which Is Configured as Low-Power Button

Voltage	Measurement Period	I_{DD} (µA)			
		Oversample			
		DEVCTRL.DLPLB = 1			
		x8	x16	x32	x64
3.3V	100	19	25.7	38.7	65.1
	200	10.3	13.5	19.9	33
	250	8.3	11.2	16.6	27.4
5V	100	28.8	37.9	55.3	91.3
	200	16.7	21.6	29.3	45.5
	250	13.3	16.7	24.1	38.7

7. Specification

7.1 Disclaimer

Unless otherwise specified, all typical values are measured at T = 25°C and V_{DD} = 3V, and all minimum and maximum values are valid across operating temperature and voltage.

7.2 Electrical Specification

Stresses beyond those listed in this section may cause permanent damage to the device, which is a stress rating only, and functional operation of the device at these or other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect the device’s reliability.

Absolute Maximum Ratings

Symbol	Description	Min.	Max.	Unit
V_{DD}	Power supply voltage	3	5	V
V_{RST}	$\overline{\text{RESET}}$pin voltage with respect to GND	V_{DD}		V
V_{PIN}	Pin voltage with respect to GND			
I_{PIN}	I/O pin sink/source current	-40	40	mA
$T_{storage}$	Storage temperature	-65	150	°C

7.3 General Operating Ratings

The device must operate within the ratings listed in this section for all other electrical and typical characteristics to be valid.

General Operating Conditions

Symbol	Description	Condition	Min.	Max.	Unit
V_{DD}	Operation supply voltage	-	3	5	V
SRON	Power-on Slope	-	-	100	V/ms
T	Operating Temperature Range	Standard temperature range	-40	105	°C

Note: Refer to the device ordering codes for the device temperature range.

7.4 I/O Pin Characteristics

I/O Pin Characteristics (TA = [-40, 105]°C, V_{DD} = [1.8, 5.5]V Unless Otherwise Stated)

Symbol	Description	Condition	Min.	Typ.	Max.	Unit
V_{IL}	Input low-voltage, except RESET pin as I/O		-0.2	-	0.3 × V_{DD}	V
V_{IH}	Input high-voltage, except RESET pin as I/O		0.7 × V_{DD}	-	V_{DD} + 0.2V	V
I_{IH}/I_{IL}	I/O pin input leakage current, except RESET pin as I/O	V_{DD} = 5V, pin high	-	< 0.05	-	µA
		VV_{DD} = 5V, pin low	-	< 0.05	-	µA
V_{OL}	I/O pin drive strength	VV_{DD} = 3.0V, I_{OL} = 7.5 mA	-	-	0.6	V
		V_{DD} = 5.0V, I_{OL} = 15 mA	-	-	1	V
V_{OH}	I/O pin drive strength	V_{DD} = 3.0V, I_{OH} = 7.5 mA	2.4	-	-	V
		V_{DD} = 5.0V, I_{OH} = 15 mA	4	-	-	V

7.5 Timing Details

The following operations require the typical time needed to complete their execution.

Operation	Typical Time Required to Complete the Execution
Initialization	250 ms

...........continued

Operation	Typical Time Required to Complete the Execution
Save	1.3 sec
Software Reset	250 ms
Reset to Default Configuration.	1.4 sec

7.6 Maximum Allowed Sensor Capacitance

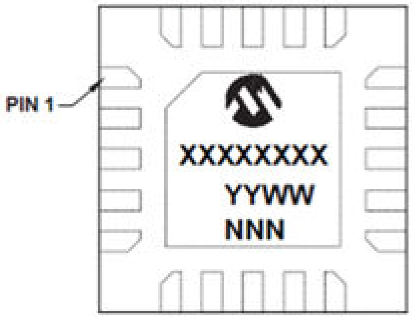
Button	Capacitance
Button 0	4 pF
Button 1	24 pF
Button 2	9 pF
Button 3	6 pF
Button 4	4 pF
Button 5	4 pF
Button 6	13 pF
Button 7	13 pF
Button 8	6 pF
Button 9	6 pF
Button 10	6 pF
Button 11	6 pF

8. Ordering Information

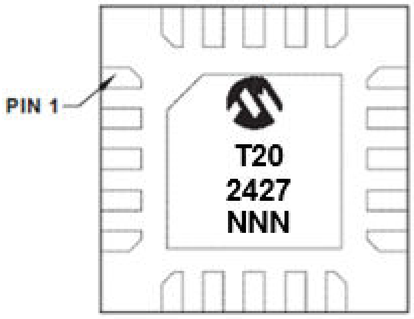
- Available ordering options can be found by:
  - Clicking on one of the following product page links:
    - MTCH2120 Product Page
    - MTCH2120 Evaluation Kit
- Searching by product name at microchipdirect.com
- Contacting the local sales representative

9. Package Marking

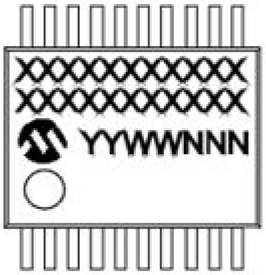
20-Lead VQFN



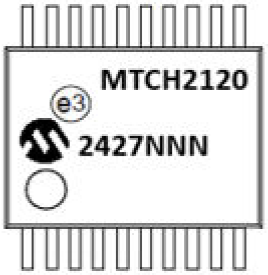
Example



20-Lead SOIC



Example



Legend:	XX...X	Customer-specific information or Microchip part number
	Y	Year code (last digit of calendar year)
	YY	Year code (last 2 digits of calendar year)
	WW	Week code (week of January 1 is week '01')
	NNN	Alphanumeric traceability code
	(e3)	b- free JEDEC ® designator for Matte Tin (Sn)
	*	This package is Pb-free. The Pb-free JEDEC designator ((e3)) can be found on the outer packaging for this package.

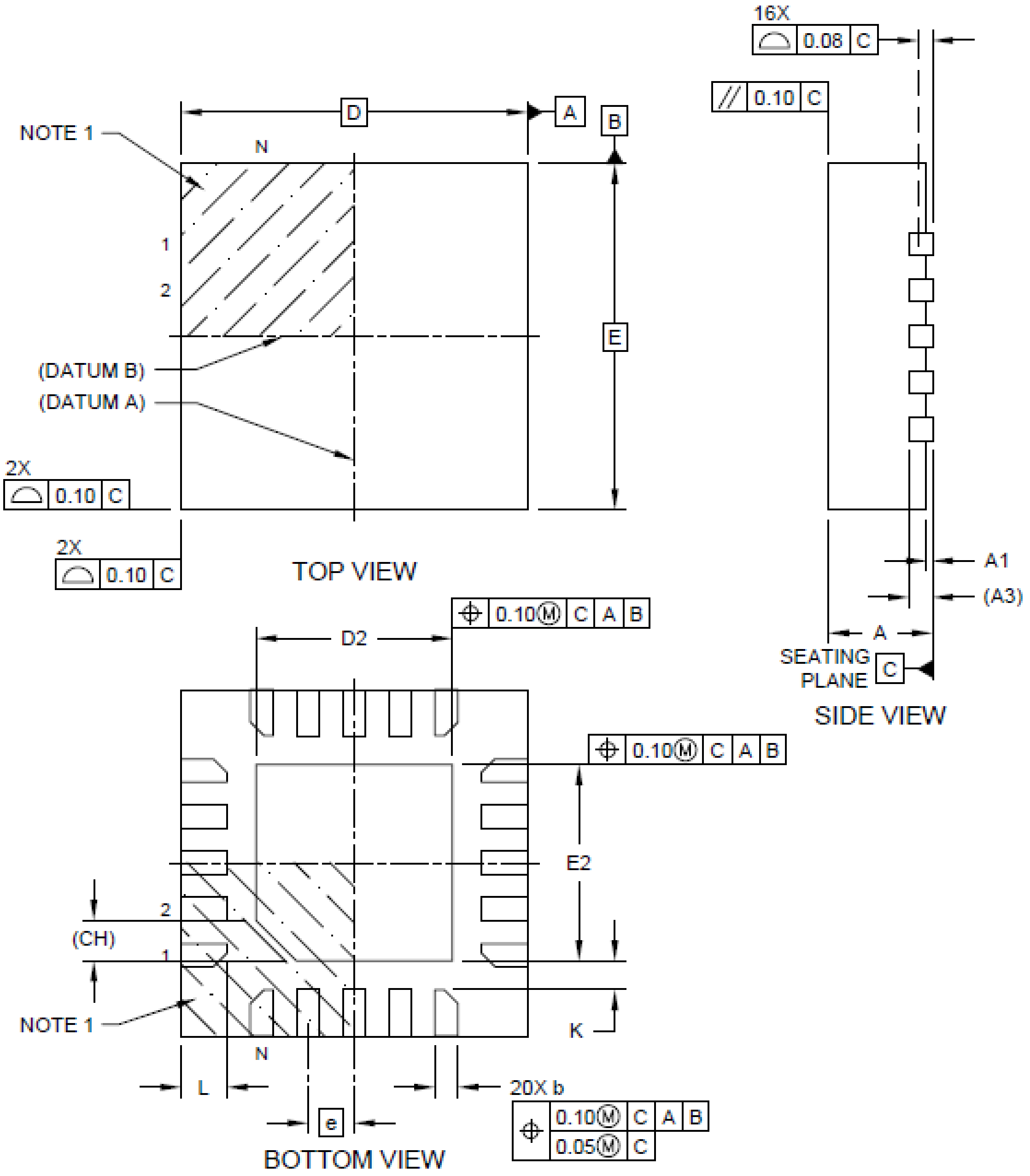
Note: In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line, thus limiting the number of available characters for customer-specific information.

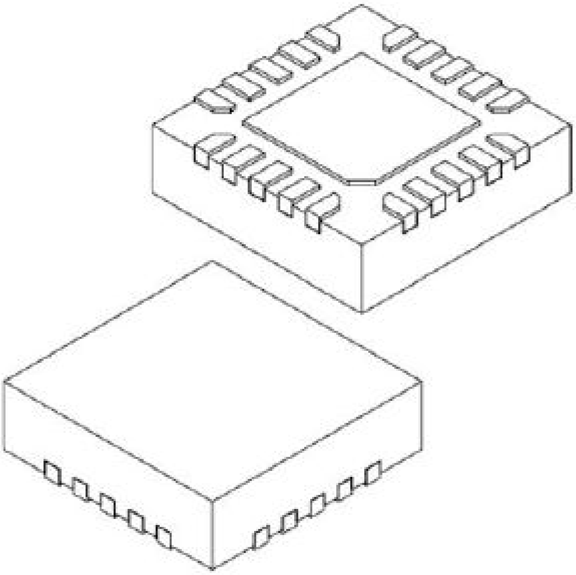
10. Package Drawing

20-Pin VQFN

20-Lead Very Thin Plastic Quad Flat, No Lead Package (REB) - 3x3 mm Body [VQFN] with 1.7 mm Exposed Pad

Note: For the most current package drawings, see the Microchip Packaging Specification located at www.microchip.com/packaging.

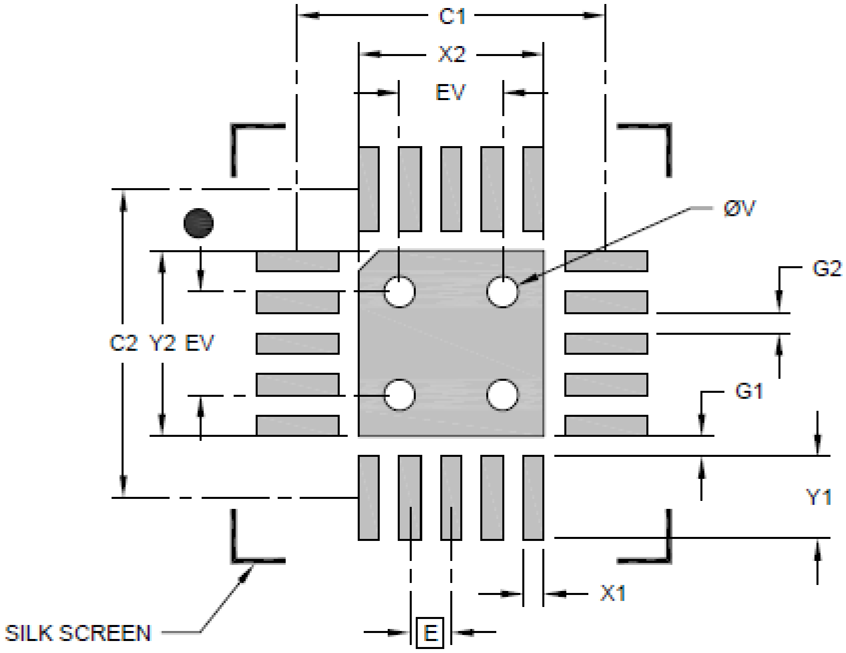




Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Terminals	N	20		
Pitch	e	0.40 BSC		
Overall Height	A	0.80	0.85	0.90
Standoff	A1	0.00	0.035	0.05
Terminal Thickness	A3	0.203 REF		
Overall Length	D	3.00 BSC		
Exposed Pad Length	D2	1.60	1.70	1.80
Overall Width	E	3.00 BSC		
Exposed Pad Width	E2	1.60	1.70	1.80
Terminal Width	b	0.15	0.20	0.25
Terminal Length	L	0.35	0.40	0.45
Terminal-to-Exposed-Pad	K	0.20	-	-
Pin 1 Index Chamfer	CH	0.35 REF		

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Package is saw singulated.
3. Dimensioning and tolerancing per ASME Y14.5M:
   BSC: Basic Dimension. Theoretically, the exact value is shown without tolerances.
   REF: Reference Dimension, usually without tolerance, for information purposes only.



RECOMMENDED LAND PATTERN

	Units	MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E	0.40 BSC		
Optional Center Pad Width	X2			1.80
Optional Center Pad Length	Y2			1.80
Contact Pad Spacing	C1		3.00	
Contact Pad Spacing	C2		3.00	
Contact Pad Width (X20)	X1			0.20
Contact Pad Length (X20)	Y1			0.80
Contact Pad to Center Pad (X20)	G1	0.20		
Contact Pad to Contact Pad (X16)	G2	0.20		
Thermal Via Diameter	V		0.30	
Thermal Via Pitch	EV		1.00	

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M:
   BSC: Basic Dimension. Theoretically, the exact value is shown without tolerances.
2. For best soldering results, thermal vias, if used, should be filled or tented to avoid solder loss during the reflow process.

Table 10-1. Device and Package Maximum Weight

Device and Package Maximum Weight	19.1 mg

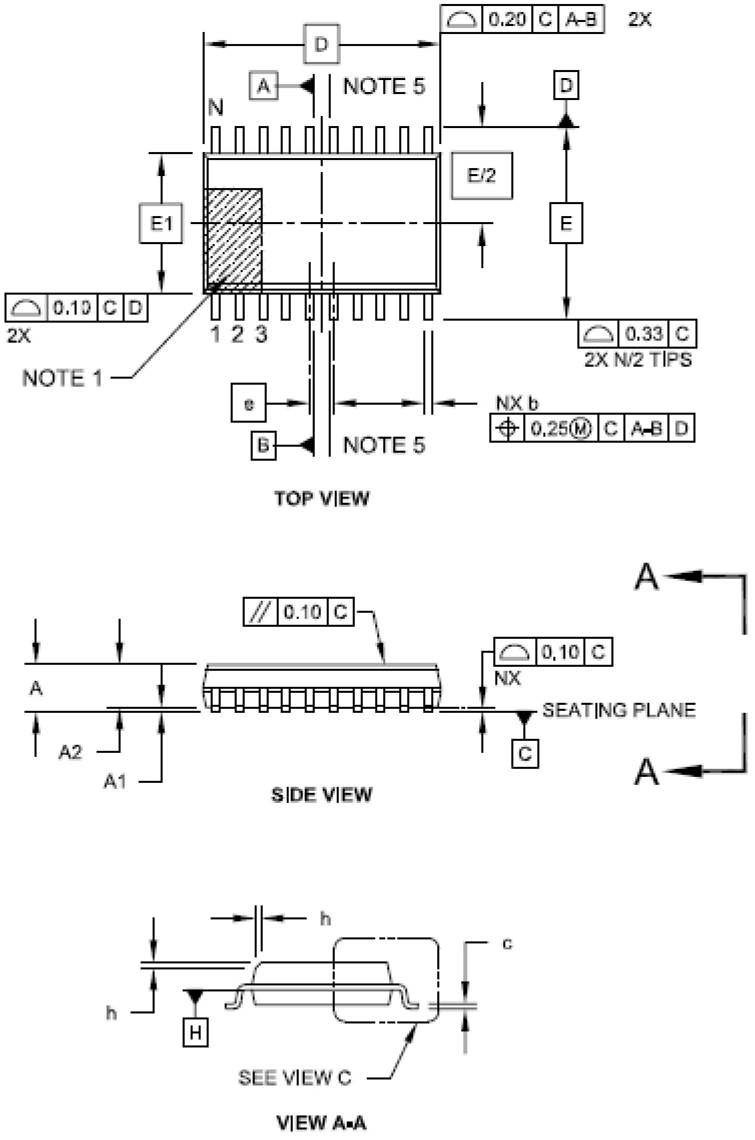
Table 10-2. Package Reference

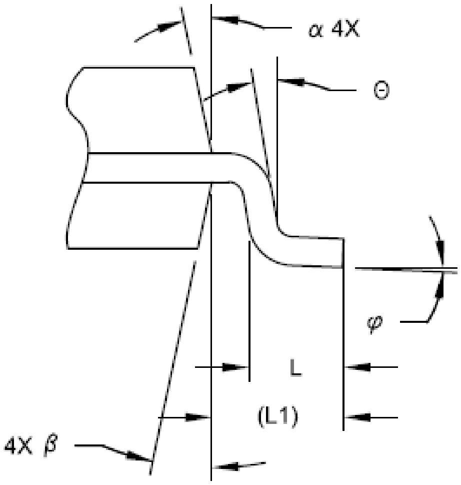
JEDEC Drawing Reference	N/A
JESD97 Classification	E3

20-Pin SOIC

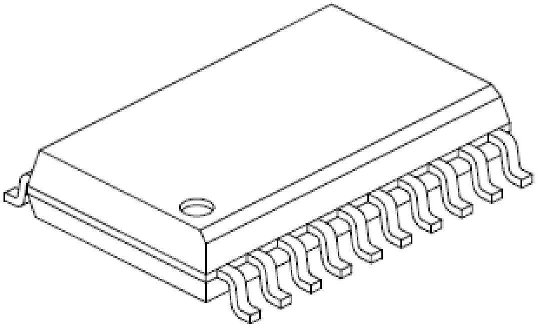
20-Lead Plastic Small Outline (SO) - Wide, 70.50 mm Body [SOIC]

Note: For the most current package drawings, see the Microchip Packaging Specification located at www.microchip.com/packaging.





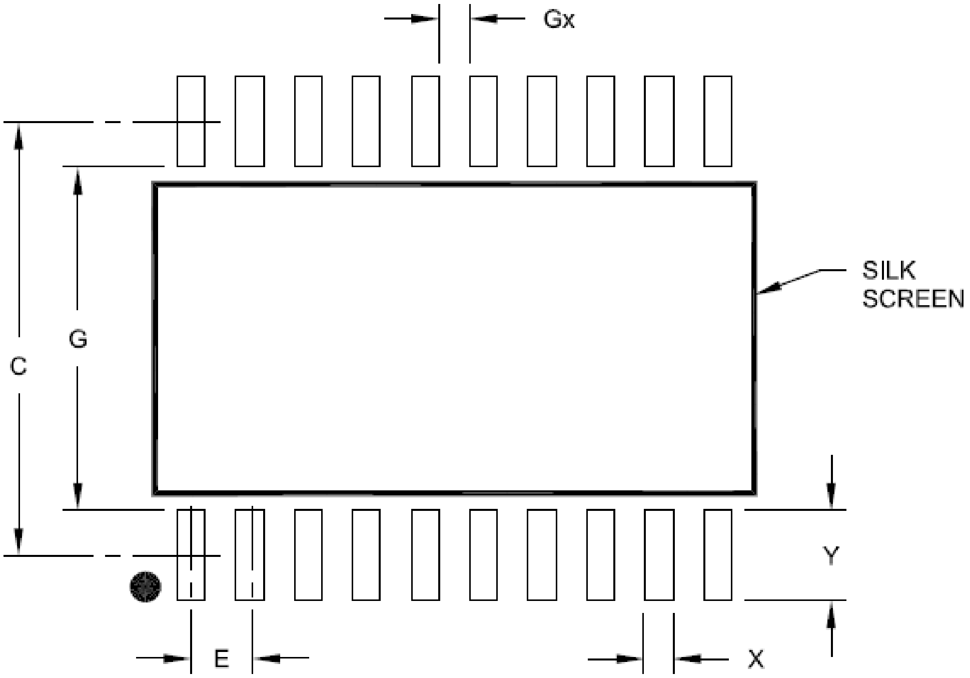
VIEW C



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N	20		
Pitch	e	1.27 BSC		
Overall Height	A	-	-	2.65
Molded Package Thickness	A2	2.05	-	-
Standoff §	A1	0.10	-	0.30
Overall Width	E	10.30 BSC		
Molded Package Width	E1	7.50 BSC		
Overall Length	D	12.80 BSC		
Chamfer (Optional)	h	0.25	-	0.75
Foot Length	L	0.40	-	1.27
Footprint	L1	1.40 REF		
Lead Angle	Θ	0°	-	-
Foot Angle	φ	0°	-	8°
Lead Thickness	c	0.20	-	0.33
Lead Width	b	0.31	-	0.51
Mold Draft Angle Top	α	5°	-	15°
Mold Draft Angle Bottom	β	5°	-	15°

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. § Significant Characteristic
3. Dimension D does not include mold flash, protrusions or gate burrs, which shall not exceed 0.15 mm per end. Dimension E1 does not include interlead flash or protrusion, which shall not exceed 0.25 mm per side.
4. Dimensioning and tolerancing per ASME Y14.5M
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.
   REF: Reference Dimension, usually without tolerance, for information purposes only.
5. Datums A & B to be determined at Datum H.



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E	1.27 BSC		
Contact Pad Spacing	C		9.40	
Contact Pad Width (X20)	X			0.60
Contact Pad Length (X20)	Y			1.95
Distance Between Pads	Gx	0.67		
Distance Between Pads	G	7.45		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Table 10-3. Device and Package Maximum Weight

Maximum Weight	542 mg

Table 10-4. Package Reference

JEDEC Drawing Reference	N/A
JESD97 Classification	E3

11. Revision History

Revision	Date	Description
B	11/2024	Title update: MTCH2120 - Twelve-Channel Capacitive Touch Controller Memory Block: Status list update (rearranged the list order)
A	10/2024	Initial document release

Microchip Information

Trademarks

The “Microchip” name and logo, the “M” logo, and other names, logos, and brands are registered and unregistered trademarks of Microchip Technology Incorporated or its affiliates and/or subsidiaries in the United States and/or other countries (“Microchip Trademarks”). Information regarding Microchip Trademarks can be found at https://www.microchip.com/en-us/about/legal-information/microchip-trademarks.

ISBN: 978-1-6683-0519-5

Legal Notice

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 www.microchip.com/en-us/support/design-help/client-support-services.

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.

Microchip Devices Code Protection Feature

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 product 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.