MICROCHIP

MTCH6303

MTCH6303 Projected Capacitive Touch Controller Data Sheet

Description

Microchip's MTCH6303 is an innovative turnkey projected capacitive touch controller that provides multi-touch coordinates as well as a readymade multi-finger surface gesture suite. MTCH6303 brings modern user interface (UI) elements – such as pinch and zoom, multi-finger scrolling, and swipes – to any embedded design, with minimal host requirements.

The MTCH6303's advanced signal processing provides noise-avoidance techniques and predictive tracking for ten fingers, typically at 100 Hz each for five touches. It also combines with Microchip's MTCH652 High-voltage Line Driver to achieve a superior signal-to-noise ratio (SNR) for outstanding touch performance in noisy environments (refer to www.microchip.com/MTCH652). These capabilities are critical in demanding environments such as industrial controls, home and office automation with security control panels, thermostat, printers and lighting controls, and various consumer applications including exercise equipment and audio systems.

Features

- Multi-Touch up to Ten Touches
- Five Touches Typically at 100 Hz+ Each
- 27RX x 19TX Channels Support Approximately 8" Touch Screens (larger possible)
- Combines with MTCH652 High-Voltage Driver for Superior Signal-to-Noise Ratio (SNR)
- Integrated Single and Multi-finger Gesture Recognition Suite including Taps, Swipes, Scrolling, Pinching and Zooming
- Advanced Processing Provides Noise Avoidance Techniques
- USB and I^{2}C™ Communication
- Supports 3D Gestures up to 20 cm when Combined with the MGC3130 GestIC® Controller

Power Management

Example:

- 27RX 19TX Sensor
  - 27 mA full-scan rate
  - 1 mA reduced-scan rate

Applications

- Touch screen designs and touch pads that require cost effective, easy to integrate, fast time to market PCAP touch solutions
- Perfect for touch screens over displays, control panels, keypads and many other input devices
- Targeting the industrial, medical, home and office automation, and consumer markets

TABLE 1: MTCH6303 SOLUTION PART NUMBERS

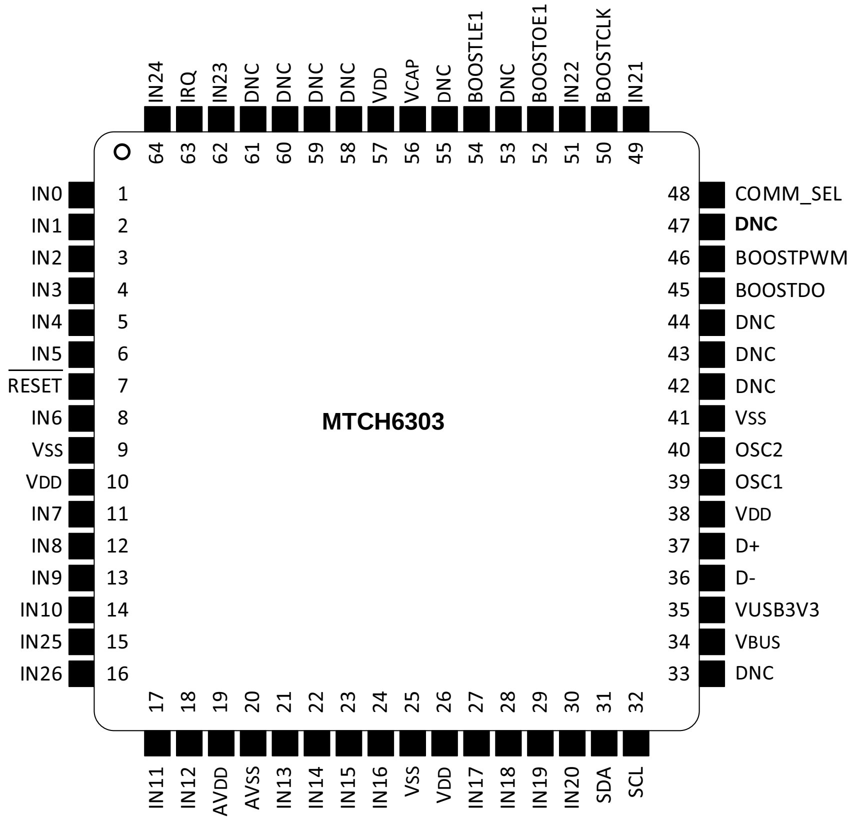
Device	Pin Count	Package Types	Touch Channels	Features
MTCH6303-I/PT	64	10 x10 mm TQFP	Up to 27 RX	Multi-touch, up to 8" sensors
MTCH6303-I/RG		9 x 9 mm QFN		
*MTCH652-I/SO	28	7.5 mm SOIC	Up to 19 TX	1.8 – 5.5V input, 6V – 18V configurable output
*MTCH652-I/SS		5.3 mm SSOP		
*MTCH652-I/MV		4 x 4 mm UQFN		

Note: *One MTCH652 high-voltage driver (boost) is required with MTCH6303.

Note: The MTCH6303 devices are pre-programmed with a Library Loader (bootloader) only. Refer to **Section 8.0, Firmware update** for more details.

PIN DIAGRAM

FIGURE 1: MTCH6303 64-PIN DIAGRAM TQFP/QFN



PIN ALLOCATION TABLE

TABLE 2: MTCH6303 PINOUT DESCRIPTION

Name	Pin	Description
IN0	1	IN 0 – 5
IN1	2	
IN2	3	
IN3	4	
IN4	5	
IN5	6	
$\overline{\text{RESET}}$	7	Reset
IN6	8	IN 6
V_{SS}	9	Ground
V_{DD}	10	Power Supply Input
IN7	11	IN 7 – 10
IN8	12	
IN9	13	
IN10	14	
IN25	15	IN 25 – 26
IN26	16	
IN11	17	IN 11 – 12
IN12	18	
AV_{DD}	19	Positive supply for analog modules. This pin must be connected at all times.
AV_{SS}	20	Ground reference for analog modules
IN13	21	IN 13 – 16
IN14	22	
IN15	23	
IN16	24	
V_{SS}	25	Ground
V_{DD}	26	Power Supply Input
IN17	27	IN 17 – 20
IN18	28	
IN19	29	
IN20	30	
SDA	31	I^2C™ Data
SCL	32	I^2C Clock

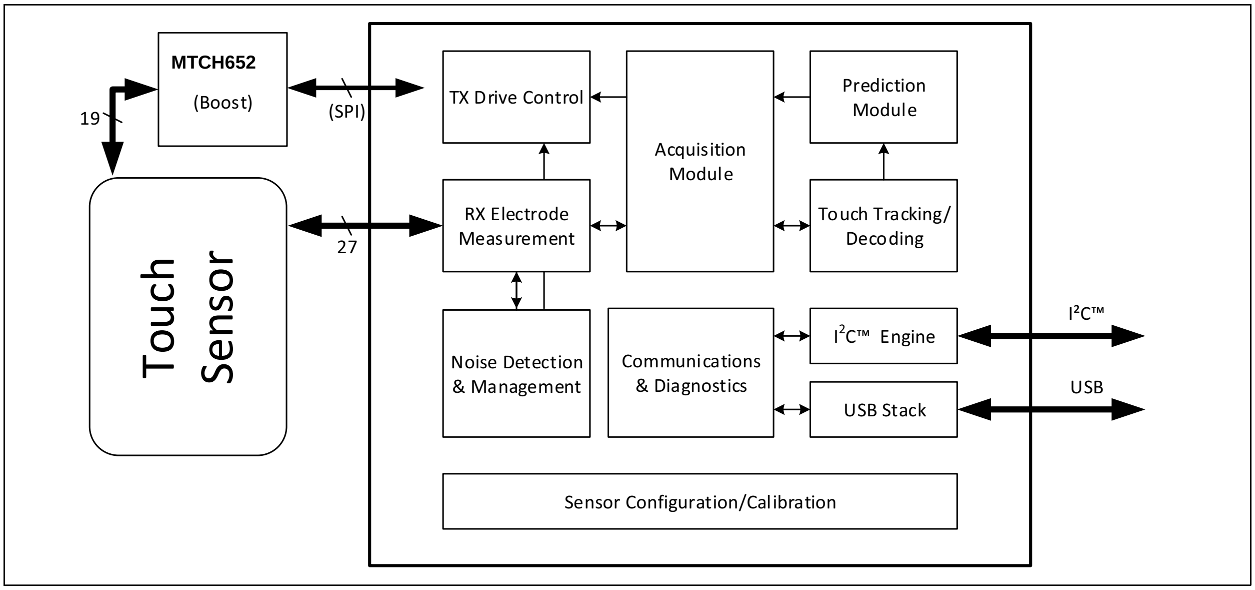
TABLE 2: MTCH6303 PINOUT DESCRIPTION (CONTINUED)

Name	Pin	Description
DNC	33	Do not connect any signal to these pins.
	42	
	43	
	44	
	47	
	53	
	55	
	58	
	59	
	60	
	61	
V_{BUS}	34	USB Bus Power Monitor
VUSB3V3	35	USB internal transceiver supply. If the USB module is not used, this pin must be connected to V_{DD}.
D-	36	USB D-
D+	37	USB D+
V_{DD}	38	Power Supply Input
OSC1	39	Oscillator Pin 1
OSC2	40	Oscillator Pin 2
V_{SS}	41	Ground
BOOSTDO	45	MTCH652 DO output/DIN Input
BOOSTPWM	46	MTCH652 PWM Out/OSCIN input
COMM_SEL	48	Communication Select Pin (V_{DD} = I^2C™, V_{SS} = USB)
IN21	49	IN 21
BOOSTCLK	50	MTCH652 CLK Output
IN22	51	IN 22
BOOSTOE1	52	MTCH652 OE Output 1
BOOSTLE1	54	MTCH652 LE Output 1
V_{CAP}	56	Capacitor for Internal Voltage Regulator
V_{DD}	57	Power Supply Input
IN23	62	IN 23
IRQ	63	I^2C Interrupt
IN24	64	IN 24
MGC_TS	42	Gesture Transfer Status
MGC_SDA	43	Gesture I^2C Data
MGC_SCL	44	Gesture I^2C Clock
MGC_MCLR	61	Gesture Reset
MGC_MODE	60	Gesture Mode Control
MGC_SYNC	47	Gesture Sync

Table of Contents

1.0 DEVICE OVERVIEW

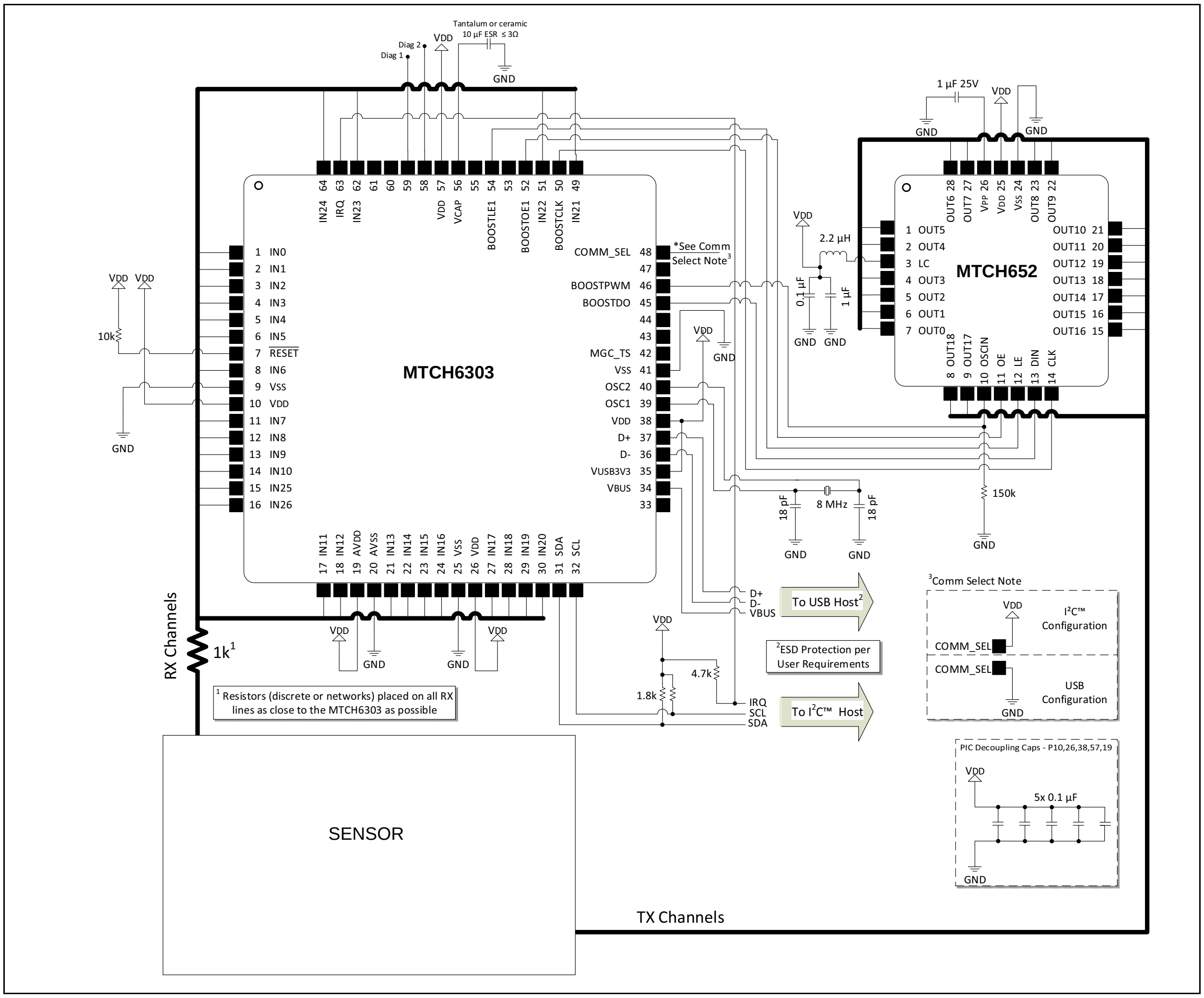
FIGURE 1-1: MTCH6303 BLOCK DIAGRAM



 Preliminary

2.0 LAYOUT

FIGURE 2-1: TYPICAL APPLICATION CIRCUIT

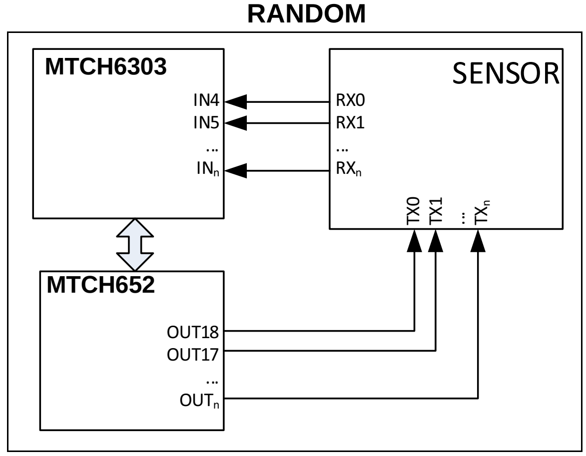


2.1 SENSOR CHANNEL NAMING CONVENTION

Throughout this document, there are references to signals such as IN, RX, OUT and TX. This is deliberately done to avoid confusion between sensor channels and physical pins on the controller. Refer to Figure 2-2 for an example of channel numbers chosen randomly.

- When referring to a sensor, the channels are labeled RX0-RXn and TX0-TXn.
- When referring to the MTCH6303 controller, the INn pins connect to any RXn on the sensor.
- When referring to the MTCH652 boost converter, the OUTn pins connect to any TXn on the sensor.

FIGURE 2-2: EXAMPLE OF CHANNEL NUMBERS CHOSEN AT RANDOM



2.2 Decoupling Capacitors

The use of decoupling capacitors on power supply pins, such as V_{DD}, V_{SS}, is required. Consider the following criteria when using decoupling capacitors.

2.2.1 VALUE AND TYPE OF CAPACITOR

A value of 0.1 µF (100 nF), 10-20V is recommended. The capacitor should be a low Equivalent Series Resistance (low ESR) capacitor and have resonance frequency in the range of 20 MHz and higher. It is further recommended that ceramic capacitors be used.

2.2.2 PLACEMENT ON THE PRINTED CIRCUIT BOARD

The decoupling capacitors should be placed as close to the pins as possible. It is recommended that the capacitors be placed on the same side of the board as the device. If space is restricted, the capacitor can be placed on another layer on the PCB; however, ensure that the trace length from the pin to the capacitor is within one-quarter of an inch (6 mm) in length.

2.2.3 HANDLING HIGH-FREQUENCY NOISE

If the board is experiencing high-frequency noise, upward of tens of MHz, add a second ceramic-type capacitor in parallel to the above described decoupling capacitor. The value of the second capacitor can be in the range of 0.01 µF to 0.001 µF. Place this second capacitor next to the primary decoupling capacitor. In high-speed circuit designs, consider implementing a decade pair of capacitances as close to the power and ground pins as possible. For example, 0.1 µF in parallel with 0.001 µF.

2.2.4 MAXIMIZING PERFORMANCE

On the board layout from the power supply circuit, run the power and return traces to the decoupling capacitors first, and then to the device pins. This ensures that the decoupling capacitors are first in the power chain. Equally important is to keep the trace length between the capacitor and the power pins to a minimum, thereby reducing PCB track inductance.

2.3 Bulk Capacitors

The use of a bulk capacitor is recommended to improve power supply stability. Typical values range from 4.7 µF to 47 µF. This capacitor should be located as close to the device as possible.

3.0 COMMUNICATION

3.1 USB/I²C™ Selection

The MTCH6303 can communicate over either USB or I²C™. The decision of which protocol is selected is made on start-up and persists until the controller is reset.

Communications are selectable between USB/I²C through the use of the COMM_SEL pin, which must be permanently tied to either VSS or VDD as follows:

TABLE 3-1: COMM_SEL SETTINGS

Setting	Communications Type
VDD	I²C™
VSS	USB

3.2 Communications Overview

Communications with the MTCH6303 fall into two main categories:

1. **Touch Data:** Data representing the current state of any contact points; this is the main function of the touch controller.
2. **Streamed Messaging:** Packet-based messaging protocol used to:

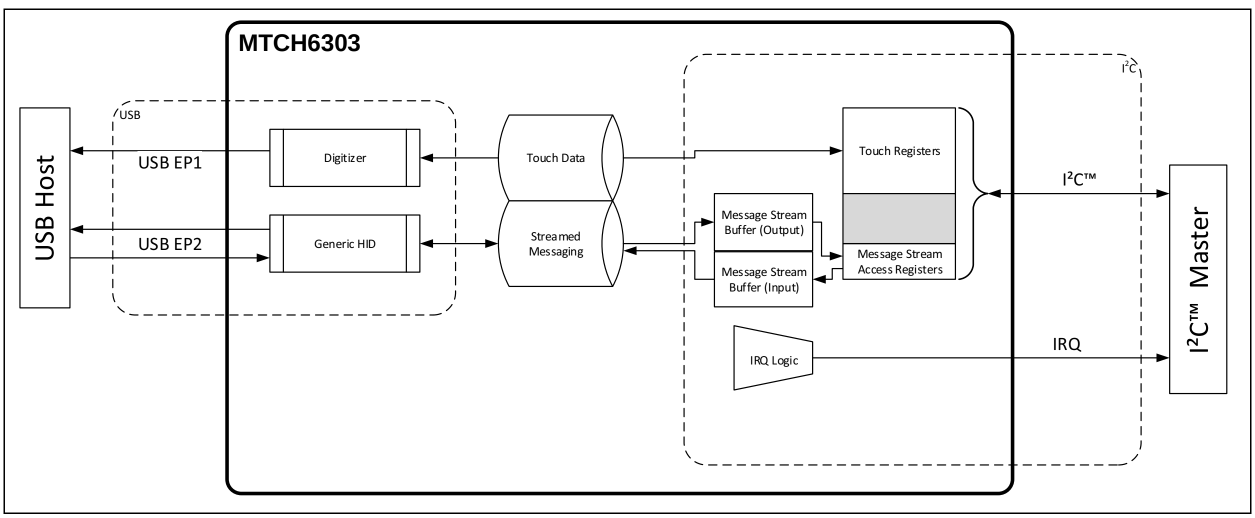
- Send controller commands
- Read/Write parameters
- Receive diagnostic reports (when enabled)
- Read 2D gesture data
- Read 3D gesture data (requires MGC3130)

Both types of data are available over either USB or I²C, as shown in the Table 3-2 below.

TABLE 3-2: COMMUNICATIONS CATEGORIES

Data Type	USB	I²C™
Touch Data	Digitizer endpoint	Register-based memory map
Streamed Messaging	Generic HID endpoint	Stream buffers accessed via I²C™ registers

FIGURE 3-1: COMMUNICATIONS OVERVIEW DIAGRAM



3.3 USB Protocol

3.3.1 HID DIGITIZER (EP 1, TOUCH DATA)

TABLE 3-3: HID DIGITIZER

Byte	7	6	5	4	3	2	1	0	
0	REPORT ID (0X01)								TOUCH 1
1	PADDING						IR	TS	
2	TOUCH ID 0								
3	X1 LSB								
4	X1 MSB								
5	Y1 LSB								
6	Y1 MSB								
7	PADDING						IR	TS	TOUCH 2
8	TOUCH ID 1								
9	X2 LSB								
10	X2 MSB								
11	Y2 LSB								
12	Y2 MSB								
..	..						..	..	TOUCHES 3-9
..	..								
..	..								
..	..								
..	..								
..	..								
47	PADDING						IR	TS	TOUCH 10
48	TOUCH ID 9								
49	X4 LSB								
50	X4 MSB								
51	Y4 LSB								
52	Y4 MSB								
53	#OF VALID TOUCHES								

Legend: IR = In Range
TS = Touch State

3.3.2 HID GENERIC (EP 2, STREAMED MESSAGES)

This generic endpoint is used to send and receive one or more messages within the payload.

FIGURE 3-2: HID GENERIC

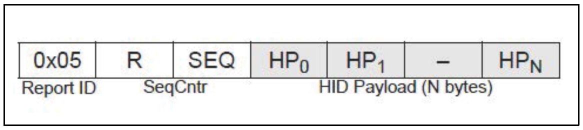


TABLE 3-4: HID GENERIC

Byte Name		Value/Description
Report ID	0x05	0x05 (Constant)
SeqCntr [7:6]	R	[reserved]
SeqCntr [5:0]	SEQ	Sequence counter, increments on every HID packet. • Values range from 0-63 • IN and OUT packets utilize independent sequence counters

3.4 I²C™ PROTOCOL

3.4.1 OVERVIEW

The MTCH6303 uses a standard register-based read/write I²C™ protocol. This protocol is similar to many other devices such as temperature sensors and serial EEPROMs. Although data can be read at any time (polling), a configurable interrupt pin (INT) is provided for flexible integration options.

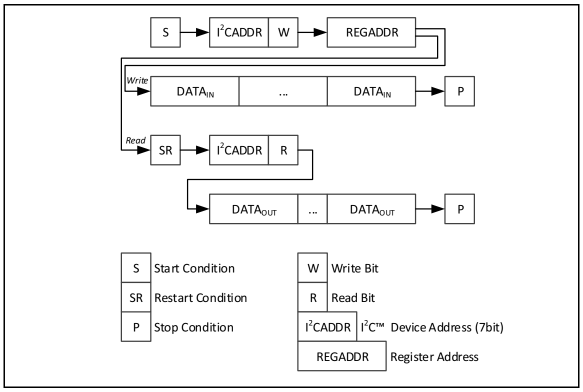
3.4.2 READING/WRITING REGISTERS

To access memory (both to read or write), the I²C transaction must start by addressing the chip with the WRITE bit set, then writing out a single byte of data representing the memory address to be operated on. After that, the host can choose to do either of the following:

1. To write memory, continue writing "n" data bytes.
2. To read memory, restart the I²C transaction (via either a Stop and Start or Restart), then address the chip with the READ bit set. Continue to read "n" data bytes.

During either of these transactions, multiple bytes may be read or written due to the device's address auto-increment feature.

FIGURE 3-3: I²C™ TRANSACTION DIAGRAM

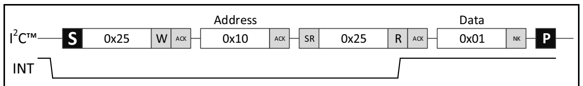


3.4.3 DEVICE ADDRESSING

The device's 7-bit base address is 0x25. Each transmission must be prefixed with this address, as well as a bit signifying whether the transmission is a MASTER WRITE (0) or MASTER READ (1). After appending this read/write bit to the base address, this first byte becomes either 0x4A (write) or 0x4B (read).

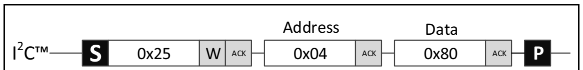
Note:	If this address conflicts with another in the system, it may be possible to customize the device. Contact Microchip support for more information.

FIGURE 3-4: EXAMPLE I^{2}C™ READ TRANSACTION



Note: Reading one byte from address 0x10, byte value is 0x01.

FIGURE 3-5: EXAMPLE I^{2}C™ WRITE TRANSACTION



Note: Writing one byte to address 0x04, value 0x80.

TABLE 3-5: I²C™ MEMORY MAP

ADDR	NAME	7	6	5	4	3	2	1	0	Description
		TOUCH								
0x00	TOUCHSTATUS	R	MGC	GST	STR	NUMTOUCHES				MGC = GestIC® data, GST = Gestures Ready, STR = Stream Ready
0x01	TOUCH 0							IR	TS	IR = In Range, TS = Touch State
0x02		TOUCH ID 1								ID = touch ID, 0-16
0x03		X1 LSB								
0x04		X1 MSB								
0x05		Y1 LSB								
0x06		Y1 MSB								
0x07	TOUCH 1							IR	TS	
0x08		TOUCH ID 1								
0x09		X1 LSB								
0x0A		X1 MSB								
0x0B		Y1 LSB								
0x0C		Y1 MSB								
0x0D	(TOUCH 2)	...								(format follows from above)
0x13	(TOUCH 3)	...								
0x19	(TOUCH 4)	...								
0x1F	(TOUCH 5)	...								
0x25	(TOUCH 6)	...								
0x2B	(TOUCH 7)	...								
0x31	(TOUCH 8)	...								
0x37	(TOUCH 9)	...								
0x42 — 0x7F		*[RESERVED]*								
		STREAM BUFFER								
0xF0 — 0xFA		*[RESERVED]*								
0xFB	RX Bytes Ready	RXRDY								Space available (bytes) for writing into RX buffer
0xFC	RX Buffer	RXBUFF								Pointer to RX Buffer
0xFD	TX Bytes Left	TXRDY								Bytes ready to be read from TX buffer
0xFE	TX Buffer	TXBUFF								Pointer to TX Buffer

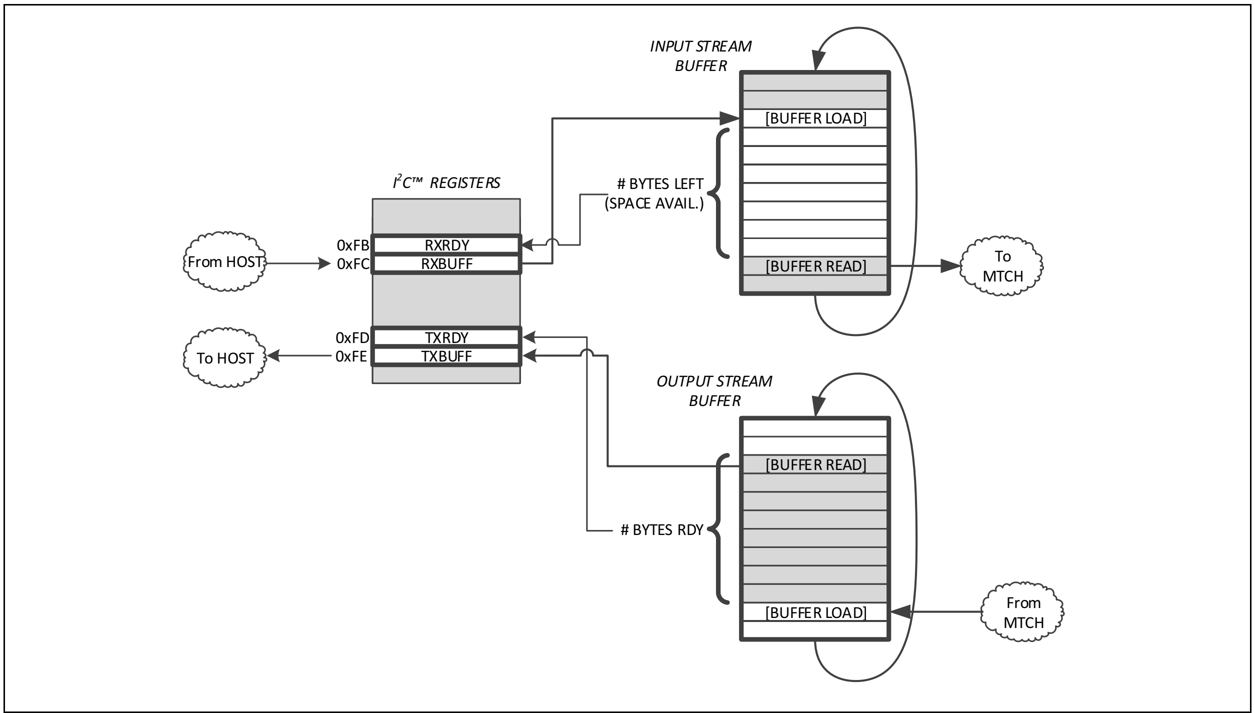
3.4.4 TOUCH REGISTERS

Touch data can be read out of the touch registers at any time, and is ensured to represent the latest state of the sensor. Use of the IRQ pin can improve efficiency by letting the host controller only read data when necessary. (See **Section 6.0, Communication Examples** for more details.)

3.4.5 MESSAGE STREAM ACCESS

For sending and receiving stream messages (described further on in this document), register-based access to the message stream is provided as shown in Figure 3-6.

FIGURE 3-6: MESSAGE STREAM ACCESS



3.4.5.1 Reading Stream Messages Over I^{2}C

The host discovers that data is ready to be read from the stream by reading a non-zero value from the TXRDY register. This register should be queried after one of the following events:

- IRQ activity
- STR bit of TOUCHSTATUS register is set
- Polled at a random interval (of the host's choosing)

To read the data, an I^{2}C register read should be started at the address of TXBUFF. The host can choose to read any amount of bytes (up to the value in TXRDY).

3.4.5.2 Writing Stream Messages Over I^{2}C

The host can write messages directly into the address of RXBUFF. Before writing, the host should check the amount of space available for writing by reading the RXRDY register.

3.4.5.3 Interrupt Pin

To alert the host that new data is ready, an interrupt pin (IRQ) is provided. The IRQ is an 'open-drain' output that is pulled to GND when asserted, and high-impedance (tri-state) when not asserted. A suitable pull-up resistor should be used on this output.

The IRQ can be configured using the parameters in Table 3-6 below (refer to **Section 5.0, Parameters** for accessing).

TABLE 3-6: IRQ CONFIGURATION PARAMETERS

Parameter	Default	Description
irqMode	1	Overall IRQ mode 0 = IRQ deactivated 1 = IRQ level maintained until data read 2 = IRQ pulsed for [irqPulseWidth] msec
irqPolarity	0	IRQ Polarity control 0 = Active-Low, 1 = Active-High
irqPulseWidth	5	Value (msec) to pulse IRQ when irqMode is set to ‘2’
irqTrigger	2	Event control for IRQ activity 0 = Off 1 = Every touch decoding frame 2 = Any touch is present 3 = Only when touch is changed

4.0 MESSAGE PROTOCOL

4.1 Overview

The MTCH6303 messaging protocol is used to send and receive streamed messages. Full or partial (fragment) messages may be exchanged with this protocol.

Messages are transmitted in an overall ‘block’ size of 64 and must be split up accordingly. Refer to **Section 6.0, Communication Examples** for depictions of messages being fragmented.

FIGURE 4-1: MESSAGE PROTOCOL

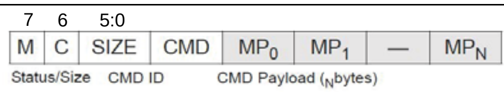


TABLE 4-1: MTCH6303 MESSAGE FORMAT

Name	Description
Status/ Size	**B5-0 SZ** Size of message fragment. If 63 (0x3f), the fragment is incomplete and uses up ALL of the parent transport layer packet
	B6 C `1` = Continued (from last fragment) `0` = Not continued (start of message)
	B7 M `1` = More messages to follow in this block `0` = Last message
CMD ID	Command ID, **only sent on first fragment of message.** For fragments after, this is a normal payload byte.
CMD Payload	Data bytes of message fragment.

4.2 Message Definitions

Messages starting with REP are reports sent from the MTCH6303 to the host. Messages starting with CMD are commands sent from the host to the MTCH6303. Messages that require further clarification are expanded upon in the following section.

TABLE 4-2: MESSAGE DEFINITIONS

ID	Name	Payload size	Payload Description (assume uint8 unless noted)	Gated by NVDM[1]	Description
0x04	REP_Echo	<varies>	[data]...[datan]	[NO GATE]	It will echo the exact payload of a received ‘echo’ command
0x17	REP_FlashContents	<varies>	[data]...[datan]	[NO GATE]	Flash contents readback (invoked by CMD_ReadFlash)
0x60	REP_AdcDbg	132	[rx] [tx] [freq] [RSVD] [*uint16* D0] [*uint16* D1]...[*uint16* Dn]	NVDM_ADC	Raw sample output from ADC
0x90	REP_Trace	2	[location][event]	NVDM_DIAG	—
0xA0	REP_Swipe	2	[flags][fingers]	NVDM_GESTURE	Swipe gesture
0xA1	REP_Scroll	8	[fingers][diamHI][*uint16* diameter][*uint16* centerx][*uint16* centery]	NVDM_GESTURE	Scroll gesture
0xA2	REP_Tap	2	[flags][fingers]	NVDM_GESTURE	Tap gesture
0xB0	REP_Noise	<varies>	[subID][data]...[datan]	NVDM_NOISE	Noise messages (see below)
0xC3	REP_MutNormSection	2+2*nodes	[rx][tx][*uint16* node0][*uint16* node1]...[*uint16* noden]	NVDM_MUTCACHE	Sends out a dynamic amount of nodes (from 1 to full RX electrode)
0xCF	REP_ParameterRead	2+len	[*uint16* address][data] (up to ‘len’ bytes)	[NO GATE]	Parameter read response
0xF0	REP_Ack	1	[command ID]	[NO GATE]	Acknowledgment of receipt of command
0xF2	REP_TouchFiltered	5*i	[STATE/ID][uint16 X][uint16 Y]	NVDM_FINGERPOS	Filtered (but not scaled) touch coordinates
0xF3	REP_TouchPredict	9	[ID][*uint16* X0][*uint16* Y0][*uint16* Xpred][*uint16* Ypred]	NVDM_RAWPOS	Prediction value for a touch
0xF4	REP_TouchRaw	5*i	[STATE/ID][*uint16* X][*uint16* Y]	NVDM_RAWPOS	Raw touch report (pre-filter)
0xF5	REP_TouchPos16	5*i	[PEN/ID][*uint16* X][*uint16* Y]	NVDM_FINGERPOS	Final scaled touch report – first byte has touch status as bit 7
0xFA	REP_SelfRaw	2*numRXch	[uint16 self0][uint16 self1]...[uint16 selfn]	NVDM_SELFRAW	Self measurements (raw)
0xFD	REP_SelfNorm	2*numRXch	[*uint16* self0][*uint16* self1]...[*uint16* selfn]	NVDM_SELFNORM	Self measurements (normalized)
0xFE	REP_ForwardGestIC	<varies>	[data]...[datan]	NVDM_GESTIC	Packet from GestIC® (direct)
0xFF	REP_FwVersion	<varies>	[fwVersionInfo]	[NO GATE]	Large array of bytes denoting all firmware information
0x04	CMD_Echo	<varies>	[data]...[datan]	n/a	Firmware will echo back any payload sent
0x17	CMD_ReadFlash	6	[uin32 address][uint16 size]	n/a	Allows host to read Flash contents of device (fw dump)
0x55	CMD_EnterBootLoader	0	(none)	n/a	Commands firmware to enter the bootloader – ACK will be sent before jumping
0xE0	CMD_SetParameter	10	[*uint16* address][*uint8*[4] data][*uint8*[4] mask]	n/a	Writes a parameter
0xE1	CMD_GetParameter	2	[*uint16* address]	n/a	Reads a parameter
0xFB	CMD_ForceBaseline	0	(none)	n/a	Forces a baseline
0xFC	CMD_ResetGestIC	0	(none)	n/a	Resets GestIC immediately
0xFD	CMD_GestIC	<varies>	(gestic command)	n/a	Sends packet directly on to GestIC
0xFF	CMD_QueryVersion	0	(none)	n/a	Requests all firmware version information – bytes 124:127 represent Rev[2].Minor.Major

Note: Refer to parameter documentation for explanation of NVDM bitfields.

4.2.1 SET PARAMETER COMMAND

FIGURE 4-2: SET PARAMETER COMMAND

ADDRESS		DATA				MASK			

Address:	16bit address location of parameter to set
Data:	32bit (4 bytes) data value to write. For data sizes smaller than 4 bytes, pad with 0x00
Mask:	32 bit (4 bytes) mask value to mask off bits that should not be set. (usually set to 0xFFFFFFFF)

4.2.2 GET PARAMETER COMMAND

FIGURE 4-3: GET PARAMETER COMMAND

ADDRESS	

Address: 16bit address location of parameter to retrieve

5.0 PARAMETERS

5.1 Operation

Default parameters are loaded on start-up, as shown in the parameter table section. These values can be modified during runtime, but will not be restored on Reset. To permanently modify parameters, the MTCH6303 Utility should be used to export and Flash a new configuration. Refer to the MTCH6303 Utility documentation for more information.

5.2 Parameter Table

Many parameters are tuned by the MTCH6303 Utility itself, so descriptions are not provided. Table 5-1 is provided for reference only.

TABLE 5-1: PARAMETER TABLE

Module	Name	Address	Format	Default	Description
pub	mgc3130	0x0102	uint8_t	0	1 = MTC3130 is present
pub	numberOfRXChannels	0x0100	uint8_t	27	Number of RX channels currently in use
pub	numberOfTXChannels	0x0101	uint8_t	19	Number of TX channels currently in use
pub	diagMask	0x0080	uint16_t	[see NVDM]	[see NVDM]
pub	activeModules	0x0081	uint16_t	[see NVAM]	[see NVAM]
pub	streamingMode	0x0082	uint8_t	0	see Operating Modes
pub	swipeDistance	0x0501	uint16_t	4*256	See Gesture definition
pub	swipeTimeout	0x0500	uint32_t	msec2ticks(1500)(1)	See Gesture definition
pub	swipeBorder	n/a (struct)		n/a	See Gesture definition
pub	swipeBorder.left	0x0502	uint16_t	3*256	See Gesture definition
pub	swipeBorder.right	0x0503	uint16_t	24*256	See Gesture definition
pub	swipeBorder.top	0x0504	uint16_t	3*256	See Gesture definition
pub	swipeBorder.bottom	0x0505	uint16_t	16*256	See Gesture definition
pub	swipeExtBorder	n/a (struct)		n/a	See Gesture definition
pub	swipeExtBorder.left	0x0506	uint16_t	2*256	See Gesture definition
pub	swipeExtBorder.right	0x0507	uint16_t	25*256	See Gesture definition
pub	swipeExtBorder.top	0x0508	uint16_t	2*256	See Gesture definition
pub	swipeExtBorder.bottom	0x0509	uint16_t	17*256	See Gesture definition
pub	tapBorder	n/a (struct)		n/a	See Gesture definition
pub	tapBorder.left	0x0540	uint16_t	1*256	See Gesture definition
pub	tapBorder.right	0x0541	uint16_t	26*256	See Gesture definition
pub	tapBorder.top	0x0542	uint16_t	1*256	See Gesture definition
pub	tapBorder.bottom	0x0543	uint16_t	18*256	See Gesture definition
pub	tapTimeout	0x0544	uint32_t	mSec2Ticks(200)(1)	See Gesture definition
pub	dblTapTimeout	0x0545	uint32_t	mSec2Ticks(500) (1)	See Gesture definition
pub	commSelectMode	0x0584	uint8_t	0	0 = use COMMSEL pin, 1 = force I^{2}C™, 2 = force USB
pub	irqPolarity	0x0581	uint8_t	0	0 = Active-Low, 1 = Active-High
pub	irqPulseWidth	0x0582	uint8_t	5	Value in msec to pulse (when mode 2)
pub	irqTrigger	0x0583	uint8_t	2	0 = Off, 1 = Set on frame, 2 = Set on touch, 3 = Set on touch changed
pub	irqMode	0x0580	uint8_t	1	0 = Off, 1 = Level-trigger, 2 = Pulse-trigger
pub	idleTime2D	0x0103	uint16_t	100	Scan period while 2D is idle (in msec)
map	txSelfTape	0x02c0	uint16_t [66]	[see below]	
map	rxPinMap	0x0200	uint8_t[27]	[see below]	
map	rxPrechargePinMap	0x0240	uint8_t[27]	[see below]	
map	txPinMap	0x0280	uint8_t[36]	[see below]	
acq	baseUpdateTime	0x0802	uint32_t	mSec2Ticks(10000) (1)	Calibration update rate
acq	selfScanPhase	0x0812	uint16_t[4]	{52,45,40,40}	Self measurement period
acq	selfScanISRPhase	0x0816	uint16_t[4]	{59,49,46,45}	Self measurement phase
acq	mutScanPeriode	0x0803	uint16_t[4]	{122,105,104,100}	Mutual measurement period

TABLE 5-1: PARAMETER TABLE (CONTINUED)

Module	Name	Address	Format	Default	Description
acq	mutScanPhase	0x0807	uint16_t[4]	{68,60,59,55}	Mutual measurement phase
acq	mutFreqHopping	0x080B	uint8_t	0	Frequency hopping control (0 = enabled, 1-4 = lock to F0-F3)
acq	mutFreqHoppingLevel	0x080C	int8_t[4]	{0,0,0,0}	Linear gain to apply to results from each frequency
acq	diagRxChannel	0x0800	uint8_t	0xff	
acq	diagTxChannel	0x0801	uint8_t	0xff	
acq	syncRxChannel	0x081A	uint8_t	0xff	
acq	syncTxChannel	0x081B	uint8_t	0xff	
acq	fullScanRxStart	0x081C	uint8_t	0	
acq	fullScanRxStop	0x081D	uint8_t	27	
acq	fullScanTxStart	0x081E	uint8_t	0	
acq	fullScanTxStop	0x081F	uint8_t	19	
dec	penDownTimer	0x0403	uint16_t	781	
dec	penUpTimer	0x0404	uint16_t	781	
dec	selfTouchThres	0x0400	uint8_t	60	
dec	mutTouchThres	0x0401	uint8_t	60	
dec	minCuspDelta	0x040b	uint8_t	25	
dec	weightThreshold	0x0402	uint8_t	20	
dec	minTouchDistance	0x040c	uint8_t	5*8	
dec	fatThreshold	0x040d	uint8_t	95	
dec	nbSampleSelf	0x0407	uint8_t	64	
dec	touchActiveHysteresis2D	0x0409	uint16_t	1000	
dec	touchActiveHysteresis2D3D	0x0401	uint16_t	50	
rep	flipState	0x0041	uint8_t	0b010	
rep	rxScale	n/a (struct)		n/a	
rep	rxScale.shift	0x0042	uint8_t	7	
rep	rxScale.divide	0x0043	uint8_t	27	
rep	rxScale.offset	0x0044	uint16_t	0	
rep	txScale	n/a (struct)		n/a	
rep	txScale.shift	0x0045	uint8_t	7	
rep	txScale.divide	0x0046	uint8_t	19	
rep	txScale.offset	0x0047	uint16_t	0	
mtc	mtch65x_active_config	none	uint32_t	0x27	
mtc	mtch65x_periode_fast_rise	0x0900	uint16_t	10	
mtc	mtch65x_periode_fast_rise_oc	0x0901	uint16_t	7	
mtc	mtch65x_fast_rise_delay	0x0902	uint16_t	300	
mtc	mtch65x_periode_self_measurement	0x090D	uint16_t[4]	{20,20,20,20}	
mtc	mtch65x_periode_self_measurement_oc	0x0911	uint16_t[4]	{10,10,10,10}	
mtc	mtch65x_periode_mutu_measurement	0x0905	uint16_t[4]	{66,60,59,58}	
mtc	mtch65x_periode_mutu_measurement_oc	0x0909	uint16_t[4]	{16,15,14,14}	

Note 1: mSec2Ticks(ms) = (((ms) * 625 + 2) / 4)

EXAMPLE 5-1: COMPLICATED INITIALIZATIONS

```
rxPinMap = {(15), (14), (13), (12), (11), (10), (9), (8), (7), (6), (0), (1), (2), (3), (4), (5),
(19), (18), (17), (16), (27), (23), (22), (21), (20), (26), (24)}
rxPrechargePinMap = {(24), (24), (24), (24), (24), (24), (24), (24), (24), (24), (24), (24), (24),
(15), (15), (15), (15), (15), (15), (15), (15), (15), (15), (15), (15), (15), (15)}
txPinMap = {(0+ 17), (0+ 18), (0+ 0), (0+ 1), (0+ 2), (0+ 3), (0+ 4), (0+ 5), (0+ 6), (0+ 7), (0+
8), (0+ 9), (0+ 10), (0+ 11), (0+ 12), (0+ 13), (0+ 14), (0+ 15), (0+ 16)}
txSelfTape = {0x0000,
0x0F,0x0010,0x0010,0x7110,0x00,0x00,0x00,0x00,0x00,0x00,0x0F,0x0010,0x0310,0x8110,0x00,0x00,0x00
,0x00,0x00,0x00,0x0F,0x0010,0x1C10,0x0110,0x00,0x00,0x00,0x00,0x00,0x00,0x0F,0x0010,0xE010,0x011
0,0x00,0x00,0x00,0x00,0x00,0x00,0x0F,0x0F10,0x0010,0x0110,0x00,0x00,0x00,0x00,0x00,0x00,0x0C,0x0
000,0x0000,0,0,0,0,0,0,0,0,0,0,0,0}
```

5.3 Special Parameters

5.3.1 ACTIVE MODULES REGISTER (NVAM)

REGISTER 5-1: ACTIVE MODULES REGISTER (NVAM)

U-x	U-x	U-x	U-x	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	—	DECODE	DIGITIZER	AUTOBASE	BESTFREQ
bit 15							bit 8

R/W-0	R/W-0	R/W-0	U-x	U-x	R/W-0	U-x	R/W-1
AW_EVENT	SW_EVENT	FL_EVENT	—	—	FULLSCAN	—	GESTURE
bit 7							bit 0

Legend:		
R = Readable bit	x = Bit is unknown	-n = Value after initialization (default)
W = Writable bit	U = Unimplemented bit	q = Conditional
‘1’ = Bit is set	‘0’ = Bit is cleared	

bit 15-12	**Unused**	
bit 11	**DECODE:**	Turns touch decoding logic on or off
bit 10	**DIGITIZER:**	Turns digitizer/I^{2}C™ register output on or off
bit 9	**AUTOBASE:**	Turns on or off automatic baseline functionality
bit 8	**BESTFREQ:**	Turns on or off bestfrequency selection algorithms
bit 7	**AW_EVENT:**	Events related to GestIC airwheel
bit 6	**SW_EVENT:**	Events related to GestIC swipes
bit 5	**FL_EVENT:**	Events related to GestIC flicks
bit 4-3	**Unused**	
bit 2	**FULLSCAN:**	Turns on full mutual scanning
bit 1	**Unused**	
bit 0	**GESTURE:**	Turns on 2d gesture recognition

5.3.2 DIAGNOSTIC MODULES REGISTER (NVDM)

REGISTER 5-2: ACTIVE DIAGNOSTICS MODULES REGISTER (NVDM)

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
GESTIC	DIAG	CUSTOM	GESTURE	FINGERPOS	RAWPOS	NOISE	TRACE
bit 15							bit 8

U-x	U-x	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	ADC_COR	ADC	MUTRAW	SELFRAW	MUTCACHE	SELFNORM
bit 7							bit 0

Legend:		
R = Readable bit	x = Bit is unknown	-n = Value after initialization (default)
W = Writable bit	U = Unimplemented bit	q = Conditional
‘1’ = Bit is set	‘0’ = Bit is cleared	

bit 15	**GESTIC:**	Forward GestIC® packets to host, also packets from host to GestIC
bit 14	**DIAG:**	Diagnostic Messages
bit 13	**CUSTOM:**	Custom Messages
bit 12	**GESTURE:**	Gesture Messages
bit 11	**FINGERPOS:**	Filtered Touch Data
bit 10	**RAWPOS:**	Unfiltered Touch Data
bit 9	**NOISE:**	Noise Messages
bit 8	**TRACE:**	Trace Messages
bit 7-6	**Unused**	
bit 5	**ADC_COR:**	Use ADC Offsets
bit 4	**ADC:**	ADC Messages
bit 3	**MUTRAW:**	Mutual Raw Data
bit 2	**SELFRAW:**	Self Raw Data
bit 1	**MUTCACHE:**	Mutual Normalized Data
bit 0	**SELFNORM:**	Self Normalized Data

6.0 COMMUNICATION EXAMPLES

6.1 Reading Touch Data

The following examples show a frame of data communicating three Touch ID contact points:

TABLE 6-1: READING TOUCH DATA

Touch ID	ID5
5	Contact at (2345,4657)
8	Contact at (9823,0023)
13	Touch Removed (last contact 7264,1893)

6.1.1 READING TOUCH DATA (USB)

Touch data is populated in the HID report (refer to **Section 3.3.2, HID Generic (EP 2, Streamed Messages)**).

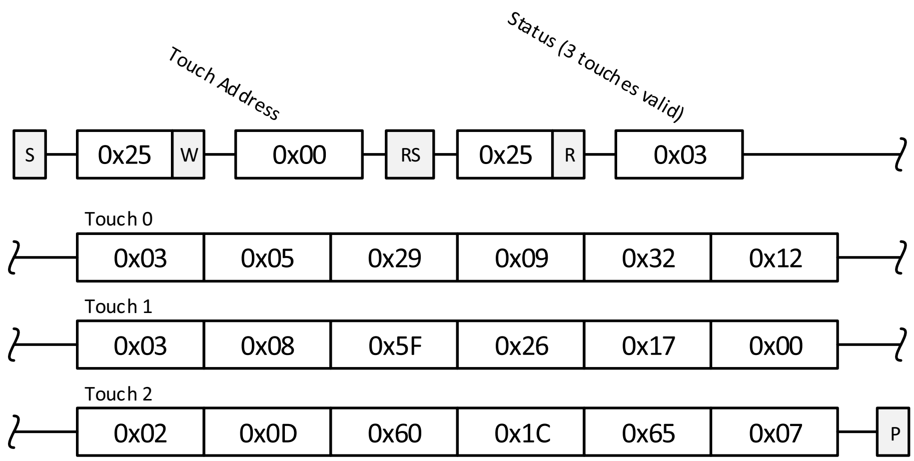
TABLE 6-2: READING TOUCH DATA HID REPORT

0	0x01 REPID	0x03 STATUS0	0x05 ID0	0x29 XLSB0	0x09 XMSB0	0x31 YLSB0	0x12 YLSB0	0x03 STATUS1	7
8	0x08 ID1	0x5F XLSB1	0x26 XMSB1	0x17 YLSB1	0x00 YMSB1	0x02 STATUS2	0x0D ID2	0x60 XLSB2	15
16	0x1C XMSB2	0x65 YLSB2	0x07 YMSB2	0x00 STATUS3	— ID3	— XLSB3	— XMSB3	— YLSB3	23
24	— YMSB3	— STATUS4	— ID4	— XLSB4	— XMSB4	— YLSB4	— YMSB4	— STATUS5	31
32	— ID5	— XLSB5	— XMSB5	— YLSB5	— YMSB5	— STATUS6	— ID6	— XLSB6	39
40	— XMSB6	— YLSB6	— YMSB6	— STATUS7	— ID7	— XLSB7	— XMSB7	— YLSB7	47
48	— YMSB7	— STATUS8	— ID8	— XLSB8	— XMSB8	— YLSB8	— YMSB8	— STATUS9	55
56	— ID9	— XLSB9	— XMSB9	— YLSB9	— YMSB9	0x03 #VALID	— —	— —	

6.1.2 READING TOUCH DATA (I^2C)

Reading touch data over I^2C must be performed in one single transaction to ensure the data is all from the same frame.

FIGURE 6-1: READING TOUCH DATA (I^2C™)



Note: The host could continue to read all 10 touches, but there is no need since the first byte indicates only three touches are valid.

6.2 Message Send/Receive

In these examples, a message setting the current number of RX channels is sent, and the response received is shown. (including acknowledgment).

6.2.1 MESSAGE TO SEND

Message ID

0xE0 (CMD_SetParameter)

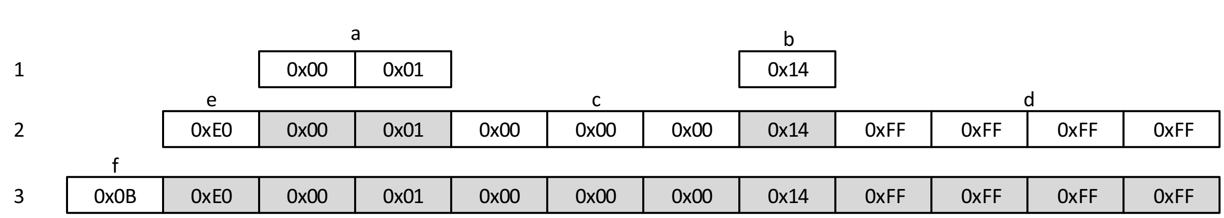
Payload (message specific)

Address: 0x0100

Data: 0x14

First, the message must be created according to the message format in Figure 6-2.

FIGURE 6-2: MESSAGE TO SEND



6.2.1.1 Steps

1. Parameter address (a) and value to write (b)
2. Message ID is added (e).

Fill bytes are added to value to make it 32 bits (c).

Data mask is added (d) – note that since the parameter is only one byte, only the last byte of the mask actually affects the behavior.

3. Status byte is added:
   - size is 11 (0x0B)
   - “more messages” is set to 0
   - “is continued” ID set to 0 (this is the start of message)

6.2.2 EXPECTED RESPONSE

Every message sent to the controller also contains an acknowledgment message back (ACK), which follows this format:

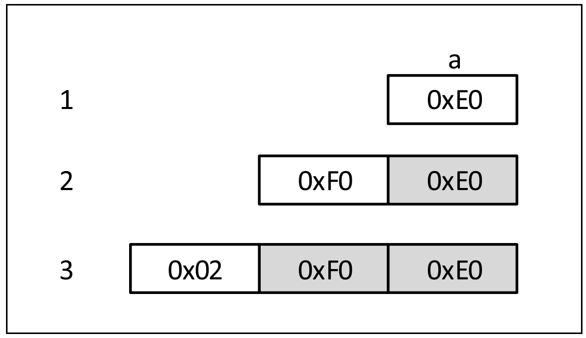
Message ID

0xF0 (REP_Ack)

Payload

0xE0 (command received was CMD_SetParameter)

FIGURE 6-3: EXPECTED RESPONSE

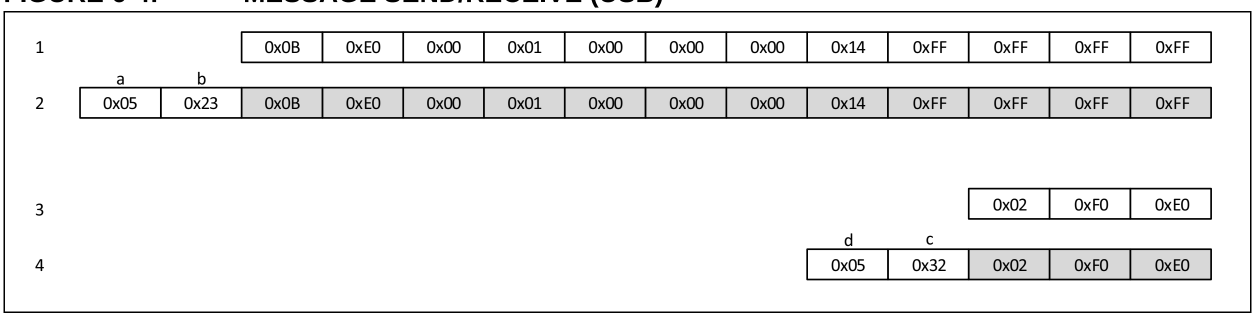


6.2.2.1 Steps

1. Expected payload for an ACK message is an echo of the command being ACK’d – in this case, 0xE0
2. Message ID is added
3. Status byte is added:
   - Size = 2
   - More messages = 0
   - Continued = 0

6.2.3 MESSAGE SEND/RECEIVE (USB)

FIGURE 6-4: MESSAGE SEND/RECEIVE (USB)



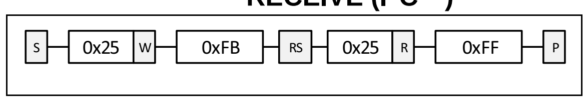
6.2.3.1 Steps

1. Message to send (from previous section)
2. Adding sequence ID (b), which was chosen at random for this example. Adding reportID (always 0x05)
3. Response expected (from previous section)
4. Adding sequence ID (c), which was chosen at random for this example. Adding reportID (always 0x05).

6.2.4 MESSAGE SEND/RECEIVE (I^2C)

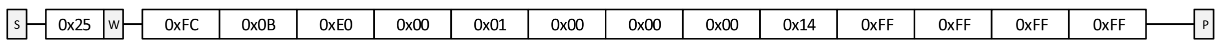
First, the host must query the RXRDY buffer to ensure there is enough space to write the command. In this case, the controller is reporting that 255 bytes are available for writing:

FIGURE 6-5: MESSAGE SEND/ RECEIVE (I^2C™)



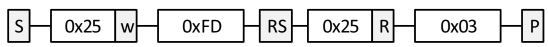
Next, the host writes the command into the controller's RXBUFF register (Figure 6-6).

FIGURE 6-6: HOST WRITE TO RXBUFF REGISTER



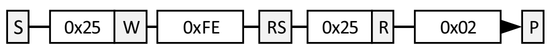
The host may now query the TXRDY buffer to see if the response is ready, either after a set amount of time or by observing IRQ (Figure 6-7).

FIGURE 6-7: HOST READ FROM TXRDY REGISTER



Since there are three bytes ready to be read, the host should now read those three bytes out of the TXBUFF register (Figure 6-8).

FIGURE 6-8: HOST READ FROM TXBUFF REGISTER



Note: The two previous steps could have been completed in a single read by a host capable of performing decisions during a read (based on the value of TXRDY).

Reading address 0xFD auto-increments the address pointer to 0xFE, the stream buffer. Further bytes read will all be from within the stream buffer, maintaining the 0xFE address. The first byte read, 0x03, would indicate that three more bytes are within the stream buffer and may be read immediately.

7.0 SENSOR DESIGN CONSIDERATIONS

7.1 Sensor Patterns and PCB Layout

With regard to touch sensor patterns, refer to the mTouch® Design Center (www.microchip.com/mtouch) for additional information on designing and laying out a touch sensor pattern, as well as using the correct techniques for PCB trace routing.

7.1.1 PROTOTYPING DESIGNS

Touch sensor designs typically require a thorough debugging phase to ensure a reliable product. If possible, it is recommended that flexible prototyping hardware be created with this in mind. A common example is providing external access to the communication lines for quick test and tuning while in circuit.

7.1.2 SENSOR OVERLAY MATERIAL

To prevent saturation of sensor levels, a minimum overlay of 0.5 mm plastic or glass is required for proper operation of the device, even during a prototyping phase. (Even if this value is different than the final design.)

Note:	At no time should the device be expected to respond correctly to a user touching a bare PCB sensor.

7.1.3 OPERATION WITH AN LCD

The MTCH6303 has integrated algorithms to detect and minimize the effects of noise, but proper care should always be taken in selecting an LCD and support components with a focus on reducing noise as much as possible. Since the interaction between the touch sensor and display is highly dependent upon the physical arrangement of the components, proper testing should always be executed with a fully integrated device. Please reference your projected capacitive touch screen manufacturer's integration guide for additional design considerations.

7.2 Sensor Layout Configuration

TABLE 7-1: REGISTERS ASSOCIATED WITH SENSORS LAYOUT CONFIGURATION

Address	Name	Description
0x0200	NUMBEROFX-CHANNELS	Number of channels used for X axis
0x0280	NUMBEROFY-CHANNELS	Number of channels used for Y axis

The MTCH6303 is designed to work with sensors with a minimum of 3 RX and 3 TX sensor channels, and a maximum of 27 RX and 19 TX channels using a single MTCH652.

7.3 Sensor Output Resolution

The MTCH6303 interpolates 256 discrete points between each sensor channel and 128 points past the centerline of each edge. These internal values are then scaled over a default range of 0-32767 (0-0x7FFF) for the default sensor configuration. If the number of TX or RX channels is modified, then the related output resolution values must also be updated.

7.3.1 MODIFYING OUTPUT RESOLUTION

TABLE 7-2: OUTPUT RESOLUTION REGISTERS

Address	Name
0x0042	RX Shift
0x0043	RX Divide
0x0044	RX Offset
0x0045	TX Shift
0x0046	TX Divide
0x0047	TX Offset

The X and Y resolution may be modified by changing addresses 0x0042 through 0x0047. The firmware uses the values in equation Equation 7-1.

EQUATION 7-1: RESOLUTION CHANGE

$$Final\ Value = \frac{Value \cdot 2^{Shift}}{Divide} + Offset$$

These values must be updated when changing the number of TX or RX channels on the sensor if the 0-0x7FFF resolution is to be maintained.

7.4 Sensor Orientation

TABLE 7-3: SENSOR ORIENTATION

Address	Name	Description
0x0041	FLIPSTATE	Determines X and Y flips, as well as swaps

REGISTER 7-1: SENSOR ORIENTATION REGISTER

U-x	U-x	U-x	U-x	U-x	R/W-0	R/W-0	R/W-0
—	—	—	—	—	XYSWAP	TXFLIP	RXFLIP
bit 7							bit 0

Legend:		
R = Readable bit	x = Bit is unknown	-n = Value after initialization (default)
W = Writable bit	U = Unimplemented bit	q = Conditional
‘1’ = Bit is set	‘0’ = Bit is cleared	

bit 7-3 **Unused**

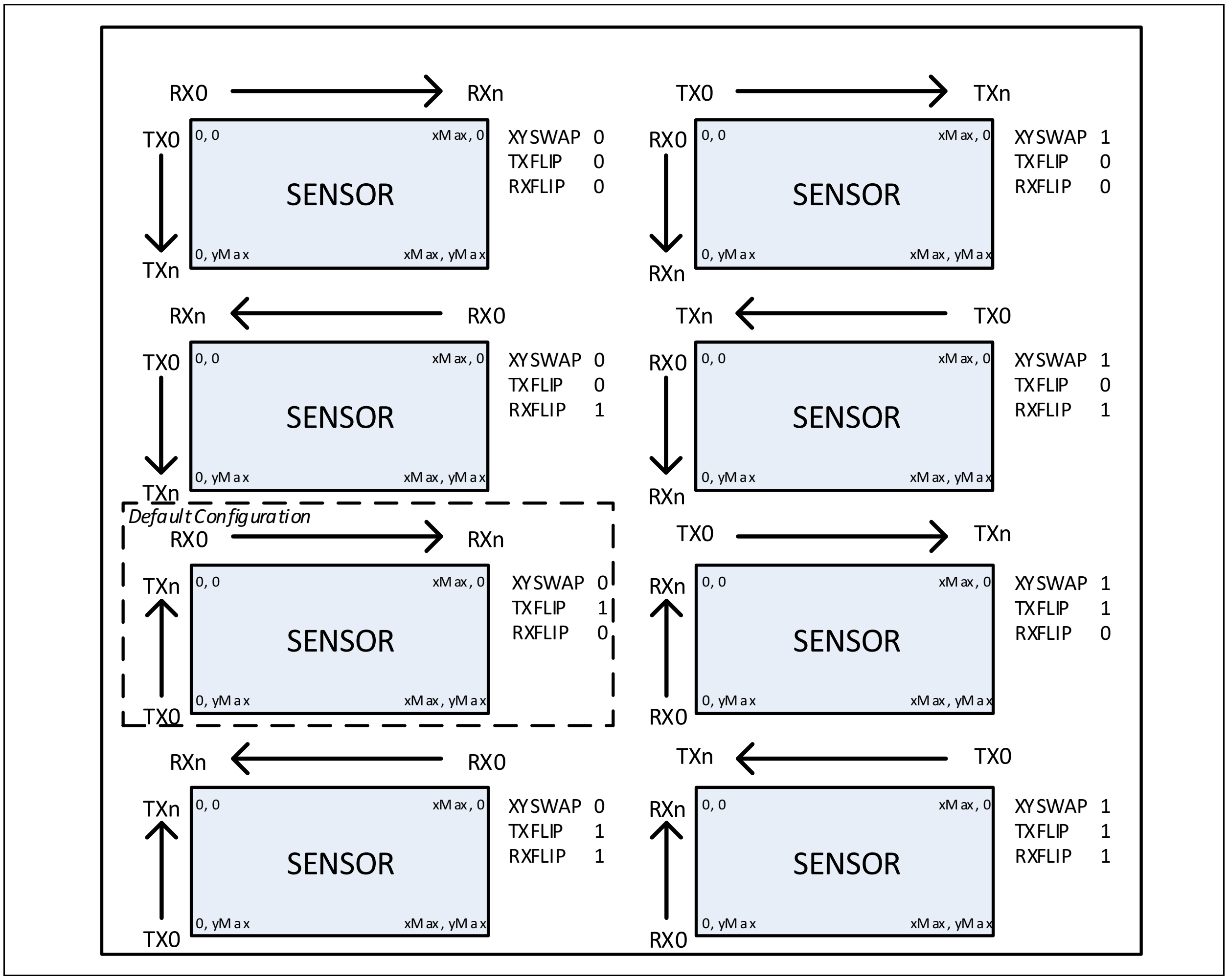
bit 2 **XYSWAP:** Swap the TX and RX coordinates

bit 1 **TXFLIP:** Swap the coordinates along the TX axis

bit 0 **RXFLIP:** Swap the coordinates along the RX axis

To aid in PCB layout, the sensor can be oriented in any direction, have either axis reversed or have the axis swapped.

FIGURE 7-1: SENSOR ORIENTATION EXAMPLES



 Preliminary

8.0 FIRMWARE UPDATE

8.1 Library Loader

The MTCH6303 devices are manufactured with a built-in Library Loader (bootloader) only. There will not be any PCAP touch decoding library preloaded. The library loader has interfaces for USB HID and I^2C, so that an MTCH6303 library can be uploaded to the MTCH6303 Flash memory.

The latest MTCH6303 PCAP touch decoding library can be found in the MTCH6303 Utility download which can be accessed from the MTCH6303 device page.

There are three ways to upload the MTCH6303 library to the MTCH6303 device, as listed in Sections 8.1.1. to 8.1.3 below.

8.1.1 UPLOAD VIA THE MTCH6303 UTILITY

The MTCH6303 Utility can be used to perform the update. For this option, USB connectivity to a PC with the MTCH6303 Utility installed will be needed.

8.1.2 UPLOAD VIA EMBEDDED HOST CONTROLLER

This option requires an embedded host controller which performs the upload using the MTCH6303 Bootloader commands (refer to Table 8-1).

Microchip pre-programmed MTCH6303 parts can be ordered through the Microchip Programming Center. Please reference www.microchipdirect.com/programming for further information.

8.1.3 QUICK TIME PROGRAMMING (QTP)

For larger quantities of pre-programmed parts with unique part numbers, contact your local Microchip sales office.

8.2 Overview

The firmware update process involves a host device transmitting a hex file to the MTCH6303 while in Bootloader mode. The hex file should be parsed and all data bytes extracted before being sent to the MTCH6303. This can either be done by the host or by software that utilizes the host as a bridge to send the bytes to the MTCH6303.

When the MTCH6303 is in Bootloader mode, the host has access to commands to read, erase and write ROM pages that contain the touch application. An outline of the update procedure is detailed in this section.

8.3 Bootloader Command Overview

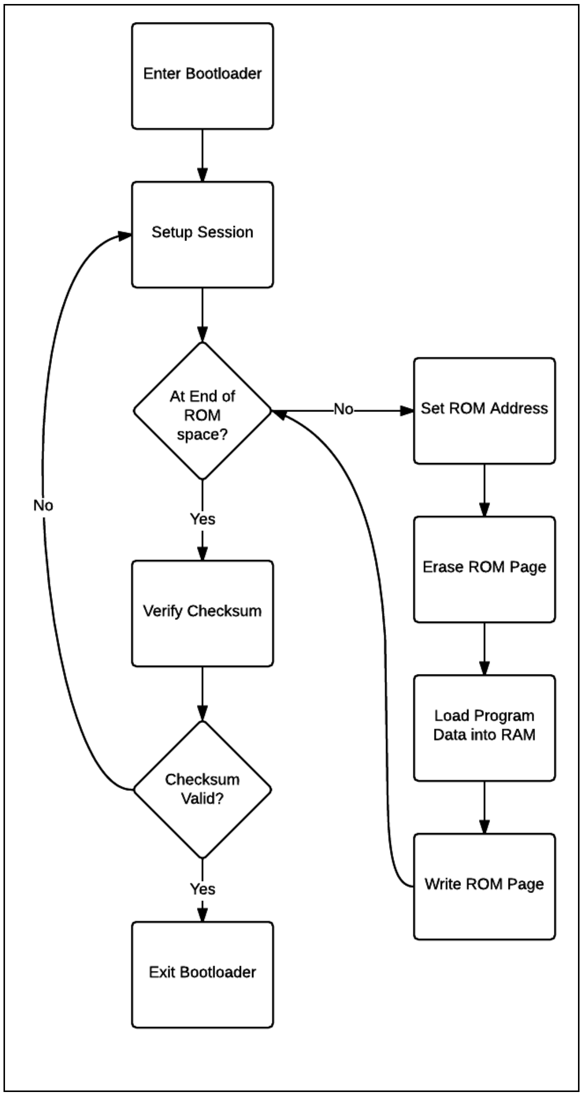
The command interpreter within the bootloader operates in a similar manner as the standard MTCH6303 command interpreter. The bootloader supports the following commands:

TABLE 8-1: BOOTLOADER COMMANDS

ID	Name	Description
0x10	EXIT_BOOTLOADER	Exit Bootloader mode
0x11	SETUP_SESSION	Setup and initiate a bootloading session
0x12	ERASE_PAGE	Erase a ROM page
0x13	SET_ADDRESS	Write the Flash address to operate on
0x14	LOAD_DATA	Load program data into RAM
0x15	WRITE_PAGE	Latch program data from RAM into ROM
0x16	VALIDATE_FW	Read from a section in Flash
0x17	READ_FLASH	Read a section in ROM
0xff	QUERY_VERSION	Read the bootloader firmware revision

8.4 Update Procedure

FIGURE 8-1: BOOTLOADER FLOWCHART



8.4.1 ENTER THE BOOTLOADER

The MTCH6303 normally runs in Application mode, so the host must communicate to the MTCH6303 to enter its Bootloader mode. To do this, issue the 'Enter bootloader' command as seen in **Section 10.5, Command: ENTER_BOOTLOADER**. If using USB, the device will disconnect from the USB bus, then re-attach as the bootloader. (VID 0x04D8, PID 0x09D5)

8.4.2 SETUP A FIRMWARE UPDATE SESSION

Once the MTCH6303 is in Bootloader mode establish an update session with the MTCH6303. The purpose of this is to setup the ROM boundaries and other various parameters for the update. Use the SETUP_SESSION command to configure the session. Prior to receiving a valid SETUP_SESSION command the bootloader will not allow modifications to the ROM. Once a SETUP_SESSION command is received, the application firmware is identified as unstable and it is no longer possible to exit the bootloader until a firmware update sequence has been completed.

8.4.3 PERFORM A SETADDRESS/ERASE/WRITE CYCLE ON EACH ROM PAGE

With a valid session in place the host can now begin accessing the MTCH6303 device's ROM to update the firmware. The typical procedure is to update the device one Flash page at a time, erasing and writing one page before moving onto the next.

First, use SET_ADDRESS to configure the address of the start of the ROM page to perform further operations on. The address should be the start of a 4 Kb ROM page.

Next, use ERASE_PAGE to erase the page starting at the address selected using the SET_ADDRESS command.

Once the page is erased, the host should send parsed hex data to update the selected page. This process will take several iterations of the LOAD_DATA command to write all 4Kb of data. The LOAD_DATA command has size and offset parameters that denote respectively the size of the current LOAD_DATA packet and the offset from the address defined by the SET_ADDRESS command.

After all 4 Kb of data has been provided to the controller, use the WRITE_PAGE command to write the data into the selected page.

Continue this process of SET_ADDRESS, ERASE_PAGE, LOAD_DATA(s), and WRITE_PAGE for each 4Kb block of ROM until the entire update is completed.

8.5 Bootloader Commands

This section describes the bootloader commands. The format for each command and its response are detailed below.

8.5.1 COMMANDS

8.5.1.1 0X10 EXIT_BOOTLOADER

When called, this command will cause the bootloader to exit, returning to the touch application if a valid application is present. If not, the controller will remain in Bootloader mode.

TABLE 8-2: EXIT BOOTLOADER

Cmd Size	Cmd ID	Data
0x01	0x10	<none>

8.5.1.2 0x11 SETUP_SESSION

This will initiate a bootloading session, defining session type, start address and end address.

TABLE 8-3: SETUP SESSION

Cmd Size	Cmd ID	Data		
0x0A	0x11	Session Type [8 bits]	Start Address [32 bits]	End Address [32 bits]

8.5.1.3 0x12 ERASE_PAGE

This command will cause the currently set page to be erased. The SET_ADDRESS command must be used to define the address of the page to be erased prior to calling ERASE_PAGE.

TABLE 8-4: ERASE PAGE

Cmd Size	Cmd ID	Data
0x01	0x12	<none>

8.5.1.4 0x13 SET_ADDRESS

This command defines the start address of the page of ROM to perform further operations upon. This address MUST be the start of one of the 4 Kb ROM pages.

TABLE 8-5: SET ADDRESS

Cmd Size	Cmd ID	Data			
0x05	0x13	Addr[7:0]	Addr[15:8]	Addr[23:16]	Addr[31:24]

8.5.1.5 0x14 LOAD_DATA

Load application data from the host into RAM.

TABLE 8-6: LOAD DATA

Cmd Size	Cmd ID	Data						
		Size		Offset		progData[0]-progData[n]**		
varies	0x14	[7:0]	[15:8]	[7:0]	[15:8]	[0]		[n]**

Note:	Max length of progData is 54 bytes.

8.5.1.6 0x15 WRITE_PAGE

Write loaded RAM data into ROM at the defined address.

TABLE 8-7: WRITE PAGE

Cmd Size	Cmd ID	Data
0x01	0x15	<none>

8.5.1.7 0x16 VALIDATE_FW

Read from a section in Flash.

TABLE 8-8: VALIDATE FW

Cmd Size	Cmd ID	Data
0x01	0x16	<none>

8.5.1.8 0x17 READ_FLASH

Read a section in ROM.

TABLE 8-9: READ FLASH

Cmd Size	Cmd ID	Data	
0x03	0x17	Size [7:0]	Size[15:8]

8.5.1.9 0xff QUERY_VERSION

Read the bootloader firmware version.

TABLE 8-10: QUERY VERSION

Cmd Size	Cmd ID	Data
0x01	0xff	<none>

8.5.2 RESPONSES

TABLE 8-11: BOOTLOAD COMMAND RESPONSE ID

Value	Description
0x00	Successful operation
0x07	Checksum mismatch
0x08	Flash read/erase/write failure
0x0a	Out-of-Range address
0x0b	No session data
0x0c	Unrecognized command ID
0x0d	Invalid number of bytes for this command
0x0e	Error exiting Bootloader mode

TABLE 8-12: BOOTLOADER COMMAND RESPONSE

Byte	Value	Description
1	0x02	Length, number of bytes to follow
2	varies	Echo Command ID
3	See Table 8-11 for values	Response Status

9.0 OPERATING MODES

The MTCH6303 allows enabling and disabling individual modules within the controller by modifying the active Modules (NVAM) register. Node control is from the NVAM in conjunction with the Streaming Modes register.

REGISTER 9-1: STREAMING MODE REGISTER (STREAMINGMODE)

U-x	U-x	U-x	U-x	U-x	U-x	U-x	U-x
MODE<7:0>							
bit 7							bit 0

Legend:		
R = Readable bit	x = Bit is unknown	-n = Value after initialization (default)
W = Writable bit	U = Unimplemented bit	q = Conditional
‘1’ = Bit is set	‘0’ = Bit is cleared	

bit 7-0 **MODE:** Mode Selection – See **Section 9.2, Controller State Machine** for more information.

0: 2D3D
1: PCAP_ONLY
2: GESTIC_BRIDGE
4: ACTIVE_STANDBY
5: 2D_SLEEP_MODE
0xFF: INVALID

9.1 Active Modules Register (NVAM)

REGISTER 9-2: ACTIVE MODULES REGISTER (NVAM)

U-x	U-x	U-x	U-x	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	—	DECODE	DIGITIZER	AUTOBASE	BESTFREQ
bit 15							bit 8

R/W-0	R/W-0	R/W-0	U-x	U-x	R/W-0	U-x	R/W-1
AW_EVENT	SW_EVENT	FL_EVENT	—	—	FULLSCAN	—	GESTURE
bit 7							bit 0

Legend:		
R = Readable bit	x = Bit is unknown	-n = Value after initialization (default)
W = Writable bit	U = Unimplemented bit	q = Conditional
‘1’ = Bit is set	‘0’ = Bit is cleared	

bit 15-12	**Unused**
bit 11	**DECODE:** Turns touch decoding logic on or off
bit 10	**DIGITIZER:** Turns digitizer/I^{2}C register output on or off
bit 9	**AUTOBASE:** Turns on or off automatic baseline functionality
bit 8	**BESTFREQ:** Turns on or off bestfrequency selection algorithms
bit 7	**AW_EVENT:** Events related to GestIC® airwheel
bit 6	**SW_EVENT:** Events related to GestIC® swipes
bit 5	**FL_EVENT:** Events related to GestIC® flicks
bit 4-3	**Unused**
bit 2	**FULLSCAN:** Turns on full mutual scanning
bit 1	**Unused**
bit 0	**GESTURE:** Turns on 2D gesture recognition

9.2 Controller State Machine

Using the Active Modules Register there are numerous different operating modes for the MTCH6303. The streaming Mode register (address 0x0082) can be used to configure the overall operational mode of the controller. Please contact Microchip for further information on using the MTCH6303 in combination with an MGC3130 GestIC® controller for 3D gestures.

FIGURE 9-1: STANDARD CONTROLLER OPERATION STATE MACHINE

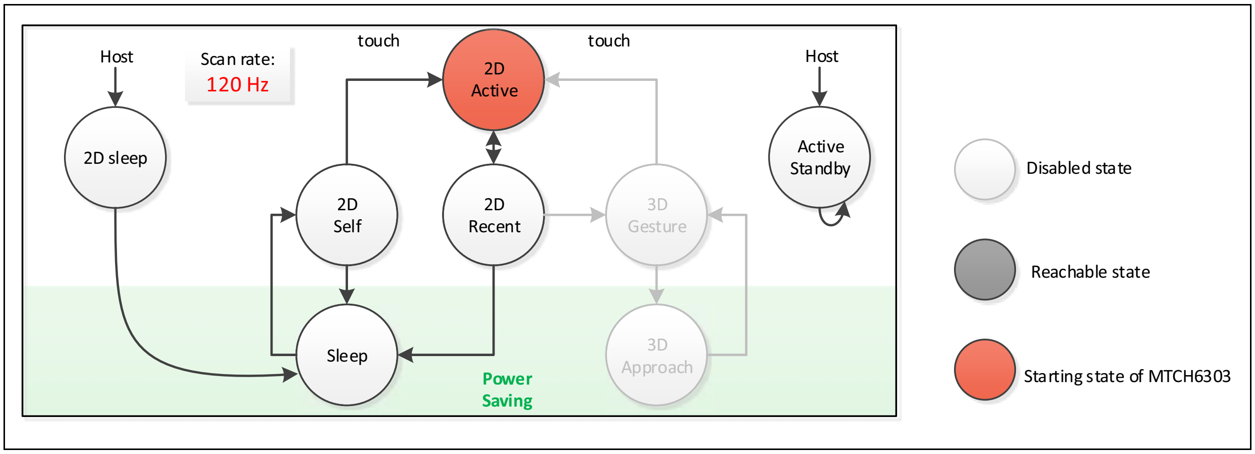
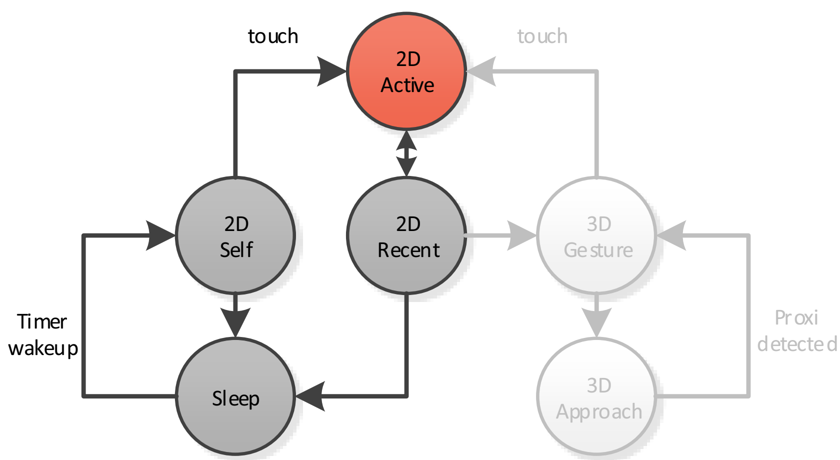
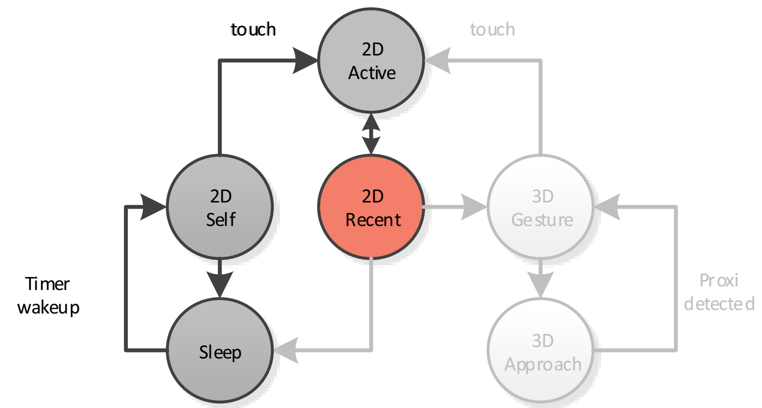


FIGURE 9-2: 2D ONLY MODE



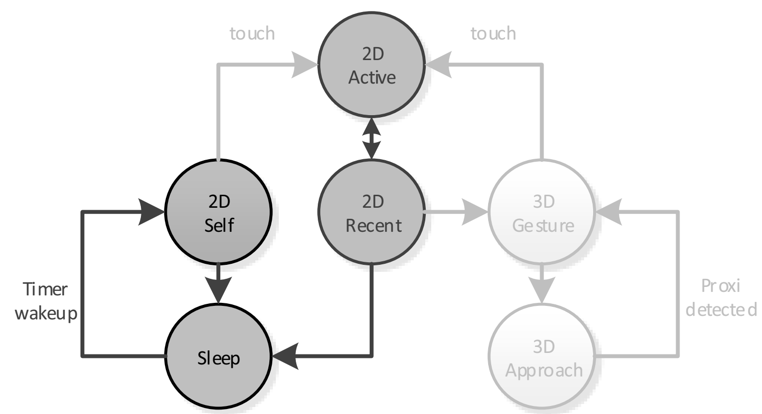
Note: Set to 2D mode only
CMD: 0x82, 1

FIGURE 9-3: DISABLE AUTO-SLEEP



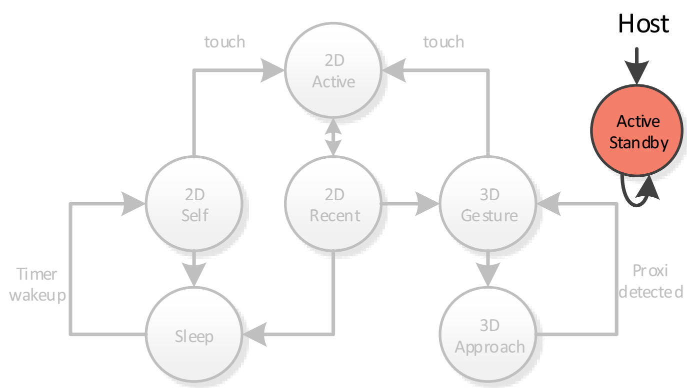
Note: Disable Auto-sleep
CMD: bit12 (NVAM)
The PCAP does not return to sleep

FIGURE 9-4: DISABLE AUTO-WAKE-UP



Note: Disable Auto-Wake-up
CMD: bit13 (NVAM)
The PCAP does not return to Active mode.
The Sleep time can be set with idleTime2D parameter
CMD: 0x103, 64 (data in ms, max 420 ms)

FIGURE 9-5: ACTIVE STANDBY MODE

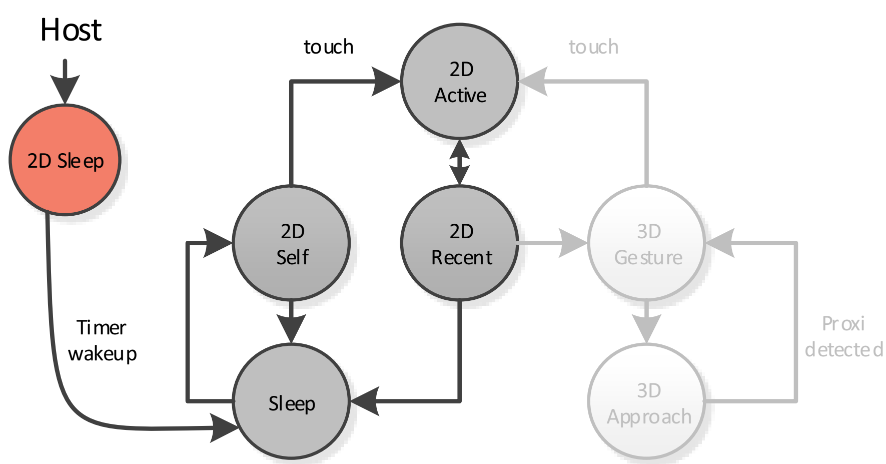


Note: Set to Active Standby mode

CMD: 0x82, 4

Safe mode to change multiple parameters without having any side effect on the code. No running code in this mode.

FIGURE 9-6: 2D SLEEP MODE



Note: Set to 2D Sleep mode

CMD: 0x82, 5

10.0 APPLICATION COMMANDS

TABLE 10-1: APPLICATION COMMANDS

CMD ID	Name	Description
0x04	ECHO	Echo back the received packet
0xfb	FORCE_BASELINE	Force the touch sensor to update its baseline measurements
0xff	QUERY_VERSION	Read the MTCH6303 firmware and application revisions
0x17	READ_FLASH	Read from a section in ROM
0x55	ENTER_BOOTLOADER	Enter Bootloader mode
0xfc	GESTIC_BRIDGE	Pass information through to an MGC3130
0xe0	SET_PARAMETER	Write a value to a register
0xe1	GET_PARAMETER	Read a value from a register

10.1 Command: ECHO

Host command to test communication. Host sends <04><01><02><03>, and the controller will respond with the exact same packet <04><01><02><03>. Any bytes following the 0x04 ID byte will not be processed by the controller, and should only be used to verify communication is working properly.

TABLE 10-2: COMMAND: ECHO

Byte	Value	Description
1	n+1	Length, # of bytes to follow
2	0X04	Command ID
3-n	Packet[0]:Packet[n]	Test packet information for confirmation

TABLE 10-3: COMMAND RESPONSE: ECHO

Byte	Value	Description
1	n+1	Length, # of bytes to follow
2	0X04	Command ID
3-n	Packet[0]:Packet[n]	Identical test packet for confirmation

10.2 Command: FORCE_BASELINE

Forces the controller to update touch sensor baseline measurements.

TABLE 10-4: COMMAND: FORCE_BASELINE

Byte	Value	Description
1	0x01	Length, # of bytes to follow
2	0Xfb	Command ID

TABLE 10-5: COMMAND RESPONSE: FORCE_BASELINE

Byte	Value	Description
1	0x02	Length, # of bytes to follow
2	0Xf0	Acknowledge CMD ID
3	0xfb	Repeat FORCE_BASELINE Command ID

TABLE 10-6: FORCE_BASELINE EXAMPLE

SEND	
0x01	0xfb
Length	CMD ID

RECEIVE		
0x02	0xf0	0xfb
Length	ACK CMD ID	Repeat FORCE_BASELINE ID

10.3 Command: QUERY_VERSION

The QUERY_VERSION command will read the MTCH6303 firmware and application revisions. Sending a QUERY_VERSION command while in Application mode will prompt two packets to be returned from the MTCH6303. The first packet will contain the 128 bytes of version data, and the second packet is the acknowledgment of the QUERY_VERSION command.

TABLE 10-7: COMMAND: QUERY_VERSION

Byte	Value	Description
1	0x01	Length, # of bytes to follow
2	0Xff	Command ID

TABLE 10-8: COMMAND RESPONSE: QUERY_VERSION

Byte	Value	Description
1	0x80	Length, # of bytes to follow
2		128 bytes of version information
130		
1	0x02	Length, # of bytes to follow
2	0Xf0	Acknowledge CMD ID
3	0xff	Repeat QUERY_VERSION Command ID

10.4 Command: READ_FLASH

Read from a section in ROM. When used as an application command, the controller will respond with two packets: one containing the read data, and a second acknowledgment packet.

TABLE 10-9: COMMAND: READ_FLASH

Byte	Value	Description
1	0x07	Length, # of bytes to follow
2	0x17	Command ID
3	addr[7:0]	4-byte (32-bit) Start address
4	addr[15:8]	
5	addr[23:16]	
6	addr[31:24]	
7	size[7:0]	Length of Flash block to read, in Bytes
8	size[15:8]	

TABLE 10-10: COMMAND RESPONSE: READ_FLASH

Byte	Value	Description
1	0x05	Length, # of bytes to follow
2	0x17	Command ID
3-[size]	Data	[size] number of bytes of data, as requested in command, starting at Start address
1	0x02	Length, # of bytes to follow
2	0Xf0	Acknowledge CMD ID
3	0x17	Repeat GET_REGISTER CMD ID

10.5 Command: ENTER_BOOTLOADER

TABLE 10-11: COMMAND: ENTER_BOOTLOADER

Byte	Value	Description
1	0x01	Length, # of bytes to follow
2	0X55	Command ID

TABLE 10-12: COMMAND RESPONSE: ENTER_BOOTLOADER

Byte	Value	Description
1	0x02	Length, # of bytes to follow
2	0Xf0	Acknowledge CMD ID
3	0x55	Repeat ENTER_BOOTLOADER command ID

10.6 Command: GESTIC_BRIDGE

Use GESTIC_BRIDGE to pass information through the MTCH6303 to the MGC3130 controller.

TABLE 10-13: COMMAND: GESTIC_BRIDGE

Byte	Value	Description
1	n+1	Length, # of bytes to follow
2	0Xfc	Command ID
3-n	Packet[0]:Packet[n]	Packets to send to MGC3130

TABLE 10-14: COMMAND RESPONSE: GESTIC_BRIDGE

Byte	Value	Description
1	n+1	Length, # of bytes to follow
2	0Xf0	Acknowledge CMD ID
3-n	Packet[0]: Packet[n]	Packets to send to MGC3130

10.7 Register Commands

There are a number of parameter registers that can be configured to modify the performance of the MTCH6303. Table details a list of all modifiable registers.

TABLE 10-15: COMMAND: SET_REGISTER

Byte	Value	Description
1	0x0b	Length, # of bytes to follow
2	0xe0	Command ID
3	addr[7:0]	2-byte (16-bit) Register Address
4	addr[15:8]	
5	value[7:0]	4-byte (32-bit) register value to be written
6	value[15:8]	
7	value[23:16]	
8	value[31:24]	
9	mask[7:0]	4-byte (32-bit) value to mask register value to be written
10	mask[15:8]	
11	mask[23:16]	
12	mask[31:24]	

TABLE 10-16: COMMAND: SET_REGISTER RESPONSE

Byte	Value	Description
1	0x02	Length, # of bytes to follow
2	0Xf0	Acknowledge CMD ID
3	0xe0	Repeat SET_REGISTER Command ID

TABLE 10-17: SET_REGISTER 0X0004 TO VALUE 0XAABBCCDD EXAMPLE

SEND											
0x0b	0xe0	0x04	0x00	0xdd	0xcc	0xbb	0xaa	0xff	0xff	0xff	0xff
Length	CMD ID	Register 0x0004		New Register Value 0xaabbccdd				Register Bit Mask			

RECEIVE		
0x02	0xf0	0xe0
Length	ACK CMD ID	Repeat SET_REGISTER ID

10.8 Command: GET_REGISTER

The MTCH6303 will respond with two packets when issued the GET_REGISTER command. The first packet will contain the data, and the second packet is the acknowledgment of the GET_REGISTER command.

TABLE 10-18: COMMAND: GET_RESGISTER

Byte	Value	Description
1	0x03	Length, # of bytes to follow
2	0xe1	Command ID
3	addr[7:0]	2-byte (16-bit) Register Address

TABLE 10-19: COMMAND: GET_REGISTER RESPONSE

Byte	Value	Description
1	0x05, 0x06, or 0x07	Length, # of bytes to follow
2	0xe1	Command ID
3	addr[7:0]	2-byte (16-bit) Register Address
4	addr[15:8]	
5	value[7:0]	Up to 4 bytes (32bit) of value data, depending on register
6	value[15:8]	
7	value[23:16]	
8	value[31:24]	

1	0x02	Length, # of bytes to follow
2	0Xf0	Acknowledge CMD ID
3	0xe1	Repeat GET_REGISTER CMD ID

TABLE 10-20: GET_REGISTER VALUE 0X00CC AT 0X0004 EXAMPLE

SEND			
0x03	0xe1	0x04	0x00
Length	CMD ID	Register 0x0004	

RECEIVE					
0x05	0xe1	0x04	0x00	0xcc	0x00
Length	CMD ID	Register 0x0004		Data	

0x02	0xf0	0xe1
Length	ACK CMD ID	Repeat GET_REGISTER CMD ID

TABLE 10-21: MTCH6303 PARAMETER REGISTERS

Register #	Name	Description	Default Value
0x0040	numOfAvg		
0x0041	flipState	bit 0 = X flip, bit 1 = Y flip, bit 2 = X/Y swap	
0x0080	diagMask		
0x0081	activeModules		
0x0082	streamingMode		
0x0100	numberOfXChannels	Number of RX Channels along long/ wide axis of touch screen	
0x0101	numberOfYChannels	Number of TX Channels along short/ narrow axis of touch screen	
0x0200 - 0x021a	rxPinMap[0] - rxPinMap[26]		
0x0280 - 0x02a3	txPinMap[0] - txPinMap[35]		
0x0400	selfTouchThres		
0x0401	mutTouchThres		
0x0402	weightThreshold	Limits the max distance a touch can travel between frames before assigning a new ID (native position units)	
0x0403	penDownTimer		
0x0404	penUpTimer		
0x0405	largeActThres		
0x0480	minCuspDelta	Slope value must be above this to determine that a ‘peak’ has been found	
0x0500	swipeTimeout		
0x0501	swipeDistance		
0x0502	swipeBorder.left		
0x0503	swipeBorder.right		
0x0504	swipeBorder.top		
0x0505	swipeBorder.bottom		
0x0540	tapBorder.left		
0x0541	tapBorder.right		
0x0542	tapBorder.top		
0x0543	tapBorder.bottom		
0x0800	diagRxChannel		
0x0801	diagTxChannel		
0x0802	baseUpdateTime	Stopwatch time for baseline counter, no touch for this duration will engage a recalibration	
0x0803 - 0x0806	mutScanPeriode[0] - mutScanPeriode[3]		

TABLE 10-21: MTCH6303 PARAMETER REGISTERS (CONTINUED)

Register #	Name	Description	Default Value
0x0807 - 0x080a	mutScanPhase[0] - mutScanPhase[3]		
0x080b	mutFreqHopping	If >0, selects Fixed Frequency mode (indexed by this value). If 0, all frequencies are in use	
0x080c - 0x080f	mutFreqHoppingLevel[0] - mutFreqHoppingLevel[3]	Provides a software gain for frequencies that provide smaller amplitude than normal. (0 = none)	
0x0810	selfSampleTime		
0x0811	mutSampleTime		
0x0812 - 0x0815	selfScanPhase[0] - selfScanPhase[3]		
0x0816 - 0x0819	selfScanISRPhase[0] - selfScanISRPhase[3]		
0x081a	syncRxChannel		
0x081b	syncTxChannel		
0x081c	fullScanRxStart		
0x081d	fullScanRxStop		
0x081e	fullScanTxStart		
0x081f	fullScanTxStop		
0x0900	mtch65x_periode_fast_rise	Period for TMR2 (pwm for 652)	
0x0901	mtch65x_periode_-fast_rise_oc	Sets OC1 for TMR2 duty cycle - divide “fast_rise” by this number to calculate DC	
0x0902	mtch65x_fast_rise_delay	TMR1 counts to wait until full boost is established	
0x0905 - 0x0908	mtch65x_periode_mutu_measurement[0] - mtch65x_periode_mutu_measurement[3]		
0x0909 - 0x090c	mtch65x_periode_mutu_measurement_oc[0] - mtch65x_periode_mutu_measurement_oc[3]		
0x090d - 0x0910	mtch65x_periode_self_measurement[0] - mtch65x_periode_self_measurement[3]		
0x0911 - 0x0914	mtch65x_periode_self_measurement_oc[0] - mtch65x_periode_self_measurement_oc[3]		

11.0 GESTURE FEATURES AND PARAMETERS

To simplify touch-based application development the controller already includes the capability to recognize a fixed set of touch gestures. The gesture recognizer supports the following kinds of gestures:

- Swipe-Gestures
- Scroll-Gestures
- Tap-Gestures

The gesture recognizer in the MTCH6303 is generic in that it supports those gestures for any number of fingers greater or equal to one. In practice, the maximum number of fingers is still limited because of the following two other factors:

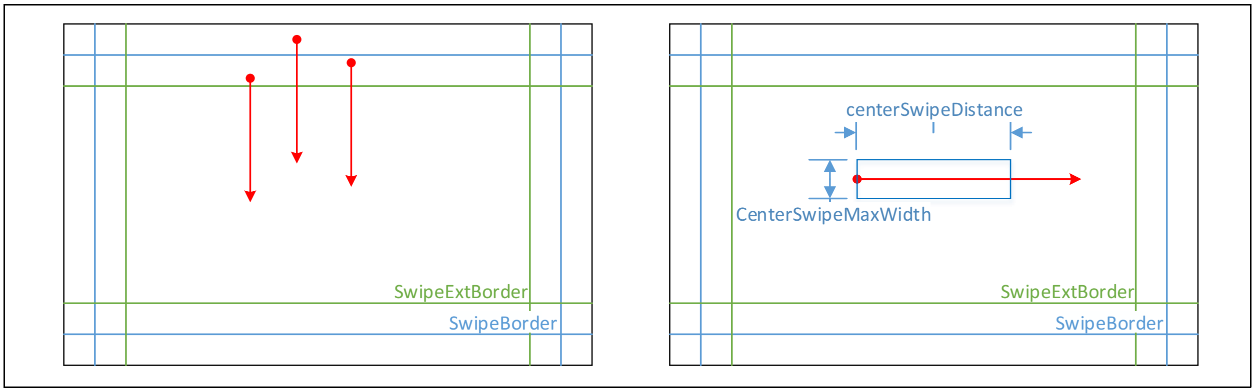
1. The number of concurrent finger contacts the touch digitizer stages of the MTCH6303 is able to trace. This is currently internally limited to 10 although HID only reports a maximum of 5.
2. Ergonomic considerations also play a role: e.g., on a 3.7-inch touch surface the user would be hard pressed to correctly perform a five finger gesture.

In order for gestures to be recognized, the gesture recognition module has to be enabled and in order to output results the gesture bit has to be set in the diagnostic mask (see **Section 7.0, Sensor Design Considerations**).

11.1 Swipe Gestures

The MTCH6303 can detect two different types of swipes: swipes starting from the edge and then going towards the center of the touch surface and swipes within the central area of the sensor. The criteria used to decide when a user's movement should be considered an edge swipe are illustrated in Figure 11-1 below.

FIGURE 11-1: SWIPE GESTURES



1. The user starts touching the surface with "n" fingers, where n is any number greater than zero. In the illustration above the user touches the surface with three fingers on the northern edge of the device.
2. The gesture recognizer checks if all fingers started on the same edge of the device. It does so in a tolerant way: i.e., if two fingers were on the northern edge, but a third finger was in the north-east corner, it would consider it to be on the northern edge. If all fingers did not start on the same edge, the gesture recognizer will abort the swipe detection at this stage.
3. To distinguish between the edge and the inner area of the surface the gesture recognizer uses two parameters: SwipeBorder and SwipeExtBorder. The two parameters are necessary for the following reasons:

- In order to avoid the user accidentally starting a swipe gesture it is desirable to make the border area as narrow as possible.
- However when touching with multiple fingers it is hard for the user to align all of them sufficiently within a narrow band. Typically the contact points, when touching with multiple fingers, would be on a slight curve, not a straight line.
- To resolve this dilemma, between having narrow border area to avoid accidental swipes and having a wider one to allow for easier use with multiple fingers, the recognizer utilizes two border areas. One, SwipeBorder, having smaller borders and another one, SwipeExtBorder, with bigger borders, which is inside of it. The recognizer then only requires one finger to start within the narrower borders described by SwipeBorder, all the other fingers are allowed to start within the wider borders from SwipeExtBorder to be considered on an edge.

4. If the controller determines that the start position is not within the edge, the recognizer checks for a center swipe to have occurred. If all fingers have moved beyond a certain distance (parameter centerSwipeDistance) and stayed within a certain range of horizontal or vertical (parameter centerSwipeMaxWidth), then a center swipe message is generated.

5. If the edge criteria have been fulfilled, the user has to move all fingers towards the center. The recognizer checks if all fingers have moved beyond a certain distance (parameter SwipeDistance) within a specific timeout (parameter SwipeTimeout). For swipes starting from the northern or the southern edge, only the vertical distance is considered, while for swipes starting from the western or eastern edge only the horizontal distance is considered. Once the user has moved all touching fingers beyond the distance threshold, a swipe is reported, unless the timeout has expired. The gesture recognizer then stops the swipe detection until the user has removed all fingers from the surface and starts touching again.

TABLE 11-1: PARAMETERS ASSOCIATED WITH SWIPE GESTURES

ID	Name	Type	Default (Sensor: 10137_100h)
0x0500	SwipeTimeout	uint32	234375 (1.5s)
0x0501	SwipeDistance	uint16	1024
0x0502	SwipeBorder.Left	uint16	768
0x0503	SwipeBorder.Right	uint16	4352
0x0504	SwipeBorder.Top	uint16	768
0x0505	SwipeBorder.Bot-tom	uint16	8448
0x0506	SwipeExtBorder.Left	uint16	1536
0x0507	SwipeExtBor-der.Right	uint16	3484
0x0508	SwipeExtBorder.Top	uint16	1536
0x0509	SwipeExtBor-der.Bottom	uint16	7680
0x050A	centerSwipeDis-tance	uint16	1024
0x050B	centerSwipeMax-Width	uint16	512

The unit for the timeout is in 1s/156250, so a value of 234375 corresponds to 1.5 seconds.

Distance and border are in units of internal digitizer resolution, without any coordinate transformation such as scaling or flipping applied.

TABLE 11-2: MESSAGE OUTPUT FOR SWIPE GESTURES

ID	Payload	
0xA0	Flags	Fingers
byte	Uint8	Uint8

ID: A0

Payload:

uint8 flags; // flags describing the swipe

uint8 fingers; // number of fingers which participated in the swipe.

Flags is a bitmask. It contains currently only one of the following values (in theory they are logically or-ed together, but practically a swipe is only from one edge, so they are mutually exclusive).

EDGE_N 0x01 // swipe started at northern edge

EDGE_E 0x02 // … eastern…

EDGE_S 0x04 // … southern…

EDGE_W 0x08 // … western…

SWIPE_SOUTH 0x10 // center swipe moving south

SWIPE_WEST 0x20 // center swipe moving west

SWIPE_NORTH 0x40 // center swipe moving north

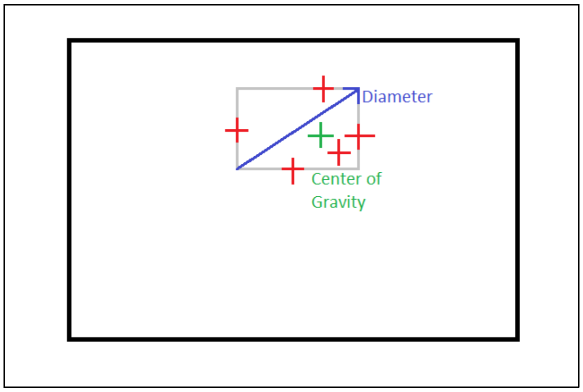
SWIPE_EAST 0x80 // center swipe moving east

Note:	Edge swipes are named by the edge the swipe starts from. Center swipes are named by the direction of travel.

11.2 Scroll Gestures

The gesture recognizer takes the incoming data of the moving fingers and derives additional data from them which can be used in an application to generate responses such as scroll, zoom and other gestures which depend on sliding fingers. As soon as the user touches the surface with n fingers, with n being any number greater than zero, the gesture recognizer continuously calculates the center of the touching points as well as the length of the diagonal of the bounding box around those points, as illustrated by the following drawing.

FIGURE 11-2: SCROLL GESTURES



As can be seen, the bounding box is orthogonal to the coordinate system of the surface, the edges being horizontal and vertical. There is no attempt to find a smaller bounding box for the surface contact points, which would be rotated against the surface. For two fingers the length of the diagonal of the bounding box is incidentally equal to the distance between those two fingers and the center of gravity would equal the midpoint between the two. For three or more fingers no such obvious interpretation is available, but since the bounding box circumscribes all touching fingers the behavior of the diagonal measure when spreading out those fingers on the surface or contracting them is often quite usable, depending on the application. The computation also runs when the user touches with only a single finger.

There are no parameters associated with scroll gestures.

TABLE 11-3: MESSAGE OUTPUT FOR SCROLL GESTURES

ID	Payload				
0xA1	Fingers	diamHi	Diam	centerX	center
byte	Uint8	Uint8	Uint16	Uint16	Uint16

ID: 0xA1

Payload:

uint8 fingers; // number of fingers touching

uint8 diamHi; // bits 16 to 23 of diameter

uint16 diam; // bits 0 to 15 of diameter

uint16 centerX; // x of center of gravity

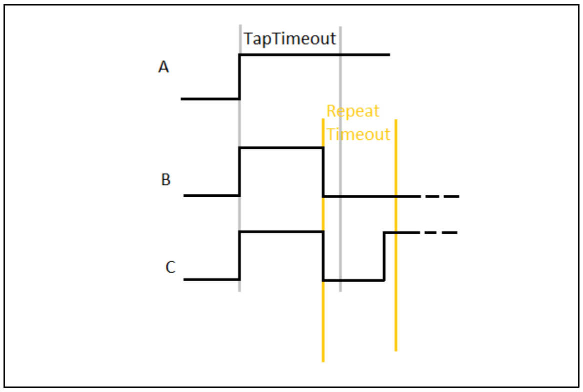
uint16 centerY; // y of center of gravity

Currently diamHi can be ignored. It is here for future compatibility.

11.3 Tap Gestures

There is also support in the gesture recognizer to detect short taps on the surface. The implemented algorithm also supports detection of repeated taps as well as taps with multiple fingers.

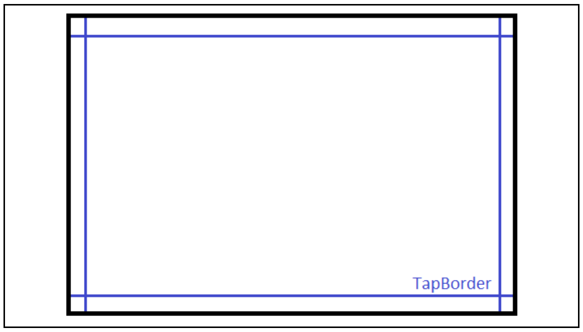
FIGURE 11-3: TAP RECOGNIZER



Once the user contacts the surface a timer starts running. If the user does not lift their finger(s) from the surface before TapTimeout has expired (as in part A of the drawing), the tap is considered aborted and the recognizer stops further processing. If the user lifts the finger without exceeding the timeout (as in part B and C), the recognizer considers it a tap.

If the gesture recognizer has recognized a tap a second timeout, which is used to distinguish repeated taps from single taps and is called RepeatTimeout, starts running. If the user does not touch the surface before the timer has expired (part B), the next tap will not be considered a repeated tap. Otherwise, if he does touch the surface while the timeout has not expired, the following touch will be indicated as a repeated tap.

FIGURE 11-4: TAPBOARDER PARAMETER



At least one finger has to be inside the area enclosed by TapBorder to start tap recognition, while the other fingers may be either outside or inside. So for a one finger tap the finger has to be inside, while for a two finger tap, one finger must be inside, while the second can be either inside or outside.

This is done because a user moving around at the edges of the surface might enter and leave the actual area where touch is detected for very short moments and thereby trigger accidental tap events. By requiring at least one finger to be inside a smaller area, this can be prevented.

TABLE 11-4: PARAMETERS ASSOCIATED WITH THE TAP RECOGNIZER

ID	Name	Type	Default (Sensor: 10137_100h)
0x0540	TapBorder.Left	uint16	256
0x0541	TapBorder.Right	uint16	4864
0x0542	TapBorder.Top	uint16	256
0x0543	TapBorder.Bottom	uint16	8960
0x0544	TapTimeout	uint32	31250 (200ms)
0x0545	RepeatTimeout	uint32	78125 (500ms)

The unit for the timeouts is 1s/156250, so a value of 31250 corresponds to 200 milliseconds and a value of 78125 to 500 milliseconds.

The border is in units of internal digitizer resolution, without any coordinate transformation such as scaling or flipping applied.

TABLE 11-5: MESSAGE OUTPUT FOR SWIPE GESTURES

ID	Payload	
0xA2	Flags	Fingers
byte	Uint8	Uint8

ID: A2

Payload:

uint8 flags; // flags with details about the tap

uint8 fingers; // number of fingers for this tap.

The flags field is a bitmask where the following values could be logically or-ed together:

TAPPED 0x01

ABORTED 0x02

NOREPEAT 0x04

REPEAT 0x08

EQFINGERS 0x10

If TAPPED is set it means that a tap has occurred, in that case the “fingers” field contains the number of fingers used for this tap.

The ABORTED flag is set if tap recognition has been aborted because TapTimeout has expired (part A of previous drawing). In this case the “fingers” field is not valid.

The REPEAT/NOREPEAT flags can be used to distinguish between taps which happened within a timespan of RepeatTimeout from the previous tap, or from which the previous tap has been a longer while back.

The EQFINGERS flag indicates that the previous tap was performed with the same number of fingers touching the surface than the current one.

Having all those flags allows the user to decide whether to distinguish between repeated taps and single isolated taps or not to do so by either looking at the REPEAT/NOREPEAT flags or by ignoring them. Likewise, the user may decide if a repeated tap should only be considered a repeated tap if it was performed with the same number of fingers as the previous tap by simply looking at the EQFINGERS field in addition to the REPEAT field. If the user instead does not care if a repeated tap has been performed with a different number of fingers, he can ignore the EQFINGERS flag. That way the user can chose the behavior which fits the application best.

12.0 ELECTRICAL SPECIFICATIONS

Absolute Maximum Ratings for the MTCH6303 devices are listed below. Stresses above those listed under the Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions, above those indicated in the operation listings of this specification, is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

Absolute Maximum Ratings(†)

Ambient temperature under bias .... -40°C to +85°C

Storage temperature .... -65°C to +150°C

Voltage on pins with respect to VSS

- on VDD pin.. -0.3V to +4.0V
- on any pin that is not 5V tolerant(2).. -0.3V to (VDD + 0.3V)
- on any 5V tolerant pin when VDD ≥ 2.3V(2) .... -0.3V to +6.0V
- on any 5V tolerant pin when VDD < 2.3V(2) .... -0.3V to +3.6V

Voltage on D+ or D- pin with respect to VUSB3V3 .... -0.3V to (VUSB3V3 + 0.3V)

Voltage on VBUS with respect to VSS .... -0.3V to + 5.5V

Maximum current

- out of VSS pin(s).. 200 mA
- into VDD pin(s)(1).. 200 mA

Maximum output current

- sourced/sunk by any 4x I/O pin .... 15 mA
- sourced/sunk by any 8x I/O pin .... 25 mA

Maximum current sunk by all ports .... 150 mA

Maximum current sourced by all ports(1) .... 150 mA

Note 1: Maximum allowable current is a function of device maximum power dissipation (see Table 12-2)

2: See the Pin Diagram section for the 5V tolerant pins.

† **NOTICE:** Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operation listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.
This device is sensitive to ESD damage and must be handled appropriately. Failure to properly handle and protect the device in an application may cause partial to complete failure of the device.

12.1 DC Characteristics: MTCH6303

Rating	Min.	Typ.†	Max.	Units	Conditions
Supply Voltage	2.3	—	3.6	V	

TABLE 12-1: THERMAL OPERATING CONDITIONS

Rating	Symbol	Min.	Typical	Max.	Unit
Industrial Temperature Devices					
Operating Junction Temperature Range	T_J	-40	—	+125	°C
Operating Ambient Temperature Range	T_A	-40	—	+85	°C
Power Dissipation: Internal Chip Power Dissipation: $P_{INT} = V_{DD} \times (I_{DD} – S\ I_{OH})$ I/O Pin Power Dissipation: $I/O = S\ ((\{V_{DD} – V_{OH}\} \times I_{OH}) + S\ (V_{OL} \times I_{OL}))$	P_D	$P_{INT} + P_{I/O}$			W
Maximum Allowed Power Dissipation	P_{DMAX}	$(T_J – T_A)/\theta_{JA}$			W

TABLE 12-2: THERMAL PACKAGING CHARACTERISTICS

Characteristics	Symbol	Typical	Max.	Unit
Package Thermal Resistance, 64-pin QFN (9x9x0.9 mm)[(1)]	θ_{JA}	28	—	°C/W
Package Thermal Resistance, 64-pin TQFP (10x10x1 mm)[(1)]	θ_{JA}	47	—	°C/W

Note 1: Junction to ambient thermal resistance, Theta-JA (θ_{JA}) numbers are achieved by package simulations.

TABLE 12-3: DC TEMPERATURE AND VOLTAGE SPECIFICATIONS

DC CHARACTERISTICS			**Standard Operating Conditions: (unless otherwise stated)** **2.3V to 3.6V** Operating temperature -40°C ≤ T_A ≤ +85°C for Industrial				
Param. No.	**Symbol**	**Characteristics**	**Min.**	**Typical**	**Max.**	**Units**	**Conditions**
Operating Voltage							
DC10	V_{DD}	**Supply Voltage**	2.3	—	3.6	V	—
DC12	V_{DR}	**RAM Data Retention Voltage (Note 1)**	1.75	—	—	V	—
DC16	V_{POR}	**V_{DD} Start Voltage** to Ensure Internal Power-on Reset Signal	1.75	—	2.1	V	—
DC17	S_{VDD}	**V_{DD} Rise Rate** to Ensure Internal Power-on Reset Signal	0.00005	—	0.115	V/µs	—

Note 1: This is the limit to which V_{DD} can be lowered without losing RAM data.

TABLE 12-4: DC CHARACTERISTICS: I/O PIN INPUT SPECIFICATIONS

DC CHARACTERISTICS			**Standard Operating Conditions: (unless otherwise stated) 2.3V to 3.6V** Operating temperature -40°C ≤ T_A ≤ +85°C for Industrial				
Param. No.	**Symb.**	**Characteristics**	**Min.**	**Typ.[1]**	**Max.**	**Units**	**Conditions**
	V_{IL}	**Input Low Voltage**					
DI18		I/O Pins	V_{SS}	—	0.2 V_{DD}	V	
		SDAx, SCLx	V_{SS}	—	0.3 V_{DD}	V	
DI19		SDAx, SCLx	V_{SS}	—	0.8	V	
	V_{IH}	**Input High Voltage**					
		I/O Pins 5V-tolerant with PMP[4]	0.25 V_{DD} + 0.8V	—	5.5	V	**(Note 5)**
		I/O Pins 5V-tolerant[4]	0.65 V_{DD}	—	5.5	V	
DI28		SDAx, SCLx	0.65 V_{DD}	—	5.5	V	
DI29		SDAx, SCLx	2.1	—	5.5	V	
	I_{IL}	**Input Leakage Current[3]**					
DI50		I/O Ports	—	—	±1	μA	V_{SS} ≤ V_{PIN} ≤ V_{DD}, Pin at high-impedance
DI51		Analog Input Pins	—	—	±1	μA	V_{SS} ≤ V_{PIN} ≤ V_{DD}, Pin at high-impedance
DI55		$\overline{MCLR}$[2]	—	—	±1	μA	V_{SS} ≤ V_{PIN} ≤ V_{DD}

Note 1: Data in “Typical” column is at 3.3V, 25°C unless otherwise stated. Parameters are for design guidance only and are not tested.

2: The leakage current on the $\overline{MCLR}$ pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.

3: Negative current is defined as current sourced by the pin.

4: See the **Pin Diagram** section for the 5V tolerant pins.

5: The V_{IH} specifications are only in relation to externally applied inputs, and not with respect to the user-selectable internal pull-ups. External open-drain input signals utilizing the internal pull-ups of the PIC32 device are ensured to be recognized only as a logic “high” internally to the PIC32 device, provided that the external load does not exceed the minimum value of I_{CNPU}. For External “input” logic inputs that require a pull-up source, to ensure the minimum V_{IH} of those components, it is recommended to use an external pull-up resistor rather than the internal pull-ups of the PIC32 device.

6: V_{IH} source > (V_{DD} + 0.3) for non-5V tolerant pins only.

7: Digital 5V tolerant pins do not have an internal high side diode to V_{DD}, and therefore, cannot tolerate any “positive” input injection current.

8: Injection currents > | 0 | can affect the ADC results by approximately 4 to 6 counts (i.e., V_{IH} Source > (V_{DD} + 0.3) or V_{IL} source < (V_{SS} - 0.3)).

9: Any number and/or combination of I/O pins not excluded under I_{ICL} or I_{ICH} conditions are permitted provided the “absolute instantaneous” sum of the input injection currents from all pins do not exceed the specified limit. If **Note 7**, I_{ICL} = (((V_{SS} - 0.3) - V_{IL} source) / R_S). If **Note 8**, I_{ICH} = ((I_{ICH} source - (V_{DD} + 0.3)) / R_S). RS = Resistance between input source voltage and device pin. If (V_{SS} - 0.3) ≤ V_{SOURCE} ≤ (V_{DD} + 0.3), injection current = 0.

TABLE 12-4: DC CHARACTERISTICS: I/O PIN INPUT SPECIFICATIONS (CONTINUED)

DC CHARACTERISTICS			Standard Operating Conditions: (unless otherwise stated) 2.3V to 3.6V Operating temperature -40°C ≤ TA ≤ +85°C for Industrial				
Param. No.	**Symb.**	**Characteristics**	**Min.**	**Typ.[1]**	**Max.**	**Units**	**Conditions**
DI60a	I_{ICL}	**Input Low Injection Current**	0	—	-5[6,9]	mA	Pins with Analog functions. Exceptions: [N/A] = 0 mA max
							Digital 5V tolerant designated pins. Exceptions: [N/A] = 0 mA max
							Digital non-5V tolerant designated pins. Exceptions: [N/A] = 0 mA max

Note 1: Data in "Typical" column is at 3.3V, 25°C unless otherwise stated. Parameters are for design guidance only and are not tested.

2: The leakage current on the $\overline{MCLR}$ pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.

3: Negative current is defined as current sourced by the pin.

4: See the **Pin Diagram** section for the 5V tolerant pins.

5: The V_{IH} specifications are only in relation to externally applied inputs, and not with respect to the user-selectable internal pull-ups. External open-drain input signals utilizing the internal pull-ups of the PIC32 device are ensured to be recognized only as a logic "high" internally to the PIC32 device, provided that the external load does not exceed the minimum value of I_{CNPU}. For External "input" logic inputs that require a pull-up source, to ensure the minimum V_{IH} of those components, it is recommended to use an external pull-up resistor rather than the internal pull-ups of the PIC32 device.

6: V_{IH} source > (V_{DD} + 0.3) for non-5V tolerant pins only.

7: Digital 5V tolerant pins do not have an internal high side diode to V_{DD}, and therefore, cannot tolerate any "positive" input injection current.

8: Injection currents > | 0 | can affect the ADC results by approximately 4 to 6 counts (i.e., V_{IH} Source > (V_{DD} + 0.3) or V_{IL} source < (V_{SS} - 0.3)).

9: Any number and/or combination of I/O pins not excluded under I_{ICL} or I_{ICH} conditions are permitted provided the "absolute instantaneous" sum of the input injection currents from all pins do not exceed the specified limit. If **Note 7**, I_{ICL} = (((V_{SS} - 0.3) - V_{IL} source) / Rs). If **Note 8**, I_{ICH} = ((I_{ICH} source - (V_{DD} + 0.3)) / RS). RS = Resistance between input source voltage and device pin. If (V_{SS} - 0.3) ≤ V_{SOURCE} ≤ (V_{DD} + 0.3), injection current = 0.

TABLE 12-4: DC CHARACTERISTICS: I/O PIN INPUT SPECIFICATIONS (CONTINUED)

DC CHARACTERISTICS			Standard Operating Conditions: (unless otherwise stated) 2.3V to 3.6V Operating temperature -40°C ≤ TA ≤ +85°C for Industrial				
Param. No.	**Symb.**	**Characteristics**	**Min.**	**Typ.(1)**	**Max.**	**Units**	**Conditions**
DI60b	I_{ICH}	**Input High Injection Current**	0	—	+5(7,8,9)	mA	Pins with Analog functions. Exceptions: [SOSCI] = 0 mA max.
							Digital 5V tolerant designated pins (V_{IH} < 5.5V)(8). Exceptions: [All] = 0 mA max.
							Digital non-5V tolerant designated pins. Exceptions: [N/A] = 0 mA max.
DI60c	$\sum I_{ICT}$	**Total Input Injection Current (sum of all I/O and control pins)**	-20(11)	—	+20(9)	mA	Absolute instantaneous sum of all ± input injection currents from all I/O pins (| I_{ICL} + | I_{ICH} |) ≤ $\sum I_{ICT}$

Note 1: Data in “Typical” column is at 3.3V, 25°C unless otherwise stated. Parameters are for design guidance only and are not tested.

2: The leakage current on the MCLR pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.

3: Negative current is defined as current sourced by the pin.

4: See the **Pin Diagram** section for the 5V tolerant pins.

5: The V_{IH} specifications are only in relation to externally applied inputs, and not with respect to the user-selectable internal pull-ups. External open-drain input signals utilizing the internal pull-ups of the PIC32 device are ensured to be recognized only as a logic “high” internally to the PIC32 device, provided that the external load does not exceed the minimum value of I_{CNPU}. For External “input” logic inputs that require a pull-up source, to ensure the minimum V_{IH} of those components, it is recommended to use an external pull-up resistor rather than the internal pull-ups of the PIC32 device.

6: V_{IH} source > (V_{DD} + 0.3) for non-5V tolerant pins only.

7: Digital 5V tolerant pins do not have an internal high side diode to V_{DD}, and therefore, cannot tolerate any “positive” input injection current.

8: Injection currents > | 0 | can affect the ADC results by approximately 4 to 6 counts (i.e., V_{IH} Source > (V_{DD} + 0.3) or V_{IL} source < (V_{SS} - 0.3)).

9: Any number and/or combination of I/O pins not excluded under I_{ICL} or I_{ICH} conditions are permitted provided the “absolute instantaneous” sum of the input injection currents from all pins do not exceed the specified limit. If **Note 7**, I_{ICL} = (((V_{SS} - 0.3) - V_{IL} source) / R_S). If **Note 8**, I_{ICH} = ((I_{ICH} source - (V_{DD} + 0.3)) / R_S). R_S = Resistance between input source voltage and device pin. If (V_{SS} - 0.3) ≤ V_{SOURCE} ≤ (V_{DD} + 0.3), injection current = 0.

TABLE 12-5: DC CHARACTERISTICS: I/O PIN OUTPUT SPECIFICATIONS

DC CHARACTERISTICS			Standard Operating Conditions: (unless otherwise stated) 2.3V to 3.6V Operating temperature -40°C ≤ T_A ≤ +85°C for Industrial				
Param.	**Symbol**	**Characteristic**	**Min.**	**Typ.**	**Max.**	**Units**	**Conditions**
DO10	V_{OL}	**Output Low Voltage** I/O Pins: 4x Sink Driver Pins – All I/O output pins not defined as 8x Sink Driver pins	—	—	0.4	V	I_{OL} ≤ 9 mA, V_{DD} = 3.3V
		Output Low Voltage I/O Pins: 8x Sink Driver Pins – RC15, RD2, RD10, RF6, RG6	—	—	0.4	V	I_{OL} ≤ 15 mA, V_{DD} = 3.3V
DO20	V_{OH}	**Output High Voltage** I/O Pins: 4x Source Driver Pins – All I/O output pins not defined as 8x Source Driver pins	2.4	—	—	V	I_{OH} ≥ -10 mA, V_{DD} = 3.3V
		Output High Voltage I/O Pins: 8x Source Driver Pins – RC15, RD2, RD10, RF6, RG6	2.4	—	—	V	I_{OH} ≥ -15 mA, V_{DD} = 3.3V
DO20A	V_{OH1}	**Output High Voltage** I/O Pins: 4x Source Driver Pins – All I/O output pins not defined as 8x Sink Driver pins	1.5(1)	—	—	V	I_{OH} ≥ -14 mA, V_{DD} = 3.3V
			2.0(1)	—	—		I_{OH} ≥ -12 mA, V_{DD} = 3.3V
			3.0(1)	—	—		I_{OH} ≥ -7 mA, V_{DD} = 3.3V
		Output High Voltage I/O Pins: 8x Source Driver Pins – RC15, RD2, RD10, RF6, RG6	1.5(1)	—	—	V	I_{OH} ≥ -22 mA, V_{DD} = 3.3V
			2.0(1)	—	—		I_{OH} ≥ -18 mA, V_{DD} = 3.3V
			3.0(1)	—	—		I_{OH} ≥ -10 mA, V_{DD} = 3.3V

Note 1: Parameters are characterized, but not tested.

TABLE 12-6: ELECTRICAL CHARACTERISTICS: BOR

DC CHARACTERISTICS			Standard Operating Conditions: (unless otherwise stated) 2.3V to 3.6V Operating temperature -40°C ≤ T_A ≤ +85°C for Industrial				
Param. No.	**Symbol**	**Characteristics**	**Min.(1)**	**Typical**	**Max.**	**Units**	**Conditions**
BO10	V_{BOR}	BOR Event on V_{DD} transition high-to-low	2.0	—	2.3	V	—

Note 1: Parameters are for design guidance only and are not tested in manufacturing.

12.2 AC Characteristics and Timing Parameters

The information contained in this section defines MTCH6303 AC characteristics and timing parameters.

FIGURE 12-1: LOAD CONDITIONS FOR DEVICE TIMING SPECIFICATIONS

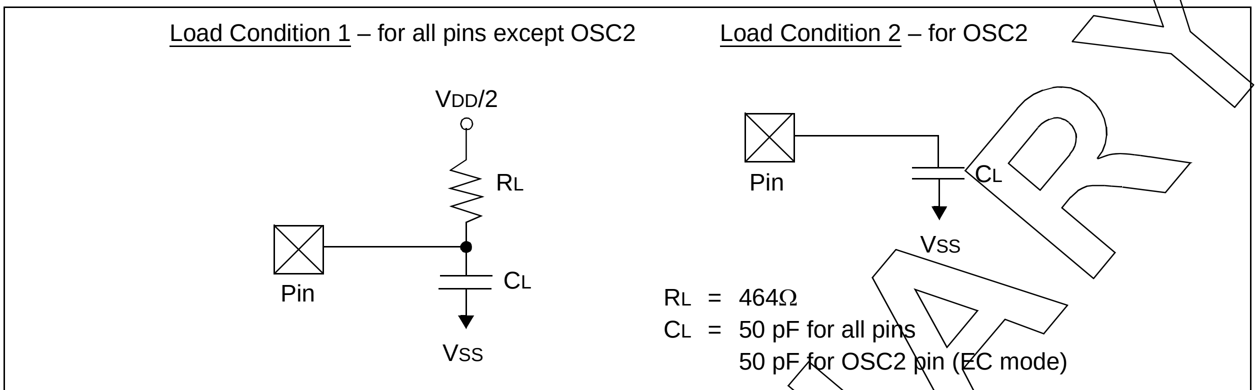


TABLE 12-7: CAPACITIVE LOADING REQUIREMENTS ON OUTPUT PINS

AC CHARACTERISTICS			Standard Operating Conditions: (unless otherwise stated) 2.3V to 3.6V Operating temperature -40°C ≤ TA ≤ +85°C for Industrial				
Param. No.	**Symbol**	**Characteristics**	**Min.**	**Typical(1)**	**Max.**	**Units**	**Conditions**
DO50	COSCO	OSC2 pin	—	—	15	pF	In XT and HS modes when an external crystal is used to drive OSC1
DO50a	CSOSC	SOSCI/SOSCO pins	—	33	—	pF	Epson P/N: MC-306 32.7680K-A0:ROHS
DO56	CIO	All I/O pins and OSC2	—	—	50	pF	EC mode
DO58	CB	SCLx, SDAx	—	—	400	pF	In I²C™ mode

Note 1: Data in "Typical" column is at 3.3V, 25°C unless otherwise stated. Parameters are for design guidance only and are not tested.

FIGURE 12-2: EXTERNAL CLOCK TIMING

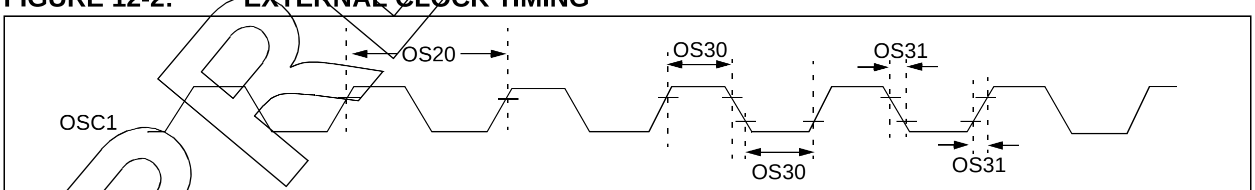
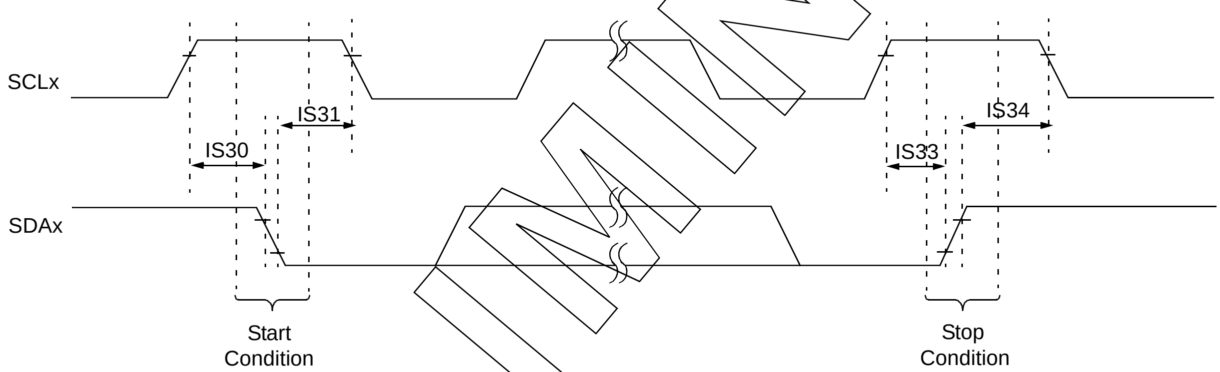


TABLE 12-8: EXTERNAL CLOCK TIMING REQUIREMENTS

AC CHARACTERISTICS			Standard Operating Conditions: (unless otherwise stated) 2.3V to 3.6V Operating temperature -40°C ≤ TA ≤ +85°C for Industrial				
Param. No.	**Symbol**	**Characteristics**	**Min.**	**Typical(1)**	**Max.**	**Units**	**Conditions**
OS11	FOSC	Oscillator Crystal Frequency	—	8	—	MHz	XT **(Note)**
OS20	TOSC	TOSC = 1/FOSC = TCY (1)	—	—	—	—	See parameter OS10 for FOSC value
OS41	TFSCM	Primary Clock Fail Safe Time-out Period	—	2	—	ms	**(Note)**
OS42	GM	External Oscillator Transconductance (Primary Oscillator only)	—	12	—	mA/V	VDD = 3.3V, TA = +25°C **(Note)**

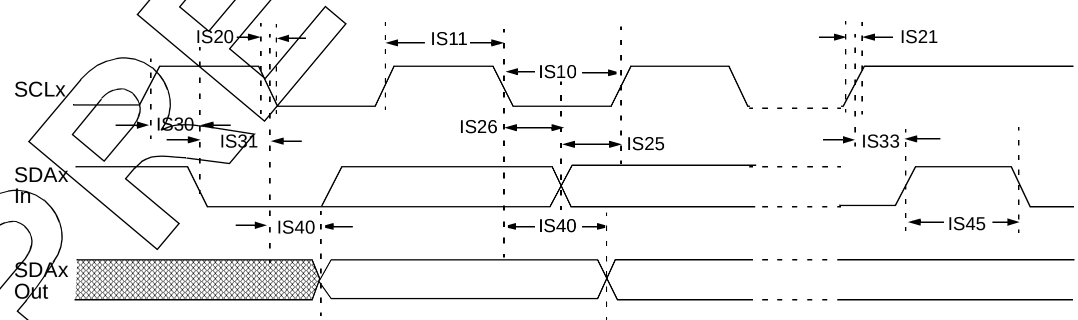
Note 1: The external clock is required for USB operation and not needed for I²C™ operation.

FIGURE 12-3: I²C™ BUS START/STOP BITS TIMING CHARACTERISTICS



Note: Refer to Figure 12-1 for load conditions.

FIGURE 12-4: I²C™ BUS DATA TIMING CHARACTERISTICS



Note: Refer to Figure 12-1 for load conditions.

TABLE 12-9: I²C™ BUS DATA TIMING REQUIREMENTS

AC CHARACTERISTICS				Standard Operating Conditions: (unless otherwise stated) 2.3V to 3.6V Operating temperature -40°C ≤ TA ≤ +85°C for Industrial			
Param. No.	Symbol	Characteristics		Min.	Max.	Units	Conditions
IS10	TLO:SCL	Clock Low Time	100 kHz mode	4.7	—	µs	PBCLK must operate at a minimum of 800 kHz
			400 kHz mode	1.3	—	µs	PBCLK must operate at a minimum of 3.2 MHz
IS11	THI:SCL	Clock High Time	100 kHz mode	4.0	—	µs	PBCLK must operate at a minimum of 800 kHz
			400 kHz mode	0.6	—	µs	PBCLK must operate at a minimum of 3.2 MHz
IS20	TF:SCL	SDAx and SCLx Fall Time	100 kHz mode	—	300	ns	CB is specified to be from 10 to 400 pF
			400 kHz mode	20 + 0.1 CB	300	ns	
IS21	TR:SCL	SDAx and SCLx Rise Time	100 kHz mode	—	1000	ns	CB is specified to be from 10 to 400 pF
			400 kHz mode	20 + 0.1 CB	300	ns	
IS25	TSU:DAT	Data Input Setup Time	100 kHz mode	250	—	ns	—
			400 kHz mode	100	—	ns	
IS26	THD:DAT	Data Input Hold Time	100 kHz mode	0	—	ns	—
			400 kHz mode	0	0.9	µs	
IS30	TSU:STA	Start Condition Setup Time	100 kHz mode	4700	—	ns	Only relevant for Repeated Start condition
			400 kHz mode	600	—	ns	
IS31	THD:STA	Start Condition Hold Time	100 kHz mode	4000	—	ns	After this period, the first clock pulse is generated
			400 kHz mode	600	—	ns	
IS33	TSU:STO	Stop Condition Setup Time	100 kHz mode	4000	—	ns	—
			400 kHz mode	600	—	ns	
IS34	THD:STO	Stop Condition Hold Time	100 kHz mode	4000	—	ns	—
			400 kHz mode	600	—	ns	
IS40	TAA:SCL	Output Valid from Clock	100 kHz mode	0	3500	ns	—
			400 kHz mode	0	1000	ns	
IS45	TBF:SDA	Bus Free Time	100 kHz mode	4.7	—	µs	The amount of time the bus must be free before a new transmission can start
			400 kHz mode	1.3	—	µs	
IS50	CB	Bus Capacitive Loading		—	400	pF	—

13.0 ORDERING INFORMATION

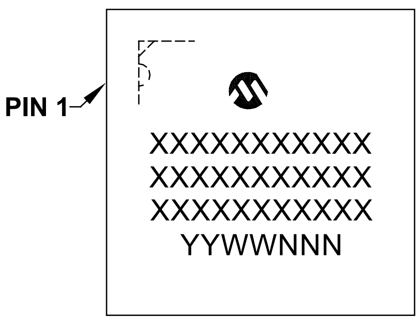
TABLE 13-1: ORDERING INFORMATION

Part Number	Pin Package	Packing
MTCH6303-I/PT	64-Lead TQFP (10x10mm)	Tray
MTCH6303-I/RG	64-Lead QFN (9x9mm)	Tube
MTCH6303T-I/PT	64-Lead TQFP (10x10mm)	T/R
MTCH6303T-I/RG	64-Lead QFN (9x9mm)	T/R

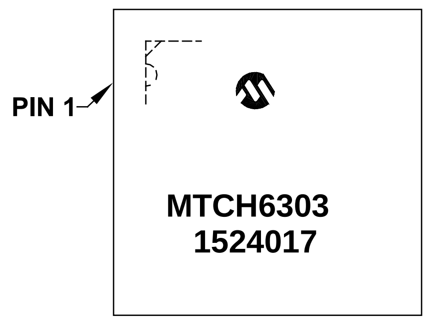
14.0 PACKAGING INFORMATION

14.1 Package Marking Information

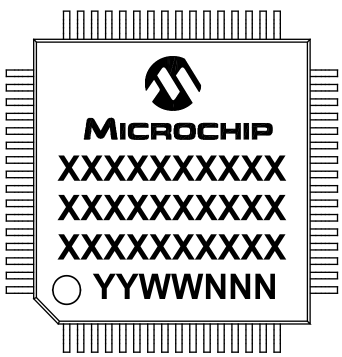
64-Lead QFN (9x9x0.9 mm)



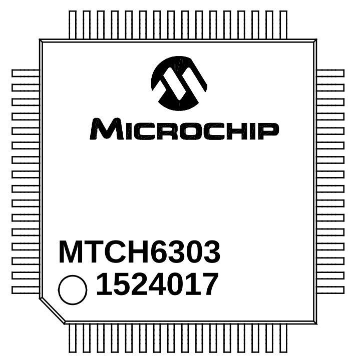
Example



64-Lead TQFP (10x10x1 mm)



Example



Legend:	XX...X	Customer-specific information
	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)	Pb-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.

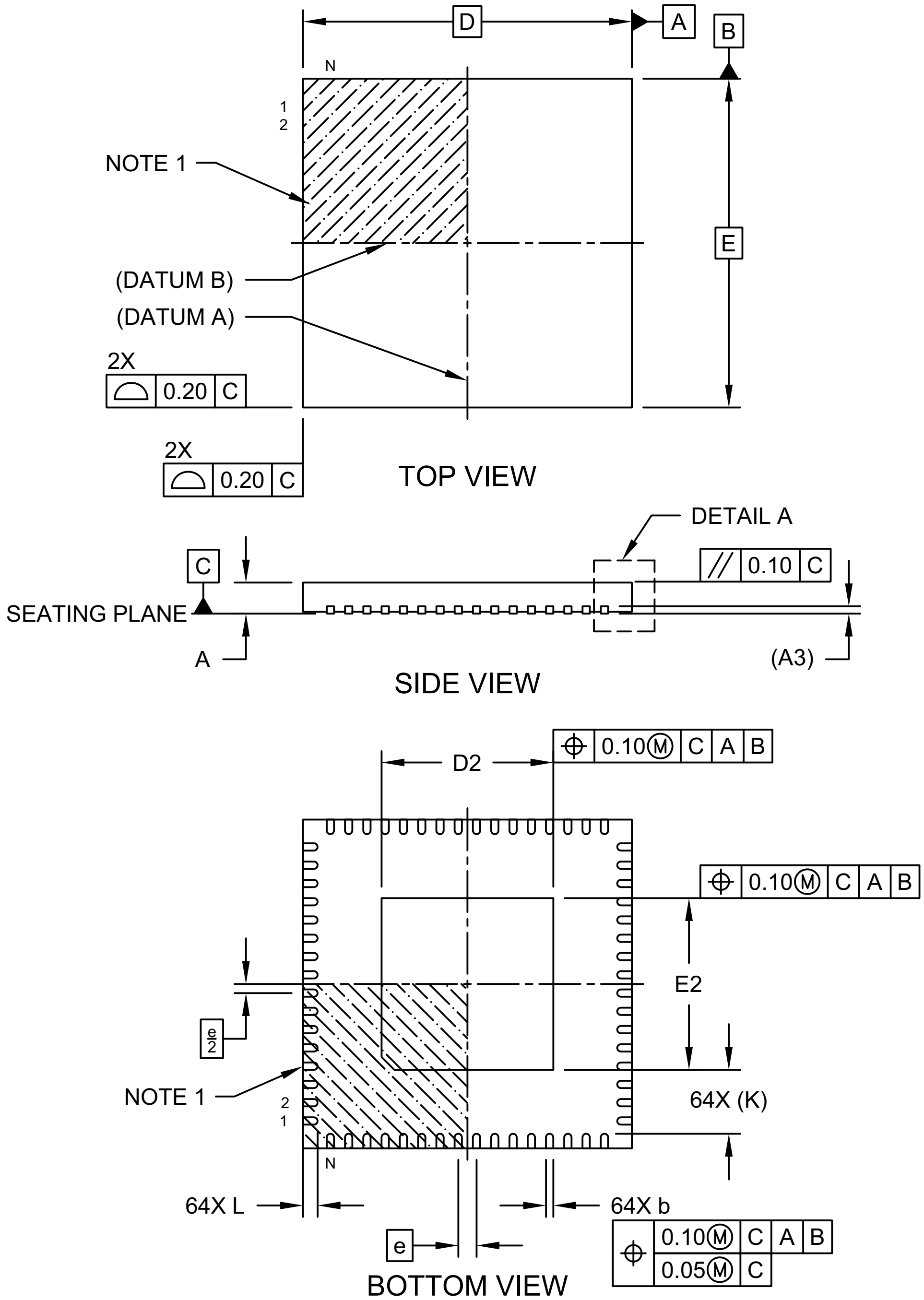
Note: In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line, thus limiting the number of available characters for customer-specific information.

14.2 Package Details

The following sections give the technical details of the packages.

64-Terminal Plastic Quad Flat Pack, No Lead (RG) 9x9x0.9 mm Body [QFN] Saw Singulated

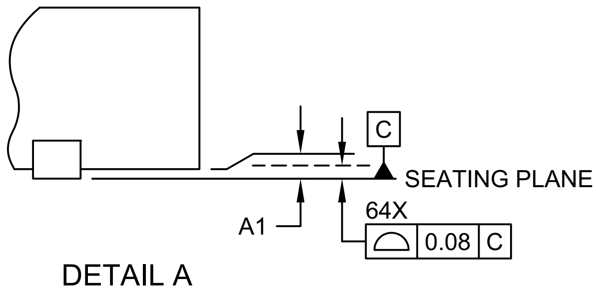
Note:	For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging

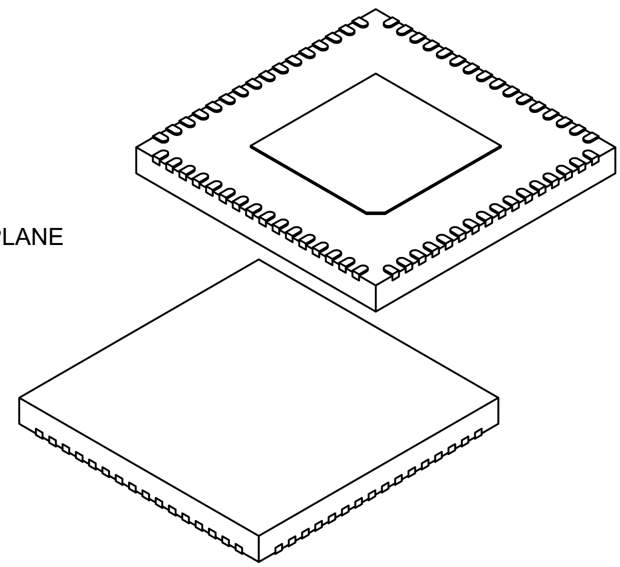


Microchip Technology Drawing C04-260A Sheet 1 of 2

64-Terminal Plastic Quad Flat Pack, No Lead (RG) 9x9x0.9 mm Body [QFN] Saw Singulated

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





Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Terminals	N		64	
Pitch	e		0.50 BSC	
Overall Height	A	0.80	0.85	0.90
Standoff	A1	0.00	0.02	0.05
Standoff	A3		0.20 REF	
Overall Width	E		9.00 BSC	
Exposed Pad Width	E2	4.60	4.70	4.80
Overall Length	D		9.00 BSC	
Exposed Pad Length	D2	4.60	4.70	4.80
Terminal Width	b	0.15	0.20	0.25
Terminal Length	L	0.30	0.40	0.50
Terminal-to-Exposed-Pad	K		1.755 REF	

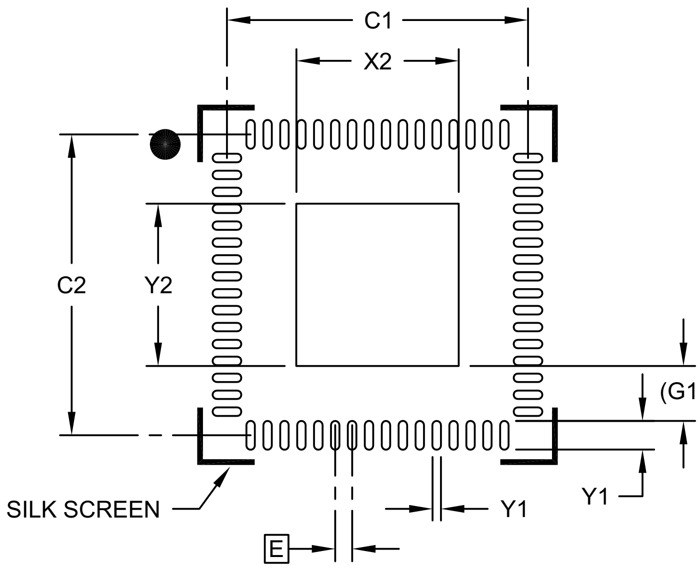
Notes:

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 exact value shown without tolerances.
   REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-260A Sheet 2 of 2

64-Lead Very Thin Plastic Quad Flat, No Lead Package (RG) - 9x9x1.0 mm Body [QFN] 4.7x4.7 mm Exposed Pad

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



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E	0.50 BSC		
Optional Center Pad Width	X2			4.80
Optional Center Pad Length	Y2			4.80
Contact Pad Spacing	C1		8.90	
Contact Pad Spacing	C2		8.90	
Contact Pad Width (X64)	X1			0.25
Contact Pad Length (X64)	Y1			0.85
Contact Pad to Center Pad (X64)	G1	1.625 REF		

Notes:

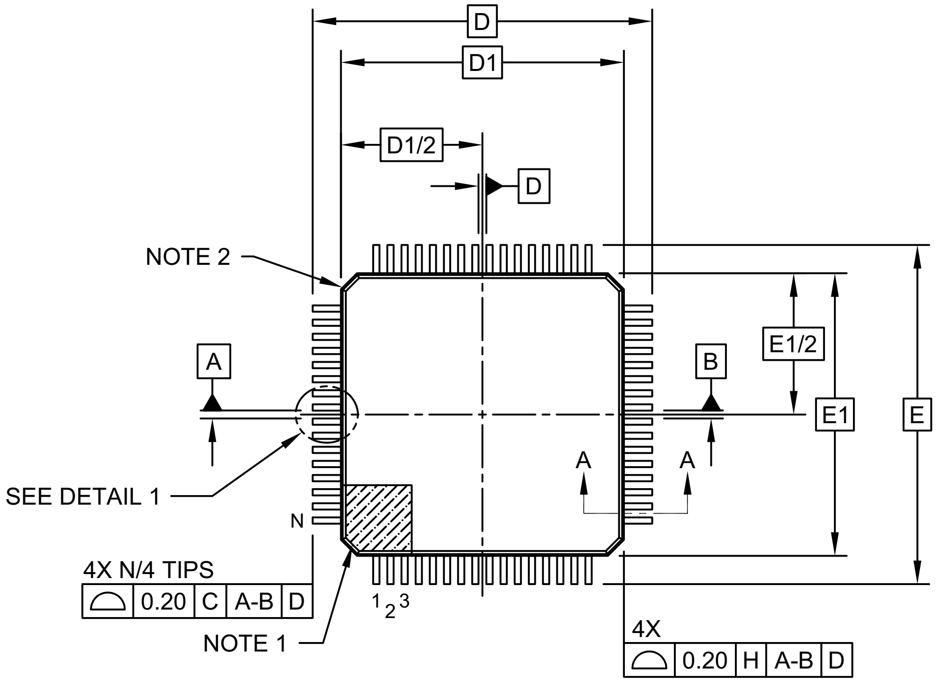
1. Dimensioning and tolerancing per ASME Y14.5M

   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

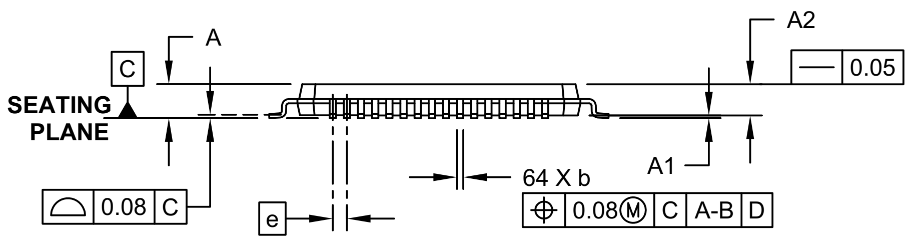
Microchip Technology Drawing C04-2260A

64-Lead Plastic Thin Quad Flatpack (PT)-10x10x1 mm Body, 2.00 mm Footprint [TQFP]

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



TOP VIEW

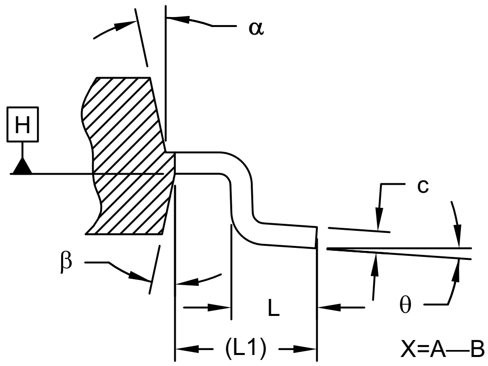


SIDE VIEW

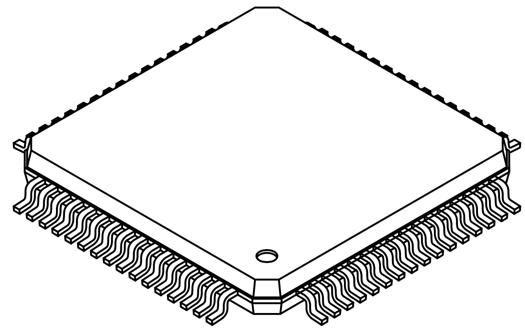
Microchip Technology Drawing C04-085C Sheet 1 of 2

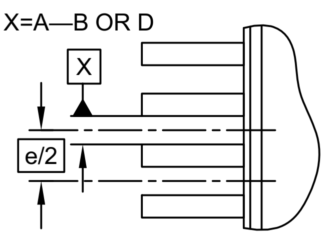
64-Lead Plastic Thin Quad Flatpack (PT)-10x10x1 mm Body, 2.00 mm Footprint [TQFP]

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



SECTION A-A





DETAIL 1

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Leads	N		64	
Lead Pitch	e		0.50 BSC	
Overall Height	A	-	-	1.20
Molded Package Thickness	A2	0.95	1.00	1.05
Standoff	A1	0.05	-	0.15
Foot Length	L	0.45	0.60	0.75
Footprint	L1		1.00 REF	
Foot Angle	φ	0°	3.5°	7°
Overall Width	E		12.00 BSC	
Overall Length	D		12.00 BSC	
Molded Package Width	E1		10.00 BSC	
Molded Package Length	D1		10.00 BSC	
Lead Thickness	c	0.09	-	0.20
Lead Width	b	0.17	0.22	0.27
Mold Draft Angle Top	α	11°	12°	13°
Mold Draft Angle Bottom	β	11°	12°	13°

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Chamfers at corners are optional; size may vary.
3. Dimensions D1 and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.25mm per side.
4. Dimensioning and tolerancing per ASME Y14.5M

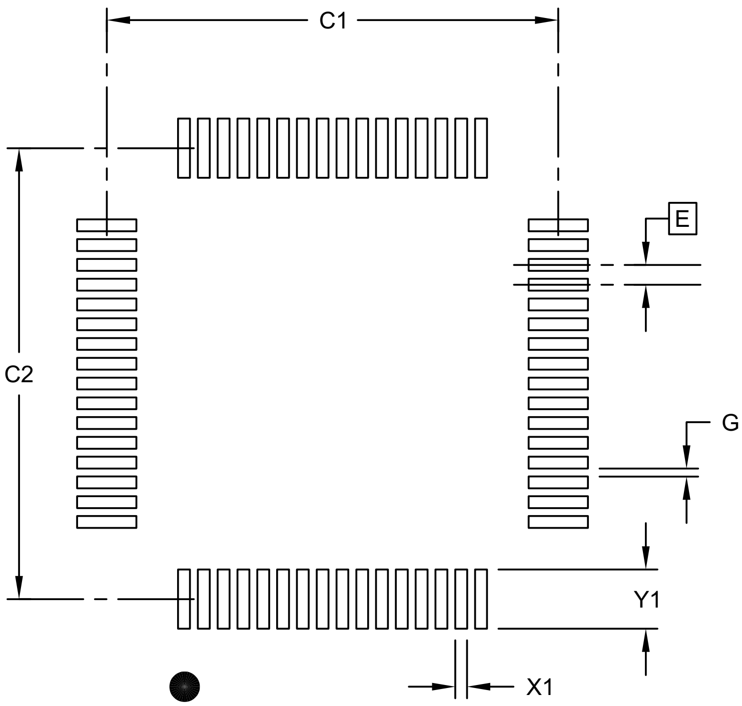
    BSC: Basic Dimension. Theoretically exact value shown without tolerances.

    REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-085C Sheet 2 of 2

64-Lead Plastic Thin Quad Flatpack (PT)-10x10x1 mm Body, 2.00 mm Footprint [TQFP]

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



RECOMMENDED LAND PATTERN

	Units	MILLIMETERS		
	Dimension Limits	MIN	NOM	MAX
Contact Pitch	E	0.50 BSC		
Contact Pad Spacing	C1		11.40	
Contact Pad Spacing	C2		11.40	
Contact Pad Width (X28)	X1			0.30
Contact Pad Length (X28)	Y1			1.50
Distance Between Pads	G	0.20		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing C04-2085B Sheet 1 of 1

APPENDIX A: REVISION HISTORY

Revision A (06/2015)

Initial release of this document.

THE MICROCHIP WEB SITE

Microchip provides online support via our web site at www.microchip.com. This web site is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the web site contains the following information:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQ), technical support requests, online discussion groups, Microchip consultant program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

CUSTOMER CHANGE NOTIFICATION SERVICE

Microchip's customer notification service helps keep customers current on Microchip products. Subscribers will receive e-mail notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, access the Microchip web site at www.microchip.com. Under "Support", click on "Customer Change Notification" and follow the registration instructions.

CUSTOMER SUPPORT

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support

Customers should contact their distributor, representative or Field Application Engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

Technical support is available through the web site at: http://www.microchip.com/support

PRODUCT IDENTIFICATION SYSTEM

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.

PART NO.	[X][(1)]	X	/XX	XXX
Device	Tape and Reel Option	Temperature Range	Package	Pattern

Device:	MTCH6303
Tape and Reel Option:	Blank = Standard packaging (tube or tray) T = Tape and Reel[(1)]
Temperature Range:	I = -40°C to +85°C (Industrial)
Package:	RG = QFN PT = TQFP
Pattern:	QTP, SQTP, Code or Special Requirements (blank otherwise)

Examples:

a) MTCH6303-I/PT = Industrial Temp TQFP package.

Note 1: Tape and Reel identifier only appears in the catalog part number description. This identifier is used for ordering purposes and is not printed on the device package. Check with your Microchip Sales Office for package availability with the Tape and Reel option.

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Trademarks

ISBN: 978-1-63277-531-3

QUALITY MANAGEMENT SYSTEM CERTIFIED BY DNV

= ISO/TS 16949 =

Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

Worldwide Sales and Service

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
http://www.microchip.com/support
Web Address:
www.microchip.com

Atlanta
Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

Austin, TX
Tel: 512-257-3370

Boston
Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

Chicago
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

Cleveland
Independence, OH
Tel: 216-447-0464
Fax: 216-447-0643

Dallas
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

Detroit
Novi, MI
Tel: 248-848-4000

Houston, TX
Tel: 281-894-5983

Indianapolis
Noblesville, IN
Tel: 317-773-8323
Fax: 317-773-5453

Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608

New York, NY
Tel: 631-435-6000

San Jose, CA
Tel: 408-735-9110

Canada - Toronto
Tel: 905-673-0699
Fax: 905-673-6509

ASIA/PACIFIC

Asia Pacific Office
Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon
Hong Kong
Tel: 852-2943-5100
Fax: 852-2401-3431

Australia - Sydney
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

China - Beijing
Tel: 86-10-8569-7000
Fax: 86-10-8528-2104

China - Chengdu
Tel: 86-28-8665-5511
Fax: 86-28-8665-7889

China - Chongqing
Tel: 86-23-8980-9588
Fax: 86-23-8980-9500

China - Dongguan
Tel: 86-769-8702-9880

China - Hangzhou
Tel: 86-571-8792-8115
Fax: 86-571-8792-8116

China - Hong Kong SAR
Tel: 852-2943-5100
Fax: 852-2401-3431

China - Nanjing
Tel: 86-25-8473-2460
Fax: 86-25-8473-2470

China - Qingdao
Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

China - Shanghai
Tel: 86-21-5407-5533
Fax: 86-21-5407-5066

China - Shenyang
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

China - Shenzhen
Tel: 86-755-8864-2200
Fax: 86-755-8203-1760

China - Wuhan
Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

China - Xian
Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

ASIA/PACIFIC

China - Xiamen
Tel: 86-592-2388138
Fax: 86-592-2388130

China - Zhuhai
Tel: 86-756-3210040
Fax: 86-756-3210049

India - Bangalore
Tel: 91-80-3090-4444
Fax: 91-80-3090-4123

India - New Delhi
Tel: 91-11-4160-8631
Fax: 91-11-4160-8632

India - Pune
Tel: 91-20-3019-1500

Japan - Osaka
Tel: 81-6-6152-7160
Fax: 81-6-6152-9310

Japan - Tokyo
Tel: 81-3-6880- 3770
Fax: 81-3-6880-3771

Korea - Daegu
Tel: 82-53-744-4301
Fax: 82-53-744-4302

Korea - Seoul
Tel: 82-2-554-7200
Fax: 82-2-558-5932 or
82-2-558-5934

Malaysia - Kuala Lumpur
Tel: 60-3-6201-9857
Fax: 60-3-6201-9859

Malaysia - Penang
Tel: 60-4-227-8870
Fax: 60-4-227-4068

Philippines - Manila
Tel: 63-2-634-9065
Fax: 63-2-634-9069

Singapore
Tel: 65-6334-8870
Fax: 65-6334-8850

Taiwan - Hsin Chu
Tel: 886-3-5778-366
Fax: 886-3-5770-955

Taiwan - Kaohsiung
Tel: 886-7-213-7828

Taiwan - Taipei
Tel: 886-2-2508-8600
Fax: 886-2-2508-0102

Thailand - Bangkok
Tel: 66-2-694-1351
Fax: 66-2-694-1350

EUROPE

Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

Denmark - Copenhagen
Tel: 45-4450-2828
Fax: 45-4485-2829

France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany - Dusseldorf
Tel: 49-2129-3766400

Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Germany - Pforzheim
Tel: 49-7231-424750

Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

Italy - Venice
Tel: 39-049-7625286

Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

Poland - Warsaw
Tel: 48-22-3325737

Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

Sweden - Stockholm
Tel: 46-8-5090-4654

UK - Wokingham
Tel: 44-118-921-5800
Fax: 44-118-921-5820

01/27/15