Keywords: TSC, 4-wire touch panel, touch screen controller, resistive

APPLICATION NOTE 5435

Quick Start Programmer's Guide for the MAX11800–MAX11803 and MAX11811 Touch-Screen Controllers

By: Faisal Tariq
Jul 12, 2012

Abstract: This application note shows how to quickly develop code for the MAX11800–MAX11803 and MAX11811 resistive touch-screen controllers and operate them in direct or autonomous mode. Software is provided to help compute data conversion based on the register settings.

Overview

The MAX11800–MAX11803 and MAX11811 touch-screen controllers can quickly be made operational in either direct or autonomous mode. This application note breaks down the procedure of communication into two steps. First, users must set all the configuration registers. The second step requires sending a set of commands that produces the desired data (X, Y, Z1, Z2). This application note also provides software that, based on the register settings, calculates the time for the MAX11800–MAX11803 and MAX11811 to complete data conversion.

The default register settings of the MAX118xx resistive touch-screen controllers are such that the touch-screen controller connected to most touch-panel systems will be functional with only a few minor changes.

It is expected that the user is familiar with SPI and I²C communication and also has access to the touch-screen controller's data sheet. Refer to the device data sheet to see how the I²C and SPI are handled.

Register Address Modification for Read and Write

The register address has been modified for a write and read operation, as required by the touch-screen controller. This is displayed in the corresponding device's data sheet. The translated register addresses are explained in **Tables 1**, **2**, and **3** for easy reference.

Table 1. Configuration Register Summary

Translated Address for Read/Write		Address	Access	Pairable (Write)*	Auto-Incr (Read)**	Data Length (Bytes)	Function
Write	Read						
—	0x01	0x00	R	N	Y	8	General status
0x02	0x03	0x01	R/W	Y	Y	8	General configuration
0x04	0x05	0x02	R/W	Y	Y	8	Measurement resolution
0x06	0x07	0x03	R/W	Y	Y	8	Measurement averaging
0x08	0x09	0x04	R/W	Y	Y	8	ADC sample time
0x0A	0x0B	0x05	R/W	Y	Y	8	Panel setup times
0x0C	0x0D	0x06	R/W	Y	Y	8	ADC delay init. conversion
0x0E	0x0F	0x07	R/W	Y	Y	8	Touch detect pullup times
0x10	0x11	0x08	R/W	Y	Y	8	Autonomous mode timing
0x12	0x13	0x09	R/W	Y	Y	8	Aperture settings (auto)
0x14	0x15	0x0A	R/W	Y	Y	8	Auxiliary meas. configuration
0x16	0x17	0x0B	R/W	Y	Y	8	Operating mode configuration

*A single command can be used to write to an indefinite number of address data pair, i.e., Addr1+Data1+Addr2+Data2, etc.

**Auto-incr (read): This address is incremented to point to the next address upon a read.

The default values of these registers typically work with almost any hardware setup (refer to the data sheet for more information). However, to optimize performance, the user should adjust the register values individually to meet their specific design needs.

Table 2. Conversion Command Summary

Translated Command		Measurement Commands	Access	Pairable (Write)*	CMND Length (Bytes)	Function
Write	Read					
0xE0	—	0x70	W	N	8	X, Y combined command
0xE4	—	0x72	W	N	8	X, Y, Z1 combined command
0xE8	—	0x74	W	N	8	X, Y, Z1, Z2 combined command
0xEC	—	0x76	W	N	8	AUX conversion
0xF0	—	0x78	W	N	8	X measurement
0xF4	—	0x7A	W	N	8	Y measurement
0xF8	—	0x7C	W	N	8	Z1 measurement
0xFC	—	0x7E	W	N	8	Z2 measurement

*A single command can be used to write to an indefinite number of address data pair, i.e., Addr1+Data1+Addr2+Data2, etc.

The commands in Table 2 initiate a conversion for the measurement to take place.

For advanced users: If multiple measurements of the same type are needed (for example, multiple X measurements) where the present panel setup is maintained, then the CONT bit can be set as 1. Refer to the device data sheet for details.

Table 3. Data Read Back Summary

Translated Address for Read/Write		Address	Access	Auto-Incr (Read)*	Data Length (Bytes)	Function
Write	Read					
—	0xA1	0x50	R	N	INF	Read next available FIFO data block
—	0xA5	0x52	R	Y	8	X MSB (direct conversion result)
—	0xA7	0x53	R	Y	8	X LSB (direct conversion result)
—	0xA9	0x54	R	Y	8	Y MSB (direct conversion result)
—	0xAB	0x55	R	Y	8	Y LSB (direct conversion result)
—	0xAD	0x56	R	Y	8	Z1 MSB (direct conversion result)
—	0xAF	0x57	R	Y	8	Z1 LSB (direct conversion result)
—	0xB1	0x58	R	Y	8	Z2 MSB (direct conversion result)
—	0xB3	0x59	R	Y	8	Z2 LSB (direct conversion result)
—	0xB5	0x5A	R	Y	8	AUX MSB (direct conversion result)
—	0xB7	0x5B	R	N	8	AUX LSB (direct conversion result)

*This address is incremented to point to the next address upon a read.

Step 1: Configuration Register Setup

Set the Configuration Registers to the values shown below. (These values are suggestions to get the system working. The user should set these values based on their hardware setup.)

Register Address	Value Written
0x01	See below
0x02	0x00
0x03	0x00
0x04	0x00
0x05	0x00
0x06	0x00
0x07	0x10
0x08	0x00
0x09	0x00
0x0A	0x00
0x0B	See below

Register 0x01

Bits 4, 5, and 6 can affect the quick startup. The other bits should be set for optimum performance after the device has started communicating with the host.

Bit 4: 0 = TIRQB output is a CMOS buffered output, 1 = TIRQB output is an open-drain NMOS output.

Bit 5: If bit 4 is set to 1, then this bit will be selected to allow for an internal pullup resistance (i.e., 0 = disable IRQ internal pullup resistance, 1 = enable IRQ internal pullup resistance).

Bit 6: This only affects the MAX11800 and MAX11802, which have SPI communication. 1 = internal bus holder enabled, 0 = internal bus holder disabled.

Register 0x0B

Direct mode: 0x0B should be set to 0x00.

Autonomous mode X, Y: 0x0B should be set to 0x20 (bits 6 and 5).

Autonomous mode X, Y, Z1: 0x0B should be set to 0x40 (bits 6 and 5).

Autonomous mode X, Y, Z1, Z2: 0x0B should be set to 0x80 (bits 6 and 5).

Important Note: The default value of bit 7 in register 0x0B after power-up is 1, which means that the part is in power-down mode. The user must write 0 to bit 7 of register 0x0B to power the part.

However, the register addresses listed above are not directly used; instead they change based on the read and write instructions shown in Table 1. The values used to write and read to these registers are shown below.

Register Address	Write	Read
0x01	0x02	0x03
0x02	0x04	0x05
0x03	0x06	0x07
0x04	0x08	0x09
etc.		

Therefore, to write 0xF0 to the register 0x01, the user will write 0xF0 to 0x02. And to read from register 0x01, the user will read from 0x03.

Step 2: Measurement/Conversion and Read-Back

Direct Mode

Reading Each Parameter X, Y, Z1, and Z2 Individually

Reading X in Direct Mode

1. **X measurement/conversion**: *Send 0xF0*. See Table 2 for translation.
2. **Put a delay for the above conversion to complete**: Use the MAX118xx Conversion Time software to calculate the wait time based on the Configuration Register setting.
3. **X data read back**: *Send 0xA5 followed by a request for 2 bytes*. See Table 3 for translation. The first read byte will be the MSByte followed by the LSByte.

Reading Y in Direct Mode

1. **Y measurement/conversion**: *Send 0xF4*. See Table 2 for translation.
2. **Put a delay for the above conversion to complete**: Use the MAX118xx Conversion Time software to calculate the wait time based on the Configuration Register setting.
3. **Y data read back**: *Send 0xA9 followed by a request for 2 bytes*. See Table 3 for translation. The first read byte will be the MSByte followed by the LSByte.

Reading Z1 in Direct Mode

1. **Z1 measurement/conversion**: *Send 0xF8*. See Table 2 for translation.
2. **Put a delay for the above conversion to complete**: Use the MAX118xx Conversion Time software to calculate the wait time based on the Configuration Register setting.
3. **Z1 data read back**: *Send 0xAD followed by a request for 2 bytes*. See Table 3 for translation. The first read byte will be the MSByte followed by the LSByte.

Reading Z2 in Direct Mode

1. **Z2 measurement/conversion**: *Send 0xFC*. See Table 2 for translation.
2. **Put a delay for the above conversion to complete**: Use the MAX118xx Conversion Time software to calculate the wait time based on the Configuration Register setting.
3. **Z2 data read back**: *Send 0xB1 followed by a request for 2 bytes*. See Table 3 for translation. The first read byte will be the MSByte followed by the LSByte.

Reading AUX in Direct Mode

1. **AUX measurement/conversion**: *Send 0xEC*. See Table 2 for translation.
2. **Put a delay for the above conversion to complete**: Use the MAX118xx Conversion Time software to calculate the wait time based on the Configuration Register setting.
3. **AUX data read back**: *Send 0xB5 followed by a request for 2 bytes*. See Table 3 for translation. The first read byte will be the MSByte followed by the LSByte.

Combined X, Y in Direct Mode

1. **Combined X, Y measurement/conversion**: *Send 0xE0*. See Table 2 for translation.
2. **Put a delay for the above conversion to complete**: Use the MAX118xx Conversion Time software to calculate the wait time based on the Configuration Register setting.
3. **Combined X, Y data read back**: *Send 0xA5 followed by a request for 4 bytes*. See Table 3 for translation. The first read byte will be the MSByte for X, the second byte will be the LSByte for X, the third byte will be the MSByte for Y, and the fourth byte will be the LSByte for Y.

Combined X, Y, Z1 in Direct Mode

1. **Combined X, Y, Z1 measurement/conversion**: *Send 0xE4*. See Table 2 for translation.
2. **Put a delay for the above conversion to complete**: Use the MAX118xx Conversion Time software to calculate the wait time based on the Configuration Register setting.
3. **Combined X, Y, Z1 data read back**: *Send 0xA5 followed by a request for 6 bytes*. See Table 3 for translation. The first read byte will be the MSByte for X; the second byte will be the LSByte for X; the third byte will be the MSByte for Y; the fourth byte will be the LSByte for Y; the fifth byte will be the MSByte for Z1; and the sixth byte will be the LSByte for Z1.

Combined X, Y, Z1, Z2 in Direct Mode

1. **Combined X, Y, Z1, Z2 measurement/conversion**: *Send 0xE8*. See Table 2 for translation.
2. **Put a delay for the above conversion to complete**: Use the MAX118xx Conversion Time software to calculate the wait time based on the Configuration Register setting.
3. **Combined X, Y, Z1, Z2 data read back**: *Send 0xA5 followed by a request for 8 bytes*. See Table 3 for translation. The first read byte will be the MSByte for X; the second byte will be the LSByte for X; the third byte will be the MSByte for Y; the fourth byte will be the LSByte for Y; the fifth byte will be the MSByte for Z1; the sixth byte will be the LSByte for Z1; the seventh byte will be the MSByte for Z2; and the eight byte will be Z2LSB.

Autonomous Mode

The autonomous mode can be handled in two ways:

Technique #1: Read the FIFO immediately when the TIRQB goes low. This way, the FIFO never fills beyond the first set of data.

Technique #2: Wait until after the TIRQB goes down for the FIFO to get a small amount of data before reading the whole buffer. This method works well when the host cannot respond fast enough to the interrupt. However, it is important when using this technique that the FIFO does not overflow. If this happens, data is getting lost. For Technique #2 to work properly, it is mandatory that the host processor services the MAX11800–MAX11803 before the FIFO overflows. One can monitor bit 3 of Register 0x00 (General Status

Register) to monitor overflow.

Technique #1

The TIRQB goes low when the FIFO receives data. It goes back up immediately when the FIFO is completely read. First make sure that the correct autonomous mode is set in register 0x0B (e.g., X, Y or X, Y, Z1 or X, Y, Z1, Z2).

Read X, Y in Autonomous Mode

X, Y read back from the FIFO: Send 0xA1 followed by a request for 4 bytes. See Table 3 for translation. The data received will be in the following order: X_1MSB, X_1LSB, Y_1MSB, Y_1LSB.

Read X, Y, Z1 in Autonomous Mode

X, Y, Z1 read back from the FIFO: Send 0xA1 followed by a request for 6 bytes. See Table 3 for translation. The data received will be in the following order: X_1MSB, X_1LSB, Y_1MSB, Y_1LSB, $Z1_1$MSB, $Z1_1$LSB.

Read X, Y, Z1, Z2 in Autonomous Mode

X, Y, Z1, Z2 read back from the FIFO: Send 0xA1 followed by a request for 8 bytes. See Table 3 for translation. The data received will be in the following order: X_1MSB, X_1LSB, Y_1MSB, Y_1LSB, $Z1_1$MSB, $Z1_1$LSB, $Z2_1$MSB, $Z2_1$LSB.

Technique #2

The MAX11800–MAX11803 and MAX11811 have different FIFO sizes. The MAX11800–MAX11803 have 128 bytes of RAM, giving out 16 sets of data as the maximum. Contrastingly, the MAX11811 has 64 bytes of RAM, giving out 8 sets of data as the maximum. The autonomous mode data handling has been separated into two sections, one for the MAX11800–MAX11803 and the other for MAX11811, for easy reading.

MAX11800–MAX11803

Read X, Y in Autonomous Mode

1. **X, Y read back from the FIFO**: *Send 0xA1 followed by a request for 64 bytes*. See Table 3 for translation. The data received will be in the following order: X_1MSB, X_1LSB, Y_1MSB, Y_1LSB, X_2MSB, X_2LSB, Y_2MSB, Y_2LSB, ..., X_{16}MSB, X_{16}LSB, Y_{16}MSB, Y_{16}LSB.
2. **Put a delay for the FIFO to fill**: Use the MAX118xx Conversion Time software to calculate the wait time based on the Configuration Register setting.

Read X, Y, Z1 in Autonomous Mode

1. **X, Y, Z1 read back from the FIFO**: *Send 0xA1 followed by a request for 96 bytes*. See Table 3 for translation. The data received will be in the following order: X_1MSB, X_1LSB, Y_1MSB, Y_1LSB, $Z1_1$MSB, $Z1_1$LSB, X_2MSB, X_2LSB, Y_2MSB, Y_2LSB, $Z1_2$MSB, $Z1_2$MSB, ..., X_{16}MSB, X_{16}LSB, Y_{16}MSB, Y_{16}LSB, $Z1_{16}$MSB, $Z1_{16}$LSB.
2. **Put a delay for the FIFO to fill**: Use the MAX118xx Conversion Time software to calculate the wait time based on the Configuration Register setting.

Read X, Y, Z1, Z2 in Autonomous Mode

1. X, Y, Z1, Z2 read back from the FIFO: Send 0xA1 followed by a request for 128 bytes. See Table 3 for translation. The data received will be in the following order: X_1MSB, X_1LSB, Y_1MSB, Y_1LSB, $Z1_1$MSB, $Z1_1$LSB, $Z2_1$MSB, $Z2_1$LSB, X_2MSB, X_2LSB, Y_2MSB, Y_2LSB, $Z1_2$MSB, $Z1_2$MSB, $Z2_2$MSB, $Z2_2$LSB, ..., X_{16}MSB, X_{16}LSB, Y_{16}MSB, Y_{16}LSB, $Z1_{16}$MSB, $Z1_{16}$MSB, $Z2_{16}$MSB, $Z2_{16}$LSB.
2. Put a delay for the FIFO to fill: Use the MAX118xx Conversion Time software to calculate the wait time based on the Configuration Register setting.

MAX11811

Read X, Y in Autonomous Mode

1. X, Y read back from the FIFO: Send 0xA1 followed by a request for 32 bytes. See Table 3 for translation. The data received will be in the following order: X_1MSB, X_1LSB, Y_1MSB, Y_1LSB, X_2MSB, X_2LSB, Y_2MSB, Y_2LSB, ..., X_8MSB, X_8LSB, Y_8MSB, Y_8LSB.
2. Put a delay for the FIFO to fill: Use the MAX118xx Conversion Time software to calculate the wait time based on the Configuration Register setting.

Read X, Y, Z1 in Autonomous Mode

1. X, Y, Z1 read back from the FIFO: Send 0xA1 followed by a request for 48 bytes. See Table 3 for translation. The data received will be in the following order: X_1MSB, X_1LSB, Y_1MSB, Y_1LSB, $Z1_1$MSB, $Z1_1$LSB, X_2MSB, X_2LSB, Y_2MSB, Y_2LSB, $Z1_2$MSB, $Z1_2$MSB, ..., X_8MSB, X_8LSB, Y_8MSB, Y_8LSB, $Z1_8$MSB, $Z1_8$LSB.
2. Put a delay for the FIFO to fill: Use the MAX118xx Conversion Time software to calculate the wait time based on the Configuration Register setting.

Read X, Y, Z1, Z2 in Autonomous Mode

1. X, Y, Z1, Z2 read back from the FIFO: Send 0xA1 followed by a request for 64 bytes. See Table 3 for translation. The data received will be in the following order: X_1MSB, X_1LSB, Y_1MSB, Y_1LSB, $Z1_1$MSB, $Z1_1$LSB, $Z2_1$MSB, $Z2_1$LSB, X_2MSB, X_2LSB, Y_2MSB, Y_2LSB, $Z1_2$MSB, $Z1_2$MSB, $Z2_2$MSB, $Z2_2$LSB, ..., X_8MSB, X_8LSB, Y_8MSB, Y_8LSB, $Z1_8$MSB, $Z1_8$LSB, $Z2_8$MSB, $Z2_8$LSB.
2. Put a delay for the FIFO to fill: Use the MAX118xx Conversion Time software to calculate the wait time based on the Configuration Register setting.

Related Parts		
MAX11800	Low-Power, Ultra-Small Resistive Touch-Screen Controllers with I²C/SPI Interface	Free Samples
MAX11801	Low-Power, Ultra-Small Resistive Touch-Screen Controllers with I²C/SPI Interface	Free Samples

MAX11802	Low-Power, Ultra-Small Resistive Touch-Screen Controllers with I^{2}C/SPI Interface	Free Samples
MAX11803	Low-Power, Ultra-Small Resistive Touch-Screen Controllers with I^{2}C/SPI Interface	Free Samples
MAX11811	TacTouch™, Low-Power, Ultra-Small, Resistive Touch-Screen Controller with Haptic Driver	Free Samples

More Information
For Technical Support: http://www.maximintegrated.com/support
For Samples: http://www.maximintegrated.com/samples
Other Questions and Comments: http://www.maximintegrated.com/contact

Application Note 5435: http://www.maximintegrated.com/an5435
APPLICATION NOTE 5435, AN5435, AN 5435, APP5435, Appnote5435, Appnote 5435

Additional Legal Notices: http://www.maximintegrated.com/legal