

4-Wire and 8-Wire Resistive Touch-Screen Controller Using the MSP430™

Neal Brenner, Shawn Sullivan
William Goh

*Microcontroller Field Applications
MSP430 Applications*

ABSTRACT

This application report describes how to use an MSP430™ microcontroller to drive and read a resistive touch screen. The hardware and software solutions provided enable the reading of user input through a 4-wire or 8-wire resistive touch screen with a low-cost low-power customizable microcontroller. The reference design includes MSP430 software and hardware schematics along with a demonstration PC application. The design was implemented using an MSP430F2012, but it can easily be modified to use any other MSP430 with an ADC.

Contents

1	Principles of Operation	2
2	MSP430 Implementation	5
3	Demonstration System	9
4	References	10

List of Figures

1	Touch Detection	2
2	4-Wire Touch-Screen Construction	3
3	4-Wire Touch Coordinate Reading	3
4	8-Wire Touch-Screen Construction	4
5	8-Wire Touch Coordinate Reading	4
6	MSP430 Touch Detection Connections	5
7	Schematic	6
8	Software Flow Charts	7
9	Communication Format	8
10	Hardware Setup	9
11	PC Software	10

List of Tables

1	Resources Used	5
---	----------------------	---

MSP430 is a trademark of Texas Instruments.
All other trademarks are the property of their respective owners.

1 Principles of Operation

1.1 Resistive Touch-Screen Concept

A resistive touch screen is constructed with two transparent layers coated with a conductive material stacked on top of each other. When pressure is applied by a finger or a stylus on the screen, the top layer makes contact with the lower layer. When a voltage is applied across one of the layers, a voltage divider is created. The coordinates of a touch can be found by applying a voltage across one layer in the Y direction and reading the voltage created by the voltage divider to find the Y coordinate, and then applying a voltage across the other layer in the X direction and reading the voltage created by the voltage divider to find the X coordinate.

1.2 Detecting a Touch

To know if the coordinate readings are valid, there must be a way to detect whether the screen is being touched or not. This can be done by applying a positive voltage (V_{CC}) to Y+ through a pullup resistor and applying ground to X-. The pullup resistor must be significantly larger than the total resistance of the touch screen, which is usually a few hundred ohms. When there is no touch, Y+ is pulled up to the positive voltage. When there is a touch, Y+ is pulled down to ground as shown in Figure 1. This voltage-level change can be used to generate a pin-change interrupt.

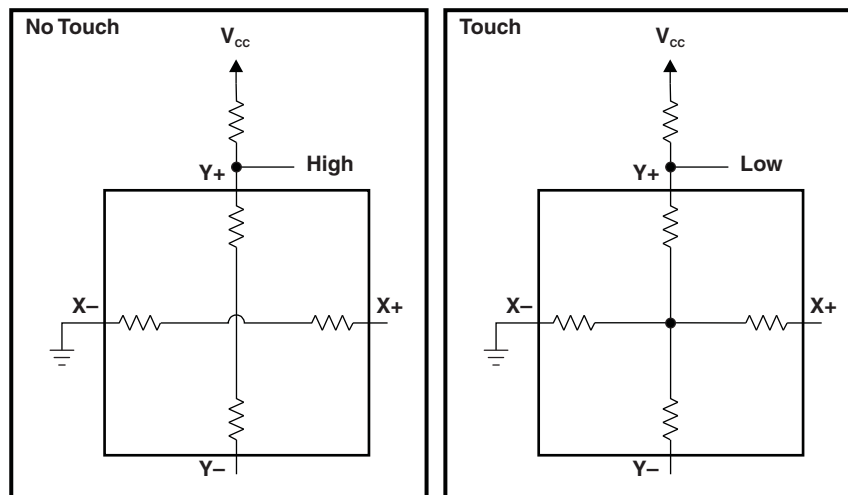


Figure 1. Touch Detection

1.3 Reading a 4-Wire Screen

A 4-wire resistive touch screen is constructed as shown in Figure 2.

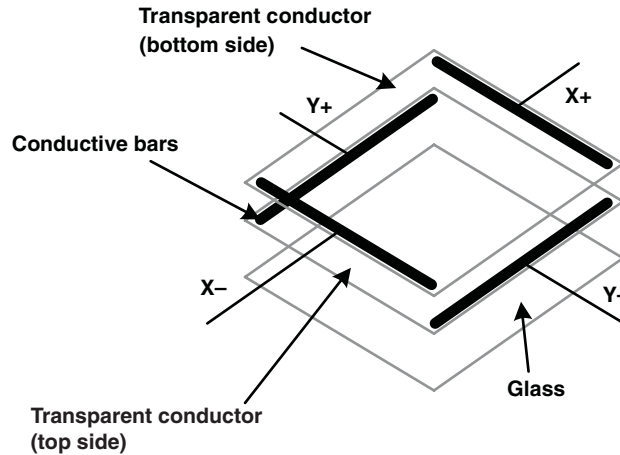


Figure 2. 4-Wire Touch-Screen Construction

The x and y coordinates of a touch on a 4-wire touch screen can be read in two steps. First, Y+ is driven high, Y- is driven to ground, and the voltage at X+ is measured. The ratio of this measured voltage to the drive voltage applied is equal to the ratio of the y coordinate to the height of the touch screen. The y coordinate can be calculated as shown in Figure 3. The x coordinate can be similarly obtained by driving X+ high, driving X- to ground, and measuring the voltage at Y+. The ratio of this measured voltage to the drive voltage applied is equal to the ratio of the x coordinate to the width of the touch screen. This measurement scheme is shown in Figure 3.

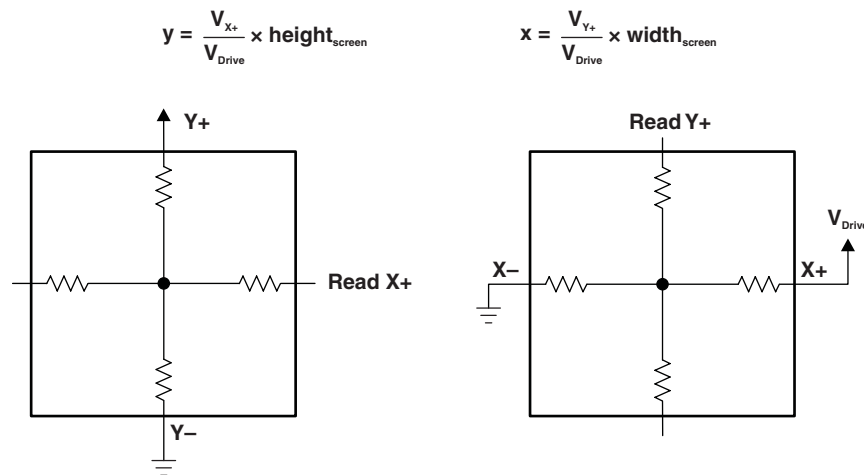


Figure 3. 4-Wire Touch Coordinate Reading

1.4 Reading an 8-Wire Screen

An 8-wire resistive touch screen is constructed as shown in Figure 4.

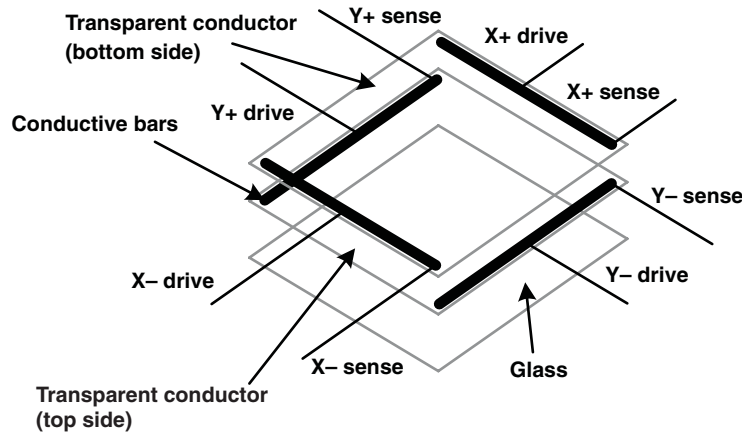


Figure 4. 8-Wire Touch-Screen Construction

In comparison to a 4-wire touch screen, an 8-wire touch screen adds sense wires to the end of each of the conductive bars. This allows any voltage offset created by the wiring or drive circuitry to be calibrated out during operation.

An 8-wire touch screen is calibrated by measuring voltage extremes on either coordinate. First, Y+ drive is driven high and Y- drive is driven low. The corresponding voltages measured at Y+ sense and Y- sense are denoted V_{Ymax} and V_{Ymin} . A similar procedure yields V_{Xmax} and V_{Xmin} . These are the maximum and minimum possible voltages across each coordinate.

The coordinates of a touch on an 8-wire touch screen can be read by first driving Y+ drive high, driving Y- drive to ground, and reading the voltage at X+ sense. Using the maximum and minimum results obtained during calibration, the y coordinate can be calculated as shown in the equations in Figure 5. The x coordinate can be obtained by driving X+ drive high, driving X- drive to ground, and reading the voltage at Y+ sense. This process is shown in Figure 5.

$$y = \frac{\left(\frac{V_{X+} - V_{Ymin}}{V_{Ymax} - V_{Ymin}} \right)}{V_{Drive}} \times \text{height}_{screen}$$

$$x = \frac{\left(\frac{V_{Y+} - V_{Xmin}}{V_{Xmax} - V_{Xmin}} \right)}{V_{Drive}} \times \text{width}_{screen}$$

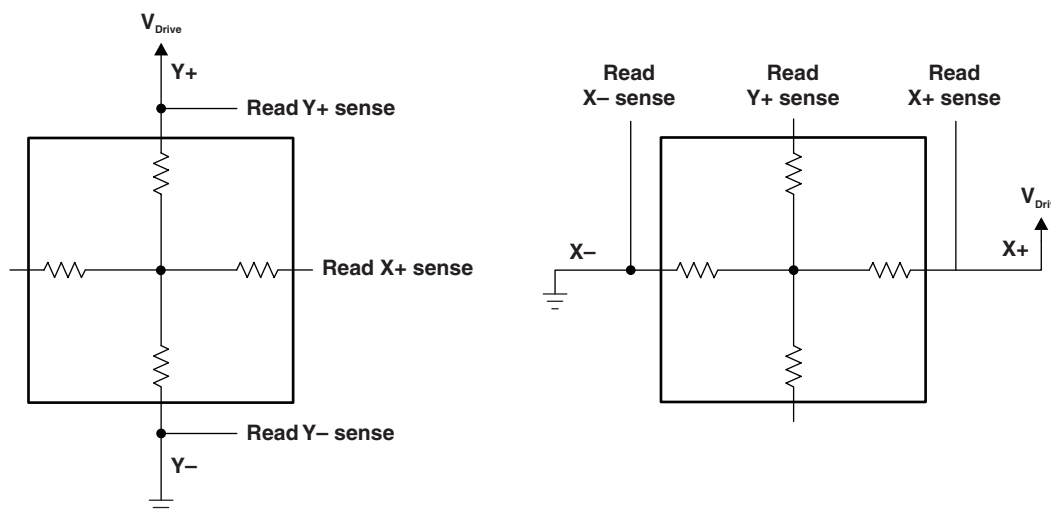


Figure 5. 8-Wire Touch Coordinate Reading

2 MSP430 Implementation

2.1 Solution Overview

Any MSP430 device with an ADC can be used as a precise, flexible, and low-power touch-screen controller. This application report provides a hardware and software proposal based on the MSP430F20x2 to implement both a 4-wire and 8-wire touch-screen controller. Both solutions wait in LPM4 (~0.1 μ A) for the screen to be touched, and then transmit the coordinates of the touch over UART or I²C. When the screen is no longer touched, the MSP430F20x2 is put back into LPM4. Table 1 shows the MSP430 resources used for each solution.

Table 1. Resources Used

Touch Screen	Communication	GPIO	Flash (bytes)	RAM (bytes)	Other
4 wire	UART	5	878	74	Two channels of ADC10, TACCR1
8 wire	UART	9	1178	82	Four channels of ADC10, TACCR1
4 wire	I ² C	6	992	75	Two channels of ADC10, USI
8 wire	I ² C	10	1292	83	Four channels of ADC10, USI

2.2 Detecting a Touch With the MSP430

The MSP430 2xx devices have programmable internal pullup and pulldown resistors on all GPIO pins. This feature can be used to detect a touch on the screen. The MSP430 GPIO pins connected to pins Y+ (Y+ drive for 8-wire screens) and X- (X- drive for 8-wire screens) from the touch screen are used. The X- pin is set to output low. The Y+ pin is set to input with the internal pullup resistor enabled as shown in Figure 6. The Y+ pin can be sampled to determine whether the screen is being pressed. If it is high, the screen is not being pressed; if it is low, the screen is being pressed. A high-to-low port pin interrupt can also be used to enable the MSP430 to sleep in LPM4 while waiting for a touch on the screen.

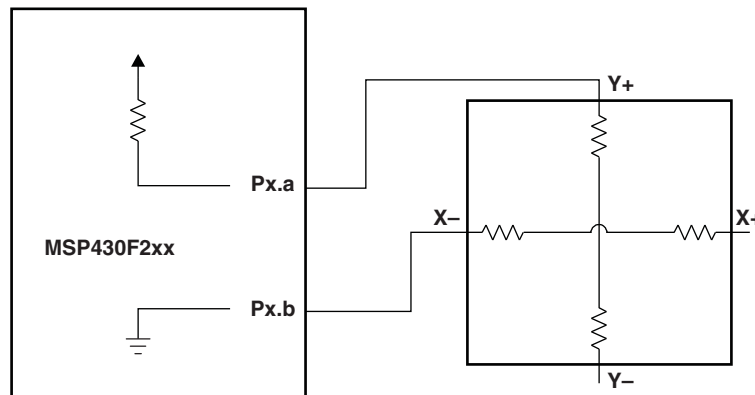


Figure 6. MSP430 Touch Detection Connections

2.3 Hardware

The schematic for the demonstration hardware is shown in Figure 7. The demonstration hardware was designed to interface to any of the Touch International (<http://www.touchinternational.com/>) 8-wire resistive touch screens (TI-8 Touch).

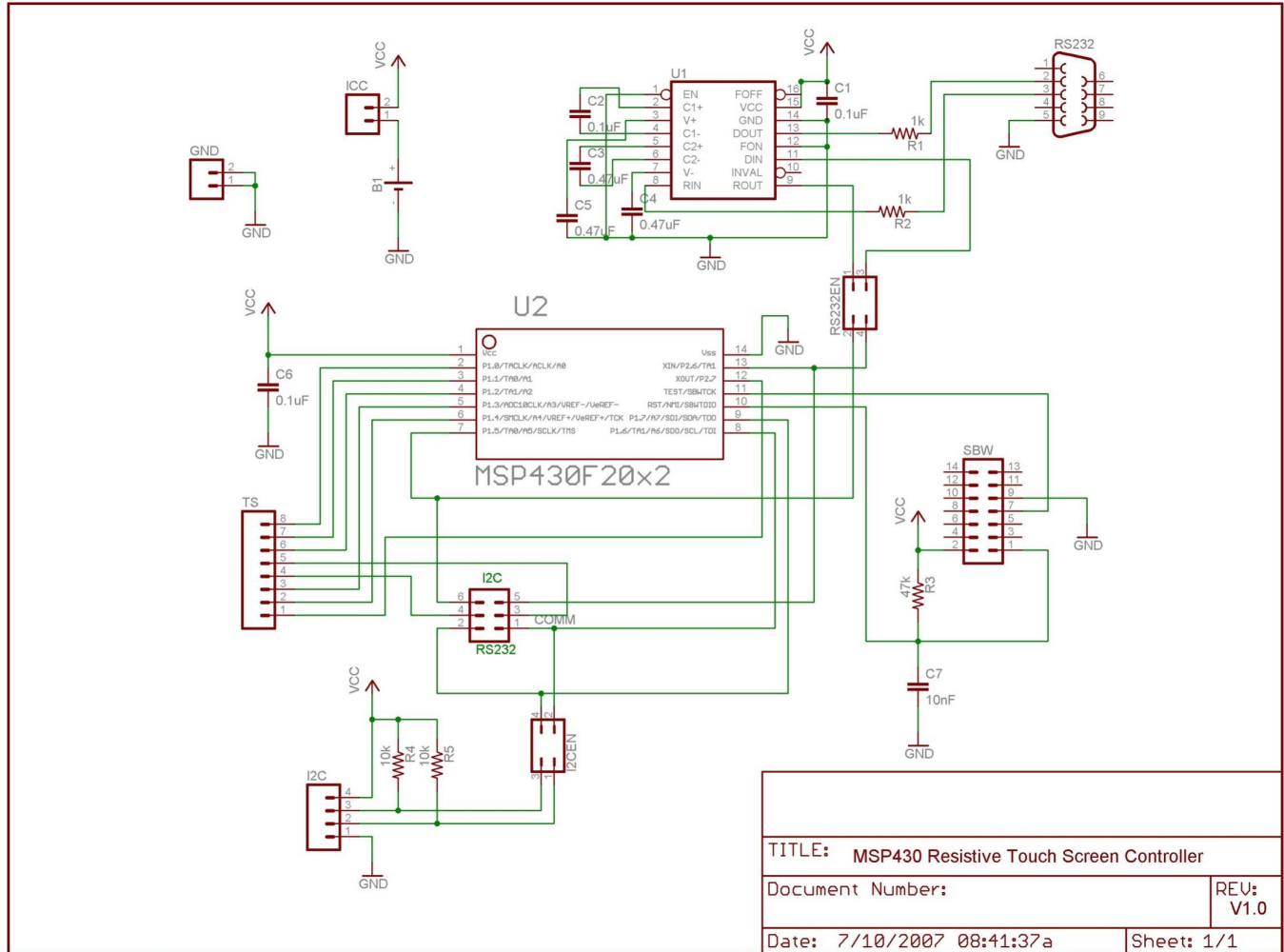


Figure 7. Schematic

2.4 Software

The software flow charts for the 4-wire and 8-wire software are shown in [Figure 8](#).

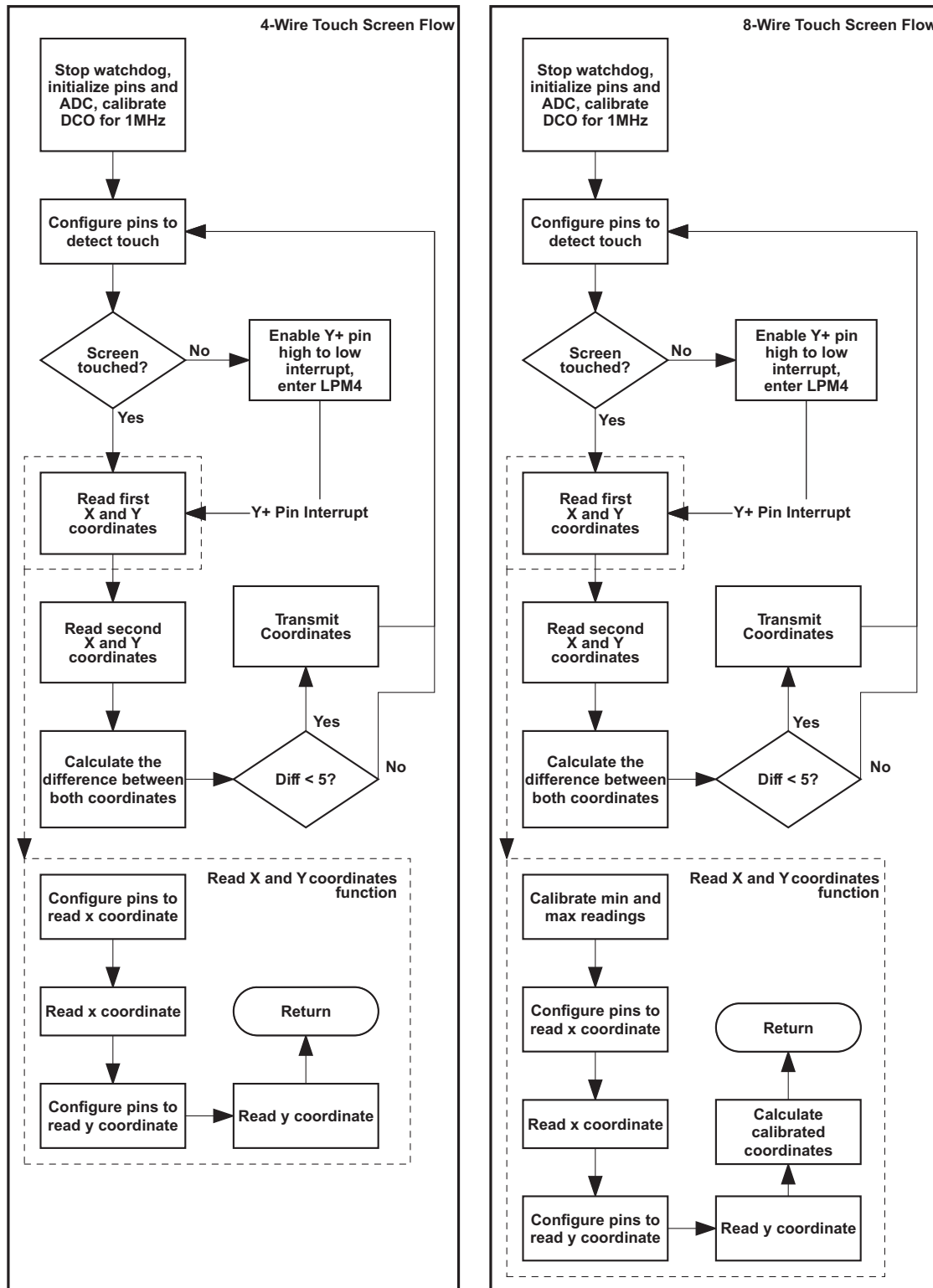


Figure 8. Software Flow Charts

2.4.1 Coordinate Reading

Once a touch is detected, the MSP430 reads the x and y coordinates of the touch. For 8-wire screens, first the minimum and maximum x and y values are read to calibrate the screen.

To read the coordinates, the pins are configured as previously described to drive the X lines and set the Y sense pin as analog ADC inputs. Four ADC conversions are taken and averaged together to obtain the x coordinate reading. For 8-wire screens, this value is calibrated as previously described. The same process is repeated to obtain the y coordinate.

2.4.2 Communication Details

After a touch coordinate is read, it is transmitted back to a host. The zip file associated with this document has example code using master I²C or an 8N1 9600 baud UART. The Timer_A module is used to implement the UART transmit function as described in *Implementing a UART Function With Timer_A3 (SLAA078)*.

Each coordinate is sent in a packet of four bytes. The first byte is a synch/control byte. The MSB is always 1 to differentiate it from the data bytes. If the screen is still being touched, a value of 0xFF is sent as the control byte to indicate that the data following is valid. If the screen is not still being touched, a value of 0x80 is sent to indicate that the data following is not valid, and this is the final pair of coordinates for the touch. The 10-bit x and y coordinates are split and packed into three data bytes. The three MSBs of the x and y are packed together in the first byte, and the seven LSBs of the x and y coordinates are sent in the third and fourth, respectively. The communication format is shown in [Figure 9](#).

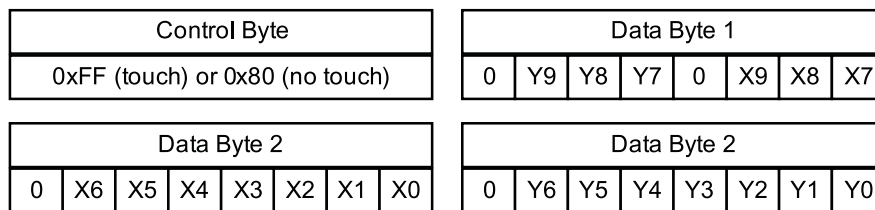


Figure 9. Communication Format

3 Demonstration System

To demonstrate the hardware and software solution proposed in this document, a PC application is provided in the accompanying zip file. This section discusses how to setup and run the demonstration.

3.1 Hardware Setup

The demonstration software uses the RS232 port of a PC, so the 4-wire or 8-wire RS232 software must be loaded into the hardware. At the beginning of the file named main.c, set the WIRES #define to match the number of wires on your touch screen. The COMM #define must be set to UART. The configuration for an 8-wire demonstration is:

```
// Define # of wires, 8 or 4 #define WIRES 8 // Define communication interface, I2C
or UART #define COMM UART
```

To enable RS232 communication on the board, five jumpers must be put in place as shown by the red boxes in [Figure 10](#). There should be no jumpers connected on the I2CEN header.

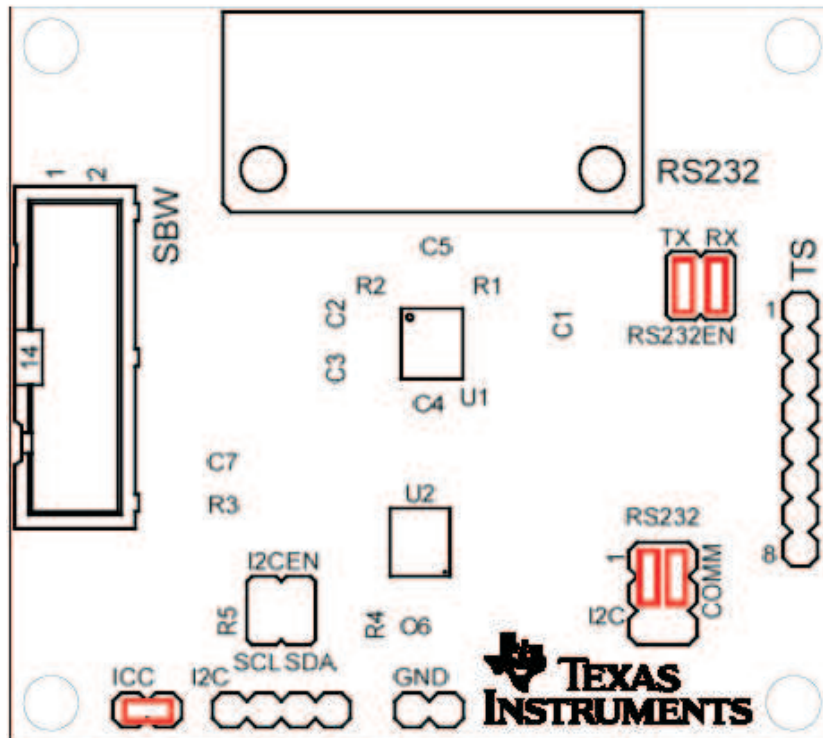


Figure 10. Hardware Setup

The touch screen should be connected to the TS header, and an RS232 cable should be connected between the PC COM1 port and the RS232 connector on the board. Finally, the board should be powered by 1.8 V to 3.6 V, either through the JTAG header or by a CR2032 battery on the back of the board.

3.2 PC Software

The PC application is provided in the associated zip file. Run setup.exe to install the software. The setup program may download and install additional components required by the software. Once the proper software is loaded on the board and the hardware is configured as previously described, launch the PC software that was just installed called Touch-Screen Reader. A window similar to the one shown in [Figure 11](#) appears.

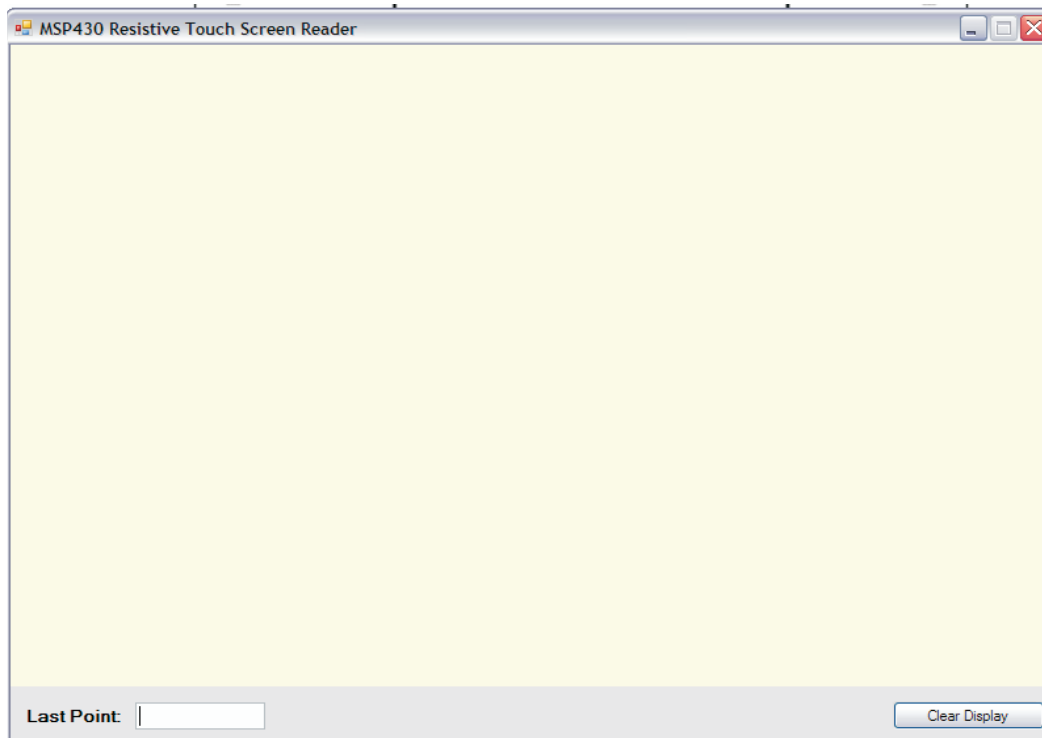


Figure 11. PC Software

The PC software opens COM1 and a connection to the board. As you touch the touch screen, the software draws the points you touch, and the coordinate of the last point are displayed in the text box at the bottom of the application. The Clear Display button at the bottom of the application clears the previous points drawn in the window.

4 References

1. MSP430x20x1, MSP430x20x2, MSP430x20x3 data sheet ([SLAS491](#))
2. *MSP430x2xx Family User's Guide* ([SLAU144](#))
3. *Implementing a UART Function With Timer_A3* ([SLAA078](#))

Revision History

Changes from Original (February 2008) to A Revision	Page
• Updated values in Flash and RAM columns in Table 1	5
• Updated Figure 8	7

NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

TI products are not authorized for use in safety-critical applications (such as life support) where a failure of the TI product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of TI products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by TI. Further, Buyers must fully indemnify TI and its representatives against any damages arising out of the use of TI products in such safety-critical applications.

TI products are neither designed nor intended for use in military/aerospace applications or environments unless the TI products are specifically designated by TI as military-grade or "enhanced plastic." Only products designated by TI as military-grade meet military specifications. Buyers acknowledge and agree that any such use of TI products which TI has not designated as military-grade is solely at the Buyer's risk, and that they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI products are neither designed nor intended for use in automotive applications or environments unless the specific TI products are designated by TI as compliant with ISO/TS 16949 requirements. Buyers acknowledge and agree that, if they use any non-designated products in automotive applications, TI will not be responsible for any failure to meet such requirements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		Applications	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DLP® Products	www.dlp.com	Communications and Telecom	www.ti.com/communications
DSP	dsp.ti.com	Computers and Peripherals	www.ti.com/computers
Clocks and Timers	www.ti.com/clocks	Consumer Electronics	www.ti.com/consumer-apps
Interface	interface.ti.com	Energy	www.ti.com/energy
Logic	logic.ti.com	Industrial	www.ti.com/industrial
Power Mgmt	power.ti.com	Medical	www.ti.com/medical
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
RFID	www.ti-rfid.com	Space, Avionics & Defense	www.ti.com/space-avionics-defense
RF/IF and ZigBee® Solutions	www.ti.com/lprf	Video and Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless-apps

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2010, Texas Instruments Incorporated