Elo TouchSystems

Troubleshooting and Technical Reference Guide

Elo T-shoot & Tech Ref

Note:

See the Elo Web site at **www.elotouch.com** for a complete list of formal documentation, downloadable drivers and complete product information.

This document is intended for reference purposes only, and is intended to be an adjunct to the formal set of Elo technical documentation. The goal is to provide information on the most common hardware and drivers, and to provide quick answers to the most common problems; therefore, some older controller models are not mentioned and not all problem situations are covered. Refer to the formal technical documentation for more detailed information.

Although some of the subject matter in the IntelliTouch technology, AccuTouch technology and General Notes sections goes into some depth, for the most part it is still a distillation of material that appears in the various Elo technical manuals.

Please send any comments, corrections or suggestions to jpeed@elotouch.com.

Elo TouchSystems Troubleshooting and Reference Guide - Index

Section 1 **elodiags.doc**

Step-by-step troubleshooting sequences. Starts with basic hardware tests, then proceeds to software checks for each operating system.

Section 2 **elodrivr.doc**

Particulars for most Elo software drivers. Lists supported controllers, filenames, default directory locations, registry keys, driver calls and locations, installation command, COM port fault error messages and driver removal procedure.

Section 3 **Driver Notes**

For driver specifics, see section 2.
These sections are for additional driver notes only.

Section 3a **DOS and Windows driver notes**
V2.0c

Section 3b **Windows 95 driver notes**
V1.82
V1.00

Section 3c **Windows NT driver notes**
V2.00

Section 4 **IntelliTouch technology notes**

Section 5 **IntelliTouch controller notes**

Section 6 **AccuTouch technology notes**

Section 7 **AccuTouch controller notes**

Section 8 **General notes**

Section 1 - elodiags.doc

Touchsystem troubleshooting

This section outlines a touchsystem troubleshooting regime. Following the guidelines in the order presented (hardware, then software checks for the applicable driver) should quickly lead to the resolution of most “no touch” problems.

Basic Touchscreen Troubleshooting Guide

Determine the operating system, touch technology and controller type

Symptom	Test Sequence
No touch response	**Setup for hardware diagnostic** **Do hardware diagnostic** **Do software operational checks**
Nonlinear touch response	**AccuTouch: Do resistance checks**
Erratic touch operation	**IntelliTouch: Check for noise**
Bands/areas of no touch response	**IntelliTouch: Check reflectors for dirt or blockage**

Which technology do they have?

Is it AccuTouch or IntelliTouch?

Look for dot grid (turn monitor so screen faces ceiling and direct bright light) -
Run ends of fingernails over screen -
Look/feel for dot grid.
If so, it's AccuTouch

If no dot grid, look at all bezel edges for reflectors-
If seen (not likely)**, it's IntelliTouch**

If they can't tell, it's probably IntelliTouch

What controller do they have?

Is it internal or external?

Female DB9 out of monitor -
-S on Model Number (Elo only) -
Next to last character of Part Number is N (Elo only) -
Internal serial

Male DB9 out of monitor -
No -S on Model Number (Elo only) -
Next to last character of Part Number is 0 (Elo only) -
No internal controller

Check the obvious - bus card; external serial box; direct connection to COM port -
but remember that they can cable it up wrong - **gender changers are a dead giveaway**!
When in doubt, **double check**.

All DB9 cables are M/F; any gender change adapters imply that they have made some wrong assumptions and hooked it up wrong.

If it's internal serial and not identified on the back of the monitor (3rd party):

Most will be SmartSet 2XX0 (2210,2300 or 2310). If there is any doubt, run COMDUMP; touch; look for data packet structure.

SmartSet: 10 bytes [55] [54] [Status] [Xlo] [Xhi] [Ylo] [Yhi] [Zlo] [Zhi] [checksum]
Accu: 2210 / Intelli: 2300/2310 (Zlo is always 0)

4002 (Intelli)	6 bytes	[Xhi] [Xlo] [Yhi] [Ylo] [Zhi] [Zlo]
140 (Accu)	4 bytes	[Xhi] [Xlo] [Yhi] [Ylo]
280 (old 4-wire)	3 bytes	[01] [X] [Y] (initial touch) ... [81] [X] [Y] (untouch)

Setup for touchscreen hardware operational check

(Macintosh - Not applicable)

For serial controllers, run COMDUMP.

- See http://www.elotouch.com/support/webtech/comdump.asp

For bus controllers, run BUSSTAT.

- To make a DOS boot diskette: On a DOS, Windows or Windows 95 system, get to a DOS prompt. Insert a floppy in the a: drive and use the command "sys a:".
- For NT systems, it is best to copy COMDUMP.EXE (or BUSSTAT.EXE) from the \elo directory of the hard drive to the a: drive (if the hard drive is formatted for NTFS, it will not be readable from DOS).

COMDUMP

DOS command: comdump 1 (for COM1 connection) or
comdump 2 (for COM2 connection).

If data scrolls on the screen when the screen is not being touched, a constant touch is indicated; for AccuTouch resistive screens, look for a damaged spot on the screen, and make sure the bezel isn't clamped down on the active area of the touchscreen.

Touch the screen; if data scrolls on-screen only when the screen is touched, the hardware is functioning. If data 55 54 repeats every ten bytes, a SmartSet 2XX0 controller is installed.

If no data scrolls on-screen:

⇒ Cycle the power to the controller (remove and re-install the power cable for monitors with internal serial controllers) and try COMDUMP again.
- The controller can lock up if an incorrect controller was specified.

⇒ Plug a serial mouse into the COM port and move the mouse; data indicates that the port is OK and the fault is in the touchscreen, controller or wiring.

⇒ Substitute touch system components if possible.

Check the Hshk item in the header

⇒ DSR CTS should be indicated
- DSR is high (5 volts) when power is applied to the controller.
- CTS is driven high (8 volts) when the controller completes the Power-On Self-Test.

⇒ The NT driver (Ver 2.0 and earlier) will not load if the proper handshaking is not indicated.

⇒ The Windows 95 driver (Ver 1.82) can be forced to load without proper handshaking by setting the following registry key to 00:

HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Class\EloTouchscreen\0000\HardwareHandShaking

The drivers for DOS will not load if the proper handshaking is not indicated, unless the -p- flag is used when loading ELODEV.

If handshaking is indicated as NONE

⇒ the controller may not be getting power
⇒ the COM port may have a problem or the wrong port may have been specified
⇒ there may be a cable problem. As a minimum, the following pins must be connected:

DB 9	to	DB9	or	DB25	
2		2		3	Rd
3		3		2	Tx
6		6		6	DSR
8		8		5	CTS
5		5		7	Gnd

BUSSTAT

DOS command: BUSSTAT

select the controller type from the list (model number is on the controller backplate)

enter the base address (ENTER key for default 280)

Data should scroll on screen only when the screen is touched.

If data scrolls on the screen when the screen is not being touched, a constant touch is indicated; for AccuTouch resistive screens, look for a damaged spot on the screen, and make sure the bezel isn't clamped down on the active area of the touchscreen. With 4035 IntelliTouch controllers, this can also be caused by an incorrect address or an address conflict with another device.

Touch the screen; if data scrolls on-screen only when the screen is touched, the hardware is functioning.

If no data scrolls on-screen:

- ⇒ With 4035 IntelliTouch controllers, make sure the IRQ is not being used by another device.
- ⇒ If AccuTouch, try the paper clip test (see page 5).
- ⇒ Substitute touch system components if possible.

BUSSTAT results with various address and IRQ conflicts

Wrong address	4035	Constant scrolling data
	2201	Controller not detected
Address shared with another device		
	4035	Constant scrolling data
	2201	BUSSTAT will appear to operate normally
IRQ shared with another device		
	4035	No touch response in BUSSTAT
	2201	BUSSTAT will appear to operate normally

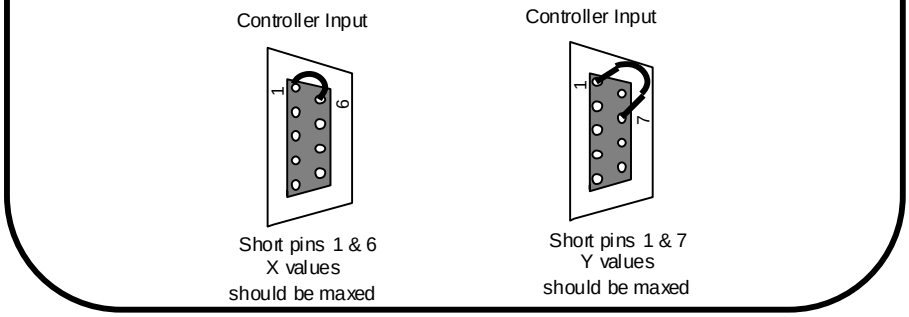
Note: The Microsoft diagnostic, MSD, cannot detect all conflicts, but it's worth running.

Paper Clip Test (for "no Touch Data" problems)

for AccuTouch ONLY!

Bus Controllers and External Serial Controllers

Run BUSSTAT/COMDUMP (as appropriate)

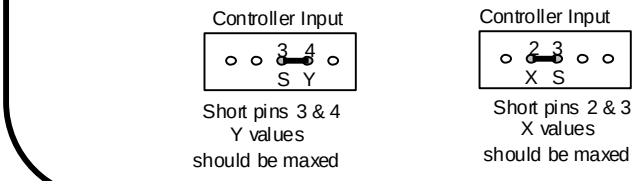


If the proper results are not obtained, the controller is at fault.

InternalSerial Controllers

(monitor must be opened to gain access to the controller input)

Run COMDUMP



In practice, shorting either pin set and seeing the data scroll is generally sufficient to verify basic controller operation.

External cables can be checked by connecting the cable to the controller and shorting the appropriate pins on the touchscreen-end of the cable; if data scrolls, the cable is OK (in the case of the cable, it might be wise to check both pin sets).

Software Operational Checks

First, if a serial controller is being used, make sure handshaking is indicated as DSR and CTS (see COMDUMP).

Next, check the controller applicability per the list below. Bus controller model numbers can be found on the controller card backing plate. Most controllers that are inside the monitor will be SmartSet 2XX0 type (2210, 2300 or 2310). *Note that a common mistake is to specify the older IntelliTouch 4002 controller when an IntelliTouch 2310 is installed; this controller should be specified as the SmartSet 2XX0.*

DOS & Windows - Supported controllers: 140, 2210, 2300, 2310 and 4002 serial; 141, 4025, 4035 and 2201 bus.

Make sure the controller type and parameters are correct in the ELODEV command line in the AUTOEXEC.BAT file.

Proceed to "DOS and Windows drivers: Format and Summary."

Windows 95 - Supported controllers: 2210, 2300, 2310 serial (Select SmartSet 2XX0 for all); 4025, 4035 and 2201 bus.

Make sure that the V1.82 driver is installed (Control Panel, Elo Touchscreen; a five-tab tabset should be identified as Elo Touchscreen Properties; the Touchscreen Control Panel will be identified as Version 2.0 in the About tab).

Make sure the controller type and parameters are correct in the Control Panel, System, Device Manager, Elo Touchscreen entry.

Proceed to "Windows 95 V1.82 drivers: Troubleshooting Checklist."

Windows NT - Supported controllers: 2210, 2300, 2310 (Select SmartSet 2XX0 for all) and 4002 serial; 4025, 4035 and 2201 bus.

Make sure the controller type and parameters are correct in the Control Panel Elo Touchscreen Setup dialog box.

Proceed to "Windows NT V2.0 drivers: Troubleshooting Checklist"

.

DOS and Windows drivers: Format and Summary

The proper driver calling sequence in AUTOEXEC.BAT is:

(path) mouse	(typically c:\mouse\mouse)
c:\touch\nomouse	(optional if mouse is called)
c:\touch\elodev (parameters)	(example: c:\touch\elodev 2210, 1, 9600 -c## ## ## ## ## ##)
	(example: c:\touch\elodev 4035, 280,5 -c## ## ## ## ## ##)
c:\touch\monmouse (parameters)	(typically c:\touch\monmouse -m6)

The -c ## ## ## ## in the ELODEV line is the calibration points. If the calibration points are not present, ELODEV will load but monmouse will not load. Run ELOCALIB to calibrate and add the calibration points to the ELODEV line.

DOS Driver Summary:

Either mouse or nomouse must load

mouse will not load if a mouse is not attached and detected

ELODEV must load and include DOS calibration points

ELODEV will not load if a controller is not detected.
ELODEV will not load if DSR and CTS are not detected (serial) (unless -p- flag is used).
ELODEV will not be calibrated if the installation was aborted

monmouse must load

monmouse will not load unless mouse (or nomouse) is loaded.
monmouse will not load if ELODEV is not loaded
monmouse will not load if ELODEV is not calibrated for DOS

In addition, if running Windows, the following lines are required in SYSTEM.INI:

[boot] section
mouse.drv=monmouse.drv

[boot description] section (this is a descriptor and is not actually required)
mouse.drv=Elo TouchSystems MonitorMouse(R) for Windows

[386Enh] section
mouse=vmmd.386

All other mouse calls should be commented out (line begins with a semicolon)

Windows calibration is done separately in the Control Panel/Touchscreen/Calibrate function.

The Windows drivers will not operate unless all the DOS drivers are properly loaded.

DOS and Windows drivers: Troubleshooting Checklist

NOTE: To correct Run Time Error 6002, Divide by Zero (P-133 and faster): Install ELODEV 1.7d

No touchscreen response

⇒ Run **TUTORIAL**.

- ♦ If the tutorial screen comes up, then all DOS drivers are loaded.
  Touch "Begin Lesson"
  If response is OK: DOS calibration is correct and no IRQ conflicts exist
  If no response: IRQ conflict exists
  If response, but not at touch point: improper calibration or nonlinearity problem

- ◊ "Mouse driver not installed" error message?
  Check AUTOEXEC.BAT for mouse or nomouse call.
  Check for correct path in mouse or nomouse call.
  Check for mouse driver in specified directory.
  Check for mouse physically attached.
  (mouse will not install if a mouse is not connected to the system).

- ◊ "Monmouse driver not installed" error message?
  - ⇒ Run **INFO**
    - ∗ "Elodev not installed" message?
      - ⇒ Try loading ELODEV from the command line, using the correct parameters for the installed controller.
        - ELODEV installed OK from command line?
          Check ELODEV command line in AUTOEXEC.BAT for errors.
        - ELODEV fails to install from command line?
          Error message? Check for line faults (if serial)

Cannot output to controller	CTS	open or grounded
Controller not detected.	DSR	open or grounded or no power to controller
Controller not responding.	RX	open or grounded
	TX	open or grounded
	RTS	grounded
	DTR	grounded

          Check power/COM port/base address as applicable.
        - Correct the problem in either case. Reboot and try again.
    - ∗ Calibration points are all zeros?
      - ⇒ Run democal or ELOCALIB (save points in file if ELOCALIB).
        - Reboot and retry.

A note on the installation sequence: After installation completes, GO.BAT will calibrate the touch system in four DOS video modes. When GO.BAT completes, it is erased and cannot be run again. Use ELOCALIB for any additional calibration.

DOS and Windows drivers: Troubleshooting (continued)

⇒ To confirm proper calls and AUTOEXEC.BAT load sequence, use the Step-by-Step boot option. Confirm that either nomouse or a mouse driver loads (note the version). Confirm that ELODEV loads (note any error messages). Confirm that monmouse loads (note any error messages).

⇒ Make sure that the touch driver calls are placed before the WIN call.

Incorrect/Erratic operation

⇒ Check mouse driver version (Microsoft: 8.2 preferred; Logitech: 6.2 preferred).
- To confirm bad mouse driver, try nomouse instead of mouse.
- Mouse 9.01 will cause extremely erratic mouse operation if not properly installed.

⇒ A number of things can cause erratic operation; study the DOS and Windows Driver Guide, or call Elo Tech Support.

⇒ For Windows no-touch problems (if all DOS drivers are loaded per above)

- Make sure the touchscreen is calibrated for Windows operation.
  Use the calibration function in the Control Panel Touchscreen applet.
- Check the Windows touchscreen and mouse driver calls in the SYSTEM.INI file

An additional DOS utility and its uses:

TOUCHES

TOUCHES displays touch coordinates. Unlike COMDUMP or BUSSTAT, *TOUCHES* uses ELODEV. Therefore, if ELODEV is calibrated, *TOUCHES* will display the touch coordinates in 80 column by 25 row format. If ELODEV is not calibrated, *TOUCHES* will display touch coordinates as raw coordinate data. Because ELODEV is interrupt-driven, *TOUCHES* can indicate IRQ conflicts.

Best uses:

COMDUMP/BUSSTAT OK, but *TOUCHES* is dead: **an IRQ conflict is indicated**.
Indicates the hardware is OK, but IRQ-dependent ELODEV cannot function.

Raw data: ELODEV not calibrated. **Column/row data: ELODEV is calibrated**

A note on the installation sequence: After installation completes, GO.BAT will calibrate the touch system in four DOS video modes. When GO.BAT completes, it is erased and cannot be run again. Use ELOCALIB for any additional calibration.

Windows 95 V1.82 drivers: Troubleshooting Checklist

NOTE: The Windows 95 driver does not support full screen DOS operation.

Windows 95 driver Ver. 1.00 should be replaced with the Ver 1.82 driver, available on the Web at www.elotouch.com, in the File download area.

For Windows Protection Errors on boot (unlikely, but possible), go directly to the last item in this checklist.

No touchscreen response

⇒ Make sure the screen is not touched during Windows 95 boot
- Touches during Windows 95 boot will make the touchscreen inoperative

⇒ If serial controller, make sure the handshake is indicated as DSR and CTS
- Use COMDUMP to verify
  - If not, see note below

Note:
Hardware handshaking lines DSR and CTS are normally required for this driver to load. To tell the driver to ignore hardware handshaking, run REGEDIT and set the following registry key to "00":

HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Class\EloTouchscreen\0000\HardwareHandShaking

⇒ If unsure about the controller type, location and parameters: go to Control Panel, System, Device Manager, Elo Touchscreen. Double-click Elo Touchscreen to display the current configuration. To change configuration (COM port, address/IRQ, etc.), double-click the secondary touchscreen icon (the one indicating current configuration), select the Properties tab and click the down arrow to view alternative configurations.

⇒ Bus Controllers: make sure there are no IRQ or I/O port conflicts in Windows 95
- Confirm in Control Panel\System\Device Manager\Properties

⇒ If all above is OK, try removing and re-installing the driver per the bulleted items below. Confirm the controller and hardware configuration first, so the correct installation can be selected when prompted. Note which controllers are supported.

- Go to Control Panel and double-click System, then click the Device Manager tab.
- Double-click the EloTouchscreen icon.
- For each instance of touchscreen, click the icon, click the Remove button, then click OK.
- Reboot the system and re-install the Win 95 V1.82 Elo touchscreen driver.

Windows NT V2.0 drivers: Troubleshooting Checklist

No touchscreen response

⇒ If serial controller, make sure the handshake is indicated as DSR and CTS.
- Boot to DOS, run COMDUMP to verify (See note below on DOS boot disk).
- The NT touchscreen driver will not load without these handshake signals.
- See COMDUMP section, next page.

⇒ If serial controller, make sure COMDUMP shows data.
- The NT touchscreen driver will load and show no errors even if the serial data line (RD) is not operational; data cannot be sent and the touchscreen can't work.

⇒ Make sure the controller and parameters are correct in the Control Panel Elo Touchscreen Setup dialog box. For 2310, 2300 and 2310 controllers, select SmartSet 2XX0. Select IntelliTouch 4002 only if you are certain that a 4002 controller is installed (this is an older controller model). The old 140 and 141 controllers are not supported.

⇒ Confirm monmouse and mouseclass Status and Startup configuration.
- Confirm in Control Panel\Devices.
  - Monmouse must be indicated as Started, System.
  - Mouseclass must be indicated as Started, System.

⇒ Bus Controllers: make sure there are no IRQ or I/O port conflicts in NT.
- Confirm in Start\Programs\Administrative Tools\Win NT Diagnostics\Resources.
  - IRQ tab shows all IRQ assignments.
  - I/O Port tab shows all I/O address assignments.
- Devices tab - double-click Monmouse to show assigned IRQ and I/O port.

⇒ Check the BIOS.
- Make sure the needed resources are not available for allocation to PCI or plug and play devices.
- Force the COM port to manual (or specific COM port) if possible.
- Reserve the needed resources if possible.
  - Disable parallel (printer) port if IRQ7 is being used for bus controller.

- *To make a DOS boot diskette: On a DOS, Windows or Windows 95 system*, get to a DOS prompt. Insert a floppy in the a: drive and use the command "format a: /s".
- It is a good idea to copy COMDUMP.exe from the hard drive to the boot disk; that way, if COMDUMP is not accessible from the hard drive (due to NTFS disk format), COMDUMP can be run directly from the boot disk.

NT error messages with Elo serial controllers and MMNT V2.0

Three error messages appear in the NT event log when Elo serial controllers are used with Windows NT. The reason is that the Elo serial driver accesses the COM port directly, using the COM port address and IRQ, while NT wants all devices to use the NT virtual communications interface. **The errors are benign and may therefore be ignored.**

The error messages are:

"The hardware resources for com(x) are already in use by another device."

"A conflict has been detected between two drivers which claimed two overlapping IO port regions. Driver Monmouse, with device <\Device\PointerPort0.Translated> claimed an IO port range with starting address in data address 0x28 and 0x2C, and length in data address 0x30."

"A conflict has been detected between two drivers which claimed two overlapping IO port regions. Driver Serial, with device <\Device\Serial10000.Translated> claimed an IO port range with starting address in data address 0x28 and 0x2C, and length in data address 0x30."

Section 2 - elodrivr.doc

Elo Driver Specifics

This section contains the particulars for most Elo TouchSystems drivers.

For each driver:

Supported controllers are listed.

Filenames and directory locations are listed.

Registry keys, if any, are listed.

Driver call locations are specified.

Installation command is listed.

COM port line fault error conditions and messages are listed.

Removal procedure is listed.

DOS & Windows Drivers, V 2.0c

Supports 140, 4002, 2210, 2300, 2310 serial; 141, 4025, 4035 and 2201 bus controllers.

File name	Directory	Size	Date
elodev.exe	\touch	19K	2/28/96
monmouse.com	\touch	16K	2/16/96
nomouse.com	\touch	2K	6/ 2/95
cursoff.exe	\windows	6K	7/14/92
monmouse.drv	\windows\system	7K	12/21/93
vmmd.386	\windows\system	10K	8/30/94
elocalw.cpl	\windows\system	62K	3/26/96
elocalw.hlp	\windows\system	17K	1/11/96

Registry entry: Not Applicable

Driver calls:

Autoexec.bat

(path) mouse

c:\touch\nomouse

c:\touch\elodev (parameters)

c:\touch\monmouse (parameters)

SYSTEM.ini

[boot]
mouse.drv=monmouse.drv

[boot description]
mouse.drv=Elo TouchSystems MonitorMouse(R) for Windows

[386Enh]
mouse=vmmd.386

To install: a:\install ***(from DOS only)***

DOS & Windows Drivers, V 2.0c (continued)

Handshake fault error messages:

Error message location: On the DOS command line following the ELODEV command.

Signal	Condition	Error Message
CTS	open or grounded	*Cannot output to controller*
DSR	open or grounded	*Controller not detected.*
RX	open or grounded	*Controller not responding*.
TX	open or grounded	*Controller not responding*
RTS	open *grounded*	Normal operation if not pulled low C*ontroller not responding*
DTR	open *grounded*	Normal operation if not pulled low C*ontroller not responding*

To remove all DOS & Windows touchscreen driver components:

Delete the contents of the \touch directory

Remove \touch directory

Delete the following file from the \windows directory:

cursoff.exe

Delete the following files from the \windows\system directory:

monmouse.drv
vmmd.386
elocalw.cpl (failure to delete this will give you and extra icon in Control Panel)
elocalw.hlp
elologo.fot
elologo_.ttf

Edit AUTOEXEC.BAT
delete all lines with the \touch\ path (ELODEV, monmouse and maybe nomouse)
delete the mouse call, if present (if running Windows 95)

Edit SYSTEM.ini
in [Boot] section, delete mouse.drv=monmouse.drv line
in [Boot Description] section, delete mouse="Elo..." line
in [Enh386] section, delete mouse=vmmd.386 line

Edit win.ini
delete [elocalw] section

Re-install the Windows mouse, if necessary.

Windows 95 Driver, V 1.82

Supports: 2210, 2300, 2310 serial; 4025, 4035 and 2201 bus controllers.

File name	Directory	Size	Date
monmouse.hlp	\windows\system	16K	11/22/96
monmouse.cpl	\windows\system	548K	3/24/97
monmouse.vxd	\windows\system	46K	5/ 6/97
elocal32.exe	\windows\system	80K	1/27/97
rbutton.exe	\windows\system	22K	2/19/97
mminst.dll	\windows\system	27K	3/25/97
oem#.inf	\windows\inf	7K	9/25/96 (internal date)
Null.cur	it's in there somewhere, but need not be removed		

Registry entry:

HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Class\EloTouchscreen

Driver call: from registry

To install:

Before installing, be sure that no needed resources are reserved in Windows 95 (Control Panel/System/Device Manager/Properties/Reserve Resources) - if resources to be used by the touchscreen driver are reserved, remove the reservations. Failure to do so will give a message saying that the resources are reserved for use by another device, and you will not be able to install this driver.

- To unzip the driver, double-click the filename; by default, a directory named elowin95 will be created on the C: drive and the driver will be unzipped to that directory.

1. Click the Start button.
2. Choose Settings/Control Panel/Add New Hardware.
3. Click Next.
4. Choose No when the system asks whether you want to search for new hardware; click Next
5. Scroll down to and double-click Other Devices.
6. Choose Have Disk
7. Click Browse; double-click the C:\ folder; scroll down to the elowin95 directory and double-click it; click OK.
8. In the Install From Disk dialog, the path should be indicated as C:\elowin95; click OK.
9. Choose the correct controller and configuration, then click Next.
10. Click Finish to complete the installation.
11. Restart the computer when prompted; touch the calibration points when they appear after boot.

Note: Hardware handshaking lines DSR and CTS are normally required for this driver to load. To tell the driver to ignore hardware handshaking, run REGEDIT and set the following registry key to "00":

HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Class\EloTouchscreen\0000\HardwareHandShaking

Windows 95 Driver, V 1.82 (continued)

Handshake fault error messages:

Error message location: Diagnostics Tab of the Elo Touchscreen Properties in the Control Panel

Signal	Condition	Error Message
CTS	open or grounded	*Controller reports an unknown error.* In either case, the driver will not load.
DSR	open or grounded	*Controller reports an unknown error.*
RX	open or grounded	DIAGNOSTICS NORMAL, CALIBRATION ENABLED, BUT NO RESPONSE TO TOUCH.
TX	open or grounded	Normal operation
RTS	open or grounded	Normal operation
DTR	open or grounded	Normal operation

To remove all driver components:

1. **Using Explorer**, delete the following:

\windows\system\elocal32.exe
\windows\system\monmouse.vxd
\windows\system\monmouse.hlp
\windows\system\monmouse.cpl
\windows\system\rbutton.exe
\windows\inf\oem#.inf

NOTE1: There may be more than one oem#.inf file (where # is a number). Double-click each one to open the text and look for the one pertaining to the touchscreen. When the oem.inf file for the touchscreen has been identified, delete that file.

NOTE 2: If the monmouse.cpl file cannot be deleted because it is in use by Windows, perform the rest of the procedure below, then select Start, Shut down, Restart the computer in MS-DOS mode; then cd \windows\system; then delete monmouse.cpl.

2. **Using Regedit**, delete the following Registry Branch:

HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Class\EloTouchscreen

(Start button, Run button, enter regedit. Double-click each key to proceed to the Class key; click Elo Touchscreen, hit the delete key, then the ENTER key). Exit regedit.

Win 95, V 1.00 Driver

Supports: 2210, 2300, 2310 serial; 4025, 4035 and 2201 bus controllers.

File name	Directory	Size	Date
monmouse.hlp	\windows\system	17K	9/ 3/96
monmouse.cpl	\windows\system	110K	9/ 3/96
monmouse.vxd	\windows\system	35K	9/ 3/96
	\Elo (readme, BUSSTAT, null.cur, COMDUMP, sawdump)		

Registry entries:

HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\VxD\MonMouse

HKEY_LOCAL_MACHINE\Software\Elo TouchSystems

HKEY_CURRENT_USER\ControlPanel\Desktop - DoubleClickHeight, DoubleClickWidth

Driver call:

SYSTEM.ini

[386Enh]
device=monmouse.vxd

Autoexec.bat: no touchscreen or mouse calls

To install: Start button, Run button, enter a:\setup

Handshake fault error messages:

Error message location: Message box - when the Control Panel Touchscreen Applet is opened

Signal	Condition	Error Message
CTS	open or grounded	NOTE: IF EITHER LINE IS GOOD , TOUCHSCREEN OPERATION WILL BE NORMAL
and	Both bad:	T*he controller specified in the Elo Touchscreen Control Panel Setup was not found.*
DSR	open or grounded	NOTE: IF BOTH LINES ARE BAD , TOUCHSCREEN WILL NOT OPERATE
RX	open or grounded	*No response from controller. Controller or touchscreen may have become disconnected or unplugged.*
TX	open or grounded	*No response from controller. Controller or touchscreen may have become disconnected or unplugged.*
RTS	open or grounded	Normal operation
DTR	open or grounded	Normal operation

Win 95, V 1.00 Driver (continued)

To remove:

1. Close the Windows Control Panel.

2. Using EXPLORER, delete the following files from \Windows\System folder:
   monmouse.vxd
   monmouse.hlp
   monmouse.cpl.

   Delete the Elo folder.

3. Start the REGEDIT program to edit the registry
   (Start, Run, type in *regedit*, hit ENTER).

   Delete the following keys from the registry:

   HKEY_CURRENT_USER\ControlPanel\desktop\DoubleClickHeight
   HKEY_CURRENT_USER\ControlPanel\desktop\DoubleClickWidth

   Delete the following registry key folders and their contents:

   HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\VxD\MonMouse HKEY_LOCAL_MACHINE\SOFTWARE\Elo TouchSystems

4. Run SYSEDIT; select SYSTEM.ini file; delete the following line from the [386Enh] section:

   device=monmouse.vxd

5. Restart Windows 95.

Windows NT, V 2.0 Driver

Supports: 4002, 2210, 2300, 2310 serial, 4025, 4035 and 2201 bus controllers.

File name	Directory	Size	Date
monmouse.hlp	\winnt\system32	16K	2/17/97
monmouse.cpl	\winnt\system32	148K	2/26/97
monmouse.sys	\winnt\system32\drivers	27K	2/25/97
(diags & readme)	\elo		

Registry entries:

HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\MonMouse

HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\MonMouse\Parameters

HKEY_CURRENT_USER\ControlPanel\Mouse\DoubleClickHeight & DoubleClickWidth

Driver: Monmouse.sys is the actual driver.

To install: Start button, Run button, enter a:\setup

NOTE: YOU MUST BE LOGGED ON AS THE SYSTEM ADMINISTRATOR TO INSTALL THE SOFTWARE
(otherwise, the install program will not be able to modify the system registry)

Handshake fault error messages:

Error message location: Message boxes at boot time and when the Control Panel Touchscreen Applet is opened

Signal	Condition	Error Message
CTS	open or grounded	At boot: *At least one driver or service failed during system startup*. At Touchscreen applet open: *Monitormouse for NT device driver not installed.* In either case, Modes and Cal will be grayed out; cal inoperative.
DSR	open or grounded	At boot: *At least one driver or service failed during system startup*. At Touchscreen applet open: *Monitormouse for NT device driver not installed.*
RX	open or grounded	DIAGNOSTICS NORMAL, CALIBRATION ENABLED, BUT NO RESPONSE TO TOUCH.
TX	open or grounded	Normal operation
RTS	open or grounded	Normal operation
DTR	open or grounded	Normal operation

Verifying the Windows NT registry (for troubleshooting only):

(the registry may be verified or changed directly using regedit - to make changes, you must be logged on as the system administrator)

From Windows NT, run regedit. Verify the following parameters in the registry.

HKEY_LOCAL_MACHINE
System
CurrentControlSet
Services
MonMouse

Type	REG_DWORD	0x1
Start	REG_DWORD	0x1

Parameters

ControllerAddress	REG_DWORD		
		0x3f8	*
		0x2f8	**
		0x280	***
ControllerLevel and Controller vector	REG_DWORD		
		0x4	*
		0x3	**
		0x5	***
ControllerModel	REG_DWORD		
		0x8a2	(2210)
		0x8fc	(2300)
		0x906	(2310)
		0xfa2	(4002/2856)
		0x899	(2201)
		0xfc3	(4035)
Controller Type	REG_DWORD		
		0x0	* or **
		0x01	***

*	if serial controller connected to com1
**	if serial controller connected to com2
***	if bus controller at default settings

To remove all NT driver components:

Delete the files:

delete from winnt\system32\drivers
monmouse.sys

delete from winnt\system32
monmouse.cpl
monmouse.hlp

delete the \elo directory and its contents

NOTE: YOU MUST BE LOGGED ON AS THE SYSTEM ADMINISTRATOR TO DELETE THE REGISTRY ENTRIES
(otherwise, you will not be able to modify the system registry)

Using regedit, remove the Monmouse Parameters and MonMouse entries from the registry.

(Run Regedit, open path to monmouse key, highlight the monmouse key and hit the DEL key to delete monmouse and its associated keys)

HKEY_LOCAL_MACHINE
System
CurrentControlSet
Services
MonMouse
Parameters

Web Enabler, V 1.00

File name	Directory	Size	Date
enabler.exe	\program files\web enabler	1382K	5/30/97
sampexe.exe	\program files\web enabler	11K	4/14/97
startup.htm	\program files\web enabler	1K	4/14/97
Uninst.isu	\program files\web enabler	37K	8/12/97
webkiosk.res	\program files\web enabler	4K	8/12/97
wininfo.exe	\program files\web enabler	15K	4/14/97
example folder	\program files\web enabler		

Registry entry:

HKEY_LOCAL_MACHINE\Software\Elo TouchSystems, Inc

To install:

d:\setup

To remove:

Control Panel, Add/Remove Programs; choose Web Enabler; click Add/Remove Programs; click Yes.

Before re-installing, you must delete the webkiosk.res file, which is located in \Program Files\Web Enabler\webkiosk.res.

Section 3 - Driver Notes

NOTE: For driver specifics, see section 2.

These sections are for additional driver notes only.

Section 3a - DOS & Windows driver notes

Section 3b - Windows 95 driver notes

Section 3c - Windows NT driver notes

3a- DOS & Windows driver notes

NOTE: For driver specifics, see section 2.

This section is for additional DOS & Windows driver notes only.

ELODEV EXE

Command line flags (options)

The ELODEV command line may be run from the DOS prompt or, typically, will be inserted into AUTOEXEC.BAT, to execute on boot up. Optional flags can be specified to alter the default settings, as described below. For a complete list of options on-screen, type ELODEV, <ENTER>.

-c<xl, xh,yl,yh,zl,zh) Calibration points, in DECIMAL
determined and added to autoexec ELODEV command line by ELOCALIB

-s<+ or -> X,Y axis swapping
-s+ rotates orientation 90 degrees
-s- is normal orientation (as is no s flag)

-i<##> Software interrupt number, in HEX
This interrupt is independent of the hardware interrupt
This software interrupt is used by other applications to access ELODEV
Default is 65; possibilities are 60 thru 66

NOTES: To change the software interrupt with the -i flag, ELODEV must first be extracted from memory (see -x flag, below), and then re-run with the desired -i<number>

-w<##> Untouch wait time
Time, in milliseconds, before the program recognizes an untouch.. Default may need to be lengthened to eliminate false untouches. Test for false untouches with the *TOUCHES* utility (but be aware that noise can also give false untouches). Try longer wait times (start at -w90):
for slow baud rates;
with >16" touchscreens and pre-2.2 firmware in 4002 controllers

-h Display quick reference info
(version #, command line syntax, controller list, flags)

-p<+ or -> Program the controller
Use -p- to ignore handshaking (when problems exist with handshake lines)
Default is -p+, which enables controller programming

-x Extract ELODEV from memory
this is a stand-alone command, i.e. elodev -x
ELODEV is normally loaded from AUTOEXEC.BATand remains in memory until shutdown. Extraction removes it from memory, but does not remove the command line from the AUTOEXEC.BAT file. Extraction works only if no other TSRs have been loaded after ELODEV was loaded.

To update an existing memory-resident copy of ELODEV with new flag choices, invoke ELODEV from the DOS prompt with the flag info you want; do not include the controller and additional normal command line info. Example: elodev -w90.

If the software interrupt has been changed using the -i flag, then the non-default interrupt must be used to access ELODEV for any future flag changes (as well as to load ELODEV in the future). Example: elodev -w90 -i61, if -i61 was previously used. This is also true for extracting ELODEV, i.e. elodev -x -i61.

Flags for MONMOUSE **for DOS**. Upper or lower-case. Spaces separate multiple flags.

NOTE: **On recalibration, MonitorMouse for DOS must be re-run so it will recognize the new values.**

Flag	Function	Defaults settings	
-m<mode>	Mouse button emulation mode 0,1, 5, 6.	-m0	Click on touch
-c<+/->	Mouse cursor on/off.	-c+	Cursor on
-b<+/->	Beeps on/off.	-b+	Beep on
-d<directory>	Path where ELOGRAPH.CAL is found.	-d	<elograph.cal>
-r<x>,<y>	Default mickey/pixel ratio.	-r8,16	8 horiz, 16 vert
-j<+/->	Jump/don't jump to touch with mickeys.	-j-	mickeys off
-s<+/->	Emulation active/inactive in spawned programs.	-s+	active in spawn
-t<+/->	Emulation active/inactive after program terminations.	-t-	inactive on term
-i<interrupt>	Software interrupt of ELODEV (hex).	-i65	(or first found)
-x	Extract MonitorMouse for DOS from memory.		
-h	Display quick reference information.		
-w	Scaling factors (undocumented)		

Mouse Button Emulation Mode Flag (-m) (DOS default = 0; Win default = 6)

Mode 0 Left button click on touch.
Mode 1 Left button click on **un**-touch (cursor follows finger until untouch).
Mode 5 Left button clicking and dragging (click on untouch).
Mode 6 Same as Mode 5, but supports double-clicking.
(Three additional modes are available for backwards compatibility but are no longer documented).

Cursor Flag (-c) Default: -c+ To turn cursor off, use -c-.

No effect if the application draws its own cursor; no effect in Microsoft Windows.

Calibration File Directory Flag (-d)

Path to ELOGRAPH.CAL, i.e., MONMOUSE -m6 -dC:\MYCAL

ELOGRAPH.CAL is produced by ELOCALIB.EXE, and contains a list of calibration points for each video mode used. MonitorMouse for DOS will automatically recalibrate itself using the values from ELOGRAPH.CAL as you switch video modes. When MonitorMouse for DOS is not active, the original calibration points are restored.

When MonitorMouse for DOS is installed, it looks for ELOGRAPH.CAL in the directory where MONMOUSE.COM is (typically \TOUCH), then the current directory, unless specified otherwise with the -d flag. If this flag is not specified, and ELOGRAPH.CAL is not found in the default directories, MonitorMouse for DOS uses ELODEV's calibration for all video modes. If ELOGRAPH.CAL is loaded, and your application uses a video mode not listed in the file, MonitorMouse for DOS will use the calibration values of the closest video mode, or the default calibration.

If you are using MonitorMouse for Windows, the Touchscreen Control Panel will allow calibration in the video mode used by Windows, even if the mode is not supported by ELOCALIB.

Mickey/Pixel Ratio Flag (-r) Default: -r 8,16 (same as mouse driver)

This flag specifies the default "mickey-to-pixel ratio" for horizontal and vertical motion. **A "mickey" is a unit of distance, typically 1/200 of an inch.** Mickey/pixel ratios specify how far the mouse must move for the cursor to move a given number of pixels. Because **MonitorMouse for DOS uses absolute positioning** rather than relative motion, your finger may appear out of calibration when using certain programs which use mickeys rather than coordinates.

The -r flag is required with some mickey-based (relative motion) programs, when:

The program uses motion from the mouse driver rather than coordinates, ***AND***

The program draws its own cursor or cursor bar, ***AND***

The program performs its own mickey-to-pixel conversion, and changes the ratios without informing the mouse driver through mouse function 15.

If your application fits these criteria, increase the value if the cursor falls behind your finger when sliding, or decrease it if the cursor speeds ahead of your finger.
The defaults for this flag are the same as they are for the mouse driver: **-r 8,16** (8 for X and 16 for Y).

Jump Flag (-j) Default: -j-

The **-j flag is only of concern in programs which use mickeys** (relative motion) rather than coordinates. With -j+, a new touch will emulate mickey motion to the new touch location from the point of last release, thus causing the cursor to jump to your fingertip. With -j-, a new touch will not simulate any motion to that point. The default is -j-

.Spawn Flag (-s) Default: -s+

Instructs MonitorMouse for DOS to remain active when an application *spawns* another program. **This flag is required for some off-the-shelf software to run properly.**

Technically speaking, if the application sets up a mouse interrupt routine, then spawns another program which expects to receive mouse interrupts, MonitorMouse for DOS must remain active after the spawn.
If the -s- flag is specified, MonitorMouse for DOS will disable itself when the program spawns. If you spawn non-mouse-driven programs within a DOS shell, you may want to use the -s- flag to disable MonitorMouse for DOS while in the spawned programs. However, if a spawned program makes a mouse call, MonitorMouse for DOS will automatically re-enable itself.

Terminate Flag (-t) Default: -t-

The -t+ flag may be required with some mouse-driven applications that do not respond at all to the touchscreen. See Appendix B.

Instructs MonitorMouse for DOS to remain active after a program *terminates*.

Normally, you will want MonitorMouse for DOS to disable itself when a program terminates. **This prevents beeps when you touch the screen at the DOS command level and in programs which do not use the mouse.** MonitorMouse for DOS will always automatically re-enable itself if a program makes a mouse call.

Interrupt Flag (-I)

The -i<n> flag allows you to specify the software interrupt of ELODEV in hex**. This flag is necessary if more than one copy of ELODEV is loaded**, as when controlling two touchscreens. You will also need it if ELODEV is installed at a software interrupt other than 60 through 66 (hex). **If the -i flag is not specified, the first ELODEV found when checking interrupts 60-66 will be used.**

Help Flag (-h)

At anytime, type: **MONMOUSE -h** to display quick reference information on all flags **and the current version of MonitorMouse for DOS**.

Scaling factors flag (-w)

Note: Undocumented. Not to be confused with ELODEV -w untouch wait time flag.

-wXlow,Xhigh,Ylow,Yhigh

-w0,16383,0,16383 (standard scaling factors)

Updating Flags

You may re-run MonitorMouse for DOS to update the flags of an installed copy (except -I flag; extract and reload monmouse with new value)

If a flag is not specified, it is left unchanged.

EloCalib.EXE

Calibration aligns the touchscreen with the display.

Each video mode requires a separate calibration.
After the command “ELOCALIB”, the following table appears:

ELOCALIB 1.6 - Copyright 1987-1994 EloTouchSystems, Inc.

Mode	Description	Resolution	Colors	Xlow Ylow Xhi Yhi Zlow Zhi
01	**CGA Text**	**40 x 25**	**16 Colors**	Note: after calibrating in one or more of the video modes, the six right columns will be filled with the calibration coordinates. These are stored in elograph.cal and accessed by the mode number.
03	**CGA Text**	**80 X 25**	**16 Colors**	
04	**CGA Graphics**	**320x200**	**4 Colors**	
06	**EGA Graphics**	**640x200**	**2 Grays**	
00	**EGA Graphics**	**320X200**	**16 Colors**	
0E	**EGA Graphics**	**640x200**	**16 Colors**	

INFO

INFO.EXE is a diagnostics program that allows you to confirm ELODEV settings. INFO is loaded in the selected directory (usually C:\Elo) during the install process of most Elo driver programs. INFO requires DOS and ELODEV INFO can be used with all Elo controllers.

```
C:\ELO> INFO
Controller:          2210 Rev: 1.3-0.0 Diags: 00-00 ID:
Base Port:
Interrupt (IRQ):
COM Port:
Baud Rate:
Data Bits:
Stop Bits:
Parity:
Wait Time:
ELODEV Version:
Software Interrupt:
Buffer Size:
Calibration Points: X=   666..  3438   Y=  3350..   809   Z=     1..
Scaling Range:      X=     1..    80   Y=     1..    25   Z=     1..
XY Swap Flag:
Mode:               0007H =
Open Flag:
Enable Flag:
Error Flag:
Zone Table:
User ISR:
TSRLockFlag:
```

NOTE: TSRLockFlag indicates that MONMOUSE (or other TSR using ELODEV) is loaded.

To use COM3 or 4 with the Elo DOS and Windows drivers:

Specify the COM port normally, then add a fourth parameter to specify the IRQ to be used with the COM port.

Standard ELODEV call for COM1:

elodev 2310,1,9600 -cXlo,Xhi,Ylo,Yhi,Zlo,Zhi

ELODEV call for COM3, using IRQ 10 for COM3:

elodev 2310,3,9600,10 -cXlo,Xhi,Ylo,Yhi,Zlo,Zhi

ELODEV call for COM4, using IRQ 15 for COM4:

elodev 2310,4,9600,15 -cXlo,Xhi,Ylo,Yhi,Zlo,Zhi

3b- Windows 95 Driver Notes

NOTE: For driver specifics, see section 2.

This section is for additional W-95 driver notes only.

Elo Windows 95 V1.82 driver anomalies

- If the Elo touchscreen Properties tabset is open but covered by another window, it does not come to the front when the Elo icon in the Control Panel is double clicked. The effect is that a double click-to-open command appears to be ignored. If the Elo icon does not appear to respond to double-click, do ALT-Tabs until the Elo Properties/Controls set is selected.

- Sound tab: The "Enable Click Sounds" touch zone is too wide (about 3 inches). At worst, this is slightly confusing if you don't know about it.

- Diagnostics tab: Most of the details are not implemented in this version. A red X will appear with a message if the driver fails to load.

- The Right Click sticky button default placement is in the upper lefthand corner of the screen. It can be dragged to a new location, but only while the Buttons tab is selected in the Elo Touchscreen Properites.

- The default placement of the Right Click Sticky button will cover the File menu selection of most applications if a high screen resolution is being used. The button can be moved (see above) or the Windows control buttons can be enlarged to "push" the menu down below the Sticky button (this also has the advantage of making Windows buttons much easier to use with a touchscreen). Select Control Panel, Display, Appearance. Click on the one of the control buttons in the "sample area" and adjust the size with the arrow key to the right of the "Captions Buttons" text; set it to approximately 40.

- Setup Wizard: The panel that says "The touchscreen was not calibrated" (displayed after calibration is aborted) has a < Back button that, when clicked, displays a panel that says "The touchscreen was calibrated successfully". In this case, the latter message is incorrect.

- The version is identified as 2.0 in the About tab; this version number applies to the Elo control panel, not the driver itself.

To remove the cursor when running Windows 95, use the null.cur file.

- Open Control Panel
- Select the Mouse icon
- Select the Cursors tab.
- Highlight the “Normal Select” item
- Select Browse
- Enter NULL.CUR and select OK.
- Select OK again.

The arrow cursor should now have disappeared.

3c - Windows NT Driver Notes

NOTE: For driver specifics, see section 2.

This section is for additional NT driver notes only.

NT Installation Notes

The latest NT driver, version 2.0, is the easiest to install and maintain. To download from the Web: www.elotouch.com (select File Download Library and download MMNT.exe). Copy MMNT.exe to a floppy and execute the file on the floppy to self-extract into the installation components.

NOTE: YOU MUST BE LOGGED ON AS THE SYSTEM ADMINISTRATOR TO INSTALL THE TOUCHSCREEN DRIVER SOFTWARE

(otherwise, the install program will not be able to modify the system registry)

To install: From NT: Start, Run, type a:setup. At approximately the 90% point, the touchscreen controller setup panel is displayed.

For most **serial controllers** (2210, 2300, 2310), **choose serial, SmartSet 2xx0** and the COM port the controller is connected to. For the older 4002 serial controller (may also be labeled 2856 or 2852), choose 4002 controller type.

For bus controllers, choose bus, the appropriate controller model and the address and IRQ you will be using.

2201 AccuTouch: Address default is 280; to change, the address must be jumper-selected on the board and selected in the controller setup panel. To change the default IRQ setting, no jumper changes are necessary (select in controller setup panel). If IRQ5 is unacceptable, try IRQ7 or IRQ2 (in that order).

4035 IntelliTouch: Address default is 280; to change, the address must be jumper-selected on the board and selected in the controller setup panel. To change the default IRQ setting, the IRQ must be jumper-selected on the 4035 board, as well as specified in the controller setup panel. If IRQ5 is unacceptable, try IRQ7 or IRQ2 (in that order).

NOTE: When changing address or IRQ, change the values in the NT Control Panel Touchscreen Setup BEFORE changing any board jumpers; if the software configuration does not match the hardware configuration, NT will hang at the "Blue Screen of Death".

After the selections have been made, click OK to finish the installation.

Reboot the system

Calibrate the touchscreen: In NT Control Panel, double-click the touchscreen icon. Select Calibrate and follow the directions.

NT error messages with Elo serial controllers and MMNT V2.0

Three error messages appear in the NT event log when Elo serial controllers are used with Windows NT. The reason is that the Elo serial driver accesses the COM port directly, using the COM port address and IRQ, while NT wants all devices to use the NT virtual communications interface. **The errors are benign and may therefore be ignored.**

The error messages are:

"The hardware resources for com(x) are already in use by another device."

"A conflict has been detected between two drivers which claimed two overlapping IO port regions. Driver Monmouse, with device <\Device\PointerPort0.Translated> claimed an IO port range with starting address in data address 0x28 and 0x2C, and length in data address 0x30."

"A conflict has been detected between two drivers which claimed two overlapping IO port regions. Driver Serial, with device <\Device\Serial10000.Translated> claimed an IO port range with starting address in data address 0x28 and 0x2C, and length in data address 0x30."

To remove the cursor when running NT, use the null.cur file as your cursor.

- Open Control Panel
- Select the Cursors icon.
- Highlight the Arrow field under System Cursors
- Select Browse
- Enter NULL.CUR and select OK.
- Select OK again.

The arrow cursor should now have disappeared.

Windows NT MonitorMouse Registry Data

Parameters for MonitorMouse in general:

Note: With the exception of the Start parameter, these parameters should never change. They are used by NT internally.

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Monmouse]

```
ErrorControl    REG_DWORD  0x1           // Used by the event viewer
Group           REG_SZ     Pointer Port  // The class of device(pointer, disk, etc.)
Start           REG_DWORD  0x1           // 0x1 designates that we load at system time, right after
                                         // the boot devices load. This can be changed by the
                                         // devices applet in the Control Panel.
Tag             REG_DWORD  0x2           //
Type            REG_DWORD  0x1           //
```

Parameters for the touchscreen specifically: Settings shown for 2310, COM1

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Monmouse\Parameters]

```
BeepOnOff               REG_DWORD  0x0    // 0x0 = Beep Off, 0x1 = Beep On
ComPort                 REG_DWORD  0x1    // 0x0 If bus controller, COM port number
                                          // otherwise.
ControllerAddress       REG_DWORD  0x3F8  // Base port address for serial and bus
                                          // controllers.
ControllerLength        REG_DWORD  0x8    // Bytes used on the bus including the base port.
ControllerLevel         REG_DWORD  0x4    // IRQ used by serial and bus controllers. This value has
                                          the potential for being different from the ControllerVector.
                                          Run MSD to verify that these are the same. For standard
                                          ISA and PCI machines, they should be the same.
ControllerModel         REG_DWORD   2310  // Controller model number. 2300, 2200, 2201,
                                          // 2210, 4035, 4002 legal.
ControllerType          REG_DWORD  0x0    // 0x0 = Serial, 0x01 = PC-Bus
ControllerVector        REG_DWORD  0x4    // Same value as "ControllerLevel".
MouseDataQueueSize      REG_DWORD  0x64.  // Always 0x64. This is the amount of mouse events that can
                                             be queued to the mouse class driver.
MouseMode               REG_DWORD  0x6    // 0x1, 0x2, 0x5, 0x6 are valid button emulation modes.
NumberOfButtons         REG_DWORD  0x2    // Number of mouse buttons.
OverrideHardwarePresent REG_DWORD  0x0    // Always 0x0. Used internally by the mouse class driver.
PointerDeviceBaseName   REG_SZ     PointerPort // The generic NT internal name for kernal
                                          // and for kernel and diagnostic recognition purposes.
SampleRate              REG_DWORD  0x28   // Used by the mouse class driver. This is meaningless to us.
                                             Recommended by Microsoft.
SwapButton              REG_DWORD  0x0    // 0x0 = monmouse emulates left button, 0x01 =
                                            right button.
SwapFlag                REG_DWORD  0x0    // 00 = Do not swap X/Y axes, 01 = Swap axes
                                            (touchscreen oriented vertically).
XLow                    REG_DWORD  0x1    // Upper left X calibration point.
XHigh                   REG_DWORD  0x107E // Lower right X calibration point.
YLow                    REG_DWORD  0x105F // Upper left Y calibration point.
YHigh                   REG_DWORD  0x24   // Lower right Y calibration point.
ZLow                    REG_DWORD  0x1    // Low Z calibration point (always 0x1).
ZHigh                   REG_DWORD  0xFF   // High Z calibration point (always 0xFF except
                                          // for 4002 = 0x0F).
```

Windows NT MonitorMouse Registry Data (cont)

Other parameters changed by the MonitorMouse Control Panel:
[HKEY_CURRENT_USER\Control Panel\Mouse]

DoubleClickHeight	REG_SZ	"4"	// "4" = 4 pixels, Mouse Mode 6 sets this to a larger number like "25".
DoubleClickWidth	REG_SZ	"4"	

Setting up an Elo touchscreen in a multiple monitor matrix under NT

Multiple monitors may be used with the Elo NT touchscreen drivers, **provided that only one is to be used as a touch monitor.** The current NT driver does not support more than one touchscreen in a matrix of monitors. A future NT driver will support multiple touchscreens.

To accomplish a single touchscreen in a monitor matrix with the current driver, first **configure the touch monitor as a single monitor, then perform the normal Elo touchscreen calibration**. After calibrating, reconfigure to the multiple monitor configuration.

Next, four additional entries must be manually added to the MonitorMouse Parameters area in the registry, using regedit ...

HKEY_LOCAL_MACHINE
SYSTEM
CurrentControlSet
Services
Monmouse
Parameters

Add the following to monmouse parameters (see Details, below):

XlateXLow
XlateXHigh
XlateYLow
XlateYHigh

Details: From the Edit menu, choose New, then DWORD VALUE. A new DWORD value box is created. Drag across the text and enter the new text **exactly** as above (observe upper-case letters). Repeat for the other new parameters.

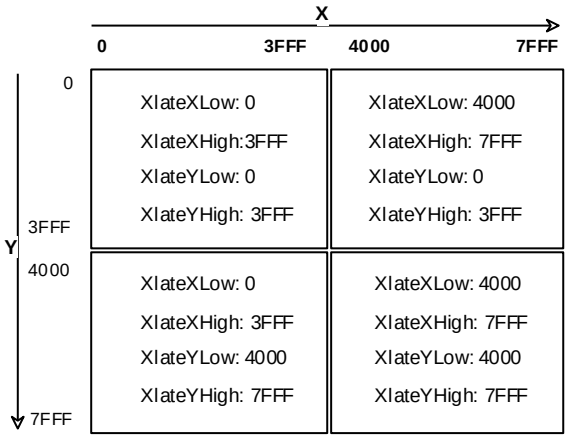
The XlateXLow and XlateXHigh values determine the starting and ending points, respectively, for the X-axis of the touch functionality. The XlateYLow and XlateYHigh values determine the starting and ending points, respectively, for the Y-axis of the touch functionality. **The origin for the touchscreen is the upper left corner**. Based on a range of 0 to 8000 **hex**, tailor the new entries for the location of the touchscreen monitor. See the diagram on the next page for values corresponding to various touchscreen locations. To edit an Xlate parameter, double-click on the value, then enter the new value from the keyboard.

Reboot the system and check operation. The numbers may have to be tailored slightly for optimum response, although calibration should take care of any touch location problems.

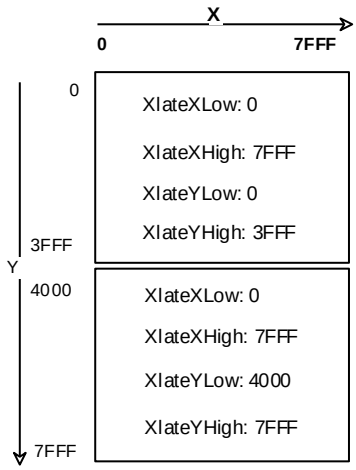
Setting up an Elo touchscreen in a multiple monitor matrix under NT (continued)

Xlate values for a single touchscreen in a monitor matrix, using Elo NT drivers

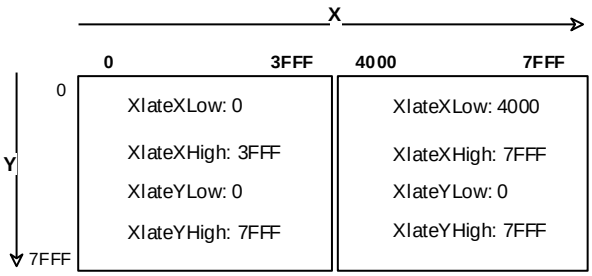
Quad orientation



Dual vertical orientation



Dual horizontal orientation



Section 4- IntelliTouch Technology Notes

Application Note:
Testing Touchscreens With SAWSTAT

December 26, 2019

NOTE

This is not a specification. The values given below are only approximates useful for understanding IntelliTouch theory of operation.

This information assumes you have a working knowledge of SAWSTAT. See SAWSTAT.DOC for instructions.

When testing IntelliTouch touchscreens, integrations, and sealing, there are several symptoms of an improperly working system: reduced sensitivity, dead regions, sudden calibration change at power on, and intermittent coordinate transmission. SAWSTAT can be used to diagnose these and other problems, typically due to low signal strength or electrical noise.

NO SIGNAL

SAWSTAT displays the controller power-up diagnostics. If there is a problem with the touchscreen cables or transducers that prevents the system from working at all, it will be indicated.

SIGNAL SHAPE

The signal shape for X and Y displayed by SAWSTAT should look somewhat like Figure 1, although the signals may be flat, have more oscillations, or even slope upward.

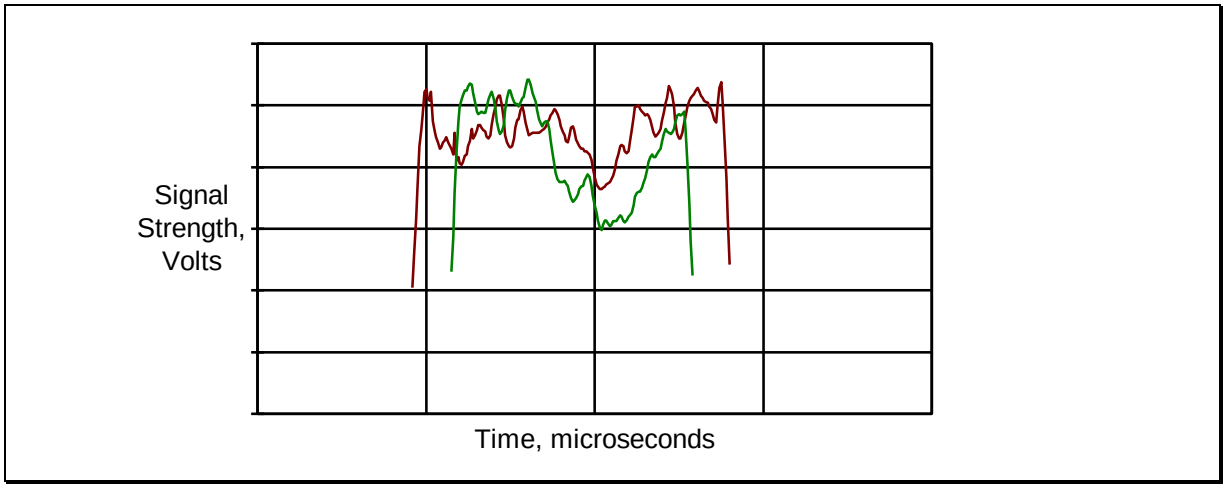


Figure 1. Typical 17-inch IntelliTouch Touchscreen Receive Waveforms

IntelliTouch controllers are designed to work with all variations of signal shape. However, the following may be indications of a faulty system. Inspect the reflectors.

- If a downward slope exists greater than in Figure 1, dirt or a seal on the reflectors may be causing a problem.
- Any signal below 1.5V is suspect.
- If the voltage of the peak signal is much more than 2X the minimum signal (3X for 19"+ touchscreens), there may be a problem. Exclude 0.4" at the beginning and end of the signals with SAWSTAT's Sample Window Limits settings.
- The length of the signal traces should be the same for like touchscreen sizes.
- The active portions of both the X and Y traces should have approximately the same centerline along the time axis.

SIGNAL STRENGTH

Next, look at the AGC (automatic gain control) levels for X and Y by pressing the [TAB] key within SAWSTAT until they are displayed. The AGC levels do not measure the touchscreen's signal strength, but they tell how much the controller must amplify the signal to get the peak in a normalized position. The range of the AGC levels is 0-255, and the reported number should not be interpreted as having any physical units or as being calibrated to any measurement standard.

The following AGC levels probably indicate a defective touchscreen, dirty reflectors, or a seal with too much signal attenuation. For the E281-2310 controller, larger numbers indicate higher gain levels at the controller, and would be associated with a weak signal returning from the touchscreen. For the E281-2300 controller, smaller numbers mean higher gain levels. Be sure to check both axes.

Size	2310 Bad AGC Levels	2300 Bad AGC Levels
14"	greater than 170	less than 120
15"	greater than 200	less than 100
19"+	greater than 220	less than 70

In general, smaller touchscreen sizes have more signal strength than larger sizes. Also, the higher the signal strength, the better. For example, an AGC level of 100 is better than 120 with a 2310 controller.

Save and compare samples in SAWSTAT for the unintegrated and integrated/sealed touchscreens with the S)ave and L)oad commands. When sealing, the signal strength should not drop out of the acceptable AGC ranges nor should the signal shape become unacceptable. A seal should not attenuate the signal by more than about 6 dB. Avoid having sealing material on the reflectors and only use materials recommended by Elo.

NOISE

IntelliTouch components, when properly installed, are designed to work in an electrically noisy environment, such as inside a plasma display or a large CRT monitor.

However, excessive noise can result from a poor controller power source, an improperly grounded controller and/or cable, as well as routing the touchscreen cable or mounting the controller near noisy display components. Low signal strength can also lead to increased noise sensitivity.

There are several features in SAWSTAT to assist in detecting noise problems:

- Use the Test Report Rate command in SAWSTAT to count the number of coordinates transmitted per second. You should get approximately the same number of coordinates per second for an integrated and unintegrated touchscreen . The default reporting rate for both E281-2310 and E281-2300 controllers is about 45 coordinates per second. A lower value indicates the controller is discarding marginal coordinates which will affect its overall reporting rate. If the controller is powered separately from the monitor, run this test with the monitor on and off to see if the monitor is generating the noise and substantially reducing the coordinate transmission rate.
- The 2310 and 2300 controllers have event counters that are useful for detecting noise. Press the [TAB] key until they are displayed by SAWSTAT. Press [SPACE] to reset the values. Touch the screen 3 times, then press [SPACE]. You should see three touch events counted. Press [SPACE] again to clear the values. Wait a few minutes without touching the screen and press [SPACE] again. All counters should still be zero unless noise was generating acceptable or unacceptable touches, relearns, or autosizing.
- The 2310 controller computes a signal-to-noise (S/N) ratio that is displayed on the counters screen within SAWSTAT. An ideal S/N Ratio is 45. This is the maximum reported by the controller. The controller may have reduced performance if the value is below 30. A value below 20 indicates a serious problem.

IN CONCLUSION

Again, nothing in this application note is meant to serve as a specification. Values given are approximates only. Touchscreens may work with very poor signal shapes, AGC values, and excessive noise. This information is only for assisting in diagnosing problem touchscreens or integrations, and to help determine if a touchscreen is getting close to the point of failure.

IntelliTouch Noise

General:

IntelliTouch controllers transmit 30 volt spikes to the touchscreen, but receive back a signal of only millivolts. Typically, the signal cables are routed inside a monitor, an environment that has very high voltages and frequencies. Because of the possibility of radiated switching noise from these sources, special cabling, shielding and proper grounding are required.

The touch threshold is normally established just below the peak of the touch waveform, so any touch will cause a dip in the waveform such that the threshold is crossed and a touch is indicated. If noise causes the waveform to chop repeatedly below the threshold, the events will be discarded and the threshold will be automatically lowered to put it below the noise; in an environment of excessive noise, the threshold can be lowered to a point when the signal can no longer be detected.

Symptoms:

- Lockup

  Severe noise can cause the controller X and Y Automatic Gain Controls (AGC's) to lower the gain until the signal can no longer be seen.

- Spurious double-touching

  Noise can cause spurious untouch reports; double-tap response can result when a single tap was intended. In an application, an on-screen simulated switch can toggle through the intended setting (on-off, instead of on).

- False touches

  Noise can cause spurious touches; an on-screen control activates without having been touched.

- Delayed/slow response

  Noise can cause touch response to be delayed. Noise can also cause erratic operation, such as a failure to respond to double-tap.

- Insensitive to touch; poor drag performance; requires high touch pressure

  Noise causes the touch threshold to be lowered. In extreme cases, extreme touch pressure is required before the threshold is reached..

IntelliTouch Noise (continued)

Causes:

- Routing the touchscreen cable (internal to monitor) too close to the CRT yoke, the flyback transformer, switching power supplies or other noise generating switching components.

- Using too long a cable run or an improper cable (for external controllers, the thicker cable with the twisted pairs and extra shielding must be used; this is an Elo-proprietary cable). This only applies to non-internal-controller monitors; output from serial controllers is standard RS232 and it not highly susceptible to noise. Examine cable to external serial or bus controllers (Elo P.N. 002863-x-2, says "To IntelliTouch Controller/Screen" on ends).

- Bus controller close to a noisy board (video card, for instance) or installed in a computer with a noisy bus. (Try another slot/another computer).

- Controller not well grounded (possible, although standard signal cabling should take care of grounding problems).

- Noisy source of power used for internal controller.

- **NOTE: EXTERNAL sources of noise may be a problem, particularly when an improper cable has been used. Try to determine if sources of excessive RF or line noise exist in the environment.**

IntelliTouch Noise (continued)

Troubleshooting

When troubleshooting, always remember to allow the controller to settle for approximately one minute after power-on; this allows thresholds to be set and waveforms to be learned.

The TOUCHES utility should report FALSE as long as the screen is being touched, and TRUE only on untouch. If spurious TRUEs are reported, noise is indicated. If a TRUE occurs while touching, followed by lockup, noise is indicated (AGC swamping the signal). If erratic TRUEs are seen, you can try disconnecting the video cable and running TOUCHES with a blank display (touch, hold, then reconnect the video cable and inspect the results). If no false TRUEs were reported with a blank screen, video switching noise problems are indicated (internal serial controllers). TOUCHES only works with ELODEV.

The SAWSTAT utility reports, among other things, the X and Y AGC levels. For pre-2310 controllers, the AGC values are initialized to 127, and they adjust from that initial value, in the range of 0 to 255. The values decrease (to decrease the gain) as the noise increases (for pre-2310 controllers). Typical values vary for different size screens; 15" monitors may run from 140 to 160; 20" screens might run in the range of 80 to 100. If one or both is at the rail (0 or 255), or if one or the other remains at 127 consistently, a problem exists. NOTE: 2310 controllers respond in the opposite way; that is, as the noise increases, the AGC values increase, so suspect 2310 noise when the values get atypically high.

You can save the waveforms generated in SAWDUMP and analyze them with SAWSTAT (SAWSTAT is not available to customers, but when in doubt you can have customers e-mail the waveforms to you, or ship you a diskette containing the waveforms. **When running SAWDUMP or SAWSTAT with a controller, always allow one minute or so after power up to allow thresholds to be established.**

Counters 9 and 10 in the SAWSTAT display show the signal-to-noise ratio for 2310 controllers; 45 is a good S/N ratio, 30 is poor.

SAWSTAT displays actual X and Y waveforms, which can be quite helpful in analyzing IntelliTouch problems. Some problems are obvious (flat-line response, spikes, jagged holes in the waveform). Consult IntelliTouch engineers for help with interpreting SAWSTAT waveforms. Also, see the Noise section in the SAWSTAT Application Note.

IntelliTouch Noise (continued)

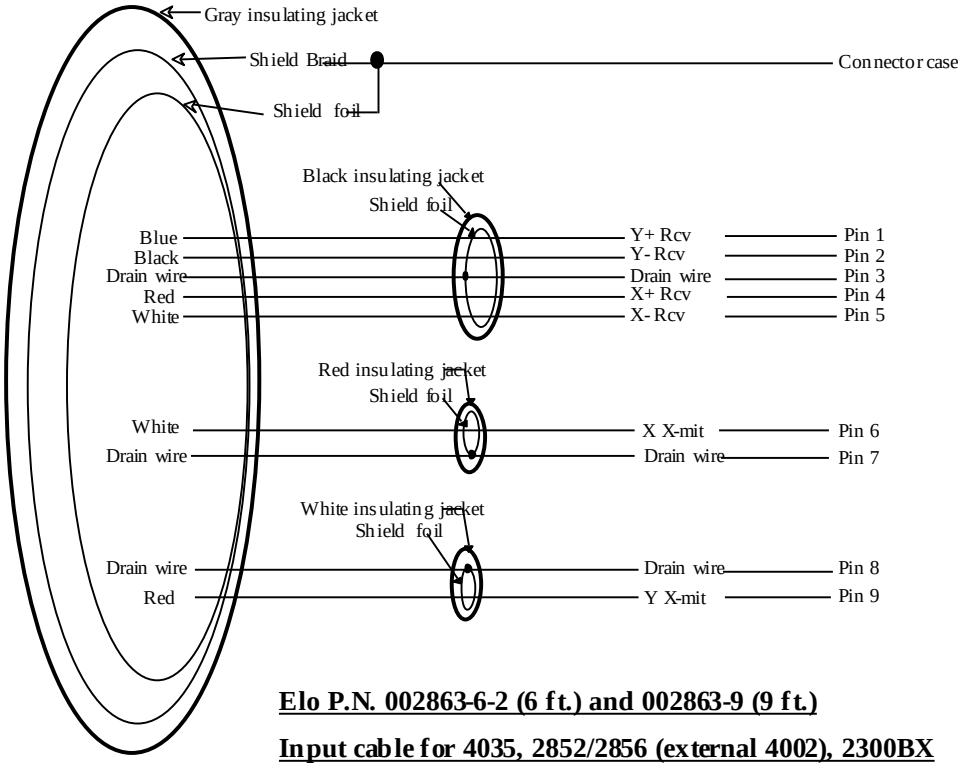
TSTIMER is another diagnostic, although it is not distributed on our diskettes. TSTIMER reports the number of touch events recorded in 10 seconds, so the touch processing rate can be analyzed (low rates indicate noise problems). This test may be run with a black screen or a white screen; the white screen generates much more video board activity, which can translate to more system noise. The monitor power can also be turned off while the test is running (if the controller is not internal) to silence flyback and switching supply noise; with internal controllers, the video cable can be disconnected to minimize video activity. 45 coordinates per second is good; 30 is poor. SAWSTAT has a similar test (See the SAWATAT Application Note).

A rough test can be performed in a lockup situation by simply turning off the monitor (again, if the controller is not internal) or unplugging the video cable, to reduce the switching (hence the noise) in the monitor. First, after the touch has ceased to respond (locked up), note the cursor position. Then disconnect the video/unplug the monitor power as appropriate to the controller configuration, wait half a minute or so for the AGCs to readjust, then touch in a known new position. Note if a touch beep is heard. Reconnect video/power and see if the new "quiet environment" touch was recorded. If so, then noise is indicated.

Corrective Action

If the noise can't be eliminated by moving a bus controller, the problem is usually due to improper routing of the touchscreen cable inside the monitor. Ask if the monitor has an Elo TouchSystems label on the front; if not, then a third party did the integration work, and improper cable routing is a prime suspect. There have been instances of Elo-integrated monitors having noise problems, but it is a rare occurrence. In general, monitors with noise should be returned to the integrator for repair.

- **A FINAL NOTE: ALWAYS give the IntelliTouch controller time to stabilize after power-up (30 to 60 seconds). If a customer complains of numerous failures of IntelliTouch products, make sure they are allowing the system to stabilize after power-up. In the course of your troubleshooting, remember to allow the system time to stabilize.**



Elo P.N. 002863-6-2 (6 ft.) and 002863-9 (9 ft.)

Input cable for 4035, 2852/2856 (external 4002), 2300BX

Use this cable, *not a standard serial cable*, for input to all external IntelliTouch serial and bus controllers (except 2310MX).

This is a straight through DB9M to DB9F cable; the connections are the same as for a standard modem serial cable, but this cable is specially constructed for immunity to noise.

It can be identified by its diameter, which is approximately 3/8" (9mm); standard serial cables are approximately 1/4"in diameter. This cable is also marked with the Elo part number and at each end with the words "Connect to IntelliTouch Controller" or "Connect to IntelliTouch Touchscreen".

The receive pairs are all bundled in a black plastic jacket, along with a common drain wire which is in contact with the shield foil. The purpose of the shielded bundle is to isolate the small-signal receive lines from external inductive noise sources.

Each of the transmit lines is encased in its own shield foil, which is in contact with its own drain wire; each drain wire is the ground for each transmit line. The purpose of the two independently shielded transmit lines is to isolate the large signal (~30 volts) transmit lines from each other and from the small-signal receive lines. The three separate bundles are encased in a shield foil and a shield braid. The braid is tied to the case of the connectors, which acts to shield the entire cable from outside inductive nose sources.

There are also five cloth cords in the main bundle to aid flexibility and strain relief.

Power-On Self-Test error codes for **IntelliTouch** **controllers**

Error Code

00-00

01 X-axis failure
02 Y-axis failure
03 X receive channel failure
04 Y receive channel failure
05 X transmit channel failure
06 Y transmit channel failure
07 General failure (T/S may not be connected)

08 DAC hardware test failure
10 PROM checksum test failure
20 External RAM read/write test failure failure
40 Microprocessor register test failure
80 Microprocessor logic test failure

These may be cumulative, i.e., 60 is both error 20 and error 40

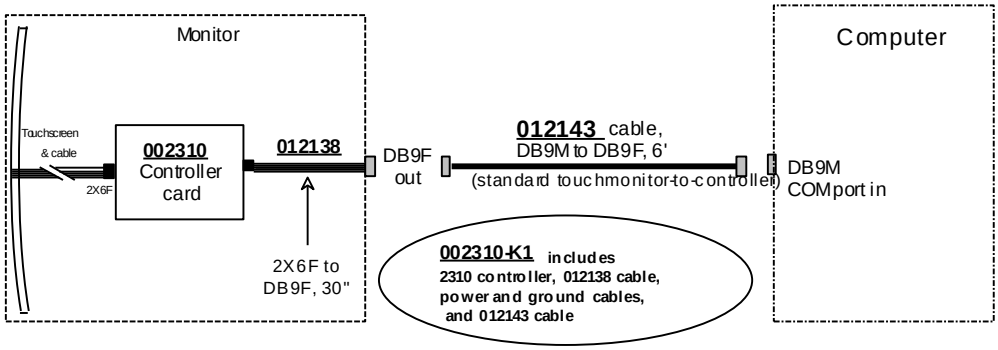
These may be cumulative

01 Firmware/hardware compatibility test failure
02 Microprocessor/logic test failure
04 ROM checksum test failure
08 RAM read/write test failure
10 Non-volitile RAM checksum test failure
20 Touch microprocessor test failure (T/S may not be connected)
40 Touch microprocessor test incomplete

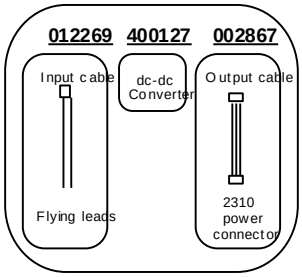
Section 5 - IntelliTouch Controller Notes

IntelliTouch internal serial controller part numbers

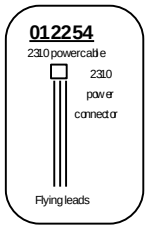
IntelliTouch Monitor (internal serial controller)



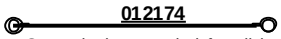
Power options for IntelliTouch internal 2310 controller



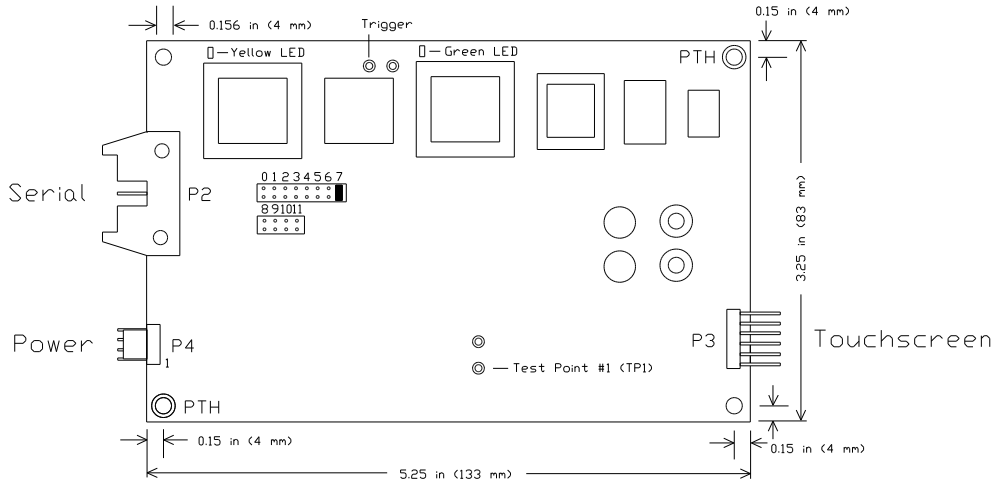
Power tap from 8 to 30 volt dc source on monitor board; regulated 5 volts out



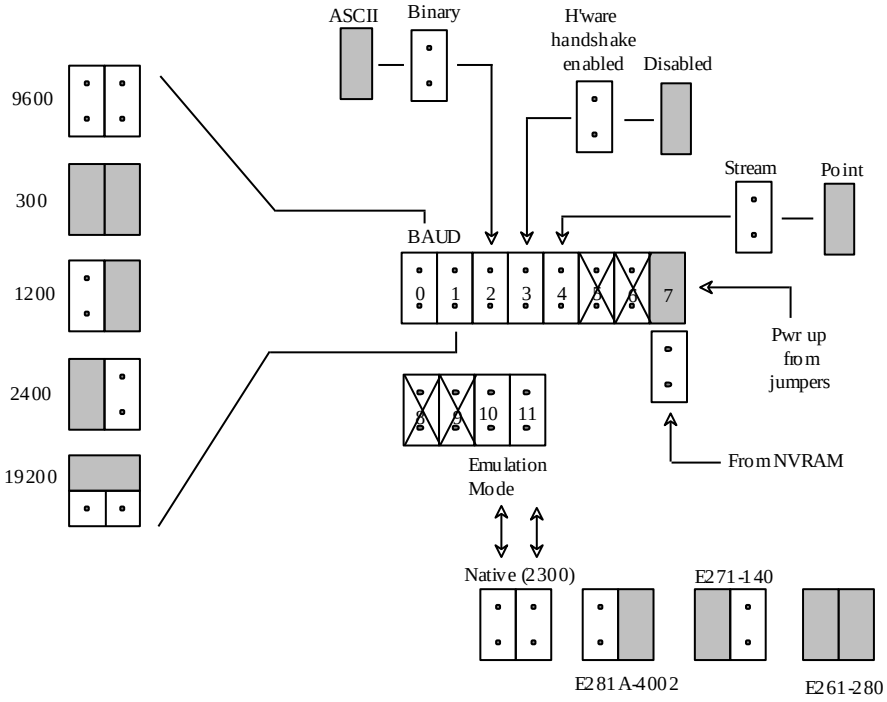
Power tap direct from +5 volt source on monitor board



Ground wire; needed for all installations



2300 IntelliTouch Controller



Diagnostic LEDs

Yellow (Communications) **Normal: Flashes 1.5/second**; lights (or flickers) with touches

Problem: Stays on solid

Host failed to remove data packet

Possible bad power supply

Problem: Flashes 4/second

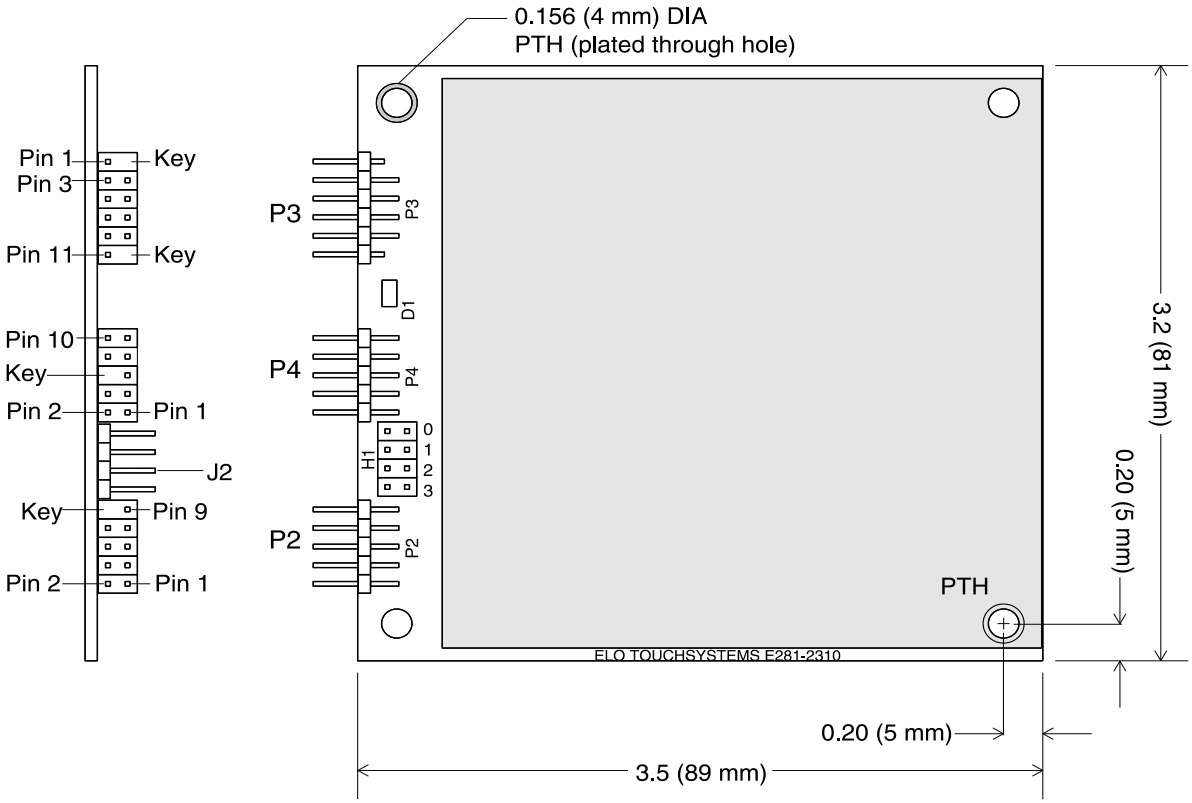
Communications problem; suspect BAUD, etc.

Green (Self-test) **Normal: Off (On during Power-On Self-Test)**

Flashes (2/sec) while screen is touched

Problem: Stays on solid; Power-On Self-Test failed

E281-2310



Factory Jumper Settings		
J0	Out	Reserved
J1	Out	Reserved
J2	Out	No Emulation Mode
J3	Out	Reserved

Power Requirments
+5 VDC, nominal (± 5%)
140 mA, typical at +5 VDC

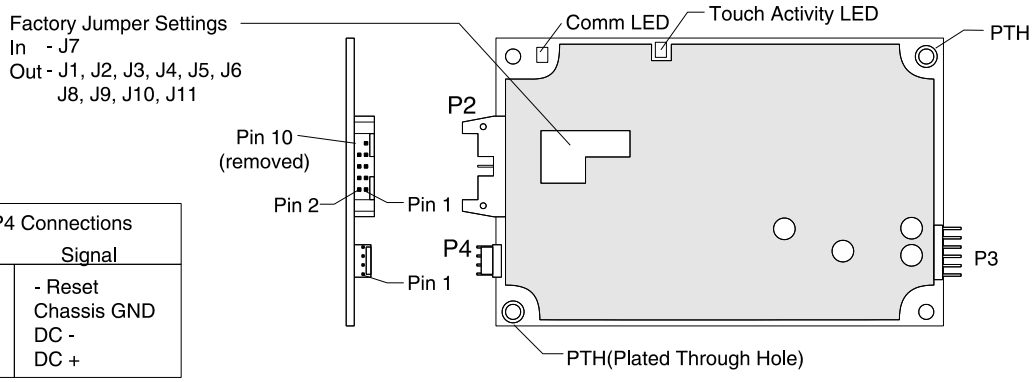
P2 Connections		Serial Communication Connections	
		DB 9	DB 25
1	Data Carrier Detect (DCD)	1	8
2	Data Set Ready (DSR)	6	6
3	Receive Data (RxD)	2	3
4	Ready To Send (RTS)	7	4
5	Transmit Data (TxD)	3	2
6	Clear To Send (CTS)	8	5
7	Data Terminal Ready (DTR)	4	20
8	N/C	9	22
9	Ground (GND)	5	7
10	Key	N/A	N/A

P4 Connections	
Pin	Signal
1	DC +
2	DC - (tied to pin 4)
3	N/C
4	DC - (tied to pin 2)
5	LED Remote
6	Key
7	N/C
8	Chassis Ground
9	- Reset
10	N/C

Jumper J2: Installed = 4002 controller emulation

Diagnostic LED: Green **Normal, without touch: Should blink at 1 Hz.**
Normal, with touch: On solid while touched.
Problem: On solid indicates Power-on Self Test failure; host failed to remove data; touchscreen cable disconnected; or touchscreen not functioning
Problem: Fast blink (2 or 3/sec) indicates Communications error

E281-2300



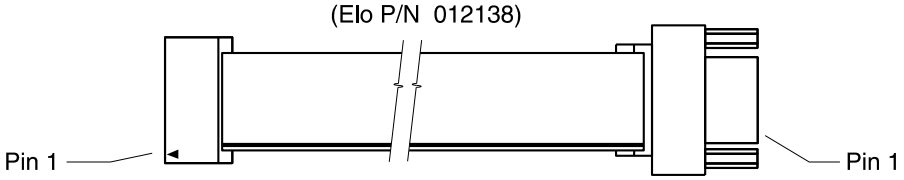
P4 Connections	
Pin	Signal
1	- Reset
2	Chassis GND
3	DC -
4	DC +

Connector	To
P2	RS-232 Serial Communication
P3	Touchscreen Cable
P4	Power Supply

Power Requirments
+12 VDC, nominal (+4.75 to +30 VDC)
90 mA, typical at +12 VDC

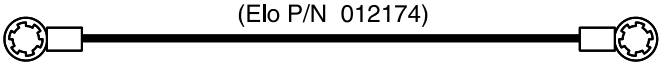
P2 Connections		Serial Communication Connections	
		DB 9	DB 25
1	Data Carrier Detect (DCD)	1	8
2	Data Set Ready (DSR)	6	6
3	Receive Data (RxD)	2	3
4	Ready To Send (RTS)	7	4
5	Transmit Data (TxD)	3	2
6	Clear To Send (CTS)	8	5
7	Data Terminal Ready (DTR)	4	20
8	N/C	9	22
9	Ground (GND)	5	7
10	Key	N/A	N/A

2x5 to DB9-Female Flat Ribbon Cable Assembly, 30 in.
Serial cable, from P2 to back of monitor
(Elo P/N 012138)



Cable, 1x4 with flying leads, 18" - from P4 to power source
(Elo P/N 012217)

Cable, Ground with Ring Terminals - from PTH to Frame Ground
(Elo P/N 012174)

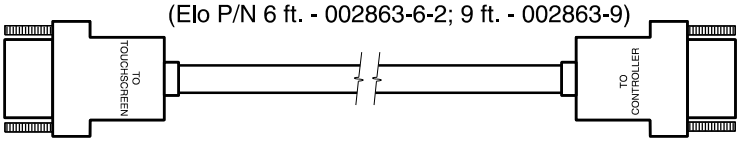


Power Supply, Internal Mount, 12 VDC, International Power (not pictured)
(Elo P/N 004100-206-K1)

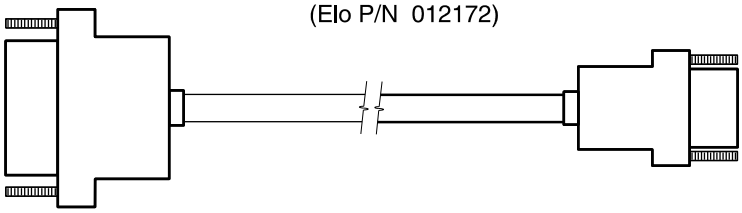
Internal Mounting Kit - Misc. hardware to install E281-2300 inside a monitor (not pictured)
(Elo P/N UK2300)

E281-2300 Accessory Cables

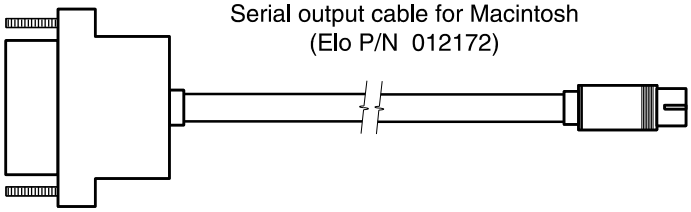
DB9-Female to DB9-Male EC2 Analog TouchScreen Cable
From TouchMonitor to Controller
(Elo P/N 6 ft. - 002863-6-2; 9 ft. - 002863-9)



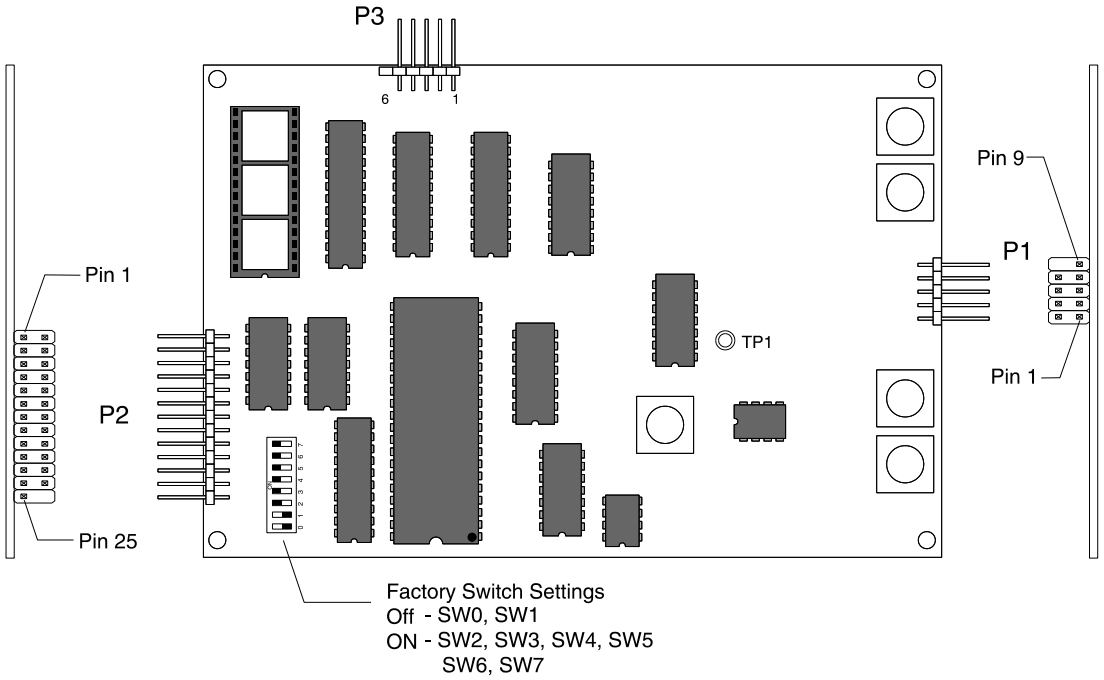
DB25-Male to DB9-Female Serial Cable, From external box to PC-Com Port
(Elo P/N 012172)



DB25-Male to 8 pin mini-din
Serial output cable for Macintosh
(Elo P/N 012172)



E281-4002



P2 Connections		Serial Communication Connections	
		DB 9	DB 25
3	Transmit Data (TxD)‡	3	2
5	Receive Data (RxD)	2	3
9	Clear To Send (CTS)	8	5
11	Data Set Ready (DSR)	6	6
13	Ground (GND)‡	5	7

‡ The controller only requires TxD and GND. Implementation of the other signals will allow two way communication between the E281-4002 and the host computer.

Connector	To
P1	Touchscreen Cable
P2	RS-232 Serial Communication
P3	Power Supply

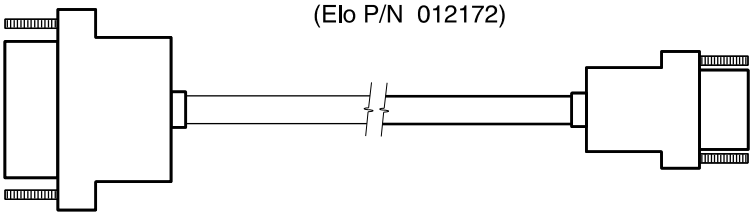
Dip Switch Settings				
Data Type	SW0	Baud Rate	SW3	SW4	SW5
Stream*	OFF	9600*	ON	ON	ON
Single Point	ON	4800	ON	ON	OFF
		2400	ON	OFF	ON
Communications	SW1	1200	ON	OFF	OFF
Binary*	OFF	600	OFF	ON	ON
ASCII	ON	300	OFF	ON	OFF
		150	OFF	OFF	ON
Z-Axis	SW2	9600	OFF	OFF	OFF
Enabled*	ON				
Disabled	OFF		SW6	SW7	
		Reserved*	ON	ON	
* Factory Default Settings					

P4 Connections	
Pin	Signal
1	-12 VDC, 0.5A
2	GND (Analog)†
3	+12 VDC, 0.12 A
4	GND (Digital)†
5	+5 VDC, 0.05A
6	Key

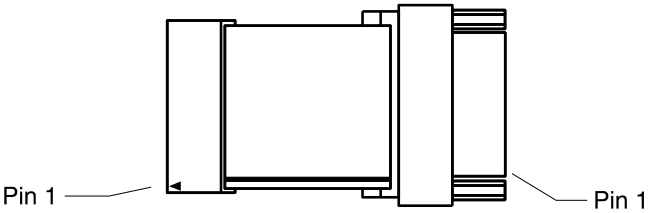
† For proper operation pins 2 and 4 must both be grounded.

E281-4002 Accessory Cables

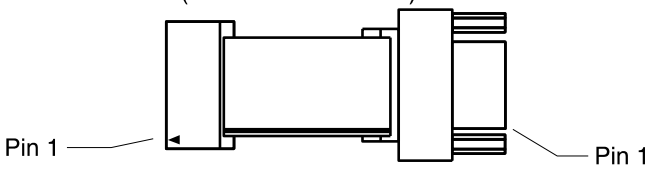
DB25-Male to DB9-Female Serial Cable, From external box to PC-Com Port
(Elo P/N 012172)



2x13 to DB25-Female Flat Ribbon Cable Assembly, 2 in.
From P2 to Serial Cable (used inside external box)
(Elo P/N 02861-2-25)

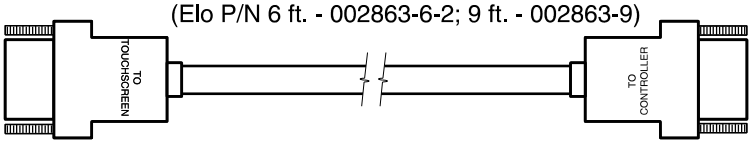


2x6 to DB9-Female Flat Ribbon Cable Assembly, 2 in.
From P1 to EC2 Touchscreen Cable (used inside external box)
(Elo P/N 02861-2-12)



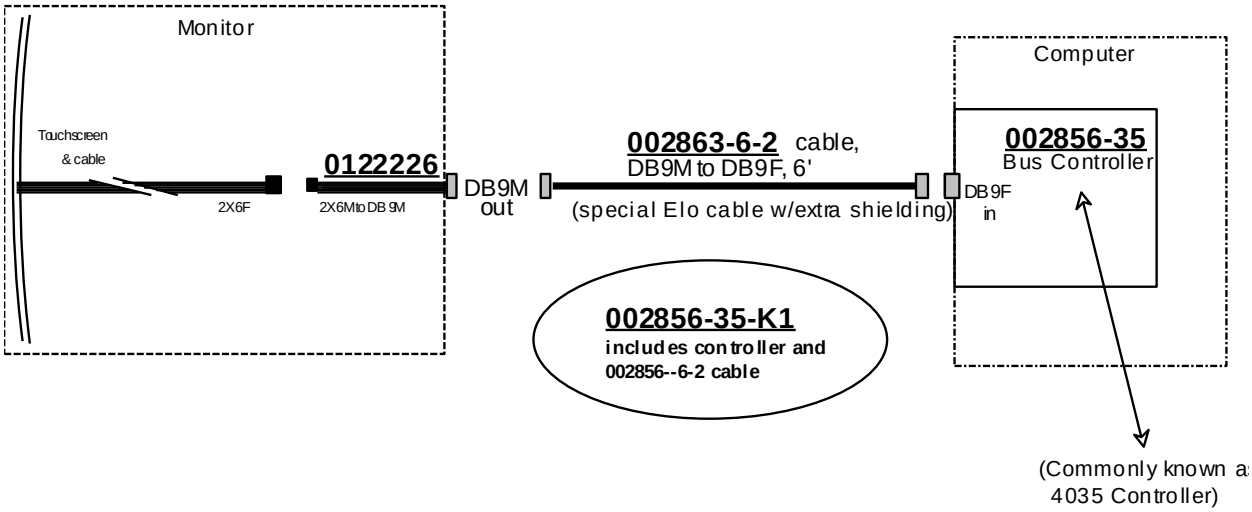
Cable, 1x6 with flying leads, 6 in.
From P3 to power source (used inside external box)
(Elo P/N 002899-18-1)

DB9-Female to DB9-Male EC2 Analog TouchScreen Cable
From TouchMonitor to Controller
(Elo P/N 6 ft. - 002863-6-2; 9 ft. - 002863-9)

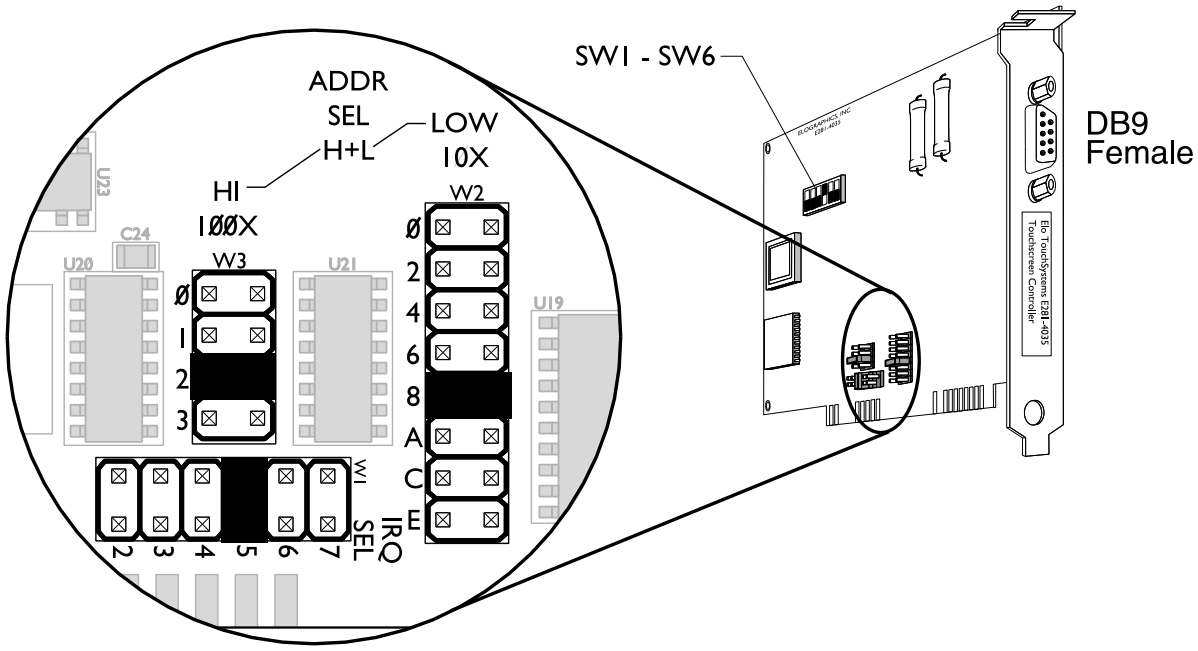


IntelliTouch bus controller part numbers

IntelliTouch Monitor



E281-4035



Jumper Installed

Jumper NOT Installed

- The E281-4035 controller only uses four consecutive I/O ports, but it occupies 32 (0x20) consecutive I/O ports. The Base I/O address is selected by jumpers W3 and W2. In the example above the Base I/O address is 280H (factory default).

  (W3 value X 100) + (W2 value X 10)

  (2 X 100) + (8 X 10) = 280H

- Jumper W1 selects the IRQ. In the example above IRQ 5 is selected.

Switch Settings		
Switch	**Closed (on)**	**Open (off)**
SW1	12 Bit Mode*	8 Bit Mode
SW2	Reserved*	N/A
SW3	Point Mode	Stream Mode*
SW4	Reserved*	N/A
SW5	Z Axis Enabled*	Z Axis Disabled†
SW6	Reserved*	N/A

* Factory Default

† E271-141 Emulation Mode

Power Requirements

5 VDC (±5%) @ 0.3 Amps

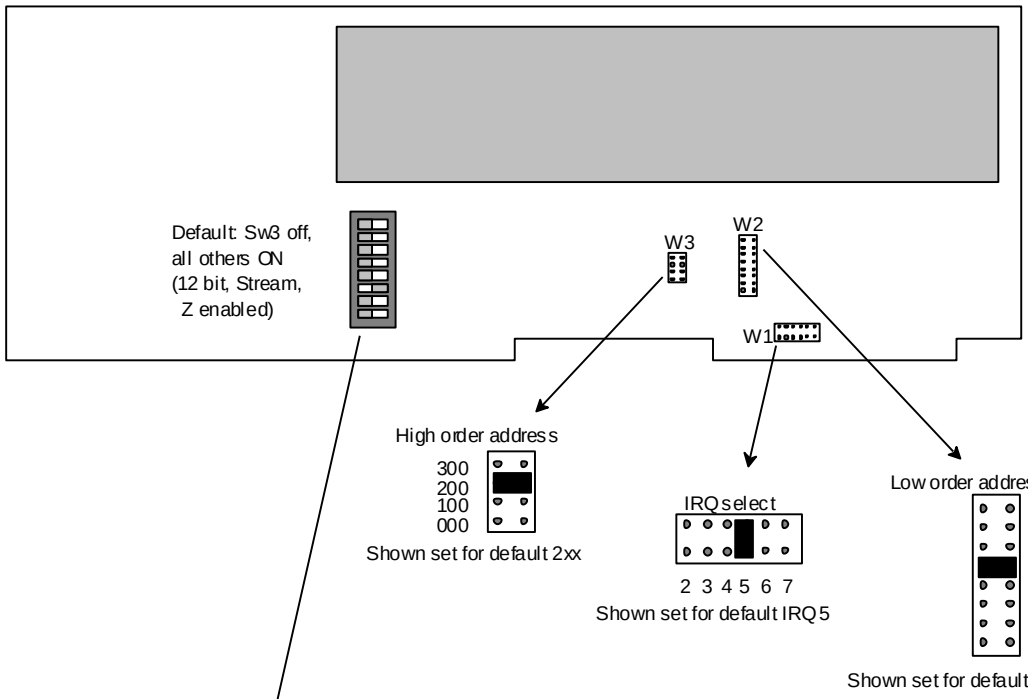
12 VDC (-5%, +10%) @ 0.1 Amps

-12 VDC (±10%) @ 0.04 Amps

Ripple for all voltages < 0.150 V p-p

281-4025 IntelliTouch Bus Controller

Default settings: Address 280, IRQ 5

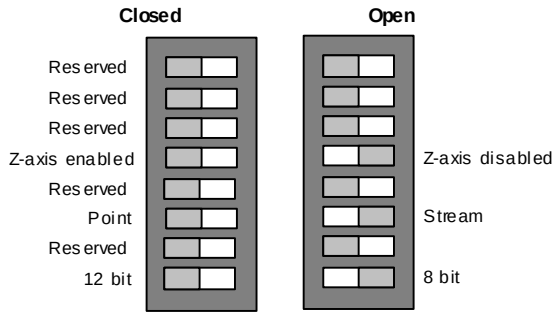


If IRQ 5 causes a conflict, try IRQ 7

IRQ 2 may or may not work

IRQs 4 and 3 are generally reserved for com ports 1 and 2, respectively

IRQ 6 is reserved for the floppy disk controller



Leave Reserved switches in the closed position

Section 6 - AccuTouch Technology Notes

AccuTouch Dead corner

A dead corner is sometimes seen on AccuTouch products. A dead corner is due to failure of one of the solder joints where the 5-pin touchscreen cable connects to the touchscreen.

A good analogy for a dead corner on a touchscreen is a rubber band stretched around four pegs in the corners of a rectangle: remove one of the pegs and the rubber band pulls in to eliminate the corner. The same sort of response occurs in the case of a dead corner on a touchscreen, except that the dead zone will be more like that in Figure 3, below.

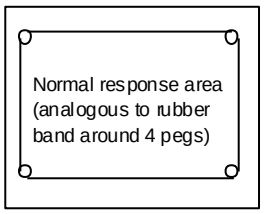


Figure 1

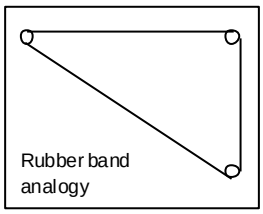


Figure 2

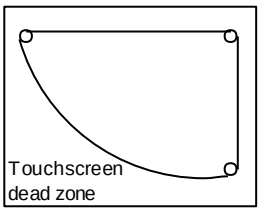
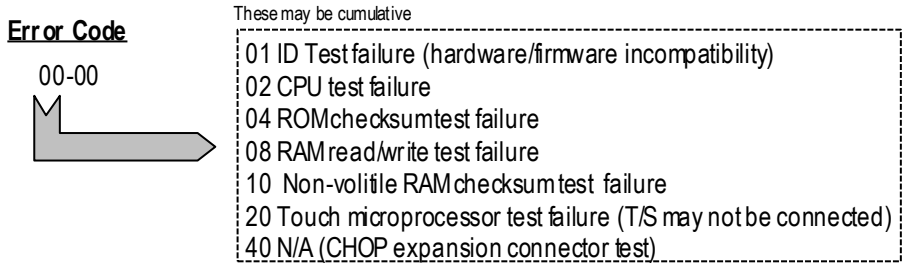


Figure 3

If someone complains about a non-linearity issue with an AccuTouch product, have them touch each corner of the screen. If the cursor fails to go to the point of touch in one corner, then the touchscreen probably has a failed solder joint.

Power-On Self-Test error codes for AccuTouch controllers

Power-on Self-Test error codes for AccuTouch Controllers



Chemical Resistance of AccuTouch® HL™ and Enhanced Antiglare FST Touchscreens

The following is chemical resistance data provided by our coversheet plastic suppliers. For economic and chemical disposal regulation reasons, Elo only retests the AccuTouch product using a subset of the harsher chemicals listed. However, customers can feel confident about the data provided below by Elo suppliers. Ultimately, it is the customer's responsibility to determine the suitability of Elo products in their application.

The active area of the AccuTouch HL and Enhanced Antiglare FST touchscreens is resistant to the following:

Acetone	Ethanol	TCE
Benzene	Isopropyl Alcohol	Toluene
Butyl Acetate	MEK	Xylene
Chloroform	Methanol	

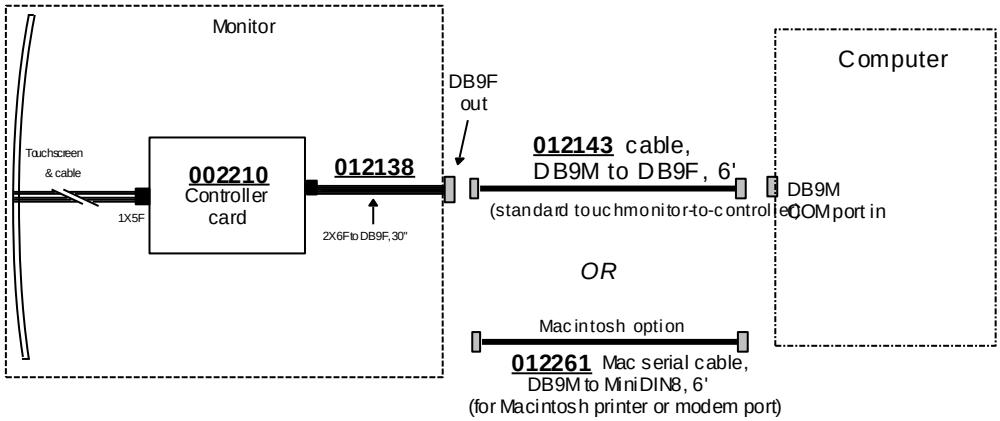
Mayonnaise	Tomato Catsup	Window Cleaner
Salad Oil	Vinegar	

Acetic Acid	Hydrochloric Acid	Sodium Hydroxide
Ammonia	Mineral Spirits	Turpentine

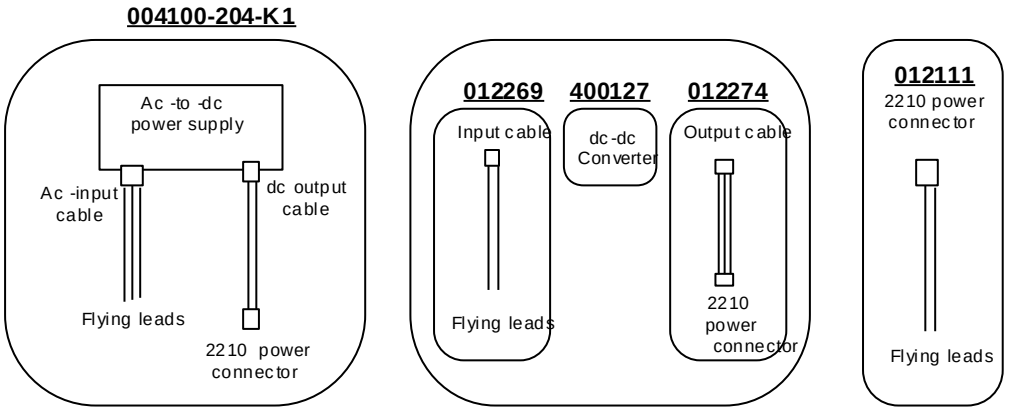
Section 7 - AccuTouch Controller Notes

AccuTouch internal serial controller part numbers

AccuTouch Monitor (internal serial controller)



Power options for AccuTouch internal serial controller



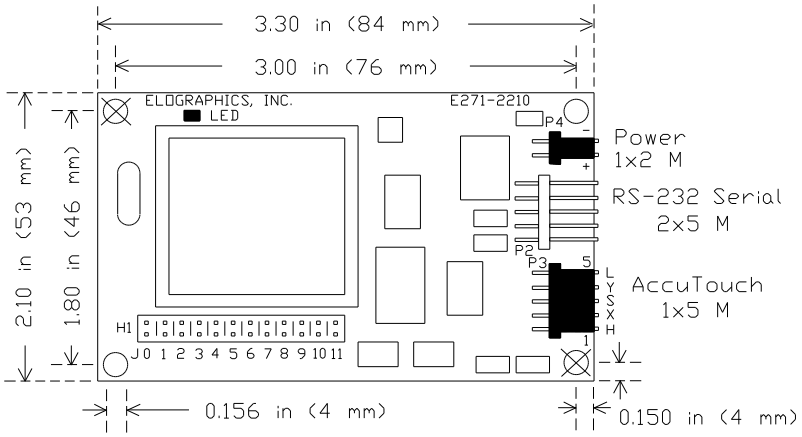
Power tap from monitor AC power. Regulated +5 volts out

Power tap from 8 to 30 volt dc source on monitor board; regulated +5 volts out

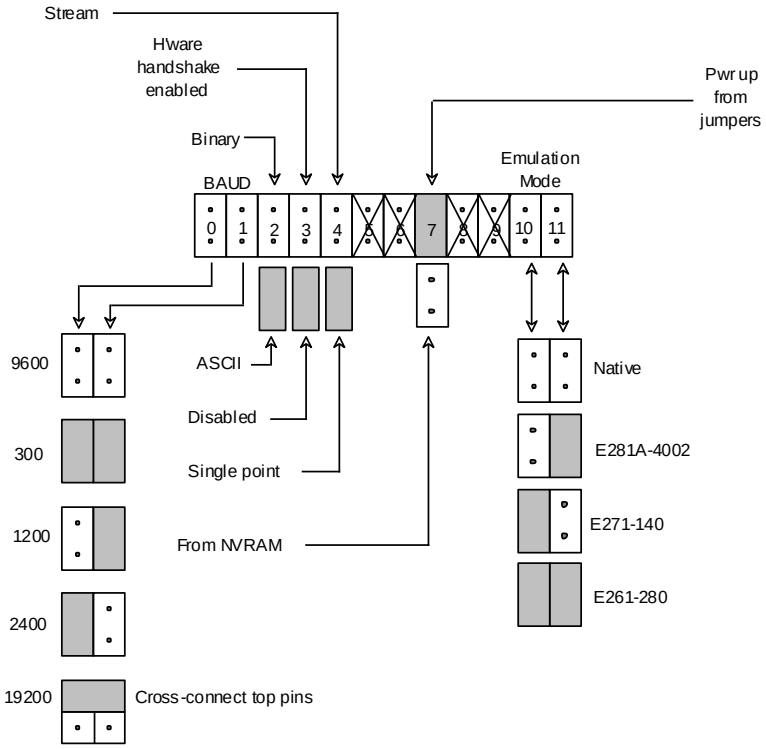
Power tap direct from +5 volt sourc on monitor board

012174

Ground wire; needed for all installations



E271-2210 AccuTouch Serial Controller



E271-2210 Diagnostic LED (Yellow only)

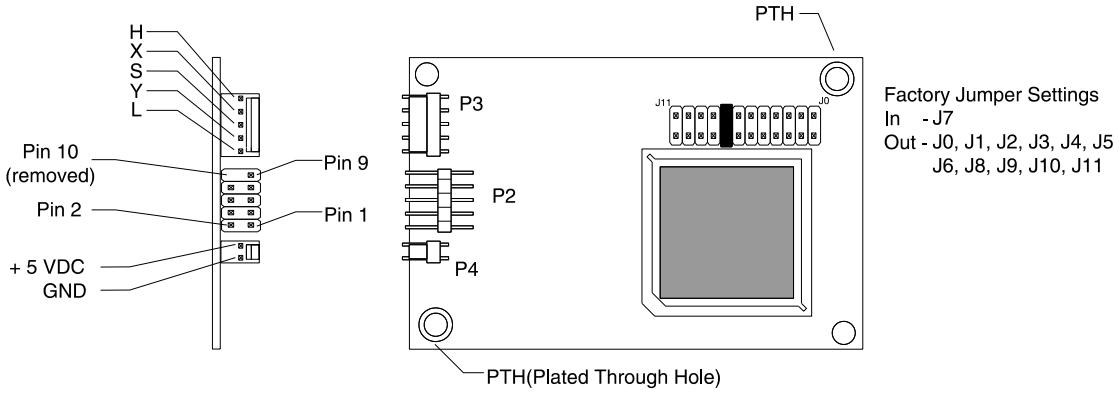
Normal: Flashes 1.5 times per second; also lights with touches

Problem: Stays on solid

Host failed to remove data packet
Shorted touchscreen or wiring
Disconnect touchscreen and cycle power to detect

Problem: Flashes four times per second

Communications problem - Suspect BAUD, etc.

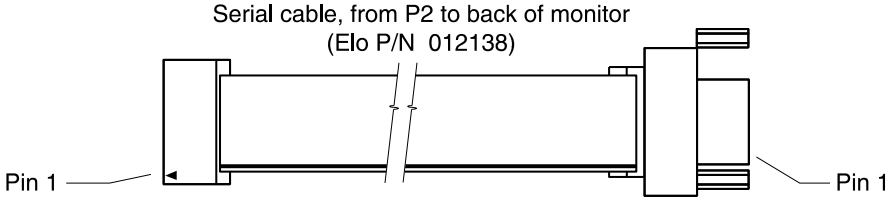


Connector	To
P2	RS-232 Serial Communication
P3	Touchscreen Cable
P4	Power Supply

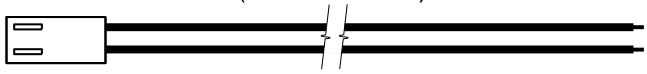
Power Requirments
55mA @ +5VDC±10% standby,
160mA average with touch,
240mA peak.

P2 Connections		Serial Communication Connections	
		DB 9	DB 25
1	Data Carrier Detect (DCD)	1	8
2	Data Set Ready (DSR)	6	6
3	Receive Data (RxD)	2	3
4	Ready To Send (RTS)	7	4
5	Transmit Data (TxD)	3	2
6	Clear To Send (CTS)	8	5
7	Data Terminal Ready (DTR)	4	20
8	N/C	9	22
9	Ground (GND)	5	7
10	Key	N/A	N/A

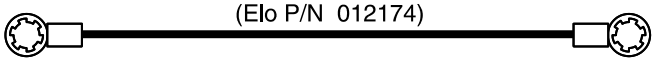
2x5 to DB9-Female Flat Ribbon Cable Assembly, 30 in.
Serial cable, from P2 to back of monitor
(Elo P/N 012138)



Two pin Red/Black flying lead - From P4 to power source
(Elo P/N 012111)

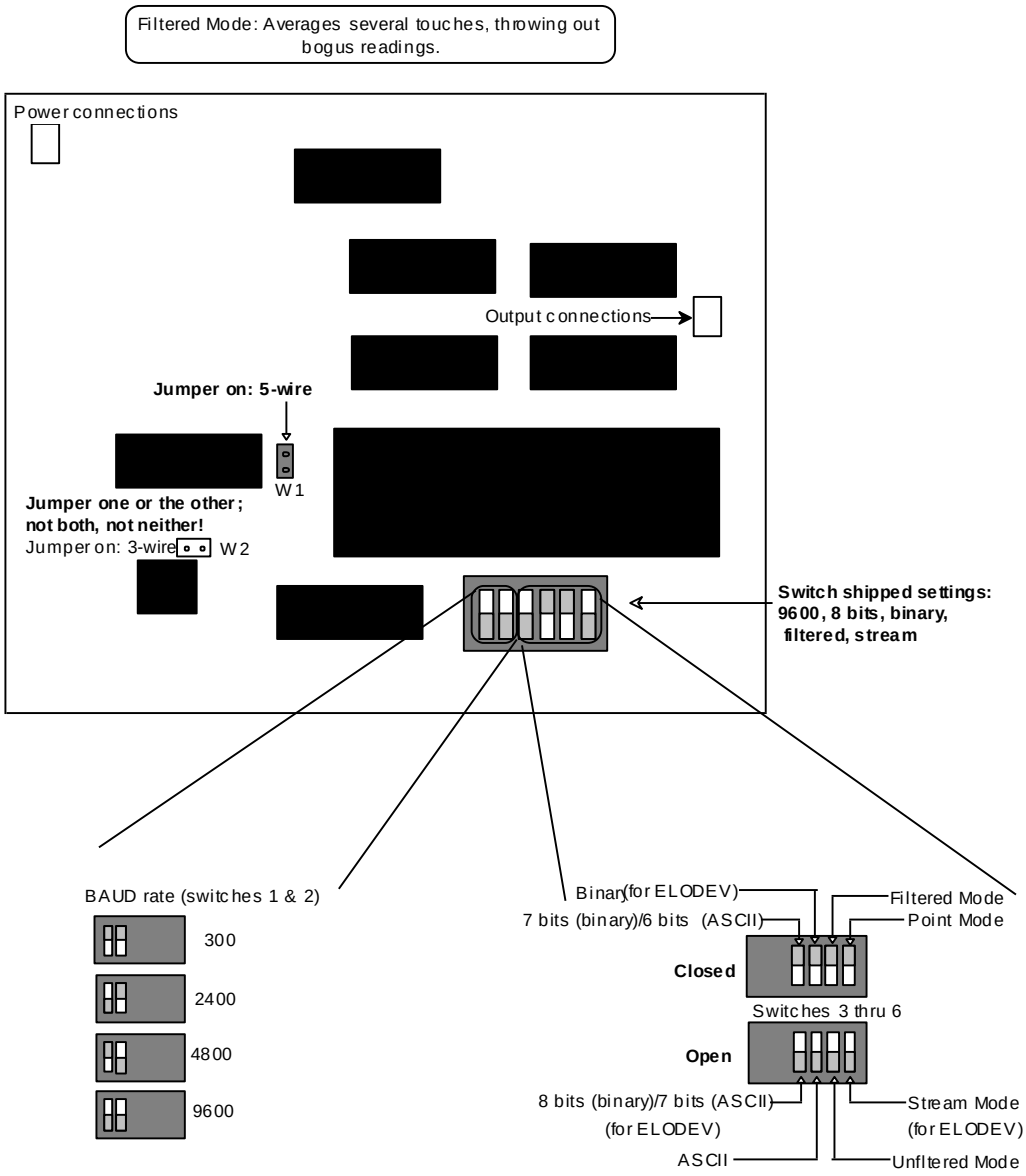


Cable, Ground with Ring Terminals - from PTH to Frame Ground
(Elo P/N 012174)



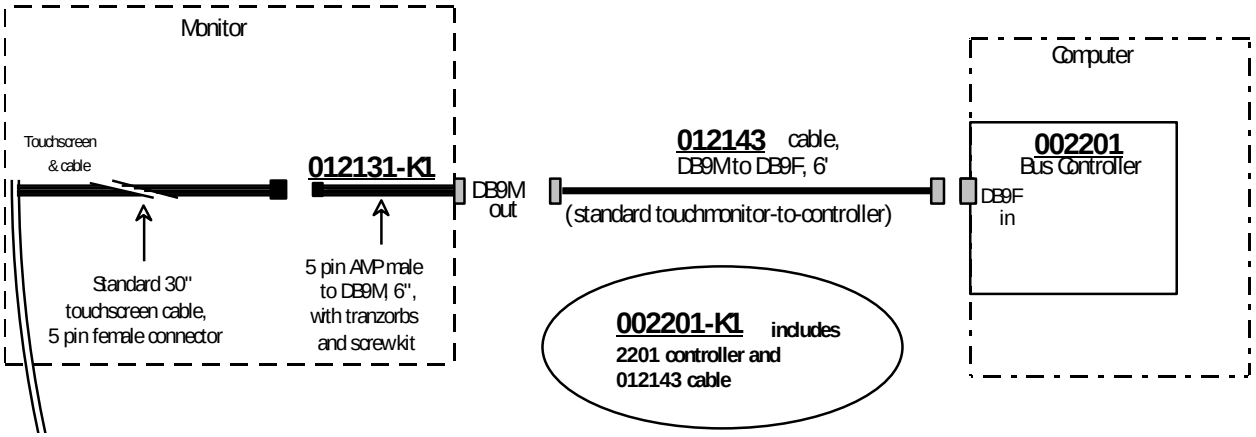
Power Supply, Internal Mount, +5 VDC, International Power (not pictured)
(Elo P/N 004100-204-K1)

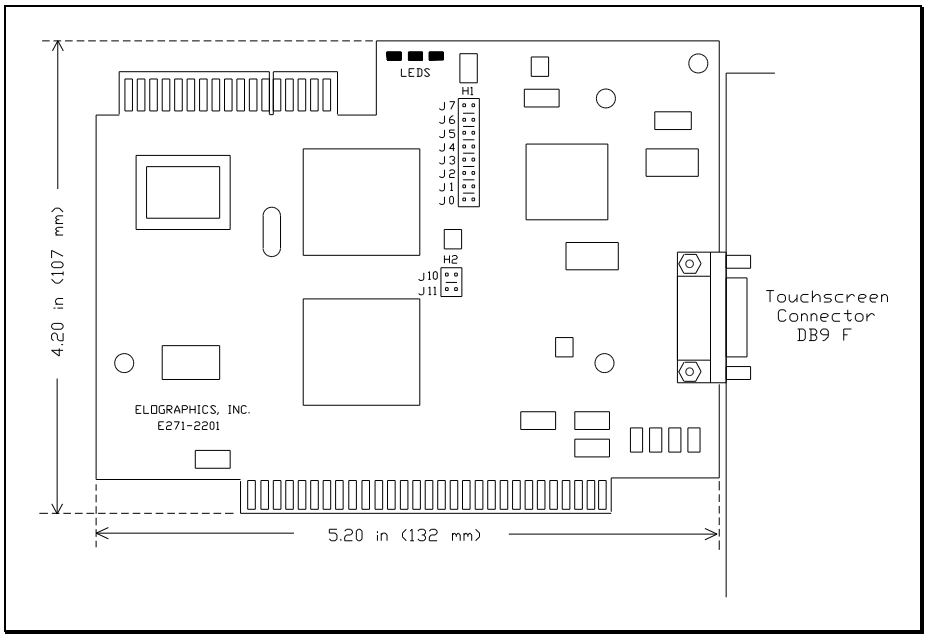
E271-140 AccuTouch Serial Controller



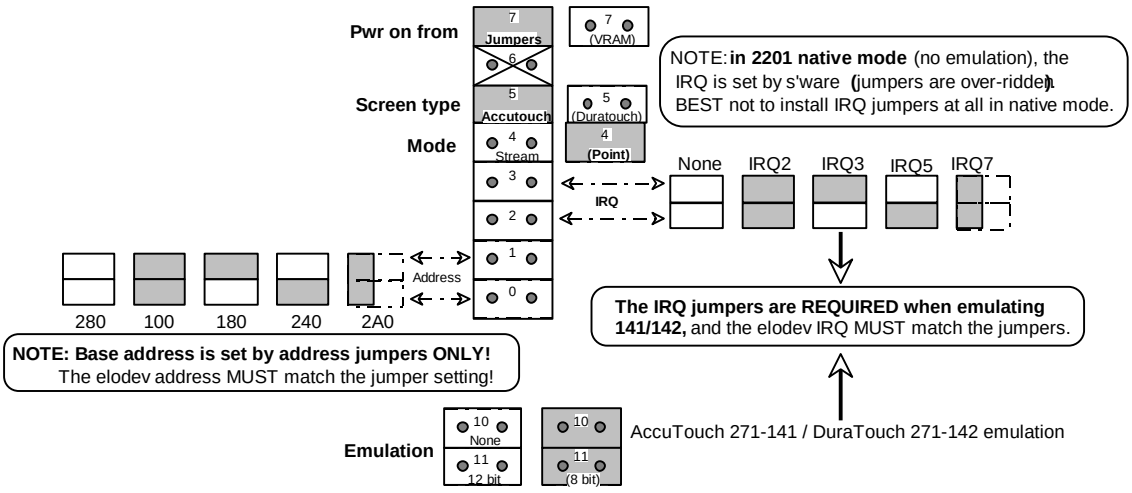
AccuTouch bus controller part numbers

AccuTouch Monitor (bus controller)





E271-2201 AccuTouch Bus Controller



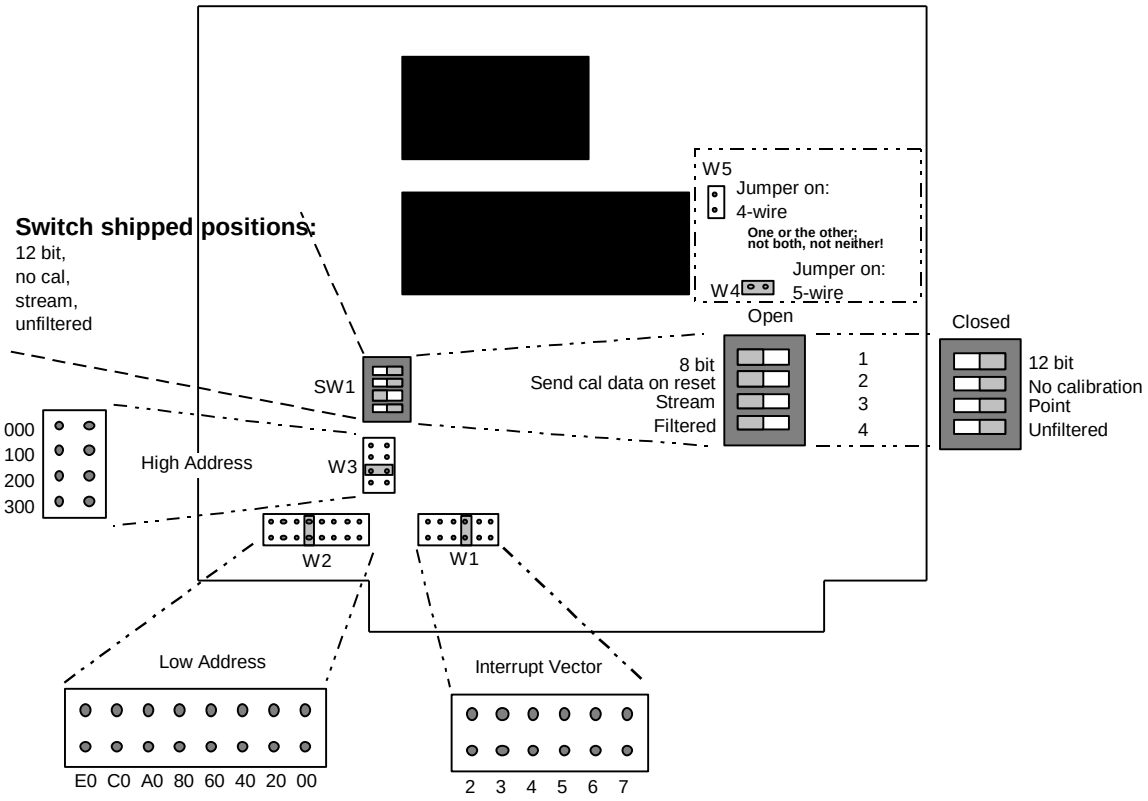
Diagnostic LEDs

Yellow (Communications) **Normal: Off; flicker on touch**
Problem: Stays on solid
Host failed to remove data packet
Shorted touchscreen or wiring

Green (Self-test) **Normal: Flashes 2/sec**
Problem: Stays on solid - Power-On Self-Test failed

Red (Warning) **Normal: Off**
Problem: Stays on solid - Improper communication

E271-141 Accutouch Bus Controller



Section 8 - General Notes

Serial Handshaking in general

RS232 handshaking was defined when the targets were mainly modems and terminals, so terms are defined in that context. The Data Terminal Equipment (or Host, or the computer) sends data to and receives data from the Data Circuit-terminating Equipment (the serial device).

Signal Abbr.	Full Name	Meaning
DTR	Data Terminal Ready	The computer is ready for serial operation
DSR	Data Set Ready	The serial peripheral device is ready to operate
RTS	Request to send	The computer is ready to receive (requesting *the peripheral device* to send)
CTS	Clear to send	The device is ready to receive
TD	Transmit data	Computer's transmit line
RD	Receive data	Device's transmit/computer's receive line
Ground		
DCD	Data Carrier Detect	Not applicable to most general serial devices
RI	Ring Indicator	Not applicable to most general serial devices

DTR is asserted by the computer COM port when the COM port is ready to operate.
DSR is asserted by the serial device when the device is ready to operate.

RTS and CTS are the actual handshaking lines.

RTS is asserted by the computer COM port whenever it can receive data; RTS is driven low when the COM port is busy processing data.

CTS is asserted by the serial device whenever it can receive data; CTS is driven low by the device when the device is busy processing data.

Elo Handshaking

In the following discussion, "asserted" means "driven high".

The DSR line is asserted when the controller is powered up (the Data Set is Ready).
The DTR line is asserted by the COM port when it has been initialized. The controller cannot transmit unless DTR is asserted, indicating that the Data Terminal is Ready. (*)

The controller keeps CTS low while it is running the Power-On Self-Test; after POST completes, the controller asserts CTS (saying, "Data terminal, I am ready to receive data, you are Clear to Send").

On receipt of CTS, the software driver instructs the computer to send a diagnostics query (literally, transmits 'd0000000', the SmartSet diagnostics query command, on the TxD line). The computer hasn't received data from the controller yet, so the computer is ready to receive and so indicates by having RTS asserted (Request *to device* To Send a reply).

When the controller receives the command and verifies that it is a valid command, it de-asserts CTS to tell the computer not to send further data until it has processed the command. Since RTS is asserted, the controller sends the response packet on the Rx line (then reasserts CTS); the driver interprets the response and displays the POST results.

For all intents and purposes, handshaking is required only when the driver loads. The sequence that matters is:

DSR	Asserted by controller on power up
DTR	Asserted by COM port on power up/initialization (or float)
CTS	Asserted by controller after POST
RTS	Asserted by host when Rx line is inactive (or float) (host ready to receive).

In reality, if neither DTR or RTS is pulled low - that is, if they are not connected at all - the only handshaking signals needed are:

DSR	Asserted by controller on power up
and	
CTS	Asserted by controller after POST.

This is because both DTR and RTS are connected through diodes to a pullup resistor in the controller. If either line is pulled to ground, the UART on the controller will see a logic low (the 0.7 volt drop across the diode); otherwise, both DTR and RTS will be seen as high.

(*) DTR, strictly speaking, doesn't HAVE to be asserted; as long as neither DTR or RTS is pulled low (meaning, both float), the controller can transmit.

In normal operation, the controller sends touch data to the computer (the computer sends no commands to the controller once the driver load/initialization sequence is finished). Normal handshaking is available, i.e., if the computer takes RTS low, the controller will stop sending until RTS is reasserted. DTR remains asserted full time. DSR and CTS remain asserted, DSR because it is connected to the power line of the controller, and CTS because the controller is never receiving data during normal operation; therefore, the controller is always "ready to receive", so it keeps Clear to Send asserted.

When the -p- flag is used with ELODEV, ELODEV is instructed not to execute the subroutine that sends the diagnostics query command and checks the diagnostics results. The code that examines handshake line status is in the subroutine that checks the diagnostics results. Because the command is not sent and the diagnostic results are not checked, the driver will load without regard to the handshake line status.

Handshaking error messages associated with DOS and Windows drivers

Error Messages as a Result of Serial Communications Handshaking or Data Line Faults **when ELODEV Loads**

Signal Name	DB9/DB25 Pin #	Signal Origin	Error Message	ELODEV-reported Handshake when forced with -p-
DCD	1/8			
Rx	2/3	Controller transmit	Controller not responding	(DSR, CTS)
TxD	3/2	COM port transmit	Controller not responding	(DSR, CTS)
DTR	4/20	COM Port	*Controller not responding	*(DSR)
Ground	5/7	Ground	Controller not detected	(CTS)
DSR	6/6	Controller (Power on)	Controller not detected	CTS)
RTS	7/4	COM port	*Controller not responding	*(DSR)
CTS	8/5	Controller (post-POST)	Cannot output to controller	(DSR)
RD	9/22			

** only when grounded; normal ops when open*

The only thing that will completely disable serial touchscreen controller communications when using ELODEV with the -p- flag is a fault on Pin 2, Receive Data, which is the controller's data transmit line. The -p- override will allow communications in all other line fault cases.

DSR is used to "detect" a controller (controller's +5 is applied as long as controller power is on); thus the message "controller not detected" if DSR is low (or if the ground is missing).

CTS signals that controller is ready to operate (asserted high at completion of Power-On Self-Test). When ELODEV loads, the computer queries the controller for the POST results. The computer cannot send the query to the controller (without -p-) if the controller does not assert the CTS line; thus the message "Cannot output to controller" if the CTS line is not asserted.

Obviously, if the controller doesn't send the POST results on Rx, the message "controller not responding" will be displayed. This condition exists when there is a fault with either Rx or TxD. Also, if either DTR or RTS is low (pulled to ground, not floating), the controller cannot send data, and the same message will be given..

Minimum Signal Requirements for **Normal** operation **(SmartSet Controllers)**

Signal Name	DB9/DB25 Pin #	Signal Origin	Error Message, if line is defective
Rx	2/3	Controller transmit	Controller not responding
TxD	3/2	COM port transmit	Controller not responding
Ground	5/7	Ground	Controller not detected
DSR	6/6	Controller (Power on)	Controller not detected
CTS	8/5	Controller (post-POST)	Cannot output to controller

(DTR and RTS must not be low)

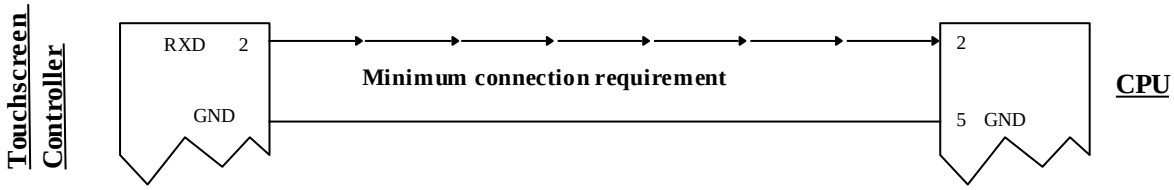
Minimum Signal Requirements for **Non-handshake** operation **(ELODEV -p- , D&W drivers only)**

Signal Name	DB9/DB25 Pin #	Signal Origin
Rx	2/3	Controller transmit
Ground	5/7	Ground

Elo "Handshaking" Connection Requirements

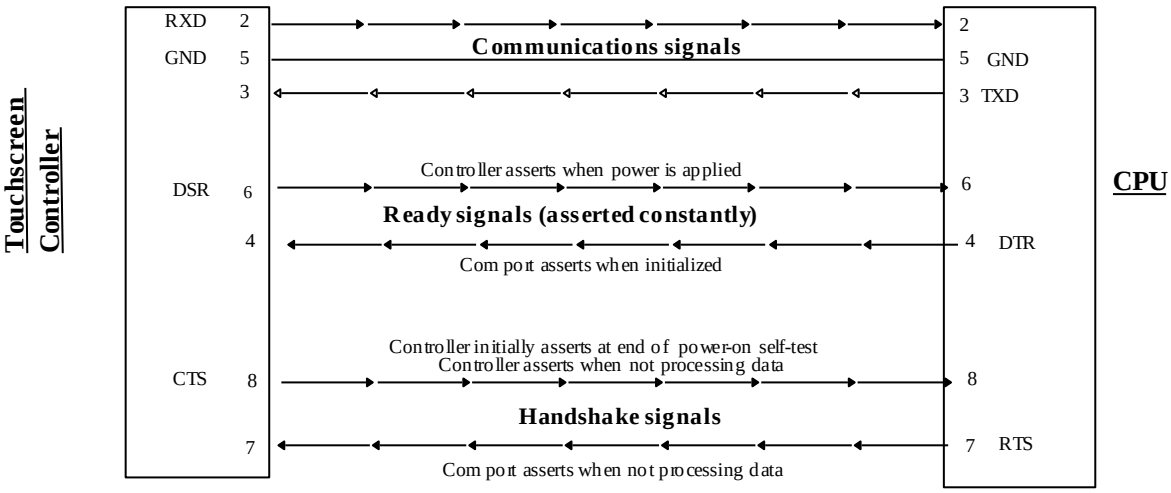
One-way communications

Pre-SmartSet controllers and SmartSet controllers with handshaking disabled (elodev -p-, or J3 installed)



Full Handshaking

SmartSet controllers



Elo SmartSet Data Packet and Command Format - Overview

The fundamental SmartSet data packet is 8 bytes; the serial data packet has a byte added to each end (the lead-in byte and the checksum byte). See page 3 of this document, **SmartSet Touch Response Serial Data Packet**.

The 55 you see at the beginning of the serial packet (using COMDUMP) is the lead-in byte. It was added to give the driver something unique to look for; when 55 hex is detected, the driver says "all right, we're going to see if this is a data packet". This allows the COM port to ignore any data that might be garbage (suppose the user selects the wrong COM port, for instance). Thus, the driver can ignore data until a 55 is detected, and only then does it need to check for a valid data packet. This cuts CPU overhead. The 55 value was chosen because it is a somewhat unique number in binary: 0101 0101 (every other bit set). It can also be expressed as an upper-case 'U' (because the ASCII standard character set defines that value to be 'U'). 0101 0101 binary = 55 hex = 85 decimal = 'U'.

The checksum byte was added as a standard method for checking serial data integrity. The sending device adds all the values of the bytes in the packet and puts the result in the checksum byte; the receiving device adds all the values of the bytes it receives in the packet and compares that sum to the value sent in the checksum byte. If the sending checksum byte is equal to the calculated received checksum, it's a fair assumption that no bits were changed or characters lost in the transmission.

In summation to this point: the actual command and data packet is 8 bytes; the serial transmission protocol adds two bytes, a lead-in to serve as a "wake up flag" and a trailing checksum to help validate the data. (In actual practice, these two are used together to indicate a valid data packet: the lead-in 55 alerts the receiving device to total the 55 with the next 8 byte values and check to see that the calculated sum is equal to the checksum in the tenth transmitted byte).

The central 8 bytes are the actual command and data. The first is the Command Byte and it is thought of in terms of an ASCII character that (generally) serves as a mnemonic for the command; hence 54 hex indicates a Touch (54 hex is the ASCII value of an upper-case T). The T command is a "one-way" command, sent by the controller to the host (computer), to report touch location. Other commands are "two-way"; 'D', for instance, can be sent to the controller to command it to perform diagnostics, and the controller can send a 'D' response back to report diagnostics results.

In a touch packet, the next byte is for status (detailed later in this document). The remaining six bytes are for data. In the case of Touch reporting, these contain X, Y and Z data; other commands use some or all of the data bytes to specify actions or results.

Commands may be sent TO the controller to modify its settings (Set commands); commands may be sent TO the controller to get controller settings (Query commands); commands may be sent BY the controller to report results/settings/events (Response "commands"); and the Acknowledge "command" may be sent BY the controller to acknowledge receipt of a command. Command byte formats are detailed later in this document, as is a summary of the data packet structure.

SmartSet Command and Reply Format

The computer (host) can send only two types of commands to the SmartSet controller: Set and Query.

Set: Modifies the controller settings
The *set* command byte is always an **upper-case** letter
Various letters specify the action to take
Followed by seven data bytes that specify how to modify the settings

Query: Tells the controller to report its settings
The *Query* command byte is always a **lower-case** letter
Various letters specify the settings to be reported
Any data bytes which follow are ignored

The controller can send only two responses: Response and Acknowledge

Response: The *Response* command byte is always an **upper-case** letter
The letter is the same as the command that evoked the response
Responses are only given to Query commands (not Set commands)
The seven data bytes report the controller settings

Acknowledge: Sent after every Response to a Query
Sent after every Set command is received
The *Acknowledge* command byte is always an upper-case 'A'
Data bytes that follow should be all 0 (non-zero indicates errors)

Summary:

Host sends Query
Controller sends Response
Controller sends Acknowledge

Host sends Set
Controller sends Acknowledge

SmartSet *Touch Response* Serial Data Packet (10 bytes)

Lead-in	Command	Data							Checksum
		Status	Xlo	Xhi	Ylo	Yhi	Zlo	Zhi	
10101010	10100100	xxxxxxxx	xxxxxxxx	0000xxxx	xxxxxxxx	0000xxxx	xxxxxxxx	0000xxxx	xxxxxxxx
55	54								
'U'*	'T'**	***	\		****				/

* Lead-in is always 'U' (55 hex)

** 'T' (54 hex) indicates a touch event; this is actually a RESPONSE, not a COMMAND, but this byte is referred to as the COMMAND BYTE

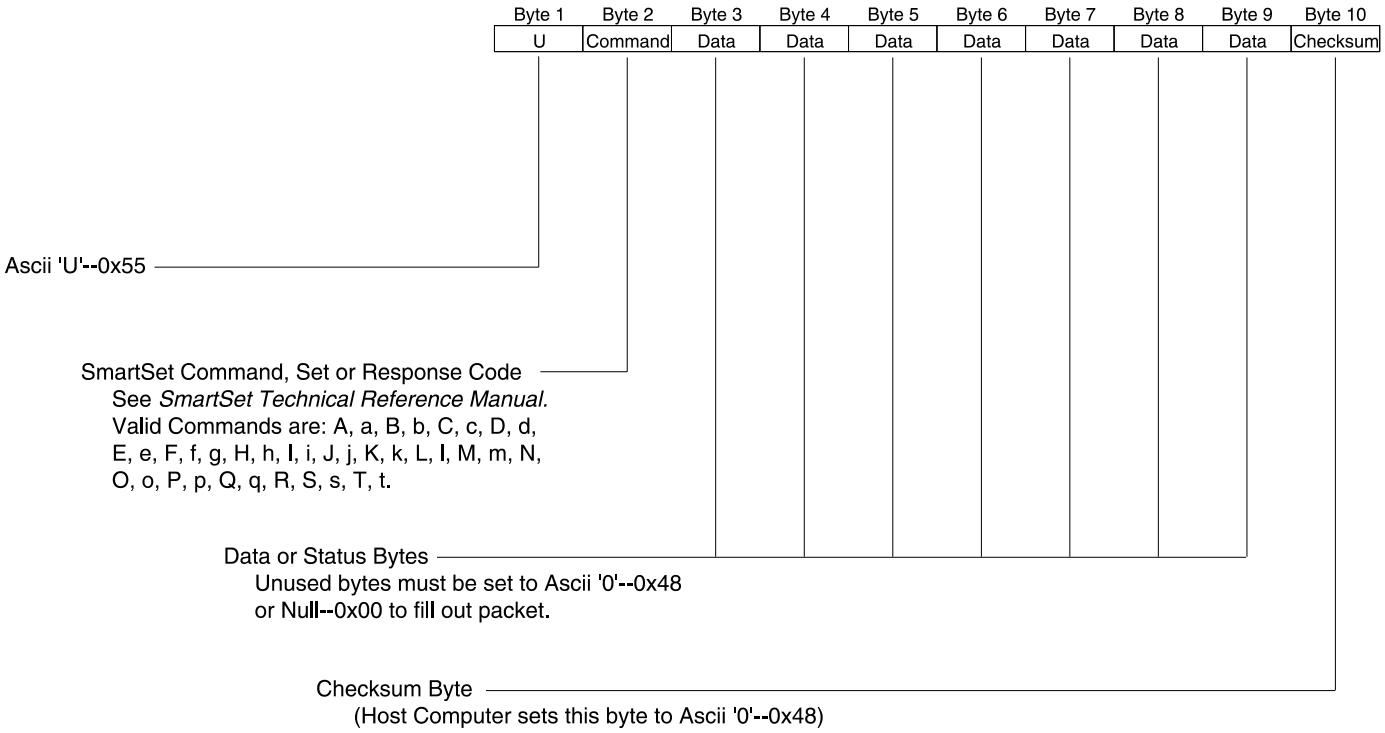
*** 1: initial touch (00000001)
2: Stream touch (00000010)
4: Untouch (00000100)
20: Warning (10000000) (additive with touch status, i.e. 21)
40: Out of range (10000000) (additive with touch status, i.e. 41)
80: Z-axis active (10000000) (additive with touch status, i.e. 81) (generally indicates IntelliTouch)

**** Lo bytes: 00000000 to 11111111 (0 to 255 decimal / 0 to FF hex)
Hi bytes: 00000000 to 00001111 (256 to 3840 decimal / 0 to 0F hex))
Maximum range: 12 bits (0 to 4095 decimal / 0 to 0FFF hex)

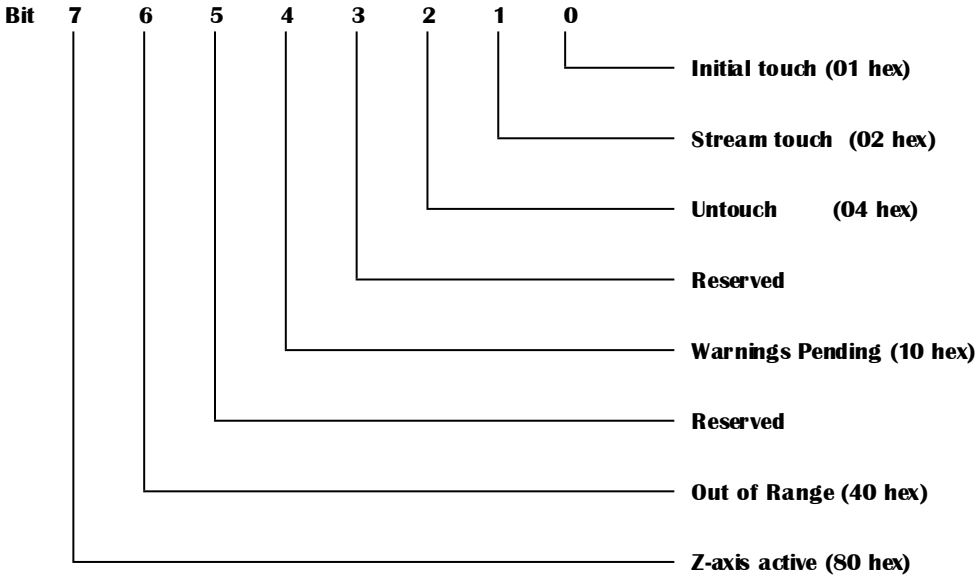
NOTES:
Above is a breakdown of the data packet received at the COM port when the screen is touched.
See the SmartSet Command and Reply part of this document for the general SmartSet controller command and reply format.
See the SmartSet Technical Reference for a detailed explanation of commands and responses.

The SmartSet Technical reference Manual notes that the checksum feature is normally disabled when sending commands from the computer to the controller , in which case the host is not required to send a properly calculated checksum;
NOTE, however, that a dummy byte (such as 0) must be sent to maintain the correct data packet length (10 bytes).
NOTE also that this applies only to commands sent TO the controller; the checksum is always enabled when the controller is sending touch packets to the computer.

SmartSet *General Command/Response* Serial Data Packet (10 bytes)



SmartSet Controller Status Byte



SmartSet Bus Controller Communications (8 bytes)

I/O address	Byte content	Comments
280	Command Byte	Bit 7 is the NOT READY bit; if set (1xxxxxxx), the controller is not ready to receive
281	Status Byte	
282	Xlo	
283	Xhi	
284	Ylo	
285	Yhi	
286	Zlo	
287	Zhi	

NOTES:
The base address is assumed to be 280 for this example; any valid base address may be used
The host must wait for an Acknowledge before issuing another command.
The NOT READY bit must be zero before issuing another command.
The Elo controller asserts the IRQ line when the NOT READY bit transitions from 1 to zero.