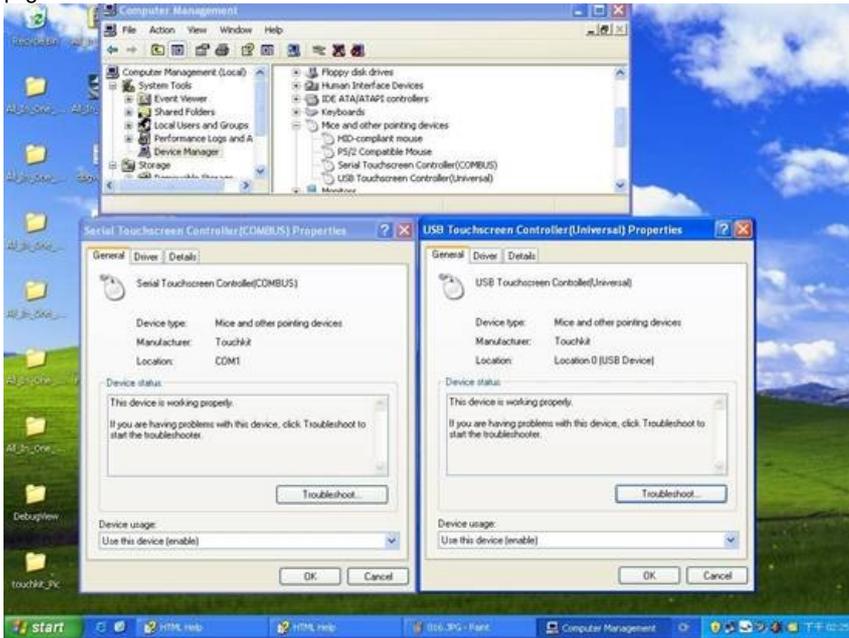


eGalaxTouch Devices in Windows® Device Manager

After driver installation completed, all of eGalaxTouch touchscreen controller devices will be list in the "Mice and other pointing devices" folder in Windows® Device Manager. Click on the specified eGalaxTouch touchscreen controller device and select the properties on the pop up context menu, the device property pages will show as below



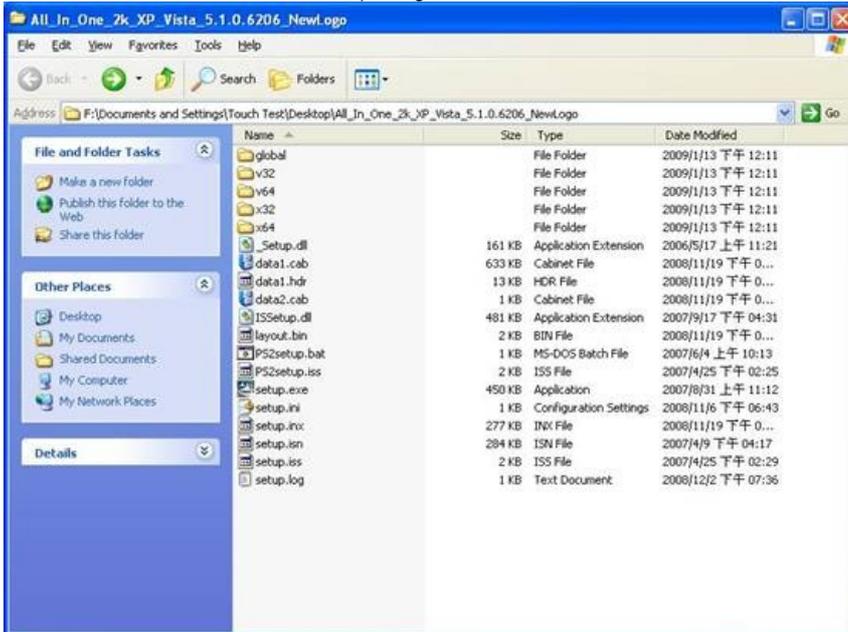
In addition to some system provided property pages for mice function setting, there are some other property pages in this property sheet for eGalaxTouch touchscreen controller specific configuration setting. If user got the driver files(.inf, .sys) from Microsoft® via Internet, user can do configuration and calibration for the devices with this property pages. If user got the complete software driver installation package to install the software, eGalaxTouch also provides another software utilities for advanced setting for some advanced application.

See Also

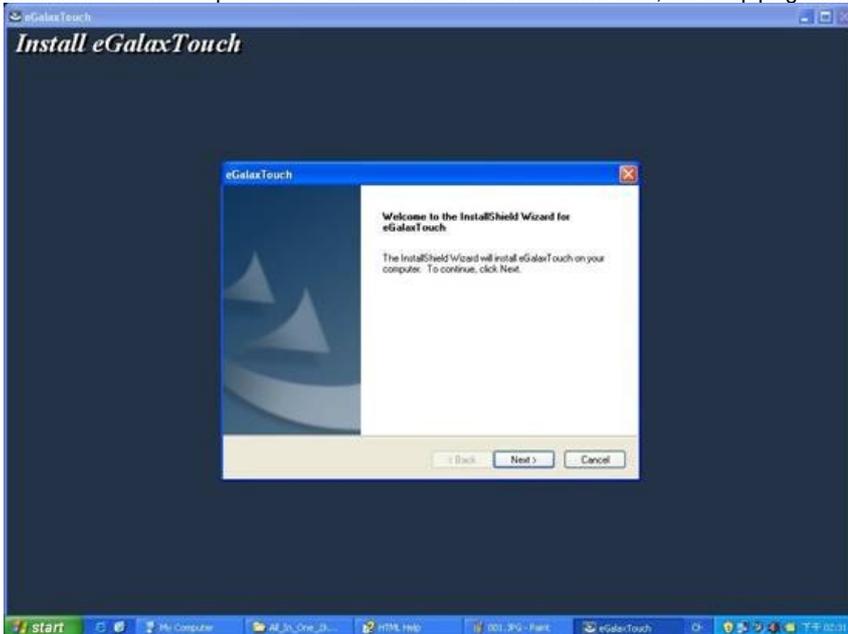
[eGalaxTouch Configuration Utility](#)
[xTouchmon Utility](#), [Monitor rotation](#), [Display Resolution Setting Property Page](#), [Beep](#), [Mouse Emulation Mode Tools Property Page](#), [Calibration](#), [Linearization](#), [Drawing Test Display Property Page](#), [Multiple Monitor](#), [Split Display](#), [Edge Compensation Property Page](#), [Edge Compensation Parameters Hardware Property Page](#), [Hardware Information About Property Page](#)

eGalaxTouch Software Installation Guide for Windows® 2000 / Windows® XP / Windows Vista® / Windows® 7 operating system

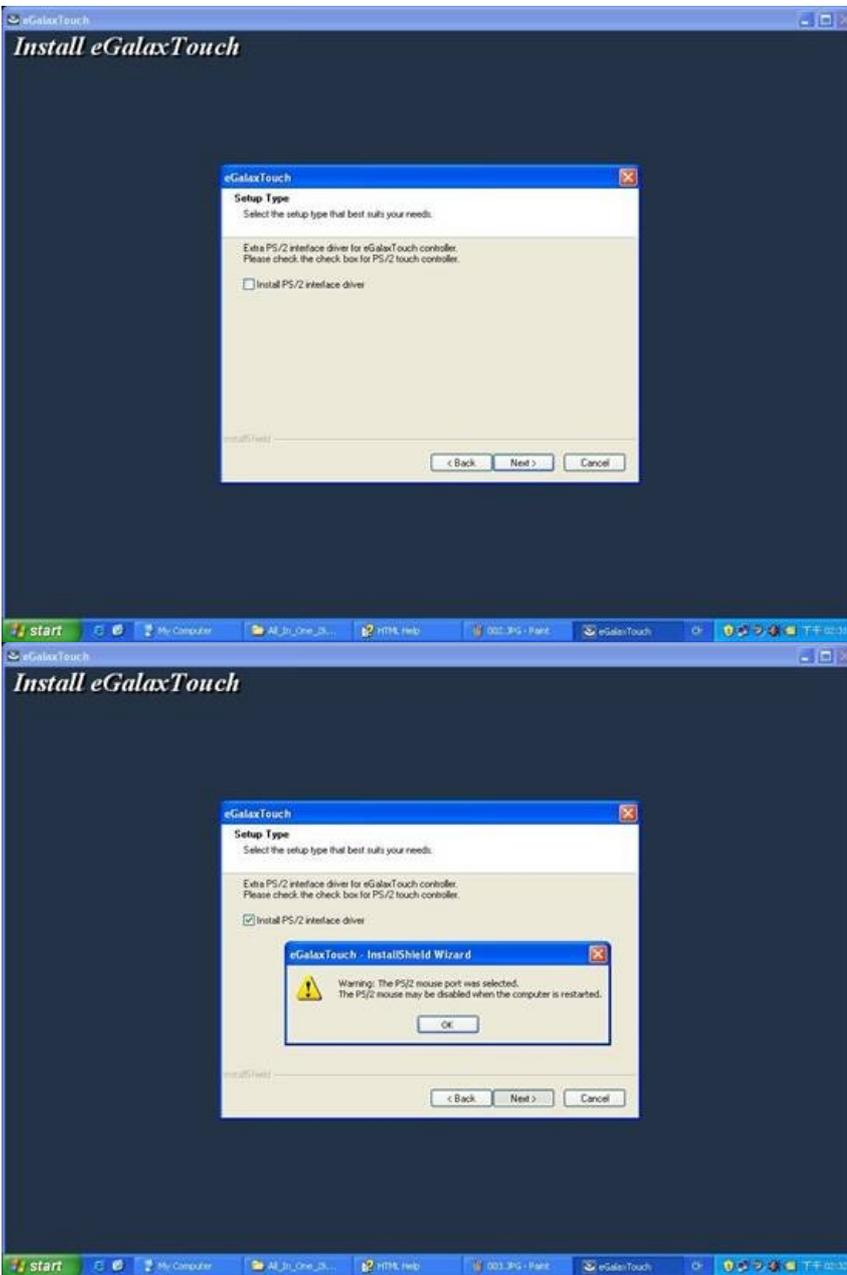
eGalaxTouch driver software installation package was created with InstallShield® tool. The installation package contains files as below



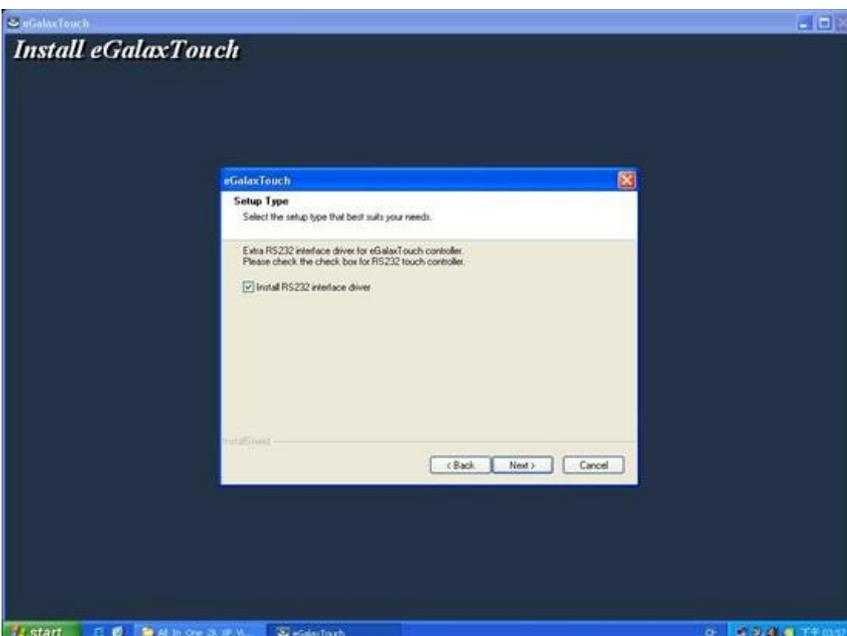
Double click at the setup.exe file to start software driver installation. Then, the setup program will guide user to complete software installation.



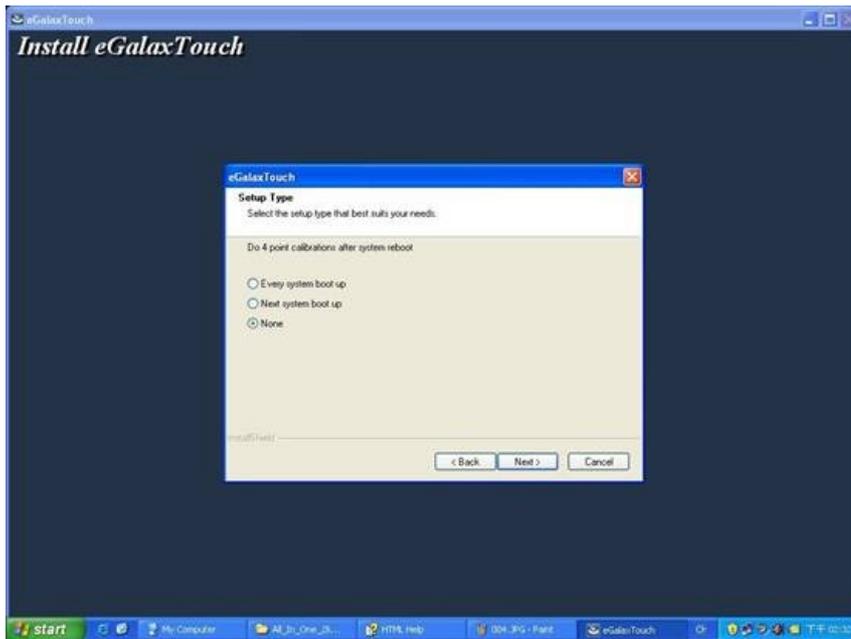
Press **Next** button to continue installation, then, a new dialog popped up as below,



This dialog shows to ask user if the PS2 filter driver for touch screen to be installed. User can check this check for PS2 filter driver installation. The standard PS2 mice can still work well after this filter driver installed because eGalaxTouch PS2 filter driver can work with both standard PS2 mice and PS2 touchscreen. But, this filter driver may does not work with other devices with the PS2 mouse port. After check or uncheck this check box, press **Next** button to continue installation. Then, it shows new dialog as below,

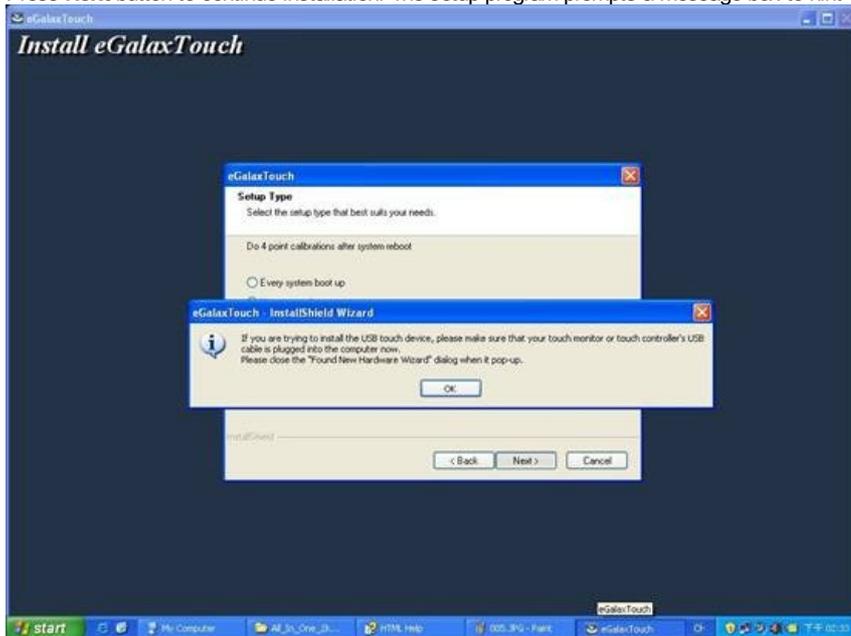


This dialog shows to ask user if the RS-232 driver for touch screen to be installed. User can check this check for RS-232 driver installation. After check or uncheck this check box, press **Next** button to continue installation. Then, it shows new dialog as below,



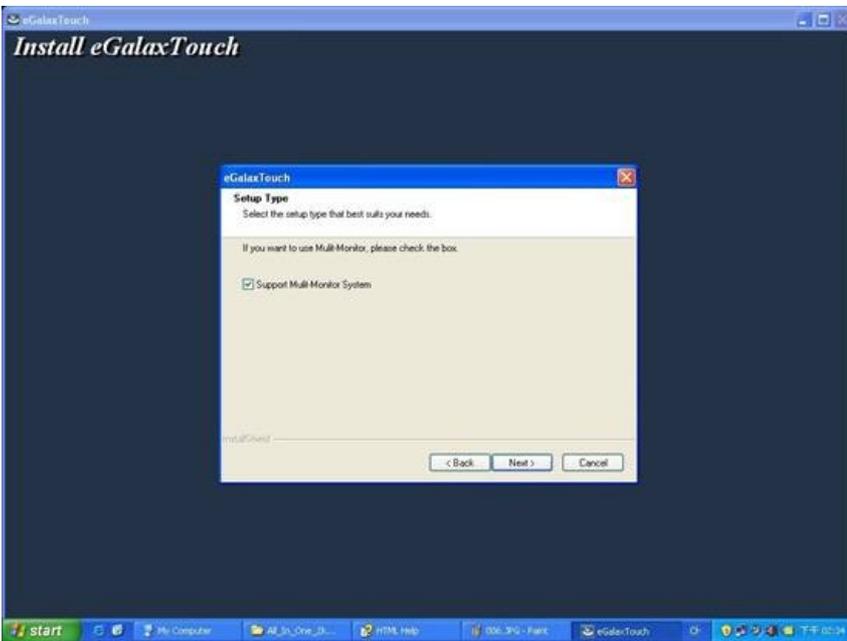
Also, eGalaxTouch software provides user with a fast 4 points calibration. If the system needs 4 points calibration to make sure the touch accuracy every system reboot, user can check this check box. The 4 points calibration window will be popped up for calibration whenever system boot up if this check box was checked during driver software installation.

Press **Next** button to continue installation. The setup program prompts a message box to hint

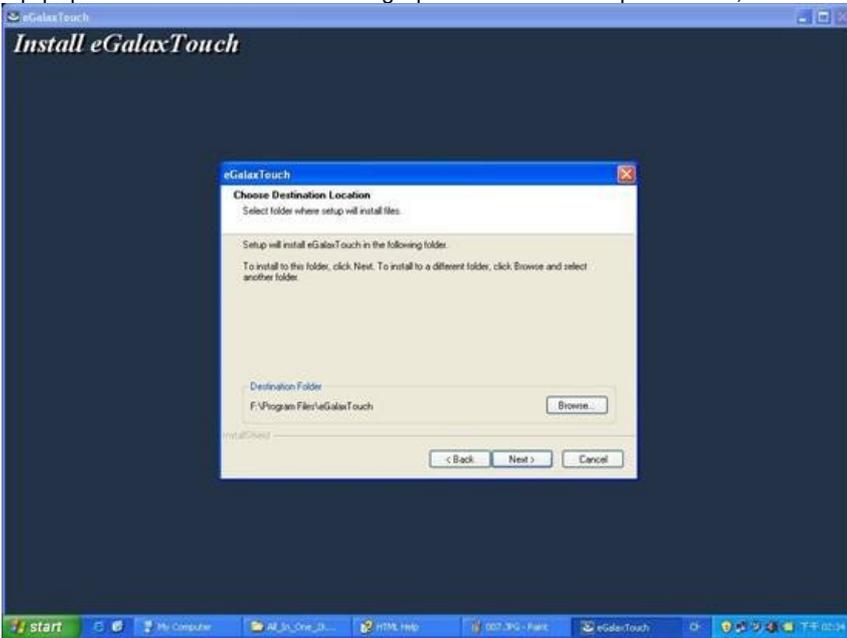


user to make sure that the eGalaxTouch USB controller devices were well connected with system USB ports to guarantee the USB eGalaxTouch device drivers updated after driver installation. Then, just press **OK** to continue,

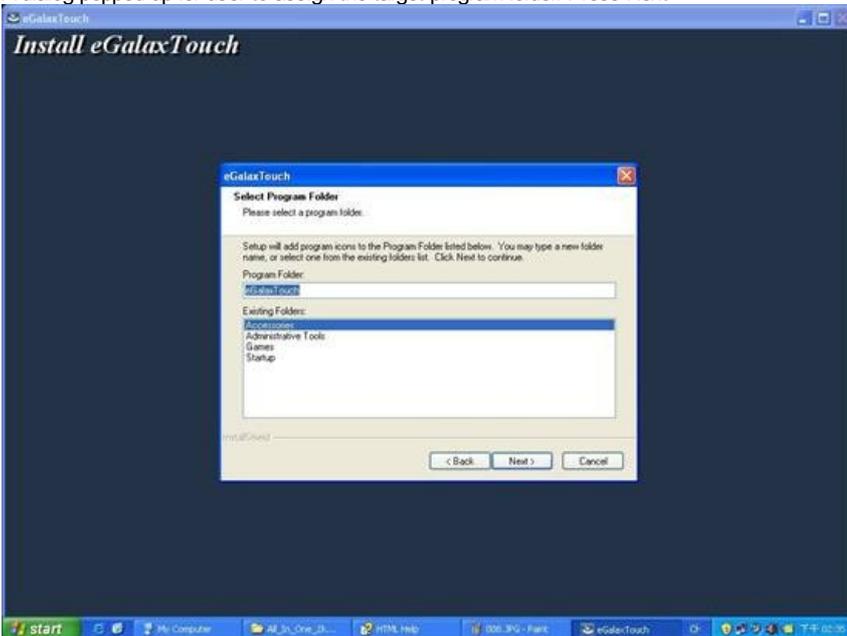
eGalaxTouch driver package supports multiple monitor operation. If your application is for multiple monitors system, please check the check box to support multiple monitors. So that, the a multiple monitor setting property page will be shown in eGalaxTouch utility. Then, press **Next** to continue.



A pop up window for user to choose the target path the files will be copied to. Then, Press **Next** to continue,

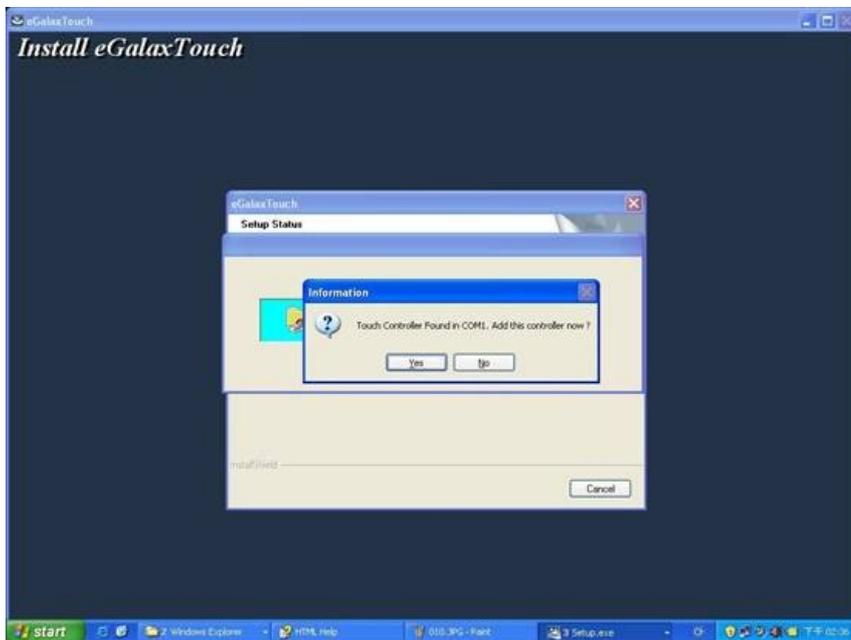
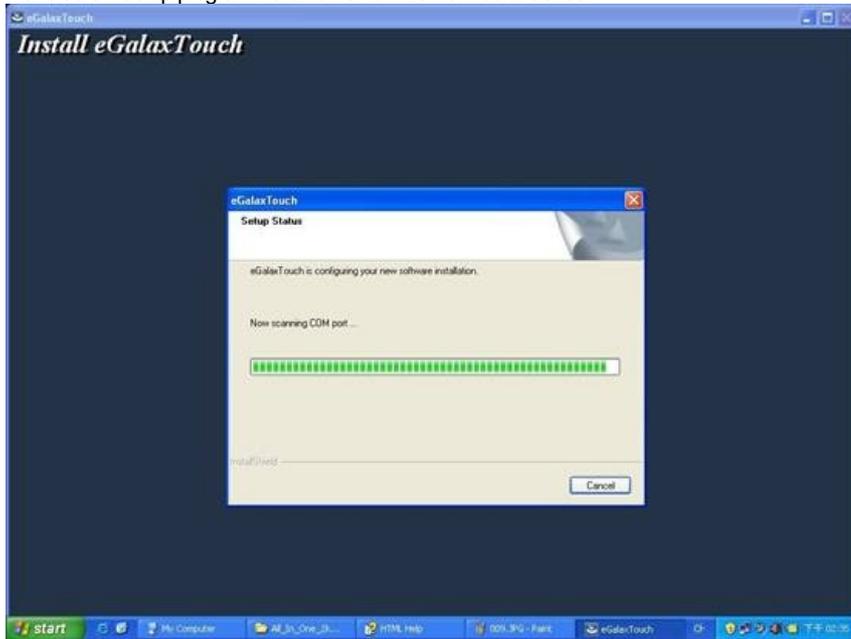


A dialog popped up for user to assign the target program folder. Press Next

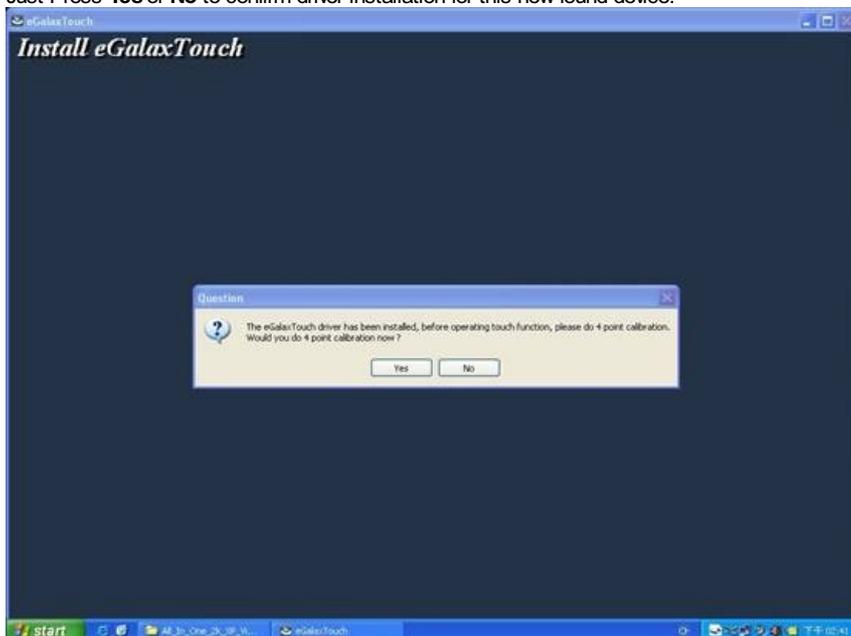


If install RS-232 driver check box was checked, the setup program will scan system COM ports.

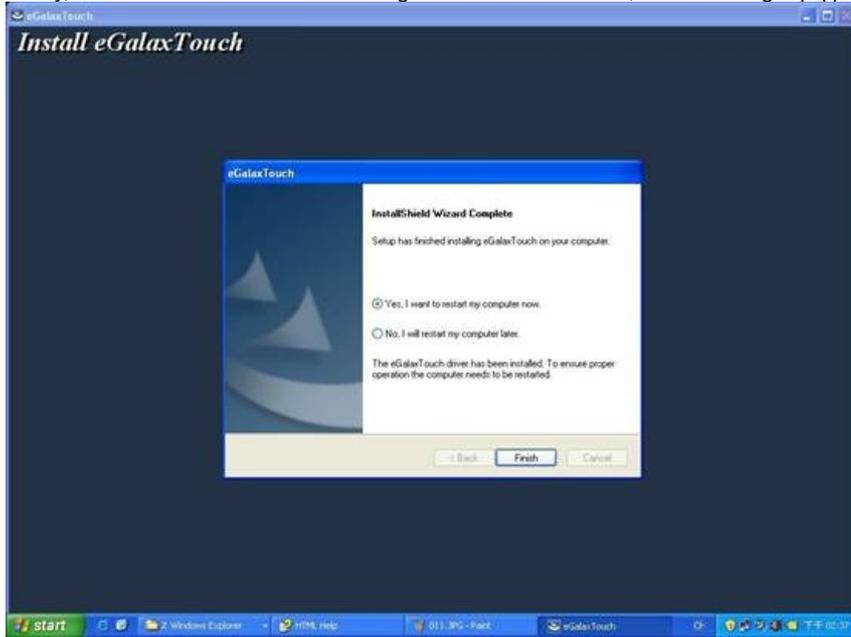
The setup program will scan system COM ports to detect if any eGalaxTouch serial device was connected. If yes, it also pops up a message box for user to make sure if setup program install driver for this new found device.



Just Press **Yes** or **No** to confirm driver installation for this new found device.



Finally, if PS2 filter driver was installed during this software installation, a reboot dialog is popped up for system reboot request to complete driver installation.

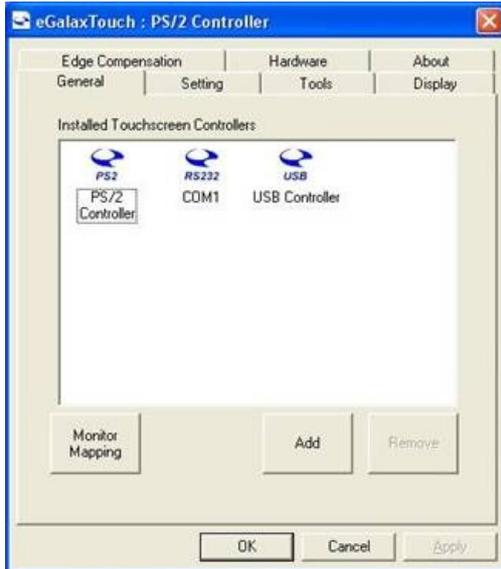


After driver installation, a shortcut will be generated and shown on the desktop. And, a new file group generated for eGalaxTouch as below,



General Property Page

The general property page in eGalaxTouch utility shows all of eGalaxTouch touchscreen controllers installed as below, including RS232, USB and PS2 interfaces.



User can select the controller list in the window to do configuration for it. When user use a mouse or other input device to select the controller device in the controller list window, The controller name will be shown in the title bar of the main window(the property sheet). Also, all of the other property pages will be updated for this selected controller. In some application, user may in place edit and change the controller name for easy identification.



In addition, there are 3 function push buttons in this property page.

1. Add

The function button is used for serial RS232 controllers only. Press this button to search the eGalaxTouch serial controllers connected with the system COM ports. Whenever it finds a new eGalaxTouch serial controller, a new serial controller icon object will be shown in the controller list window automatically.

USB eGalaxTouch device supports plug and play, the icon object for USB controller will be shown in the controller list window automatically when the USB controller is connected with the system USB port. And, the icon object for the USB controller will disappear automatically as soon as the device was removed from the system USB port.

eGalaxTouch PS2 driver support PS2 mice and eGalaxTouch touchscreen controller. It can works with both PS2 mice and eGalaxTouch touchscreen PS2 controller. After the eGalaxTouch PS2 driver was installed, this utility assumes the PS2 touchscreen controller exists and is always shown in the controller list window.

2. Remove

This function button is used for serial RS232 controllers only. This button will be greyed and disabled automatically when the selected controller in the controller list window is not RS232 type. Press to remove and uninstall the selected serial RS232 controller from the system. Then, this serial RS232 icon object in controller list window disappears automatically.

USB eGalaxTouch device supports plug and play, the icon object for USB controller will be shown in the controller list window automatically when the USB controller is connected with the system USB port. And, the icon object for the USB controller will disappear automatically as soon as the device was removed from the system USB port.

eGalaxTouch utility does not allow user to remove/uninstall the PS2 device driver dynamically. To uninstall the eGalaxTouch PS2 driver, user needs to go to Windows® Device Manager to do un-installation. In addition, after PS2 un-installation, it needs to system reboot to complete un-installation.

3. Monitor Mapping

eGalaxTouch driver utility supports multiple monitor and display system. This function will help user to does monitor mapping. a new window will be popped-up on a monitor as below. Please touch the touchscreen panel to do mapping. In addition , please do calibration after finish monitor mapping.

1



Press ESC to quit.

Press SPACE bar to skip to Next Screen.



Auto skip to next screen...

See Also

[4 points, linearization/25 points calibration, draw testing](#)
[Beep, Mouse emulation, Double click, Auto right click](#)
[Multiple monitor and split display configuration](#)

General Property Page

The general property page in eGalaxTouch utility shows all of eGalaxTouch touchscreen controllers installed as below, including RS232, USB and PS2 interfaces.



User can select the controller list in the window to do configuration for it. When user use a mouse or other input device to select the controller device in the controller list window, The controller name will be shown in the title bar of the main window(the property sheet). Also, all of the other property pages will be updated for this selected controller. In some application, user may in place edit and change the controller name for easy identification.



In addition, there are 3 function push buttons in this property page.

1. Add

The function button is used for serial RS232 controllers only. Press this button to search the eGalaxTouch serial controllers connected with the system COM ports. Whenever it finds a new eGalaxTouch serial controller, a new serial controller icon object will be shown in the controller list window automatically.

USB eGalaxTouch device supports plug and play, the icon object for USB controller will be shown in the controller list window automatically when the USB controller is connected with the system USB port. And, the icon object for the USB controller will disappear automatically as soon as the device was removed from the system USB port.

eGalaxTouch PS2 driver support PS2 mice and eGalaxTouch touchscreen controller. It can works with both PS2 mice and eGalaxTouch touchscreen PS2 controller. After the eGalaxTouch PS2 driver was installed, this utility assumes the PS2 touchscreen controller exists and is always shown in the controller list window.

2. Remove

This function button is used for serial RS232 controllers only. This button will be greyed and disabled automatically when the selected controller in the controller list window is not RS232 type. Press to remove and uninstall the selected serial RS232 controller from the system. Then, this serial RS232 icon object in controller list window disappears automatically.

USB eGalaxTouch device supports plug and play, the icon object for USB controller will be shown in the controller list window automatically when the USB controller is connected with the system USB port. And, the icon object for the USB controller will disappear automatically as soon as the device was removed from the system USB port.

eGalaxTouch utility does not allow user to remove/uninstall the PS2 device driver dynamically. To uninstall the eGalaxTouch PS2 driver, user needs to go to Windows® Device Manager to do un-installation. In addition, after PS2 un-installation, it needs to system reboot to complete un-installation.

3. Monitor Mapping

eGalaxTouch driver utility supports multiple monitor and display system. This function will help user to does monitor mapping. a new window will be popped-up on a monitor as below. Please touch the touchscreen panel to do mapping. In addition , please do calibration after finish monitor mapping.

1



Press ESC to quit.

Press SPACE bar to skip to Next Screen.



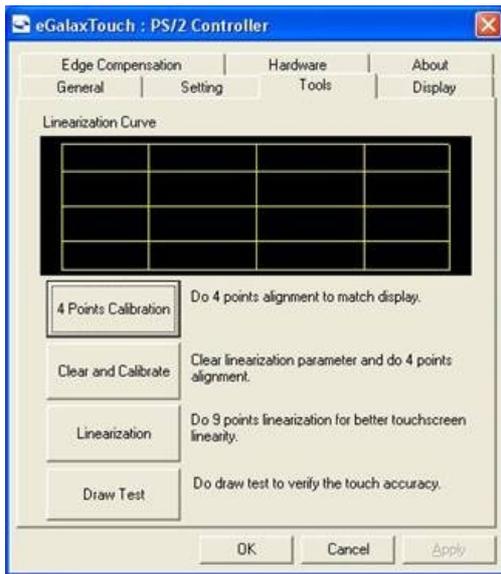
Auto skip to next screen...

See Also

[4 points, linearization/25 points calibration, draw testing](#)
[Beep, Mouse emulation, Double click, Auto right click](#)
[Multiple monitor and split display configuration](#)

Tool Property Page

Calibration, draw test tools and the linearity curve of the touchscreen were list in this property page shown as below for user to do touchscreen calibration and touch position test.



User can do calibration or draw test by pressing the function push buttons.

1. Linearization Curve

Linearization curve of the touchscreen is list in this page for reference and trouble shooting purpose.

2. 4 points calibration

It needs calibration before the touchscreen can work accurately. Whenever the user feel the accuracy lost, user can do calibration again to get a more accuracy touch function.

Pressing this button, a new window will be popped-up at the location when the touchscreen was mapped to area for this touch system to guide the user do 4 points calibration.

User should follows the guide to touch and hold the blinking symbol in the calibration window until it shows '¡\$OK¡' to make sure that the utility can gather enough data for computation.



3. Clear and Calibration

Press this button to erase the 25 points calibration/linearization parameters and force user to do 4 points calibration again. After 25 points calibration data was clear, the 4 points calibration data will be invalid. It needs to do 4 points calibration.

4. Linearization

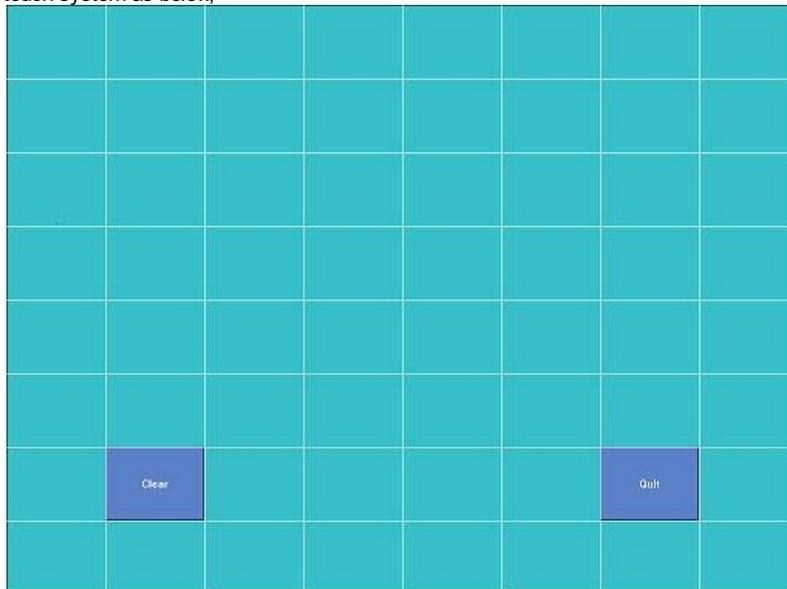
Linearization (25 or 9 points calibration) function is used to compensate the touchscreen linearity. After linearization completed, the linearity of the touchscreen will be shown in the Linearity curve window.

Pressing this button, a new window will be popped-up at the location when the touchscreen was mapped to area for this touch system to guide the user do 25 points calibration. User should follows the guide to touch and hold the blinking symbol in the calibration window until it shows '¡\$OK¡' to make sure that the utility can gather enough data for computation.



5. Draw Test

This function is used for accuracy and performance check. Press this button and a new pop up window will be popped up in the location where the touchscreen was mapped to the touch system as below,



User can press the Clear button to clear the window. Press Quit button to terminate this draw test.

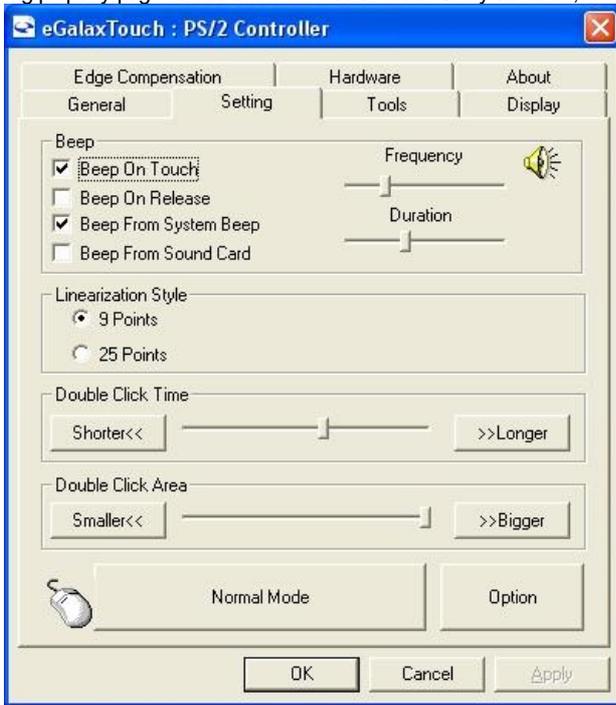
See Also

[Multiple monitor](#)

[Beep, Mouse Emulation Mode](#)

eGalaxTouch Setting Property Page

The Setting property page can be shown in eGalaxTouch Utility as below.



There are function buttons and check boxes described as below

1. Beep

1-1) Beep On Touch

Check this check box to enable driver to generate a beep sound when touch touchscreen state is switched from untouched to touched state.

1-2) Beep On Release

Check this check box to enable driver to generate a beep sound when touchscreen state is switched from touched state to un-touch state.

1-3) Beep From System Beep

Check this check box to make the beep from system speaker.

1-4) Beep From Sound Card

Check this check box to make the beep from sound card.

1-5) Frequency

Adjust this frequency to control the beep sound frequency generated by the driver.

1-6) Duration

Adjust this duration to control the beep sound duration.

2 Linearization Style

eGalaxTouch utility provides user with both 9 points and 25 points calibration for linearization. User can select the suitable kind of linearization type.

3 Double Click Time

The double Click Time group is used to set system double click time. Change this value will affects the double click behavior for all of the mice devices in the system. Two continuous clicks at the same area within this specified time period will be recognized as a double click event.

4 Double Click Area

The double click area group is used to set the system double click area. Change this value will affects the double click behavior for all of the mice devices in the system. Two continuous click with this specified area in the specified double click time will be recognized as a double click event.

5 Mouse Emulation mode

There are 5 mouse emulation modes for eGalaxTouch touchscreen controllers. Press on the button to change the emulation mode,

4-1) Normal Mode

Normal mode behaves mouse button down and mouse move. User can select this mode to select object, and dragging the object.

4-2) Click On Touch

With this Click On Touch mode, the driver emulates a mouse click event when the touchscreen state was switched from un-touched state to touched state. Then, the driver always generate mouse move event and is tracking the touch position until the touchscreen state switched to un-touch state.

4-3) Click On Release

With this Click On Release mode, the driver emulates a mouse click event when the touchscreen state was switched from touched state to un-touched state.

4-4) Click On Touch without moving cursor

With this mode, the driver behaves similar as Click On Touch mode. The cursor does not move to the touch position except the first touch point.

4-4) Click On Release without moving cursor

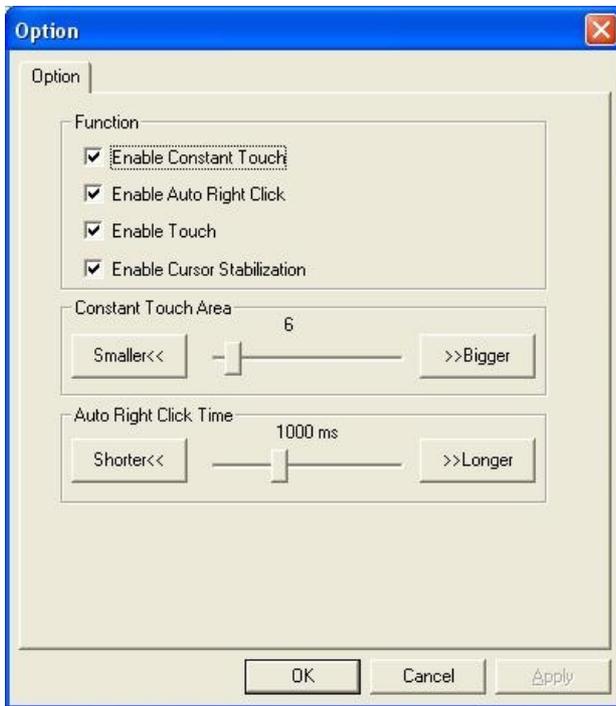
With this mode, the driver behaves similar as Click On Release mode. The cursor does not move to the touch position except the lift-off point.

4-5) Desktop Mode

With this mode, the driver behaves similar as Normal Mode. But the driver will not report mouse button down immediately after user touches down. User needs to touch and stay at one point for a few milli-second, then the driver will report mouse touch down.

6 Option

User can set configuration for some advanced functions with this option button. Press this button, a pop up property sheet window will be popped up and shown as below.



The advanced functions are

5-1) Enable Constant Touch(Hold)

Constant Touch is the function to check if the most recent touched position is same as the previous touched point. If the points difference is smaller than the defined area, the driver does not generate any mouse event to reduce system loading.

Check this check box to enable this function (Hold) and un-check it to disable this function.

5-2) Enable Auto Right Click

If the touchscreen was kept touched for a specified time, the driver will generate a mouse right button click event if this function was enabled.

Check the check box to enable this function and uncheck it to disable this function.

5-3) Enable Touch

The driver read the data input from controller to generate mouse event. However, it can be enabled or disabled to generate the mouse event.

Check this check box to make driver to generate the mouse event when it receives the touch point input from eGalaxTouch touchscreen control and un-check it to stop driver generating the mouse event.

5-4) Enable cursor stabilization

A software filter was implemented inside the driver to filter some noise to stabilize and smooth the touch points. Then, the user can see a more stable cursor.

Check this check box to enable this software filter and un-check it to disable this function.

5-5) Constant Touch Area (Hold area)

Adjust the parameter for Constant Touch(Hold) function. This is a criterion to judge if the most recent touched point is same as the previous touched point. If the points difference is within this area, it will be recognized as the same touch point and the driver does not generate new mouse event for this new touch point.

5-6) Auto Right Click Time

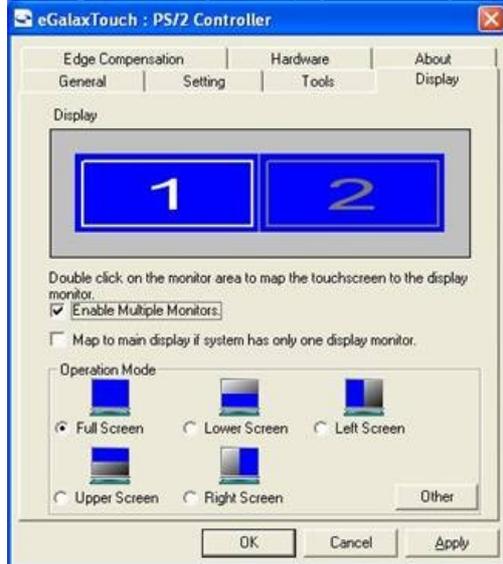
Adjust the Right click time for auto right click function. If the touchscreen was touched and hold for this period of time, the driver generates a mouse right click event.

See Also

- [4 points, linearization/25 points calibration, draw testing](#)
- [Multiple monitor and split display configuration](#)

Monitor Property Page

eGalaxTouch driver utility supports multiple monitor and display system. To work with multiple monitor system, user needs to do configuration and map the touchscreen working area to the system display area correctly. User can do such configuration with the property page shown as below,



Display

The system monitor display geometry is shown in the Monitors window in this page to show the locations of all of the monitors of the system.

Note: [xtouchmon.exe](#) is another utility for eGalaxTouch touchscreen system. This tray icon utility can monitor the system monitor display configuration change and correct the touchscreen monitor mapping relationship automatically as soon as the system monitor display configuration changed. We strongly suggest the user to make sure [xtouchmon.exe](#) is launched for multiple monitor system.

Users can follow below instructions to do the configuration

1. Enable multiple monitor

Check this check box to enable multiple monitor support and uncheck it to disable multiple monitor support. When this function is disabled, the touchscreen will be mapped to the primary monitor automatically.

When this function is enabled, user can double click on the monitor area in the monitor geometry window to assign the monitor area where the touchscreen will be mapped. In other words, the touchscreen will work with the selected monitor. Then, the selected monitor area rectangle line will be changed to white and the other monitor rectangles line will be grey.

2. Map to main monitor when the system has only one monitor

When the multiple monitor function was enabled, and the system has only one monitor,

Driver allows user to generate the mouse event for the primary monitor or not when the touchscreen is not mapped to the primary monitor. Check the check box to enable this function, then, the driver will generate the mouse event for the primary monitor even though the touchscreen is configured as other monitor mapping and multiple monitor function is enabled.

3. Operation Mode

eGalaxTouch driver supports split display mode for those applications which do not map the touchscreen to the full screen of the monitor.

3-1) Full screen

The touchscreen will be mapped to the full screen of the specified monitor.

3-2) Right screen

The touchscreen will be mapped to the half right screen of the specified monitor.

3-3) Left screen

The touchscreen will be mapped to the half left screen of the specified monitor.

3-4) Upper screen

The touchscreen will be mapped to the upper half screen of the specified monitor.

3-5) Lower screen

The touchscreen will be mapped to the lower half screen of the specified monitor.

3-6) Other operation mode

3-6-1) Quarter 1

The touchscreen will be mapped to the first quarter area of the specified monitor display.

3-6-2) Quarter 2

The touchscreen will be mapped to the 2nd quarter area of the specified monitor display.

3-6-3) Quarter 3

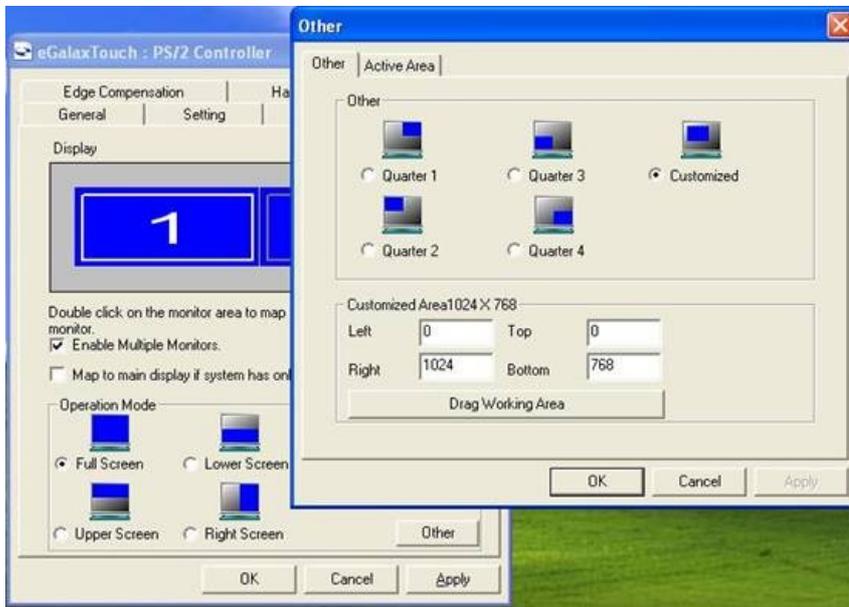
The touchscreen will be mapped to the 3rd quarter area of the specified monitor display.

3-6-4) Quarter 4

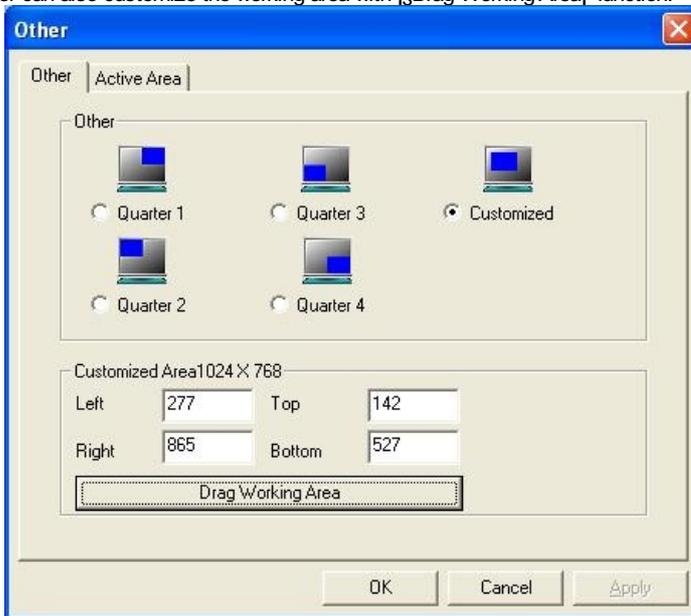
The touchscreen will be mapped to the 4th quarter area of the specified monitor display.

3-6-5) Customized

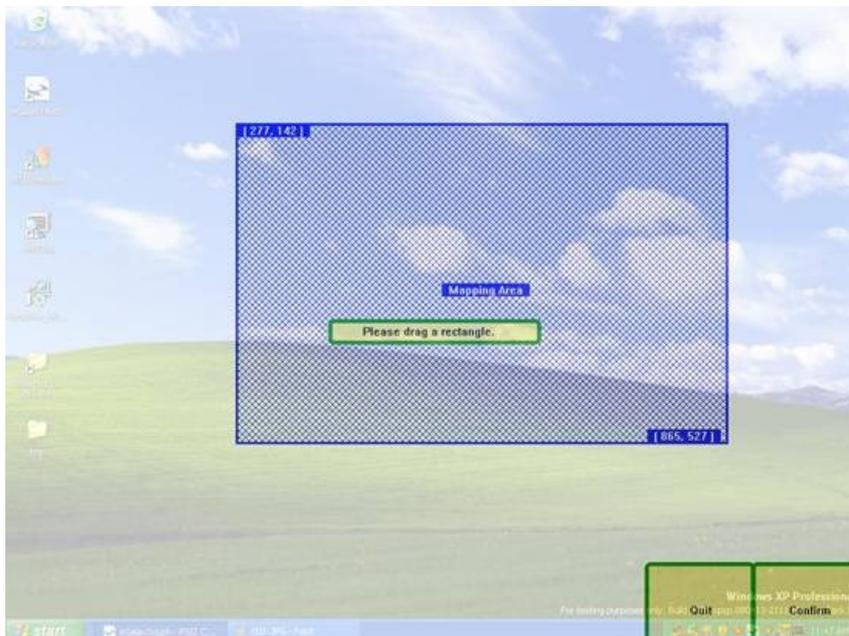
User can type in the range (Left, Right, Top, Bottom) to customize the working area. **With this mode, the driver does not correct the mapping area when the display resolution changed. It needs to do configuration setting again whenever the display resolution changed.**



User can also customize the working area with `i$Drag Working Area` function.



Click the `i$Drag Working Area` button, the image will switch back to the desk top, then user can customize the working area by drawing and dragging the rectangle. Then click `i$confirm` to set the working area. Or `i$Quit` to cancel.



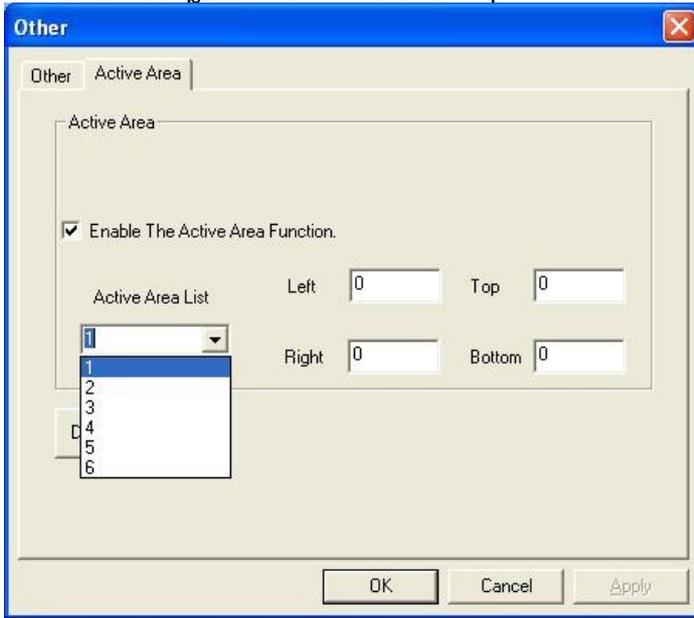
3-7) Active Area

If several areas but not the full screen are needed as working area, eGalaxTouch allows user to customize up to 6 active areas, only in which

touches will be recognized.

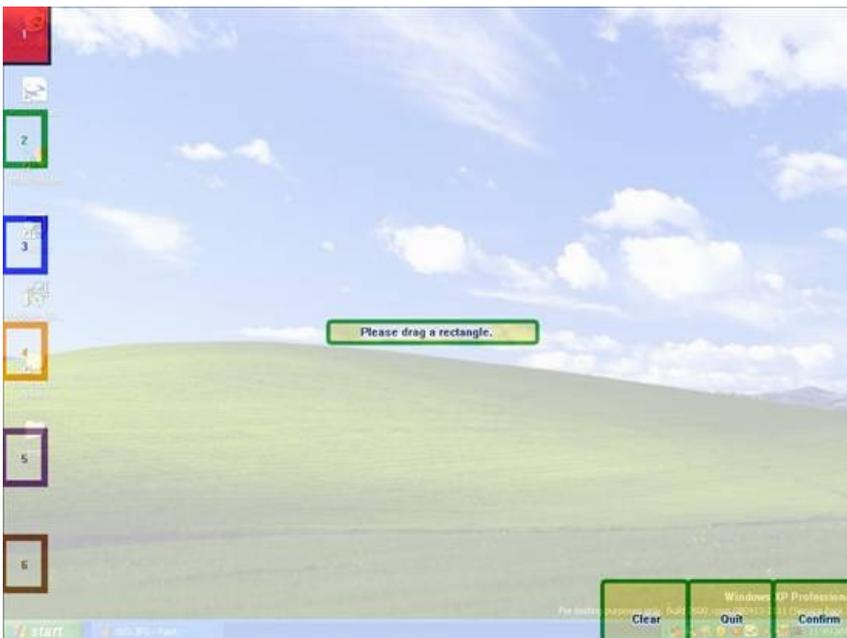
Go to 'Active Area' Tab first, user can type in the range of the preferred active areas or customize the active areas by using 'Drag Active Area' function.

Check the box for 'Enable The Active Area Function'

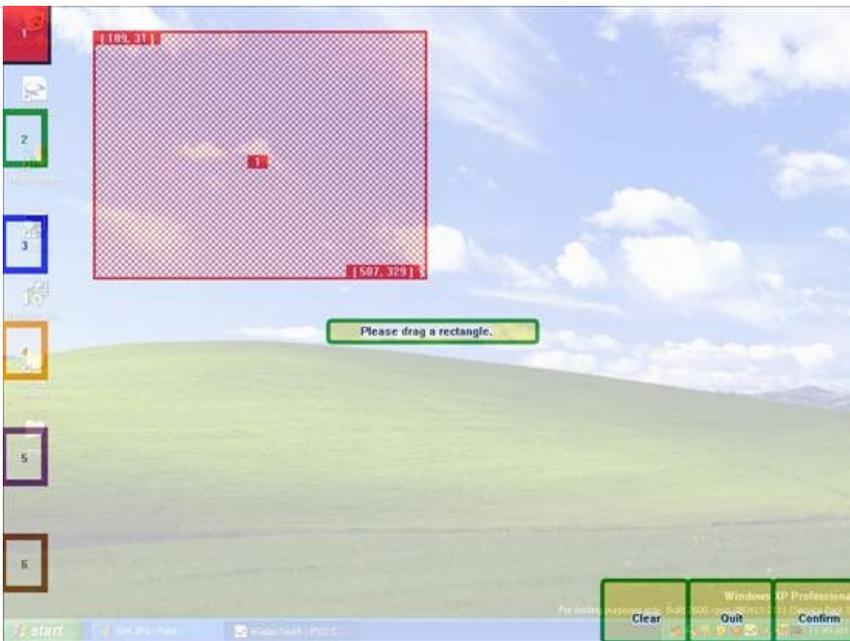


If user prefer to type in the range of the active area, please use the scroll bar to number and choose the active areas.

If user prefer to use 'Drag Active Area' function, please click on the 'Drag Active Area' button. The image will switch to desk top, and there are six box in different colors numbered from 1~6.



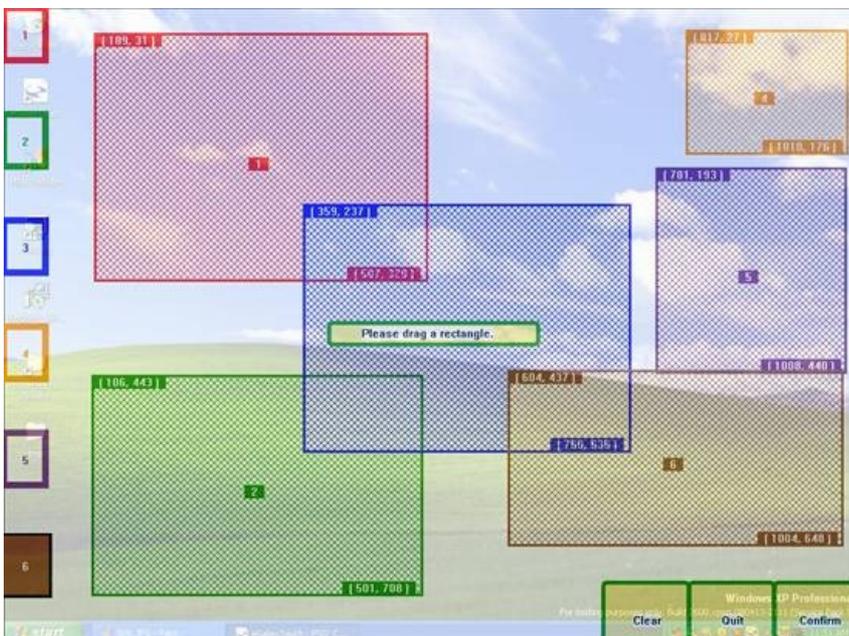
When a box is touched (or clicked), user can draw/drag a rectangle as the customized active area.



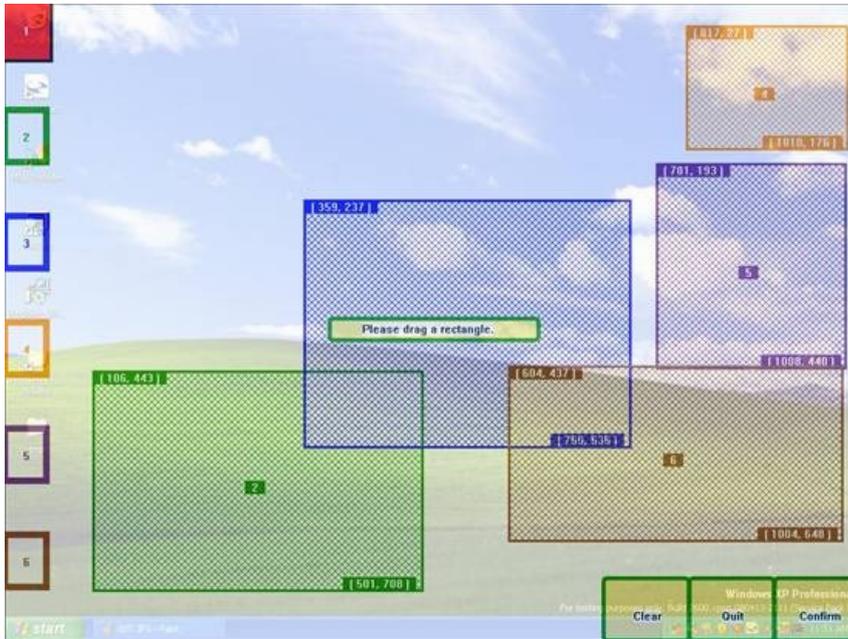
When another box is touched, user can draw/drag another rectangle as another active area.



User can use this function to customize six active areas, these areas can be overlapped.



If user wishes to cancel a particular active area, simply click on the box on the left hand side, and click "Clear".

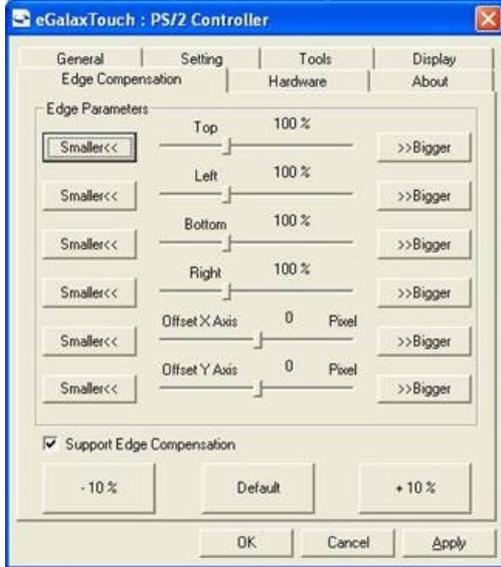


For example, Area 1 is now cleared.
Once everything is done, click "Confirm".
User can "Quit" anytime during customizing the active areas.

See Also
[4 points, linearization/25 points calibration and drawing test](#)
[Beep, Mouse emulation, Double click, Auto right click](#)

eGalaxTouch Edge Compensation Property Page

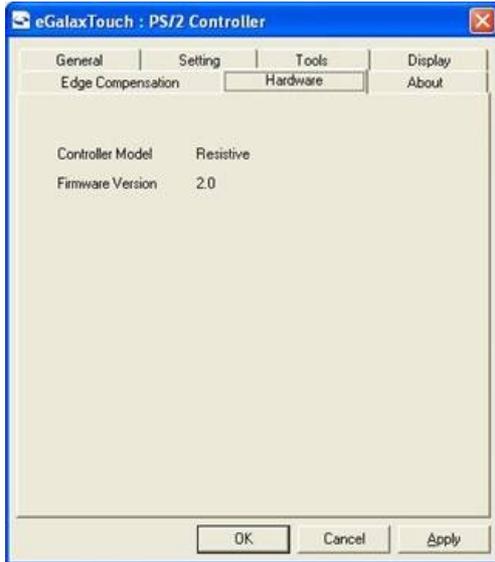
For some special touchscreen which can not reach to the edge area of the full screen, eGalaxTouch utility provides user with edge compensation tool to solve such problem and make it easy to touch the edge area without accuracy lost. Stretch a little bit near the edge area.



User can set the stretch percentage for the 4 edges. In addition, it allows user to set an offset for the touch point for special application.

Hardware Property Page

This hardware property page shows the model and firmware version of the eGalaxTouch touchscreen controller. The software will query the hardware information from controller and show the information as below.



About Property Page

The 'About' tab shows some information for eGalax Touch driver information. User can click Download to download the latest driver from web site.



xTouchmon utility

eGalaxTouch software package contains an utility to monitor some events related to touchscreen application. After eGalaxTouch software package installed, a tray icon utility was shown in the system task bar as below. Whenever the mouse cursor moves on the tray icon, a tool tip shows as



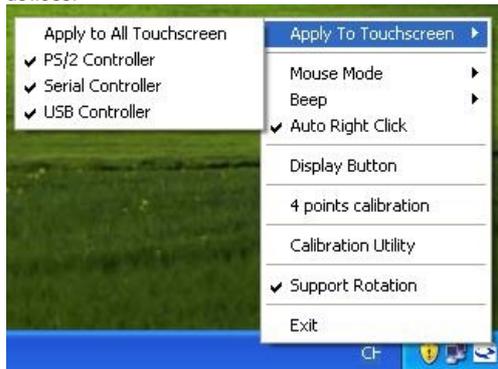
This is a fast tool for user to change some driver setting without launching the eGalaxTouch configuration utility. When user right click on this tray icon, a pop up context menu shows as below,



This pop up menu shows the all of the functions this utility supports.

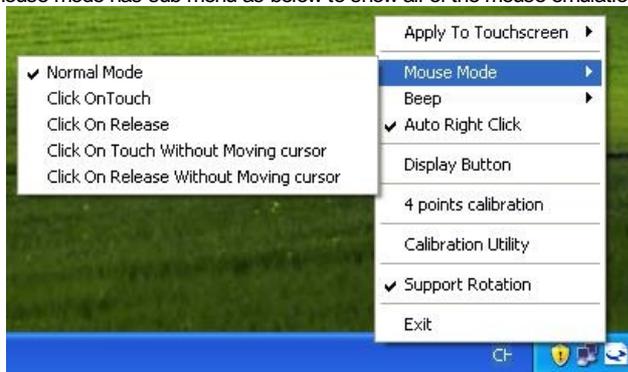
1. Apply To Touchscreen

When user selects this function, the sub-menu shows all of the eGalaxTouch devices installed in the system. User can choose the next function configuration settings will be applied to which devices. xTouchmon allows the configuration settings applied to all of the eGalaxTouch devices or some specified devices.



2. Mouse Mode

This Mouse mode has sub-menu as below to show all of the mouse emulation modes the driver supports.



This [mouse emulation mode](#) changed will be applied to the specified devices.

3. Beep

User can change the [Beep Mode](#) with this function menu. And, the change will be applied to the specified devices immediately.

4. Auto Right Click

User can enable/disable auto right click function with this function menu. The change will be applied immediately to those specified devices.

5. Display Button

This button is designed to show/hide the below right click button window. User can click on this right button before generating a right click event.



6. 4 Points calibration

Select this function to pop up the 4 points calibration window for user to do [calibration](#) for those specified devices.

7. eGalaxTouch Utility

eGalaxTouch configuration utility will be launched if user selects this function.

8. Support Rotation

Monitor display rotation support. Enable it, this utility will correct the touchscreen orientation to match with monitor display orientation whenever it detects monitor rotation. This correction will be done automatically if the system video adapter driver supports Microsoft Win32® APIs for monitor rotation. If the video adapter driver does not support Microsoft Win32® API for monitor rotation, the utility will pop up a one point calibration window to correct the orientation mapping.

9. Exit

Exit function to terminate the xTouchmon daemon process. After this daemon process terminated, all of functions the utility supports will be disabled.

In addition to above functions, xTouchmon also monitors monitor display setting change. It will correct the coordination mapping automatically except the touchscreen work with [customized](#) area mode whenever it detect monitor resolution change, numbers of system monitors change.

Packet Format for eGalaxTouch Touchscreen controller

eGalaxTouch Touchscreen system defines two kinds of packet format for communication between Host and controller device. Both of these two kinds of packet format are valid for all of series of eGalaxTouch touchscreen controllers with RS232, USB, PS2, and I2C interfaces.

Diagnostics Packet

These packets are issued from the host for querying some device information. The controller firmware will report the corresponding data to the host. The packet format is as follows:

0x0A	LengthInByte + 1	Command	Response
1 Byte	1 Byte	1 Byte	LengthInByte

The maximum packet size is 16 bytes. The first byte is Start of Packet as 0X0A. The second byte is the length of Response. The third byte is the issued command and the last part (length is defined in second byte) is the response from controller firmware.

1. Check active : this packet is to check if the device is working properly.

Host issues

0x0A	1	0xA1
------	---	------

Device responds when active

0x0A	1	0xA1
------	---	------

2. Get firmware version

Host issues

0x0A	1	0xD1
------	---	------

Controller firmware responds

0x0A	Length	0xD1	Response
------	--------	------	----------

The response is an ASCII string, such as 0.99

3. Get type

This packet is to request the controller type.

Host issues

0x0A	1	0xE1
------	---	------

Controller firmware responds

0x0A	Length	0xE1	Response
------	--------	------	----------

Report Packet

eGalaxTouch USB HID Touchscreen controllers support Microsoft HID touch digitizer. By default, eGalaxTouch HID compatible controller report with HID format for coordination data according to the HID report descriptor it reported to Host system. In addition, eGalaxTouch serial RS232 controllers support emulation modes.

Serial controller's report format depends on the format of command sets it receives from Host. By default, it reports with non-emulated packet format as below. **To make sure the controllers to report with the below format, the host driver should issue any one of diagnostics packet data to controller.** For example, host driver may send a Check Active (0x0A, 1, 0xA1) packet data to controller to make it report with below report format.

Each report packet may contain 5 or 6 bytes as below:

	MSB				LSB			
Byte0	1	Z	M	0	0	AD1	AD0	Status
Byte1	0	A13	A12	A11	A10	A9	A8	A7
Byte2	0	A6	A5	A4	A3	A2	A1	A0
Byte3	0	B13	B12	B11	B10	B9	B8	B7
Byte4	0	B6	B5	B4	B3	B2	B1	B0
Byte5	0	P6	P5	P4	P3	P2	P1	P0

Byte0: Byte0 is the header of the point packet. It contains below point Information

Z: pressure bit. eGalaxTouch controller SAW technology may report with pressure information.

Z=0 means no pressure information

Z=1 means Byte5 is pressure information.

M: Player ID. eGalaxTouch multiplier controller report player ID information

M=0 means no player ID information

M=1 means Byte5 is player ID

Status: touch down status.

Status = 1 means touch down

Status = 0 means lift off point

Byte1~Byte4:

AD1,AD0: resolution information of the current point coordination.

AD1:AD0 = 0:0 means the coordination resolution is 11 bits

AD1:AD0 = 0:1 means the coordination resolution is 12 bits

AD1:AD0 = 1:0 means the coordination resolution is 13 bits

AD1:AD0 = 1:1 means the coordination resolution is 14 bits

indicates the touch status: 1 for touch down and 0 for touch up.

A10/A11/A12/A13 \setminus A0: 11/12/13/14 bits of 1st direction raw data

B10/B11/B12/B13 \setminus B0: 11/12/13/14 bits of 2nd direction raw data

Please be aware that A and B just represent 2 resolution directions of the touch panel.

Byte5: Pressure or player ID

The point packet has 6th byte only when Z=1 or M=1. Otherwise, the point packet has 5 bytes only. If Z=1, this byte is pressure value. If M=1, this byte is player ID.

See Also

[Communication interface](#)

Communication Interface

eGalaxTouch provides controllers to communicate with Host devices with RS232, USB, PS2, and I2C interfaces.

RS232 Interface

The RS232 interface controllers communicate with Host device with RS232 protocol definition as below

Baud rate: 9600 bps
Data bits: 8 bit
Stop bit: 1 bit
Parity check: NONE

PS/2 Mouse Interface

eGalaxTouch PS2 controller follows the IBM PS2 mouse specification to communicate with Host device.

USB Interface

eGalaxTouch USB controllers communicate with Host with USB Low speed or Full Speed and $\{ \$Vendor\ Specific\ Class\}$.

The VID(Vendor ID) and PID(Product ID) may be one of

- ◆ 1.) VID = 0123, PID=0001
- ◆ 2.) VID = 0EEF, PID=0001
- ◆ 3.) VID = 0EEF, PID=0002

- **Two end points are used:**

- ◆ **Control pipe:**

for standard USB PnP and writing packets to controller device controller.

- ◆ **Interrupt or Bulk IN pipe:**

for reading packet from controller device.

- **It needs to wait for 3ms at least to issue another write command after one write command is issued**
- **The polling interval of reading with interrupt pipe is 5 ms**
- **Control Write Urb format is as follows**

```

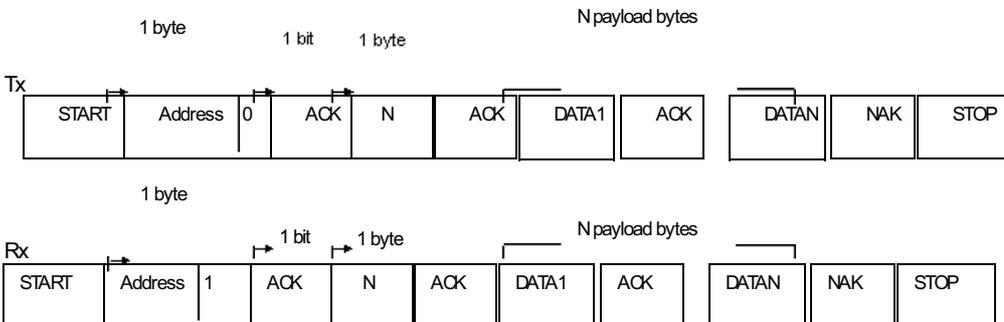
/*=====*/
/* Vendor specific request Urb format for Touch Panel controller kit */
/* with Win2000 DDK */
/*=====*/

UsbBuildVendorRequest( pWriteUrb, //IN PURB Urb,
    URB_FUNCTION_VENDOR_DEVICE, //IN USHORT Function,
    sizeof( struct _URB_CONTROL_VENDOR_OR_CLASS_REQUEST ), //IN USHORT Length,
    0, //IN ULONG TransferFlags,
    0, //IN UCHAR ReservedBits,
    0, //IN UCHAR Request,
    0, //IN USHORT Value,
    0, //IN USHORT Index,
    pTxBuffer, //IN P/VOID TransferBuffer OPTIONAL,
    NULL, //IN P/MDL TransferBufferMDL OPTIONAL,
    ulBytesToSend, //IN ULONG TransferBufferLength,
    NULL, //IN PURB Link OPTIONAL,
);
    
```

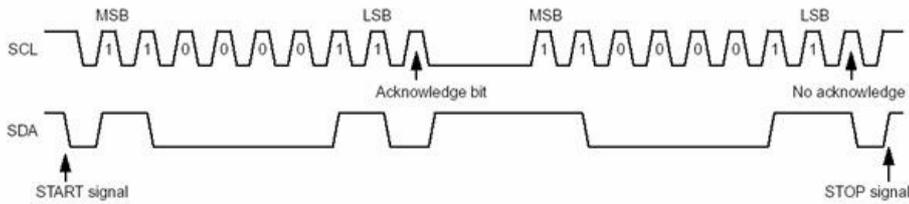
I²C Interface

- Baud rate: less than 12.5 kbps
- I²C transaction package: (Packet) = Bytes of transfer + Payload
Where the payload is packet for touch screen feature.
- Default address: 04
- Software needs to reassemble the payload as the eGalaxTouch packet.

Example: Bytes of transaction = N



- **I²C Timing:**



Packet Parser Sample Code

```

#define MAX_BUFFER          1024
#define MOUSE_PACKET_LEN   5
#define MAX_CMD_LEN        16
#define POLLING_BUFFER_SIZE 3

unsigned __stdcall PortThreadRoutine( LPVOID pContext )
{
    CPort *pPort = ( CPort* ) pContext;
    CHAR    pBuffer[ MAX_BUFFER ];
    CHAR    pMsgBuffer[ MAX_BUFFER ];
    DWORD   dwRead = 0;
    DWORD   dwCnts = 0;
    BOOL    bPointPacket = FALSE ;
    BOOL    bCmdPacket = FALSE;
    DWORD   dwCmdPacketLen;
    UCHAR   ucChar;
    INT     i;

    while( TRUE )
    {
        if( WAIT_OBJECT_0 == ::WaitForSingleObject( pPort->m_hStopEvent, 0 ) )
        {
            return 100;
        }
        // read packet from COM port or USB port
        if ( pPort->Read( pBuffer, POLLING_BUFFER_SIZE, &dwRead, pPort->m_hReadEvent ) )
        {
            // parse the packet
            for( i = 0; i < (INT)dwRead; i++ )
            {
                ucChar = pBuffer[ i ];
                if( ( pBuffer[ i ] & 0xF0 ) == _SYNCBIT ) && !bCmdPacket )
                {
                    dwCnts = 0;
                    pMsgBuffer[ dwCnts ] = pBuffer[ i ];
                    bPointPacket = TRUE;
                    dwCnts++;
                    continue;
                }
                else if( _SOP == ucChar && !bPointPacket && !bCmdPacket )
                {
                    bCmdPacket = TRUE;
                    dwCmdPacketLen = ( DWORD )-1;
                    bPointPacket = FALSE;
                    continue;
                }
                else if( bCmdPacket )
                {
                    if( ( DWORD )-1 == dwCmdPacketLen )
                    {
                        dwCmdPacketLen = ( DWORD )pBuffer[ i ];
                        dwCnts = 0;
                        if( dwCmdPacketLen > MAX_CMD_LEN )
                            dwCmdPacketLen = MAX_CMD_LEN;
                        continue;
                    }
                    pMsgBuffer[ dwCnts ] = pBuffer[ i ];
                    dwCnts++;
                    if( dwCmdPacketLen == dwCnts )
                    {
                        dwCmdPacketLen = 0;
                        pMsgBuffer[ dwCnts ] = 0;
                        dwCnts++;
                        // Here, a completely Cmd packet received !!!
                        // Do what you want to do!
                        // For instance,
                        // pPort->DispatchMessage( pMsgBuffer, dwCnts );
                        dwCnts = 0;
                        bCmdPacket = FALSE;
                        continue;
                    }
                }
            }
            continue;
        }
    }
}

```

```
}
if( bPointPacket)
{
    pMsgBuffer[ dwCnts ] = pBuffer[ i ];
    dwCnts++;
    if( MOUSE_PACKET_LEN == dwCnts )
    {
        // Here, a completely point packet received !!!
        // Do what you want to do!
        // For instance,
        //pPort->DisPachMessage( pMsgBuffer, dwCnts );
        dwCnts = 0;
        bPointPacket = FALSE;
    }
    continue;
}
}
}
}
```

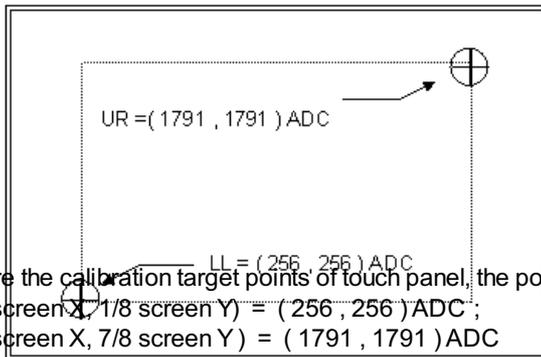
See Also

[Data Packet format](#)

2 Points Calibration for Position Decoding

(0, 2047) ADC

(2047, 2047) ADC



- LL and UR are the calibration target points of touch panel, the points are setup at
 $LL = (1/8 \text{ screen } X, 1/8 \text{ screen } Y) = (256, 256) \text{ ADC}$;
 $UR = (7/8 \text{ screen } X, 7/8 \text{ screen } Y) = (1791, 1791) \text{ ADC}$
- During calibration, press on these two target points, the raw data are obtained as LL_i and UR_i :
 $LL_i = (LL_x, LL_y)$;
 $UR_i = (UR_x, UR_y)$
- After the calibration, whenever the panel was touched, firmware report the raw data X and Y. Then, the calibrated position X_i and Y_i are calculated with the formulation as follows:

X_i

$$X_i = \frac{X - LL_x}{UR_x - LL_x} \times 1536 + 256$$

$$Y_i = \frac{Y - LL_y}{UR_y - LL_y} \times 1536 + 256$$

Interface to eGalaxTouch Device Driver.

(This Document is for eGalaxTouch driver Version 4.0 or later for Microsoft® Windows® 2000/ Windows® XP / Windows Vista® / Windows® 7 operating system only.)

eGalaxTouch allows user to communicate with the device driver. User can either use eGalaxTouch API or Win32® APIs such as **CreateFile**, **CloseHandle**, **ReadFile**, **WriteFile**, **DeviceIoControl** to communicate with device driver.

1. Using eGalaxTouch API

eGalaxTouch API implements a set of functions with Win32® API for user to interact with eGalaxTouch device driver easily. The system must have MFC42.dll support to work with eGalaxTouch API.

Program designer call [EnumerateTouchInstance](#) to enumerate all of eGalaxTouch devices installed in the system. And then, open the device with [CreateDevice](#) to work with the device driver. [StartDeviceThread](#) to start reading data comes from device. Pls make sure that no other process or thread is reading data the device with **ReadFile** so that the API can read correct and complete data packet from driver. In addition, designer can call other API functions to configure driver behavior.

2. Win32® API

eGalaxTouch device driver export the device interface for its device objects. User can enumerate all of these devices with the **GUID = { 0xf771fb3b, 0x7de, 0x4f40, { 0xbb, 0xdf, 0xf5, 0xda, 0x97, 0x71, 0x65, 0x5d } }**.

In addition, the program designer can open the enumerated device with **CreateFile**. Then,

It can read data from device with **ReadFile**, write data to device with **WriteFile**, and configure device driver behavior with [DeviceIoControl](#).

Example : enumerate all eGalaxTouch devices.

```
guidDeviceInterface = { 0xf771fb3b, 0x7de, 0x4f40, { 0xbb, 0xdf, 0xf5, 0xda, 0x97, 0x71, 0x65, 0x5d } };
SP_DEVICE_INTERFACE_DATA InterfaceData;
PSP_DEVICE_INTERFACE_DETAIL_DATA pIfDetailData;
DWORD dwIndex = 0;
HDEVINFO hDevInfo = SetupDiGetClassDevs(
    &guidDeviceInterface,
    NULL,
    (DIGCF_DEVICEINTERFACE | DIGCF_PRESENT));
hWhd, //INHWND hwndParent, PTIONAL

if( INVALID_HANDLE_VALUE == hDevInfo )
{
    return 0;
}
InterfaceData.cbSize = sizeof( SP_DEVICE_INTERFACE_DATA );
while( SetupDiEnumDeviceInterfaces( hDevInfo,
    NULL,
    &guidDeviceInterface,
    dwIndex,
    &InterfaceData ) )
{
    SP_DEVINFO_DATA did;
    DWORD needed;
    dwIndex++;
    SetupDiGetDeviceInterfaceDetail( hDevInfo,
        &InterfaceData,
        NULL,
        0,
        &needed,
        NULL);
    pIfDetailData = ( PSP_DEVICE_INTERFACE_DETAIL_DATA )malloc( needed );
    if( NULL == pIfDetailData )
    {
        break;
    }
    pIfDetailData->cbSize = sizeof( SP_DEVICE_INTERFACE_DETAIL_DATA );
    did.cbSize = sizeof( SP_DEVINFO_DATA );
    SetupDiGetDeviceInterfaceDetail( hDevInfo,
        &InterfaceData,
        pIfDetailData,
        needed,
        NULL,
        &did );
    // Note
    // pIfDetailData->DevicePath is the symbolic link name of the device object.
    // Program designer has to save this symbolic name and append ;$#EDO; to it to
    // open the device with Win32®API CreateFile.
    //
    // Do something here!
    //
    free( pIfDetailData );
}
SetupDiDestroyDeviceInfoList( hDevInfo );
return dwIndex;
```

eGalaxTouch API for Windows 2000/XP/Vista/7

eGalaxTouch driver package contains a dynamic Link Lib(xtkutility.dll) which exports some APIs for special application. User can communicate with eGalaxTouch driver by way of this APIs. The exported APIs were list as below

[CloseDevice](#)
[CreateDevice](#)
[DoCalibration](#)
[DoDrawTest](#)
[DoLinearization](#)
[EnableAutoRightButton](#)
[EnableConstTouch](#)
[EnableTouch](#)
[EnumerateTouchInstance](#)
[GetBranchInfo](#)
[GetCallInfo](#)
[GetConstTouchInfo](#)
[GetEdgeInfo](#)
[GetLinzInfo](#)
[GetMonitorInfo32](#)
[GetMouseMode](#)
[GetTouchBeepInfo](#)
[RegisterDeviceRemovalHook](#)
[RegisterTPNotifyMessage](#)
[RestartFirmware](#)
[SetBranchFormat](#)
[SetCallInfo](#)
[SetConstTouchInfo](#)
[SetEdgeInfo](#)
[SetConstTouchInfo](#)
[SetLinzInfo](#)
[SetMonitorInfo32](#)
[SetMouseMode](#)
[SetTouchBeepInfo](#)
[StartDeviceThread](#)
[StopDeviceThread](#)
[UpdateControllerCallInfo](#)
[UpdateControllerLinzInfo](#)
[VerifyEEPROM](#)