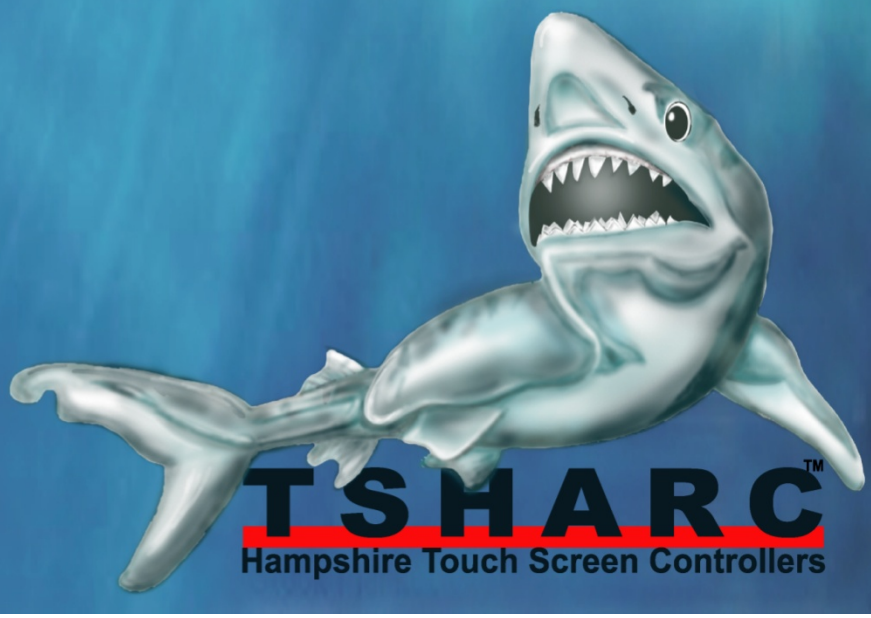
TSHARC™ Linux Driver 3.0.4
Setup and Users Manual
Linux

Document Revision and Copyright
Document Name: LinuxDriverUserManual_3.0.4_071001.docx
Document Date: 10/1/2007 2:35:00 PM

Contact Information
Hampshire Company, Inc.
9055 N. 51st Street Unit H
Brown Deer, WI 53223
Main Phone: 414-355-4675
Main Fax: 414-355-4775
www.hampshirecompany.com

Features
ARM Support
5pt, 9pt, 25pt, 28pt Calibration Support
X11R7 (Support
User-Mode Calibration Support
Text-Mode Control Panel
Autodetect and Configure of TSHARC PS2 Components
Minimal Configuration After Installation
Uninstall Script Provided
EEPROM Reading and Writing
Full Capacitive Controller Support



Important Documents and User Information

Use this document as a guide only. The information contained in this document is subject to change without notice. This document, and all of its contents, is protected under International and U.S. copyright law. Hampshire Company trademarks, patents and copyrights are the exclusive property of Hampshire Company, Inc and may not be used without the express written consent of Hampshire Company, Inc.

Due to the vast range of application variables that fall outside of Hampshire's expertise and control, Hampshire assumes no responsibility for the usability or the suitability of Hampshire products in customer application(s). Hampshire assumes those responsible for the application and use of Hampshire Company products and documentation have taken all necessary steps to insure that the application of Hampshire products meet any and all safety and performance requirements including any laws, regulations, codes and standards associated with the application. In no case does Hampshire warrant the usability or suitability of its products in any medical, aviation, military or other life critical applications. User should contact Hampshire Company before integrating any Hampshire hardware or software product into these types of applications.

All Hampshire standard software products, including but not limited to device driver software, are provided "as is" and may be used exclusively with an authentic Hampshire TSHARC touch screen controller. Hampshire does not warrantee or guarantee the usability of any of its software products.

Hampshire Company warrants to the original purchaser, or a third party who is listed on the Hampshire "Multi Party Authorization Agreement", that for the warranty period identified by Hampshire warranty for the specific product that you have purchased, that the goods will be in conformance with Hampshire specifications. In addition to Hampshire specifications, the warrantee includes Non-Hampshire specifications agreed-to, in writing, by Hampshire in advance for the limited purpose between the customer and Hampshire Company. The warrantee also insures that the goods will be free from defects in design and material for the warrantee period. Hampshire Company, at its option, will repair or replace the nonconforming or defective goods, issue a credit memorandum, or refund the purchase price, as its sole obligation: provided that the nonconforming or defective goods are rejected in writing (including a reasonable failure analysis statement) to Hampshire Company and returned to Hampshire Company within the relevant warranty period and provided that the nonconformance or defect was not caused by misuse, alteration, accident, or abnormal conditions of storage, operation or handling, as determined by Hampshire Company.

THIS WARRANTY IS YOUR SOLE AND EXCLUSIVE REMEDY AND IS IN LIEU OF ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE, FURTHER, HAMPSHIRE COMPANY SHALL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGE OR LOSSES, INCLUDING LOSS OF PROFITS, BUSINESS INTERRUPTION, OR OTHER PECUNIARY LOSS, WHETHER ARISING FROM BREACH OF WARRANTY OR BASED ON CONTRACT, FUNDAMENTAL BREACH OR ANY OTHER THEORY. SINCE SOME JURISDICTIONS DO NOT ALLOW LIMITATIONS OF THE TERMS OF ANY IMPLIED WARRANTY, OR EXCLUSION OR LIMITATION OF INCIDENTAL OF CONSEQUENTIAL DAMAGES, THE LIMITATIONS AND EXCLUSIONS OF THIS WARRANTY MAY NOT APPLY TO YOU.

General Information

Mailing address:
Hampshire Company, Inc.
9055 N 51st Street
Brown Deer, WI 53223
Phone: 414-355-4675
Fax: 414-355-4775

World Wide Web:
www.hampshirecompany.com

Support E-Mail Address:
support@hampshirecompany.com

Copyright Information

Edition Information
Printed on: October 1, 2007
Document Part number: LinuxDriverUserManual_3.0.4_071001.docx

General Installation Notes

Hampshire drivers use the display-driver-software settings to configure various touch screen driver setup files. Install the display and display driver cards properly before installing any TSHARC touch screen controller drivers. If the display is not configured and working properly prior to loading the TSHARC controller drivers, the TSHARC controllers will not function properly.

While Hampshire goes to great lengths to insure that all of the controllers and drivers will provide the highest possible performance and will even improve the performance and extend the life of a poor quality or failing touch screen, the overall accuracy and stability of the calibration will be dependent on the quality of the touch screen. Linearity, sheet resistance, contact resistance, tail assembly, capacitance and the printed silver linearization pattern vary between touch screen manufacturers, construction, assembly and technology. For more information regarding touch screen constructions and types, contact Hampshire Company.

Hampshire controllers are universal by design. Configuration of the controller is available for any number of touch screen types as well as communication and power settings. Please check the TSHARC controller board to insure that the user-configurable configuration settings are set correctly for the desired application. It is essential that the system used is set-up properly. Please review this manual to insure the system is ready. **The users' manual for each Hampshire controller board is available at www.hampshirecompany.com.** Once at the website, select the "Products" tab along the left side of the window and navigate to the corresponding TSHARC controller product. Once the system is verified to be working properly, proceed with the TSHARC controller driver installation procedure.

Features and Changes

Features and changes to this driver include:

- Serial, USB, and PS2 support
- Capacitive support
- Edge acceleration
- Right click emulation
- Touch modes – Normal, Pendown, Penup
- EEPROM read and write
- 4, 5, 9, 25, and 28 point calibrations
- Configurable primary and alternate buttons
- Multi-display support
- Multi-screen support
- No script command dependencies
- Working daemon on systems without X-Windows
- ARM support
- "Cpconsole" is now named "hlincal"
- Tinyx / Kdrive Support
- X11R7 (Xorg 7.x) support
- X11R6 support
- No reboot required
- Text-mode Calibration Support
- Setup Interface for installing driver components
- Autodetect and configure of TSHARC PS2 components
- Autodetect USB controllers upon start of daemon
- Uninstall script provided
- Reduced library dependencies

Up to 3 controllers can be operating on the same system concurrently.

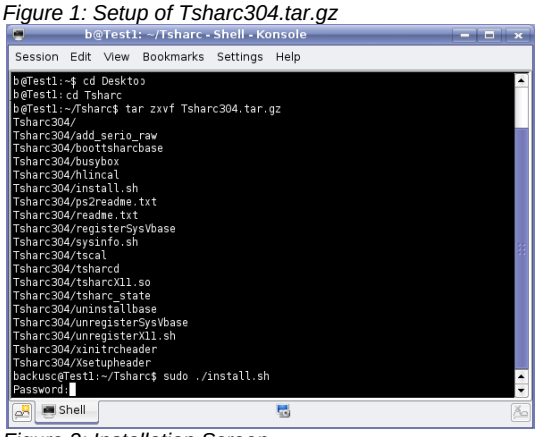
Since there is a wide variance among Linux distributions, there may be issues with the daemon automatically starting after a reboot for some distributions.

Setup

To uncompress Tsharc304.tar.gz into any directory on the local system, first navigate the terminal to the directory containing Tsharc304.tar.gz.

Enter “tar zxvf Tsharc304.tar.gz”. A list of the uncompressed files will be displayed. A new file folder containing these files will be in the directory.

Figure 1: Setup of Tsharc304.tar.gz

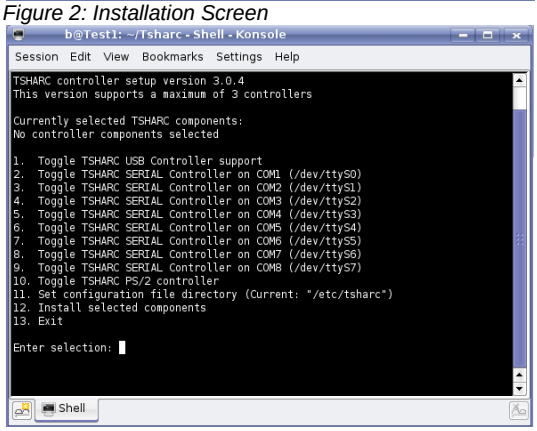


Installation Options

Once Tsharc304 is uncompressed, run the command “sudo ./install.sh” at the console.

Select up to three controller components to install from the listed menu. Make a selection by entering in the corresponding number and pressing “Enter”. Each additional component will take a moment to add to the list to be installed.

Figure 2: Installation Screen



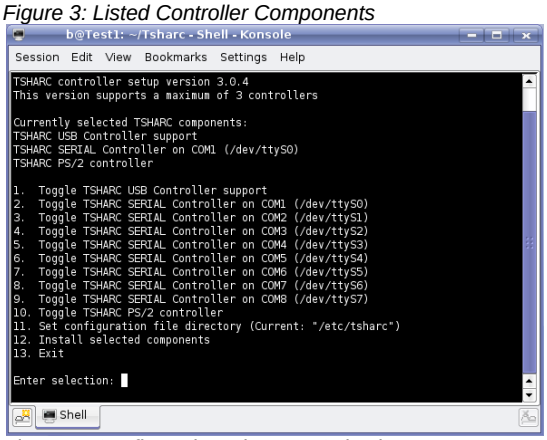
Controller Selection

Select the install menu choice.

It is recommended that all controllers be connected before install, but it is not required.

USB controllers will function, but may not be calibrated depending on which controller is being used. The daemon will have to be started before serial controllers will function. This is shown in Fig. 4.

Figure 3: Listed Controller Components

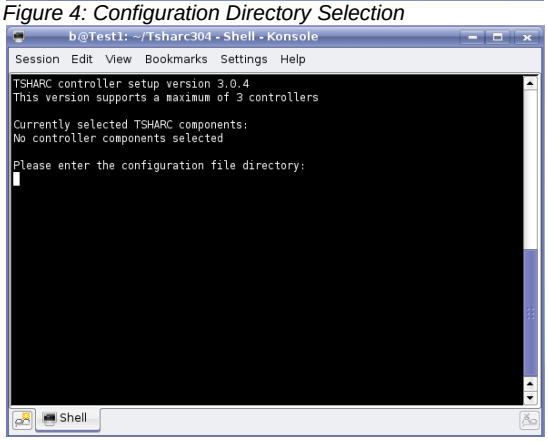


Configuration File Directory Selection

In this screen, the configuration directory can be modified.

Enter the desired directory name and press “Enter”.

Figure 4: Configuration Directory Selection



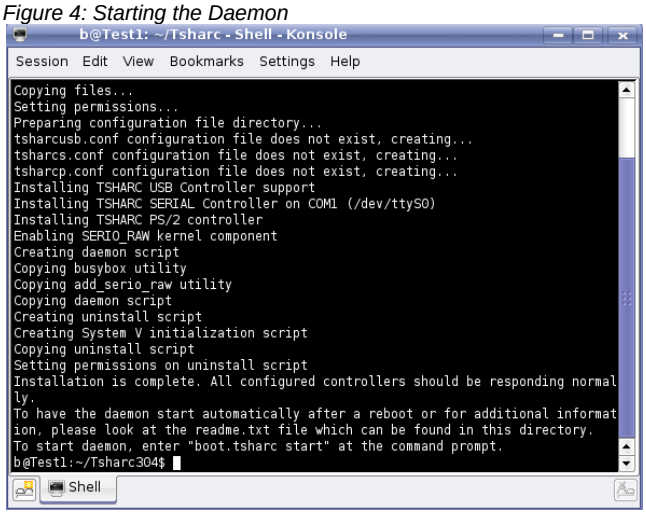
Starting the Daemon

The daemon must be started before the connected serial controllers will function correctly.

Start the daemon by entering “boot.tsharc start” in the command prompt window. To configure the daemon to launch automatically, refer to the “Configuring Daemon to Automatically Start” section of this manual.

All of the configured controllers should now be responding normally to the touch. The touch screen may not be correctly calibrated at this point.

Figure 4: Starting the Daemon



Configuring Daemon to Automatically Start

There are two daemon registration scripts included with the TSHARC Linux daemon, ‘registerSysVexample’ and ‘registerX11example.sh’. Both scripts try to ensure that the command “xhost local:” script command “boot.tsharc start” is executed during startup. “Xhost local” ensures the system is allowed to move the Xserver mouse cursor. “Boot.tsharc start” starts the daemon.

The “registerSysVexample.sh” script attempts to register the daemon with the target system that utilizes System V initialization. There are two main types of initialization normally found on Linux distributions:

- System V
- BSD

It is only necessary to use the script if the TSHARC control panel is to be run without X-Windows. **When the daemon is started at this stage, X-Windows communications become disabled.** To enable X-Windows communications, run the script command “boot.tsharc start” again after X-Windows has started. If X-Windows is installed on the target system, then the “registerx11example.sh” script should be executed in addition to executing “registerSysVexample.sh” so X-Windows communications is enabled when X-Windows starts.

The “registerx11example.sh” script attempts to register the daemon with the x11 startup scripts on the target system. Because x11 is generally the default environment for using a touch screen controller, the “registerx11example.sh” is typically the only script that needs to be executed.

- To register the daemon to use System V initialization, run “sudo ./registerSysVexample.sh”.
- To register the daemon to use x11 startup scripts, run “sudo ./registerx11example.sh”.

For information on how to unregister the daemon, refer to the “How to Uninstall” section of this manual.

To start the daemon manually, enter “boot.tsharc start” at the command prompt. The file “./registerSysVexample.sh” will not be present in the installation directory until after running the script “./uninstall.sh”.

The files and directories that are required to be present for proper operation of the TSHARC daemon are:

- The files contained within the TSHARC configuration directory. These files are listed under “Installation File Descriptions”.
- The “/usr/bin” directory and the files “tscal” and “hlincal”
- The “/var/log/tsharc” directory
  - The file “tsharc.log” within directory
- The “/usr/lib” library
  - The file “tsharcX11.so” within directory

Installation File Descriptions

After installation is complete, the following files will be copied to the installation directory.

- add_serio_raw
- boot.tsharc
- busybox
- tsharcs.conf
- tsharcp.conf
- tsharcusb.conf
- uninstall.sh

<configuration directory>/add_serio_raw

This file is required only if a PS2 controller is implemented. This file ensures that the serio_raw Kernel module is loaded to allow direct data access from the PS2 port.

<configuration directory>/boot.tsharc

The script controls the status of the daemon. Currently, the supported parameters are:

- "start" – starts daemon
- "stop" – stops daemon
- "restart" – restarts daemon
- "status" – Reports if the daemon is running or not

<configuration directory>/busybox

Busybox is an open source project (http://www.busybox.net) that is a tool that combines many Linux utilities and commands into a single file. It is used by boot.tsharc, install.sh, and uninstall.sh scripts to ensure that no additional Linux commands are required by the target system and that the Hampshire scripts remain compatible across Linux distributions.

<configuration directory>/tsharcp.conf

This file is not currently implemented.

<configuration directory>/tsharcs.conf

This file is read by the TSHARC daemon to determine which device paths to read from. If the file were to contain the following lines:

/dev/ttyS1
/dev/tsharcps2

Then the daemon would create the files "ttyS1.conf" and tsharcps2.conf" to be used to save controller settings and to attempt to read and decode bytes from the COM2 and PS2 ports.

<configuration directory>/uninstall.sh

This file will uninstall all the TSHARC daemon files setup on the target system.

/usr/bin/tsharcd

This is the main daemon executable that communicated with the TSHARC controllers set up on the target system. Typically, this is started by using the "boot.tsharc" script.

/usr/bin/hlincal

This is the executable for the text-based control panel.

/usr/bin/tscal

The text-based control panel uses this file to run a GUI calibration using X-windows on the target system.

/usr/bin/uninstall.sh

/usr/bin/boot.tsharc

These files are symbolic links to the files in the <configuration directory>. These files assure the scripts are in the path and can be run from any directory.

/usr/lib/tsharcX11.so

This library file is used by the daemon to communicate with X-windows.

/tmp/tsharcctl

/tmp/tsharcoutput

/tmp/thsarcevents

These files are special files used by the TSHARC daemon as a named pipe. If these files do not exist, the TSHARC daemon will create them. If creating these files manually, use the "mkfifo" command. **Do not use the "touch" command to create these files**.

After the command "boot.tsharc start" is executed, the boot.tsharc script checks if the X-windows environment is running. If it is running, the script enables X-windows functionality for all serial and PS2 controllers installed and all detected USB controllers. To have X-windows functioning means the TSHARC daemon will run the cursor by sending events to X-windows. The tsharcX11.so library file is responsible for the communication with X-windows. If the X-windows environment is not running, the tsharcX11.so file is not used until all the TSHARC controllers are X-windows enabled for functionality.

If the daemon is started without X-windows running, but X-windows is then started manually, the X-windows functionality for TSHARC controllers may be enabled by running "boot.tsharc start" again at the command prompt once X-windows is started.

Additional Installation Notes for PS2 Kernel Configuration

If the TSHARC Installation script gives a "**serio_raw component not found**" error, please follow this procedure.

In all other cases, this procedure should not be necessary.

PS2 Communication System Requirements

- Kernel version 2.4.x or
- Kernel version 2.6.12 or above

Including Required Components with Kernel for PS2 Communication

1. Download the Kernel source for the current Kernel version. Typically, any Kernel version 2.6.9 or newer will be acceptable. If possible, use the Kernel source that comes with the distribution.
2. Log into the system as root or enter "su" at the command prompt.
3. Extract the source code into a directory under "/usr/src".
4. Change the current directory to the directory the Kernel source code was extracted to.
5. Enter "make menuconfig" at the command prompt, select the "Provide legacy /dev/psaux device" and "Raw access to serio ports" items found under "Device Driver > Input device support". This location in the menu varies between Kernel versions.

It is recommended that an asterisk (*) be placed to the left of SERIO to compile it into the Kernel and an "M" to the left of SERIO_RAW to install it as a module.

7. Exit and save the current configuration
8. Enter "Make" at the command prompt
9. Enter "Make modules_install" at the command prompt.

If the Kernel source is for the current version of the Kernel, and SERIO_RAW was selected as a module, then the last step can be omitted

10. Enter "make install" at the command prompt.

The "make install" typically will automatically add a grub, or lilo, menu item on most new Kernel versions. If this does not modify the grub menu configuration, the "menu.lst", "lilo.conf", or "grub.conf" file may need to be modified.

Using the Control Panel

The general process for setting up a controller on the target system follows these steps. This process must be repeated for every controller used on the system.

1. Start the control panel
2. Select a controller to configure
3. Calibrate the current controller
4. Set Edge Acceleration(Optional)
5. Set Click Settings(Optional)
6. Set Touch Settings(Optional)
7. Set Advanced Settings(Optional)
8. Write Current Controller Data
9. Exit Control Panel

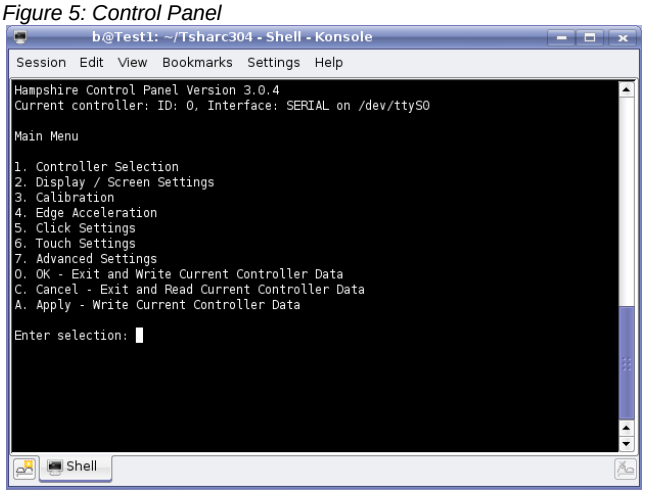
Start the Control Panel

To run the control panel, enter “hlincal” in the command prompt console. Enter in the numbered choice and press “Enter” to continue. **The daemon must be started in order for “hlincal” to run.**

A menu of options will appear, as shown in Fig. 5.

- “Controller Selection” allows the user to configure each controller individually on the system.
- “Display / Screen Settings” selects a monitor on a multi-monitor system.
- “Calibration” runs the calibration/linearization routine
- “Edge Acceleration” modifies the edge acceleration settings.
- “Click Settings” modifies the click options of the touch screen.
- “Advanced Settings” includes EEPROM reading and writing, X and Y offset, and custom width and height.
- “OK” exits the control panel **and** writes the configuration data to the controller.
- “Cancel” exits the control panel and reads the original configuration data from the controller.
- “Apply” will apply the current configurations to the controller, but will not write this data to the EEPROM.

Figure 5: Control Panel



Controller Selection

If using multiple controllers on a system, each controller must be configured individually. Use this screen to choose between available connected controllers.

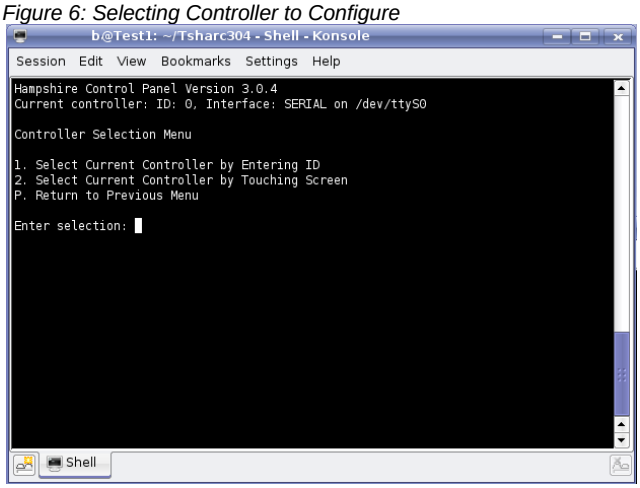
At the top of the screen, shown in Fig. 6, the current controller information, version number, and the controller’s device path are listed.

If multiple controllers are to be configured, there are two ways to select the current controller.

- By entering the ID
- By touching the touch screen

If no other controllers are connected to the system, enter “P” and press “Enter” to return to the Control Panel options.

Figure 6: Selecting Controller to Configure



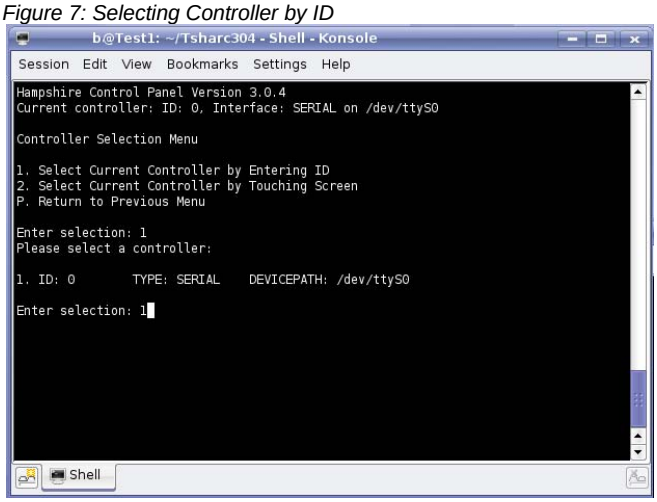
Controller Selection – By Entering ID

After selecting this option, a list of controllers connected to the system will be available, as shown in Fig. 7. Fig. 7 shows only one controller connected to the system.

Enter the corresponding number for the desired controller. Press “Enter”.

After selecting the controller, the current controller’s information will be displayed on the screen shown in Fig. 6.

Figure 7: Selecting Controller by ID



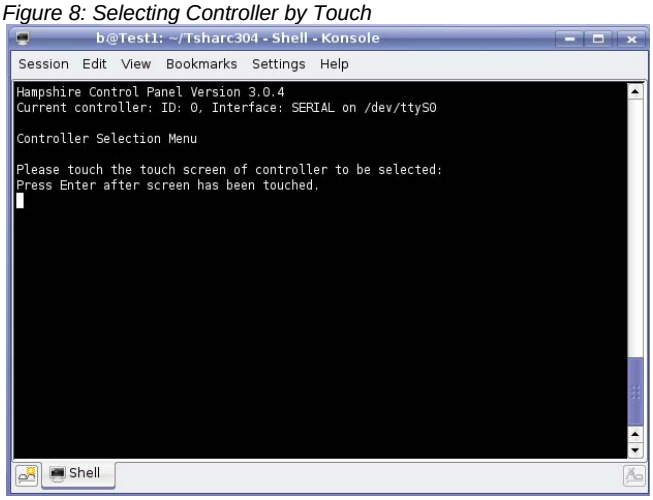
Controller Selection – By Touch

A prompt will appear, as shown in Fig. 8, to touch the screen of the controller to be configured.

Touch the screen and press “Enter”.

The new controller’s information will be displayed on the “Selecting Controller to Configure” screen, shown is Fig. 6.

Figure 8: Selecting Controller by Touch

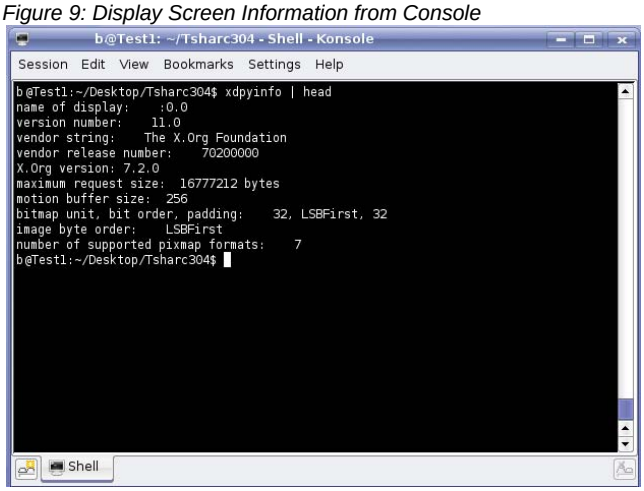


To Display Screen Information From Console

To find the current display and screen for the target system, enter “xdpyinfo | head” at the console. It is not required to be in the directory when running this command.

The first line of the output of the command will provide a text string in the format :<display>.<screen>. Fig. 9 shows “0.0”.

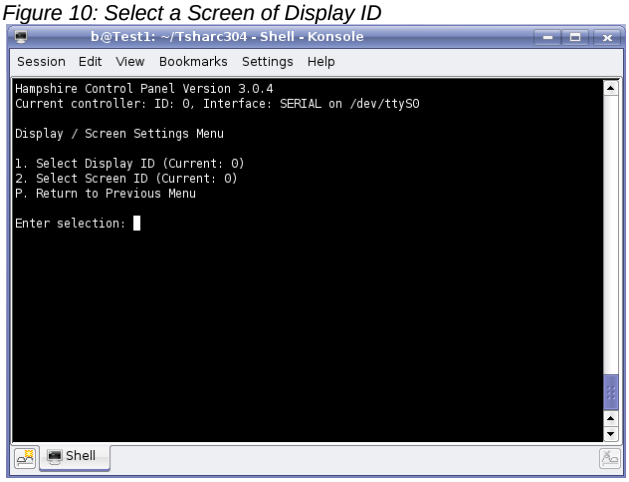
Figure 9: Display Screen Information from Console



Select Screen or Display ID

If there are multiple displays or screens used in the target system, the screen ID will need to be selected.

Figure 10: Select a Screen of Display ID



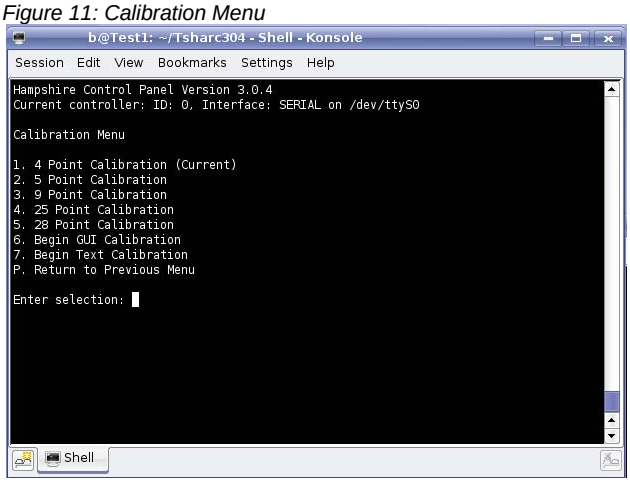
Selected Display ID
This option sets the Display ID of the current controller. This setting does not need to be changed if the mouse cursor is on the correct display after touching the current controller's screen. This is only necessary on systems with multiple monitors in the setup. Changes to this value are applied immediately. It is important that a valid Display ID be chosen. Changes made are applied immediately. It is important that a valid Screen ID be chosen. **Specifying a remote display is not current supported.**

Select Screen ID
This option sets the Screen ID of the current controller. This setting does not need to be changed if the mouse cursor is on the correct display after touching the current controller's screen. This is only necessary on systems with multiple monitors in the setup. Changes to this value are applied immediately. It is important that a valid Screen ID be chosen. Changes made are applied immediately. It is important that a valid Screen ID be chosen.

If the incorrect ID is chosen, a new one can be selected by selecting the menu choice shown in Fig. 10 and re-entering the display of screen ID.

Calibration

Figure 11: Calibration Menu



To select the number of calibration points used in the routine, select one of the first five menu choices. The selection made is indicated by "(Current)" next to the selection. Shown in Fig. 11, the 4 point calibration is selected. Enter a number, 1 through 5, and press "Enter". The (Current) will now be next to the chosen calibration.

To run a graphic calibration, a GUI calibration, enter 6 and press "Enter". A calibration routine will launch.

To run a text calibration, select 7 and press "Enter".

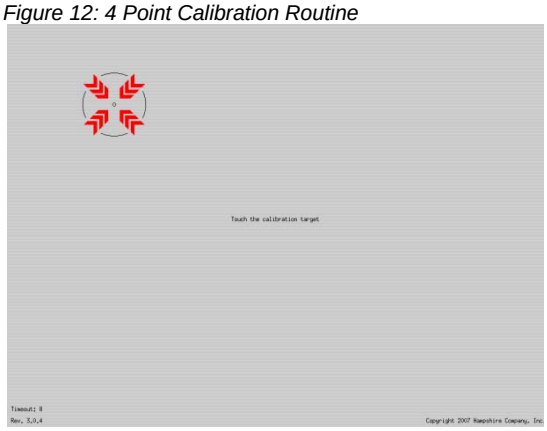
Calibration options are provided to allow for user control of the calibration results of the touch screen. Any calibration involving more than 9 points will linearize the touch screen. While a 9 point calibration will reduce the edge-bow, a 25 point calibration will be even more accurate. A 28 point calibration differs from a 25 point calibration in that more emphasis is added to the corners of the touch screen. A 4 or 5 point calibration will align the screen. Aligning the screen will move the screen relative to the current display. The more points used in the calibration routine, the more linear the touch screen will be.

The calibration targets are inset to bring the calibration routine targets away from the edge of the display. This eliminates physical limitations put into place by the screen's bezel. Default calibration inset values depend on the number of points selected.

4pt., 5pt. – 20% Inset
9pt., 25pt., 28pt. – 2% Inset

GUI Calibration

Figure 12: 4 Point Calibration Routine



It is important that all calibration routines be completed using a finger or a stylus. Using the available mouse will exit the routine immediately.

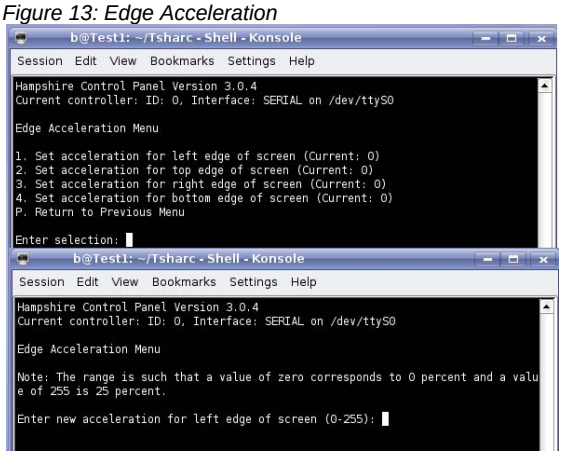
Each target of the calibration routine will appear one at a time. Touch the center of each target as accurately as possible. Hampshire's specially designed calibration targets assist in calibrating the touch screen as accurately as possible. If using a stylus during operation, calibrate with a stylus. Be sure to position in front of the touch screen as it will normally be used, sitting or standing. This will reduce error when calibrating the touch screen.

The calibration screen will automatically time out and return to the calibration menu if not touched within 10 seconds. This time-out feature insures that the user can exit the calibration screen in the event that the user has incorrectly calibrated the touch screen or the touch screen has been damaged or disconnected from the system.

If a custom calibration or test application is needed that works with or without X-installed installed as part of the touch solution, please contact Hampshire company for a source code example of how to communicate the daemon that provides all the necessary functional components to develop such a custom solution.

Edge Acceleration

Figure 13: Edge Acceleration



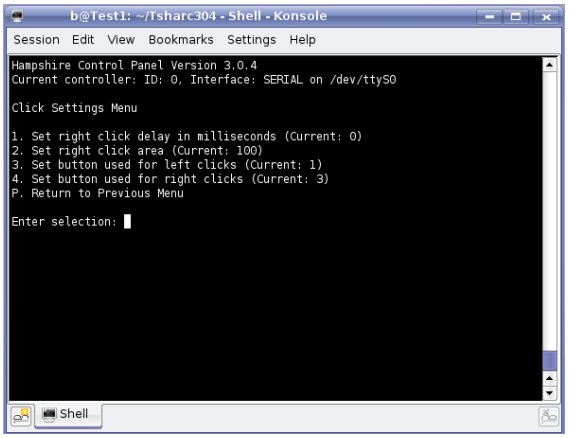
Edge acceleration enables the user to accelerate the cursor to the edge of a touch screen display without touching the active area edge. Without this feature, it is generally difficult to reach the outer edge because of the limitations the display bezel may impose on the user. Use edge-acceleration for systems that implement an "auto-hide" task bar or buttons at the edge of the display.

The Edge Acceleration menu enables the user to set the point at which edge acceleration will take effect.

Setting these values too high may make it difficult to use the screen normally. The maximum amount of edge acceleration is 25%. Typically, only a 5% edge acceleration is needed. Values between 0 and 255 are relative to 0% to 25% edge acceleration, where 5% has the value of 51.

Click Settings

Figure 14: Click Settings



Right Click Delay
Set the "Right-Click Delay" value to the preferred time, in milliseconds, needed to produce a right click event. A value of 0 will disable this function.
Right Click Area
The event area should be set to an area that is slightly larger than the activator tip. If the activator is a fingertip, the right click area should be at least as big as your fingertip.
Use the following formula to find the appropriate value:

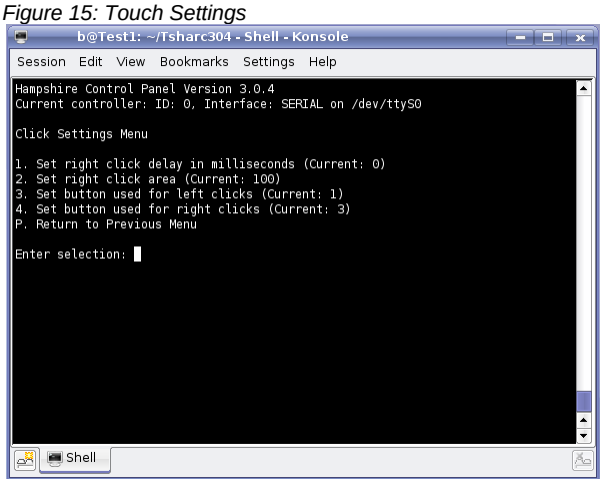
$$area = 4096 \times \frac{event\ width}{width\ of\ touch\ screen}$$

Button used for Left Clicks
This is used to change the button used for a left-click event when the screen is touched.
Button used for Right Clicks
This is used to change the button used for right-click events when the screen is touched. This may need to be changed to "2" depending on the target system configuration.

Enter in the corresponding menu number and press "Enter".

Touch Settings

Figure 15: Touch Settings



Normal Mode
This mode emulates a standard mouse. Selecting “Normal” will allow for single click, double click, drawing, dragging and right click option. This mode will allow the cursor to operate as a computer mouse typically would.

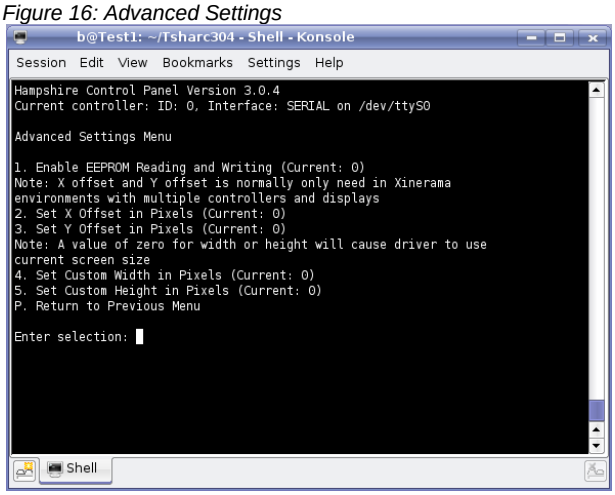
Touch Down Mode
Touch-down mode will allow for a click event to take place at “touch-down”. This mode will allow the cursor to operate as a single button-press or a single left-click of the computer mouse. The user will not be able to draw or drag if selecting this option.

Touch Up Mode
A touch is sent only at touch-up in this mode. Once lifted, the touch will register as a single left-click or button press. This disables right click and double click functions.

Enter in the corresponding menu number and press “Enter”.

Advanced Settings

Figure 16: Advanced Settings



Enable EEPROM Reading and Writing
Configurations made on the control panel will be saved to the controller for the next time it is connected to a touch screen. This includes calibration data as well as touch modes. This allows the controller to maintain calibration between systems with similar configuration if both have the driver installed and EEPROM enabled. Most of Hampshire’s controllers support EEPROM reading and writing. When enabled, this feature reads the calibration data and touch modes when starting the deamon.

X and Y Offset
These settings typically need changing when using a Xinerama environment with multiple controllers and displays. Enter in the necessary values, in pixels. This value will select the start of the range in the X or Y direction for the mouse cursor to remain in as the result of a touch.

Custom Screen Size
Enter a custom height or width using these settings. This value will select the end of the range in the X or Y direction for the mouse cursor to remain in as the result of a touch. A value of 0 for width and height will cause the driver to recognize the current screen size.

If the target system is setup with multiple displays and Xinerama is setup for the Xserver, both the X and Y offset and the custom screen size will need to be configured for every controller on the system. The X and Y offset will choose the minimum starting point in the most upper-left corner of the touch screen. If a second touch screen is placed to the right of another touch screen with a width of 1024 pixels, the X offset for the controller of the second touch screen would be set to 1024 while the X offset of the first screen would remain at 0.

Write Current Controller Data

Apply
To save settings for the current controller so that the next time the TSHARC daemon is started the current settings are used, select “Apply – Write Current Controller Data” from the Control Panel.

Exit the Control Panel

Cancel
To exit and revert settings to their last save for the current controller, select “Cancel – Exit and Read Current Controller Data” from the Control Panel.

Save
To exit and save settings for the current controller’s configuration to be read the next time the TSHARC daemon is started, select “Ok – Exit and Write Current Controller Data” from the Control Panel.

Command Line Parameters

Figure 17: Command Line Parameters

Command	Usage	Description
-h or --help	-h, --help	Displays help text
-q or --quick	-q NumCalPoints, --quick=<NumCalPoints>	Specify number of calibration points for quick calibration
-d or --display	-d <DisplayNum>, --display=<DisplayNum>	Specify display to use (use with -q)
-s or --screen	-s <ScreenNum>, --screen=<ScreenNum>	Specify screen to use (use with -q)
-c or --controller	-c <Controller Num>, --controller=<ControllerNum>	Specify controller to use (use with -q)

Uninstallation

To uninstall the TSHARC daemon:

- Return to command prompt
- Run “sudo uninstall.sh”
- Select “Y” to uninstall

If the scripts “registerSysVexample.sh” or registerX11example.sh” were executed during installation, the corresponding scripts should be run to unregister the daemon with the target system.

- sudo ./unregisterSysV.sh
- sudo ./unregisterX11example.sh

Troubleshooting

If after installation the controller does not respond, these additional steps may correct the problem.

- Enter “xhost +localhost:root:” at the command prompt using the same username as the initial X-Windows login. On some distributions, this command is required for the daemon to acquire permissions to communicate with the X-Server.

- As part of the daemon installation files, tsharc_state is an executable file that is included. This utility is used to ensure that basic communication is made with the controller.
    - Run “boot.tsharc stop” to ensure the daemon is not running during this communications test.
    - At the command prompt, change to the directory containing the installation files and run “.tsharc_state <device path>” where <device path> is the path for the device.
        - /dev/ttyS0 for COM1 serial controller
        - /dev/tsharcps2 for a PS2 controller
        - tsharcusb” for USB controllers

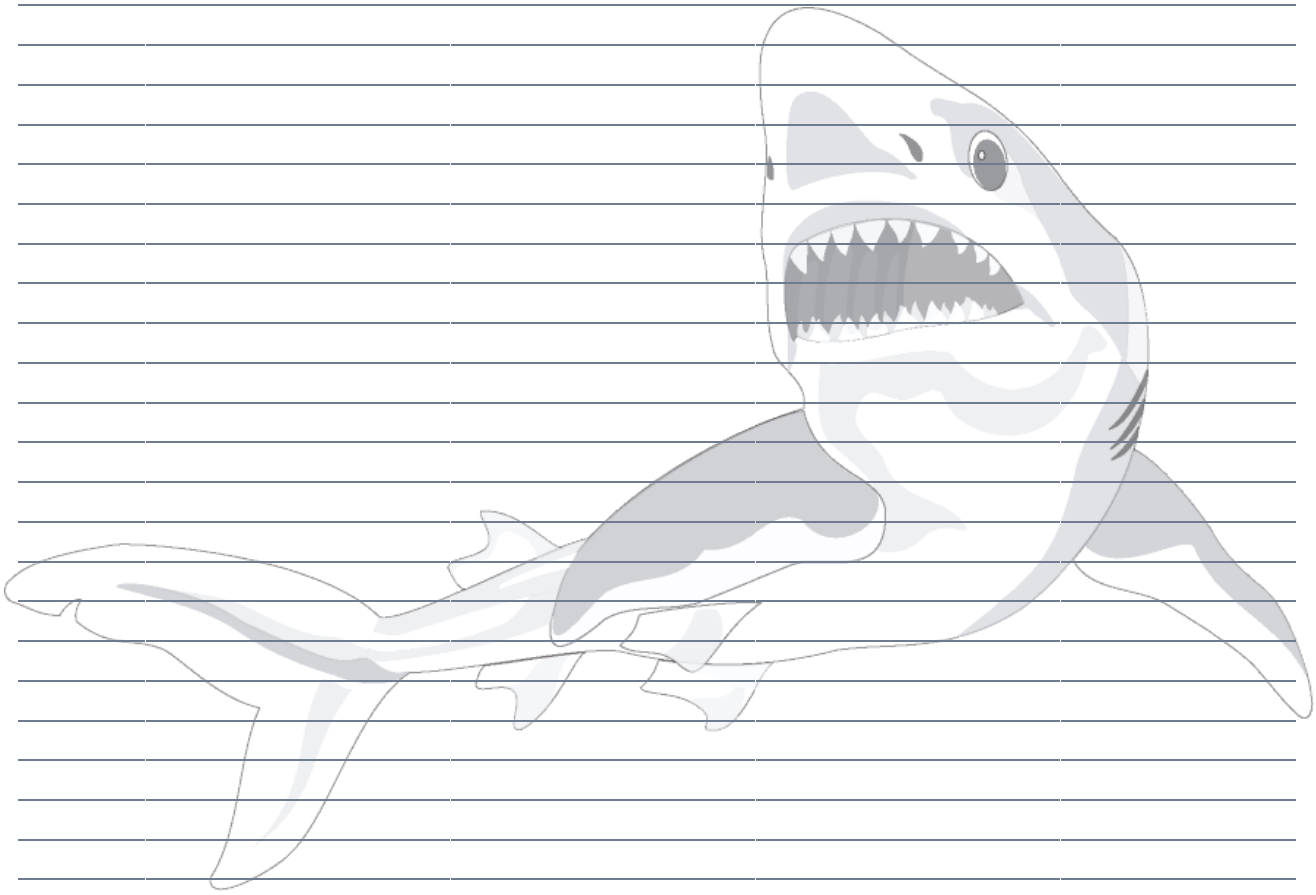
      After running the tsharc_state, touch the screen and the coordinates should be displayed within the console window. If coordinates are not shown, then additional paths may be tested until the correct one is found.
- If other solutions do not work, change the current console directory to the installation directory and run “./sysinfo.sh”. This command will generate a “sysinfo.tar.gz” file in the “/tmp” directory which can be sent to Hampshire Company for examination and troubleshooting purposes.

Recompiling Busybox

Busybox sometimes returns an error that the GLIBC on the system is incompatible. Recompiling the source code for Busybox corrects this issue. Additional instructions are available. Please contact Hampshire Company.

H A M P S H I R E

NOTES



Hampshire Company, Inc. 9055 N. 51st Street, Brown Deer, WI 53223
Ph: 414-355-4675 www.hampshirecompany.com Fax: 414-355-4775