Xeon - Action Item #694
Verify touchscreen works with launcher

02 Mar 2019 12:23 - Hammel

Status:	Closed	**Start date:**	02 Mar 2019
Priority:	Immediate	**Due date:**	
Assignee:	Hammel	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:	v0.1		
Severity:	01 - Critical		

Description

The touch library appears to be okay, so now try the launcher and see what happens.

Associated revisions

Revision eda95880 - 02 Mar 2019 16:56 - Hammel

RM #694: Add test that displays the active touch screen device, if any.

Revision a2ba7f08 - 03 Mar 2019 17:03 - Hammel

RM #694: Add pbtouchstate tool.

Revision 7f59a746 - 03 Mar 2019 18:22 - Hammel

RM #694: Add piboxGetDisplayTouch() so apps can test if we have a touch screen.

Revision 3ca6730f - 03 Mar 2019 22:37 - Hammel

RM #694: Make tool build use shared library so it works in Buildroot.

Revision 263b998f - 09 Mar 2019 18:28 - Hammel

RM #694: Add PIBOX_DISPLAY_NORMAL as display size if not BIG or SMALL. Support substring name comparison for supported touchscreens.

Revision dd76eb6b - 09 Mar 2019 20:39 - Hammel

RM #694: Added pbtouchstate tool to libpibox build and install.

Revision 4d1d0cda - 09 Mar 2019 20:40 - Hammel

RM #694: Add missing evdev module for touch screens.

Revision 4f68ea54 - 09 Mar 2019 20:40 - Hammel

RM #694: Merge in change for testing if touchscreen is the official RPi touchscreen to firstboot.

Revision 0df85a2c - 09 Mar 2019 20:41 - Hammel

RM #694: Add config.txt options to disable rainbow box and warning icons from firmware for Xeon.

Revision f209e1af - 10 Mar 2019 10:05 - Hammel

RM #694: Clean up display type reporting and testing for touch screen.

Revision e96a4084 - 10 Mar 2019 20:27 - Hammel

RM #694: Don't enable key and button presses if using a touch screen.

Revision aadf06c6 - 10 Mar 2019 20:28 - Hammel

RM #694: Switch back to exact match when looking for supported touchscreen because some touchscreens can have multiple devices with the same prefix with at least one having no additional text (so you can't identify it with a substring since all devices could match).

Revision 4ac04dc2 - 16 Mar 2019 13:22 - Hammel

RM #694: Fix up firstboot and associated files so firstboot properly identifies touchscreens.

Revision 258da0e2 - 16 Mar 2019 14:03 - Hammel

RM #694: Fix typo when referencing rpi-ft5406 driver in firstboot.

History

#1 - 02 Mar 2019 17:30 - Hammel

- Status changed from New to In Progress

- % Done changed from 0 to 10

The launcher doesn't work with the touch screen because
a) firstboot doesn't properly test if the screen is a touchscreen
b) piboxLoadDisplayConfig() doesn't properly check if the configuration file (pibox-config) shows it's a touch screen.

There are two things that need to be fixed.
1. Fix firstboot. There is a section that tests if the display size is 800x400, which just happens to match the official Raspberry Pi touchscreen display size. This now needs to use *ptests -t17* which will either report the name of the touch screen (only two are currently supported) or *Not found* if no touchscreen is found. This is a little slow but it only has to happen on firstboot.
2. Fix piboxLoadDisplayConfig() to use the third line of the pibox-config file to determine if we're on a touch screen.

There is also a new piboxTouchGetDeviceName() API call but that can't be used until the touch screen service is started and the launcher (and other apps) don't start it unless piboxLoadDisplayConfig() tells them they have a touch screen. So there is no function (yet, maybe in the future - some variation of the ts_setup in touchProcessor.c) in the APIs that can query tslib directly to find out if we're using a touchscreen.

Note that I can't call piboxTouchGetDeviceName() from piboxLoadDisplayConfig() because the latter doesn't know if the touchscreen service has been initialized yet. So it reads the static configuration to find out.

#2 - 03 Mar 2019 11:22 - Hammel

ptest -t17 doesn't always work unless I set a long timeout. This is because it sets up a thread before it does it's work and the work becomes async to the test. So this is far from ideal for testing if I have a touchscreen device.

A better solution is to simply use the scan_devices code from tslib. This is exactly what I want. But it's not an exposed API in tslib. Instead I need to call ts_setup and then check the return value to see if it's non-null. If it is, then we have a touchscreen device. Of course, this doesn't necessarily map to one of the devices we support. So we'd need to have a new function in libpibox:touchscreen.c that runs the EVIOCGNAME ioctl on a path (returned in the struct tsdev from tslib:ts_setup) to check if the name matches one we support.

Since this isn't a test of the library I'll need to add another dir for a special program generated from libpibox that can do all this and skip use of ptests by firstboot.

#3 - 03 Mar 2019 17:07 - Hammel

- % Done changed from 10 to 20

Implemented pbtouchstate tool.

Now I can update firstboot to use it.

#4 - 07 Mar 2019 17:45 - Hammel

All of the changes to libpibox and launcher have been made (though not all are checked in). I've built a libpibox pkg and installed it on the Xeon. I'm building the meta packages now and will install them as well. Then I'll build a launcher package (with latest updates) and install those. At that point I can test the launcher.

I need to verify the current installation on the Xeon boots without the packages, just to be safe. I'll need network access up and running automatically to verify my tests.

#5 - 10 Mar 2019 18:03 - Hammel

This still isn't working after many updates to clean up the code and rebase with upstream kernel, etc.

I need to find a way to debug why X works but tslib via my library is not. I can try various things like ts_read_raw() or ts_read_mt() and nonblocking reads (again). But I don't really know which way to go at this point.

Still digging around.

#6 - 10 Mar 2019 19:40 - Hammel

I disabled the launcher so X comes up in dev mode. Then I ran evtest. Interestingly: there is no /dev/input/event0. They start with event1. Anyway, evtest doesn't work. Also, the mouse on the favi(like) keyboard doesn't work either. So something else is up here. It's not my library. It's some kind of config problem.

#7 - 10 Mar 2019 20:24 - Hammel

I found the problem: The display gets two event devices, one for the touch screen one for a mouse (not sure where the second one comes from - seems like I did something to make that show up because it wasn't there before - I don't think). The hid_multitouch driver hides the first one so only the mouse device is still there after that driver loads. That device doesn't work with the touchscreen. So don't load that driver and you see both devices. And now I need to go back to exact matches for device names because the two devices from this display have the same prefix but with "(Mouse)" appended to the second one, so a substring match won't work to get the device I actually need.

I fixed these things and tested with evtest. That worked fine just under an xterm. So I re-enabled the launcher and tried again. This time I could tell it got the right device and when I pressed the terminal icon I got the xterm!! Woohoo! The device now works.

So now I need to make the changes in the source trees and rebuild the image. Once that's verified and changes pushed I can close this issue and move on to making the real phone.

#8 - 16 Mar 2019 17:05 - Hammel

- Status changed from In Progress to Closed

- % Done changed from 20 to 100

I had to get all my eggs in a row with this. There were changes to the dev build and libpibox and launcher and I had to make sure the dev build picked up the changes to libpibox, while the launcher was compiled with those changes. Plus I had to get changes integrated into the firstboot to properly identify the touchscreens.

Touchscreen is now working properly for Xeon. I've also verified that the changes are backwards compatible with the official RPi touchscreen. Also verified changes do not regress use on non-touchscreens.

All changes committed and pushed.

Closing issue.