



Virtio-loopback: a common device interface between Virt-AGL and Non-Virt AGL



Michele Paolino
2022-10-19

m.paolino@virtualopensystems.com
www.virtualopensystems.com



VOSYS activities update - Introduction

The “Common device interface between Virtualized and non virtualized AGL” project is composed by two main tasks:

- Virtio-loopback
- Touchscreen with sensitivity support

This presentation will go through the current status of both.



Presentation index

virtio-loopback

- Objectives
- Requirements
- Design overview
- Benefits
- Current Status
- Virtio-loopback components
 - Kernel space
 - User space
- Next steps

touchscreen-sensitivity

- Requirements
- Current Status
- Next steps

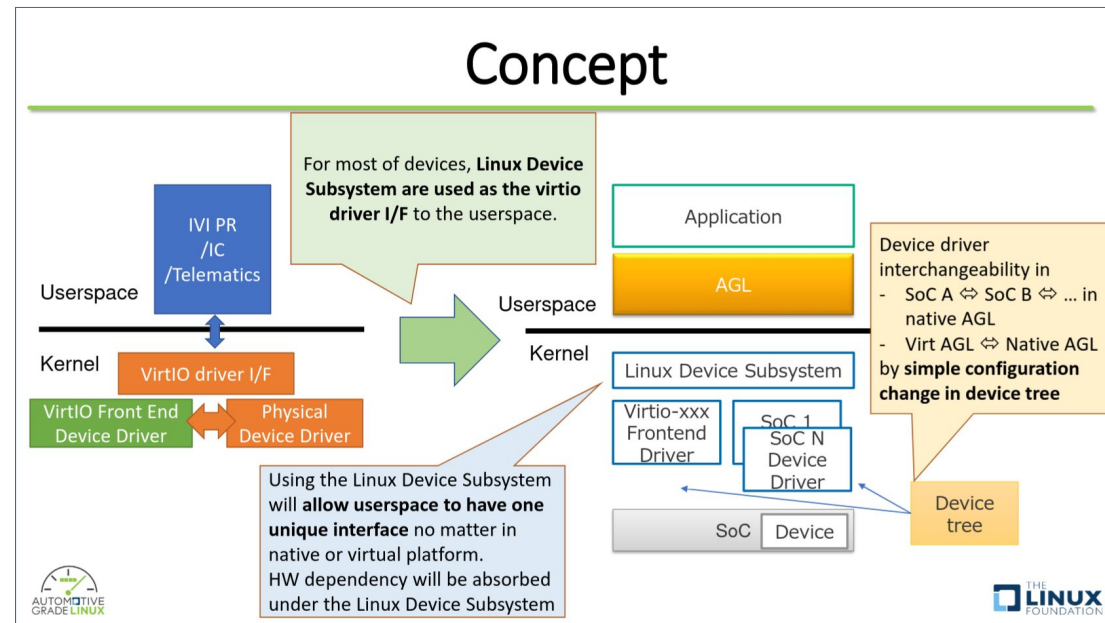


virtio-loopback objectives

Build a virtio based hardware abstraction layer that:

- Enables applications portable execution on:
 - Native and virtual machines environments
 - Several hypervisors

- Provides a standardized solution to build userspace drivers for both native and virtualized environments

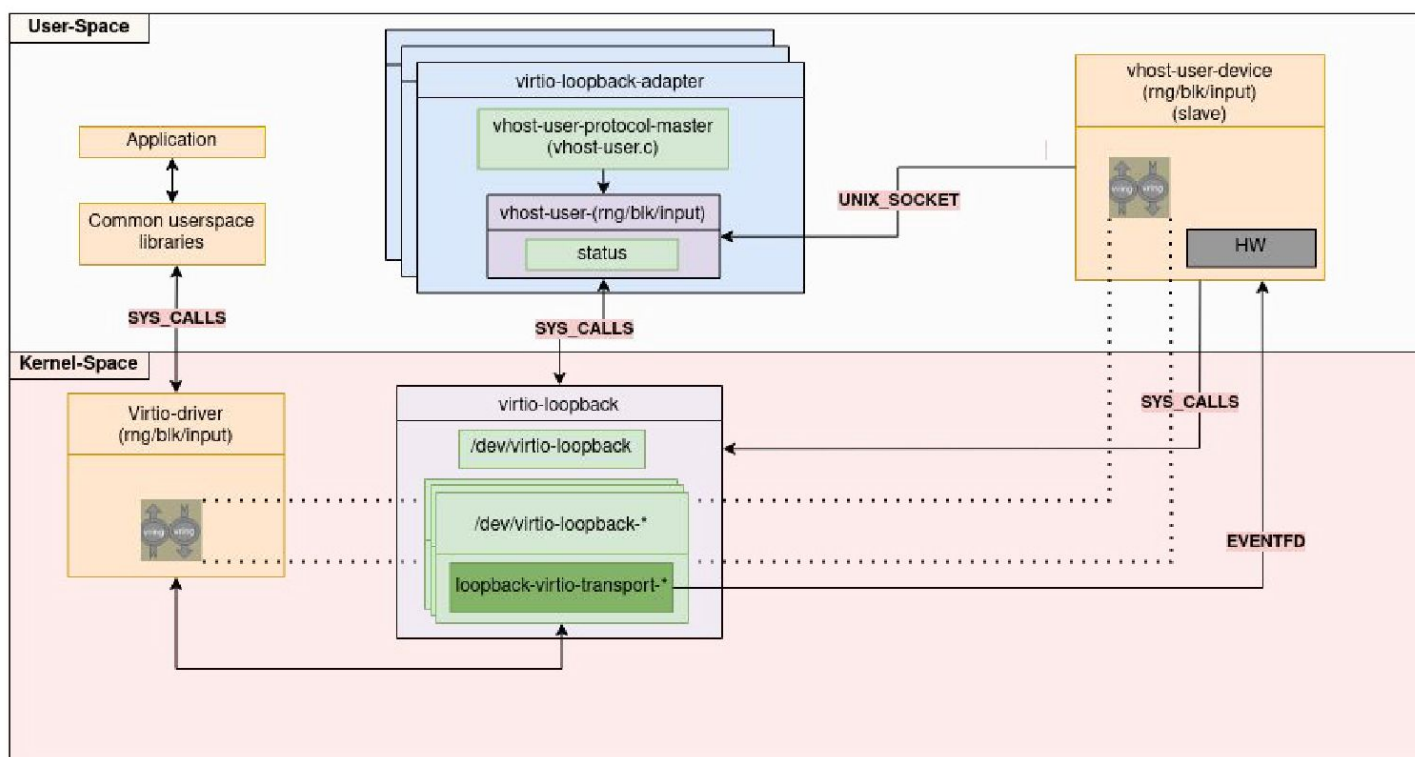




Design overview

Two new components have been designed and developed

- A user space application (virtio-loopback-adapter)
- A kernel driver (virtio-loopback)

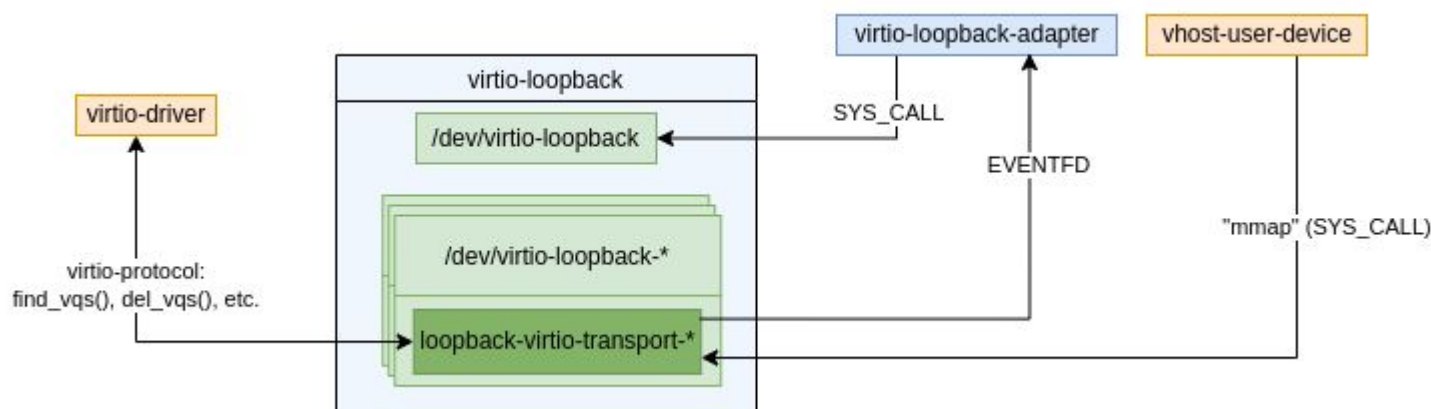




virtio-loopback

Kernel component composed by

- Char device (interface with virtio-loopback adapter used to set up control plane)
- Virtio transport (used as interface between the virtio driver and the vhost-user device)

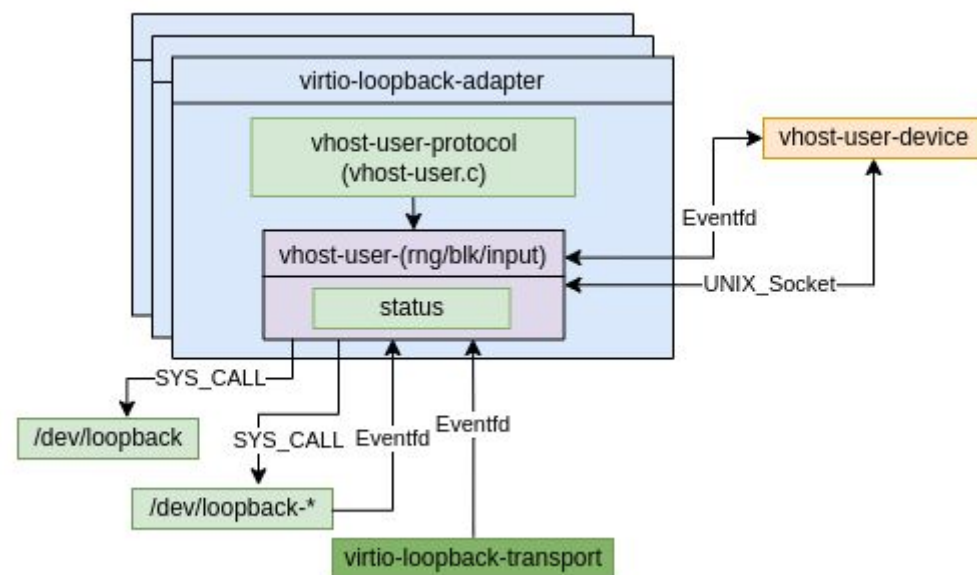




virtio-loopback-adapter

Userspace application that build the connection between the device and the virtio kernel driver

- It implements vhost-user protocol to communicate with the vhost-user device
- Uses syscalls (write/ioctl) to interact with virtio-loopback





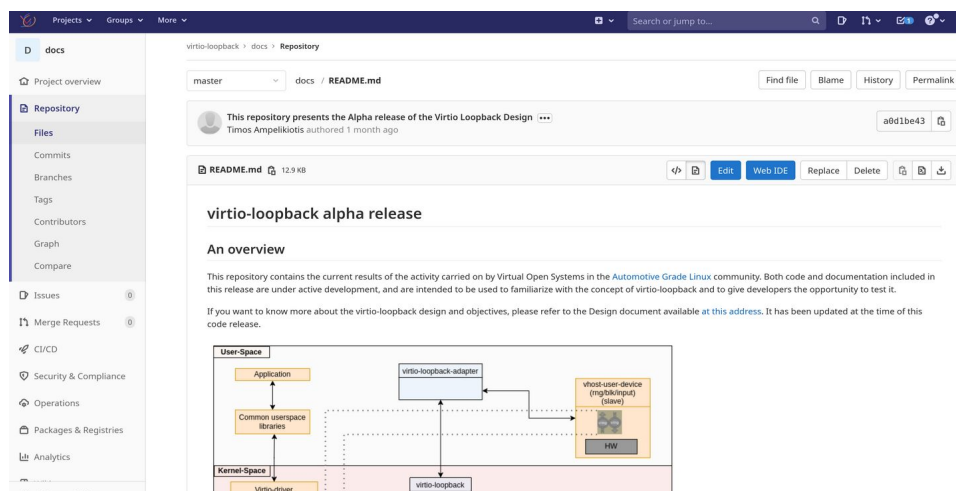
Benefits

- Existing user-space implementations can be reused
- Hypervisors that support virtio/vhost-user standards are fully compliant
- Data (vrings) exposed by the virtio driver in kernel space and directly mmap()ed by the device in user space
 - no copies, higher performance!
- Host and user space components are fully compliant with virtio and vhost-user open standards
 - virtio/vhost-user guarantee openness and stability



virtio-loopback current status

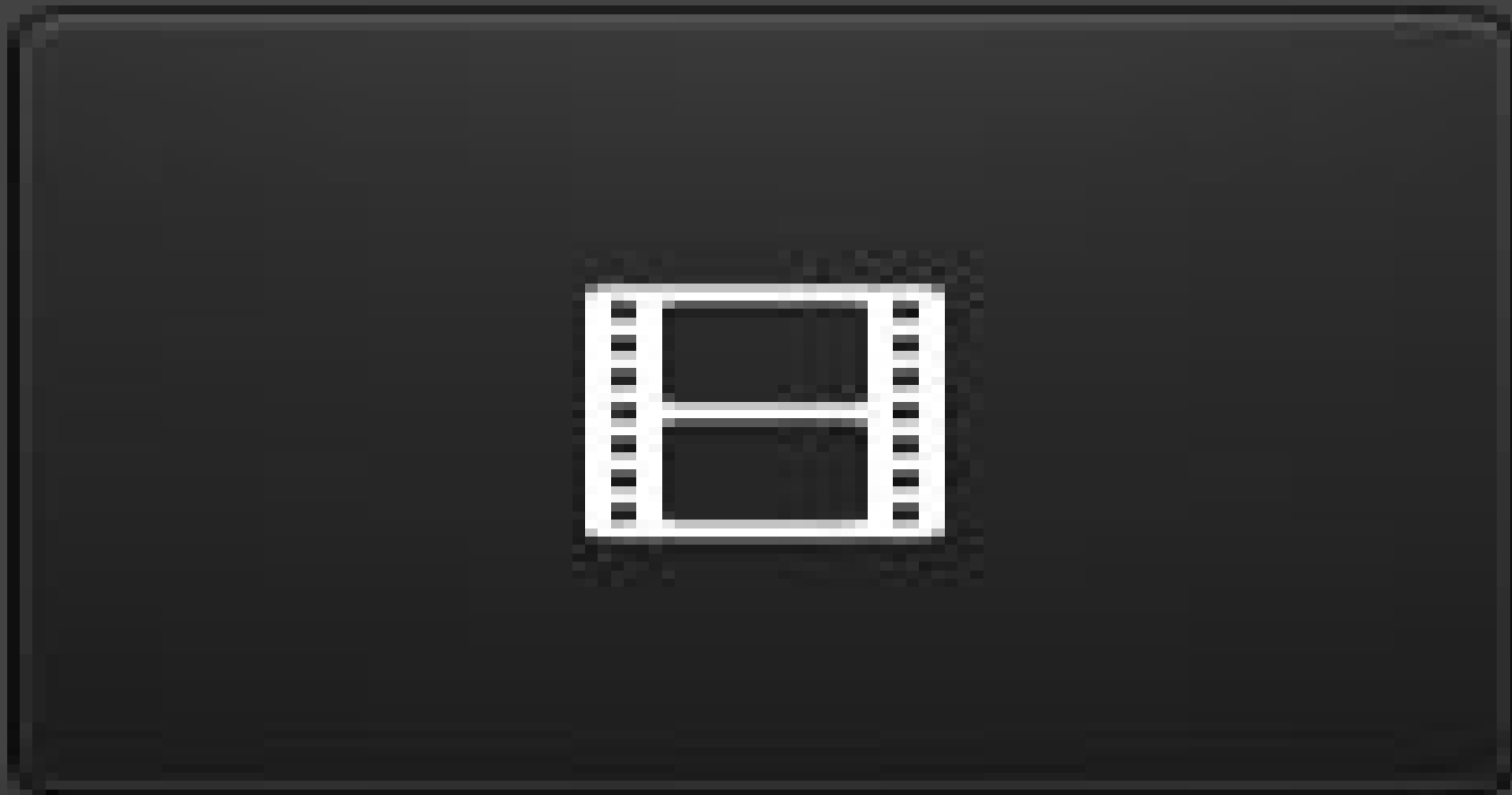
Alpha release publicly available for review, test and download:
<https://git.virtualopensystems.com/virtio-loopback/>



- Support for both x86 and Arm processors (tested on KVM virtual machines and R-Car M3 with AGL needlefish)
- Runs with both C and RUST vhost-user-rng implementations



virtio-loopback alpha release demo





virtio-loopback current status

Video demonstration publicly available !

<http://www.virtualopensystems.com/en/solutions/demos/virtio-loopback-agl-lf-alpha-release/>

Virtio-Loopback Alpha Release By Vos For Linux Foundation Automotive Grade Linux (AGL) virtio-loopback alpha release, an open source virtio-based hardware abstraction layer to enable portable application-driver interactions on native and virtualized environments



virtio-loopback alpha release: the virtio-based hardware A...



Copia link

During the video you will see:

- 1) Start the vhost-user-rng device
- 2) Insert the *virtio-loopback-transport* driver into the Kernel
- 3) Start the *virtio-loopback-adapter* user-space application
- 4) Execute a user-space application that requests 8 random numbers (1 bytes each), from the *virtio-rng* driver requests new data from *Vhost-user-rng*.

Guarda su YouTube

Virtual Open Systems Confidential & Proprietary

Virtual Open Systems

Virtual Open Systems Virtualization Services Products Solutions Research

Solutions >> Demos > Virtio-loopback-agl-lf-alpha-release

Experience kvm virtualization extensions

virtio-loopback: the open source virtio-based Hardware Abstraction Layer (HAL) for Linux

The alpha release of virtio-loopback supports random number generator devices and is available on VOSYS' gitlab website

Virtual Open Systems After the first *virtio-loopback Intermediate video demonstration* published in July 2022, with this new video demo *Virtual Open Systems* provides an update of the virtio-loopback developments with an important news: the code of the alpha release is publicly available on the company *Gitlab repository*! More in details, virtio-loopback is a *virtio-based hardware abstraction layer* that provides user-space applications with the ability to take advantage of user-space drivers using the standard virtio kernel interface. Virtual Open Systems develops virtio-loopback on behalf of the Linux Foundation and in the context of the *Virtualization Expert Group* of the Automotive Grade Linux (AGL) project.

The new components and features demonstrated in this latest video are summarized as:

- An updated version of the virtio-loopback-transport driver
- First implementation of the virtio-loopback-adapter
- Support for RNG vhost-user devices both in *C* and *Rust programming languages*
- Initial version of Rust vhost-user modified library
- HW heterogeneity: executes on x86 and arm64 targets, and has been tested on the Renesas R-Car M3 (AGL needfish)

Alpha release virtio-loopback components and features (2)

With virtio-loopback, applications using user-space drivers can be executed transparently in virtualized or native environments.

virtio-loopback is under active development, and in the Virtual Open Systems *Gitlab repository* you can find the current implementation together with documentation about how to reproduce the demonstration on your system. The next development steps are to support new devices (virtio-blk, virtio-input), make them work together concurrently, and introduce performance optimizations.



virtio-loopback next steps

As of today, a the alpha release supporting RNG has been delivered and reviewed by part of the community. Next steps:

- Gather and address further comments from AGL community
- Add support for input and block devices
- Prepare a Request for Comments (RFC) for relevant communities
 - To be evaluated proposals to kernel, QEMU, rust-vm
- Final version before AGL ALS



Touchscreen

The target solution agreed with EG-IVI is based on the MatrixOrbital HTT70A R1.0.0 device

- Supports different levels of touchscreen sensitivity
- HDMI touchscreen, open source driver
 - Can be used with different hardware platforms

```
155 int set_sensitivity(hid_device *handle, int sensitivity)
156 {
157     unsigned char buf[256];
158     buf[0] = REPORT_MXT_SENSITIVITY;
159     buf[1] = sensitivity;
160     int res = hid_send_feature_report(handle, buf, 2);
161     if (res < 0) {
162         return 0;
163     }
164     return 1;
165 }
166
```



Touchscreen - updates

The touchscreen device has been tested on AGL needlefish (Rcar M3)

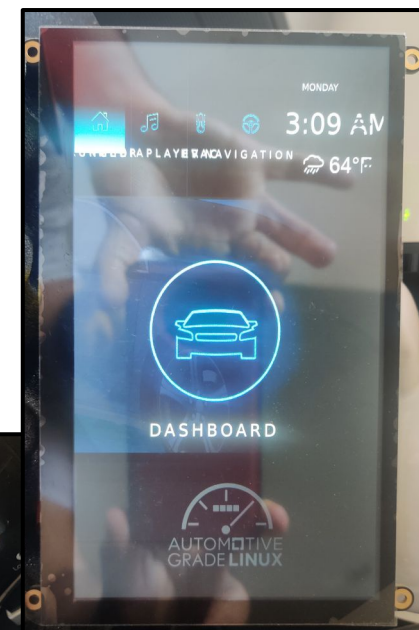
```
Terminal
root@m3ulcb:~# uname -a
Linux m3ulcb 5.10.41-yocto-standard #1 SMP PREEMPT Fri Sep 16 12:06:40 UTC 2022 aarch64 aarch64 aarch64 GNU/Linux
root@m3ulcb:~# ./htt_util --scan
HTT Detected.
- Device           : 0
- Firmware Rev     : 11953
- Driver Type      : GT9xx (3)
- Screen Rotation  : 0 degrees
- Default Backlight : 255
- Touch feedback   : 3 (Haptic and Piezo)
- Backlight fade   : 500
- Backlight dimming : Disabled
- Touch Sensitivity : 0 (normal).

root@m3ulcb:~# ./htt_util --sensitivity high
Setting sensitivity to high : Success!

The sensitivity comman[ 1957.315476] usb 2-1: USB disconnect, device number 5
d reboots the unit, further commands will not executed.
root@m3ulcb:~# [ 1958.732843] usb 2-1: new full-speed USB device number 6 using ohci-platform
[ 1959.000012] input: Matrix Orbital Multi-Touch Device as /devices/platform/soc/ee0a0000.usb/usb2/2-1/2-1:1.0/0003:1B3D:14C9.0005/input/input5
[ 1959.012956] hid-multitouch 0003:1B3D:14C9.0005: input,hidraw0: USB HID v1.11 Device [Matrix Orbital Multi-Touch Device] on usb-ee0a0000.usb-1/0

root@m3ulcb:~# ./htt_util --scan
HTT Detected.
- Device           : 0
- Firmware Rev     : 11953
- Driver Type      : GT9xx (3)
- Screen Rotation  : 0 degrees
- Default Backlight : 255
- Touch feedback   : 3 (Haptic and Piezo)
- Backlight fade   : 500
- Backlight dimming : Disabled
- Touch Sensitivity : 1 (high).

root@m3ulcb:~#
```





Code changes - HIDRAW kernel support

```
mpaol@vosys-s9:~/shared/test/meta-agl/meta-agl-bsp/meta-rcar-gen3/recipes-kernel$ git show
commit 0ef116f65cd867922fe102b5a7ad1582e118b685
Author: Michele Paolino <m.paolino@virtualopensystems.com>
Date:   Fri Sep 23 10:24:02 2022 +0200

    Add kernel support to HIDRAW for R-Car gen3

    Signed-off-by: Michele Paolino <m.paolino@virtualopensystems.com>
    Change-Id: I0260888954bc96bf7c4819c83e8ccc65667ae0e0

diff --git a/meta-agl-bsp/meta-rcar-gen3/recipes-kernel/linux/files/hidraw.cfg b/meta-agl-bsp/meta-rcar-gen3/recipes-kernel/linux/files/hidraw.cfg
new file mode 100644
index 0000000..bbd271a
--- /dev/null
+++ b/meta-agl-bsp/meta-rcar-gen3/recipes-kernel/linux/files/hidraw.cfg
@@ -0,0 +1 @@
+CONFIG_HIDRAW=y
diff --git a/meta-agl-bsp/meta-rcar-gen3/recipes-kernel/linux/linux-renesas_%.bbappend b/meta-agl-bsp/meta-rcar-gen3/recipes-kernel/linux/linux-renesas_%.bbappend
index 78996bf..0bbd346 100644
--- a/meta-agl-bsp/meta-rcar-gen3/recipes-kernel/linux/linux-renesas_%.bbappend
+++ b/meta-agl-bsp/meta-rcar-gen3/recipes-kernel/linux/linux-renesas_%.bbappend
@@ -10,6 +10,7 @@ SRC_URI:append = " \
    AGL_KCONFIG_FRAGMENTS += "namespace_fix.cfg"
    AGL_KCONFIG_FRAGMENTS += "Set_GOV_PERFORMANCE.cfg"
    AGL_KCONFIG_FRAGMENTS += "vivid.cfg"
+AGL_KCONFIG_FRAGMENTS += "hidraw.cfg"

# For Xen
AGL_KCONFIG_FRAGMENTS += " \
```



touchscreen next steps

As of today, touchscreen has been found and tested on the RCar M3. Next steps:

- Integrate it with vhost-user-input and virtio-loopback (AGL ALS)



contact@virtualopensystems.com

Web: virtualopensystems.com

Products: <http://www.virtualopensystems.com/en/products/>

Demos: [virtualopensystems.com/en/solutions/demos/](http://www.virtualopensystems.com/en/solutions/demos/)

Guides: [virtualopensystems.com/en/solutions/guides/](http://www.virtualopensystems.com/en/solutions/guides/)

Research projects: [virtualopensystems.com/en/research/innovation-projects/](http://www.virtualopensystems.com/en/research/innovation-projects/)