2022 - Common Device I/F between Virt-AGL and Non-Virt AGL

virtio-loopback design & touch sensitivity support deliverable

This section contains the virtio-loopback design file together with the touchscreen activity report. It represents the first delivery related to the Non-Virt Agl RFQ.

File	version	Description	Changes
EG-VIRT-VOSYS_virtio_loopback_design_v1.3.pdf	v1.3, 2022-09-21	Full review to comply with the current implementation. Next steps removed to keep the document only technical	6, 7.2, All
EG-VIRT-VOSYS_virtio_loopback_design_v1.2.pdf	v1.2, 2022-06-22	With new section about vring sharing	Section 1, 2, 8, 9
EG-VIRT-VOSYS_virtio_loopback_design_v1.1.pdf	v1.1, 2022-05-12	New delivery after review from Panasonic and OpenSynergy	Section 7
EG-VIRT_VOSYS_virtio_loopback_design_v1.0.pdf	v1.0, 2022-05-11	First delivery	-

virtio-loopback code deliverable

link	Description
https://git.virtualopensystems.com/virtio-loopback	Contains the code repositories - alpha version
https://git.virtualopensystems.com/virtio-loopback/docs	Lists the steps needed to reproduce the first virtio-rng demonstration based on vhost-user-rng device (Rust implementation)

Code review

Review meeting held on the 09-11-2022. Slides presented available here.

Attendees:

- Jerry Zhao (Panasonic) [JZ]

- Alex Benneé (Linaro) [AB]

- Maciej Szymanski (OpenSynergy) [MS]

- Michele Paolino (Virtual Open Systems) [MP]

- Alvise Rigo (Virtual Open Systems) [AR]

- Timos Amp (Virtual Open Systems) [TA]

Points of interest:

- VOS: presented details of the control plane, mapping mechanisms, etc. A video of the current implementation is shown

- MJ: Mmio memory used by the device - fixed memory address

- TA: Action Point VOS: to address this and make it dynamic

- MJ: Interrupt to be used should be dynamically selected

- TA: New version of the code do not use interrupts anymore. We are now using a non-blocking ioctl command

- MJ: Compiling with different flags should be removed if possible

- TA: Action Point VOS: stop using different flags at compile time and evaluate the possibility to take the device type (id)

from vhost-user-device and pass it to the kernel at runtime by the adapter

- AB: what are the changes done in vhost rust?

- TA: Changes on how the device is accessing the shared memory

- TA: Changes in vhost-user lib

- Add new vhost-user message for exchanging loopback driver fd

- Changing the mechanism that the device accesses the shared memory (vrings, SG list)

- Bypass the check, if a memory exists into the memory region table (The memory region table is not used in the current implementation).

- AB: Is it possible the vhost-user backend to be agnostic about how the memory is mmap-ed?

- AR, TA: This might be possible by using existing Vhost-user messages as "SET_MEM_TABLE". Even in that case the vhost-user-lib used into the

back-end might be slightly modified when it comes to "mmap"-ing the memory for the virtio-loopback case. In any case, a generic solution

to support also the virtio-loopback use case will be investigated. Maybe, the introduction of a new message to the vhost-user protocol

will be needed to accomplish more "fine grained" mappings.

- AB: Is it safe to access kernel memory - Security concerns?

- TA: The device asks the driver access to a number of host physical addresses. The virtio devices have a table with all these regions which can be shared with

the vhost-user device (Under the descriptor table -> attribute 'addr'). In this way the driver is able to check each time if the requested address is

under the descriptor table of the virtio device and if so, share a pointer to that memory region with the vhost-user device.

- TA: So the driver "whitelists" the addresses that the vhost-user device can have access to. Related to this, we don't find any critical security concern.

Action Points summary:

- VOS: update the adapter to stop using different flags at compile time and evaluate the possibility to take the device type (id)

- VOS: update current implementation to address mmio memory and make it dynamic
- VOS: Investigate more on what could be done on the backend mmap and proceed the discussion with the community

Weekly activity reports

This page contains weekly reports of the Virtual Open Systems activity about the "AGL_Virtio_On_Native_2021_RFQ" project.

Week 38

Activity report: This week we prepared the ALS event and the slides for the event. We are going to show benchmark results for blk and input, as well as a description of the architecture and a video demo. In the meantime, we are testing the yocto integration that has been released at this address. Finally, this is the last weekly report for this project. We are going to keep attending EG-VIRT calls to update the community about results follow ups.

Plan for next week: Next week we plan to present project results at ALS.

Week 37

Activity report: This week we pushed an update of the beta release, that includes the implementation of two out of the three action points (dynamic vrings allocation, single adapter binary) resulted from the review meeting. The third one, set_mem_table, but this opened a new challenge which is related to the synchronization between the virtio-driver and the vhost-user device that we believe needs a solution at the virtio-community level. The result is that virtio-loopback with set_mem_table works, but only for low bandwidth applications (input and rng). For this reason, the set_mem_table function will not be merged in our beta branch and will remain in "next". More info will come on this point.

At the same time, we are finalizing the yocto integration. After proper testing, we found an issue in the compilation of qemu vhost-user-devices (missing dependency) that is going to be fixed next week.

Plan for next week: Next week we plan to improve the beta-release documentation as well as we plan to finalize the yocto integration.

Week 36

Activity report: This week we worked on the action points of the review meeting to improve the quality of the code. We worked on the implementation of the SET_MEM_TABLE mechanism that has now been completed and is under testing. In the meantime, the yocto integration proceeds. We are able to build automatically the rust vhost-user-device and include it in the rootfs.

Plan for next week: Next week we plan to address the remaining action points from the review meeting. Also, we plan to finalize the yocto integration and to push the meta-layers via gerrit on the AGL repositories. All the activities are expected to be over by the ALS event, however we will continue to support where needed (up to three patches follow ups are included on the current contract)

Week 35

Activity report: This week we had the code review meeting on the 2022-11-09. We shared both slides and report of the call and we are already working on the action points that resulted from the discussion. In parallel, we started the creation of the yocto recipes to be integrated in meta-eg virt. We have today a layer that installs the virtio loopback driver and the adapter application. Next week we will focus on the vhost-user devices.

Plan for next week: Progress in addressing code review action points and in the development of the meta-layer.

Week 34

Activity report: This week we tested the current implementation of blk (dd application) and input (real keyboard and mouse). Also, we started preparation of the repository for the review planned on the 2022-11-09: updated version of the code has been pushed in the repo https://git.virtualopensystems.com/virtio-loopback with branch **beta-release-for-review**. Finally, we started the integration with yocto which will enable AGL to automatically build virtio-loopback.

Plan for next week: Code review, progress on the integration and testing/benchmarking activities.

Week 33

Activity report: This week we worked on the stability of the system and fixed the blk crashes that were mentioned last week. Now we are able to read and write from the blk device (e.g. /dev/vdb). Also, we completed the implementation of the vhost-user-input support that has been tested with a real keyboard.

Plan for next week: Test existing implementation of blk, rng and input. Start integration with AGL layers.

Week 32

Activity report: This week we presented the current status at the AGL F2F meeting with a video demo about the virtio-loopback alpha release. Also we published the video demo on our website at this address: www.virtualopensystems.com/en/solutions/demos/virtio-loopback-agl-lf-alpha-release/

On the development side, we found and solved a bug that was preventing writes on the blk device. At this point we are able to correctly read and write data using the blk device. We are now experiencing some random crashes that will be investigated next week.

Plan for next week: Investigate current blk issue, and start virtio-input activity in parallel.

Week 31

Activity report: This week we finalized both slides and demonstrations for the F2F meeting that will be held next week. Also, we responded to the first comments coming from Mikhail on the design document. On the development side, we tested the delivered code on the M3 board and with AGL needlefish. Also, we have an initial infrastructure for blk in place with an updated mmap mechanism for the driver. Data exchanges seem to work correctly, however further testing is needed.

*Plan for next week: P*resentation at AGL F2F with demonstration on R-Car M3 (needlefish). Continue development of the blk device support. Update design document after received comments.

Week 30

Activity report: This week we proceeded with the integration of vhost-usr blk in virtio-loopback. We have the initial negotiation between the device and the adapter working and we developed some enhancements on both driver and Vhost-user-lib used in vhost-user-dev:

- Overflow control: On vhost-user-dev side, after we mmap a buffer we check if all the buffer is contained into the mmap area, otherwise we add a new mapping for the remaining.
- Avoiding overlapping mmappings: Before we mmap a buffer we first ask the driver if its pa (Physical address) exists already in a shared mmapping. If yes, we find its offset in the existing mmapping, otherwise, we create a new one.

Plan for next week: Keep working on alpha release optimizations and on the vhost-user-blk device. Finalize presentation for AGL F2F and demonstration on R-Car M3

Week 29

Activity report: This week we released an update for the alpha version of virtio-loopback including code optimization and polishing. Then, waiting for comments from the community, we proceeded the development by starting the integration of vhost-usr blk in virtio-loopback.

Plan for next week: Keep working on apha release optimization and on the vhost-user-blk device. Finalize demonstration on R-Car M3

Week 28

Activity report: This week we released the alpha version of virtio-loopback, including adapter application, kernel driver and all the information needed to test the release in a Virtual Machine (x86 and arm both supported). Also, the first version of the slides for the next (postponed) F2F meeting was presented during the EG-Virt call.

Plan for next week: Update repos, finalize demonstration on R-Car M3

Week 27

Activity report: We continued working on the preparation of the repository for release. We added a docs folder with a README that will serve as an entry point for testing the full solution. In addition, we started preparing 1) a video demo of the current status and 2) the slides that will be presented in the next AGL F2F meeting. Finally we responded to the Call For Papers (CFP) for the next ALS summit in December, proposing a talk about virtio-loopback.

Plan for next week: Release repos, demo and presentation.

Week 26

Activity report: We modified vhost-user-rng rust version in order to work with our design and build a demo. Now we are preparing the repositories for release, including documentation on how to reproduce the work done (rust virtio-rng and userspace application communicating via virtio-loopback).

Plan for next week: Continue repos preparation activity. Start preparing presentation for the AGL event in Tokyo

Week 25

Activity report: We continue the work on the adapter side and we are now able to share data between the RNG device and the adapter. As a result, we have a first prototype of the full communication flow from the application to the device. The device today is a C implementation of vhost-user-rng

Plan for next week: Clean the code and start preparing repo sharing

Week 24

Activity report: We started adding vhost-user support in the adapter application. The adapter and the vhost user device (RNG) are able to exchange messages. No data sharing is yet possible, this is going to be developed next week.

Plan for next week: Adding vhost-user data sharing support between adapter and vhost-user device

Week 23

Activity report: Adapter extended to support both inbound and outbound communication with the kernel driver. We extend the adapter to respond to RNG requests from drivers (no vhost-user device in the picture yet) with fixed values. We have now a small PoC of an application that requests an RNG to the driver, which is forwarding the request to the adapter. The adapter is for the time being replying with hardcoded values. Plan for next week is to start adding vhost-user in the picture to make sure the vhost-user device provides the date.

Plan for next week: Start adding vhost-user support to the adapter

Week 22

Activity report: Ongoing activity is focused on the adapter component. Our adapter is now able to address all the three phases defined in Weekly #19. As a result, we are now able to share information between the kernel driver and the adapter.

Plan for next week: Complete an initial version of the kernel/adapter communication (kernel to adapter communication missing). Extend the adapter to respond to RNG requests from drivers (no vhost-user device in the picture yet) with fixed values.

Week 21

Activity report: Ongoing activity is focused on the adapter component. We developed a first version of the component that is able to address the initial negotiation with the vhost-user RNG device (Step 1 of the 3-phases defined in weekly 19) and is able to complete virtio negotiation with the kernel driver (Step 2). Also, we presented the development plan during the last EG-VIRT call, aiming at providing an early version of adapter, kernel driver and simplified vhost-user RNG device by end of September.

Plan for next week: Start to address step 3 of the 3-phases defined in weekly #19

Week 20

Activity report: Ongoing activity is focused on the adapter component. We developed a first version of the component that is able to address the initial (when VM is not running) negotiation with the vhost-user RNG device (Step 1 of the 3-phases defined in the last weekly). Also, we started to address Step 2 - the virtio negotiation with the kernel driver (e.g., to get the address of the vrings, etc).

Plan for next week: Virtio negotiation with the kernel driver (e.g., to get the address of the vrings, etc)

Week 19

Activity report: we published the video demo here: http://www.virtualopensystems.com/en/solutions/demos/virtio-loopback-agl-lf/. Also, we continued the activity on the adapter component, focusing to address the first negotiation phase between the rng device and the adapter. More in detail, we conceptually split the vhost-user devices operation in 3 phases:

1) Vhost-user negotiation: negotiation of information about Vhost-user and Virtio protocols
2) Virtio negotiation with the kernel driver (e.g., to get the address of the vrings, etc)
3) Data exchange between adapter and kernel driver

Plan for next week: The objective for next week is to address step 1.

Week 18

Activity report: We continued extracting the code written in Qemu to the user-space application (Adapter). At this point the app is in an intermediate state and the plan is to be able to use the same basic mechanisms used into the demo, (SYS_CALLS, EventFD, shared messaging struct) in order to start and exchange messages with the virtio-loopback-transport driver.

Plan for next week: keep working on the adapter application. Publish video demo

Week 17

Activity report: During the EG-VIRT this week we have shown an intermediate version of virtio-loopback-transport that redirects virtio-rng requests to user space. We are now finalizing the video for publication on youtube and on the Virtual Open Systems website. In parallel, we started the activities on the adapter side as agreed during the call. We started extracting part of the re-usable code we have developed for qemu in a newly created user-space application (i.e., our adapter).

Plan for next week: work on the adapter application*,* porting basic functions from QEMU.

Week 16

Activity report: We prepared a small demo to be shown during the next EG-VIRT meeting. The demo shows the current status of development, with an intermediate version of virtio-loopback-transport that redirects virtio-rng requests to user space. In userspace, QEMU with a virtio-rng is used.

Plan for next week: Share current status and reply to havy comments/questions.

Week 15

Activity report: This week we shared version 1.2 of the design document. It Contains more information about vrings sharing in section 8. Moreover, we started finalizing the early PoC (described in week 13 with a picture) to share the results with the community.

Plan for next week: Share early PoC results*.*

Week 14

Activity report: This week we started preparing the new version of the architecture document. A new section related to vring sharing has been prepared, explaining differences between buffer addresses resolution in vhost-user with qemu and our virtio-loopback case.

Plan for next week: Finalize and share the updated document

Week 13

Activity report: Mapping issue with the early PoC fixed. We are able to request and receive a random number to a QEMU virtio device using a draft virtio-loopback transport driver. As shown in the Figure below, we have a system with a modified QEMU and draft virtio-loopback transport driver. When an application calls a virtio-rng function (e.g. hexdump /dev/hw-rng) the call is intercepted by the virtio-loopback driver and is forwarded in user-space to the QEMU process.

This is a first example of kernel call redirection in user space.

blocked URL

Plan for next week: Prepare an updated version of the architecture document, with lessons learnt from the above PoC development. In particular, the focus will be given to virt-queue sharing.

Week 12

Activity report: Continue the development of the PoC with qemu acting as adapter. We created a new mapping to share the vrings in the transport driver. Qemu is receiving it, but is not yet fully functional.

Plan for next week: Fix mapping issue of the early PoC.

Week 11

Activity report: Developing an early PoC with virtio-rng driver, a custom version of the transport module and a qemu instance that works as a first version of our adapter. Today, we are able to exchange basic messages between the driver, the transport and the qemu instance.

Plan for next week: Address any additional questions/comments on the design. Continue developments on virtio-loopback kernel early PoC (control plane)

Week 10

Activity report: Design document (v1.1) presented at the AGL Workshop event. Questions were related to: 1) possibility to use the virtio-loopback concept also on AWS graviton 2) difference between DPDK and virtio-loopback use of vhost-user and 3) possibility to add net support to virtio-loopback (vhost-user-net for RCar M3). This points will be discussed in the next EG-VIRT meetings. Development experiments done with virtio-rng driver and a custom version of the transport module.

Plan for next week: Address any additional questions/comments on the design. Continue developments on virtio-loopback kernel side

Week 9

Activity report: First version of the design document prepared and shared with the community. First comments received led to version 1.1. Slides for next week Hybrid workshop prepared. Development of initial loopback component in kernel side continued this week.

Plan for next week: Present the document at the AGL Hybrid meeting. Continue developments on virtio-loopback kernel side.

Week 8

Activity report: On the touchscreen sensitivity side, we found that CONFIG_HIDRAW is not set in the standard M3 configuration. This kernel driver is needed to expose the /dev/hidraw devices used to set sensitivity (via ioctl). As a result, with AGL today the touchscreen device works but it is not possible to change sensitivity.

For what concerns the virtio-loopback, we are working on the document to be shared with you next week. We have at this point an initial draft that describes the components to develop, as well as it gives a description of both data and control planes of virtio-loopback.

Plan for next week: Share design file and discuss it during the next EG-VIRT call.

Week 7

Activity report: On the touchscreen sensitivity side, *tests on a guest system are continuing. Today, the VM is not able to set the sensitivity directly because there is no hidraw interface inside the guest.*

For what concerns the virtio-loopback, we had a call with Mikhail and his colleagues. We agree on the key requirements of the work (no modifications to the userspace devices, applications untouched). Moreover, we experimented with the implementation of a dummy virtio-mmio-transport for testing purposes. In addition, we focused on the draft design document with the objective of presenting it during the next EG_VIRT meeting. Today we have a list of the new components that need to be developed, which will be detailed in the following weeks.

Plan for next week: Proceed with the design proposal to be shared on May the 11th, especially related to the adapter part.

Week 6

Activity report: On the touchscreen sensitivity side, the device arrived on time this week. We started testing on the host side, and we can confirm sensitivity can be set via the HTT_Utility application. Also, we are today able to run a KVM virtual machine that reads device events using vhost-user-input. The "dmesg" log of the VM is the following:

```
input: Matrix Orbital Multi-Touch Device as /devices/pci0000:00/0000:00:03.0/virtio0/input/input1
```

For what concerns the transport layer, the focus of the week was the communication between the user space drivers and the adapter. On this we investigated the qemu side (vhost-user-* devices in qemu) and the libvhost-user library. This latter is particularly of interest, it is integrated in QEMU as a subproject and provides vhost-user devices implementing a library with common functions (queue push, pop, notify, etc).

Plan for next week: Continue touchscreen testing, move HTT_Utility for sensitivity test on the guest side. Proceed with the design proposal to be shared on May the 11th.

Week 5

Activity report: On the touchscreen sensitivity side, thanks to Jerry support we got positive feedback from production IVI EG and as a result, we proceeded with the purchase of the device which is expected to be in our hands next week.

For what concerns the transport layer, we investigated the mmio implementation and we started having a look at vhost-user-net to evaluate the possibility to swap rng with net (as requested by Jerry during the last call). We also started detailing a draft design that will be presented at EG-VIRT level on the 11th of May and to the whole AGL community the week after.

Plan for next week: Start touchscreen testing. Move investigations on the user space adapter layer.

Week 4

Activity report: On the touchscreen sensitivity side, we investigated the library provided by the producer to get/set sensitivity. It is HID based, and the linux driver ends up with an ioctl() call to the touchscreen device file. Producer claims Linux is already supported (to be tested). As a consequence, we explored the virtio-input-hid. No news, but not even counter-indications from EG-IVI related to the htt70a-ips. Let's give them a few days more and then we proceed with the purchase of the device, followed by testing phase and eventually virtio-input extensions.

We proceeded with the user space devices exploration and best solutions seems to be virtio-user-blk and virtio-user-input available here, as well as the vhost-user-rng solution developed in the context of the rust-vmm project. In addition, we started experiments based on the rng, investigating both the standard virtio-rng and the vhost-user version based on rust.

Plan for next week: Purchase the touchscreen device and start testing. Move investigations on the virtio-transport layer.

Week 3

Activity report: Got response on the input-device-abstraction confluence page. The htt70a-ips is perceived as a good solution. We are under discussion with IVI-EG (this week call was canceled, we proceeded on confluence), if no counter-indication we proceed with the purchase next week. In parallel, investigations on the possible target user space drivers for the virtio-loopback activity. An initial point could be represented by the acrn userland virtio drivers (to be further investigated)

In parallel, the virtio activity continued with investigations aiming to assess the state of the art (SoA) on the kernel side.

Plan for next week: Finalize investigations about the htt70-ips display device (hidapi, virtio-input), and if no counter indication from IVI-EG proceed with the purchase of the device.

Week 2

Activity report: Related to the touch screen task, no response yet on the input-device-abstraction confluence page. Anyway, we continued the quest for a suitable device and we found the htt70a-ips, a solution from Matrix Orbital that supports touch sensitivity and has an open source support utility based on hidapi able to change sensitivity level.

In parallel, the virtio activity continued with investigations aiming to assess the state of the art (SoA) on the kernel side.

Plan for next week: SoA investigations focusing on the user space side (available drivers, etc).

Week 1

Activity report: received news from the technical support of the gechic T151a. Basically they say sensitivity is exposed only to On Screen Display (OSD) and not to the OS attached via USB. As a consequence, we attended the EG-IVI call on Thursday and since Takeuchi-san was not in the call, we sent him a message in the Input Device Abstraction page on confluence. The objective on our side is to understand better the requirement to find a way to fulfill the request from the IVI guys. Main open questions at this point are 1) whether they need this feature for a device they have in their hands, and 2) if this is not the case, what is the target usage for this feature. I hope we will be able to initiate a discussion on this topic soon.

In parallel, we started the virtio activity. We prepared an initial virtual development environment based on KVM that will be used for explorations and trials.

Plan for next week: proceed with investigations on the virtio activity

Week 0 (started 2022-03-09)

Activity report: investigations related to Task 1. Looking for touchscreen displays with sensitivity support, we found this gechic T151a that seems to have sensitivity support. We contacted the producer to understand if the drivers are open source and whether the sensitivity feature is exposed to the OS.

Plan for next week: continue the display quest, and we hope to have more information for you already for our call on Wednesday.