

[GSHA] Display Sharing Tech Brief Work

The wiki text below was used to collaborate on the draft text for this tech brief. Final development has moved to editing the text in the attached word document



Sharing a physical display across multiple operating systems

Summary

Five primary techniques for distributing graphics and HMI between cooperating systems have been identified in the GENIVI Graphics Sharing and Distributed HMI project (GSHA).

- *Surface Sharing*
- *API Remoting*
- *GPU Sharing*
- *Display Sharing*
- *Shared State, independent rendering.*

Display Sharing provides support in hardware for compositing the display output from multiple operating systems into a single final display buffer. ~~In this way~~ *Using this, a combined HMI can be created from independently operating systems without any additional interaction.*

Description

Display Sharing is used to allow multiple operating systems to share a physical display whilst retaining control over how their output is combined. Typically, this control over the separation is an important requirement in the system design. For example, the surface area of a physical display may be mostly dedicated to display cluster graphics, but areas of the display may also show IVI information such as multimedia or navigation, or the output from a reversing camera. In this case the IVI graphics should not be allowed to overwrite the cluster graphics.

This separation is enabled by a hardware block that provides separate hardware display layers with support for composition and per pixel alpha blending. Each operating system can be assigned a dedicated layer to draw into. The hardware block combines (composites) all layers into a final display buffer when is then passed on in the Display Unit for output on the physical display. By providing control over the Z-ordering (which layer is "in front" or "behind" in the drawing order) the compositor can ensure that one layer does not overwrite the other. For example, placing the layer assigned to the Linux operating system providing IVI behind the layer assigned to the safety-critical OS providing the Cluster in the Z-ordering. Per pixel alpha blending allows a visually complex composition of the different layers both in terms of transparency between the layers and the shape of the bounding area of each layer. A layer does not need to constrain its rendering into a simple box for example. In the case of the Renesas R-Car Gen 3 SoCs this functionality is provided by the VSPD hardware block.

Since the hardware block composites the different display layers each operating system does not need to know about the other and there is no need to invent protocols to pass graphics between operating systems.

For simplicity of description so far we have talked about sharing between multiple operating systems, however it could also be the case that a single operating system provides all functionality and assigns the hardware layers appropriately.

Where hardware display layers are not available a form of Display Sharing may be provided in software by the Display Manager of certain Hypervisors. The Display Manager provides a mechanism for guest VMs to pass display buffers which it then combines.

Key Characteristics

- The physical display can be shared across multiple operating systems or major functional units of a single operating system.
- H/W Compositor block provides multiple layers that can be assigned to different functional units. The hardware then combines (composites) them to create the final display buffer.
- The Compositor hardware block is typically close to or part of the hardware Display Unit block.
- The different systems do not need to know about each other.
- No need to invent protocols to pass graphics between operating systems.
- H/W may provide additional features that enhance isolation of the different functional units.

Display Sharing can be usefully combined with virtualisation and functional safety technology in both software and hardware. For example, the Renesas R-Car H3 SoC has multiple hardware virtualisation features to prevent one operating system stalling the graphics processing of another. Along with functional safety and safe rendering features. This protection can be further enhanced with a Hypervisor.

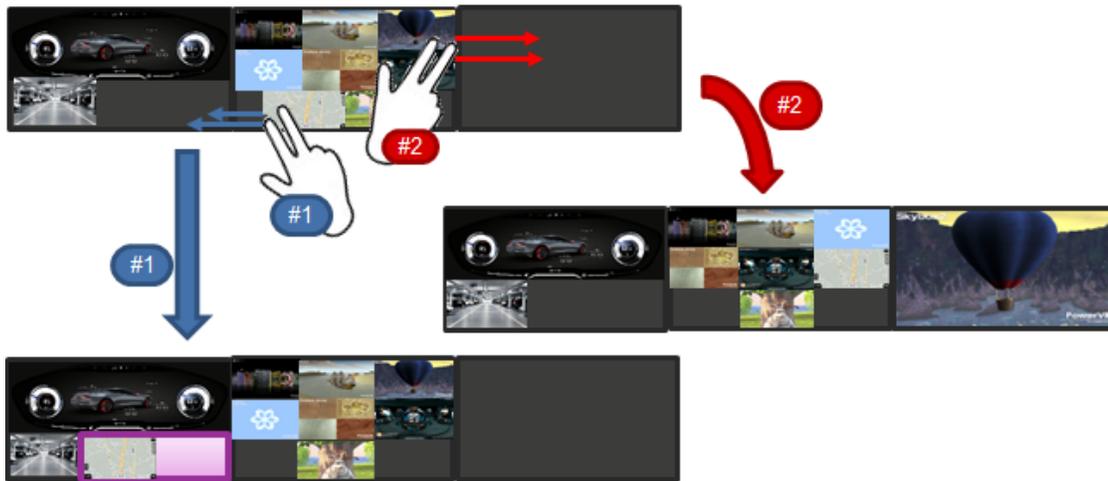
Usage in virtualised and functional safety environments

Display Sharing can be usefully combined with virtualisation. Particularly in the case of consolidated hardware where multiple ECUs or SoCs are consolidated into a single ECU or SoC. An environment in which mixed functional safety requirements are also common. For example, in the case of the Renesas R-Car H3 SoC a functional safety environment may be running on its R7 CPU providing early reversing camera video, whilst an IVI environment running the Linux operating system and/or Android runs on the general purpose A6x CPUs.

The Display Manager of a Hypervisor can enhance the functional safety of Display Sharing by controlling access to the hardware block. The Display Manager may also enhance how displays are shared by offering control over how the hardware layers are represented to each operating system. It may describe a layer as being smaller than the physical display area to a specific operating system for example. Allowing this smaller area to be offset anywhere on the larger screen.

The Hypervisor may also combine Display Sharing with other operating system separation or functional safety features in the hardware to create a more robust and flexible platform. For example the Renesas R-Car H3 SoC provides hardware GPU virtualisation features that separate the processing of graphics from each operating system before it reaches the hardware composition layers. This prevents one operating system stalling the graphics processing of another. Graphics sharing in a virtualisation environment is covered in more detail in an upcoming Tech Brief.

Case Study



The Renesas R-Car Canvas Demo demonstrates consolidation of cockpit functionality onto a single H3 SoC running on a Salvator-X board driving three displays and multiple operating systems. In the version discussed here the Integrity Multivisor hypervisor is used to virtualise an Integrity RTOS instance running a cluster demo and Linux operating system instance running an IVI stack and other functionality. The first display is shared between Integrity and Linux, whilst the second and third display is dedicated to Linux. For display one Integrity and Linux are assigned different hardware display layers in the VSPD hardware block, with Linux being placed behind the Integrity cluster in the Z-ordering to ensure the Linux applications cannot overwrite the cluster graphics.

The demo implements touch gestures in the Linux instance. These can be used to size and swipe applications together and between the screens. When applications are pushed to display one they are automatically docked into a dedicated area in the cluster. A small amount of transparency allows Linux applications placed behind the cluster to show through.

The cluster does not need to understand what the Linux applications are doing and the hardware support for OS separation ensures cluster performance is maintained.

Authors

- Stephen Lawrence, Renesas Electronics

Related technology

- Virtual display
- Surface sharing
- GPU virtualisation
- Display Manager in virtualisation technology such as Hypervisor
- ~~Alternatively provided by a Display Manager in virtualisation technology such as a Hypervisor. Possibly with less protection of layers not over-writing each other~~

Open questions:

1. ~~Do we want to be specific and say **Physical** Display Sharing or not? Is there a reason to be that specific or is it limiting?~~
A: Decision in 29/11 GSHA meeting is it is sufficient to call it simply Display Sharing
2. ~~Move the HV alternative in key characteristics into the body text so as not to confuse main points?~~
A: Decision in 29/11 GSHA meeting is to mention HV as an alternative at some point in the text.
3. ~~The virtualisation section seems a little large, but although not necessary to use display sharing, they partner well — particularly in H3 where it brings in the other gfx functional safety features — and it seems right to mention it. Particularly as it used in the Canvas demo.~~
A: Decision in 29/11 GSHA meeting is to simplify where possible. Can mention where an additional feature changes in some way when combined with Display Sharing.

I used [this page](#) to analyze and put in more review comments / Gunnar.