Display and GPU Sharing Case Study
October 10th, 2018 | using Renesas R-Car H3 Canvas Demo

Stephen Lawrence
Principal Engineer, Renesas
BIT Lead, Genivi

GENIVI is a registered trademark of the GENIVI Alliance in the USA and other countries.
Copyright © Renesas Electronics 2018.
Display and GPU Sharing in Canvas Demo

• Overview

• Video demonstrations
  • Video 1: showing gfx interaction
  • Video 2: focus on gfx sharing into cluster

• Enabling technology
  • Major building blocks
  • Some benefits of the approach
Canvas Demo Overview

• Created to demonstrate consolidation of cockpit onto single H3 SoC running on single Salvator-X board
• Various variants created, combining cluster/adas/ivi, but today showing you Linux gfx sharing into cluster running Integrity
• Demo has three Displays
  • Integrity and Linux share display 1. Display 2 and 3 dedicated to Linux.
• Touch gestures allows Linux apps to be swiped and scaled on screen
**Canvas Demo Overview**

*1: Make the cluster partially transparent in order to see the Linux graphics.

*2: When Linux app windows are swiped to the display shared with Integrity apps they are put in this space.
Enabling Technology

• H/W (Renesas R-Car H3)
  • GPU Virtualisation
    • IPMMU and GPU Multiple Input Ports, OS ID and GPU Scheduler provides OS guest separation and protection
    • Each OS “sees” its own GPU
  • Display compositor layers with per pixel alpha blending (VSPD IP)
    • Each OS has its own dedicated layer to draw into
    • Compositing them so Linux is behind the Cluster in Z-ordering protects the Cluster from being overwritten
    • Per pixel alpha allows flexible combination of the layers
  • Image/video processing and transformation (various IP)
    • Provides smooth application window transformation

• Green Hills Integrity RTOS and Multivisor Hypervisor (HV)
  • Further strengthens guest separation and protection
  • Cluster Safety OS
  • Display Manager provides Display Sharing for display 0
    • Describes display and GPU each OS sees
Positives

• Linux/Android does not need to know about cluster or concern itself about maintaining a high cluster fps

• Safety OS gains protection from being drawn over by Linux, whilst its performance (fps) is maintained

• If HV already present, no extra protocols to invent to get gfx/interaction etc, from Linux to “other OS” and back. Native apps just do their thing.

• HV Display Sharing can provide flexibility on display size and position for each OS, which can be enhanced further by H/W compositor layer alpha blending.
  • E.g. Present window between cluster dials to Linux for it to draw maps, multimedia.

• But like all tech it of course is not a silver bullet for all use cases...
  • Inter-ECU vs consolidated ECU. Display Sharing vs Surface Sharing.
  • Lucky we have a day to discuss tomorrow!
Thank you!

Visit GENIVI at http://www.genivi.org or http://projects.genivi.org
Contact us: help@genivi.org

GENIVI is a registered trademark of the GENIVI Alliance in the USA and other countries.