Low-level IPC / ICC (Inter Chip Communication) Deep Dive

2021-12-06, 10:00 AM CET

Agenda

- Presentation of one option from Thomas Bruss (Renesas)
- Discussion of presentation
- Further discussion, driven by the questions below.

Participants

Adam L, Artem M, Dan M, François O, Oleksandr T, StephenL, Thomas B, Alex A, José B, Bernhard R, Christian S, Matt S

Minutes

Thomas Bruss (RENESAS)
presented current solutions CR7A5x. Including ring-buffer and shared memory solution. VIRTIO layered on top of this. Paravirtualization (modified drivers) on A5X side. A paravirt server on CR7 side takes display output from A5X. MFIS hardware provides support but in this solution it is not used so much - only to generate interrupts. CR7 often booted before A5X (e.g. Linux). For example synchronize on VBLANK interrupts - and a message is also transferred associated with interrupts.

Selection criteria:

- Portable solution - not only Linux.
  - On CR7 is assumed a number of RTOS alternatives, on A5x Linux of course but also QNX, Integrity, T-kernel ... are options. For example, the concept has been implemented on Integrity.
- System should be as "unaware" as possible of the ICC.
  - To make this work CR7SDK A5X, only the Linux kernel driver is modified
- Low latency – for some cases it needs to be as if you had direct hardware access, even though it goes via the ICC, e.g. framebuffer switch on VBLANK
  - Ringbuffer impl. is too much copy overhead for this, and for streaming data cases
  - Therefore pointers (to shared mem) are transferred with the message

Q: (Artem): Is it built on RP message?

A: (Thomas): That's an option but we have a specific virtio setup with two queues.... basically the standard VIRTIO driver doe snot work on top of our protocol without some changes.

François: LINARO is working on OpenAMP+VirtIO

Dan: Working in OpenAMP project with VIRTIO on top of it. Shared mem & notification - similar. Main difference we are exposing full VIRTIO infrastructure = Multiple queue support. This supports additional features compared to RP message.   This is not to suggest that it replaces RP Message completely - it can still be useful for some cases.

Q: ... other services?

A: (Thomas): We have also made a CAN driver implementation.
Gunnar note: - I wanted to ask a follow up question about CAN, at another time.

Discussion on use cases...

Q: How do use-cases drive the chosen solution?
A: E.g. ring-buffer vs shared memory = performance difference.

... slow progress on this side
Gunnar: What about going from the other direct - is there a single solution that ought to fulfil all (conceivable) use cases.
... or can we agree on 2 (3) options - that can be chosen depending on need?
...How close are we to "standardization" of some sort (since VIRTIO is mentioned by several as an appropriate standard protocol to build on.)

Francois: Different variants. Number of affected cache lines. It's a zoo of different interfaces and variations.
.. also don't miss end-to-end performance. Cache-lines usage make a big difference in performance.

Gunnar: OK so analyzing benefits (of each choice) are measurable (only) on the real system workload.
Francois agreed.

(Gunnar: Does it mean we can at least agree on 2-3 options that can be chosen depending on need. And they can be tried against the real workload when the system is built.)

FM tuner use case mentioned.
Is that something to drive the discussion. We could start from that case and drill into to see what understanding we gain from it. It includes audio streams and other features.

Bernhard: But also consider also ADAS use cases, not only IVI/Cluster side. Diagnostic monitoring. CR52 software, how does it affect? (need to seek out details)

... further discussion

Adjourned after 1h – continue where we left of on the same time and place next week.

Discussion questions

What factors affect the choice of IPC / ICC?
- Technology / optimization?
- Tier-1 preference (previous experience)?
- Silicon Vendor preference?
- Why is a certain choice preferred?

Are there any standard ways that can be agreed upon?
Is IC-COM a standard? Are there others?

Does AUTOSAR have specific choices, and how does that apply on non-AUTOSAR systems?

Is diversity/fragmentation of IPC choice a challenge for efficient
development?

What (if anything) is preventing simply agreeing on one choice?

Detailed operation - what and how does it work
e.g. shared memory, buses (SPI..), PCI, ?
Development effort: the driver must be implemented

Flexibility (available implementation) towards upper and lower protocols
(think OSI-stack).
Are higher-level protocols layered on top of the simple communication,
or not? How does the choice of low-level IPC affect that?
In relation to connecting downward to hardware, what are the implementation challenges? Does it depend on particular hardware features (hardware-supported mailboxes, dual-port memories, com. buses,...?)
(E.g. if you choose A, then you have simple integration of technologies
1, 2, and 3, but for choice B available integrations are more limited)

Applicability?
- Communication between different cores on an SoC.
- Inter-ECU? Between SoCs / CPUs.
  E.g. shared PCI network,
  Full network (Ethernet)