RPMsg to accelerate transition between multi-SoC and multi-processor SoC solutions.

Arnaud Poulquier / Loïc Pallardy
STMicroelectronics
Who am I

Arnaud Pouliquen
- Embedded software Engineer at STMicroelectronics on STM32MP1 MPU.
  - Audio technical leader.
  - Coprocessor management technical leader:
    - ensure processors coexistence and inter-communication.
- Contributions:
  - Linux
    - Contributor and maintainer on some ST drivers (asoc / iio / remoteproc / rpmsg).
  - OpenAMP library
    - Contributor to OpenAMP restructuring and footprint reduction.
  - Zephyr
    - Introduction of the stm32mp1 SoC in Zephyr.
    - Contributor/expertise for OpenAMP library integration.
“RPMsg to accelerate transition between muti-SoC and multi-processor SoC solutions.”

- **Muti–SoC solution:**
  - several processors in separate devices
  - communicate together by a physical link (only common serial link is considered)

- **Multi-processor SoC solution:**
  - several processors embedded in one chip
  - hardware mailbox and optional shared memory for inter-processing communication.

- Focus on the inter-processor communication.
Remote Processor Messaging

- ST implementation is based on open source solutions:
  - **VirtIO** and **RPMsg** frameworks for Linux (Ohad Ben-Cohen 2011)
  - **OpenAMP** library for the co-processor (Xilinx and Mentor Graphic 2014).

![Diagram of Remote Processor Messaging](attachment:image.png)

**Communication link** (shared memory + inter-core Interrupts)

**Transport layer**

**Virtio/virtqueue**

**Mac layer**

**Physical layer**

**RPMsg structure**

- **Source address** (32 bits)
- **Destination address** (32 bits)
- **Reserved** (32 bits)
- **Length** (16 bits)
- **Flags** (16 bits)
- **Payload**
Remote Processor Messaging

- One physical link can offer multiple communication channels.
- A channel implements a service relying on one or several end-points.
- An end-point provides a logical connections through a channel.

![Diagram showing Master and Slave with channels and end-points](image)
Motivation

- Preserve applications but migrate from a serial link to RPMsg.
Motivation

- Extend platform by offloading some services on an external coprocessor.
- Manage diversity of services on a single link.
Migrate to an internal co-processor

Serial link

Virtual serial link over RPMsg
Implementation status

• **Linux developments**
  - Virtual serial drivers over RPMsg implemented.
    - rpmsg_tty driver
    - rpmsg_i2c driver
    - rpmsg_spi driver

• **Co-processor firmwares**
  - Virtual drivers over RPMsg and associated examples available:
    - on STM32Cube
    - on Zephyr
Extend to an external co-processor
Co-processor

application

rpmsg
rpmq virtio bus
virtio

Shared memory

rpmsg

application

rpmsg
rpmq UART bus
rpmq UART Peripheral

uart API
uart controller

UART Peripheral
Implementation status?

- **Linux developments**
  - RPMMsg serial buses:
    - RPMMsg serdev bus implemented,
    - RPMMsg I2C/SPI bus not implemented

- **Co-processor firmwares**
  - Examples on STM32Cube distribution and Zephyr available
  - Serial buses not yet implemented in OpenAMP library.
What’s next (current target)

- Upstream of the Linux RPMsg clients drivers and RPMsg serial buses.
- Support serial buses in OpenAMP.
- Provide support of the virtual serial drivers in different ecosystems (STM32Cube, Zephyr, arm® Mbed, Arduino…).
Thank you!

Questions ? Suggestions, comments?

Arnaud Pouliquen  
arnaud.pouliquen@st.com

Loïc Pallardy  
loic.pallardy@st.com

Meet us and see the demos at the ST Booth!