

Unikernels - RTOS combination (Motivation)

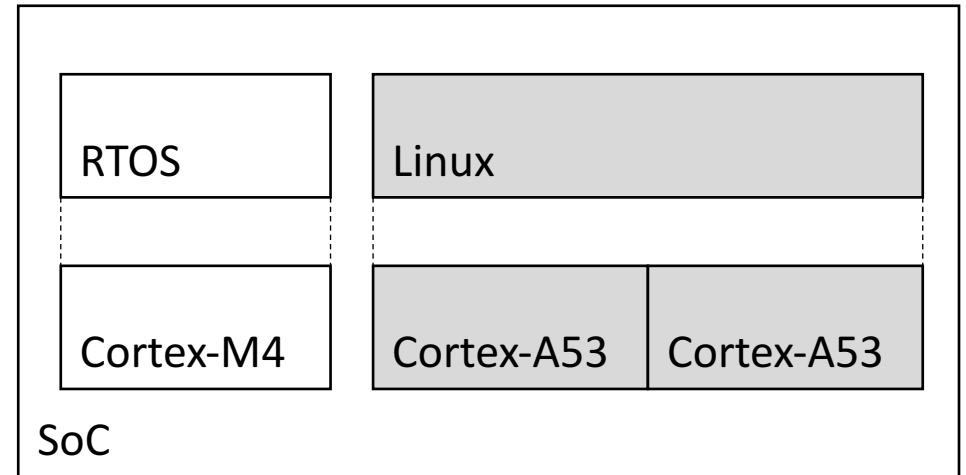
■ Background

➤ RTOS + Linux on a SoC (System on a Chip) with heterogeneous cores is becoming attractive!

- Real time processing (RTOS)
- A variety of APIs, software and devices (Linux)
- Cost effective (SoC)

➤ Requirements

- RTOS users often want to give RTOS a master role
→ Linux is only a slave OS
- RTOS should be able to manage life cycle of the slave OS



Ex.) NXP i.MX8 series

STMicroelectronics STM32MP1 series

■ Problems

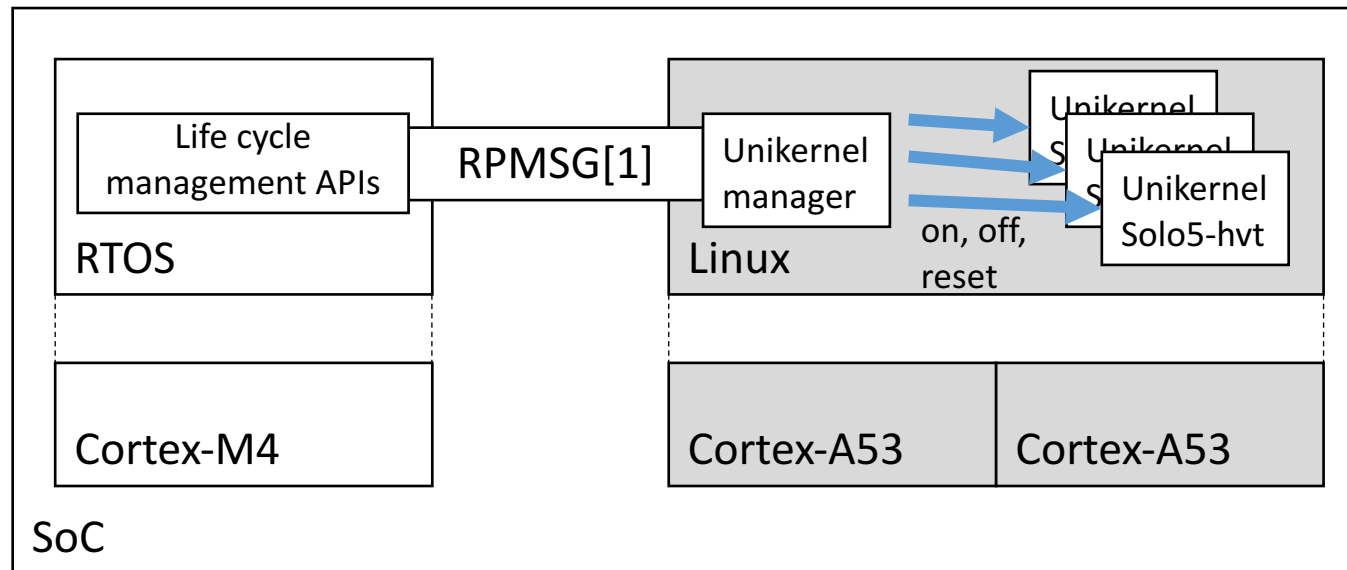
➤ OSes - SoC dependency if we use Linux as a slave OS

- Slave OS shutdown (or reboot) causes overall system shutdown (or reboot)
→ Of course, RTOS will be put into the shutdown state too!!

➤ Difficult to remove or mitigate the dependency from the hardware point of view

Unikernels - RTOS combination (Solution)

- Employing Unikernel-based slave OSes
 - Life cycle management can be quite easy
 - Unikernel shutdown (reboot) does not affect RTOS operation
 - Small and isolated environments for RTOS and Unikernels
 - Good for IoT edge devices



[1] <https://www.kernel.org/doc/Documentation/rpmsg.txt>

Prototype

- SoC

- STMicroelectronics STM32MP157A

- Board: <https://www.st.com/en/evaluation-tools/stm32mp157a-dk1.html> (not so expensive 😊)

- RTOS

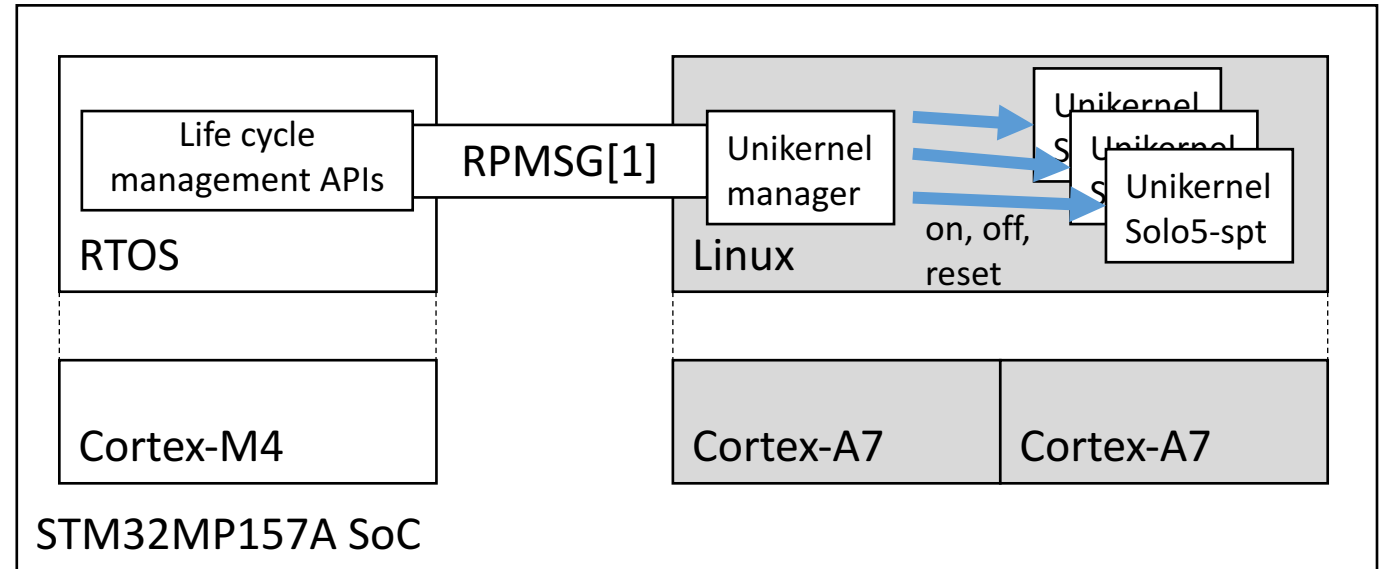
- FreeRTOS

- Linux

- 32-bit kernel
without virtualization support

- Unikernels

- running on Solo5-spt



[1] <https://www.kernel.org/doc/Documentation/rpmsg.txt>