i.MX 7 - Hetereogenous

Multiprocessing Architecture

Overview Toradex

Innovative Business Model

- Independent Companies
- Direct Sales
- Publicly disclosed Sales Prices
- Local Warehouses
- In-house HW and SW Development
- Free Support by the Developers
- 10+ Year Product Life Cycle

Toradex Product Families

Two complementary, pin-compatible product families targeting industrial applications:

Apalis: MXM3-type small-form-factor SOM

- High-performance product family
- High-speed interface support: PCIe, GigE, SATA, USB 3.0, HDMI, eDP, MIPI CSI/DSI, etc.
- Advanced multimedia support with HD 3D Graphics and multiple Display and Camera ports

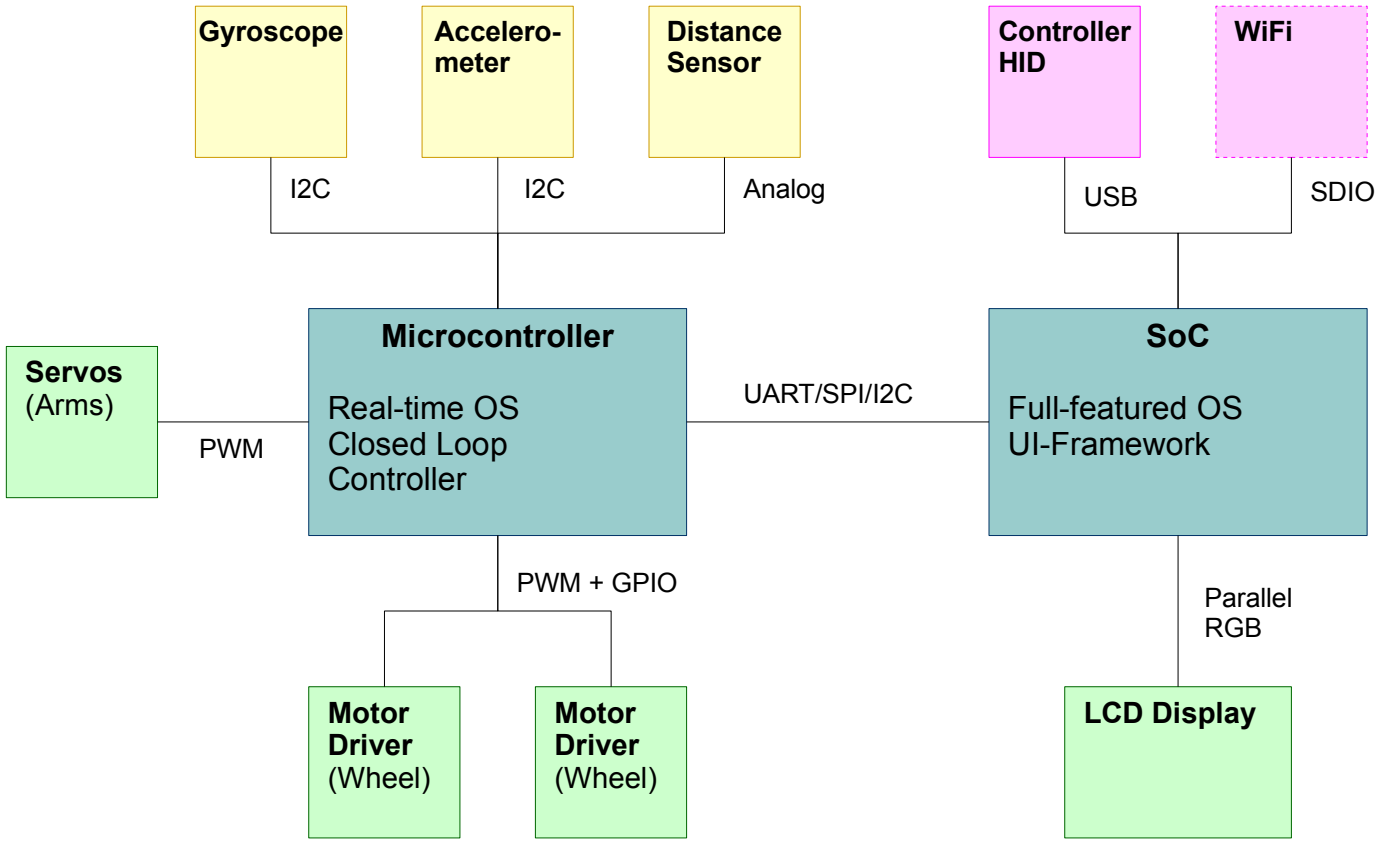
Colibri: SODIMM small-form-factor SOM

- Cost/performance optimized product family
- Common industrial interface support: Ethernet, UART, SPI, I2C, RGB/LVDS display, etc.
- Hugely popular for over 10 years

Build a self-balancing Robot with a Graphical User Interface

- Requirements
  - Robot needs to balance on two wheels
  - User Interface should act on changes in balance
- Challenges
  - Balancing… Inertial sensors, motor control
    Closed loop controller (real-time)
  - User Interface...
    Resolution, Animations…
  - Communication?

Robot: Traditional approach



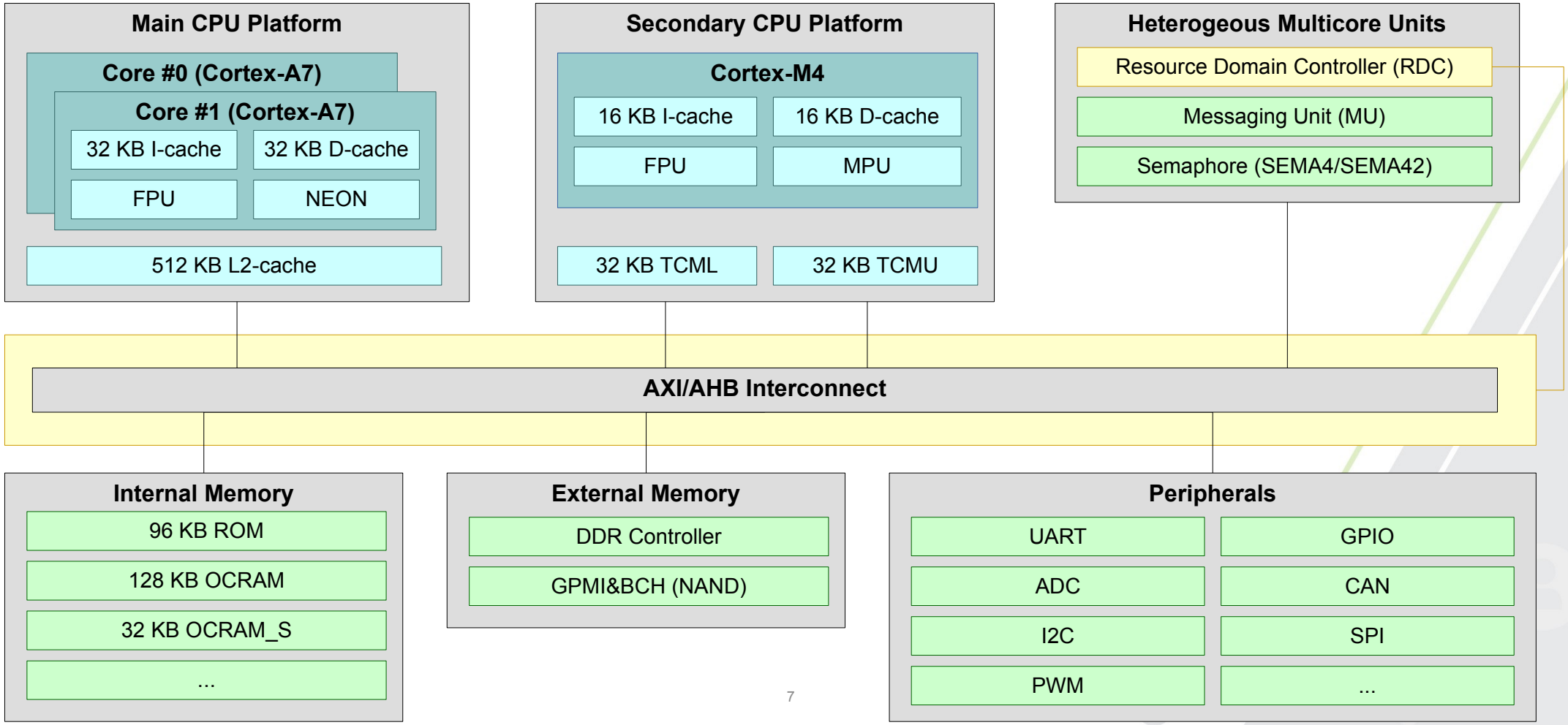
Toradex
Embedded. Computing.

NXP i.MX 7 Heterogeneous Multiprocessing Architecture

- Multiprocessing
  Two independent CPU cores in a single Chip
- Heterogeneous
  Do not use the same ISA (almost though)

- ARM Cortex-A7@1GHz (two in the i.MX 7D case)
- ARM Cortex-M4@200MHz
- Shared Bus Topology

NXP i.MX 7 Heterogeneous Multiprocessing Architecture



NXP i.MX 7 HMP Applications

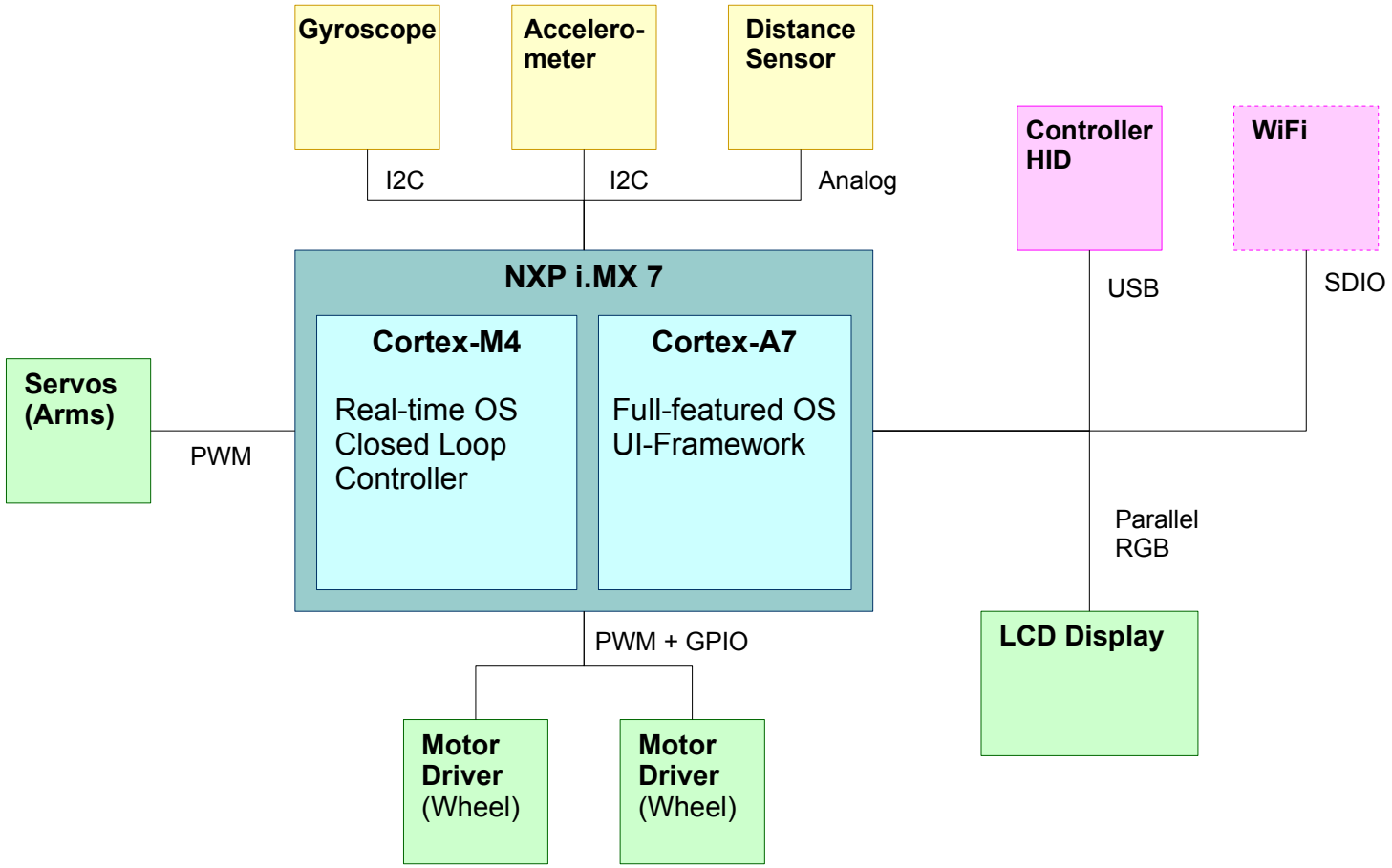
- Offload Real-time Tasks
  - Real-time Operating System on Cortex-M4
- Optimize Power
  - Powerdown main CPU and its peripherals
  - Keep DDR memory in Self-Refresh
- Increase Security
  - Run Secure world only on one CPU complex
  - Hardware partitioning of peripherals

Toradex
Embedded. Computing.

NXP i.MX 7 HMP Advantages

- Fast communication between the two systems
- Simple Firmware upgrade
- Less Hardware
  - Cost savings
  - Availability
- Dynamic hardware
  - Reallocation of Devices as Software evolves

Robot: HMP Approach

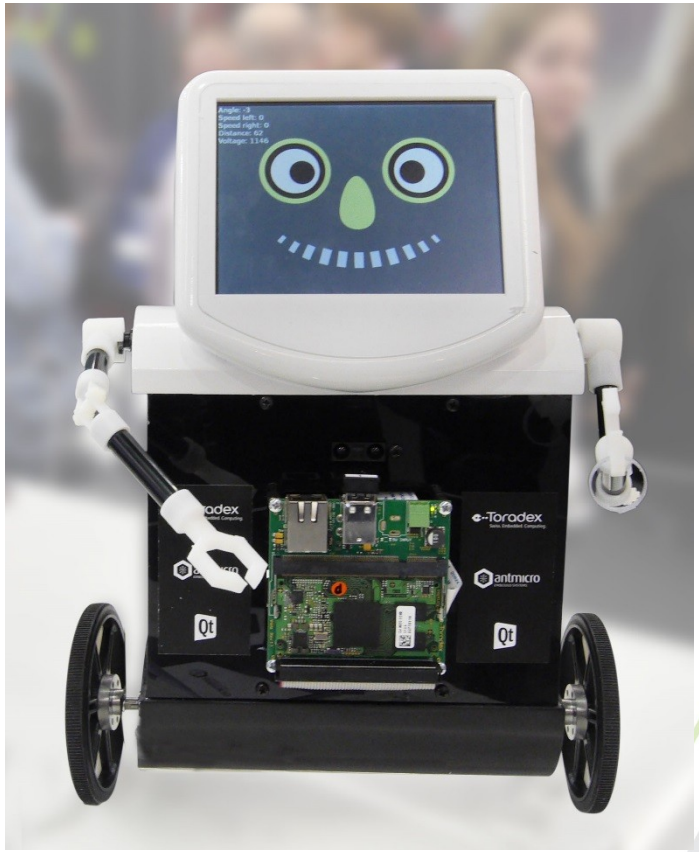


Balancing Robot “TAQ”

Toradex
Swiss. Embedded. Computing.

antmicro
EMBEDDED SYSTEMS

Qt The Qt Company



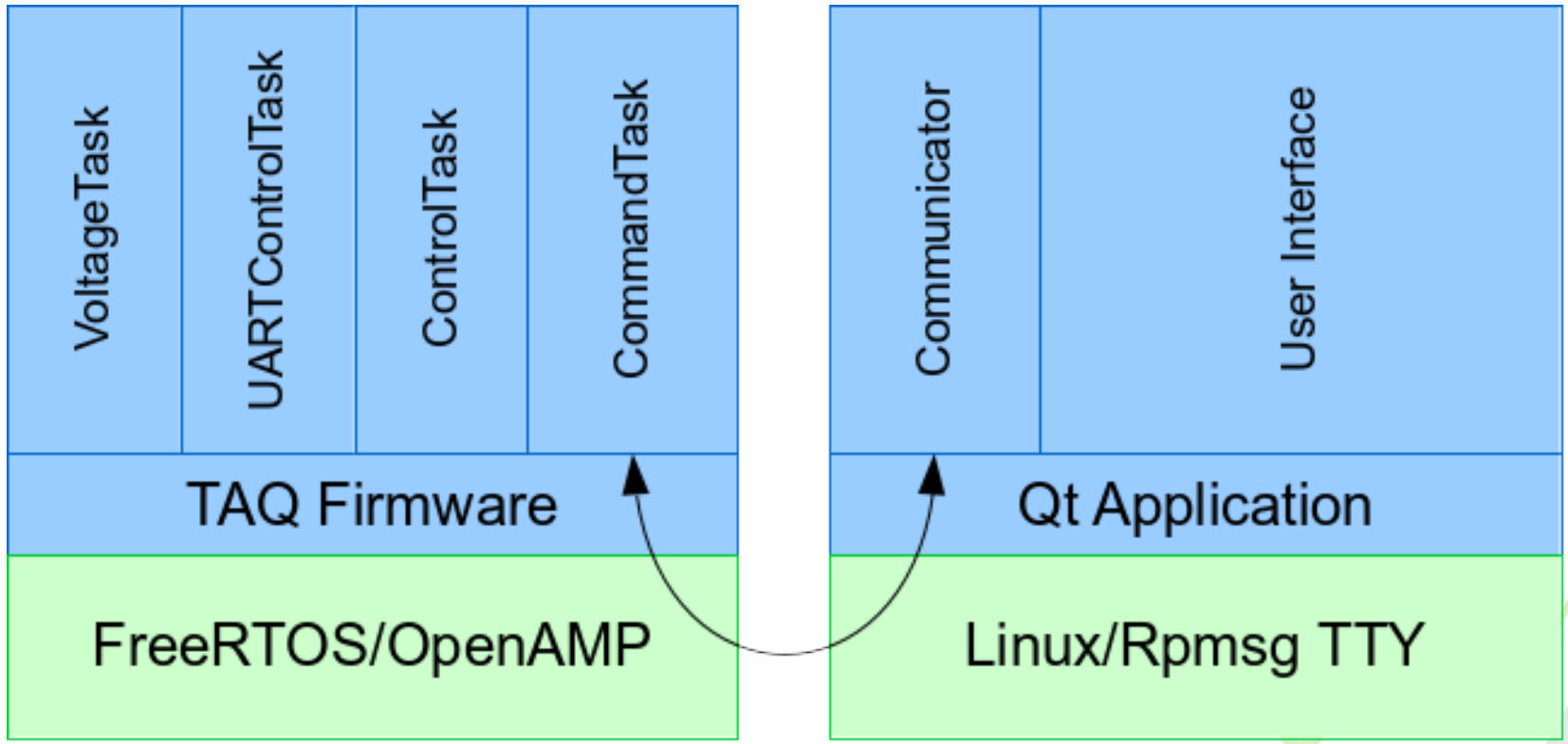
Toradex
Embedded. Computing.

Demo

Robot: Software Stack

- Main CPU Platform
  - Linux/Boot2Qt root file system
  - Qt 5 User Interface
- Secondary CPU Platform
  - FreeRTOS with Custom Firmware
- Communication
  - Linux Rpmsg with Rpmsg based TTY driver
  - FreeRTOS OpenAMP (bare-metal Rpmsg implementation)

Robot: Software Architecture



Robot: Simple Line-Based Protocol

- Implemented commands:

```
?angle
?speedLeft
?speedRight
?distance
?voltage
!move value
!turnLeft value
!turnRight value
!stop
!servo nr, value
```

HMP History

- NXP Vybrid: MQX, MCCv1 (Multi-Core Communcation)
  - Kernel driver
  - User-space Library
  - CPU2CPU interrupts
- NXP i.MX 6 SoloX: MQX, MCCv2
  - Using Messaging Unit (MU)
- NXP i.MX 7: FreeRTOS, Rpmsg
  - Kernel drivers

Boot flow

- The system (boot ROM) always starts on Cortex-A7
- Cortex-M4 is clock gated
- Vector base is in OCRAM_S (0x00180000)
- U-Boot implements custom command “bootaux”
  - `fatload mmc 0:1 0x7F8000 hello_world.bin`
  - `dcache flush`
  - `bootaux 0x7F8000`
- Image Format:
  - First word: Initial stack pointer
  - Second word: Entry point

Toradex
Embedded. Computing.

Software Stack in detail

- Linux Kernel
  - NXP Beta BSP Linux 3.14.52 based (+ ~2.4k patches)
  - NXP Final BSP Linux 4.1.15 based (+ ~1.6k patches)
- FreeRTOS
  - FreeRTOS v8.0.0 kernel
  - Various build systems supported (DS-5, CMake + gcc, IAR)
  - Modified version of OpenAMP
  - Drivers: ADC, GPIO, I2C, UART, WDOG, ECSPI, FlexCAN
- Some Mainline Support (starting with 4.4)

Remoteproc/Rpmsg Frameworks

- Upstream Linux Frameworks to support HMP systems
- Introduced in Linux 3.3 around 2011 (Ohad Ben-Cohen)

- Remoteproc: Remote Processor management Framework
  - Currently *no* implementation for i.MX 7
  - Loading firmware
  - Starting/Stopping the (remote) Processor
  - Allocating and providing resources (e.g. memory)
  - Setting up VirtIO devices

Rpmsg: Remote Processor Messaging Framework

- Used as communication Framework for i.MX 7
- Rpmsg is a VirtIO driver (`VIRTIO_ID_RPMSG 7`)
- Bus driver (in the Linux driver framework)
  - Each endpoint has a callback
  - Endpoint created for each Rpmsg driver
  - Created with control messages in mind
  - Fixed message size of 512 bytes
- i.MX 7 specific VirtIO driver, making use of Messaging Unit

Rpmsg: Drivers

- 2 Rpmsg based Kernel drivers provided by NXP
- Pingpong (`CONFIG_IMX_RPMSG_PINGPONG`, drivers/rpmsg/imx_rpmsg_pingpong.c)
  - In Kernel test driver
  - Sends back a message upon message reception
- Serial Port (`CONFIG_IMX_RPMSG_TTY`, drivers/rpmsg/imx_rpmsg_tty.c)
  - TTY driver (`/dev/ttyRPMSG`)
  - One message per write
  - Use it like a regular TTY (open/read/write/close)

Rpmsg: OpenAMP

- Rpmsg implementation for bare-metal firmware and RTOS
- Started by Mentor Graphics, Xilinx and NXP
- Open source project on Github
  https://github.com/OpenAMP
  - BSD License
  - VirtIO implementation of FreeBSD
- RTOS-aware extension of RPMsg API by NXP
  - Blocking, sequential API
  - Like sockets… but locally… and zero-copy!

Rpmsg: Example OpenAMP/RTOS API

```
struct remote_device *rdev = NULL;
struct rpmsg_channel *app_chnl = NULL;
void *rx_buf;
int len;
unsigned long src;

/* Init */
rpmsg_rtos_init(0, &rdev, RPMSG_MASTER, &app_chnl);

/* Receive */
rpmsg_rtos_recv_nocopy(app_chnl->rp_ept, &rx_buf, &len, &src, 0xFFFFFFFF);

/* Process data in rx_buf... just sending them back... */
tx_buf = rpmsg_rtos_alloc_tx_buffer(app_chnl->rp_ept, &size);
memcpy(tx_buf, rx_buf, len);
rpmsg_rtos_send_nocopy(app_chnl->rp_ept, tx_buf, len, src);

/* Free Receive buffer */
rpmsg_rtos_recv_nocopy_free(app_chnl->rp_ept, rx_buf);
```

Resources

- Balancing Robot “TAQ” Firmware
  https://github.com/antmicro/imx7-taq-demo (soon)
- Balancing Robot “TAQ” User Interface
  https://github.com/mitchcurtis/robot-faces
- Toradex Developer Article
  FreeRTOS on the Cortex-M4 of a Colibri iMX7
- Linux Kernel Sources
  Documentation/rpmsg.txt
- OpenAMP
  https://github.com/OpenAMP
- FreeRTOS BSP (including NXP documentation)
  http://git.toradex.com/cgit/freertos-toradex.git/
  Release Notes, Getting Started, API Reference Manual

Toradex
Embedded. Computing.

Stefan Agner
stefan.agner@toradex.com
+1 (206) 319-5616