QtCon BR

Developing Embedded Solutions on Asymmetric Systems using QT

Fernando Luiz Cola
fernando.cola@emc-logic.com

Whoami?

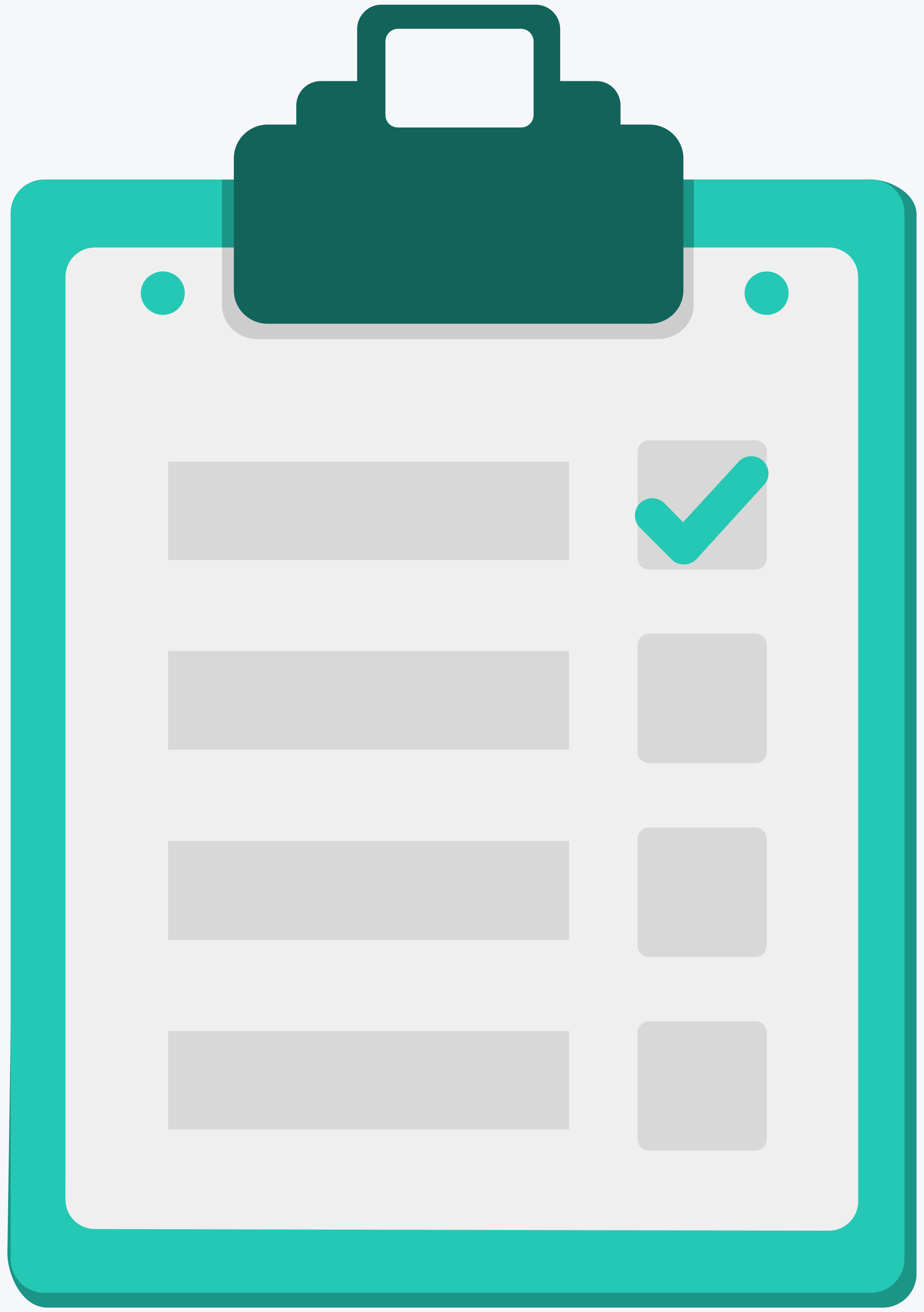
Fernando Luiz Cola

Embedded Software Developer
Emc Logic

emc logic;

fernando.cola@emc-logic.com

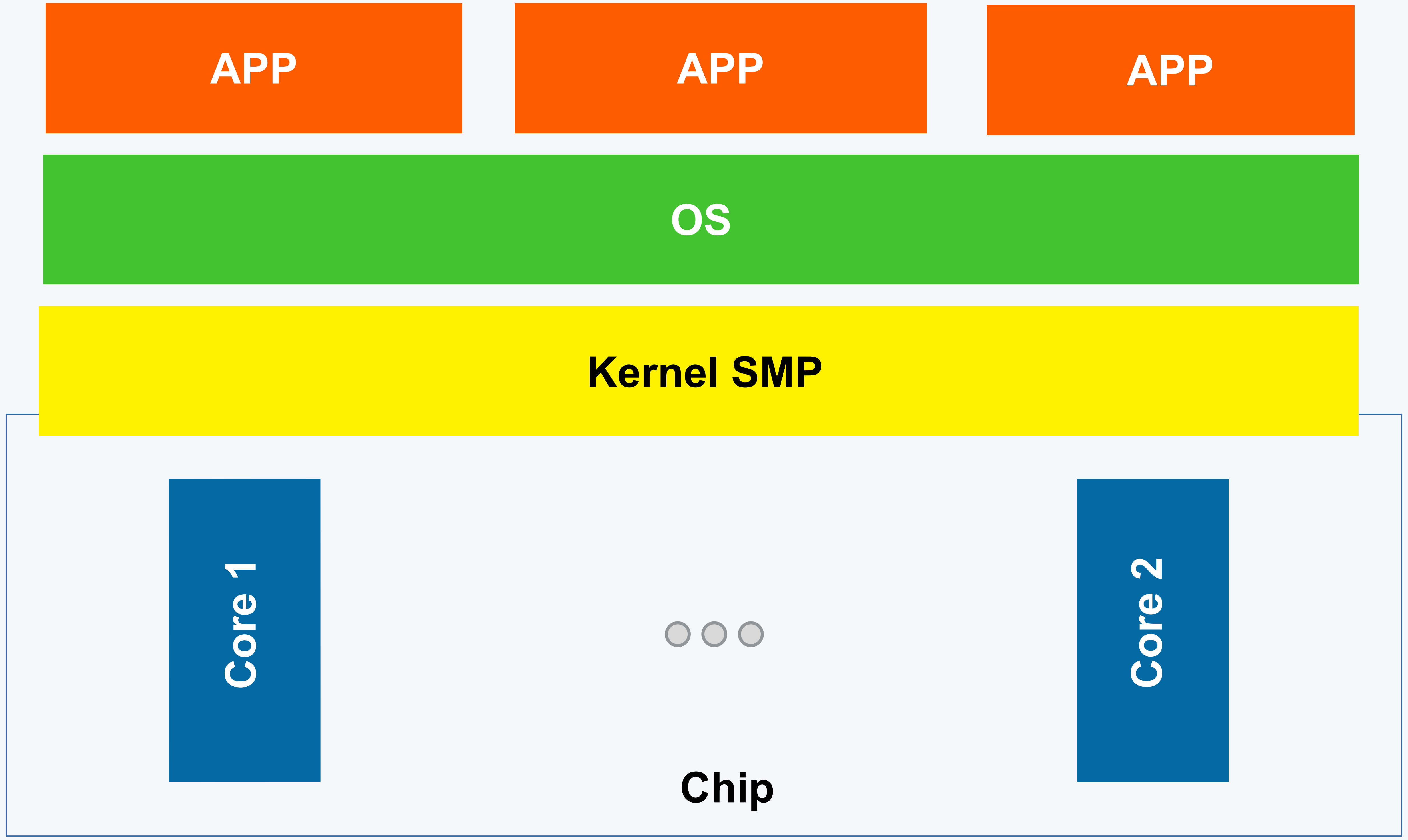
Agenda / Objectives



1. Understand AMP and SMP architecture
2. Applications and solution where AMP is a good fit
3. Overview I.MX7 Processor
4. RPMSG and Inter Processor Communication
5. Bottom-up example application with Qt / Linux / FreeRTOS

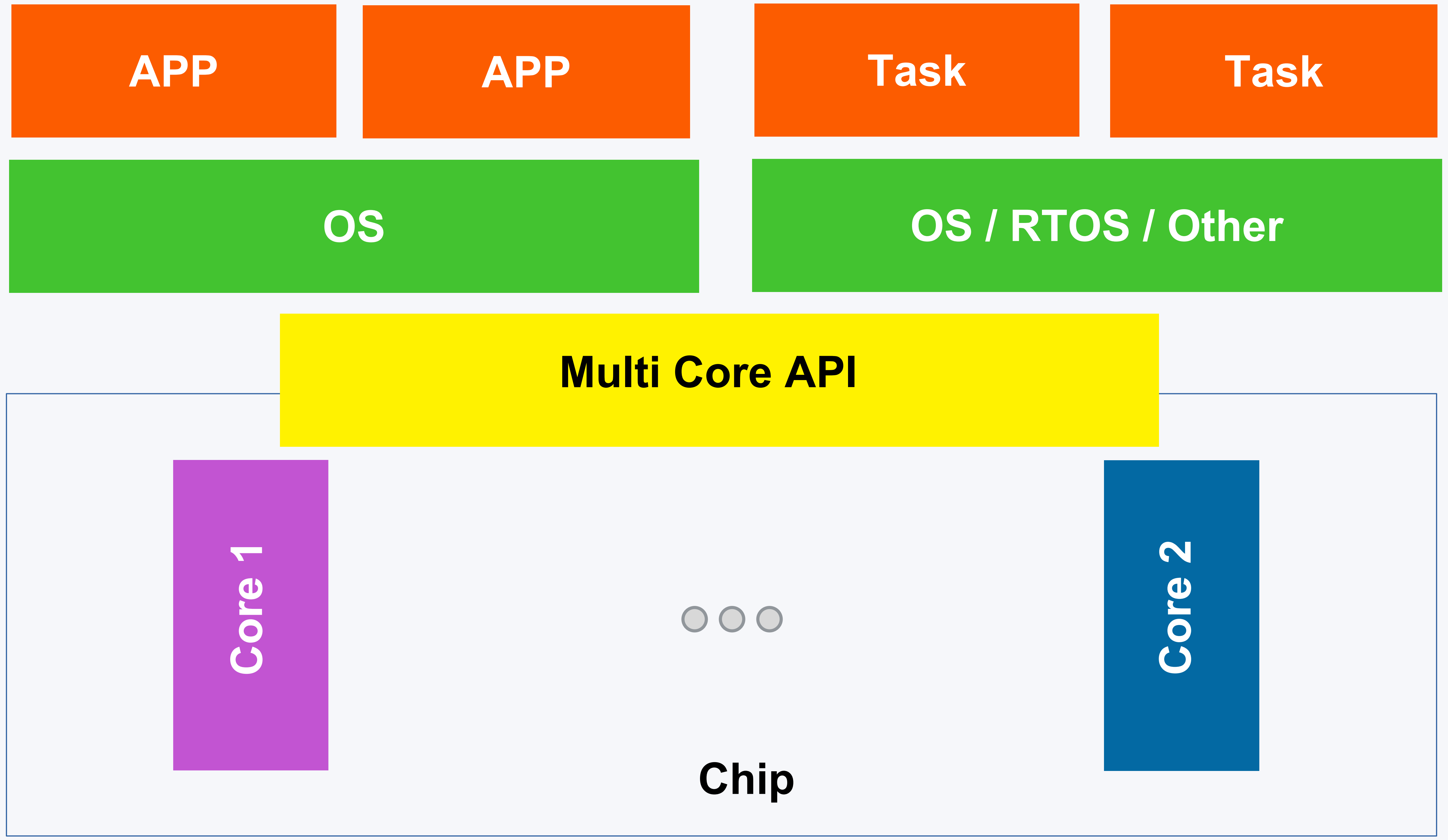
Introduction AMP vs SMP

Symmetric Multi-Processing

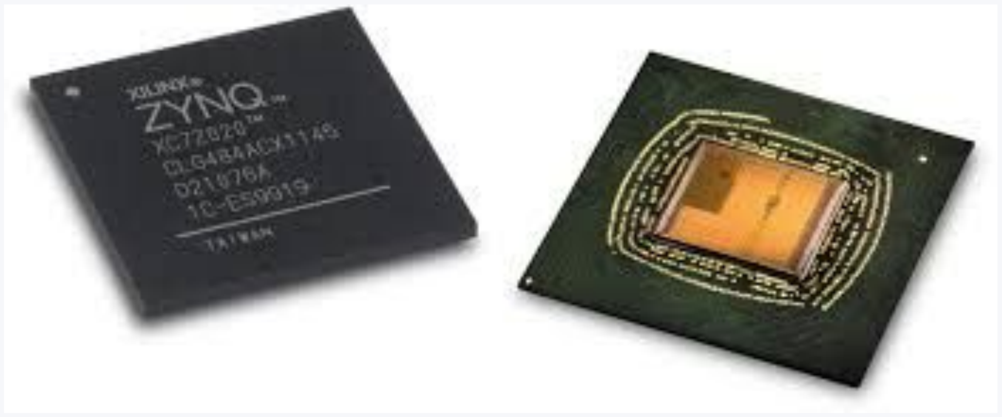


- Single OS controlling two or more cores of the same architecture
- CPU shares memory space
- Dynamic Scheduling e load balancing

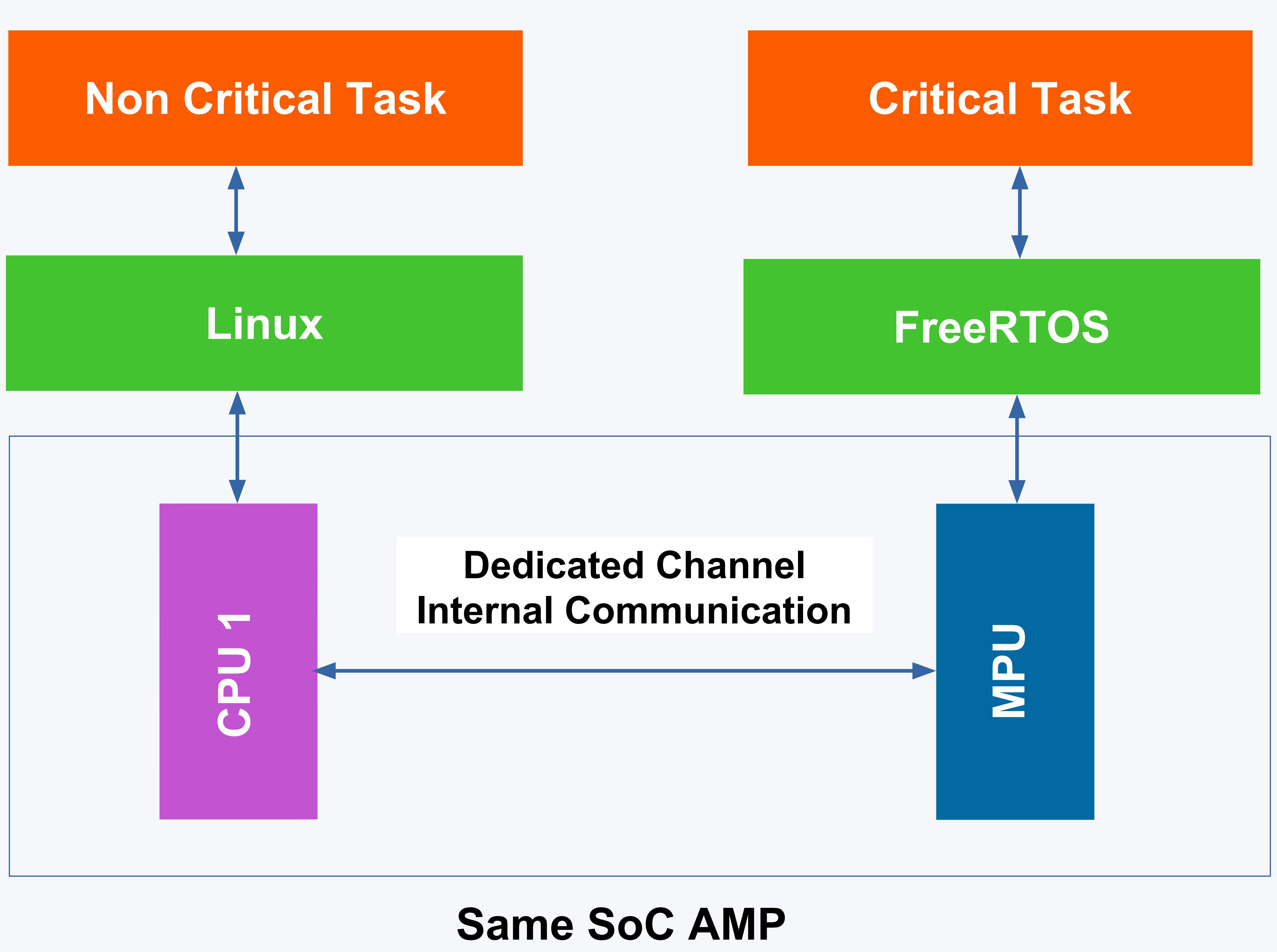
Asymmetric Multi-Processing

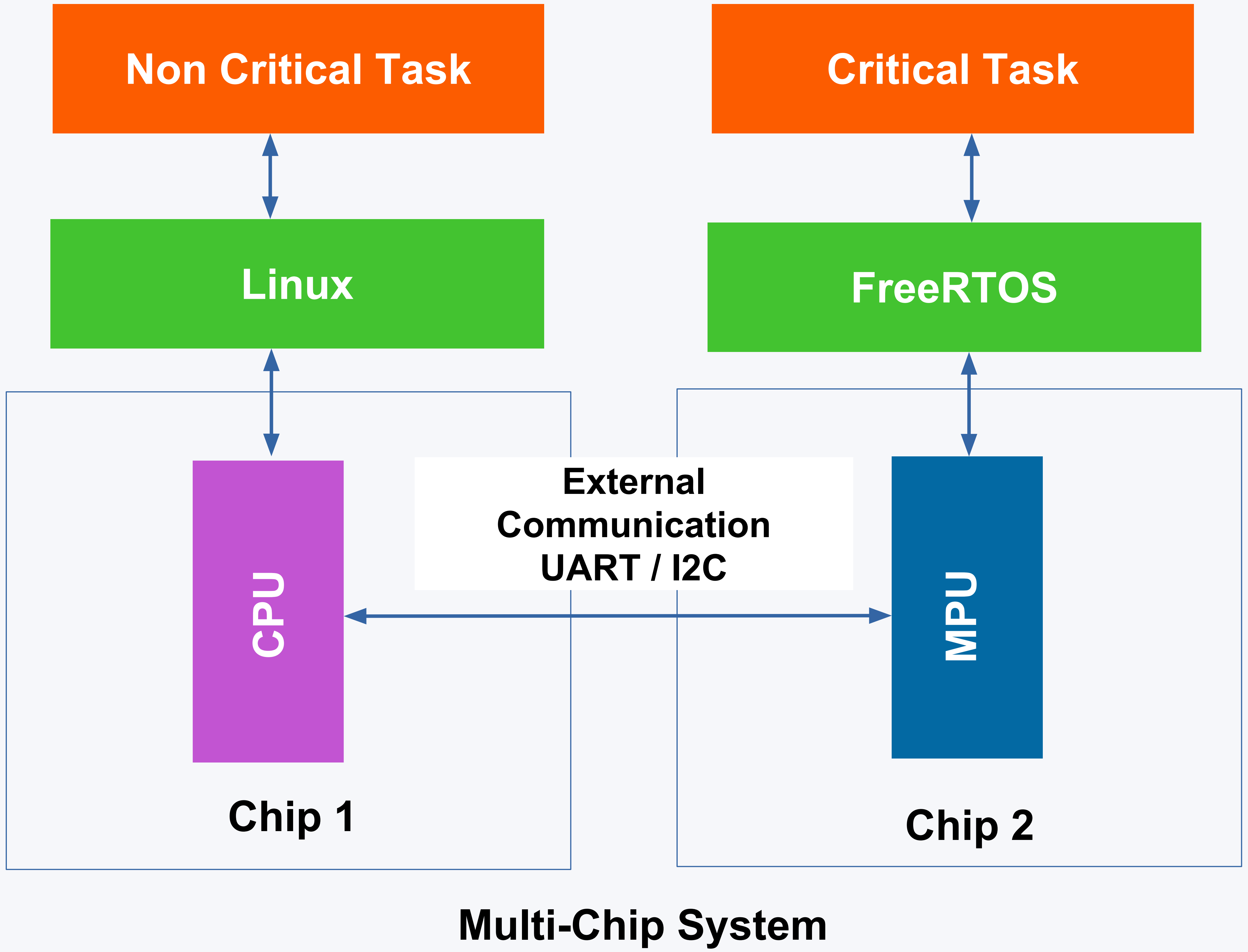


- Different OS on each core
- Different core architectures
- Each core may run full-feature OS, Real-time OS or baremetal code
- Inter process communication protocol
- Efficient when the application can be statically partitioned across cores

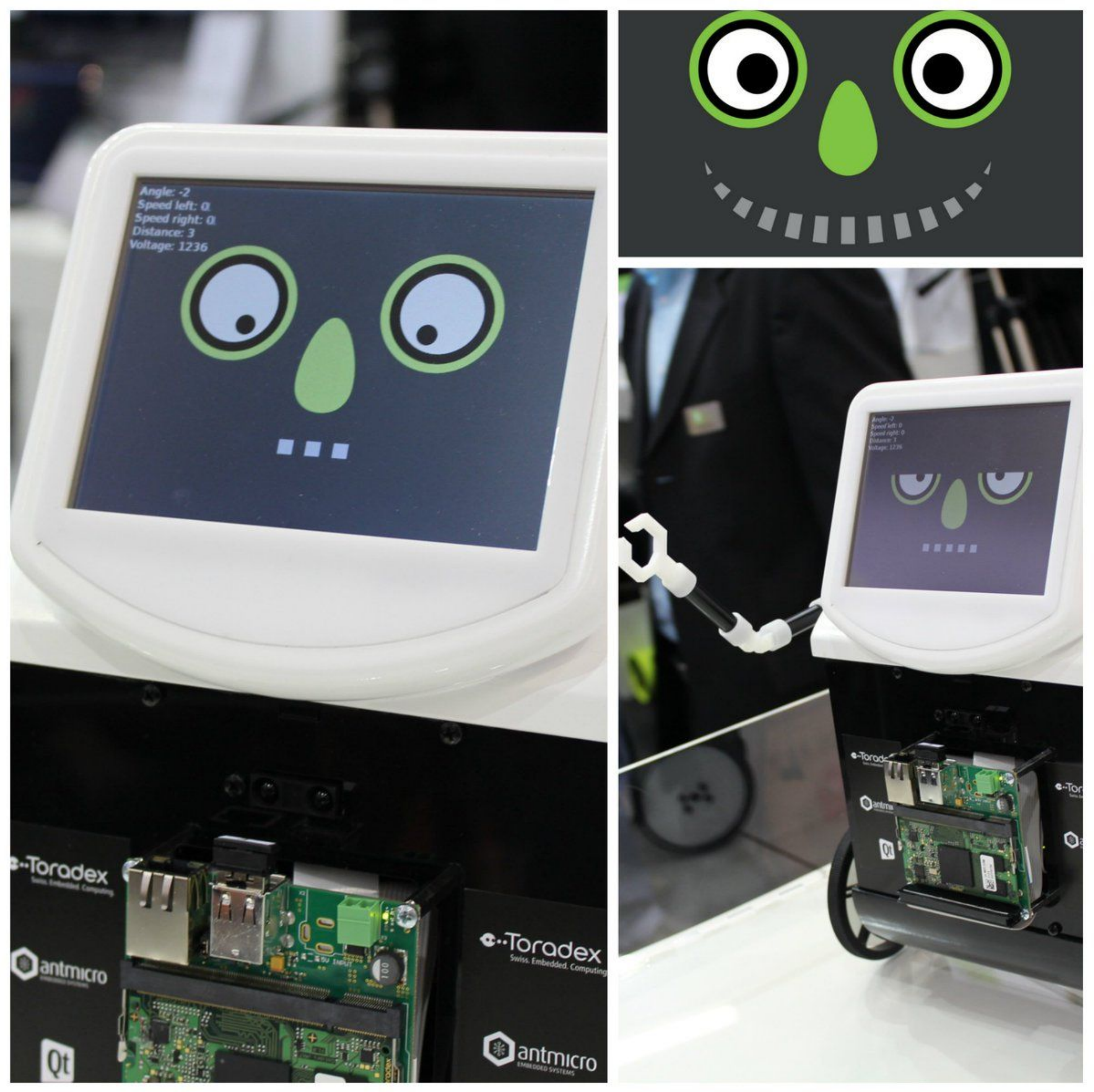


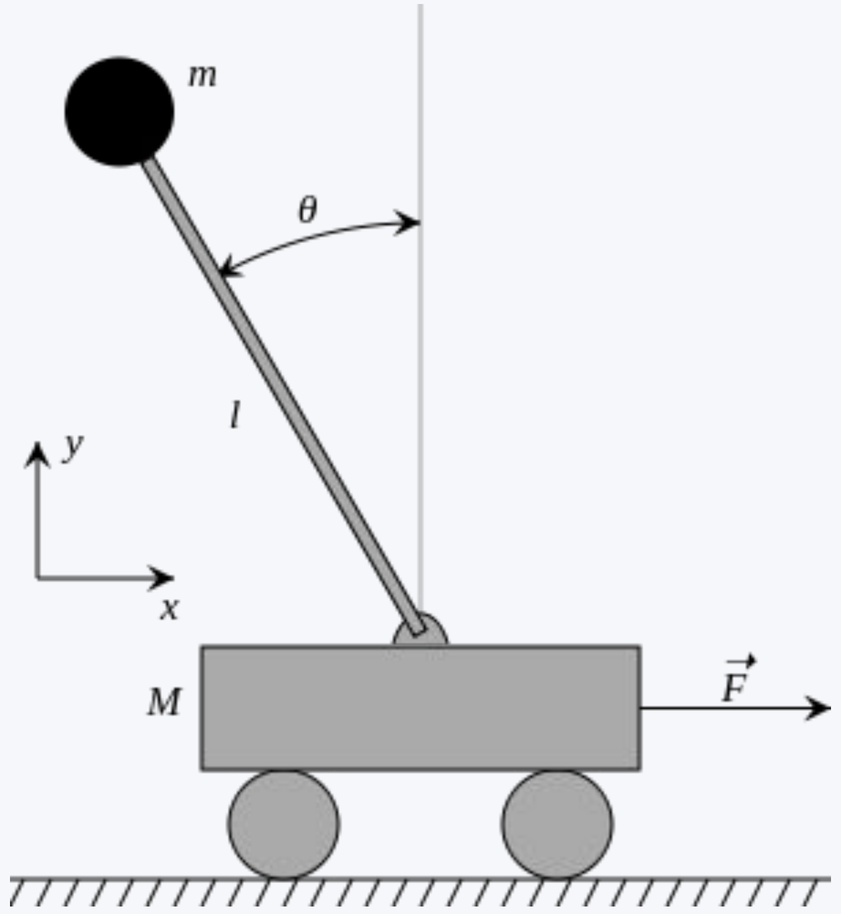
Example

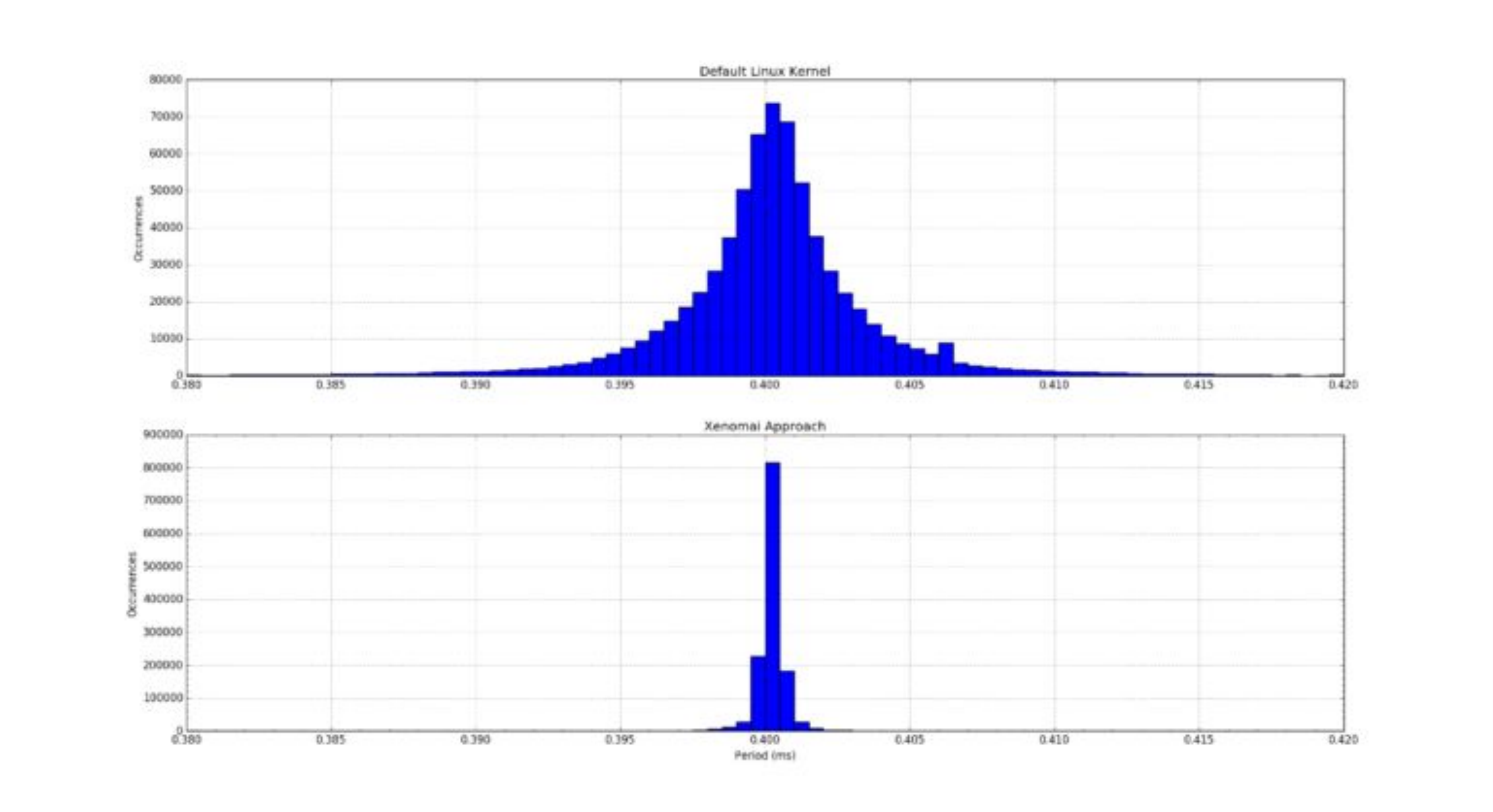

Non Critical Task
Critical Task
Linux
FreeRTOS
CPU 1
Dedicated Channel
Internal Communication
MPU
Same SoC AMP


Non Critical Task
Critical Task
Linux
FreeRTOS
CPU
External
Communication
UART / I2C
MPU
Chip 1
Chip 2
Multi-Chip System

Applications Examples

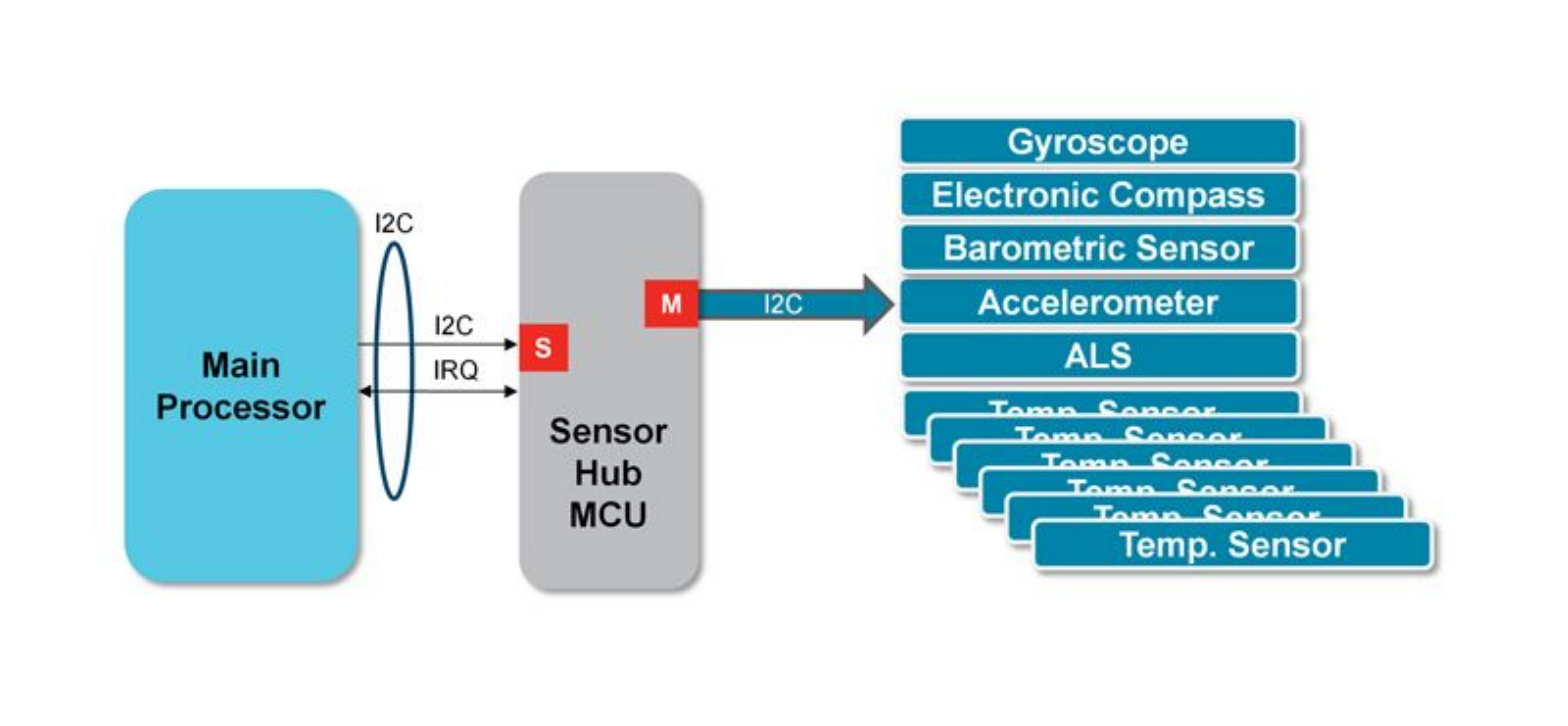


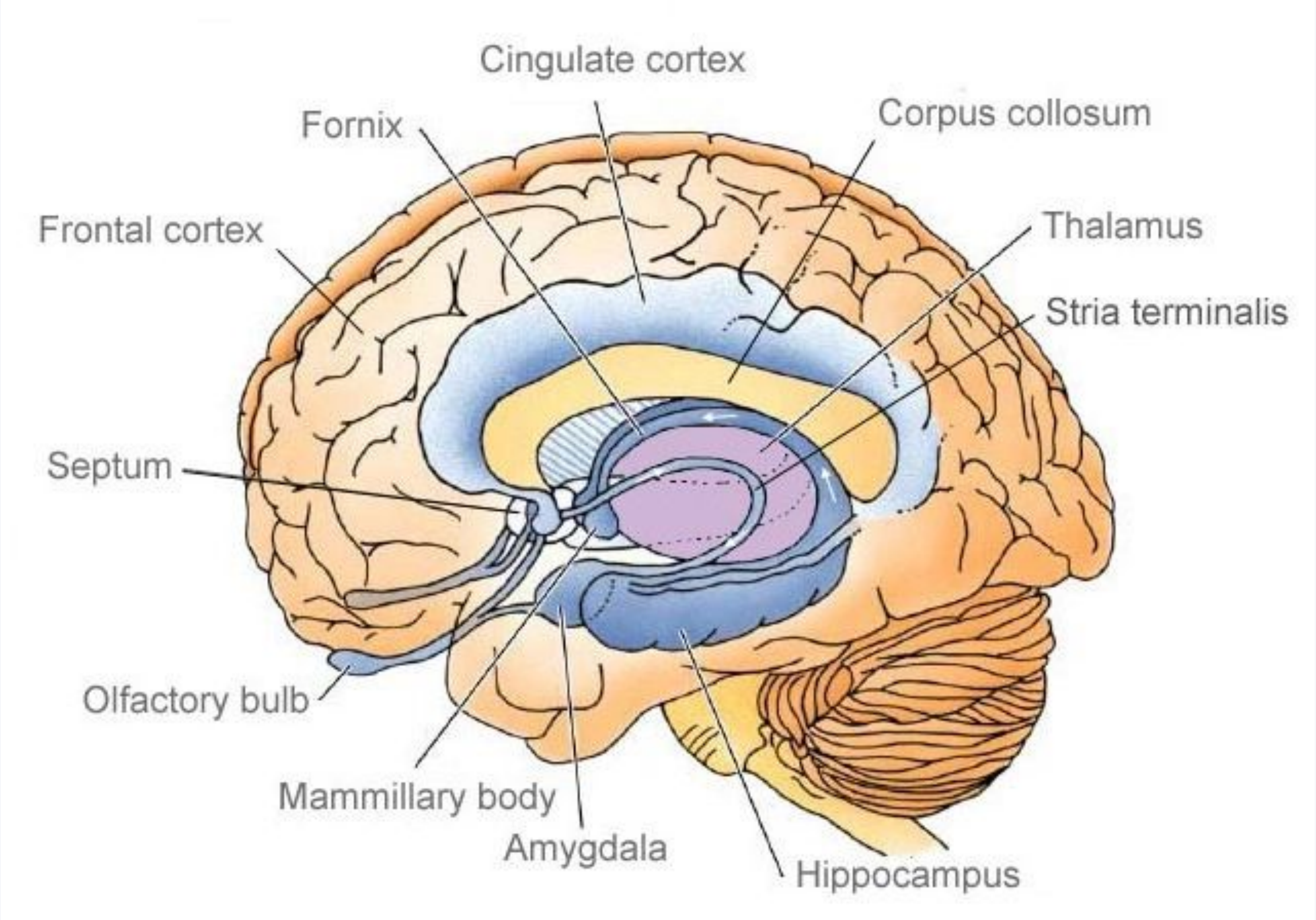




Robotics / Real-Time

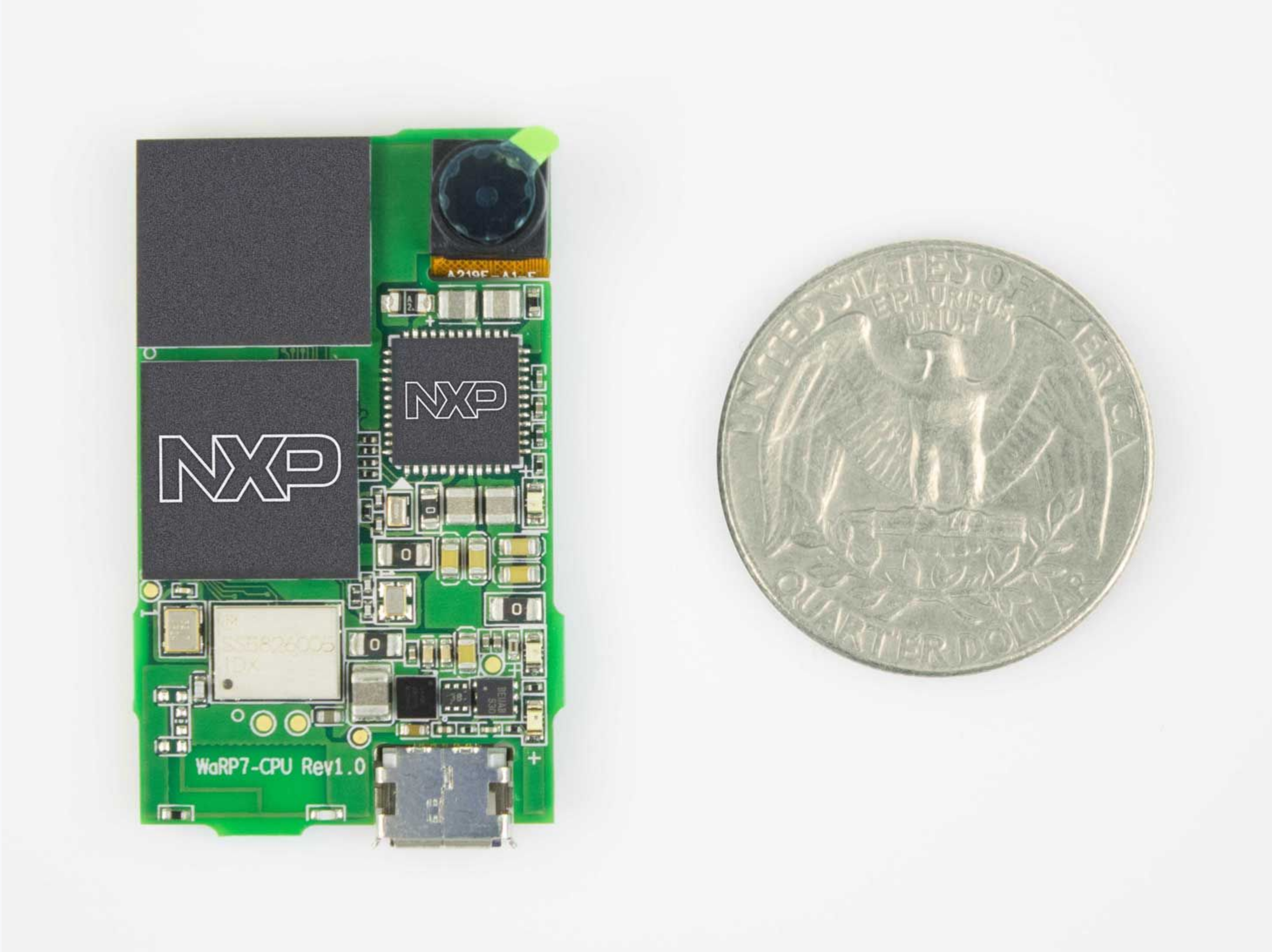
Applications Examples





Mobile / Sensor Acquisition

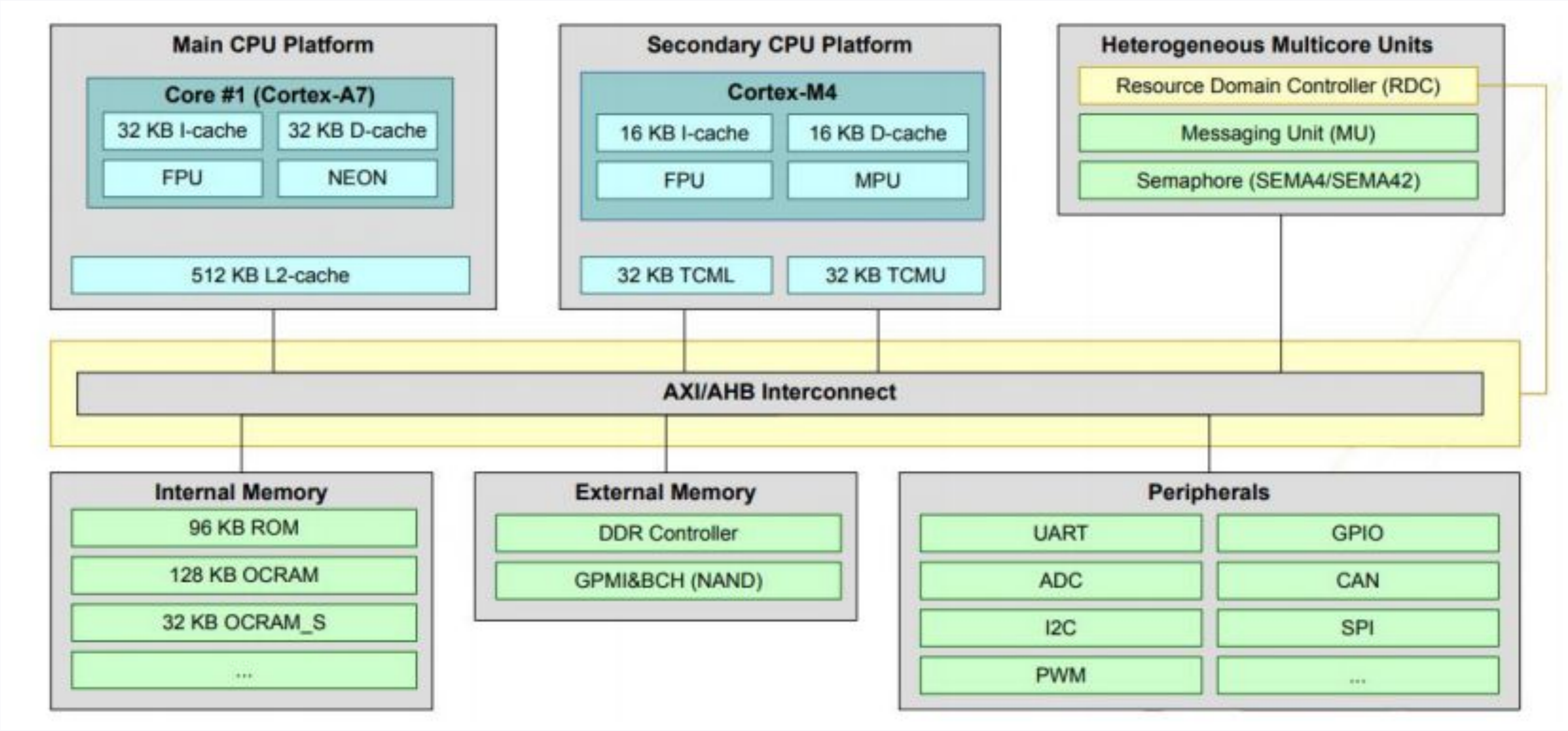
Applications Examples



Wearable / Low-Power

Overview I.MX7 Architecture

NXP I.MX7 Overview



- **Dual Cortex A7 core + Cortex M4 core**

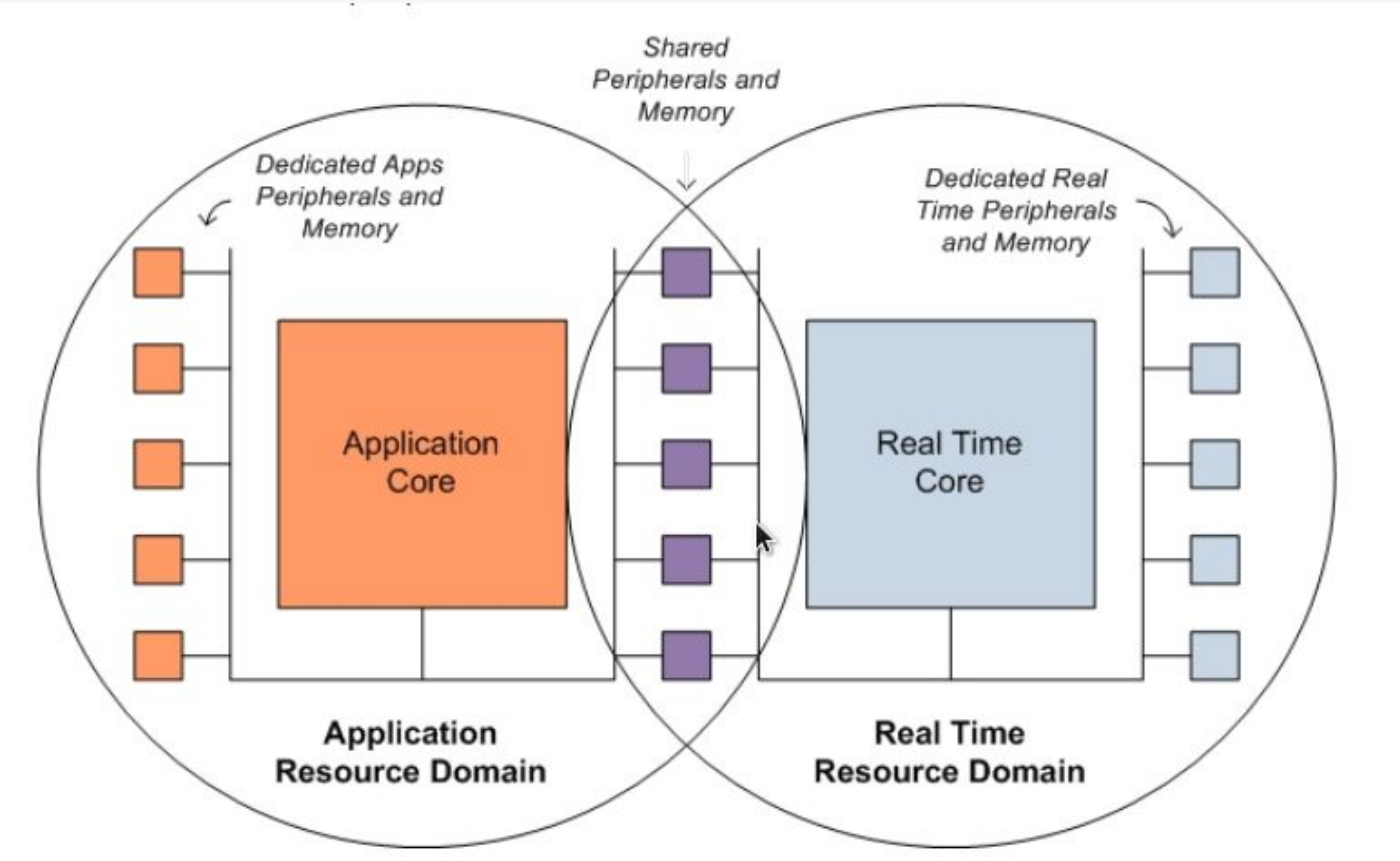
- Master/Slave architecture
- A7 is the master
- M4 is the Slave

- Inter processor communication

- **MU** – Message Unit
- **RDC** – Resource Domain Controller

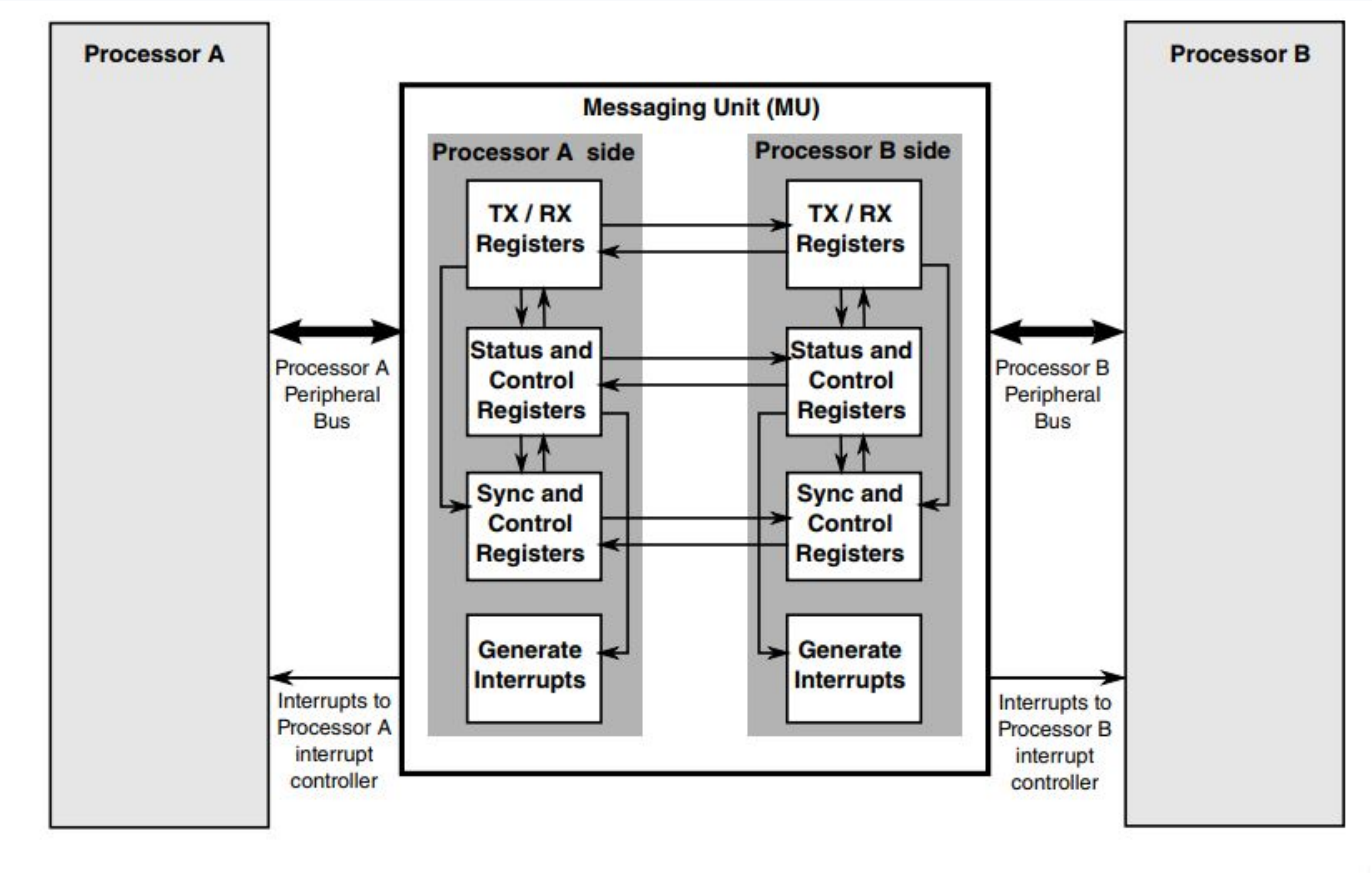
NXP I.MX7 Overview

RDC – Resource Domain Controller


Shared Peripherals and Memory
Dedicated Apps Peripherals and Memory
Dedicated Real Time Peripherals and Memory
Application Core
Real Time Core
Application Resource Domain
Real Time Resource Domain

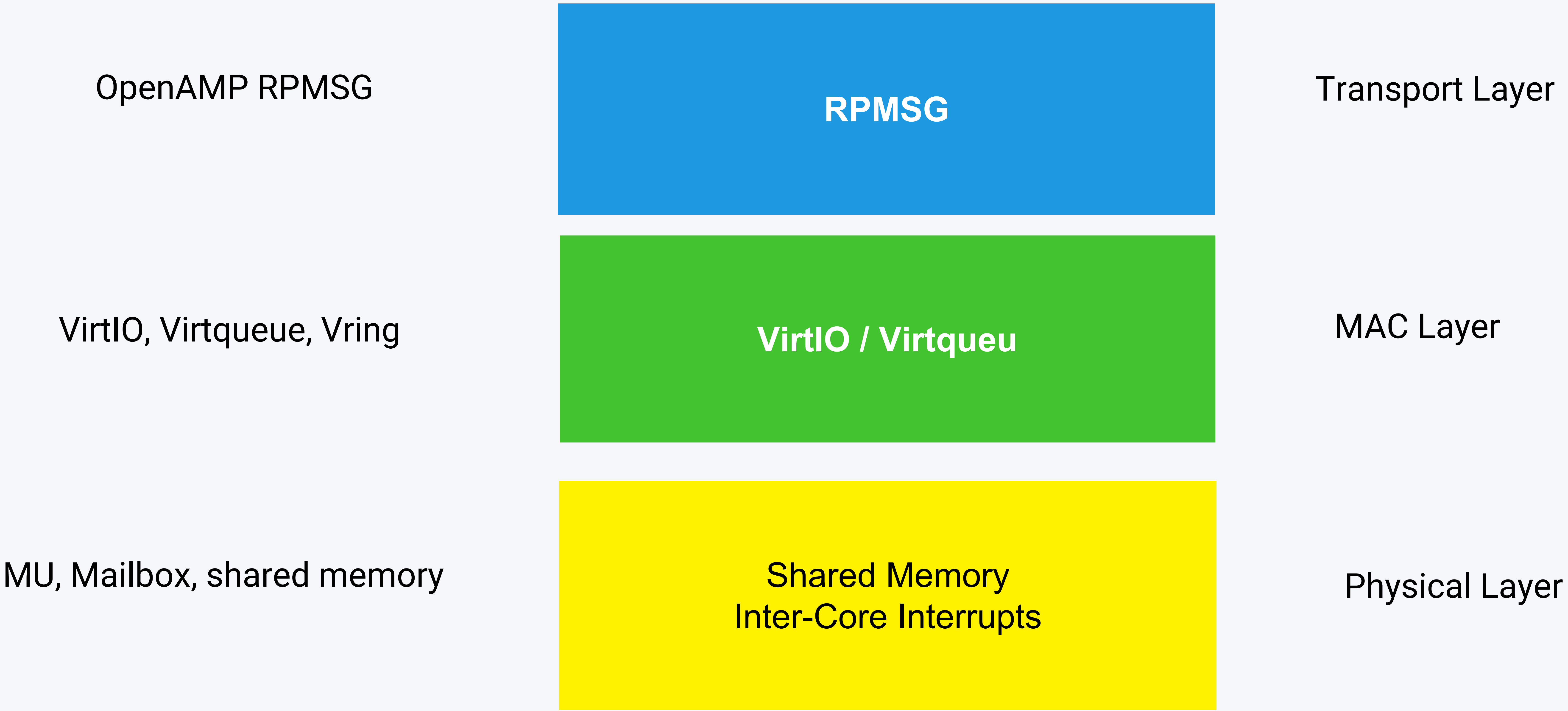
NXP I.MX7 Overview

MU – Message Unit


Processor A
Messaging Unit (MU)
Processor A side
Processor B side
TX / RX Registers
Status and Control Registers
Sync and Control Registers
Generate Interrupts
TX / RX Registers
Status and Control Registers
Sync and Control Registers
Generate Interrupts
Processor B
Processor A Peripheral Bus
Processor B Peripheral Bus
Interrupts to Processor A interrupt controller
Interrupts to Processor B interrupt controller

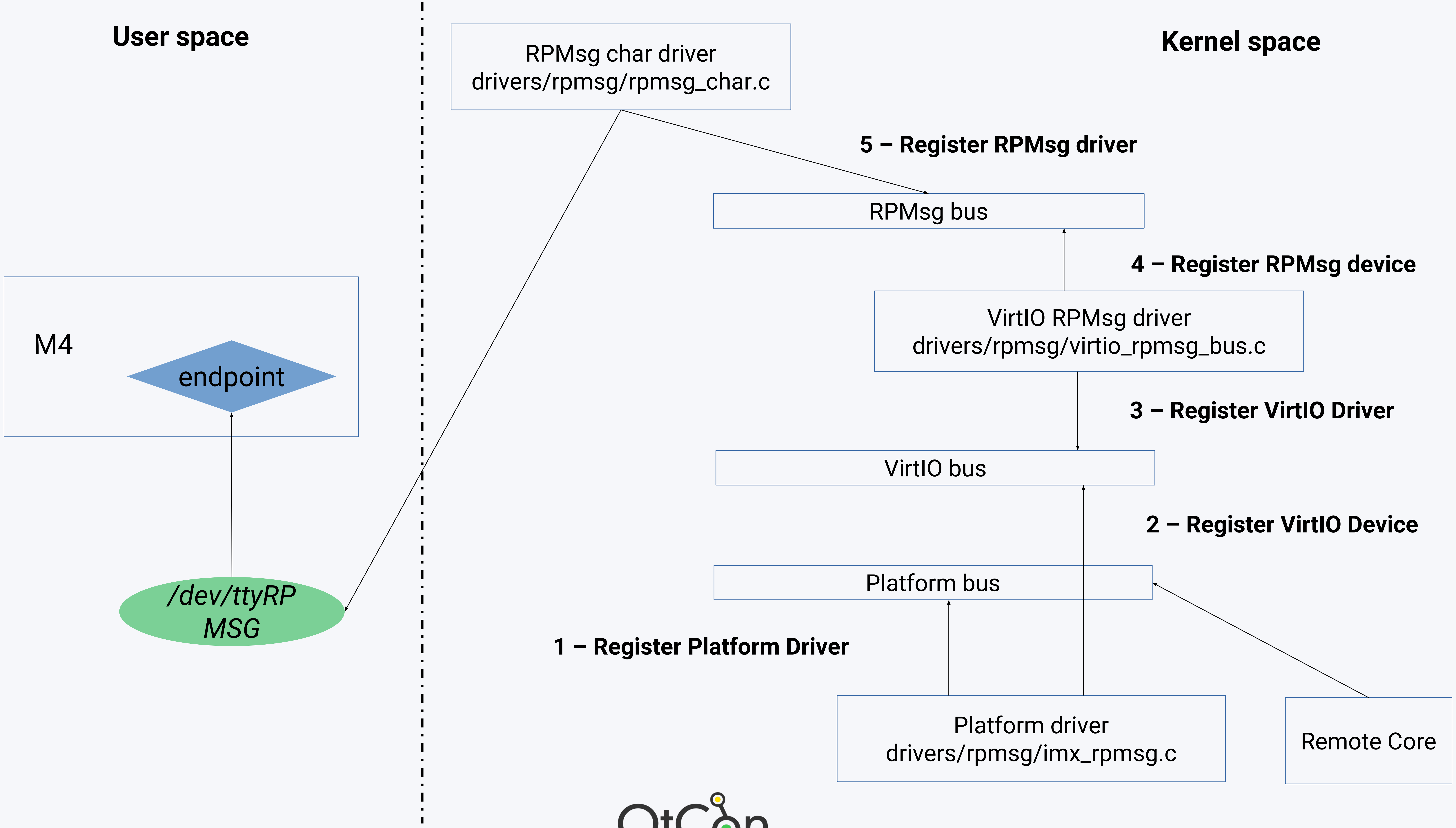
- Enables two processors within the SoC to communicate and coordinate by passing messages(ex: data, status and control)
- Signal the other processor using interrupts

Inter Processor Communication



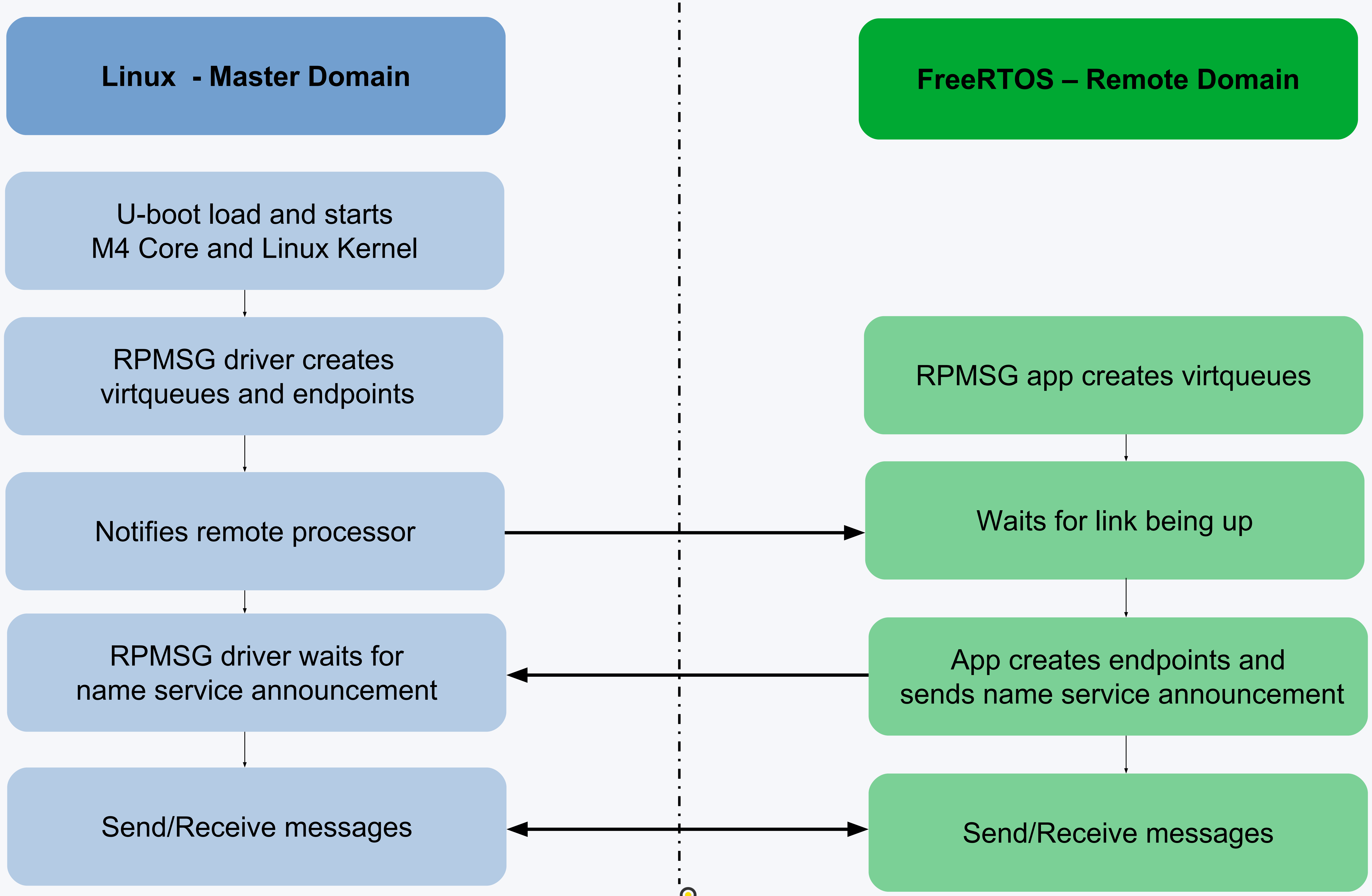
OpenAMP RPMSG	**RPMSG**	Transport Layer
VirtIO, Virtqueue, Vring	**VirtIO / Virtqueu**	MAC Layer
MU, Mailbox, shared memory	Shared Memory Inter-Core Interrupts	Physical Layer

RPMSG on Linux


User space
Kernel space
RPMsg char driver
drivers/rpmsg/rpmsg_char.c
5 – Register RPMsg driver
RPMsg bus
4 – Register RPMsg device
VirtIO RPMsg driver
drivers/rpmsg/virtio_rpmsg_bus.c
3 – Register VirtIO Driver
VirtIO bus
2 – Register VirtIO Device
Platform bus
1 – Register Platform Driver
Platform driver
drivers/rpmsg/imx_rpmsg.c
Remote Core
M4
endpoint
/dev/ttyRP
MSG

QtCon BR

Linux and FreeRTOS talking

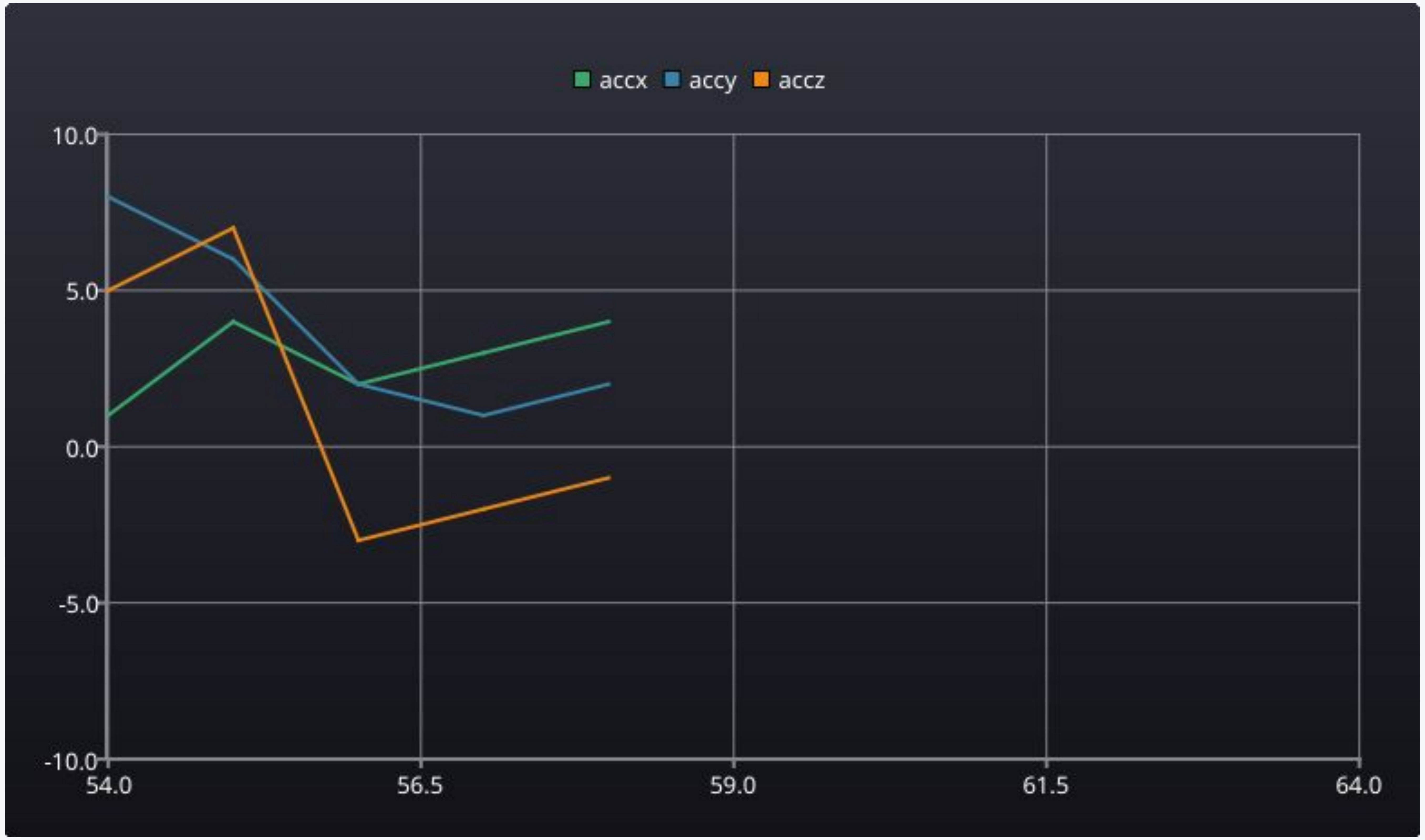


Linux - Master Domain
FreeRTOS – Remote Domain
U-boot load and starts M4 Core and Linux Kernel
RPMSG driver creates virtqueues and endpoints
RPMSG app creates virtqueues
Notifies remote processor
Waits for link being up
RPMSG driver waits for name service announcement
App creates endpoints and sends name service announcement
Send/Receive messages
Send/Receive messages

QtCon BR

Application
evelopment with Qt

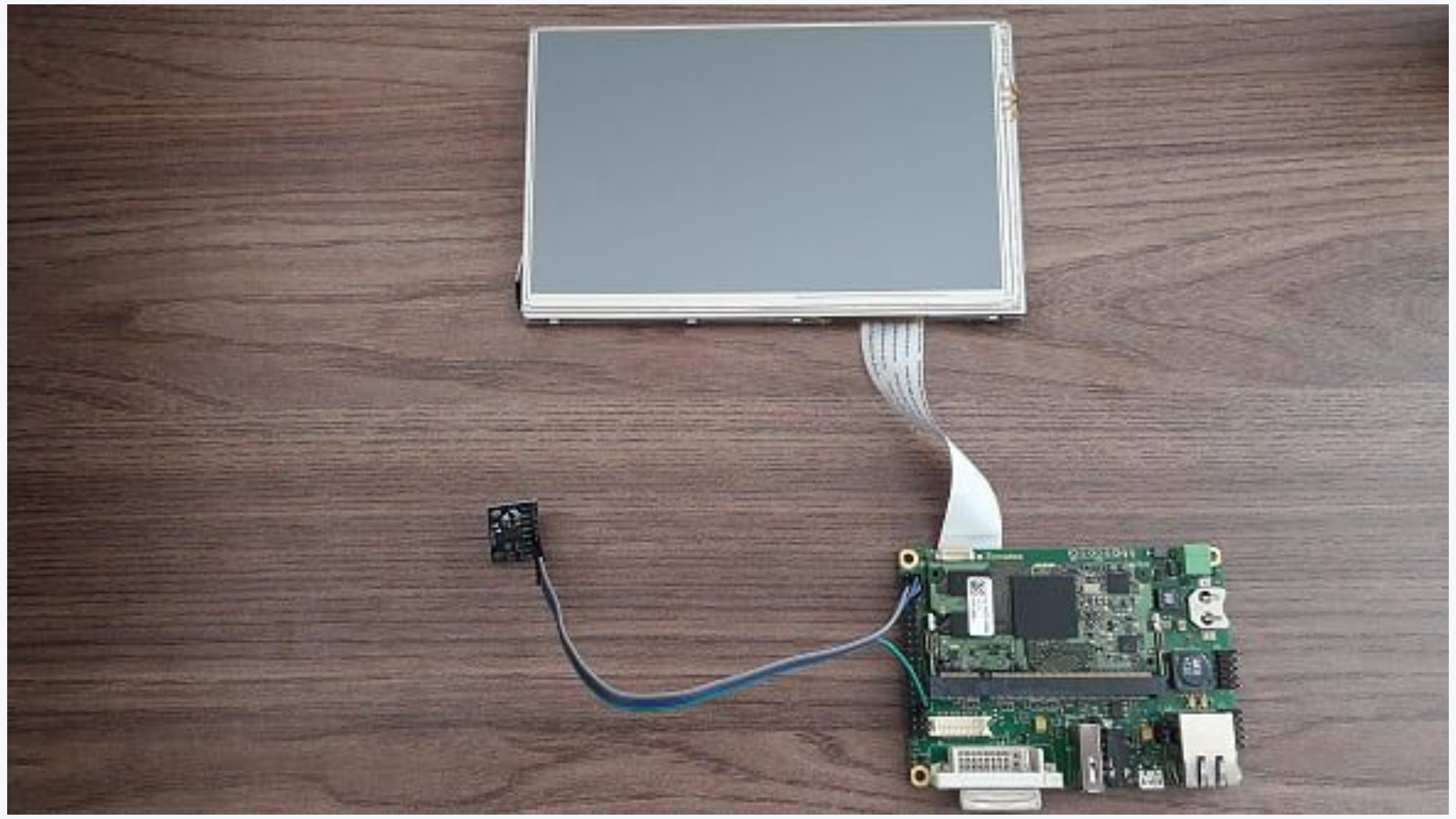
Hybrid Linux Qt / FreeRTOS Demo

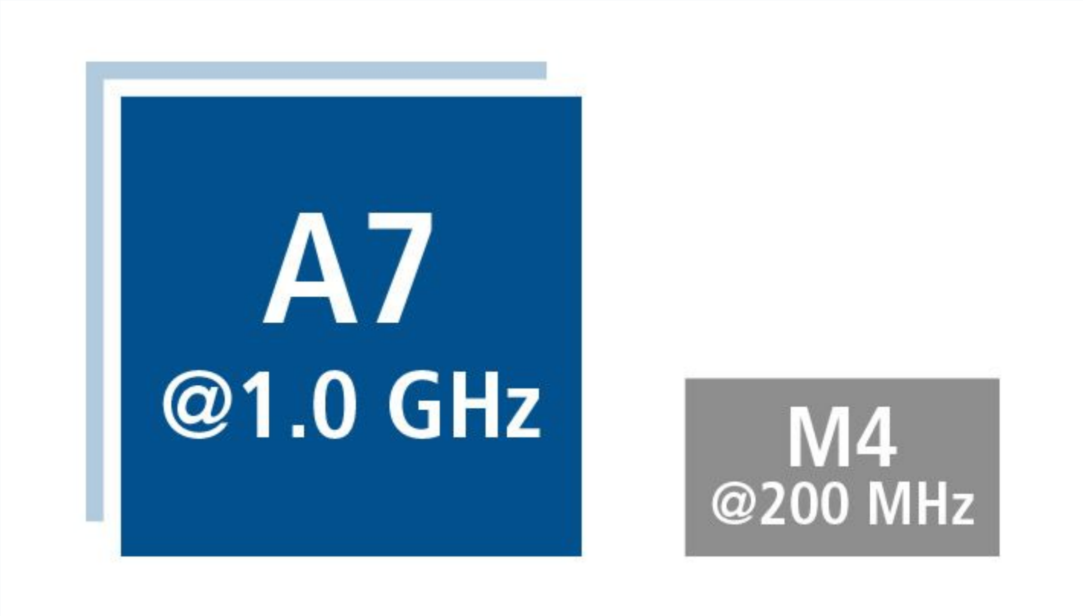


- IMU sensor (I2C) read by MCU
- Qt App read data from MCU using RPMSG
- Plot data on Linux using QtCharts

Hardware Setup

- Colibri iMX7D 512MB
- Iris Carrier Board
- 7” display
- MPU6050

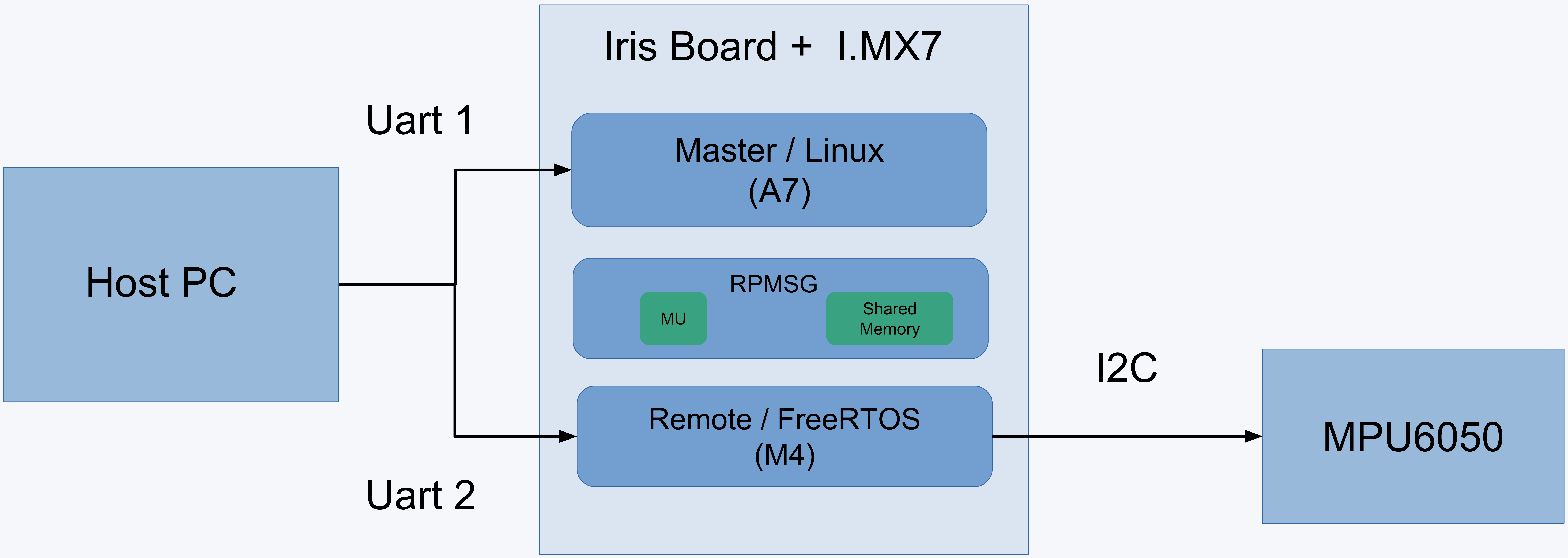




A7
@1.0 GHz

M4
@200 MHz

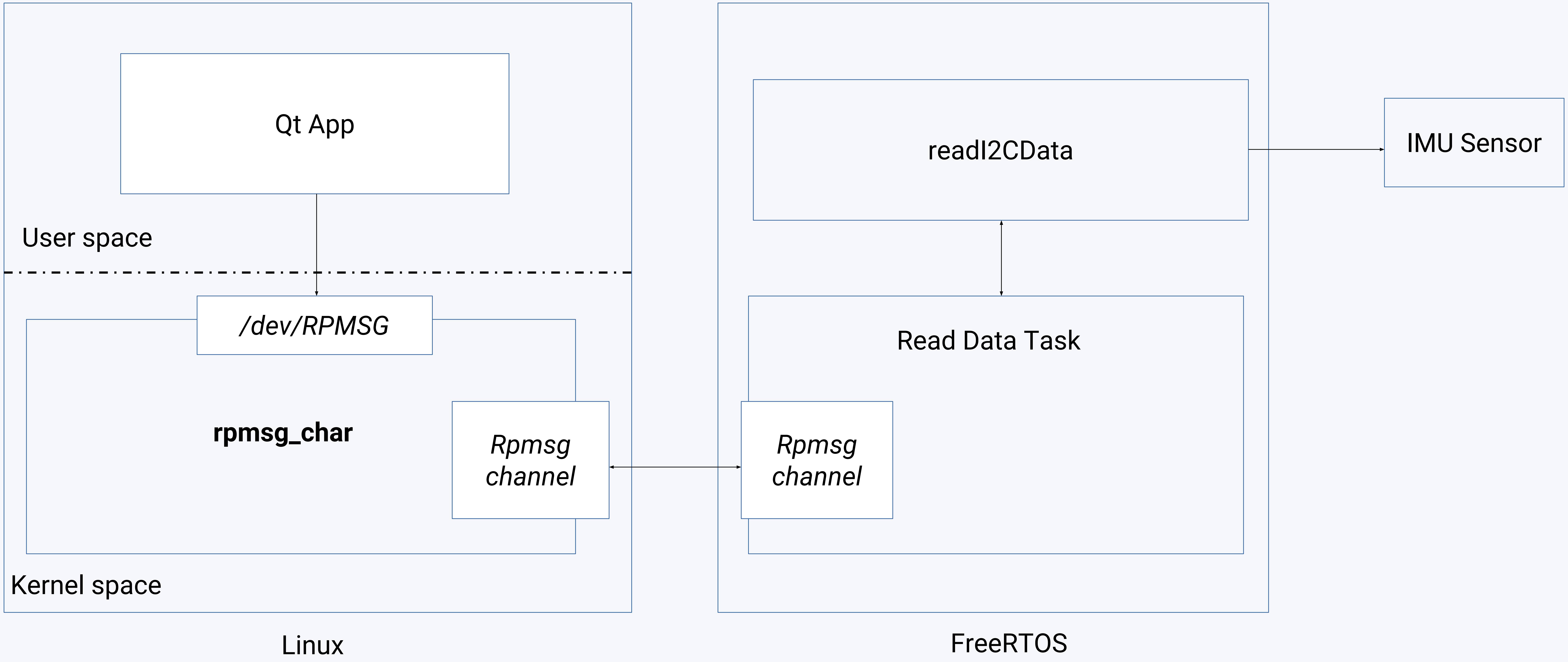
Hardware Setup



Qt Development

- Toolchain(cross-compile, rootfs, libraries) generated by Yocto-Project
- Configure Qt Kit for I.MX7 using toolchain generated by Yocto
- QtQuick and QML on i.MX7(no-GPU) Qt 2D Software Rendering
- **qputenv("QMLSCENE_DEVICE", QbyteArray("softwarecontext"))**;
- Chart Visualization via QtCharts
  - Add to your .pro: **QT += charts**
- **QtCharts is GPLv3!**

Architecture Overview



Realtime Class

```
class Realtime : public  QObject
{
   Q_OBJECT
   Q_PROPERTY(int accX READ XAcc NOTIFY accChanged)
   Q_PROPERTY(int accY READ YAcc NOTIFY accChanged)
   Q_PROPERTY(int accZ READ ZAcc NOTIFY accChanged)
)

public:
   Realtime(QObject *parent = nullptr);
   virtual ~Realtime();
private:
   QFile rpmsgDevice;
signals:
   void accChanged();
public slots:
   void update();
};
```

Realtime.h

```
Realtime {
        id: realtime
    }
```

Main.qml

Realtime Class

```
rpmsDevice.setFileName("/dev/ttyRPMSG");
rpmsgDevice.open(QIODevice::ReadWrite);

 qDebug() << "Get Sensor Realtime Data";
    if(!rpmsgDevice.isOpen()){
        qDebug() << "RPMSG Device not open";
    } else {
            int accx, accy, accz;
            QByteArray query("acc");
            rpmsgDevice.write(query);
            rpmsgDevice.flush();
            QbyteArray response = rpmsgDevice.readLine(64);
            sscanf(response.constData(),
                    "x:%d,y:%d,z:%d", &accx, &accy, &accz);
}
```

Realtime.cpp

QML

```
Timer {
    id: timer
    property int index: 0
    running: true
    repeat: true
    interval: 1000
    onTriggered: {
           realtime.update();
        accx.append(index,realtime.accX);
        accy.append(index,realtime.accY);
        accz.append(index,realtime.accZ);

      index++;
      axisX.min++;
      axisX.max++;
    }
  }
```

Main.qml

QML

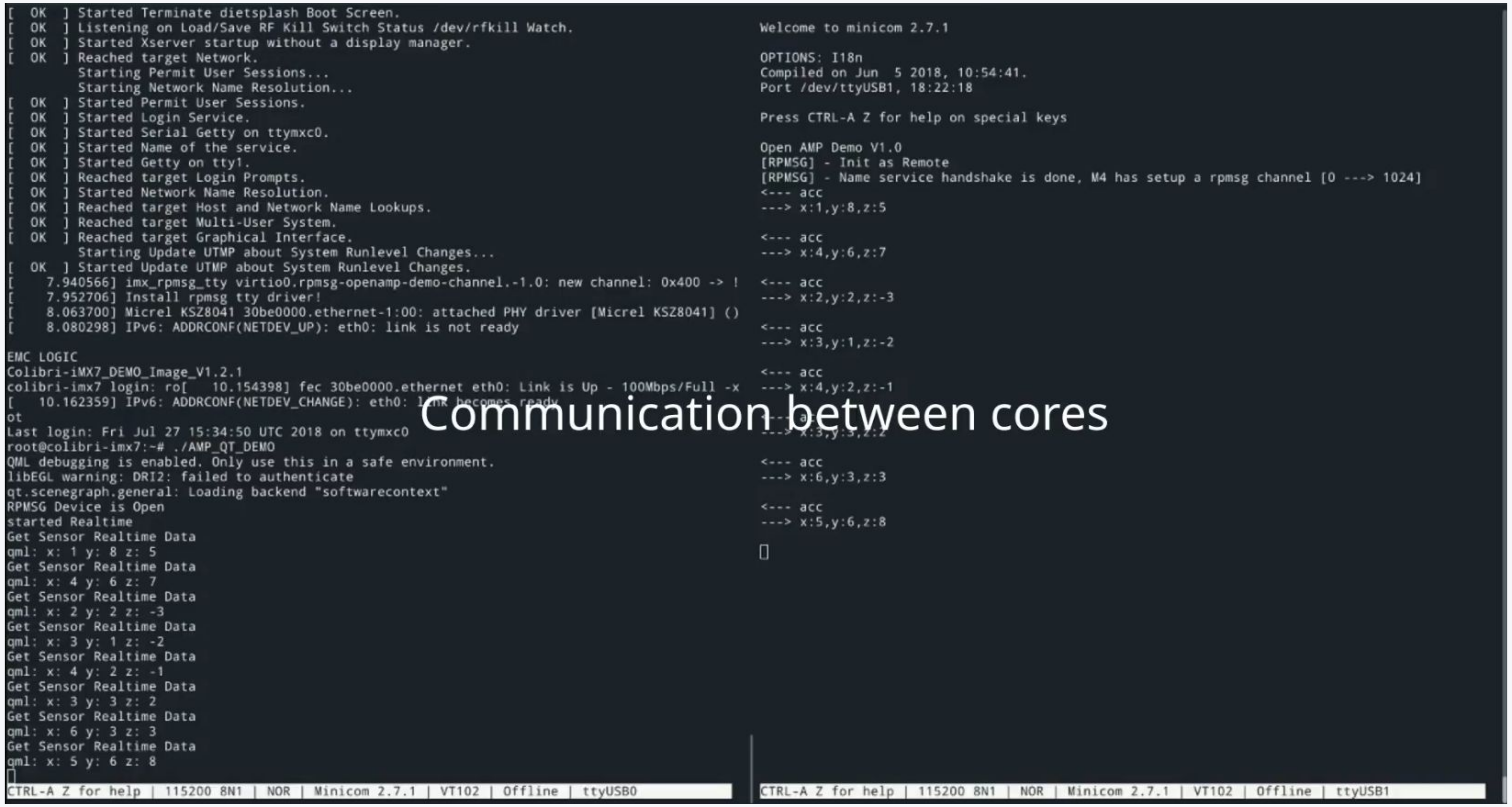
```
ChartView {
        id: chartview
        animationOptions: ChartView.NoAnimation
        theme: ChartView.ChartThemeDark
        antialiasing: true
        anchors.fill: parent

        ValueAxis {
            id: axisX
            min: -5
            max: 5
        }

        ValueAxis {
            id: axisY
            min: -10
            max: 10
        }
     }
```

```
LineSeries {
    id: accx
    name: "accx"
    axisY: axisY
    axisX: axisX
}

LineSeries {
    id: accy
    name: "accy"
    axisY: axisY
    axisX: axisX
}

LineSeries {
    id: accz
    name: "accz"
    axisY: axisY
    axisX: axisX
}
```

Main.qml

Demo Communication between cores

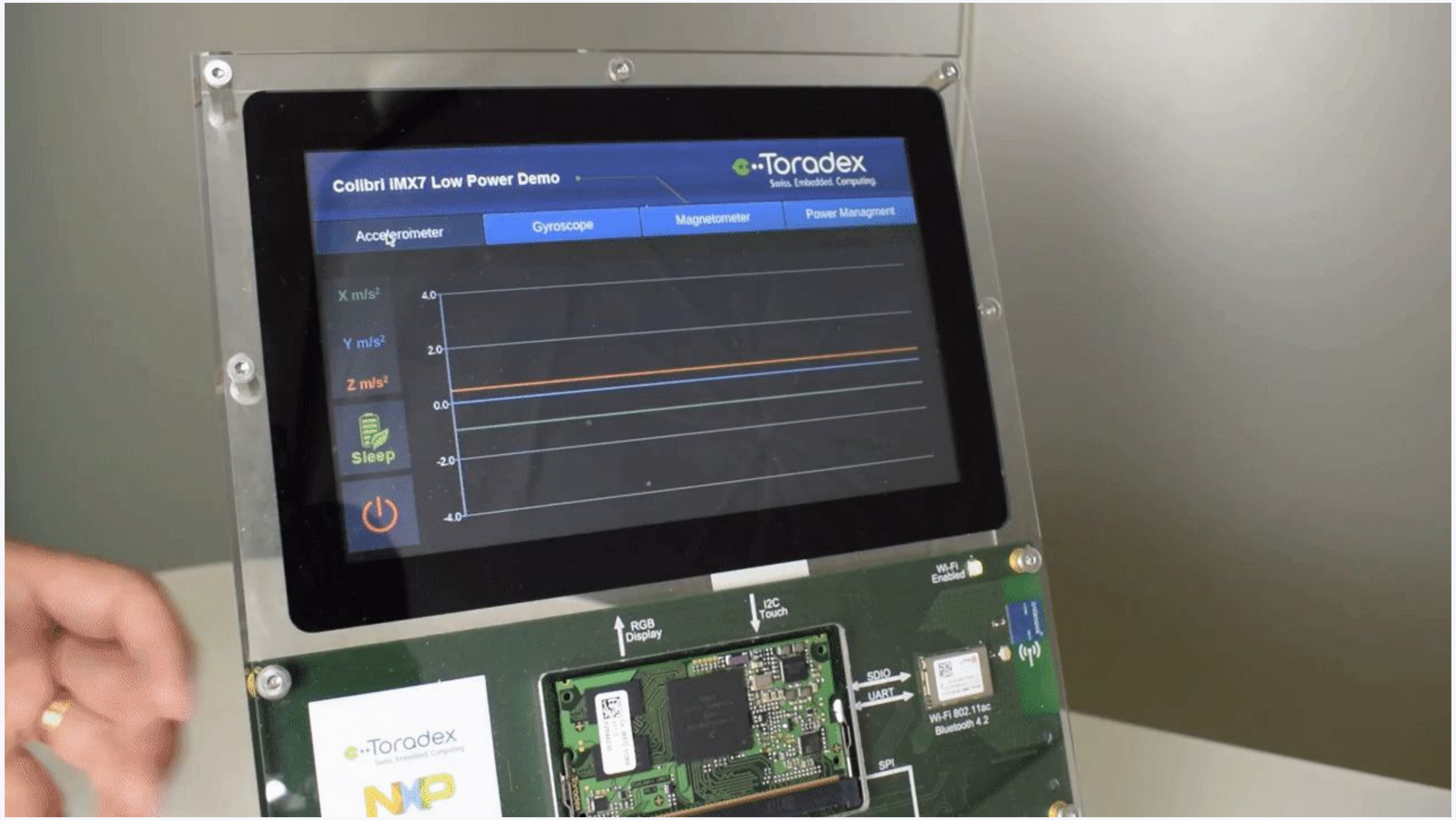
https://www.youtube.com/watch?v=SnLAySJPCBU



```
[  OK  ] Started Terminate dietsplash Boot Screen.
[  OK  ] Listening on Load/Save RF Kill Switch Status /dev/rfkill Watch.
[  OK  ] Started Xserver startup without a display manager.
[  OK  ] Reached target Network.
         Starting Permit User Sessions...
         Starting Network Name Resolution...
[  OK  ] Started Permit User Sessions.
[  OK  ] Started Login Service.
[  OK  ] Started Serial Getty on ttymxc0.
[  OK  ] Started Name of the service.
[  OK  ] Started Getty on tty1.
[  OK  ] Reached target Login Prompts.
[  OK  ] Started Network Name Resolution.
[  OK  ] Reached target Host and Network Name Lookups.
[  OK  ] Reached target Multi-User System.
[  OK  ] Reached target Graphical Interface.
         Starting Update UTMP about System Runlevel Changes...
[  OK  ] Started Update UTMP about System Runlevel Changes.
[    7.940566] imx_rpmsg_tty virtio0.rpmsg-openamp-demo-channel.-1.0: new channel: 0x400 -> !
[    7.952706] Install rpmsg tty driver!
[    8.063700] Micrel KSZ8041 30be0000.ethernet-1:00: attached PHY driver [Micrel KSZ8041] ()
[    8.080298] IPv6: ADDRCONF(NETDEV_UP): eth0: link is not ready

EMC LOGIC
Colibri-iMX7_DEMO_Image_V1.2.1
colibri-imx7 login: ro[   10.154398] fec 30be0000.ethernet eth0: Link is Up - 100Mbps/Full -x
[   10.162359] IPv6: ADDRCONF(NETDEV_CHANGE): eth0: link becomes ready
ot
Last login: Fri Jul 27 15:34:50 UTC 2018 on ttymxc0
root@colibri-imx7:~# ./AMP_QT_DEMO
QML debugging is enabled. Only use this in a safe environment.
libEGL warning: DRI2: failed to authenticate
qt.scenegraph.general: Loading backend "softwarecontext"
RPMSG Device is Open
started Realtime
Get Sensor Realtime Data
qml: x: 1 y: 8 z: 5
Get Sensor Realtime Data
qml: x: 4 y: 6 z: 7
Get Sensor Realtime Data
qml: x: 2 y: 2 z: -3
Get Sensor Realtime Data
qml: x: 3 y: 1 z: -2
Get Sensor Realtime Data
qml: x: 4 y: 2 z: -1
Get Sensor Realtime Data
qml: x: 3 y: 3 z: 2
Get Sensor Realtime Data
qml: x: 6 y: 3 z: 3
Get Sensor Realtime Data
qml: x: 5 y: 6 z: 8

CTRL-A Z for help | 115200 8N1 | NOR | Minicom 2.7.1 | VT102 | Offline | ttyUSB0
```

```
Welcome to minicom 2.7.1

OPTIONS: I18n
Compiled on Jun  5 2018, 10:54:41.
Port /dev/ttyUSB1, 18:22:18

Press CTRL-A Z for help on special keys

Open AMP Demo V1.0
[RPMSG] - Init as Remote
[RPMSG] - Name service handshake is done, M4 has setup a rpmsg channel [0 ---> 1024]
<--- acc
---> x:1,y:8,z:5

<--- acc
---> x:4,y:6,z:7

<--- acc
---> x:2,y:2,z:-3

<--- acc
---> x:3,y:1,z:-2

<--- acc
---> x:4,y:2,z:-1

<--- acc
---> x:3,y:3,z:2

<--- acc
---> x:6,y:3,z:3

<--- acc
---> x:5,y:6,z:8

CTRL-A Z for help | 115200 8N1 | NOR | Minicom 2.7.1 | VT102 | Offline | ttyUSB1
```

Demo QT charts



References

- **M4 Firmware - https://github.com/ferlzc/Asymmetric_QT_demo_firmware**
- **QT Application - https://github.com/ferlzc/Asymmetric_QT_demo**

References

- **Linux and Zephry “talking” to each other in the same SoC**

- https://events.linuxfoundation.org/wp-content/uploads/2017/12/Linux-and-Zephyr-%E2%80%9CTalking%E2%80%9D-to-Each-Other-in-the-Same-SoC-Diego-Sueiro-Sepura-Embarcados.pdf

- **OpenAMP Project Page** - https://github.com/OpenAMP/

- **An Introduction to Asymmetric Multiprocessing: When this Architecture can be a Game Changer and How to Survive It (ELC 2018)**
https://elinux.org/images/6/6e/AMP_-_Kynetics_ELC_2018_Portland.pdf

- **Asymetric Multiprocessing and Embedded Linux (ELC 2017)**
https://elinux.org/images/3/3b/NOVAK_CERVENKA.pdf

- **Toradex FreeRTOS on Cortex-M4 of Colibri IMX7**
https://developer.toradex.com/knowledge-base/freertos-on-the-cortex-m4-of-a-colibri-imx7

Obrigado!

Fernando Luiz Cola
fernando.cola@emc-logic.com

https://www.linkedin.com/in/ferlzc/

https://www.emc-logic.com/