

Using the Cortex-M4 core on VisionSOM-8MM

October 2020



SOM System on Module B Carrier Board D Development Kit Engineeging Since 2003 delivering proven designs

Agenda

Enabling the Cortex-M4 and RPMSG in device tree
Managing resources access
Obtaining the SDK for Cortex-M4
Building and running the example application



Exercises

- home/dev/Excercises/Workshop3/
- Lab1 Preparing the system image
- □ Lab2 Building and running Cortex-M4 application
- Lab3 Running Cortex-M4 application automatically



home/dev/Excercises/Workshop3/Lab1/linux-imx_5.4.bbappend

SRC_URI = "\${KERNEL_SRC};branch=\${KERNEL_BRANCH} \
 file://0001-Added-support-for-Cortex-M4-rpmsg.patch \
 "

home/dev/Excercises/Workshop3/Lab1/ 0001-Added-support-for-Cortex-M4-rpmsg.patch



Documentation/devicetree/bindings/remoteproc/imx-rproc.yaml

```
imx8mm-cm4 {
    compatible = "fsl, imx8mm-cm4";
    rsc-da = <0xb800000>;
    clocks = <&clk IMX8MM CLK M4 DIV>;
   mbox-names = "tx", "rx", "rxdb";
   mboxes = < mu 0 1
              &mu 1 1
              &mu 3 1>;
   memory-region = <&vdev0vring0>, <&vdev0vring1>, <&vdevbuffer>;
    syscon = <&src>;
```



abs www.somlabs.com

Documentation/devicetree/bindings/soc/fsl/fsl,rpmsg.txt

```
&rpmsg{
    vdev-nums = <1>;
    reg = <0x0 0xb8000000 0x0 0x10000>;
    memory-region = <&vdevbuffer>;
    status = "okay";
};
```



cd ~/imx-yocto-bsp/sources/meta-somlabs/recipes-kernel/linux

cp ~/Excercises/Workshop3/Lab1/linux-imx_5.4.bbappend .

cp ~/Excercises/Workshop3/Lab1/0001-Added-support-for-Cortex-M4-rpmsg.patch linux-imx/



Resource Domains





GOLD

SoMLabs | www.somlabs.com

home/dev/Excercises/Workshop3/Lab1/imx-atf_2.0.bbappend

SRC_URI += "file://0001-Visionsom-8mm-fix.patch \
 file://0002-Assign-M4-and-Uart3-to-domain-1.patch \
"



home/dev/Excercises/Workshop3/Lab1/ 0002-Assign-M4-and-Uart3-to-domain-1.patch

/* Assign M4 to domain 1 */
mmio_write_32(IMX_RDC_BASE + 0x204, 0x1);
/* Assign UART3 to domain 1 (M4) */
mmio_write_32(IMX_RDC_BASE + 0x5A0, 0xfc);



i.MX 8M Mini Applications Processor Reference Manual (IMX8MMRM.pdf) 3.2 Resource Domain Controller (RDC)

Table 3-2. Master Assignment Mapping

Master	RDC MDA Register
Quad A53	RDC_MDA0
M4	RDC_MDA1



RDC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
303D_0000	Version Information (RDC_VIR)	32	R	0376_E204h	3.2.5.1/60
303D_0024	Status (RDC_STAT)	32	R/W	0000_0100h	3.2.5.2/61
303D_0028	Interrupt and Control (RDC_INTCTRL)	32	R/W	0000_0000h	3.2.5.3/62
303D_002C	Interrupt Status (RDC_INTSTAT)	32	R/W	See section	3.2.5.4/62
303D_0200	Master Domain Assignment (RDC_MDA0)	32	R/W	0000_0000h	3.2.5.5/63
303D_0204	Master Domain Assignment (RDC_MDA1)	32	R/W	0000_0000h	3.2.5.5/63

SoMLabs | www.somlabs.com

GOLD PARTNER



RDC_MDA*n* field descriptions

Field	Description				
31 LCK	0 Not Locked 1 Locked				
30–2 Reserved	This field is reserved.				
DID	Domain ID Indicates the domain to which the Master is assigned				
	 Master assigned to Processing Domain 0 Master assigned to Processing Domain 1 Master assigned to Processing Domain 2 Master assigned to Processing Domain 3 				



home/dev/Excercises/Workshop3/Lab1/ 0002-Assign-M4-and-Uart3-to-domain-1.patch

/* Assign M4 to domain 1 */
mmio_write_32(IMX_RDC_BASE + 0x204, 0x1);
/* Assign UART3 to domain 1 (M4) */
mmio_write_32(IMX_RDC_BASE + 0x5A0, 0xfc);



i.MX 8M Mini Applications Processor Reference Manual (IMX8MMRM.pdf)

□ 3.2 Resource Domain Controller (RDC)

Table 3-3. RDC Peripheral Mapping

UART3	RDC_PDAP104	SEMA42 B2 / G40
UART2	RDC_PDAP105	SEMA42 B2 / G41

RDC memory map

303D_05A0	Peripheral Domain Access Permissions (RDC_PDAP104)	32	R/W	0000_00FFh	3.2.5.6/64



MLabs www.somlabs.com



SoMLabs | www.somlabs.com

GOLD PARTNER

Field	Description
31 LCK	Peripheral Permissions Lock
	When set prevents further modification of the Peripheral Domain Access Permissions (sticky bit until reset)
	0 Not Locked
	1 Locked
30 SREQ	Semaphore Required
	When set the hardware semaphore state enforces the semaphore lock. If a domain has access permissions and a semaphore has locked a shared peripheral then only the domain holding the semaphore signal can access this peripheral.
	0 Semaphores have no effect
	1 Semaphores are enforced
29–8 Reserved	This field is reserved.
7 D3R	Domain 3 Read Access
	0 No Read Access
	1 Read Access Allowed
6 D3W	Domain 3 Write Access
	0 No Write Access
	1 Write Access Allowed



GOLD

0xFC -> All domains except 0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LCK	SRE Q							Rese	erved						
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W				Rese	erved				D3R	D3W	D2R	D2W	D1R	D1W	D0R	D0W
Reset	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

SoMLabs | www.somlabs.com

GOLD

cd ~/imx-yocto-bsp/sources/meta-somlabs/recipes-bsp/imx-atf

cp ~/Excercises/Workshop3/Lab1/imx-atf_2.0.bbappend .

cp ~/Excercises/Workshop3/Lab1/0002-Assign-M4-and-Uart3-to-domain-1.patch imx-atf/



Obtaining the SDK for Cortex-M4

Obtaining the SDK for Cortex-M4

 /home/dev/Excercises/Workshop3/Lab2/ SDK_2.6.1_MIMX8MM6xxxKZ.zip
 https://mcuxpresso.nxp.com

MCUXpresso SDK Builder

The MCUXpresso SDK brings open source drivers, middleware, and reference example applications to speed your software development. Customize and download an SDK specific to your processor or evaluation board selections.

Select Development Board

Q Explore and filter devices

Access My SDK Dashboard



SoMLabs | www.somlabs.com

Obtaining the SDK for Cortex-M4

 \otimes

Select Development Board

Search for your board or kit to get started.

Search by Name

imx8mm6

Select a Board, Kit, or Processor



Don't use: (<, >, :, ", I, J, ?, *, 1) in the name of your SDK



Hardware Details

Included Part	MIMX8MM6CVTKZ
Numbers	
Board(s)	EVK-MIMX8MM
Device	MIMX8MM6
Core Type / Max	Cortex-M4F / 400MHz
Freq	
Device Memory Size	0 KB Flash
	256 KB RAM





SoMLabs | www.somlabs.com

imx-yocto-bsp/sources/meta-imx/meta-sdk/recipes-fsl/m4-demos/ README

The M4 demo app version of each SoCs are followed:

- * 2.6.1 -- i.MX 8MM
- * 2.6.0 -- i.MX 7ULP/8DXL-Phantom
- * 2.7.0 -- i.MX 8QXP
- * 2.5.2 -- i.MX 8QM
- * 2.3.0 -- i.MX 8MQ
- * 1.0.1 -- i.MX 7D



Obtaining the SDK for Cortex-M4

Developer Environment Settings

Selections here will impact files and examples projects included in the SDK and Generated Projects

2.6.1 20	019-11-29	-	Toolchain / IDE H GCC ARM Embedd€ ▼	lost OS Linux -
Filter by I	Name, Category, or Des	scription	Select All	Unselect All
¢	Name 🗘	Category	Description	Dependencies
	CMSIS DSP Library		CMSIS DSP Software Library	
V	multicore	Middleware	Multicore Software Development Kit	Amazon- FreeRTOS
\checkmark	Amazon- FreeRTOS	Operating System	Amazon-FreeRTOS	

Click the link below to request this specific MCUXpresso SDK Build

In general, SDK builds should complete within a few minutes. You will be notified via email and notifications in the upper right corner of this webpage.



Archive Name SDK 2.6.1 MIMX8MM6xxxLZ

Don't use: (<,>,:,*,I,],?,•,1) in the name of your SDK

.

Hardware Details

Included Part Numbers	MIMX8MM6DVTLZ
Board(s)	EVK-MIMX8MM
Device	MIMX8MM6
Core Type / Max Freq	Cortex-M4F / 400MHz
Device	0 KB Flash
Memory Size	256 KB RAM
SDK Details	

Middleware:	Amazon-FreeRTOS, CMSIS DSP Library, multicore
Toolchain:	GCC ARM Embedded
Host OS:	Linux
SDK Version:	2.6.1 (released 2019-11-29)

Documentation

API Reference Manual



SoMLabs | www.somlabs.com

Preparing the environment (ARMGCC_DIR and PATH)

cd ~/Excercises/Workshop3/Lab2

. setenv.sh

Building the example

cd SDK_2.6.1_MIMX8MM6xxxKZ/boards/visionsom-8mm/multicore_examples/rpmsg_lite_str_echo_rtos/armgcc

./build_debug.sh



Copying the binary to boot partition

scp debug/rpmsg_lite_str_echo_rtos_imxcm4.bin root@visionsom-8mmcb-std.local:/run/media/mmcblk2p1/



Running the binary - u-boot

fatload mmc 2:1 0x48000000 rpmsg_lite_str_echo_rtos_imxcm4.bin
cp.b 0x48000000 0x7e0000 35000
bootaux 0x7e0000
boot

Running the binary - kernel

modprobe imx_rpmsg_tty
echo "on" > /dev/ttyRPMSG30
echo "off" > /dev/ttyRPMSG30



```
/home/dev/Excercises/Workshop3/Lab3/
  u-boot-imx 2019.04.bbappend
SRC URI = "\
       ${UBOOT SRC};branch=${SRCBRANCH} \
       file://splash.bmp \
       file://m4.bin \
       file://0001-Added-loading-m4-binary-to-boot-command.patch \
       11
do install append visionsom-8mm-cb() {
        install -d ${DEPLOY DIR_IMAGE}
        install -m 0644 ${WORKDIR}/m4.bin ${DEPLOY DIR IMAGE}/m4.bin
                                                                  GOLD
```

abs www.somlabs.com

home/dev/Excercises/Workshop3/Lab3/

0001-Added-loading-m4-binary-to-boot-command.patch

"loadm4bin=fatload mmc 2:1 0x48000000 m4.bin; cp.b 0x48000000 0x7e0000 35000; bootaux 0x7e0000;\0" \

```
"mmcboot=echo Booting from mmc ...; " \
```

"run mmcargs; " \

"run loadm4bin; " \



home/dev/Excercises/Workshop3/Lab3/visionsom-8mm-cb.inc

```
IMAGE_BOOT_FILES = " \
${KERNEL_IMAGETYPE} \
${@make_dtb_boot_files(d)} \
m4.bin \
```

П



cd ~/imx-yocto-bsp/sources/meta-somlabs/recipes-bsp/u-boot

cp ~/Excercises/Workshop3/Lab3/u-boot-imx_2019.04.bbappend .

cp ~/Excercises/Workshop3/Lab3/0001-Added-loading-m4-binary-to-bootcommand.patch u-boot-imx/

cp ~/Excercises/Workshop3/Lab3/m4.bin u-boot-imx/



cd ~/imx-yocto-bsp/sources/meta-somlabs/conf/machine/include

cp ~/Excercises/Workshop3/Lab3/visionsom-8mm-cb.inc .



