# Build Freescale MQX_RTOS

Rev 1.1 20160301

TechNexion

# Contents

# 1. Supported hardware

These are the systems covered in this guide:

System-on-Modules:

- EDM1-CF-IMX6SX

Carrier Boards:

- EDM1-FAIRY
- EDM1-GOBLIN

MQX BSP is provided to the SOC with Cortex-M core. Freescale i.mx6sx has Cortex-A9 and Cortex-M4 cores. Cortex-A9 runs Linux, and Cortex-M4 runs MQX.

# 2. Get MQX Source Code

There are two ways that you can get EDM Yocto BSP source code.

**1. From Technexion website:**

Download the source tarball “edm1-cf-imx6_edm1-fairy_MQX-4.1.0-linux_source.zip”.

**2. From Technexion github:**

https://github.com/TechNexion/fsl_mqx

To get the source code:

```
git clone git@github.com:TechNexion/fsl_mqx.git
cd fsl_mqx
git checkout mqx_4.1.0_imx6sx_linux
```

# 3. Build MQX RTOS

## 3.1 Set up toochain

Download and install toolchain to compile MQX:

```
$ wget 'https://launchpad.net/gcc-arm-embedded/4.8/4.8-2014-q1-update/+download/gcc-arm-none-eabi-4_8-2014q1-20140314-linux.tar.bz2'

$ tar jxvf gcc-arm-none-eabi-4_8-2014q1-20140314-linux.tar.bz2
```

Export the environment of toolchain:

```
$ export TOOLCHAIN_ROOTDIR=${absolute_path}/gcc-arm-none-eabi-4_8-2014q1
```

```
$ export PATH=$PATH:${absolute_path}/gcc-arm-none-eabi-4_8-2014q1/bin
```

Compile the MQX:

```
$ cd fsl_mqx/build/edm1-cf-imx6sx/make

$ ./build_gcc_arm.sh
```

Then, you can start to compile the example you would like to run.

# 4. Run an example of MQX

## 4.1 Build an example of MQX

Here, we take “pingpong application” as example.

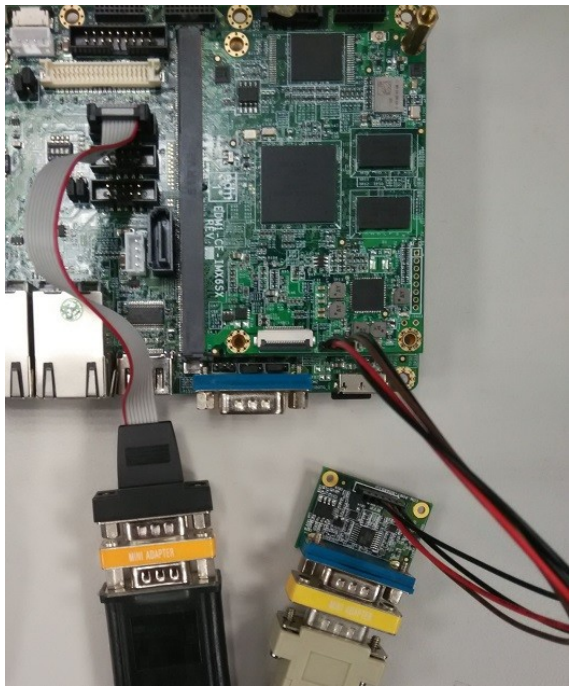
```
$ cd fsl_mqx/mcc/examples/pingpong/build/make/pingpong_example_edm1-cf-imx6sx/

$ ./build_gcc_arm.sh

$ cd gcc_arm/extflash_release/

$ arm-none-eabi-objcopy -O binary pingpong_example_edm1-cf-imx6sx.elf m4_qspi.bin
```

Copy “m4_qspi.bin” and “imx6sx-edm1-cf-m4.dtb” into first partition of eMMC/SD.

```
imx6sx-edm1-cf.dtb  imx6sx-edm1-cf-m4.dtb m4_qspi.bin  zImage
```

## 4.2 Run pingpong example

As the picture below, connect these two UARTs to your PC and set the baud rate as “115200”. Left UART is for MQX, and right UART is for linux.



**Run pingpong example on edm1-cf-imx6sx:**

In u-boot prompt:

```
root@edm-goblin-imx6sx:~# run update_m4_from_sd

root@edm-goblin-imx6sx:~# run m4boot

root@edm-goblin-imx6sx:~# setenv fdt_file imx6sx-edm1-cf-m4.dtb

root@edm-goblin-imx6sx:~# setenv mmcargs "${mmcargs} uart_from_osc"

root@edm-goblin-imx6sx:~# setenv bootcmd "run m4boot;${bootcmd}"

root@edm-goblin-imx6sx:~# boot
```

Then you will see the following message on left UART:

```
***** MCC PINGPONG EXAMPLE *****
Please wait :
   1) A9 peer is ready
Then press "S" to start the demo
*******************************
```

Press "S" to start the demo : S

In the Linux prompt, activate pingpong application:

```
root@~# echo 1 > /sys/devices/soc0/soc.0/2200000.aips-bus/mcctest.16/pingpong_en &
```

Then, cortex-A9 and cortex-M4 will start to communication via UARTs.