

# How to Run FreeRTOS Demo on Cortex-M4 of the phyBOARD-Zeta i.MX7

The following is a guide for running a FreeRTOS demo on the Cortex-M4. The demo showcases RPSMsg communication between the Cortex-A7 (running Linux) and the Cortex-M4. In the demo, the Cortex-M4 receives and prints a message from Cortex-A7. It will toggle an LED if a certain message is received.

<b>Targeted Hardware</b>	phyCORE-i.MX7, phyBOARD-Zeta i.MX7
<b>Targeted Software</b>	Cortex-M4: <a href="#">BSP FreeRTOS FSL iMX7 ALPHA1</a> Cortex-A7: <a href="#">BSP Yocto FSL iMX7 ALPHA2</a>
<b>Date</b>	04 Oct 2016

## Running the Demo

### Requirements

- SD card formatted with sdcard image from ALPHA2 release. Instructions for formatting [here](#)
- Hardware set up as described in [Getting Started](#) section of Quickstart
- PEB-EVAL-02 module required. Demo uses LED1 and debug UART2 for the Cortex-M4

### Step-by-step guide

1. Download demo image from Artifactory [Here](#) and add to Boot partition of SD card.
2. Insert SD card into microSD slot on the underside of the board. Connect a serial cable from UART2 RS232 DB-9 connector on the evaluation board to your host PC.
3. In your preferred terminal software (such as Minicom or TeraTerm) on your host PC, configure an additional terminal for UART2 with 115200 baud, 8 data bits, no parity, and 1 stop bit (8n1), no handshake. Note that UART5 should be connected as well and is the debug UART for Cortex-A7)
4. Set the boot switch located on the underside of the board to boot from SD (See [quickstart](#) for more info).
5. Power on the board. Press the power button if board doesn't automatically boot.
6. Press any key to stop autoboot and enter U-Boot.
7. In U-Boot, edit the environment to boot imx7d-phyboard-zeta-m4.dtb. This DTS file configures memory allocation, enables rpmsg bus, and disables UART2 in Linux.

```
=> editenv fdt_file
edit: imx7d-phyboard-zeta-m4.dtb

=> saveenv
=> reset
```

8. In U-Boot, type the following to load the application image from the SD card to TCM, flush cached content, and start the M4 demo.

```
=> fatload mmc 0:1 0x7F8000 rpmsg_gpio_toggle_freertos_example.bin
=> dcache flush
=> bootaux 0x7F8000
```

9. You should see the following appear in the serial terminal connected to UART2:

```
RPMSG GPIO Toggle FreeRTOS RTOS API Demo...
Load module imx_rpmsg_tty in Linux to continue
```

10. Now boot Linux on the Cortex-A7.

```
=> boot
```

11. Type root to log in, and then load the kernel module imx\_rpmsg\_tty which will configure RPMSG on the A7.

```
modprobe imx_rpmsg_tty
```

12. You can now echo content to /dev/ttyRPMSG and it will be received by the M4 and printed on the FreeRTOS serial console.
13. In order to toggle LED1 on the evaluation board, echo "led1" to /dev/ttyRPMSG:

```
echo "led1" > /dev/ttyRPMSG
```

## Demo Source Code

The FreeRTOS release is based on NXP's FreeRTOS\_BSP\_1.0.1\_iMX7D. PHYTEC's source can be found on [stash.phytec.com](http://stash.phytec.com). The demo is located at: [examples/imx7d\\_phyboard\\_zeta/demo\\_apps/rpmsg/gpio\\_toggle\\_freertos/](#).

See NXP's Getting Started Guide located at [doc/Getting\\_Started\\_with\\_FreeRTOS\\_BSP\\_for\\_i.MX\\_7Dual.pdf](#) for instructions on building the demo with ARM GCC or DS-5.

## Additional Info

NXP Documentation is located in the doc/ subdirectory of the source code. They provide extensive documentation on building and running demos, a reference manual, and information on NXP's implementation of RPMsg.