Document Number:T1040RDBQS
Rev 0, 04/2014

QorIQ T1040 Reference Design Board Quick Start

1 Introduction

The T1040 reference design board (RDB) system is a hardware board, having a Freescale QorIQ T1040 processor with four e5500 cores and speed up to 1.4 GHz.

The part number of the T1040 reference design board (RDB) system is T1040RDB-PA, (for a board based upon T1040 Rev 1.0 silicon).

Contents

2 Related documentation

The table below lists and explains the additional documents that you can refer to, for more information about C29x PCIe.

Some of the documents listed below may be available only under a non-disclosure agreement (NDA). To request access to these documents, contact your local field applications engineer or sales representative.

Table 1. Useful references

Document	Description
T1040 QorIQ Advanced Multiprocessing Processor Reference Manual	Provides details about individual implementations
Data sheet	Provides specific data regarding bus timing, signal behavior, and AC, DC, and thermal characteristics, as well as other design considerations
Application note	Addresses specific design issues useful to programmers and engineers working with Freescale processors
T1040 Product Brief	Provides an overview of the T1040 features, and examples of its usage

3 Preparing board

The figure below shows the front panel of the T1040 RDB.

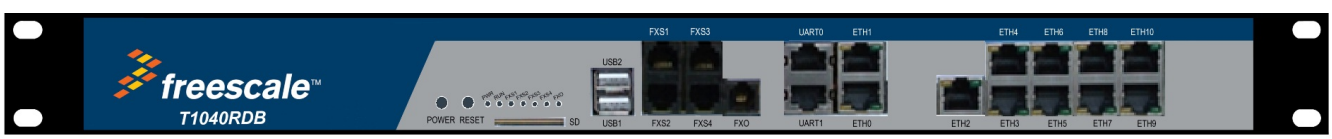

Figure 1. T1040RDB front panel

To prepare the T1040RDB for use, default configuration should be: CPU: 1.4 GHz, DDR: 1600 MT/s. The steps are:

1. Attach an RS-232 cable between the T1040RDB UART0 port and host computer
2. Open a serial console tool on the host computer to communicate with the T1040RDB
3. Configure the host computer's serial port with the following settings:
   - Data rate: 115200 bps
   - Number of data bits: 8
   - Parity: None
   - Number of stop bits: 1
   - Flow control: Hardware/None

Push the power button on the front side of the chassis. The board will boot and show the u-boot console messages.

```
U-Boot 2014.01QorIQ-SDK-T1040-BSP0.2 (Mar 07 2014 - 01:04:58)

CPU0:  T1040E, Version: 1.0, (0x85280010)
Core:  e5500, Version: 2.0, (0x80241020)
Clock Configuration:
       CPU0:1400 MHz, CPU1:1400 MHz, CPU2:1400 MHz, CPU3:1400 MHz,
       CCB:600  MHz,
       DDR:800  MHz (1600 MT/s data rate) (Asynchronous), IFC:150  MHz
```

```
       FMAN1: 600 MHz
       QMAN:  300 MHz
       PME:   300 MHz
L1:    D-cache 32 KiB enabled
       I-cache 32 KiB enabled
Reset Configuration Word (RCW):
       00000000: 0c18000e 0e000000 00000000 00000000
       00000010: 66000002 80000002 ec027000 01000000
       00000020: 00000000 00000000 00000000 00032810
       00000030: 00000000 0342500f 00000000 00000000
Board: T1040RDB
Board rev: 0x01 CPLD ver: 0x02, vBank: 4
I2C:   ready
SPI:   ready
DRAM:  Initializing....using SPD
Detected UDIMM 18KSF51272AZ-1G6K1
2 GiB left unmapped
    DDR: 4 GiB (DDR3, 64-bit, CL=11, ECC on)
       DDR Chip-Select Interleaving Mode: CS0+CS1
Flash: 256 MiB
L2:    256 KiB enabled
Corenet Platform Cache: 256 KiB enabled
Using SERDES1 Protocol: 102 (0x66)
NAND:  1024 MiB
MMC:  FSL_SDHC: 0
PCIe1: Root Complex, no link, regs @ 0xfe240000
PCIe1: Bus 00 - 00
PCIe2: Root Complex, x1 gen1, regs @ 0xfe250000
  02:00.0     - 8086:10d3 - Network controller
PCIe2: Bus 01 - 02
PCIe3: Root Complex, no link, regs @ 0xfe260000
PCIe3: Bus 03 - 03
PCIe4: Root Complex, no link, regs @ 0xfe270000
PCIe4: Bus 04 - 04
In:    serial
Out:   serial
Err:   serial
Net:   Initializing Fman
Fman1: Uploading microcode version 106.4.14
FSL_MDIO0:0 is connected to FM1@DTSEC1.  Reconnecting to FM1@DTSEC2
FSL_MDIO0:0 is connected to FM1@DTSEC2.  Reconnecting to FM1@DTSEC3
e1000: 68:05:ca:04:d5:6a
       FM1@DTSEC1, FM1@DTSEC2, FM1@DTSEC3, FM1@DTSEC4 [PRIME], FM1@DTSEC5, e1000#0
Warning: e1000#0 MAC addresses don't match:
Address in SROM is         68:05:ca:04:d5:6a
Address in environment is  00:04:9f:ef:00:00

Hit any key to stop autoboot:  0
```

The system auto boots and shows the following Linux login screen.

```
t1040rdb
login: root
root@t1040rdb:~#
```

4 System board interface

The figure below shows the top view of the T1040 RDB system board interface.

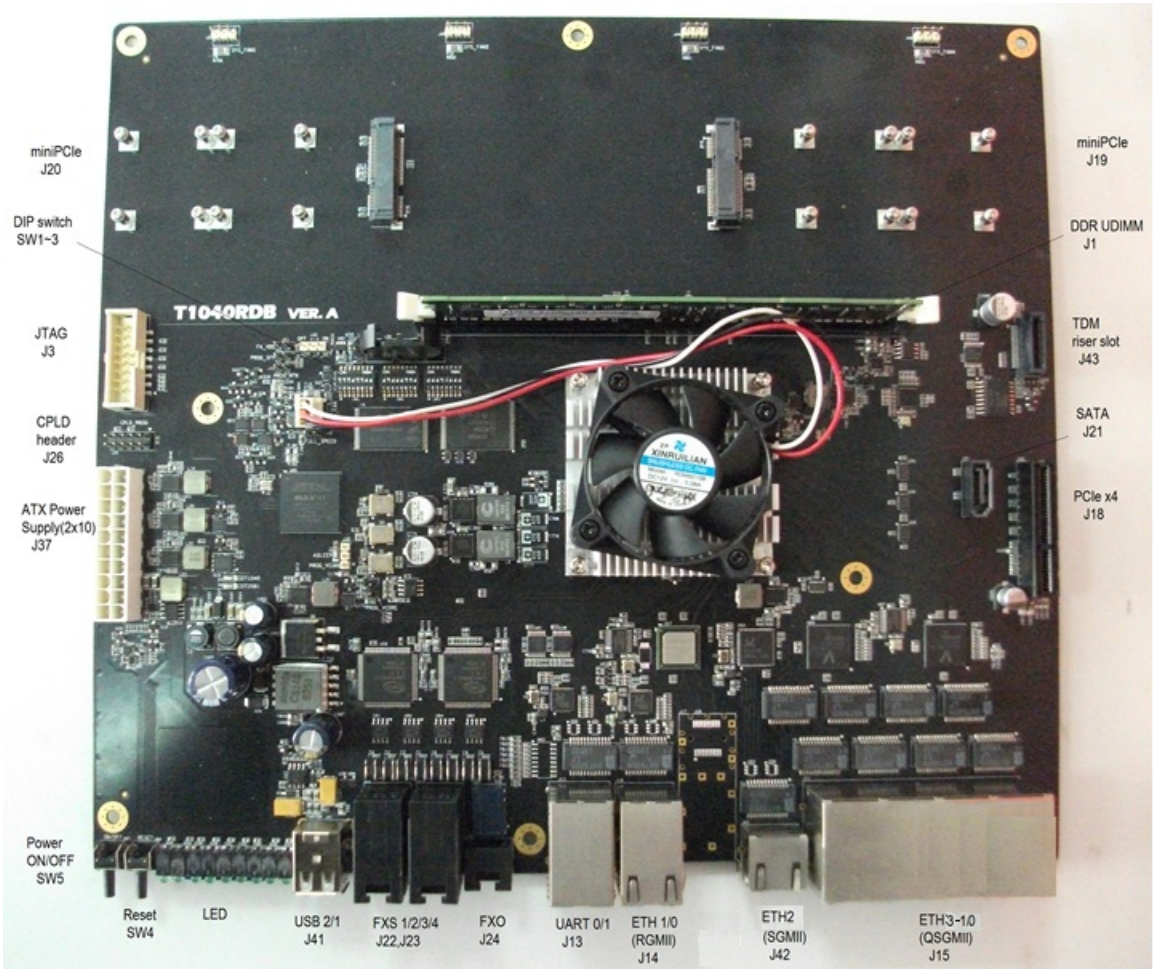

Figure 2. T1040RDB system board interface

4.1 Block diagram

The figure below shows the high-level block diagram of T1040RDB.

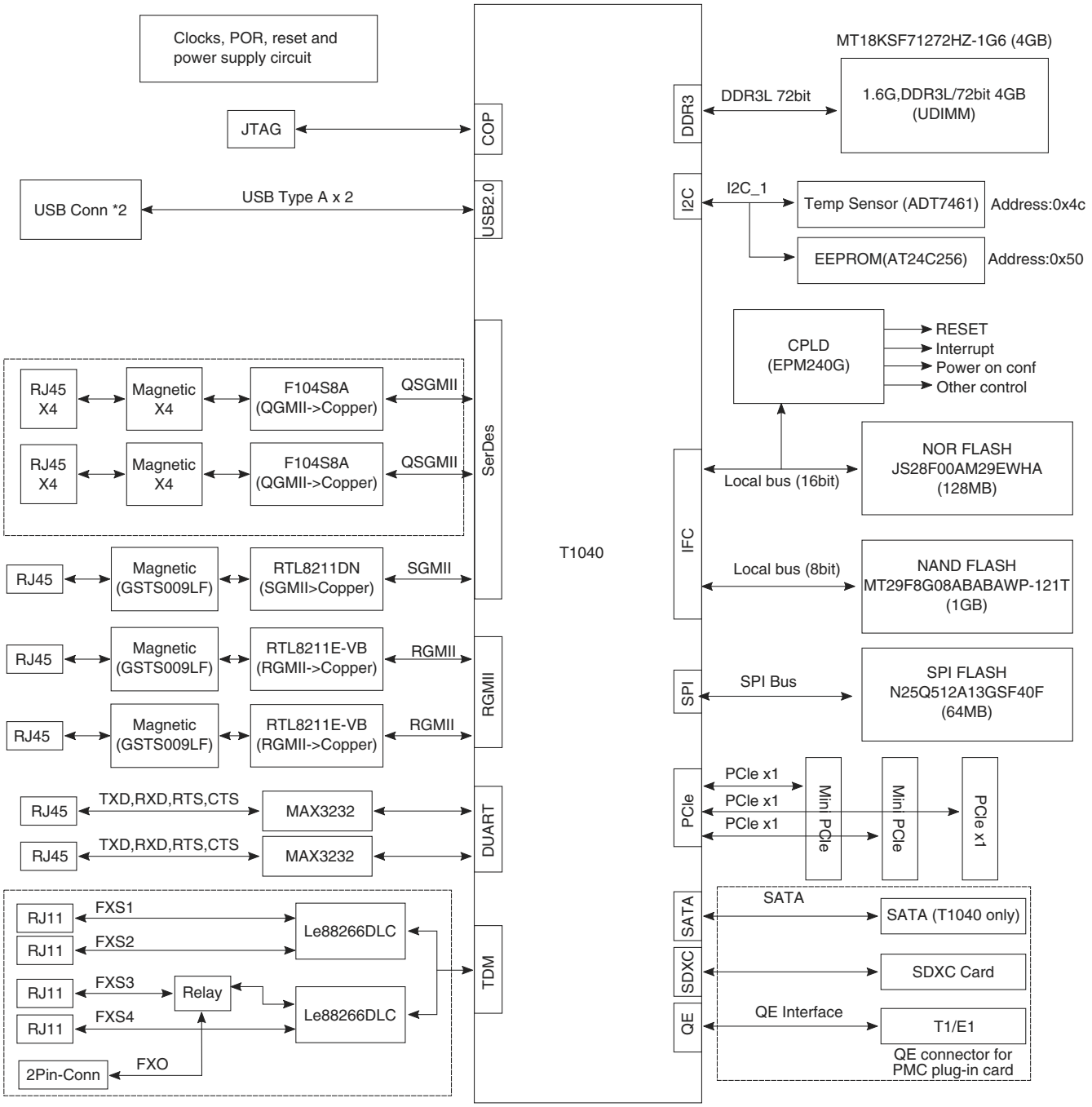

Figure 3. Block diagram

4.2 Features

Some key features of T1040RDB are:

- Freescale QorIQ processing platform
  - QorIQ T1040 communications processor with four e5500 cores, 1.4 GHz
- Memory subsystem
  - DDR3L SDRAM

        - 1 DIMM slots; supports 2 GB per DIMM
        - Supports DDR3 1600 UDIMM/RDIMM
    - NOR flash
        - 128 MB 16-bit NOR flash, MICRON: JS28F00AM29EWHA
    - NAND flash
        - 1 GB SLC NAND flash, MICRON: MT29F8G08ABABAWP-ITX:B
    - SD connector interface
    - SATA interface
- PCIe
    - One PCIe-x4 connector
    - Two mini- PCIe connectors
- USB 2.0
    - One dual USB slot, connected to USB PHY
- Ethernet
    - ETH0 - ETH 1: Connected to two independent RGMII PHY s - RTL8211E
    - ETH 2: Connected to SGMII PHY - RTL8211DN
    - ETH 3 - ETH10: Connected to two independent QSGMII PHY s - F104
- UART
    - Supports two UARTs, up to 115200 bps for console display; uses dual RJ45 slot for the two ports
- TDM
    - Supports one FXO and four FXS port

4.3 Port map

The table below shows how ETH matches to Linux and Uboot.

Label on 1U box	Port in Uboot	Port in Linux	FMAN address	Comments
ETH0	FM1@DTSEC4	fm1-gb3	0xfe4e6000	
ETH1	FM1@DTSEC5	fm1-gb4	0xfe4e8000	
ETH2	FM1@DTSEC3	fm1-gb2	0xfe4e4000	
ETH3	Not supported in u-boot	fm1-gb0	0xfe4e0000	
ETH4	Not supported in u-boot	fm1-gb0		
ETH5	Not supported in u-boot	fm1-gb0		
ETH6	Not supported in u-boot	fm1-gb0		
ETH7	Not supported in u-boot	fm1-gb1	0xfe4e2000	
ETH8	Not supported in u-boot	fm1-gb1		
ETH9	Not supported in u-boot	fm1-gb1		
ETH10	Not supported in u-boot	fm1-gb1		

5 Default boot mode

In the T1040 RDB, the boot loader, by default, executes from the NOR flash.

6 Flash image layout

Start address	End address	Image	Max size
0xEFF40000	0xEFFFFFFF	u-boot (current bank)	768 KB
0xEFF20000	0xEFF3FFFF	u-boot env (current bank)	128KB
0xEFF10000	0XEFF1FFFF	QE Ucode (current bank)	64KB
0xEFF00000	0xEFF0FFFF	FMAN Ucode (current bank)	64KB
0xED300000	0xEFEFFFFF	rootfs (alt bank)	43MB
0xEC800000	0xEC8FFFFF	Hardware device tree (alt bank)	1MB
0xEC020000	0XEC7FFFFF	Linux.uImage (alt bank)	7MB+875KB
0xEC000000	0XEC01FFFF	RCW (alt bank)	128KB
0xEBF40000	0XEBFFFFFF	u-boot (alt bank)	768KB
0xEBF20000	0XEBF3FFFF	u-boot env (alt bank)	128KB
0xEBF10000	0XEBF1FFFF	QE ucode (alt bank)	64KB
0XEBF00000	0XEBF0FFFF	FMAN UCODE (alt bank)	64KB
0xE9300000	0XEBEFFFFF	rootfs (current bank)	43MB
0XE8800000	0XE88FFFFF	Hardware device tree (cur bank)	1MB
0xE8020000	0xE86FFFFF	Linux.uImage (current bank)	7MB+875KB
0xE8000000	0xE801FFFF	RCW (current bank)	128KB

7 Default RCW setting

No	RCW words	Description
1	0x0c18000e	SYS_PLL_RAT = 1:6 (SYSCLK is 100 MHz), DDR_PLL_RAT=1:24(DDRCLK is 66.66MHz) CGA_PLL1_RAT=1:14(SYSCLK is 100 MHz)
2	0x0e000000	CGA_PLL2_RAT=1:14(SYSCLK is 100 MHz)
3	0x00000000	Reserved
4	0x00000000	Default setting
5	0x66000002	6600: Serdes protocol is 0x66 0002: MAC is operating at 2.5G interface
6	0x80000002	Serdes PLL1 operating at high frequency, Serdes PLL2 operating at low frequency DDR_FDBK_MULT=2
7	0xec027000	PBI_SRC=IFC, boot location = IFC
8	0x01000000	0100: DDR parameter

Table continues on the next page...

No	RCW words	Description
		0000: reserved
9	0x00000000	Default setting, PCIe work at Host mode.
10	0x00000000	Default setting, GPIO information.
11	0x00000000	Default setting, TDM option.
12	0x00032810	0003: UART option 0810: ASLEEP,RTC,SDHC_BASE,IRQ_OUT, IRQ_BASE,SPI_BASE option.
13	0x00000000	Default setting, IFC option
14	0x0342500f	0342: 1588,SDHC,RGMII, I2C, TDM option 500f: LVDD, L1VDD, CVDD, EVDD, HDLC, DMA option
15	0x00000000	Reserved
16	0x00000000	Reserved

8 Switch settings

8.1 Switch default setting (NOR flash boot)

SW1	0001 0011	ON ON ON OFF ON ON OFF OFF
SW2	1011 1001	OFF ON OFF OFF OFF ON ON OFF
SW3	1110 0001	OFF OFF OFF ON ON ON ON OFF

8.2 Other boot source setting

NAND boot settings:

SW1	1000 1000	OFF ON ON ON OFF ON ON ON
SW2	0011 1001	ON ON OFF OFF OFF ON ON OFF
SW3	1111 0001	OFF OFF OFF OFF ON ON ON OFF

SPI boot SW settings:

SW1	0010 0010	ON ON OFF ON ON ON OFF ON
SW2	1011 1001	OFF ON OFF OFF OFF ON ON OFF
SW3	1110 0001	OFF OFF OFF ON ON ON ON OFF

SD boot settings:

SW1	0010 0000	ON ON OFF ON ON ON ON ON
SW2	0011 1001	ON ON OFF OFF OFF ON ON OFF
SW3	1110 0001	OFF OFF OFF ON ON ON ON OFF

8.3 Switch detailed description

Switch	POR configure	Signal name	Signal meaning	Setting
SW1[1]	cfg_rcw_src0	IFC_AD8	RCW source	010011011: Hard-code RCW for JTAG debug 000100111: Nor Flash boot mode
SW1[2]	cfg_rcw_src1	IFC_AD9		
SW1[3]	cfg_rcw_src2	IFC_AD10		
SW1[4]	cfg_rcw_src3	IFC_AD11		
SW1[5]	cfg_rcw_src4	IFC_AD12		
SW1[6]	cfg_rcw_src5	IFC_AD13		
SW1[7]	cfg_rcw_src6	IFC_AD14		
SW1[8]	cfg_rcw_src7	IFC_AD15		
SW2[1]	cfg_rcw_src8	IFC_CLE		
SW2[2]	cfg_ifc_te	IFC_TE		OFF(1): IFC drives logic 0 for TE assertion.
SW2[3]	cfg_pll_config_sel_b	IFC_A18		
SW2[4]	cfg_por_ainit	IFC_A19		
SW2[5]	cfg_svr0	IFC_A16		
SW2[6]	cfg_svr1	IFC_A17		
SW2[7]	cfg_dram_type	IFC_A21		ON(0): T1040 work with DDR3L
SW2[8]	cfg_rsp_dis	IFC_AVD		
SW3[1]	cfg_eng_use0	IFC_WE_N		OFF(1): SYSCLK clock source ON (0): single-clock source using diff_sys_clk
SW3[2]	cfg_eng_use1	IFC_OE_N		
SW3[3]	cfg_eng_use2	IFC_WP_N		
SW3[4]		BOOT_FLASH_SEL		ON(0): Select NOR Flash on CS0
SW3[5]		CFG_VBANK0	Alter Flash bank	111: NOR flash bank 0 select 011: NOR flash bank 1 select
SW3[6]		CFG_VBANK1		
SW3[7]		CFG_VBANK2		
SW3[8]		TEST_SEL_N		

9 Programming flash (without Uboot)

To program flash for the first time (without Uboot), perform the following tasks:

1. In u-boot source code, add below line into boards.cfg, and run “make T1040RDB_RAM_config” to create u-boot_ram.bin:

   ```
   Active powerpc mpc85xx – freescale t104xrdb T1040RDB_RAM
   T1040RDB:PPC_T1040,RAMBOOT_PBL,
   SPIFLASH,SYS_TEXT_BASE=0x11000000,RESET_VECTOR_ADDRESS=0x110bfffc
   ```
2. Set DIP switch as given below:
   - SW1: 0100 1101
   - SW2: 1011 1001
   - SW3: 1110 0001
3. Run T1040RDB_RCW_override.cfg in CCS, to override RCW
4. Run T1040RDB_core_init.tcl in Codewarrior with Halt mode to configure DDR SDRAM
5. Download Ram u-boot u-boot_ram.bin to 0x11000000, and set PC reg to 0x110bf000,
6. Run with Codewarrior
7. After enter u-boot command, stop Codewarrior
8. Memory Download fman_ucode at 0x100000, download rcw at 0x200000, download u-boot.bin at 0x1000000
9. Run Codewarrior again
10. In u-boot, execute command:

    Erase all

    - cp.b 100000 eff00000 8000
    - cp.b 200000 e8000000 100
    - cp.b 1000000 eff40000 c0000
11. Power down, set DIP switch at below:
    - SW1: 0001 0011
    - SW2: 1011 1001
    - SW3: 1110 0001
12. Turn power on, and system will enter u-boot environment

10 Build details

10.1 Toolchain

Install toolchain provided in the release. This will be used for building the images. Here forth path in which tool-chain is installed will be refer to as <tool-chain-path>.

10.2 Building bootloader

Download the u-boot source code tarball from the Release location.

T1040RDB (NOR boot)

```
#make ARCH=powerpc CROSS_COMPILE= <tool-chain-path>/ppce5500-
fsl_networking-linux/powerpc-fsl_networking-linux- distclean

#make ARCH=powerpc CROSS_COMPILE= <tool-chain-path>/ppce5500-
fsl_networking-linux/powerpc-fsl_networking-linux- T1040RDB
```

10.3 Building RCW

Download the rcw source code tarball from the Release location. On a Linux machine, type make to build the rcw binaries.

```
#make clean
#make
```

Note that Python2 should be installed on your Linux machine.

10.4 Building Linux

Download the linux source code tarball from the Release location.

- Building uImage for 32-bit Kernel Configuration

```
#make ARCH=powerpc CROSS_COMPILE= <path>/ppce5500-fsl_networking-
linux/powerpc-fsl_networking-linux- t1040_32bit_smp_defconfig

#make ARCH=powerpc CROSS_COMPILE= <path>/ppce5500-fsl_networking-
linux/powerpc-fsl_networking-linux- uImage
```

- Building uImage for 64-bit Kernel Configuration

```
#make ARCH=powerpc CROSS_COMPILE= <path>/ppc64e5500-fsl_networking-
linux/powerpc64-fsl_networking-linux- t1040_64bit_smp_defconfig

#make ARCH=powerpc CROSS_COMPILE= <path>/ppc64e5500-fsl_networking-
linux/powerpc64-fsl_networking-linux- uImage
```

- Compiling dts (From linux directory)
  - T1040RDB

```
#./scripts/dtc/dtc -f -I dts -O dtb -R 0x8 -p 0x3000
arch/powerpc/boot/dts/t1040rdb.dts > t1040rdb.dtb
```

Note that Python2 should be installed on your Linux machine.

11 Flashing and updating images

11.1 Flashing images on NOR flash and booting from NOR Flash

Update the switch settings as mentioned below and restart the board. The board should boot with the updated images.

- RCW programming on current bank (from u-boot prompt)

  For T1040RDB, use rcw/t1040rdb/RR_P_66/ rcw_1400MHz.bin

  NOTE

  This RCW can change depending upon the requirements

```
=>    tftp 1000000 <rcw>.bin
=>    protect off all
=>    erase 0xe8000000 0xe801ffff
=>    cp.b 0x1000000 0xe8000000 $filesize
```

- Fman micro code programming on current bank (from u-boot prompt)

```
=> tftp 0x3000000 <fsl_fman_ucode>.bin
=> protect off all
Un-Protect Flash Bank # 1
=> erase 0xEFF00000 eff1ffff

. done
Erased 1 sectors
=> cp.b 3000000 0xEFF00000 10000
Copy to Flash...
9....8....7....6....5....4....3....2....1....done
```

- U-boot binary programming on current bank (from u-boot prompt)

```
=> tftp 0x1000000 u-boot.bin
=> protect off all
Un-Protect Flash Bank # 1
=> erase 0xeff40000 0xeffffff

.... done
Erased 4 sectors
=> cp.b 0x1000000 0xeff40000 0xc0000
Copy to Flash...
9....8....7....6....5....4....3....2....1....done
```

- RCW programming on alternate bank (from u-boot prompt)

  For T1040RDB, use rcw/t1040rdb/RR_P_66/ rcw_1400MHz.bin

  NOTE

  This RCW can change depending upon the requirements

```
=>    tftp 1000000 <rcw>.bin
=>              protect off all
=>    erase 0xec000000 0xec01ffff
=>    cp.b 0x1000000 0xec000000 $filesize
```

- Fman micro code programming on alternate bank (from u-boot prompt)

```
=> tftp 0x3000000 <fsl_fman_ucode>.bin
=> protect off all
Un-Protect Flash Bank # 1
=> erase 0xEbF00000 ebf1ffff

. done
```

```
Erased 1 sectors
=> cp.b 3000000 0xEbF00000 10000
Copy to Flash...
9....8....7....6....5....4....3....2....1....done
```

- U-boot binary programming on alternate bank (from u-boot prompt)

```
=> tftp 0x1000000 u-boot.bin
=> protect off all
Un-Protect Flash Bank # 1
=> erase 0xebf40000 0xebffffff

.... done
Erased 4 sectors
=> cp.b 0x1000000 0xebf40000 0xc0000
Copy to Flash...
9....8....7....6....5....4....3....2....1....done
```

11.2 Booting Linux

- Setting bootargs and serverip at u-boot prompt:

```
=>    setenv bootargs "root=/dev/ram rw console=ttyS0,115200
ramdisk_size=700000”
=>    setenv ethact  FM1@DTSEC4
=>    setenv ipaddr  <ipaddr>
=>    setenv serverip <serverip>
```

- From u-boot prompt for booting Linux with 32-bit configuration on T1040RDB

```
=>    setenv my_kern 'tftp 0x1000000  <uImage>’
=>    setenv my_fs 'tftp 0x2000000  <rfs_e5500.bin>'
=>    setenv my_dtb 'tftp 0x00c00000 <t1040rdb.dtb>'
=>    setenv my_boot bootm 0x1000000 0x2000000 0x0c00000
=>    setenv boot run my_dtb my_fs my_kern my_boot
=>    save
=>    run boot
```

- From u-boot prompt for booting Linux with 64-bit configuration on T1040RDB

```
=>    setenv my_kern 'tftp 0x1000000  <uImage_64bit>’
=>    setenv my_fs 'tftp 0x2000000  <rfs_e5500.bin>'
=>    setenv my_dtb 'tftp 0x00c00000 <t1040rdb.dtb>'
=>    setenv my_boot bootm 0x1000000 0x2000000 0x0c00000
=>    setenv boot run my_dtb my_fs my_kern my_boot
=>    save
=>    run boot
```

12 Revision history

This table summarizes revisions to this document.

Table 2. Revision history

Revision	Date	Description
Rev 0	04/2014	Initial public release.

How to Reach Us:

Home Page:
freescale.com

Web Support:
freescale.com/support

Document Number: T1040RDBQS Rev0
Agile Number: 926-28215