

SITRE sarl
 La Grande Arche – Paroi Nord
 92044 Paris laDéfense

RELEASE NOTES

for
 QNX 6.5.0 SP1 BSP for the Karo TX6Q
 (Freescale i.MX6 Quad)

QNX 6.5.0 SP1 BSP for KaRo Electronics's TX6Q StarterKit V	Version: 1.7
Release Notes & Evaluation Instructions	Date: May 15th, 2014
Auther/Owner	SITRE sarl

Document history

Date	Version	Description	Author
April 10 th , 2014	1.0	Initial document	AR
April 18 th , 2014	1.1	Formatting and review	JMD
May 15 th , 2014	1.2	Specific to TX6Q	JMD
July 09 th , 2014	1.3	Added download info	JMD

Table of content

1. Introduction	2
2. Supported Features.....	2
3. Additional Documentation	2
4. System Requirements	2
4.1. Host Requirements.....	2
4.2 Target Requirements.....	3
5. Contents of BSP	3
5.1 List of Components.....	3
5.2 Location of BSP source code and other BSP directories	4
5.3 Binaries, buildfiles, IPLs, and other files.	4
6. Booting up the Karo TX6 Board	5
6.1 Connect the hardware.....	5
6.2 Building the BSP	5
6.3 Place and run the IFS-file on the Karo TX6Q Board	6
6.4 Boot log example	6
7 Summary of driver commands	8
8 Known issues	8
9 Technical support	8
10 Download /Purchase	8

1. Introduction

This document provides instructions for running QNX 6.5.0 SP1 on the Karo TX6Q (i.MX6Q based) board.

2. Supported Features

This release contains support for the following:

- Startup code
- SMP (all cores are supported)
- Serial driver
- i2c
- MMC/SD
- USB port
- Ethernet Controller
- RTC
- Watchdog

3. Additional Documentation

For more information about i.MX6 based reference boards, please refer to Freescale's website:

http://www.freescale.com/webapp/sps/site/overview.jsp?code=IMX6X_SERIES

For the latest general QNX documentation, use the following link:

<http://www.qnx.com/developers/docs/6.5.0/index.jsp>

We recommend the sections below for an introduction to BSP building with QNX:

Document	Section
Building Embedded Systems	Making an OS image
Building Embedded Systems	Working with a BSP
QNX Momentics IDE User's Guide	Building OS and Flash Images; for customizing the BSP within the IDE

4. System Requirements

4.1. Host Requirements

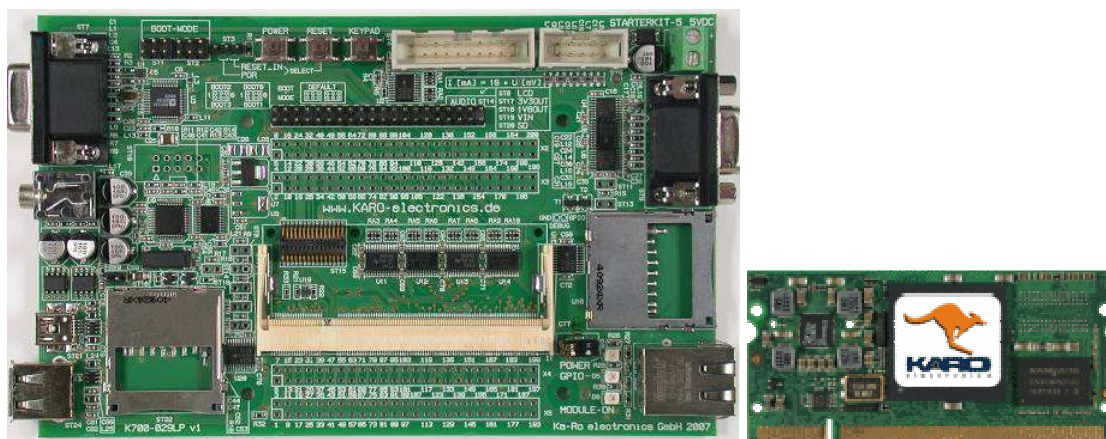
In order to use this BSP, we recommend one of the following host operating systems:

- QNX Momentics 6.5.0 SP1 as a self-hosted QNX® Neutrino® system
- Linux Red Hat Enterprise Linux 5.4 Desktop 32- and 64-bit, Red Hat Fedora 12, Ubuntu 9.10 32- and 64-bit, or openSUSE 11.2

- Microsoft Windows 7 Professional 32- and 64-bit, Windows Vista Business 32- and 64-bit, Windows XP Professional SP3, or Windows 2000 SP4.

4.2 Target Requirements

This BSP supports the **Karo TX6Q** (i.MX6 based) C(omputer) O(n) M(odule) board installed on the starter kit **Karo-STK5** board pictured below.



5. Contents of BSP

5.1 List of Components

This BSP contains the following components:

Component	Format (Binary or source)
startup	source
serial driver	source
MMC/SD driver	source
i2c driver	source
USB driver	binary
Ethernet driver	source
Watchdog timer	Source (experimental)

5.2 Location of BSP source code and other BSP directories

When you install a BSP by importing it into the QNX Momentics Integrated Development Environment, you'll find the source code for individual BSP components in directories at the root of your IDE workspace directory. BSPs installed at the command line under Windows, Linux, or self-hosted QNX, may be placed in any directory of your choice.

Note: when building the BSP at the command line in a Windows environment, run the 'bash' shell, before building the BSP.

5.3 Binaries, buildfiles, IPLs, and other files.

When a QNX BSP is extracted from the .zip archive, there are normally four directories created:

/prebuilt :

contains various header files necessary to build the source components of the BSP, as well as any pre-built binaries or libraries for which the source code is not included with the BSP.

/src :

contains the source code to all BSP components for which the source is provided, including device drivers, libraries, and utilities.

/images :

the QNX Image File System (IFS), which is the BSP image suitable for running on the target hardware, is created in this directory when the BSP is built. Any other related binaries (such as an IPL or combined IPL/IFS image) will also be created in this directory. In addition, the default IFS build file will also reside in this directory, after the BSP is built.

/install :

when the BSP is built, the contents of the /prebuilt directory are copied here. Any binaries that are generated as a result of building the BSP source (contained in the BSP's /src directory) are also copied to the /install directory. The directory structure within the /install directory reflects the structure of a typical QNX host machine. However, when the IFS image is built, the /install directory will be searched first for any required binaries or libraries, before the remaining host system directories are searched. This ensures that the intended BSP components are used in the IFS, rather than older versions of the same components, which may have been present on the host system prior to the installation of the BSP.

Here are some typical locations of various BSP components. \$BSP_ROOT_DIR is the name of the directory you extracted the BSP archive in; \$CPU_VARIANT is the particular CPU architecture that this BSP is targeted to (ex. **armle-v7**. ppcbe, x86, etc.).

File	Location
Buildfile	\$BSP_ROOT_DIR/images
IPL and/or startup	\$BSP_ROOT_DIR/install/\$CPU_VARIANT/boot/sys
“sbin” drivers (serial, flash, block, PCI, PCMCIA, USB)	\$BSP_ROOT_DIR/install/\$CPU_VARIANT/sbin
“dll” drivers (audio, graphics, network)	\$BSP_ROOT_DIR/install/\$CPU_VARIANT/lib/dll
Generic header files (not architecture specific)	\$BSP_ROOT_DIR/install/usr/include

6. Booting up the Karo TX6 Board

6.1 Connect the hardware

Connect an available serial port of your host machine (e.g. /dev/ser1 on a Neutrino host, /dev/ttyS* on Linux, COM1 on Windows, etc) with a straight-through serial cable (3 wire: TX, RX, GND) to the serial port of the Karo STK board and configure it to 115200 8N1.

6.2 Building the BSP

6.2.1 Command Line Instructions

To work with this BSP at the command line on a Windows, Linux, or self-hosted QNX create a new directory for the BSP, and copy the **bsp-karo-tx6q-vX.X-YYYYMMDD.zip** archive into that directory. Once the .zip archive is in place, extract it and build the BSP using ‘make’ utility.

The generated IFS can be found in the images directory.

6.2.2 Momentics IDE Instructions

1. Open QNX Momentics IDE
2. Click File->**Import...**->‘**QNX**’ tab and select ‘**QNX Source Package and BSP**’ and click Next
4. Select ‘**Import from local archive file**’ and browse to the BSP-zip file and click Finish
5. Notice, that **two** QNX projects have been created **bsp-karo-tx6q** and **bsp-karo-tx6q-src**.
6. Build the BSP:
 - First, build the BSP by right-clicking **bsp-karo-tx6q** project and selecting ‘Build Project’
 - Then the **bsp-karo-tx6q** can be used to select, build and modify “.ifs” components.
 - When the BSP build is finished, the resulting bootable “**karo-tx6q.raw.ifs**” file will be located in the **Images** directory.

6.3 Place and run the IFS-file on the Karo TX6Q Board

U-boot loader of the TX6 board provides many different ways to load and boot an image. At the beginning we recommend two of them: from the **SD slot** or via **TFTP** protocol.

6.3.1 Boot from dos filesystem (recommended):

- Format the SD card to have a FAT32 partition on the host (Windows or Linux).
- Once the FAT32 partition is created, simply copy the “**karo-tx6q.raw.ifs**” to the partition.
- Insert the card into the SD card slot of the Karo TX6 board and power on the board
- Press any key to stop the autoboot mode (if selected) of the U-Boot
- Change boot parameters such as (please refer to U-boot manual):

```
(set bootcmd mmc dev 0\; fatload mmc 0 0x10800000 karo-tx6q.raw.ifs\; go 0x10800000)
```

```
bootcmd=mmc dev 0; fatload mmc 0 0x10800000 karo-tx6q.raw.ifs; go 0x10800000;
```

- Save u-boot settings using “**saveenv**” command
- Reset (or power cycle) the board to boot a new image

6.3.2 Boot from the network using **TFTP** protocol:

- Place the **karo-tx6q.raw.ifs** image file on the TFTP server (Momentics has an TFTP server)
- Power on the board and set the u-boot parameters such as:

```
ipaddr=<IP address of the target>  
serverip=<IP address of the TFTP server>  
netmask=<netmask of your network in form XXX.XXX.XXX.XXX>  
bootcmd= tftpbboot 0x10800000 karo-tx6q.raw.ifs; go 0x10800000
```

- Reset (or power cycle) the board to boot a new image

6.4 Boot log example

The boot process goes quite similar independently of way, which the boot image is loaded.

6.4.1 U-Boot part (SD card)

```
U-Boot 2013.07-00502-g65c16a6 (Nov 21 2013 - 18:29:01)
```

```
CPU: Freescale i.MX6Q rev1.1 at 792 MHz  
Reset cause: SOFT  
Temperature: 51 C, calibration data 0x59f50769  
Board: Ka-Ro TX6Q-1010  
I2C: ready  
DRAM: 1 GiB  
VDDCORE set to 1421mV  
VDDSOC set to 1421mV  
VDDIO set to 3307mV  
NAND: 128 MiB  
MMC: FSL_SDHC: 0, FSL_SDHC: 1  
IPU HW Rev: 4  
CPU clock set to 792.000 MHz  
Baseboard: stk5-v3  
MAC addr from fuse: 00:0c:c6:XX:XX:XX  
Net: FEC  
Hit any key to stop autoboot: 0  
mmc0 is current device
```

reading karo-tx6q.raw.ifs

6.4.2 QNX-Boot part

```
3166068 bytes read in 159 ms (19 MiB/s)
## Starting application at 0x10800000 ...
SCU_CONFIG = 00000501, 2 cpus
Enabling Dcache and MMU
L2 cache enabled
CPU0: L1 Icache: 1024x32
CPU0: L1 Dcache: 1024x32 WB
CPU0: VFP-d32 FPSID=41033094
CPU0: NEON MVFR0=10110222 MVFR1=01111111
CPU0: 412fc09a: Cortex A9 MPCore rev 10 792MHz

FEC MAC address:00:0c:c6:XX:XX:XX

Detected i.MX6 Quad, revision T01.1

PLL1 - System      : 792MHz
PLL2 - System Bus  : 528MHz
PLL3 - OTG USB     : 480MHz
PLL4 - Audio       : 36MHz
PLL5 - Video       : 72MHz
PLL7 - Host USB    : 480MHz
PLL8 - Enet        : 50MHz
IPG clock          : 660000000Hz
IPG per clock     : 660000000Hz
UART clock        : 800000000Hz
CSPI clock        : 600000000Hz
AHB clock         : 1320000000Hz
AXI clock         : 1980000000Hz
EMI_SLOW clock    : 220000000Hz
DDR clock         : 3960000000Hz
USDHC1 clock     : 1980000000Hz
USDHC2 clock     : 1980000000Hz
USDHC3 clock     : 1980000000Hz
USDHC4 clock     : 1980000000Hz
ENFC clock       : 990000000Hz

decompressing...done
CPU1: L1 Icache: 1024x32
CPU1: L1 Dcache: 1024x32 WB
CPU1: VFP-d32 FPSID=41033094
CPU1: NEON MVFR0=10110222 MVFR1=01111111
CPU1: 412fc09a: Cortex A9 MPCore rev 10 792MHz
cpu_startnext: cpul -> fc409c0c

System page at phys:10017000 user:fc408000 kern:fc408000
Starting next program at vfe0521ec
cpu_startnext: cpu0 -> fe0521ec
QNX Neutrino 6.5 on Karo TX6 (ARM Cortex-A9 MPCore) Sitre, 20140410
switch console to /dev/ser1
Starting the I2C driver (/dev/i2c1 @ 0x68 ds1339)
Starting Ethernet driver
Starting SD1 memory card driver... (USDHC1)
Starting USB host1
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 33192
    inet 127.0.0.1 netmask 0xff000000
fec0: flags=8803<UP,BROADCAST,SIMPLEX,MULTICAST> mtu 1500
    address: 00:0c:c6:7a:39:aa
    media: Ethernet 100baseTX (none)
    inet 192.168.20.40 netmask 0x0 broadcast 0.0.0.0
# Path=0 - FREESCALE USDHC
target=0 lun=0      Direct-Access(0) - SD:2 SA08G Rev: 1.1
```

7 Summary of driver commands

The driver command lines below are specific to the **Karo TX6Q** BSP and are taken from the build file. The build file will launch all drivers by default, unless stated otherwise below. See the online QNX Momentics 6.5.0 docs for each driver (or driver family) for additional command-line options and other details.

Feature	Device	Command
Serial	i.MX6Q UART1	devc-sermx1 -u1 -e -F -S -c80000000 0x02020000,58
Ethernet	i.MX6Q ENET	io-pkt-v4 -v -dmx6x speed=100,duplex=0,verbose=5
I2C	i.MX6Q I2C1 (/dev/i2c1 @ 0x68 "ds1339")	i2c-mx35 -p 0x021A0000 -i68 -c66000000 --u 1
SD Card1 SD Card2	i.MX6Q USDHC Controller	devb-mmcsd-karo-tx6 mmcsd ioport=0x02190000,irq=54 devb-mmcsd-karo-tx6 mmcsd ioport=0x02194000,irq=55
USB	i.MX6Q USB Host	io-usb -dehci-mx28 ioport=0x02184300,irq=72,phy=0x020ca000
Real time clock	Not started.	rtc hw or rtc -s hw (Note: there there is no battery backup on the Karo TX6Q module, so RTC information will be lost when powered down).

8 Known issues

Problem	Source of Problem	Workaround
Network driver does not work at 1Gbps	hardware	When running Ethernet driver on 1Gbps link, specify speed and duplex: io-pkt-v4 -dmx6x speed=100,duplex=1

9 Technical support

This product is supported by Sitre. To obtain more information on technical support options visit: <http://www.sitre.fr/contact.html> or contact the technical support by mail: tech@sitre.fr

10 Download /Purchase

Further details about free download of binary file (evaluation only) and/or purchase of the BSP source code can be found at www.sitre.fr/telechargement.html