QCA9377 FCC Certification User's Guide

Rev 1.0 20200227



Contents

1. Introduction	3
1.1 Purpose	3
2. Preparing for hardware device	4
3. Build yocto operating system for certification	5
3.1 Set up Linux build environment	5
3.2 Get yocto BSP	6
3.3 Build image for TechNexion target platform	6
3.3.1 Configurations for setup script	6
3.3.2 Build instructions	7
3.4 Image deployment	8
3.4.1 Use mfgtool "uuu" to flash eMMC under Linux	8
4. QDART software operating instructions	9
4.1 Connect a Box PC and DUT using an Ethernet network	9
4.2 Using QDART	9
4.2.1 QDART preaction	9
4.2.2 WLAN Certification	10
4.2.3 Bluetooth Certification	13

1. Introduction

1.1 Purpose

This document describes:

- how to build yocto for certification
- how to connect devices
- $\ensuremath{\bullet}$ the steps for connect to DUT on WLAN and BT

2. Preparing for hardware device

- 1. Host PC used to build yocto operating system for certification.
- 2. TechNexion TEK3-BSW Box PC with Qualcomm QDART software.
- 3. TechNexion SoM with QCA9377 WiFi module.
- 4. Monitor(HDMI or miniDP), USB Keyboard, USB Mouse and Ethernet cable.



TechNexion SoM + Baseboard

3. Build yocto operating system for certification

3.1 Set up Linux build environment

Method 1:

Set up build environment on host PC:

Our build environment is under ubuntu 16.04.

Install required packages:

\$: sudo apt-get install gawk wget git git-core diffstat unzip texinfo gcc-multilib buildessential \ chrpath socat cpio python python3 python3-pip python3-pexpect \ xz-utils debianutils iputils-ping libsdl1.2-dev xterm \ language-pack-en coreutils texi2html file docbook-utils \ python-pysqlite2 help2man desktop-file-utils \ libgl1-mesa-dev libglu1-mesa-dev mercurial autoconf automake \ groff curl lzop asciidoc u-boot-tools libreoffice-writer \ sshpass ssh-askpass zip xz-utils kpartx vim screen

Method 2:

Download virtual machine with pre-installed Ubuntu 16.04 and packages.

ftp://ftp.technexion.net/development_resources/development_tools/vm

This virtual machine is validated to build Yocto 2.5.

Method 3:

Use the dockerfile to setup the build environment:

Install docker on ubuntu 16.04, please refer to:

https://www.digitalocean.com/community/tutorials/how-to-install-and-use-docker-on-ubuntu-16-04

After fetch yocto source code, create a docker image from a dockerfile.

\$: cd sources/meta-tn-imx-bsp/tools/container

\$: docker build -t tn_ubuntu1604 .

\$: docker run -it -u jenkins -v \${directory_in_host_machine}:\${directory_in_docker}

tn_ubuntu1604 /bin/bash

(-v: use to bind volume to the directory in host machine)

(password: jenkins)

3.2 Get yocto BSP

This BSP is a TechNexion release providing support NXP i.mx series processors:

\$: mkdir ~/bin

\$: curl http://commondatastorage.googleapis.com/git-repo-downloads/repo > ~/bin/repo

```
$: chmod a+x ~/bin/repo
```

Download the BSP source:

```
$: PATH=${PATH}:~/bin
$: mkdir edm_yocto
$: cd edm_yocto
$: repo init -u https://github.com/TechNexion/tn-imx-yocto-manifest.git -b sumo_4.14.y_GA-
next -m imx-4.14.98-2.0.1_patch.xml
$: repo sync -j8
```

3.3 Build image for TechNexion target platform

3.3.1 Configurations for setup script

Please visit TechNexion yocto github for more details: <u>https://github.com/TechNexion/tn-imx-yocto-manifest/tree/sumo_4.14.y_GA-next</u>

For PICO-IMX8MQ:

\$: TOKEN=SbtQ_mC4fvJRA88_9jB7 WIFI_FIRMWARE=y DISTRO=fsl-imx-wayland MACHINE=pico-imx8mq source edm-setup-release.sh -b build-imx8mq

For EDM-IMX8MQ:

\$: TOKEN=SbtQ_mC4fvJRA88_9jB7 WIFI_FIRMWARE=y DISTRO=fsl-imx-wayland MACHINE=edm-imx8mq source edm-setup-release.sh -b build-imx8mq

For PICO-IMX8MM:

\$: TOKEN=SbtQ_mC4fvJRA88_9jB7 WIFI_FIRMWARE=y DISTRO=fsl-imx-wayland MACHINE=pico-imx8mm source edm-setup-release.sh -b build-imx8mm

For FLEX-IMX8MM:

\$: TOKEN=SbtQ_mC4fvJRA88_9jB7 WIFI_FIRMWARE=y DISTRO=fsl-imx-wayland MACHINE=flex-imx8mm source edm-setup-release.sh -b build-imx8mm

For AXON-IMX8MM:

\$: TOKEN=SbtQ_mC4fvJRA88_9jB7 WIFI_FIRMWARE=y DISTRO=fsl-imx-wayland MACHINE=axon-imx8mm source edm-setup-release.sh -b build-imx8mm

For XORE-IMX8MM:

\$: TOKEN=SbtQ_mC4fvJRA88_9jB7 WIFI_FIRMWARE=y DISTRO=fsl-imx-wayland MACHINE=xore-imx8mm source edm-setup-release.sh -b build-imx8mm

For PICO-IMX7:

\$: TOKEN=SbtQ_mC4fvJRA88_9jB7 WIFI_FIRMWARE=y DISTRO=fsl-imx-x11 MACHINE=pico-imx7 BASEBOARD=pi source edm-setup-release.sh -b build-imx7

For EDM-IMX7:

\$: TOKEN=SbtQ_mC4fvJRA88_9jB7 WIFI_FIRMWARE=y DISTRO=fsl-imx-x11 MACHINE=edm-imx7 BASEBOARD=gnome source edm-setup-release.sh -b build-imx7

For PICO-IMX6:

\$: TOKEN=SbtQ_mC4fvJRA88_9jB7 WIFI_FIRMWARE=y DISTRO=fsl-imx-x11 MACHINE=pico-imx6 BASEBOARD=pi source edm-setup-release.sh -b build-imx6

For EDM-IMX6:

\$: TOKEN=SbtQ_mC4fvJRA88_9jB7 WIFI_FIRMWARE=y DISTRO=fsl-imx-x11 MACHINE=edm-imx6 BASEBOARD=fairy source edm-setup-release.sh -b build-imx6

For AXON-IMX6:

\$: TOKEN=SbtQ_mC4fvJRA88_9jB7 WIFI_FIRMWARE=y DISTRO=fsl-imx-x11 MACHINE=axon-imx6 source edm-setup-release.sh -b build-imx6

For PICO-IMX6UL/ULL:

\$: TOKEN=SbtQ_mC4fvJRA88_9jB7 WIFI_FIRMWARE=y DISTRO=fsl-imx-x11 MACHINE=pico-imx6ul BASEBOARD=pi source edm-setup-release.sh -b build-imx6ul

Note: You need to read and accept the EULA. Do you accept the EULA you just read? (y/n)

3.3.2 Build instructions

Set a specific DISTRO for FCC certification and build image:

\$: echo "DISTRO_FEATURES_append = \" fcc\"" >>conf/local.conf

\$: bitbake tn-image-fcc-qca

When build completes, the generated release image is under "\${BUILD-TYPE}/tmp/deploy/ images/\${MACHINE}":

To decompress the .bz2:

\$: bzip2 -fdk tn-image-fcc-qca-<MACHINE>.sdcard.bz2

Note: The special drivers and special firmwares in this image file are only used on certification and cannot be used in general mode.

3.4 Image deployment

3.4.1 Use mfgtool "uuu" to flash eMMC under Linux

Download the image deploy tool(uuu) from TechNexion FTP: <u>ftp://ftp.technexion.net/development_resources/development_tools/installer/imx-mfg-uuu-</u>

tool 20200212.zip

Please visit TechNexion yocto github wiki for more details: https://github.com/TechNexion/u-boot-tn-imx/wiki/Use-mfgtool-%22uuu%22-to-flash-eMMC

1. Install required packages for executing mfgtool uuu:

\$: sudo apt-get install libusb-1.0.0-dev libzip-dev libbz2-dev

2. First, Set the boot jumpers to Boot from serial download. Then, attach a USB Type-C peripheral cable to the board, and the other end to the host PC. Boot configuration settings:

https://www.technexion.com/support/knowledgebase/boot-configuration-settings-for-picobaseboards/

3. Excute uuu to start flashing process.

Different instructions to flash image into eMMC:

For iMX6UL/6ULL:

\$: sudo uuu/linux64/uuu -b emmc_imx6ul_img imx6ul/imx6ul-SPL imx6ul/imx6ul-uboot.img tn-image-fcc-qca-pico-imx6ul.sdcard

For iMX6DL and iMX6Q:

\$: sudo uuu/linux64/uuu -b emmc_imx6_img imx6/imx6-SPL imx6/imx6-u-boot.img tn-image-fcc-qca-pico-imx6.sdcard

For iMX7D:

\$: sudo uuu/linux64/uuu -b emmc_imx7_img imx7/imx7-SPL imx7/imx7-u-boot.img tn-image-fcc-qca-pico-imx7.sdcard

For iMX8MM:

\$: sudo uuu/linux64/uuu -b emmc_img imx8mm/pico imx8mm-flash.bin tn-image-fcc-qca-pico-imx8mm.sdcard

For iMX8M(Q):

\$: sudo uuu/linux64/uuu -b emmc_img imx8mq/pico imx8mq-flash.bin tn-image-fcc-qca-pico-imx8mq.sdcard

4. Once the flash process completes. Set the boot jumpers to Boot from eMMC and reset the board to boot from eMMC..

4. QDART software operating instructions

4.1 Connect a Box PC and DUT using an Ethernet network

Qualcomm QDART software is already installed on TEK3-BSW Box PC.



1. Plug in an Ethernet cable and power supply on the DUT.

2. Connect a monitor, keyboard, mouse, Ethernet(LAN1) and power supply to TEK3-BSW Box PC.

Note: Please replug the DUT power if there is an operation error or network disconnect.

4.2 Using QDART

4.2.1 QDART preaction

1. Open QRCT.



2. First click on the "Device Connect" to make the connection smoother.



4.2.2 WLAN Certification

1. Connect to DUT using remote device.

🔌 QCA Radio Control Toolkit
<u>File View FTM Command Tool Window Custom APIs H</u> elp
Device Connect - HW Ver QMSL Library Mode - Target -
C Disconnect
192.168.0.10:2390 (UDT WLAN)
connectiondetails=COM1,iotype=SERIAL,PORT=2391,IPADDRESS=192.168.0.10 (UDT Bluetooth)
Add New UDT
01:38:38 QLIB_GetAvailablePhonesPortList() Com Port List =

2. If successful you can see the connection message in "QRCT Debug Message".



3. Open WLAN window in "FTM Command".

🔊 QCA	A Radio	Control Toolkit					
<u>F</u> ile	<u>V</u> iew	FTM <u>C</u> ommand		<u>T</u> ool	<u>W</u> indow	Custom <u>A</u> PIs	<u>H</u> elp
Device	Conne	<u>R</u> F	×	SL Lib	orary Mode	▼ Target ▼ ↓	
QRCT De	ebug N	<u>A</u> udio					
Clear	Save	<u>B</u> T FM	•				
01:37:47 01:37:47	QLIB_S OLIB (<u>N</u> FC		NTER stO	NAL)		
Com Port I	List =	<u>W</u> LAN		ľ			
01:38:37	QLIB_S	<u>W</u> iGig		NTER	NAL)		
01:38:38 (Com Port 1	QLIB_C List =	AN <u>T</u>	١.	st()			
01:41:37	Connec	<u>W</u> iPower	_	'ranpo	rt: DLL:C:\Pi	rogram Files (x86)'	QUALCOMI

4. Select Chipset and Board Data File(BDF).

WLAN	x
Select Chipset	^
Select BDF Load DUT Skip Load DUT Close DUT screen	~
< > > Chinest Select	

5. Select QCA9377 BDF.

🔌 Please select BD file				×
← → • ↑ <mark> </mark> ›	This PC > Desktop > QCA9377	✓ Ö Searc	h QCA9377	Q
Organize 🔻 New f	folder			- 🔳 🕜
- Quick accord	^ Name	Date modified	Туре	Size
Desktop *	utfbd30-QCA9377-3.bin	8/6/2019 5:32 PM	BIN File	8 KB
🕹 Downloads 🖈				
🗎 Documents 🖈				
Pictures *				
QCA9377	¥			
Fi	ile <u>n</u> ame: utfbd30-QCA9377-3.bin	∽ bin f	ïles (*.bin)	~
			<u>O</u> pen	Cancel

6. Load DUT.

WLAN	x
Select Chipset	^
QC6174 -	
Load/Unload DUT	-
Select BDF Load DUT DataFile	
Skip Load DUT	
Close DUT screen	~
Chinest Salest	<i>,</i>

7. Start certification.

🔌 QCA Radio Control Toolkit		– 🗆 X
File View FTM Command Tool V	/indow Custom APIs Help	
Device Connect 👻 HW Ver 🛛 QMSL Libra	ry Mode 🔻 Target 👻	
QRCT Debug Message 🛛 🖛 🗆 🗙	WLAN	₽ □×
Clear Save Print	Radio Control MACOTP Settings Spectral Scan Ow Tone RSSI GoldenBinGen	^
02:33:48 QLIB_SetLibraryMode (QMSL_INTERNAL) 02:33:48 QLIB_GetAveilablePhonesPortList() Com Port List = 02:33:49 QLIB_SetLibraryMode (QMSL_INTERNAL) 02:33:50 QLIB_GetAveilablePhonesPortList() Com Port List = 02:33:54 QLIB_GetAveilablePhonesPortList() ComnectServerWithUserDefinedTranport: DLL-C: Wrogram Files (x86)QUALCOMMQDARTbin DVSL_WIA_DTransport DIP	Tx Rx Tx Rx Cont. Tx TX99 TX Mode 40 (5200) Channel (MHz)	# of Packets (0 for Cont. TX) ANI Algorithm Scrambler
VLMSL_WLAN_ITANSport.dll Resource ID: 192.168.0.10-2390 OLIB_FTM_WLAN_Atheros_LoadDUT (gc6174,D:VUbuntsShared/gualcomm/gca9377-3 wthbd30-OCA9377-3 in: 5.62)	TxPowerAuto TX Power Control 1 5 TX Power (dBm) 1500 VHT80_0 It Mode 0	AIFSN Packet Size Antenna
	RATE_AC_MCS_0_80 Data Rate TxChain0 PN15_PATTERN TX Pattern 5	TX Chain Gain Index
	Don't Use Short Guard 1 Aggregate	PA CFG
	STOP TX SET TX C Flags Setting Image: Stop TX LDPC STBC DPDmode HeavyClip Tx report 0 Good Packets 0 Temp Chain1 0	N pdadc Gain Index
Ready	Chipset Select QC MSM/MDM QMSL Int No Phone Con	nected 192.168.0.10:2390

4.2.3 Bluetooth Certification

1. Connect to DUT using remote device.

🔌 QCA Radio Control Toolkit
<u>F</u> ile <u>V</u> iew FTM <u>C</u> ommand <u>T</u> ool <u>W</u> indow Custom <u>A</u> PIs <u>H</u> elp
Device Connect V HW Ver QMSL Library Mode V Target V
C Disconnect
192.168.0.10:2390 (UDT WLAN)
connectiondetails=COM1,iotype=SERIAL,PORT=2391,IPADDRESS=192.168.0.10 (UDT Bluetooth)
0
Add New UDT
03:26:41 OLIB GetAvailablePhonesPortList()
Com Port List =

2. If successful you can see the connection message in "QRCT Debug Message".



3. Open HCI Commands window in "FTM Command \rightarrow BT".

🔌 QCA Radio	o Control Toolkit			
<u>F</u> ile <u>V</u> iew	FTM <u>C</u> ommand	<u>T</u> ool <u>W</u> indow Custom	<u>A</u> PIs <u>H</u> elp	
Device Conne	<u>R</u> F ►	SL Library Mode 🔻 🛛 Targe	et 🕶 🖕	
QRCT Debug N	<u>A</u> udio			
Clear Save	<u>B</u> T ►	<u>H</u> CI Commands		
Clear Save	<u>F</u> M	<u>E</u> PTM		
03:26:38 QLIB_S	<u>N</u> FC	LE		
Com Port List =	<u>W</u> LAN	<u>S</u> tandalone		
03:26:41 QLIB_S	<u>W</u> iGig	<u>L</u> istMode		
03:26:41 QLIB_0	AN <u>T</u> •	sıŲ		
Com Fort List = 03:28:00 Connec	<u>W</u> iPower	ranport: DLL:C:\Program File:	s (x86)\QUAL(
connectiondetails=COM1,iotype=SERIAL,PORT=2391,IPADDRESS=192.168.0.10				

4. Click Disable Legacy Support \rightarrow HCl Reset \rightarrow HCl DUT Mode.



5. Open EPTM window in "FTM Command \rightarrow BT".

🔌 QCA Radio	Control Toolkit			
<u>F</u> ile <u>V</u> iew	FTM <u>C</u> ommand	<u>T</u> ool <u>W</u> indow Custom <u>A</u> PIs		
Device Conne	<u>R</u> F	▶ SL Library Mode 👻 Target 💌		
QRCT Debug N	<u>A</u> udio			
Clear Save	<u>B</u> T	HCI Commands		
Clear Save	<u>F</u> M	<u>E</u> PTM		
03:26:38 QLIB_S	<u>N</u> FC	LE		
Com Port List =	<u>W</u> LAN	<u>S</u> tandalone		
03:26:41 QLIB_S	WiGig	ListMode		
03:26:41 QLIB_0	AN <u>T</u>			
Com Port List = 03:28:00 Connect	<u>W</u> iPower	ranport: DLL:C:\Program Files (x86		
connectiondetails=COM1, iotype=SERIAL, POR T=2391, IPADDRESS=192.168.0				
03:33:31 QLIB_F	TM_BT_SET_Legacy	rLoggingMode(0)		

6. Start certification.

🔌 QCA Radio Control Toolkit			– 🗆 🗙
File View FTM Command To	ool Window Custom APIs Help		
Device Connect - HW Ver QMS	L Library Mode V Target V		
QRCT Debug Message 🗜 🗆 🗙 Blu	etoothEPTM		4 □ ×
Clear Save Print	ttings		
Clear Save Print 04:12:52 QLIB_SetLibraryMode (QMSL_INTERNAL) 04:12:52 QLIB_GetAvailablePhonesPortList() Com Port List = 04:12:56 QLIB_GetAvailablePhonesPortList() Com Port List = 04:12:58 ConnectServerWithUserDefined Tran port DLL:C:Program Files (x86) YQMSL_BT_Transport.dll Resource ID: connectiondetails=COM1 interne_SE	Select Bluetooth Chip Test BD Address (Hex: MSB->LSB) bd 35 9c bd 35 9c 6 Space separated values Hopping Mode (Channels in Decimal) Enable Hopping 39 39 39 39 5 Space separated values	Image: Construction of the second system Image: Construction of the second system Image: Construction of the second system Image: Construction of the second system Image: Construction of the second system Image: Construction of the second system Image: Construction of the second system Image: Construction of the second system Image: Construction of the second system Image: Construction of the second system Image: Construction of the second system Image: Construction of the second system Image: Construction of the second system Image: Construction of the second system Image: Construction of the second system Image: Construction of the second system Image: Construction of the second system Image: Construction of the second system	Payload Length < 339 IDO_RANDOM Transmit Pattern hitening Enabled Rx Burst Config
RIALFOR = 2991, FADDRESS= 192.168.0.10 04:13:05 QLIB_FTM_BT_SET_LegacyLoggi ngMode(0)	Tx Only Burst Rx Only Bu	ust Rx Test Stats	Stop
04:13:06 QMCC_FTM_BT_HCI_Reset() 04:13:07 QLIB_FTM_BT_Enable_Bluetooth() 04:13:10 QLIB_FTM_BT_SET_CHIP_ID(0)	Tx Continuous 7 Power 39 Continuous Transmit Char Carrier Test Type 1 Pattern Length (1-32) 0 Pattern (Hex UINT32)	BT Receiver Test Statistics Total Packets Received : Total Packet Access Emors: Total HEC Emors: Total CRC Emors: Total Packet Bit Emors: Total Packet Bit Emors: Total First Half Bit Emors: Total Last Half Bit Emors: Avg. RSSI Reading:	
	Tx Continuous Stop	Bit Error Rate (BER %):	非數值
) Ready		MSM/MDM QMSL Int	No Phone Connected con