ConnectCore 6N

Migration Guide

Reference Manual

Revision history—90002504

Revision	Date	Description
A	December 2021	Initial release.

Trademarks and copyright

Disclaimers

Information in this document is subject to change without notice and does not represent a commitment on the part of Digi International. Digi provides this document “as is,” without warranty of any kind, expressed or implied, including, but not limited to, the implied warranties of fitness or merchantability for a particular purpose. Digi may make improvements and/or changes in this manual or in the product(s) and/or the program(s) described in this manual at any time.

Warranty

To view product warranty information, go to the following website:

www.digi.com/howtobuy/terms

Customer support

Gather support information: Before contacting Digi technical support for help, gather the following information:

Product name and model

Product serial number (s)

Firmware version

Operating system/browser (if applicable)

Logs (from time of reported issue)

Trace (if possible)

Description of issue

Steps to reproduce

Contact Digi technical support: Digi offers multiple technical support plans and service packages. Contact us at +1 952.912.3444 or visit us at www.digi.com/support.

Feedback

To provide feedback on this document, email your comments to

techcomm@digi.com

Include the document title and part number (ConnectCore 6N Migration Guide, 90002504 A) in the subject line of your email.

Contents

Migrating to the ConnectCore 6N

Hardware differences

Module identification

Certifications

Software

Migrating to the ConnectCore 6N

Due to the current market situation and to guarantee the continuous supply of products, Digi has replaced the wireless chipset of the ConnectCore 6 to the Qualcomm QCA6564A, which has better availability than the previously used AR62X3. The new Wi-Fi/Bluetooth interface is functionally compatible with the current Wi-Fi/Bluetooth interface with the following benefits:

- Update to newer Wi-Fi 5 standard (802.11ac)
- Higher throughput / better performance
- Updated Bluetooth 5.0 standard
- Continued availability and longevity

The ConnectCore 6N is a pin-to-pin compatible variant of the ConnectCore 6 SOM. The ConnectCore 6N carries the Qualcomm QCA-6564A wireless MAC, while the ConnectCore 6 has the Atheros AR6233 wireless MAC.

To facilitate the transition to the ConnectCore 6N, Digi has updated the software repositories with support for the new platform. Once these changes are applied, the resulting firmware images are compatible with both the ConnectCore 6N and the ConnectCore 6. Certification testing will be completed for North America, Canada, Europe, Japan, Australia, and New Zealand.

This document describes the process of migrating a design from a ConnectCore 6 SOM to a ConnectCore 6N SOM. It includes step-by-step instructions for adapting the firmware to support the QCA-6564A wireless MAC while retaining compatibility with existing ConnectCore 6 SOMs.

This migration guide provides all the information you need to successfully migrate your designs from the ConnectCore 6 to the new ConnectCore 6N. The following sections describe these steps in more detail.

Hardware differences

While the ConnectCore 6N is pin-to-pin compatible with the ConnectCore 6 platform, there are some internal differences.

See the ConnectCore 6N Hardware Reference Manual for more detailed information on the ConnectCore 6N platform.

Connectivity

- The ConnectCore 6N uses the Qualcomm QCA-6564A wireless MAC, whereas the ConnectCore 6 uses the Atheros AR6233. It is internally connected to the CPU in a similar way as on the ConnectCore 6:
  - The WLAN interface is tied to the CPU through the SD1 bus.
  - The Bluetooth is tied to the CPU through the UART2 bus.
  - Bluetooth enable, WLAN, and Bluetooth wakeup lines are connected to the same GPIOs.
- The ConnectCore 6N provides a single antenna; dual antennae are not supported.
- The PMIC linear regulator available on pad AA11 of the SOM has changed. It is:
  - **LDO7** on the ConnectCore 6
  - **LDO2** on the ConnectCore 6N
- The output voltage range on this regulator is different:
  - LDO7 is **0.6 - 3.6 V** on the ConnectCore 6
  - LDO2 is **0.9 - 1.86 V** on the ConnectCore 6N

The following table summarizes the differences between the ConnectCore 6 and ConnectCore 6N platforms:

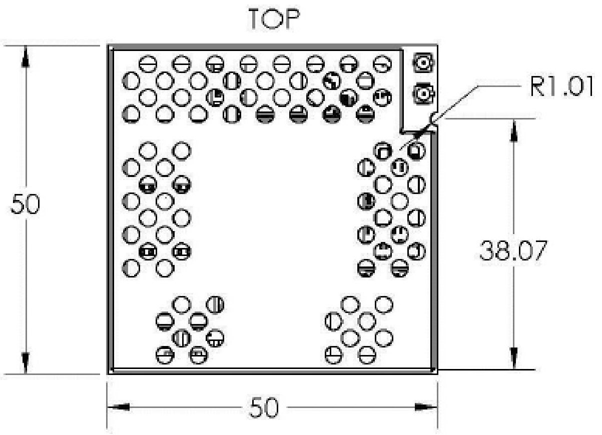
Item		ConnectCore 6		ConnectCore 6N	
Wireless MAC	**Wireless**	Atheros AR6233	IEEE 802.11 a/b/g/n	Qualcomm QCA-6564A	IEEE 802.11 a/b/g/n/ac
	Bluetooth		Bluetooth 4.0 dual mode		Bluetooth 5.0 dual mode
	BT_DISABLE# signal		Only connected to LGA pad D13		Connected to LGA pad D13 and internally to a GPIO

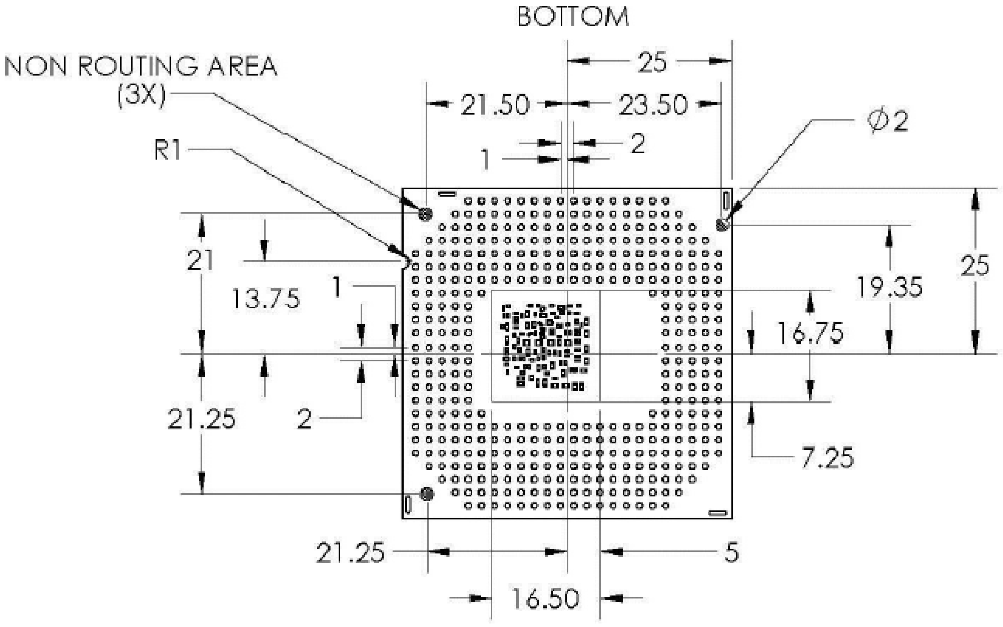
Item		ConnectCore 6	ConnectCore 6N
Pinout	**AA11**	Connected to DA9063 LDO7: ■ Default output voltage: 1.8V ■ Configurable output range: 0.9 - 3.6V ■ Maximum output current: 200mA	Connected to DA9063 LDO2: ■ Default output voltage: 1.8V ■ Configurable output range: 0.6 - 1.86V ■ Maximum output current: 200mA

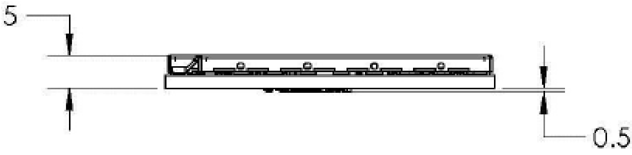
Mechanical specifications

The ConnectCore 6N platform has the same form factor as the ConnectCore 6, so their mechanical dimensions and LGA pin matrices are identical.

Note The following drawings are of the ConnectCore 6, not the ConnectCore 6N.







Note The ConnectCore 6 shield contains holes, so heat is dissipated via conduction to the shield and convection to the air. The shield on the ConnectCore 6N does not contain holes, so heat from SOM components is evacuated only to the shield through conduction and then from the shield to the air through convection. This solution used on the ConnectCore 6N is more efficient and provides better thermal management.

Power consumption

The power consumption of the ConnectCore 6N is very similar to that of the ConnectCore 6. The following tables list power consumption values for use cases described in more detail in the ConnectCore 6N Hardware Reference Manual.

Wireless disabled		
Use case	**ConnectCore 6**	**ConnectCore 6N**
Suspend	131 mA	126 mA
Power-off	5 mA	1 mA
IDLE	546 mA	540 mA
CPU Stress	1431 mA	1436 mA

Wireless enabled		
Use case	**ConnectCore 6**	**ConnectCore 6N**
Suspend	142 mA	171 mA
Power-off	5 mA	1 mA
IDLE	553 (periodic spikes to 650) mA	609 mA
CPU Stress	1426 (periodic spikes to 1550) mA	1537 mA

Wireless functionality

The ConnectCore 6N module carries the Qualcomm QCA-6564A Wireless MAC. This extends WLAN and Bluetooth capabilities.

- Wi-Fi
  - **Standards:** 802.11ac, 802.11a/b/g, 802.11n
  - **MIMO Configuration:** 1x1 (1-stream)
  - **Wi-Fi Spectral Bands:** 2.4 GHz, 5 GHz
  - **Channel Utilization:** 20/40/80 MHz
  - **Peak Speed:** 433 Mbps
  - **Security:** WPA3 Easy Connect among other authentication/encryption configurations
  - Advanced proprietary LTE/Wi-Fi co-existence
- Bluetooth

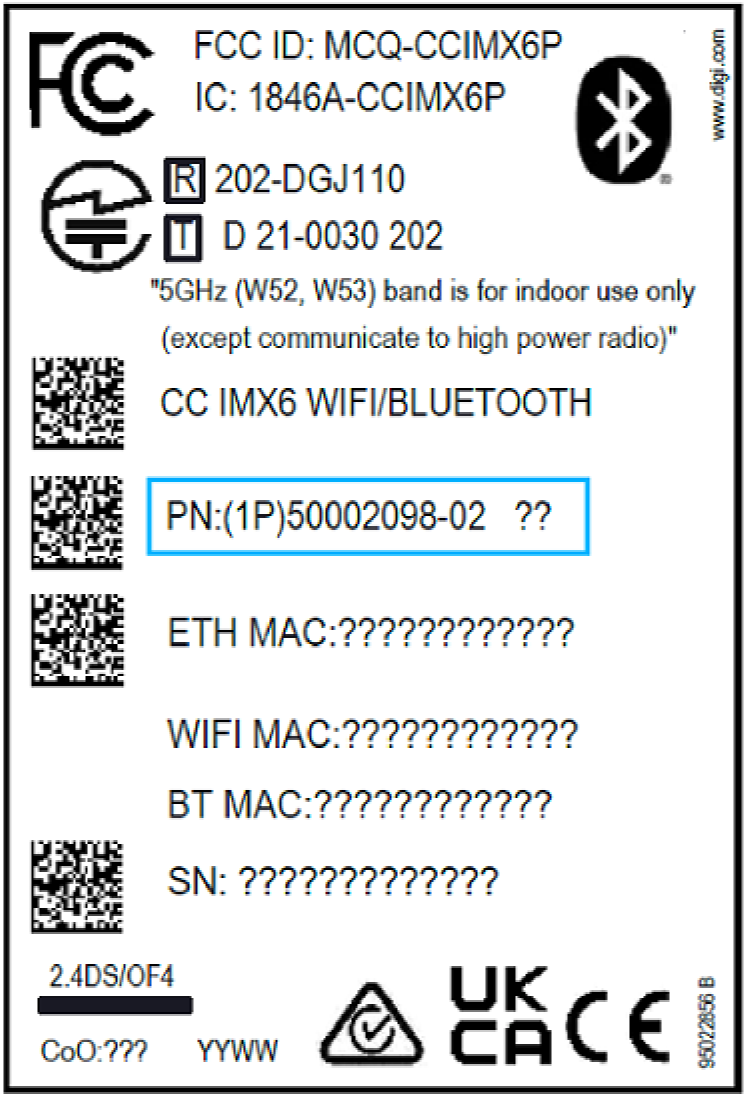
- **Bluetooth Specification Version:** Bluetooth 5.0
- **Connection Technology:** Bluetooth Low Energy
- Bluetooth supported on shared Wi-Fi antenna

Data throughput of the **QCA6564A** wireless interface is higher than the data throughput of the AR6233 wireless interface.

Module identification

To identify whether a module is a ConnectCore 6 or a ConnectCore 6N, check the part number on the label on the top of the shield.

- **50002098:** ConnectCore 6N
- **50001475:** ConnectCore 6



Identify the hardware variant in runtime

If you need to differentiate between the ConnectCore 6N and the ConnectCore 6 in runtime for your applications or scripts, use either of the following methods:

HWID

From U-Boot and/or userspace, parse the hardware version field from the HWID.

U-Boot:

```
=> hwid read
```

Userspace:

```
$ cat /proc/device-tree/digi,hwid,hv
```

- If it’s **0xA or lower**, the SOM is a ConnectCore 6.
- If it’s 0xB or higher, the SOM is a ConnectCore 6N.

Wi-Fi MAC ID

- From userspace, parse the **modalias** of the SDIO device associated with the wireless MAC:

```
$ find /sys/devices -name modalias -print0 | xargs -0 sort -u -z | grep -a "sdio:c00v0271d"
```

- If it’s sdio:c00v0271d0301, the SOM is a ConnectCore 6.
- If it’s sdio:c00v0271d050A, the SOM is a ConnectCore 6N.

Certifications

The ConnectCore 6N is based on the ConnectCore 6 Plus design. Since it shares the same PCB and the same components, the ConnectCore 6N is considered a variation of he ConnectCore 6 Plus and all certification of ConnectCore 6 Plus applies to ConnectCore 6N.

This means that for USA and Canada, the same identifier as for ConnectCore 6 *Plus* shall be used for the ConnectCore 6N (FCC ID: MCQ-CCIMX6P and IC ID: 1846A-CCIMX6P). You must make sure the product labels and product documentation reflect these new IDs.
The antenna approved for the ConnectCore 6N are also the same ones as for the ConnectCore 6 Plus. See the ConnectCore 6 Plus regulatory information and certifications for more details.

For Europe and UK, once the ConnectCore 6N module is incorporated into a product, the manufacturer must ensure compliance of the final product with articles 3.1a and 3.1b of the RE Directive (Radio Equipment Directive). A Declaration of Conformity must be issued for each of these standards and kept on file as described in the RE Directive (Radio Equipment Directive). Digi recommends you contact your local certification lab to evaluate the potential retesting required for your product.

For Japan, certifications are in progress.

Software

To facilitate the transition to the ConnectCore 6N, Digi has updated the software repositories with support for the new platform. Once these changes are applied, the resulting firmware images are compatible with both the ConnectCore 6N and the ConnectCore 6. Note that custom implementations may require additional changes.

Digi Embedded Yocto

Follow these instructions to get the latest Digi Embedded Yocto source code compatible with both the ConnectCore 6N and the ConnectCore 6.

1. Download

To get the new code, update the Digi Embedded Yocto sources by running the following commands in your Digi Embedded Yocto source folder:

```
# repo init -b <release name>
# repo sync
```

where <release name> by Digi Embedded Yocto version is

- **gatesgarth** (for DEY-3.2)
- **zeus** (for DEY-3.0)
- **thud** (for DEY 2.6)
- **rocko** (for DEY-2.4)
- **morty** (for DEY 2.2)
- **jethro** (for DEY 2.0)
- **daisy** (for DEY-1.6)

2. Integrate

Firmware images from the updated source code are compatible with both the ConnectCore 6N and the ConnectCore 6. However, any project-specific customizations you have made to Digi Embedded Yocto may need additional modification. To understand the scope of the ConnectCore 6N support and how it’s been implemented in the different components of our Digi Embedded Yocto operating system, see Description of changes (Digi Embedded Yocto). These descriptions provide the details you need to port the ConnectCore 6N support to any custom software.

3. Build

Build images from a *ccimx6sbc* Yocto project, for example, to build dey-image-qt images:

```
# bitbake dey-image-qt
```

Once the images have been built, you must perform a full system update (including an update of the bootloader) to bring all the code changes into your system. To do so, program the firmware as you usually do.

Note To facilitate the evaluation of the ConnectCore 6N, Digi also provides prebuilt images on the website at https://ftp1.digi.com/support/cc6n/digiembeddedyocto.

4. Test

Once you have installed the firmware, you can test it on both the ConnectCore 6 and the ConnectCore 6N. The interfaces affected by the code changes include:

- uSD access: Make sure uSD cards are accessible from both U-Boot and userspace.
- Wi-Fi: Verify that the wireless interface works.
- Bluetooth: Verify that the Bluetooth interface works.
- Suspend/resume: Verify that the system can go to suspend-to-ram mode and also resume from it, while verifying that the interfaces mentioned above still function after the cycle.

Description of changes (Digi Embedded Yocto)

This section lists technical details about the software changes performed in U-Boot, Linux (kernel and DTs), and the meta-digi Yocto layer.

U-Boot changes for the ConnectCore 6N

- The new U-Boot code enables LDO9 of the PMIC. On the ConnectCore 6N, LDO9 powers the uSD interface, so the change is required for the uSD interface to function.
- The new U-Boot code implements a different memory calibration due to minor layout differences in the DDR memory interface.

The latest source code that supports the new ConnectCore 6N SOM is at:

- Digi Embedded Yocto 3.2 through 2.4: https://github.com/digi-embedded/u-boot/tree/v2017.03/maint
- Digi Embedded Yocto 2.2 and 2.0: https://github.com/digi-embedded/u-boot/tree/v2015.04/maint
- Digi Embedded Yocto 1.6: https://github.com/digidotcom/yocto-uboot/tree/v2013.04/dub-2.3%2Fmaint

Linux kernel and device tree changes for the ConnectCore 6N

- Changes for all Digi Embedded Yocto versions:
  - LDO9 and LDO10 are added to the device tree and also designated as power supplies for the uSD interface and the Wi-Fi MAC, respectively.
- Changes specific to versions between Digi Embedded Yocto 3.2 and 2.2:

  - The pwrdown and disable GPIOs have been removed from the Bluetooth node in the device tree. The presence of these GPIOs triggers a sequence in the kernel code that interferes with the ConnectCore 6N Bluetooth initialization sequence done in userspace.
- Changes specific to versions between Digi Embedded Yocto 3.2 and 2.0:
  - Remove the regulator-boot-on property from LDO2 and configure it to turn off when going to suspend-to-ram. This is necessary for the ConnectCore 6N to boot.
- Changes specific to versions between Digi Embedded Yocto 2.2 and 1.6:
  - Enable the HOSTAP configuration option.
- Changes specific to versions between Digi Embedded Yocto 2.0 and 1.6:
  - Apply a backported version of the cfg80211_3.10.17 patch.
  - Enable the BT_HCIUART_H4 configuration option.

The latest source code that supports the new ConnectCore 6N SOM is at:

- **Digi Embedded Yocto 3.2:** https://github.com/digi-embedded/linux/tree/v5.4/dey-3.2%2Fmaint
- **Digi Embedded Yocto 3.0:** https://github.com/digi-embedded/linux/tree/v5.4/dey-3.0%2Fmaint
- **Digi Embedded Yocto 2.6:** https://github.com/digi-embedded/linux/tree/v4.9/dey-2.6%2Fmaint
- **Digi Embedded Yocto 2.4:** https://github.com/digi-embedded/linux/tree/v4.9/dey-2.4%2Fmaint
- **Digi Embedded Yocto 2.2:** https://github.com/digi-embedded/linux/tree/v4.1/dey-2.2%2Fmaint
- **Digi Embedded Yocto 2.0:** https://github.com/digi-embedded/linux/tree/v3.14/dey-2.0%2Fmaint
- **Digi Embedded Yocto 1.6:** https://github.com/digidotcom/yocto-linux/tree/v3.10/dey-1.6%2Fmaint

Digi Embedded Yocto changes for the ConnectCore 6N

- Modify Qualcomm driver/firmware recipes to be compatible with the ConnectCore 6 and ConnectCore 6N.
- Include Wi-Fi and Bluetooth files (drivers and firmware) for both Wi-Fi chip models (Atheros AR6233 on the ConnectCore 6 and Qualcomm QCA6564 on the ConnectCore 6N).

  Note The Atheros and Qualcomm files can co-exist on the same images without any secondary effects.

- The following packages detect the Wi-Fi chip model (either during installation or during runtime) to install files and/or run scripts depending on the chip.
  - In every Digi Embedded Yocto version:
    - **bluez5**: Bluetooth initialization script
    - **busybox**: standby/suspend script
  - In Digi Embedded Yocto versions between 3.2 and 2.2:

    - **bluez5**: main configuration file
    - **wpa_supplicant**: p2p wpa_supplicant configuration file
    - **init-ifupdown**: network interface configuration file (/etc/network/interfaces)
- Changes specific to versions between Digi Embedded Yocto 2.2 and 2.0:
    - Build U-Boot v2015.04 from the latest version rather than a tag, and modify the bootscript accordingly to account for changes in the default image type expected by U-Boot (zImage instead of uImage).
    - Update firmware-qualcomm and bluez to disable deep sleep and IBS bluetooth modes.
- Changes specific to Digi Embedded Yocto 2.0:
    - Modify kernel-module-qualcomm to load the module via udev.
    - Patch kernel-module-qualcomm to build with Linux v3.14.
- Changes specific to Digi Embedded Yocto 1.6:
    - Port QCA6564A support (firmware and module) from Digi Embedded Yocto 2.2.
    - Upgrade bluez to v5.33, backported from Digi Embedded Yocto 2.0.
    - Modify kernel-module-atheros to load the module via udev, like in kernel-module-qualcomm.
    - Update firmware-qualcomm and bluez to disable deep sleep and IBS bluetooth modes.
    - Add small changes to miscellaneous packages (pulseaudio, hostap-daemon, gst-plugins-base) to make them compatible with the rest of the changes.

The latest source code that supports the new ConnectCore 6N SOM is at:

- **Digi Embedded Yocto 3.2:** https://github.com/digi-embedded/meta-digi/tree/gatesgarth
- **Digi Embedded Yocto 3.0:** https://github.com/digi-embedded/meta-digi/tree/zeus
- **Digi Embedded Yocto 2.6:** https://github.com/digi-embedded/meta-digi/tree/thud
- **Digi Embedded Yocto 2.4:** https://github.com/digi-embedded/meta-digi/tree/rocko
- **Digi Embedded Yocto 2.2:** https://github.com/digi-embedded/meta-digi/tree/morty
- **Digi Embedded Yocto 2.0:** https://github.com/digi-embedded/meta-digi/tree/jethro
- **Digi Embedded Yocto 1.6:** https://github.com/digidotcom/meta-digi/tree/daisy

Digi Embedded for Android

Follow these instructions to get the latest Digi Embedded for Android source code compatible with both the ConnectCore 6N and the ConnectCore 6.

1. Download

To get the new code, follow these steps:

Digi Embedded for Android 8.0 (Oreo)

To get the new code, update the Digi Embedded for Android sources by running the following command in your Digi Embedded for Android 8.0 source folder:

```
# repo init -b oreo/maint
# repo sync
```

Digi Embedded for Android 4.4.3 (Kit-Kat) and 5.1.1 (Lollipop)

This process will overwrite several files of your working directory. Digi recommends you back up your code before executing these steps.

1. Download the patch file from this location:
   a. **Android 4.4.3 (Kit-Kat):** https://ftp1.digi.com/support/cc6n/digiembeddedandroid/4.4.3/DEA_4.4.3_CC6N.zip
   b. **Android 5.1.1 (Lollipop):** https://ftp1.digi.com/support/cc6n/digiembeddedandroid/5.1.1/DEA_5.1.1_CC6N.zip
2. Uncompress the patch file and follow the README instructions to install it.

2. Integrate

Firmware images from the updated source code are compatible with both the ConnectCore 6N and the ConnectCore 6. However, any project-specific customizations you have made to Digi Embedded for Android may need additional modification. To understand the scope of the ConnectCore 6N support and how it's been implemented in the different components of our Digi Embedded for Android operating system, see Description of changes (Digi Embedded for Android). These descriptions provide the details you need to port the ConnectCore 6N support to any custom software.

DEA 8.0 (Oreo)

All the required changes for this release are integrated in the public GitHub repositories:

- **U-Boot:** https://github.com/digi-embedded/u-boot/commits/v2017.03/maint
- **Kernel:** https://github.com/digi-embedded/linux/commits/android/v4.9/maint
- **QCA6564 binaries:** https://github.com/digi-embedded/android_vendor_digi_firmware/tree/oreo/maint
- **QCA6564 module:** https://github.com/digi-embedded/qcacld-2.0/tree/qca65X4/dea-8.0/maint

- **QCA6564 module integration in the platform:** https://github.com/digi-embedded/android_device_digi/tree/oreo/maint
- **Driver load/unload on suspend:** https://github.com/digi-embedded/android_platform_frameworks_base/tree/oreo/maint

DEA 4.4.3 (Kit-Kat) and DEA 5.1.1 (Lollipop)

All the required changes for these two releases are in form of a patch included in the ZIP file you have downloaded. Digi recommends you review these changes carefully before applying them to your development tree to avoid conflicts.

3. Build

You must perform a clean build of the Android sources to create up-to-date images. To do so, follow these steps:

1. Open a shell and navigate to the root of the Android sources.
2. Load the Android build environment:

   ```
   # . build/envsetup.sh
   ```

3. Prepare the system:

   ```
   # lunch <platform>
   ```

   Where <platform> is the target platform that depends on the DEA version:

   - Android 8.0 (Oreo): ccimx6sbc-eng
   - Android 4.4.3 (Kit-Kat) and Android 5.1.1 (Lollipop): imx6_ccimx6_sbc-eng

4. Clear any previous build:

   ```
   # make clobber
   ```

5. Perform a build:

   ```
   # make -j<N>
   ```

   Where *N* is the number of CPUs of your computer.

6. The resulting files are located at **<android_root>/out/target/product/<platform>/**

   Note To facilitate the evaluation of the ConnectCore 6N, Digi also provides prebuilt images on the website https://ftp1.digi.com/support/cc6n/digiembeddedandroid**.**

4. Program firmware

You must perform a full system update (including an update of the bootloader) to bring all the code changes into your system. Program the firmware as you usually do.

5. Test

Once you have installed the firmware, you can test it on both the ConnectCore 6 and the ConnectCore 6N.

- **uSD access:** make sure uSD cards are accessible from both U-Boot and userspace.
- **WiFi:** verify that the wireless interface works.
- **Bluetooth:** Bluetooth is not expected to work on ConnectCore 6N running Android.
- **Suspend/resume:** verify that the system can both go to suspend-to-ram mode and also resume from it, while verifying that the interfaces mentioned above still function after the cycle.

Description of changes (Digi Embedded for Android)

U-Boot changes for the ConnectCore 6N

- The new U-Boot code enables LDO9 of the PMIC. On the ConnectCore 6N, LDO9 powers the uSD interface, so the change is required for the uSD interface to function.
- The new U-Boot code implements a different memory calibration due to minor layout differences in the DDR memory interface.

Linux kernel and device tree changes for the ConnectCore 6N

- A new LDO2 configuration specifies that the state of LDO2 remains unchanged from how the bootloader initially configures it. This is necessary for the ConnectCore 6N to boot.
- LDO9 and LDO10 are added and also designated as power supplies for the uSD interface and the Wi-Fi MAC, respectively.
- GPIOs are no longer in the Bluetooth node. The presence of the GPIOs in the Bluetooth node triggers a kernel Bluetooth chip initialization that interferes with the ConnectCore 6N Bluetooth initialization sequence done in userspace.
- Changes specific to DEA 4.4.3 (Kit-Kat) and DEA 5.1.1 (Lollipop):
  - Applied a backported version of the cfg80211_3.10.17 patch
  - Enable the HOSTAP configuration option

Digi Embedded for Android changes for the ConnectCore 6N

- Add the QCA6564 kernel module driver source code and firmware files to the build.
- Modify the build scripts to support both MAC types in a single image (Atheros AR6233 on the CC6 and Qualcomm QCA6564 on the ConnectCore 6N).

  Note The Atheros and Qualcomm files can co-exist on the same images without any secondary effects.

- Modify the init scripts to determine the Wi-Fi MAC installed so that the system performs initialization tasks for the correct module.
- Modify the Wi-Fi HAL (hardware abstraction layer) to determine the Wi-Fi MAC installed and load/unload the correct kernel module.
- Modify the PowerManager service to disable the Wi-Fi interface and unload the QCA6564 kernel module on suspend and enable it on resume. The QCA6564 driver does no support low power mode in this version, thus the kernel driver module must be unloaded and re-loaded when the device goes to suspend and resumes. This only applies to the ConnectCore 6N module.