Qualcomm Technologies International, Ltd.

QCC514x_QCC304x.SRC.1.0 Earbud Application Quick Start

User Guide

80-CH213-1 Rev. AE

August 12, 2020

For additional information or to submit technical questions, go to: https://createpoint.qti.qualcomm.com

Qualcomm Technologies International, Ltd. is a company registered in England and Wales with a registered office at:
Churchill House, Cambridge Business Park, Cowley Road, Cambridge, CB4 0WZ, United Kingdom.
Registered Number: 3665875 | VAT number: GB787433096.

Revision history

Revision	Date	Description
AA	August 2019	Initial release. Alternative document number CS-00417564-UG.
AB	September 2019	Added information on running `test.h` functions.
AC	November 2019	Documented change to the directory path of the Earbud application. Added SODIMM Ids.
AD	April 2020	Title updated.
AE	August 2020	Updated for ADK 20.2.

Contents

Tables

1 To build and run the Earbud application

Prerequisites

- A PC with the following software installed:
  - Qualcomm® Multicore Development Environment (MDE) v2.3.2 or later.
  - An ADK Toolkit compatible with the device:

ADK Toolkit	Appropriate for
ADK_Toolkit.WIN.1.0.x.x	QCC514x, QCC304x

NOTE: The ADK Toolkit provides tools to build, deploy and debug the Earbud application, see Section 1.3. The Earbud application source code is obtained from the Qualcomm ChipCode™ repository, see the *ADK Toolkit and Source Code Installation Application Note* available in the Product Kit on CreatePoint.

- Two CDA Development boards (20-CF376-H1), one for each earbud.
- A charged battery connected to the **J2** connector on the CDA Development boards
- Two device modules one for each development board. The Earbud application is supported by the modules described in Table 1-1.

Table 1-1 Supported modules

Module	Development Kit	SODIMM
QCC5141_94WLCSP_AA	65-CG254-410	20-CH128-2
QCC5144_90VFBGA_AA	Not available	20-CH140-1
QCC3040_90VFBGA_AA	65-CG253-400	20-CH140-2
QCC3044_90VFBGA_AA	Not available	20-CH140-3
QCC3046_94WLCSP_AA	65-CG253-460	20-CH128-5

WARNING: The batteries must be sufficiently charged to run the development boards. Until the batteries are sufficiently charged the boards can be powered through the USB connector.

Process flowchart

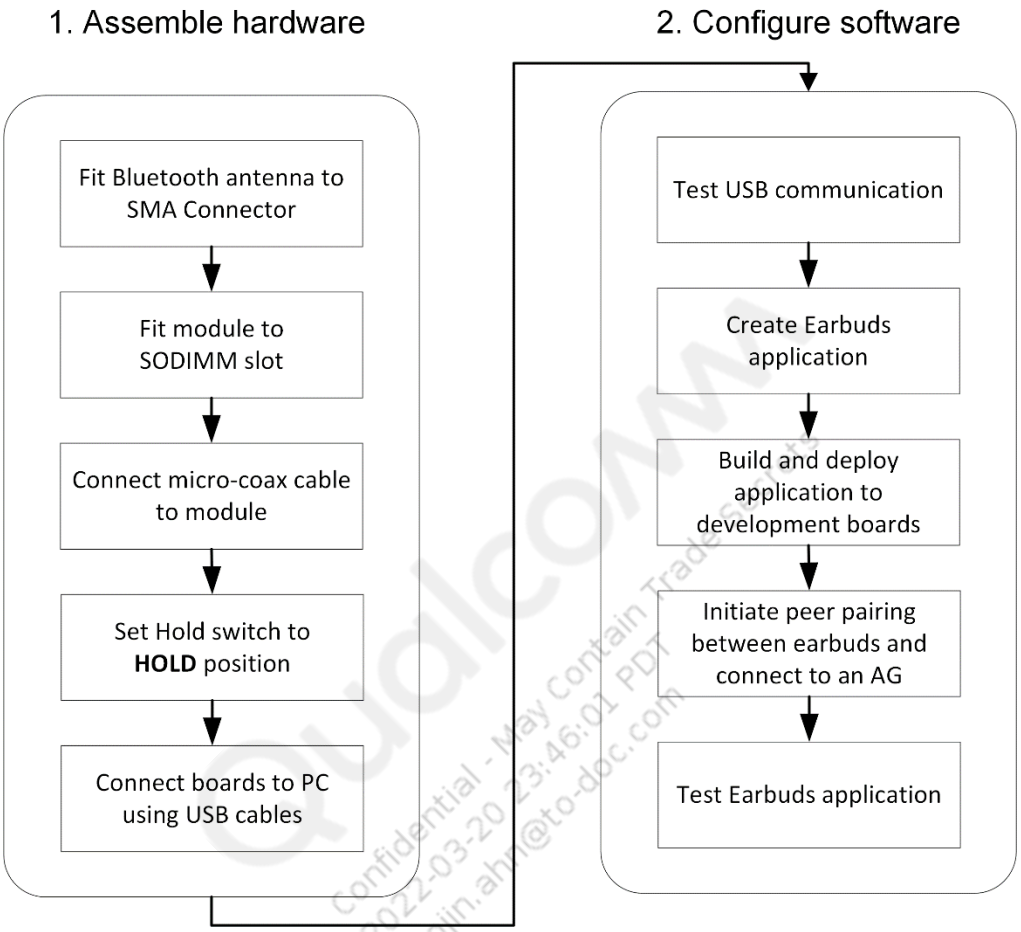
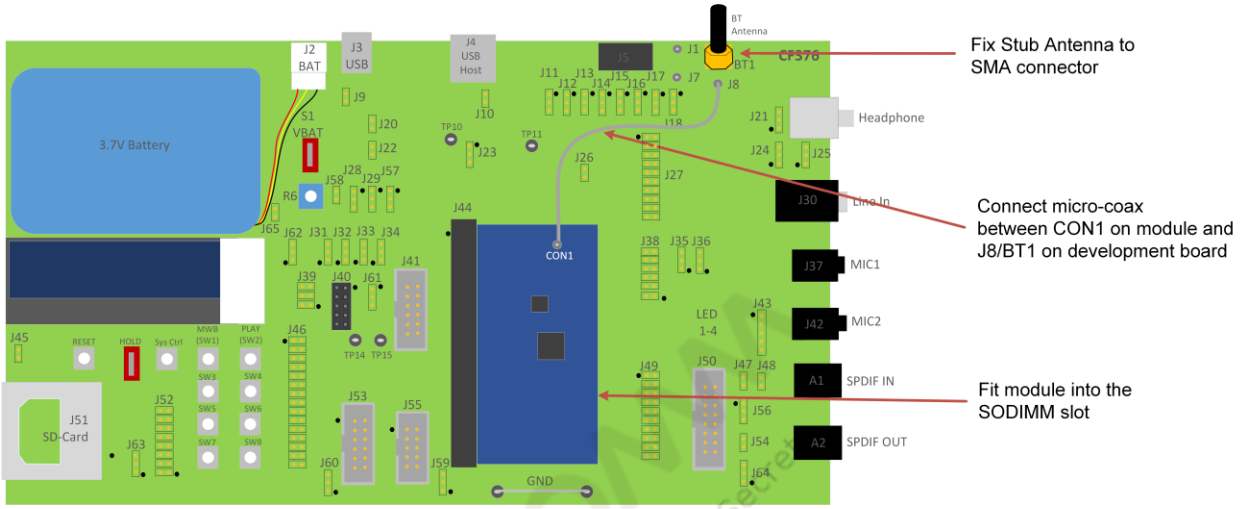


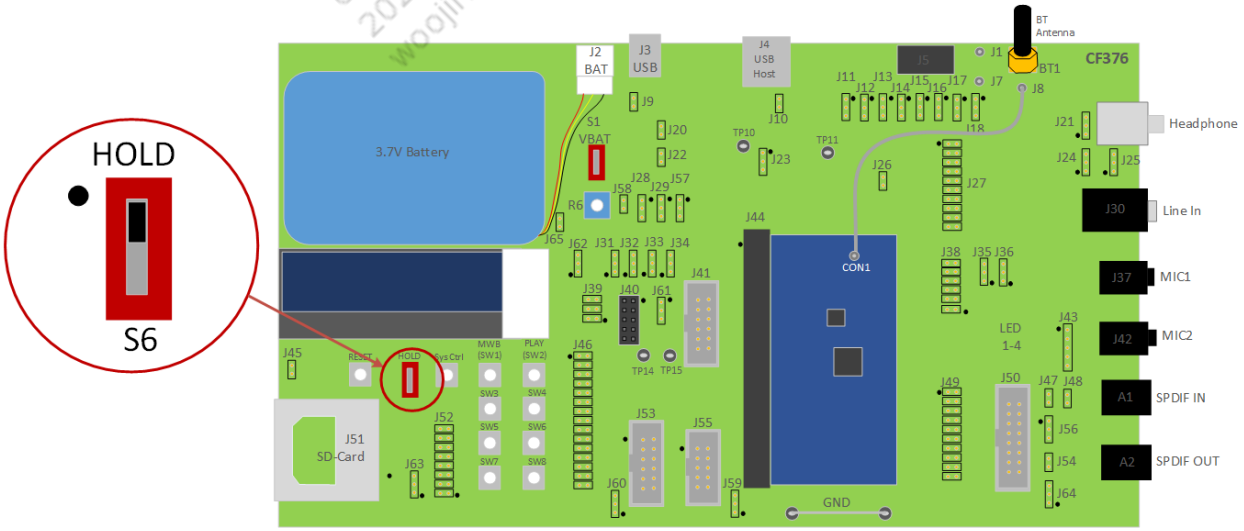
Figure 1-1 Process flowchart

1.1 Assembly of hardware components



Set up each CDA development board as follows:

1. Fit the Bluetooth antenna to the SMA connector.
2. Insert the module securely into the SODIMM slot.
3. Connect the microcoax cable to the module.
4. Ensure the battery is charged and connected to J2
5. Set the **VBAT** switch to the **OFF** position.
6. Ensure **J23** is connecting pins **1** and **2**.
7. Set the HOLD switch to the **HOLD** position:



8. Set the **VBAT** switch to the **ON** position.

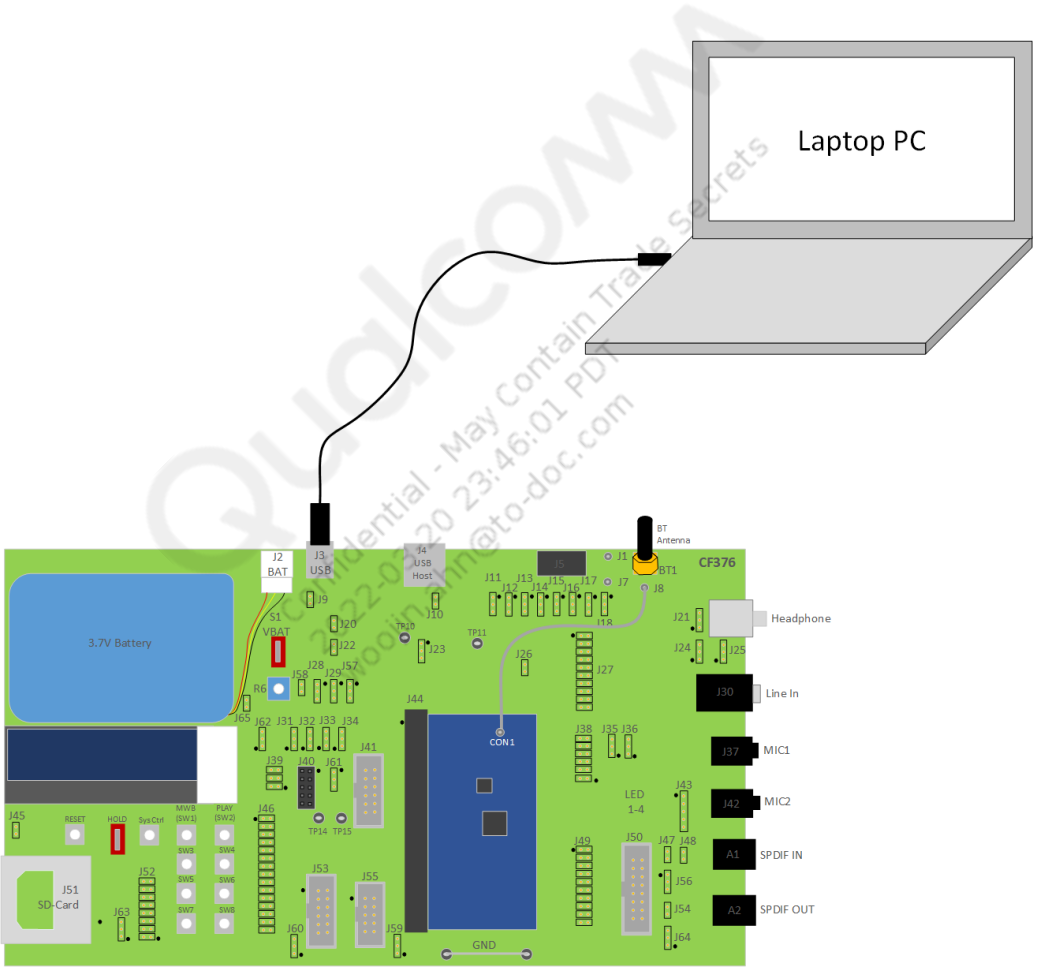
1.2 Test the USB communication

The USB drivers are distributed with the ADK Toolkit and need to be installed as part of the ADK Toolkit installation. When the module is connected, the drivers are loaded by Windows to enable USB communication to the module.

QTIL recommends checking that the correct drivers are installed and that USB is unlocked by testing the USB debug communication.

To test the USB debug communication:

1. Connect the development board to the PC using a USB cable.



2. If this is a first installation of the software listed in Chapter 1:
   a. Open a command window as administrator.
   b. Change directory:
      ```
      CD C:\qtil\ADK_Toolkit_x.x.x.x\tools\bin
      ```
   c. Run:
      ```
      echo 00000000000000000000000000000000 > c:\qtil\unlock.txt
      TransportUnlock.exe writeunlockkey c:\qtil\unlock.txt
      ```
   d. Disconnect the USB cable from the PC.
   e. Reattach the USB cable from the PC.

3. Open a command window.
4. Change directory:

   ```
   CD C:\qtil\ADK_Toolkit_x.x.x.x\tools\bin
   ```

5. Run:

   ```
   nvscmd.exe -usbdbg 1 -deviceid 4 0 identify
   ```

   If this command executes successfully the USB Debug communication with the development boards is operating correctly.

If the USB debug communication is not working use the following documents, available in the Product Kit on CreatePoint, to troubleshoot the possible causes:

- Qualcomm USB Debug Setup Guide
- Qualcomm USB Debug Drivers Installation and Troubleshooting User Guide

1.3 Build and deploy the Earbud application

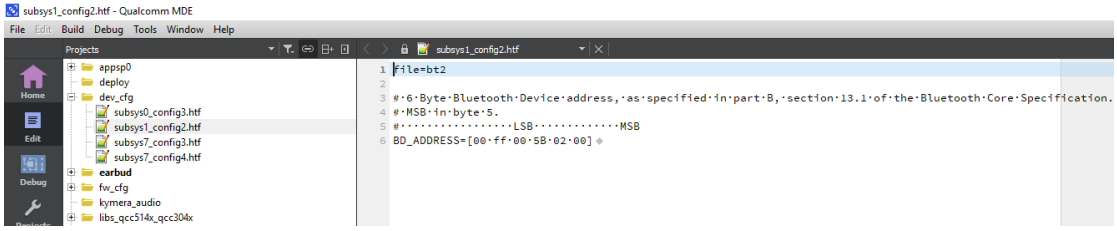
NOTE: See the *QCC514x_QCC304x.SRC.1.0 Release Note* available in the Product Kit on CreatePoint for a list of supported features, known issues and limitations.

To build an Earbud project:

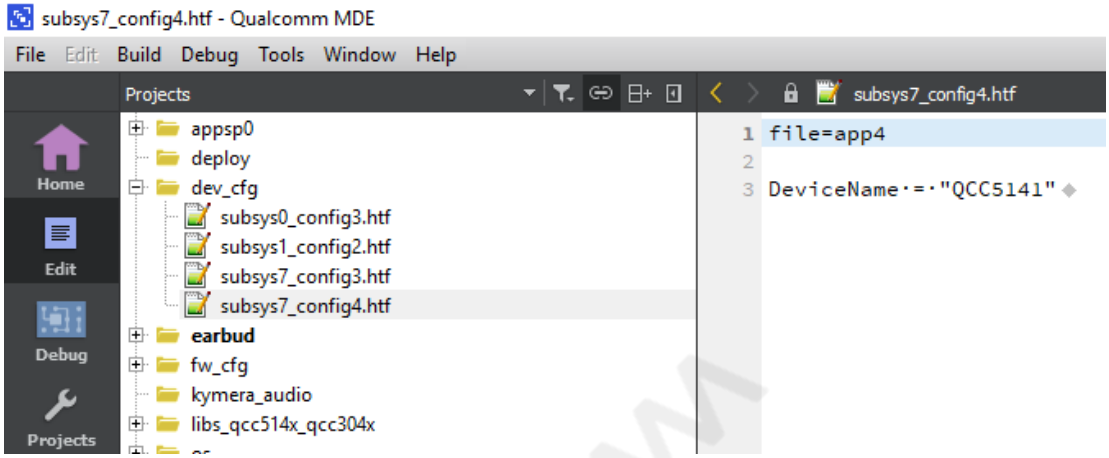
1. Open MDE.
2. Click **Open Project**.
3. Browse to the appropriate ADK workspace location on the local GitHub repository on your PC for the platform being used, for example:

   C:\qtil\QCC514x_QCC304x.SRC.1.0\earbud\workspace\QCC5141-AA_DEV-BRD-R2-AA

4. Double-click on the `earbud.x2w` file to open the project in MDE.
5. Open the `subsys1_config2.htf` file from the MDE **Projects** navigation tree:



6. Edit the value of **`BD_ADDRESS`** as required to set the Bluetooth address of the device
7. Open the `subsys7_config5.htf` file from the MDE **Projects** navigation tree:



8. Edit the value of **`DeviceName`** as required to set the friendly name of the device

NOTE: (1) The Bluetooth Addresses of both Earbud must be in the range `00025b00ff00` to `00025b00ff0f`. The bdaddr of one earbud must be even and the other earbud must have an odd bdaddr.

(2) The earbud with the even `bdaddr` channels right-channel audio and the one with the odd `bdaddr` channels the left-channel audio.

(3) The Device Name should be the same for both.

(4) For optimal performance edit the `XtalLoadCapacitance` and `XtalFreqTrim` values in the `subsys0_config3.htf` file to the values printed on the individual chip modules.

9. Press **F5**, to build and deploy the application to the connected development board.
10. Disconnect the development board you have programmed.
11. To program the other earbud, connect the second development board and repeat steps 6 and 7.

1.4 Peer Pairing and connection to a handset

The first time the earbuds are in the Out-of-Case state Peer Pairing is automatically initiated. Remove the USB cable from both development boards to emulate taking the earbuds out of the case.

The earbuds then negotiate to decide which is to currently act as the Primary earbud. The Primary earbud then becomes discoverable and the green LED flashes.

WARNING: The batteries must be sufficiently charged to run the development boards.

Using the **Bluetooth Settings** on your handset search for, and initiate pairing with, the earbud that is discovered by the handset. The handset and earbud complete pairing and then connect.

NOTE: You only need to pair with one earbud.

When the Earbuds are connected to the handset, audio on the handset is routed through the Earbud application running on the development boards.

1.5 Testing the earbuds

The Earbud application can be tested and evaluated using the test functions described in the `earbud_test.h` file.

Table 1-2 lists a small selection of useful test functions that help when tesingt the earbud application.

Table 1-2 Earbud application test functions

Function name	Description
`appTestFactoryReset()`	Initiates a factory Reset.
`appTestPhyStateInCaseEvent()`	Generates the event that indicates the earbud is now in the case.
`appTestPhyStateOutOfCaseEvent()`	Generates the event that indicates the earbud is now out of the case.
`appTestBatteryStateIsOk()`	Returns TRUE if the current battery state is `battery_level_ok`.
`appTestGetBatteryVoltage()`	Returns the current battery voltage

See the `earbud_test.h` file for a full list of the available test functions.

To run a test function:

1. Open a command window.
2. Change directory to:

   ```
   CD C:\qtil\ADK_Toolkit_1.0.x.x\apps\fw\tools
   ```

3. Enter the appropriate command:

   a. For releases upto an including r00010.1.

   ```
   python pydbg.py -d usb:<usb device id> -f
   apps1:C:\qtil\<path_to_framework_sourcecode>\adk\src\apps\earbud\work
   space\<QCC504x-AA_DEV-BRD_R2-
   AA>\depend_debug_qcc514x_qcc304x\earbud.elf
   ```

   b. For releases after r00010.1

   ```
   python pydbg.py -d usb:<usb device id> -f
   apps1:C:\qtil\<path_to_framework_sourcecode>\earbud\workspace\<QCC504
   x-AA_DEV-BRD_R2-AA>\depend_debug_qcc514x_qcc304x\earbud.elf
   ```

   This opens a pydbg session.

NOTE: `<usb device id>` is typically 101.

4. You can now issue pydbg commands using the following syntax:

   ```
   apps1.fw.call.appTestFactoryReset()
   ```

NOTE: For more information on the user experience including button configuration see the *ADK Application Framework Earbud Application Programming Guide.*

Document references

Document
ADK Application Framework Earbud Application Programming Guide
ADK_Toolkit.WIN.1.1 Release Note
ADK Toolkit and Source Code Installation Application Note
QCC512x and QCC302x/3x Development Kit and ADK <release ID> *User Guide*
Qualcomm USB Debug Setup Guide
Qualcomm USB Debug Drivers Installation and Troubleshooting User Guide
QCC514x_QCC304x.SRC.1.0 Release Note

NOTE: The referenced documents for each release are available in the specific Product Kit on CreatePoint, the <release ID> may vary depending on the Product Kit.

Terms and definitions

Term	Definition
ADK	Audio Development Kit
CDA	Combo Digital Architecture
LAP	Lower Address Part
MDE	Qualcomm Multicore Development Environment
PC	Personal Computer
QTIL	Qualcomm Technologies International, Limited
SMA	Sub Miniature version A connector
SODIMM	Small Outline Dual In-Line Memory Module
USB	Universal Serial Bus