AN12593

LPC54114 Headset with NXH3670

Rev. 0 — September 2019

Application Note

Contents

1 Introduction

1.1 Overview

The Headset contains a speaker, a microphone and some User Interface (UI) components, such as, buttons, sliders, rotary switches and LED. The main functions are summarized as below,

- **Send**: To transmit recorded audio or control signal to Dongle.
- **Receive**: To receive audio stream sent from Dongle and playback using CODEC.
- **OTA**: To receive `OTA_Headset` firmware sent from Dongle and write it to host controller's Flash.

To give the audience a systematic view of Headset in **LPC54114 BLE Audio System**, this document describes the hardware design and software architecture (top level design).

1.2 Reference documents

Table 1. References

Reference	Definition
[LPC BLE Audio System]	LPC54114 BLE Audio System introduction
[LPC Dongle]	LPC54114 USB Dongle with NXH3670
[LPC OTA]	LPC54114 BLE Audio System OTA operation steps

2 System overview

2.1 Block diagram

The block diagram of LPC54114_Headset is as shown in Figure 1.

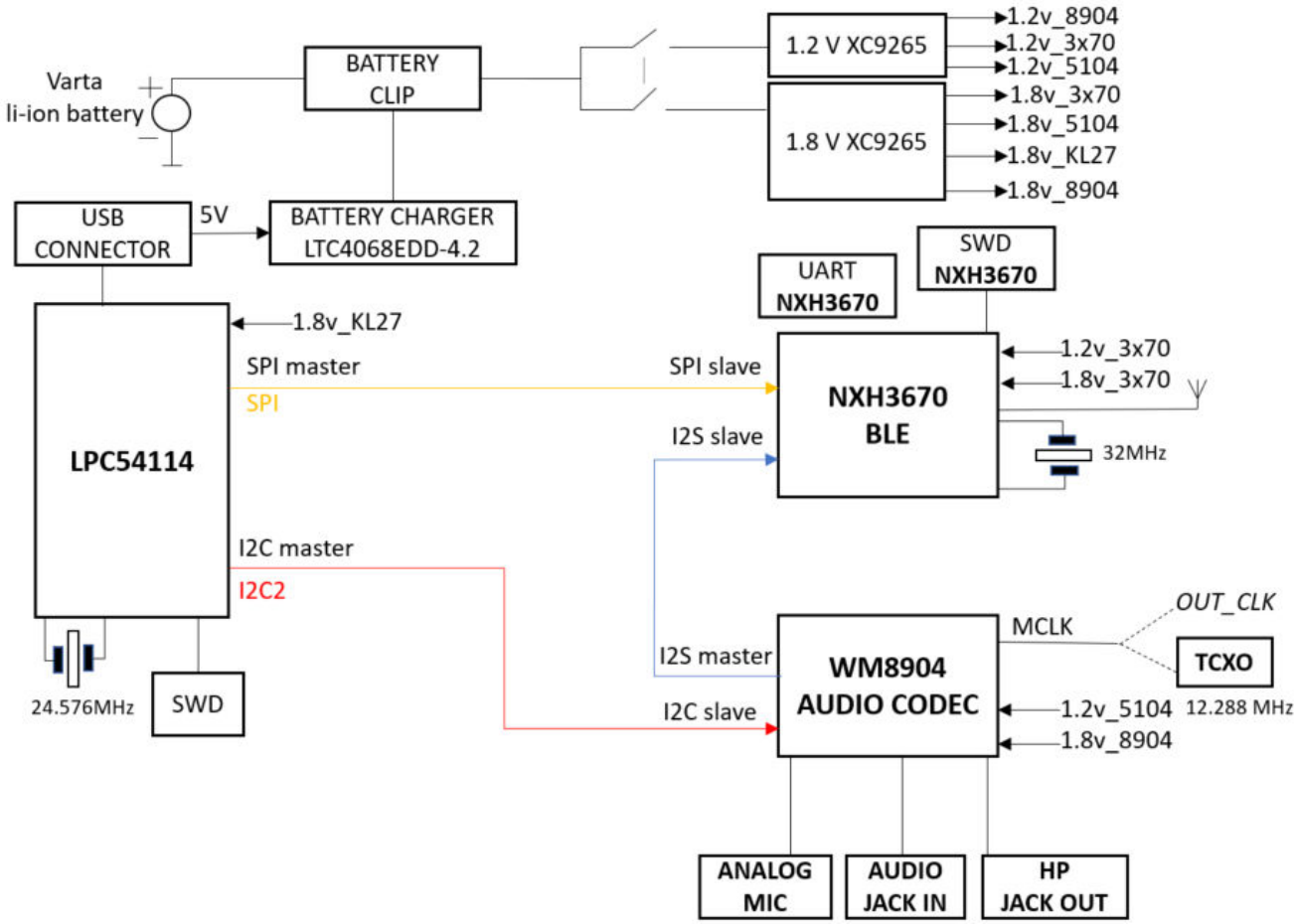

Figure 1. Block diagram of Headset configuration

As we can see in Figure 1,

- The host controller (LPC54114) is used to run Headset and OTA_Headset demos.
- The NXH3670 communicates with the LPC54114 through the SPI interface.
- CODEC (WM8904) is programed to encode or decode a digital data stream or signal, use I^2C interface to configure the CODEC in software design.
- The NXH3670 (I^2S slave) communicates with CODEC (I^2S master) directly throught the I^2S interface and no MCU processing is required. The NXH3670 receives the audio stream sent from Dongle and transmits to CODEC through the I^2S interface.

2.2 USB Headset software architecture

The software architecture of USB Headset is as shown in Figure 2.

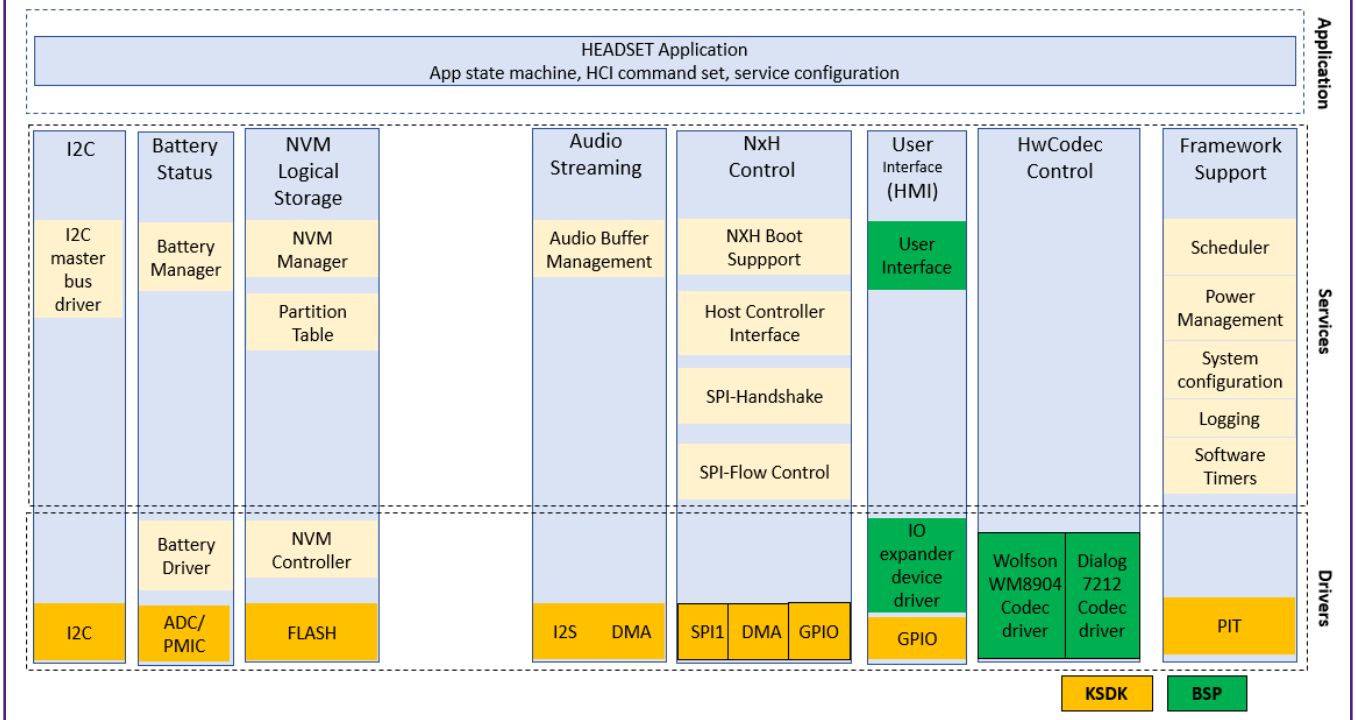

Figure 2. Headset Application Architecture

As seen in Figure 2, the Headset contains **NVM service**, **CODEC service**, **NXH service** and **UI service** . This document lists the following functions.

1. **Nvm service**: to read **Partition Table**.
2. **NxH Control**: to boot, start and transfer data with LPC54114 through the SPI interface.
3. **UI service**: to use buttons to control the volume, start and pause.
4. **CODEC service**: to configure CODEC via the I^2C interface.

In hardware design, NXH3670 and CODEC are connected through the I^2S interface. Audio data is transmitted directly from NXH3670 to CODEC through the I^2S interface, so users need to initialize the I^2C peripheral instead of the I^2S peripheral.

The audio transfer process is as shown in Figure 3.

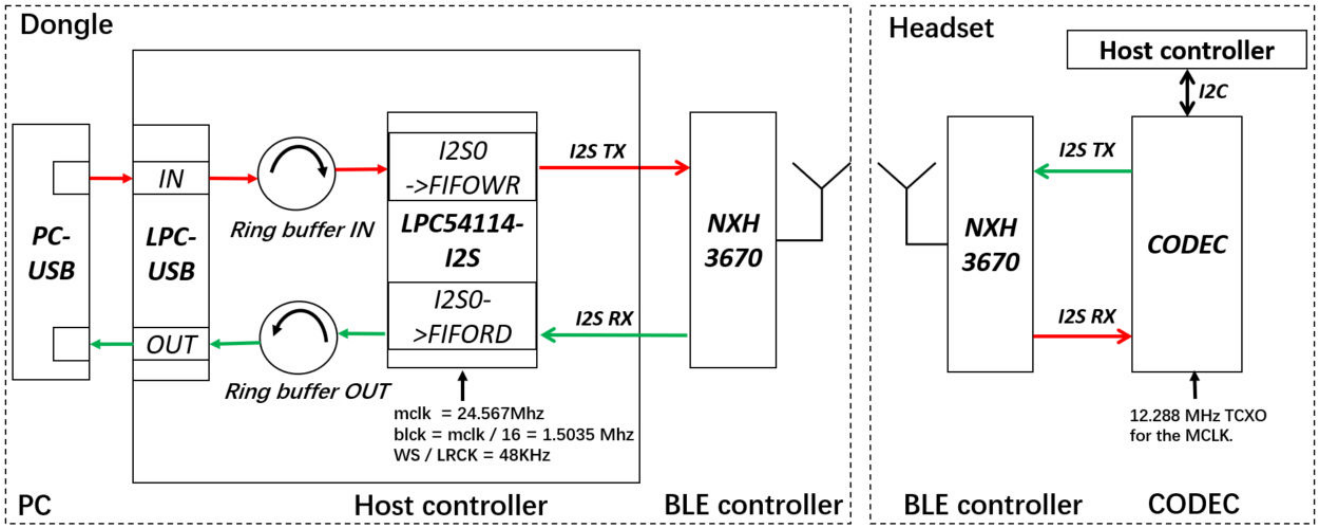

Figure 3. Audio path

Playback(forward channel): the audio path from the PC to the Headset.

Record(backward channel): The audio path from the Headset to the PC.

NOTE

As shown in Figure 3, the Headset software does not handle audio stream, so only NXH3670 and CODEC are required to tbe configured. For more information about audio transfer process, refer to LPC54114 USB Dongle with NxH3670 (document AN12568).

Users can download the demo for 48 KHz 16-bit downstream to test playback function and 16 KHz 16-bit upstream to test record function.

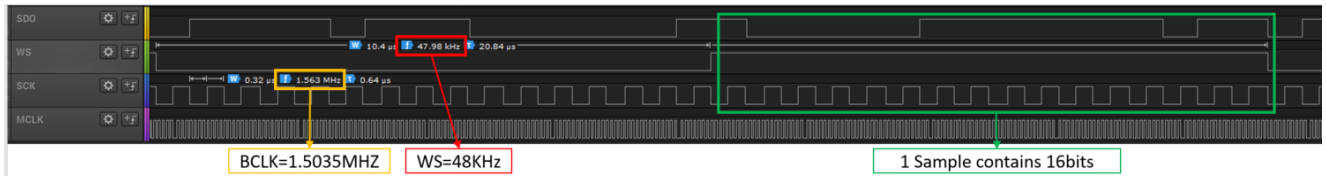

Figure 4. Sample of audio data

This document introduces the audio transfer process of the Headset section. For more information of Dongle section, refer to LPC54114 USB Dongle with NxH3670 (document AN12568).

3 Components of USB headset

3.1 LPC54114

3.1.1 Host controller

The following describes the features used in **LPC54114 USB Headset with NXH3670**. The LPC5411x are Arm® Cortex®-M4 based microcontrollers for embedded applications. These devices include:

- an optional Arm Cortex-M0+ coprocessor
- up to 192 KB of on-chip SRAM
- up to 256 KB on-chip flash
- full-speed USB device interface
- a DMIC subsystem with dual-channel PDM microphone interface and I^2S
- one 24-bit Multi-Rate Timer (MRT)
- eight flexible serial communication peripherals (each of which can be a USART, SPIs, or I2C interface)

3.1.2 Clocks

The following two crystals are used on the board.

- 32 MHz crystal connected with the NxH3670
- 12.288 MHz TCXO for the CODEC's MCLK (currently, it is an external master clock and possible to be used to output of PLL as MCLK later)

Uses can select their clock source via J10, as shown in Figure 5.

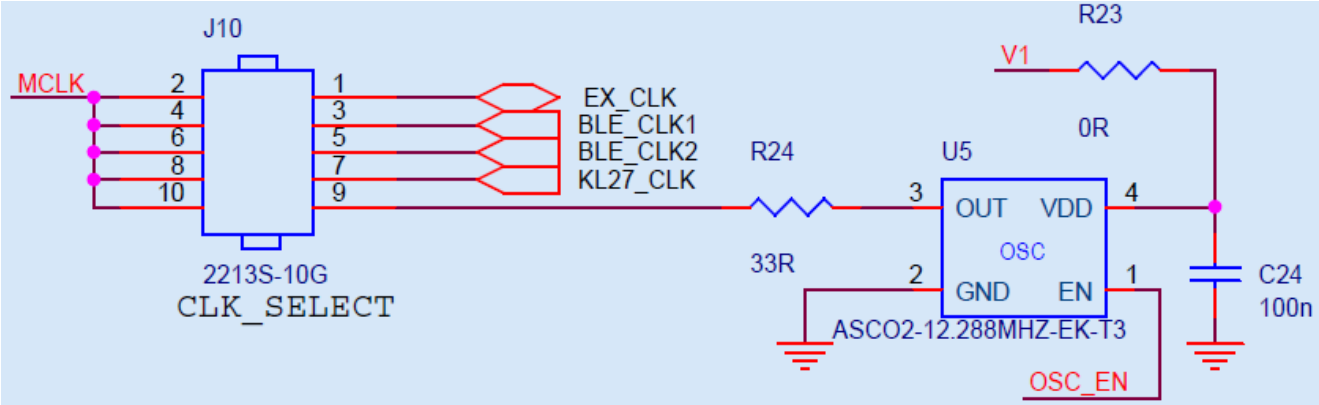

Figure 5. Selection of MCLK (CLK_SELECT)

3.1.3 Pin connections

Table 2 lists the connection information between LPC54114 and other components.

Table 2. Pin connections

Function	Jumper (LPC54114 Headset)	Name	Jumper (NXH3670)	Name
I^2S (no need to connect MCU)	—	CODEC_SDI	J12_1/9 (I2S_CONFIG)	BLE_SDO
	—	CODEC_SDO	J12_3/11(I2S_CONFIG)	BLE_SDI
	—	CODEC_WS	J12_5/13 (I2S_CONFIG)	BLE_WS
	—	CODEC_SCK	J12_7/15 (I2S_CONFIG)	BLE_SCK

Table continues on the next page...

Table 2. Pin connections (continued)

Function	Jumper (LPC54114 Headset)	Name	Jumper (NXH3670)	Name
I²C	J1_3 (PIN P0.26)	LPC54114_SDA	J11_2 (PERIPHERAL_I2C)	PH_SDA
	J1_1 (PIN P0.25)	LPC54114_SCL	J11_4 (PERIPHERAL_I2C)	PH_SCL
NXH Handshake	J2_18 (PIN P1.4)	BLE_SPIS_INTN	J16_ 9 (BLE_SPI)	SWM4 (- INTN)
	J2_20 (PIN P1.3)	BLE_SPIS_SRQ	J16_13 (BLE_SPI)	SRQ
SPI	J4_3 (PIN P0.13)	BLE_SPIS_MISO	J16_ 1 (BLE_SPI)	SW0
	J4_2 (PIN P0.12)	BLE_SPIS_MOSI	J16_3 (BLE_SPI)	SW1
	J4_4 (PIN P0.11)	BLE_SPIS_SCLK	J16_5 (BLE_SPI)	SW2
	J4_7 (PIN P0. 4)	BLE_SPIS_SSN	J16_7 (BLE_SPI)	SW3
NXH Reset	J4_8 (PIN P0.22)	BLE_RESETN	J20_5 (BLE_SWD)	POR_RESETN

As we do not make PCB for **LPC54114+NXH3670**, Figure 6 shows the demo using LPCXpresso54114 board and NXH3670 board with extra connection line.

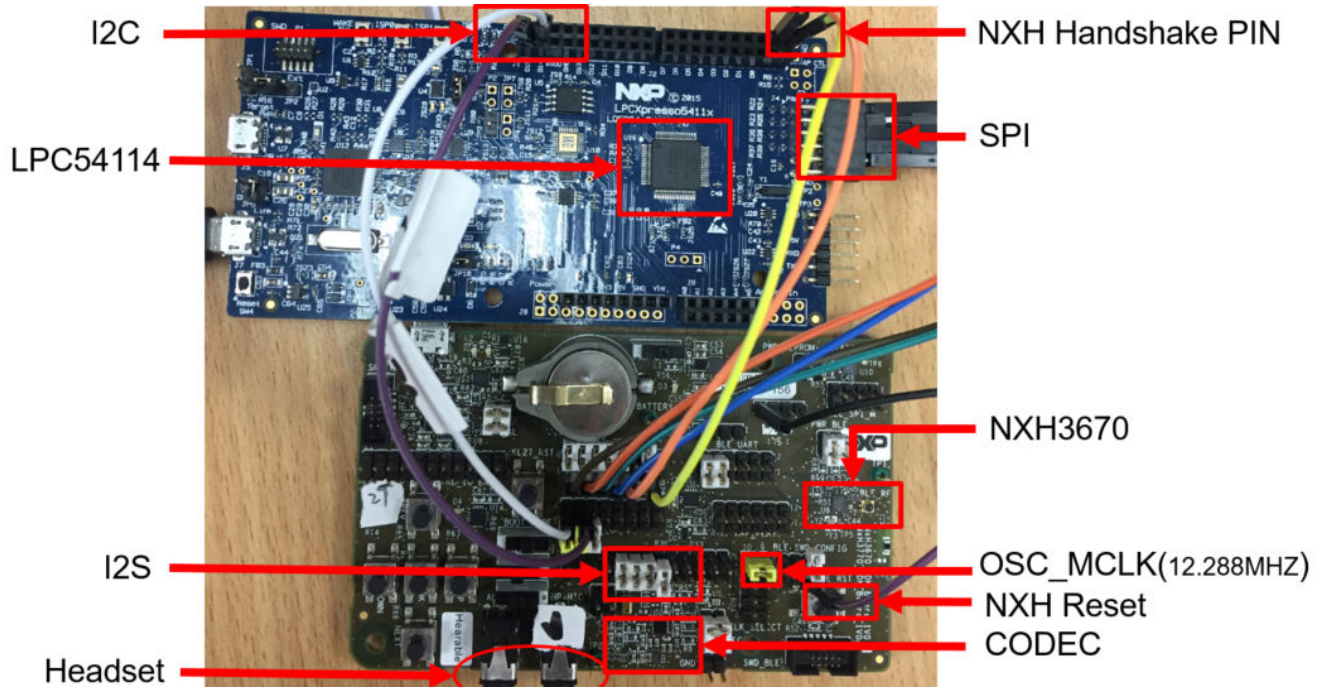

Figure 6. LPC Headset hardware design

3.1.4 Schematic

1. Audio transfer
   - I²C

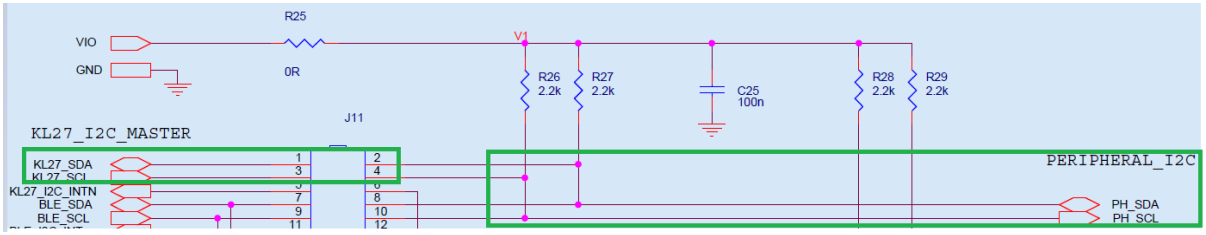

Figure 7. I²C interface connection

Audio data is transmitted directly from NXH3670 to CODEC throught the I²S interface. For the software, I²C peripheral is required to be initialized to configure CODEC instead of the I²S peripheral.

- I²S

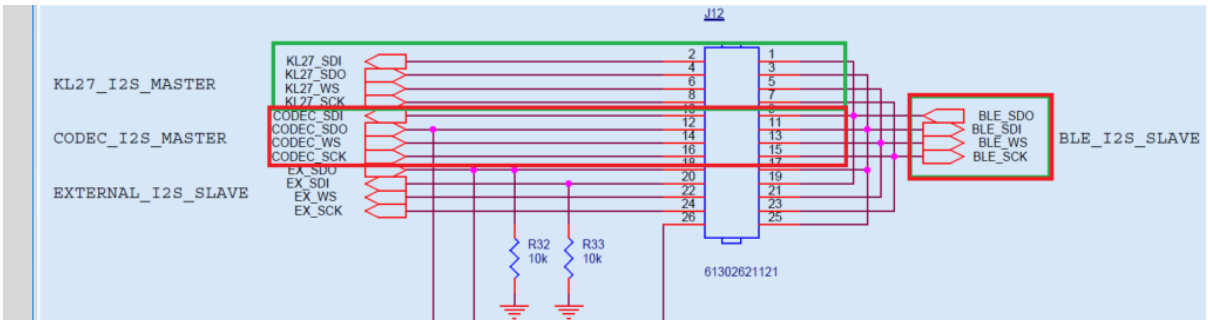

Figure 8. I²S interface connection

The NXH3670 is connected with CODEC instead of Host Controller (LPC54114) through the I²S interface, so I²S peripheral is not required to be initialized.

2. NXH3670
   - NXH Handshake

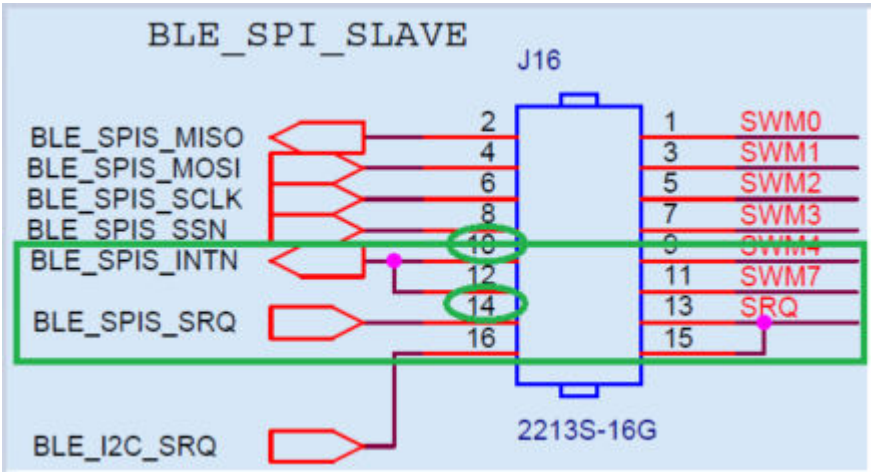

Figure 9. Handshake pin (SRQ and INTN) connection

- SPI

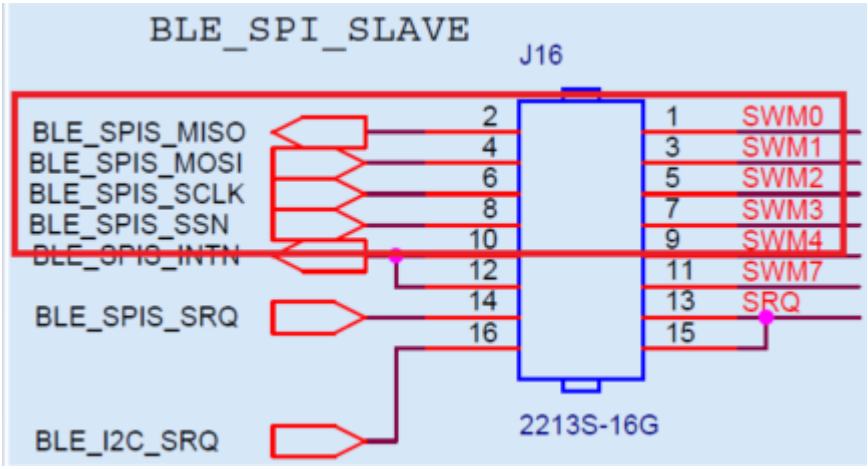

Figure 10. SPI interface connection

- Power On Reset (POR)

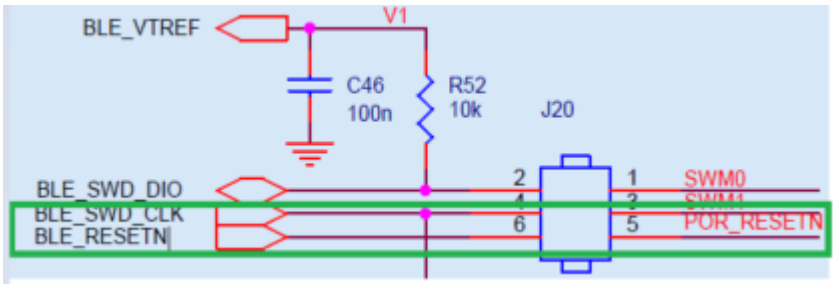

Figure 11. POR pin connection

3.1.5 Pins configurations

- SPI
  - **Interface**: SPI3
  - **Pin**: CS(P0.4), SCK(P0.11), MISO(P0.12), MOSI(P0.13)
  - **Polarity**: Active-high SPI clock (idles low)
  - **Phase**: First edge on SPSCK occurs at the middle of the first cycle of a data transfer
  - **Baud Rate**: configured to **8000000u** for SPI
- I^2C
  - **Interface**: I2C4.
  - **Pin**: SCL(P0.25), SDA(P0.26)
  - Configured to **0x1A** for i2cAddress.
- NxH3670 relevant pins
  - **INIT (P1.4)**: configured to digital input
  - **SRQ (P1.3)**: configured to digital output
  - **POR (P0.22)**: configured to digital output

3.2 NXH3670

For more information of NXH3670, refer to LPC54114 USB Dongle with NxH3670 (document AN12568).

4 Porting guide and demo introduction

Headset project is similar with Dongle. For example, the **NXH Service** part remains same. This document lists only the service configuration used in the Headset project.

4.1 I²C

No API is required to be modified in `i2c_common` and `i2c`. In SDK of KL27, APIs related to the I²C are same as of LPC54114. Figure 12 shows the configurations based on KL27 and LPC54114, which can be copied for your project.

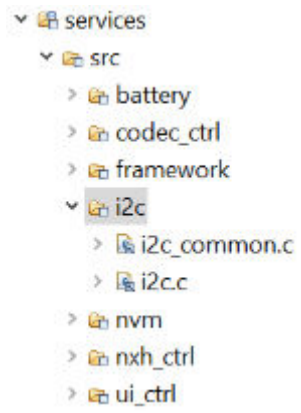

Figure 12. I²C service porting introduction

4.2 CODEC

No API is required to be modified in `codec_ctrl_power`, `codec_ctrl_updated`, and `codec_ctrl`. As CODEC is connected with BLE through the jumper, so the data is transferred between CODEC and NXH3670 without other operations.

Figure 13 shows the configurations based on KL27 and LPC54114, which can be copied for your project.

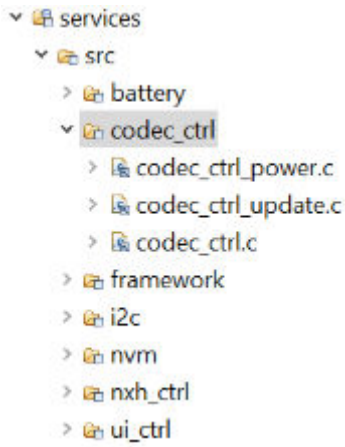

Figure 13. CODEC service porting introduction

5 Conclusions

This document describes the hardware design and software architecture (top-level design) of LPC54114_Headset in **LPC54114 BLE Ausdio System**. It can be used as a reference for your demo.

How To Reach Us

Home Page:

nxp.com

Web Support:

nxp.com/support

Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters that may be provided in NXP data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including “typicals,” must be validated for each customer application by customer's technical experts. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address: nxp.com/SalesTermsandConditions.

While NXP has implemented advanced security features, all products may be subject to unidentified vulnerabilities. Customers are responsible for the design and operation of their applications and products to reduce the effect of these vulnerabilities on customer’s applications and products, and NXP accepts no liability for any vulnerability that is discovered. Customers should implement appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP, the NXP logo, NXP SECURE CONNECTIONS FOR A SMARTER WORLD, COOLFLUX, EMBRACE, GREENCHIP, HITAG, I2C BUS, ICODE, JCOP, LIFE VIBES, MIFARE, MIFARE CLASSIC, MIFARE DESFire, MIFARE PLUS, MIFARE FLEX, MANTIS, MIFARE ULTRALIGHT, MIFARE4MOBILE, MIGLO, NTAG, ROADLINK, SMARTLX, SMARTMX, STARPLUG, TOPFET, TRENCHMOS, UCODE, Freescale, the Freescale logo, AltiVec, C-5, CodeTEST, CodeWarrior, ColdFire, ColdFire+, C-Ware, the Energy Efficient Solutions logo, Kinetis, Layerscape, MagniV, mobileGT, PEG, PowerQUICC, Processor Expert, QorIQ, QorIQ Qonverge, Ready Play, SafeAssure, the SafeAssure logo, StarCore, Symphony, VortiQa, Vybrid, Airfast, BeeKit, BeeStack, CoreNet, Flexis, MXC, Platform in a Package, QUICC Engine, SMARTMOS, Tower, TurboLink, and UMEMS are trademarks of NXP B.V. All other product or service names are the property of their respective owners. AMBA, Arm, Arm7, Arm7TDMI, Arm9, Arm11, Artisan, big.LITTLE, Cordio, CoreLink, CoreSight, Cortex, DesignStart, DynamIQ, Jazelle, Keil, Mali, Mbed, Mbed Enabled, NEON, POP, RealView, SecurCore, Socrates, Thumb, TrustZone, ULINK, ULINK2, ULINK-ME, ULINK-PLUS, ULINKpro, µVision, Versatile are trademarks or registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere. The related technology may be protected by any or all of patents, copyrights, designs and trade secrets. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. The Power Architecture and Power.org word marks and the Power and Power.org logos and related marks are trademarks and service marks licensed by Power.org.

© NXP B.V. 2019. **All rights reserved.**

For more information, please visit: http://www.nxp.com

For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: September 2019

Document identifier: AN12593

arm