AN12567

Getting Started with LPC54114 + NxH3670 Gaming Use-case

Rev. 0 — September 2019 Application Note

Contents

1 Introduction.....1
2 Hardware introduction..... 2
3 Software introduction.....6
4 Verification..... 12
5 Conclusion..... 15

1 Introduction

1.1 Overview

This document describes the hardware design of **LPC54114 BLE Audio System** and software architecture (top-level design) of Host Controller (LPC54114). This document is intended to be used by those who have a systematic view of **LPC54114 BLE Audio System**. For more introduction about Dongle ([LPC Dongle]), Headset ([LPC Headset]) and OTA ([LPC OTA]), refer to relevant application notes.

- In the software section, use relevant services to run the reference application and store NxH3670-BLE firmware. The relevant demos include:
  - Dongle (to transfer audio stream to Headset)
  - OTA_ Dongle (to transfer binary/firmware file to Headset)
  - Headset (to play audio stream and record)
  - OTA_ Headset (to receive firmware and update Headset)
- The hardware includes:
  - a Dongle that can run Dongle and OTA_Dongle demos
  - a Headset board that can run Headset and OTA_Headset demos

1.2 Summary

This document provides the necessary information on how to get started on the **LPC54114 BLE Audio System** based on LPCXpresso54114 board and NxH3670 SDK boards.

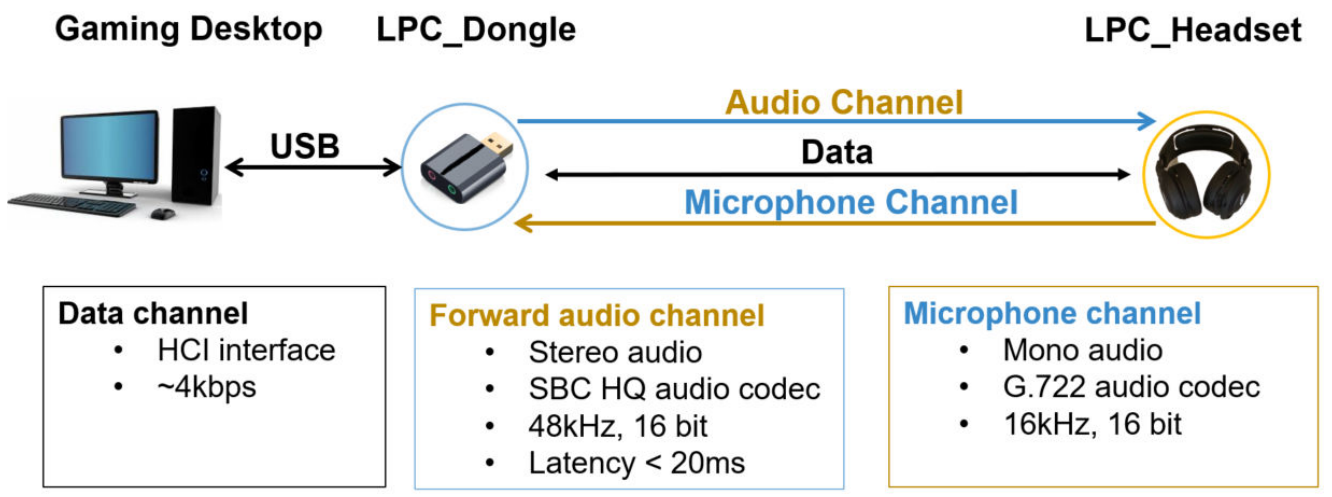

Figure 1. LPC54114 + NxH3670 wireless headset use case

The system consists of a Dongle and a Headset, using LPC54114 as Host Controller.

- **Dongle**: The Dongle has a USB interface that connects to PC. Dongle is responsible to set up a wireless audio link with Headset.
- **Headset**: The Headset has a speaker, a microphone and some User Interface (UI) components, such as,buttons, sliders, rotary switches and LED. Headset is responsible for receiving audio data from Dongle and sending the recorded audio to Dongle.

1.3 Reference documents

Table 1. References

Reference	Definistion
[LPC Dongle]	LPC54114 USB Dongle with NxH3670
[LPC Headset]	LPC54114 USB Headset with NxH3670
[LPC OTA]	OTA operation steps of **LPC54114 BLE Audio System**

2 Hardware introduction

2.1 System overview

The simplified system overview is as shown in Figure 2.

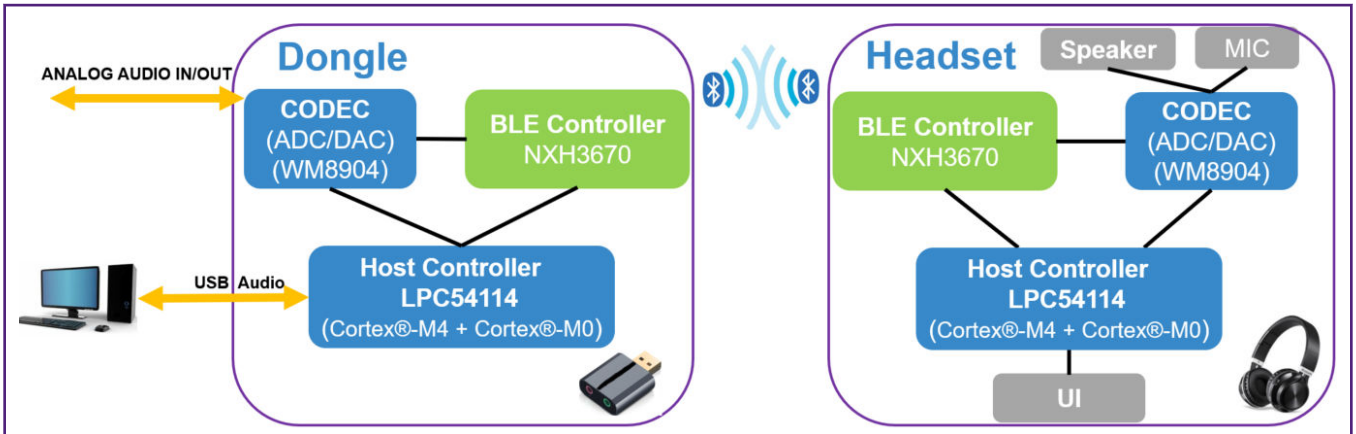

Figure 2. Simplified system overview

As seen in Figure 2, the main steps of the audio transfer process are:

1. The NxH3670 can boot, start and communicate with LPC54114 through the SPI interface.
2. Assuming the NxH3670 can work well after Step 1, we use the USB interface to transfer audio stream from PC to host controller. The host controller will convert the 48 kHz USB audio to an I^2S signal, and then transmit it to NxH3670 of Dongle through I^2S interface.
3. The audio stream can be transmitted to NxH3670 of Headset automatically.

Now, users can hear voice using their Headset.

The current **LPC54114 BLE Audio System** includes LPCXpresso54114 board and NxH3670 SDK boards (KL27 Dongle and Headset board is used to provide the basic Audio function respectively). The platform can:

- send audio stream from PC to Headset.
- receive control signal and recorded audio from Headset to PC.
- update new firmware through Over The Air (OTA).

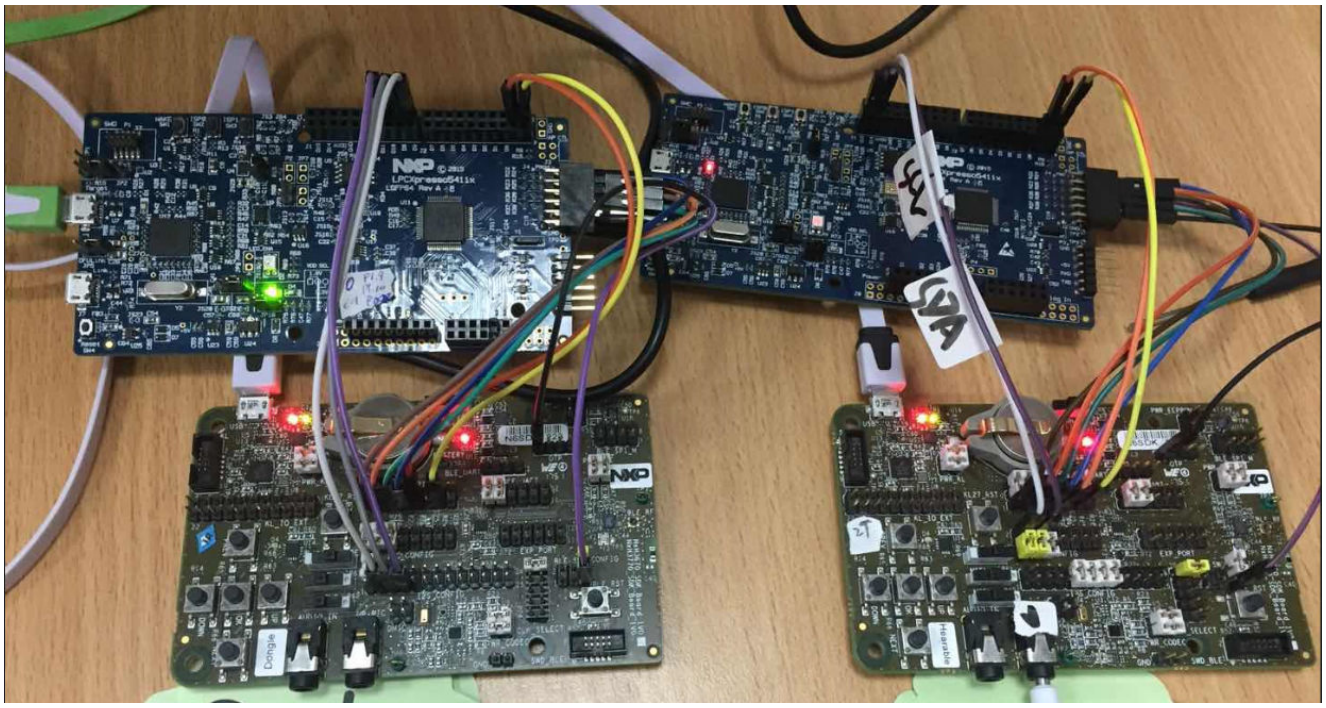

Figure 3. LPC54114 BLE Audio System hardware

As seen in Figure 3,

- In **Dongle** part, the host controller is LPC54114 and the BLE device is NxH3670 on KL27 Dongle board. LPC54114 can configure and communicate with NxH3670 through the SPI interface. LPC54114 transfer audio data to NxH3670 through the I^2S interface.
- In **Headset** part, the host controller is LPC54114 and the BLE device is NxH3670 on KL27 Headset board. LPC54114 can configure and communicate with NxH3670 through the SPI interface and configure CODEC using the I^2C interface.

2.2 LPC Dongle

This section gives current hardware design of LPC Dongle based on LPCXpresso54114 and KL27 Dongle board, as well as components and interfaces.

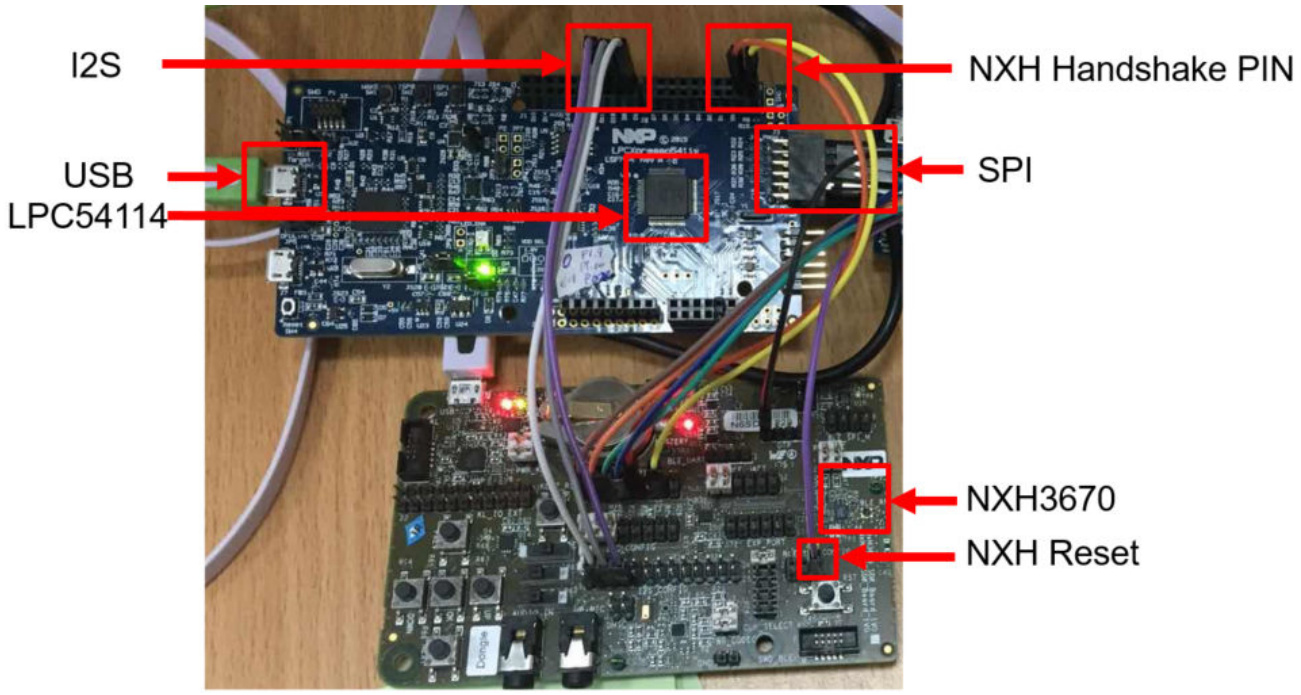

Figure 4. LPC dongle hardware design

Use a USB cable to connect J5 (LPCXpresso54114) with PC to power ON or download firmware.

2.3 LPC Headset

This section gives current hardware design of LPC Headset based on LPCXpresso54114 and KL27 Headset board, as well as components and interfaces.

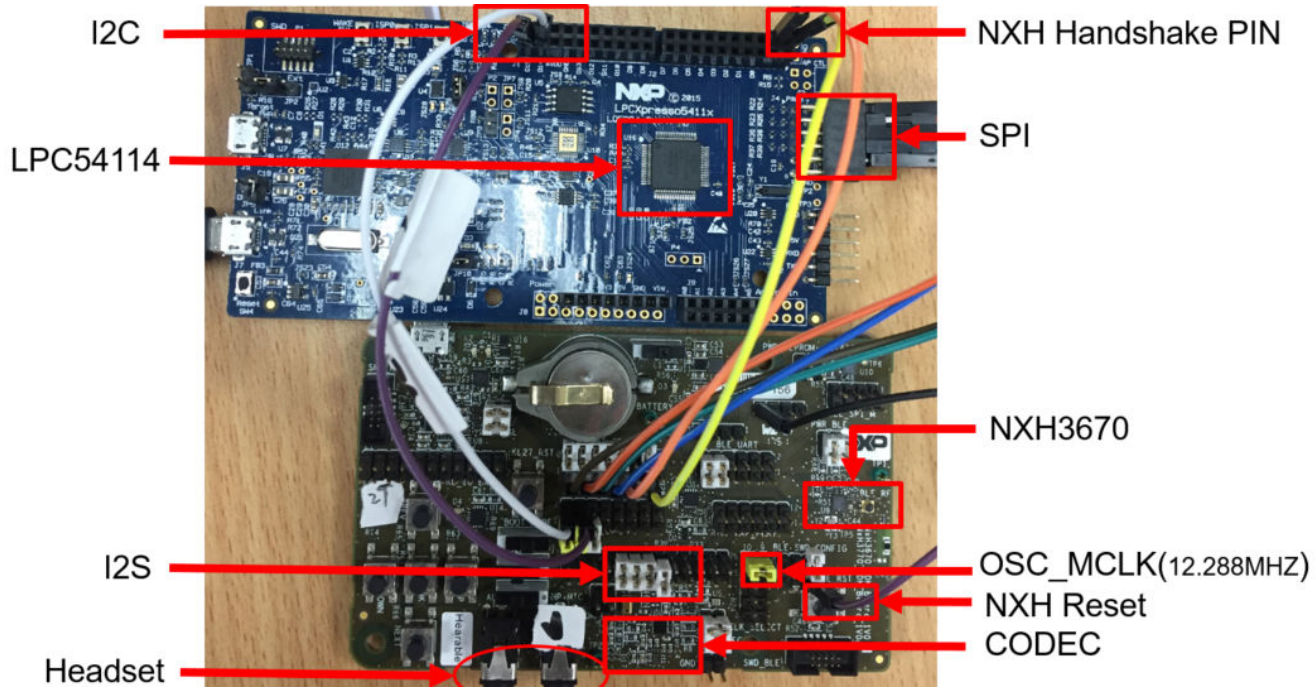

Figure 5. Headset hardware

NOTE

- In Headset design, the I^2C interface is used to to configure CODEC.
- NxH3670 communicates with CODEC using I^2S (as seen in Figure 5, the attached jumpers of 9-10, 11-12, 13-14 indicate that they can transfer data directly without the extra operation of LPC54114).

NOTE

- Jumper 9-10 (J10 CLK_SELECT, the yellow jumper `OSC_MCLK (12.288MHZ)` in Figure 5) is obligatory, as it is used to obtain 12.288 MHz frequency and then provide 24.576 Mhz frequency to I^2S's MCLK. PLL is recommended to be used to obtain higher frequency, if users want to reduce hardware cost and do not want use extra OSC.
- In addition, the Audio class implements Audio Speaker Class supporting stereo 48 kHz/16 bit.
  - The bit clock of the I^2S interface is derived from the master clock. The clock division is 16, so BCLK = MCLK/16 = 24.576 Mhz/16 = 1.536 MHz.
  - The Word Select (WS)/Left-Right Clock (LRCK) is 48 KHz.

3 Software introduction

3.1 Audio path

The BLE audio system consists of two channels:

1. The forward-channel transmits the audio from the PC to the headset.
2. The backward-channel transmits the microphone signal from the headset to the PC.

3.1.1 The forward audio channel

- The forward-channel is a stereo-channel.
- The RF transports the forward-channel as 16-bit samples @ 48 kHz sample-rate.
- The I^2S and USB signals use 48 kHz sample-rate.

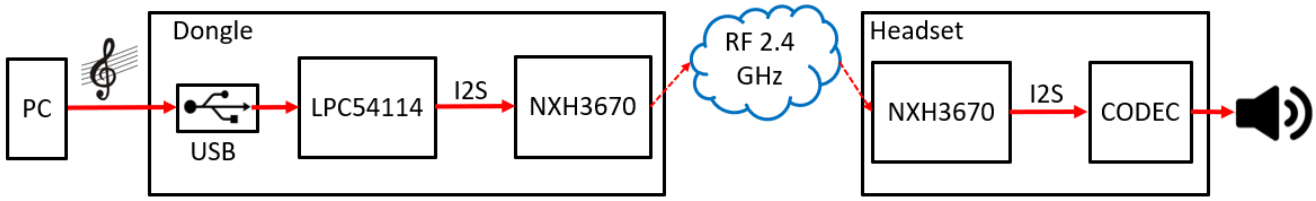

Figure 6. Data path of LPC54114 BLE forward audio system

3.1.2 The backward audio channel

- The backward-channel is a mono-channel, only using the left channel.
- The RF transports the backward channel as 16-bit samples @ 16 kHz sample-rate.
- The I^2S and USB signals use 48 kHz sample-rate.

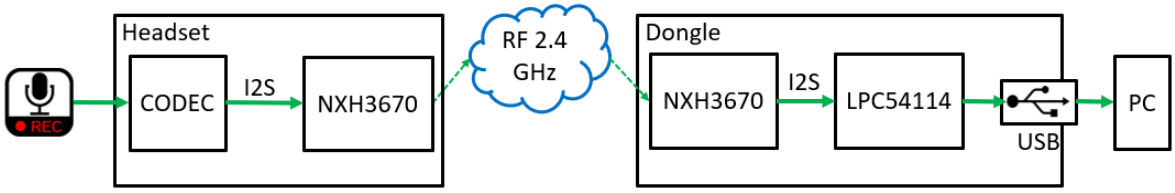

Figure 7. Data path of LPC54114 BLE backwardaudio system

3.2 Application framework

The application framework defines the software architecture of the reference application. The focus is modularity, code reuse and software maintainability.

The entire application architecture is as shown in Figure 8.

1. The top layer and the application layer are strictly application specific.
2. The layer below is called the services layer.
3. The driver layer is under the service layer and it controls the hardware.
4. The Board Support Package (BSP) contains the board specific software, such as, hardware initialization, GPIO-pin configuration, clock settings.

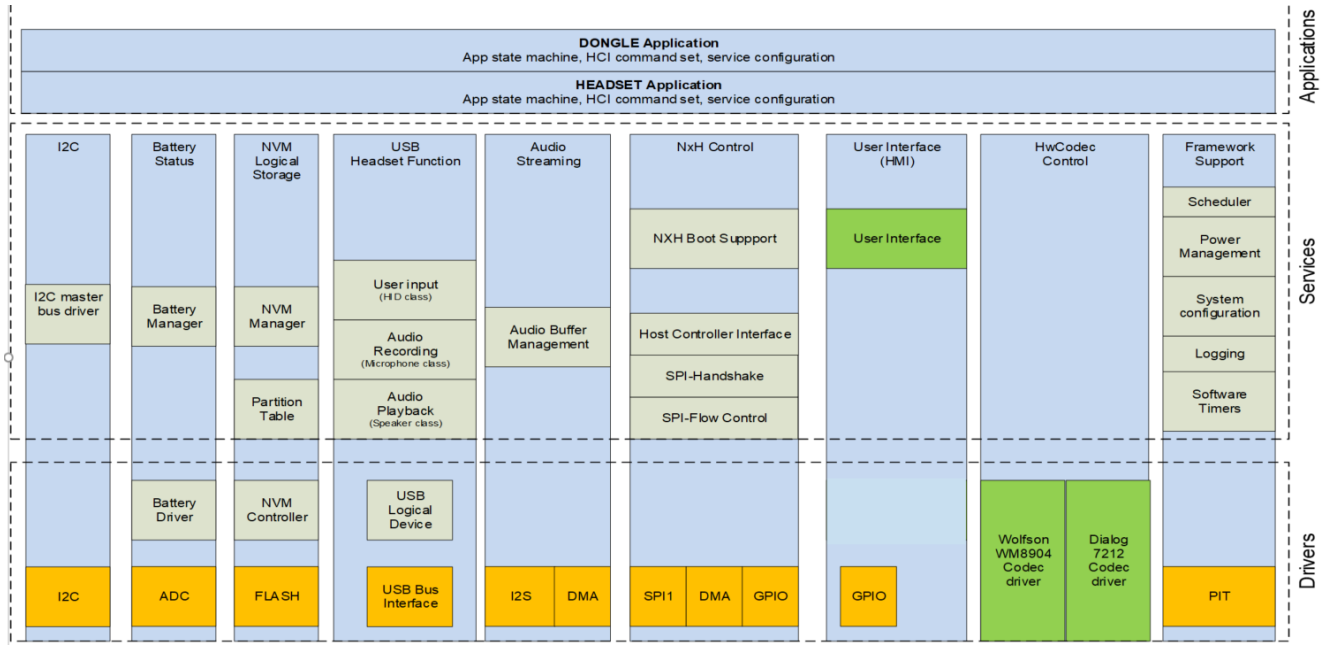

Figure 8. Application framework architecture

Users can design their own application or service as required. The below is a brief introduction of state machine that will be used to control the other services and the application. Users can decide on the service in a state as they wish.

- For example, if the state of an uninitialized USB-service is uninitialized, then the state will change from uninitialized to stopped by calling API initialize (and `cfg`) and executing successfully.

The mandatory API and corresponding state machine is as shown in Figure 9.

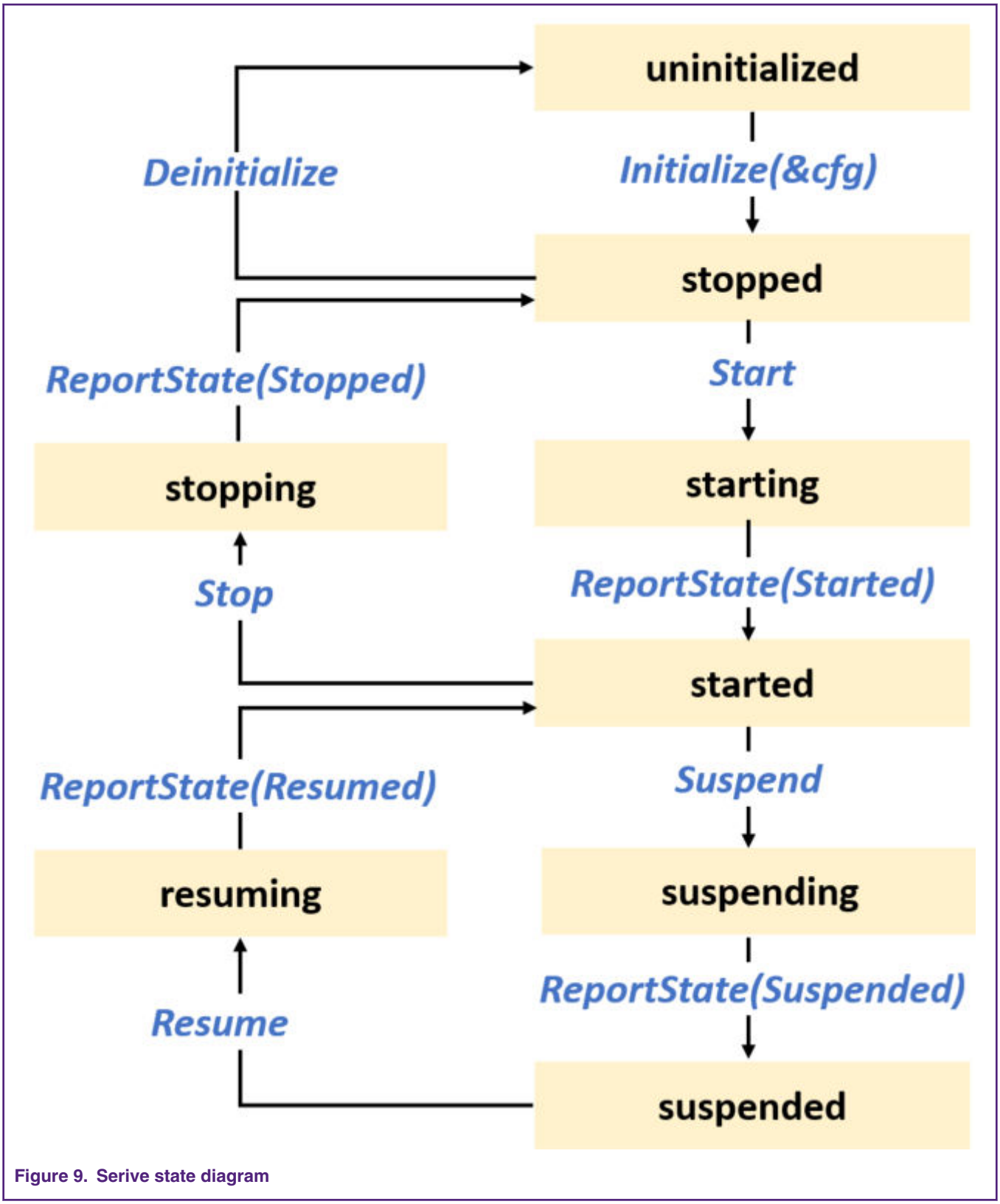

Figure 9. Serive state diagram

For example, the state can jump from **starting** to **started** by calling `ReportState(Started)`.

```
FRAMEWORK_ServiceReportState(&g_XXXServiceApi, kSTATE_Started);
```

3.3 Firmware development

3.3.1 Setting up the environment

This section gives an overview on the different steps to set up the MDK environment and the the materials required are listed in Table 2.

Table 2. Materials in firmware development

List	Description
PC	Host device connected to the development board
Debugger	• Default **CMSIS-DAP** firmware in Debugger onboard. • Replace the default **CMSIS-DAP** firmware with **J-LINK** firmware if users want to use `JLink.exe`to download the `Bin` file without IDE.
IDE	MDK (V5.26.2.0)
Demos	The `LPC54114+NxH3670.zip` file for Gaming use case includesing: • Bin files that can be download through `JLink.exe`. • Debug version of demo that can be used to re-developed.

3.3.2 Software based on MDK

In this document, we ported five demos, including:

- Dongle
- LPC_Headset
- LPC_OTA_Dongle
- LPC_OTA_Headset
- LPC_SSB

Two demos are listed as references to porting the demos to other boards.

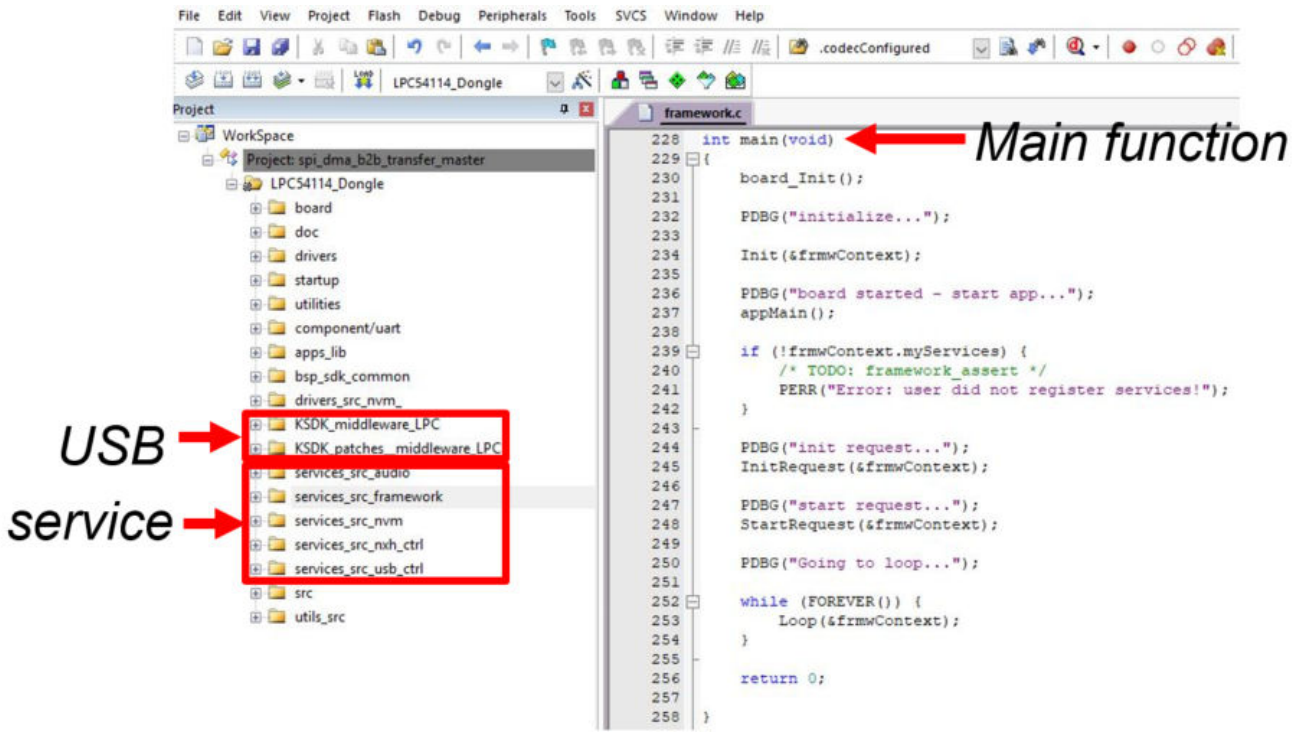

Figure 10. LPC Dongle project in MDK

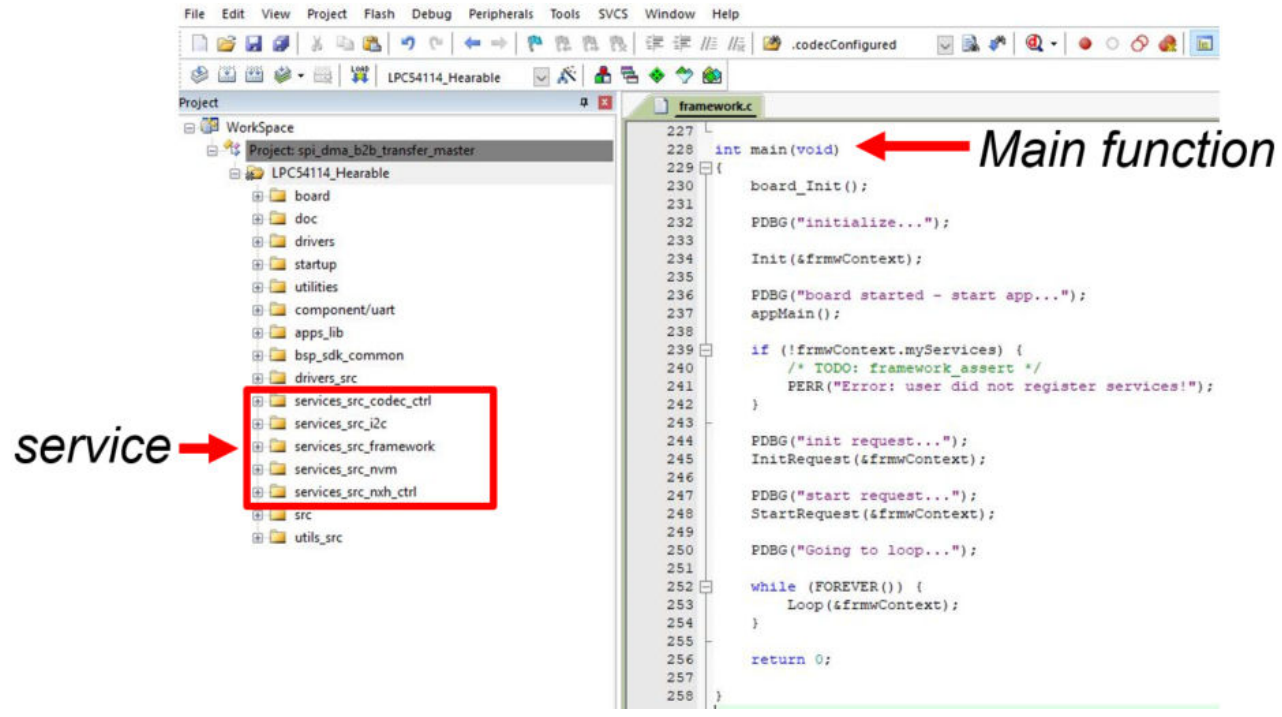

Figure 11. Headset project in MDK

If using the `JLink.exe` to download the `Bin` file, make sure the `J-LINK` firmware is in the Debugger onboard (MDK can be used for checking).

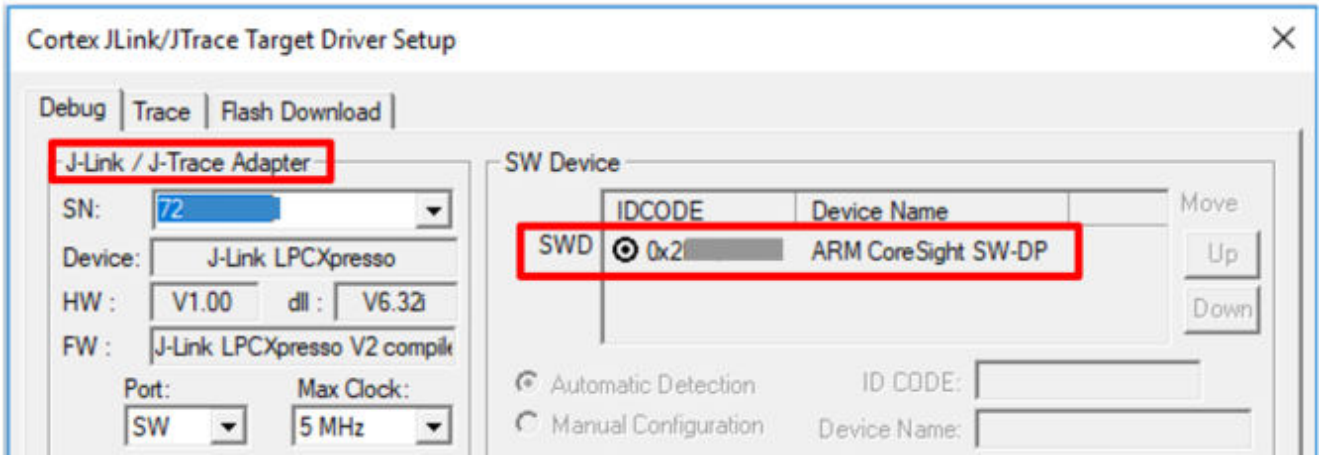

Figure 12. JLINK firmware check

To download `Bin` files from PC to LPC54114, users can use file, such as, `J-LINK.exe`, `JLinkARM.dll`, `XXXX.bat`, `XXXX.txt`, and `Bin`. Figure 13 shows an example.

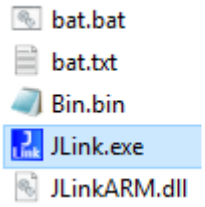

Figure 13. Test `zip` files list

- `bat.bat`: It is responsible for calling `JLink.exe` and the content is `call JLink.exe -CommanderScript bat.txt`.
- `bat.txt`: It defines where to download the specified `Bin` files.
- `Bin.bin`: Users can provide the related `Bin` file as required..

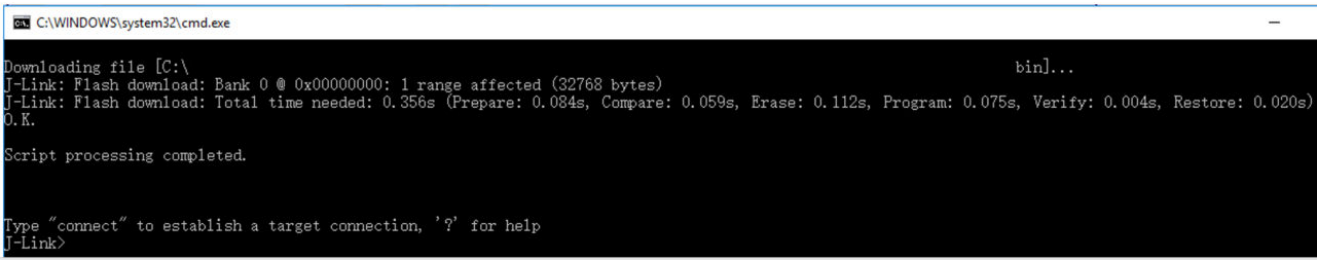

Figure 14. Downloading process through JLINK

4 Verification

4.1 Logic analyzer results

To make sure that **LPC54114 BLE Audio System** works well, refer to logic analyzer results shown in Figure 15, Figure 16, and Figure 17.

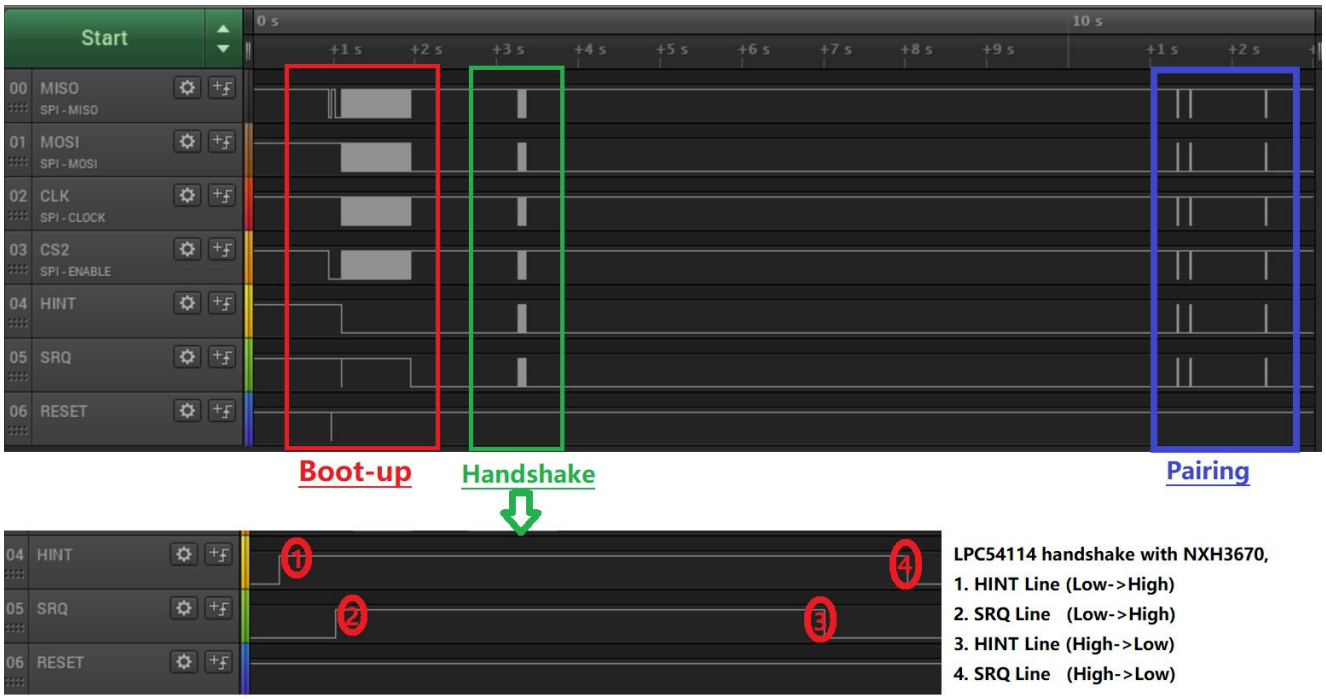

Figure 15. Logic analyzer result of Dongle

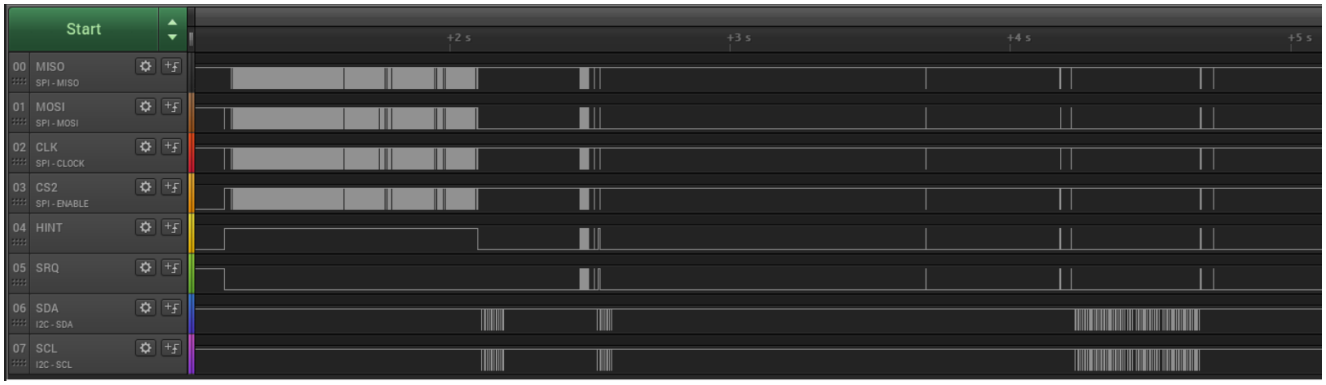

Figure 16. Logic analyzer result of Headset

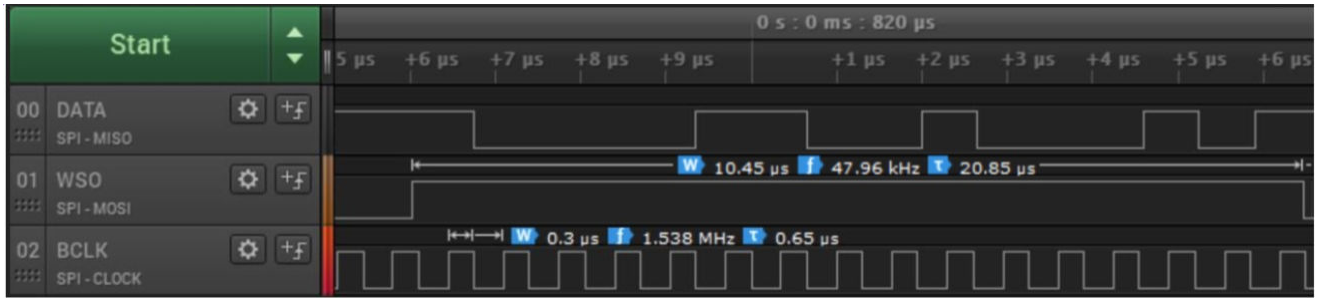

Figure 17. Logic analyzer result of I²S

4.2 Getting started with gaming uer-case

In this document, we provide two cases for user, **Play Audio** and **OTA**.

4.2.1 Play audio

Users can follow the steps below to verify audio play function of **LPC54114 BLE Audio System**.

1. Connect the hardware by following Hardware introduction.
2. Make sure that related demos have been correctly downloaded, no matter with IDE or `JLINK.exe`.
   - Use the **PWR ON/OFF** button to power ON/OFF NxH3670. Then, you can boot, start and communicate with it.
3. Wait until Dongle is paired and connected with Headset successfully.

   The two NxH3670 are paired firstly and then connected.

   In KL27 Dongle, the red LED blinks while pairing, ON when paired, and OFF when connected. In LPC54114 Dongle, the blue LED ON while pairing and OFF when paired, and the red LED OFF when connected.
4. Select **Headphones** as playback device and play music.

   Connect J5 with PC using USB cable instead of J7.

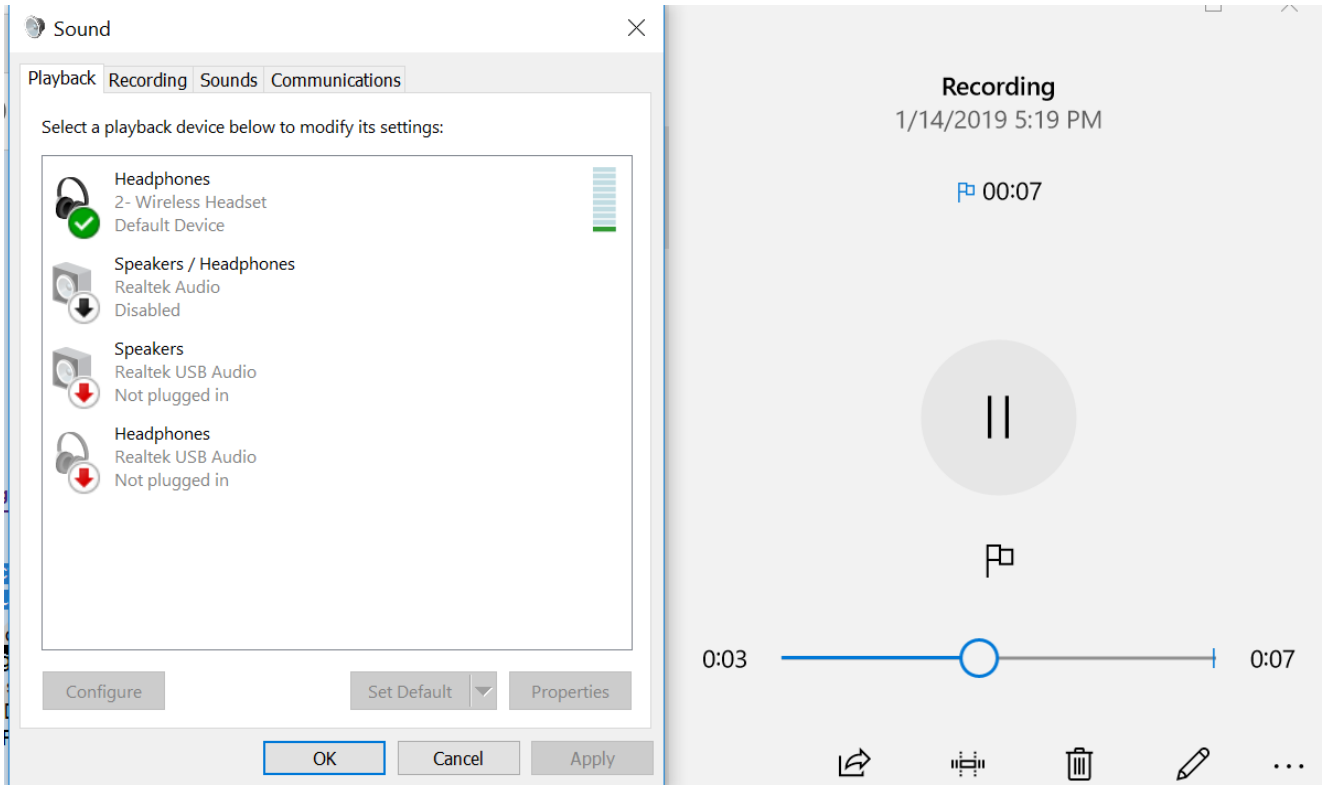

Figure 18. Windows sound control panel

4.2.2 OTA

User can upload latest firmware that brings new features through OTA.

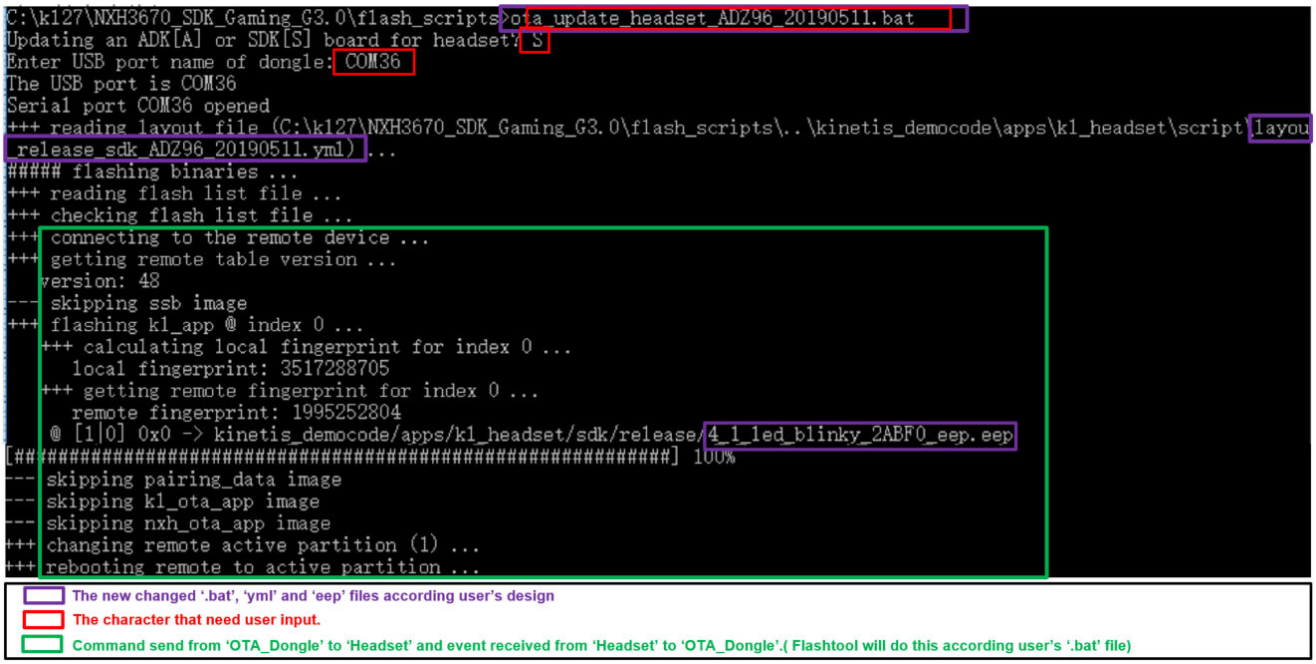

Figure 19. CMD of OTA process

To save time for users, the **flashtool** does not update existing firmware. For more information, refer to [LPC OTA].

5 Conclusion

This document provides basic introduction of **LPC54114 BLE Audio System**. Users can use Dongle and Headset to verify related functions.

How To Reach Us

Home Page:

nxp.com

Web Support:

nxp.com/support

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: AN12567

arm