AN13081

Wireless Gamepad Solution based on LPC5528 and NxH3670

Rev. 1 — 21 May, 2021

Application Note

Contents

1 Introduction

This application note introduces a wireless gamepad solution based on LPC5528 and NxH3670. This solution implements a wireless gamepad and supports wireless audio transmission. LPC5528 is a Cortex®-M33 based microcontroller for embedded applications with the main frequency of up to 150 MHz, the flash size of up to 512 KB and the RAM size of up to 256 KB. It has two USB device controllers which are key peripherals for the gamepad solution: Full-speed USB and High-speed USB device controllers. All of these features make LPC5528 very suitable as a gamepad or headset host controller. NxH3670 is a ultra-low-power 2.4 G wireless transceiver. It integrates a Cortex-M0 processor and has passed the Bluetooth Low Energy 4.2 certification. It has low latency and low power consumption characteristics and is used in wireless gaming headset products. This solution is developed based on NxH3670 SDK v5.2 and LPC5528 SDK v2.7. The development tool is MDK v5.28. Figure 1 shows the system block diagram.

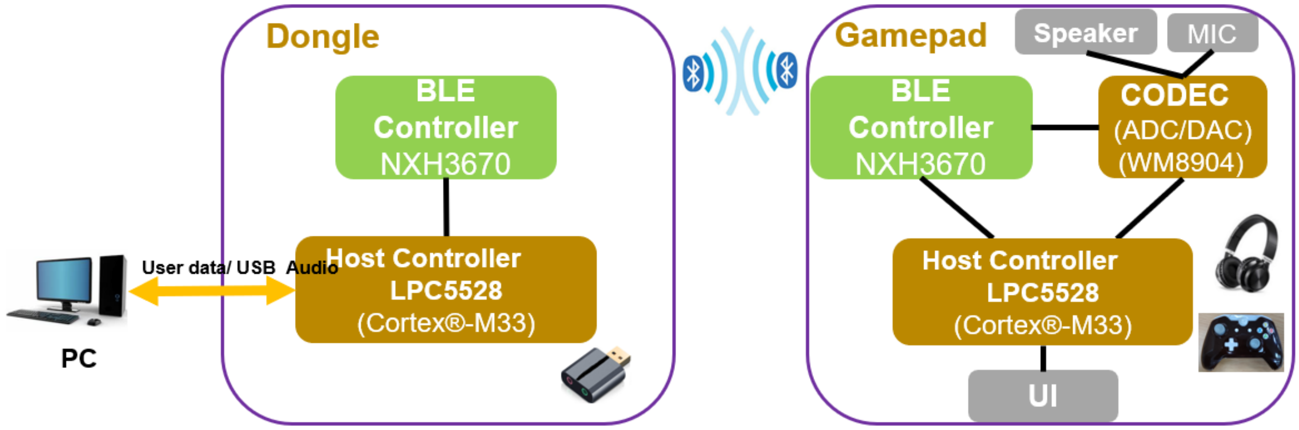

Figure 1. Wireless gamepad solution based on LPC5528 and NxH3670

The main functions implemented are as follows:

- Wireless mode

  USB cables are not required to connect PC (game console) and gamepad.

- Wired mode

  When the battery of gamepad is exhausted or there is no USB dongle module, the gamepad can be connected to the PC (game console) through a USB cable to make the gamepad work in wired mode and charge the battery.

- Audio playback and recording function

This solution also supports the function of wireless transmission and playback of audio stream. When the gamepad is working in wireless mode, the audio data on the PC is transmitted to dongle board through the USB interface, and then the audio is transmitted to the gamepad through the NxH3670 for playback. The gamepad can also transmit microphone data to the PC. When the gamepad works in wired mode, the audio data is directly transmitted to the gamepad through the USB interface. Similarly, the audio data input to the codec through the microphone will also be directly transmitted to the PC through the USB interface.

- USB VCOM and OTA upgrade function

  During the OTA upgrade process, the PC sends the new firmware to the dongle board via USB VCOM. Then, the dongle forwards the firmware to the gamepad. This new firmware will be written to the specified flash address.

2 Audio path

This chapter introduces the transmission channel of the audio stream in this solution. The audio channel includes two channels:

- **Forward channel**: Transmit audio data from the PC to the gamepad
- **Backward channel**: Transmits microphone data from the gamepad to the PC

2.1 Wireless mode

When the gamepad works in wireless mode, the audio path is as shown in Figure 2.

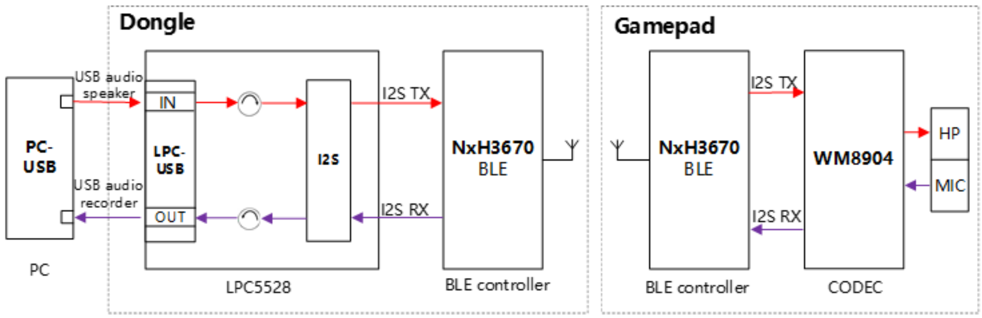

Figure 2. Audio path in wireless mode

- In the forward channel, the audio data is transmitted from the PC to the dongle through the USB audio speaker channel and then to the gamepad via NxH3670.
- In the backward channel, audio data is transmitted from the gamepad to the dongle via NxH3670 and then to the PC through the USB audio recorder channel.

NOTE

In fact, in wireless mode, once gamepad and dongle have established a BLE connection, the LPC5528 on the gamepad is not required to participate in the wireless audio transmission process.

Table 1 describes the audio features supported in this solution.

Table 1. Audio channel characteristics

	Forward channel	Backward channel
Signal	Stereo	Mono
Source	USB host, using USB Audio Class (Dongle)	Analog (external codec) microphone

Table continues on the next page...

Table 1. Audio channel characteristics (continued)

	Forward channel	Backward channel
Sample rate Over-The-Air	48 kHz	16 kHz
Encoding Over-The-Air	SBC	G722
Streaming Over-The-Air	Proprietary ICO (2 Mbps)	Proprietary ICO (2 Mbps)
NxH Audio Interface	I2S slave @ 48 kHz	I2S slave @ 48 Hz
Output	Headphone (Headset)	USB host, using USB Audio Class (Dongle)

2.2 Wired mode

When the gamepad works in wired mode, the audio path is as shown in Figure 3.

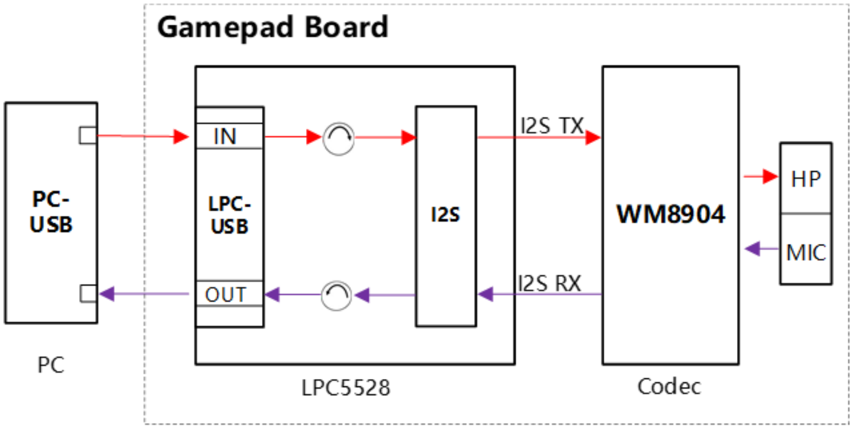

Figure 3. Audio path in wired mode

1. When the gamepad works in wired mode, the audio data on the PC is directly transmitted to the host controller through USB interface.
2. Then, the host controller transmits the data to the codec through the I^{2}S interface.
3. At the same time, the data input to the codec through the microphone is also transmitted to the host controller.
4. The host controller transmits the audio data to PC through USB interface.

In this mode, no USB dongle is used.

3 Control path

In addition to the audio channel, a control channel is also required to transmit user information, such as the state of the buttons and volume control signal. The host controller transmits the volume control signal to the PC through the USB HID class and transmits the gamepad control signal through the USB vendor specified class.

3.1 Wireless mode

3.1.1 Volume control

There are two ways to update the volume:

- Triggered by PC

  The PC can send an update over USB interface after user has changed its volume settings on PC.

- Triggered by Gamepad

  Pressing related buttons on the gamepad board.

3.1.1.1 PC-initiated volume update

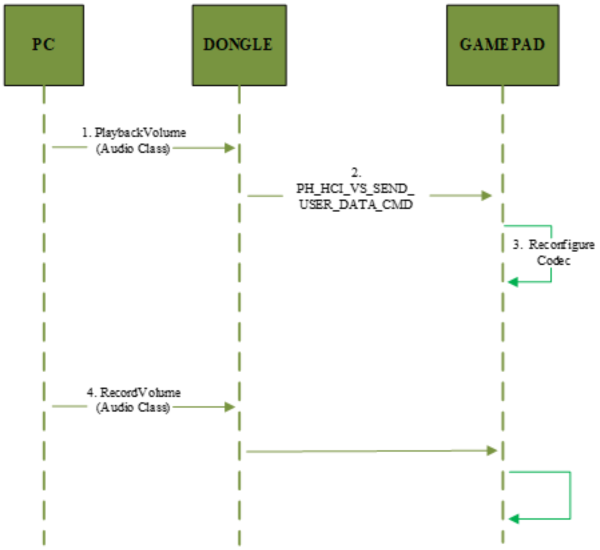
PC
DONGLE
GAME PAD
1. PlaybackVolume (Audio Class)
2. PH_HCI_VS_SEND_ USER_DATA_CMD
3. Reconfigure Codec
4. RecordVolume (Audio Class)

Figure 4. Volume update process initiated by PC

1. The user changes the volume setting on the PC. A USB volume update command is sent to the dongle.
2. The application on the host controller of the dongle reads out the updated volume value from the USB interface. Next, the application forwards this volume update over-the-air to the gamepad. The application uses a generic ‘user data’ HAPI command. For the details of HAPI commands, see *NxH3670 HAPI gaming* (document UM11148).
3. The application of the gamepad receives this command and configures the codec chip with the updated volume setting.
4. The exact same flow is followed for the configuration of the recording volume. To distinguish a playback-volume from a record-volume command, the application uses a parameter inside the HAPI user command.

3.1.1.2 Gamepad (button)-initiated volume update

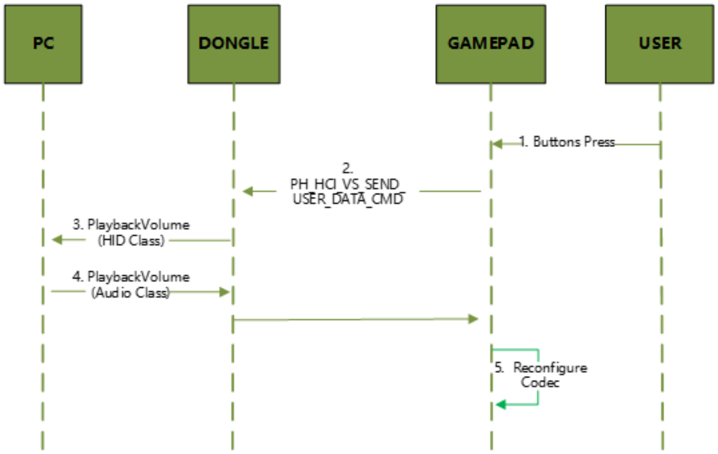
PC
DONGLE
GAMEPAD
USER
1. Buttons Press
2. PH_HCI_VS_SEND_USER_DATA_CMD
3. PlaybackVolume (HID Class)
4. PlaybackVolume (Audio Class)
5. Reconfigure Codec

Figure 5. Volume update process initiated by the gamepad

1. The user presses the volume up/down buttons.
2. The gamepad application sends a volume-update request to the dongle.
3. The dongle application receives the update request and sends a PlaybackVolume USB-command, part of the HID class.
4. The PC responds with an updated volume settings command, the application forwards the settings to the gamepad.
5. The gamepad reconfigures the codec with updated volume settings.

The gamepad does not apply the volume updates directly to the codec because the PC defines the step size of the volume update.

In this solution, use the following methods to control the volume:

- Volume UP: XE button + UP button
- Volume Down: XE button + DOWN button

For more details on how to use the gamepad, see *Getting Started with LPC5528 Wireless Gamepad Solution* (document AN13082).

3.1.2 Gamepad control

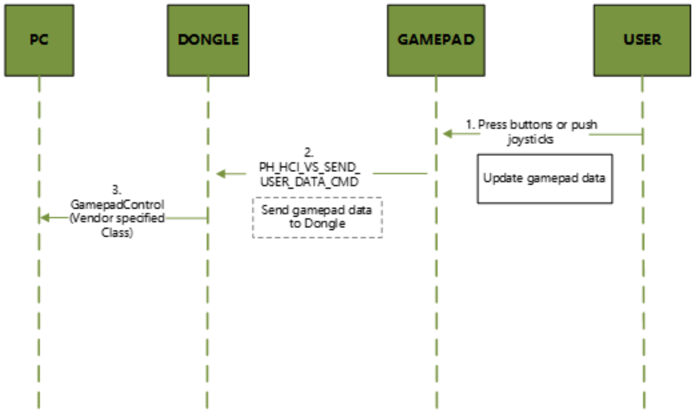
PC
DONGLE
GAMEPAD
USER
1. Press buttons or push joysticks
Update gamepad data
2. PH_HCI_VS_SEND_USER_DATA_CMD
Send gamepad data to Dongle
3. GamepadControl (Vendor specified Class)

Figure 6. Gamepad control

When the user press a button or push joysticks, the host controller on the gamepad will update the user data and send new user data to dongle. The command used to send the user data is also `PH_HCI_VS_SEND_USER_DATA_CMD`. After the dongle receives the user data, it will analyze the type of the user data. If it is the gamepad data, then the host controller on dongle board will update gamepad data and wait for the USB host (PC) to periodically obtain the gamepad data. If the USB host initiates a data acquisition request, dongle sends the latest gamepad data to the USB host through USB vendor specified class, the PC will perform the corresponding operation after receiving the new gamepad data.

3.2 Wired mode

The volume control and gamepad control in wired mode are similar to those in wireless mode, except that there is no process of forwarding control signals through Bluetooth LE.

4 Software

NXP provides the NxH3670 SDK board and the corresponding software development kit. The NxH3670 SDK can be downloaded from NxH3670 SDK Gaming Package. This solution is developed based on the NxH3670 SDK Rev5.2 with LPC5528 replacing KL27. The SDK version of LPC5528 is SDK v2.7. Figure 7 shows the software framework.

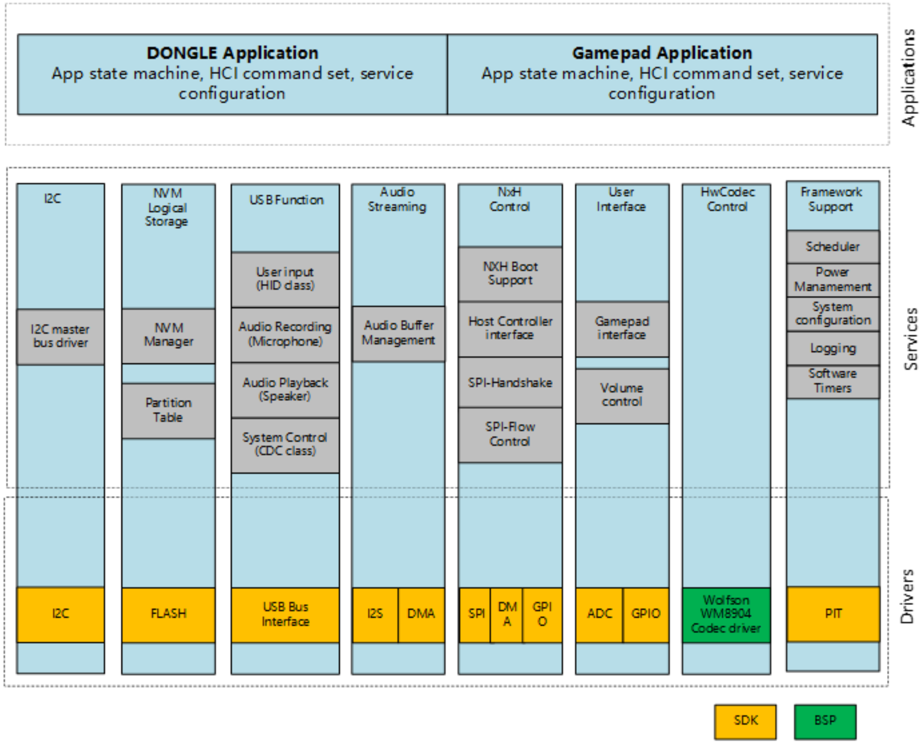

Figure 7. Application framework architecture

The main services are as follows:

- Non-Volatile Memory (NVM) service

  NVM service implements functions to read and write and erase flash.

- USB service

  The USB service implement a composite class device of Audio+CDC+HID+Vendor specified. The USB audio class is used to transfer audio data. The USB CDC class is used to configure system parameters through VCOM and to transfer new firmware during OTA upgrade process. The USB HID class is used to transmit volume control signal and the vendor specified class is used to transmit gamepad control signals.

- Audio service

  Audio service implements the configuration of I^{2}S transmitting and receiving channels.

- NxH3670 control

  The NxH3670 control service implements the configuration of the LPC5528 and NxH3670 communication interface, that is, the configuration of the SPI interface and also implements the handshake protocol between two processors.

- Codec service

  The codec service implements the configuration of the I2C interface and WM8904 codec.

- User Interface

  UI service implements the configuration of gamepad interface, including GPIO ports and ADC channels.

- Framework support

  The system software framework implements the scheduling mechanism between various tasks. In this solution, the deferred procedure calls(DPC) scheduling mechanism is used. For more details about the scheduling mechanism, please refer to the application note AN12280 in NxH3670 SDK Rev5.2.

4.1 Dongle framework architecture

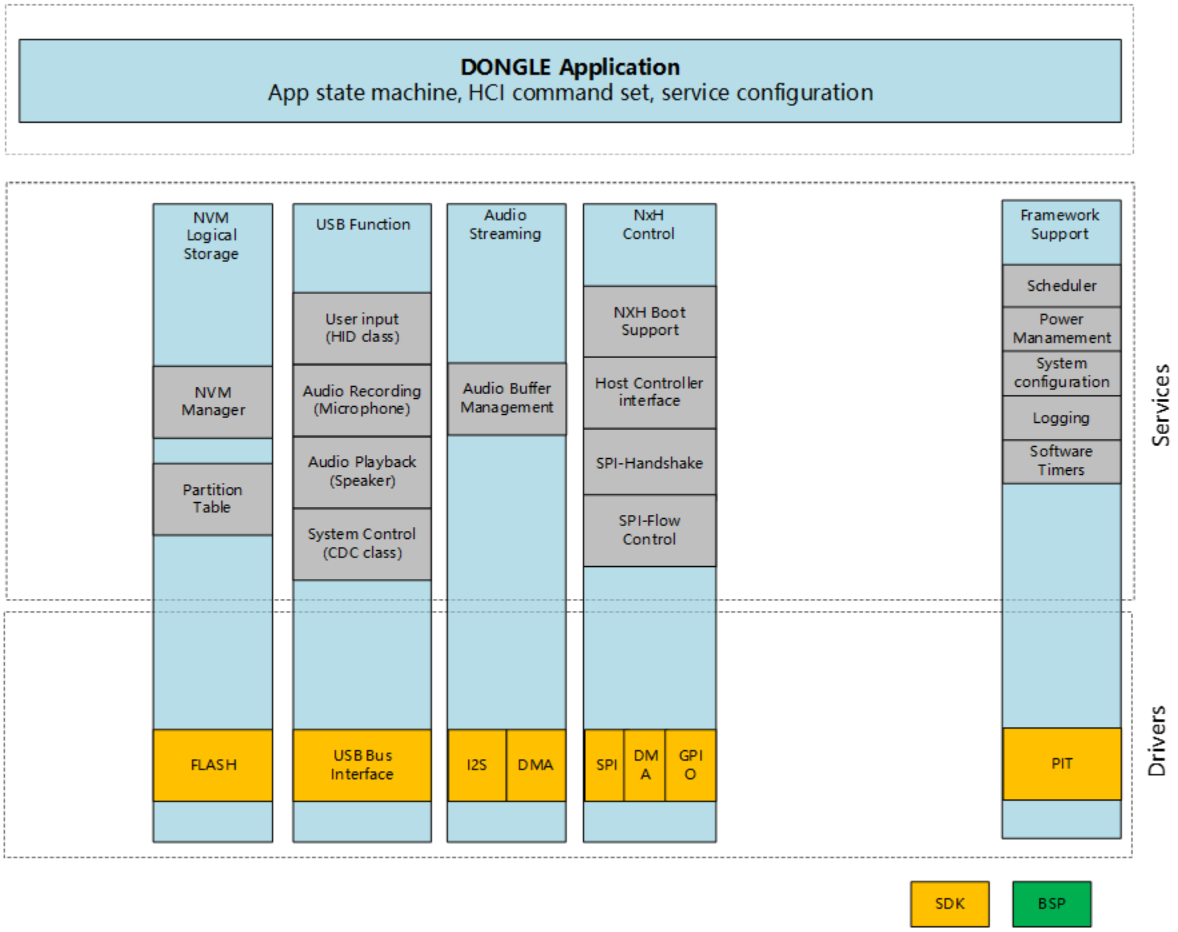

Figure 8. Software framework of dongle application

The dongle application implements the configuration of NxH3670 and the transmission of audio data.

4.2 Gamepad framework architecture

Figure 9 shows the software framework of the gamepad application.

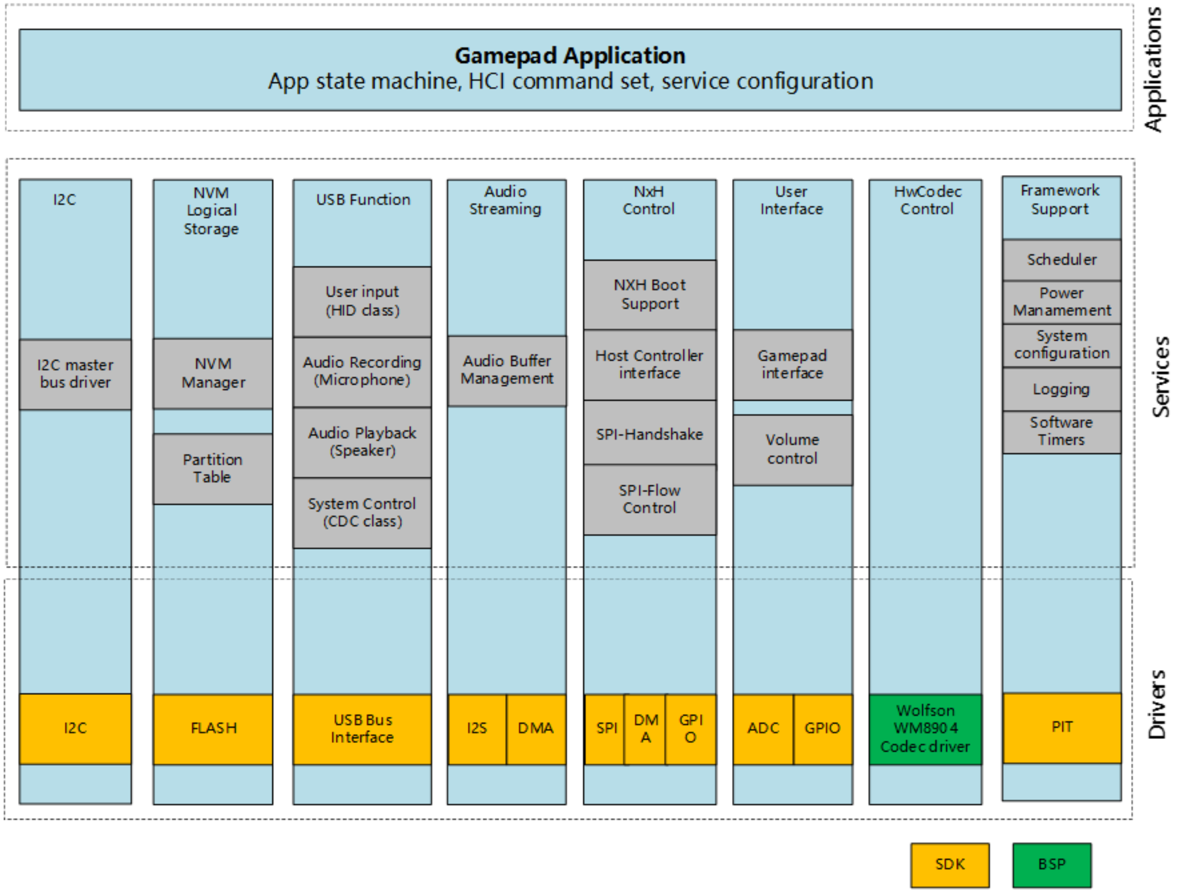

Figure 9. Software framework of the gamepad application

5 Hardware

This section introduces the hardware resources required in this solution. The hardware part consists of two boards: an LPC5528 dongle board and a gamepad main board.

5.1 Gamepad Main Board

Figure 10 shows the system block diagram of the gamepad main board.

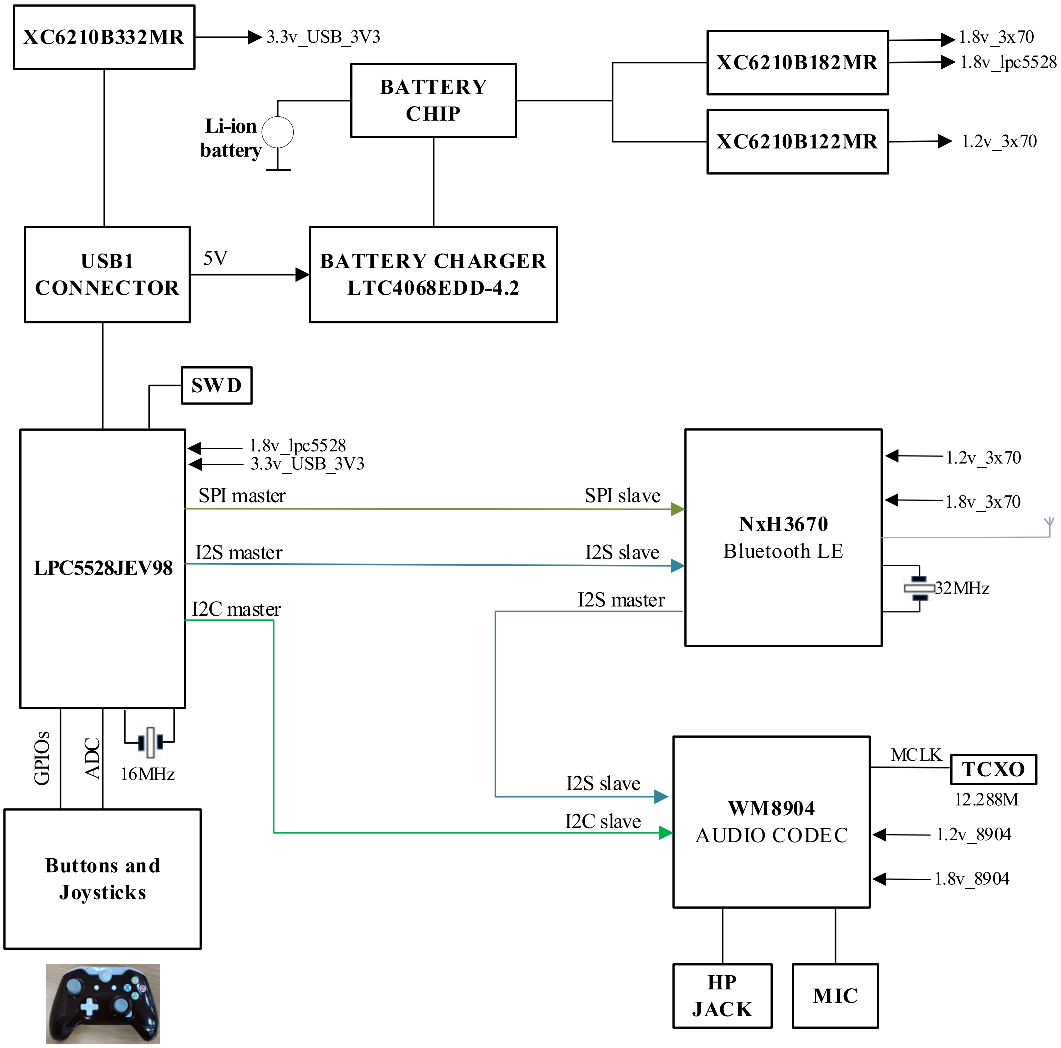

Figure 10. System block diagram of gamepad main board

The physical picture of the gamepad main board are as shown in Figure 11 and Figure 12.

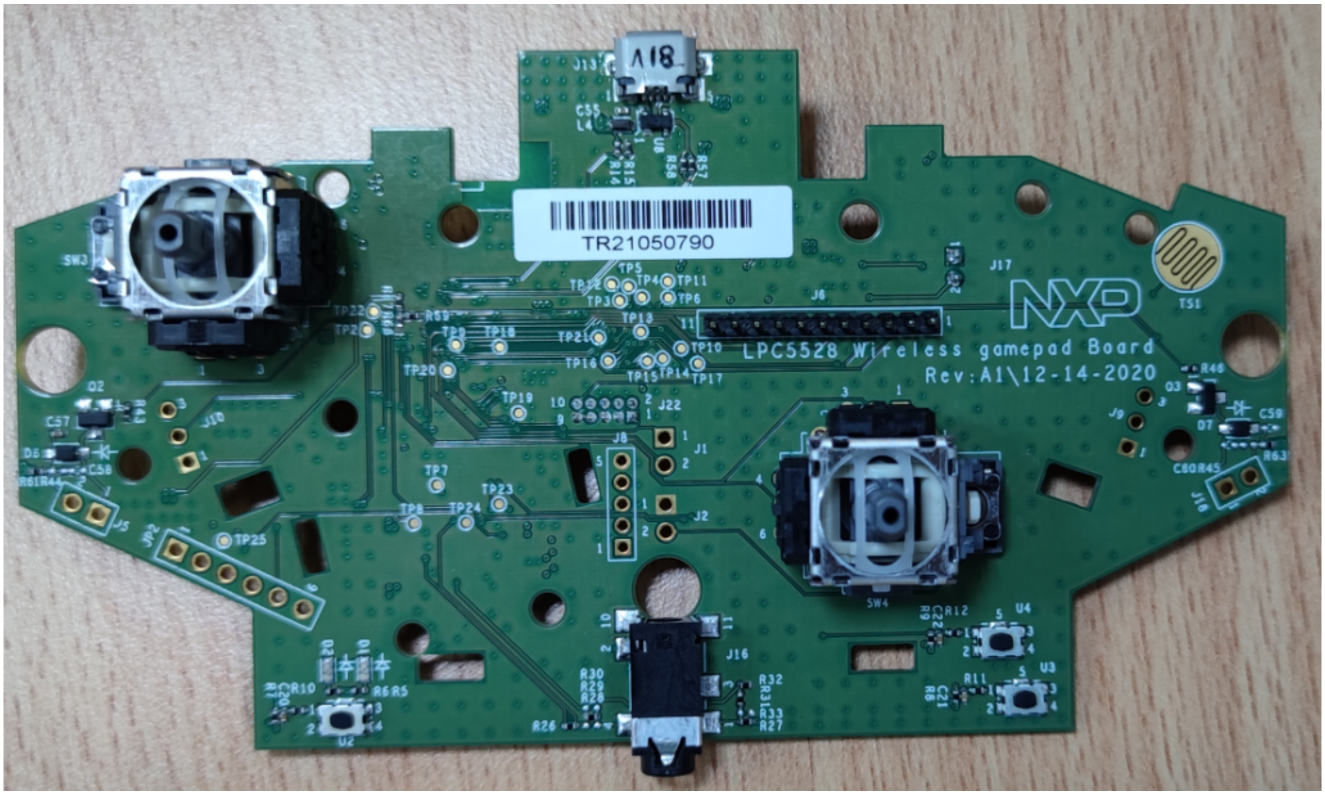

Figure 11. Front view of the gamepad main board

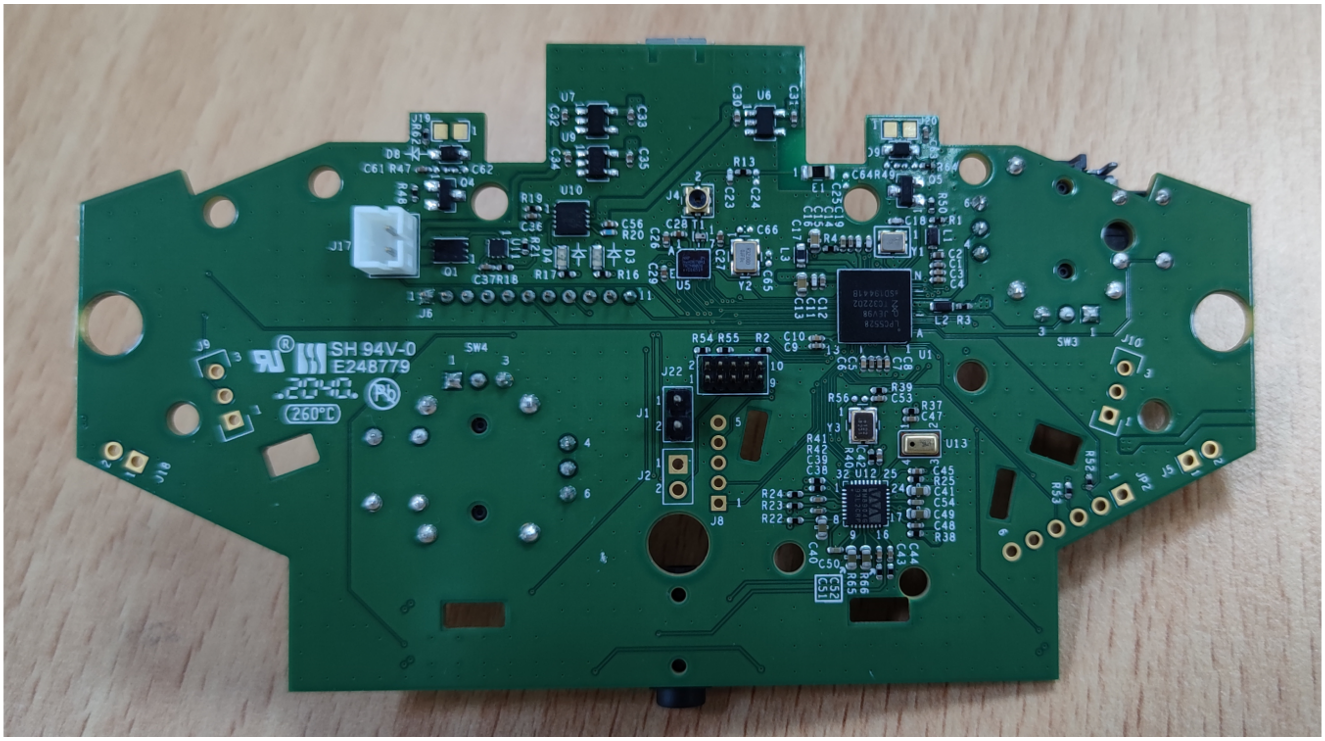

Figure 12. Back view of the gamepad main board

5.2 USB Dongle board

Figure 13 shows the system block diagram of dongle board.

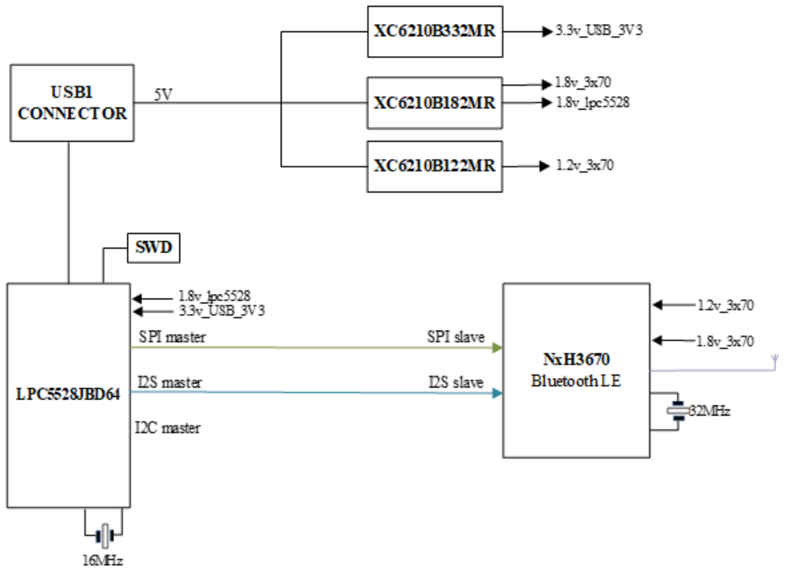

Figure 13. Dongle board block diagram

Figure 14 shows the physical picture of dongle board.

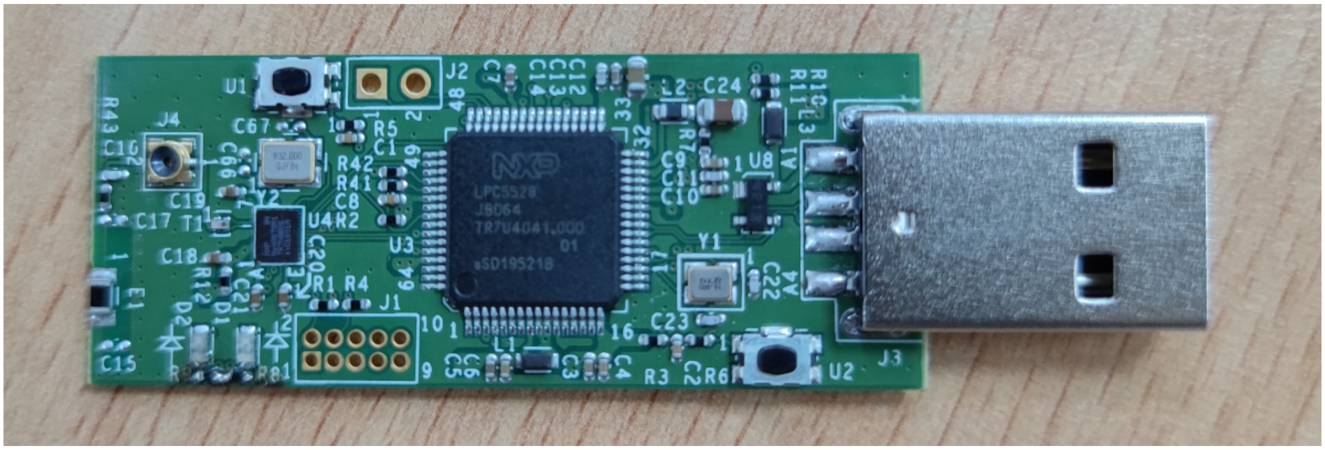

Figure 14. LPC5528 dongle board

Figure 15 shows the physical picture after assembly.

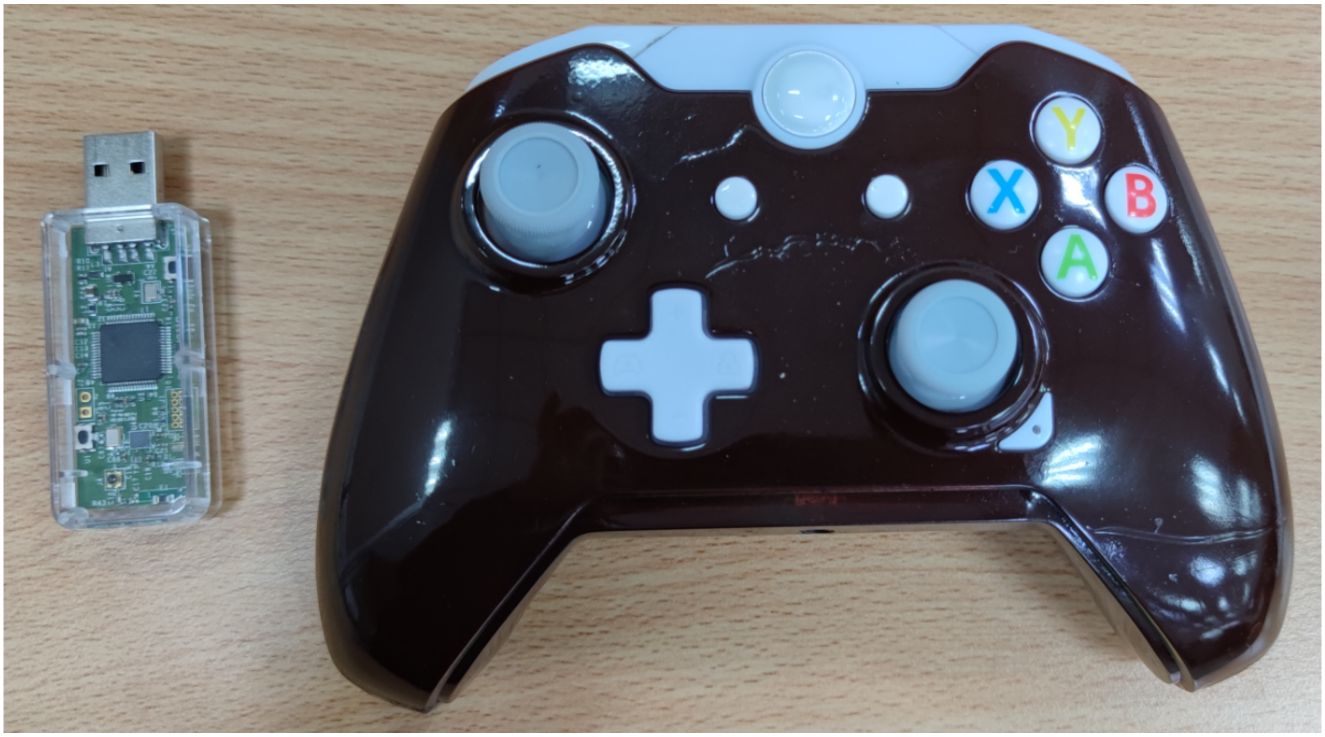

Figure 15. Gamepad board and dongle board after assembly

For more information about the hardware resources of the board and the gamepad usage, see *Getting Started with LPC5528 Wireless Gamepad Solution* (document AN13082).

6 OTA upgrade

This solution also supports OTA upgrade and the block diagram is as shown in Figure 16.

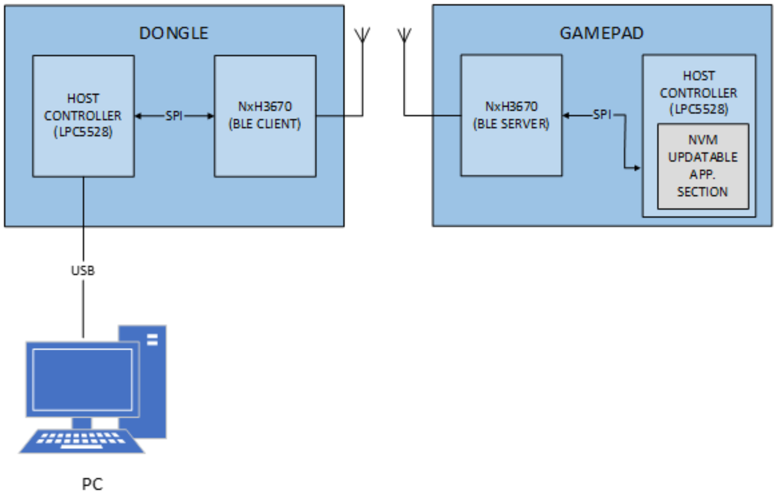
DONGLE
HOST CONTROLLER (LPC5528)
SPI
NxH3670 (BLE CLIENT)
GAMEPAD
NxH3670 (BLE SERVER)
SPI
HOST CONTROLLER (LPC5528)
NVM UPDATABLE APP. SECTION
USB
PC

Figure 16. Block diagram during OTA upgrade

The steps are as follows:

1. PC sends OTA upgrade command to dongle.

2. Dongle forwards this command to the gamepad and establish BLE connection.
3. The host controller on gamepad board check the partition table before enabling the OTA application.
4. Gamepad enable OTA application.
5. If the OTA application is successfully enabled, restart the gamepad and run the OTA application.
6. The gamepad re-establishes a BLE connection with dongle.
7. Dongle notifies the PC after the connection is established.
8. The PC requests the partition table version of gamepad and verify that it is compatible.
9. The PC starts to send the firmware to dongle.
10. Dongle forwards each packet to the gamepad.
11. Gamepad check the updated partition table before enabling new firmware.
12. Gamepad enable new firmware.
13. If an error occurs before enabling the new firmware, continue to run the OTA application and upgrade again.
14. If the new firmware is successfully enabled, the host controller restarts and runs the new firmware, OTA upgrade is completed.

For more details of OTA upgrade, see *LPC5528 Wireless Gamepad OTA Upgrade* (document AN13083).

7 Reference

1. *NxH3670 Reference Application Software Architecture* (document AN12280)
2. *Getting Started with LPC5528 Wireless Gamepad Solution* (document AN13082)
3. *LPC5528 Wireless Gamepad OTA Upgrade* (document AN13083)
4. Release Notes NxH3670 SDK Gaming
5. *NxH3670 HAPI gaming* (document UM11148)
6. *NxH3670 SDK board* (document UM11150)

8 Revision history

Rev.	Date	Substantive changes
0	16 December, 2020	Initial release
1	21 May, 2021	Updated Figure 1, Figure 10, Figure 11, and Figure 15

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.

Right to make changes - NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Security — Customer understands that all NXP products may be subject to unidentified or documented vulnerabilities. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer’s applications and products. Customer’s responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer’s applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately. Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP. NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

NXP, the NXP logo, NXP SECURE CONNECTIONS FOR A SMARTER WORLD, COOLFLUX,EMBRACE, GREENCHIP, HITAG, 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, CodeWarrior, ColdFire, ColdFire+, the Energy Efficient Solutions logo, Kinetis, Layerscape, MagniV, mobileGT, PEG, PowerQUICC, Processor Expert, QorIQ, QorIQ Qonverge, SafeAssure, the SafeAssure logo, StarCore, Symphony, VortiQa, Vybrid, Airfast, BeeKit, BeeStack, CoreNet, Flexis, MXC, Platform in a Package, QUICC Engine, Tower, TurboLink, EdgeScale, EdgeLock, eIQ, and Immersive3D 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.

For more information, please visit: http://www.nxp.com
For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 21 May, 2021
Document identifier: AN13081