UM11082

User Manual for LPC8N04 Development Board

Rev. 1.1 — 12 January 2018 User Manual

Document information

Info	Content
Keywords	LPC8N04 Development Board, OM40002, LPC8N04, NFC, NTAG
Abstract	LPC8N04 Development Board User Manual

Revision history

Rev	Date	Description
1.0	20171218	First draft
1.1	20180112	Corrected SWD/LED jumpers. Added information about SWD pins at start up.

1. Introduction

The LPC8N04 Development Board board is developed by NXP to enable evaluation of and prototyping with the LPC8N04 MCU. Figure 1 shows each side of LPC8N04 Development Board. The LPC8N04 device, buttons, jumpers and debug probe circuitry are on the top side (along with the board name). On the bottom side is a 5x7 array of LEDs, plus a surface mount speaker (LS1).

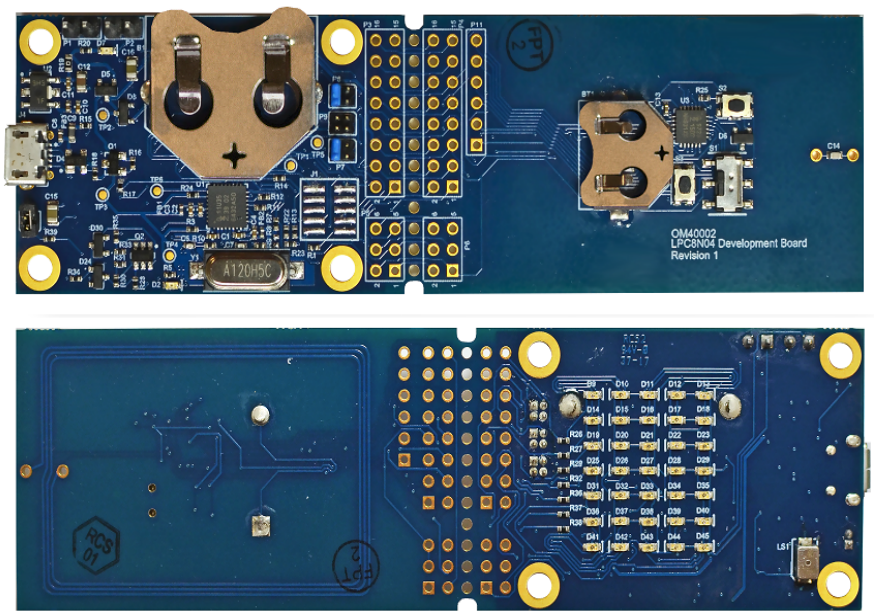

Fig 1. LPC8N04 Development Board

The LPC8N04 Development Board board includes the following features:

- Compatible with MCUXpresso IDE and other popular toolchains (incl. IAR and Keil)
- Detachable, two-section board featuring minimal LPC8N04 system and debug probe / demo circuitry (LEDs and speaker)
- LPC8N04 Arm Cortex-M0+ MCU running at up to 8MHz
- Integrated NFC antenna
- Dual coin cell battery holders for
- On-board CMSIS-DAP (debug probe) with VCOM port, based on LPC11U35 MCU
- Debug connector to allow debug of target LPC8N04 MCU using an external probe
- LPC8N04 User button
- LPC8N04 Reset button
- 5x7 LED array for user applications
- Surface mounted speaker with amplifier
- Expansion connections with pin-outs compatible with I^2C Grove and Pmod connectors for easy connection of sensors

- Pre-programmed with demo message scrolling / tune playing application (requires NFC-enabled phone to use; iOS 11 phones only support limited functionality)

The Board is divided into two parts - the Debug Probe (DP) and Main Processor (MP) Sections. The two sections may be snapped apart by scoring the row of holes between them using a knife, then applying firm pressure to this region of the board. Once separated, the MP board may be used as a standalone target, either powered via a button cell or via the P4 expansion connector. The boards can be reconnected by soldering 0.1" pitch right angled connectors(not supplied) in the P3/P4 and P5/P6 header locations.

2. Board Layout

Figure 2 below shows the layout of the LPC8N04 Development Board board, indicating location of jumpers, buttons and connectors/expansion options.

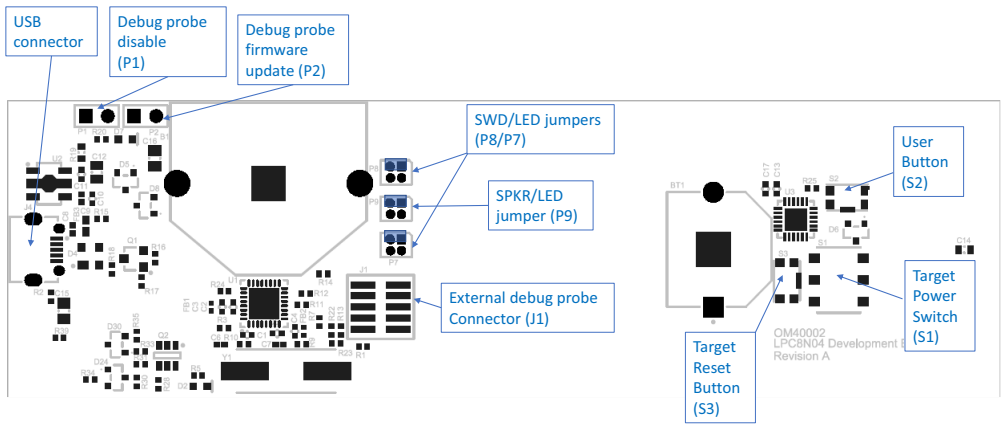

Fig 2. Board layout (top silkscreen)

Table 1 below shows the layout of the LPC8N04 Development Board board, indicating location of jumpers, buttons, connectors/expansion options and MCU devices.

Table 1. Jumpers and connectors

Circuit reference	Description	Reference section
P1	Inserting a jumper on P1 holds the LPC11U35 debug probe in reset. Install P1 when using an external debug probe.	[4.1]
P2	Inserting a jumper on P2 forces the LPC11U35 debug probe into Device Firmware Update (DFU) mode when the board is power cycled, allowing its on-board firmware to be updated.	[4.1]
P3, P4, P5, P6	These connectors provide access to the PIO lines of the LPC8N04, plus power and reset.	[5.1]
P7, P8	PIO0_11 and PIO0_10 from the LPC8N04 can be used as SWD pins or GPIOs. P7 (for PIO0_11) and P8 (for PIO0_10) are used to determine whether those signals are routed to the SWD header and LPC11U35 debug probe, or to the 3rd and 4th columns of the LED array.	[4], [7.2]
P9	P9 is use to selected whether PIO0_3 from the LPC8N04 is routed to the speaker amplifier input of row 6 of the LED array.	[7.2], [7.3]

Table 1. Jumpers and connectors

Circuit reference	Description	Reference section
P10	P10 is provided in order to connect a piezo-type speaker (after first removing the on-board SMD speaker). This jumper is bypassed by R2 (zero ohm resistor) by default.	[7.3]
J1	External debug probe connector. Install P1 when using this header.	[4.1]
S1	LPC8N04 power on/off switch. When the switch is open the power is disconnected from the LPC8N04. This switch is provided to prevent accidental draining of the batteries.	[6]
S2	User button. This resistor is connected to the PIO0_3 pin of the LPC8N04 and can be used in user applications. When S2 is pressed, PIO0_3 is pulled to ground via a 1k ohm resistor. Note that PIO0_3 also drives row 5 of the LED array.	[7.1]
S3	LPC8N04 reset button. This button resets the LPC8N04 (note that no other circuitry is reset by this button.	n/a

3. Getting Started

The LPC8N04 device flash is pre-programmed with an example application, which scrolls a message across the LED array on the bottom side of the board and plays a tune through the micro speaker. The jumpers are positioned for that demonstration by default. This section describes how to run the demonstration, and how to re-configure the board to start code development. If using Windows 7 or 8, before powering up the board, you will need to install device drivers for the VCOM port.

3.1 Installing device drivers (Windows 7 and 8 only)

The firmware for the on-board debug probe and Windows 7/8 driver for the virtual com (VCOM) port are available from http://nxp.com/demoboard/OM40002 under the Software and Tools tab (look under “Software” download types). The debug probe firmware is factory installed so should not normally need updating, but if using Window 7 or 8, you will need to install the VCOM driver by running the installer program before attempting to use the board. After downloading and unzipping the package, run the installer program provided to install the driver (see installation notes included in the package for further information.)

3.2 Running the out-of-box demo

The LPC8N04 on the Development Board is pre-programmed with a demonstration program, which designed illustrate the features of the device by working with an NFC-enabled smartphone running a free App from NXP. For best results, an smart phone running Android Nougat or later is recommended. The demonstration program (in conjunction with the App) shows two way communication between a smartphone and the LPC8N04, passing messages, temperature and tune selection information between the two devices. Some aspects of the demonstration application can be used without any power being supplied to the Development Board; these are mentioned in this section below.

The LPC8N04 supports SWD debug, requiring two I/O pins. Alternatively, these pins can be used as GPIOs. On the Board these two SWD/GPIO signals are connected to two of the LED array columns via jumpers P7 and P8. To use the full number of LED columns, the P8 and P7 jumpers need to be set as shown in Figure 3.

On GPIO from the LPC8N04 is shared between a row of the LED array and the speaker, using a jumper (P9) to select which function is used, as shown in Figure 3. If no jumper is fitted then neither the LED row nor the speaker will be enabled.

On the Target (right) side of the board, there is a power switch. This is to prevent accidental running down of the battery, when used. The switch should be in the up position to provide power to the LPC8N04.

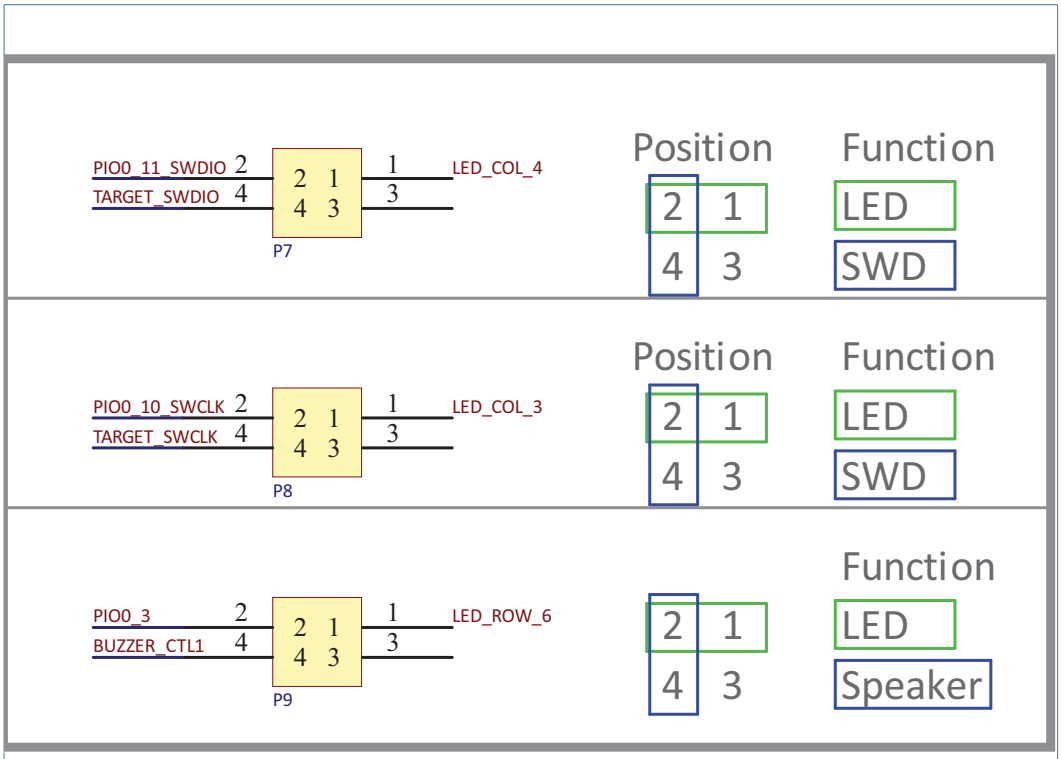

Fig 3. SWD/Speaker/LED selection jumpers

3.2.1 Starting the application

The LPC8N04 demo application can be downloaded for free from the Google Play Store. Search for “NXP LPC8N04 NFC Demo.”

Power up the LPC8N04 Board and press the reset (S3) button (note that the RESET label on Rev A boards is adjacent to S2, but S3 is the reset switch). Turn the board over so the LED array is on top and oriented to the left. The center column of LEDs will light in turn in a repeating pattern. Note that after 100 seconds of inactivity, the demonstration firmware will go into a low power state, and the center LED will blink approximately once every second. This is to prevent accidentally draining a battery (if fitted). If this occurs, press the reset button to re-activate the demo.

Ensure NFC is enabled on the smartphone, then activate the App. Tap on the “LED TEXT AND MUSIC” tab at the top left, then the Read NDEF button.

3.2.2 Enabling and reading the and scrolling message & temperature

Bring the phone close to the antenna section of the board, to the right of the antenna section of the Board. As the smartphone detects the Board it will either vibrate and/or play a tone (depending on its settings). Once this happens, a message will start scrolling across the LED array, and the message content will appear in the text box. This message is stored in the EEPROM, along with the current temperature of the LPC8N04 and will have been read over NFC. The default message is “The LPC Experience; Your World Reimagined Temp is 78.9F” (although the temperature value will vary). The read performance will also be read, and the currently set scroll speed will be shown.

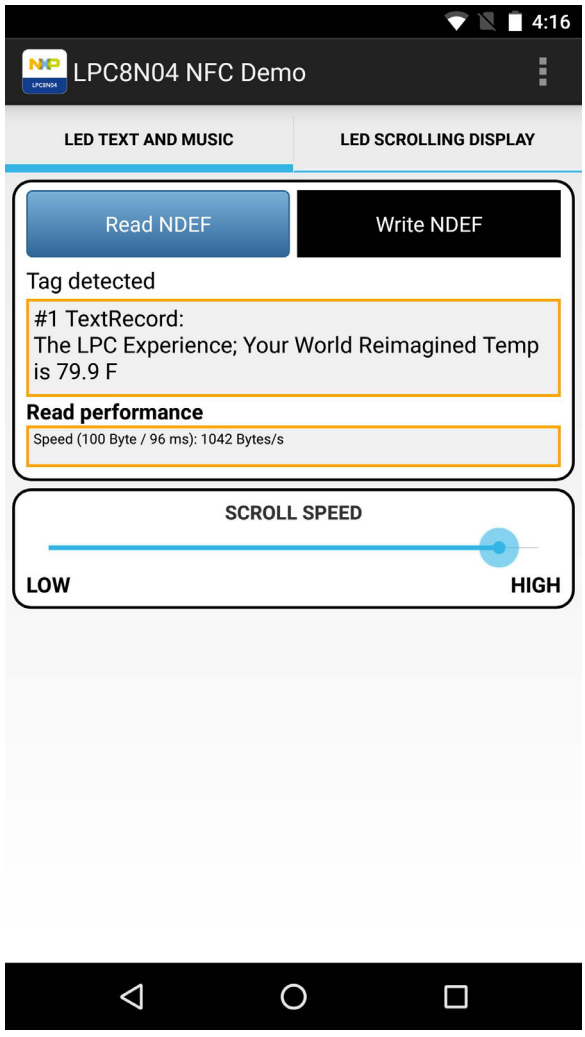

Fig 4. Read NDEF operation, after scanning the Board

3.2.3 Updating the scrolling message

Tap the “Write NDEF” button; it will turn blue. Type a message in the Text box, then tap “Write NDEF” to write the message to the LPC8N04. The message “TAG successfully written” should appear, along with the speed at which the write took place. The LED array will show your new text scrolling across the display. The speed of the message can be varied by moving the blue circle along the slider bar just below the text entry / write performance box; note that Write NDEF must be pressed again to update the scrolling speed.

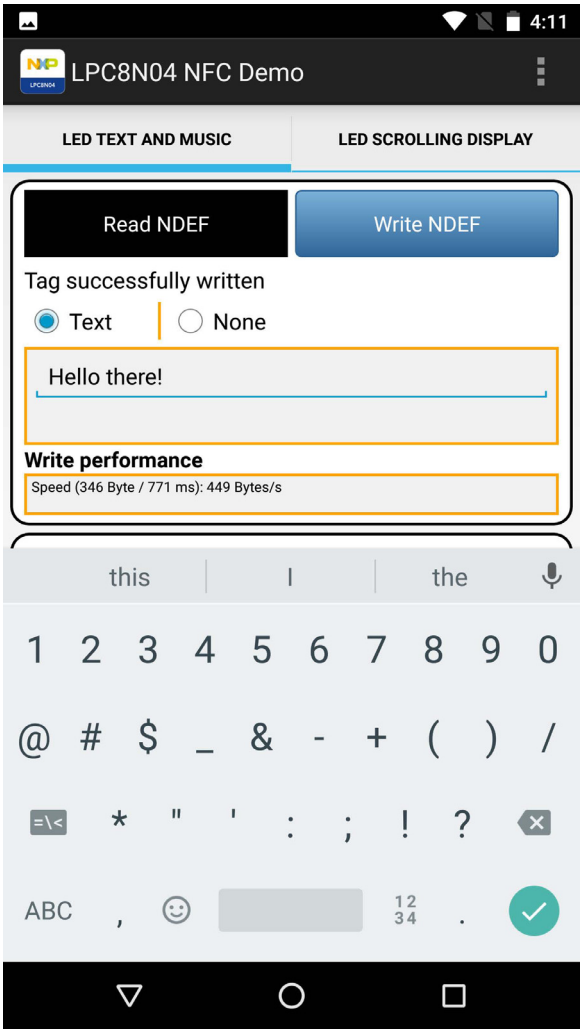

Fig 5. Updating the scrolling message

3.2.4 Playing a tune

To enable the speaker, jumper JP9 needs to be set such that the speaker is enabled instead of the top row of LEDs (see Figure 3). By default, the firmware will assume this GPIO is being used for the LED row, but will switch to driving the speaker once a push of button S2 is detected. Because of this, the speaker will making a buzzing sound until S2 is pressed.

There are 3 tunes available on the App. To change the tune, simply tap on the name of the tune to be played, then tap Write NDEF. Fur Elise is programmed into the EEPROM during manufacture of the Development Board.

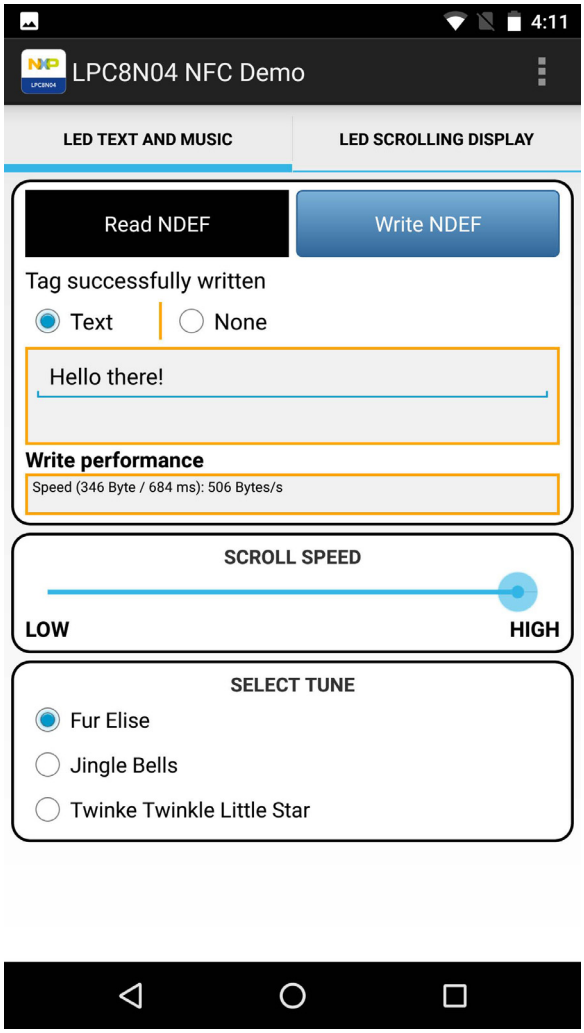

Fig 6. Playing tunes

3.2.5 Running demos using energy harvesting

If a smart phone with sufficient NFC reader power is used,a demo can be run using energy harvesting to power the board (i.e. with no battery installed and no USB power). Before tying to run this demo, ensure jumper JP9 is set such for LEDs (not speaker).

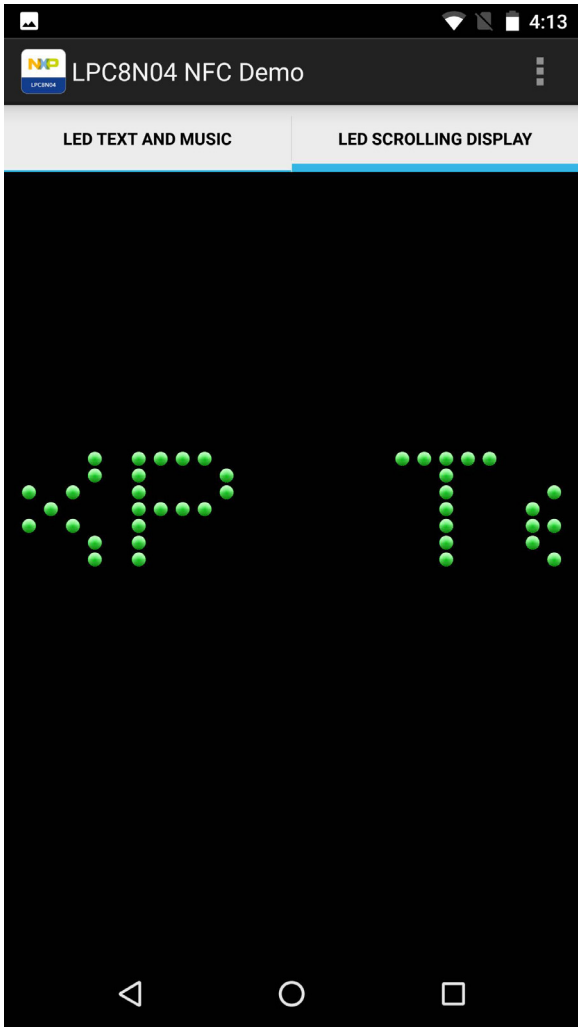

Fig 7. LED scrolling display

3.3 Setting up the board for Development

The on-board debug probe (LPC11U35) on the LPC8N04 Development Board comes pre-programmed with firmware supporting the CMSIS-DAP protocol. This protocol is supported by MCUXpresso IDE, and other popular IDEs such as Keil MDK and IAR EWARM. It is not normally necessary to update the debug probe firmware, but this can be done by following the steps in Section 3.4.

For debug to be possible, jumpers must be installed on P7 and P8 so that the SWD signals are connected from the target side of the board to the debug probe side. See Figure 3 for the jumper positions. Note that the factory fault jumper settings do not support debug and must be changed in order for development to begin.

Note that in the demo application (app_demo) from the Code Bundle (and pre-programmed into the board, as described earlier) P0_10 and P011 are set up for SWD functionality when the board starts up, but if/when NFC is activated, the I/O functionality changes to GPIO mode. To allow debugging when NFC is activate, the code to switch this I/O functionality should be removed.

The rest of this section describes specific setup steps for MCUXpresso IDE, and provides guidelines for using other IDEs, using the on-board debug probe. When using an external debug probe the steps are identical except for the selection of the target probe (which will vary by IDE use). See Section 4.1 for more information on using an external debug probe.

3.3.1 Using MCUXpresso IDE (using on-board debug probe)

Note that MCUXpresso IDE has built-in knowledge of the LPC8N04 part family, so does not require any SDK installation steps. Follow the steps below to build and run the application example from the LPC8N04 Board support package provided by NXP.

1. LPC8N04 driver firmware and the demo application for the LPC8N04 can be downloaded from nxp.com: http://www.nxp.com/demoboard/OM40002 (look under Software and Tools tab.) Select the MCUXpresso IDE version when downloading.
2. Open a new workspace in the MCUXpresso IDE.
3. In the Quickstart panel of the IDE, click in “Import a new project from the file system”
4. In the “Import project(s) from file system...” dialog box that opens, click “Browse...” in the Project Archive (from zip) section, and select the LPC8N04 code zip file downloaded in step (1) above. Click “Next >” on the “Import project(s) from file system...” dialog to continue.
5. You will several projects listed in the project bundle; click “Finish” to import them all.
6. The dialog box will close, and you will see the imported projects in the Project tab at the upper left window of the IDE. Click on app_demo to select it, then select Build from the Quickstart panel. This is the same firmware as pre-loaded in the Board during manufacture. You will see the build processing in the Console window to the right of the Quickstart panel. The projects are set up to include dependency checking, so the build process will automatically build the utility and peripheral libraries as well as the example program.
7. Ensuring the LPC8N04 Development Board is connected to the host computer, click Debug in the Quickstart panel. The IDE will search for available debug probes. Select the debug probe that appears for your board, then click OK. Note that the IDE will remember your selection for the next time you debug this project, so will not prompt for this again, unless it cannot find the board.
8. The code will execute to main. Press F8 to resume and run the program. You will now be able to interact with the demonstration, as described in Section 3.2.

3.3.2 Using 3rd Party IDEs

LPC8N04 driver firmware and the demo application for the LPC8N04 can be downloaded (from February 2018) from nxp.com: http://www.nxp.com/demoboard/OM40002 (look under Software and Tools tab.) Projects are provided with Keil MDK and IAR EWARM projects.

Set up will vary between Development Tools, but the board debug probe should supported as a CMSIS-DAP probe when using the on-board debug probe.

When using Keil MDK, install the Device Pack for the LPC8N04 (version 1.7 or later) before attempting to use the board.

When using IAR EWARM, ensure that you have version 8.20.2 or later in order to have LPC8N04 device family support.

3.4 Debugger firmware update (optional)

The MP board comes pre-programmed with a CMSIS-DAP firmware image; it is not normally necessary to update the debugger firmware image, but this can be done by following the steps below.

1. Download the LPC11U35 debug probe firmware from nxp.com (see Software and Tools tab for the board: http://www.nxp.com/demoboard/OM40002.)
2. Place a jumper on P2.
3. Press and release the reset. Using File Explorer (or equivalent on Mac/Linux platforms), look at the available drives on your system. A device called CRP_DISABLED will appear.
4. Delete the firmware.bin file on the CRP_DISABLED drive.
5. Drag and drop the firmware.bin file you downloaded from nxp.com on to the CRP_DISABLED drive.
6. Remove jumper on P2.
7. Repower the board.
8. The board should now enumerate on your system - allow 20-30 seconds for this to complete.

4. Debug Probe

The LPC11U35 debug probe on the LPC8N04 Development Board is pre-programmed with firmware to support the CMSIS-DAP protocol. See Section 3.4 for information on how to update the debug probe firmware, if needed.

The an LED is connected to the LPC11U35 device. The behavior of the LEDs will vary depending on firmware used, typically the LED will blink when debug communication is occurring.

The SWD connections from the LPC8N04 target are routed to headers P4 and P6. If the MP and DP boards are snapped apart, the P6 (MP side) and P5 (DP side) headers can be populated with right angled connectors (not supplied) and reconnected, so that the debug probe on the DP section can be used (or an external probe via the J1 connector), and power provided to the MP section from the DP section.

4.1 Using an external debug probe

An external debug probe that supports ARM's SWD interface, such as a SEGGER J-Link or PE Micro probe, can be used with the LPC8N04 Development Board board. The external probe must be connected J1. When an external debug probe is used, the on-board probe must be held in reset by placing a jumper on P1. It is recommended that P1 is fitted before powering the board.

5. Expansion connectors/headers

All the usable I/O signals from the LPC8N04 Target are routed to the P4 header, along with a reset signal and 3.3V power. 12 I/Os are shared with the 5x7 LED array, including the two SWD signals (PIO0_11 and PIO0_10), and PIO0_3 which is shared with the speaker driver. These three signal may be disconnected for all these functions using P7, P8 and P9; for other I/Os to be used the MP and DP sections must be separated or the 560 ohm LED row resistor (R26, R27, R29, R32, R36, R37 or R38) removed. For further details please refer to the board schematics.

Header P4 is designed to accept either a Grove I2C connector or PMod 6 pin SPI Type 2 connector, for easy connection of off-the-shelf sensors

5.1 Main expansion connectors (DP Board P3, MP Board P4)

The

Table 2. Arduino expansion connector pin mappings (CN3)

P3 pin	P4 pin	LPC8N04 pin	PMod connection	Grove connection
1	2	PIO0_0 (LED ROW5 / User Button)		
2	1	PIO0_4 (LED COL2)		SCL
3	4	PIO0_1 (LED ROW4)		
4	3	PIO0_5 (LED COL1)		SDA
5	6	PIO0_3 (SPKR/LED ROW6)		
6	5	VCC	VCC	VCC
7	8	PIO0_7 (LED ROW1)		
8	7	GND	GND	GND
9	10	PIO0_10 (SWCLK / LED COL3)		
10	9	PIO0_6 (LED COL0)	SCK	
11	12	PIO0_11 (SWDIO / LED COL4)		
12	11	PIO0_9 (LED ROW3)	SDO	
13	14	nc		
14	13	PIO0_8 (LED ROW2)	SDI	
15	16	RESET		
16	15	PIO0_2 (LED ROW0)	SSEL	

6. Power supplies and batteries

This section describes various ways to power the Development Board. The Board can be power distribution is shown in Figure 8. The LPC11U35 Debug Probe (and its associted LEDs) can only be powered by USB, but the rest of the circuitry can be powered by USB, one of the two button cells, the SWD connector or by using energy harvesting from an off-board NFC antenna. Two battery options are provided; Coin cell A (DP Section) is provided for demo applications that use the LED array. Coin cell B (MP section) is provided for use cases where the MP section has been separated from the DP section.

A power switch is provided on the MP section of the board to prevent inadvertent draining of the batteries when the Board is not in use; note that energy harvesting can still operate when the power switch is set in the off position.

Protection diodes are provided to prevent reverse powering of the coin cell batteries or external debug probe.

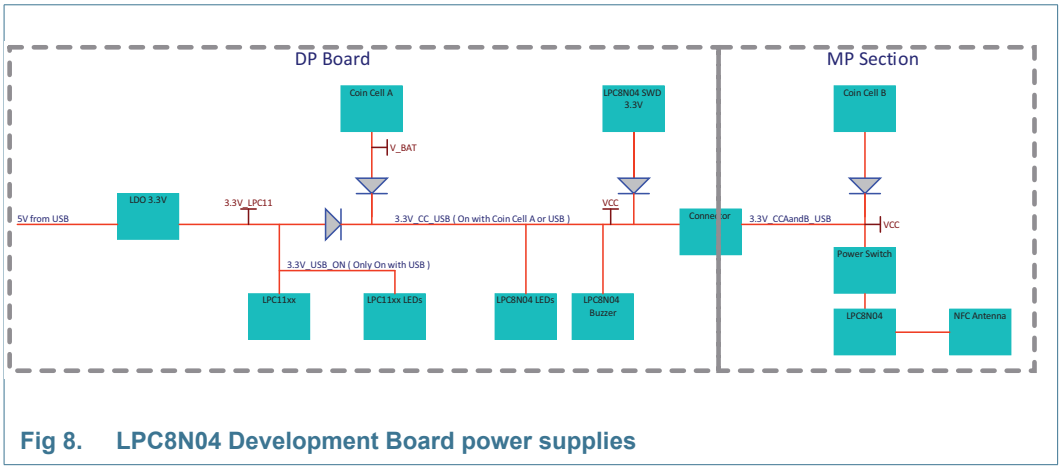

Fig 8. LPC8N04 Development Board power supplies

6.1 Measuring power consumption

Due to LED array and power amplifier circuits on the DP section of the board, it is necessary to separate the MP section of the board before attempting accurate power consumption measurements. Once the sections are separated, an ammeter may be connected in series with the power supply providing powe rto the P4 or P6 headers.

7. Other board features

This section describes other board features not detailed elsewhere in this document.

7.1 User button

The User button (S2) is for general purpose use by LPC8N04 applications. It is connected from 3.3V to LED row 5 via a 1k ohm resistor to the LPC8N04 PIO0_0 pin.

7.2 LED Array

An array of 35 LEDs, arrange as 5 columns by 7 rows, is provided on the reverse side of the DP section of the board. Connections are shown in Table 3 and Table 4. The row signals connect to the anode of all 5 LEDs in that row, and the cathodes of each LED are connected to the column signals. To turn on an LED, the row signal should be driven high and the column signal driven low. The LEDs are low current devices, with a nominal current draw of around 5mA, but the LED array is intended to be used with a row scanning technique, so only one row is driven at a time. If all LEDs are driven continuously this may lead to damage to the LPC8N04 device. An example of driving the LEDs with a scanning technique is provided in the app_demo example code mentioned in Section 3.3

Table 3. LED array row connections

LED row	LPC8N04 pin	Shared with
0	PIO0_2	
1	PIO0_7	
2	PIO0_8	
3	PIO0_9	
4	PIO0_1	
5	PIO0_0	
6	PIO0_3	Speaker (selected using P9)

Table 4. LED array column connections

LED column	LPC8N04 pin	Shared with
0	PIO0_6	
1	PIO0_5	
2	PIO0_4	
3	PIO0_10	SWCLK (selected using P8)
4	PIO0_11	SWDIO (selected using P7)

7.3 Speaker and Speaker Driver

A surface mount speaker (LS1) is provided on the reverse side of the DP section of the Board. The speaker is driven by a driver built from discrete transistors in a Class B topology, with PIO0_3 used to provide the input signal. P9 must be set to connect PIO0_3 to the driver circuit (see Figure 3). An 820 resistor to ground is connected at the amplifier input in order to avoid excessive current draw when the amplifier is not being used (i.e. when PIO0_3 is not driven or is disconnected at P3.)

Note that the amplifier current draw is relatively high, so it is not recommended to use the speaker with Coin cell B, and energy harvesting will not support speaker operation.

8. Legal information

8.1 Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the *Terms and conditions of commercial sale* of NXP Semiconductors.

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.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors accepts no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from national authorities.

8.2 Trademarks

Notice: All referenced brands, product names, service names and trademarks are the property of their respective owners.

9. Contents