

UM11057

LPCXpresso845MAX

Rev. 1.0 — 9 June 2017

User Manual

Document information

Info	Content
Keywords	LPCXpresso845MAX, OM13097, LPC845, LPC844
Abstract	LPCXpresso845MAX User Manual



Revision history

Rev	Date	Description
1.0	20170526	First draft

Contact information

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

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

1. Introduction

The LPCXpresso-MAX family of boards provides a powerful and flexible development system for NXP's low end Cortex-M0+ MCUs. They can be used with a range of development tools, including the MCUXpresso IDE toolchain. The LPCXpresso845MAX board is developed by NXP to enable evaluation of and prototyping with the LPC84x family of MCUs. [Figure 1](#) shows the LPCXpresso845MAX Board.

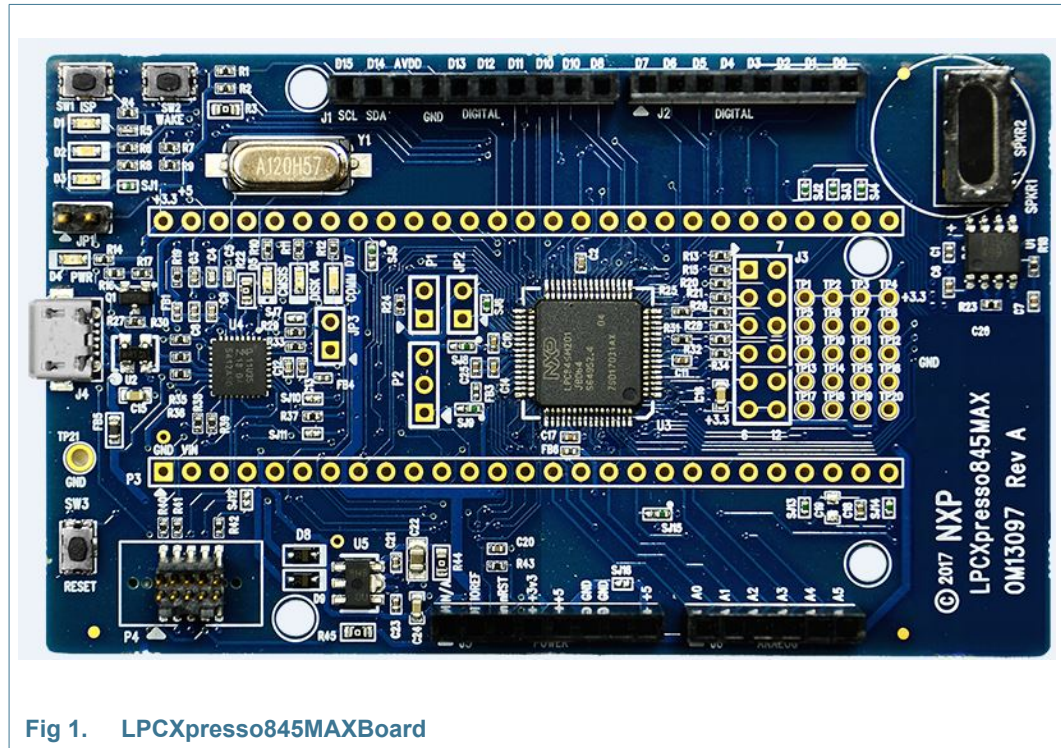


Fig 1. LPCXpresso845MAXBoard

The LPCXpresso845MAX board includes the following features:

- Compatible with MCUXpresso IDE and other popular toolchains (incl. IAR and Keil).
- On-board CMSIS-DAP (debug probe) with VCOM port, based on LPC11U35 MCU.
- Debug connector to allow debug of target MCU using an external probe.
- Red, Green and Blue User LEDs.
- Target ISP and User/Wake buttons.
- Target Reset button.
- LPCXpresso expansion connector.
- DAC output via speaker driver and speaker.
- Arduino connectors compatible with the “Arduino UNO” platform.
- Pmod® compatible expansion header.
- Prototyping area.

2. Board Layout

Figure 2 below shows the layout of the LPCXpresso845MAX board, indicating location of jumpers, buttons, connectors/expansion options and MCU devices.

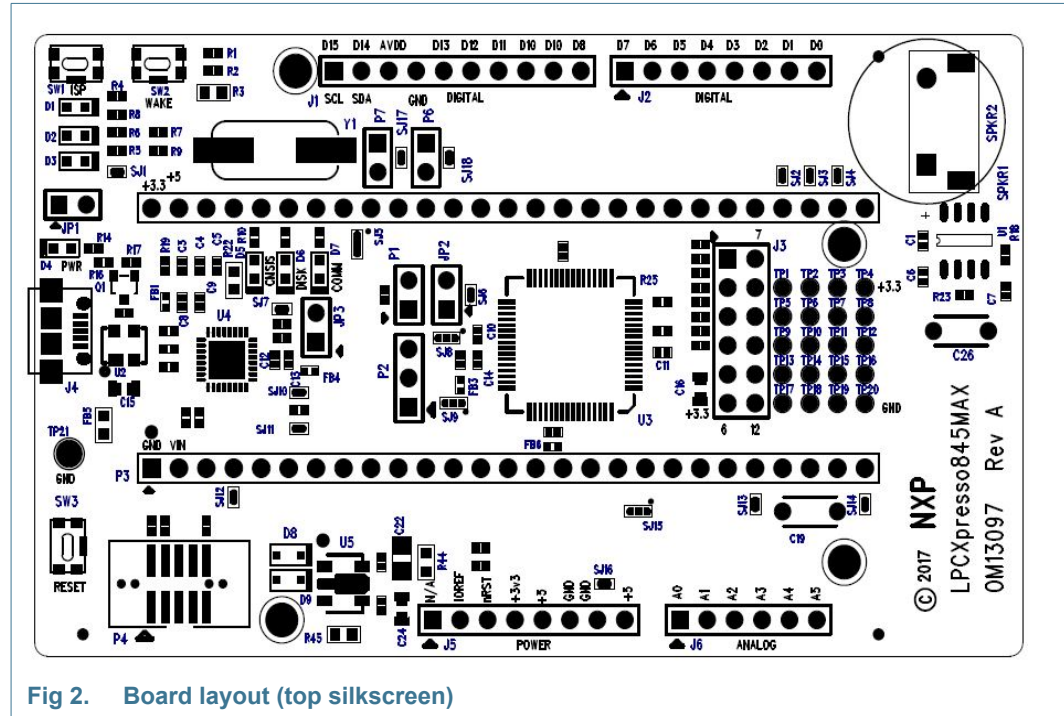


Fig 2. Board layout (top silkscreen)

Table 1 shows the layout of the LPCXpresso845MAX board, indicating location of jumpers, buttons, connectors/expansion options and MCU devices.

Table 1. Jumpers and connectors

Circuit reference	Description	Reference section
JP1	Inserting a jumper on JP1 holds the LPC11U35 debug probe in reset. Install JP1 when using an external debug probe.	Section 4.1
JP2	Ammeter connection. After removal of SJ6, an ammeter can be installed between the pins of JP2 to measure current flow to the LPC845. Once SJ6 has been removed, JP2 must either be installed or an ammeter in place in order for the LPC845 to be powered.	Section 6
JP3	LPC11U35 Device Firmware Update (DFU) mode enable. Install JP3 before powering up the board via USB in order to place the LPC11U35 in DFU mode (thus enabling software update via drag and drop programming on a host computer.)	Section 3.1
P1	Power measurement point. Connect a voltmeter across P1 to measure voltage drop across a 2.43 ohm resistor connected in-line with the supply to the LPC845 device.	Section 6
P2	VREFP/VREFN header connection. By default, VREFP of the LPC845 is connected to the 3.3V regulator output, and VREFN to ground. By removing connection 1-2 on SJ9 and bridging pads 2-3 instead, VREFP can be injected via pin 1 of header P2. By removing connection 1-2 on SJ8 and bridging pads 2-3 instead, VREFN can be injected via pin 3 of header P2.	n/a

Table 1. Jumpers and connectors

Circuit reference	Description	Reference section
P3	LPCXpresso expansion connector.	Section 5.3
P4	External SWD debug connector for LPC845.	Section 4.1
P5	External SWD debug connector for LPC11U35 (not installed, would be mounted on underside of PCB.)	Section 4
P6	Once SJ18 has been removed, P6 can be used to connect/disconnect the LPC845 UART TX from the LPC11U35 debug probe (VCOMM connection).	Section 5.3
P7	Once SJ17 has been removed, P7 can be used to connect/disconnect the LPC845 UART RX from the LPC11U35 debug probe (VCOMM connection).	Section 5.3
J1, J2, J5, J6	Arduino expansion connectors.	Section 5.1
J3	PMod-compatible header.	Section 5.2
J4	USB connector for debug probe / board power.	Section 4

3. Getting Started

The LPCXpresso845MAX board is pre-programmed with a simple program to blink a user LED, indicating that the target MCU is running. Connect a micro USB cable from connector J4 to a host computer or power supply to power up the board and run this program.

The rest of this section describes how to set up your board for use with MCUXpresso IDE and/or third party tools.

3.1 Debugger firmware update

The LPCXpresso845MAX board comes pre-programmed with a CMSIS-DAP firmware image. Early production boards are programmed with firmware based on the mbed CMSIS-DAP port which requires installation of the mbed serial port driver on Windows platforms before the board can be used (note that the LPCXpresso845 board is not mbed enabled). Later boards do not require this driver installation, and it is recommended that the latest debugger firmware is downloaded onto the board before starting a debug session. The debugger firmware is available from <http://nxp.com/demoboard/om13097> under the Software and Tools tab (look under “Software” download types).

After downloading the debugger firmware image (which will be called firmware.bin), follow these steps:

1. With the board unpowered, install jumper JP3.
2. Power the board. Using File Explorer (or equivalent on Mac/Linux platforms), look at the available drives on your system. A device called CRP_DISABLED will appear.
3. Delete the firmware.bin file on the CRP_DISABLED drive.
4. Drag and drop the firmware.bin file you downloaded from nxp.com on to the CRP_DISABLED drive.
5. Remove JP3, unplug then repower the board.
6. The board should now enumerate on your system - allow 20-30 seconds for this to complete.

3.2 Using the board with MCUXpresso IDE

Once the board debugger firmware has been installed (as described in [Section 3.1](#)), to use the board during a debug session in MCUXpresso IDE, connect to the host computer then follow the steps below.

Note that MCUXpresso IDE has built-in knowledge of the LPC84x part family, so does not require any SDK installation steps. Follow the steps below to build and run a simple example from the LPC84x Code Bundles provided by NXP.

Code Bundles for the LPC8xx family are included in the MCUXpresso IDE installation. These can also be downloaded from nxp.com (in case of any updates between IDE releases): <https://www.nxp.com/LPC800-Code-Bundles>. (Note that MCUXpresso IDE has a link to the Code Bundle Page in the Help -> Additional Resources menu).

1. Open a new workspace in the IDE.
2. In the Quickstart panel of the IDE, click in “Import project(s) from the file system”

3. In the “Import project(s) from file system...” dialog box that opens, click “Browse...” in the Project Archive (from zip) section, and select the LPC84x Code Bundle zip file from the Code Bundles directory in the MCUXpresso IDE installation (or select a version downloaded from nxp.com, as described in Step 1 above.) Click “Next >” on the “Import project(s) from file system...” dialog to continue.
4. You will see several projects listed in the Code Bundle; click “Finish” to import them all.
5. 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 Example_Multi_Timer_Blinky to select it, then select Build from the Quickstart panel. 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.
6. Ensuring the LPCXpresso845MAX 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.
7. The code will execute to main. Press F8 to resume and run the program. You will now see the User LEDs light, each color in turn.

3.3 Using LPCXpresso845MAX with 3rd Party IDEs

Once the board debugger firmware has been installed (as described in [Section 3.1](#)), to use the board during a debug session in MCUXpresso IDE, it can be used with 3rd party development tools. Set up will vary between Development Tools, but the board debug probe should be supported as a CMSIS-DAP probe in all cases).

Code Bundles, including sample projects for Keil MDK and IAR EWARM, are available from NXP’s website at <http://www.nxp.com/products/software-and-tools/software-development-tools/software-tools/lpc800-code-bundles:LPC800-Code-Bundles>. Refer to the readme files provided with each code bundle in order to build/debug.

When using Keil MDK, install the Device Pack for the LPC84x (version 1.5 or later) before attempting to use the board.

When using IAR EWARM, ensure that you have version 8.12.2 or later in order to have LPC84x device family support.

4. Debug Probe

The on-board LPC11U35 provides CMSIS-DAP debug probe functionality, plus a virtual comm port (VCOM) capability via PIO1_16 and PIO1_17 of the LPC845 target. This functionality bridges the LPC845 serial port via USB, so that PC applications such as TeraTerm and PuTTY can communicate directly with the target.

Initial production boards have been programmed with firmware based from mbed, so require the mbed serial port driver to be installed while the board is attached in order to run correctly. See: <http://mbed.org/handbook/Windows-serial-configuration#1-download-the-mbed-windows-serial-port> for more information. Note that the LPCXpresso845MAX is not an mbed certified platform, and does not support drag and drop programming of the LPC845 target. When using the mbed firmware image, the VCOM settings used should be 9600 baud, 8 bits, no parity, 1 stop bit, no flow control.

Later production LPCXpresso845MAX boards have been programmed with an updated version of CMSIS-DAP, and a standard UART VCOM port. It is recommended that the LPC11U35 firmware be updated to the latest version available on the LPCXpresso845 board page, under the Software and Tools tab. See [Section 3.1](#) for information on how to update the debug probe firmware.

The CMSIS, DISK and COMM LEDs are connected to the LPC11U35 device. The behavior of the LEDs will vary depending on firmware used, typically the CMSIS LED will blink when debug communication is occurring, and the COMM LED will blink when data is being transferred over the VCOMM port.

The board design includes pads for an external debug probe to be connected to the LPC11U35 (connector P5). This connector is not installed, and is not intended for customer use.

4.1 Using an external debug probe

An external debug probe that supports ARM's SWD interface, such as an LPC-Link2 (OM13054), SEGGER J-Link or PE Micro probe, can be used with the LPCXpresso845MAX board. The external probe must be connected to header P4. When an external debug probe is used, the on-board probe must be held in reset by placing a jumper on JP1. It is recommended that JP1 is fitted before powering the board.

5. Expansion connectors/headers

The LPCXpresso845MAX board provides multiple options to add additional circuitry or off the shelf expansion boards; this section describes these options. For further details please refer to the board schematics.

5.1 Arduino UNO Rev 3 expansion connectors

The Arduino UNO Rev 3 compatible connectors provided on the LPCXpresso845MAX board provide I2C, SPI, UART, PWM and analog function connections to shield boards that are available from various 3rd part suppliers, or for customer use. The pin mappings are shown in the tables below. Some connections are shared with the LPCXpresso connector (P3), as shown.

Table 2. Arduino expansion connector pin mappings (Digital connector J1)

J1 Pin	Arduino signal	LPC845 pin	LPCXpresso connector pin
1	I2C SCL	PIO0_10	41
2	I2C SDA	PIO0_11	40
3	AREF	n/a (3.3V)	n/a
4	GND	n/a (GND)	n/a
5	SPI SCK	PIO0_6	7
6	SPI MISO	PIO0_7	6
7	SPI MOSI	PIO1_19	5
8	SPI SSEL	PIO1_18	42
9	PWM	PIO1_14	n/a
10		PIO0_16	n/a

Table 3. Arduino expansion connector pin mappings (Digital connector J2)

J2 pin	Arduino signal	LPC845 pin	LPCXpresso connector pin
1		PIO0_1	43
2	PWM	PIO0_28	n/a
3	PWM	PIO1_20	n/a
4		PIO1_21	n/a
5	INT/PWM	PIO0_20	n/a
6	INT	PIO0_21	n/a
7	TX	PIO0_25	48
8	RX	PIO0_24	49

Table 4. Arduino expansion connector pin mappings (Analog connector J6)

J1 Pin	Arduino signal	LPC845 pin	LPCXpresso connector pin
1	A0	PIO0_14	15
2	A1	PIO0_23	16
3	A2	PIO0_19	17
4	A3	PIO0_18	19
5	A4	PIO0_17	20
6	A5	PIO0_29	18

Note that the default ports for ISP boot are connected to J1 (I2C) J2 (USART).

The Arduino power connector (J5) pin 2 (IOREF) and pin 3 (+3V3) are routed to the 3.3V regulator output. J5 pin 5 (5V) is routed to the 5V supply from USB. J5 pin 8 (VIN) can also be connected to this 5V supply by closing solder jumper SJ16 (this connection is required by some Arduino shields).

5.2 PMod connector

The LPCXpresso845MAX board is designed to accommodate a 0.1” header for PMod compatible boards (the header is not installed during manufacture). The circuit reference for this header is J3. The board provides SPI, I2C and GPIO connections from the LPC845 target, each signal routed via a 100 ohm connector to prevent accidental damage.

Table 5. PMod connector pin mappings (J3)

J3 Pin	PMod signal	LPC845 pin
1	SPI SSEL	PIO0_22
2	SPI MOSI	PIO0_27
3	SPI MISO	PIO0_26
4	SPI SCK	PIO0_15
7	GPIO	PIO1_10
8	GPIO	PIO0_13
9	I2C SCL	PIO1_12
10	I2C SDA	PIO1_13

5.3 LPCXpresso expansion connector (P3)

The LPCXpresso expansion connector follows the signal / port mappings common to other MAX-style LPCXpresso, LPCXpresso V2, LPCXpresso-CD and the original LPCXpresso boards. Refer to the schematic for a full list of pin connections.

The USART connections from LPC845 that are routed to the LPC11U35 debug probe are also present on P3 (pins 13 & 14). If using P3 to access this peripheral, disconnect the debug probe connections by removing SJ17 and SJ18. Note that if SJ17 and SJ18 are removed then headers P6 and P7 may be installed to provide a convenient way of connecting/disconnecting the USART to the debug probe.

5.4 Prototyping area

A prototyping (or “breadboarding”) area is provided, adjacent to the PMod connector, made up of 20 holes. The top row (TP1 through TP4) are connected to the 3.3V regulator output, and the bottom row (TP17 through 20) are connected to ground. The remaining holes (TP5 through TP16) are not connected to anything else on the board.

6. Power measurement

JP2 is provided to enable supply current to the LPC845 to be measured by placing an ammeter in line with JP2 pins. By default this jumper is bypassed by solder jumper (SJ6). Header P1 may be installed to measure supply power to the LPC845 by measuring voltage drop across a 2.43 ohm resistor on the board.

7. Other board features

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

7.1 ISP booting and the ISP button

The LPC845 can be forced into ISP boot mode by holding down the ISP button (SW1) and then holding and releasing reset. The ISP button is connected via a 100 ohm resistor to LPC845 pin PIO0_12, which is also routed to the LPCXpresso expansion connector (pin 45) and to the cathode of the red user LED. LPC845 pin PIO0_12 can be reconfigured by software so that the button can be used by an application as a general purpose button (refer to the LPC84x User Manual).

7.2 Wake button

The Wake button is for general purpose use by LPC845 applications. It is connected via a 100 ohm resistor to a 100K ohm pullup to 3.3V, and to the LPC845 PIO0_4 pin. This pin can wake the LPC845 from deep power-down. Note that PIO0_4 is also connected to the P3 LPCXpresso expansion connector (pin 50).

7.3 User LEDs

Three users LEDs are provided on the board, one blue, one green and one red. The anodes of the LEDs are connected together, through solder jumper SJ1 to the 3.3V regulator output. SJ1 can be opened to prevent leakage from the LPC845 pins through the diodes if accurate power measurement measurements are being performed. The LEDs are connected to the LPC845 as shown in [Table 6](#).

Table 6. LED connections

LED	LPC845 pin	Notes
Blue (D2)	PIO1_15	Shared with LPCXpresso connector pin 44
Red (D3)	PIO0_12	Shared with ISP pin and LPCXpresso connector pin 45
Green (D1)	PIO0_0	-

7.4 Speaker driver

The LPC845 DACOUT output PIO0_29 is route to an IS31AP4991 speaker driver (see Integrated Silicon Solutions web site for datasheet). This device can drive 8 to 32 ohm speakers; a 32 ohm speaker is supplied with the board and resistors have been selected to provide a reasonable volume level for musical tones. If desired, a capacitor can be added in parallel with the speaker driver feedback resistor, in order to reduce the speaker circuit bandwidth and reject DAC output sampling artifacts. This capacitor can be added using the through holes with circuit reference designator C26 (also called CAP NP in the schematic.)

8. Legal information

8.1 Definitions

Draft — The document is a draft version only. The content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included herein and shall have no liability for the consequences of use of such information.

8.2 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.

8.3 Trademarks

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

9. Contents

1	Introduction	3
2	Board Layout	4
3	Getting Started	6
3.1	Debugger firmware update	6
3.2	Using the board with MCUXpresso IDE	6
3.3	Using LPCXpresso845MAX with 3rd Party IDEs	7
4	Debug Probe	8
4.1	Using an external debug probe	8
5	Expansion connectors/headers	9
5.1	Arduino UNO Rev 3 expansion connectors ...	9
5.2	PMod connector	10
5.3	LPCXpresso expansion connector (P3)	10
5.4	Prototyping area	10
6	Power measurement	11
7	Other board features	11
7.1	ISP booting and the ISP button	11
7.2	Wake button	11
7.3	User LEDs	11
7.4	Speaker driver	12
8	Legal information	13
8.1	Definitions	13
8.2	Disclaimers	13
8.3	Trademarks	13
9	Contents	14

Please be aware that important notices concerning this document and the product(s) described herein, have been included in section 'Legal information'.

© NXP B.V. 2017.

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: 9 June 2017

Document identifier: UM11057