UM11035

LPCXpresso546x8/540xx Board User Manual

Rev. 2.0 — 12th January 2018　　　　　　　　　　　　　　**User manual**

Revision history

Rev	Date	Description
2.0	20180112	Added LPCXpresso54018, modified title
1.2	20170525	Added LPCXpresso54628, included reference to OM13099.
1.1	20170407	Updated for MCUXpresso and CAN-FD Kit/Shield information
1.0	20161118	Initial revision

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™ family of boards provides a powerful and flexible development system for NXP's LPC Cortex®-M family of MCUs. They can be used with a wide range of development tools, including NXP's MCUXpresso IDE. The LPCXpresso54608 (OM13092), LPCXpresso54618 (board provided as part of the OM13094 CAN-FD Kit) and LPCXpresso54628 (OM13098) share the same design and have been developed by NXP to enable evaluation of and prototyping with the LPC546x8/540xx family of MCUs. There is no functional difference between these boards except the (1) functionality of the LPC546x8/540xx device installed, and (2) that the LPC54618 board included in the OM13094 does not include the LCD panel. The LPCXpresso54018 also uses the same board design and components, but has different booting behavior due to differences in the LPC54018 MCU boot ROM. All boards use a BGA180 package.

Note that the LPCXpresso54628 board features an LPC54628 device which has a superset of functionality compared to the LPC54608 and LPC54618. This means code written for the LPC54608 and LPC54618 can run unchanged on the LPC54628. The LPC54628 is capable of running at up to 220MHz, but requires a different power library (from the MCUXpresso SDK) than the other devices in order to run at this clock speed.

The LPCXpresso54018 board features and LPC54018 MCU which can operate at up to 180MHz. This MCU also includes CAN-FD support. While most of the peripherals are identical between LPC540xx and LPC546x8 devices, a different SDK is required for each family.

The schematics show that the board was designed for either 1.8 V or 3.3 V build configurations, but all production boards are built in the 3.3 V configuration.
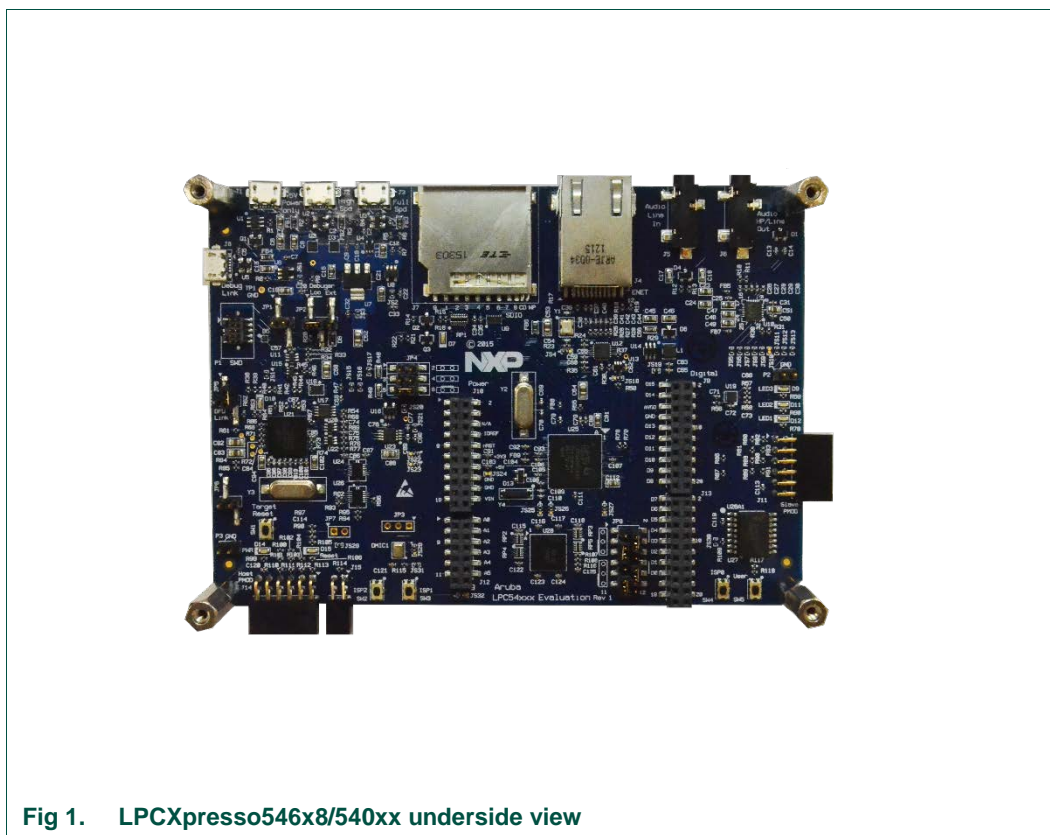
Fig 1.   LPCXpresso546x8/540xx underside view

This document describes the hardware of the LPCXpresso546x8/540xx boards, plus information on the CAN-FD shield supplied in the OM13094 kit (this shield board is also available as a standalone product, order code OM13099.) This shield board can be used with any of the LPCXpresso boards described, but the LPCXpresso54608 does not support CAN-FD.

The following aspects of interfacing to the board are covered by this guide:

- Main board features.
- Setup for use with development tools.
- Board interface connectors.
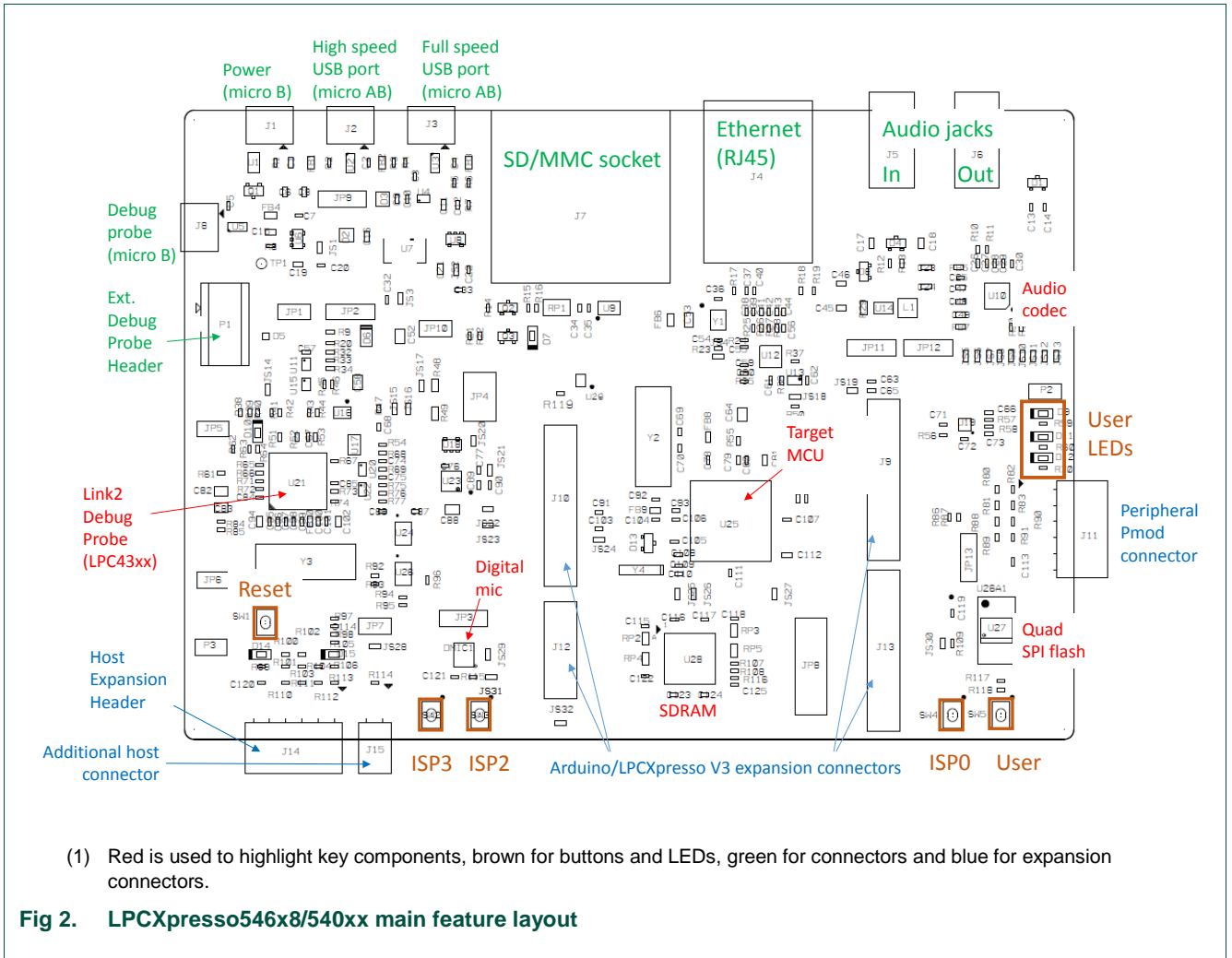- Jumper settings.

2.   Feature summary

The LPCXpresso546x8/540xx board includes the following features:

- On-board, high-speed USB based, Link2 Debug Probe with CMSIS-DAP and SEGGER J-Link protocol options:
  - Link2 probe can be used with on-board LPC546x8/540xx or external target.
  - UART and SPI port bridging from LPC546x8/540xx target to USB via the on-board Debug Probe.
  - Support for external Debug Probe.

- 3 x user LEDs
- Target Reset, ISP (3) and user buttons
- Expansion options based on popular standards:
  - Arduino UNO compatible expansion site with additional LPCXpresso V3 standard connections
  - PMod™ compatible expansion port
  - Host connection / general purpose expansion port
- On-board 3.3V regulator with external power supply options.
- Built-in power consumption measurement for target LPC546x8/540xx MCU.
- Winbond 128Mb W25Q128JVFIM Quad-SPI flash (Micron devices used on all revisions up to Rev C and some Rev D.)
- Winbond 128Mb W9812G6JB-6I SDRAM (Micron MT48LC8M16A2B4 used on all revisions up to Rev C and some Rev D.).
- Knowles SPH0641LM4H digital microphone.
- Full size SD/MMC card slot.
- NXP MMA8652FCR1 accelerometer.
- Stereo audio codec with line in/out.
- High and full speed USB ports with micro A/B connector for host or device functionality.
- 10/100Mbps Ethernet (RJ45 connector).
- 272x480 color LCD with capacitive touch screen.

2.1 Board layout and settings

This section provides a quick reference guide to the main board components, configurable items, visual indicators, and expansion connectors. Fig 2 shows the layout of the components on the LPCXpresso546x8/540xx board.

(1) Red is used to highlight key components, brown for buttons and LEDs, green for connectors and blue for expansion connectors.

Fig 2.   LPCXpresso546x8/540xx main feature layout

The LCD panel is mounted on the reverse side of the board, connected to the circuitry via two flex cable connectors. It should not normally be necessary to remove the LCD or access these connectors; the LCD is held in place by 4 double-sided adhesive pads.

Fig 3 shows the location of indicators and jumpers.

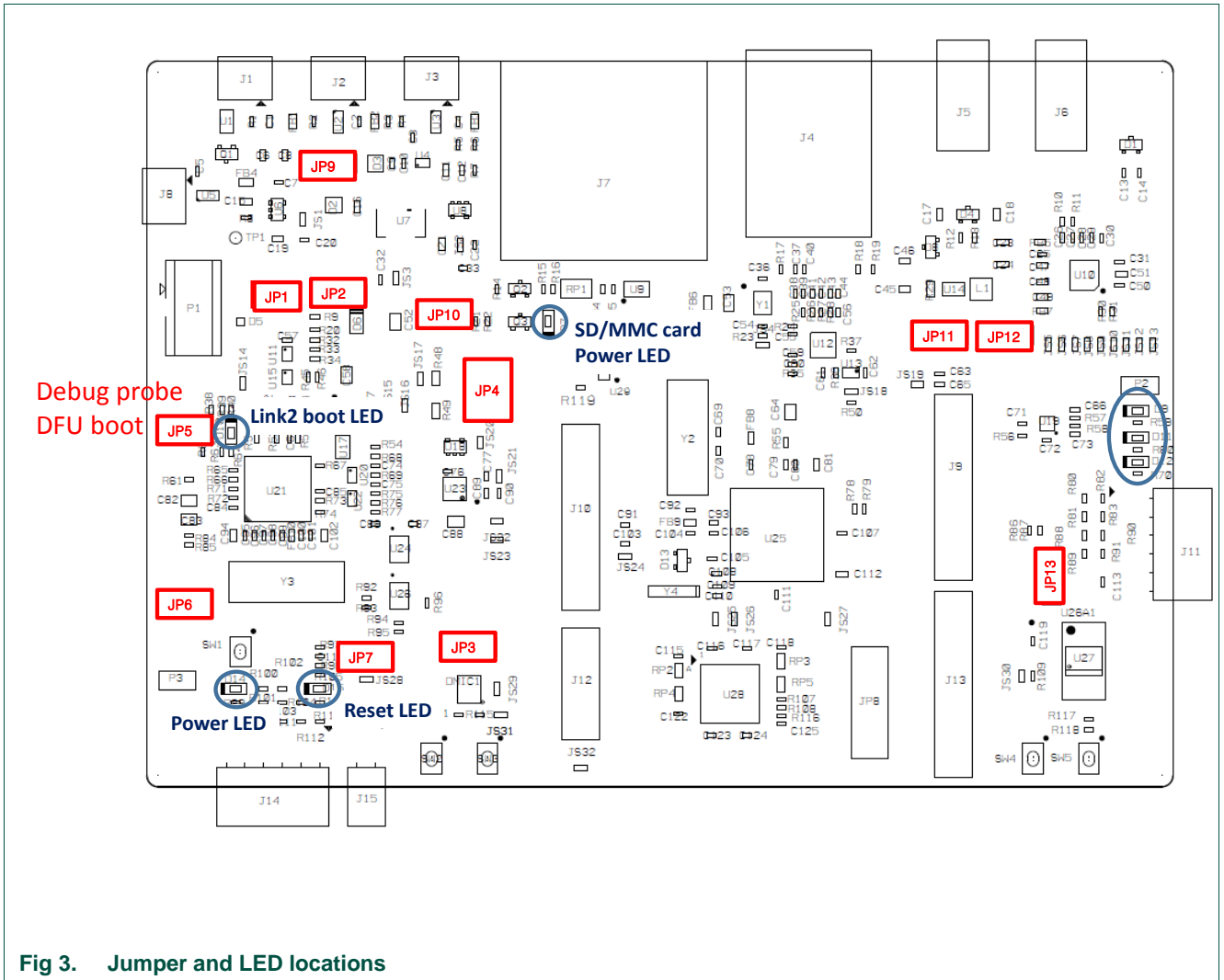**Fig 3.   Jumper and LED locations**

[Table 1](#) lists the function of each jumper.

Table 1.   Jumpers

Circuit ref	Description	Section
JP1	**Target processor selection for the on-board Debug Probe.**	[3](#), [4](#)
	Jumper open (default) the LPC546x8/540xx Target SWD interface enabled. Normal operating mode where the Target SWD is connected to either the on-board Link2 Debug Probe or an external Debug Probe.	
	Jumper shunted, the LPC546x8/540xx Target SWD interface is disabled. Use this setting only when the on-board Link2 Debug Probe is used to debug an off-board target MCU.	

UM11035

© NXP B.V. 2017, 2018. All rights reserved.

User manual | **Rev. 2.0 — 12th January 2018** | **7 of 34**

Circuit ref	Description	Section
JP2	**Buffer Power Selection** For On-board Target place in position 1-2 (default) For Off-board Target place in position 2-3	3, 4
JP3	This header (not installed by default) provides a convenient connection point to provide external ADC positive and negative voltages. To inject these voltages at this header SJ22 (for VREFN) and/or SJ23 (for VREFP) need to be moved from the default 1-2 position to the 2-3 position.	See Schematic
JP4	This set of 3 jumpers control various selections for power measurement: Position 1-2 and 3-4 are in parallel with 1 ohm resistors. Current can be measured across these jumper headers to determine current flow into the LPC546x8/540xx target. Position 5-6 (installed by default) can be left open and a current meter connected between these pins to directly measure current flow into the LPC546x8/540xx target.	
JP5	Link2 (LPC43xx) force DFU boot – 2 position jumper pins. 1) Jumper open (default) for Link2 to follow the normal boot sequence. The Link2 will boot from internal flash if image is found there. With the internal flash erased the Link2 normal boot sequence will fall through to DFU boot. Jumper shunted to force the Link2 to DFU boot mode. Use this setting to reprogram the Link2 internal flash with a new image (using the LPCScrypt utility) or to use the MCUXpresso IDE with CMSIS-DAP protocol. Note that the LPCXpresso546x8/540xx Link2 flash is pre-programmed with a version of CMSIS-DAP firmware by default.	
JP6	**Bridge / Host Expansion Header selector**. **Revision B boards:** When open (default), the SPI connections from Flexcomm3 the LPC546x8/540xx are driven to the Link2 Debug Probe. Install JP6 when using the SPI interface at connector J14. Note that this disables the Link2 SPI (bridge) probe connection. **Revision C and later boards:** When open (default), the "Bridge" UART and SPI connections from the Link2 probe are driven to the LPC546x8/540xx target. Install JP6 when using the SPI interface at connector J14 and/or FC0 UART at P4 (FTDI). Note that this disables the Link2 SPI and UART (bridge) probe connections. **Note that the I2C SCL/SDA connections are reversed on Rev C and earlier boards (relative to the PMod standard). This was corrected from Revision D onwards. See Section 6.3.**	

Circuit ref	Description	Section
JP7 (not installed by default)	JP7 may be fitted to provide a convenient way to enable/disable the reset signal to/from the expansion connectors. Solder jumper JS28 should be removed if JP7 is to be used.	See schematic
JP8	This set of jumpers is used to configure 3.3V vs 1.8V board configurations. OM13092 boards are all 3.3V, so these jumpers should not be changed from the default. JP8 is not present on Revision C boards (functionality has been replaced by zero ohm links).	See schematic
JP9	**USB host Vbus selection** Note that only one of USB0 or USB1 can be configured as a USB host port at any given time (this is a board restriction, not a limitation of the LPC546x8/540xx.) Install jumper in position 1-2 for USB1 (High Speed) to provide Vbus (i.e. enable USB host capability) (Default) Install jumper in position 2-3 for USB0 (Full Speed) to provide Vbus (i.e. enable USB host capability)	
JP10	**USB host power control selection** This jumper selects routing of USB port power and overcurrent detect from either the USB0 or USB1 ports of the LPC546x8/540xx. Note that only one of USB0 or USB1 can be configured as a USB host port at any given time (this is a board restriction, not a limitation of the LPC546x8/540xx.) Leave open when using USB1 (High Speed) as a USB host (Default) Install jumper for USB0 (Full Speed) to provide Vbus (i.e. enable USB host capability)	
JP11 & JP12	**USB0 host / Ethernet TXD/RXD selection** Due to sharing of pin functionality on this development board, it is not possible to support the Ethernet port and USB0 overcurrent feature simultaneously. For both JP11 and JP12: Install jumper in position 1-2 to enable Ethernet (Default) Install jumper in position 2-3 for USB0 (Full Speed) (i.e. enable USB host capability)	
JP13	**USB0 host / Ethernet selection** Due to sharing of pin functionality on this development board, the P4-7 port pin is used for either the USB0 port or as a general purpose signal on the expansion connector. Install jumper in position 1-2 to route P4-7 to the expansion connector. Install jumper in position 2-3 for USB0 (Full Speed) (i.e. enable USB host capability)	

Table 2 describes the board LED and button functions, and connectors.

Table 2.   LEDs, buttons and connectors

Circuit Ref	Description	Section
D7	**SD card slot power enable** This LED illuminates when power is enabled to the SD card slot (controlled by LPC546x8/540xx port P2-5.)	n/a
D9, D11, D12	**User LEDs** These LEDs are for application use. They are illuminated when the driving signal from the LPC546x8/540xx is low. The LEDs are driven by ports P2-2 (D9), P3-3 (D11) and P3-14 (D12).	n/a
D10	**Link2 boot mode** Link2 LPC43xx BOOT0_LED indicator. Reflects the state of LPC43xx Link2 MCU P1_1. When the boot process fails, D1 will toggle at a 1 Hz rate for 60 seconds. After 60 seconds, the LPC43xx is reset.	n/a
D14	**Target power** This LED illuminates when the 3.3V supply to the LPC546x8/540xx is present.	n/a
D15	**Reset LED** This LED illuminates when reset is asserted either via the expansion connector or when reset button SW1 is pressed.	n/a
SW1	**Reset button** Press and release this button to reset the LPC546x8/540xx. Note that this does not reset the Link2 Debug Probe.	n/a

UM11035

User manual **Rev. 2.0 — 12th January 2018** **10 of 34**

Circuit Ref	Description	Section
SW2, SW3, SW4	**ISP / User buttons** These switches can be used to force the LPC546x8/540xx in to ISP boot modes, as shown below. Note that ISP boot is also affected by OTP bit settings, and behavior will also be modified based on port activity as the boot ROM executes. Refer to the device User Manual for more information. Signal is low when the button is pressed.	

LPC546xx boot modes:

Mode / Boot source	ISP2 (P0-6)	ISP1 (P0-5)	ISP0 (P0-4)
Internal flash boot	High	High	High
USB1 (High speed)	Low	High	High
CAN	High	Low	Low
USB0 (Full speed)	High	Low	High
USART/I2C/SPI	High	High	Low

LPC540xx boot modes:

Mode / Boot source	ISP2 (P0-6)	ISP1 (P0-5)	ISP0 (P0-4)
Auto boot: If OTP BOOT_SRC not set, LPC540xx will look for a valid image in SPIFI flash, external SPI flash then external parallel memory. If not image is found, ISP boot will commence.	High	High	High
USART/I2C/SPI	High	High	Low
SPI Boot: Boot from SPI NOR flash connected to Flexcomm 0. If no valid image is found, ISP boot will commence.	High	Low	High
Reserved	High	Low	Low
SPIFI boot: Boot from QSPI NOR flash device connected to SPIFI interface. If no valid image is found, ISP boot will commence.	Low	High	High
USB0 (full speed) ISP DFU	Low	High	Low
USB1 (high speed) ISP DFU	Low	High	Low

The ISP pins are sampled by the LPC546x8/540xx boot ROM code immediately following reset, so to initiate an ISP boot press and hold the required ISP buttons while pressing and releasing the reset button (SW1.)

Following reset, these buttons may also be used by a user application.

UM11035

User manual **Rev. 2.0 — 12th January 2018** 11 of 34

Circuit Ref	Description	Section
SW5	**User button**	
	This button is connected to LPC546x8/540xx port pin P1-1, and is provided for user applications. Port P1-1 is pulled to ground when the button is pressed.	
J1	**External +5V power**	
	Micro USB connection for power to the LPC546x8/540xx target and peripheral circuitry (excluding Link2 Debug Probe).	
J2	**LPC546x8/540xx High Speed USB connector (USB1)**	
	This micro AB connector enables connection from the LPC546x8/540xx USB1 port to host or slave devices. An adaptor (not supplied) is typically required to connect USB slave devices (mouse, keyboard, etc.)	
	Note that when using this USB port as a host, power must be supplied via the J1 connector in order to power the USB device being connected to the board.	
J3	**LPC546x8/540xx Full Speed USB connector (USB0)**	
	This micro AB connector enables connection from the LPC546x8/540xx USB1 port to host or slave devices. An adaptor (not supplied) is typically required to connect USB slave devices (mouse, keyboard, etc.)	
	Note that when using this USB port as a host, power must be supplied via the J1 connector in order to power the USB device being connected to the board. Also note that jumpers JP9 through JP13 must be changed from their default position to use this port in host mode.	
J4	**LPC546x8/540xx Ethernet connector**	
	This RJ45 connector provides a 10/100Mbps connection to the Ethernet PHY being driven by the LPC546x8/540xx.	
J5	**Audio line input jack**	
	3.5mm audio input jack for the audio codec	
J6	**Audio line output jack**	
	3.5mm audio input jack for the audio code	
J7	**SD/MMC card slot**	
	Full size SD/MMC card slot connected to the SDIO interface of the LPC546x8/540xx.	
J8	**Link2 Debug Probe connector**	
	Micro USB type B connection for the on-board Link2 Debug Probe. Note: do not use this connection when using an external Debug Probe.	
J9, J10, J12, J13	**Expansion connectors**	
	0.1" pitch connectors for addition of Arduino R3 shield or other expansion daughter boards / circuitry.	
J11	**Peripheral expansion PMod connector**	
	0.1" pitch 2x6 connector following the PMod standards. This connector is primarily intended for adding external peripherals using I^2C and/or SPI bus, but is also suitable for general purpose I/O connections.	

Circuit Ref	Description	Section
J14	**Host/peripheral expansion connector**	
	0.1" pitch 2x6 connector for host connection / expansion. This connector is primarily intended for connection and external host using I^{2}C and/or SPI bus, but is also suitable for general purpose I/O connections or peripherals.	
J15	**Host reset control**	
	This connector provides a reset input to the LPC546x8/540xx along with ground signals.	
J16, J17	**LCD and touch screen display connectors**	n/a
	These connector is dedicated for the LCD and touch screen. They are located under the LCD panel and should not be handled by the user unless instructed to do so by NXP.	

3. Getting Started

The LPCXpresso54608 and LPCXpresso54628 boards are pre-programmed with a demo application showcasing TouchGFX from Draupner Graphics. Connect a micro USB cable from connector J8 or J1 to a power source (computer or power supply) and the board will boot within a few seconds and run this demonstration.

The LPCXpresso54618 board, as supplied in the OM13094 CAN-FD kit, is pre-programmed with a CAN loopback example. For more information on this example please refer to Section 9.

The LPCXpresso54018 board is pre-programmed with a demo application using emWIN graphics. Connect a micro USB cable from connector J8 or J1 to a power source (computer or power supply) and the board will boot within a few seconds and then run this demonstration. The LPCXpresso54018 demo is the "touch and draw" emWIN example from the SDK.

The rest of this section describes how to download other sample programs using mass storage boot and how to start code development with the board, assuming the on-board Link2 Debug Probe will be used. Note that the binary code for the out of box demo (along with other demo code) is available under the Downloads tab on the OM13092 and OM13098 board sites at http://nxp.com/demoboard/om13092 and http://nxp.com/demoboard/om13098 respectively. For further information on TouchGFX please visit the Draupner Graphics website at http://touchgfx.com/en/nxp-semiconductors/

This section also describes how to download pre-built binary images to the LPCXpresso546x8 boards using its mass storage device capability, and how to set up the board to start debugging sessions using popular development tools. Note that the mass storage boot mode does not program off-chip flash devices (such as the quad SPI flash on the board), so some application examples (such as the TouchGFX demos) will require a different utility (J-flasher lite from SEGGER, MCUXpresso IDE's flash utility or Flash Magic). The demo binary packages mentioned earlier include instructions on using these tools.

3.1 Loading applications using USB mass storage boot (MSC class) mode (LPCXpresso546xx boards only)

The LPC546xx device family supports mass storage boot mode, enabling "drag and drop" programming of on-chip flash. Note that this method requires a binary file, which can be generated from any toolchain supporting the LPC546xx family. Refer to the documentation for your tools to determine how to generate a binary file. Note that this method cannot be used to load binary files that require data to be programmed into SPI/SPIFI flash.

Mass storage mode boot is possible via the High Speed (USB1) or Full Speed (USB0) ports. The High Speed port will provide significantly shorter programming times for larger binary files.

To program the LPC546x8 in mass storage mode follow these steps:

1. If using the full-speed port:

    a. Connect host computer to J3. Press and hold the ISP1 button while pressing and the releasing the Reset button.

    b. If using the high-speed port: connect host computer to J2. Press and hold the ISP2 button while pressing and the releasing the Reset button.

2. The host computer should show a mass storage device called CRP_DISABLD.

3. Open the MSC device and delete the file firmware.bin. Note that afterwards, even if the flash is in principle empty, the PC will read it as fully occupied memory.

4. Drag and drop the new binary file to the drive. Note that this new file must have the name firmware.bin, otherwise the file seems to be transferred and program but in fact it is not.

5. Reset the board. The application will now run.

3.2 Starting a debug session using the on-board (Link2) Debug Probe

By default, the LPCXpresso546x8/540xx is configured to use the on-board Debug Probe (Link2) to debug the on-board target (LPC546x8/540xx), using the CMSIS-DAP debug protocol pre-programmed into the Link2 Flash memory. The MCUXpresso IDE or other development tools that support the CMSIS-DAP protocol can be used in the default configuration. Check with your toolchain vendor for availability of specific device support packs for the LPC54xxx series devices.

Note that when using the MCUXpresso IDE, the on-board Link2 can also be booted in DFU mode by installing a jumper on JP5; if this is done then the IDE will download CMSIS-DAP to the probe as needed. Using DFU boot mode will ensure that the most up-to-date / compatible firmware image is used with the MCUXpresso IDE. Note that spare jumpers are provided in the board packaging.

NOTE: if the Debug Probe is set up to boot in DFU mode, the USB bridge functions (virtual comm port) and Debug Probe features will not be available if the board is not first initialized by the MCUXpresso IDE.

For further information and tutorial videos please visit the Getting Started tab on the landing page for the board being used (http://nxp.com/demoboard/om13092,

http://nxp.com/demoboard/om13098, http://nxp.com/demoboard/om13094 or http://nxp.com/demoboard/om40003)

3.2.1 Installation steps for use with MCUXpresso IDE

1. Download and install the MCUXpresso IDE.

2. Configure and download an SDK package (with the MCUXpresso IDE tool chain option selected) from the MCUXpresso SDK Builder utility (http://mcuxpresso.nxp.com).

3. Recommended: Install JP5 to force the Link2 Debug Probe to boot in DFU mode (see notes above).

4. Ensure jumper JP2 is fitted in position 1-2 (local target powered), and JP1 is not installed (target SWD enabled). These are the default positions set during board manufacture.

5. Connect the LPCXpresso546x8/540xx board to the USB port of your host computer, connecting a micro USB cable to connector J8 ("USB Debug-Link"). The board will boot and run the pre-installed demo.

6. Allow about 10 seconds for the LPCXpresso546x8/540xx devices to enumerate for the first time; the device will appear as "LPC Device".

7. If the first attempt to debug a project fails in the IDE, cancel the debug session and repower the board. On some machines the drivers take longer to enumerate for the first time, so these steps should correct this issue.

The board is now ready to be used with the MCUXpresso SDK examples for LPCXpresso546x8 or LPCXpresso54018.

When the board is used for the first time, it is recommended to force the LPC546x8 target into a known state by performing an ISP boot before attempting to run your first example code. This can be achieved by pressing and holding down one of the ISP buttons while pressing and releasing the reset button.

3.2.2 Installation steps to use Keil and IAR tools

1. Download and install LPCScrypt or the Windows drivers for LPCXpresso boards (http://www.nxp.com/lpcutilities). This will install required drivers for the board.

2. Ensure JP5 is open to force the Link2 Debug Probe to boot from internal flash.

3. Ensure jumper JP2 is fitted in position 1-2 (local target powered), and JP1 is not installed (target SWD enabled). These are the default positions set during board manufacture.

4. Connect the LPCXpresso546x8/540xx board to the USB port of your host computer, connecting a micro USB cable to connector J8 ("USB Debug-Link"). The board will boot and run the pre-installed demo.

5. Allow about 10 seconds for the LPCXpresso546x8/540xx devices to enumerate for the first time. It is not necessary to check the Hardware Manager, however if this is done there will be five devices; four under Human Interface Devices (CMSIS-DAP,

LPC-SIO, two HID Compliant Devices, and a USB Input Device) and one under Ports (LPC-LinkII Ucom.)

6. Run either the "Program LPC-Link2 with CMSIS-DAP" or "Program LPC-Link2 with SEGGER J-link" script provided in your LPCScrypt installation, and follow the on-screen instructions. These scripts can be seen in the Windows Start menu for the LPCScrypt installation.

7. After the script has run, remove JP5 and power cycle the board (note that resetting the board does not reset the Link2, so power cycling is required).

8. Your board is now ready to use with your 3rd party tool. Follow the instructions for those tools for using a CMSIS-DAP probe.

When the board is used for the first time, it is recommended to force the LPC546x8/540xx target into known state by performing an ISP boot before attempting to run your first example code. This can be achieved by pressing and holding down one of the ISP buttons while pressing and releasing the reset button.

3.3 Starting a debug session using an external Debug Probe

Code running on the LPC546x8/540xx target can be debugged using an external Debug Probe that conforms to the standard ARM debug connector. To use an external Debug Probe connect the probe to the SWD connector (P1) and connect power via the micro USB connector J1.

Note: The Debug link connector J8 must be left unconnected so that the Link2 Debug Probe is left unpowered and does not contend with the SWD interface signals from the external Debug Probe.

4. Link2 debug features

This section describes the features provided by the on-board Link2 Debug Probe. See the Getting Started section for more information on how to program the device.

The Link2 Debug Probe is implemented using an LPC432x MCU (circuit reference U21), which provides a high speed USB port interface to the host computer that runs the development tools. This device is not intended for developer use, and should only be used with approved firmware images from NXP. The Link2 on-chip flash memory is factory programmed with a firmware image that supports CMSIS-DAP debug protocol, but also includes several other USB end point functions:

- Virtual COM (VCOM) port: a serial device that can be used with any host computer application design for serial port communication (e.g. Teraterm, puTTY, etc.) Set the terminal program for baud rate to 115200, no parity, 8 bit data, 1 stop bit, no flow control.

- SWO trace end point: this virtual device is used by MCUXpresso to retrieve SWO trace data. See the MCUXpresso IDE documentation for more information.

- Power measurement probe: this virtual device is used by MCUXpresso to retrieve power measurement data from boards that support this feature (e.g. LPCXpresso V3 boards, including the LPCXpresso546x8/540xx.) See Section 7 for more details on this feature.

- I2S/SPI bridges: bridge device from I2C and SPI ports of the LPC546x8/540xx target.

All of these devices are independent of each other and of the CMSIS-DAP debug device that is enumerated when the board is connected to a host computer; for example, the VCOM port can be used if the board is running an application when no debugger is running.

In order to correctly install and use the Link2 device on the LPCXpresso546x8/540xx (required for any debugging purpose) for Windows host computers, install the drivers first. These drivers will automatically be installed when MCUXpresso IDE has already been installed. If these IDEs are not being used, it is recommended LPCScrypt be installed as this also includes the required drivers. All these tools and utilities are available for free download at www.nxp.com.

The CMSIS-DAP firmware image installed at the factory (and by LPCScrypt) will uniquely identify itself to the host computer so that more than one board can be connected to that host computer at any time. Some toolchains cannot discern between multiple debug devices; refer to your toolchain documentation for more information (note the MCUXpresso does support multiple LPCXpresso board targets.) It is strongly recommended that LPCScrypt be used to update the Debug Probe firmware in order to ensure the latest version is being used (see Section 4.2.)

Note: The Link2 only boots when the board is power cycled; the reset button on the board does not reset the Link2.

When using MCUXpresso IDE, the Link2 can be automatically booted with the latest / most appropriate firmware for that IDE version by installing JP5 before powering up the board. This is the recommended approach for the MCUXpresso IDE. Note that if JP5 is installed the VCOM port (and other devices mentioned above) device will not appear until the IDE boots the Debug Probe. The Debug Probe is booted once a debug session is started (that is, the IDE attempts to download code to the target).

4.1 What the Link2 boot LED indicates

LED D10 is the Link2 MCU BOOT0_LED indicator. This LED reflects the state of Link2 MCU pin P1_1. When the boot process fails, D1 will toggle at a 1 Hz rate for 60 seconds. After 60 seconds, the Link2 MCU is reset.

4.2 Programming the Link2 firmware

As mentioned earlier in this section, it is not normally necessary to program the Link2 firmware. However, this can easily be accomplished using the supporting utility, LPCScrypt.

To program the Link2 Flash the Link2 device (LPC432x) must be in DFU mode. If the Link2 already has a valid image in the flash, it will need to be forced into DFU mode by placing a jumper shunt on JP5, and power cycling (disconnecting then reconnecting power via J8.) Link2 MCU programming is performed using the LPCScrypt utility (see http://www.nxp.com/lpcutilities). Instructions for using the tool are located at the same web page.

4.3 VCOM port

The identifier of the VCOM port will vary between boards and hosts. To determine the COM port, open the Windows operating system Device Manager. This can be achieved by going to the Windows operating system Start menu and typing "Device Manager" in

the search bar. In the device manager look under "Ports"; the LPC-LinkII UCom Port device and its name should be visible. NOTE: this VCOM port will only appear if (a) the Debug Probe has been programmed with the CMSIS-DAP firmware and the Debug Probe DFU link is removed at power up, or (b) if the Debug Probe has been configured for DFU boot (JP5 installed) at power up and MCUXpresso IDE has booted it (by starting a debug session.)
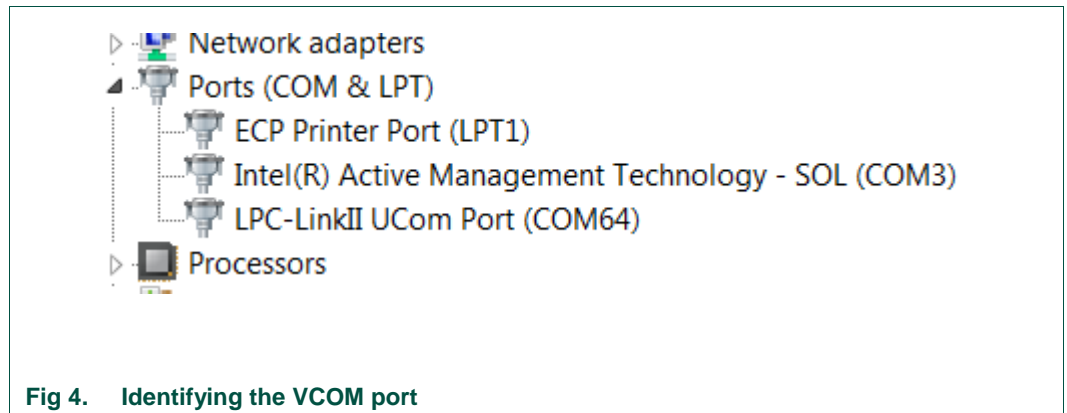

Fig 4.   Identifying the VCOM port

If the J-Link firmware image has been programmed into the Debug Probe and DFU boot mode is not being used, then a VCOM device called Jlink CDC UART port will appear instead of the LPC-LinkII UCom port.

4.4 Configuring the LPCXpresso546x8/540xx to debug an external target

The LPCXpresso546x8/540xx board's Link2 Debug Probe may be used to debug an off-board target MCU. The on-board Link2 Debug Probe is capable of debugging target MCU's with a VDDIO range of 1.6V to 3.6V. To keep the on-board target LPC546x8/540xx MCU from interfering with the SWD interface, JP1 must be fitted. The Link2 Debug Probe SWD should be connected by a ribbon cable between the P1 connector to the off-board target MCU SWD interface. Power the LPCXpresso546x8/540xx board from the Link USB connector J8, and fit jumper JP2 across pins 2 - 3 (External Target).

5. Board power connections & measurement

The LPCXpresso546x8/540xx board requires +5V input to power the on-board voltage regulators which in turn power the Link2 Debug Probe and other +3.3V circuits, the LPC546x8/540xx target and other +1.8V circuits, and the Arduino +5V and +3.3V power rails. When the main external power source is from the Link2 side USB micro B-type connector (J8), both the Link side and LPC546x8/540xx Target sections of the board are powered. When the main external power is from the power only USB micro B-type connector (J1), or one of more of the USB device ports (USB0 at J3 or USB1 at J2), only the LPC546x8/540xx target and supporting devices and peripherals are powered.

5.1 LPCXpresso546x8/540xx current measurement

The LPC546x8/540xx current can be measured by measuring the voltage across a sense resistor in series with the supply, a current meter or using the on board current measurement circuit. Each of these methods will be described in subsections below. There is no current monitoring of the Link2 section circuits on the board. The Target side power going to LEDs and support ICs is not monitored by the current measurement

circuit. The LPC546x8/540xx BGA package has the core and IO power both sourced from the same VDD pins.

When a shield board is attached, attempting to measure the lowest possible power the LPC546x8/540xx IO pins must be configured according to how the software has configured the shield board to ensure there is no extra current from the LPC546x8/540xx IO ports that have external pull-up or pull-down resistors enabled.

There are several leakage paths through the various devices on the LPCXpresso546x8/540xx board, so this board should not be used to measure the lowest possible leakage current achievable in a target application.

5.1.1 LPC546x8/540xx Vsense resistor current measurement

The voltage across a pair of 1Ω resistors in series with the target LPC546x8/540xx VDD can be manually measured at JP4 across pins 1 and 4 on the PCB. For a higher range of current a jumper should be installed between pins 3 and 4 of JP4 to short out on of these resistors. The voltmeter positive probe is applied to JP4 pin 1 (see silkscreen labelling) and pin 4. Use Ohm's law to calculate the current (LPC546x8/540xx current = measured voltage / 2 Ω). As an example, if the measured voltage is 20 mV, then 20e-3 / 2 Ω = 10 mA.

Note: The input current to the MAX9634 used in the on-board current measurement will be included in the voltage measured across this resistor.

5.1.2 LPC546x8/540xx VDD current measurement using a current meter

A current meter may be inserted at JP4 between pins 5 and 6 to measure the LPC546x8/540xx VDD input current. Note that a jumper needs to be installed in the 5-6 position when an ammeter is not present so power can reach the LPC546x8/540xx target.

5.1.3 LPC546x8/540xx VDD current measurement

The LPCXpresso546x8/540xx board has an on-board current measurement circuit consisting of a MAX9634T (U18) current monitor chip and a 12-bit ADC (ADC122S021, U23) with a 12-bit sample at 50k to 200ksps. The on-board MAX9634T current monitor measures the voltage across the LPC546x8/540xx VDD Vsense resistors; either 2 Ω or 1 Ω if JP4 jumper 3-4 is installed. The MAX9634 multiplies the sense voltage by 25 to provide a voltage range suitable for the ADC to measure. A 2-input analog mux selects between the LPC546x8/540xx current monitor and the output off a MAX9634T current monitor chip on an expansion board (with compatible current measurement circuit on-board). The current measurement circuit is controlled by the Link2 processor and is not user programmable. Power measurement utilities to use this feature are available in LPCXpresso and IDE MCUXpresso IDE installations.

Due to input offset voltage variations in the MAX9634, the current measurement circuit is not recommended for measuring current below 150 uA. See Fig 5 as a guideline for measurement error versus measured current.
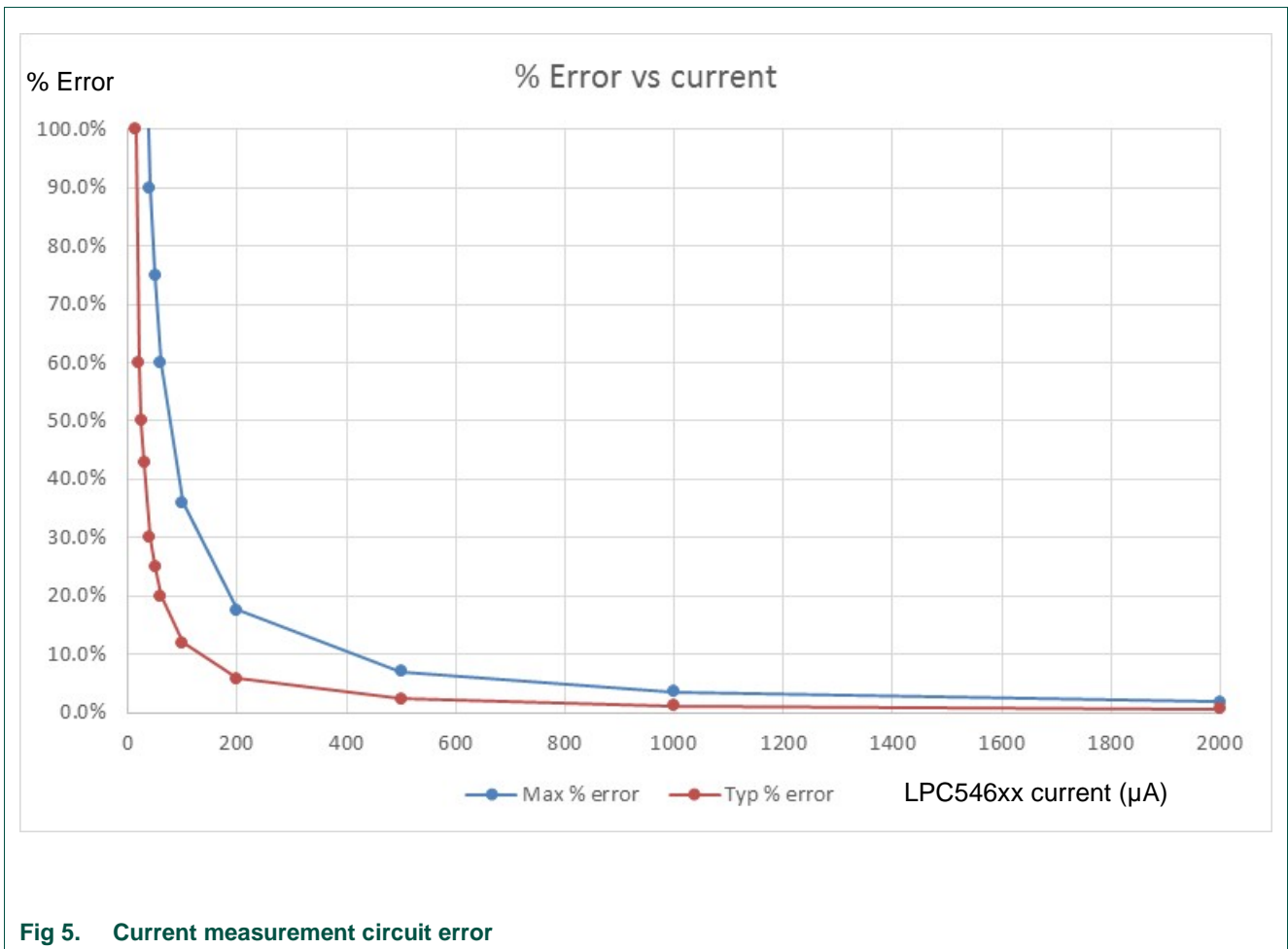
Fig 5.   Current measurement circuit error

5.1.4 Shield board current measurement

To use the on-board current measurement circuitry, any expansion board must match the functionality of the LPCXpresso546x8/540xx. Refer to the board schematics for more information.

6. Board peripheral connections

6.1 LPC546x8/540xx USART ports and VCOM support

The LPCXpresso546x8/540xx board provides access to USART connections from Flexcomm4 at the expansion connector J13.

On LPC546x8/540xx the USART from Flexcomm0 supports ISP UART mode booting, and is used by default for semi-hosting debug messages. This USART is connected to a serial port on the Link2 Debug Probe, which provides a bridging function to USB to a virtual com port (or "VCOM" port) on a host computer connected to the debug link connector, J8. On revision C boards TXD and RXD from this USART are also available at header P4, for an external serial connection to be made. When using the P4

connector, JP6 must be installed; note that when JP6 is installed the VCOM port functionality is disabled.

The factory default CMSIS-DAP Link2 image includes UART bridge functionality (VCOM support), and this firmware is also available with the LPCScrypt utility, available at http://www.nxp.com/lpcutilities. The J-Link firmware image available from LPCScrypt (version 1.9 onwards) or directly from SEGGER also has this feature (also see Section 4.3.)

6.1.1 P3 USART header

The P4 header is designed to be used with an external serial to USB or other 3.3V serial device. P4 is only present on Revision C (and later) boards. Table 3 shows the pin out information. below. Note that the connections at P4 are made directly to the LPC4608 device, so care should be taken to avoid ESD discharge and voltages outside the ratings of the device I/O pins.

Table 3.   P4 connections

Pin	Function
1	Ground
2	RXD (from LPC546x8/540xx)
3	TXD (from LPC546x8/540xx)

6.2 USB ports

The LPCXpresso546x8/540xx incorporates connectors for the two USB ports of the LPC546x8/540xx device – a full speed (USB0) and a high speed port (USB1). Both of these USB ports support device and host functionality, but only one can be used as a host port at any given time. Jumper settings must be set appropriately for host mode selection, as described in this section.

The MCUXpresso SDK for LPC546x8/540xx includes USB stacks for both host and device, with several supporting application examples.

6.2.1 USB full-speed port (USB0)

The LPC546x8/540xx Full Speed (FS) USB port is connected to micro AB USB connector J3. This section describes functionality support by this board and associated jumper settings for this board.

6.2.1.1 Mass Storage Boot (MSC)

The FS USB port supports mass storage boot mode (also known as MSC). When the board is booted in this mode using the FS port, the device will enumerate in a host computer (when that computer is connected using a standard micro B USB cable via J3) as a mass storage class device called CRP_DISABLED. To force the board to boot into this mode press and hold down the ISP1 button while pressing and releasing the reset button (or power cycling the board). See Section 3.1 for further information on how to use this mode.

6.2.1.2 Device mode

The FS port supports USB device mode operation; no jumper settings are needed to configure this mode. When using the port in device mode, a standard micro USB cable can be used.

6.2.1.3 Host mode

The FS port supports USB device mode operation but, due to sharing of some signals and power supply circuitry, requires the following settings:

- Connect external +5V power via USB connector J1. This is required for the FS USB port to be able to power the devices connected to it.
- Install JP10
- For each of JP9, JP11, JP12, JP13 install a jumper in position 2-3.
- Use of a micro AB adaptor to connect an external USB device may be required, as most USB device have either a type A or micro B connector.

Note: The Ethernet and HS USB Host functions are disabled when the board is configured for FS USB Host operation. Also, note that Expansion connector port PIO4-7 is no longer available when using this mode.

6.2.2 USB High Speed port (USB1)

The LPC546x8/540xx High Speed (HS) USB port is connected to micro AB USB connector J2. This section describes functionality support by this board and associated jumper settings for this board.

6.2.2.1 Mass Storage Boot (MSC)

The HS USB port supports mass storage boot mode (also known as MSC). When the board is booted in this mode using the HS port, the device will enumerate in a host computer (when that computer is connected using a standard micro B USB cable via J2) as a mass storage class device called CRP_DISABLED. To force the board to boot into this mode press and hold down the ISP2 button while pressing and releasing the reset button (or power cycling the board). See Section 3.1 for further information on how to use this mode.

6.2.2.2 Host mode

The HS port supports USB device mode operation but, due to sharing of some signals and power supply circuitry with the FS port, requires the following settings:

- Connect external +5V power via USB connector J1. This is required for the HS USB port to be able to power the devices connected to it.
- Ensure JP10 is not installed (default).
- Install a jumper at JP9 in position 1-2 (default).
- Use of a micro AB adaptor to connect an external USB device may be required, as most USB device have either a type A or micro B connector.

(Note: Although not required for the HS USB port operation, it is recommended to check JP11, JP12 and JP13 have a jumper in position 1-2 to enable Ethernet operation and availability of the I/O at the Expansion connector D8.)

6.3 Host Expansion Header (J14)

This header provides connectivity from the LPC546x8/540xx target to a remote Host, peripherals or other devices. Table 4 shows the connections.

Table 4. Host Expansion Header signals

Pin	J14 signal	Flexcomm # / Port
1	GPIO/SPI-SSEL in/out	3 / PIO0-1
2	GPIO/SPI-MOSI in/out	3 / PIO0-3
3	GPIO/SPI-MISO out/in	3 / PIO0-2
4	GPIO/SPI-SCK in/out	3 / PIO0-0
5	GND	
6	VDD (3.3V)	
7	GPIO/INT (out/in)	PIO0-5 (default) or PIO0-4, set by JS32
8	GPIO/RESET (out)	PIO3-13
9	SCL	1 / PIO0-14 Rev C and earlier
		1 / PIO0-13 Rev D and later
10	SDA	1 / PIO0-13 Rev C and earlier
		1 / PIO0-14 Rev D and later
11	GND	
12	VDD (3.3V)	

This connector shares SPI interface signal connections with the SPI bridge function (i.e. connection from the LPC546x8/540xx to the Link2 Debug Probe); only one of these functions may be used at once. By default jumper JP6 is open, connection the Flexcomm 3 SPI port to the Link2 device; install JP6 if using the SPI port on J14.

By default, solder jumpers JS15 and JS16 connect the LPC546x8/540xx Flexcomm 1 I²C port to the Link2 device; these may be removed if they will interfere with other devices interfaced to J14. Note there are 2.2KΩ pull-up resistors to 3.3V for these SDA and SCL connections.

6.4 PMod Slave Expansion Header

This header provides connectivity from the LPC546x8/540xx target to PMod peripherals or other devices. Table 5 shows the connections.

Table 5. PMod Slave Expansion Header signals

Pin	PMod standard connection	J11 signal	Flexcomm # / Port
1	GPIO/SPI-SSEL out / UART CTS in	GPIO/SPI-SSEL in/out	9 / PIO4-6 (default) or PIO3-30, set by JS27
2	GPIO/SPI-MOSI out / UART TXD out	GPIO/SPI-MOSI in/out	9 / PIO3-21
3	GPIO/SPI-MISO in / UART RXD in	GPIO/SPI-MISO out/in	9 / PIO3-22
4	GPIO/SPI-SCK out / UART RTS out	GPIO/SPI-SCK in/out	9 / PIO3-20
5	GND	GND	

Pin	PMod standard connection	J11 signal	Flexcomm # / Port
6	VCC (3.3V)	VDD (3.3V)	
7	GPIO/INT in	GPIO/INT (out/in)	PIO2-0
8	GPIO/RESET (out)	GPIO/RESET (out)	PIO3-11
9	GPIO/SCL	SCL	2 / PIO3-24
10	GPIO/SDA	SDA	2 / PIO3-23
11	GND	GND	
12	VCC (3.3V)	VDD (3.3V)	

Note that the I²C and SPI ports on J11 are also connected to the Expansion connector J9, and the INT and RESET connections to J12.

6.5 Expansion connectors (including Arduino connectivity)

The LPCXpresso546x8/540xx board includes four expansion connectors (J9, J10, J12 and J13) incorporate an Arduino Uno revision 3 footprint in their inner rows. These connectors provide access to the CAN interfaces, additional digital microphone support signals, I²S, I²C, USART, SPI and GPIO/INT/PWM connections. Note that several of the signals available at these connectors are shared with other connectors or board functions, so may not be usable if those other functions are being used.

Table 6 shows the connector circuit references and purposes.

Table 6. Expansion connectors

Reference	Description
J9	The odd number pins are compatible with Arduino Uno rev3 Digital 15:8, AREF, SDA & SCL connector. The even numbered pins are used for external access and expansion of LPC546x8/540xx signals not used by the Arduino Uno rev3 compatible interface.
J13	The odd numbered pins 1 – 15 are compatible with Arduino Uno rev3 Digital 7:0 connector. The even numbered pins, and odd numbered pins 17 and 19, are used for external access and expansion of LPC546x8/540xx signals not used by the Arduino Uno rev3 compatible interface.
J10	The even numbered pins 6 – 20 are compatible with the Arduino Uno rev3 Power connector standard. On Revision E and later boards, pins 13 and 15 are used to provide access to Smart Card 0 port (board modifications required to use this feature – see Section 6.7.
J12	The even numbered pins 2 – 12 are compatible with Arduino Uno rev3 Analog connector. The odd numbered pins are used for external access and expansion of LPC546x8/540xx signals not used by the Arduino Uno rev3 compatible interface.

Below shows the sharing of signals between the expansion connectors and other connectors or circuit functions. Refer to the board schematics for more details.

Table 7.   **Expansion Connector signal sharing**

Function	Connector	Shared with
I^{2}C	J9 pin 14 & 15	Accelerometer, PMod Slave Connector, Audio Codec, LCD touch panel controller
SPI	J9 pins 9, 11,13,15	PMod Slave Connector
GPIO D8	J9 pin 19	Jumper (JP13) selection for USB0 FS host port. Only one of USB0 Host mode or GPIO function on this port may be used.
GPIO	J9 pin 6	User LED 3
GPIO	J9 pin 8	User LED 2
I2S	J9 pins 12, 14, 16, 18, 20	Audio codec. To use an I^{2}S device connected to these pins solder jumpers JS6 through JS9 and JS11 through JS13 may need to be removed.

6.6 Ethernet port

The LPC546x8/540xx incorporates an on-board 10/100 Mbps Ethernet MAC. The LPCXpresso546x8/540xx board includes an on-board LAN8720A-CP PHY and RJ45 jack with integrated link status LEDs.

The MCUXpresso SDK for LPCXpresso546x8/540xx includes drivers, LwIP stack and several examples to show how the Ethernet function can be used.

6.7 Smart Card (revision E and onwards only)

Smart Card 0 interface connections are routed to J10 pins 13 (P0_20) and 15 (P0_21) via solder jumpers JS38 and JS37. Since these signals are also used for the EMC (SDRAM) address lines A2 and A3, these solder jumpers are left open by default. If these solder jumpers are closed, then care should be taken not to access the SDRAM and thus cause bus contention.

7. On-board memory

7.1 SDRAM

The board includes a Winbond 128Mb SDRAM (Micron SDRAM used on revision C and earlier, and some rev D builds), connected to the 16-bit external memory interface available on the LPC546x8/540xx BGA180 part. The MCUXpresso SDK includes drivers that have optimized external memory settings for use of this memory.

7.2 Quad SPI flash

The board includes a 128Mb Winbond W25Q128JVFIM Quad-SPI flash (Micron MX25L12835FM2I-10G used on revision C and earlier, and some rev D builds). The MCUXpresso SDK includes drivers for use of this memory as a data storage/retrieval device. Note that drivers for this device may be required for the development tool chain being used (these are included in the MCUXpresso IDE). Refer to your tool chain vendor for more information. SEGGER J-link includes drivers for both of these devices.

UM11035

User manual   **Rev. 2.0 — 12th January 2018**   **25 of 34**

Note that the LPCXpresso546x8/540xx board is also designed to accommodate a Macronix MX25R6435FM2IL0 quad SPI flash as a reference for customers considering a 1.8V based design.

8. Other board features

This section describes board functions not mentioned elsewhere in this document.

8.1 LCD with capacitive touch

The LPCXpresso546x8/540xx includes a Rocktech RK043FN02H-CT color LCD display panel with a resolution of 272x480 pixels and a capacitive touch controller. The LCD and capacitive touch controller interface to the main board via flex cables routed beneath the LCD panel. The LCD panel is secured using multiple adhesive pads. It is not recommended to remove the panel because these adhesive pads will need to be replaced afterwards to ensure the mechanical integrity of the flex cable connections.

Note: The LCD data lines are shared with the PDM0 interface of the LPC546x8/540xx device (the LCD is enabled by default), and solder jumpers need to be modified to change between these configurations (see section 8.5 for more details.)

On-board voltage pump circuitry is provided to drive the LCD backlight using an AP5724WG device. This is enabled using PIO3-31.

The MCUXpresso SDK includes drivers and example applications for the LCD and capacitive touch controller.

8.2 Audio codec

The LPCXpresso546x8/540xx board incorporates a Cirrus Logic (Wolfson) WM8904 audio codec, powered via a dedicated regulator. This codec has both I²C (for control) and I²S (for data) interfaces.

The I²C interface of the codec is routed to Flexcomm 2 off the LPC546x8/540xx, the same connection as used for the other I²C devices on the board; the Codec has an address of 0b0011010.

The Flexcomm 6 and Flexcomm 7 interfaces of the LPC546x8/540xx are connected respectively to the Codec for data sent to / received from the Codec. These I²S connections from the LPC546x8/540xx are also connected to the Expansion connectors (see Section 6.5 for more information); solder jumpers may be removed to disconnect the on-board codec if an external I²S device is to be used instead.

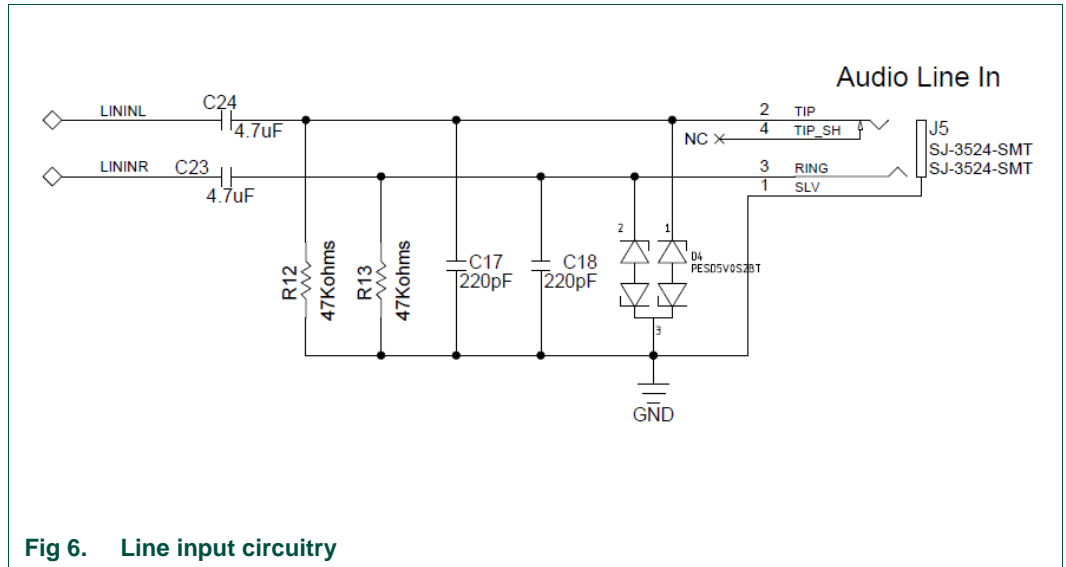The line input to the codec is routed to a 3.5mm stereo jack socket (J5) through the circuit shown in Fig 6.

UM11035

© NXP B.V. 2017, 2018. All rights reserved.

User manual **Rev. 2.0 — 12th January 2018** **26 of 34**

Fig 6.   Line input circuitry

A second 3.5 mm stereo jack socket (J6) provides a headphone / line out from the codec, via the circuit shown in .
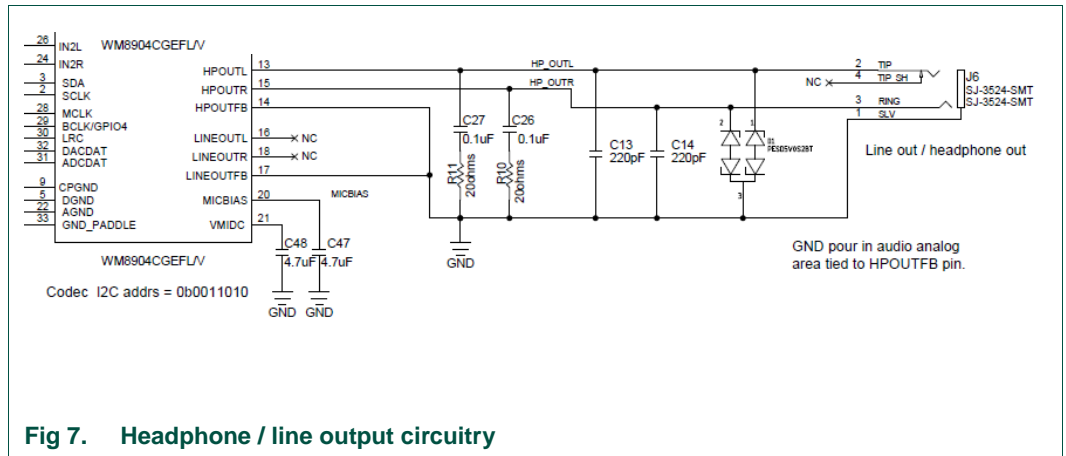

Fig 7.   Headphone / line output circuitry

The MCUXpresso SDK includes drivers and example code for the audio codec.

8.3 SD card

The full size SD card (J7) includes in the LPCXpresso546x8/540xx board provides a 4-bit SDIO interface to support memory cards, plug-in WiFi modules, etc. Power enable to the socket is provided via PIO2-5, with LED D7 providing a visual indication when power is applied.

The MCUXpresso SDK includes drivers and examples of using this interface with SD memory cards.

8.4 Accelerometer

The board includes an NXP MMA8652FCR1 accelerometer, interfaced to Flexcomm 2 (ports PIO3-23/PIO3-24) of the LPC546x8/540xx. The accelerometer has an I^2C address of 0b0011101.

8.5 Digital Microphone

The LPC546x8/540xx device incorporates dual digital microphone interfaces. A low power Knowles SPH0641LM4H digital microphone is incorporated on the board, interfaced to the PDM1 microphone interface. Note that the Knowles digital microphone is designed to be mounted on the underside of a board, with audio passing through a hole in the board.

Note that the PDM0 interface of the LPC546x8/540xx is also available for interfacing to an external microphone via the Expansion connectors, but due to signal sharing with this function and LCD data lines 14 and 15, solder jumpers JS33 and JS36 need to be moved to the 2-3 position to enable this function (thus disabling correct LCD operation).

The MCUXpresso SDK includes drivers and demonstration code for capturing data from the on-board digital microphone and an external microphone of the same type (the latter requires board modification).

8.6 USER button

This button can be used to generate an interrupt by pulling down the PIO1-1 of the LPC546x8/540xx to ground.

9. Dual CAN transceiver Shield (OM13094 CAN-FD kit or OM13099)

The OM13094 CAN-FD kit includes a dual CAN/CAN-FD transceiver shield board, which is also available as a standalone item (product number OM13099). This board features the NXP TJA1059 dual transceiver, plus an RS232C transceiver. This shield board is also compatible with all the LPCXpresso546x8/540xx boards (although LPCXpresso54608 only supports classic CAN, not CAN-FD.)

9.1 Board layout and jumpers

Fig 8 shows the Shield board layout. The diagram highlights the connections used by the Shield board on the expansion connectors, and the location of the jumpers.

(1)

Fig 8.   **CAN-FD Shield Layout**
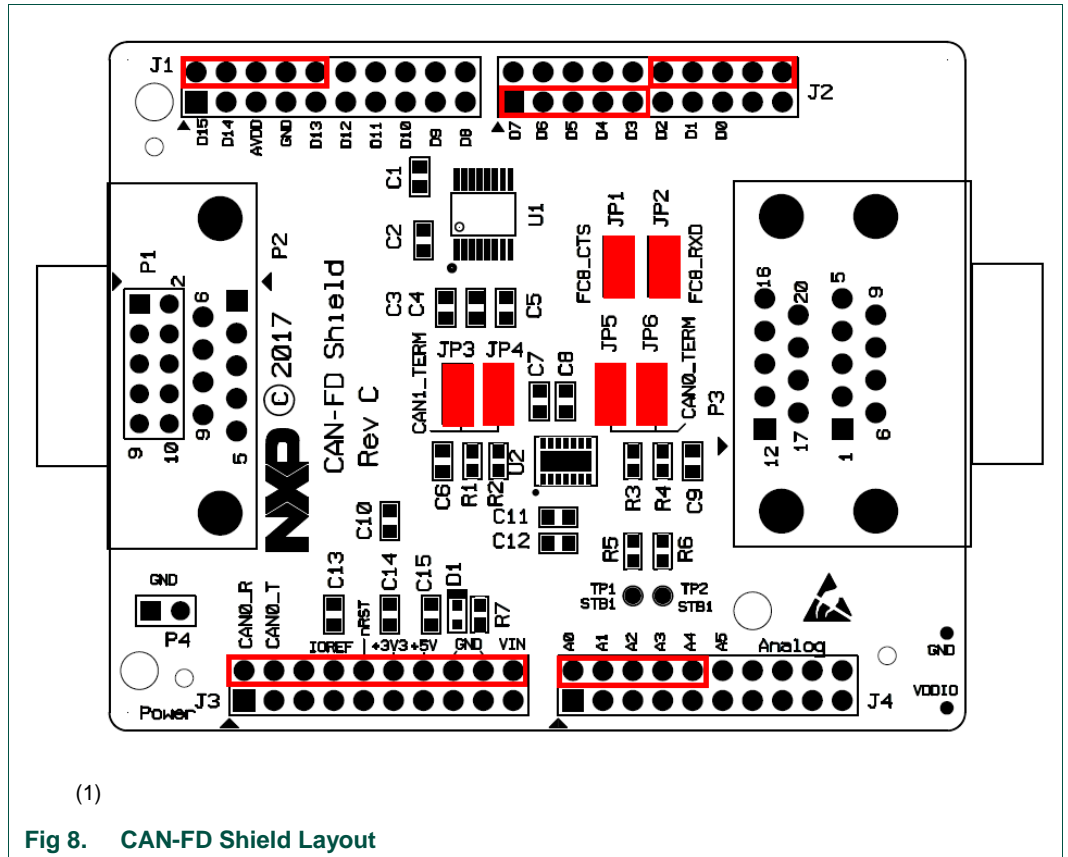
The CAN-FD Shield has a double-height DB9 connector installed by default (P3) for CAN/CAN-FD connections.

The Shield board layout includes provision for a DB9 connector to be installed (P2) but a standard 0.1" header (P1) is installed by default; P1 would need to be unsoldered in order for P2 to be fitted.

Note that header P4 is provided to allow convenient access to ground.

9.2 Jumper settings

Jumper settings are described in Table 8.

Table 8.   **CAN-FD Shield Jumper settings**

Circuit ref	Description	Default
JP1	RS232C transceiver CTS connection. When fitted, the RS232C transceiver CTSn signal is connected to expansion connector J4 pin 8 (Arduino A4). On the LPCXpresso546x8/540xx boards A4 is connected to pin PIO3_4; this pin is shared with the accelerometer interrupt output.	Installed
JP2	RS232C transceiver RXD connection. When fitted, the RS232C transceiver R1OUT signal is connected to expansion connector J2 pin 20. On the	Installed

Circuit ref	Description	Default
	LPCXpresso546x8/540xx boards A4 is connected to pin PIO2_17.	
JP3 & JP4	Install these jumpers to provide termination on CAN/CAN-FD connection 2.	Installed
JP5 & JP6	Install these jumpers to provide termination on CAN/CAN-FD connection 2.	Installed

9.3 Installation

To install the Shield board, carefully align the J1 pin 1 of the Shield board with J9 pin 1 of the LPCXpresso board, and J3 pin 1 of the Shield board with J10 pin 2 of the LPCXpresso board, ensuring that all leads from the Shield are lined up with the corresponding sockets of the LPCXpresso board. Apply even pressure to the Shield board edges until its leads are fully seat in the sockets. It should not require very much pressure to install the Shield board; if the required force seems high, stop and re-check the pins are aligned properly.

9.4 Out-of-the-box example code

The pre-installed demo code runs a simple loop back test between the two CAN ports, using CAN protocol. The test requires a female to female, straight through connection cable, connected between the two connectors on P3. As the test runs, user LEDs 1 and 2 will blink alternately as a series of data blocks are transferred over the CAN ports; once the test completes successfully, LEDs 1, 2 and 3 will light and stay on. The test can be repeated easily by pressing the reset button on the LPCXpresso54618 board.

If the test fails (as will happen if no cable is connected), LED1 and LED2 will blink briefly, then LED3 will light and stay on.

Once drivers have been installed for the board, a virtual UART connection can be used to see test progress. Refer to Section 3 for more information on driver installation and virtual com port use.

9.5 Other example code

The MCUXpresso SDK includes drivers (called mcan drivers) and examples using polling and interrupt methods.

CAN-FD drivers from Embedded Systems Academy, compatible with the MCUXpresso SDK, are available separately under the Downloads tab of the CAN-FD kit page and(http://nxp.com/demoboard/om13094) and also from the standalone shield board page (http://nxp.com/demoboard/om13099)

10. Legal information

10.1 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.

Evaluation products — This product is provided on an "as is" and "with all faults" basis for evaluation purposes only. NXP Semiconductors, its affiliates and their suppliers expressly disclaim all warranties, whether express, implied or statutory, including but not limited to the implied warranties of non-infringement, merchantability and fitness for a particular purpose. The entire risk as to the quality, or arising out of the use or performance, of this product remains with customer.

In no event shall NXP Semiconductors, its affiliates or their suppliers be liable to customer for any special, indirect, consequential, punitive or incidental damages (including without limitation damages for loss of business, business interruption, loss of use, loss of data or information, and the like) arising out the use of or inability to use the product, whether or not based on tort (including negligence), strict liability, breach of contract, breach of warranty or any other theory, even if advised of the possibility of such damages.

Notwithstanding any damages that customer might incur for any reason whatsoever (including without limitation, all damages referenced above and all direct or general damages), the entire liability of NXP Semiconductors, its affiliates and their suppliers and customer's exclusive remedy for all of the foregoing shall be limited to actual damages incurred by customer based on reasonable reliance up to the greater of the amount actually paid by customer for the product or five dollars (US$5.00). The foregoing limitations, exclusions and disclaimers shall apply to the maximum extent permitted by applicable law, even if any remedy fails of its essential purpose.

10.2 Trademarks

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

11. List of figures

UM11035

All information provided in this document is subject to legal disclaimers.

© NXP B.V. 2017, 2018. All rights reserved.

User manual

Rev. 2.0 — 12th January 2018

32 of 34

12. List of tables

13. Contents

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