UM11121

LPCXpresso51U68

Rev 1.1 — 27 April 2018 User manual

Document information

Info	Content
Keywords	LPCXpresso51U68, LPC51U68, OM40005
Abstract	LPCXpresso51U68 User Manual

Revision history

Rev	Date	Description
1.0	20180112	Initial version
1.1	20180427	First release version

Contact information

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

1. Introduction

The LPCXpresso™ family of boards provides a powerful and flexible development system for NXP's Cortex®-M family of MCUs. They can be used with a wide range of development tools, including the NXP's MCUXpresso IDE. The LPCXpresso51U68 board has been developed by NXP to enable evaluation of and prototyping with the LPC51U68 MCUs, and is based on the LPC51U68JBD64 version of the MCU.

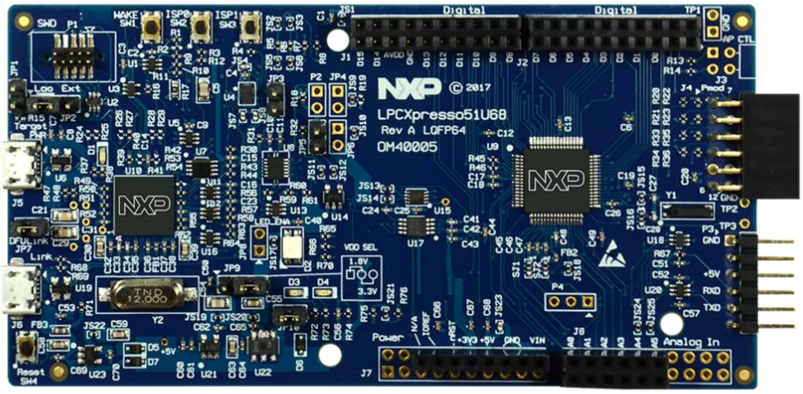

Fig 1. LPCXpresso51U68 Board

This document describes the LPC51U68 LPCXpresso LQFP board hardware. The following aspects of interfacing to the board are covered by this guide:

- Main board features.
- Setup for use with development tools.
- Supporting software drivers.
- Board interface connector pin out.
- Jumper settings.

2. Feature summary

The LPCXpresso51U68 board includes the following features:

- On-board, high-speed USB based, Link2 debug probe with ARM’s CMSIS-DAP and SEGGER J-Link protocol options.
- Link2 probe can be used with on-board LPC51U68 or external target.
- Support for external debug probes.
- Tri-color LED.
- Target Reset, ISP, and WAKE buttons.
- Expansion options based on Arduino UNO and PMod™, plus additional expansion port pins.
- On-board 1.8/3.3V or external power supply options.
- Built-in power consumption measurement.
- UART, I^{2}C and SPI port bridging from LPC51U68 target to USB via the on-board debug probe.
- FTDI UART connector.

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. The layout of the components on the LPCXpresso51U68 board is shown in Fig 2.

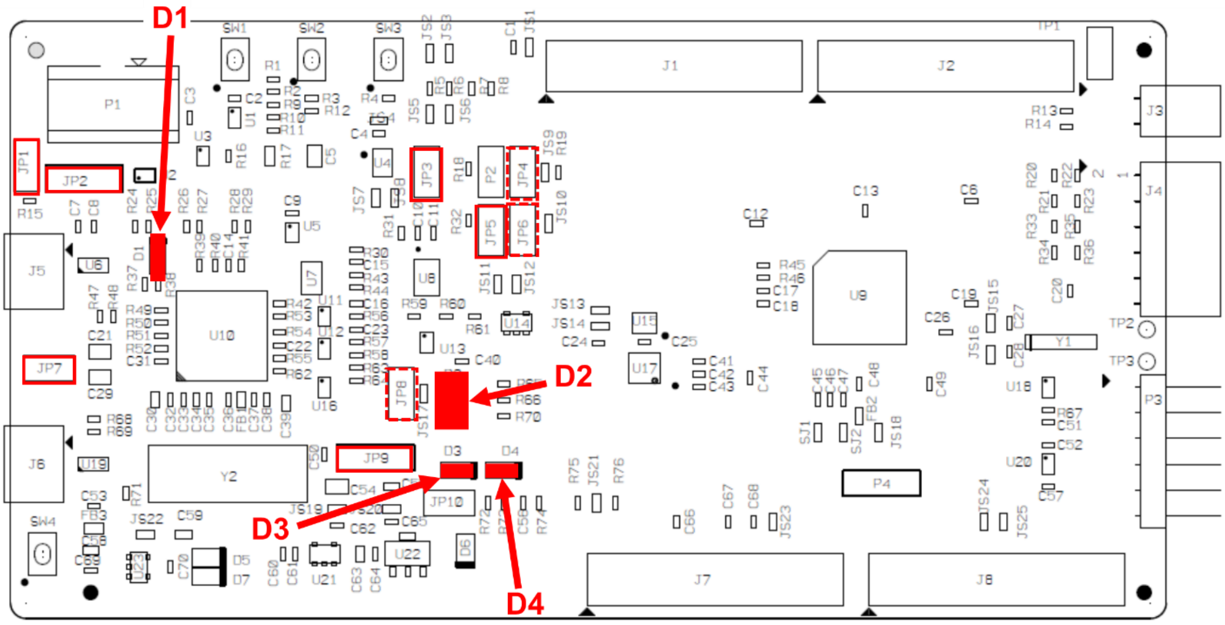

Fig 2. LPCXpresso51U68 layout

The function of each identified component is listed in Table 1.

Table 1. Table Board interface components

Circuit ref	Description	Ref section
D1	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.	7
D2	Tri-color LED driven by Target LPC51U68 MCU. JP8 must be shunted for +3.3V to be applied to D2 anode. The default shunt for JP8 is a 0Ω resistor installed at JS19.	7
D3	Target LPC51U68 Power LED.	7
D4	Target LPC51U68 Reset LED – LED is on anytime the Target RESETn is pulled low.	7
J1, J2, J7, J8	Expansion connectors, including Arduino Uno rev3 compatible connectivity.	8
J3	External processor control header. This connector provides access to the LPC51U68 control and ISP0 boot control to enable an external device control reset and entry into I2C/SPI boot mode.	n/a
J4	PMod™ (SPI / I^2C) Bridge connector. An external Application Processor (AP) or PMod™ peripheral may be connected to the LPC51U68 Target MCU SPI and I2C via this connector.	8

Circuit ref	Description	Ref section
J5	Target MCU Power / USB Device connector. Connect this micro USB B-type connector to a +5V power source when it is desired to power only the Target MCU, and leave the on-board Link2 debug probe unpowered.	6
J6	Link2 micro USB B-type connector. Powers both the Link2 side of the board and LPC51U68 Target side of the board. Power the board from this connector when using the on-board debug probe to debug the LPC51U68 Target MCU.	6
JP1	LPC51U68 Target SWD disable – 2-position jumper pins. 1) Jumper open (default) the LPC51U68 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. 2) When the jumper in installed, the LPC51U68 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.	6
JP2	Buffer power selection. Install jumper in position 1-2 during normal use. Install in position 2-3 when debugging an off-board target and using the LPCXpresso51U68 board to power that target via the SWD header (P1).	6
JP3	JP3 is used to isolate the Link2 debug probe (SPI bridge function) from the LPC51U68 target to prevent leakage current in power critical applications / current consumption analysis. JP3 needs to be fitted to use the SPI bridging function between the LPC51U68 and Link2. This jumper is not fitted by default.	n/a
JP4	JP4 (not installed by default) provides a method to disconnect all reset sourced to the LPC51U68 device except for the reset from the AP control port (J3). If this jumper is to be used, remove SJ9 first.	n/a
JP5	JP5 can be installed to reduce the voltage sense resistance (used by the on-board current measurement circuitry) from 8.24 ohms to 4.12 ohms.	5.1
JP6	A current meter may be installed across JP6 terminals to measure the LPC51U68 current consumption. By default JP6 is shunted by a 0Ω resistor installed at JS10; remove this shunt to measure current at JP6.	5.1.2
JP7	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. 2) Jumper shunted to force the Link2 to DFU boot mode. Use this setting to reprogram the Link2 internal flash with a new image or to use the MCUXpresso IDE to boot the probe.	6.1.3
JP8	Tri-color LED anode voltage enable. By default JP8 is shunted by a 0Ω resistor installed at JS17. To disable +3.3V to the tri-color led (D2) common anode, remove the 0Ω at JS17.	5.1
JP9	Power supply voltage selection – 3 position jumper pins. 1) Jumper 1–2: LPC51U68 is powered at 1.8V. 2) Jumper 2–3 (default): LPC51U68 is powered at 3.3V.	n/a
JP10	Target connector USB (J5) VBUS to LPC51U68 connection: 1) Installed (default): JP10 connects VBUS from Target USB connector to the LPC51U68. 2) Not installed: VBUS from J5 is unconnected.	n/a
P1	10-pin SWD connector – The SWD connector is used to debug the LPC51U68 Target from an external debug probe. The same SWD connector can also be used to connect the on-board Link2 debug probe to an off-board target MCU (for this JP1 must be shunted).	6
P2	LPC51U68 VDD current monitor Vsense measurement. The Vsense can be measured with a volt meter. Pin 1 (square pad) is positive and pin 2 is	5.1.1

<table>
<tr><th>Circuit ref</th><th>Description</th><th>Ref section</th></tr>
<tr><td></td><td>negative. LPC51U68 current is calculated by dividing the measured voltage at P2 by the Vsense resistance value of 4.12Ω.</td><td></td></tr>
<tr><td>P3</td><td>FTDI serial header. In addition to provide a serial output from LPC51U68, the Target side of the board can be powered from the FTDI header. The LPC51U68 supports serial ISP boot from the FTDI header.</td><td>4.1</td></tr>
<tr><td>P4</td><td>External ADC reference input. The pads of this header enable external VREF (negative and positive) ADC reference voltages to be injected. Pin 1 (indicated by the square pad) of the connector footprint can connected to the VREFP pin of the LPC51U68 by removing the zero ohm resistor at SJ from position 1-2 and bridging pads 2-3 instead. Similarly, Pin 3 of P4 can be connected to the VREFN of the LPC51U68 by removing the zero-ohm resistor at SJ1 from position 1-2 and bridging pads 2-3 instead.</td><td>LPC51U68 User Manual</td></tr>
<tr><td>SW1</td><td>LPC51U68 Target WAKE pushbutton. When pressed the WAKE switch will drive LPC51U68 P0_24 to a low level.</td><td>9.3</td></tr>
<tr><td>SW2, SW3</td><td>These switches can be used to force the LPC51U68 in to ISP boot modes:
<table>
<tr><th>Boot mode</th><th>ISP0</th><th>ISP1</th><th>Vbus (from J5)</th></tr>
<tr><td>I2C/SPI boot</td><td>Pressed</td><td>Pressed</td><td>X</td></tr>
<tr><td>UART boot</td><td>Pressed</td><td>Not pressed</td><td>0</td></tr>
<tr><td>USB Mass storage</td><td>Pressed</td><td>Not pressed</td><td>1</td></tr>
<tr><td>Boot from internal flash</td><td>Not pressed</td><td>Not pressed</td><td>X</td></tr>
</table>
After reset these pins may also be used to generate interrupts and/or GPIO inputs.</td><td>9.2</td></tr>
<tr><td>SW4</td><td>LPC51U68 Target Reset pushbutton.</td><td>9.1</td></tr>
<tr><td>TP1</td><td>Ground terminal test point.</td><td>n/a</td></tr>
<tr><td>U10</td><td>Link2 MCU</td><td>n/a</td></tr>
<tr><td>U9</td><td>LPC51U68 Target LQFP64 MCU</td><td>n/a</td></tr>
</table>

3. Getting Started

By default, the LPCXpresso51U68 is configured to use the on-board debug probe (Link2) to debug the on-board target (LPC51U68), using the CMSIS-DAP debug protocol pre-programmed into the Link2 Flash memory. The MCUXpresso IDE (available for free download at http://www.nxp.com/mcuxpressoide) or 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 LPC54110 family of 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 IDE. Note that spare jumpers are provided in the board packaging, taped to the card insert beneath the board.

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

By default, the LPCXpresso51U68 is configured to use the on-board Debug Probe (Link2) to debug the on-board target, 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 LPC51U68 devices (also see Section 3.1.2 for details when using IAR EWARM or Keil MDK.)

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.

3.1.1 Installation steps for use with MCUXpresso IDE

1. Download and install the MCUXpresso IDE (version 10.1 or later.)
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 JP7 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 LPCXpresso51U68 board to the USB port of your host computer, connecting a micro USB cable to connector J7 (“Link”). The board will boot and run the pre-installed demo (user RGB LED blinks).

6. Allow about 10 seconds for the LPCXpresso51U68 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 LPCXpresso51U68.

When the board is used for the first time, it is recommended to force the LPC51U68 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.

3.1.2 Installation steps to use Keil and IAR tools

Using tools other than MCUXpresso IDE require that the debug probe is programmed with firmware to support the CMSIS-DAP or SEGGER J-link protocols. While the board is programmed with CMSIS-DAP compatible firmware during manufacture, you may update to the latest release available from NXP’s website, as described below.

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 JP7 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 LPCXpresso51U68 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 LPCXpresso51U68 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 JP7 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 LPC51U68 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.

3.2 Starting a debug session using an external Debug Probe

Code running on the LPC51U68 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 J5.

Note: The Debug link connector J7 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. LPC51U68 Serial ports

By default, the LPC51U68 UART0 is connected to the FTDI header at J5. This can be used for ISP booting or sending debug messages out to a host computer via a suitable cable. The LPC51U68 UART0 can also be connected through a virtual communication port (VCOM) UART bridge Link2 function to a host computer connected to the J6 USB Link2.

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. When running this firmware, the default source of data to the LPC51U68 RXD is the FTDI header. Once the Link2 receives any data via the VCOM port of a host computer it will set P2_2 low to select the Link2 UART0 data to the LPC51U68. In order to reset this so the FTDI connection can be used it is necessary to power cycle the board.

4.1 P3 FTDI header

The FTDI header P3 mates with FTDI cable TTL-232R-3V3. P3 interfaces the LPC51U68 UART0 to a Host PC virtual serial port. The location of P3 is shown in Fig 3. The pin out and a description of the signals at P3 are listed in Table 2.

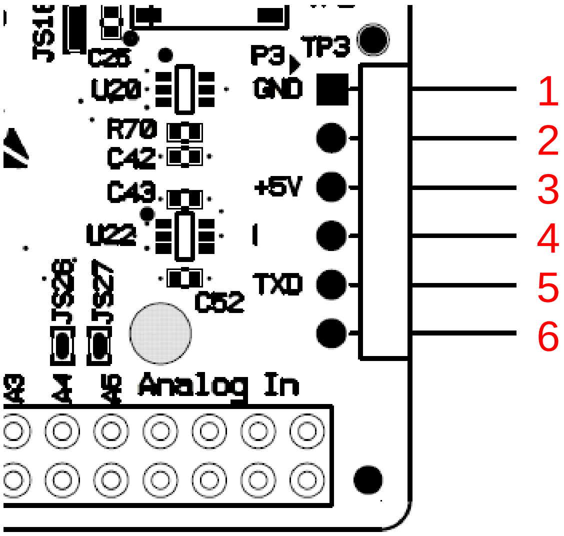

Fig 3. FTDI connector

Table 2. P3 FTDI interface

LPC51U68 Signal	FTDI signal	Pin #	Direction	LPC51U68 Signal
GND	GND	1		GND
No connect	CTS	2		No connect
Board +5V	5V	3		Board +5V
UART0_RXD	TXD	4	From host	UART0_RXD
UART0_TXD	RXD	5	To host	UART0_TXD
No connect	RTS	6		GND

5. Board power connections and measurement

The LPCXpresso51U68 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 LPC51U68 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 (J7), both the Link side and LPC51U68 Target sections of the board are powered. When the main external power is from the Target side USB micro B-type connector (J5), or FTDI header (P3) only the LPC51U68 Target section of the board is powered.

A block diagram of the board power tree is shown in Fig 4. When the LPC51U68 Target is to be debugged from an external debug probe, instead of the on-board Link2 debug probe, the Link USB connector (J7) must be disconnected. The circle with I indicates where the current monitoring circuitry measures / where an ammeter can be inserted.

The LPC51U68 Target VDD selection of 1.8V or 3.3V is made at JP9, with 3.3V set as the default.

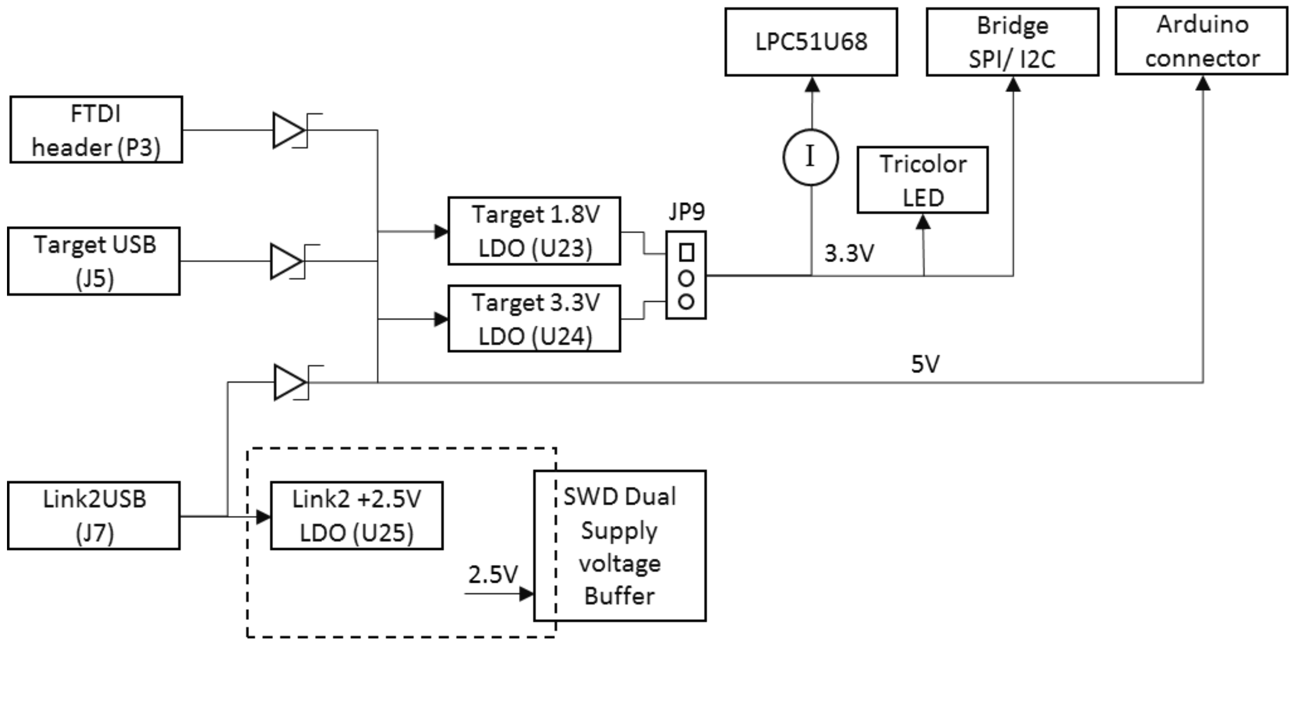

Fig 4. Board power diagram

5.1 LPCXpresso51U68 current measurement

The LPC51U68 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 LPC51U68 LQFP 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 LPC51U68 IO pins must be configured according to how the software has configured the shield board to ensure there is no extra current from the LPC51U68 IO ports that have external pull-up or pull-down resistors enabled. JS17 should be opened to ensure no leakage through the tri-color LED and +3.3V supply, and JP3 installed to avoid leakage to the Link2 via the I2C and SPI connections between it and the LPC51U68.

5.1.1 LPC51U68 Vsense resistor current measurement

The voltage across a series 4.12Ω resistor with the target LPC51U68 VDD can be manually measured at P2 on the PCB. The voltmeter positive probe is applied to P2 pin 1 (square pad) and negative probe to P2 pin 2. Use Ohm’s law to calculate the current (LPC51U68 current = measured voltage / 4.12Ω). As an example, if the measured voltage is 10mV, then 10e-3 / 4.12Ω = 2.44mA. Note that 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 LPC51U68 VDD current measurement using a current meter

A current meter may be inserted at JP6 to measure the LPC51U68 VDD input current. The 0Ω resistor at JS10 must be removed and the current meter connected at the positive input at JP6 pin 1 (square pad) and negative input at pin 2***.***

5.1.3 LPC51U68 VDD current measurement

The LPCXpresso51U68 board has an on-board current measurement circuit consisting of a MAX9634T current monitor chip and a 12-bit ADC (ADC122S021) with a 12-bit sample at 50k to 200ksps. The on-board MAX9634T current monitor measures the voltage across the LPC51U68 VDD vsense resistors; either 8.24Ω or 4.12 Ω if JP5 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 LPC51U68 current monitor and the output on 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 the MCUXpresso IDE installation.

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

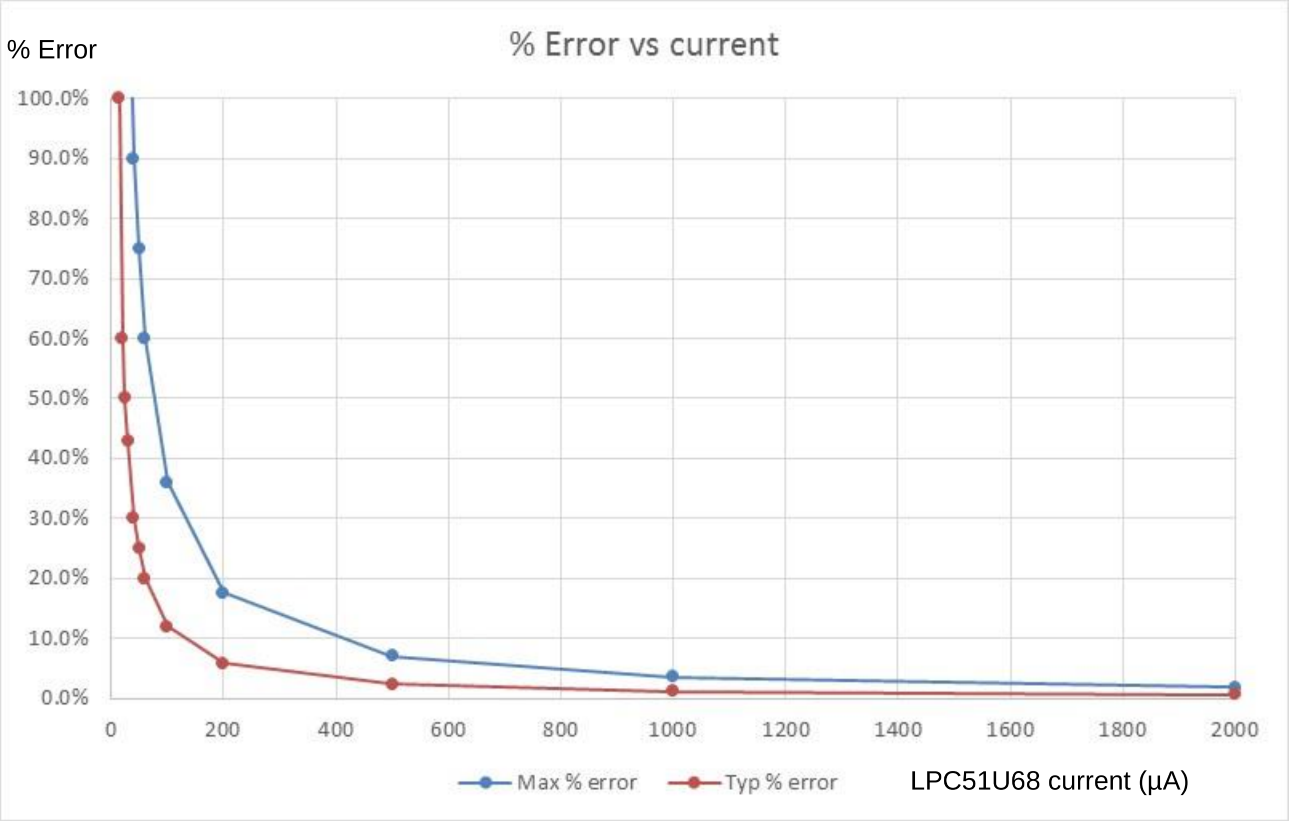

Fig 5. Current measurement circuit error

5.1.4 Shield board current measurement

To use the on-board current measurement circuitry, any expansion board must match the functionality of the LPCXpresso51U68. Refer to the board schematics for more information.

6. Debug Configurations

The LPCXpresso51U68 LQFP board has a built-in debug probe referred to as “Link2”. The LPC51U68 target MCU can be debugged by the Link2 debugging probe, or from an external debug probe installed at P1. On-board jumpers JP1 and JP2 must be correctly positioned for each mode. The on-board Link2 debug probe is capable of debugging target MCU’s with a VDDIO range of 1.6V to 3.6V. Use JP9 to set the LPC51U68 chip VDD to the desired voltage level (+1.8V or +3.3V). Check the sections below for the appropriate jumper settings and how to properly power the board.

6.1.1 Debugging LPC51U68 target using on-board (Link2) debug probe

To use the on-board Link debug probe, the LPCXpresso51U68 board must be powered from the Link2 USB connector J6, and jumper JP2 must be fitted in position pin 1 - 2 (Local Target). Jumper JP1 must be open to enable the target LPC51U68. Connecting the micro USB J6 to a host computer will power the Link and Target sections of the board and provide the USB link to the debug tool software.

6.1.2 Debug LPC51U68 target using external debug probe

To use an external debug probe, connect the probe to the SWD (P1) connector, power the LPC51U68 Target section of the board from the Target micro USB connector J5, and fit a jumper to JP2 across pin 1 - 2 (Local Target)*.* Jumper JP1 must be open to enable the target LPC51U68. The on-board Link2 debug probe must be unpowered, by leaving J7 unconnected.

6.1.3 On-board Link2 flash programing

To program the Link2 onboard flash memory, the Link2 MCU must be in DFU mode. If the Link2 already has a valid image in the flash (as it normally will after coming from the factory), you will need to force it into DFU mode by placing a jumper shunt on JP7, then power the board by connecting the micro USB J6 to a host computer*.* 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.

6.2 Using on-board Link2 to debug an off-board target LPC MCU

The LPCXpresso51U68 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 LPC51U68 MCU from interfering with the SWD interface, JP1 must be fitted. The Link2 debug probe SWD is connected by a ribbon cable between the P1 connector to the off-board target MCU SWD interface. Power the LPCXpresso51U68 board from the Link USB connector J6, and fit jumper JP2 across pins 2 - 3 (**External** Target)***.***

7. LED indicators

The LPCXpresso51U68 board LED locations are shown in Fig 2. A description of each on-board LED indicator is shown in Table 3.

Table 3. LED indicator functions

LED reference	Description
D1	Link2 MCU BOOT0_LED indicator. Reflects the state of Link2 MCU 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. It will be ON when the Link2 MCU is Booting using DFU (See description for JP6).
D2	Tri-color LED – Driven by Target LPC51U68 MCU. The red led is driven by P0_29. The green led is driven by P1_10. The blue led is driven by P1_9. LEDs are on when the LPC51U68 port is output low. By default the 0Ω resistor at JS17 provides the shunt for JP6. To measure the lowest possible current from LPC51U68 remove the 0Ω resistor at JS17 (remove +3.3V from led common anode). After removing JS17, to light any of the tri-color leds, JP8 must be installed.
D3	Target LPC51U68 power LED. This LED is on any time power is applied to the Target LPC51U68 MCU. LED will have a brighter intensity when the LPC51U68 is powered from +3.3V than when powered by +1.8V.
D4	Target Reset LED. This LED is on anytime the Target RESETn is pulled low.

8. Expansion connectors

The LPCXpresso51U68 board includes four expansion connectors plus a PMod™ compatible connector (J3). The expansion connectors (J1, J2, J7 and J8) incorporate an Arduino Uno revision 3 footprint in their inner rows. Not all connector locations are populated on the expansion connectors since the LPC51U68 does not have enough I/O to utilize all of the available connections (additional pin locations are provided for compatibility with future LPCXpresso boards).

Table 4. Expansion connectors

Reference	Description
J1	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 LPC51U68 signals not used by the Arduino Uno rev3 compatible interface.
J2	The odd numbered pins 1 – 13 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 LPC51U68 signals not used by the Arduino Uno rev3 compatible interface.
J3	PMod™ connector. Connected to the LPC51U68 Target MCU Flexcomm 3 port (for SPI or I2C use). This port is shared with the Link2 debug probe for SPI to USB bridging; install JP3 when using the PMod connector in order to disable the Link2 connection.
J7	The even numbered pins 6 – 20 are compatible with Arduino Uno rev3 Power connector. The odd numbered pins, and even numbered pins 2 and 4 are used.
J8	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 LPC51U68 signals not used by the Arduino Uno rev3 compatible interface.

9. Buttons

The LPCXpresso51U68 board has 4 push buttons available to control the operation of the LPC51U68 (target) MCU. Their functions are as described below.

9.1 Reset

This button is used to reset the LPC51U68 (note that the Link2 is NOT reset by this button.)

9.2 ISP0 and ISP1

These button connect to the LPC51U68 P0_31 (ISP0 button) and P0_4 (ISP1 button) pin and may be used to force the LPC51U68 into its various ISP boot modes. See Table 5.

Table 5. ISP boot mode control

Boot mode	ISP0	ISP1	Vbus (from J5)
I2C/SPI boot	Pressed	Pressed	X
UART boot	Pressed	Not pressed	0
USB Mass storage	Pressed	Not pressed	1
Boot from internal flash	Not pressed	Not pressed	X

To force entry into one of these boot modes hold down the required button(s), press and release the Reset button, then release the ISP button(s).

Note that for USB Mass Storage boot to work, a USB host must be connected to J5 and JP10 must be installed in order to route Vbus to the LPC51U68. When the LPC51U68 has been booted in mass storage mode a firmware image (which must be named firmware.bin) can be virtually “dragged and dropped” onto the board (which appears as a mass storage device called “CRP DISABLED” on the host computer) using a file manager utility (such as File Explorer on Windows, or Finder on MacOS). Before dropping new firmware onto the board the existing firmware.bin needs to be deleted.

This can be useful when the LPC51U68 flash has been programmed with code that disables the SWD debug pins or changes timing settings such that the debug probe has problems communicating with it. The ISP buttons can be used to force the LPC51U68 into a state where the debugger can regain debug control.

The ISP buttons can also be used to trigger an interrupt by configuring the P0_31 / P0_4 pins and associated interrupt controls within your application code.

9.3 WAKE

This button can be used to generate an interrupt by pulling down the P0_24 of the LPC51U68. Note that this signal is shared with the Link2 SPI bridge function.

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. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

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 and its suppliers accept 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 competent authorities.

Translations — A non-English (translated) version of a document is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

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.

<Name> — is a trademark of NXP Semiconductors N.V.

11. List of figures

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'.

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

Date of release: April 27, 2018

Document identifier: UM11121