UM10830

User Manual for LPCXpresso824-MAX Board

Rev. 1.2 — 20 June 2019

Document information

Info	Content
Keywords	LPCXpresso824-MAX, OM13071, LPC824
Abstract	LPCXpresso824-MAX User Manual

Revision history

Rev	Date	Description
1.0	20180209	First release
1.1	20190613	Document update

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 Cortex-M MCUs. They can be used with a range of development tools. The LPCXpresso824-MAX board is developed by NXP to enable evaluation of and prototyping with the LPC82x family of MCUs.

The LPCXpresso824-MAX includes a standard 10-pin JTAG/SWD connector plus analog/digital expansion headers, making it a highly extensible platform. Headers conforming to the LPCXpresso, Arduino UNO expansion connector standards give several options to developers wanting to leverage existing peripheral boards.

The LPCXpresso824-MAX can be configured to use external debug probes from Keil, IAR, Segger, P&E and other vendors that support CMSIS-DAP. Figure 1 shows the LPCXpresso824-MAX main board.

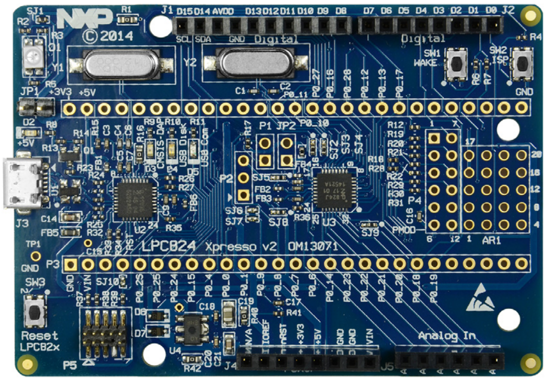


Fig 1. LPCXpresso824-MAX Board

The LPCXpresso824-MAX board includes the following features:

- Supports MCUXpresso IDE and other popular toolchains (incl. IAR and Keil)
- LPC824 Arm Cortex-M0+ MCU
- On-board CMSIS-DAP (debug probe) based on LPC11U35 MCU
- Debug connector to allow debug of target LPC824 MCU using an external probe
- Target ISP and WAKE buttons.
- Target Reset button.
- Target pins available on ‘standard LPCXpresso/mbed’ expansion connector.
- Pmod® expansion header.
- Tri-color LED
- Powering options from multiple sources.
- Options to allow measurement of current consumed by target MCU.

2. Board Layout

Figure 2 and 3 below shows the layout of the LPCXpresso824-MAX board, indicating location of jumpers, buttons and connectors/expansion options.

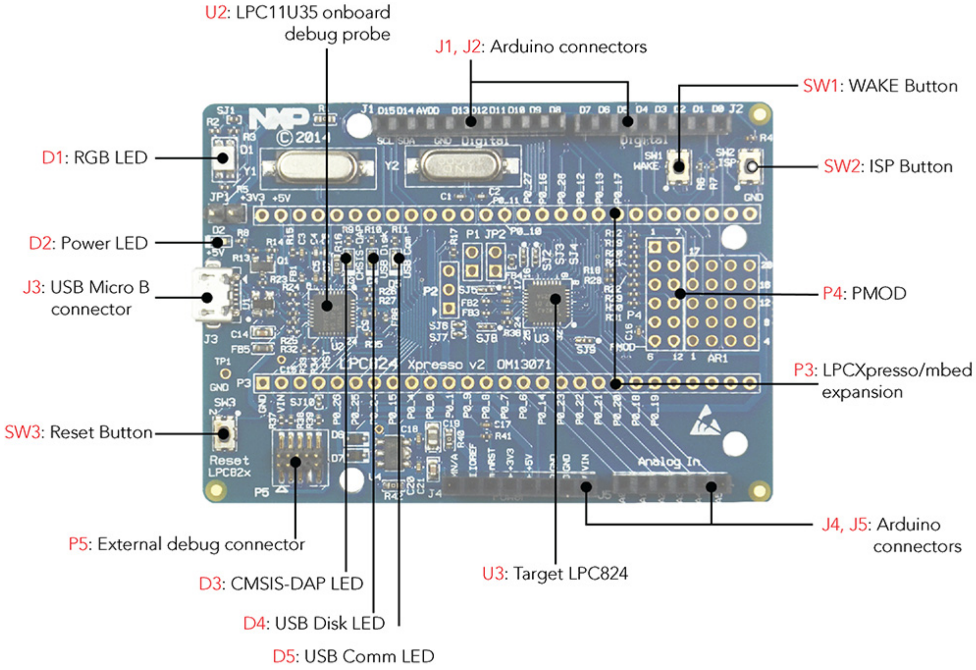

U2: LPC11U35 onboard debug probe
J1, J2: Arduino connectors
SW1: WAKE Button
D1: RGB LED
SW2: ISP Button
D2: Power LED
J3: USB Micro B connector
P4: PMOD
P3: LPCXpresso/mbed expansion
SW3: Reset Button
J4, J5: Arduino connectors
P5: External debug connector
U3: Target LPC824
D3: CMSIS-DAP LED
D4: USB Disk LED
D5: USB Comm LED

Fig 2. Board layout

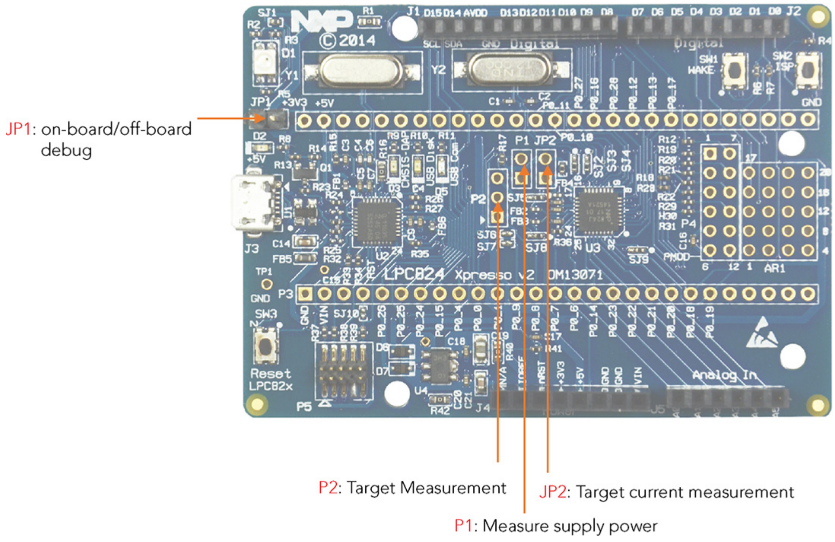

JP1: on-board/off-board debug
P2: Target Measurement
JP2: Target current measurement
P1: Measure supply power

Fig 3. Default jumpers

Table 1 below shows the layout of the LPCXpresso824-MAX board, indicating location of jumpers, buttons, connectors/expansion options and MCU devices.

Table 1. Default jumpers, button and expansion connectors

Circuit reference	Description	Reference section
JP1	On-chip debug probe disable. Insert a jumper on this header to disable the on-board debug probe when using an external probe.	5.1.1
JP2	Current measurement. Insert an Ammeter across JP2 to measure current supply to the LPC824. Solder jumper SJ2 must be removed first. Jumper not installed by default.	5.1.2
SW1	Wake button/switch	5.2.3
SW2	ISP button/switch	5.2.2
SW3	Reset button/switch	5.2.1
D1	RGB LED	5.3
D2	Power LED	5.3
D3	LED indicator for CMSIS-DAP	5.3
D4	LED indicator for USB Disk	5.3
D5	LED indicator for USB Com	5.3
P1	P1 is in parallel with 2.43 ohm resistor, which is in series with the VDD supply of the LPC824. Measuring the voltage across P1 can hence be used to determine current flow into the MCU.	5.1.2
P2	P2 provides access to inject VREFN and VREFP ADC references to the LPC824. SJ5 and SJ8 need to be changed to position 2-3 to use this header.	5.1.3
P3	LPCXpresso and mbed headers	6.1
P4	PMOD expansion	6.1.3
P5	External SWD debug connector	4.1
J1, J2, J4, J5	Arduino Analog and Digital connectors.	6.1.1
J3	USB Micro B connector	2
U2	On-board debug probe	4.1
U3	LPC824 MCU	2

3. Getting Started

This section describes the operation of the factory programmed demo program, and how to set up your board for code development with MCUXpresso IDE and/or third party tools.

3.1 Installing drivers and updating firmware

The virtual com (VCOM) port provided by the debug probe on the board requires drivers to be installed when using Windows host computers. These drivers are not required when using a Mac or Linux host. These drivers are available from http://nxp.com/demoboard/OM13071 under the Software and Tools tab (look under “Software” download for the LPC11U35 debug probe firmware). After downloading and unzipping the package, run the installer program provided to install the driver (see installation notes included in the package for further information.)

Some LPC boards are factory programmed with a CMSIS-DAP/MBED compatible firmware image. It is always recommended to update the debug probe firmware in order for the device to run properly.

For CMSIS-DAP, follow these steps to update the firmware:
1. Search “lpc11u35 debug probe firmware” and download the files on the NXP.com site.

Note: This is not required on other supported platforms (Linux and MacOS)

2. Unzip/extract the files on your PC and run the executable file.
3. Run the entire setup process for the driver installation.
4. Hold down the reset button on the board and connect the USB connector to your board and computer.
5. The board should appear on your system as a disk called CRP DISABLED.
6. Delete the file firmware.bin on that disk.
7. Drag and drop or copy the new binary image(firmware.bin) downloaded in step 1 to the disk.
8. Disconnect and re-connect USB.

For MBED, follow the steps mentioned on this site to download and update mbed firmware:

https://os.mbed.com/handbook/Firmware-LPCXpresso824-MAX

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

3.2 Connecting your board to a PC

Start by connecting to a PC using the USB connector. The status LED (D2) light will come on, indicating it has power. After a few seconds of activity, the PC will recognize the board as a device. The board runs a factory programmed demo to blink user LED.

3.3 Running Example/Demos

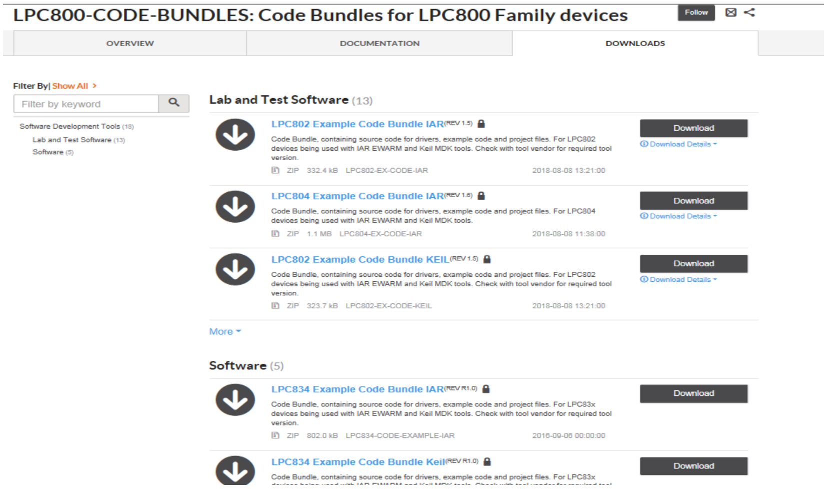
NXP provides various packages of drivers and examples for this board / device family - Code Bundles, LPCOpen and MCUXpresso SDK. It is recommended that either Code Bundles or MCUXpresso SDK are used by new users. LPCOpen support is built into MCUXpresso IDE; for more information on how to use LPCOpen please refer to the MCUXpresso IDE documentation.

The rest of this section explains how to run an example from each of the MCUXpresso SDK and Code Bundle options. Code Bundles are intended for users who prefer a lower level, register-like interface in a direct, driver-free approach. MCUXpresso SDK is intended for developers who prefer a more abstracted interface.

The LPCXpresso824-MAX board may also be used with the mbed development environment, but is currently limited to mbed 2.0. Please refer to the mbed developer site for more information https://os.mbed.com/platforms/LPCXpresso824-MAX/.

3.3.1 Using Code Bundles provided by MCUXpresso IDE.

1. Code Bundles for the LPC8xx family are included in the MCUXpresso IDE installation. These can also be downloaded from nxp.com (in case of any updates between IDE releases):
   http://www.nxp.com/products/software-and-tools/software-development-tools/software-tools/lpc800-code-bundles:LPC800-Code-Bundles.



LPC800-CODE-BUNDLES: Code Bundles for LPC800 Family devices

Lab and Test Software (13)

LPC802 Example Code Bundle IAR(REV 1.5)
Code Bundle, containing source code for drivers, example code and project files. For LPC802 devices being used with IAR EWARM and Keil MDK tools. Check with tool vendor for required tool version.
ZIP 332.4 kB LPC802-EX-CODE-IAR 2018-08-08 13:21:00

LPC804 Example Code Bundle IAR(REV 1.6)
Code Bundle, containing source code for drivers, example code and project files. For LPC804 devices being used with IAR EWARM and Keil MDK tools.
ZIP 1.1 MB LPC804-EX-CODE-IAR 2018-08-08 11:38:00

LPC802 Example Code Bundle KEIL(REV 1.5)
Code Bundle, containing source code for drivers, example code and project files. For LPC802 devices being used with IAR EWARM and Keil MDK tools. Check with tool vendor for required tool version.
ZIP 323.7 kB LPC802-EX-CODE-KEIL 2018-08-08 13:21:00

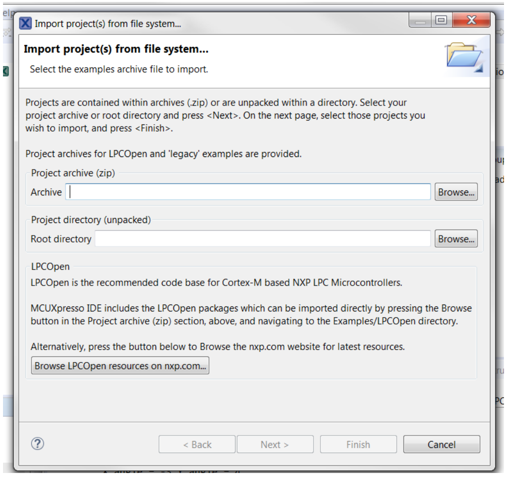
Software (5)

LPC834 Example Code Bundle IAR(REV R1.0)
Code Bundle, containing source code for drivers, example code and project files. For LPC83x devices being used with IAR EWARM and Keil MDK tools. Check with tool vendor for required tool version.
ZIP 802.0 kB LPC834-CODE-EXAMPLE-IAR 2016-09-06 00:00:00

LPC834 Example Code Bundle Keil(REV R1.0)
Code Bundle, containing source code for drivers, example code and project files. For LPC83x

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

4. In the “Import project(s) from file system...” dialog box that opens



Import project(s) from file system...

Import project(s) from file system...

Select the examples archive file to import.

Projects are contained within archives (.zip) or are unpacked within a directory. Select your project archive or root directory and press <Next>. On the next page, select those projects you wish to import, and press <Finish>.

Project archives for LPCOpen and 'legacy' examples are provided.

Project archive (zip)

Archive

Project directory (unpacked)

Root directory

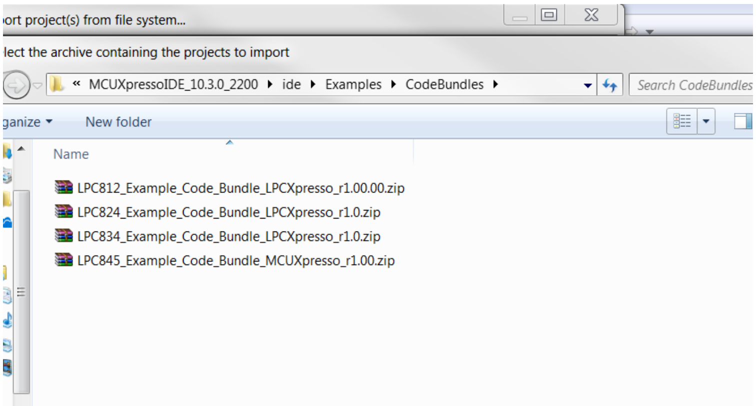
LPCOpen

LPCOpen is the recommended code base for Cortex-M based NXP LPC Microcontrollers.

MCUXpresso IDE includes the LPCOpen packages which can be imported directly by pressing the Browse button in the Project archive (zip) section, above, and navigating to the Examples/LPCOpen directory.

Alternatively, press the button below to Browse the nxp.com website for latest resources.

5. Click “Browse...” in the Project Archive (from zip) section, and select the LPC824 Code Bundle zip file from the Examples\Code Bundles directory in the MCUXpresso IDE installation (or select a version downloaded from nxp.com, as described in Step 1 above.)
6. Click “Next >” on the “Import project(s) from file system...” dialog to continue.
7. Select a project for LPC824 and click “Finish” to import the project.



ort project(s) from file system...

lect the archive containing the projects to import

« MCUXpressoIDE_10.3.0_2200 ▸ ide ▸ Examples ▸ CodeBundles ▸

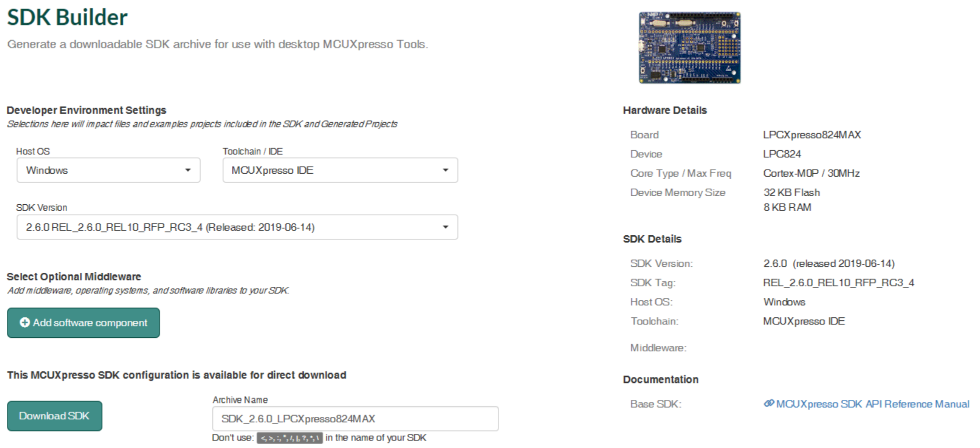
Name

LPC812_Example_Code_Bundle_LPCXpresso_r1.00.00.zip
LPC824_Example_Code_Bundle_LPCXpresso_r1.0.zip
LPC834_Example_Code_Bundle_LPCXpresso_r1.0.zip
LPC845_Example_Code_Bundle_MCUXpresso_r1.00.zip

8. The imported projects will be located in the Project tab at the upper left window of the IDE.
9. Click on Example_Multi_Timer_Blinky to select it, then select Build from the Quickstart panel. You will see the build processing in the Console window to the right of the Quickstart panel. The projects are set up to include dependency checking, so the build process will automatically build the utility and peripheral libraries as well as the example program.
10. Connect the LPCXpresso824-MAX board the host computer, click Debug in the Quickstart panel. The IDE will search for available debug probes. Select the debug probe that appears for your board, then click OK. Note that the IDE will remember your selection for the next time you debug this project, so will not prompt for this again, unless it cannot find the board.
11. The code will execute to main. Press F8 to resume and run the program. You will now see the User LEDs light, each color in turn.

3.3.2 Using NXP MCUXpresso Software Development Kit (SDK).

1. Go to the website https://mcuxpresso.nxp.com/en/welcome
2. Click on the "Select Development Board" option.
3. For the next page, you can either use the “search by name” field or you can select the board from the “Boards” list.
4. Click on the LPCXpresso824MAX, and on the right side of the page, click “Build MCUXpresso SDK”.
5. Select your Host OS and Toolchain/IDE from the drop down menu. (Optional) You can also create or edit the name of your SDK, for now we are using the default name.
6. Click “Download SDK” and save it to your PC. (Note that the SDK version number may be a later on than that shown below).



SDK Builder

Generate a downloadable SDK archive for use with desktop MCUXpresso Tools.

Developer Environment Settings

Selections here will impact files and examples projects included in the SDK and Generated Projects

Host OS: Windows

Toolchain / IDE: MCUXpresso IDE

SDK Version: 2.6.0 REL_2.6.0_REL10_RFP_RC3_4 (Released: 2019-06-14)

Select Optional Middleware

Add middleware, operating systems, and software libraries to your SDK.

This MCUXpresso SDK configuration is available for direct download

Archive Name: SDK_2.6.0_LPCXpresso824MAX

Don't use: <, >, :, ", /, |, ?, *, \ in the name of your SDK

Hardware Details

Board	LPCXpresso824MAX
Device	LPC824
Core Type / Max Freq	Cortex-M0P / 30MHz
Device Memory Size	32 KB Flash 8 KB RAM

SDK Details

SDK Version:	2.6.0 (released 2019-06-14)
SDK Tag:	REL_2.6.0_REL10_RFP_RC3_4
Host OS:	Windows
Toolchain:	MCUXpresso IDE
Middleware:	

Documentation

Base SDK:	MCUXpresso SDK API Reference Manual

7. Run MCUXpresso IDE. To get the latest version of MCUXpresso IDE, visit http://www.nxp.com/mcuxpresso/ide and click on download.
8. Once you have the IDE installed, create a workspace for you first project and click Launch.
9. Locate the SDK file you created.
10. To install the SDK, drag and drop the file to the Installed SDK view from MCUXpresso IDE and follow the prompts.
11. When the SDK installation is finished, click on “Import SDK example(s)” and select LPCXpresso824MAX board for your project.
12. Click Next and select the “led_blinky” checkbox from the demo_apps list then finish when done.
13. Select “lpcxpresso824max_led_blinky” from the project explorer, then select **Build** from the Quickstart panel. You will see the build processing in the Console window to the right of the Quickstart panel.
14. Connect the LPCXpresso824-MAX board the host computer.
15. Click **Debug** in the Quickstart panel. The IDE will search for available debug probes. Select the debug probe that appears for your board, then click OK. Note that the IDE will remember your selection for the next time you debug this project, so will not prompt for this again, unless it cannot find the board.

16. The code will execute to main. Press F8 to resume and run the program.

3.4 Using LPCXpresso824-MAX with 3rd Party IDEs

This section explains using the LPCXpresso824-MAX board with other IDEs.

3.4.1 Using Code Bundles

Code Bundles, including sample projects for Keil MDK and IAR EWARM, are available from NXP’s website at http://www.nxp.com/products/software-and-tools/software-development-tools/software-tools/lpc800-code-bundles:LPC800-Code-Bundles.

Refer to the readme files provided with each code bundle in order to build/debug.

3.4.2 Using SDK Builder

SDK Builder includes sample projects for MCUXpresso IDE, GCC ARM, IAR Embedded Workbench for ARM, and Keil MDK. These are available from the NXP website at https://mcuxpresso.nxp.com/en/welcome

Refer to the getting started guide from the page in order to build/download a customized SDK for a specific platform.

When using Keil MDK, install the Device Pack for the LPC824 (available for download from within the Keil IDE website) before attempting to use the board.

4. Debug Probe

LPCXpresso824-MAX is pre-programmed with CMSIS-DAP/mbed firmware and will work out of the box with the MCUXpresso IDE. The debug probe firmware also provides a virtual serial port (VCOM port) which will be called LPC11U3x CMSIS-DAP or mbed. See Section 3 for details on how to update the debug probe firmware.

4.1 Using an external debug probe

An external debug probe that supports ARM’s SWD interface, such as a SEGGER J-Link, LPC Link2 or PE Micro probe, can be used with the LPCXpresso824-MAX board. The external probe must be connected to header P5. When an external debug probe is used, a jumper shunt on JP1 must be installed before powering up the board in order to prevent the on-board debug contending with the external debug probe.

5. Jumpers, buttons and LEDs

5.1 Jumpers

This section describes the function of the on-board jumpers.

5.1.1 JP1: on-board/off-board debug

When this jumper is open (default), the on-board debug probe (LPC11U35 device) is used as the debug interface. The debug probe boots from flash, enumerates as a CMSIS-DAP debug probe and virtual serial port.

When JP1 is closed (jumper fitted), the on-board debug probe is held in reset, and an external debug probe can be connected using the P5 connector.

5.1.2 JP2 and P1: Target power consumption

The JP2 jumper is provided to enable by placing an ammeter in line with JP2 pins. By default, the solder jumper (SJ2) bypasses the JP2 jumper. Header P1 can be installed to measure supply (across a 2.43 ohm resistor on the board). See the schematic for more information.

P1 is in parallel with 2.43 ohm resistor, which is in series with the VDD supply of the LPC824. Measuring the voltage across P1 can hence be used to determine current flow into the MCU.

5.1.3 P2

P2 provides access to inject VREFN and VREFP ADC references to the LPC824. SJ5 and SJ8 need to be changed to position 2-3 to use this header.

5.2 Buttons

The LPCXpresso824-MAX has three push buttons available to control the operation of the LPC824 (target) MCU. Their functions are described in this sections.

5.2.1 Reset

This button is normally used to reset the LPC824. Holding down this button when the board is powered-up also places the LPC11U35 debug probe in device firmware update mode (see Section 4).

5.2.2 ISP

The ISP button connects to the LPC824 PIO_12 pin and can be used to force the LPC824 in ISP boot mode then the part is reset. This can be useful when the LPC824 flash is programmed with code that disables the SWD debug pins or changes timing settings so that the debug probe has problems communicating with it. To force ISP boot, hold the ISP button down while pressing and releasing the reset button.
The ISP button can also be used to trigger an interrupt by configuring the PIO_12 pin and associated interrupt controls within your application code.

5.2.3 Wake

Depressing this button triggers a wake interrupt by pulling-down the WAKEUP input (PIO0_4) of the LPC824.

5.3 LEDs

There are five LED devices on the board: one tricolor device (with red, green, and blue channels) for user programs, three discrete LEDs that indicate the activity of the debug probe, and a power LED.

The Tricolor LED is driven by PIO_12 (red), PIO_16 (green) and PIO_27 (blue), with the LEDs illuminating when those lines are driven low.

Note: The red tricolor LED channel is also controlled by PIO_12, so the tricolor LED turns red when ISP is depressed.

The status LEDs are connected to the LPC11U35 debug probe. They operate as follows:

• USB Disk LED (red): illuminates when the mbed disk device is being accessed.
• CMSIS-DAP (green): illuminates when debug activity (control of the target SWD port) is occurring.
• USB Comm (blue): illuminates when the mbed serial port device is active.

6. Expansion connectors/headers

The LPCXpresso824-MAX board provides an Arduino shield to add additional peripherals, sensors or other circuitry, including off-the-shelf expansion boards; this section describes these options. For further details please refer to the board schematics.

6.1 Expansion connectors

The LPCXpresso824-MAX board provides three options for expanding the capabilities of the board by adding hardware to it:

- Arduino UNO style headers
- LPCXpresso/mbed headers
- Pmod header

See the board schematics for more information.

6.1.1 Arduino UNO headers

Arduino is a popular hobbyist platform, with a standardized set of expansion connectors. The Arduino connectors on the LPCXpresso824-MAX board are compatible with the“Arduino UNO” platform. Several compatible expansion cards (shields) can be obtained from vendors such as Sparkfun, Adafruit, and others. Shield designs might vary in their implementation because the Arduino UNO platform is based on guidelines and is not a fully defined specification. See the LPCXpresso824-MAX schematics and those of the board(s) you are interfacing before attempting to connect the two together.

The Arduino UNO headers are factory installed on the LPCXpresso824-MAX.

6.1.2 LPCXpresso/mbed headers

The LPCXpresso headers on the LPCXpresso824-MAX are suitable for use with existing LPCXpresso compatible baseboards or breadboards, available from 3rd parties.

LPCXpresso expansion headers can be mounted on the bottom side of the PCB but are not factory fitted.

6.2 Arduino expansion connectors pin mappings

The Arduino compatible connectors provided on the LPCXpresso824-MAX board provide I^2C, SPI, UART, PWM and analog function connections to shield boards that are available from various 3rd part suppliers, or for customer use. The pin mappings are shown in the tables below.

Table 2. Arduino expansion connector pin mappings (J4)

Pin	Arduino signal	LPC824
1	NC	RFU
2	+3.3V	IOREF
3	Reset	RESET
4	+3V	+3V3
5	+5V	+5V0
6	GND	GND
7	GND	GND
8	NC	VIN

Table 3. Arduino expansion analog connector pin mappings (J5)

Pin	Arduino signal	LPC824 pin
1	A0	P0_6
2	A1	P0_14
3	A2	P0_23
4	A3	P0_22
5	A4	P0_21
6	A5	P0_20

Table 4. Arduino expansion connector pin mappings (J1)

Pin	Arduino signal	LPC824 pin
1	I2C SCL	P0_10
2	I2C SDA	P0_11
3	AREF	
4	GND	
5	SPI SCK	P0_24
6	SPI MISO	P0_25
7	SPI MOSI	P0_26
8	SPI SSEL	P0_15
9	PWM	P0_27
10	NC	P0_13

Table 5. Arduino expansion connector pin mappings (J2)

Pin	Arduino signal	LPC824 pin
1		P0_17
2	PWM	P0_16
3	PWM	P0_28
4		P0_18
5	PWM/INT	P0_12
6	INT	P0_19
7	UART TX	P0_4
8	UART RX	P0_0

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

6.2.1 Pmod™ headers

Pmod support is available on the LPCXpresso824-MAX:.

- Pins 1 – 6 (left side) will support Pmod™ Type 2 (SPI) or Type 1 (GPIO).
- Pins 7 – 12 (right side) design to support I2C. Will Pmod™ Type 1 (GPIO) interface.

Mount the Pmod connectors on the top side of the PCB.

Note: Connectors are not factory fitted for this interface.

7. Board Specifications

Recommended operating conditions: 0 to 70C ambient temperature
Weight: 0.7 ounces
Size: 2.15 x 3.05 inches

8. Legal information

8.1 Disclaimers

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

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

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

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

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

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

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

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

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

8.2 Trademarks

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

9. Contents