UM11181

User Manual for LPC845 Breakout board

Rev. 1.1 — 4 February 2019 User Manual

Document information

Info	Content
Keywords	LPC845 Breakout, LPC845 Breakout Board, LPC845-BRK, LPC845
Abstract	LPC845 Breakout board User Manual

Revision history

Rev	Date	Description
1.0	20181218	First release
1.1	20190201	Corrected jumper labeling

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 LPC845 Breakout board has been developed by NXP to enable quick and easy evaluation of and prototyping with the LPC84x family of MCUs. The Board has been designed to be small and simple, yet flexible and powerful, and for easy use with the MCUXpresso suite of tools and popular IDEs from NXP's partners such as Keil MDK and IAR EWARM. Figure 1 shows the LPC845 Breakout board.

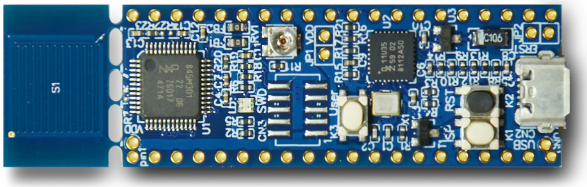

Fig 1. LPC845 Breakout Board

The LPC845 Breakout board includes the following features:

- LPC84x family MCU LPC845M301JBD48 in a 48 QFN package.
- Compatible with MCUXpresso IDE and other popular toolchains
- On-board CMSIS-DAP (debug probe) with VCOM port, based on LPC11U35 MCU
- Debug connector to allow debug of target MCU using an external probe
- Red, Green and Blue User LEDs
- ISP and User/Wake buttons
- Reset button
- Capacitive Touch evaluation button
- Potentiometer
- Access to 38 ports of the LPC845 in a DIP format for easy breadboarding
- Options to allow measurement of current consumed by target LPC845

2. Board Layout

Figure 2 below shows the layout of the LPC845 Breakout board, indicating location of jumpers, buttons and connectors/expansion options.

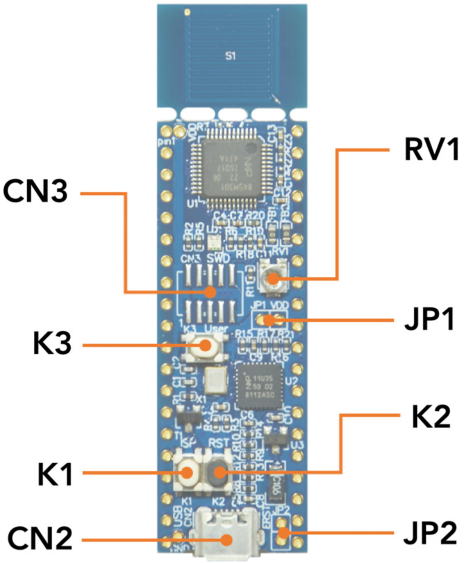

Fig 2. Board layout

Table 1 below shows the layout of the LPC845 Breakout board, indicating location of jumpers, buttons, connectors/expansion options and MCU devices.

Table 1. Jumpers and connectors (LPC845 Breakout board)

Circuit reference	Description	Reference section
JP1	Ammeter connection (not factory installed). An Ammeter can be installed between the pins of JP1 to measure current flow to the LPC845.	Section 7
JP2	Inserting a jumper on JP2 holds the LPC11U35 debug probe in reset. Install JP2 when using an external debug probe to prevent contention between the LPC11U35 and external probe.	Section 4.1
CN2	USB connector for debug probe / board power.	Section 4
CN3	External SWD debug connector for LPC845 (not fitted by default)	Section 4
RV1	Potentiometer to provide an adjustable analog signal level to the LPC84x ADC.	Section 8.6
K1	ISP button. Connected to PIO0_12	Section 8.1
K2	Reset button. Connected to PIO0_5	Section 8.2
K3	User button. Connected to PIO0_4	Section 8.3

3. Getting Started

This section describes how to get started with your new LPC845 Breakout board. This information is also provided in a step-by-step guide, including video tutorials, at the board page located at https://www.nxp.com/LPC845Breakout. Click on the Getting Started tab on this page to access this tutorial.

The LPC845 Breakout board is normally powered via USB so is easy to connect and use, but the LPC11U35 MCU on the board that provides the debug probe and VCOM (USB to serial) functionality will need a driver to be installed first if you are using Windows 7 or 8, before you plug it in. If you are using Windows 10, MacOS or Linux, or have already installed MCUXpresso IDE (version 10.1 or later) you can just plug the supplied USB cable into the micro USB connector (CN2) and your host computer to get started.

If you are using Windows 7 or 8 first you must install the VCOM device driver; this is done for you when you install MCUXpresso IDE (available for free download at http://www.nxp.com/mcuxpresso/ide). If you are using a different toolchain you can download and install the device driver package by searching for “LPC11U35 Debug Probe” on nxp.com.

When the board is first plugged in the host computer may take several seconds to enumerate the VCOM and debug probe for the first time, but the Board will start running a simple demo example as soon as it powers up. The Board includes an RGB LED, with separate red, green and blue LEDs in a single package. Once the board powers up, the green LED acts as a "breathing" LED, slowly turning on and off. When the touch pad is pressed the red LED turns on and the green one turns off. Finally, when the user button is pressed the blue LED will turn on for about half a second (the green LED isn’t affected by the User button).

Note that the Board will also output some diagnostic messages over the VCOM port, which is initialized for 9600 baud, 8 data bits, 1 stop bit. More information about the VCOM port can be found in .

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

3.1 Using the board with MCUXpresso IDE

Once the device drivers have been installed (as described above), to use the board during a debug session in MCUXpresso IDE, connect to the host computer then follow the steps below.

1. If you have note already done so, download and install the MCUXpresso IDE (version 10.3 or later is recommended).

2. Go to NXP’s SDK package build utility, located at https://mcuxpresso.nxp.com. Press the Select Development board button, then type LPC845 in the Search by Name box. Click on the LPC845 link that appears under select a Device, Board, or Kit > Processor, then select the Build MCUXpresso SDK button that is now highlighted in Green on the right of the screen.

3. The SDK Builder page will now appear - click on the Download SDK button. You will need to accept the end user license agreement, then the SDK package will be available for download from your web browser. Save it, and make a note of its location.

4. Open MCUXpresso IDE and select a workspace to use in the dialog box that appears. The IDE can create a new workspace for you if you just type in the name of the name you wish that workspace to have.

5. Once the IDE has finished launching, drag and drop the LPC845 SDK package (zip file) you downloaded in Step 3 into the Installed SDK window at the bottom right of the IDE. This gives the IDE baseline support for the LPC845 MCU.

6. Download the LPC845 Breakout example code set from the Software and Tools tab of the board site http://www.nxp.com/LPC845Breakout. This is a zip file that contains a set of examples for the Board.

7. In the Quickstart panel at the bottom right of the MCUXpresso IDE, click on “Import projects from file system...” and, in the dialog that appears, browse to the example code set zip file you downloaded in step 6, then click Finish. The IDE will now import the examples, and you will see them populate in the Project window at the top left of the IDE.

8. Click on the project LPC845_BoB_Blinky from the project window. Click Build in the Quickstart panel and observe the Console window, watching until the build process completes.

9. Click on Debug from the Quickstart panel; the IDE will search for the board (this only happens the first time the example is run). Click OK to connect and download the code. Once this completes, you will see the source code window update with the code execution stopped and the next line of code to be run highlighted in green. Press F8 or go to the Run menu option and click resume to run the code.

10. The blue LED on the board should now blink on and off every second.

Congratulations - you just ran your first program on the board!

3.2 Using 3rd Party IDEs

The instructions below assume the on-board debug probe will be used. As mentioned previously, this debug probe supports CMSIS-DAP protocol, so is compatible with Keil MDK, IAR EWARM and various other IDEs. MCUXpresso SDK packages currently support MDK and EWARM.

1. Download and install the MCUXpresso Config tool IDE.

2. Configure and download an LPC845 SDK package (with the third party tool chain option selected) from the MCUXpresso SDK Builder utility (https://mcuxpresso.nxp.com).

3. Open MCUXpresso Config Tool IDE.

4. In the wizard that comes up, select the “Clone SDK example and create a new configuration” radio button and click on Next.

5. On the next screen, select the location of the SDK that you had unzipped in Step 2. Then select the IDE that is being used. Note that only IDEs that were selected in the online SDK builder when the SDK was built will be available.

6. Then select the project to clone. For this example, we want to use the Led Blinky project. You can filter for this by typing “led_blinky” in the filter box and then selecting the “led_blinky” project. You can then also specify where to clone the project and the name. Then click on Finish.

7. Use the Switch Package button toolbar in the top right corner of the Package window to select the “LQFP 48 package”.

8. To enable the GREEN LED as GPIO, click on the pin name that matches the schematic (PIO1_0) under the “GPIO” column in the Pins window, enter GREEN as the identifier for the pin.

9. Switch to the Clock Tools using the perspective icon in the upper right.

10. Select the “Clock Diagram” view and select the “System clock” to highlight the clock tree path.

11. With the “System clock” highlighted, review the “Details” panel for specific clock configurations related to this clock path.

12. Change the SYSAHBCLKDIV -> fro setting to “direct from FRO oscillator”, this provides a 30MHz System Clock, but introduces an error with the PLL, which has a max frequency of 25MHz (see error message and tooltip).

13. Resolve the PLL error by changing the input source from “fro” to “fro_div”.

14. Optionally change the PLL Multiplier to *2 to set the SYSPLL clock output to the 30MHz maximum.

15. Use the Update Project to save the config settings (.mex) and generate the pin and clock source files.

16. Open the project generate by the config tools.

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

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

17. In the led_blinky.c file, find the definitions:

```
#define BOARD_LED_PORT 0U
```

```
#define BOARD_LED_PIN 12U
```

18. Copy the variable names BOARD_INITPINS_GREEN_PORT and BOARD_INITPINS_GREEN_PIN from the pinmux.h file to replace the hard coded values of the definitions in led_blinky.c (0U and 12U respectively).

19. Build, Debug, and Run the application to see the green LED flashing.

4. Debug Probe

The on-board LPC11U35 provides CMSIS-DAP debug probe functionality, plus a virtual comm port (VCOM) capability via PIO0_24 and PIO0_25 of the LPC845. This functionality bridges the LPC845 serial port via USB, so that host computer applications such as TeraTerm and PuTTY can communicate with the target LPC845.

The LPC845 Breakout board debug probe is factory programmed with an updated version of CMSIS-DAP, and a standard UART VCOM port.

4.1 Using an external debug probe

An external debug probe that supports ARM's SWD interface, such as a SEGGER J-Link or PE Micro probe, can be used with the LPC845 Breakout board. The external probe must be connected to header CN1, which is not installed during manufacture but can be purchased separately (suggested part number Samtec FTSH-105-01-L-DV-K). When an external debug probe is used, the on-board probe must be held in reset by placing a jumper on JP2. It is recommended that JP2 is fitted before powering the board.

4.2 Debugger firmware update

The board comes pre-programmed with a CMSIS-DAP firmware image; it is not normally necessary to update the debugger firmware image unless the version is earlier than 1.0.7, but this can be done by following the steps below. Note that the Windows driver version is not the same at the debugger firmware version, which can be observed using MCUXpresso IDE when connecting to the target.

1. Download the driver package from nxp.com by searching for LPC11u35 debug probe firmware (or use this direct URL: https://www.nxp.com/webapp/sps/download/license.jsp?colCode=LPC11Uxx-Debug-Probe-VCOM).

2. Unzip the package downloaded in Step 1.

3. Hold down the reset button and keep it held down while applying power to the board. Release reset. Using File Explorer (or equivalent on Mac/Linux platforms), look at the available drives on your system. A device called CRP_DISABLED will appear.

4. Delete the firmware.bin file on the CRP_DISABLED drive.

5. Drag and drop the firmware.bin file you downloaded from nxp.com on to the CRP_DISABLED drive.

6. Repower the board.

7. The board should now enumerate on your system - allow 20-30 seconds for this to complete. The board will be visible in your system as a COM port called NXP LPC11Uxx COM.

5. Exploring MCUXpresso SDK examples

The LPC845 SDK package includes drivers for all on-board peripherals of that MCU, and these can be found in MCUXpresso IDE by using the “Import SDK example” link in the Quickstart menu. The examples for this device are currently based on the LPCXpresso845MAX board, but can be easily modified to use the LPC845Breakout board.

In the LPC845 Breakout examples package NXP has added a series of examples of interfacing the Board to various off-the-shelf components, easily available from NXP’s etailer partners.

An MCUXpresso SDK package specifically for the LPC845 Breakout board is in development and will be available in SDK release 2.6 (due for release in Q2 2019).

6. Expansion connectors/headers

The board incorporates two expansion connectors to give access to the I/Os of the LPC845 device. Some of these pins are also used for functions on the board:

- SWD debug port (PIO0_2, PIO0_3)

- LEDs (PIO1_0, PIO1_1, PIO1_2)

- RST button (PIO0_5)

- ISP button (PIO0_12)

- User button(PIO0_4)

- VCOM UART (PIO0_24, PIO0_25)

- Cap touch button (PIO1_8, PIO1_9, PIO0_30, PIO0_31)

For full information on I/O configuration, refer to the user manual for the LPC84x family devices.

The LPC845 Breakout board is provided with 20x1 headers in case the board is to be used with a breadboard.

Note: the VDD connection of the LPC845 Breakout board expansion connectors is an output from the board. The Board may only be powered via USB. The low drop out Torex regulator used on the board is capable of providing a total of 250mA, including on-board and off-board current.

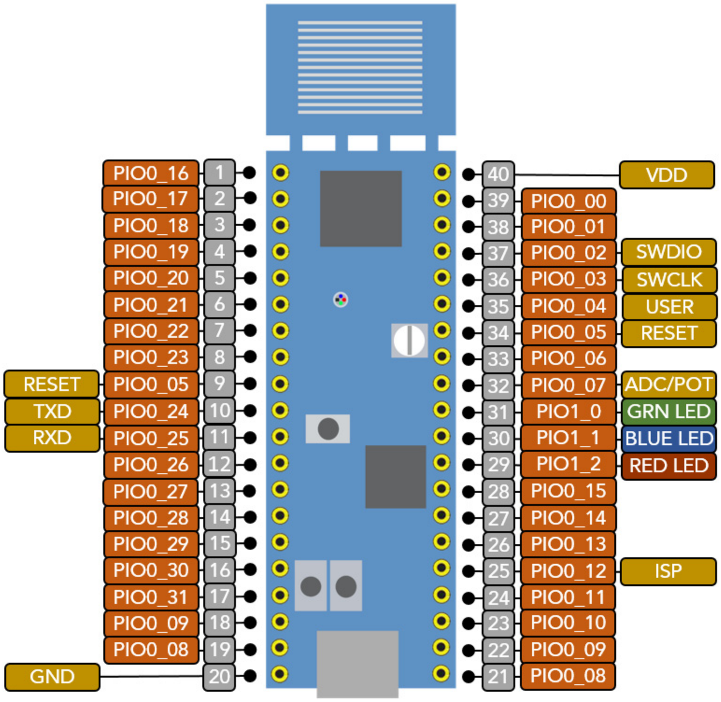

Fig 3. LPC845 Breakout board expansion connectors

7. Power supplies and current measurement

This section describes available power options for the board, and how current consumption of the LPC804 can be measured.

7.1 Power measurement

JP1 is provided to enable supply current to the LPC845 to be measured by placing an ammeter in line with JP1 pins. This header is not installed during manufacture and must be added if used, but the Board comes with a spare 1x2 header for this purpose. A narrow PCB trace connects the two holes for JP1, so this must be cut with a knife before installing the header in order for an ammeter to be able to measure the current. The LPC845 has a single supply to both internal logic and I/O pads, so when measuring supply current be careful to consider the possible effects of I/O activity.

8. Other board features

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

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

8.1 ISP booting and the ISP button (K1)

The LPC845 samples port PIO0_12 while booting and so can be forced into ISP (in-system programming) mode by holding down the ISP button (K2) while holding and releasing the Reset button (K1) or when connecting power to the board. LPC845 pin PIO0_12 can be reconfigured by software so that the ISP button can be used by an application as a general purpose button once the LPC845 has booted (refer to the LPC84x User Manual).

Using the ISP button can be a very useful way to regain control of a board if application code puts the LPC845 into a state where the debugger cannot get control.

8.2 Reset Button (K2)

The Reset Button (K2) can be used to reset the LPC845, and is also used when updating the firmware in the LPC11U35 debug probe. Holding down the Reset button when connecting power to the board will force the LPC11U35 into USB device firmware update mode; see Section 4.2 for more information. Note that the Reset button is connected to PIO0_5 of the LPC845, and can be reconfigured in software to be a GPIO if reset is not needed in the application. Care should be taken when doing this, since it will prevent a debug probe from being able to reset the LPC845, and could result in difficulties debugging the board.

8.3 User button (K3)

The User button (K3) is for general purpose use by LPC845 applications. It is connected to a 10K ohm pull-up to 3.3V, and to the LPC845 PIO0_4 pin.

8.4 User LEDs

The LPC845 Breakout Board provides an RGB LED for user applications, The LEDs are connected to the LPC845 as shown in the table below. The LEDs light when the associated port pin is pulled low.

Table 2. RGB LED connections

LED	LPC845 pin
Green	PIO1_0
Blue	PIO1_1
Red	PIO1_2

8.5 Capacitive Touch

The LPC845 Capacitive Touch button enables easy evaluation of the capacitive touch features of the LPC84x family of devices.

The connections for the capacitive touch button are shown in Table 3 below. If the Cap Touch button is not being used, the ports connected to it can be used for other purposes (such as GPIO), but note that PIO0_30 and PIO0_31 are effectively shorted together through resistor R19. If this zero ohm resistor may be removed if the Cap Touch button is not required.

Table 3. Capacitive touch button signals

Button	LPC845 pin	Capacitive touch signal
S1	PIO0_31	CAPT_X0
Common	PIO1_8	CAPT_YL
Common	PIO1_9	CAPT_YH
Common	PIO0_30	ACMP_15

8.6 Potentiometer

Potentiometer RV1 is connected to the voltage reference. The wiper of the potentiometer is connected to PIO0_7. If the potentiometer is not required resistor R23 may be removed so that PIO_7 is not affected by it.

9. Legal information

9.1 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.

9.2 Trademarks

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

10. Contents