AN10955

Full-duplex software UART for LPC111x and LPC13xx

Rev. 1 — 13 July 2010 — Application note

Document information

Info	Content
Keywords	LPC111X, LPC13XX, UART, software
Abstract	This application note illustrates how software running on an LPC111X or LPC13XX processor can implement the behavior of a standard Universal Asynchronous Receiver/Transmitter (UART)

Revision history

Rev	Date	Description
1	20100713	Initial version.

Contact information

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

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

1. Introduction

The LPC111X and LPC13XX families of microcontrollers offer a wide range of communication interfaces. Each microcontroller contains one on-chip Universal Asynchronous Receiver/Transmitter (UART). However, some applications require more UARTs than are available on the device. While the addition of an external stand-alone UART IC could be used in some of these situations, software based UARTs result in reduced BOM costs and take up no additional space on the PCB. This allows designers to meet their ever increasing miniaturization requirements and ever decreasing cost targets. This application note describes one possible implementation of the full-duplex software UART.

This software UART for LPC111X and LPC13XX is migrated from the LPC2000 software UART. For the principle of software UART implementation, please refer to AN10689 – LPC2000 software UART for details. This application note made use of the Embedded Artists' LPCXpresso 1114 Rev A and LPCXpresso Base Board Rev A evaluation kits with Code Red LPCXpresso 3.3.4. Care should be taken to make sure that the proper jumper configuration on the Base Board is in place, in particular that the thermal sensor is not connected to P1.5 nor is the potentiometer connected to P0.11

Be aware that the example software packages are not optimized. The additional space required by adding the software UART functionality to an existing design can be as low as 1 kB when additional care is taken to optimize the code density. If full-duplex operation is not required by a particular application, code size can be further reduced by trimming the unused features of the UART.

2. Implementation

This solution relies on a general purpose 32-bit timer and its ability to toggle a dedicated pin while simultaneously monitoring the state of another pin. While the capture and toggle functions of the timer are used, the receive routine also uses an additional match register of the timer. Each timer has four match registers; Table 1 indicates their allocation. In this application note, Timer32_0 was used. See Table 2 for the transfer format used in the example code.

Table 1. Match register usage

Timer MRn	Function
MR0	RX
MR1	RX
MR2	Available to user
MR3	TX

The software implementation has variable depth FIFOs enabling buffering for both transmission and reception. It should be noted that the UART baud rate, as well as transmitter and receiver FIFO depth, are configured during the compilation process and thus cannot be changed at runtime.

Table 2. UART transfer format

Parameter	Value
Start Bits	1
Data Bits	8
Stop Bits	1
Parity	None

The demo source code takes advantage of the pin compatibility between the LPC111X and LPC13XX and therefore requires minimal modification to change between target platforms. Table 3 lists the functions of the pins used in the demo. Both of the targeted evaluation modules used in this application note are equipped with hardware to convert the peripheral UART signals to either RS-232 or USB. The software UART pins must be externally connected to these signals in order to communicate with a host PC. Because the pins of the LPC111X/LPC13XX default to using configurable pull-up resistors P1.6 and P1.7 should be set to inactive mode to prevent contention with the software UART signals.

Table 3. Pin functions

Pin Number	Configuration	Description
P0.11	CT32B0_MAT3	Software Tx
P1.5	CT32B0_CAP0	Software Rx
P1.6	GPIO (Inactive)	RS-232 Driver Rx
P1.7	GPIO (Inactive)	RS-232 Driver Tx
P1.11	Output	CALL indicator (active low)
P3.0	Output	TX_PRO indicator (active low)
P3.1	Output	INT CT32B0 interrupt indicator (active low)
P3.2	Output	INT_TX Tx interrupt indicator (active low)
P3.3	Output	INT_RX Rx interrupt indicator (active low)

3. Software environment

Included with this application note are several versions of the example software project. They have been tested in the environments listed in Table 4. Default (i.e., without optimization) sizes for the example project are listed in Table 5.

Table 4. Tool versions used

Tool	Version	LPC1114 PCB	LPC1343 PCB
LPCXpresso	v3.3.4	LPCXpresso LPC1114 Rev. A, LPCXpresso Base Board Rev. A	LPCXpresso LPC1343 Rev A. LPCXpresso Base Board Rev. A
IAR C/C++ Compiler for ARM	5.50.0.31878	LPC1114-SK Rev. B	LPC1343-SK Rev. B
Keil C/C++ Compiler	V4.0.0.728	LPC1114-SK Rev. B	LPC1343-SK Rev. B

Table 5. Estimated build size

Tool	Device	Text (Bytes)
LPCXpresso	LPC1114	3632
LPCXpresso	LPC1343	5236
IAR	LPC1114	3636
IAR	LPC1343	3710
Keil	LPC1114	3796
Keil	LPC1343	4216

4. Hardware configuration – IAR LPC1114 SK

An example wiring connection can be seen in Fig 1. Please be aware that the jumpers connecting P0.11 to P1.7 interfere with the 20 pin standard connector for debugging. This does not affect debugging via the on board JLink, however, it requires external debuggers such as the Keil ULINK2 use the 10 pin SWD header.

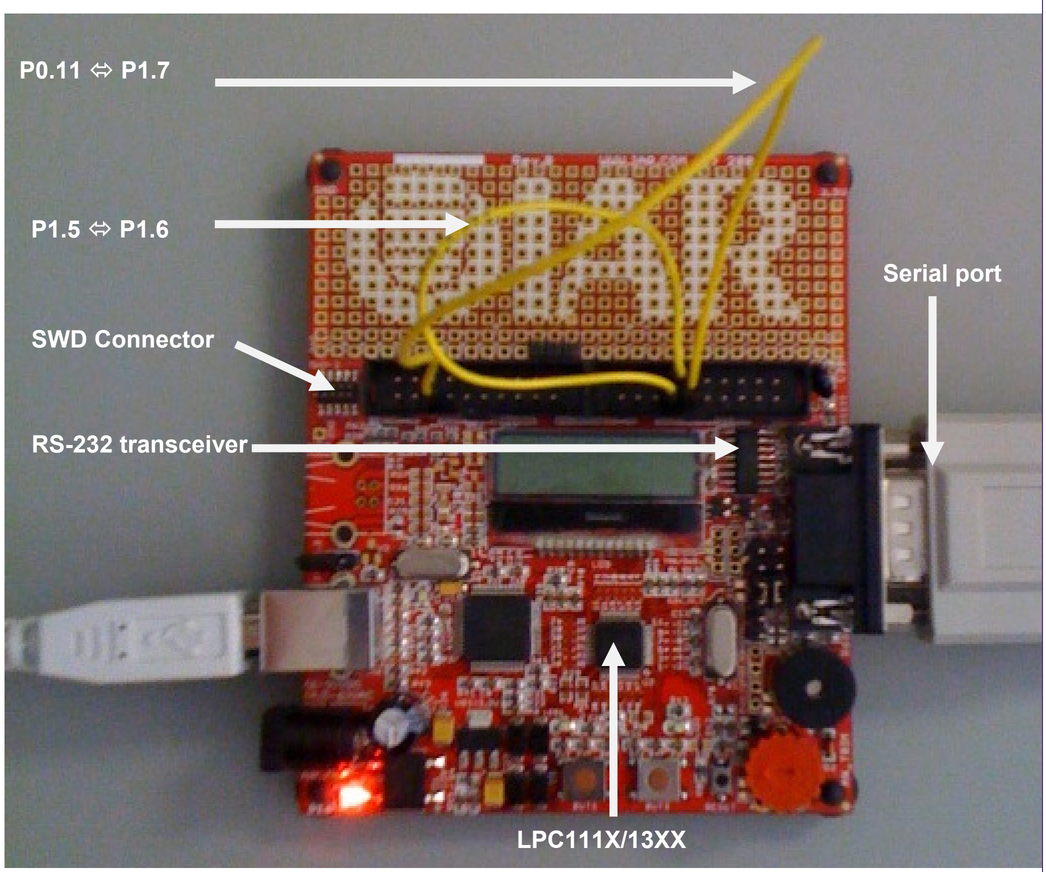

Fig 1. IAR LPC1114 SK jumpers

5. Hardware configuration – LPC1114 LPCXpresso

Fig 2 illustrates how to connect the software UART signals to the USB translation IC.

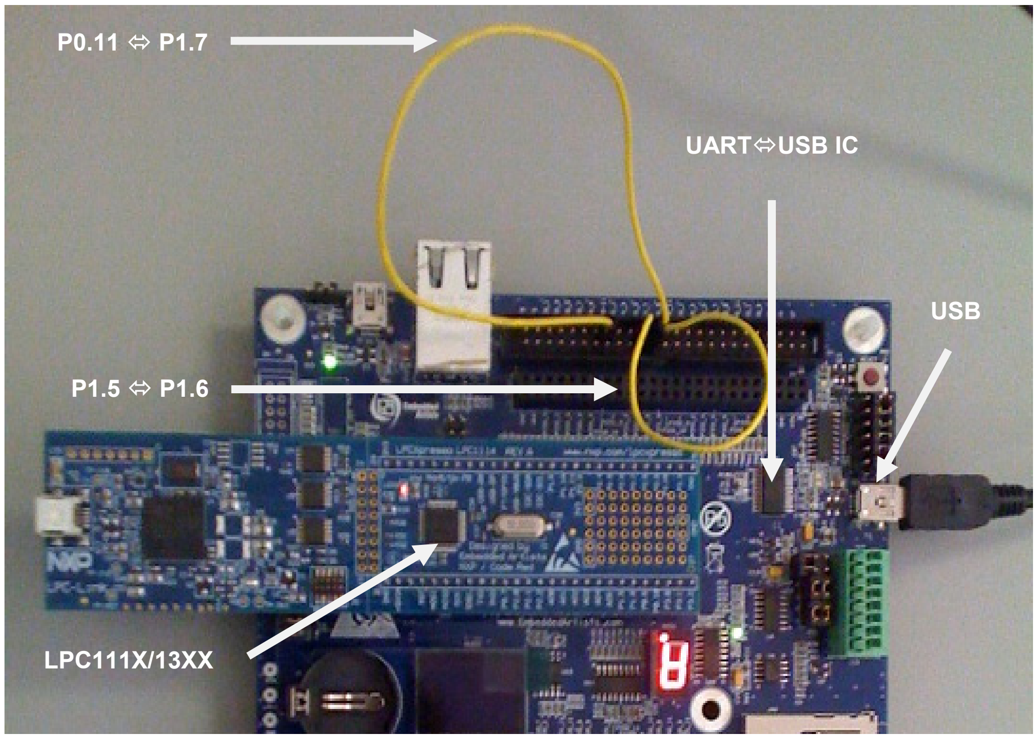

Fig 2. LPC1114 LPCXpresso and base board jumpers

6. Operation

The demo code for LPC13XX family was tested under several conditions: in a system running an external 12 MHz crystal without the use of a system PLL, as well as in systems using the PLL to generate 24 MHz, 60 MHz and 72 MHz system clocks. The LPC111X family was tested in a system running an external 12 MHz crystal as well as systems using the PLL to generate 24 MHz and 48 MHz system clocks.

It should be noted that the UART’s baud rate is dependent upon the system clock frequency. Therefore systems using a clock frequency which differ from the example code’s 48 MHz/72 MHz must update the BIT_LENGTH definitions in the project’s source code. For example, in a 12 MHz system transferring serial data at 9600 bit/s, the BIT_LENGTH is equal to 12000000/9600 = 1250. If the system clock is changed to 48 MHz, BIT_LENGTH should be changed to 48000000/9600 = 5000.

The demo application will initially prompt the user with a message, and will then enter a perpetual loop in which received characters are echoed back to the user. This is illustrated in Fig 3.

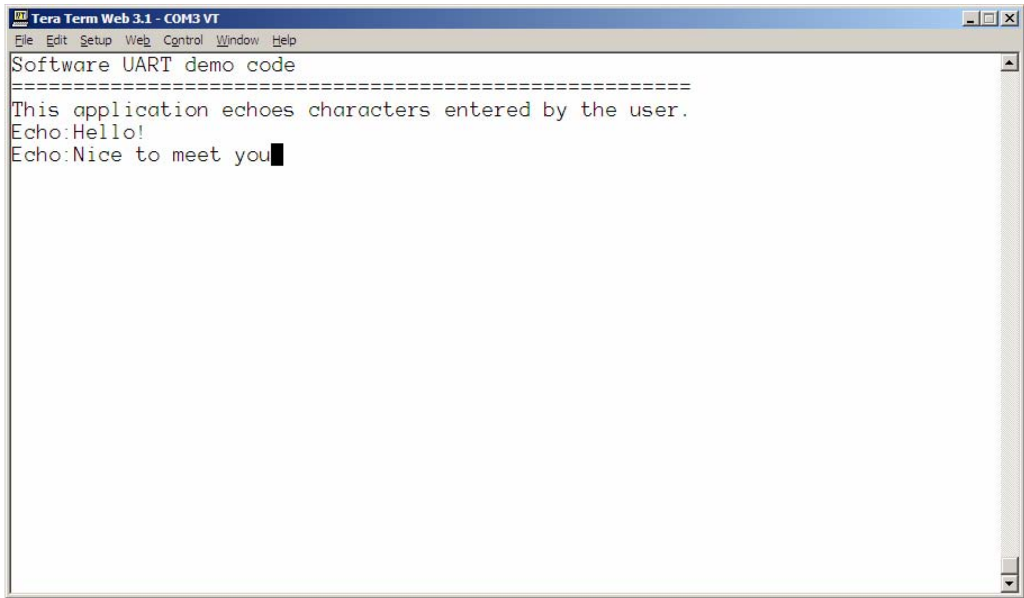

Fig 3. Example output

7. Results

The oscilloscope screenshots of signal lines could be helpful when debugging. Since they are similar to the LPC2000 software UART, please refer to AN10689 for details. Based on the demo application, the baud rates in Table 6 and Table 7 were determined to be the maximum reliable speed for each system clock frequency listed.

Table 6. LPC111X test results

System clock (MHz)	Max. transmit (bit/s)	Max. receive (bit/s)
12	9600	4800
24	38400	9600
48	57600	19200

Table 7. LPC13XX test results

System Clock (MHz)	Max. transmit (bit/s)	Max. receive (bit/s)
12	9600	4800
24	38400	9600
60	115200	19200
72	115200	38400

8. Conclusion

Despite the LPC11XX/LPC13XX part families being equipped with a single UART peripheral, with this example designers can augment their products with an additional software UART channel. By using the software based implementation no additional ICs will need to be added to the PCB, thus saving space and reducing the cost of manufacturing.

9. Legal information

9.1 Definitions

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

9.2 Disclaimers

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

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

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

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

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

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

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

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

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

9.3 Trademarks

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

10. List of figures

Fig 1. IAR LPC1114 SK jumpers.................................5
Fig 2. LPC1114 LPCXpresso and base board jumpers 6
Fig 3. Example output...7

11. List of tables

Table 1. Match register usage 3
Table 2. UART transfer format..4
Table 3. Pin functions 4
Table 4. Tool versions used..4
Table 5. Estimated build size 5
Table 6. LPC111X test results 7
Table 7. LPC13XX test results..7

12. Contents

1. Introduction 3
2. Implementation 3
3. Software environment 4
4. Hardware configuration – IAR LPC1114 SK 5
5. Hardware configuration – LPC1114 LPCXpresso 6
6. Operation 6
7. Results 7
8. Conclusion 7
9. Legal information 8
9.1 Definitions 8
9.2 Disclaimers 8
9.3 Trademarks 8
10. List of figures 9
11. List of tables 10
12. Contents 11

Date of release: 13 July 2010
Document identifier: AN10955