AN12126

Using LPC802 as I/O expander

Rev.1.0 — 20 February 2018 Application note

Document information

Info	Info
Keywords	I/O expander, I^{2}C-bus, LPC802
Abstract	This application note introduces how to use LPC802 as I/O expander via I^{2}C-bus interface.

Revision history

Rev	Date	Description
1.0	20180220	Initial version

Contact information

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

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

1. Overview

LPC802 is a new member of the LPC800 series, which satisfies the demand for improved power efficiency. It is an ideal product for the migration of market from 8-bit architecture with a very low cost. LPC802 has up to 15 MHz Cortex-M0+ core with 16 kB flash and 2 kB RAM. It is suitable for I/O expander, small programmable logic unit, etc. This application note discusses about LPC802 as an I/O expander.

As LPC802 uses ARM® Cortex®-M0+ core, it can perform single-cycle I/O toggle operation. It is suitable to use LPC802 as an I/O expander. Such kind of devices are widely used in:

- Factory automation and process control
- Portable and battery operated devices
- Cellular data devices

For more information about I/O expanders, see the following datasheets:

- https://www.nxp.com/docs/en/data-sheet/PCA9502.pdf
- https://www.nxp.com/docs/en/data-sheet/PCA8575.pdf

General IO expander selection guide for NXP can be referred from:

https://www.nxp.com/parametricSearch#/&c=c704_c674&c=c704_c674_c680&page=1

This application not uses I²C-bus interface of LPC802 to simulate on-the-market I²C-bus I/O expander devices. It has thee following information:

- General description, memory resources and pin layout, and available peripherals
- GPIO operation and pin interrupt system
- Usage of I²C-bus module, especially for I²C-bus slave mode

Note: A basic knowledge of the I²C-bus is required. For the I²C bus specification, see http://www.i2c-bus.org/specification/.

2. Hardware

2.1 MCU overview

The LPC802 is ARM Cortex-M0+ based, low-cost 32-bit MCU family operating at CPU frequencies of up to 15 MHz. It supports 16 kB of flash memory and 2 kB of SRAM. LPC802 is available in TSSOP16, TSSOP20, HVQFN33, and WLCSP16 packages. In addition, the dual power supply parts provide level shifter function, which reduces the corresponding external components and reduce the total system BOM cost.

The peripherals of the LPC802 includes:

- One I²C-bus interface
- Up to two USARTs
- One SPI interface
- One multi-rate timer, self-wake-up timer, one general purpose 32-bit counter/timer
- One 12-bit ADC, one analog comparator

- Function-configurable I/O ports through a switch matrix, and up to 17 general-purpose I/O pins. Three I/O pins have high drive capability, providing up to 20 mA source current.

2.1.1 GPIO and package information

LPC802 offers five different types of packages:

- **TSSOP20**: This package provides maximum 17 GPIOs
- **TSSOP20 with VDDIO pin:** Pin 2 (PIO_0_17) is replaced with VDDIO pin, which is a separate power supply pin for digital I/O. It enables the GPIO voltage level operating to be isolated from main V_{DD}. So, this package can provide up to 16 GPIOs
- **TSSOP16:** This package provides up to 13 GPIOs
- **HVQFN33:** This package provides up to 17 GPIOs
- **WLCSP16:** This package provides up to 13 GPIOs

In this application note, we select LPC802 TSSOP20 package.

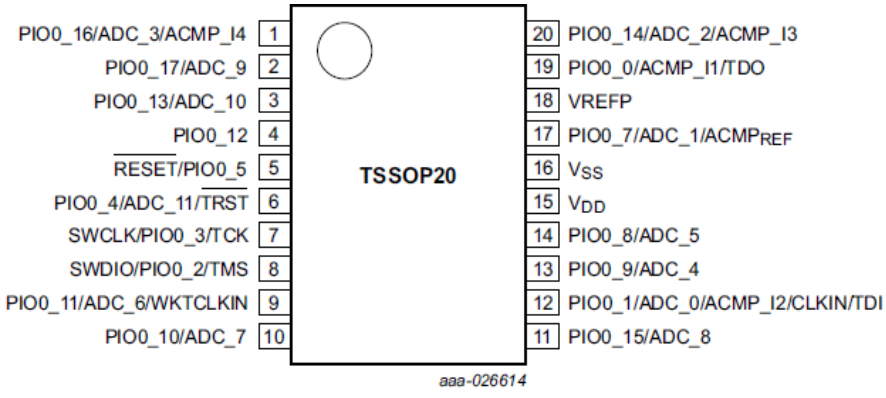


Fig 1. LPC802 TSSOP20 package

2.1.2 I²C-bus features

In this application note, LPC802 has one I²C-bus interface which is used in slave mode. I²C-bus interface features are as follows:

- Independent master, slave, and monitor functions
- Supports both multi-master and multi-master with slave functions
- Multiple I²C-bus slave addresses supported in hardware
- One slave address can be selectively qualified with a bit mask or an address range in order to respond multiple I²C-bus address

In this application, I²C0 is configured in slave mode and used as communication interface for I/O expander device.

2.2 Hardware and external connection

The hardware connection is simple, with only two pins used.

- I²C _SDA: P0_10
- I²C _SCL: P0_16

The I²C-bus requires two 4.7 kΩ pull-up resistors for SDA and SCL lines

- GPIO interface, for the demo purpose, this demo supports up to eight pins: PIO0.0 – PIO0.7, but because LPC802 does not have PIO0.6 routed out, totally seven pins are supported.
  Note that PIO0.2, PIO0.3, PIO0.4 is SWDIO, SWDCLK and RESET pins, by default. They are not configured as GPIOs. In the attached firmware, these three pins are configured as GPIO in the initialization sequence. If the user wants to program a new code after flashing the attached firmware to LPC802, the only way is to hold the ISP0 pin and re-power the board. It would make the MCU enter ISP mode and the code can be downloaded normally.
  For GPIO timing specification, see https://www.nxp.com/docs/en/data-sheet/PCA9502.pdf
- IRQ pin: If a GPIO pin is enabled to generate an interrupt and there is a level change at its input, IRQ pin will generate a 10 µs pulse."

3. Software

3.1 I²C-bus interface programming

In this application, I²C-bus is configured as slave function. To use I²C-bus module, correct initialization steps must be performed before using it. It includes clock gate control, clock routing, pin MUX etc. The following code snippet shows the initializing the I²C-bus and enabling the I²C-bus interrupt:

```
1    void app_i2c_slave_init(uint8_t slv_addr)
2    {
3        /* pin mux */
4        ConfigSWM(I2C0_SDA, P0_10);
5        ConfigSWM(I2C0_SCL, P0_16);
6
7        /* using main clock */
8        LPC_SYSCON->I2C0CLKSEL = 1;
9
10       /* give I2C a reset */
11       LPC_SYSCON->PRESETCTRL[0] &= (I2C0_RST_N);
12       LPC_SYSCON->PRESETCTRL[0] |= ~(I2C0_RST_N);
13
14       /* I2C PCLK is AHBCLK div by 3 */
15       LPC_I2C0->DIV = 2;
16       LPC_I2C0->CFG = CFG_MSTENA | CFG_SLVENA;
17
18       /* set i2c slave address */
19       LPC_I2C0->SLVADR0 = (slv_addr << 1) | 0;
20
```

```
21        // Enable the I2C0 slave pending interrupt
22        LPC_I2C0->INTENSET = STAT_SLVPEND | STAT_SLVDESEL;
23        NVIC_EnableIRQ(I2C0_IRQn);
24    }
```

I²C-bus slave operation is done by software interrupt handling. Two major I²C-bus interrupt sources are used:

- SLVPENDING: Indicates that the slave function is waiting to continue communication on the I²C-bus and needs software service
- SLVDESEL: A stop condition occurred or the new address on bus does not match current slave address

In I²C interrupt, the software should check the interrupt state register. When a STOP condition occurs, SLVDESEL is generated. Software posts a message to main thread to process more operations such as, writing the received data into NVM. When SLVPENDING is generated, software should check the state code in SLVSTATE filed in I²C0->STAT register to determine the next operation.

SLVSTATE = 0X00 (slave address received and matched)

- Software should record address by reading SLVDATA and judge whether it is a reading or writing operation
- SLVSTATE = 0X01 (slave received a new byte)
- Software should read SLVDATA to get transferred data and store in RAM. Then set CTL_SLVCONTINUE bit in SLVCTRL register, to let I²C-bus hardware continue processing bus transition
- SLVSTATE = 0X02 (slave need transmit a new byte to master)

Software should feed data into SLVDATA, then set CTL_SLVCONTINUE bit in SLVCTRL register, to let I²C-bus hardware continue processing bus transition.

3.2 I/O expander register definition

Register definition are as follows:

Table 1. I/O expander register overview

Register name	Address	Read mode	Write mode
IODir	0X0A	I/O pin direction	I/O pin direction
IOState	0X0B	I/O pin status	I/O pin logic level
IOIntEna	0X0C	I/O interrupt enable	I/O interrupt enable

3.2.1.1 I/O pin direction register(IODIr)

This register is used to set I/O pin directions. Bit 0 to Bit 7 control GPIO0 – GPIO7

Table 2. I/O pin direction register

Bit	Symbol	Description
7:0	IODir	Set GPIO pins 7:0 to input or output 0 = input 1 = output

3.2.1.2 I/O pin state register (IOState)

When read, this register returns the actual state of all I/O pins. When write, each register bit will be transferred to corresponding I/O pin programmed as output.

Table 3. I/O pin state register

Bit	Symbol	Description
7:0	IOState	write this register; set to logic level on the output pins 0 = set output pin to zero 1 = set output pin to one read this register: return state of all pins

3.2.1.3 I/O pin interrupt enable register (IOIntEna)

This register enables the interrupt due to a change in the I/O configured as input.

Table 4. I/O pin interrupt enable register

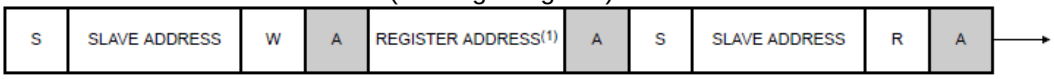
Bit	Symbol	Description
7:0	IOIntEna	input interrupt enable 0 = a change in the input pin will not generate an interrupt 1 = a change in the input will generate an interrupt When interrupt is generated, the IRQ pin will generate a 10 µs high-level pulse.

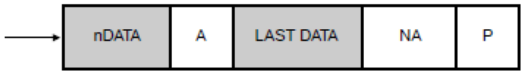
3.3 Communication protocol

The I²C-bus R/W timing is same with PCA9502.

- Master write to slave (writing a register)

- Master read data from slave (reading a register)





White block: host to LPC802

Grey block: LPC802 to host

Note that for write operation, one write sequence can only write one register. For read sequence, there is no limit for read data length. The data read always represents the latest register contents.

3.4 Summary

For slave address, we choose 0X13 to be compatible with PCA9502.
For software workflow, a standard foreground or background system is used. Only I²C-bus interrupt is enabled. Software waits for I²C-bus transition and once I²C-bus interrupt is generated, software handles the top-half process and pushes message to main thread for bottom half-task. The main thread handles the task such as writing data into GPIO register. Fig 2 shows the software workflow.

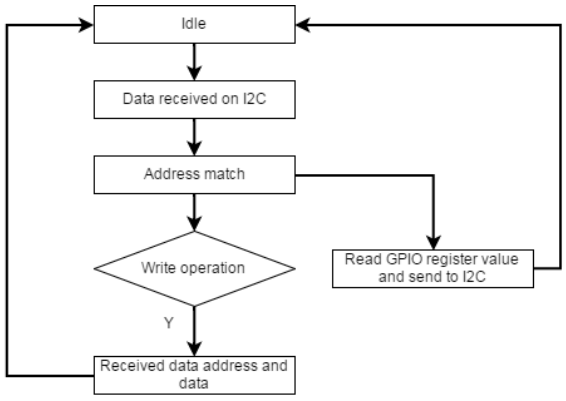


Fig 2. Diagram for software workflow

4. Test and result

4.1 Environment setup

In this section a test environment is built using LPC845 as master to read/write LPC802 through predefined I²C-bus interface. See Section 3.1
Error! Reference source not found. shows the block diagram of test environment.

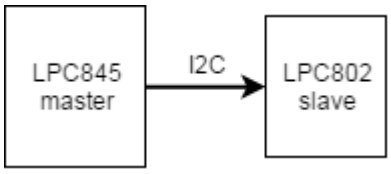


Fig 3. Block diagram of test environment

4.2 Hardware and connection:

- Master: LPC845 XpressoMAX board
  - I2C_SDA: P0_11
  - I2C_SCL: P0_10
- Slave: LPC802 demo board

- I2C_SDA: P0_10
- I2C_SCL: P0_16
- A low power LED connected between PIO0_1 and GND to see the test result

Fig 4 shows the hardware connection.

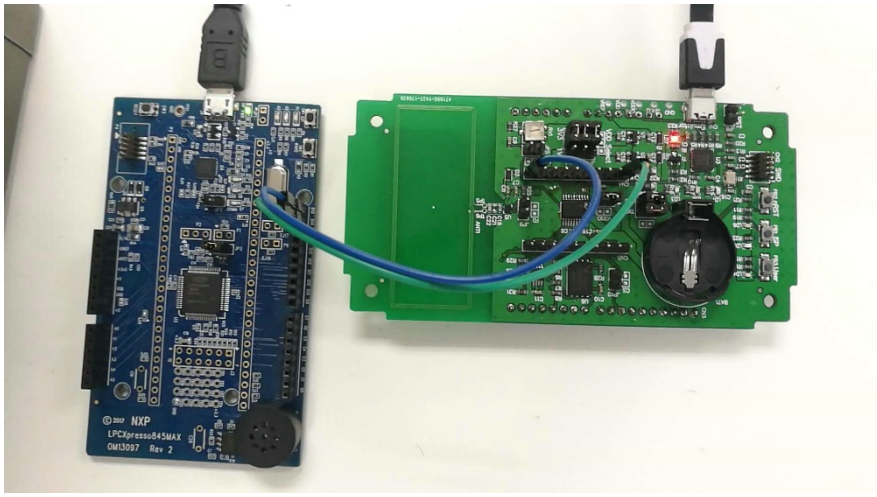


Fig 4. Hardware connection

4.3 Test step and result

- Prepare and connect hardware as described in the Section 4.2
- Download firmware to each board
  - For master demo project (LPC845): compile project under “sources/lpc845_io_expander_master.zip” and download image into LPC845 board, or download the pre-compiled image under “/bin/ lpc845_io_expander_master_test.bin” into LPC845.
  - For slave firmware project (LPC802), compile project under “lpc802_io_expander_slave.zip” and download image into LPC845 board or download the pre-compiled image under “/bin/lpc802_io_expander_slave.bin” into LPC802.

  The LPC845 will set pin direction register to 0XFF (all pins output) and writes 0X00 in pin status register for ~500 ms and then 0XFF for the next ~500 ms. It means that pin 0 to pin 7 will toggle at 1 Hz, which makes the LED connected to P0_1 to blink at 1 Hz.

5. Conclusion

This application note mainly discusses the following topics:

- LPC802 general features: Includes GPIO, pin interrupt and example code
- LPC802 I^2C-bus features, slave mode, knowledge about how to write software to co-work with I^2C-bus module and how to handle I^2C-bus transition
- A demo software using LPC802 as an I/O expander to demonstrate the above two features
- A test demo based on LPC845 Xpresso board that acts as I^2C-bus master to communicate with LPC802, also providing sources and project file

This demo helps the user to understand on how to use LPC802 as a smart device with ADC interface, multi-channel mixed signal and function expandable device.

6. Legal information

6.1 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.

6.2 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. 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.

6.3 Licenses

Purchase of NXP <xxx> components
<License statement text>

6.4 Patents

Notice is herewith given that the subject device uses one or more of the following patents and that each of these patents may have corresponding patents in other jurisdictions.

<Patent ID> — owned by <Company name>

6.5 Trademarks

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

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

7. List of figures

8. List of tables

9. Contents

For more information, visit: http://www.nxp.com
For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 20 February 2018
Document identifier:AN12126