AN11015

Adding ISP to LPC1102 systems

Rev. 1 — 5 January 2011 Application note

Document information

Info	Content
Keywords	LPC1102, ISP, FLASH, DEBUGGING
Abstract	This application note details a simple implementation which adds ISP functionality to a design. The principle can be extended in various ways, but is a recommended starting point for projects targeting LPC1102.

Revision history

Rev	Date	Description
1	20110105	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

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

© NXP B.V. 2011. All rights reserved.

1. Introduction

Because of the reduced pin count on LPC1102 devices it is difficult to anticipate what functionality a customer will use. Thus, the designer was given flexibility in how to add In-System Programming (ISP) support to their devices.

Many other similar devices in the NXP ARM Cortex-M0 and Cortex-M3 device families have a dedicated ISP pin which is evaluated by the bootloader. On the LPC1102, the designer must add this functionality to their application by taking into account the hardware and features used by their application.

The GUI tool Flash Magic (http://www.flashmagictool.com/) can be used to program LPC1102 (as well as many other NXP devices) using serial communication once ISP has been [re]invoked.

Blank LPC1102 parts will not have valid user code, and thus the bootloader will automatically invoke ISP. However, once a part is programmed with a valid application there is no way of automatically invoking ISP in hardware.

It should also be mentioned that certain applications may require pins normally used for Serial Wire Debugging (SWD) to be used for other peripherals (such as SPP/SPI). When SWD is no longer available (especially during product development) reprogramming the device can become difficult, or in some cases impossible. By adding a user controlled ISP invocation, these applications can allow themselves to be programmed via SWD flash loaders because SWD pins are enabled when ISP is invoked. If this step is neglected, the device may need to be reset to cause ISP to be invoked.

Should an application use P0_10 or P1_3 as GPIO or peripheral function pins, the designers should plan to add user invoked ISP functionality to the design as early as possible. Doing so will avoid conditions in which the LPC1102's flash cannot be reprogrammed.

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

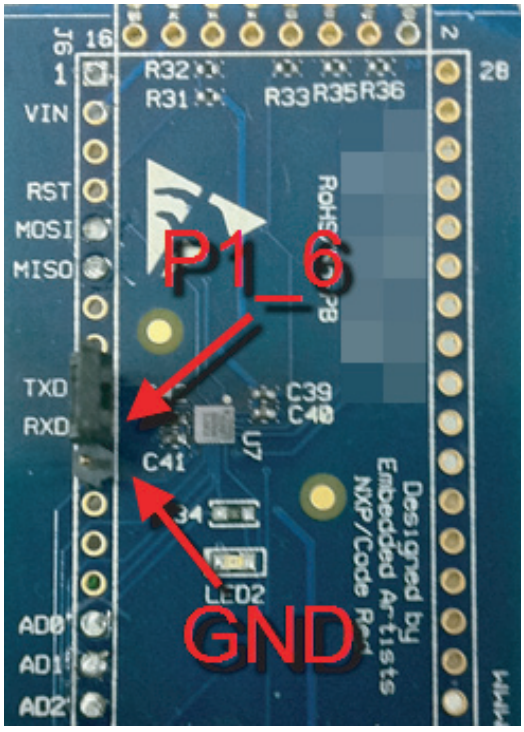
© NXP B.V. 2011. All rights reserved.

2. Hardware design

This application note was designed around example LPC1102 PCBs. On these boards the UART pins TX and RX were unused. The decision was made to allocate RX (P1_6) as the custom user ISP pin.

A header was installed on the PCB with P1_6 and a wire connected to GND. If a jumper was installed, ISP would be activated by the application code. This is illustrated in Fig 1.

Additionally, in this design, the SWCLK pin must be used for SSP communication. The application code will force the pin into SWD mode at runtime should this jumper be installed, as a means of making reprogramming via LPC-Link (or other) debuggers easier during application development.



019aab321

Fig 1. Jumper detail: When connected P1_6 is connected to GND

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

© NXP B.V. 2011. All rights reserved.

3. Software implementation

Implementing the user ISP mode in firmware is fairly straight forward. Because most development environments may abstract away low level initialization, care should be taken to ensure that the main function executes relatively early after the LPC1102 resets. Some tools may initialize C language runtimes, or do additional hardware-related initialization tasks, which may in unlikely situations prevent the code illustrated here from operating as intended.

The flow of execution can be summarized as follows:

1. Enable the relevant LPC_GPIO block
2. Configure the selected ISP pin as an input
3. Read the level on the ISP pin and conditionally invoke ISP via the “Reinvoke ISP” IAP command.

```
int main(void)
{
  /* Check for ISP... */
  InitGPIO();
  /* Assign P1_6 as INPUT */
  SetGPIOIn(ISP_PORT,ISP_PIN);
  LPC_GPIO1->DATA = 0;

  /*Check for the assertion (low) of user ISP Pin*/
  if ( 0 == GetGPIOBit(ISP_PORT,ISP_PIN) )
  {
    ReinvokeISP();
  }
  /* Setup system PLL */
  SystemInit();
  …
}
```

(1) Note – As stated previously, this project ensured that main() is called very early in the reset exception handler. Always confirm how your development environment behaves prior to executing main().

Fig 2. Example C code executed at startup

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

© NXP B.V. 2011. All rights reserved.

```
void ReinvokeISP(void)
{
  IAP iap_entry = (IAP)0x1fff1ff1;
  uint32_t command[5], result[4];
  /* make sure 32-bit Timer 1 is turned on before
  calling ISP */
  LPC_SYSCON->SYSAHBCLKCTRL |= 0x00400;
  /* make sure GPIO clock is turned on before calling ISP */
  LPC_SYSCON->SYSAHBCLKCTRL |= 0x00040;
  /* make sure IO configuration clock is turned
  on before calling ISP */
  LPC_SYSCON->SYSAHBCLKCTRL |= 0x10000;
  /* make sure AHB clock divider is 1:1 */
  LPC_SYSCON->SYSAHBCLKDIV = 1;

  /* Send Reinvoke ISP command to ISP entry point*/
  command[0] = 57;

  /* Set stack pointer to ROM value (reset default).
  This must be the last piece of code executed before
  calling ISP, because most C expressions and function
  returns will fail after the stack pointer is changed. */
  __set_MSP(*((uint32_t *)0x1FFF0000)); /* inline asm */

  /* Invoke ISP. We call "iap_entry" to invoke
  ISP because the ISP entry is done through the same
  command interface as IAP. */
  iap_entry(command, result);

  // Code will never return!
}
```

Fig 3. Example C code to invoke ISP mode manually

It is also possible to conditionally set the function of the SWD pins as needed. Fig 4 shows an example that will only allow the device to use SSP functionality when the ISP pin is de-asserted.

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

© NXP B.V. 2011. All rights reserved.

```
while(1)
{
  /* Only enable SSP when jumper is absent */
  if (GetGPIOBit(ISP_PORT,ISP_PIN))
  {
    LPC_IOCON->JTAG_TCK_PIO0_10 &= ~0x07;
    /* SSP CLK */
    LPC_IOCON->JTAG_TCK_PIO0_10 |= 0x02;
  }

  /* SSP data transfer takes place */
  …

  /* Switch Back to  SWCLK */
  LPC_IOCON->JTAG_TCK_PIO0_10 &= ~0x07;
  LPC_IOCON->JTAG_TCK_PIO0_10 |= 0x00;
}
```

Fig 4. Example C code which will only allow pins to be used for SSP when jumper is not connected

4. Conclusion

By implementing a user controlled ISP invocation, designers can ensure that their parts can be reprogrammed under all conditions. It should be noted that this example application is fairly straight forward, but more complicated methods of triggering the IAP call to invoke ISP can be used. For example, if an application has a serial command interface, rather than using a GPIO pin, a command could invoke ISP.

By ensuring that SWD pins can be forced into debugging mode, development becomes much easier, as the device does not need to be power cycled in order to be reprogrammed.

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

© NXP B.V. 2011. All rights reserved.

5. Legal information

5.1 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.

5.2 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.

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.

5.3 Trademarks

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

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

© NXP B.V. 2011. All rights reserved.

6. Contents

Please be aware that important notices concerning this document and the product(s) described herein, have been included in the section 'Legal information'.

© NXP B.V. 2011. **All rights reserved.**

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

Date of release: 5 January 2011
Document identifier: AN11015