AN11780

LPC82x I2C secondary bootloader

Rev. 1.0 — 12 November 2015 Application note

Document information

Info	Content
Keywords	LPC82x, secondary bootloader, image creator tool, I2C utility, firmware update, field update
Abstract	This application note introduces the image creator tool and I2C utility programs to help facilitate the use of an I2C SBL with an LPC82x application to enable firmware updates in the field.

Revision history

Rev	Date	Description
1	20151112	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 LPC82x provides the user a convenient way to update the flash content in the field for bug fixes or product updates. This can be achieved using the following two methods:

- ISP: In-System programming mode can be used to program or re-program the on-chip flash memory, using the internal bootloader and UART0 serial port. This can be done when the part resides on the end-user board.
- IAP: In-Application programming performs erase and write operations on the on-chip flash memory, as directed by the end-user application code.

A secondary bootloader (SBL) is a piece of code that allows a user application code to be downloaded using alternative channels other than the standard UART0 used by the internal bootloader. The primary bootloader is the firmware that resides in the microcontroller's boot ROM block and is executed on power-up and resets. After the boot ROM's execution, the secondary bootloader is executed, which then executes the end-user application.

The purpose of this document is to explain how to use two tools provided by NXP to easily incorporate an I2C SBL with any given LPC82x application binary.

2. Contents of package

Fig 1 shows the extracted contents of the package.

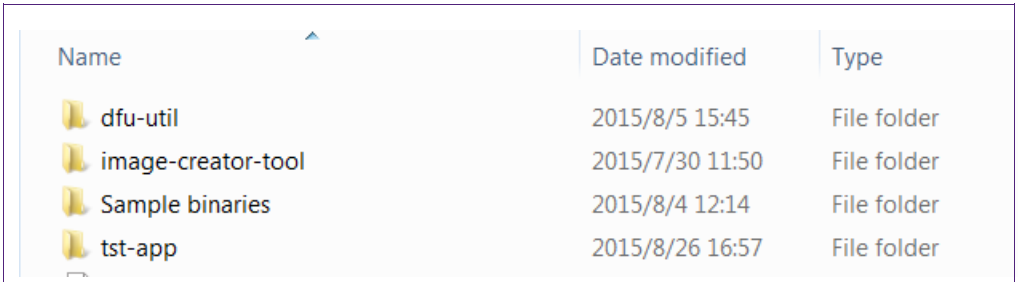


Fig 1. Package contents

A brief description for each of the folders is explained here:

1. **I2C-util** – This folder contains the I2C-util.exe with an NXP patch. This will be used to interface with the SBL through I2C.
2. **Image-creator-tool** – This folder contains the lpc82x_secimgcr.exe program that is used to create encryption keys, generating and inserting a valid CRC, encrypting firmware images, and generating factory images. The I2C SBL is embedded inside this tool.
3. **Sample binaries** – This folder contains sample binaries of all of the files that can be generated with the image creator tool.

a. tst_82x_i2c.bin – Sample application binary that was used to create all the sample firmware images in this folder.

b. secure_fac_img.bin – Secure factory image is a merged image of the I2C SBL with CRP2 activated and application binary that has been encrypted with the 'key' file in this folder. CRC is generated and inserted into application binary. Key file is placed in SBL region of memory so it can decrypt the application code.

c. plain_fac_img.bin – Plain factory image which is a merged image of the I2C SBL with no CRP enabled and unencrypted application binary. CRC is generated and inserted into application binary.

d. tst_82x_i2c_crc.bin – Unencrypted application binary with CRC generated and inserted.

e. encrypted_app_img.bin – Application binary with CRC generated and inserted, encrypted with the 'key' file in this folder.

f. key – Key file used to encrypt and decrypt the application binary. The 'key' file used to encrypt any given application image must also be used by the SBL to decrypt.

Test-app – This folder contains a Keil project for the test application.

3. Hardware and software environment

The sample test application can be tested using Keil MDK IDE v.5.14.0.0 along with LPC82x Xpresso board (#OM13071) and LPCXpresso54102 board (#OM13077) used as USB-to-I2C tool. I2C-Util tool uses I2C protocol in OM13077 board to send firmware updates to OM13071 board. Connections between LPC82x board and LPCXpresso54102 boards are shown in Fig 4.

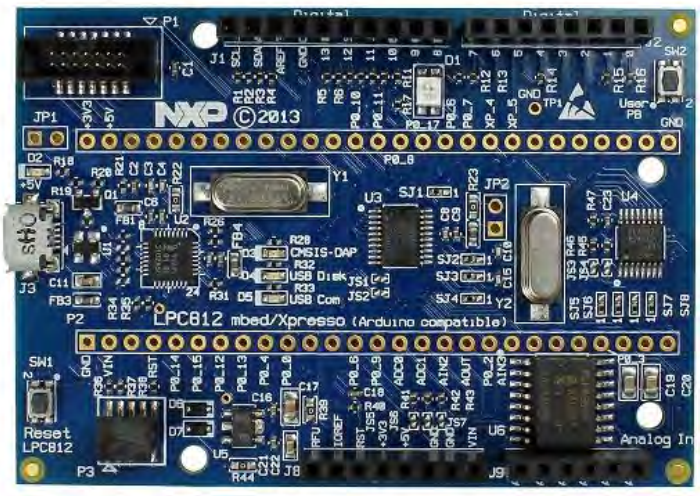
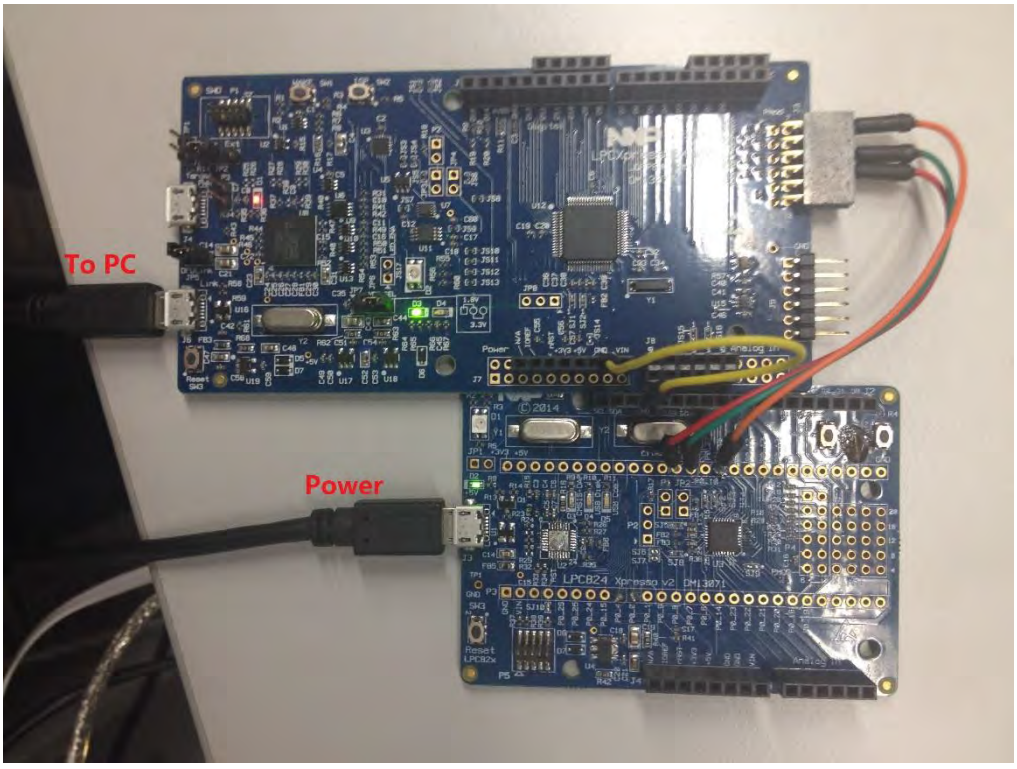


Fig 2. LPC82x Xpresso board

Fig 3. LPCXpresso54102 board as USB-to-I2C tool



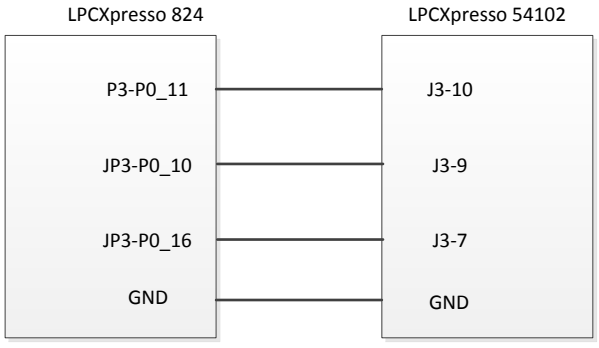


Fig 4. Connection between LPC82x board and LPCXpresso54102 board

For more information visit the following link:

http://www.nxp.com/demoboard/OM13071.html

4. Development flow

A factory image is created with the image creator tool which merges the secondary bootloader (embedded inside the tool) and the application as a single binary image. Fig 5 shows the process of creating a secure factory image.

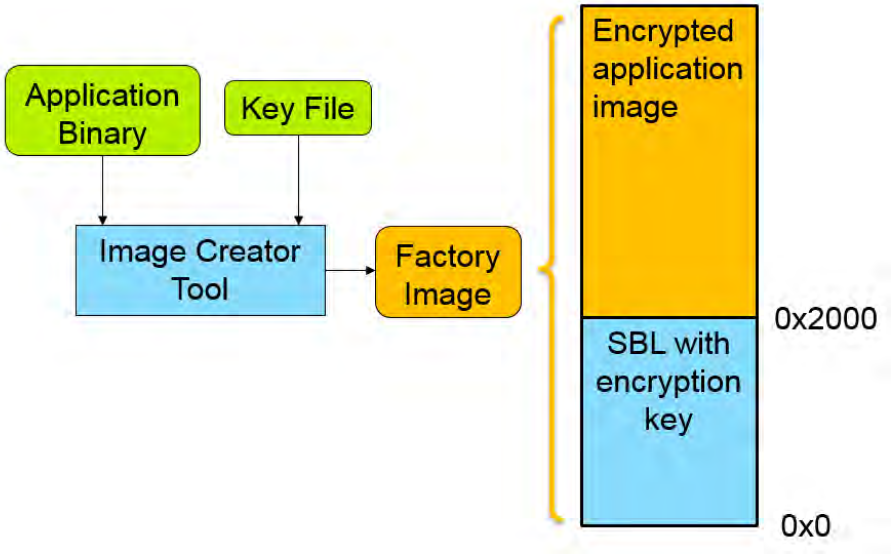


Fig 5. Creating a secure factory image

Fig 6 & Fig 7 outline the flow of the demo.

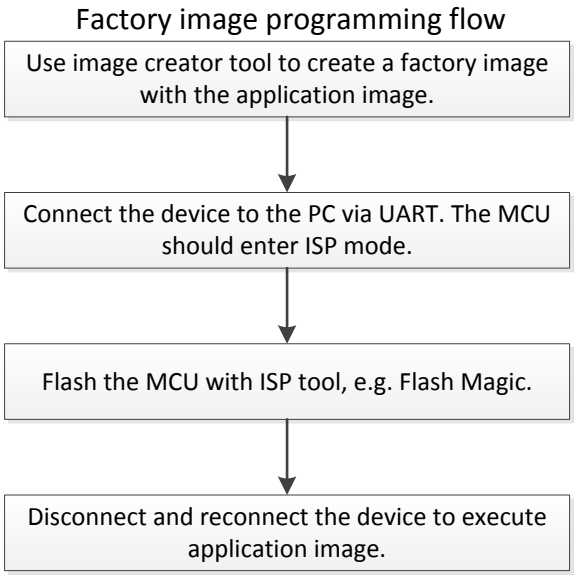


Fig 6. Factory programming flow

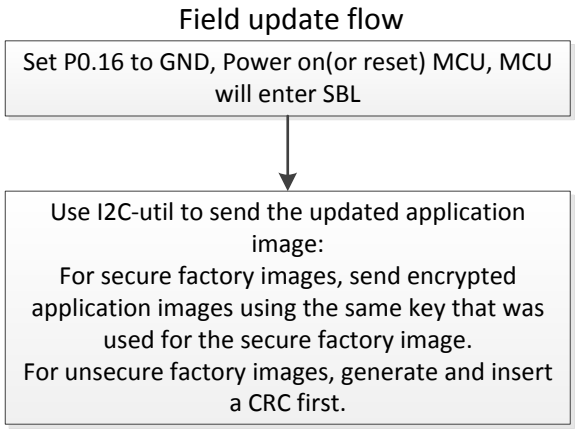


Fig 7. Field Firmware update flow

5. Download the SBL or factory file

SBL or factory file is downloaded onto the target using the factory image and ISP mode. Flash Magic can be used to download SBL or factory file, and 'bin2hex' to convert bin file to hex file. To create a factory image see section 6.2, Image creator tool.

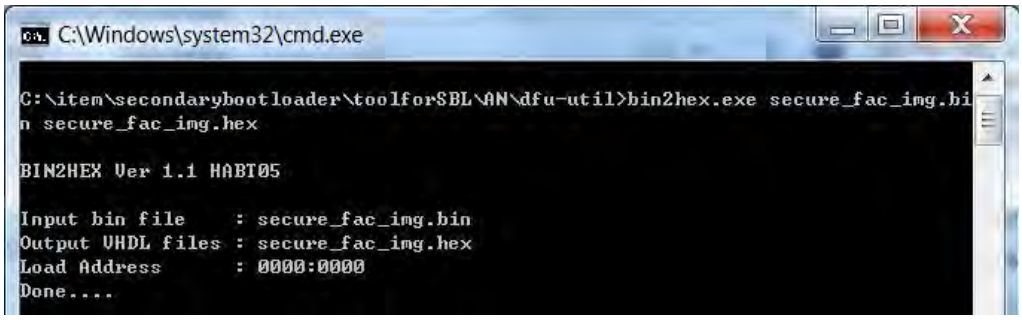


Fig 8. Convert bin file to hex file

6. Test application

The test application is a blinky example. It toggles LED D1 on the LPC824 LPCXpresso board.

The SBL occupies the first eight sectors of user flash and the test application is located at an offset 0x2000.

Fig 9 shows the *.ini file of the test application.

```
FUNC void Setup (void) {
  SP = _RDWORD(0x2000);
  PC = _RDWORD(0x2004);
}

LOAD .\keil_output\tst_82x_i2c.axf INCREMENTAL          // Download

Setup();                                        // Setup for Running

// g, main
```

Fig 9. *.ini file of test application

Build the project ‘tst_82x’ in ‘prj_82x_sbl_i2c_test’ workspace by clicking Project->Build. After build is completed a binary file ‘tst_82x_i2c.bin’ is generated in the ‘keil output’ folder.

The linker script generated by the test application is shown in Fig 10.

```
LR_IROM1 0x00002000 0x00002000  {    ; load region size_region
  ER_IROM1 0x00002000 0x00002000  {  ; load address = execution address
   *.o (RESET, +First)
   *(InRoot$$Sections)
   .ANY (+RO)
  }
  RW_IRAM1 0x10000000 0x00001000  {  ; RW data
   .ANY (+RW +ZI)
  }

  SBL_DUMMY 0x00004000 FIXED 0x0
  {
    ;Dummy section to Fill unused space with 0xFFFFFFFF
  }
}
```

Fig 10. Linker script test application

6.1 Re-invoke I2C SBL from test application

When the SBL is re-invoked from the test application, the test application sends a pin configuration table 'PINONLYCFGTABLEFLASH' to the SBL. See the SBL invoke function 'bootSecondaryLoader ()' in 'sl_prorocol.h' file.

The PINONLYCFGTABLEFLASH table is defined in the startup file along with the image header marker IMG_HEADER and CRC length. The Image header is stored at an offset 0x100 in the test application.

```
PINCFGTABLEFLASH
                DCD     0xFEEDA5A5                  ; Image header marker
                ; img_type : 0    = Normal image check IRQ line to halt boot
                ; img_type : 1    = Wait for AP to send SH_CMD_BOOT command
                ; img_type : 2    = Boot image with no AP checks
                ; img_type : 3    = No CRC or AP checks needed. Used during development
                ; img_type : 0xA5 = Image type used with SH_CMD_PROBE command
                EXPORT PINONLYCFGTABLEFLASH
PINONLYCFGTABLEFLASH
                DCB     0                           ; img_type:         See img_type values above
                DCB     1                           ; ifSel:            Interface selection for host (0,=AUTODETECT, 1=I2C0,
                DCB     ((0 << 5) + 16)             ; hostIrqPortPin:   Host IRQ port (bits 7:5) and pins (bits 4:0)
                DCB     ((1 << 5) + 4)              ; hostMisoPortPin:  SPI MISO port (bits 7:5) and pins (bits 4:0)
                DCB     ((0 << 5) + 12)             ; hostMosiPortPin:  SPI MOSI port (bits 7:5) and pins (bits 4:0)
                DCB     ((0 << 5) + 14)             ; hostSselPortPin:  SPI SEL port (bits 7:5) and pins (bits 4:0)
                DCB     ((1 << 5) + 3)              ; hostSckPortPin:   SPI SCK port (bits 7:5) and pins (bits 4:0)
                DCB     0 ^ 1 ^ ((0 << 5) + 16) ^ ((1 << 5) + 4) ^ ((0 << 5) + 12) ^ ((0 << 5) + 14) ^ ((1 << 5) + 3)    ; ch
                EXPORT  CRC32_LEN
                EXPORT  CRC32_VAL
CRC32_LEN       DCD     0                           ; Length for CRC32 check starting at offset 0, in 32-bit words
CRC32_VAL       DCD     0                           ; CRC32 value
```

Fig 11. Test application startup file

6.2 Image creator tool

A PC application is used to generate images used for the I2C secondary loader. The PC application is provided in the 'image-creator-tool' folder. Open the command prompt and navigate to the directory where the executable is located.

TIP: It is convenient to navigate to image-creator-tool folder then press 'shift' + 'right- mouse-click' and click on 'Open command window here'.

NOTE: If command prompt cannot find the input bin file, required bin file can be relocated to the folder that the command prompt is in by default (in this case '..\lpc82x_seximgcr\bin>' folder) or the navigation path must be added before input bin filename in command prompt.

6.2.1 Inserting CRC in application image

The syntax to invoke the tool to create an output binary file with image header from an input binary file is:

C:\<path>\lpc82x_secimgcr.exe <input filename.bin> <output filename.bin>

Fig 12 shows the syntax to generate the CRC for the input application binary file 'tst_82x_i2c.bin' and creates an output file 'tst_82x_i2c_crc.bin'.

```
C:\item\secondarybootloader\toolforSBL\AN\Sample binaries\lpc82x_secimgcr\bin>lp
c82x_secimgcr.exe  tst_82x_i2c.bin tst_82x_i2c_crc.bin
LPC82x Secondary Boot Loader Image Creator Utility v1.2

Opening tst_82x_i2c.bin
Generating CRC32 for the entire file!
File size = 8192
Image header offset = 0x100
Aligning image to a 32-bit alignment, adding 8192 bytes
CRC length (bytes) = 0x2000
offsetCrc = 272
img_type     = 0x0
ifSel        = 0x0
IrqPortPin   = 0x13
MisoPortPin  = 0x24
MosiPortPin  = 0xc
SselPortPin  = 0xe
SckPortPin   = 0x23
checksum     = 0x16
version     = 0x46854803
Generating CRC on bytes 0 - 0x110
Skipping CRC at bytes 0x110 - 0x113
Generating CRC on bytes 0x114 - 0x2000
CRC length:      0x00002000
CRC value :      0x6d44d449
```

Fig 12. Image with CRC header

The CRC can be generated over the image header or over the entire length of the image.

The syntax is:
C:\<path>\lpc11xx_secimgcr.exe –n[1,2] <input filename.bin> <output filename.bin>

-n indicates length of image over which CRC is generated, n1 is the full application image and n2 is just the image header. If –n[1,2] parameter is not specified the default is n1.

6.2.2 Creating a plain factory image

A plain factory image contains I2C SBL and application image as a single binary file. The syntax to create a plain factory image is:

C:\<path>\lpc82x_secimgr.exe –n[1,2] –f <application image> <factory image>

-n indicates length of image over which CRC is generated, n1 is the full application image and n2 is just the image header. If –n[1,2] parameter is not specified the default is n1.

Fig 13 shows the generation of a plain factory image ‘plain_fac_img.bin’ from the input application image ‘tst_82x_i2c.bin’.

```
C:\item\secondarybootloader\toolforSBL\AN\Sample binaries\lpc82x_secimgcr\bin>lp
c82x_secimgcr.exe -n1 -f tst_82x_i2c.bin plain_fac_img.bin
LPC82x Secondary Boot Loader Image Creator Utility v1.2
SBL binary size: 8192, application binary size: 8192
SBL Version: v0.1
Generating CRC32 for the entire file!
Image header offset = 0x100
Aligning image to a 32-bit alignment, adding 8192 bytes
CRC length (bytes) = 0x2000
offsetCrc = 272
img_type     = 0x0
ifSel        = 0x0
IrqPortPin   = 0x13
MisoPortPin  = 0x24
MosiPortPin  = 0xc
SselPortPin  = 0xe
SckPortPin   = 0x23
checksum     = 0x16
version     = 0x46854803
Generating CRC on bytes 0 - 0x110
Skipping CRC at bytes 0x110 - 0x113
Generating CRC on bytes 0x114 - 0x2000
CRC length:      0x00002000
CRC value :      0x6d44d449
```

Fig 13. Plain factory image generation

6.2.3 Generating key files for encryption

A key is required to encrypt a plain image. The syntax to generate a key file is:

C:\<path>\lpc82x_secimgr.exe –g <key file name>.

The syntax is given in Fig 14 that generates a key file ‘key’ containing the encryption key.

```
C:\item\secondarybootloader\toolforSBL\AN\Sample binaries\lpc82x_secimgcr\bin>lp
c82x_secimgcr.exe -g key
LPC82x Secondary Boot Loader Image Creator Utility v1.2
```

Fig 14. Key generation

NOTE: Store the key file in a safe location as the key file will be used in the future for field firmware updates.

6.2.4 Generating secure mode factory image

A secure mode factory image contains I2C SBL and application image as a single binary file with encryption key inserted and CRP2 enabled.

C:\<path>\lpc82x_secimgr.exe –f <key filename> <application image> <factory image>

In secure mode CRC must be calculated over the entire image length. Fig 15 lists the syntax to generate a secure factory image ‘secure_fac_img.bin’ from application image ‘tst_82x_i2c.bin’.

```
C:\item\secondarybootloader\toolforSBL\AN\Sample binaries\lpc82x_secimgcr\bin>lp
c82x_secimgcr.exe -f key tst_82x_i2c.bin secure_fac_img.bin
LPC82x Secondary Boot Loader Image Creator Utility v1.2
SBL binary size: 8192, application binary size: 8192
SBL Version: v0.1
Generating CRC32 for the entire file!
Image header offset = 0x100
Aligning image to a 32-bit alignment, adding 8192 bytes
CRC length (bytes) = 0x2000
offsetCrc = 272
img_type     = 0x0
ifSel        = 0x0
IrqPortPin   = 0x13
MisoPortPin  = 0x24
MosiPortPin  = 0xc
SselPortPin  = 0xe
SckPortPin   = 0x23
checksum     = 0x16
version     = 0x46854803
Generating CRC on bytes 0 - 0x110
Skipping CRC at bytes 0x110 - 0x113
Generating CRC on bytes 0x114 - 0x2000
CRC length:      0x00002000
CRC value :      0x6d44d449
```

Fig 15. Secure factory image generation

6.2.5 Generating encrypted image for field update

The syntax to create an encrypted image with CRC header from plain application image for field update is:

C:\<path>\lpc82x_secimgr.exe –e <key filename> <application image> <encrypted application image>

```
C:\item\secondarybootloader\toolforSBL\AN\Sample binaries\lpc82x_secimgcr\bin>lp
c82x_secimgcr.exe -e key tst_82x_i2c.bin encrypted_app_img.bin
LPC82x Secondary Boot Loader Image Creator Utility v1.2

Opening tst_82x_i2c.bin
Generating CRC32 for the entire file!
File size = 8192
Image header offset = 0x100
Aligning image to a 32-bit alignment, adding 8192 bytes
CRC length (bytes) = 0x2000
offsetCrc = 272
img_type     = 0x0
ifSel        = 0x0
IrqPortPin   = 0x13
MisoPortPin = 0x24
MosiPortPin = 0xc
SselPortPin = 0xe
SckPortPin   = 0x23
checksum     = 0x16
version     = 0x46854803
Generating CRC on bytes 0 - 0x110
Skipping CRC at bytes 0x110 - 0x113
Generating CRC on bytes 0x114 - 0x2000
CRC length:      0x00002000
CRC value :      0x6d44d449
```

Fig 16. Encrypted Image for field updates

7. Programming and updating factory Image

7.1 Programming factory image onto the target

The very first time when LPC82x is connected there is no firmware present. In order to program a secure factory image (CRP2 enabled) or unsecured factory image, programmer tool can be used to download the ‘secure_fac_img.bin’ or ‘plain_fac_img.bin’ file. See section 5, Download the SBL or factory file.

7.2 Field firmware update

After downloading SBL or factory file which includes SBL and application file, set P0.16 to GND and power on LPC824 board and LPC54102 board. Connect J6 of LPC54102 board to PC, then run I2C-util tool, see Fig 17.

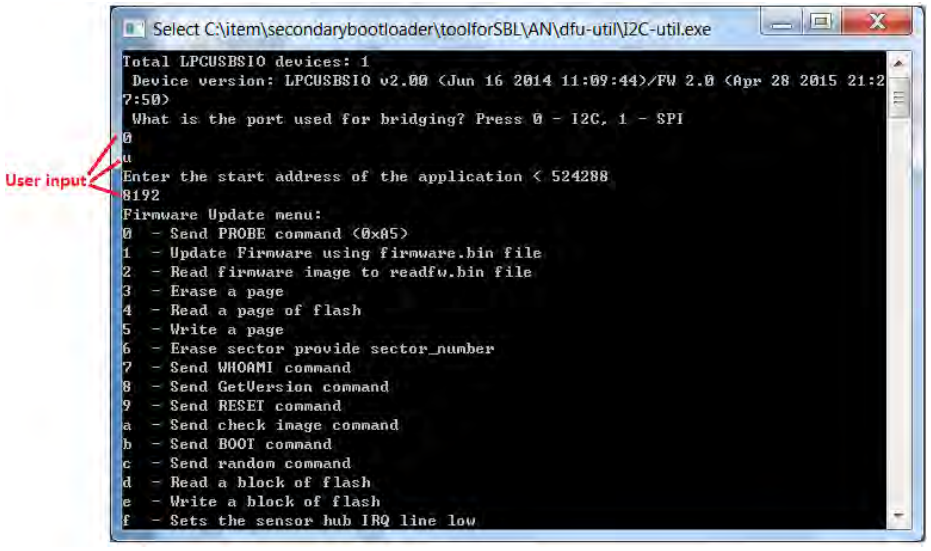


Fig 17. Run I2C_util.exe

To confirm the connection is established, you can get the SBL version, see Fig 18.

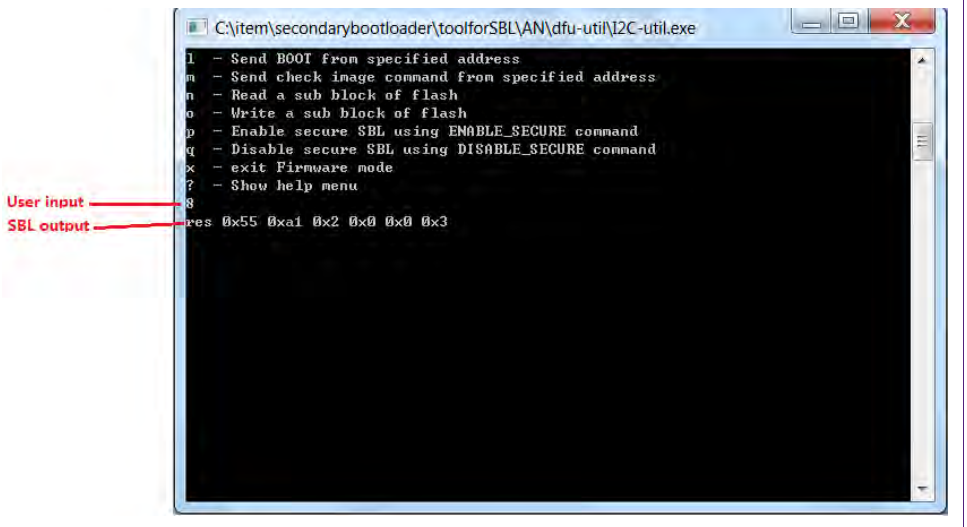


Fig 18. Get SBL version

Field firmware update in secure mode, see Fig 19.

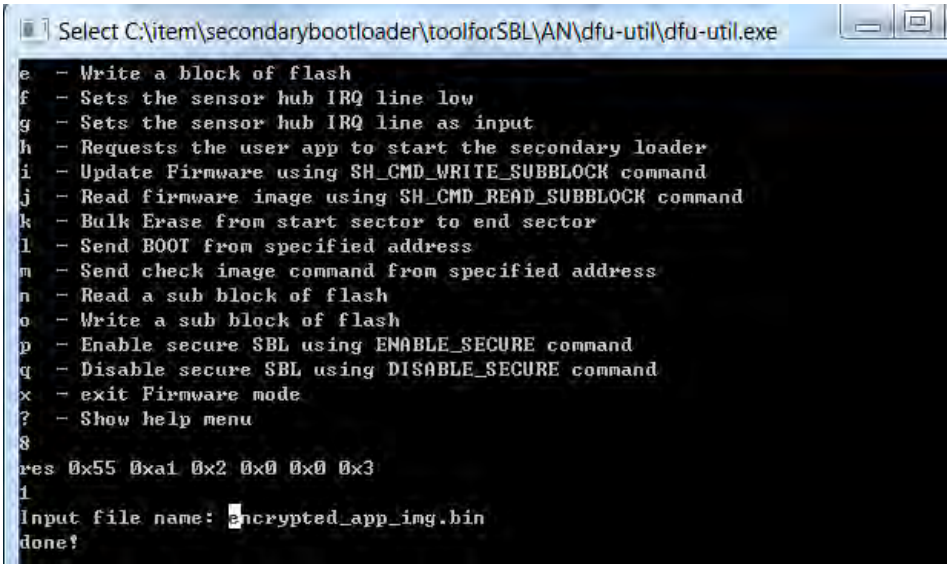


Fig 19. Field firmware update

Firmware updates can now be performed via I2C-util. An encrypted application image (see section 6.2.5, Generating encrypted image for field update) is downloaded via I2C-util. Once download is complete the application starts running.

Follow the steps in Fig 20 to update a plain field firmware image in unsecure mode.

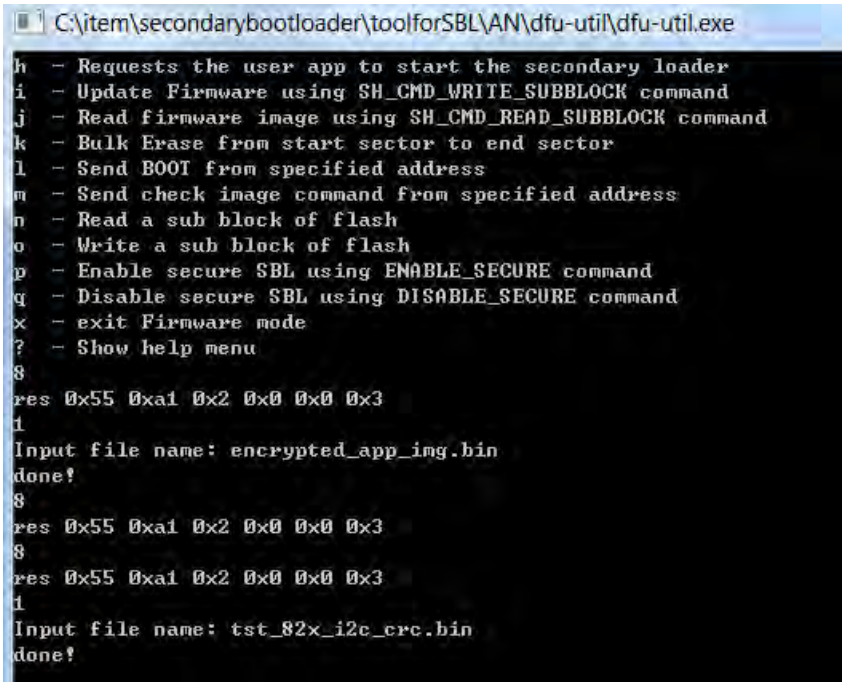


Fig 20. Field firmware update

8. Conclusion

The LPC82x provides the user a convenient way to update the flash content in real-time for bug fixes or product updates using In-Application Programming (IAP) via secondary bootloader using I2C. The functionality allows user to update the firmware using two tools, provided by NXP, to incorporate an I2C SBL with any given LPC82x application binary. A secondary bootloader (SBL) is a piece of code that allows to download a user application code using alternative channels other than the standard UART0 used by internal bootloader.

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. 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.

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.

9.3 Trademarks

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

10. Contents

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

Date of release: 12 November 2015
Document identifier: AN11780