

1 LPC55(S)1x Introduction

The LPC55S1x/LPC551x is an Arm® Cortex®-M33 based microcontroller for embedded applications. These devices include CASPER Crypto engine, up to 256 KB on-chip flash, up to 96 KB of on-chip SRAM, PRINCE module for on-the-fly flash encryption/decryption, high-speed/full-speed USB host and device interface with crystal-less operation for full-speed, CAN FD, five general-purpose timers, one SCTimer/PWM, one RTC/alarm timer, one 24-bit Multi-Rate Timer (MRT), a Windowed Watchdog Timer (WWDT), nine flexible serial communication peripherals (which can be configured as a USART, SPI, high speed SPI, I2C, or I2S interface), Programmable Logic Unit (PLU), one 16-bit 2.0 Msamples/sec ADC, comparator, and temperature sensor.

LPC55(S)1x On-chip ROM bootloader supports :

- Booting of images from on-chip flash.
- CRC32 image integrity checking.
- Flash programming through In System Programming (ISP) commands over following interfaces: USB1 interfaces using HID Class device, UART interface (Flex COMM 0) with auto baud, SPI slave interfaces (flex COMM 3 or 9) using mode 3 (CPOL = 1 and CPHA = 1), and I2C slave interface (flex COMM 1).
- ROM API functions: Flash programming API, Power control API, and Secure firmware update API using NXP Secure Boot file format, version 2.0 (SB2 files).
- Booting of images from PRINCE encrypted flash regions.
- NXP Debug Authentication Protocol version 1.0 (RSA-2048) and 1.1 (RSA-4096).
- Setting a sealed part to Fault Analysis mode through Debug authentication.

This application note focuses on un-secure firmware update to LPC55(S)1x. The firmware update uses high-speed USB port (USB port 1) and cooperates with NXP open source software `blhost` on host computer to achieve firmware update.

2 ROM boot process in non-secure

2.1 ROM boot process in non-secure

This document focuses on the boot process of ROM in non-secure condition.

Figure 1 shows the ROM boot startup flowchart. This document does not enable TrustZone and security startup, so follows the orange line for the normal boot-up sequence.

Contents

1	LPC55(S)1x Introduction.....	1
2	ROM boot process in non-secure... 1	1
2.1	ROM boot process in non-secure	1
2.2	Three ways to enter ROM USB HID firmware update process.....	2
3	Hardware and software tools.....	2
3.1	LPC55S16-EVK evaluation board	2
3.2	BLHOST - PC firmware update software tool.....	3
3.3	ELFTOSB secure firmware generation software tool.....	3
4	Update new firmware through USB port 1.....	4
4.1	blhost firmware update commands	4
4.2	How to enter ISP mode and update firmware with blhost on EVK.....	5
4.3	Phenomenon after upgrading the routine.....	5
5	How to use ELFToSB tool to generate CRC enabled firmware.....	5
5.1	LPC55(S)1x ROM code supporting firmware CRC check.....	5
5.2	How to use ELFToSB-GUI tool to generate firmware with CRC check	6
5.3	Enter USB HID ISP boot mode after damaged the firmware with CRC	8
6	KEIL, IAR and MCUXpresso generate bin Format Firmware file..	8
6.1	KEIL IDE generate bin file.....	8
6.2	IAR IDE output bin file.....	9
6.3	MCUXpresso IDE output bin file	10
7	Conclusion.....	10
8	Reference.....	10



The boot-up sequence marked with green is the on-chip flash firmware has enabled CRC check boot process. If the CRC enabled image is damaged, the chip will go to ISP firmware update status when MCU reset.

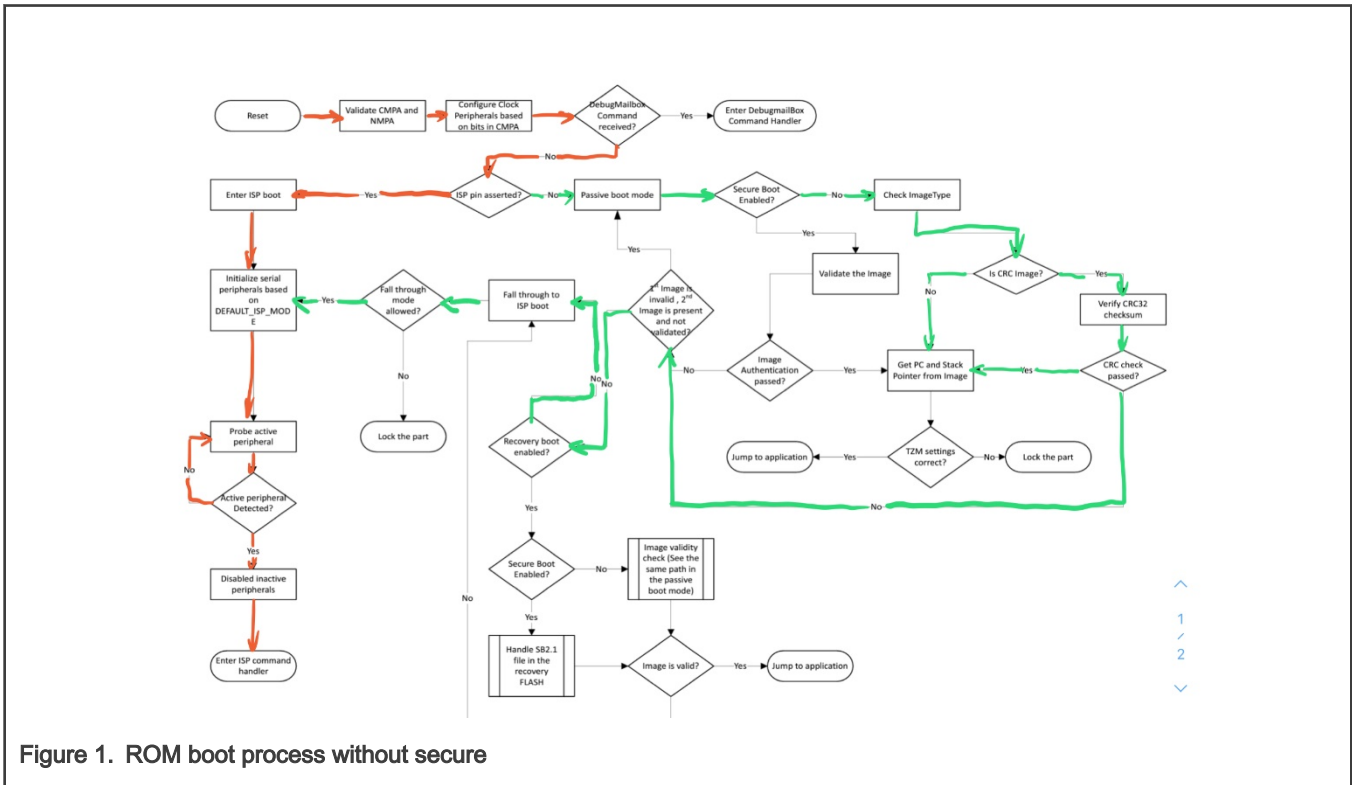


Figure 1. ROM boot process without secure

2.2 Three ways to enter ROM USB HID firmware update process

LPC55(S)1x on-chip ROM boot supports update firmware through USB port1, the high-speed USB port, in USB device HID class. There are two ways to start the ROM code to enter update firmware mode: using the `ISP (PIO0_5)` pin or calling `runBootloader()` API of the ROM in the user application.

Except for these two methods, there is another way that determines whether to enter USB HID ISP update firmware mode according to the validity or invalidity of the firmware CRC checksum.

From the startup process, the firmware binary can support CRC checksum or not during un-secure ROM boot process.

- If the firmware binary does not enable CRC checksum, MCU will start normal boot process.
- If the firmware binary has CRC checksum, ROM code will check whether the CRC value of the on-chip firmware is correct or not after MCU reset. If the on-chip firmware is incorrect, which means the on-chip firmware is damaged or changed, the ROM boot code will enter the ISP firmware update mode.

3 Hardware and software tools

3.1 LPC55S16-EVK evaluation board

Figure 2 shows the LPC55(S)1x official evaluation board – LPC55S16-EVK. This evaluation board can enter ISP mode through the **SW4-ISP** button and **SW2-RESET** button. The USB HID ISP function needs to connect the USB interface **J4 (USB1)** with PC through a micro USB cable.

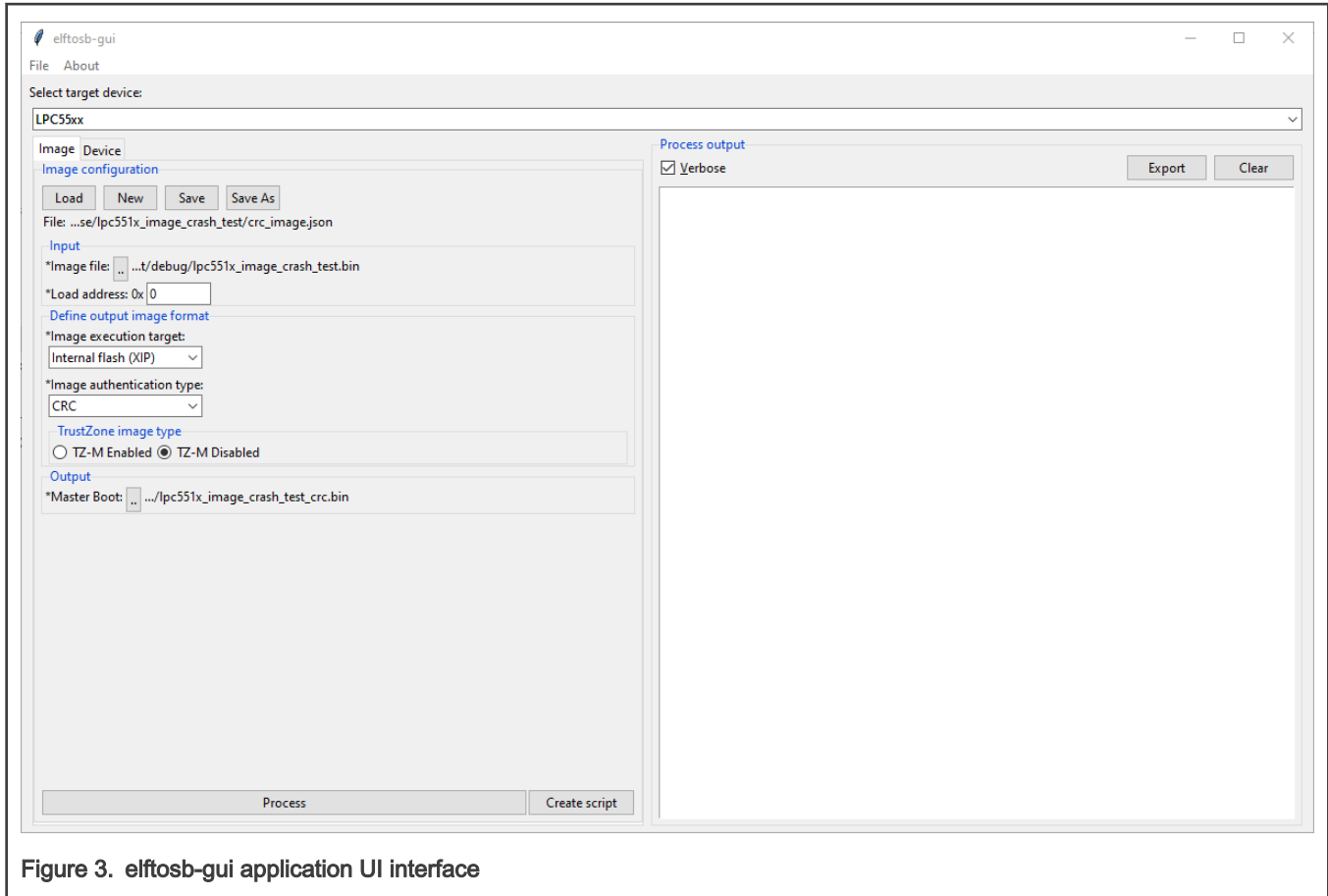


Figure 3. elftosb-gui application UI interface

For `blhost` and `elftosb` tools, download the latest software from [MCUBOOT: MCU Bootloader for NXP microcontrollers](#).

4 Update new firmware through USB port 1

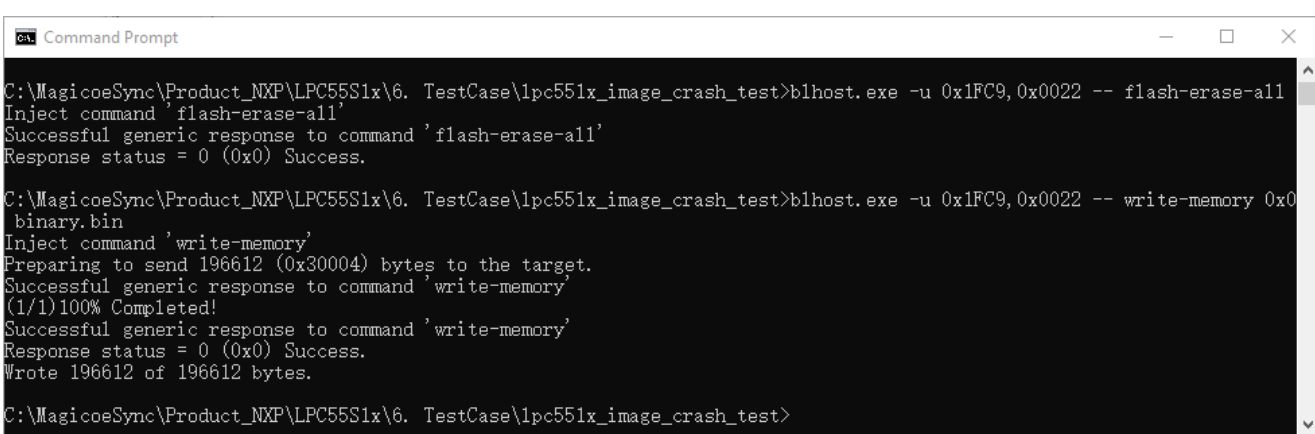
4.1 blhost firmware update commands

Because the `elftosb` tool can only add CRC checksum value into bin format file, all the firmware update in this article are based on bin format. The firmware update process will not check the CRC value of the binary and the CRC value will be checked when MCU reboots.

For how to generate bin format firmware file under different IDE, see [KEIL, IAR and MCUXpresso generate bin Format Firmware file](#).

`blhost` upgrade firmware need to use the following commands, as shown in [Figure 4](#).

```
blhost.exe -u 0x1FC9,0x0022 -- flash-erase-all
blhost.exe -u 0x1FC9,0x0022 -- write-memory 0x0 binary.bin
```



```

C:\MagicoeSync\Product_NXP\LPC55S1x\6. TestCase\lpc551x_image_crash_test>blhost.exe -u 0x1FC9,0x0022 -- flash-erase-all
Inject command 'flash-erase-all'
Successful generic response to command 'flash-erase-all'
Response status = 0 (0x0) Success.

C:\MagicoeSync\Product_NXP\LPC55S1x\6. TestCase\lpc551x_image_crash_test>blhost.exe -u 0x1FC9,0x0022 -- write-memory 0x0
binary.bin
Inject command 'write-memory'
Preparing to send 196612 (0x30004) bytes to the target.
Successful generic response to command 'write-memory'
(1/1)100% Completed!
Successful generic response to command 'write-memory'
Response status = 0 (0x0) Success.
Wrote 196612 of 196612 bytes.

C:\MagicoeSync\Product_NXP\LPC55S1x\6. TestCase\lpc551x_image_crash_test>

```

Figure 4. blhost update firmware commands

4.2 How to enter ISP mode and update firmware with blhost on EVK

Before updating firmware on LPC55S16-EVK with the `blhost` commands, enable the MCU to enter the ISP mode, as described below:

1. When the EVK board has external power supply, press and hold the **ISP** (SW4) button, press the **RESET** (SW2) button, and then release it. Connect the J4 USB micro connector (USB1/HS_USB) with PC via the Micro USB cable.
2. If the EVK's board is not powered on, press and hold the **ISP** (SW4) button, power on the EVK board, and then connect J4 interface with PC through a Micro USB cable.

Make sure that after performing the above steps, open the Command Prompt on the PC and enter the project folder with the `cd` command. Enter the `blhost` commands, as described in [blhost firmware update commands](#).

4.3 Phenomenon after upgrading the routine

If the EVK board successfully updates the firmware, press the reset (SW2) and the green LED will be blinking.

User may use the following commands through `blhost` to reset the MCU on the EVK

```
blhost.exe -u 0x1FC9,0x0022-- reset
```

5 How to use ELFtoSB tool to generate CRC enabled firmware

5.1 LPC55(S)1x ROM code supporting firmware CRC check

For the details on the working principle of ROM code, see the **Boot ROM** chapter in *LPC55S1x/LPC551x User Manual* (document [UM11295](#)). Here is only a brief introduction.

If the firmware is a CRC image, the `imageLength` field value is used as the length to perform a CRC ON, as shown in [Table 1](#). The CRC is performed on the image in internal flash. The CRC calculation begins at offset `0x0` from the beginning of the image sector and continues up to the number of bytes specified by the length. The length does not include the `offsetToSpecificHeader` field that makes up the CRC value field, which means that the CRC calculated skips the CRC value field. The result is then compared to the `offsetToSpecificHeader` entry in the structure and the image is considered valid if a match exists. Otherwise, the image is considered invalid. CRC is not performed if the image is not a CRC image.

Table 1. LPC55S1x/LPC551x image header information

Offset	Size in bytes	Symbol	Description
0x00	4	Initial SP	Stack pointer
0x04	4	Initial PC	The application first execution instruction.
0x08	24	Vector table	Cortex-M33 Vector table entries.
0x20	4	imageLength	The length of the current image. Set to 0 if the image type is 0 as well. Set to actual image length if the image type is other value.
0x24	4	imageType	Image type <ul style="list-style-type: none"> • 0x0000: Normal image for unsecure boot • 0x0001: Plain signed Image • 0x0002: Plain CRC Image • 0x0004: Plain signed XIP Image • 0x0005: Plain CRC XIP Image • 0x8001: Signed plain Image with KeyStore Included.
0x28	4	offsetToSpecificHeader	Offset to specific header It means offset to certificate block header if the image type is 0x01, 0x04, or 0x8001. It means the <code>crcChecksum</code> if the image type is 0x02 or 0x05.
0x2c	8	Vector table	Cortex-M33 Vector table entries.
0x34	4	imageExecutionAddress	The execution address of the image. Set to 0 if image type is 0. Set to actual image execution address if the image type is other value.
0x38	8	Vector table	Cortex-M33 Vector table entries.

5.2 How to use ELFToSB-GUI tool to generate firmware with CRC check

ELFToSB-GUI tool is in AN13183SW. The tool's path is `mcu-boot\bin\Tools`. Once opening the `elftosb-gui(win)` executable file, select LPC55xx in **Select target device**.

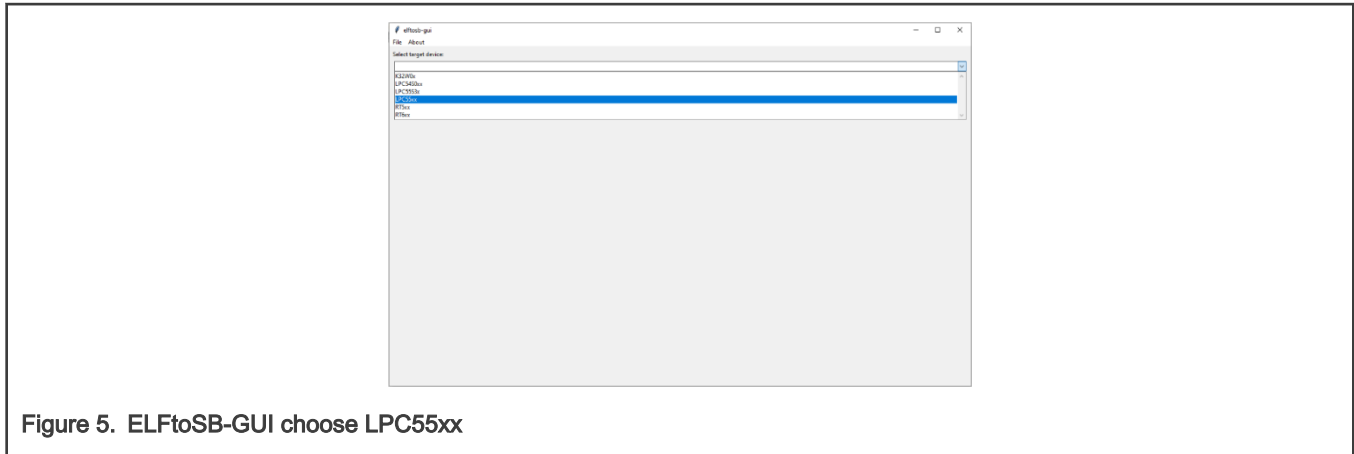


Figure 5. ELFToSB-GUI choose LPC55xx

As shown in [Figure 6](#), set up the configurations as below:

- For **Image Configuration**, click **New**.
- In the **Input** pane:
 - For ***Image file**, select the native *bin* file.
 - Use the default **0** for ***Load address 0x**.
- In the **Define output image format** pane:
 - For **Image execution target**, select **Internal flash (XIP)**.
 - For **Image authentication type**, select **CRC**.
- For **TrustZone image type**, select **TZ-M Disabled**.
- In the **Output** pane:
 - Modify the path and name the output bin file for **Master Boot**

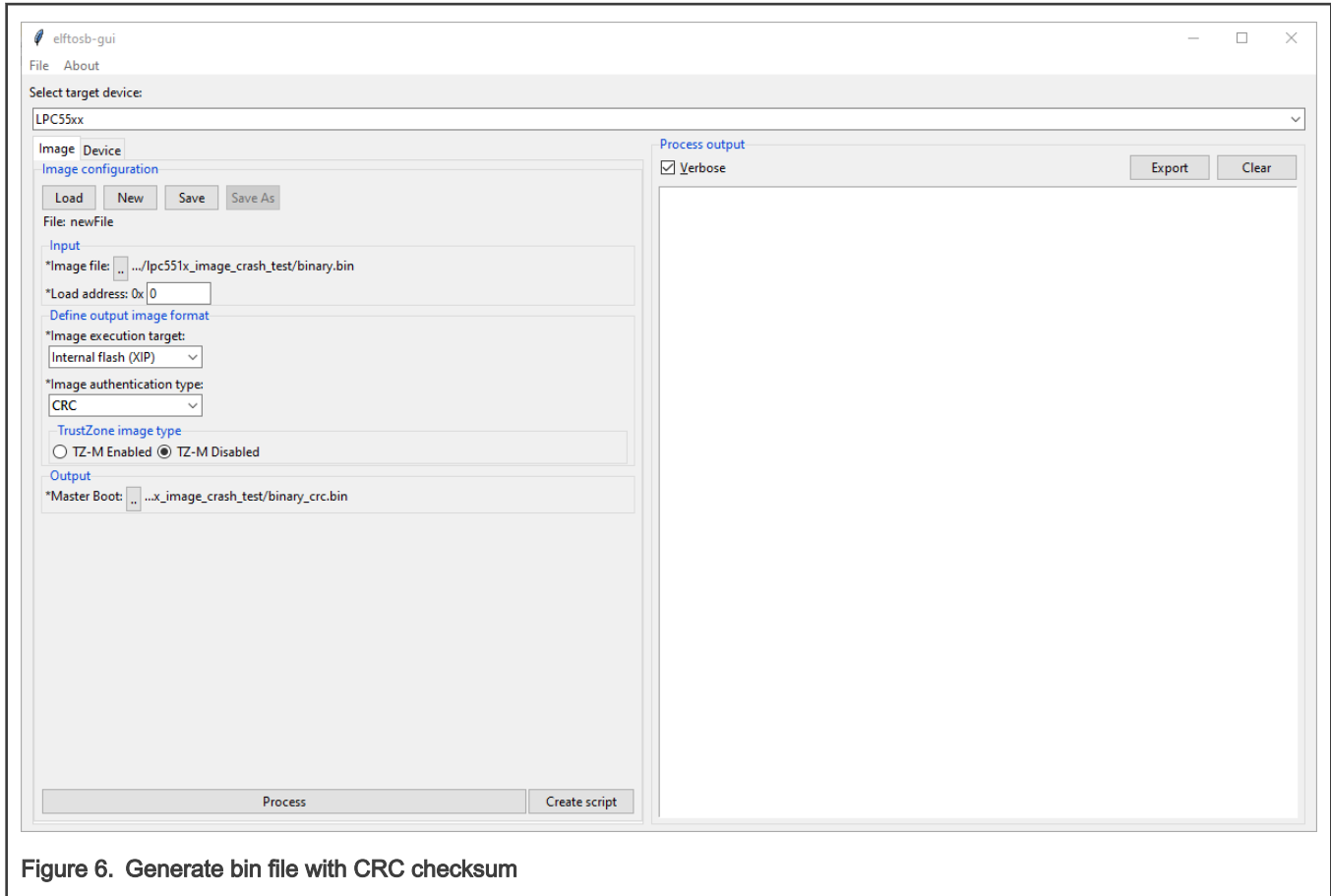


Figure 6. Generate bin file with CRC checksum

After setting correctly, click the **Process** button to generate the firmware with CRC enabled.

5.3 Enter USB HID ISP boot mode after damaged the firmware with CRC

After updating a CRC enabled firmware to LPC55S16-EVK according to the method in [Update new firmware through USB port 1](#), the green LED is flashing under normal conditions. At this time, press the **SW1** button and this application code will erase the 512-byte content starting from `0x30000`. The illusion that the firmware in on-chip flash is damaged or changed some bit. After this actions, it depends on whether to reset the MCU through reset button. Re-power off/on the EVK or use `NVIC_SystemReset()` to reset the MCU, and the MCU will re-enter ROM ISP function. If the USB1 interface(J4) on EVK is connected with PC, then the system will enter USB HID ISP update mode.

6 KEIL, IAR and MCUXpresso generate bin Format Firmware file

6.1 KEIL IDE generate bin file

In the configuration options window of the KEIL project, select the **User** tab. In **Run #1** of the **After Build/Rebuild** column, fill in `xxx\ARM\ARMCLANG\bin\fromelf.exe -bin -o ./binary.bin./output/@L.axf`. Here, `xxx` is the installation path of KEIL. In this way, the IDE will automatically generate a bin file after each compilation.

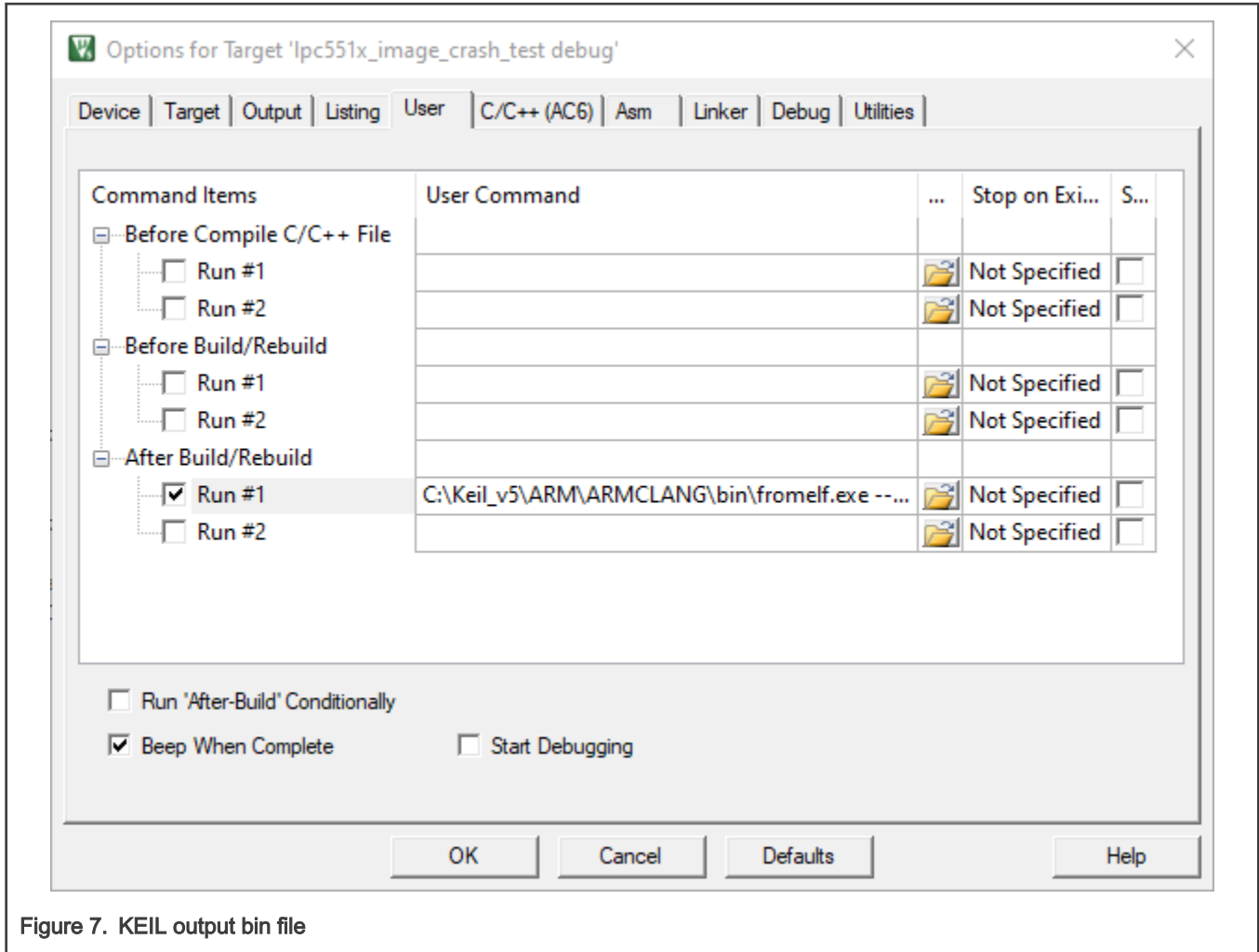


Figure 7. KEIL output bin file

6.2 IAR IDE output bin file

In the project configuration options, select **Output Converter**. Select **Generate additional output** in the tab and select Raw binary in **Output format**.

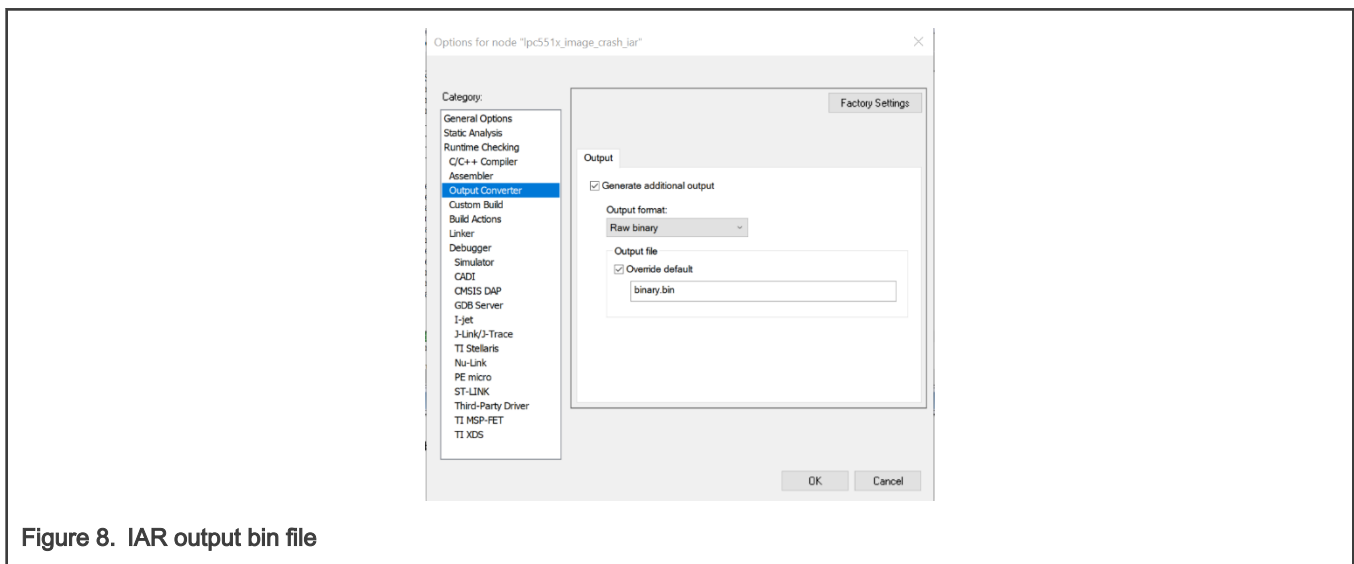


Figure 8. IAR output bin file

6.3 MCUXpresso IDE output bin file

In the output folder after compiled by MCUXpresso, find the corresponding *axffile* and right click. Select **Binary Utilities -> Create Binary**, and the *.bin* file can be automatically generated.

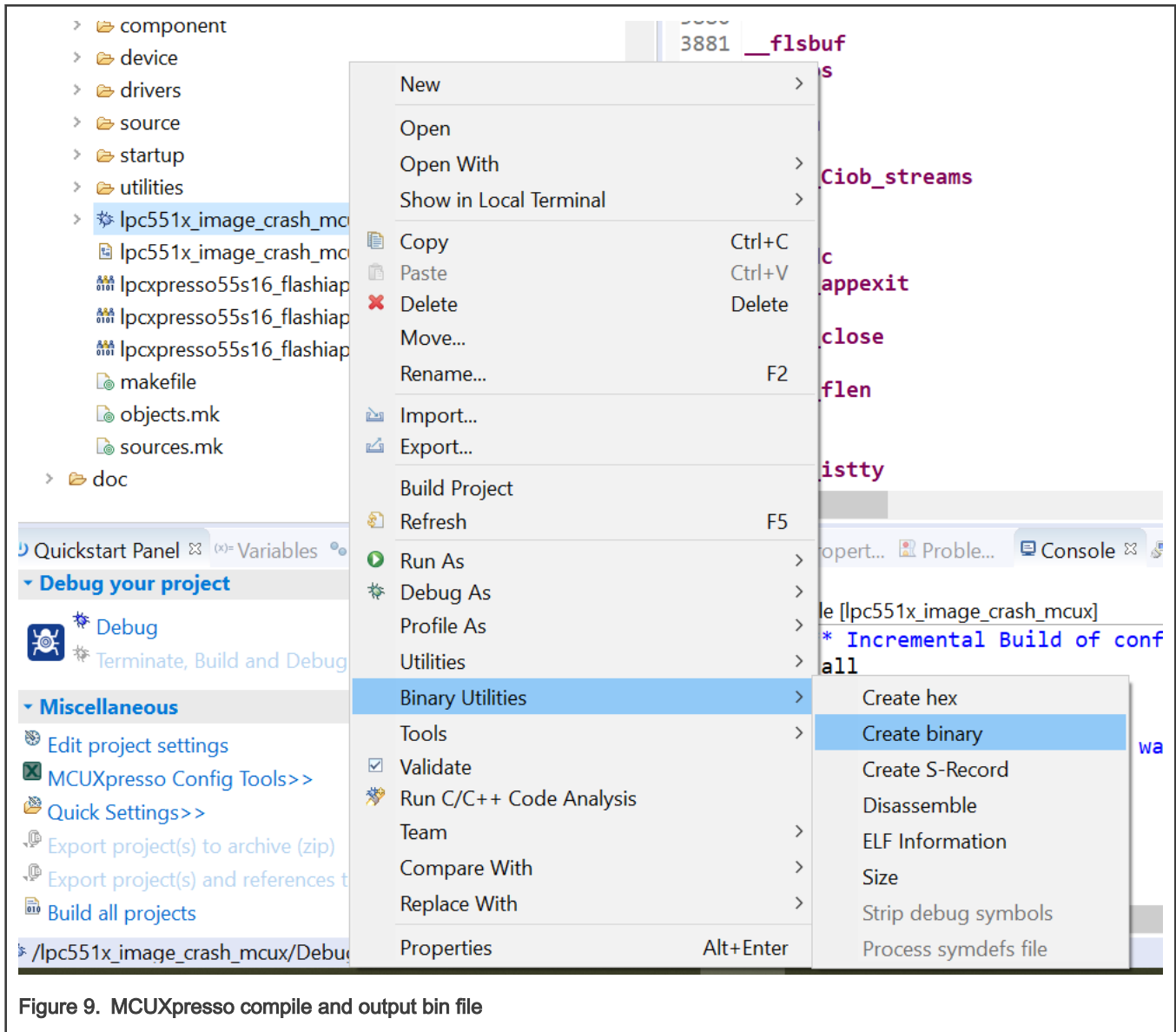


Figure 9. MCUXpresso compile and output bin file

7 Conclusion

The un-secure boot ROM startup process for LPC55(S)1x chip can meet most firmware update functions. It supports CRC check of the firmware and checks the integrity of the on-chip firmware through CRC checksum value. If the on-chip flash's firmware with CRC function enabled has changed a little, the MCU will automatically enter the ROM boot upgrade program mode after resetting or restarting the chip.

8 Reference

- *LPC55S1x/LPC551x User Manual* (document [UM11295](#))

How To Reach Us

Home Page:

nxp.com

Web Support:

nxp.com/support

Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in NXP data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including "typicals," must be validated for each customer application by customer's technical experts. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address: nxp.com/SalesTermsandConditions.

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.

Security — Customer understands that all NXP products may be subject to unidentified or documented vulnerabilities. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer's applications and products. Customer's responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer's applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately. Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP. NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

NXP, the NXP logo, NXP SECURE CONNECTIONS FOR A SMARTER WORLD, COOLFLUX, EMBRACE, GREENCHIP, HITAG, ICODE, JCOP, LIFE, VIBES, MIFARE, MIFARE CLASSIC, MIFARE DESFire, MIFARE PLUS, MIFARE FLEX, MANTIS, MIFARE ULTRALIGHT, MIFARE4MOBILE, MIGLO, NTAG, ROADLINK, SMARTLX, SMARTMX, STARPLUG, TOPFET, TRENCHMOS, UCODE, Freescale, the Freescale logo, Altivec, CodeWarrior, ColdFire, ColdFire+, the Energy Efficient Solutions logo, Kinetis, Layerscape, MagniV, mobileGT, PEG, PowerQUICC, Processor Expert, QorIQ, QorIQ Qonverge, SafeAssure, the SafeAssure logo, StarCore, Symphony, VortiQa, Vybrid, Airfast, BeeKit, BeeStack, CoreNet, Flexis, MXC, Platform in a Package, QUICC Engine, Tower, TurboLink, EdgeScale, EdgeLock, eIQ, and Immersive3D are trademarks of NXP B.V. All other product or service names are the property of their respective owners. AMBA, Arm, Arm7, Arm7TDMI, Arm9, Arm11, Artisan, big.LITTLE, Cordio, CoreLink, CoreSight, Cortex, DesignStart, DynamIQ, Jazelle, Keil, Mali, Mbed, Mbed Enabled, NEON, POP, RealView, SecurCore, Socrates, Thumb, TrustZone, ULINK, ULINK2, ULINK-ME, ULINK-PLUS, ULINKpro, μ Vision, Versatile are trademarks or registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere. The related technology may be protected by any or all of patents, copyrights, designs and trade secrets. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. The Power Architecture and Power.org word marks and the Power and Power.org logos and related marks are trademarks and service marks licensed by Power.org.

© NXP B.V. 2021.

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: March 15, 2021

Document identifier: AN13183

