Application Note

Document No.: AN1131

G32R501 Tool User Manual

Version: V1.0

1 Introduction

Due to the chip characteristics of the G32R5xx series MCU, this chip requires data stream settings for some generated files before specific routines. For example, data stream of flashapi_ex2_uart_kernel. “geehy_tool” will help you generate the required data stream format to ensure normal operation of the program.

Please read this instruction document before officially using the “geehy_tool” tool. Operating environment of the tool:

- Windows 10/11 system

Contents

2 Instructions for Use of Geehy_Bin

2.1 Commands Supported by Geehy_Bin.exe

-h: Display the help information.

-v: Display detailed log information.

-boot: Convert all segments into bootable form.

-uart8: Specify the UART format, 8-bit mode.

- spi8: Specify the SPI format, 8-bit mode.

-lospcp=value: Specify the initial value of the lospcp register.

-spibrr=value: Specify the initial value of the spibrr register.

-i2c8: Specify the I2C format, 8-bit mode.

-i2cpsc=value: Specify the initial value of the i2cpsc register

-i2cclkh=value: Specify the initial value of the i2cclkh register

-i2cclkl=value: Specify the initial value of the i2cclkl register

pc=value: Specify the value of PC.

-a: Select ASCII hexadecimal.

-o: Specify the output file.

-m: Select the primary secure boot mode.

-e: Choose the extended secure boot mode.

-me: Select the primary secure boot and extended secure boot modes.

- mesta=: If the primary and extended boot modes are selected, select the start address of the extended boot.

- melen=: If the primary and extended secure boot modes are selected, extend the flash length of the secure boot.

-- cmac=file: Specify the txt file for storing the CMAC key.

xxx.cmd: Specify the CMD file for storing the start address of CMAC calculation, checked flash length and filled value.

-tag=value: Specify the start address for placing the tag value.

-load=value: Specify the start address for downloading the file.

--load_image: Select to generate BIN file.

--image: Select to generate TXT file.

2.2 Use Examples

This section will introduce how to use the Geehy_Bin.exe tool to perform some common command operations. These examples will help you understand how to convert binary files into different formats, configure boot parameters, and implement such functions as secure boot mode. Through the following example, users can understand basic usage of the tool, and modify and apply it according to actual needs.

2.2.1 UART_Example

This section will introduce how to use the Geehy_Bin.exe to convert files to UART 8-bit mode. Each example demonstrates how to set different PC values and output the results in ASCII hexadecimal format.

- Convert the flashapi_ex2_uart_kernel_CPU0_RAM.bin file into bootable UART 8-bit mode format, set the initial PC value to 0x00000000, output in ASCII hexadecimal format, and save the results to flashapi_ex2_uart_kernel_CPU0_RAM.txt.

Geehy_Bin.exe -boot -uart8 -pc=0x00000000 -a -o flashapi_ex2_uart_kernel_CPU0_RAM.txt flashapi_ex2_uart_kernel_CPU0_RAM.bin

- Similar to the previous instruction, except that the initial value of the PC is set to 0x08000000, and the file of flashapi_ex2_uart_kernel_BANK0_LDFU.txt is output.

Geehy_Bin.exe -boot -uart8 -pc=0x08000000 -a -o flashapi_ex2_uart_kernel_BANK0_LDFU.txt flashapi_ex2_uart_kernel

- Perform operations similar to the previous two, except that the initial value of the PC is set to 0x08020000, and the file of flashapi_ex2_uart_kernel_BANK1_LDFU.txt is output.

Geehy_Bin.exe -boot -uart8 -pc=0x08020000 -a -o flashapi_ex2_uart_kernel_BANK1_LDFU.txt flashapi_ex2_uart_kernel_BANK1_LDFU.bin

2.2.2 SPI_Example

Convert the project.bin file to bootable SPI 8-bit mode format, lospcp to 2, spibrr to 127; set the initial value of PC to 0x08000000, output in ASCII hexadecimal format, and output to the project.txt.

Geehy_Bin.exe -boot -spi8 -lospcp=0x02 -spibrr=0x7F -pc=0x08000000 -a project.bin -o project.txt

2.2.3 I2C_Example

Convert the project.bin file to bootable I2C 8-bit mode format, i2cpsc to 4, i2cclkh to 45, i2cclkh to 45; set the initial value of PC to 0x08000000, output in ASCII hexadecimal format, and output to the project.txt.

Geehy_Bin.exe –boot –i2c8 -i2cpsc=0x04 -i2cclkh=0x2d -i2cclkl=0x2d -pc=0x08000000 -a project.bin -o project.txt

2.2.4 Secure_Example

Some examples of secure boot mode are introduced below. Through commands, files can be encrypted using CMAC keys and the format of generated files can be specified (e.g. txt or bin). The following examples show different application scenarios of the primary and extended secure boot modes, and how to configure tag values and download the start address.

- Use the TEST.cmd file to specify the start address and other related parameters for CMAC calculation. It selects the primary secure boot mode, generates the txt file format, processes user.bin using the CMAC key in key.txt, places the tag value at the address 0x08000008, and finally outputs to *test_major_cmac.txt*.

Geehy_Bin.exe test_major.cmd -m --image --cmac=key.txt user*.bin -tag=0x08000008 -o test_major_cmac.txt*

- Similar to the previous instruction, but a bin file format is generated and the result is saved to the CMAC.bin.

Geehy_Bin.exe test_major.cmd -m --load_image --cmac=key.txt user.bin -tag=0x08000008 -o test_major_cmac.bin

- Use the extended secure boot mode to generate the txt file format, using the CMAC key in key.txt. It processes user.bin, places the tag value at the address 0x08003000, downloads files starting from the address 0x08000000, and finally outputs them to *test_extend_cmac.txt*.

Geehy_Bin.exe test_extend.cmd -e --image --cmac=key.txt user.bin -tag=0x08003000 -load=0x08000000 -o test_extend_cmac.txt

- Similar to the previous instruction, but a bin file format is generated and the result is saved to the *test_extend_cmac.bin*.

Geehy_Bin.exe test_extend.cmd -e --load_image --cmac=key.txt user.bin -tag=0x08003000 -load=0x08000000 -o test_extend_cmac.bin

3 Geehy_Serial_flash_Programmer

This chapter will introduce how to use Geehy_Serial_flash_Programmer.exe and related commands. This tool is mainly used for programming devices through serial ports.

3.1 Commands Supported by Geehy_Serial_flash_Programmer.exe

-d <device>: The name of loaded device. e.g.: r501.

-k <file>: The file name of the flash core. This file must be in UART boot format.

-a <file>: The file name used for downloading. This file must be in UART boot format.

-p COM<num>: Set the COM port used for communication.

-b <num>: Set the baud rate of the COM port.

-? Or -h: Display help information.

-q: Quiet mode. Disable output to stdout.

-w: Wait for the button before exiting.

-v: Enable detailed output.

Note: -d, -f and -p are required parameters. If the baud rate is omitted, communication will be made at a baud of 9600.

3.2 Use Examples

There are two different versions of the upper computer in the G32R5xx_SDK\utilities\geehy_tool path, in which Geehy_Serial_flash_Programmer.exe and MCU BOOTROM are used together, but the programs cannot be burnt. Geehy_Serial_flash_Programmer_appln.exe and uart_flash_kernel program are used together. Some examples using the command line tools are provided below. These examples demonstrate how to set the device, specify flash core files, download files, and configure the communication port and baud rate.

- How to use Geehy_Serial_flash_Programmer.exe to program the device r501. We have set the flash core file and download file name, and specified the use of COM11 port for communication with a baud rate of 38400.

Geehy_Serial_flash_Programmer.exe -d r501 -k flashapi_ex2_uart_kernel_CPU0_RAM.txt -a LED1_0x0800C000.txt -p COM11 -b 38400

Geehy_Serial_flash_Programmer_appln.exe -d r501 -k flashapi_ex2_uart_kernel_CPU0_RAM.txt -a LED2_0x0800C000.txt -p COM11 -b 38400

4 Revision

Table 1 Document Revision History

Date	Version	Change History
January, 2025	1.0	New

Statement

This document is formulated and published by Geehy Semiconductor Co., Ltd. (hereinafter referred to as “Geehy”). The contents in this document are protected by laws and regulations of trademark, copyright and software copyright. Geehy reserves the right to make corrections and modifications to this document at any time. Read this document carefully before using Geehy products. Once you use the Geehy product, it means that you (hereinafter referred to as the “users”) have known and accepted all the contents of this document. Users shall use the Geehy product in accordance with relevant laws and regulations and the requirements of this document.

1. Ownership

This document can only be used in connection with the corresponding chip products or software products provided by Geehy. Without the prior permission of Geehy, no unit or individual may copy, transcribe, modify, edit or disseminate all or part of the contents of this document for any reason or in any form.

The “极海” or “Geehy” words or graphics with “®” or “™” in this document are trademarks of Geehy. Other product or service names displayed on Geehy products are the property of their respective owners.

2. No Intellectual Property License

Geehy owns all rights, ownership and intellectual property rights involved in this document.

Geehy shall not be deemed to grant the license or right of any intellectual property to users explicitly or implicitly due to the sale or distribution of Geehy products or this document.

If any third party’s products, services or intellectual property are involved in this document, it shall not be deemed that Geehy authorizes users to use the aforesaid third party’s products, services or intellectual property. Any information regarding the application of the product, Geehy hereby disclaims any and all warranties and liabilities of any kind, including without limitation warranties of non-infringement of intellectual property rights of any third party, unless otherwise agreed in sales order or sales contract.

3. Version Update

Users can obtain the latest document of the corresponding models when ordering Geehy products.

If the contents in this document are inconsistent with Geehy products, the agreement in the sales order or the sales contract shall prevail.

4. Information Reliability

The relevant data in this document are obtained from batch test by Geehy Laboratory or cooperative third-party testing organization. However, clerical errors in correction or errors caused by differences in testing environment may occur inevitably. Therefore, users should understand that Geehy does not bear any responsibility for such errors that may occur in this document. The relevant data in this document are only used to guide users as performance parameter reference and do not constitute Geehy's guarantee for any product performance.

Users shall select appropriate Geehy products according to their own needs, and effectively verify and test the applicability of Geehy products to confirm that Geehy products meet their own needs, corresponding standards, safety or other reliability requirements. If losses are caused to users due to user's failure to fully verify and test Geehy products, Geehy will not bear any responsibility.

5. Legality

USERS SHALL ABIDE BY ALL APPLICABLE LOCAL LAWS AND REGULATIONS WHEN USING THIS DOCUMENT AND THE MATCHING GEEHY PRODUCTS. USERS SHALL UNDERSTAND THAT THE PRODUCTS MAY BE RESTRICTED BY THE EXPORT, RE-EXPORT OR OTHER LAWS OF THE COUNTRIES OF THE PRODUCTS SUPPLIERS, GEEHY, GEEHY DISTRIBUTORS AND USERS. USERS (ON BEHALF OR ITSELF, SUBSIDIARIES AND AFFILIATED ENTERPRISES) SHALL AGREE AND PROMISE TO ABIDE BY ALL APPLICABLE LAWS AND REGULATIONS ON THE EXPORT AND RE-EXPORT OF GEEHY PRODUCTS AND/OR TECHNOLOGIES AND DIRECT PRODUCTS.

6. Disclaimer of Warranty

THIS DOCUMENT IS PROVIDED BY GEEHY "AS IS" AND THERE IS NO WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, TO THE EXTENT PERMITTED BY APPLICABLE LAW.

GEEHY'S PRODUCTS ARE NOT DESIGNED, AUTHORIZED, OR WARRANTED FOR USE AS CRITICAL COMPONENTS IN MILITARY, LIFE-SUPPORT, POLLUTION CONTROL, OR HAZARDOUS SUBSTANCES MANAGEMENT SYSTEMS, NOR WHERE FAILURE COULD RESULT IN INJURY, DEATH, PROPERTY OR ENVIRONMENTAL DAMAGE.

IF THE PRODUCT IS NOT LABELED AS "AUTOMOTIVE GRADE," IT SHOULD NOT BE CONSIDERED SUITABLE FOR AUTOMOTIVE APPLICATIONS. GEEHY ASSUMES NO LIABILITY FOR THE USE BEYOND ITS SPECIFICATIONS OR GUIDELINES.

THE USER SHOULD ENSURE THAT THE APPLICATION OF THE PRODUCTS COMPLIES WITH ALL RELEVANT STANDARDS, INCLUDING BUT NOT LIMITED TO SAFETY, INFORMATION SECURITY, AND ENVIRONMENTAL REQUIREMENTS. THE USER ASSUMES FULL RESPONSIBILITY FOR THE SELECTION AND USE OF GEEHY PRODUCTS. GEEHY WILL BEAR NO RESPONSIBILITY FOR ANY DISPUTES ARISING FROM THE SUBSEQUENT DESIGN OR USE BY USERS.

7. Limitation of Liability

IN NO EVENT, UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL GEEHY OR ANY OTHER PARTY WHO PROVIDES THE DOCUMENT AND PRODUCTS "AS IS", BE LIABLE FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, DIRECT, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE DOCUMENT AND PRODUCTS (INCLUDING BUT NOT LIMITED TO LOSSES OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY USERS OR THIRD PARTIES). THIS COVERS POTENTIAL DAMAGES TO PERSONAL SAFETY, PROPERTY, OR THE ENVIRONMENT, FOR WHICH GEEHY WILL NOT BE RESPONSIBLE.

8. Scope of Application

The information in this document replaces the information provided in all previous versions of the document.