Application Note

Document No.: AN1125

G32R501 SDK Quick Start Guide

Version: V1.1

1 Introduction

G32R5xx SDK (Software Development Kit) is a set of comprehensive development software and documents designed especially for the G32R5xx series real-time control MCU, for the purpose of minimizing software development time. From device-specific drivers and libraries to peripheral examples, the G32R5xx SDK provides a solid foundation for starting to develop and evaluate products.

The materials mentioned in this application note can be obtained and downloaded from www.geehy.com.

To quickly evaluate the G32R5xx SDK, you may need to prepare the following environment or content:

1. Windows10/11
2. MDK-ARM v5.40 or newer version
3. IAR EW for Arm 9.60.2 or newer version
4. Eclipse 4.35 or newer version
5. LLVM-ET-Arm 19.1.1 or newer version
6. arm-gnu-toolchain 14.2 or newer version
7. G32R5 EVAL board

Note: The SDK containing Eclipse projects is available in version V1.1.0 and later. For inquiries, please contact Geehy official to get it (Click the link: https://docs.google.com/forms/d/1rU47O-eqIbkfPWIivAiHErBF-7sId8JRhNkdJNGYVqE/edit).

Contents

2 SDK Structure

The G32R5xx SDK has the following hierarchical structure:

SDK directory

- boards
- device_support
- docs
- driverlib
- examples
- libraries
- package
- utilities

3 Content About boards

Note: The boards content described in this chapter is located in the following positions of G32R5xx_SDK:

G32R5xx_SDK_VERSION#/boards

This directory includes the hardware design schematic diagram for G32R5xx series boards.

4 Content About device_support

Note: The device_support content described in this chapter is located in the following position on G32R5xx_SDK:

G32R5xx_SDK_VERSION#/device_support

This directory includes all device-specific support files (including core files from Arm), bit field headers, and device development user guide.

1. CMSIS (Cortex Microcontroller Software Interface Standard) folder includes

   header files related to the Arm® Cortex®-M52 core, which define the registers and functions of the core.

2. The Geehy folder includes files specific to the G32R501 device. Wherein:

   - g32r501.h: The device header file, which defines the registers of the device.
   - startup_g32r501.c: The startup file, which includes the reset vector table and startup code.
   - The system_eval folder includes the files related to system initialization:
     - system_g32r501.h and system_g32r501.c: System initialization and configuration.
     - system_g32r501_dwtmeasure.h: Measurement functions related to DWT (Data Watchpoint and Trace).

3. The include folder includes various device driver and definition files:

   - device.h, driverlib.h, etc.: General device and driver library header files.
   - g32r501_adc_defines.h, g32r501_dma_defines.h, etc.: The definition file for specific modules, which includes registers and macros related to ADC, DMA, GPIO, etc.

4. The icf folder contains linker scripts, which are suitable for IAR EW for Arm.

5. The sct folder contains the linker scripts and the document:

   - doc: It includes the document *AN1130_G32R501 Keil Linker File Application Note*, about the ".sct" linker script file.
   - *.sct: Linker script file, used for configuring the memory layout and mapping of G32R5xx series MCU.

6. examples: Example code containing various peripherals, demonstrating how to use various peripheral functions of G32R5xx. Please refer to Chapter 7 for the documents of routines.

7. Headers: The header file directory containing the device support library, which defines the peripheral interfaces and functions of the device.

5 Content About docs

Note: The docs content described in this chapter is located in the following positions of G32R5xx_SDK:

G32R5xx_SDK_VERSION#/docs

This directory contains the *G32R5xx SDK User Guide* and all software package documents.

6 Content About driverlib

Note: The driverlib content described in this chapter is located in the following positions of G32R5xx_SDK:

G32R5xx_SDK_VERSION#/driverlib

This directory contains the driver libraries for specific device and examples of peripherals based on drivers.

- boards_support:
  - eval: It contains supporting files related to the eval development board, e.g. board-level header files and source codes.
- docs.lnk: Document link, pointing to relevant development documents.
- driverlib:
  - adc.c / adc.h: Driver code and header file related to ADC (analog-to-digital converter).
  - as.c / as.h: Driver code and header file related to AS (analog subsystem).
  - can.c / can.h: Driver code and header file related to CAN (controller area network).
  - cap.c / cap.h: Driver code and header file related to CAP (Catcher).
  - comp.c / comp.h: Driver code and header file related to the comparator.
  - cortex.c / cortex.h: Driver code and header file related to the Cortex core.
  - Cpu.h: Header file related to CPU.
  - dac.c / dac.h: Driver code and header file related to DAC (digital-to-analog converter).
  - dbgmcu.c / dbgmcu.h: Driver code and header file related to debugging MCU.
  - dccomp.c / dccomp.h: Driver code and header file related to DCCOMP (dual-clock comparison).
  - dcs.c / dcs.h: Driver code and header file related to DCS (dual-code security).
  - debug.h: Debug-related header file.
  - dma.c / dma.h: Driver code and header file related to DMA (direct memory access).
  - driver_ex.h: Header file for driver extension.
  - exti.c / exti.h: Driver code and header file related to external interrupts.
  - flash.c / flash.h: Driver code and header file related to flash.
  - gpio.c / gpio.h: Driver code and header file related to GPIO (general-purpose input/output).

- hrcap.c / hrcap.h: Driver codes and header file related to HRCAP (pulse width modulator).
- hrpwm.c / hrpwm.h: Driver code and header file related to HRPWM (high-resolution pulse width modulation).
- i2c.c / i2c.h: Driver code and header file related to I2C (inter-integrated circuit communication).
- inc: Includes the header file of hardware registers.
- interrupt.c / interrupt.h: Driver code and header file related to interrupts.
- ipc.c / ipc.h: Driver code and header file related to IPC (dual-core communication).
- lin.c / lin.h: Driver code and header file related to LIN (local interconnect network).
- pin_map.h / pin_map_legacy.h: Header file related to pin mapping.
- pmbus.c / pmbus.h / pmbus_common.h: Driver code and header file related to PMBus (power management bus).
- pwm.c / pwm.h: Driver codes and header file related to PWM (pulse width modulation).
- qep.c / qep.h: Driver code and header file related to QEP (quadrature encoder pulse).
- sdf.c / sdf.h: Driver code and header files related to SDF (signal definition file).
- spi.c / spi.h: Driver code and header file related to SPI (serial peripheral interface).
- sysctl.c / sysctl.h: Driver code and header file related to system control.
- tmr.c / tmr.h: Driver code and header file related to timers.
- uart.c / uart.h: Driver code and header file related to UART (universal asynchronous receiver/transmitter).
- version.c / version.h: Driver code and header file related to version information.
- xbar.c / xbar.h: Driver code and header file related to X-BAR.
- zidian_cde.h / zidian_math.h: Header file related to dictionary.

- examples:
  - eval: Example code containing various peripherals and functional modules, demonstrating how to use various functions of G32R501. Please refer to Chapter 7 for the documents of routines.

7 Content About examples

Note: The examples content described in this chapter is located in the following positions of G32R5xx_SDK:

G32R5xx_SDK_VERSION#/examples

This directory contains examples involving multiple devices/peripherals to demonstrate different applications. Before officially using examples, you need to complete the following preparation:

1. For description of program examples, please refer to the *AN1127_G32R501 SDK Examples User Manual*.

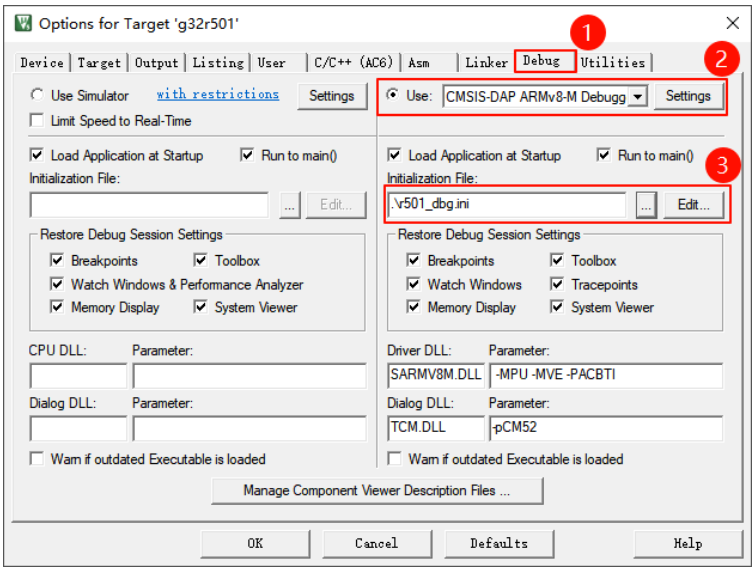
2. To run examples, please complete the following work:

   a) Software environment preparation, installation of necessary development tools. e.g. compilers, debuggers, and IDE (integrated development environment).

   b) Hardware environment preparation, with corresponding simulation debugging tools (e.g. GEEHY-LINK, J-Link), related boards or G32R5xx series MCU controller boards with programs that can run normally.

   c) In MDK environment, when you need to simulate, please compile the program once and select the appropriate simulator and simulation debugging script under "Option for Target XXX" → "Debug" (following the steps in Figure 1**).**

Figure 1 Select Suitable Simulator and Simulation Script



8 Content About libraries

Note: The libraries content described in this chapter is located in the following positions of G32R5xx_SDK:

G32R5xx_SDK_VERSION#/libraries/

This directory contains specific devices and core libraries for users to use in the development process. These library files are classified and stored in different subdirectories, e.g. DSP library, math library, Flash API library, etc. The following is an introduction to each library and its subdirectories.

8.1 Calibration

This directory contains the hrpwm library. This library provides a set of tools and methods for high-precision PWM (HRPWM) calibration. By using the SFO (Self-calibrating Frequency Oscillator) algorithm, automatic calibration can be realized to ensure high precision and stability of PWM signals.

For the use of SFO library functions, please refer to the *AN1134_G32R501 SDK SFO Library Function Application Note*.

8.2 DSP

This directory contains three main library files, namely the Fixed Point library, FPU library, and VCU library. These library files provide rich DSP (digital signal processing) functions, suitable for various fixed-point, floating-point, and hardware acceleration DSP operations.

8.2.1 Fixed Point

This directory contains the Fixed Point library, which is used for fixed-point DSP operations. It provides a set of tools and methods for fixed-point DSP operations. These operations include fast Fourier transform (FFT), finite impulse response (FIR) filtering, and infinite impulse response (IIR) filtering. This library is suitable for scenarios that require fixed-point arithmetic, for example in resource-limited embedded systems, or in situations where high precision is required but floating-point arithmetic capability is limited.

8.2.2 FPU

This directory contains the FPU library, which is used for floating-point DSP operations. It provides a set of tools and methods specially for floating-point DSP operations, including fast Fourier transform (FFT), finite impulse response (FIR) filtering, infinite impulse response (IIR) filtering, and vector operations. This library is particularly suitable for application scenarios that require high computational accuracy and performance, e.g. signal processing and real-time control systems. Use of FPU can significantly improve operational speed and efficiency.

8.2.3 VCU

This directory contains the VCU library, which is a library for DSP operations using VCU hardware accelerators. It provides a set of DSP operation tools and methods for VCU (Vector Control Unit) hardware accelerators, which improves the execution efficiency of algorithms through hardware acceleration. These operations include fast Fourier transform (FFT), cyclic redundancy check (CRC), etc. This library is suitable for DSP operation scenarios that require efficient hardware acceleration. By utilizing the VCU (Vector Control Unit) hardware accelerators, the execution efficiency of algorithms can be significantly improved, so it is suitable for high-performance applications such as motor control and signal processing.

8.3 FLASH API

This directory contains the FLASH API library, which is used for operating Flash storage areas on G32R5 series MCU. It provides a set of API for Flash storage operations, especially for G32R5 series MCU. Through these APIs, users can easily perform operations such as reading, writing, and erasing of Flash. This library is suitable for scenarios that require Flash storage area operations, such as firmware upgrade, data storage, and reading in embedded systems. These APIs provide support for reading, writing, erasing, and other operations on Flash storage areas.

8.4 math

8.4.1 Fix32math

This directory contains the Fix32mat library, which is used for fixed-point mathematical operations. It provides a set of tools and methods for fixed-point mathematical operations. These operations include basic arithmetic operations, trigonometric functions, logarithmic and exponential operations, etc. The Fix32math library is suitable for scenarios that require fixed-point mathematical operations, especially in embedded systems where resources are limited but high-precision mathematical operations are required.

8.4.2 FPUfastRTS

This directory contains the FPUfastRTS library, which is used for fast floating-point mathematical operations. It provides a set of tools and methods for fast floating-point mathematical operations. These operations include basic arithmetic operations, trigonometric functions, logarithmic and exponential operations, etc., and use optimization algorithms to improve operational speed. This library is suitable for scenarios that require efficient floating-point mathematical operations, especially in embedded applications that require high computational accuracy and real timeliness.

8.5 secure_rom_apis

This directory contains the Secure ROM APIs library, which is used for secure area codes. It

provides a set of ROM API for use in secure area codes. These APIs include secure boot, encryption operation, etc., ensuring the security and integrity of the system. The Secure ROM APIs library is suitable for the codes that need to run in secure areas, e.g. application scenarios involving secure boot, encryption operation, and system protection.

9 Content About utilities

Note: The utilities content described in this chapter is located in the following positions of G32R5xx_SDK:

G32R5xx_SDK_VERSION#/utilities

This directory contains development tool applications, e.g. flash programmers, Windows drivers, and third-party software. This directory contains the following files:

8. geehy_tool/: Includes CMAC tool, e.g. Geehy_Bin.exe. Used for users to generate firmware containing CMAC checksum values. For the use methods, please refer to the *AN1131_G32R501 Tool User Manual*.

9. keil_dbg_tool/: Includes Debug tools of MDK-ARM IDE: keil_dbg_tool.exe. Used by user for simulation in MDK. For the use methods, please refer to the *AN1129_G32R501 Keil Debug Tool User Manual*.

10. G32R5xx_AddOn/: Includes chip support installation package for IAR EW for Arm IDE, used by users to add G32R5 series MCU chip selection in IAR EW for Arm IDE. For the use methods, please refer to Section 3.3 in *Instructions for Use of AN1126_G32R501 IDE and Tool Chain*.

10 Revision

Table 1 Document Revision History

Date	Version	Change History
January 2025	1.0	New
April 2025	1.1	Errata description.

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.