Application Note

Document No.: AN1181

G32A14xx

Low-Power Debugging Notes

Version: V1.0

1 Introduction

This document describes the low-power system architecture of the G32A14xx series automotive-grade MCUs. It explains the chip power modes, hardware and software design considerations, practical debugging experience, example use cases, and current measurement results.

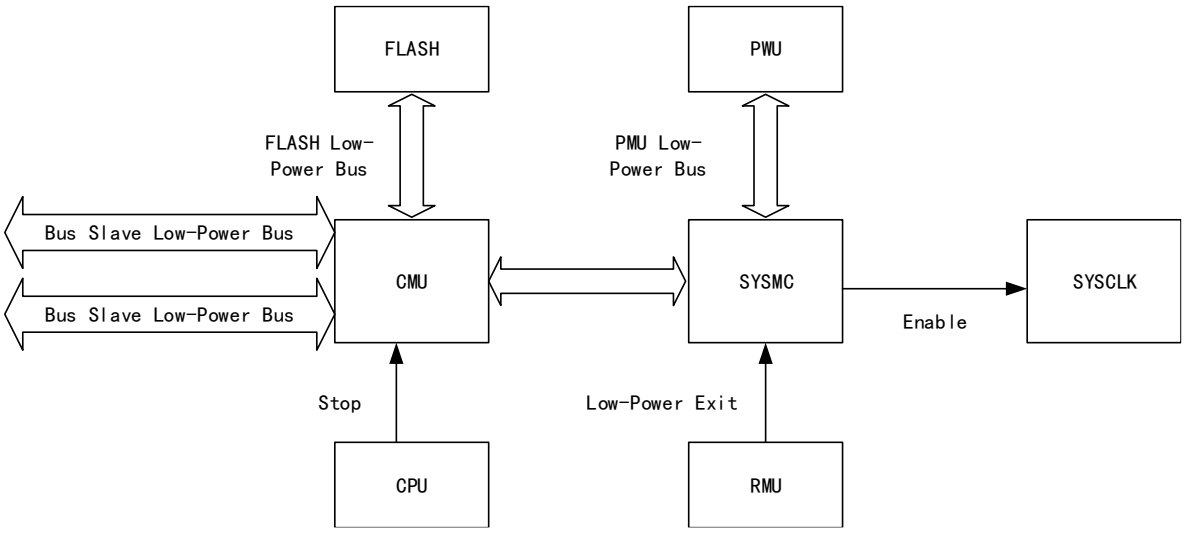
Content

2 Low-Power System Architecture

The G32A14xx low-power system mainly consists of the PMU (Power Management Unit), CMU (Clock Management Unit), and SYSMC (System Mode Control Unit).

- PMU (Power Management Unit): The power supply is the basis for stable system operation. The operating voltage range is 2.7–5.5V. The PMU integrates a voltage regulator, an LVD (Low-Voltage Detection) system, and Power-On Reset (POR).
- CMU (Clock Management Unit): The system clock sources include SYSOSC, HSI, LSI, LPO, and SYSPLL. For clock-source characteristics, refer to the “Electrical Characteristics” section of the datasheet.
- SYSMC (System Mode Control unit): The SYSMC controls entry into and exit from all low-power modes. It monitors events that trigger power-mode transitions and synchronously controls system power, clocks, and memory in low-power modes to achieve the required power consumption and functionality. The SYSMC can operate in very-low-power modes.

Figure 1 Composition of the Low-Power System



3 Chip Power Mode Descriptions

3.1 CPU Operating Modes

The CPU supports three main operating states: Run, Sleep, and Deep Sleep. The Run power modes include High-Speed Run mode (HSR), Run mode (RUN), and Very-Low-Power Run mode (VLPR). The Sleep power modes include Stop modes (STOP1 and STOP2) and Very-Low-Power Stop mode (VLPS).

Table 1 Power Modes

Power Mode	Functional Description
Stop mode (STOP)	Turns off the core clock. STOP mode includes STOP1 and STOP2. In STOP1, both the system clock and bus clock are turned off. In STOP2, the bus clock is enabled while the system clock is turned off.
Very-Low-Power Stop mode (VLPS)	Turns off the core clock. After all supported peripherals provide valid stop-acknowledge signals, the system clock and bus clock are turned off.
Very-Low-Power Run mode (VLPR)	Limits the maximum frequencies of the core clock, system clock, bus clock, and FLASH clock.
Run mode (RUN)	Allows the MCU to run at full speed with the internal power supply fully regulated and stable.
High-Speed Run mode (HSR)	Allows the MCU to run at a higher frequency than RUN mode, with the internal power supply fully regulated and stable.

Note: During a mode transition, the system clock changes. It is recommended to stop all peripheral communication before switching modes.

3.2 Module Functionality in the Available Power Modes

Table 2 Full Terms, Abbreviations, and Descriptions

Full Name (Chinese)	Full Name (English)	Abbreviation	Description
全部功能	Full functionality	FF	The module is fully functional. In VLPR mode, the system frequency is limited.
异步操作	Async operation	ASYNCO	The module can operate asynchronously if the selected clock source remains enabled.
禁止	Disable	D	The clock source and module are disabled.
电源	Power	P	Memory must remain powered to retain data.
唤醒	Wake-up	AWUP	The module can be used as an MCU wake-up source.
工作模式	Operation mode	-	Supports low-power mode or full-performance mode. Memory must remain powered in low-power mode to retain data.

Table 3 Module Functionality in the Available Power Modes

Module		Stop	VLPS[1]	VLPR	HSR	Run
Core	NVIC	D		FF		
System	SYSMC/PMU	FF				
	Voltage regulator	Enabled	LPM		FPM	
	LVD/LVR	Enabled	LVD disabled/LVR enabled		Enabled	
	Brown-out detection	FF				
	DMA	ASYNCO		FF		
	Watchdog	STOP1: async operation; STOPEN bit must be set. STOP2: FF	Async operation; STOPEN bit must be set. Supports async wake-up	FF; only LPOCLK and LSICLK can be selected as clock sources	FF	
	EWDT	STOP1: static; STOP2: FF	D		FF	
Clock	128KHz LPO	FF				
	PLL	FF	D[2]		FF	
	HSICLK					
	SYSOSC					
	LSICLK	FF				
	SCG	All clock sources available	Only LSICLK available		All clock sources available	
	FLASH clock	FLASH clock disabled in STOP1 and enabled in STOP2	D	$f_{FLASHCLK(VLPR)}$ [3]LSICLK as clock source	$f_{FLASHCLK(HSR)}$ [3]	$f_{FLASHCLK(RUN)}$ [3]
	Core clock	D		$f_{CORECLK(VLPR)}$ [3]LSICLK as clock source	$f_{CORECLK(HSR)}$ [3]	$f_{CORECLK(RUN)}$ [3]
	Bus clock	Disabled in STOP1 and enabled in STOP2	D	$f_{BUSCLK(VLPR)}$ [3]LSICLK as clock source	$f_{BUSCLK(HSR)}$ [3]	$f_{BUSCLK(RUN)}$ [3]

Module		Stop	VLPS[1]	VLPR	HSR	Run
	System clock	D		$f_{SYSCLK\ (VLPR)}$ [3]LSICLK as clock source	$f_{SYSCLK\ (HSR)}$ [3]	$f_{SYSCLK\ (RUN)}$ [3]
Memory & Memory Interface	SRAM_H and SRAM_L	LPM; no read/write operations		LPM; read/write operations supported	FF	
	FLASH memory	LPM; no read, program, or erase operations		Read-only, up to 1MHz. No FLASH commands are available, including CSE commands for CSEc parts	Read-only. No FLASH commands are available, including CSE commands for CSEc parts	FF
	CFGMemory as EEERAM	LPM; no read/write operations		LPM; read/write operations supported. Read-only; no FLASH commands	Read-only; no FLASH commands	FF
	CFGMemory (CFGRAM as conventional SRAM); LPM; read/write operations supported			LPM; read/write operations supported	FF	
	Cache controller					
	ENET	Remote wake-up by magic packet is supported in STOP1 and STOP2	D	FF; only LSICLK can be selected as clock source		
Communication Interfaces	IO		D	FF; only LSICLK can	FF	
	LPI2C		ASYNCO			

Module		Stop	VLPS[1]	VLPR	HSR	Run
	LPSPI	STOP1: async operation; STOP2: FF		be selected as clock source		
	LPUART					
	CAN	CAN0 supports PN in both STOP1 and STOP2	D			
Timers	PDU	D		FF; only LSICLK can be selected as clock source	FF	
	CFMTMR					
	LPTMR	STOP1: async operation; STOP2: FF	ASYNCO			
	RTC		FF			
	LPITMR		ASYNCO	FF; only LPOCLK and LSICLK can be selected	FF	
Analog Peripherals	ADC	Disabled in STOP1; FF in STOP2	D		FF	
	COMP	Low-speed or high-speed comparison mode is FF in STOP2	Only low-speed comparison mode supported			
Security Overview	CRC	Disabled in STOP1 and enabled in STOP2	D	FF		
Human-Machine Interface	GPIO/PM	STOP1: async operation; STOP2: FF	D[4]	FF		

Notes:

(1) The BEN bit must be set to 1 in VLPS/VLPR mode.

(2) Software must disable this module before any mode transition from RUN mode.

(3) For detailed data, refer to the datasheet.

(4) The G32A1445 supports asynchronous wake-up from VLPS mode through GPIO.

3.3 Power Mode State Transitions

Any reset returns the MCU to RUN mode.

Table 4 Power Mode Transitions

Mode Transition	Trigger Method
RUN → HSR	ALLOWHS=1, RUNCTRL=11
HSR → RUN	Reset or RUNCTRL=00
RUN → VLPS	ALLOWLP=1, STOPCTRL=010. Enter immediate sleep or sleep-on-exit mode by setting SLEEPDEEP in the core System Control Register.
VLPS → RUN	Reset, or interrupt after entering VLPS directly from RUN mode.
RUN → VLPR	ALLOWLP=1, RUNCTRL=01. In this mode, the core clock, system clock, bus clock, FLASH clock, and SCG clock modes are limited.
VLPR → RUN	Reset or RUNCTRL=00
VLPR → VLPS	STOPCTRL=000 or STOPCTRL=010. Enter immediate sleep or sleep-on-exit mode by setting SLEEPDEEP in the core System Control Register.
VLPS → VLPR	Interrupt Note: If VLPS mode is entered directly from RUN mode, hardware forces the MCU to return to RUN mode and does not allow a transition to VLPR mode.
RUN → STOP	RUNCTRL=00, STOPCTRL=000. Enter immediate sleep or sleep-on-exit mode by setting SLEEPDEEP in the core System Control Register.
STOP → RUN	Reset or interrupt

3.4 Power Clocks and Valid Wake-Up Sources

The operating frequency of the G32A14xx series depends on the operating mode and clock configuration. After power-on, the default clock is HSI, the internal high-speed clock, at 48MHz. A higher frequency requires SPLL (Phase-Locked Loop) frequency multiplication. In general, higher clock frequency increases power consumption. The clock and wake-up source status in each mode is shown below.

Table 5 Clock and Wake-Up Source Status Comparison

Operating Mode	System Clock Status	Valid Wake-Up Sources
HSR (High-Speed Run)	SYSPLL main PLL enabled, up to 112MHz. Supports HSE/HSI high-speed clocks.	-
RUN	Supports HSE (4–40MHz) and HSI (48MHz). Clock dividers are configurable. PLL can be enabled or disabled as needed.	-

Operating Mode	System Clock Status	Valid Wake-Up Sources
VLPR (Very-Low-Power Run)	PLL is forced off by hardware. Only LSI (8MHz) and LPO (128kHz) low-speed clocks remain available.	GPIO external interrupts and low-power peripheral interrupts, including RTC, LPTMR, LPITMR, LPUART, LPSPI, and LPI2C.
STOP	Main system clock is off. Only the LPO 128 kHz basic clock remains. The bus clock can be configured on or off.	External IO interrupts, RTC alarm or periodic wake-up, and LPTMR timed wake-up.
VLPS (Very-Low-Power Stop)	Only the LPO 128 kHz wake-up clock remains. All other clock domains are off.	RTC periodic/alarm wake-up, LPTMR timed wake-up, LPITMR timed wake-up, and LPUART wake-up.

4 Hardware/Software Considerations and Practical Experience

To reduce MCU power consumption, follow the recommendations below.

4.1 Hardware Considerations

Unused pins, especially pins with analog functions, should be handled properly to avoid extra current consumption.

- ADC pins should be connected to ground.

4.2 Software Considerations

To save power, most module clocks can be disabled by configuring the CGC field in the PCC peripheral control register. These fields are cleared after any reset, disabling the corresponding peripheral clocks by default. Always disable a module before turning off its clock.

Several peripherals support sleep mode. In this mode, register fields can disable the peripheral while it remains in a low-power state.

4.3 Practical Experience

(1) Macro-Level System Strategy (Architecture and Logic Optimization)

Low-power design requires coordination between hardware configuration, system architecture, and software logic.

- Power and supply management: Use the lowest operating voltage allowed by the specification. Use multi-domain power partitioning. Add controlled switches to peripheral circuits, such as sensors and operational amplifiers, so they can be powered off when not used. Remove unnecessary external pull-up and pull-down resistors to reduce static leakage.
- Clock system control (greatest impact on power): Reduce the main clock frequency whenever the application allows. Switch to an internal RC clock or low-speed crystal during idle periods, and disable the high-speed crystal. Fully disable clock gates for unused buses and peripherals. Use dynamic frequency scaling for different load conditions.
- Operating mode selection: Select the appropriate sleep depth based on application needs. For example, Idle stops the CPU, Stop disables the main clock, and VLPS/PowerDown keeps only the wake-up source active. Use periodic wake-up so the MCU works only when needed.
- Software execution logic: Group tasks together to shorten active time. Avoid empty polling loops such as while(1). Use an interrupt-plus-sleep mechanism instead. Simplify code execution and disable unnecessary floating-point operations and hardware acceleration units.

(2) Meso-Level Hardware and Peripheral Control (Detail and Leakage-Current Management)

Peripherals and pins are common sources of hidden current and leakage current, so they must be carefully managed.

- GPIO pin optimization (very easy to overlook):
  - Idle IOs: Set unused IOs to high-impedance input, or drive them to a fixed high or low level based on the external circuit. Do not leave pins floating. Disable internal pull-ups and pull-downs. Disable analog, comparator, and ADC functions. Reduce pin toggle frequency.
  - Wake-up source IOs: Configure wake-up IOs to match the external circuit. For external pull-up with falling-edge wake-up, configure the internal setting as input with pull-up. For external pull-down with rising-edge wake-up, configure the internal setting as input with pull-down.
- Peripheral and bus control:
  - Turn off when unused: Fully power off or disable the clocks of unused peripherals such as ADC, UART, and SPI. Reduce communication baud rates and sampling rates where possible.
  - Streamline redundancy: Disable the watchdog and buzzer if they are not required. Keep LED indicators on for the shortest possible time. In mass production, disable JTAG/SWD debug ports and unnecessary interrupts.
- Storage and bus optimization: Place data in RAM when possible to reduce Flash access. In low-power scenarios, disable Flash prefetch and cache if appropriate. Turn off unused AHB/APB bus clocks.

(3) Micro-Level Standard Operating Procedure (SOP for Entering Low Power)

Before entering a low-power mode, strictly follow these steps. The hardware power supply must be normal before these steps are executed.

- System keep-alive: Feed the watchdog once to prevent reset during sleep.
- Interrupt and flag cleanup (critical):
  - Disable all peripheral and NVIC interrupts that are not wake-up sources.
  - Clear all interrupt flags of enabled modules and all pending flags in the NVIC to prevent low-power entry failure.
- Peripheral and clock shutdown: Turn off all peripheral modules that are not wake-up sources, and disable their clocks.
- GPIO state configuration:
  - Disable alternate functions for unused IOs and non-wake-up IOs. Configure them to a safe fixed level or high-impedance state.
  - Correctly configure the internal and external pull-up/pull-down states of wake-up IOs.
- Sleep configuration: Set the target low-power mode, such as VLPS, and configure the wake-up

source. This must be done before executing WFI to avoid locking the MCU in sleep.

- Global interrupt and final cleanup: Enable global IRQ interrupts, then clear all IRQ flags again.
- Execute sleep: Execute the low-power entry instruction, such as WFI or WFE.

(4) Practical Pitfall-Avoidance and Lessons Learned

- Prevent “bricking”: Configure the wake-up source before executing WFI. Otherwise, if the MCU enters deep sleep, such as VLPS, it may not wake up, and the system may appear “bricked.”
- Prevent “instant wake-up” and entry failure: Before entering sleep, clear all remaining interrupt flags, especially NVIC pending flags, and disable all non-wake-up interrupts. Otherwise, the MCU may wake immediately after entering sleep, causing high power consumption.
- Handle IO states carefully: Floating pins can greatly increase power consumption. All IO states must be configured properly in software.

Troubleshooting methodology: If power consumption is higher than expected, do not modify code blindly. Follow the SOP step by step. Use a multimeter or oscilloscope to check for peripheral clocks that remain enabled, floating GPIO pins, and uncleared interrupt flags.

5 Power Mode Use Cases

5.1 Hardware Preparation

- G32A1445 EVAL board × 1
- J-Link programmer × 1, including Type-B to Type-A USB cable and FC-20P dual-head cable/2.54mm pitch
- Mini USB data cable × 1
- Multimeter
- Several jumper wires
- Computer × 1

5.2 Software Preparation

- G32A1xxx_SDK_x.x.x
- Keil MDK and G32A series support package
- SEGGER – J-Link Vx.xx
- Serial port assistant

5.3 Test Procedure

（1） Open the corresponding Keil MDK project, modify it, compile it, and download it to the evaluation board:

“G32A1xxx_SDK_x.x.x\Examples\G32A1445\POWER\Power_SwitchMode\Project\MDK\Power_SwitchMode.uvprojx”. In “main.c”, comment out the statement that turns on the LED, namely the call to “LED_On()”. In STOP and VLPS modes, configure SLOWCLK as the clock output and turn off unused peripheral clocks to further reduce power consumption. Compile the project, download it to the G32A1445 EVAL board, and disconnect the J-Link programmer from the board.

（2） To further reduce the effect of peripheral hardware on power consumption, modify the G32A1445 EVAL board as follows:

- Remove pull-up resistor R17 on the LIN_RX pin.
- Remove potentiometer RV1.
- Remove button pull-up resistors R22, R29, and R11.
- Remove RGB LED series resistors R26, R27, and R28.
- Remove SWD programming port pull-up resistors R3, R4, and R5.

（3） Device connection: Remove the jumper cap at J12 on the G32A1445 EVAL board. Connect the red probe of the multimeter to VDD at J12 and the black probe to V_MCU at J12. Use the jumper cap at J13 to connect VDD to either 5V or 3.3V. Set the multimeter to the mA range. Connect the evaluation board to the computer using the Mini USB cable. Open the Serial Port Assistant on the computer, configure the serial parameters, and open the corresponding port.

（4） Send characters 1–6 through the serial port assistant to enter the corresponding power modes: HSRUN, RUN, VLPR, STOP1, STOP2, and VLPS. After sending the command, when VDD is supplied at +3.3V, remove the jumper caps at J9 and J10. Then observe and record the current reading on the multimeter. This reading is the power consumption of the EVAL board in the selected mode. The serial print for entering VLPS mode is shown in Figure 2. The measured power consumption is shown in Table 6.

Figure 2 Composition of the Low-Power System

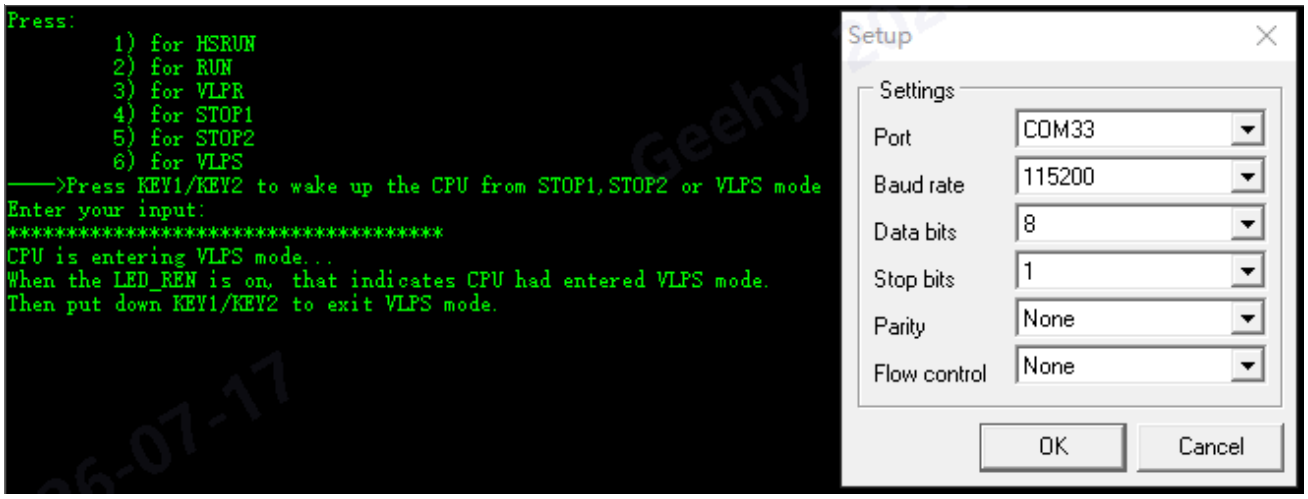


Table 6 Power Consumption Comparison of the G32A1445 EVAL Board at VDD = +5 V / +3.3 V in Different Power Modes at Room Temperature (25 °C)

VDD Supply	Power Mode	HSRUN	RUN	VLPR	STOP1	STOP2	VLPS
+5.0	Current/mA	21.893	14.451	1.197	6.247	6.531	0.049
+3.3		22.06	14.668	1.201	6.343	6.634	0.05

(5) End of experiment: Disconnect all connections and restore the original jumper caps and hardware connections on the G32A1445 EVAL board.

6 Revision History

Table 7 Document Revision History

Date	Version	Revision History
July, 2026	1.0	● Initial version

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.