Application Note

Application Note

Document No.: AN1123

APM32F402 Migration Guide

Version: V1.0

1. Introduction

For embedded system designers, it is crucial to easily migrate applications between different series of MCU. With continuous increase in product demand, the demand for resources such as memory size and I/O quantity also increases, so designers often need to migrate applications to the MCU with higher performance or more resources.

This Migration Guide is intended to help users to analyze the steps required to migrate from existing APM32E103xx device to APM32F402xx device. This document collects the most important information and lists the key matters needing attention in the migration process. The main aspects involved in the migration process include hardware migration, peripheral migration, firmware migration, and tool chain migration.

To make full use of the information in this guide, users shall be familiar with the characteristics and development environment of APM32F402xx series MCU. Users can refer to the following relevant technical documents:

- APM32F402xx Series User Manual and Datasheet
- APM32F402xx series core related documents

This document will systematically introduce the steps of migration from APM32E103xx to APM32F402xx, and provide practical examples and codes to help designers to smoothly complete the application migration work and improve the overall performance and reliability of the system. The MCU applicable to this application note is APM32F402.

Through this Migration Guide, the designers can better understand and address various challenges encountered in the migration process, to ensure that optimal performance of applications can be implemented on the new MCU platform.

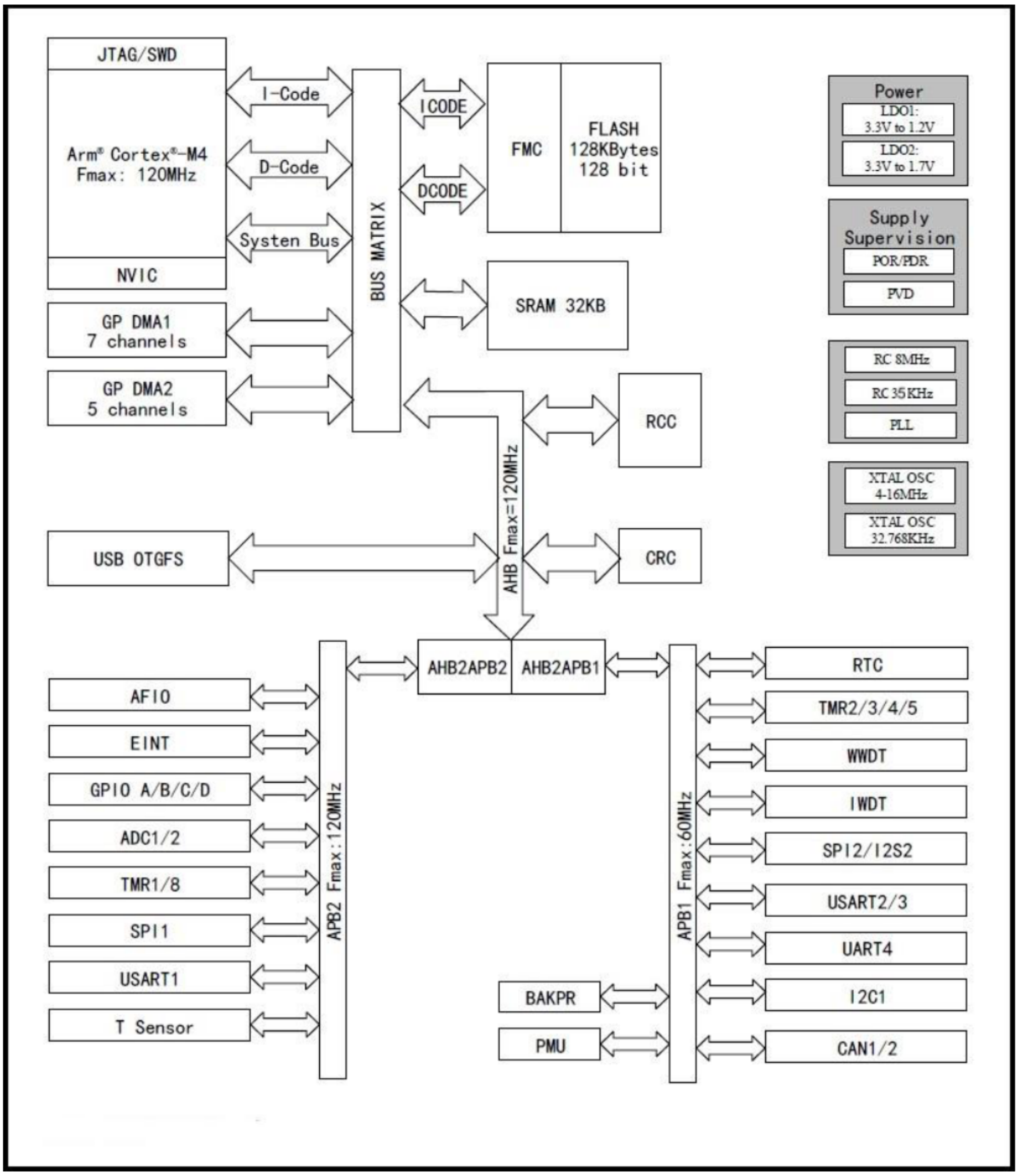
1.1. APM32F402xx Introduction

The Arm® Cortex®-M4F core in the product is equipped with an FPU, and the system mainly consists of four driver units and three passive units.

The four driver units are connected to DCode (D-bus), system bus (S-bus), general-purpose DMA1, and DMA2 of the Arm® Cortex®-M4F core. The three passive units are internal SRAM, internal flash memory, and AHB-to-APB bridge (AHB/APBx), among which AHB/APBx connects all APB devices. They are all connected through a multi-level AHB bus architecture.

The architecture of APM32F402xx is as shown in Figure 1 below.

Figure 1 APM32F402xx System Block Diagram



Contents

1. Introduction 1

1.1. APM32F402xx Introduction 1

2. Startup mode 4

2.1. Startup configuration 4

2.2. Embedded Bootloader 4

3. Peripheral 5

3.1. Resource Comparison List 5

3.2. CPU core 6

3.3. Interrupt Vector Table Differences 6

3.4. Memory mapping 8

3.5. Clock tree differences 9

4. Standard peripheral library 11

5. Tool chain support 15

5.1. IDE support 15

5.2. Simulator support 15

6. Revision history 16

2. Startup mode

2.1. Startup configuration

The method to enter the boot mode of APM32F402 is consistent with that of APM32E103. By configuring the BOOT[1:0] pin parameters, three different boot modes can be switched.

2.2. Embedded Bootloader

APM32F402 is embedded with two sets of Bootloader, and supports five communication protocols, i.e. USART, I2C, SPI, CAN, and OTG FS, which can be switched through the option byte NROMSEL. APM32E103 is embedded with a set of Bootloader, which only supports USART. Their differences can be seen in Table1.

Table1 Bootloader Comparison Table

APM32F402		APM32E103
NROMSEL = 1	NROMSEL = 0	
USART1(PA9\PA10) USART2(PA2\PA3)	I2C1(PB6\PB7) SPI1(PA2\PA3\PA4\PA5) CAN2(PB5\PB6) OTG_FS(PA11\PA12)	USART1(PA9\PA10)

Note: When using the OTG_FS function, PA9 needs to be connected to an external VBUS pin or 5V power pin.

3. Peripheral

It is crucial to understand the differences in peripheral resources between the two MCU in the migration process from APM32E103 to APM32F402. The selection and configuration of peripheral resources not only directly affects the performance and function of the system, but also concerns the implementation and optimization of applications. This chapter will compare the peripheral resources of the two MCU in detail.

3.1. Resource Comparison List

In order to clearly demonstrate the differences in peripheral resources between APM32E103 and APM32F402, the main peripheral resources of these two MCU are compared Table 2 based on the maximum package of APM32F402.

Table 2 Peripheral Resource Comparison between APM32F402 and APM32E103

Product		APM32F402RB	APM32E103RC
Package		LQFP64	LQFP64
Core and maximum working frequency		Arm® 32-bit Cortex®-M4F@120MHz	Arm® 32-bit Cortex®-M3@120MHz
Flash (KB)		128	512
SRAM(KB)		32	128
GPIOs		51	51
Communication interface	USART/UART	3/1	3/2
	SPI/I2S	2/1	3/2
	I2C	1	2
	I2C3	0	1
	USBD	0	1
	USB OTG FS	1	0
	CAN	2	2
	SDIO	0	1
Timer	16-bit advanced	2	2
	32-bit general	1	0
	16-bit general	3	4
	16-bit basic	0	2
	System tick timer	1	1
	Watchdog	2	2
Real-time Clock		1	1
12-bit ADC	Unit	2	3
	Channel	16	16
12-bit DAC	Unit	0	2
	Channel	0	2

3.2. CPU core

Compared with the APM32E103 MCU, the ARM Cortex-M4F core on which the APM32F402 is based additionally supports FPU, enabling it to efficiently perform floating-point arithmetic.

3.2.1. Floating point unit (FPU)

APM32F402 supports FPU, so it has significant advantages in applications that require high-precision computation. The floating-point unit (FPU) can significantly improve the speed and accuracy of floating-point arithmetic, and its performance of performing floating-point arithmetic is significantly improved compared with the cores without FPU.

3.3. Interrupt Vector Table Differences

The interrupt numbers used by APM32F402 can support up to 66 interrupt lines. All interrupts are sent to the CPU through the nested vector interrupt controller (NVIC). The external interrupt controller (EINT) provides external interrupts (GPIO).

Table 3 Comparison of Interrupts and Abnormal Functions

Function	APM32F402	APM32E103
Number of interrupts	66	65
Interrupt priority (bit)	4	4

In the interrupt vector table, APM32F402 and APM32E103 are not the same, and for the differences between them, refer to Table 4.

Table 4 Interrupt Vector Comparison Table

Vector ID	APM32F402	APM32E103
0	WWDT	WWDT
1	PVD	PVD
2	TAMPER	TAMPER
3	RTC	RTC
4	FLASH	FLASH
5	RCM	RCM
6	EINT0	EINT0
7	EINT1	EINT1
8	EINT2	EINT2
9	EINT3	EINT3
10	EINT4	EINT4
11	DMA1 Channel 1	DMA1 Channel 1

Vector ID	APM32F402	APM32E103
12	DMA1 Channel 2	DMA1 Channel 2
13	DMA1 Channel 3	DMA1 Channel 3
14	DMA1 Channel 4	DMA1 Channel 4
15	DMA1 Channel 5	DMA1 Channel 5
16	DMA1 Channel 6	DMA1 Channel 6
17	DMA1 Channel 7	DMA1 Channel 7
18	ADC1_2	ADC1_2
19	CAN1 TX	USBD1 HP / CAN1 TX
20	CAN1 RX0	USBD1 LP / CAN1 RX0
21	CAN1 RX1	CAN1 RX1
22	CAN1 SCE	CAN1 SCE
23	EINT9_5	EINT9_5
24	TMR1 BRK	TMR1 BRK
25	TMR1 UP	TMR1 UP
26	TMR1 TRG	TMR1 TRG
27	TMR1 CC	TMR1 CC
28	TMR2	TMR2
29	TMR3	TMR3
30	TMR4	TMR4
31	I2C1 EV	I2C1 EV
32	I2C1 ER	I2C1 ER
33	Reserved	I2C2 EV
34	Reserved	I2C2 ER
35	SPI1	SPI1
36	SPI2	SPI2
37	USART1	USART1
38	USART2	USART2
39	USART3	USART3
40	EINT15_10	EINT15_10
41	RTC Alarm	RTC Alarm
42	OTG FS WKUP	USBD WakeUp
43	TMR8 BRK	TMR8 BRK

Vector ID	APM32F402	APM32E103
44	TMR8 UP	TMR8 UP
45	TMR8 TRG COM	TMR8 TRG COM
46	TMR8 CC	TMR8 CC
47	Reserved	ADC3
48	Reserved	EMMC
49	Reserved	SDIO
50	TMR5	TMR5
51	Reserved	SPI3
52	UART4	UART4
53	Reserved	UART5
54	Reserved	TMR6
55	Reserved	TMR7
56	DMA2 Channel 1	DMA2 Channel 1
57	DMA2 Channel 2	DMA2 Channel 2
58	DMA2 Channel 3	DMA2 Channel 3
59	DMA2 Channel 4_5	DMA2 Channel 4_5
60	FPU	Reserved
61	CAN2 TX	USBD2 HP / CAN2 TX
62	CAN2 RX0	USBD2 LP / CAN2 RX0
63	CAN2 RX1	CAN2 RX1
64	CAN2 SCE	CAN2 SCE
65	OTG FS	Reserved

3.4. Memory mapping

Compared with APM32E103, the Flash of APM32F402 adds an adaptive real-time memory accelerator, which can improve the execution speed of Flash. The configurable range of wait periods is wider. Table 5 is a comparison table of the Flash function between APM32F402 and APM32E103.

Table 5 Flash Function Comparison between APM32F402 and APM32E103

Function	APM32F402	APM32E103
Capacity	Maximum 128KB	Maximum 512KB
Flash wait period	0 (0 < SYSCLK ≤ 30) 1 (30 < SYSCLK ≤ 60)	0 (0< SYSCLK ≤ 24) 1 (24 < SYSCLK ≤ 48) 2 (48 < SYSCLK ≤ 72)

Function	APM32F402	APM32E103
	2 (60 < SYSCLK ≤ 90) 3 (90 < SYSCLK ≤ 120)	3 (72 < SYSCLK ≤ 96) 4 (96 < SYSCLK ≤ 120)
Sleep mode	Supported	Supported
Standby mode	Supported	Supported
Stop mode	Supported	Supported
FACC acceleration	Supported	Not supported

The memory mapping of APM32F402 is shown in Table 6.

Table 6 Memory Mapping of APM32F402

Block	Name	Address range	Size (bytes)
Main memory block	Page 0	0x0800 0000 – 0x0800 03FF	1K
Main memory block	Page 1	0x0800 0400 – 0x0800 07FF	1K
Main memory block	Page 2	0x0800 08400 – 0x0800 0BFF	1K
Main memory block	..	…	…
Main memory block	Page 127	0x0801 FC00 – 0x0801 FFFF	1K
Information block	System memory area	0x1FFF E400 – 0x1FFF F7FF	5K
Information block	Option byte	0x1FFF F800 – 0x1FFF F80F	16

3.5. Clock tree differences

The clock usage of APM32F402 is similar to that of APM32E103, it supports AHB, APB1, and APB2, but there are also some differences, as shown in Table 7 .

Table 7 Clock Structure Comparison Table

Clock	APM32F402	APM32E103
LSICLK	Supported	Supported
LSECLK	Supported	Supported
HSICLK	Supported	Supported
HSECLK	Supported	Supported
FMCCLK	Supported	Supported
SYSCLK	Supported	Supported
HCLK	Supported	Supported
PCLK1	Supported	Supported
PCLK2	Supported	Supported
OTG FS CLK	Supported	Not supported

Clock	APM32F402	APM32E103
USBD CLK	Not supported	Supported
FPUCLK	Not supported	Supported
SMCCLK	Not supported	Supported
SDRAMCLK	Not supported	Supported
SDIOCLK	Not supported	Supported
TMRxCLK	Supported	Supported
ADCCLK	Supported	Supported
I2SxCLK	Supported	Supported

4. Standard peripheral library

The standard firmware library of APM32F402 is basically consistent with APM32E103, but due to differences in some IP, they have significant differences in naming and implementation methods. Therefore, in the migration process, it is necessary to conduct detailed analysis and adaptation for the existing firmware library.

This chapter will introduce how to migrate the firmware library from the APM32E103 series to the APM32F402 series, focusing on describing the differences in driver library.

4.1.1. CAN module

The CAN module of APM32F402 adds the configuration field for the CAN2 starting storage address in the FCTRL register. The API differences are shown below.

Table 8 CAN API Differences

APM32F402	APM32E103
CAN_ConfigFilter(CAN_FilterConfig_T* filterConfig)	CAN_ConfigFilter(CAN_T * can, CAN_FilterConfig_T* filterConfig)
CAN_SlaveStartBank(uint8_t bankNum)	/

4.1.2. FMC module

The FMC module of APM32F402 is basically compatible with the FMC module of APM32E103, but a series of new functions are added, e.g. extending the wait period settings of APM32F402 to 32. The API differences are shown below.

Table 9 FMC API Differences

APM32F402	APM32E103
FMC_EnableInstructionCache	/
FMC_DisableInstructionCache	/
FMC_EnableDataCache	/
FMC_DisableDataCache	/
FMC_ResetInstructionCache	/
FMC_ResetDataCache	/
FMC_EnablePrefetch	/
FMC_DisablePrefetch	/

4.1.3. PMU module

The PMU module of APM32F402 is basically compatible with the PMU module of APM32E103, but a series of API are added. The API differences are as follows.

Table 10 PMU API Differences

APM32F402	APM32E103
PMU_EnterSleepMode	/

4.1.4. RCM module

The RCM module of APM32F402 is basically compatible with the RCM module of APM32E103, and the main differences come from IP differences. The API differences are as follows.

Table 11 RCM API Differences

APM32F402	APM32E103
/	RCM_ConfigFPUCLK
RCM_EnableAHBPeriphReset	/
RCM_DisableAHBPeriphReset	/

4.1.5. TMR module

The TMR driver library of APM32F402 is compatible with the TMR driver library of APM32E103. But some API has differences in the number of transfer parameters and the returned value.

- Naming difference in driver library API
  - No difference
- Differences in driver library API
  - Please refer to Table 12 for quantity differences Table 12
  - Please refer to Table 13 for transfer parameter type Table 13
  - There is no difference in the number of transfer parameters
- Naming difference in driver library macro definition
  - No difference
- Differences in driver library enumeration body
  - Please refer to Table 14 for differences Table 14

Table 12 Differences in API Quantity of TMR Module Driver Library

APM32F402	APM32E103
TMR_ConfigRemap	/
TMR_ReadRemap	/

Table 13 Differences in API Transfer Parameter Types of TMR Module Driver Library

APM32F402	APM32E103
TMR_ConfigCounter(TMR_T* tmr, uint32_t counter)	TMR_ConfigCounter(TMR_T* tmr, uint16_t counter)
TMR_ConfigAutoreload(TMR_T* tmr, uint32_t autoReload)	TMR_ConfigAutoreload(TMR_T* tmr, uint16_t autoReload)
TMR_ConfigCompare1(TMR_T* tmr, uint32_t compare1)	TMR_ConfigCompare1(TMR_T* tmr, uint16_t compare1)
TMR_ConfigCompare2(TMR_T* tmr, uint32_t compare2)	TMR_ConfigCompare2(TMR_T* tmr, uint16_t compare2)
TMR_ConfigCompare3(TMR_T* tmr, uint32_t compare3)	TMR_ConfigCompare3(TMR_T* tmr, uint16_t compare3)
TMR_ConfigCompare4(TMR_T* tmr, uint32_t compare4)	TMR_ConfigCompare4(TMR_T* tmr, uint16_t compare4)
uint32_t TMR_ReadCaputer1(TMR_T* tmr)	uint16_t TMR_ReadCaputer1(TMR_T* tmr)
uint32_t TMR_ReadCaputer2(TMR_T* tmr)	uint16_t TMR_ReadCaputer2(TMR_T* tmr)
uint32_t TMR_ReadCaputer3(TMR_T* tmr)	uint16_t TMR_ReadCaputer3(TMR_T* tmr)
uint32_t TMR_ReadCaputer4(TMR_T* tmr)	uint16_t TMR_ReadCaputer4(TMR_T* tmr)
uint32_t TMR_ReadCounter(TMR_T* tmr)	uint16_t TMR_ReadCounter(TMR_T* tmr)

Table 14 Differences in API Emulation Body of TMR Module Driver Library

APM32F402	APM32E103
TMR_REMAP_T	/

4.1.6. USB OTG module

There are differences between the APM32F402 USB module and the APM32E103 USB module, which can be briefly understood from the followings. For specific usage, please refer to APM32F402 SDK OTG Routine.

- IP differences
  - The USB module of APM32F402 is OTG_FS module, and it supports both host mode and device mode. The USB module of APM32E103 is USB_D module, and it only supports device mode.
- API differences
  - For specific API differences, please refer to Driver Library

- Differences in use process
  - APM32F402 adds the pin for character recognition detection and the pin for VBUS detection. The data pins are the same.
  - There are differences in interrupt vector between APM32F402 and APM32E103, as shown below

Table 15 Usage Differences in USB Interrupt

APM32F402	APM32E103
OTG_FS_WKUP_IRQn（42）	USBD2_HP_CAN2_TX_IRQn（61）
OTG_FS_IRQn（65）	USBD2_LP_CAN2_RX0_IRQn（62）

- Differences in configuration process
  - There are differences in the data buffer configuration between OTG FS and USBD.
  - There is no conflict between USB and CAN of APM32F402, while there are conflicts between USB and CAN of APM32E103.

5. Tool chain support

In embedded development, the use of tool chains such as IDE and simulators is crucial. Below are the IDE and emulators supported by APM32F402.

5.1. IDE support

Support of IDE by APM32F402:

- MDK-ARM
- IAR-EWARM V8.50.5.26295
- ECLIPSE-EMB

5.2. Simulator support

Support of simulators by APM32F402:

- Geehy-Link (WinUSB), DAP Link (the firmwar version is CMSIS-DAP V2 and above)
- J-Link

6. Revision history

Table 16 Document Revision History

Date	Version	Revision History
2024.12	V1.0	New

Statement

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.