

Readme

APM32F003 SDK

Rev: V1.1

1 Introduction

The Geehy Semiconductor APM32F003MINI board software development kit includes a series driver library, a group of example applications that demonstrate key peripheral functionality, and other development files.

Software development kit have a hierarchy as follows:

- SDK directory
 - * [Boards](#)
 - * [Documents](#)
 - * [Examples](#)
 - * [Libraries](#)
 - * [Middlewares](#)
 - * [Package](#)

Table of Contents

1	Introduction	1
2	About boards	5
3	About documents.....	6
4	About examples.....	7
4.1	ADC_Channel_2.....	10
4.2	ADC_Channel_8.....	10
4.3	ADC_ContinuousConversion.....	10
4.4	ADC_DiffMode.....	11
4.5	ADC_MultiChannelScan.....	11
4.6	DMA_ADC.....	11
4.7	DMA_MemoryToMemory	12
4.8	DMA_TMRBurst.....	12
4.9	EINT	13
4.10	FMC_Program.....	13
4.11	FMC_WriteProtection.....	13
4.12	GPIO_Toggle	13
4.13	I2C_TwoBoardsInterrupt	14
4.14	I2C_TwoBoardsPolling	14
4.15	Application1.....	14
4.16	Application2.....	15
4.17	BootLoader	15
4.18	IWDT	15
4.19	LPM_StopMode	15
4.20	LPM_WaitMode.....	16
4.21	NVIC.....	16
4.22	RCM_ClockSwitch	16

4.23	RCM_CRS	17
4.24	FreeRTOS	17
4.25	RT-Thread	17
4.26	RTX.....	17
4.27	SPI_TwoBoards	18
4.28	SPI_TwoBoardsDMA	18
4.29	SysTick	19
4.30	Template	19
4.31	TMR1_6stepswithTMR2	19
4.32	TMR1_ADC_Trigger_Sampling.....	20
4.33	TMR1_ComplementaryOutput.....	20
4.34	TMR1_InputCapture	20
4.35	TMR1_PWMOutput.....	21
4.36	TMR2_OCAActive.....	21
4.37	TMR2_OCInActive.....	21
4.38	TMR2_PWMOutput.....	22
4.39	TMR2_SinglePulse	22
4.40	TMR2_SynchronizationWithTMR1	22
4.41	TMR4_TimeBase.....	23
4.42	UART_DMA	23
4.43	UART_HalfDuplex	23
4.44	UART_Interrupt	24
4.45	UART_Polling	24
4.46	UART_Printf	24
4.47	WUPT.....	25
4.48	WWDT	25
5	About libraries	26
6	About middlewares	27

7	About Package	28
8	Revision History	29

2 About boards

The boards folder includes a board support package for APM32F003MINI board. It can help drive the peripheral circuit or components on the board quickly. The BSP can be found in the [Boards](#) directory.

The BSP provided are built for APM32F003MINI board compatibility. For other user development board use, some minor modifications may be required.

Boards have a hierarchy as follows:

- Boards folder
 - * Board_APM32F003_MINI
 - inc
 - src
 - * Board.c
 - * Board.h

Board APM32F003 MINI include following board support package:

- Board_APM32F003_MINI src folder
 - * Board_APM32F003_MINI

3 **About documents**

The documents folder includes a link file that can be redirected to the technical support center of Geehy semiconductor. The document can be found in the [Documents](#) directory.

4 About examples

The example applications can be found in the [Examples](#) directory.

The examples provided are built for APM32F003MINI board compatibility. For other user development board use, some minor modifications may be required.

Example projects have a hierarchy as follows:

- Example folder
 - * Include
 - * Project
 - Eclipse
 - IAR
 - MDK
 - * Source

All example applications tested with: **APM32F003StdPeriphDriver v1.0**, include the following examples:

- Examples
 - * ADC
 - [ADC_Channel_2](#)
 - [ADC_Channel_8](#)
 - [ADC_ContinuousConversion](#)
 - [ADC_DiffMode](#)
 - [ADC_MultiChannelScan](#)
 - * DMA
 - [DMA_ADC](#)
 - [DMA_MemoryToMemory](#)
 - [DMA_TMRBurst](#)
 - * EINT
 - [EINT](#)
 - * FMC

- [FMC Program](#)
- [FMC WriteProtection](#)
- * GPIO
 - [GPIO Toggle](#)
- * I2C
 - [I2C_TwoBoardsInterrupt](#)
 - [I2C_TwoBoardsPolling](#)
- * IAP
 - [Application1](#)
 - [Application2](#)
 - [BootLoader](#)
- * IWDT
 - [IWDT](#)
- * LPM
 - [LPM_StopMode](#)
 - [LPM_WaitMode](#)
- * RCM
 - [RCM_ClockSwitch](#)
 - [RCM_CRs](#)
- * RTOS
 - [FreeRTOS](#)
 - [RT-Thread](#)
 - [RTX](#)
- * SPI
 - [SPI_TwoBoards](#)
 - [SPI_TwoBoardsDMA](#)
- * SysTick
 - [SysTick](#)

- * Template
 - [Template](#)
- * TMR1
 - [TMR1_6stepswithTMR2](#)
 - [TMR1_ADC_Trigger_Sampling](#)
 - [TMR1_ComplementaryOutput](#)
 - [TMR1_InputCapture](#)
 - [TMR1_PWMOutput](#)
- * TMR2
 - [TMR2_OCActive](#)
 - [TMR2_OCInActive](#)
 - [TMR2_PWMOutput](#)
 - [TMR2_SinglePulse](#)
 - [TMR2_SynchronizationWithTMR1](#)
- * TMR4
 - [TMR4_TimeBase](#)
- * USART
 - [UART_DMA](#)
 - [UART_HalfDuplex](#)
 - [UART_Interrupt](#)
 - [UART_Polling](#)
 - [UART_Printf](#)
- * WUPT
 - [WUPT](#)
- * WWDT
 - [WWDT](#)

4.1 ADC_Channel_2

4.1.1 Example Description

This example describes how to use ADC peripheral to convert ADC_Channel_4 or ADC_Channel_5 input voltage in Continuous conversion mode with polling.

After Initialization, ADC start to convert. If press the Key1, ADC will choose the ADC Channel 5 to convert, else ADC will choose the ADC Channel 4 to convert.

If the converted voltage greater than 3100mV, the LED2 on and LED3 off;

If the converted voltage less than 800mV, the LED2 off and LED3 on;

If the converted voltage is between 3100mV and 800mV, the LED2 and LED3 on.

4.1.2 Directory content

This example can be found in the [ADC_Channel_2](#) directory.

4.2 ADC_Channel_8

4.2.1 Example Description

This example describes how to use ADC peripheral to convert ADC_Channel_8 internal voltage in continuous conversion mode with polling.

After Initialization, the voltage of ADC_Channel_8 will be printed on UART1.

4.2.2 Directory content

This example can be found in the [ADC_Channel_8](#) directory.

4.3 ADC_ContinuousConversion

4.3.1 Example Description

This example describes how to use ADC peripheral to convert ADC Channel 4 input voltage in continuous conversion mode with interrupt.

After Initialization, ADC will use ADC Channel 4 to convert voltage.

If the converted voltage greater than 3100mV, the LED2 on and LED3 off;

If the converted voltage less than 800mV, the LED2 off and LED3 on;

If the converted voltage is between 3100mV and 800mV, the LED2 and LED3 on.

4.3.2 Directory content

This example can be found in the [ADC_ContinuousConversion](#) directory.

4.4 ADC_DiffMode

4.4.1 Example Description

This example provides how to use ADC's Difference mode.

In this example, ADC will calculate the difference between two channels, and output the value to upper computer by Uart1.

4.4.2 Directory content

This example can be found in the [ADC_DiffMode](#) directory.

4.5 ADC_MultiChannelScan

4.5.1 Example Description

This example describes how to use ADC peripheral to scan multiple channels input voltage in continuous scan mode with polling.

After Initialization, ADC_Channel_0, ADC_Channel_1 and ADC_Channel_2 start to convert voltage. The value of ADC channels will be printed on UART1.

4.5.2 Directory content

This example can be found in the [ADC_MultiChannelScan](#) directory.

4.6 DMA_ADC

4.6.1 Example Description

This example provides an example of how to use a DMA channel to transfer continuously a data from a peripheral (ADC) to DMA transfer. The ADC channel1 for APMF002 MINI Board is configured to be converted when device startup.

The value of ADC is shown in UART1.

4.6.2 Directory content

This example can be found in the [DMA_ADC](#) directory.

4.7 DMA_MemoryToMemory

4.7.1 Example Description

This example shows how to configure the DMA peripheral to transmit data from memory to memory. After system reset, data transmit from one group to another through DMA. If the data received is equal to the data send, LED2 will light, otherwise, LED3 will light.

4.7.2 Directory content

This example can be found in the [DMA_MemoryToMemory](#) directory.

4.8 DMA_TMRBurst

4.8.1 Example Description

The program shows how to configure the TMR1 channel period and the duty cycle by DMA burst to generate 2 PWM with 2 different duty cycles (80%, 70%).

On the DMA update request, the DMA will do 3 transfers of half words into TMR2 registers (AUTORLD, CC1, CC2).

999 will be transferred into AUTORLD

800 will be transferred into CC1.

700 will be transferred into CC2.

4.8.2 Directory content

This example can be found in the [DMA_TMRBurst](#) directory.

4.9 EINT

4.9.1 Example Description

This example describes how to use EINT peripheral. When the KEY1 is pressed, the LED2 and LED3 will twinkle.

4.9.2 Directory content

This example can be found in the [EINT](#) directory.

4.10 FMC_Program

4.10.1 Example Description

This example describes how to use Program flash.

When program is ok, LED3 on else LED2 and LED3 twinkle together.

4.10.2 Directory content

This example can be found in the [FMC_Program](#) directory.

4.11 FMC_WriteProtection

4.11.1 Example Description

This example describes how to set write protection.

When write protection error is triggered, LED3 turn on.

When write successful, LED2 turn on.

4.11.2 Directory content

This example can be found in the [FMC_WriteProtection](#) directory.

4.12 GPIO_Toggle

4.12.1 Example Description

This example describes how to use GPIO peripheral. LED2 and LED3 will alternate twinkle.

4.12.2 Directory content

This example can be found in the [GPIO_Toggle](#) directory.

4.13 I2C_TwoBoardsInterrupt

4.13.1 Example Description

This example describes how to use I2C peripheral establish communication in interrupt mode. After initialization, master will send data to slave, after a period of delay, master will read the slave's received data.

4.13.2 Directory content

This example can be found in the [I2C_TwoBoardsInterrupt](#) directory.

4.14 I2C_TwoBoardsPolling

4.14.1 Example Description

This example describes how to use I2C peripheral establish communication in polling mode. After initialization, master will send data to slave, after a period of delay, master will read the slave's received data.

4.14.2 Directory content

This example can be found in the [I2C_TwoBoardsPolling](#) directory.

4.15 Application1

4.15.1 Example Description

This example shows how to generate a APP firmware to IAP.

LED2 are toggled with a timing defined by the Delay function.

4.15.2 Directory content

This example can be found in the [Application1](#) directory.

4.16 Application2

4.16.1 Example Description

This example shows how to generate a APP firmware to IAP.

LED3 are toggled with a timing defined by the Delay function.

4.16.2 Directory content

This example can be found in the [Application2](#) directory.

4.17 BootLoader

4.17.1 Example Description

The example aims to show how to configure a bootloader firmware to IAP.

4.17.2 Directory content

This example can be found in the [BootLoader](#) directory.

4.18 IWDT

4.18.1 Example Description

This example describes how to use IWDT peripheral.

After initialization, system keep feed dog and the LED2 keep twinkle. If KEY1 is pressed, system will stop feed dog then the IWDT will generate a reset and LED3 on.

4.18.2 Directory content

This example can be found in the [IWDT](#) directory.

4.19 LPM_StopMode

4.19.1 Example Description

This example provides how to use LVD interrupt or EINT interrupt to wake up CPU which is in a stop mode.

4.19.2 Directory content

This example can be found in the [LPM_StopMode](#) directory.

4.20 LPM_WaitMode

4.20.1 Example Description

This example provides how to use UART1 to receive a falling edge to wake up CPU which is in a waiting low-power manage state.

4.20.2 Directory content

This example can be found in the [LPM_WaitMode](#) directory.

4.21 NVIC

4.21.1 Example Description

This example describes how a high priority interrupt preempts a low priority interrupt.

When connect PA3 to GND, LED3 will blink five times. If press KEY1 at this moment, LED3 stops until LED2 has blinked five times.

4.21.2 Directory content

This example can be found in the [NVIC](#) directory.

4.22 RCM_ClockSwitch

4.22.1 Example Description

This example describes how to switch the master clock.

Pressing KEY1, master clock will switch between HSE and HSI.

The clock can be visualized on CCO pin.

4.22.2 Directory content

This example can be found in the [RCM_ClockSwitch](#) directory.

4.23 RCM_CRS

4.23.1 Example Description

This example describes how to use the CRS peripheral.

Short-circuiting the external crystal oscillator make it disable. And system switch clock by CRS so that the flashing frequency of the light changes.

4.23.2 Directory content

This example can be found in the [RCM_CRS](#) directory.

4.24 FreeRTOS

4.24.1 Example Description

This example describes how to use FreeRTOS create multiple tasks.

Led toggle task: The LED2 and LED3 is configured to toggle constantly

4.24.2 Directory content

This example can be found in the [FreeRTOS](#) directory.

4.25 RT-Thread

4.25.1 Example Description

This example describes how to use RT-Thread create task.

UART1 will print a string of information.

4.25.2 Directory content

This example can be found in the [RT-Thread](#) directory.

4.26 RTX

4.26.1 Example Description

This example describes how to use RTX5 create multiple tasks.

www.geehy.com

Uart test task: UART1 and UART2 send or received data to each other. Verification will occur after transmission,

if send and receive data pass, LED3 will be on all the time.

if send and receive data fault, LED3 will be off all the time.

if send or receive data fault, LED3 will be constantly flickered alternately.

Led toggle task: The LED2 is configured to toggle constantly

4.26.2 Directory content

This example can be found in the [RTX](#) directory.

4.27 SPI_TwoBoards

4.27.1 Example Description

This example provides a small application in which system sends and receives data by polling though using SPI firmware library. All received information will be displayed by serial assistant.

To use this example, you need to load it on two APM32F003 boards (let's call them Board master and Board slave). Then connect these two boards through SPI lines and must master and slave connect to the same GND.

When the program is running, the LED3 is toggle.

When compare buffer is the same, the LED2 is on.

The phenomenon of data interaction process can be displayed using serial assistant.

4.27.2 Directory content

This example can be found in the [SPI_TwoBoards](#) directory.

4.28 SPI_TwoBoardsDMA

4.28.1 Example Description

This example provides a small application in which system sends and receives data by DMA though using SPI firmware library. All received information will be displayed by serial assistant.

To use this example, you need to load it on two APM32F003 boards (let's call them Board master and Board slave). Then connect these two boards through SPI lines and must master and slave connect to the same GND.

When the program is running, the LED3 is toggle.

When compare buffer is the same, the LED2 is on.

The phenomenon of data interaction process can be displayed using serial assistant.

4.28.2 Directory content

This example can be found in the [SPI_TwoBoardsDMA](#) directory.

4.29 SysTick

4.29.1 Example Description

This example describes how to configure the SysTick to generate a time base equal to 1 ms. After Initialization, LED2 and LED3 will toggle interval 1ms.

4.29.2 Directory content

This example can be found in the [SysTick](#) directory.

4.30 Template

4.30.1 Example Description

This example provides a template project.

4.30.2 Directory content

This example can be found in the [Template](#) directory.

4.31 TMR1_6stepswithTMR2

4.31.1 Example Description

This example describes how to configure the TMR1 peripheral and TMR2 peripheral to generate 6 steps PWM signal.

The waveform can be show be displayed using an oscilloscope.

4.31.2 Directory content

This example can be found in the [TMR1_6stepswithTMR2](#) directory.

4.32 TMR1_ADC_Trigger_Sampling

4.32.1 Example Description

This example uses the Update event of TMR1 to periodically trigger ADC1 acquisition.

After the ADC is initialized, the update event of TMR1 is used to trigger the ADC acquisition.

In Debug mode, set a breakpoint at the entry of the ADC_Isr function and click Run at Full Speed, and the program will stop there.

Or you can run it directly and see LED2 switching back and forth between on and off.

4.32.2 Directory content

This example can be found in the [TMR1_ComplementaryOutput](#) directory.

4.33 TMR1_ComplementaryOutput

4.33.1 Example Description

This example describes how to configure the TMR1 peripheral to generate complementary TMR1 signals, to insert a defined dead time value.

4.33.2 Directory content

This example can be found in the [TMR1_ComplementaryOutput](#) directory.

4.34 TMR1_InputCapture

4.34.1 Example Description

This example describes how to configure the TMR1 peripheral to capture the internal clock source. PC4 COC pin should be connected to PC3 then LED2 and LED3 will alternate twinkle.

4.34.2 Directory content

This example can be found in the [TMR1_InputCapture](#) directory.

4.35 TMR1_PWMOutput

4.35.1 Example Description

This example describes how to configure the TMR1 peripheral to generate PWM signal.

The waveform can be displayed using an oscilloscope.

4.35.2 Directory content

This example can be found in the [TMR1_PWMOutput](#) directory.

4.36 TMR2_OCActive

4.36.1 Example Description

The program to show how to configure the TMR2 peripheral to generate 3 different signals with three different delays.

TMR2 waveform will be display by oscilloscope.

4.36.2 Directory content

This example can be found in the [TMR2_OCActive](#) directory.

4.37 TMR2_OCInactive

4.37.1 Example Description

The program to show how to configure the TMR2 peripheral in Output Compare Inactive mode.

While the counter is lower than the Output compare registers values, the PC4, PC5, and PC6 are set.

When the counter value reaches the Output compare registers values,

the PC4, PC5, and PC6 are reset.

Reset system and display the PC4, PC5, and PC6 waveform by oscilloscope.

4.37.2 Directory content

This example can be found in the [TMR2_OCInActive](#) directory.

4.38 TMR2_PWMOutput

4.38.1 Example Description

This example describes how to configure the TMR2 peripheral to generate PWM signal.

The waveform can be show be displayed using an oscilloscope by conneceting to PC5(TMR2_CH1).

4.38.2 Directory content

This example can be found in the [TMR2_OCInActive](#) directory.

4.39 TMR2_SinglePulse

4.39.1 Example Description

This example describes how to use the TMR2 to generate a Single Pulse Mode.

Each time KEY1 pressed, a pulse would be generated. The pulse can be detected using an oscilloscope by connecting to PD2.

4.39.2 Directory content

This example can be found in the [TMR2_SinglePulse](#) directory.

4.40 TMR2_SynchronizationWithTMR1

4.40.1 Example Description

This example describes how to use the TMR2 peripheral synchronization by TMR1.

TMR2 is configured as a slave of TMR1.

The waveform can be show be displayed using an oscilloscope.

4.40.2 Directory content

This example can be found in the [TMR2_SynchronizationWithTMR1](#) directory.

4.41 TMR4_TimeBase

4.41.1 Example Description

This example describes how to use the TMR4 peripheral to generate time base.

LED2 and LED3 will twinkle together.

4.41.2 Directory content

This example can be found in the [TMR4_TimeBase](#) directory.

4.42 UART_DMA

4.42.1 Example Description

This example provides an example of how to use UART to print a string by DMA.

After system reset, DMA transfers data to UART1. And UART1 send data to host computer.

4.42.2 Directory content

This example can be found in the [UART_DMA](#) directory.

4.43 UART_HalfDuplex

4.43.1 Example Description

This example provides a basic communication between UART1 and UART2 using half-duplex polling.

After Initialization, UART2 will send data to UART1 using half-duplex polling, if UART1 received data equal to UART2 sent data, LED2 will on, else LED2 will twinkle.

4.43.2 Directory content

This example can be found in the [UART_HalfDuplex](#) directory.

4.44 UART_Interrupt

4.44.1 Example Description

This example provides a basic communication between UART1 and UART2 using interrupts.

After Initialization, UART2 will send data to UART1 using interrupts, if UART1 received data equal to UART2 sent data, LED2 will on, else LED2 will twinkle.

4.44.2 Directory content

This example can be found in the [UART_Interrupt](#) directory.

4.45 UART_Polling

4.45.1 Example Description

This example provides a basic communication between UART1 and UART2 using polling.

After Initialization, UART2 will send data to UART1 using polling, if UART1 received data equal to UART2 sent data, LED2 will on, else LED2 will twinkle.

4.45.2 Directory content

This example can be found in the [UART_Polling](#) directory.

4.46 UART_Printf

4.46.1 Example Description

This example provides how to remap the "printf" function in the C standard library to USART. the USART will use "printf" function send data to uppor computer.

4.46.2 Directory content

This example can be found in the [UART_Printf](#) directory.

4.47 WUPT

4.47.1 Example Description

This example describes how to use WUPT peripheral. Each time the KEY1 is pressed, the MCU will toggle some while than enter STOP mode and OFF all LED. When pressing the KEY1 again, the MCU will be awakened, the LED2 and LED3 will ON.

4.47.2 Directory content

This example can be found in the [WUPT](#) directory.

4.48 WWDT

4.48.1 Example Description

This example describes how to use WWDT in peripheral.

After initialization, System keep feeding watch dog on time, when KEY1 is pressed, system will stop feeding dog, and a reset will be generated. If generate a WWDT reset, LED3 on.

4.48.2 Directory content

This example can be found in the [WWDT](#) directory.

5 About libraries

The libraries folder includes a series library. It can provide supports for APM32F003MCU such as device support and standard peripheral. The libraries can be found in the [Libraries](#) directory.

APM32F003MCU include following library:

- Libraries folder
 - * APM32F003_StdPeriphDriver
 - * CMSIS
 - * Device

6 About middlewares

The middlewares can be found in the [Middlewares](#) directory.

The middlewares used by APM32F003MINI include following:

- Middlewares folder
 - * FreeRTOS
 - * RealThread
 - * RTX5

7 About Package

The Package folder includes Geehy APM32F003_DFP Package and SVD files. The Package can be found in the [Package](#) directory.

The Package used by APM32F003MINI include following:

- Package folder
 - * SVD
 - * Geehy.APM32F003_DFP.1.0.6.pack

8 Revision History

Table 1 File Revision History

Date	Rev	Description
2023.08.31	1.0	First release version of APM32F003SDK
2025.05.31	1.1	Add the examples for ADC and TMR

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. Please 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 “TM” 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, 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 the 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 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 PROVIDE THE DOCUMENT "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 (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY USERS OR THIRD PARTIES).

8. Scope of Application

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

© 2023 Geehy Semiconductor Co., Ltd. - All Rights Reserved