Readme

APM32E10x SDK

Rev：V1.0

1 Introduction

The Geehy Semiconductor APM32E10x MINI 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

2 About boards

The boards folder includes a board support package for APM32E10x MINI 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 APM32E10x MINI board compatibility. For other user development board use, some minor modifications may be required.

Boards have a hierarchy as follows:

- Boards folder
  - Board_APM32E103_MINI
    - inc
    - src
  - Board.c
  - Board.h

Board APM32E103 MINI include following board support package:

- Board_APM32E103_MINI src folder
  - Board_APM32E103_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 APM32E10x MINI 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
    - IAR
    - MDK
  - Source

All example applications tested with: **APM32E10x StdPeriphDriver v1.0.2**, include the following examples:

- Examples
  - ADC
    - ADC_AnalogWatchdog
    - ADC_ContinuousConversion
    - ADC_DMA
    - ADC_DualRegulSimulMode
    - ADC_MultiChannelScan
  - BAKPR
    - BAKPR_Tamperk
  - CAN
    - CAN_LoopBack
    - CAN_Normal
  - CRC
    - CRC_CalcMessage
  - DAC

- DAC_ADC
- DAC_NoiseWave

* DMA
  - DMA_MemoryToMemory
  - DMA_UsartToMemory
* EINT
  - EINT_Config
* EMMC
  - DMC_SDRAM
* FMC
  - FMC_Program
  - FMC_Protection
* FPU
  - FPU_Math
* GPIO
  - GPIO_Toggle
* I2C
  - I2C_TwoBoards
* I2S
  - I2S_Interrupt
* I2S
  - IAP_Application1
  - IAP_Application2
  - IAP_BootLoader
* IWDT
  - IWDT_Reset

NVIC

- NVIC_Priority

- NVIC_WFI

* PMU
  - PMU_Standby
  - PMU_Stop
* RCM
  - RCM_ClockConfig
* RTC
  - RTC_Alarm
  - RTC_Second
* RTOS
  - FreeRTOS
  - RT-thread
  - RTX5
* SPI
  - SPI_FullDuplex
* SysTick
  - SysTick_TimeBase
* Template
  - Template
* TMR
  - TMR_6Steps
  - TMR_32BitCounte
  - TMR_CascadeSynchro
  - TMR_EncoderInterface
  - TMR_ExtTriggerSynchro
  - TMR_OCActive
  - TMR_OCInactive
  - TMR_OCToggle

- TMR_PWMInput
- TMR_PWMOutput
- TMR_SinglePulse
- TMR_TimeBase
- TMR_TMR1DMA
- TMR_TMR1DMABurst

* USART
  - USART_Interrupt
  - USART_IrDA
  - USART_LIN
  - USART_Printf
  - USART_Smartcard
* USB
  - USB_CDC_VirtualCOMPort
  - USB_HID_Mouse
  - USB_MSC_Dis
* WWDT
  - WWDT_Reset

4.1 ADC_AnalogWatchdog

4.1.1 Example Description

This example describes how to use ADC1 to monitor the voltage of ADC1_Channel14 continuously.

If the voltage on ADC1_Channel14(PC4) is not in the thresholds which is set before, analog watchdog interrupt will generate and light LED2.

4.1.2 Directory content

This example can be found in the ADC_AnalogWatchdog directory.

4.2 ADC_ContinuousConversion

4.2.1 Example Description

This example describes how to use the ADC1 to convert continuously the voltage applied to the APM32E103 MINI ADC1_Channel0 input.

The converted voltage is displayed on serial assistant through USART1.

4.2.2 Directory content

This example can be found in the ADC_ContinuousConversion directory.

4.3 ADC_DMA

4.3.1 Example Description

This example provides an example of how to use a DMA channel to transfer continuously a data from a peripheral (ADC1) to DMA transfer.

The ADC channel 0 for APM32E103 MINI Board is configured to be converted when device startup.

The converted voltage is displayed on serial assistant through USART1.

4.3.2 Directory content

This example can be found in the ADC_DMA directory.

4.4 ADC_DualRegulSimulMode

4.4.1 Example Description

This example describes how to use ADC1 and ADC2 in regular simultaneous dual mode.

The ADC1 are configured to convert ADC Channel 0(PA0).

The ADC2 are configured to convert ADC Channel 1(PA1).

4.4.2 Directory content

This example can be found in the ADC_DualRegulSimulMode directory.

4.5 ADC_MultiChannelScan

4.5.1 Example Description

This example describes how to use the ADC1 to scan continuously the voltage applied to the APM32E103 MINI ADC1_Channel0, ADC1_Channel1 and ADC1_Channel2 input.

The converted voltage is displayed on serial assistant through USART1.

4.5.2 Directory content

This example can be found in the ADC_MultiChannelScan directory.

4.6 BAKPR_Tamper

4.6.1 Example Description

This example describes how to write the backup registers. After initialization,

System enters into an infinite loop. if data in the backup registers is equal to the data write to the registers before, LED2 keeps blinking, otherwise, LED3 keeps blinking. TAMPER pin is also enabled, if the pin changes from 0 to 1 or from 1 to 0.

The TAMPER pin generates a Tamper detection event to reset all data backup registers.

4.6.2 Directory content

This example can be found in the BAKPR_Tamper directory.

4.7 CAN_LoopBack

4.7.1 Example Description

This example describes how to configure a communication the CAN in loopback mode. CAN transmit a message to self. Then compare the received message with transmitted message.

4.7.2 Directory content

This example can be found in the CAN_LoopBack directory.

4.8 CAN_Normal

4.8.1 Example Description

This example describes how to configure a communication the CAN. CAN transmit a message in normal mode.

4.8.2 Directory content

This example can be found in the CAN_Normal directory.

4.9 CRC_Calculation

4.9.1 Example Description

Write the calculated data to CRC DATA register and get the calculated result. The phenomenon of ComputedCRC compases ExpectedCRC. The results will be displayed on serial assistant through USART1.

4.9.2 Directory content

This example can be found in the CRC_Calculation directory

4.10 DAC_ADC

4.10.1 Example Description

This example provides an example of how to use DAC channel 1(PA4) to output voltage to ADC channel 0(PA0). The converted voltage of PA4 is detected by ADC channel 0(PA0) and displayed on serial assistant through USART1.

4.10.2 Directory content

This example can be found in the DAC_ADC directory

4.11 DAC_NoiseWave

4.11.1 Example Description

This example shows how to configure the DAC peripheral to generate NoiseWave. The waveform can be displayed using an oscilloscope. using DAC_CHANNEL_1(PA4) to output NoiseWave.

4.11.2 Directory content

This example can be found in the DAC_NoiseWave directory.

4.12 DMA_MemoryToMemory

4.12.1 Example Description

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

4.12.2 Directory content

This example can be found in the DMA_MemoryToMemory directory.

4.13 DMA_UsartToMemory

4.13.1 Example Description

This example provides a basic communication between USART1 and USART2 using DMA1 capability.

After system reset, the DMA transfers data from DMA_USART_TxBuf buffer to USART2. Transmit data register, then this data is sent to USART1. Data received by USART1 is transferred by DMA and stored in DMA_USART_RxBuf. Then compared with the two buffers, If the data of DMA_USART_TxBuf and DMA_USART_RxBuf are the same, LED2 will light, otherwise LED3 will light.

4.13.2 Directory content

This example can be found in the DMA_UsartToMemory directory.

4.14 EINT_Config

4.14.1 Example Description

This example shows how to configure external interrupt lines.

In this example, 2 EINT lines (KEY1, KEY2) when using the APM32E103 MINI BOARD are configured to generate an interrupt on each falling edge. In the interrupt routine a led connected to a specific GPIO pin is toggled.

4.14.2 Directory content

This example can be found in the EINT_Config directory.

4.15 DMC_SDRAM

4.15.1 Example Description

The program shows how to send data by using USART, in this case, USART1 sends data to upper computer. You can check the data in a Serial Port Utility.

4.15.2 Directory content

This example can be found in the DMC_SDRAM directory.

4.16 FMC_Program

4.16.1 Example Description

This example shows how to program the flash address of APM32E103.

After system reset, the Flash will be unlocked. Then erase the specifies address and write a data in the address. In the end, lock the flash. If the data in the address is equal to the data to be written, LED2 will light, otherwise, LED3 will light.

4.16.2 Directory content

This example can be found in the FMC_Program directory.

4.17 FMC_Protection

4.17.1 Example Description

This example shows how to set write protection for the flash address of APM32E103.

Select the 'FLASH_WRITE_PROTECTION' macro, the specific flash address set write protection.

Select the 'FLASH_DISABLE_PROTECTION' macro, the specific flash address remove write protection.

Select the 'FLASH_PAGE_WRITE' macro, the specific flash address will be programed.

4.17.2 Directory content

This example can be found in the FMC_Protection directory.

4.18 FPU_Math

4.18.1 Example Description

This example shows how to use FPU. There are some results about sin, cos and so no which are calculated by FPU. The results will be displayed on serial assistant through USART1.

4.18.2 Directory content

This example can be found in the FPU_Math directory.

4.19 GPIO_Toggle

4.19.1 Example Description

This example describes how to use DOUT for toggling IO. The IO of LED2 and LED3 is configured to toggle constantly. The phenomenon of LED2 and LED3 constantly flickered alternately.

4.19.2 Directory content

This example can be found in the GPIO_Toggle directory.

4.20 I2C_TwoBoards

4.20.1 Example Description

This example shows how to control I2C devices and communicate between two different boards.

4.20.2 Directory content

This example can be found in the I2C_TwoBoards directory.

4.21 I2S_Interrupt

4.21.1 Example Description

This example describes how to use I2S peripheral by making a communication between the I2S2 and the I2S3. If communication success, LED2 and LED3 will turn on.

4.21.2 Directory contents

This example can be found in the I2S_Interrupt directory.

4.22 IAP_Application1

4.22.1 Example Description

This example shows how to generate a APP firmware to IAP. LED2 are toggled with a timing defined by the Delay function.

4.22.2 Directory contents

This example can be found in the IAP_Application1 directory.

4.23 IAP_Application2

4.23.1 Example Description

This example shows how to generate a APP firmware to IAP. LED3 are toggled with a timing defined by the Delay function.

4.23.2 Directory contents

This example can be found in the IAP_Application2 directory.

4.24 IAP_BootLoader

4.24.1 Example Description

The example aims to show how to configure a bootloader firmware to IAP. When device connect to HyperTerminal right, a usart menu will show to user.

4.24.2 Directory contents

This example can be found in the IAP_Application1 directory.

4.25 IWDT_Reset

4.25.1 Example Description

The example shows how to configure IWDT and feed dog to prevent a system reset. After IWDT initialization, System enters into an infinite loop, feed dog before the counter reach a given timeout value to prevent system reset and keep LED2 blinking regularly. Pressing KEY1 to stop feed dog will trigger system reset when the counter reach a given timeout value. LED3 will be lighted when a system reset is triggered by IWDT.

4.25.2 Directory contents

This example can be found in the IWDT_Reset directory.

4.26 NVIC_Priority

4.26.1 Example Description

This example describes how to use NVIC priority.

At startup, press KEY1(PA1) to occur enter EINT1 Interrupt, and device will enter Infinite loop mode.

The device will enter higher priority EINT0 Interrupt if press KEY2.

Now press KEY1 again will not enter EINT1 Interrupt.

The status of device is displayed on serial assistant through USART1.

4.26.2 Directory content

This example can be found in the NVIC_Priority directory

4.27 NVIC_WFI

4.27.1 Example Description

This example describes how to use WFI event to enter sleep mode and wake up using external interrupt.

At startup, press KEY2(PA0) to wait for Interrupt(WFI) event, and device will enter sleep mode. The device will wake up if press KEY2 again.

4.27.2 Directory content

This example can be found in the NVIC_WFI directory.

4.28 PMU_Standby

4.28.1 Example Description

This example shows how to enter STANDBY mode and wake up from this mode through RTC alarm event's rising edge. There is an infinite loop that will keep LED2 blinking in main program which means program is running. Press KEY1, configure RTC alarm event and then system enters STANDBY mode. After a rising edge is generated by RTC alarm event, if system recover to normal state, LED2 keep blinking and light LED3 which means system wake up from STANDBY mode.

4.28.2 Directory content

This example can be found in the PMU_Standby directory.

4.29 PMU_Stop

4.29.1 Example Description

This example shows how to enter STOP mode and wake up from this mode through EINT interrupt. There is an infinite loop that will keep LED2 blinking in main program which means program is running.

Press KEY1, system enters STOP mode and LED2 turn off, LED3 turn on.

Press KEY2, system wake up from stop mode and LED2 blink, LED3 turn off.

4.29.2 Directory content

This example can be found in the PMU_Stop directory.

4.30 RCM_ClockConfig

4.30.1 Example Description

This example shows how to:

- Configure the PLL (clocked by HSE) as System clock source.
- Output the System clock on MCO pin(PA8).

4.30.2 Directory content

This example can be found in the RCM_ClockConfig directory.

4.31 RTC_Alarm

4.31.1 Example Description

This example provides an example of how to use RTC alarm in APM32E103 MINI Board.

There will have an alarm for 5 s when reset system.

And press KEY1 can set an alarm for 5 s.

The data for alarm interrupt and RTC time is showing by USART1.

4.31.2 Directory content

This example can be found in the RTC_Alarm directory.

4.32 RTC_Second

4.32.1 Example Description

The example shows how to use RCT to generate second interrupt.

Download program and then reset (power off to power on).

LED2 will blink every second which is generated by RCT interrupt.

4.32.2 Directory content

This example can be found in the RTC_Second directory.

4.33 FreeRTOS

4.33.1 Example Description

This example describes show how to how to use FreeRTOS create multiple tasks.

4.33.2 Directory content

This example can be found in the FreeRTOS directory.

4.34 RT-thread

4.34.1 Example Description

This example describes how to use RT-Thread for APM32E10x.

The IO of LED2 and LED3 is configured to toggle constantly.

The phenomenon of LED2 and LED3 constantly flickered alternately.

4.34.2 Directory content

This example can be found in the RT-thread directory.

4.35 RTX5

4.35.1 Example Description

This example describes show how to how to use RTX5 create multiple tasks.

4.35.2 Directory content

This example can be found in the RTX5 directory.

4.36 SPI_FullDuplex

4.36.1 Example Description

This example describes how to use SPI peripheral.

By making a board, the master/slave full duplex communication between the SPI and UART1.

If communication success, LED2 will turn on, LED3 will blink.

4.36.2 Directory content

This example can be found in the SPI_FullDuplex directory.

4.37 SysTick_TimeBase

4.37.1 Example Description

The example shows how to configure the SysTick to generate a time base equal to 1ms. A "Delay" function is implemented based on the SysTick end-of-count event which delays exactly half a second, and the LED's on-off state changes every second.

4.37.2 Directory content

This example can be found in the SysTick_TimeBase directory.

4.38 Template

4.38.1 Example Description

The example is routine. Users can add own functional code to main.c.

4.38.2 Directory contents

This example can be found in the Template directory.

4.39 TMR_6Steps

4.39.1 Example Description

The program to show how to configure the TMR1 peripheral to generate 6 Steps. In this example, a software COM event is generated each 100 milliseconds.

The TMR1 is configured in Timing Mode, each time a COM event occurs, a new TMR1 configuration will be set in advance.

4.39.2 Directory contents

This example can be found in the TMR_6Steps directory.

4.40 TMR_32BitCount

4.40.1 Example Description

This example describes how to configure the TMR3 and TMR4 realize the 32-bit timer.

TMR3 as High 16 bit count value, TMR4 as Low 16 bit count value.

User can view the counter value through serial terminal.

4.40.2 Directory contents

This example can be found in the TMR_32BitCount directory.

4.41 TMR_CascadeSynchro

4.41.1 Example Description

This example shows how to synchronize TMR peripherals in cascade mode.

4.41.2 Directory contents

This example can be found in the TMR_CascadeSynchro directory.

4.42 TMR_ExtTriggerSynchro

4.42.1 Example Description

This example shows how to synchronize TMR1 and TMR peripherals in cascade mode with an external trigger.

4.42.2 Directory contents

This example can be found in the TMR_ExtTriggerSynchro directory.

4.43 TMR_EncoderInterface

4.43.1 Example Description

This example describes how to configure the TMR1 peripheral to Encoder mode.

4.43.2 Directory contents

This example can be found in the TMR_EncoderInterface directory.

4.44 TMR_InputCapture

4.44.1 Example Description

This example describes how to use TMR5 Channel2(PA1) measure frequency of external signal.

User can view the "Frequency" value through serial terminal.

4.44.2 Directory contents

This example can be found in the TMR_InputCapture directory.

4.45 TMR_OCActive

4.45.1 Example Description

The program to show how to configure the TMR2 peripheral to generate 4 different signals with four different delays.

4.45.2 Directory contents

This example can be found in the TMR_OCActive directory.

4.46 TMR_OCInactive

4.46.1 Example Description

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

4.46.2 Directory contents

This example can be found in the TMR_OCInactive directory.

4.47 TMR_OCToggle

4.47.1 Example Description

The program to show how to configure the TMR2 peripheral to generate 4 waveform with 4 different frequencies (2.5KHz, 5KHz, 25KHz and 50KHz).

4.47.2 Directory contents

This example can be found in the TMR_OCToggle directory.

4.48 TMR_PWMInput

4.48.1 Example Description

This example describes how to use TMR4 Channel_2 (PB7) measure frequency and duty cycle of external signal.

User can view the "DutyCycle" "Frequency" value through serial terminal.

4.48.2 Directory contents

This example can be found in the TMR_PWMInput directory.

4.49 TMR_PWMOutput

4.49.1 Example Description

This example shows how to configure the TIM1 peripheral to generate PWM signals with different duty cycles. The TMR1 waveform can be displayed using an oscilloscope.

4.49.2 Directory contents

This example can be found in the TMR_PWMOutput directory.

4.50 TMR_SinglePulse

4.50.1 Example Description

This example shows how to configure TMR peripherals to generate a Single Pulse with an external trigger.

4.50.2 Directory content

This example can be found in the TMR_SinglePulse directory..

4.51 TMR_TimeBase

4.51.1 Example Description

This example shows how to realize timing one second by using TMR1 peripheral generating time base.

LED2 will toggle by second.

4.51.2 Directory content

This example can be found in the TMR_TimeBase directory.

4.52 TMR_TMR1DMA

4.52.1 Example Description

The program to show how to use DMA to transfer Data from memory to TMR1 Capture Compare Register1 to change the Duty Cycle.

4.52.2 Directory content

This example can be found in the TMR_TMR1DMA directory.

4.53 TMR_TMR1DMABurst

4.53.1 Example Description

The program to show how to configure the TMR1 channel period and the duty cycle by DMA burst to generate 6 PWM with 6 different duty cycles (80%, 70%, 60%, 40%, 30% and 20%).

4.53.2 Directory content

This example can be found in the TMR_TMR1DMABurst directory.

4.54 USART_Interrupt

4.54.1 Example Description

The program shows how to send data by using USART1 interrupt, and get data by USART2 interrupt.

In this case, USART2 sends data to upper computer. You can check the data in a Serial Port Utility.

4.54.2 Directory content

This example can be found in the USART_Interrupt directory.

4.55 USART_IrDA

4.55.1 Example Description

The program shows how to using USART2 and USART3 in IrDA mode, in this case,

USART1 sends data to upper computer. You can check the data in a Serial Port Utility.

4.55.2 Directory content

This example can be found in the USART_IrDA directory.

4.56 USART_LIN

4.56.1 Example Description

The program shows how to using USART LIN mode, in this case,

USART1 sends data to upper computer. You can check the data in a Serial Port Utility.

4.56.2 Directory content

This example can be found in the USART_LIN directory.

4.57 USART_Printf

4.57.1 Example Description

The program shows how to send data by using USART, in this case, USART1 sends data "Hello USART1" to upper computer. You can check the data in a Serial Port Utility

4.57.2 Directory content

This example can be found in the USART_Printf directory.

4.58 USART_Smartcard

4.58.1 Example Description

The program shows how to using USART Smartcard mode, in this case,

USART1 sends data to upper computer. You can check the data in a Serial Port Utility.

4.58.2 Directory content

This example can be found in the USART_Smartcard directory.

4.59 USB_CDC_VirtualCOMPort

4.59.1 Example Description

This example describes how to use the USB device module on APM32E103 to enumerate as a Virtual Com Port. This example use PC as host, you can use serial assistant to transfer USB data. Once serial assistant send data to device through the Virtual Com Port that USB enumerated, then device will send the same data back to PC.

4.59.2 Directory content

This example can be found in the USB_CDC_VirtualCOMPort directory.

4.60 USB_HID_Mouse

4.60.1 Example Description

This example shows how to use USB. Download the program, then connect the device to your computer through USB. If the USB is configured correctly, you can see a new HID-mouse in your computer.

4.60.2 Directory content

This example can be found in the USB_HID_Mouse directory.

4.61 USB_MSC_Disk

4.61.1 Example Description

This example describes how to use the USB device module on APM32E103 to enumerated as a MSC USB disk. This example use PC as host, and APM32E103 use ram to simulate usb flash drives. PC will recognize the motherboard as a disk, and formatting the U disk. This example will be Used to test usb speed.

4.61.2 Directory content

This example can be found in the USB_MSC_Disk directory.

4.62 WWDT_Reset

4.62.1 Example Description

This example aims to show how to use WWDT.

If is_OverTime = 0 , System would not reset for feeding dog timely. LED2 Toggle. Pressing KEY1 to stop feed dog will trigger system reset when the counter reach a given timeout value. LED3 will be lighted when a system reset is triggered by IWDT.

4.62.2 Directory content

This example can be found in the WWDT_Reset directory.

5 About libraries

The libraries folder includes a series library. It can provide supports for APM32E10x MCU such as device support and standard peripheral. The libraries can be found in the Libraries directory.

APM32E10x MCU include following library:

- Libraries folder
  - APM32E10x_StdPeriphDriver
  - CMSIS
  - Device
  - USB_Device_Lib

6 About middlewares

The middlewares can be found in the Middlewares directory.

The middlewares used by APM32E10x MINI include following:

- FreeRTOS
- RealThread
- RTX5
- fat_fs

7 About Package

The Package folder includes Geehy APM32E10x_DFP Package. The Package can be found in the Package directory.

The Package used by APM32E10x MINI include following:

- Package folder
  - Geehy.APM32E10x_DFP.1.0.2.pack

8 Revision History

Table 1 File Revision History

Date	Rev	Description
2021.07.29	1.0	First release version of APM32E10x SDK V1.0
2022.08.08	1.1	Update Package for 1.0.1. Update the folder directory structure. Update EMMC and TMR Example. Added IAR Support. Added USART and RTOS Example.
2022.12.31	1.2	Update Package for 1.0.2. Update Library for 1.0.2. Update Examples. Added Eclipse Support.

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 thesales 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 loses 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 COUNTIRIES 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.

Geehy Semiconductor Co.,Ltd.
Bldg.1, No.83 Guangwan Street, Zhuhai, Guangdong, China +86 0756 6299999 www.geehy.com