Readme

APM32F00x SDK

Rev：V1.0

© Geehy Semiconductor Co., Ltd.

1 Introduction

The Geehy Semiconductor APM32F00x 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 APM32F00x 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 APM32F00x MINI 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 APM32F003MINI include following board support package:

- Board_ APM32F003_MINI src folder
  - Board_ APM32F003_MINI
  - bsp_delay

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 APM32F00x 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
    - Eclipse
  - Source

All example applications tested with: **APM32F00x StdPeriphDriver v1.0**, include the following examples:

- Examples
  - ADC
    - ADC_Channel_2
    - ADC_Channel_8
    - ADC_ContinuousConversion
    - ADC_MultiChannelScan
  - BUZZER
    - BUZZER
  - EINT
    - EINT
  - FMC
    - FMC_AFR
    - FMC_OPT
    - FMC_Program

* GPIO
  - GPIO_AF
  - GPIO_Toggle
* I2C
  - I2C_TwoBoardsInterrupt
  - I2C_TwoBoardsPolling
* IAP
  - Application1
  - Application2
  - Bootloader
* IWDT
  - IWDT
* NVIC
  - NVIC
* RCM
  - RCM_ClockSwitch
  - RCM_CRS
* RTOS
  - FreeRTOS
  - RT-Thread
  - RTX
* SPI
  - SPI_FullDuplex
* SysTick
  - SysTick
* Template
  - Template
* TMR1

- TMR1_6stepswithTMR2
- TMR1_ComplementaryOutput
- TMR1_InputCapture
- TMR1_PWMOutput
- TMR1_PWMOutput_AFR

* TMR2
  - TMR2_OCActive
  - TMR2_OCInactive
  - TMR2_SinglePulse
  - TMR2_SynchronizationWithTMR1
* TMR4
  - TMR4_TimeBase
* USART
  - USART_HalfDuplex
  - USART_Interrupt
  - USART_MultiProcessor
  - USART_Polling
  - USART_Printf
  - USART_Synchronous
* 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 always be printed on USART1.

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_MultiChannelScan

4.4.1 Example Description

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

After Initialization, PC5/PC6/PC4 start to convert voltage. The value of ADC channels will always be printed on USART1.

4.4.2 Directory content

This example can be found in the ADC_MultiChannelScan directory.

4.5 BUZZER

4.5.1 Example Description

This example describes how to use BUZZER peripheral.

Each time the key1 button is pressed the BUZZER frequency changes and a corresponding led will be switched on.

When Buzzer output clock frequency is 1khz, LED2 is on and LED3 is off.

When Buzzer output clock frequency is 2khz, LED2 is off and LED3 is on.

When Buzzer output clock frequency is 4khz, LED2 is on and LED3 is on.

The waveform can be show be displayed using an oscilloscope by connecting to PD4.

4.5.2 Directory content

This example can be found in the BUZZER directory.

4.6 EINT

4.6.1 Example Description

This example describes how to use EINT peripheral.

Each time the key1 button is pressed, the LED2 and LED3 will toggle.

4.6.2 Directory content

This example can be found in the EINT directory

4.7 FMC_AFR

4.7.1 Example Description

This example describes how to use GPIO Alternate function and configure the TMR1 peripheral to generate PWM signal.

PC6 is set to TMR1_CH1, while PC3 is set to TMR1_CH1N.

The waveform can be show be displayed using an oscilloscope by connecting to PC6 and PC3.

4.7.2 Directory content

This example can be found in the FMC_AFR directory

4.8 FMC_OPT

4.8.1 Example Description

This example describes how to use OPT download algorithm to modify option byte, the result will show by USART1.

4.8.2 Directory content

This example can be found in the FMC_OPT directory.

4.9 FMC_Program

4.9.1 Example Description

This example describes how to use Program flash.

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

4.9.2 Directory content

This example can be found in the FMC_Program directory.

4.10 GPIO_AF

4.10.1 Example Description

This example describes how to Configure OSCIN and OSCOUT pin as normal GPIO peripheral. After LED initalization, LED2 and LED3 on, and OSCIN and OSCOUT pin are configured as normal pin, system will try to let LED3 off. after change system clock source, system will try to toggle LEDs.

4.10.2 Directory content

This example can be found in the GPIO_AF directory.

4.11 GPIO_Toggle

4.11.1 Example Description

This example describes how to use GPIO peripheral.

LED2 and LED3 will alternate twinkle.

4.11.2 Directory content

This example can be found in the GPIO_Toggle directory.

4.12 I2C_TwoBoardsInterrupt

4.12.1 Example Description

This example describes how to use I2C peripheral in interrupt mode.

It is an example of how to use the I2C software library to ensure the steps of an I2C communication between slave Receiver/transmitter and master transmitter/receiver using interrupts.

4.12.2 Directory content

This example can be found in the I2C_TwoBoardsInterrupt directory.

4.13 I2C_TwoBoardsPolling

4.13.1 Example Description

This example describes how to use I2C peripheral in polling mode.

It is an example of how to use the I2C software library to ensure the steps of an I2C communication between slave Receiver/transmitter and master transmitter/receiver using polling.

4.13.2 Directory content

This example can be found in the I2C_TwoBoardsPolling directory.

4.14 Application1

4.14.1 Example Description

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

4.14.2 Directory contents

This example can be found in the Application1 directory.

4.15 Application2

4.15.1 Example Description

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

4.15.2 Directory contents

This example can be found in the Application2 directory.

4.16 BootLoader

4.16.1 Example Description

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

4.16.2 Directory contents

This example can be found in the BootLoader directory.

4.17 IWDT

4.17.1 Example Description

This example describes how to use IWDT peripheral.

The IWDT time out is set to 0.5s. After initialization, system keep feed dog and the LED2 keep twinkle. If KEY1 is pressed, system will stop feed dog and the IWDT will generate a reset and LED3 on.

4.17.2 Directory contents

This example can be found in the IWDT directory.

4.18 NVIC

4.18.1 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.

4.18.2 Directory contents

This example can be found in the NVIC directory.

4.19 RCM_ClockSwitch

4.19.1 Example Description

This example describes how to switch the master clock.

Pressing KEY1, master clock will switch between HXT and HIRC. The clock can be visualized on CCO pin.

4.19.2 Directory content

This example can be found in the RCM_ClockSwitch directory.

4.20 RCM_CRS

4.20.1 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.

4.20.2 Directory content

This example can be found in the RCM_CRS directory.

4.21 FreeRTOS

4.21.1 Example Description

This example describes how to use FreeRTOS create multiple tasks.

Usart test task: USART1 received data from upper computer and then keep sending the received data to upper computer.

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

4.21.2 Directory content

This example can be found in the FreeRTOS directory.

4.22 RT-Thread

4.22.1 Example Description

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

LED Toggle Task: The LED2 and LED3 constantly flickered alternately.

4.22.2 Directory content

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

4.23 RTX

4.23.1 Example Description

This example describes how to use RTX5 create multiple tasks.

Usart test task: USART1 and USART2 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 received data fault, LED3 will be constantly flickered alternately.

Led toggle task: The LED2 is configured to toggle constantly

4.23.2 Directory content

This example can be found in the RTX directory.

4.24 SPI_FullDuplex

4.24.1 Example Description

This example describes how to use SPI peripheral.

Making a master/slave full duplex communication between the SPI and the UART1.

4.24.2 Directory content

This example can be found in the SPI_FullDuplex directory.

4.25 SysTick

4.25.1 Example Description

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

4.25.2 Directory content

This example can be found in the SysTick directory.

4.26 Template

4.26.1 Example Description

This example provides a template project.

4.26.2 Directory contents

This example can be found in the Template directory.

4.27 TMR1_6stepswithTMR2

4.27.1 Example Description

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

4.27.2 Directory contents

This example can be found in the TMR1_6stepswithTMR2 directory.

4.28 TMR1_ComplementaryOutput

4.28.1 Example Description

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

PC6 is set to TMR1_CH1, while PC3 is set to TMR1_CH1N.

The waveform can be show be displayed using an oscilloscope by connecting to PC6 and PC3.

4.28.2 Directory contents

This example can be found in the TMR1_ComplementaryOutput directory.

4.29 TMR1_InputCapture

4.29.1 Example Description

This example describes how to configure the TMR1 peripheral to capture the internal clock source. PB4(COC pin) should be connected to PC3, then LED2 and LED3 will alternate twinkle.

4.29.2 Directory contents

This example can be found in the TMR1_InputCapture directory.

4.30 TMR1_PWMOutput

4.30.1 Example Description

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

The waveform can be show be displayed using an oscilloscope by connecting to PC3(TM R1_CH3).

4.30.2 Directory contents

This example can be found in the TMR1_PWMOutput directory.

4.31 TMR1_PWMOutput_AFR

4.31.1 Example Description

This example describes how to configure the TMR1 peripheral to generate PWM signal. The waveform can be show be displayed using an oscilloscope by connecting to PC6(TM R1_CH1).

4.31.2 Directory contents

This example can be found in the TMR1_PWMOutput_AFR directory.

4.32 TMR2_OCActive

4.32.1 Example Description

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

4.32.2 Directory content

This example can be found in the TMR2_OCActive directory..

4.33 TMR2_OCInactive

4.33.1 Example Description

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

4.33.2 Directory content

This example can be found in the TMR2_OCInactive directory..

4.34 TMR2_SinglePulse

4.34.1 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 PA3.

4.34.2 Directory content

This example can be found in the TMR2_SinglePulse directory..

4.35 TMR2_SynchronizationWithTMR1

4.35.1 Example Description

This example describes how to use the TMR2 peripheral synchronized by TMR1. TMR2 is configured as a slave of TMR1.

The waveform can be show be displayed using an oscilloscope by connected to PD3.

4.35.2 Directory content

This example can be found in the TMR2_SynchronizationWithTMR1 directory.

4.36 TMR4_TimeBase

4.36.1 Example Description

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

LED2 and LED3 will twinkle together.

4.36.2 Directory content

This example can be found in the TMR4_TimeBase directory.

4.37 USART_HalfDuplex

4.37.1 Example Description

This example provides a basic communication between USART1 and USART2 using halt-duplex polling. After Initialization, USART2 will send some data to USART1 using halt-duplex polling.

If USART1 received data equal to USART2 sent data, LED2 will on, else LED2 and LED3 will twinkle.

4.37.2 Directory content

This example can be found in the USART_HalfDuplex directory.

4.38 USART_Interrupt

4.38.1 Example Description

This example provides a basic communication between USART1 and USART2 using interrupts. After Initialization, USART2 will send some data to USART1 using interrupts.

4.38.2 Directory content

This example can be found in the USART_Interrupt directory.

4.39 USART_MultiProcessor

4.39.1 Example Description

This example provides how to establish multi-Processor communication in USART.

In this example, USART1 as the slave and enter mute mode, USART2 as the master will send the address flame to wake up USART1, and then send data. If USART1 received data equal to USART2 sent data, LED2 will on, else LED2 will twinkle.

4.39.2 Directory content

This example can be found in the USART_MultiProcessor directory.

4.40 USART_Polling

4.40.1 Example Description

This example provides a basic communication between USART1 and USART2 using polling. After Initialization, USART2 will send some data to USART1 using polling.

4.40.2 Directory content

This example can be found in the USART_Polling directory.

4.41 USART_Printf

4.41.1 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 upper computer.

4.41.2 Directory content

This example can be found in the USART_Printf directory.

4.42 USART_Synchronous

4.42.1 Example Description

This example describes how to use USART peripheral Synchronous mode. It will make a full duplex communication between the SPI and the UART1. The SPI as the slave and the USART1 as the master.

4.42.2 Directory content

This example can be found in the USART_Synchronous directory.

4.43 WUPT

4.43.1 Example Description

This example describes how to use WUPT peripheral.

Each time the key1 be pressed, the MCU will toggle some while than enter STOP mode and OFF all LED. When the MCU be awakened, the LED2 will OFF and LED3 will ON.

4.43.2 Directory content

This example can be found in the WUPT directory.

4.44 WWDT

4.44.1 Example Description

This example describes how to use WWDT in peripheral.

When KEY1 be pressed, WWDT counter disable update, and a reset will be generated. If generate a WWDT reset, only LED2 twinkle.

4.44.2 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 APM32F00x MCU such as device support and standard peripheral. The libraries can be found in the Libraries directory.

APM32F00x MCU include following library:

- Libraries folder
  - APM32F00x_StdPeriphDriver
  - CMSIS
  - Device

6 About middlewares

The middlewares can be found in the Middlewares directory.

The middlewares used by APM32F00x MINI include following:

- FreeRTOS
- RealThread
- RTX5

7 About Package

The Package folder includes Geehy APM32F00x_DFP Package. The Package can be found in the Package directory.

The Package used by APM32F00x MINI include following:

- Package folder
  - Geehy.APM32F00x_DFP.1.0.5.pack

8 Revision History

Table 1 File Revision History

Date	Rev	Description
2022.06.04	1.0.0	First release version of APM32F00x SDK V1.0
2020.06.29	1.0.1	Add USART3
2020.10.19	1.0.2	Update Library： 1.Modify function name in GPIO module. 2.Modify function reset value in ADC，RCM module. 3.Add AFR5 function in FLASH module. 4.Modify register name in TMR2, RCM, SPI module. Update Package： Geehy.APM32F00x_DFP.1.0.3.pack Add Example： ADC（Channel_2）,FLASH（AFR）, GPIO(GPIOAF)，Template.
2022.04.11	1.0.3	Update Library： 1.Update Library for V1.0.1. 2.Add copyright notice and rewrite readme to all examples. 3.Add IAP and ADC and USART examples. Update Package： Geehy.APM32F00x_DFP.1.0.4.pack.
2023.03.31	1.0.4	Update Library： 1.Update Library for V1.0.2. 2. Update Boards for V1.0.2 3.Add Eclipse and OPT, NVIC, RCM, TMR, USART examples. Update Package： Geehy.APM32F00x_DFP.1.0.5.pack.

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.

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

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