

Readme

APM32S10x SDK

版本: V1.0

1 Introduction

The Geehy Semiconductor apm32s10x 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	4
3	About documents.....	5
4	About examples.....	6
4.1	ADC_AnalogWatchdog	10
4.2	BAKPR_Tamper	10
4.3	CAN_LoopBack	10
4.4	CAN_TwoBoards	11
4.5	CRC_Calculation.....	11
4.6	DMA_MomeryToMomery	11
4.7	EINT_Config	12
4.8	FMC_Program.....	12
4.9	FPU_Math.....	12
4.10	GPIO_Toggle	13
4.11	I2C_TwoBoards.....	13
4.12	Application1.....	13
4.13	Application2.....	14
4.14	BootLoader	14
4.15	IWDT_Reset	15
4.16	NVIC_WFI.....	15
4.17	PMU_Standby.....	16
4.18	RCM_ClockConfig	16
4.19	RTC_Second	16
4.20	SPI_FullDuplex	16
4.21	SysTick_TimeBase.....	17
4.22	Template	17

4.23	TMR_32BitCount	17
4.24	TMR_EncoderInterface	18
4.25	TMR_PWMOutput	18
4.26	TMR_TimeBase	18
4.27	USART_Printf	18
4.28	USB_CDC_VirtualCOMPort	19
4.29	USB_HID_Mouse	19
4.30	USB_MSC_Disk	19
4.31	WWDT_Reset	20
5	About libraries	21
6	About middlewares	22
7	About package.....	23
8	Revision History	24

2 About boards

The boards folder includes a board support package for APM32S103 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 APM32S103 MINI board compatibility. For other user development board use, some minor modifications may be required.

Boards have a hierarchy as follows:

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

Board APM32S103 MINI include following board support package:

- Board_APM32S103_MINI src folder
 - * Board_APM32S103_MINI

3 **About documents**

The documents folder includes a link file that can be redirected to the technical support center of Geehy semiconductor. The BSP 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 APM32S103 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: **APM32S10x_StdPeriphDriver**, include the following examples:

Table 1 APM32S103 examples of peripherals

APM32S10x_SDK_V1.0				
ADC	AnalogWatchdog			
BAKPR	Tamper			
CAN	LoopBack	TwoBoards		
CRC	Calculation			
DMA	MomeryToMomery			
EINT	Config			
FMC	Program			
GPIO	Toggle			
I2C	I2C_TwoBoards			
I2P	Application1	IApplication2	BootLoader	
IWDT	Reset			
PMU	Standby			
RCM	ClockConfig			
SPI	FullDuplex			
SYSTICK	TimeBase			
Template	Template			
TMR	32BitCount	EncoderInterface	PWMOutput	TimeBase
USART	Printf			
USB	USB_CDC_VirtualCOMPort	USB_HID_Mouse	USB_MSC_Disk	
WWDT	Reset			

■ Examples

- * ADC
 - [ADC_AnalogWatchdog](#)
- * BAKPR
 - [BAKPR_Tamper](#)
- * CAN
 - [CAN_LoopBack](#)
 - [CAN_TwoBoards](#)
- * CRC
 - [CRC_Calculation](#)
- * DMA
 - [DMA_MomeryToMomery](#)
- * EINT
 - [EINT_Config](#)
- * FMC
 - [FMC_Program](#)
- * FPU
 - [FPU_Math](#)
- * GPIO
 - [GPIO_Toggle](#)
- * I2C
 - [I2C_TwoBoards](#)
 - I2C_TwoBoards_Master
 - I2C_TwoBoards_Slave
- * IAP
 - [Application1](#)
 - [Application2](#)
 - [BootLoader](#)
- * IWDT

- [IWDT_Reset](#)
- * NVIC
 - [NVIC_WFI](#)
- * PMU
 - [PMU_Standby](#)
- * RCM
 - [RCM_ClockConfig](#)
- * RTC
 - [RTC_Second](#)
- * SPI
 - [SPI_FullDuplex](#)
- * SysTick
 - [TimeBase](#)
- * Template
 - [Template](#)
- * TMR
 - [TMR_32BitCount](#)
 - [TMR_EncoderInterface](#)
 - [TMR_PWMOutput](#)
 - [TMR_TimeBase](#)
- * USART
 - [USART_Printf](#)
- * USB
 - [USB_CDC_VirtualCOMPort](#)
 - [USB_HID_Mouse](#)
 - [USB_MSC_Disk](#)
- * 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 setted before, analog watchdog interrupt will generate and light LED2.

4.1.2 Directory contents

This example can be found in the [~ \Examples\ADC\ AnalogWatchdog](#) directory.

4.2 BAKPR_Tamper

4.2.1 Example Description

This This example describes how to write the backup registers. After initialization, System enters into a infinite loop. if data in the backup registers equal to the data write to the registers before, LED2 keeps blinking, otherwise, LED3 keeps bliinking. 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.

4.2.2 Directory contents

This example can be found in the [~ \Examples\BAKPR\Tamper](#) directory.

4.3 CAN_LoopBack

4.3.1 Example Description

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

- Polling transmit success: The LED2 turns, otherwise LED2 turns off.
- Interrupt transmit success: The LED3 turns, otherwise LED3 turns off.

4.3.2 Directory contents

This example can be found in the [~ \Examples\CAN\LoopBack](#) directory.

4.4 CAN_TwoBoards

4.4.1 Example Description

This example describes how to config a communication the CAN. CAN transmit a message from one board to another.

Transmit:

If press the KEY1,CAN1 transmit the USART1 Printf “CAN1 Transmit 0x55” ,LED2 turns on.

If press the KEY2,CAN2 transmit the USART1 Printf “CAN2 Transmit 0xAA” ,LED2 turns on.

Receive:

If CAN1 receive success, the USART1 Printf“CAN1 Receive 0x55”, LED2 turns on.else “CAN1 Receive Error ”, LED3 turns on.

If CAN2 receive success, the USART1 Printf“CAN2 Receive 0xAA”,LED2 turns on.else “CAN2 Receive Error ”, LED3 turns on.

4.4.2 Example Description

This example can be found in the [~ \Examples\CAN\TwoBoards](#) directory.

4.5 CRC_Calculation

4.5.1 Example Description

This example describes how to Write the calculated data to CRC DATA register and get the calculated result.The phenomenon of Computed CRC compases Expected CRC. The results will be displayed on serial assistant through USART1.

4.5.2 Directory contents

This example can be found in the [~ \Examples\CRC\Calcuation](#) directory.

4.6 DMA_MomeryToMomery

4.6.1 Example Description

This example describes how to configure the DMA peripheral to transmit data from momery to momery. 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.

4.6.2 Directory contents

This example can be found in the [~\Examples\DMA\MomeryToMomery](#) directory.

4.7 EINT_Config

4.7.1 Example Description

This example describes how to configure external interrupt lines. In this example, 2 EINT lines (KEY1、KEY2) when using the APM32S103 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. In this example

- EINT0 is mapped to PA0(KEY2) - EINT1 is mapped to PA1(KEY1)

when falling edge is detected on EINT0 , LED2 toggles

when falling edge is detected on EINT1 , LED3 toggles.

4.7.2 Directory contents

This example can be found in the [~\Examples\EINT\Config](#) directory.

4.8 FMC_Program

4.8.1 Example Description

This example describes how to program the flash address of APM32S103. After system reset, the Flash will be unlock. 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.

4.8.2 Directory contents

This example can be found in the [~\Examples\FMC\Program](#) directory.

4.9 FPU_Math

4.9.1 Example Description

This example describes how to use FPU. There are some results about sin, cos and so on which are calculated by FPU. The results will be displayed on serial assistant through USART1.

4.9.2 Directory contents

This example can be found in the [~ \Examples\FPU\Math](#) directory.

4.10 GPIO_Toggle

4.10.1 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.

4.10.2 Directory contents

This example can be found in the [~ \Examples\GPIO\Toggle](#) directory.

4.11 I2C_TwoBoards

4.11.1 Example Description

This example describes how to control I2C devices and communicate between two different boards. To use this example, you need to load the software into two APM32S103_MINI boards (let's call them Board master and Board Slave) then connect these two boards through I2C lines and GND.

- At startup, Boards master and slave are both in slave receiver mode and wait for messages to be received.

- When KEY1 is pressed on Board master, the master transmitter sent "Hello slave" to Board slave. The message is displayed on serial assistant through USART1 on Board slave.

- When KEY1 is pressed on Board slave, the slave transmitter sent "Hello master" to Board master. The message is displayed on serial assistant through USART1 on Board master.

4.11.2 Directory contents

This example can be found in the [~ \Examples\I2C\TwoBoards](#) directory.

4.12 Application1


```
* 1.Download Flash application 1 -----> 1 *
* 2.Upload Flash application 1 -----> 2 *
* 3.Jump to user application 1 -----> 3 *
* 4.Download Flash application 2 -----> 4 *
* 5.Upload Flash application 2 -----> 5 *
* 6.Jump to user application 2 -----> 6 *
```

4.14.2 Directory contents

This example can be found in the [~ \Examples\IAP\BootLoader](#) directory.

4.15 IWDT_Reset

4.15.1 Example Description

The example describes how to configure IWDT and feed dog to prevent a system reset. After IWDT initialization, System enters into 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.

4.15.2 Directory contents

This example can be found in the [~ \Examples\IWDT\Reset](#) directory.

4.16 NVIC_WFI

4.16.1 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 occur Wait For Interrupt(WFI) event, and device will enter sleep mode. The device will wake up if press KEY2 again.phenomenon:

- program running: LED2, LED3 blink. - sleep mode: LED2, LED3 off.

4.16.2 Directory contents

This example can be found in the [~ \Examples\NVIC\WFI](#) directory.

4.17 PMU_Standby

4.17.1 Example Description

This example describes how to enter STANDBY mode and wake up from this mode through RTC alarm event's rising edge. There is a infinite loop that will keep LED2 blinking in mian 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,Directory contents

This example can be found in the [~ \Examples\PMU\Standby](#) directory.

4.18 RCM_ClockConfig

4.18.1 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)

4.18.2 Directory contents

This example can be found in the [~ \Examples\RCM\ClockConfig](#) directory.

4.19 RTC_Second

4.19.1 Example Description

The example describes 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.

4.19.2 Directory contents

This example can be found in the [~ \Examples\RTC\Second](#) directory.

4.20 SPI_FullDuplex

4.20.1 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.

4.20.2 Directory contents

This example can be found in the [~ \Examples\SPI\FullDuplex](#) directory.

4.21 SysTick_TimeBase

4.21.1 Example Description

This example describes how to configure the SysTick to generate a time base equal to 1 ms. 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.

4.21.2 Directory contents

This example can be found in the [~ \Examples\SysTick\TimeBase](#) directory.

4.22 Template

4.22.1 Example Description

This example provides a template project.

4.22.2 Directory contents

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

4.23 TMR_32BitCount

4.23.1 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.

4.23.2 Directory contents

This example can be found in the [~\Examples\TMR\32BitCount](#) directory.

4.24 TMR_EncoderInterface

4.24.1 Example Description

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

using Input Capture Channel TMR1_CH1(PA8),TMR1_CH2(PA9).

4.24.2 Directory contents

This example can be found in the [~\Examples\TMR\EncoderInterface](#) directory.

4.25 TMR_PWMOutput

4.25.1 Example Description

This example describes how to configure the TIM1 peripheral to generate PWM signals with different duty cycles.The TMR1 waveform can be displayed using an oscilloscope. using TMR1 CHANNEL1(PA8) to output PWM.

4.25.2 Directory contents

This example can be found in the [~\Examples\TMR\PWMOutput](#) directory.

4.26 TMR_TimeBase

4.26.1 Example Description

This example describes how to realize timing one second by using TMR1 peripheral generating time base.LED2 will toggle pre second.

4.26.2 Directory contents

This example can be found in the [~\Examples\TMR\TimeBase](#) directory.

4.27 USART_Printf

4.27.1 Example Description

This program describes 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

4.27.2 Directory contents

This example can be found in the [~ \Examples\USART\Printf](#) directory.

4.28 USB_CDC_VirtualCOMPort

4.28.1 Example Description

This example describes how to use the USB device module on APM32S103 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.

4.28.2 Directory contents

This example can be found in the [~ \Examples\USB\USB_CDC_VirtualCOMPort](#) directory.

4.29 USB_HID_Mouse

4.29.1 Example Description

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

4.29.2 Directory contents

This example can be found in the [~ \Examples\USB\USB_HID_Mouse](#) directory.

4.30 USB_MSC_Disk

4.30.1 Example Description

This example describes how to use the USB device module on APM32S103 to enumerated as a MSC USB disk. This example use PC as host, and APM32S103 use ram to simulate usb flash drives. PC will recognizes the motherboard as a disk, and formatting the U disk. This example will be Used to test usb speed.

4.30.2 Directory contents

This example can be found in the [~ \Examples\USB\USB_MSC_Disk](#) directory.

4.31 WWDT_Reset

4.31.1 Example Description

This example describes 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 IWDG.

4.31.2 Directory contents

This example can be found in the [~ \Examples\WWDT\Reset](#) directory.

5 About libraries

The libraries folder includes a series library. It can provide supports for APM32S103 MINI such as device support and standard peripheral etc. The libraries can be found in the [~/Libraries](#) directory.

APM32S103 include following library:

- Libraries folder
 - * APM32S10x_StdPeriphDriver
 - * CMSIS
 - * Device
 - * USB_Device_Lib

6 About middlewares

The middlewares folder includes a series third-party middleware. The middlewares can be found in the [~/Middlewares](#) directory.

7 About package

The package folder includes a APM32S103 Device Family Pack(DFP). The DFP can be found in the [~/Package](#) directory.

8 Revision History

Table 1 File Revision History

Date	Rev	Description
2022.06.07	1.0	First Release version of APM32S10x_SDK

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.

© 2022 Geehy Semiconductor Co., Ltd. - All Rights Reserved