

Readme

APM32F402_403 SDK

Rev: V1.0

1 Introduction

The Geehy Semiconductor APM32F402_403 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	6
3	About documents.....	7
4	About examples.....	8
4.1	ADC_AnalogWindowWatchdog.....	13
4.2	ADC_ContinuousConversion.....	13
4.3	ADC_DMA.....	13
4.4	ADC_DualRegulSimulMode	13
4.5	ADC_MultiChannelScan.....	14
4.6	ADC_TemperatureSensor	14
4.7	BAKPR_Tamper	15
4.8	CAN_LoopBack	15
4.9	CAN_Normal.....	16
4.10	CRC_Calculation.....	16
4.11	DMA_MemoryToMemory	16
4.12	EINT_Config.....	16
4.13	FMC_Protection.....	17
4.14	FMC_Protection_RAM.....	17
4.15	FMC_Read_Write	17
4.16	FMC_Write	18
4.17	GPIO_Toggle	18
4.18	I2C_TwoBoardsDMA.....	18
4.19	I2C_TwoBoardsInterrupt	19
4.20	I2C_TwoBoardsPolling	19
4.21	I2S_TwoBoardsInterrupt.....	20
4.22	Application1.....	20

4.23	Application2.....	21
4.24	BootLoader	21
4.25	IWDT_Reset	21
4.26	NVIC_Priority	22
4.27	NVIC_WFI.....	22
4.28	PMU_Standby.....	22
4.29	PMU_Stop.....	23
4.30	PMU_Consumption	23
4.31	RCM_ClockConfig	24
4.32	RTC_Alarm	24
4.33	RTC_Second	24
4.34	CMSIS_FreeRTOS	24
4.35	CMSIS_RTX.....	25
4.36	SPI_FullDuplex	25
4.37	SysTick_TimeBase.....	25
4.38	Template	26
4.39	TMR_32BitCount	26
4.40	TMR_6Steps.....	26
4.41	TMR_CascadeSynchro	27
4.42	TMR_EncoderInterface.....	27
4.43	TMR_ExtTriggerSynchro	27
4.44	TMR_InputCapture	27
4.45	TMR_OCActive.....	28
4.46	TMR_OCInactive	28
4.47	TMR_OCToggle	28
4.48	TMR_ParallelSynchro.....	29
4.49	TMR_PWMInput.....	29
4.50	TMR_PWMOutput.....	29

4.51	TMR_SinglePulse	29
4.52	TMR_TimeBase	30
4.53	TMR_TMR1DMABurst	30
4.54	TMR_TMR1PWMOutput	30
4.55	TMR_TMR1Synchro	30
4.56	TMR_TMR2PWMOutput	31
4.57	TMR_TMR8DMA	31
4.58	USART_IrDA	31
4.59	USART_LIN	32
4.60	USART_SmartCard	32
4.61	USART_TwoBoardsDMA	32
4.62	USART_TwoBoardsInterrupt	32
4.63	USART_TwoBoardsPolling	33
4.64	OTGD_CDC	33
4.65	OTGD_Composite_CDC	33
4.66	OTGD_Composite_CDC_WINUSB	34
4.67	OTGD_Custom_HID	34
4.68	OTGD_Custom_HID_Keyboard	34
4.69	OTGD_HID	35
4.70	OTGD_HID_Keyboard	35
4.71	OTGD_MSC	35
4.72	OTGD_MSC_IAP	35
4.73	OTGD_WINUSB	36
4.74	OTGH_CDC	36
4.75	OTGH_DynamicSwitch	36
4.76	OTGH_HID	37
4.77	OTGH_MSC	37
4.78	OTGH_MSC_FWUpgrade	37

4.79	WWDT_OverTime.....	37
5	About libraries	39
6	About middlewares	40
7	About Package	41
8	Revision History	42

2 About boards

The boards folder includes a board support package for APM32F402_403 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 APM32F402_403 board compatibility. For other user development board use, some minor modifications may be required.

Boards have a hierarchy as follows:

- Board_APM32F402_403_Tiny folder
 - * Include
 - board_apm32f402_403_tiny.h
 - board_delay.h
 - * Source
 - board_apm32f402_403_tiny.c
 - board_delay.c

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 APM32F402_403 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: **APM32F402_403 StdPeriphDriver V1.0.0**, include the following examples:

- Examples
 - * ADC
 - [ADC_AnalogWindowWatchdog](#)
 - [ADC_ContinuousConversion](#)
 - [ADC_DMA](#)
 - [ADC_DualRegulSimulMode](#)
 - [ADC_MultiChannelScan](#)
 - [ADC_TemperatureSensor](#)
 - * BAKPR
 - [BAKPR_Tamper](#)
 - * CAN
 - [CAN_LoopBack](#)
 - [CAN_Normal](#)
 - * CRC

- [CRC Calculation](#)
- * DMA
 - [DMA MemoryToMemory](#)
- * EINT
 - [EINT Config](#)
- * FMC
 - [FMC Protection](#)
 - [FMC Protection_RAM](#)
 - [FMC Read_Write](#)
 - [FMC Write](#)
- * GPIO
 - [GPIO Toggle](#)
- * I2C
 - [I2C TwoBoardsDMA](#)
 - [I2C TwoBoardsInterrupt](#)
 - [I2C TwoBoardsPolling](#)
- * I2S
 - [I2S TwoBoardsInterrupt](#)
- * IAP
 - [Application1](#)
 - [Application2](#)
 - [BootLoader](#)
- * IWDT
 - [IWDT Reset](#)
- * NVIC
 - [NVIC Priority](#)
 - [NVIC WFI](#)
- * PMU

- [PMU Standby](#)
- [PMU Stop](#)
- [PMU Consumption](#)
- * RCM
 - [RCM_ClockConfig](#)
- * RTC
 - [RTC_Alarm](#)
 - [RTC_Second](#)
- * RTOS
 - [CMSIS_FreeRTOS](#)
 - [CMSIS_RTX](#)
- * SPI
 - [SPI_FullDuplex](#)
- * SysTick
 - [SysTick_TimeBase](#)
- * Template
 - [Template](#)
- * TMR
 - [TMR_32BitCount](#)
 - [TMR_6Steps](#)
 - [TMR_CascadeSynchro](#)
 - [TMR_EncoderInterface](#)
 - [TMR_ExtTriggerSynchro](#)
 - [TMR_InputCapture](#)
 - [TMR_OCAActive](#)
 - [TMR_OCInactive](#)
 - [TMR_OCToggle](#)
 - [TMR_ParallelSynchro](#)

- [TMR_PWMInput](#)
- [TMR_PWMOutput](#)
- [TMR_SinglePulse](#)
- [TMR_TimeBase](#)
- [TMR_TMR1DMABurst](#)
- [TMR_TMR1PWMOutput](#)
- [TMR_TMR1Synchro](#)
- [TMR_TMR2PWMOutput](#)
- [TMR_TMR8DMA](#)
- * USART
 - [USART_IrDA](#)
 - [USART_SmartCard](#)
 - [USART_TwoBoardsDMA](#)
 - [USART_TwoBoardsInterrupt](#)
 - [USART_TwoBoardsPolling](#)
 - [USART_LIN](#)
- * USB_OTG
 - [OTGD_CDC](#)
 - [OTGD_Composite_CDC](#)
 - [OTGD_Composite_CDC_WINUSB](#)
 - [OTGD_CUSTOM_HID](#)
 - [OTGD_CUSTOM_HID_Keyboard](#)
 - [OTGD_HID](#)
 - [OTGD_HID_Keyboard](#)
 - [OTGD_MSC](#)
 - [OTGD_MSC_IAP](#)
 - [OTGD_WINUSB](#)
 - [OTGH_CDC](#)

- [OTGH_HID](#)
- [OTGH_DynamicSwitch](#)
- [OTGH_MSC](#)
- [OTGH_MSC_FWUpgrade](#)
- * WWDT
 - [WWDT_OverTime](#)

4.1 ADC_AnalogWindowWatchdog

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 contents

This example can be found in the [~/ADC/ADC_AnalogWindowWatchdog](#) 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 ADC1 Channel 0 input. The converted voltage is displayed on serial assistant through USART1.

4.2.2 Directory contents

This example can be found in the [~/ADC/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 Board is configured to be converted when device startup. The converted voltage is displayed on serial assistant through USART1.

4.3.2 Directory contents

This example can be found in the [~/ADC/ADC_DMA](#) directory.

4.4 ADC_DualRegulSimulMode

4.4.1 Example Description

www.geehy.com

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).

A DMA request is generated each time 2 data items are available.

1st request: ADC->CDATA[31:0] = (ADC2_CH1_REGDATA[15:0] << 16) |
ADC1_CH0_REGDATA[15:0]

4.4.2 Directory contents

This example can be found in the [~/ADC/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 ADC1 Channel 0, ADC1 Channel 1 and ADC1 Channel 2 input. The converted voltage is displayed on serial assistant through USART1.

4.5.2 Directory contents

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

4.6 ADC_TemperatureSensor

4.6.1 Example Description

This example describes how to use the ADC1 to convert the internal temperature sensor's voltage applied to the ADC1_Channel 16.

And Calculate the temperature according to the formula as follow:

$$T = (V - 1.4822)/0.0024 + 28;$$

The converted voltage and temperature are displayed on serial assistant through USART1.

note:

According to actual test data of multiple batches of chips,

V28 is adopted instead of V25 for this example. And 1.4822V is the voltage for 28 degrees Celsius

4.6.2 Directory contents

This example can be found in the [~/ADC/ADC_TripleInterleavedMode](#) directory.

4.7 BAKPR_Tamper

4.7.1 Example Description

This example describes how to write the backup registers. After initialization, the system enters into an infinite loop. If data in the backup registers is equal to the data written to the registers before, LED2 keeps blinking; otherwise, LED3 keeps blinking. The 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.

4.7.2 Directory contents

This example can be found in the [~/BAKPR/BAKPR_Tamper](#) directory.

4.8 CAN_LoopBack

4.8.1 Example Description

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

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

4.8.2 Directory contents

This example can be found in the [~/CAN/CAN_LoopBack](#) directory.

4.9 CAN_Normal

4.9.1 Example Description

This example describes how to config a communication the CAN in normal mode. CAN1 transmit a message to CAN2. Then compare the received message with transmitted message. The result of Polling transmit and Interrupt transmit will displayed on serial assistant through USART1.

4.9.2 Directory contents

This example can be found in the [~/CAN/CAN_Normal](#) directory.

4.10 CRC_Calculation

4.10.1 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.

4.10.2 Directory contents

This example can be found in the [~/CRC/CRC_Calculation](#) directory.

4.11 DMA_MemoryToMemory

4.11.1 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.

4.11.2 Directory contents

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

4.12 EINT_Config

4.12.1 Example Description

This example shows how to configure external interrupt lines. In this example, 2 EINT lines (KEY1,KEY2) when using the APM32F402/403 TINY 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.

4.12.2 Directory contents

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

4.13 FMC_Protection

4.13.1 Example Description

This example shows how to set write protection for the flash address of APM32F402/403. 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 programmed.

4.13.2 Directory contents

This example can be found in the [~/FMC/FMC_Protection](#) directory.

4.14 FMC_Protection_RAM

4.14.1 Example Description

This example shows how to set write protection for the flash address of APM32F402/403. 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 programmed.

4.14.2 Directory contents

This example can be found in the [~/FMC/FMC_Protection_RAM](#) directory.

4.15 FMC_Read_Write

4.15.1 Example Description

This example describes how to using the onchip Flash memory of the APM32F402/403 device to store user data. After running this example, if the reading and writing successful, LED2 is on, otherwise it is off. The print data will have displayed on serial assistant through USART1.

4.15.2 Directory contents

This example can be found in the [~/FMC/FMC_Read_Write](#) directory.

4.16 FMC_Write

4.16.1 Example Description

This example provides a description of how to program the flash address. After reset, the flash will be unlocked and erases the specifies address, then writes data to the specified address, finally locks the flash and compares the data in the flash with the written data. if they are consistent, LED2 will light, otherwise LED3 will light.

4.16.2 Directory contents

This example can be found in the [~/FMC/FMC_Write](#) directory.

4.17 GPIO_Toggle

4.17.1 Example Description

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

4.17.2 Directory contents

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

4.18 I2C_TwoBoardsDMA

4.18.1 Example Description

This example shows how to control I2C devices and communicate between two different boards. To use this example, you need to load the software into two APM32F402/403_TINY 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 data to Board slave.
If the data received from the device is consistent with that sent by the main device,
the LED2 of the device will be lit up.

4.18.2 Directory contents

This example can be found in the [~/I2C/I2C_TwoBoardsDMA](#) directory.

4.19 I2C_TwoBoardsInterrupt

4.19.1 Example Description

This example shows how to control I2C devices and communicate between two different boards. To use this example, you need to load the software into two APM32F103_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.

4.19.2 Directory contents

This example can be found in the [~/I2C/I2C_TwoBoardsInterrupt](#) directory.

4.20 I2C_TwoBoardsPolling

4.20.1 Example Description

This example shows how to control I2C devices and communicate between two different boards. To use this example, you need to load the software into two APM32F402/403_TINY 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 data to Board slave.
If the data received from the device is consistent with that sent by the main device, the LED2 of the device will be lit up.

4.20.2 Directory contents

This example can be found in the [~/I2C/I2C_TwoBoardsPolling](#) directory.

4.21 I2S_TwoBoardsInterrupt

4.21.1 Example Description

This example demonstrates I2S (Inter-IC Sound) communication between a master device and a slave device using SPI peripherals. The master device transmits audio data, while the slave device receives the data and compares it with the expected values. After pressing KEY1, the main device begins to send data. The subordinate device receives the data and verifies its correctness. If the data is consistent, LED2 of the subordinate device is illuminated.

4.21.2 Directory contents

This example can be found in the [~/I2S/I2S_TwoBoardsInterrupt](#) directory.

4.22 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 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 BootLoader

4.24.1 Example Description

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

4.24.2 Directory contents

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

4.26 NVIC_Priority

4.26.1 Example Description

This example describes how to use NVIC priority. At startup, press KEY1(PA0) 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 contents

This example can be found in the [~/NVIC/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 KEY1(PA0) to occur Wait for Interrupt(WFI) event, and device will enter sleep mode. The device will wake up if press KEY1 again. The status of device is displayed on serial assistant through USART1.

4.27.2 Directory contents

This example can be found in the [~/NVIC/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 contents

This example can be found in the [~/PMU/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 contents

This example can be found in the [~/PMU/PMU_Stop](#) directory.

4.30 PMU_Consumption

4.30.1 Example Description

This example shows how to configure system to measure different low power modes consumption. Includes the following modes:

- Sleep Mode
- Standby mode with WKUP pin

To run this example, users has to follow the following steps:

- Select different modes by setting 1 of the following code:

```
/*  
  
* #define SYS_SLEEP_MODE  
  
* #define SYS_STANDBY_MODE  
  
*/
```

- Use an external ammeter to measure the IDD current.

4.30.2 Directory contents

This example can be found in the [~/PMU/PMU_Consumption](#) directory.

4.31 RCM_ClockConfig

4.31.1 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)

4.31.2 Directory contents

This example can be found in the [~/RCM/RCM_ClockConfig](#) directory.

4.32 RTC_Alarm

4.32.1 Example Description

This example provides an example of how to use RTC alarm function. Pressing KEY2 can set an alarm for 5 seconds.

4.32.2 Directory contents

This example can be found in the [~/RTC/RTC_Alarm](#) directory.

4.33 RTC_Second

4.33.1 Example Description

The example shows how to use RTC to generate second interrupt. LED2 will blink every second which is generated by RCT interrupt. USART1 will report the RTC counter every second.

4.33.2 Directory contents

This example can be found in the [~/RTC/RTC_Second](#) directory.

4.34 CMSIS_FreeRTOS

4.34.1 Example Description

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

Usart test task : USART1 send a string

Led toggle task : The IO of LED2 is configed to toggle constantly

4.34.2 Directory contents

This example can be found in the [~/RTOS/CMSIS_FreeRTOS](#) directory.

4.35 CMSIS_RTX

4.35.1 Example Description

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

Usart test task : Printf log information by USART1.

Led toggle task : The IO of LED2 is configed to toggle constantly.

4.35.2 Directory contents

This example can be found in the [~/RTOS/CMSIS_RTX](#) 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 SPI1 and SPI2. If communication success, LED2 will turn on, otherwise LED3 will turn on.

4.36.2 Directory contents

This example can be found in the [~/SPI/SPI_FullDuplex](#) directory.

4.37 SysTick_TimeBase

4.37.1 Example Description

This example describes how to use SysTick_Delay for toggling IO. The IO of LED2 and LED3 is configured to toggle constantly every 1000 milliseconds. The phenomenon of LED2 and LED3 constantly flickered alternately.

4.37.2 Directory contents

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

4.38 Template

4.38.1 Example Description

This demo is based on the APM32F402_403 SDK . It provides a template project for standard peripheral driver library.

4.38.2 Directory contents

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

4.39 TMR_32BitCount

4.39.1 Example Description

This example describes how to configure the TMR2 and TMR3 realize the 32-bit timer.

TMR2 as High 16 bit count value, TMR3 as Low 16 bit count value.

User can view the counter value through serial terminal.

4.39.2 Directory contents

This example can be found in the [~/TMR/TMR_32BitCount](#) directory.

4.40 TMR_6Steps

4.40.1 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.

4.40.2 Directory contents

This example can be found in the [~/TMR/TMR_6Steps](#) 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/TMR_CascadeSynchro](#) directory.

4.42 TMR_EncoderInterface

4.42.1 Example Description

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

4.42.2 Directory contents

This example can be found in the [~/TMR/TMR_EncoderInterface](#) directory.

4.43 TMR_ExtTriggerSynchro

4.43.1 Example Description

This example shows how to synchronize TMR2 and TMR peripherals in cascade mode with an external trigger.

4.43.2 Directory contents

This example can be found in the [~/TMR/TMR_ExtTriggerSynchro](#) directory.

4.44 TMR_InputCapture

4.44.1 Example Description

This example describes how to use TMR5 Channel_2 (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/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/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/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/TMR_OCToggle](#) directory.

4.48 TMR_ParallelSynchro

4.48.1 Example Description

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

4.48.2 Directory contents

This example can be found in the [~/TMR/TMR_ParallelSynchro](#) directory.

4.49 TMR_PWMInput

4.49.1 Example Description

This example describes how to use TMR5 Channel_2 (PA1) measure frequency and duty cycle of external signal. User can view the "DutyCycle" "Frequency" value through serial terminal.

4.49.2 Directory contents

This example can be found in the [~/TMR/TMR_PWMInput](#) directory.

4.50 TMR_PWMOutput

4.50.1 Example Description

This example shows how to configure the TMR5 peripheral to generate PWM signals with different duty cycles. The TMR5 waveform can be displayed using an oscilloscope.

4.50.2 Directory contents

This example can be found in the [~/TMR/TMR_PWMOutput](#) directory.

4.51 TMR_SinglePulse

4.51.1 Example Description

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

4.51.2 Directory contents

This example can be found in the [~/TMR/TMR_SinglePulse](#) directory.

4.52 TMR_TimeBase

4.52.1 Example Description

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

4.52.2 Directory contents

This example can be found in the [~/TMR/TMR_TimeBase](#) 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 contents

This example can be found in the [~/TMR/TMR_TMR1DMABurst](#) directory.

4.54 TMR_TMR1PWMOutput

4.54.1 Example Description

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

4.54.2 Directory contents

This example can be found in the [~/TMR/TMR_TMR1PWMOutput](#) directory.

4.55 TMR_TMR1Synchro

4.55.1 Example Description

This example shows how to synchronize TMR1 and TMR peripherals in parallel mode.

4.55.2 Directory contents

This example can be found in the [~/TMR/TMR_TMR1Synchro](#) directory.

4.56 TMR_TMR2PWMOutput

4.56.1 Example Description

This example shows how to configure the TMR2 peripheral to generate PWM signals with different duty cycles. The TMR2 waveform can be displayed using an oscilloscope.

4.56.2 Directory contents

This example can be found in the [~/TMR/TMR_TMR2PWMOutput](#) directory.

4.57 TMR_TMR8DMA

4.57.1 Example Description

The program to show how to use DMA to transfer Data from memory to TMR8 Capture Compare Register1 to change the Duty Cycle.

4.57.2 Directory contents

This example can be found in the [~/TMR/TMR_TMR8DMA](#) directory.

4.58 USART_IrDA

4.58.1 Example Description

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

4.58.2 Directory contents

This example can be found in the [~/USART/USART_IrDA](#) directory.

4.59 USART_LIN

4.59.1 Example Description

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

4.59.2 Directory contents

This example can be found in the [~/USART/USART_LIN](#) directory.

4.60 USART_SmartCard

4.60.1 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.

4.60.2 Directory contents

This example can be found in the [~/USART/USART_SmartCard](#) directory.

4.61 USART_TwoBoardsDMA

4.61.1 Example Description

The program aims to show how to use KEY1 button to trigger USART communication using DMA, in this case, you need to load program on two APM32F402/403_TINY boards. Then connect these two boards through USART lines and GND. If USART RX Board receives the correct data from USART TX Board, USART RX Board LED2 will turn on.

4.61.2 Directory contents

This example can be found in the [~/USART/USART_TwoBoardsDMA](#) directory.

4.62 USART_TwoBoardsInterrupt

4.62.1 Example Description

The program aims to show how to use KEY1 button to trigger USART communication using interrupts, in this case, you need to load program on two APM32F402/403_TINY boards. Then connect these two boards through USART lines and GND. If USART RX Board receives the correct data from USART TX Board, USART RX Board LED2 will turn on.

4.62.2 Directory contents

This example can be found in the [~/USART/USART_TwoBoardsInterrupt](#) directory.

4.63 USART_TwoBoardsPolling

4.63.1 Example Description

This project demonstrates how to implement bidirectional polling communication between two development boards using the APM32F402/403 microcontroller via the USART interface.

The two boards are referred to as Board A and Board B, where Board A sends data to Board B, and Board B receives and echoes back the data.

4.63.2 Directory contents

This example can be found in the [~/USART/USART_TwoBoardsPolling](#) directory.

4.64 OTGD_CDC

4.64.1 Example Description

This example describes how to use OTG to simulate a CDC device. When CDC device receive data will send back the same data to USB host.

4.64.2 Directory contents

This example can be found in the [~/USB_OTG/Device_Examples/OTGD_CDC](#) directory.

4.65 OTGD_Composite_CDC

4.65.1 Example Description

This example describes how to use OTG to implement a composite device with two CDC functional interfaces.

4.65.2 Directory contents

This example can be found in the [~/USB_OTG/Device_Examples/OTGD_Composite_CDC](#) directory.

4.66 OTGD_Composite_CDC_WINUSB

4.66.1 Example Description

This example describes how to use OTG to implement a composite device with CDC and WINUSB functional interfaces.

4.66.2 Directory contents

This example can be found in the [~/USB_OTG/Device_Examples/OTGD_Composite_CDC_WINUSB](#) directory.

4.67 OTGD_Custom_HID

4.67.1 Example Description

This example describes how to use OTG to simulate a custom hid. This is a template example. User can customize HID reports to implement the desired functionality.

4.67.2 Directory contents

This example can be found in the [~/USB_OTG/Device_Examples/OTGD_Custom_HID](#) directory.

4.68 OTGD_Custom_HID_Keyboard

4.68.1 Example Description

This example describes how to use OTG to simulate a custom HID keyboard. Press KEY1 will send the report descriptor of a - z or Enter to the USB host. The state of Capslock and numlock determine whether LED2 and LED3 are on or off.

4.68.2 Directory contents

This example can be found in the [~/USB_OTG/Device_Examples/OTGD_Custom_HID_Keyboard](#) directory.

4.69 OTGD_HID

4.69.1 Example Description

This example describes how to use OTG to simulate a HID mouse.

Press KEY1 will move the cursor to the left.

Press KEY2 will move the cursor to the right.

4.69.2 Directory contents

This example can be found in the [~/USB_OTG/Device_Examples/OTGD_HID](#) directory.

4.70 OTGD_HID_Keyboard

4.70.1 Example Description

This example describes how to use OTG to simulate a HID keyboard.

Press KEY1 will send the report descriptor of a - z or Enter to the USB host.

4.70.2 Directory contents

This example can be found in the [~/USB_OTG/Device_Examples/OTGD_HID_Keyboard](#) directory.

4.71 OTGD_MSC

4.71.1 Example Description

This example describes how to use sram array to simulate a fake U disk.

4.71.2 Directory contents

This example can be found in the [~/USB_OTG/Device_Examples/OTGD_MSC](#) directory.

4.72 OTGD_MSC_IAP

4.72.1 Example Description

This example describes how to use a simulated USB drive device for IAP upgrade.

After plugging into the computer, the USB drive will display a 78KB BACKUP.bin file

and ready.txt. The former indicates the data readback of the APP area, and the latter indicates that the USB drive is properly enumerated for the IAP upgrade.

4.72.2 Directory contents

This example can be found in the [~/USB_OTG/Device_Examples/OTGD_MSC_IAP](#) directory.

4.73 OTGD_WINUSB

4.73.1 Example Description

This example describes how to use OTG to simulate a WINUSB device.

Program will send hello + num string to USB host. And when WINUSB device receive data will send back the same data to USB host.

4.73.2 Directory contents

This example can be found in the [~/USB_OTG/Device_Examples/OTGD_WINUSB](#) directory.

4.74 OTGH_CDC

4.74.1 Example Description

This example describes how to use the usb host to enum a CDC device.

And use UART to print CDC device operation information.

4.74.2 Directory contents

This example can be found in the [~/USB_OTG/Host_Examples/OTGH_CDC](#) directory.

4.75 OTGH_DynamicSwitch

4.75.1 Example Description

This example describes how to use the OTG HS2 host to enum a CDC device.

And use UART to print CDC device operation information.

4.75.2 Directory contents

This example can be found in the [~/USB_OTG/Host_Examples/OTGH_DynamicSwitch](#) directory.

4.76 OTGH_HID

4.76.1 Example Description

This example describes how to use the usb host to enum a HID device(mouse or keyboard).

And use UART to print mouse or keyboard operation information.

4.76.2 Directory contents

This example can be found in the [~/USB_OTG/Host_Examples\OTGH_HID](#) directory.

4.77 OTGH_MSC

4.77.1 Example Description

This example describes how to use the usb host to enum a U disk.

And use FATFS to write and read file to U disk.Press KEY1 to write file to U disk and press KEY2 to read file from U disk.

4.77.2 Directory contents

This example can be found in the [~/USB_OTG/Host_Examples/OTGH_MSC](#) directory.

4.78 OTGH_MSC_FWUpgrade

4.78.1 Example Description

This example describes how to use the usb host to enum a U disk. And use it to upgrade application firmware.

4.78.2 Directory contents

This example can be found in the [~/USB_OTG/Host_Examples/OTGH_MSC_Upgrade](#) directory.

4.79 WWDT_OverTime

4.79.1 Example Description

www.geehy.com

This example aims to show how to use WWDT. When KEY1 is not pressed, System is not reset due to feeding dog timely, and 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 WWDT.

4.79.2 Directory contents

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

5 About libraries

The libraries folder includes a series library. It can provide supports for APM32F402/403xx MCU such as device support and standard peripheral etc. The libraries can be found in the [~/Libraries](#) directory.

APM32F402/403xx MCU include following library:

- Libraries folder
 - * APM32F402_403_StdPeriphDriver
 - * CMSIS
 - * Device

6 About middlewares

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

The middlewares used by APM32F402/403xx include following:

- Middlewares folder
 - * APM32_USB_Library
 - * FatFs
 - * CMSIS-FreeRTOS

7 About Package

The Package folder includes Geehy APM32F4xx DFP Package. The Package can be found in the [~/Package](#) directory.

The middlewares used by APM32F402/403xx include following:

- Package folder
 - * SVD
 - * Geehy.APM32F4xx_DFP.1.0.7.pack

8 Revision History

Table 1 File Revision History

Date	Rev	Description
2024.11.25	1.0	First Release version of APM32F402_403 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.

© 2024-2025 Geehy Semiconductor Co., Ltd. - All Rights Reserved