

Readme

APM32F4xx SDK

Rev: V1.0

1 Introduction

The Geehy Semiconductor APM32F4xx 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

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_DualInterleavedMode.....	13
4.5	ADC_DualRegulSimulMode	14
4.6	ADC_MultiChannelScan.....	14
4.7	ADC_TripleInterleavedMode	14
4.8	ADC_VBAT.....	15
4.9	CAN_LoopBack	15
4.10	CAN_Normal.....	15
4.11	CRC_Calculation.....	16
4.12	CRYP_AES.....	16
4.13	CRYP_DES-TDES	16
4.14	DAC_ADC	17
4.15	DCI_OV2640	17
4.16	DMA_ADC.....	17
4.17	DMA_FMCToRAM.....	18
4.18	DSP_bayes	18
4.19	DSP_class_marks	18
4.20	DSP_convolution	18
4.21	DSP_dotproduct	19
4.22	DSP_fft_bin.....	19

4.23	DSP_fir	19
4.24	DSP_graphic_equalizer	20
4.25	DSP_linear_interp.....	20
4.26	DSP_matrix	20
4.27	DSP_signal_converge.....	20
4.28	DSP_sin_cos.....	21
4.29	DSP_svm	21
4.30	DSP_Template.....	21
4.31	DSP_variance	21
4.32	EINT_Config	22
4.33	ETH_Ping.....	22
4.34	FMC_Write	22
4.35	GPIO_Toggle	23
4.36	HASH_SHA1	23
4.37	I2C_TwoBoards_Master	23
4.38	I2C_TwoBoards_Slave.....	24
4.39	I2S_Interrupt	24
4.40	Application1.....	24
4.41	Application2.....	25
4.42	BootLoader	25
4.43	IWDT_Reset	25
4.44	NVIC_Priority	26
4.45	NVIC_WFI	26
4.46	PMU_STANDBY	26
4.47	PMU_STOP	27
4.48	RCM_ClockConfig	27
4.49	RNG_MultiRNG.....	27
4.50	RTC_Alarm	27

4.51	FreeRTOS	28
4.52	RT-thread	28
4.53	SDIO_SDCard	28
4.54	SPI_FullDuplex	29
4.55	SysTick_TimeBase.....	29
4.56	Template	29
4.57	TMR_32BitCount	30
4.58	TMR_6Steps.....	30
4.59	TMR_CascadeSynchro	30
4.60	TMR_EncoderInterface.....	30
4.61	TMR_ExtTriggerSynchro	31
4.62	TMR_InputCapture	31
4.63	TMR_OCActive.....	31
4.64	TMR_OCInactive	31
4.65	TMR_OCToggle	32
4.66	TMR_ParallelSynchro.....	32
4.67	TMR_PWMInput.....	32
4.68	TMR_PWMOutput.....	33
4.69	TMR_SinglePulse	33
4.70	TMR_TimeBase.....	33
4.71	TMR_TMR11PWMOutput	33
4.72	TMR_TMR1DMABurst	34
4.73	TMR_TMR1PWMOutput	34
4.74	TMR_TMR1Synchro	34
4.75	TMR_TMR2PWMOutput	34
4.76	TMR_TMR8DMA.....	35
4.77	TMR_TMR9OCToggle.....	35
4.78	USART_Interrupt.....	35

4.79	USART_Polling	36
4.80	WWDT_OverTime.....	36
5	About libraries	37
6	About middlewares	38
7	About Package	39
8	Revision History.....	40

2 About boards

The boards folder includes a board support package for APM32F4xx 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 APM32F4xx 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 APM32F407 MINI include following board support package:

- Board_APM32F407_MINI src folder
 - * Board_APM32F407_MINI
 - * bsp_delay
 - * bsp_i2c
 - * bsp_key
 - * bsp_usart

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 APM32F4xx 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: **APM32F4xx StdPeriphDriver v1.0.2**, include the following examples:

- Examples

- * ADC
 - [ADC_AnalogWindowWatchdog](#)
 - [ADC_ContinuousConversion](#)
 - [ADC_DMA](#)
 - [ADC_DualInterleavedMode](#)
 - [ADC_DualRegulSimulMode](#)
 - [ADC_MultiChannelScan](#)
 - [ADC_TripleInterleavedMode](#)
 - [ADC_VBAT](#)
- * CAN
 - [CAN_LoopBack](#)
 - [CAN_Normal](#)
- * CRC
 - [CRC_Calculation](#)

- * CRYPT
 - [CRYPT_AES](#)
 - [CRYPT_DES-TDES](#)
- * DAC
 - [DAC_ADC](#)
- * DCI
 - [DCI_OV2640](#)
- * DMA
 - [DMA_ADC](#)
 - [DMA_FMCToRAM](#)
- * DSP
 - [DSP_bayes](#)
 - [DSP_class_marks](#)
 - [DSP_convolution](#)
 - [DSP_dotproduct](#)
 - [DSP_fft_bin](#)
 - [DSP_fir](#)
 - [DSP_graphic_equalizer](#)
 - [DSP_linear_interp](#)
 - [DSP_matrix](#)
 - [DSP_signal_converge](#)
 - [DSP_sin_cos](#)
 - [DSP_svm](#)
 - [DSP_Template](#)
 - [DSP_variance](#)
- * EINT
 - [EINT_Config](#)
- * ETH

- [ETH Ping](#)
- * FMC
 - [FMC Write](#)
- * GPIO
 - [GPIO Toggle](#)
- * HASH
 - [HASH_SHA1](#)
- * I2C
 - [I2C_TwoBoards_Master](#)
 - [I2C_TwoBoards_Slave](#)
- * I2S
 - [I2S Interrupt](#)
- * IAP
 - [Application1](#)
 - [Application2](#)
 - [BootLoader](#)
- * IWDT
 - [IWDT_Reset](#)
- * NVIC
 - [NVIC_Priority](#)
 - [NVIC_WFI](#)
- * PMU
 - [PMU_STANDBY](#)
 - [PMU_STOP](#)
- * RCM
 - [RCM_ClockConfig](#)
- * RNG
 - [RNG_MultiRNG](#)

- * RTC
 - [RTC_Alarm](#)
- * RTOS
 - [FreeRTOS](#)
 - [RT-thread](#)
- * SDIO
 - [SDIO_SDCard](#)
- * 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_TMR11PWMOutput](#)
- [TMR_TMR1DMABurst](#)
- [TMR_TMR1PWMOutput](#)
- [TMR_TMR1Synchro](#)
- [TMR_TMR2PWMOutput](#)
- [TMR_TMR8DMA](#)
- [TMR_TMR9OCToggle](#)
- * USART
 - [USART Interrupt](#)
 - [USART Polling](#)
- * 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_Channel0 continuously. When input Voltage of ADC1_Channel0(PA0) voltage is lower than 0.62v or higher than 2.27V, LED2 is on, otherwise, LED2 is off. That is If the voltage on ADC1_Channel0(PA0) is not in the thresholds which is setted before, analog watchdog interrupt will generate and light LED2. The converted voltage is displayed on serial assistant through USART1.

4.1.2 Directory contents

This example can be found in the [~ /Examples/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 APM32F407 MINI ADC1_Channel0 input. The voltage converted is displayed on serial assistant through USART1.

4.2.2 Directory contents

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

4.3 ADC_DMA

4.3.1 Example Description

This example provides example of how to use a DMA channel to transfer continuously a data from a peripheral (ADC1) to DMA transfer. The ADC channel0 for APMF407 MINI Board is configured to be converted when device startup. The value of ADC is shown in USART1.

4.3.2 Directory contents

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

4.4 ADC_DualInterleavedMode

4.4.1 Example Description

This example describes how to use the ADC to convert Channel0 in Dual interleaved mode using DMA in mode 3.

4.4.2 Directory contents

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

4.5 ADC_DualRegulSimulMode

4.5.1 Example Description

This example describes how to use the ADC to convert Channel0, Channel1 and Channel2 simultaneously in dual mode using DMA in mode 1.

4.5.2 Directory contents

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

4.6 ADC_MultiChannelScan

4.6.1 Example Description

This example describes how to use the ADC1 to scan continuously the voltage applied to the APM32F407 MINI ADC1_Channel0 and ADC1_Channel1 and ADC1_Channel2 input. The converted voltage is displayed on serial assistant through USART1.

4.6.2 Directory contents

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

4.7 ADC_TripleInterleavedMode

4.7.1 Example Description

This example describes how to use the ADC to convert Channel0 in Triple interleaved mode using DMA in mode 2.

4.7.2 Directory contents

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

4.8 ADC_VBAT

4.8.1 Example Description

This example describes how to use the ADC1 to convert VBAT voltage applied to the APM32F407 MINI ADC1_Channel18. The converted voltage is displayed on serial assistant through USART1.

4.8.2 Directory contents

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

4.9 CAN_LoopBack

4.9.1 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. The data 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 [~/Examples/CAN/LoopBack](#) directory.

4.10 CAN_Normal

4.10.1 Example Description

This example describes how to configure 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.

- Polling transmit success: The LED2 turns on, printf "CAN polling test passed!". Otherwise LED2 toggles, printf "CAN polling test failed!".

- Interrupt transmit success: The LED3 turns on, printf "CAN interrupt test passed!". Otherwise LED3 toggles, printf "CAN interrupt test failed".

4.10.2 Directory contents

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

4.11 CRC_Calculation

4.11.1 Example Description

This example shows how to use CRC (Cyclic Redundancy Check) calculation unit to get a CRC code of a given buffer of data word(32-bit), based on a fixed generator polynomial(0x4C11DB7). an interactive human interface is developed to allow user to display CRC 32bit numbers using the eval board USART1 with PC HyperTerminal.

4.11.2 Directory contents

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

4.12 CRYPT_AES

4.12.1 Example Description

This example describes the CRYPT processor performs data encryption and decryption using AES Igorithms in Electronic codebook (ECB) or Cipher block chaining (CBC) or counter(CTR) mode. The data that needs to be decrypted and encrypted will be displayed on the serial assistant through USART1.

4.12.2 Directory contents

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

4.13 CRYPT_DES-TDES

4.13.1 Example Description

This example describes the CRYPT processor performs data encryption and decryption using DES and TDES Igorithms in Electronic codebook (ECB) or Cipher block chaining (CBC) mode. The data that needs to be decrypted and encrypted will be displayed on the serial assistant through USART1. if define PLAIN_TEXT_SHORT then the length of plaintext is 64, else the length of plaintext is 128.

4.13.2 Directory contents

This example can be found in the [~/Examples/CRYP/CRYP_DES-TDES](#) directory.

4.14 DAC_ADC

4.14.1 Example Description

This example provides 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 and displayed on serial assistant through USART1.

4.14.2 Directory contents

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

4.15 DCI_OV2640

4.15.1 Example Description

The program aims to show how to use interrupt or DMA to get camera image data by using DCI, in this case, DCI continuously get the data of the OV2640 camera through DMA and sends it to the host by USART2(jpeg mode). In another regb 565 mode, DCI continuously get the data of the OV2640 camera through DMA and transfer it to LCD data register to display. And OV2640 can be set to jpeg and rgb565 data output format.

4.15.2 Directory contents

This example can be found in the [~/Examples/DCI/DCI_OV2640](#) directory.

4.16 DMA_ADC

4.16.1 Example Description

This example provides a example of how to use a DMA channel to transfer continuously a data from a peripheral (ADC1) to DMA transfer. The ADC channel0 for APMF407 MINI Board is configured to be converted when device startup. The value of ADC is shown in USART1.

4.16.2 Directory contents

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

4.17 DMA_FMCToRAM

4.17.1 Example Description

This example shows how to use a DMA channel to transfer a word data buffer from FLASH memory to embedded SRAM memory.

4.17.2 Directory contents

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

4.18 DSP_bayes

4.18.1 Example Description

Example code demonstrating how to use Bayes functions. The demo can be displayed using serial assistant. After power on, can press KEY1 to start demo test.

4.18.2 Directory contents

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

4.19 DSP_class_marks

4.19.1 Example Description

Example code to calculate Minimum, Maximum Mean, std and variance of marks obtained in a class. The demo can be displayed using serial assistant. After power on, can press KEY1 to start demo test.

4.19.2 Directory contents

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

4.20 DSP_convolution

4.20.1 Example Description

Example code demonstrating Convolution of two input signals using fft. The demo can be displayed using serial assistant. After power on, can press KEY1 to start demo test.

4.20.2 Directory contents

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

4.21 DSP_dotproduct

4.21.1 Example Description

Example code computing dot product of two vectors. The demo can be displayed using serial assistant. After power on, can press KEY1 to start demo test.

4.21.2 Directory contents

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

4.22 DSP_fft_bin

4.22.1 Example Description

Example code demonstrating calculation of Max energy bin of frequency domain of input signal. The demo can be displayed using serial assistant. After power on, can press KEY1 to start demo test.

4.22.2 Directory contents

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

4.23 DSP_fir

4.23.1 Example Description

Example code demonstrating how an FIR filter can be used as a low pass filter. The demo can be displayed using serial assistant. After power on, can press KEY1 to start demo test.

4.23.2 Directory contents

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

4.24 DSP_graphic_equalizer

4.24.1 Example Description

Example showing an audio graphic equalizer constructed out of Biquad filters. The demo can be displayed using serial assistant. After power on, can press KEY1 to start demo test.

4.24.2 Directory contents

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

4.25 DSP_linear_interp

4.25.1 Example Description

Example code demonstrating usage of sin function and uses linear interpolation to get higher precision. The demo can be displayed using serial assistant. After power on, can press KEY1 to start demo test.

4.25.2 Directory contents

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

4.26 DSP_matrix

4.26.1 Example Description

Example code demonstrating least square fit to data using matrix functions. The demo can be displayed using serial assistant. After power on, can press KEY1 to start demo test.

4.26.2 Directory contents

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

4.27 DSP_signal_converge

4.27.1 Example Description

Example code demonstrating convergence of an adaptive filter. The demo can be displayed using serial assistant. After power on, can press KEY1 to start demo test.

4.27.2 Directory contents

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

4.28 DSP_sin_cos

4.28.1 Example Description

Example code demonstrating sin and cos calculation of input signal. The demo can be displayed using serial assistant. After power on, can press KEY1 to start demo test.

4.28.2 Directory contents

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

4.29 DSP_svm

4.29.1 Example Description

Example code demonstrating how to use SVM functions. The demo can be displayed using serial assistant. After power on, can press KEY1 to start demo test.

4.29.2 Directory contents

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

4.30 DSP_Template

4.30.1 Example Description

This demo is based on the APM32F407 MIN board. it provides a DSP template project.

4.30.2 Directory contents

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

4.31 DSP_variance

4.31.1 Example Description

Example code demonstrating variance calculation of input sequence. The demo can be displayed using serial assistant. After power on, can press KEY1 to start demo test.

4.31.2 Directory contents

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

4.32 EINT_Config

4.32.1 Example Description

This example shows how to configure external interrupt lines. In this example, 2 EINT lines (KEY1、KEY2) when using the APM32F407 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.

4.32.2 Directory contents

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

4.33 ETH_Ping

4.33.1 Example Description

This example describes how to use ethernet PHY by using APM32F4xx_ETH_Driver library. After configured ethernet mainboard will be use USART1 to printf static IP address. if computer ping static IP address (192.168.73.22) , computer will be visit mainboard normality.

The phenomenon of data interaction process can be displayed using cmd.exe.

4.33.2 Directory contents

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

4.34 FMC_Write

4.34.1 Example Description

This example provides a description of how to program the flash address of APM32F407. After Reset, the Flash will be unlock. Then erase the specifies address and write a data in the address. In the end, lock the flash. The data of the address after erasing and programing will displayed on serial assistant through USART1.

4.34.2 Directory contents

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

4.35 GPIO_Toggle

4.35.1 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.

4.35.2 Directory contents

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

4.36 HASH_SHA1

4.36.1 Example Description

This example describes how to use DOUT for HASH Calculation. The HASH of SHA1 calculate the data that need to be digested. The phenomenon of HASH Calculation are the data outputed by DOUT. display HASH-SHA1 20 numbers using the eval board USART1 with PC HyperTerminal.

4.36.2 Directory contents

This example can be found in the [~ /Examples/HASH/HASH_SHA1](#) directory.

4.37 I2C_TwoBoards_Master

4.37.1 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 APM32F407_MINI boards (let's call them Board master and Board Slave) then connect these two boards through I2C lines and GND.

4.37.2 Directory contents

This example can be found in the [~ /Examples/I2C/I2C_TwoBoards_Master](#) directory.

4.38 I2C_TwoBoards_Slave

4.38.1 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 APM32F407_MINI boards (let's call them Board master and Board Slave) then connect these two boards through I2C lines and GND.

4.38.2 Directory contents

This example can be found in the [~ /Examples/I2C/I2C_TwoBoards_Slave](#) directory.

4.39 I2S_Interrupt

4.39.1 Example Description

This example describes how to use I2S peripheral. by making a communication between the I2S2 and the I2S3. If communication success, LED2 will turn on. "Transfer OK!" will be shown on serial assistant trough usart1. if fail, LED2 will turn off, "Transfer Fail!" will be shown on serial assistant trough usart1. LED3 blinking shows system is running.

4.39.2 Directory contents

This example can be found in the [~ /Examples/I2S/I2S_Interrupt](#) directory.

4.40 Application1

4.40.1 Example Description

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

4.40.2 Directory contents

This example can be found in the [~ /Examples/IAP/Application1](#) directory.

4.41 Application2

4.41.1 Example Description

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

4.41.2 Directory contents

This example can be found in the [~ /Examples/IAP/Application2](#) directory.

4.42 BootLoader

4.42.1 Example Description

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

4.42.2 Directory contents

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

4.43 IWDT_Reset

4.43.1 Example Description

The example shows how to configure IWDT and feed dog to prevent a system reset. After IWDT initialization, if press KEY1 to start feed watchdog, System enters into a infinite loop, feed dog before the counter. reach a given timeout value to prevent system reset and keep LED2 blinking regularly. Otherwise System will reset about 4 seconds! Pressing KEY1 again to stop feed dog will trigger system reset in 4 4 seconds, when the counter reach a given timeout value. LED3 will be lighted when a system reset is triggered by IWDT. if system reset is triggered by IWDT, "IWDT Reset" will be print with USART1.

4.43.2 Directory contents

www.geehy.com

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

4.44 NVIC_Priority

4.44.1 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.

4.44.2 Directory contents

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

4.45 NVIC_WFI

4.45.1 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.

4.45.2 Directory contents

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

4.46 PMU_STANDBY

4.46.1 Example Description

This example shows how to enter the system by external interrupt to STANDBY mode and wake-up from this mode either with the RESET or give PA0 a rising edge to recover.

4.46.2 Directory contents

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

4.47 PMU_STOP

4.47.1 Example Description

This example shows how to enter the system by external interrupt to STOP mode and wake-up from this mode either with the RESET or press KEY2 to recover.

4.47.2 Directory contents

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

4.48 RCM_ClockConfig

4.48.1 Example Description

This example shows how to: Configure the PLL1 (clocked by HSE) as System clock source
Output the System clock(33.6MHz) on MCO pin(PA8).

4.48.2 Directory contents

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

4.49 RNG_MultiRNG

4.49.1 Example Description

This example shows how to use the RNG peripheral to generate Random 32bit numbers. For this example, random 32bit numbers will display on serial assistant through USART1.

4.49.2 Directory contents

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

4.50 RTC_Alarm

4.50.1 Example Description

This example shows how to configure RTC and ALARM.

phenomenon:

- After initialization, Alarm begin to count down with LED2 is on. Five second later, Alarm is waking up and LED2 is off.
- You can monitor the system state on serial assistant through USART1.

4.50.2 Directory contents

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

4.51 FreeRTOS

4.51.1 Example Description

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

4.51.2 Directory contents

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

4.52 RT-thread

4.52.1 Example Description

This example describes how to use RT-Thread for APM32F4xx. The IO of LED2 and LED3 is configed to toggle constantly. The phenomenon of LED2 and LED3 constantly flickered alternately..

4.52.2 Directory contents

This example can be found in the [~/Examples/RTOS/RT-thread](#) directory.

4.53 SDIO_SDCard

4.53.1 Example Description

The program aims to show how to DMA or polling mode to write and read SD card data by SDIO, in this case, The data can write to SD card or read data from SD card by SDIO. Verification will occur after transmission,

The SD card sector data can be displayed using serial assistant. After power on, can switch between single block test or multi block test by KEY1 and KEY2.

KEY1 ----> single block test

KEY2 ----> multi block test

4.53.2 Directory contents

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

4.54 SPI_FullDuplex

4.54.1 Example Description

This example shows how to use SPI Peripheral to transfer data. Press KEY1 to send data from SPI1 to SPI2, if communication success, LED2 will turn on. The data will be shown on serial assistant trough usart1. Press KEY2 to use SPI1 to SPI2 to Realize full-duplex transmission. If communication success, LED3 will turn on. The data will be shown on serial assistant trough usart1.

4.54.2 Directory contents

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

4.55 SysTick_TimeBase

4.55.1 Example Description

This example describes how to use Systick_Delay for toggling IO. The IO of LED2 and LED3 is configed to toggle constantly every 1000 milliseconds. The phenomenon of LED2 and LED3 constantly flickered alternately. Delay time is printed with USART1.

4.55.2 Directory contents

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

4.56 Template

4.56.1 Example Description

This demo is based on the APM32F407 MIN board. it provides a template project.

4.56.2 Directory contents

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

4.57 TMR_32BitCount

4.57.1 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.

4.57.2 Directory contents

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

4.58 TMR_6Steps

4.58.1 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 100ms.

4.58.2 Directory contents

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

4.59 TMR_CascadeSynchro

4.59.1 Example Description

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

4.59.2 Directory contents

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

4.60 TMR_EncoderInterface

4.60.1 Example Description

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

4.60.2 Directory contents

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

4.61 TMR_ExtTriggerSynchro

4.61.1 Example Description

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

4.61.2 Directory contents

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

4.62 TMR_InputCapture

4.62.1 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.

4.62.2 Directory contents

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

4.63 TMR_OCAActive

4.63.1 Example Description

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

4.63.2 Directory contents

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

4.64 TMR_OCInactive

4.64.1 Example Description

The program to show how to configure the TMR3 peripheral in Output Compare Inactive mode.

4.64.2 Directory contents

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

4.65 TMR_OCToggle

4.65.1 Example Description

The program to show how to configure the TMR4 peripheral to generate 4 waveform with 4 different frequencies (2.5KHz, 5KHz, 25KHz and 50KHz).

4.65.2 Directory contents

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

4.66 TMR_ParallelSynchro

4.66.1 Example Description

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

4.66.2 Directory contents

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

4.67 TMR_PWMInput

4.67.1 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.

4.67.2 Directory contents

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

4.68 TMR_PWMOutput

4.68.1 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. using TMR1 CHANNEL1(PA8 and PA7) to output PWM.

4.68.2 Directory contents

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

4.69 TMR_SinglePulse

4.69.1 Example Description

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

4.69.2 Directory contents

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

4.70 TMR_TimeBase

4.70.1 Example Description

This example describes how to use TMR1 for toggling IO. The IO of LED2 is configed to toggle constantly every one second. The phenomenon of LED2 constantly flickered alternately.

4.70.2 Directory contents

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

4.71 TMR_TMR11PWMOutput

4.71.1 Example Description

The program to show how to configure the TMR11 peripheral in PWM mode.

4.71.2 Directory contents

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

4.72 TMR_TMR1DMABurst

4.72.1 Example Description

The program to show how to configure the TMR1 channel period and the duty cycle by DMA burst to generate 7 PWM with 7 different duty cycles (80%, 70%, 60%, 50%, 40%, 30% and 20%).

4.72.2 Directory contents

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

4.73 TMR_TMR1PWMOutput

4.73.1 Example Description

The program to show how to configure the TMR1 peripheral to generate 7 PWM with 7 different duty cycles (80%, 70%, 60%, 50%, 40%, 30% and 20%).

4.73.2 Directory contents

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

4.74 TMR_TMR1Synchro

4.74.1 Example Description

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

4.74.2 Directory contents

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

4.75 TMR_TMR2PWMOutput

4.75.1 Example Description

The program to show how to configure the TMR2 peripheral to generate 4 PWM with 4 different duty cycles (80%, 70%, 60%, and 50%).

4.75.2 Directory contents

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

4.76 TMR_TMR8DMA

4.76.1 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.

4.76.2 Directory contents

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

4.77 TMR_TMR9OCToggle

4.77.1 Example Description

The program to show how to configure the TMR9 peripheral to generate 2 waveform with 2 different frequencies (5KHz and 50KHz).

4.77.2 Directory contents

This example can be found in the [~ /Examples/TMR/TMR_TMR9OCToggle](#) directory.

4.78 USART_Interrupt

4.78.1 Example Description

The program aims to show how to use interrupt to send or received data by using USART, in this case, USART1 and USART2 send or received data to each other. Verification will occur after transmission, if Data transmission pass from USART1 to USART2, LED2 will be on. If data transmission pass from USART2 to USART1, LED3 will be on.

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

4.78.2 Directory contents

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

4.79 USART_Polling

4.79.1 Example Description

The program aims to show how to use polling to send or received data by using USART, in this case, USART1 and USART2 send or received data to each other. Verification will occur after transmission, if Data transmission pass from USART1 to USART2, LED2 will be on. If data transmission pass from USART2 to USART1, LED3 will be on.

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

4.79.2 Directory contents

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

4.80 WWDT_OverTime

4.80.1 Example Description

This example aims to show how to use WWDT. At start, is OverTime = 0, System would not reset for feeding dog timely. LED2 Toggle. if press KEY1, then is OverTime = 1, System will reset. LED3 ON.

4.80.2 Directory contents

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

5 About libraries

The libraries folder includes a series library. It can provide supports for APM32F4xx MCU such as device support and standard peripheral and USB OTG etc. The libraries can be found in the [~/Libraries](#) directory.

APM32F4xx MCU include following library:

- Libraries folder
 - * APM32F4xx_ETH_Driver
 - * APM32F4xx_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 APM32F4xx MINI include following:

- Middlewares folder

- * fat_fs
- * FreeRTOS
- * lwip-1.4.1
- * RealThread

7 About Package

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

The middlewares used by APM32F4xx MINI include following:

- Package folder
 - * Geehy.APM32F4xx_DFP.1.0.2.pack

8 Revision History

Table 1 File Revision History

Date	Rev	Description
2022.06.23	1.0	First Release version of APM32F4xx 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