Readme

APM32A10x SDK

Rev：V1.0

© Geehy Semiconductor Co., Ltd.

1 Introduction

The Geehy Semiconductor APM32A10x 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_AnalogWatchdog 13

4.2 ADC_ContinuousConversion 13

4.3 ADC_DMA 13

4.4 ADC_DualRegulSimulMode 14

4.5 ADC_MultiChannelScan 14

4.6 ADC_ADC3DMA 14

4.7 ADC_ExtLinesTrigger 15

4.8 ADC_TemperatureVref 15

4.9 BAKPR_Tamper 15

4.10 CAN_LoopBack 16

4.11 CAN_Normal 16

4.12 CRC_Calculation 17

4.13 DAC_ADC 17

4.14 DAC_NoiseWave 18

4.15 DAC_IndependentTriggerLFSR_Noise 18

4.16 DAC_LFSR_NoiseNode 18

4.17 DAC_OneChannelDMA_Escalator 19

4.18 DAC_OutputVoltage 19

4.19 DAC_softwareTriggerLFSR_Noise 20

4.20 DAC_TriangleNoiseMode 20

4.21 DMA_ADC 20

4.22 DMA_MemoryToMemory 21

4.23 DMA_UsartToMemory 21
4.24 DMA_I2C_RAM 22
4.25 DMA_ReloadExti 22
4.26 DMA_SPI_RAM 22
4.27 EINT_Config 23
4.28 DMC_NAND 23
4.29 DMC_NOR 24
4.30 DMC_OneNAND 24
4.31 FMC_Program 24
4.32 FMC_Protection 25
4.33 Flash_Emulation_Eeprom 25
4.34 FMC_OptionOperation 26
4.35 FPU_Math 26
4.36 GPIO_Toggle 26
4.37 GPIO_JTAG_Remap 27
4.38 I2C_TwoBoards 27
4.39 I2S_Interrupt 27
4.40 IAP_Application1 28
4.41 IAP_Application2 28
4.42 IAP_BootLoader 28
4.43 IWDT_Reset 28
4.44 NVIC_Priority 29
4.45 NVIC_WFI 29
4.46 PMU_Standby 30
4.47 PMU_Stop 30
4.48 RCM_ClockConfig 30
4.49 RTC_Alarm 31
4.50 RTC_Second 31

4.51 FreeRTOS 32
4.52 RT-Thread 32
4.53 RTX5 32
4.54 SPI_FullDuplex 33
4.55 SPI_CRC 33
4.56 SPI_DMA 33
4.57 SPI_FLASH 34
4.58 SPI_Interrupt 34
4.59 SysTick_TimeBase 34
4.60 Template 35
4.61 TMR_6Steps 35
4.62 TMR_32BitCount 35
4.63 TMR_CascadeSynchro 36
4.64 TMR_EncoderInterface 36
4.65 TMR_ExtTriggerSynchro 36
4.66 TMR_InputCapture 37
4.67 TMR_OCActive 37
4.68 TMR_OCInactive 37
4.69 TMR_OCToggle 38
4.70 TMR_ParallelSynchro 38
4.71 TMR_PWMInput 38
4.72 TMR_PWMOutput 39
4.73 TMR_SinglePulse 39
4.74 TMR_TimeBase 39
4.75 TMR_TMR1DMA 40
4.76 TMR_TMR1DMABurst 40
4.77 TMR_TMR1Synchro 40
4.78 USART_Interrupt 40

4.79 USART_IrDA 41
4.80 USART_LIN 41
4.81 USART_Printf 41
4.82 USART_Smartcard 42
4.83 USART_DMA_Interrupt 42
4.84 USART_HalfDuplex 42
4.85 USART_Polling 43
4.86 USART_SPI_Synchronous 43
4.87 USB_CDC_VirtualCOMPort 44
4.88 USB_HID_Mouse 44
4.89 USB_MSC_Disk 44
4.90 WWDT_Reset 45
5 About libraries 46
6 About middlewares 47
7 About Package 48
8 Revision History 49

2 About boards

The boards folder includes a board support package for APM32A10x 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 APM32A10x MINI board compatibility. For other user development board use, some minor modifications may be required.

Boards have a hierarchy as follows:

- Boards folder
  - Board_APM32A103_MINI
    - inc
    - src
  - Board.c
  - Board.h

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 APM32A10x 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
    - Eclipse
    - IAR
    - MDK
  - Source

All example applications tested with: **APM32A10x StdPeriphDriver v1.8**, include the following examples:

- Examples
  - ADC
    - ADC_AnalogWindowWatchdog
    - ADC_ContinuousConversion
    - ADC_DMA
    - ADC_DualRegulSimulMode
    - ADC_MultiChannelScan
    - ADC_ADC3DMA
    - ADC_ExtLinesTrigger
    - ADC_TemperatureVref
  - BAKPR
    - BAKPR_Tamper
  - CAN
    - CAN_LoopBack

- CAN_TwoBoards

* CRC
  - CRC_Calculation
* DAC
  - DAC_ADC
  - DAC_NoiseWave
  - DAC_IndependentTriggerLFSR_Noise
  - DAC_LFSR_NoiseNode
  - DAC_OneChannelDMA_Escalator
  - DAC_OutputVoltage
  - DAC_softwareTriggerLFSR_Noise
  - DAC_TriangleNoiseMode
* DMA
  - DMA_ADC
  - DMA_MemoryToMemory
  - DMA_UsartToMemory
  - DMA_I2C_RAM
  - DMA_ReloadExti
  - DMA_SPI_RAM
* EINT
  - EINT_Config
* EMMC
  - DMC_NAND
  - DMC_NOR
  - DMC_OneNAND
* FMC
  - FMC_Program
  - FMC_Protection

- Flash_Emulation_Eeprom
- FMC_OptionOperation

* FPU
  - FPU_Math
* GPIO
  - GPIO_Toggle
  - GPIO_JTAG_Remap
* I2C
  - I2C_TwoBoards
* I2S
  - I2S_Interrupt
* IAP
  - IAP_Application1
  - IAP_Application2
  - IAP_BootLoader
* IWDT
  - IWDT_Reset
* NVIC
  - NVIC_Priority
  - NVIC_WFI
* PMU
  - PMU_Standby
  - PMU_Stop
* RCM
  - RCM_ClockConfig
* RTC
  - RTC_Alarm
  - RTC_Second

* RTOS
  - FreeRTOS
  - RT-Thread
  - RTX5
* SPI
  - SPI_FullDuplex
  - SPI_CRC
  - SPI_DMA
  - SPI_FLASH
  - SPI_Interrupt
* SysTick
  - SysTick_TimeBase
* Template
  - Template
* TMR
  - TMR_6Steps
  - TMR_32BitCount
  - TMR_CascadeSynchro
  - TMR_EncoderInterface
  - TMR_ExtTriggerSynchro
  - TMR_InputCapture
  - TMR_OCActive
  - TMR_OCInactive
  - TMR_OCToggle
  - TMR_ParallelSynchro
  - TMR_PWMInput
  - TMR_PWMOutput
  - TMR_SinglePulse

- TMR_TimeBase
- TMR_TMR1DMA
- TMR_TMR1DMABurst
- TMR_TMR1Synchro

* USART
  - USART_Interrupt
  - USART_IrDA
  - USART_LIN
  - USART_Printf
  - USART_Smartcard
  - USART_Polling
  - USART_DMA_Interrupt
  - USART_HalfDuplex
  - USART_SPI_Synchronous
* 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(PC4) continuously.

When input Voltage of ADC1_Channel14(PC4) voltage is lower than 0.62v or higher than 2.27V, analog watchdog interrupt will be generated and LED2 is on.

4.1.2 Directory content

This example can be found in the ADC_AnalogWatchdog directory.

4.2 ADC_ContinuousConversion

4.2.1 Example Description

This example describes how to use the ADC1 to convert continuously the voltage applied to the APM32A103 MINI ADC1_Channel0 input.

The converted voltage is displayed on serial assistant through USART1.

4.2.2 Directory content

This example can be found in the ADC_ContinuousConversion directory.

4.3 ADC_DMA

4.3.1 Example Description

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

The ADC channel 0 for APM32A103 MINI Board is configured to be converted when device startup.

The converted voltage is displayed on serial assistant through USART1.

4.3.2 Directory content

This example can be found in the ADC_DMA directory.

4.4 ADC_DualRegulSimulMode

4.4.1 Example Description

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

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

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

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

4.4.2 Directory content

This example can be found in the ADC_DualRegulSimulMode directory.

4.5 ADC_MultiChannelScan

4.5.1 Example Description

This example describes how to use the ADC1 to scan continuously the voltage

applied to the APM32A103 MINI ADC1_Channel0, ADC1_Channel1 and ADC1_Channel2 input.

The converted voltage is displayed on serial assistant through USART1.

4.5.2 Directory content

This example can be found in the ADC_MultiChannelScan directory.

4.6 ADC_ADC3DMA

4.6.1 Example Description

This example describes how to use the 3 ADCs in independent conversions.

Two of them; ADC1 and ADC3 are transferring continuously converted data using DMA

while ADC2 converted data are stored using End of conversion interrupt.

4.6.2 Directory content

This example can be found in the ADC_ADC3DMA directory.

4.7 ADC_ExtLinesTrigger

4.7.1 Example Description

This example describes how to trigger ADC regular and injected groups channels conversion using two external line events. Discontinuous mode is enabled for regular group channel conversion and configured to convert one regular channel on each external trigger. ADC1 is configured to start regular group channel conversion on EXTI15 event. On detection of the first rising edge on PE15 pin, the conversion of the first regular channel (ADC1 channel1) is done and its converted value is transferred by DMA to ADC_RegularConvertedValueTab table.

4.7.2 Directory content

This example can be found in the ADC_ExtLinesTrigger directory.

4.8 ADC_TemperatureVref

4.8.1 Example Description

This example describes how to use the ADC to convert analog signal to digital data.

The ADC is configured in dependent mode, inner channel16(temperature sensor channel) and channel17(Vref channel) are chosen as analog input pin.

The converted voltage is displayed on serial assistant through USART1.

4.8.2 Directory content

This example can be found in the ADC_TemperatureVref directory.

4.9 BAKPR_Tamper

4.9.1 Example Description

This example describes how to write the backup registers.

After initialization, System enters into an infinite loop.

if the data in the backup registers is equal to the data write to the registers before, LED2 keeps blinking, otherwise, LED3 keeps blinking. TAMPER pin is also enabled, if the pin changes from 0 to 1 or from 1 to 0, The TAMPER pin generates a Tamper detection event to reset all data backup registers.

4.9.2 Directory content

This example can be found in the BAKPR_Tamper directory.

4.10 CAN_LoopBack

4.10.1 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.

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

4.10.2 Directory content

This example can be found in the CAN_LoopBack directory.

4.11 CAN_Normal

4.11.1 Example Description

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

Transmit:

If press the KEY1, CAN1 transmit, the USART1 print "CAN1 Transmit 0x55", LED2 turns on.

If press the KEY2, CAN2 transmit, the USART1 print "CAN2 Transmit 0xAA", LED2 turns on.

Receive:

If CAN1 receive success, the USART1 print "CAN1 Receive 0xAA", LED2 turns on, else "CAN1 Receive Error", LED3 turns on.

If CAN2 receive success, the USART1 print "CAN2 Receive 0x55", LED2 turns on, else "CAN2 Receive Error", LED3 turns on.

4.11.2 Directory content

This example can be found in the CAN_Normal directory.

4.12 CRC_Calculation

4.12.1 Example Description

Write the calculated data to CRC DATA register and get the calculated result.

The phenomenon of Computed CRC compasses Expected CRC.

The results will be displayed on serial assistant through USART1.

4.12.2 Directory content

This example can be found in the CRC_Calculation directory.

4.13 DAC_ADC

4.13.1 Example Description

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

-DAC generates voltage from 0V to 3.3V

-when the voltage is greater than 2.0V, then LED2 turns on and LED3 turns off.

-when the voltage is less than 0.8V, then LED2 turns off and LED3 turns on.

-when the voltage is greater than 0.8V but less than 2.0V, then LED2 and LED3 turn on.

4.13.2 Directory content

This example can be found in the DAC_ADC directory.

4.14 DAC_NoiseWave

4.14.1 Example Description

This example shows how to configure the DAC peripheral to generate noise wave.

The waveform can be displayed using an oscilloscope.

using DAC_CHANNEL_1(PA4) to output noise wave.

4.14.2 Directory content

This example can be found in the DAC_NoiseWave directory.

4.15 DAC_IndependentTriggerLFSR_Noise

4.15.1 Example Description

This example shows how to use DAC mode independent trigger to generate LFSR noise wave with different configurations. TIMER6 is chosen to trigger DAC_CHANNEL1. TIMER7 is chosen to trigger DAC_CHANNEL2. The DAC_CHANNEL1 output pin is configured PA4 and DAC_CHANNEL2 output pin is configured PA5. The LFSR noise wave can be observed through the oscilloscope.

4.15.2 Directory content

This example can be found in the DAC_IndependentTriggerLFSR_Noise directory.

4.16 DAC_LFSR_NoiseNode

4.16.1 Example Description

This example shows how to use DAC_CHANNEL1 to generate LFSR noise wave. TIMER6 is chosen to trigger DAC_CHANNEL. The DAC_CHANNEL1 output pin is configured PA4. The LFSR noise wave can be observed through the oscilloscope.

4.16.2 Directory content

This example can be found in the DAC_LFSR_NoiseNode directory.

4.17 DAC_OneChannelDMA_Escalator

4.17.1 Example Description

This example describes how to use one DAC channel mode with DMA to generate an escalator signal on DAC channel1 output.

DAC channel1 conversion is configured to be triggered by TIM6 TRGO triggers and without noise/triangle wave generation. 8bit right data alignment is selected since we choose to access DAC_DHR8R1 register.

DMA2 channel3 is configured to transfer continuously, byte by byte, a 6-byte buffer to the DAC1 register DAC_DHR8R1.

The transferred 6bytes buffer is made to have an escalator signal on DAC channel1 output. DAC channel1 is then enabled. Once TIM6 is enabled, each TIM6 TRGO update event generate a DMA request which transfer data to the DAC1 register and DAC conversion is started. The escalator signal can be visualized by connecting PA.04 pin to an oscilloscope.

4.17.2 Directory content

This example can be found in the DAC_OneChannelDMA_Escalator directory.

4.18 DAC_OutputVoltage

4.18.1 Example Description

This example shows how to use DAC concurrent mode output voltage. No trigger source is chosen to trigger DAC. The DAC_CHANNEL1 output pin is configure PA4 and DAC_CHANNEL2

output pin is configured PA5. There are two different voltage in PA4 and PA5.

4.18.2 Directory content

This example can be found in the DAC_OutputVoltage directory.

4.19 DAC_softwareTriggerLFSR_Noise

4.19.1 Example Description

This example shows how to use DAC mode software trigger to generate trigger LFSR noise wave with different configurations. The DAC_CHANNEL1 output pin is configured PA4 and DAC_channel2 output pin is configured PA5. The LFSR noise wave can be observed through the oscilloscope.

4.19.2 Directory content

This example can be found in the DAC_softwareTriggerLFSR_Noise directory.

4.20 DAC_TriangleNoiseMode

4.20.1 Example Description

This example shows how to use DAC_CHANNEL1 to generate LFSR noise wave. TIMER6 is chosen to trigger DAC_CHANNEL. The DAC_CHANNEL1 output pin is configured PA4. The LFSR noise wave can be observed through the oscilloscope.

4.20.2 Directory content

This example can be found in the DAC_TriangleNoiseMode directory.

4.21 DMA_ADC

4.21.1 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 channel0 for APMF103 MINI Board is configured to be converted when device startup.

The data displayed on serial assistant through USART1.

4.21.2 Directory content

This example can be found in the DMA_ADC directory.

4.22 DMA_MemoryToMemory

4.22.1 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.

4.22.2 Directory content

This example can be found in the DMA_ MemoryToMemory directory.

4.23 DMA_UsartToMemory

4.23.1 Example Description

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

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

4.23.2 Directory content

This example can be found in the DMA_UsartToMemory directory.

4.24 DMA_I2C_RAM

4.24.1 Example Description

This example provides a description of how to use two DMA channels to transfer a data buffer from memory to I2C2 through I2C1.

I2C1 use DMA_Channel6 to send data, and I2C2 use DMA_Channel5 to receive data.

If the data sent and received, LED2 is on.

4.24.2 Directory content

This example can be found in the DMA_I2C_RAM directory.

4.25 DMA_ReloadExti

4.25.1 Example Description

The example provides a description of how to reconfigure the DMA transfer number in discontinuous mode, when use DMA channel3 to transfer data from RAM memory to USART1 transmit data register.The start of transfer is triggered by software. If the DMA transfer operation is finished correctly, data stored array DataBuffer[] will be transfered to a serial port tool by USART1 and LED2 toggles.

When the USER key is pressed, the DMA transfer number is reconfigured and DMA transmission is restarted.

Press KEY1, DMA will reload.

The data is displayed on serial assistant through USART1.

4.25.2 Directory content

This example can be found in the DMA_ReloadExti directory.

4.26 DMA_SPI_RAM

4.26.1 Example Description

This example provides a description of how to set a communication between the two

SPIs in simplex mode and performs a transfer from SPI_MASTER in polling mode to the SPI_SLAVE in DMA receive mode. A dedicated DMA channel is configured for SPI_SLAVE Rx request to store received data in SPI_SLAVE_Buffer_Rx. If SPI data for transmiting is true, the LED2 is toggle.

4.26.2 Directory content

This example can be found in the DMA_SPI_RAM directory.

4.27 EINT_Config

4.27.1 Example Description

This example shows how to configure external interrupt lines.

In this example, 2 EINT lines (KEY1 and KEY2) when using the APM32A103 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)

After EINT configuration

when falling edge is detected on EINT0, LED2 toggles

when falling edge is detected on EINT1, LED3 toggles

4.27.2 Directory content

This example can be found in the EINT_Config directory.

4.28 DMC_NAND

4.28.1 Example Description

This example provides a basic example of how to use the EMMC firmware library and an associate driver to perform erase/read/write operations on the NAND512W3A2 memory.

4.28.2 Directory content

This example can be found in the DMC_NAND directory.

4.29 DMC_NOR

4.29.1 Example Description

This example provides a basic example of how to use the EMMC firmware library and an associate driver to perform erase/read/write operations on the M29W128FL, M29W128GL or S29GL128P NOR memories.

4.29.2 Directory content

This example can be found in the DMC_NOR directory.

4.30 DMC_OneNAND

4.30.1 Example Description

This example shows how to configure the FSMC to drive the OneNAND memory.
In this example a basic example of how to use the FSMC firmware library and an associate driver to perform read/write operations on the KFG1216U2A/B-DIB6 OneNAND memory.

4.30.2 Directory content

This example can be found in the DMC_OneNAND directory.

4.31 FMC_Program

4.31.1 Example Description

This example shows how to program the flash address of APM32A103.

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

4.31.2 Directory content

This example can be found in the FMC_Program directory.

4.32 FMC_Protection

4.32.1 Example Description

This example shows how to set write protection for the flash address of APM32A103.

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

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

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

The programming performance is showing by USART1.

4.32.2 Directory content

This example can be found in the FMC_Protection directory.

4.33 Flash_Emulation_Eeprom

4.33.1 Example Description

The example shows how to use FLASH to emulation EEPROM.

You can read/write bytes to Flash using the read/write interface provided in this module.

4.33.2 Directory content

This example can be found in the Flash_Emulation_Eeprom directory.

4.34 FMC_OptionOperation

4.34.1 Example Description

This example shows how to operate option bytes to enable the chip's read or write protection.

Press the KEY1 to turn on the read protection, and it must be powered off and restarted to take effect; Press KEY1 again to turn off the read protection, wait 20s for 20s to run the flash programming program; Pressing the KEY2 will turn on the write protection, pressing KEY2 again will turn off the write protection, and pressing the reset button will take effect.

4.34.2 Directory content

This example can be found in the FMC_OptionOperation directory.

4.35 FPU_Math

4.35.1 Example Description

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

4.35.2 Directory content

This example can be found in the FPU_Math directory.

4.36 GPIO_Toggle

4.36.1 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.

4.36.2 Directory content

This example can be found in the GPIO_Toggle directory.

4.37 GPIO_JTAG_Remap

4.37.1 Example Description

This example provides a short description of how to use the JTAG IOs as standard GPIOs and gives a configuration sequence. USART1 sends data to upper computer. You can check the data in a Serial Port Utility.

4.37.2 Directory content

This example can be found in the GPIO_JTAG_Remap directory.

4.38 I2C_TwoBoards

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 APM32A103_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 content

This example can be found in the I2C_TwoBoards 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 and LED3 will turn on.

4.39.2 Directory contents

This example can be found in the I2S_Interrupt directory.

4.40 IAP_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 IAP_Application1 directory.

4.41 IAP_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 IAP_Application2 directory.

4.42 IAP_BootLoader

4.42.1 Example Description

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

When device connect to HyperTerminal right, a usart menu will show to user.

4.42.2 Directory contents

This example can be found in the 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, 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.

4.43.2 Directory contents

This example can be found in the 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 content

This example can be found in the 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 KEY2(PA0) to occur wait for Interrupt(WFI) event, and device will enter sleep mode. The device will wake up if press KEY2 again.

4.45.2 Directory content

This example can be found in the NVIC_WFI directory.

4.46 PMU_Standby

4.46.1 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.

4.46.2 Directory content

This example can be found in the PMU_Standby directory.

4.47 PMU_Stop

4.47.1 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.

4.47.2 Directory content

This example can be found in the PMU_Stop directory.

4.48 RCM_ClockConfig

4.48.1 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)

4.48.2 Directory content

This example can be found in the RCM_ClockConfig directory.

4.49 RTC_Alarm

4.49.1 Example Description

This example provides an example of how to use RTC alarm in APM32A103 MINI Board.

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

And press KEY1 can set an alarm for 5 seconds.

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

4.49.2 Directory content

This example can be found in the RTC_Alarm directory.

4.50 RTC_Second

4.50.1 Example Description

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

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

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

4.50.2 Directory content

This example can be found in the RTC_Second directory.

4.51 FreeRTOS

4.51.1 Example Description

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

USART test task: USART1 and USART2 send or received data to each other. Verification will occur after transmission.

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

The phenomenon of LED2 constantly flickered alternately,

if data transmission pass, LED3 will be on,

if data transmission fault, LED3 will be off.

4.51.2 Directory content

This example can be found in the FreeRTOS directory.

4.52 RT-Thread

4.52.1 Example Description

This example describes how to use RT-Thread for APM32A10x.

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

The phenomenon of LED2 and LED3 constantly flickered alternately.

4.52.2 Directory content

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

4.53 RTX5

4.53.1 Example Description

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

USART test task: USART1 and USART2 send or received data to each other.

Verification will occur after transmission,

Led toggle task: The IO of LED2 is configured to toggle constantly,

The phenomenon of LED2 constantly flickered alternately,

if Data transmission pass, LED3 will be on,

if Data transmission fault, LED3 will be off.

4.53.2 Directory content

This example can be found in the RTX5 directory.

4.54 SPI_FullDuplex

4.54.1 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.

4.54.2 Directory content

This example can be found in the SPI_FullDuplex directory.

4.55 SPI_CRC

4.55.1 Example Description

This example provides a description of how to set a communication between two

SPIs in full-duplex mode and performs a transfer from Master to Slave and

Slave to Master followed by CRC transmission.

4.55.2 Directory content

This example can be found in the SPI_CRC directory.

4.56 SPI_DMA

4.56.1 Example Description

This example provides a description of how to set a communication between the two

SPIs in simplex mode and performs a transfer from SPI_MASTER in polling mode to the SPI_SLAVE in DMA receive mode. A dedicated DMA channel is configured for SPI_SLAVE Rx request to store received data in SPI_SLAVE_Buffer_Rx. If SPI data for transmiting is true, the LED3 will be on.If the program is running normally, LED2 will blink.

4.56.2 Directory content

This example can be found in the SPI_DMA directory.

4.57 SPI_FLASH

4.57.1 Example Description

This example provides a basic example of how to use the SPI firmware library and an associate SPI FLASH driver to communicate with an W25QXX FLASH.

4.57.2 Directory content

This example can be found in the SPI_FLASH directory.

4.58 SPI_Interrupt

4.58.1 Example Description

This example provides a description of how to set a communication between two SPIs in simplex mode and performs a data buffer transfer from SPI1 interrupt for master to SPI2 interrupt for slave. If data is sent correctly, the LED2 will light. The LED3 will blinky.

4.58.2 Directory content

This example can be found in the SPI_Interrupt directory.

4.59 SysTick_TimeBase

4.59.1 Example Description

The example shows how to configure the SysTick to generate a time base equal to 1 millisecond. 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.

4.59.2 Directory content

This example can be found in the SysTick_TimeBase directory.

4.60 Template

4.60.1 Example Description

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

4.60.2 Directory contents

This example can be found in the Template directory.

4.61 TMR_6Steps

4.61.1 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.

Display TMR1 waveform by oscilloscope.

4.61.2 Directory contents

This example can be found in the TMR_6Steps directory.

4.62 TMR_32BitCount

4.62.1 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.

4.62.2 Directory contents

This example can be found in the TMR_32BitCount directory.

4.63 TMR_CascadeSynchro

4.63.1 Example Description

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

4.63.2 Directory contents

This example can be found in the TMR_CascadeSynchro directory.

4.64 TMR_EncoderInterface

4.64.1 Example Description

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

4.64.2 Directory contents

This example can be found in the TMR_EncoderInterface directory.

4.65 TMR_ExtTriggerSynchro

4.65.1 Example Description

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

The starts and stops of the TMR1 counters are controlled by the external trigger.

The TMR2 starts and stops are controlled by the TMR1, and the TMR3 starts and stops are controlled by the TMR2.

4.65.2 Directory contents

This example can be found in the TMR_ExtTriggerSynchro directory.

4.66 TMR_InputCapture

4.66.1 Example Description

This example describes how to use TMR5 Channel2(PA0) measure frequency of external signal. User can view the "Frequency" value through serial terminal.

4.66.2 Directory contents

This example can be found in the TMR_InputCapture directory.

4.67 TMR_OCActive

4.67.1 Example Description

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

The CHx delay correspond to the time difference between PE6 falling edge and TMR2_CHx signal rising edges. Reset system and display TMR2 waveform by oscilloscope.

4.67.2 Directory contents

This example can be found in the TMR_OCActive directory.

4.68 TMR_OCInactive

4.68.1 Example Description

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

While the counter is lower than the Output compare registers values, the PA1, PA2, PA3, and PA4 are set.

When the counter value reaches the Output compare registers values, the PA1, PA2, PA3,

and PA4 are reset.

The CHx delay correspond to the time difference between PAx rising edges and falling edge. Reset system and display PA1, PA2, PA3, and PA4 waveform by oscilloscope.

4.68.2 Directory contents

This example can be found in the TMR_OCInactive directory.

4.69 TMR_OCToggle

4.69.1 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).

Display TMR2 waveform by oscilloscope.

4.69.2 Directory contents

This example can be found in the TMR_OCToggle directory.

4.70 TMR_ParallelSynchro

4.70.1 Example Description

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

4.70.2 Directory contents

This example can be found in the TMR_ParallelSynchro directory.

4.71 TMR_PWMInput

4.71.1 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.

4.71.2 Directory contents

This example can be found in the TMR_PWMInput directory.

4.72 TMR_PWMOutput

4.72.1 Example Description

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

4.72.2 Directory contents

This example can be found in the TMR_PWMOutput directory.

4.73 TMR_SinglePulse

4.73.1 Example Description

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

4.73.2 Directory content

This example can be found in the TMR_SinglePulse directory..

4.74 TMR_TimeBase

4.74.1 Example Description

This example shows how to realize timing one second by using TMR1 peripheral generating time base. LED2 will toggle every 500 milliseconds.

4.74.2 Directory content

This example can be found in the TMR_TimeBase directory.

4.75 TMR_TMR1DMA

4.75.1 Example Description

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

Display TMR1 waveform by oscilloscope.

4.75.2 Directory content

This example can be found in the TMR_TMR1DMA directory.

4.76 TMR_TMR1DMABurst

4.76.1 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%).

Display TMR1 waveform by oscilloscope.

4.76.2 Directory content

This example can be found in the TMR_TMR1DMABurst directory.

4.77 TMR_TMR1Synchro

4.77.1 Example Description

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

4.77.2 Directory content

This example can be found in the TMR_TMR1Synchro directory.

4.78 USART_Interrupt

4.78.1 Example Description

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

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

4.78.2 Directory content

This example can be found in the USART_Interrupt directory.

4.79 USART_IrDA

4.79.1 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.

4.79.2 Directory content

This example can be found in the USART_IrDA directory.

4.80 USART_LIN

4.80.1 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.

4.80.2 Directory content

This example can be found in the USART_LIN directory.

4.81 USART_Printf

4.81.1 Example Description

The program shows how to send data by using USART, in this case,

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

4.81.2 Directory content

This example can be found in the USART_Printf directory.

4.82 USART_Smartcard

4.82.1 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.

4.82.2 Directory content

This example can be found in the USART_Smartcard directory.

4.83 USART_DMA_Interrupt

4.83.1 Example Description

This example provides a basic communication between USART1 and USART2 using DMA capability, flags and interrupts.

First, the DMA transfers data from TxBuffer2 buffer to USART2 Transmit data register, then this data is sent to USART1.

In the same time, the DMA transfers data from TxBuffer1 buffer to USART1 Transmit data register, then this data is sent to USART2.

4.83.2 Directory content

This example can be found in the USART_DMA_Interrupt directory.

4.84 USART_HalfDuplex

4.84.1 Example Description

The USART1 will send data from TxDataBuffer1 to USART2, Data received by USART2 is stored in TxDataBuffer1 then compared with the sent ones. If the result is true, the LED2 will be on.

The USART2 will send data from TxDataBuffer2 to USART1, Data received by USART1

is stored in TxDataBuffer2 then compared with the sent ones. If the result is true, the LED3 will be on.

4.84.2 Directory content

This example can be found in the USART_HalfDuplex directory.

4.85 USART_Polling

4.85.1 Example Description

The program aims to show how to 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.

4.85.2 Directory content

This example can be found in the USART_Polling directory.

4.86 USART_SPI_Synchronous

4.86.1 Example Description

This example shows a basic communication between USART1 (Synchronous mode) and SPI1 using flags.

First, USART1 will send data to SPI1 through TxBuffer1. if the data transmitted from USART1 and received by SPI1 are the same, the LED2 will on. Then, SPI1 will send data to USART1 through TxBuffer2. If the data transmitted from SPI1 and received by USART1 are the same, the LED3 will on.

4.86.2 Directory content

This example can be found in the USART_SPI_Synchronous directory.

4.87 USB_CDC_VirtualCOMPort

4.87.1 Example Description

This example describes how to use the USB device module on APM32A103 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.

4.87.2 Directory content

This example can be found in the USB_CDC_VirtualCOMPort directory.

4.88 USB_HID_Mouse

4.88.1 Example Description

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

4.88.2 Directory content

This example can be found in the USB_HID_Mouse directory.

4.89 USB_MSC_Disk

4.89.1 Example Description

This example describes how to use the USB device module on APM32A103 to enumerated as a MSC USB disk. This example use PC as host, and APM32A103 use ram to simulate USB flash drives. PC will recognize the motherboard as a

disk, and formatting the U disk. This example will be Used to test USB speed.

4.89.2 Directory content

This example can be found in the USB_MSC_Disk directory.

4.90 WWDT_Reset

4.90.1 Example Description

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.

4.90.2 Directory content

This example can be found in the WWDT_Reset directory.

5 About libraries

The libraries folder includes a series library. It can provide supports for APM32A10x MCU such as device support and standard peripheral. The libraries can be found in the Libraries directory.

APM32A10x MCU include following library:

- Libraries folder
  - APM32A10x_ETH_Driver
  - APM32A10x_StdPeriphDriver
  - CMSIS
  - Device
  - USB_Device_Lib
  - USB_OTG_Lib

6 About middlewares

The middlewares can be found in the Middlewares directory.

The middlewares used by APM32A10x MINI include following:

- Middlewares folder
  - fat_fs
  - FreeRTOS
  - lwip-1.4.1
  - RealThread
  - RTX5

7 About Package

The Package folder includes Geehy APM32A10x_DFP Package. The Package can be found in the directory.

The Package used by APM32A10x MINI include following:

- Package folder
  - Geehy.APM32A10x_DFP.1.0.0.pack

8 Revision History

Table 1 File Revision History

Date	Rev	Description
2023.04.20	1.0	First release version of APM32A10x 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 thesales order or the sales contract shall prevail.

4. Information Reliability

The relevant data in this document are obtained from batch test by Geehy Laboratory or cooperative third-party testing organization. However, clerical errors in correction or errors caused by differences in testing environment may occur inevitably. Therefore, users should understand that Geehy does not bear any responsibility for such errors that may occur in this document. The relevant data in this document are only used to guide users as performance parameter reference and do not constitute Geehy's guarantee for any product performance.

Users shall select appropriate Geehy products according to their own needs, and effectively verify and test the applicability of Geehy products to confirm that Geehy products meet their own needs, corresponding standards, safety or other reliability requirements. If loses are caused to users due to the user's failure to fully verify and test Geehy products, Geehy will not bear any responsibility.

5. Legality

USERS SHALL ABIDE BY ALL APPLICABLE LOCAL LAWS AND REGULATIONS WHEN USING THIS DOCUMENT AND THE MATCHING GEEHY PRODUCTS. USERS SHALL UNDERSTAND THAT THE PRODUCTS MAY BE RESTRICTED BY THE EXPORT, RE-EXPORT OR OTHER LAWS OF THE COUNTIRIES OF THE PRODUCTS SUPPLIERS, GEEHY, GEEHY DISTRIBUTORS AND USERS. USERS (ON BEHALF OR ITSELF, SUBSIDIARIES AND AFFILIATED ENTERPRISES) SHALL AGREE AND PROMISE TO ABIDE BY ALL APPLICABLE LAWS AND REGULATIONS ON THE EXPORT AND RE-EXPORT OF GEEHY PRODUCTS AND/OR TECHNOLOGIES AND DIRECT PRODUCTS.

6. Disclaimer of Warranty

THIS DOCUMENT IS PROVIDED BY GEEHY "AS IS" AND THERE IS NO WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, TO THE EXTENT PERMITTED BY APPLICABLE LAW.

GEEHY WILL BEAR NO RESPONSIBILITY FOR ANY DISPUTES ARISING FROM THE SUBSEQUENT DESIGN OR USE BY USERS.

7. Limitation of Liability

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL GEEHY OR ANY OTHER PARTY WHO PROVIDE THE DOCUMENT "AS IS", BE LIABLE FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, DIRECT, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE DOCUMENT (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY USERS OR THIRD PARTIES).

8. Scope of Application

The information in this document replaces the information provided in all previous versions of the document.

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

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