

Readme

APM32F072 EVAL SDK

Rev: V1.0

1 Introduction

The Geehy Semiconductor APM32F072 EVAL 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	3
3	About documents.....	4
4	About examples.....	5
4.1	ADC_Potentiometer	7
4.2	CAN_LoopBack	7
4.3	CAN_TwoBoards	7
4.4	CEC_TwoBoardsExchange.....	7
4.5	I2C_EEPROM.....	8
4.6	LCD_Menu	8
4.7	RTC_Clock	8
4.8	TencentOS_Tiny	9
4.9	SPI_FLASH.....	9
4.10	Template	9
4.11	TSC_KeyLinearRotate.....	10
4.12	USART_Interrupt.....	11
4.13	USART_Polling	11
4.14	USB_CDC_VirtualCOMPort	11
4.15	USB_HID_Mouse	12
4.16	USB_MSC_Disk	12
5	About libraries	13
6	About middlewares	14
7	About Package	15
8	Revision History	16

2 About boards

The boards folder includes a board support package for APM32F072 EVAL 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 APM32F072 EVAL 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 APM32F072 EVAL include following board support package:

- Board_ APM32F072_ EVAL src folder
 - * board_apm32f072_eval
 - * bsp_cec
 - * bsp_delay
 - * bsp_i2c_ee
 - * bsp_key
 - * bsp_lcd
 - * bsp_w25q16

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 APM32F072 EVAL 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: **APM32F0xx StdPeriphDriver v1.0.3**, include the following examples:

- Examples

- * ADC
 - [ADC_Potentiometer](#)
- * CAN
 - [CAN_LoopBack](#)
 - [CAN_TwoBoards](#)
- * CEC
 - [CEC_TwoBoardsExchange](#)
- * I2C
 - [I2C_EEPROM](#)
- * RTC
 - [RTC_Clock](#)

- * RTOS
 - [TencentOS_Tiny](#)
- * SPI
 - [SPI_FLASH](#)
- * LCD
 - [LCD_Menu](#)
- * TSC
 - [TSC_KeyLinearRotate](#)
- * Template
 - [Template](#)
- * USART
 - [USART_Interrupt](#)
 - [USART_Polling](#)
- * USB
 - [USB_CDC_VirtualCOMPort](#)
 - [USB_HID_Mouse](#)
 - [USB_MSC_Disk](#)

4.1 ADC_Potentiometer

4.1.1 Example Description

This example shows how to use ADC Peripheral to detect potentiometer. Press KEY1 to start test. If detect operation is success, LCD will display the test information.

4.1.2 Directory contents

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

4.2 CAN_LoopBack

4.2.1 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.

4.2.2 Directory contents

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

4.3 CAN_TwoBoards

4.3.1 Example Description

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

4.3.2 Directory contents

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

4.4 CEC_TwoBoardsExchange

4.4.1 Example Description

To use this example, you need to load the same program into two APM32F072 EVAL boards

(let's call them device 01 and device 02) then connect these two boards through CEC lines.

4.4.2 Directory contents

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

4.5 I2C_EEPROM

4.5.1 Example Description

This example shows how to use I2C Peripheral to read and write EEPROM. Press KEY1 to write data to EEPROM. And read data from EEPROM. if read and write success, LED2 will turn on. The data will be shown on serial assistant trough usart1.if error, LED3 will turn on. And LCD will display the test information.

4.5.2 Directory contents

This example can be found in the [~/Examples/I2C/I2C_EEPROM](#) directory.

4.6 LCD_Menu

4.6.1 Example Description

This example shows how to use SPI Peripheral to driver LCD to display multi-level menu.

4.6.2 Directory contents

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

4.7 RTC_Clock

4.7.1 Example Description

This example shows how to use RTC Peripheral to make a clock. After power on, program will read the backup area data to determine whether to reconfigure RTC time. Then the RTC time will display in the LCD and written to the backup area.

4.7.2 Directory contents

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

4.8 TencentOS_Tiny

4.8.1 Example Description

This example shows how to use TencentOS Tiny for APM32F0xx. There are 4 task entries in the system. The "start task" will create 3 tasks with the some priority.

Task 1 is used to update the ADC sample values displayed on LCD. Task 2 is used to convert the sample value of ADC to the voltage value. Task 3 will print the "hello world" string at an interval of one second.

4.8.2 Directory contents

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

4.9 SPI_FLASH

4.9.1 Example Description

This example shows how to use SPI Peripheral to write data to W25Q16 flash and read data from W25Q16 flash. Press KEY1 to start test. If detect operation is success, LCD will display the test information.

4.9.2 Directory contents

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

4.10 Template

4.10.1 Example Description

This demo is based on the APM32F072 EVAL board. It provides a template project.

4.10.2 Directory contents

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

4.11 TSC_KeyLinearRotate

4.11.1 Example Description

This example shows how to use TSC (Touch sensing controller) to achieve Key, Linear, Rotation control. The linear rotation touch will display the value on the LCD.

4.11.2 Directory contents

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

4.12 USART_Interrupt

4.12.1 Example Description

This example shows how to use USART1 Peripheral to send data and receive data from serial assistant by interrupt. Press KEY1 to write data to serial assistant. And you can send data to device use serial assistant.

4.12.2 Directory contents

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

4.13 USART_Polling

4.13.1 Example Description

This example shows how to use USART1 Peripheral to send data and receive data from serial assistant by polling. Press KEY1 to write data to serial assistant. And you can send data to device use serial assistant.

4.13.2 Directory contents

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

4.14 USB_CDC_VirtualCOMPort

4.14.1 Example Description

This example describes how to use the USB device module on APM32F072 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.

4.14.2 Directory contents

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

4.15 USB_HID_Mouse

4.15.1 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.

4.15.2 Directory contents

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

4.16 USB_MSC_Disk

4.16.1 Example Description

This example describes how to use the USB device module on APM32F072 to enumerated as a MSC USB disk. This example use PC as host, and APM32F072 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.

4.16.2 Directory contents

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

5 About libraries

The libraries folder includes a series library. It can provide supports for APM32F0xx MCU such as device support and standard peripheral and USB OTG etc. The libraries can be found in the [~/Libraries](#) directory.

APM32F0xx MCU include following library:

- Libraries folder
 - * APM32F0xx_StdPeriphDriver
 - * CMSIS
 - * Device
 - * TSC_Device_Lib
 - * USB_Device_Lib

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 APM32F072 EVAL include following:

- Middlewares folder

7 About Package

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

The middlewares used by APM32F072 EVAL include following:

- Package folder
 - * Geehy.APM32F0xx_DFP.1.0.7.pack

8 Revision History

Table 1 File Revision History

Date	Rev	Description
2022.08.26	1.0	First Release version of APM32F072 EVAL 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