

User Manual

APM32F107VC EVAL Board

Version: V1.0

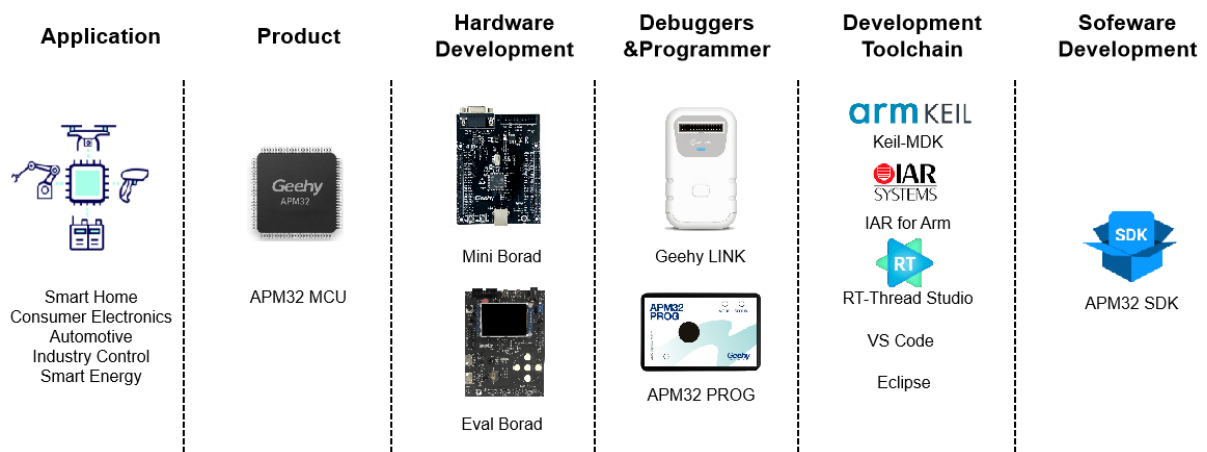
1 Introduction

This User Manual describes the functions, on-board resources and supporting SDK of APM32F107VC EVAL Board. The SDK and related data mentioned in the document can be obtained from the official website of Geehy (www.geehy.com).

1.1 APM32 Ecosystem

The APM32 ecosystem includes product application solution, hardware development board, download simulation tool, development tool chain and SDK. Moreover, the development tool chain is suitable for many development tools, such as Keil-MDK, IAR for Arm, Eclipse, etc., and all of them are equipped with relevant engineering in the SDK to meet the needs of different users in different platforms.

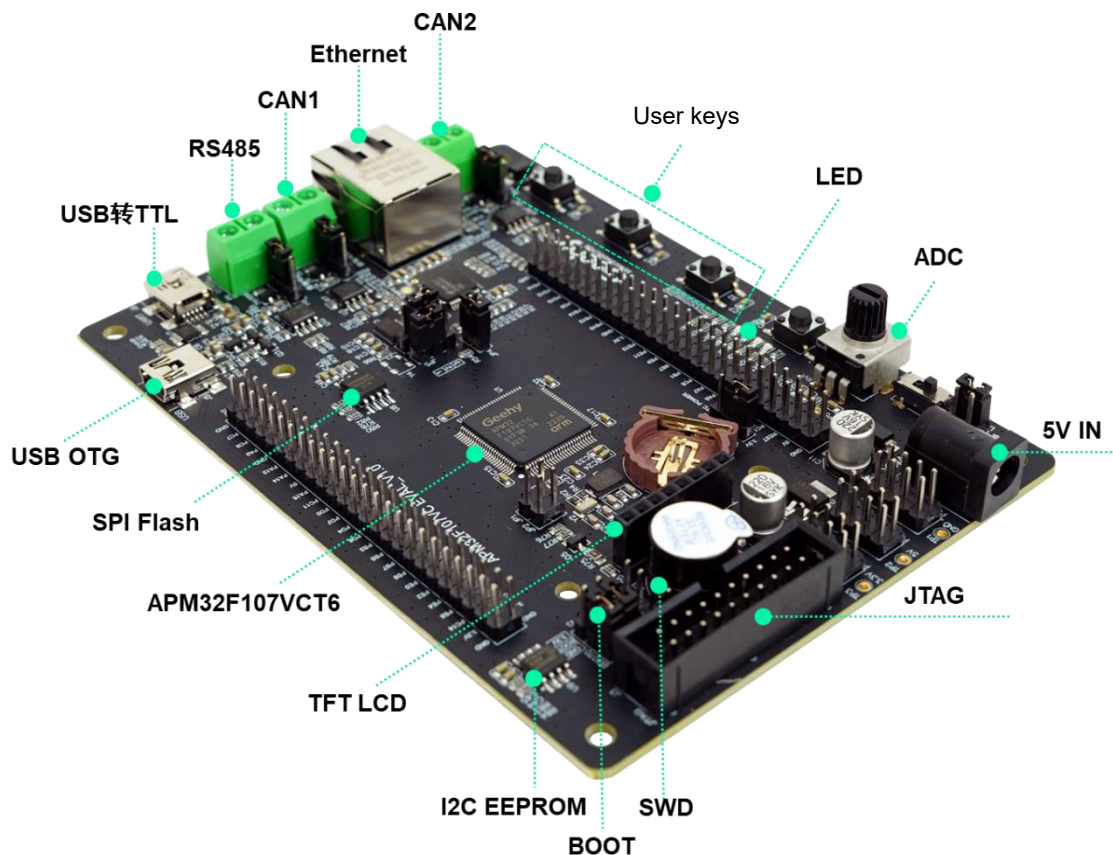
Figure 1 APM32 Ecosystem



1.2 Evaluation board

The APM32F107VC EVAL Board is a complete demonstration and development platform for the interconnected APM32F107/F105 series MCU. It carries an APM32F107VCT6 MCU chip, which is based on Arm® Cortex®-M3 core, operating frequency 96MHz, Flash 256KB, SRAM 64KB. Based on low development cost and low power consumption, this evaluation board can provide excellent computing performance and advanced system interrupt response. At the same time, it is equipped with EVAL SDK, which can help developers effectively evaluate the performance of APM32F107VCT6 chip or related development applications.

Figure 2 APM32F107VC EVAL Board



Contents

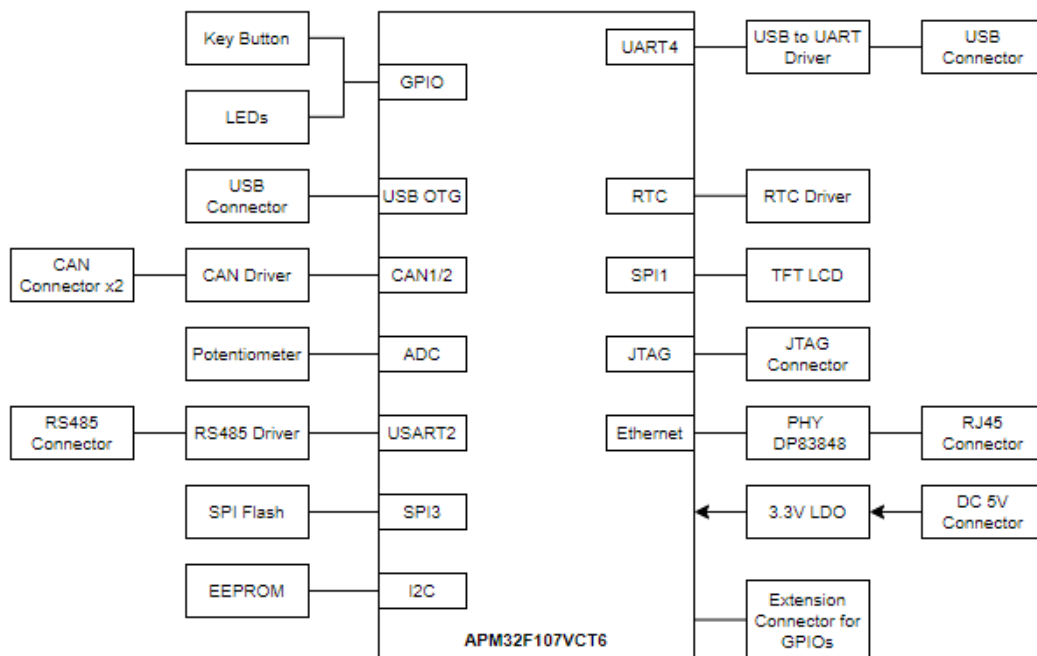
1	Introduction.....	1
1.1	APM32 Ecosystem	1
1.2	Evaluation board.....	2
2	Function overview	4
2.1	Power supply	5
2.2	Battery interface	5
2.3	Clock.....	5
2.4	Reset control	5
2.5	Simulation and download interface	5
2.6	LCD screen.....	6
2.7	LED lights	6
2.8	Keys.....	6
2.9	I2C EEPROM	6
2.10	SPI Flash	6
2.11	RS485 interface.....	6
2.12	CAN interface	6
2.13	Ethernet interface	6
3	SDK Overview	7
3.1	On-board driver	8
3.2	Library file	8
3.3	Middleware	8
3.4	IDE support.....	8
3.5	Routine	8
4	References	18
5	Revision History	19

2 Function overview

The APM32F107VC EVAL Board mainly includes the following functional peripherals:

1. TFT LCD
2. EEPROM
3. SPI FLASH
4. Potentiometer
5. RS485
6. Button x 3
7. LED x 3
8. RTC
9. CAN x 2
10. USB OTG
11. Ethernet

Figure 3 APM32F107VC EVAL Board Overview

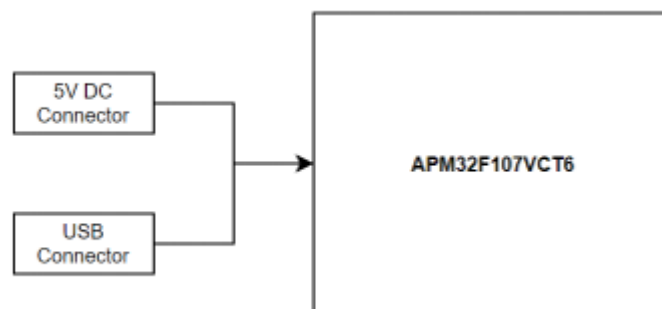


2.1 Power supply

The APM32F107VC EVAL Board can be powered by an external 5V DC power supply, or by selecting USB-to-serial port MiniUSB interface through the jumper cap. Other required voltages are provided by the on-board voltage regulator. In addition, the JTAG interface can also supply power to the evaluation board, but it can only supply 3.3V voltage to the evaluation board.

Besides, rich 5V/3.3V/GND pin headers are reserved on the board for users to conduct extended experiment.

Figure 4 Power Supply of APM32F107VC EVAL Board



2.2 Battery interface

On-board CR1220 standard battery interface, used as backup power supply of RTC.

2.3 Clock

APM32F107VC EVAL Board carries two external clocks:

- Y2, 32.768KHz clock is used for RTC or other required peripherals.
- Y1, 25MHz clock is used for APM32F107VCT6 MCU. If the internal RC clock of the chip is used, the clock can be removed or the HSE can be disabled

2.4 Reset control

Provide two reset controls, which can be generated by hardware or software:

- RESET key; press it to send a reset signal.
- JTAG reset signal.

2.5 Simulation and download interface

- Standard 20-pin IDC JTAG connecting interface.
- Four-wire SWD connecting interface

2.6 **LCD screen**

APM32F107VC EVAL Board carries a SPI LCD screen interface. The supporting SDK provides the routine of 2.4-inch TFT LCD screen with 240 x 320 pixels and ILI9341 LCD driver chip.

2.7 **LED lights**

APM32F107VC EVAL Board carries 3 user LED lights, which can be used for display by users or for indication purpose during the experiment.

2.8 **Keys**

APM32F107VC EVAL Board carries 3 user keys, which can be used for LCD menu switching or other input purposes.

2.9 **I2C EEPROM**

APM32F107VC EVAL Board carries AT24C32 EEPROM chip, and is connected with this chip through I2C1 interface of MCU.

2.10 **SPI Flash**

APM32F107VC EVAL Board carries a SPI Flash chip W25Q16, provides 2MB of external storage space, and is connected to this chip through the SPI3 interface of MCU

2.11 **RS485 interface**

APM32F107VC EVAL Board is connected to the RS485 chip through the USART2 peripheral and provides a RS485 interface. It should be noted that the RS485 function requires 5V power supply to work normally.

2.12 **CAN interface**

APM32F107VC EVAL Board has two-way on-board CAN interfaces, which can be used by users for CAN communication. It should be noted that the CAN function module requires 5V power supply to work normally.

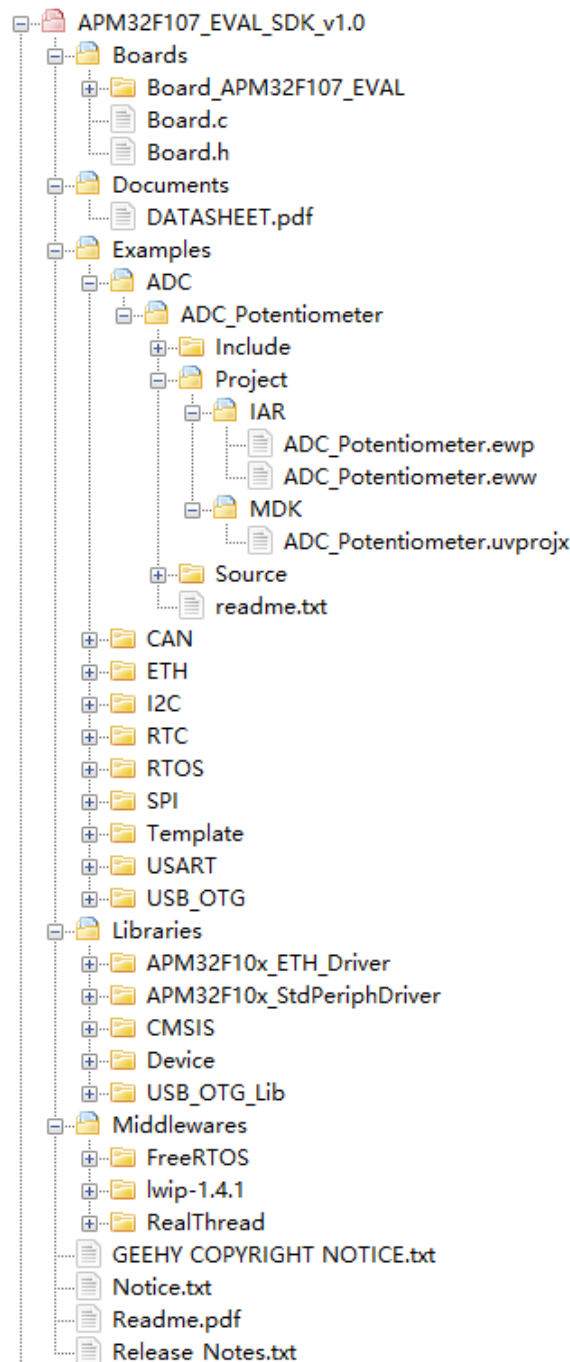
2.13 **Ethernet interface**

APM32F107VCT6 MCU is equipped with Ethernet peripheral module, which can communicate with external physical layer chip through MII/RMII interface. The physical layer chip on the APM32F107VC EVAL Board is DP83848. You can select MII or RMII as the communication interface with MCU through the jumper cap. The default jumper cap configuration of the development board is RMII interface.

3 SDK Overview

The SDK^[1] is provided in the form of compressed packages, including on-board driver packages, such as basic LED, Button and COM driver, I2C for EEPROM driver package, LCD screen driver package and W25Q16 Flash driver package, as well as multiple necessary libraries, such as APM32F10x standard library, Ethernet peripheral driver library and USB OTG peripheral driver library. It also includes many routines that are easy to reuse, such as EEPROM reading and writing, LCD multi-level menu, RS485 communication, CAN double-computer communication, and USB communication.

Figure 5 SDK Directory Architecture

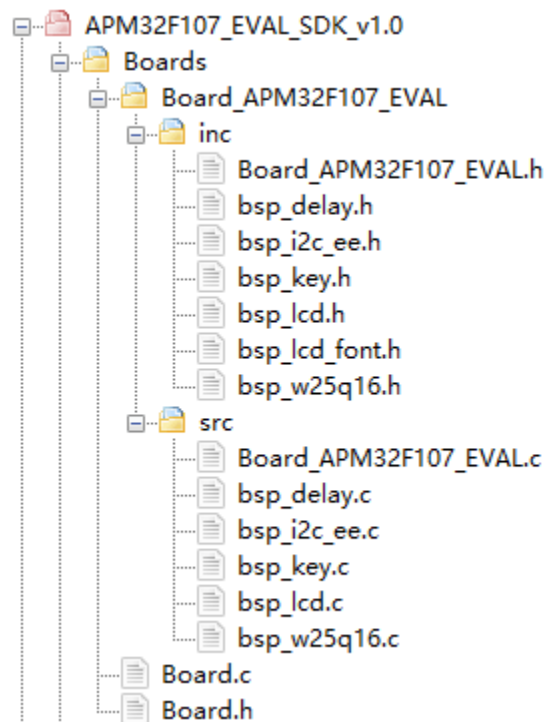


[1]. There is no package supported by IDE in the SDK of APM32xx_EVAL Board. You need to obtain it on the official website of Geehy www.geehy.com.

3.1 On-board driver

The on-board drivers contain basic LED, Button and COM drivers, I2C for EEPROM driver package, LCD screen driver package and W25Q16 Flash driver package.

Figure 6 On-board Drive



3.2 Library file

The library file contains APM32F0x standard peripheral driver library, Ethernet peripheral driver library and USB OTG peripheral driver library.

3.3 Middleware

The middleware used in the SDK package includes FreeRTOS, RT-Thread, and lwip-1.4.1.

3.4 IDE support

This SDK routine is equipped with the engineering of two development tools of Keil-MDK and IAR for Arm.

3.5 Routine

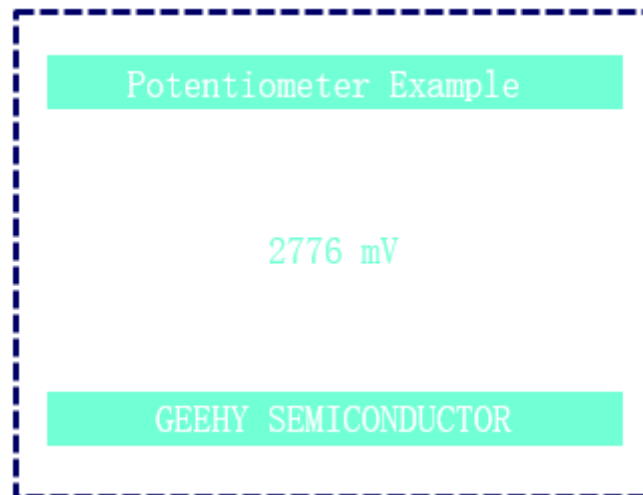
The routines provided in the SDK package involve all hardware modules of the APM32F107VC EVAL Board, such as LCD screen, CAN communication, RS485, Ethernet and other routines. Users can develop applications based on the routines in the

SDK package. The following briefly introduces each routine in the SDK package.

3.5.1 ADC

ADC part adopts a 10K potentiometer as the sampling source, which can be adjusted by the knob. The AD sampling channel is PC0 (channel 10). After downloading and running this routine, the voltage value sampled by ADC will be displayed on the LCD screen in real time.

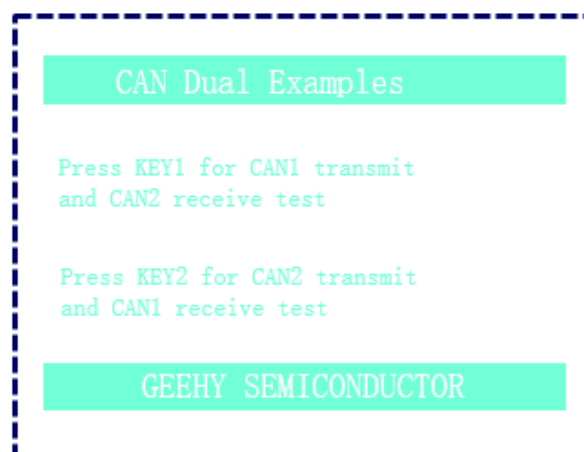
Figure 7ADC Menu



3.5.2 CAN Dual

The CAN Dual routine is two-way CAN peripherals of APM32F107 conducts communication and is used to show how to use the CAN module. During the experiment, the H/L wires of CAN1 and CAN2 need to be short-circuited, and it shall be ensured that 5V power supply is provided to the board. When KEY1 is pressed, CAN1 will send data to CAN2. After CAN2 successfully receives the data, it will display "Test OK" on the screen. Similarly, when KEY2 is pressed, CAN2 will send data to CAN1. After CAN1 successfully receives the data, it will display "Test OK" on the screen.

Figure 8 CAN Dual Menu



3.5.3 I2C EEPROM

EEPROM adopts AT24C02, the device address is 0xA0, and the addressing address is 16bit. The routine test process is to write and read 255 data to EEPROM and compare them. If the compared read and written data are equal, the screen will display "Test OK".

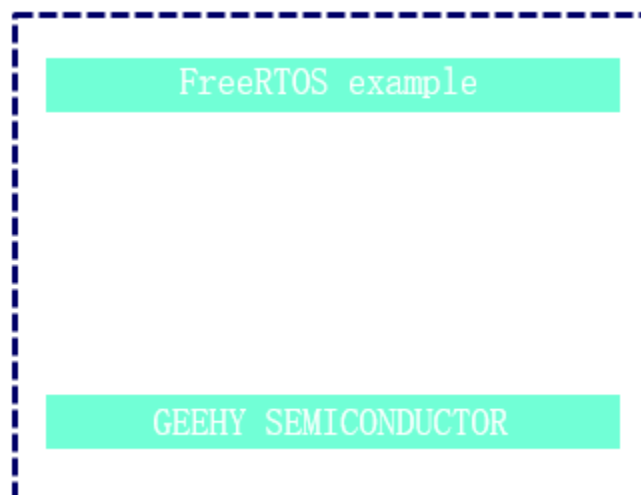
Figure 9 I2C EEPROM Menu



3.5.4 FreeRTOS

This routine provides a Demo running the FreeRTOS system. The routine creates two simple tasks: LED flashing and string printing through the serial port for users to judge whether the system is running normally.

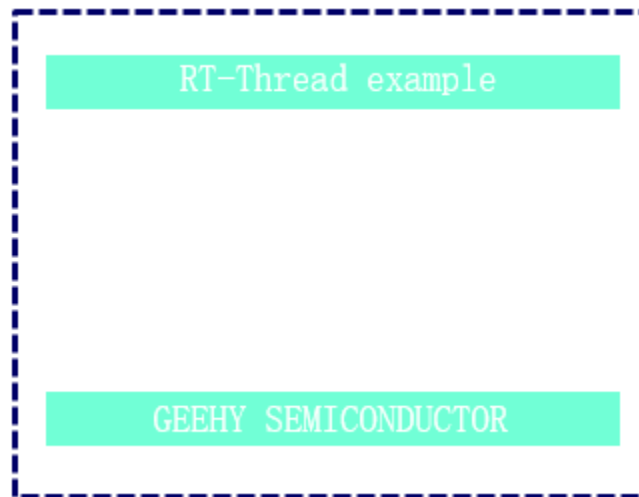
Figure 10 FreeRTOS Menu



3.5.5 RT-Thread

This routine provides a Demo running the RT-Thread system. The routine creates an LED flashing task for users to judge whether the system is running normally. In addition, the shell component of the RT-Thread system is migrated, and users can conduct command line interaction through the serial terminal.

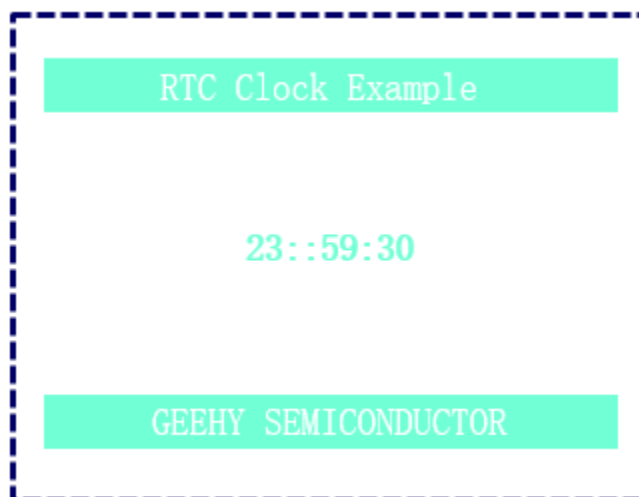
Figure 11 RT-Thread Menu



3.5.6 RTC Clock

RTC_Clock routine simulates a 24-hour clock and the time is displayed in real time on the screen.

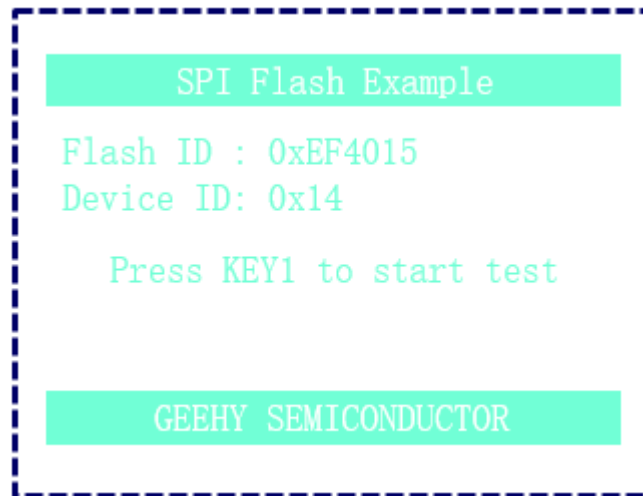
Figure 12 RTC Clock Menu



3.5.7 SPI Flash

This routine uses SPI interface to access the external flash chip W25Q16, with a size of 2MB. The test process is to write and read data to Flash and compare them. If the compared read and written data are equal, the screen will display "Test OK".

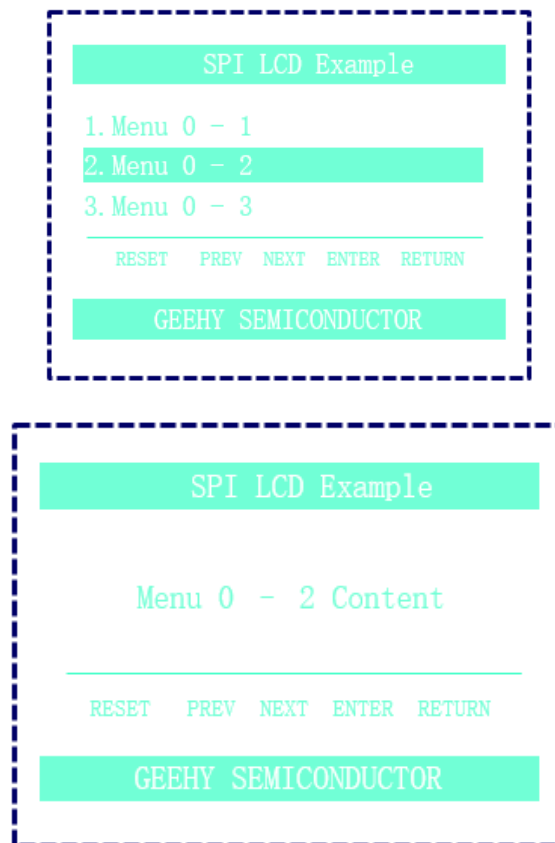
Figure 13 SPI Flash Menu



3.5.8 SPI LCD

This routine uses SPI peripheral interface to drive a 2.4- inch LCD screen to display a sample of multi-level menu.

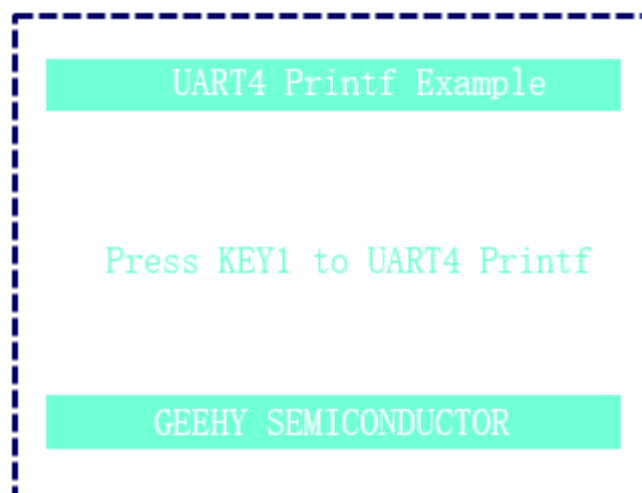
Figure 14 SPI LCD Menu



3.5.9 USART Printf

This routine shows how to send serial port data by polling. The test process is to press KEY1 and then UART4-to-USB interface to send the string "APM32F107 EVAL UART4 Printf Test.\r\n" to the upper computer.

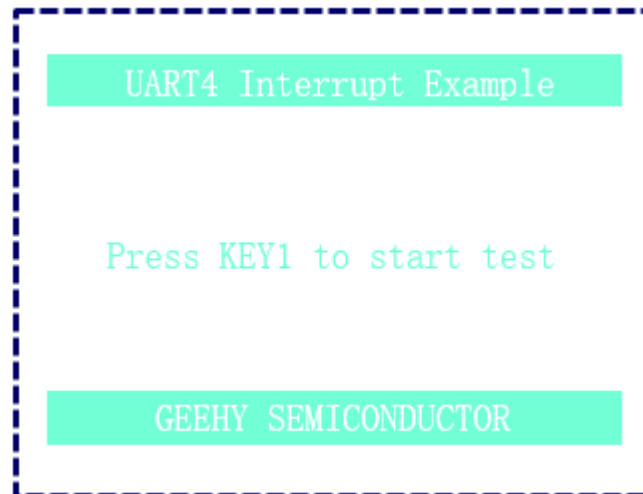
Figure 15 USART Printf Menu



3.5.10 USART Interrupt

This routine shows how to transmit and receive serial port data by interrupt. The test process is to send the string to the upper computer through the UART4-to-USB interface by interrupt; users can also send data to the development board through the upper computer, and then the development board will receive the data through an interrupt and return the same data to the upper computer.

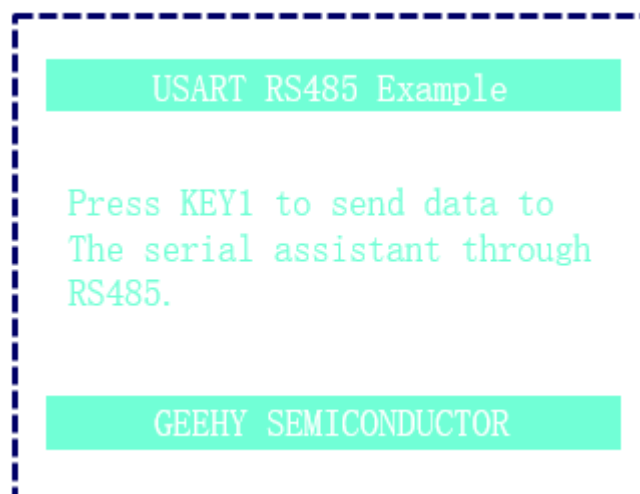
Figure 16USART Interrupt Menu



3.5.11 USART RS485

This routine shows the RS485 communication process. The test process is to send data to the upper computer through the on-board RS485 module; or the upper computer can actively send data to the development board, and the development board will return the same data to the upper computer after receiving the data. RS485 module is a USART2 peripheral connected to MCU.

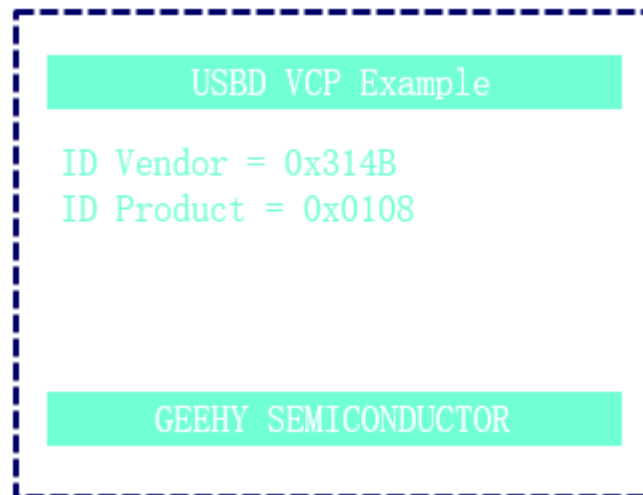
Figure 17USART RS485 Menu



3.5.12 USBD_VCP

This routine implements virtual serial port device through the USB OTG peripheral of APM32F107VCT6, and the evaluation board is configured as the Device. When the USB port of the evaluation board is connected to the PC terminal, you can see an additional COM port in the device manager. You can open this COM port through the serial port debugging tool, and then send data to the evaluation board through this COM port. After receiving the data, the evaluation board will return the same data to the serial port debugging tool.

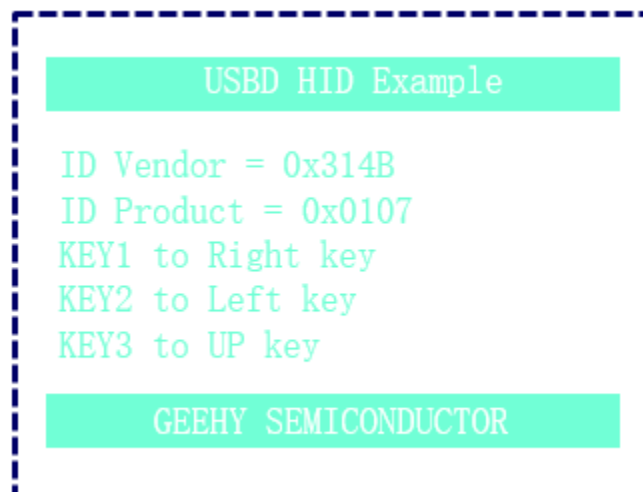
Figure 18 USB D VCP Menu



3.5.13 USB HID Mouse

This routine simulates the implementation of a USB mouse through the USB OTG peripheral of APM32F107VCT6, and the evaluation board is configured as the Device. When the USB interface of the evaluation board is connected to the PC terminal, you can see an additional mouse device in the device manager. Users can operate the KEY1 - KEY3 keys on the board to simulate movement function of the mouse.

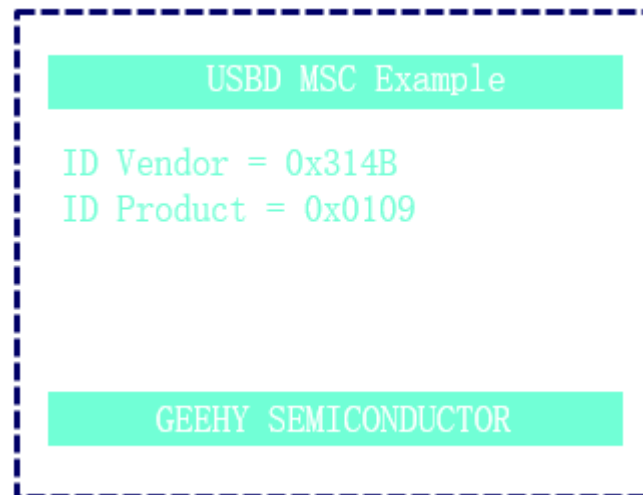
Figure 19 USB HID Menu



3.5.14 USB MSC Disk

This routine simulates the implementation of a USB device through the USB OTG peripheral of APM32F107VCT6, and the evaluation board is configured as the Device. When the USB interface of the evaluation board is connected to the PC terminal, you can see an additional disk device in the device manager. This routine uses internal RAM of APM32F107 to simulate a U disk device.

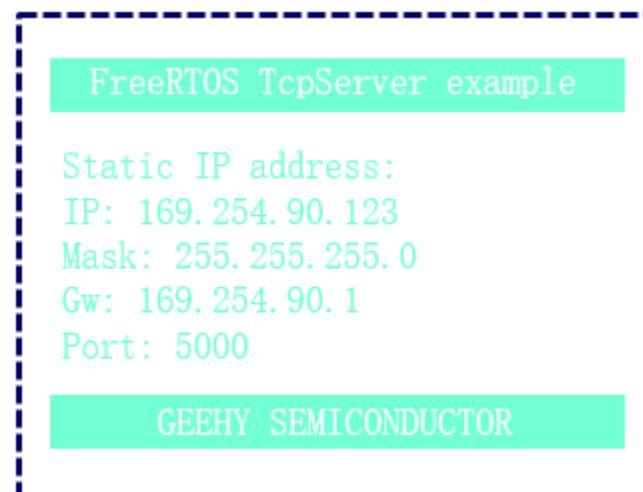
Figure 20 USB MSC Menu



3.5.15 ETH FreeRTOS TCP Server

This routine is an Ethernet communication routine running lwip1.4.1 network protocol stack under FreeRTOS system. The evaluation board runs as a Tcp Server and has been waiting for the connection of the client. The static IP configured by the routine is 169.254.90.123, and the server port is 5000. The tools such as network debugging assistant can be used as a client to connect the evaluation board. After successful connection, you they can send and receive network data from each other, and the received data can be printed through the serial port.

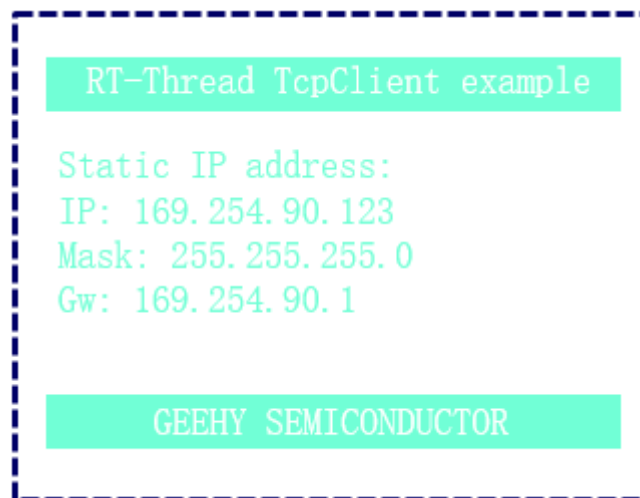
Figure 21FreeRTOS TcpServer Menu



3.5.16 ETH_RT-Thread_TcpClient

This routine is an Ethernet communication routine running lwip1.4.1 network protocol stack under RT-Thread system. The evaluation board runs as a Tcp client and has been waiting for the connection of the server. The static IP configured by the routine is 169.254.90.123. The tools such as network debugging assistant can be used as a server to connect the evaluation board. After successful connection, you they can send and receive network data from each other, and the received data can be printed through the serial port.

Figure 22RT-Thread TcpClient Menu



4 References

For chip specifications and peripheral details, see *APM32F107 105xx User Manual*, *APM32F107 105xx Data Manual*, and *APM32F107VC EVAL Schematic Diagram*. For more technical support, please visit the official website of Geehy: www.geehy.com.

5 Revision History

Table1 Document Revision History

Date	Revision	Changes
2023.1.18	1.0	New

Statement

This manual is formulated and published by Zhuhai Geehy Semiconductor Co., Ltd. (hereinafter referred to as "Geehy"). The contents in this manual are protected by laws and regulations of trademark, copyright and software copyright. Geehy reserves the right to correct and modify this manual at any time. Please read this manual carefully before using the product. Once you use the product, it means that you (hereinafter referred to as the "users") have known and accepted all the contents of this manual. Users shall use the product in accordance with relevant laws and regulations and the requirements of this manual.

1. Ownership of rights

This manual can only be used in combination with chip products and software products of corresponding models 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 manual for any reason or in any form.

The "Geehy" or "Geehy" words or graphics with "®" or "TM" in this manual 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 manual.

Geehy shall not be deemed to grant the license or right of any intellectual property to users explicitly or implicitly due to the sale and distribution of Geehy products and this manual.

If any third party's products, services or intellectual property are involved in this manual, 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 of Geehy.

3. Version update

Users can obtain the latest manual of the corresponding products when ordering Geehy products.

If the contents in this manual are inconsistent with Geehy products, the agreement in Geehy sales order or sales contract shall prevail.

4. Information reliability

The relevant data in this manual 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 manual. The relevant data in this manual 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. Compliance requirements

Users shall abide by all applicable local laws and regulations when using this manual 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 product suppliers, Geehy, Geehy distributors and users. Users (on behalf of 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

This manual is provided by Geehy "as is". To the extent permitted by applicable laws, Geehy does not provide any form of express or implied warranty, including without limitation the warranty of product merchantability and applicability of specific purposes.

Geehy will bear no responsibility for any disputes arising from the subsequent design and use of Geehy products by users.

7. Limitation of liability

In any case, unless required by applicable laws or agreed in writing, Geehy and/or any third party providing this manual "as is" shall not be liable for damages, including any general damages, special direct, indirect or collateral damages arising from the use or no use of the information in this manual (including without limitation data loss or inaccuracy, or losses suffered by users or third parties).

8. Scope of application

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

©2023 Zhuhai Geehy Semiconductor Co., Ltd. - All Rights Reserved