Design and implementation of FATFS data exchange with multiple storage media based on single-chip microcomputer

Zerui Chen
First Research Office, Kunming Shipborne Equipment Research and Test Center, KunMing, YunNan, 650051, China
xiaoruibest@yeah.net

Abstract. At present, a single-chip microcomputer without an operating system is used for data exchange with single storage media and the transmission speed is limited. In this paper, FATFS module is transplanted on a single-chip microcomputer based on an ARM Cortex-M7 micro-controller. The data acquisition and storage are carried out in parallel with multiple storage media such as SD, the NAND FLASH and the U disk on a high-performance STM32H7 platform. The read-write speed is tested finally. The results show that the file management function can be completed well and the operation is stable.

1. Introduction

With the rapid development of the information age, the requirements for data exchange have increased gradually. The selection of appropriate memory is the first step to complete the efficient transmission and fast storage of data. Embedded system has been widely used in industrial testing and control because of its small size and low cost, so it is necessary to manage data files effectively by transplanting open FATFS file management module to the single-chip microcomputer. In this paper, based on high-performance STM32H7 development board platform, the FATFS migration with small occupied hardware resources is completed without the operating system and the module functions are used to find, read, write and delete the data of NAND FLASH, SD, U disk and other storage media. The test results show that the data exchange between the microcomputer and internal storage and other peripherals can be met by testing and comparing the reading and writing speed of the three devices.

2. Chip selection and introduction

Chip selection includes MCU and NAND FLASH

2.1. MCU

The MCU is a 32-bit microprocessor based on ARM Cortex-M7 produced by ST company. The main frequency of STM32H743IIT6 is 400MHz with six-stage pipelines, dual-precision hardware floating-point computing unit (DPFPU), DSP instruction integrated with instruction and data Cathe, 1060KB SRAM, 2048KB FLASH, 9 serial ports and 2 USBs for HOST / SLAVE, 2 CANs, 1 RTC (with calendar function), 2 SDMMC interfaces, 1 FMC interface. STM32 also has outstanding on-chip peripherals, such as USB, USART, SPI, I2C, GPIO, PWM and so on[1]. The MCU is placed on the STM32H743 core board and is supplied by 3.3 V. The single-chip microcomputer can be used in motor drive and application control, medical electronics, industry, Applications such as PLC, mobile applications and the Internet of things, wearable equipment such as smartwatches[2].
2.2. NAND FLASH
Model: H27U4G8F2 with 512M bytes. The chip is also on the FMC interface of the STM32H743. With this chip, the storage space of the STM32 is greatly expanded and the 512M data storage can be realized.

3. The introduction of FATFS

![Application Layer](image1)

Figure 1. FATFS module and structure

At present, the used file systems include FAT32, NTFS and EXT2 in Linux system. FATFS is a completely free and open-source FAT file system module which does not depend on I/O layer. It is completely written in standard C language, so it has independence and portability in the hardware platform and can be ported to 8051, ARM and other series of single-chip microcomputer. There are separate buffers that can be used for multiple read-write texts. The structure of the FATFS can be displayed in Figure 1. Application Layer uses a series of application interface functions. FATFS Module provides the FAT file read/write a protocol and it does not need to be modified. Storage medium and RTC need to be modified. FATFS is designed to read and write in a very special way, which makes it high reading and writing efficiency. Sector local data is transmitted through the file I/O buffer, but I/O Rush zone is not used for the sector-aligned block of data.

4. The hardware resource description

As shown in Figure 2, the data exchange system includes main processor MCU, NAND FLASH, U disk, SD card, key, LCD and three switching data ports including CAN, RS232 and EMAC, which are connected to PC. The general workflow of the system is as follows: the single-chip microcomputer system waits for the external interrupt of SD, U disk, FLASH after initialization. When MCU captures the external interrupt signal or the corresponding state value, PC operates on the data of three kinds of numbered and interrupt priority storage medium through CAN, RS232, EMAC interfaces, which can read the data in contents and single file, establish the file and write the data, modify the file information and so on. The keyboard is used as the input device and the LCD screen is used as the output device that can display the reading and writing information of each disk.

![Data exchange system block diagram](image2)
5. The software design module

5.1 Transplantation and implementation of FATFS

There are six parts in FATFS:
- Variable.c: Define global variables and get disk information.
- Test.c: Encapsulates FATFS functions for reading and writing performance.
- Ff.c: FAT file reading and writing protocols.
- Ffsystem.c: Get the time and allocate and release memory.
- Ffunicode: Unicode.
- Diskio.c: The underlying driver interface function.

Then six interface functions are designed in diskio.c:
- Disk_initialize: Initialize disk drive.
- Disk_status: Get drive status including initialization, device connection, writing protection, disk normal drive.
- Disk_read: Read one or more sector data in the disk.
- Disk_write: Write one or more sector data in the disk.
- Disk_ioctl: Complete the function except reading and writing.
- Get_fattime: Get the current time.

In diskio.c, the symbols of SD, NAND FLASH, USB disk are defined as 0, 1, 2, corresponding to disk_reading/writing, and then determining volume number which need be operated. Finally, the appropriate read and write functions are selected to operate on the corresponding storage medium, that is: #define SD_CARD 0  #define EX_FLASH 1  #define EX_USB 2

5.2 The creation of main function

Mounting workspace for SD, NAND FLASH and USB after initialization. Then, entering the loop and testing by serial port with Usmart software.

6. The analysis and optimization of test results

The hardware interface is a serial port (RS232), the software is XCOM V2.0. Some API application functions which are shown in figure 3 can be used to finish the operation of directory and file data. The operations of the directory include creating, deleting and renaming files in the directory. The operations of files include displaying a list of disk files, opening the specified files in the directory, reading the data in the file and writing data to the files.

![Figure 3. API application interface functions](image)

| Storage Medium | Read      | Write     |
|----------------|-----------|-----------|
| NAND FLASH     | 0.72MB/s  | 0.60MB/s  |
| SD Card        | 0.80MB/s  | 0.72MB/s  |

The NAND FLASH and SD card are used to test. From the Table 1, the performance of SD card is optimal under this FATFS because its reading and writing rates are higher than NAND FLASH.
However, its transmission rate is still limited. In order to improve the efficiency of data exchange, EMMC chip which is BGA package can be used and it can achieve the maximum rate of 330MB/s for reading and 200MB/s for writing.

7. Conclusion and prospect

This paper designs a file data exchange system under embedded multi-storage medium based on single-chip microcomputer, which can finish the operations of directory and files for the data in SD, NAND FLASH and U disk. The results show that the rate in SD is higher and it is 0.8MB/s. However, the rate is limited. In future, EMMC chip of 8-bit with faster transmission efficiency. The system realizes the functions of data access without PC host and the program is programmed in C language, and it is easy to transplant to other processors.

References

[1] Zhang, T., Zuo, J.P., Ma, H.L. (2010) Transplantation of FatFs on 32-bit microcontroller STM32. Electronic Technique, 47: 25-27.
[2] Chen, C. (2018) ARM MCU Development Scheme based the STM32H743 high performance. http://www.elecfans.com/d/635707.html
[3] Yang, J. (2008) Realization of Embedded USB Host System Based on MCU. The microcomputer information, 14:93-94.
[4] Hong, Y.W., Wang, B.M., Xie, C.Y. (2008) FatFs Module that is easy to migrate and use. Application of MCU and Embedded System, 05:29-31.
[5] Gao, Z.Q. (2017) The Transplantation and Application of FATFS in Embedded Operating System FressRTOS. Modern Electronic technology, 40:65-67.