Methods of increasing the reliability of external memory of digital substation bay controllers

D Lantsev*, V Frolov

Peter the Great St. Petersburg Polytechnic University, 195251 St. Petersburg, Russia

* E-mail: mank-qwe-9874@mail.ru

Abstract. The paper studies main technical requirements for digital substation bay controllers; substantiates the high reliability requirements for file systems running on Flash drives of such devices. An experimental test analysis of the most popular file systems for microcontrollers with limited resources with start-up systems on the STM32F407IGT6 controller have been made. The algorithm of Flash translation layer (FTL) is offered, and also its check on simulation model of the Flash drive and the experimental equipment is carried out. The results of comparing the characteristics of file systems and FTL algorithm are presented.

1. Introduction

Electronic equipment at digital traction substations, according to the standard structure [1, 2], can be divided into three levels: lower, middle, upper. Currently, in Russia, among the bay controllers performing the role of middle-level, the most common controllers are based on microcontrollers with the core ARM Cortex-M4 or DSC controllers Texas Instruments [3]. Their use is economically justified due to the fact that they can confidently support the real-time operating system, have a sufficiently large clock speed and developed periphery. Microcontrollers are not expensive, their power consumption much lower than the older arm A9 controllers used at the upper level, can be provided with natural cooling. Have excellent debugging tools and advanced technical support. They are assigned the tasks of horizontal communication and the survey of the lower level, as well as communication with the upper level. A typical connection controller connection diagram is presented in Figure 1.

At the middle level of the digital substation system, it is necessary to record emergency waveforms for subsequent download and analysis of the accident development in accordance with IEC 60255-24 Ed.2, for example, in the publication [4-5], oscillograms are obtained as a result of modeling the operation of a telemechanical relay.
Connection controllers used in modern digital substations must support Modbus TCP/RTU, IEC 60870-104, IEC 61850 protocols. IEC 60870-104 Protocol is widely used for the transmission of telemetry information in the power system, its application is discussed in detail in [6]. Connection controllers must also support an HTTP server to configure and configure the settings. FTP server support is required for remote configuration and downloading of alarm waveforms. In addition, the actual problem is the remote update of software controllers. IEC 61850 requires a file describing the capabilities of the device in ICD format, with proper programming, the device reads the configuration from it, which is a more flexible solution than static compilation of a given configuration, since during operation it is possible to change the peripheral modules, and also allows you to create a single firmware for the entire line of devices. [7] describes an abstract model of the system in accordance with IEC 61850, as well as both GOOSE and MMS communication protocols. The intention of this document is to focus on security monitoring, and it addresses Protocol formats in detail. More detailed technical reference for the development of relay protection devices can be found in standard organization of PAO "ROSSETI" 34.01-4.1-007-2018 [8].

On the bay controller attach the traction substation KPL600f is now applied AT45DB161E SPI Flash 2MB. In connection with the modernization of the software in order to support the IEC 61850 Protocol, it was proposed to increase the amount of non-volatile memory. For this purpose, the prototype was tested chip NOR Flash S29GL01GS11TF102.

2. Methods

Initially, the upgraded controller used the FAT32 file system implemented by the FATFS library [9], this system is also popular with other manufacturers of equipment for RZA.

This is a time-tested solution, but it has serious drawbacks. These include the lack of a mechanism for uniform wear of the Flash drive. Erase-write operations are designed for only 100,000 operations after which the drive must be replaced. FAT32 does not protect data when power is lost while writing to the drive. Each time a write operation is carried out erasing the boot sector. There is a developed set of API functions resembling POSIX [10], and the library is well documented. However, running it directly on S29GL01GS11TF102 is extremely inefficient, since the size of the erasable sector is 128 KB, and the file system for a dozen files does not make sense.

To address the shortcomings of FATFS and the increased reliability requirements of the device due to the fact that the configuration file must be downloaded from the drive, it was decided to test the main file systems available for use.

Options of small file systems with systems of uniform wear of the drive such as LittleFS [11] used in the mbed OS operating system developed by developers of the ARM kernel and the SPIFFS system [12] developed by Peter Andersson fan were considered and tested on the test controller. In the paper [13] discusses a file system for microcontrollers that meet the requirements of the controllers. But it
can not be used in a real device without the consent of the author. Other similar systems file systems are either too demanding to the resources of the microcontroller such as JFFS or have a proprietary nature of the license.

Consider a system LittleFS, it has a low footprint microcontroller. The code does not use recursive calls and works without dynamic memory allocation. "LittleFS" the amount of consumed memory always remains constant. The availability of software for wear leveling (wear leveling), to minimize the reuse of media blocks. Resistance to power failures. This feature is normally considering the termination of the power supply of the media, and uses the mechanism of "copy-on-write (COW)", in which the data is not overwritten, and saved to a new location. The system supports a full set of POSIX functions for working with files and directories.

To test the system, an experimental firmware for STM32F4 was written, with a basic set of shell commands for file system operations. It should be noted that to start system on S29GL01GS11TF102 it was not possible, on SPI Flash the system worked not steadily at record of files from 300 KB. Also, the system does not support timestamps, and the version with support for timestamps is under development. As the system did not meet the requirements above, it was decided to move on to another system.

The SPIFS file system was designed specifically for embedded systems. In its development, the following considerations were taken into account: the system only for microprocessors, static buffers under the control structures, which prevents defragmentation of the heap, the large size of the erasable sector, the system of a uniform drive, the units recorded in bits can become zeros only when the sector is erased.

As a drawback, SPIFFS does not support directories it is a flat file system. Unlike traditional file systems, you can specify slashes ("/") in SPIFFS file names, so functions that work with folder lists – for example, openDir("/comtrade") simply filter file names to find those that begin with the requested prefix. The maximum size for a file name is 32 characters. Moreover, the symbol "\0" is reserved for string termination, so in the end we are left with 31 character.

Thus, spiffs file names should be short, which implies another limitation – file names should not be too multi-level. Thus, this file system, although launched on NOR Flash S29GL01GS11TF102, does not fully meet the requirements of the connection controller level and is more suitable for use only at the lower level.

Since both the promising file systems and the FAT system do not fully meet the file system requirements on the attach controller, it was decided to develop a Flash Layer Translation Level (FTL) algorithm and implement it between the SPI Flash driver and the FATFS library.

In articles [14-15] methods and algorithms of the organization of a layer between classical file systems and the driver of SPI or NOR Flash drive are considered. In [16] the method of static wear leveling, which is called "Level wear leveling" (RbWL), is proposed. The basic idea of RbWL is to perform static wear leveling at minimum levels because the net transfer of raw data under static wear leveling results in significant overhead in the NAND flash system. RbWL adjusts the execution frequency according to a threshold value that reflects the difference in lifetime and the total lifetime of the NAND flash system. The results of the evaluation show that RbWL increases the life of NAND flash systems by 52%, and reduces the overhead of wear leveling from 8% to 42% and from 13% to 51% in terms of the number of erase operations and the number of page transfers, respectively, compared to other algorithms.

Different granularity of the display in FTL will change the performance of read-write and the size of the mapping table. For cases in which high-capacity storage should support high performance, based on previous mapping algorithms, are considered in [17]. This document develops a mapping method based on the granularity of the page grouping and an SSD system architecture based on the parallelism of the flash chip array. In addition, taking into account the peculiarities of the display method, an appropriate wear-leveling algorithm and a garbage collection algorithm are implemented.
Taking into account the above, the algorithm FTL implemented in the language C. The proposed algorithm provides a mechanism for monitoring bad sectors, it also has a mechanism for protecting data from sudden power loss, garbage collection mechanism, supports any partition into the size of sectors and blocks, respectively, the algorithm is applicable to almost any NOR Flash. The principle of operation of the algorithm is to use an intermediate level of translation of logical fat sectors in the physical sectors of the Flash drive. When running, FTL reads the configuration from the structure, determines the size of the sector and block. After that, each sector is scanned at the beginning of a separate block, if the "MagicWord" label is not found there, the block is formatted. After that, it checks whether the recording was successful. At the beginning of each block is the control FTL sector, where the broadcast address is stored, the sector erase counter since its initial formatting. FTL provides the function of displaying the statistics of erasures by sectors, the functional scheme FTL is shown in Figure 2.

3. Results and Discussion
A model was developed in the C language launched on a PC, as operations with FLASH, it performs operations with dynamically allocated RAM, at the start of the program, the flash image is read flash.bin file, for later recording, erasing, and reading are simulated in the RAM with the features of Flash memory. When the program ends, the exit command displays the sector usage statistics, the program terminates and stores the dump in flash.bin. This image is then available for research in any hex editor. Using the erase statistics and FATFS API functions, we will conduct a synthetic test to simulate typical actions with the file system. Figure 3 shows the result of FTL operation after 400 MB Erasure recording. This FTL was correctly launched on the prototype device with the chip S29GL01GS11TF102 and microcontroller STM32F407IGT6.
We will conduct a comparative analysis of the results of testing file systems, display it in Table 1.

FATFS takes up less of all the flash memory of the microcontroller programs, the proposed algorithm takes up slightly more space than working with different sizes of SPIFFS sectors. SPIFFS is a flat system with a 31-character name limit. Other systems do not have such restrictions. All systems except the original FAT support the mechanism of uniform wear of the external drive. All considered to improve the system allow you to keep the system's ability to mount after a power failure during recording. System LittleFS and SPIFFS poorly documented despite the rather broad use in software on microprocessor element base. On a Flash drive with a parallel interface and non-standard sector sizes, only SPIFFS and the algorithm with FTL started correctly. The test for writing a large file of 500 KB did not pass LittleFS after such a recording, the system lost its contents, but retained the ability to mount, when writing small files, such problems were not observed.

Table 1. Comparison of file systems by main features.

| File system | ROM, kB | Directories | Wear-leveling | Energy lost | Documents | FSMC NOR | Stability |
|-------------|---------|-------------|---------------|-------------|-----------|----------|-----------|
| FATFS       | 8       | +           | -             | -           | +         | -        | +         |
| LittleFS    | 10      | +           | +             | +           | -         | -        | -         |
| SPIFFS      | 14      | -           | +             | +           | -         | +        | +         |
| FATFS + FTL | 15      | +           | +             | +           | +         | +        | +         |

4. Summary
As a result of consideration of all the above options FATFS with FTL surely surpasses the rest of the system. As a disadvantage, you can specify the requirement of 15 KB of permanent program memory and 2 KB of RAM allocated on the heap once during FTL initialization. The use of the system will allow to solve the shortcomings of FATFS, to increase the reliability of the file system controller connection. Reduce the number of failures and increase the warranty period of the equipment.

References
[1] Shen B, Yuan Y, Xie L, Zeng X, Liu Y and Luo H 2017 Reasch on IED configurator automatic
modeling based on IEC 61850 IEEE Conference on Energy Internet and Energy System Integration (EI2) Beijing 1–5

[2] Zamyatin E O and Shklyarskiy Y E 2016 Concept for electric power quality indicators evaluation and monitoring stationary intellectual system development, Int. J. Appl. Eng. Res. 11(6) 4270–4

[3] Frolov V, Bystrov A and Neelov A 2017 Imitating model of a microprocessor trip unit of a circuit breaker Proceedings of the 2017 IEEE Russia Section Young Researchers in Electrical and Electronic Engineering Conference EIConRus 838–40

[4] Othman M, Ishak A and Wahab N 2014 Modeling and simulation of the industrial numerical distance relay aimed at knowledge discovery in resident event reporting Malaysia Journal Citation Reports 90 660–86

[5] Frolov V, Neelov A, Zhiligotov R and Bystrov A 2018 Identification of the protection parameters of the local electrical network taking into account the detuning of the inrush current Proceedings of the 2018 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering, EIConRus 2018 626–8

[6] Belousov A, Koshlich Y and Bashkatov A 2017 Investigation of communication standard IEC 60870 used to create substations remote control systems IEEE 58th International Scientific Conference on Power and Electrical Engineering of Riga Technical University (RTUCON) 1–4

[7] Matousek P 2018 Description of IEC61850 Technical Report no. FIT-TR-2018-01 (Faculty of Information Technology Brno University of Technology)

[8] PAO “Rosseti” 2018 Technical requirements for automated monitoring of relay protection and automation devices, including those working according to IEC 61850 standard. Organization standard

[9] Open Source 2019 Fatfs - generic fat file system module http://elm-chan.org/fsw/ff/00index_e.html

[10] Godunov A and Soldatov V 2015 Scientific Research Institute for System Analysis of the Russian Academy of Sciences (SRISA), Moscow №6, pp 3-17 SOFTWARE ENGINEERING New Technologies Publishing House (Moscow)

[11] Open Source LITTLEFS 2019 https://github.com/ARMmbed/littlefs

[12] Open Source SPIFFS 2019 https://github.com/pellepl/spiffs

[13] Dudyak E 2015 Lightweight file system for in-line devices oriented to operation with NOR flash memory Nauchno-tekhnikheskiy sbornik "Tekhnika radiosvyazi" 95–101

[14] Zhao M, Jiang L, Zhang Y and Xue C J 2014 SLC-enabled wear leveling for MLC PCM considering process variation IEEE Proceedings of the 51st Annual Design Automation Conference 1–6

[15] Dubrovin A 2018 Advantages of a hybrid solution driver controller, analized using the wear leveling algorithm Works of scientific research institute of system researches of the Russian Institute for System Studies of the Russian Academy of Sciences 8 61–70

[16] Hwang S, Ho S and Wook J 2018 Recency-based static wear leveling for lifetime extension and overhead reduction in NAND flash memory systems Ieice Transactions on information and systems E101D(10) 2518–22

[17] Wang L, Zhu M, Yang C, Qiu X and Jiao Y 2016 Research on the flash translation layer based on grouping pages 2016 Sixth International Conference on Instrumentation & Measurement, Computer, Communication and Control IMCCC 121–6