Tool for Management of Flash Memory Wear-Leveling on Embedded System Device

Martin Havlík

Supervisor: Ing. Václav Šimek

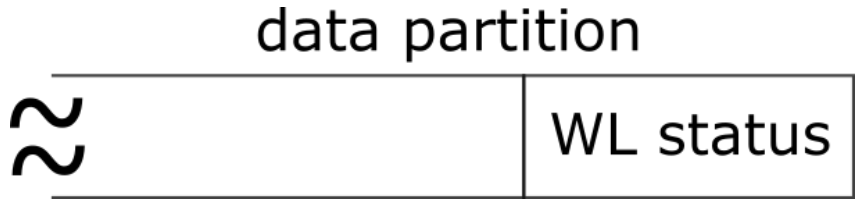
January 31, 2023

Goal

- for ESP-IDF WL implementation
- embedded BE + Python FE
- state, statistics, erase spread, life expectancy

Assignment objectives

1. Study the fundamental **principles of the wear-leveling technique**. Survey the existing approaches for various types of modern data storage devices.
2. Based on the evidence collected in point 1) of the assignment, make a comparison to the **wear-leveling features available in the latest version of the ESP-IDF framework**.
3. **Outline the conception** of a tool that would provide diagnostic information on all necessary operations and the status of a target Flash-based memory module on the ESP32 platform...




```
static wl_instance_t s_instances[MAX_WL_HANDLES];
```

```
typedef struct ALIGNED_(32) WL_State_s {
public:
    uint32_t pos;           /*!< current dummy block position*/
    uint32_t max_pos;       /*!< maximum amount of positions*/
    uint32_t move_count;    /*!< total amount of move counts. Used to calculate the address*/
    uint32_t access_count;  /*!< current access count*/
    uint32_t max_count;     /*!< max access count when block will be moved*/
    uint32_t block_size;    /*!< size of move block*/
    uint32_t version;       /*!< state id used to identify the version of current library implementation*/
    uint32_t device_id;     /*!< ID of current WL instance*/
    uint32_t reserved[7];   /*!< Reserved space for future use*/
    uint32_t crc;           /*!< CRC of structure*/
} wl_state_t;
```

Approaches to BE

- backlog: real time updates with linker wrapped read/write/erase
- tested: separate partition for BE application
- C++ lib usable in Python for partition dump analysis
- **current focus: reconstruct WL instance $\rightarrow$ access to status**

- https://github.com/omnitex/espwlmon
- **Concept of BE implemented:** detect WL partition, reconstruct status, print as JSON
- other approaches tested and experimented with
- settling on non-breaking (towards ESP-IDF) implementation


```
"pos":"0x14","max_pos":"0xfb","move_count":"0x0",
"access_count":"0x0","max_count":"0x10",
"block_size":"0x1000","version":"0x2",
"max_count":"0x10","device_id":"0xc0dbc154",
"crc":"0x9fec91f6"
```

Studied literature

- Bez, R., Camerlenghi, E., Modelli, A. and Visconti, A. **Introduction to flash memory**. Proceedings of the IEEE. 2003, vol. 91, no. 4, p. 489–502. DOI: 10.1109/JPROC.2003.811702.
- Chang, Y.-H., Hsieh, J.-W. and Kuo, T.-W. **Improving Flash Wear-Leveling by Proactively Moving Static Data**. IEEE Transactions on Computers. 2010, vol. 59, no. 1, p. 53–65. DOI: 10.1109/TC.2009.134.
- Kim, S. and Kwak, J. **Prediction of elapsed time based wear leveling for NAND flash memory in embedded systems**. february 2016, vol. 11, p. 578–585.
- Elm chan. **FatFs - Generic FAT Filesystem Module** (online). November 2022. Available at: `http://elm-chan.org/fsw/ff/00index_e.html`.
- SPIFFS, ESP-IDF wear leveling...

Plan

- resolving issues (listed on Github); linux target, `performance` and `safe` modes etc.
- focus on other mentioned approaches, derive statistics from gathered info
- once reasonable BE is done, begin visualization in Python
- many possibilities; real time, predicting memory lifetime, manipulating WL parameters on live instance...

Plan

- resolving issues (listed on Github); linux target, `performance` and `safe` modes etc.
- focus on other mentioned approaches, derive statistics from gathered info
- once reasonable BE is done, begin visualization in Python
- many possibilities; real time, predicting memory lifetime, manipulating WL parameters on live instance...

Thank you for your attention

Zhodnocení firemním mentorem

Pan Havlík pracuje na vývoji softwarového monitoru a analyzéru wear-levelling komponenty pro embedded aplikace na platformě ESP32. Vývoj vyžaduje pochopení WL mechanismu a návrh vlastního monitorovacího/analytického mechanismu (neinvazivní forma). Pan Havlík nároky projektu zvládá velice dobře, výsledky práce průběžně konzultujeme osobně i prostřednictvím veřejného GitHub repozitáře.

Martin Vychodil