AN ALGORITHM OF EXTENDING EEPROM WRITE CYCLES

Gheorghe PURCI, Veaceslav NASTASENCO

Technical University of Moldova, 168, Stefan cel Mare Blvd., MD - 2004, Chisinau, Republic of Moldova

Purci Gheorghe, gepurice@gmail.com
Nastasenco Veaceslav, vnastasenco@adm.utm.md

Actually EEPROM is commonly used to store configuration parameters and operating history information in embedded processors. For example, to record the most recent operating data in case there is a system failure or power loss. This sort of things might require recording data every few seconds [1].

The issue here is that EEPROM works for a limited number of write cycles. After from 100,000 to 1,000,000 (depending on particular chip), some of deployed systems will start exhibiting EEPROM wearout and a field failure might be get. A million writes sounds like a lot, but they go by pretty quickly [1].

If certain EEPROM cells are periodically updated a danger of EEPROM wearout arise. Reducing per-cell write frequency (sometimes more than about once per hour) the problem can be solved.

If the set values, recorded in EPROM at power loss forms a 'data frame', abovementioned problem might be solved by organizing a consecutive data frames in EEPROM (named Consecutive FIFO). An additional counter, stored into EEPROM, pinpoint most recent data frame in CFIFO [1].

In [1] Microchip suggests to use Gray code so that only one byte out of a multi-byte counter has to be updated on each count. It is recommended also to use error correcting code to compensate wear-out.

Another idea is to use Hamming code, depend upon whether bit failures are independent within the counter data bytes. Analyzing the Gray code, it was concluded that the number of registrations can increase twice.

These solutions reduces EEPROM wearout proportionally to the number of data frames stored in CFIFO. For example, if 10 spaces for data frames are used to record data, each frame space is modified 1/10th times and EEPROM wearout is improved by a factor of 10, but wearout of EEPROM .

The authors of article propose an algorithm to avoid frequent storing of pinpoint counter in EEPROM, even more extending its write cycles. The idea is to use one bit of 'data frame' to pinpoint the last writen cell in EEPROM.

Keywords: *EEPROM, divide et impera, bit, data,* wearout

References

1. Avoiding EEPROM Wearout [online]. [Accessed: 19.09.2019]. Available: https://betterembsw.blogspot.com/2015/07/avoiding-eeprom-wearout.html
2. High-Reliability and High-Frequency EEPROM Counter [online]. [Accesed: 19.09.2019]. Available: http://ww1.microchip.com/downloads/en/AppNotes/01449A.pdf