**Робота із Serial DataFlash через SPI1**

Пам’ять мікросхеми аналогового реєстратора AR45DB641E (Serial DataFlash) – складається зі 32768 сторінок розміром по 264 байт. Тобто загальна пам’ять - 32768×264 = 8 650 752 (Байт) = 8 448 кБайт = 8,25 МБайт

Пам’ять мікросхеми дискретного реєстратора AR45DB081D (Serial DataFlash) – складається зі 4096 сторінок розміром по 264 байт. Тобто загальна пам’ять - 4096×264 = 1081344 (Байт) = 1056 кБайт = 1,03125 МБайт

Ці мікросхеми обслуговується через інтерфейс SPI1, який крім даних мікросхем обслуговує ще і мікросхеми EEPROM і RTC. Це означає, що якщо виконується якась операція (з будь-якою мікросхемою), то наступна операція виконається тільки коли попередня повністю закінчиться.

При встановленні одночасно декількох задач на виконання – першою починає виконуватися та операція, яка має більший пріоритет згідно зі нижче поданим списком (задачі з меншим порядковим номером мають більший пріоритет)

1. Початок запису таблиці настройок захистів
2. Початок запису таблиці настройок БС-МРЗВ
3. Початок запису юстуючих коефіцієнтів і серійного номеру пристрою
4. Початок запису тригерних світлоіндикаторів і сигнальних реле
5. Початок запису триґерних функцій
6. Початок запису таблиці для аналогового реєстратора
7. Початок запису таблиці для дискретного реєстратора
8. Початок запису таблиці для реєстратора програмних подій
9. Початок читання юстуючих коефіцієнтів і серійного номеру пристрою
10. Початок читання таблиці настройок захистів
11. Початок читання таблиці настройок БС-МРЗВ
12. Початок читання тригерних світлоіндикаторів і сигнальних реле
13. Початок читання триґерних функцій
14. Початок читання таблиці для аналогового реєстратора
15. Початок читання таблиці для дискретного реєстратора
16. Початок читання таблиці для реєстратора програмних подій
17. Початок зчитування системної дати і часу, а також калібровочного числа
18. Початок запису системної дати і часу
19. Початок запису калібровочного числа
20. Початок повного стирання всієї мікросхеми DataFlash
21. Початок запису інформації у мікросхеми DataFlash[[1]](#footnote-1)
22. Початок читання інформації з мікросхеми DataFlash

Для мінімізації «простою» каналу SPI1 зроблено, що коли ідуть трансакції між мікросхемами DataFlash аналогового реєстратора чи дискретного реєстратора, то вони взаємно чергуються, тобто по завершенні трансакції з однією мікросхемою буде виконуватися трансакція з другою мікросхемою (якщо вона, звичайно, стоїть у черзі) бо більшість операцій з DataFlash вимагають певного часу на їх виконання, а за цей час, можна виконати «звернення» до іншої мікросхеми DataFlash.

Для виконання операцій читання/запису мікросхеми аналогового реєстратора використовуються такі глобальні змінні:

***unsigned int address\_read\_write\_ar*** – початкова адреса у мікросхемі DataFlash з якої треба почати зчитування, або з якої треба почати запис

***unsigned int number\_bytes\_read\_write\_ar*** - кількість інформації (у байтах) яку треба зчитати або записати

***unsigned int buffer\_serial\_DataFlash\_read\_write\_ar [SIZE\_BUFFER\_SERIAL\_DATAFLASH\_READ\_WRITE\_AR]*** – масив в який записується прочитана інформація, або з якого записується інформація

Для виконання операцій читання/запису мікросхеми дискретного реєстратора використовуються такі глобальні змінні:

***unsigned int address\_read\_write\_dr*** – початкова адреса у мікросхемі DataFlash з якої треба почати зчитування, або з якої треба почати запис

***unsigned int number\_bytes\_read\_write\_dr*** - кількість інформації (у байтах) яку треба зчитати або записати

***unsigned int buffer\_serial\_DataFlash\_read\_write\_dr [SIZE\_BUFFER\_SERIAL\_DATAFLASH\_READ\_WRITE\_DR]*** – масив в який записується прочитана інформація, або з якого записується інформація

Ці змінні об’явлені у файлі ***variable\_global\_mal.h*** і посиланні на неї є у файлі ***variables\_external\_mal.h***

Константи ***SIZE\_BUFFER\_SERIAL\_DATAFLASH\_READ\_WRITE\_AR*** і ***SIZE\_BUFFER\_SERIAL\_DATAFLASH\_READ\_WRITE\_DR*** об’явлена у файлі ***EEPROM\_RTC\_DataFlash.h*** і на сьогоднішній день становить 5 сторінок мікросхем DataFlash.

**Увага!**

1. **У випадку встановлення читання чи запису кількості інформації (у байтах) більшої за розмір масиву *buffer\_serial\_DataFlash\_read\_write\_Xr* це приведе до виклику зациклення *while(1)*, яка зупинить виконання програмного забезпечення і приведе до спрацювання watchdog (якщо його контакти є з’єднані перемичкою)**
2. **Змінні *address\_read\_write\_Xr* і *number\_bytes\_read\_write\_Xr,* можна змінювати тільки до подачі команди читання чи запису при умові, що попередня операція читання чи запису гарантовано закінчена. У випадку, якщо у процесі виконання операції читання чи запису відбувається зміна змінних *address\_read\_write\_Xr*, або *number\_bytes\_read\_writ\_Xr e*, то це у кращому випадку приведе до виклику зациклення *while(1)*, а у гіршому випадку приведе до непередбачуваної роботи програмного забезпечення.**
3. **Змінювати вмістиме масиву *buffer\_serial\_DataFlash\_read\_write\_Xr* можна тільки до подачі команди запису. Інакше дані , записані у мікросхему DataFlash будуть недостовірні.**
4. **Читати вмістиме масиву *buffer\_serial\_DataFlash\_read\_write\_Xr* можна тільки після завершення команди читання при умові. Інакше проінтерпретовані дані будуть недостовірні.**

**Операція стирання всієї пам’яті мікросхеми DataFlash**

Операція стирання всієї пам’яті мікросхеми DataFlash ініціюється будь-якою системою встановленням біту **TASK\_START\_ERASE\_DATAFLASH\_AR\_BIT** (для аналогового реєстратора) або **TASK\_START\_ERASE\_DATAFLASH\_DR\_BIT** (для дискретного реєстратора) (константа об’явлена у ***EEPROM\_RTC\_DataFlash.h***)) у масиві ***control\_spi1\_taskes[2]*** (масив ***control\_spi1\_taskes*** об’явлений у ***variable\_global\_mal.h/variables\_external\_mal.h*** як ***unsigned int control\_spi1\_taskes[2] = {0,0}***)

Ця операція виконується з допомогою макросу **\_SET\_BIT**

**Приклад:**

**\_SET\_BIT(control\_spi1\_taskes, TASK\_START\_ERASE\_DATAFLASH\_AR\_BIT);**

Умовою завершення подачі команди (не завершення всієї операції, яка триває декілька секунд і визначається читанням регістру статусу при подачі наступної команди) є відсутність встановлених двох біт:

**TASK\_START\_ERASE\_DATAFLASH\_AR\_BIT/ TASK\_START\_ERASE\_DATAFLASH\_DR\_BIT**

**TASK\_ERASING\_DATAFLASH\_BIT/ TASK\_ERASING\_DATAFLASH\_BIT**

Перевірку встановлених біт можна виконати макросом \_CHECK\_SET\_BIT

**Приклад:**

**if (**

**(\_CHECK\_SET\_BIT(control\_spi1\_taskes, TASK\_START\_ERASE\_DATAFLASH\_AR\_BIT) == 0) &&**

**(\_CHECK\_SET\_BIT(control\_spi1\_taskes, TASK\_ERASING\_DATAFLASH\_AR\_BIT) == 0)**

**)**

**{**

**//Операція стирання всієї пам’яті мікросхеми DataFlash аналогового реєстратора коректно подана і зараз виконується або вже завершилася**

**}**

**else**

**{**

**//Операція стирання всієї пам’яті мікросхеми DataFlash аналогового реєстратора ще не завершилася**

**}**

**Операція читання пам’яті мікросхеми DataFlash**

Операція читання пам’яті мікросхеми DataFlash ініціюється будь-якою системою виконанням такої процедури:

1. Упевнитися, що на даний момент не ідуть читання чи запис мікросхеми DataFlash, тобто:

Для аналогового реєстратора

**if (**

**(\_CHECK\_SET\_BIT(control\_spi1\_taskes, TASK\_START\_WRITE\_SERIAL\_DATAFLASH\_AR\_BIT == 0) &&**

**(\_CHECK\_SET\_BIT(control\_spi1\_taskes, TASK\_WRITING\_SERIAL\_DATAFLASH\_AR \_BIT) == 0) &&**

**(\_CHECK\_SET\_BIT(control\_spi1\_taskes, TASKTASK\_START\_READ\_SERIAL\_DATAFLASH\_AR \_BIT) == 0) &&**

**(\_CHECK\_SET\_BIT(control\_spi1\_taskes, TASK\_ERASING\_DATAFLASH\_AR \_BIT) == 0)**

**)**

**{**

**//Можна змінювати змінні *address\_read\_write\_ar* і *number\_bytes\_read\_write\_ar*** і вмістиме буферу ***buffer\_serial\_DataFlash\_read\_write\_ar***

**}**

**else**

**{**

**//ще не можна змінювати змінні *address\_read\_write\_ar* і *number\_bytes\_read\_write\_ar*** і вмістиме буферу ***buffer\_serial\_DataFlash\_read\_write\_ar – бо не завершена попередня операція роботи з мікросхемою DataFlash***

**}**

Для дискретного реєстратора

**if (**

**(\_CHECK\_SET\_BIT(control\_spi1\_taskes, TASK\_START\_WRITE\_SERIAL\_DATAFLASH\_DR\_BIT == 0) &&**

**(\_CHECK\_SET\_BIT(control\_spi1\_taskes, TASK\_WRITING\_SERIAL\_DATAFLASH\_DR \_BIT) == 0) &&**

**(\_CHECK\_SET\_BIT(control\_spi1\_taskes, TASKTASK\_START\_READ\_SERIAL\_DATAFLASH\_DR \_BIT) == 0) &&**

**(\_CHECK\_SET\_BIT(control\_spi1\_taskes, TASK\_ERASING\_DATAFLASH\_DR \_BIT) == 0)**

**)**

**{**

**//Можна змінювати змінні *address\_read\_write\_dr* і *number\_bytes\_read\_write\_dr*** і вмістиме буферу ***buffer\_serial\_DataFlash\_read\_write\_dr***

**}**

**else**

**{**

**//ще не можна змінювати змінні *address\_read\_write\_dr* і *number\_bytes\_read\_write\_dr*** і вмістиме буферу ***buffer\_serial\_DataFlash\_read\_write\_dr – бо не завершена попередня операція роботи з мікросхемою DataFlash***

**}**

2. У змінній ***address\_read\_write\_Xr*** встановити початкову адресу адресного простору мікросхеми DataFlash з якого почнеться зчитування даних

3. У змінній ***number\_bytes\_read\_write\_Xr*** встановити кількість байт для читання (**НЕ БІЛЬШЕ НІЖ РОЗМІА МАСИВУ *buffer\_serial\_DataFlash\_read\_write\_Xr***)

4. Встановити біт **TASK** **TASK\_START\_READ\_SERIAL\_DATAFLASH\_AR\_BIT/ TASK** **TASK\_START\_READ\_SERIAL\_DATAFLASH\_DR\_BIT** (константа об’явлена у ***EEPROM\_RTC\_DataFlash.h***)) у масиві ***control\_spi1\_taskes[2]*** (масив ***control\_spi1\_taskes*** об’явлений у ***variable\_global\_mal.h/variables\_external\_mal.h*** як ***unsigned int control\_spi1\_taskes[2] = {0,0}***)

Ця операція виконується з допомогою макросу **\_SET\_BIT**

**Приклад:**

**\_SET\_BIT(control\_spi1\_taskes, TASK\_START\_READ\_SERIAL\_DATAFLASH\_AR\_BIT);**

Умовою завершення читання є відсутність встановлених двох біт:

**TASK\_START\_READ\_SERIAL\_DATAFLASH\_AR\_BIT/ TASK\_START\_READ\_SERIAL\_DATAFLASH\_DR\_BIT**

**TASK\_READING\_SERIAL\_DATAFLASH\_AR\_BIT/ TASK\_READING\_SERIAL\_DATAFLASH\_DR\_BIT**

Перевірку встановлених біт можна виконати макросом \_CHECK\_SET\_BIT

**Приклад:**

**if (**

**(\_CHECK\_SET\_BIT(control\_spi1\_taskes, TASK\_START\_READ\_SERIAL\_DATAFLASH\_AR\_BIT) == 0) &&**

**(\_CHECK\_SET\_BIT(control\_spi1\_taskes, TASK\_READING\_SERIAL\_DATAFLASH\_AR\_BIT) == 0)**

**)**

**{**

**//Операція читання визначеної кількості інформації завершено**

**}**

**else**

**{**

**//Операція читання ще продовжується**

**}**

5. Можна користуватися прочитаними даними з масиву ***buffer\_serial\_DataFlash\_read\_write\_Xr***.

**Примітка:** У масиві буде перша половина (розмір визначається змінною ***number\_bytes\_read\_write\_Xr***) заповнена новозчитаними даними, а друга половина (до кінця масиву) буде містити «баласт» з попередніх операцій читання/запису.

**Операція запису пам’яті мікросхеми DataFlash**

Операція запису пам’яті мікросхеми DataFlash ініціюється будь-якою системою виконанням такої процедури:

1. Упевнитися, що на даний момент не ідуть читання чи запис мікросхеми DataFlash, тобто:

Для аналогового реєстратора

**if (**

**(\_CHECK\_SET\_BIT(control\_spi1\_taskes, TASK\_START\_WRITE\_SERIAL\_DATAFLASH\_AR\_BIT == 0) &&**

**(\_CHECK\_SET\_BIT(control\_spi1\_taskes, TASK\_WRITING\_SERIAL\_DATAFLASH\_AR\_BIT) == 0) &&**

**(\_CHECK\_SET\_BIT(control\_spi1\_taskes, TASKTASK\_START\_READ\_SERIAL\_DATAFLASH\_AR\_BIT) == 0) &&**

**(\_CHECK\_SET\_BIT(control\_spi1\_taskes, TASK\_ERASING\_DATAFLASH\_AR\_BIT) == 0)**

**)**

**{**

**//Можна змінювати змінні *address\_read\_write\_ar* і *number\_bytes\_read\_write\_ar*** і вмістиме буферу ***buffer\_serial\_DataFlash\_read\_write\_ar***

**}**

**else**

**{**

**//ще не можна змінювати змінні *address\_read\_write\_ar* і *number\_bytes\_read\_write\_ar*** і вмістиме буферу ***buffer\_serial\_DataFlash\_read\_write\_ar – бо не завершена попередня операція роботи з мікросхемою DataFlash***

**}**

Для дискретного реєстратора

**if (**

**(\_CHECK\_SET\_BIT(control\_spi1\_taskes, TASK\_START\_WRITE\_SERIAL\_DATAFLASH\_DR\_BIT == 0) &&**

**(\_CHECK\_SET\_BIT(control\_spi1\_taskes, TASK\_WRITING\_SERIAL\_DATAFLASH\_DR\_BIT) == 0) &&**

**(\_CHECK\_SET\_BIT(control\_spi1\_taskes, TASKTASK\_START\_READ\_SERIAL\_DATAFLASH\_DR\_BIT) == 0) &&**

**(\_CHECK\_SET\_BIT(control\_spi1\_taskes, TASK\_ERASING\_DATAFLASH\_DR\_BIT) == 0)**

**)**

**{**

**//Можна змінювати змінні *address\_read\_write\_dr* і *number\_bytes\_read\_write\_dr*** і вмістиме буферу ***buffer\_serial\_DataFlash\_read\_write\_dr***

**}**

**else**

**{**

**//ще не можна змінювати змінні *address\_read\_write\_dr* і *number\_bytes\_read\_write\_dr*** і вмістиме буферу ***buffer\_serial\_DataFlash\_read\_write\_dr – бо не завершена попередня операція роботи з мікросхемою DataFlash***

**}**

2. Масив ***buffer\_serial\_DataFlash\_read\_write\_Xr*** заповнити даними для запису

2. У змінній ***address\_read\_write\_Xr*** встановити початкову адресу адресного простору мікросхеми DataFlash з якого почнеться запис даних

3. У змінній ***number\_bytes\_read\_write\_Xr*** встановити кількість байт для запису (**НЕ БІЛЬШЕ НІЖ РОЗМІА МАСИВУ *buffer\_serial\_DataFlash\_read\_write\_Xr***)

4. Встановити біт **TASK\_START\_WRITE\_SERIAL\_DATAFLASH\_AR\_BIT/ TASK\_START\_WRITE\_SERIAL\_DATAFLASH\_DR\_BIT** (константа об’явлена у ***EEPROM\_RTC\_DataFlash.h***)) у масиві ***control\_spi1\_taskes[2]*** (масив ***control\_spi1\_taskes*** об’явлений у ***variable\_global\_mal.h/variables\_external\_mal.h*** як ***unsigned int control\_spi1\_taskes[2] = {0,0}***)

Ця операція виконується з допомогою макросу **\_SET\_BIT**

**Приклад:**

**\_SET\_BIT(control\_spi1\_taskes, TASK\_START\_WRITE\_SERIAL\_DATAFLASH\_AR\_BIT);**

Умовою завершення запису є відсутність встановлених двох біт:

**TASK\_START\_WRITE\_SERIAL\_DATAFLASH\_AR\_BIT/ TASK\_START\_WRITE\_SERIAL\_DATAFLASH\_DR\_BIT**

**TASK\_WRITING\_SERIAL\_DATAFLASH\_AR\_BIT/ TASK\_WRITING\_SERIAL\_DATAFLASH\_DR\_BIT**

Перевірку встановлених біт можна виконати макросом \_CHECK\_SET\_BIT

**Приклад:**

**if (**

**(\_CHECK\_SET\_BIT(control\_spi1\_taskes, TASK\_START\_WRITE\_SERIAL\_DATAFLASH\_AR\_BIT) == 0) &&**

**(\_CHECK\_SET\_BIT(control\_spi1\_taskes, TASK\_WRITING\_SERIAL\_DATAFLASH\_AR\_BIT) == 0)**

**)**

**{**

**//Операція запису визначеної кількості інформації завершено**

**}**

**else**

**{**

**//Операція запису ще продовжується**

**}**

1. Записувати у мікросхему можна будь-яку кількість інформації і з будь-якої адреси. Але є два режими запису:

   повністю ціла сторінка (коли початкова адреса є кратною 264 байтам і кількість байт для запису є не менше (можна більше або рівне) розміру однієї сторінки – 264 байт)

   частина сторінки (коли початкова адреса не є кратною 264 байтам. або кількість байт для запису є менше розміру однієї сторінки – 264 байт)

   У першому випадку операція виконується за одну трансакцію через інтерфейс SPI1. У другому випадку операція виконується за три основні трансакції через інтерфейс SPI1, причому між кожними трансакціями встановлюється додаткові трансакції читання регістру статусу, які виконуються стільки раз. поки мікросхема DataFlash не стане вільною від внутрішніх дій. які були викликані попередньою основною трансакціє.

   Тобто перший режим запису є швидшим за другий режим запису [↑](#footnote-ref-1)