

# Современные 32-разрядные ARM-микроконтроллеры серии STM32: флэш-память

Олег Вальпа (г. Миасс, Челябинская обл.)

В статье приводится описание организации флэш-памяти 32-разрядных ARM-микроконтроллеров серии STM32 компании STMicroelectronics. Рассмотрены её состав, архитектура и особенности использования, а также приведён пример программы, иллюстрирующий работу с флэш-памятью.

## ВВЕДЕНИЕ

Несмотря на все преимущества микроконтроллеров серии STM32 [1], они всё же имеют определённый недостаток, который заключается в отсутствии электрически перепрограммируемой постоянной памяти EEPROM. Многие микроконтроллеры семейства AVR и PIC имеют такую память, и это позволяет без проблем решать некоторые задачи, связанные с оперативным изменением и долговременным хранением данных. К сожалению, в микроконтроллерах серии STM32 такой памяти нет. Но выход всё же существует. Например, можно подключить внешнюю память EEPROM через последовательный интерфейс I<sup>2</sup>C или SPI. Если же не хочется подключать дополнительную внешнюю микросхему, то в качестве энергонезависимой памяти в STM32 можно использовать энергонезависимые ячейки памяти ВКР [2] или часть флэш-памяти, которая предназначена для

хранения программ. Правда, запись в такую память со стороны процессора требует соблюдения определённых процедур. Тем не менее, это лучше, чем ничего.

Использование энергонезависимых ячеек памяти ВКР возможно лишь при наличии внешней батареи питания. Кроме того, количество таких ячеек невелико, поэтому предпочтительнее вариант использования некоторой части флэш-памяти.

Поскольку STM32 имеет большой объём флэш-памяти, частичное её использование для хранения энергонезависимых данных не причинит больших неудобств. Необходимо ещё учесть, что количество циклов перезаписи для флэш-памяти ограничено 10 000 циклов, в отличие от 100 000 циклов обычной EEPROM памяти. Однако 10 000 циклов тоже немало, а если выделить для энергонезависимых данных несколько областей флэш-памяти и производить поочерёдное их

использование, то количество циклов можно многократно увеличить.

## ОРГАНИЗАЦИЯ ФЛЭШ-ПАМЯТИ

Флэш-память состоит из главной памяти (Main Memory), информационного блока (Information Block) и блока регистров интерфейса с флэш-памятью (Flash Memory Interface Registers).

Для записи данных во флэш-память STM32 используется блок FPFC (Flash Program and Erase Controller), то есть контроллер записи и стирания флэш-памяти.

Интерфейс с флэш-памятью обладает следующими особенностями:

- интерфейс чтения с буфером предварительной выборки (два слова по 64 бита);
- опционный байтовый загрузчик;
- операции программирования и стирания флэш-памяти;
- защита от записи и чтения.

В таблице 1 представлена адресная организация флэш-памяти младшего семейства микроконтроллеров серии STM32.

Первый блок представляет собой основную память, в которую записывается программа и другие данные. Блок основной памяти поделён на страницы объёмом несколько килобайт.

Информационный блок содержит в себе два раздела: системную память и опционные байты. В системной памяти хранится загрузчик, который позволяет программировать контроллер через последовательный порт USART. Этот раздел памяти невозможно удалить или модифицировать. Раздел опционных байт хранит информацию о защите основной памяти. Он позволяет включить или отключить защиту от чтения и записи данных в основную флэш-память.

Прочитать программно 32-разрядные данные с именем переменной data из флэш-памяти можно при помощи следующей команды: `data = (*(_IO uint32_t*) address)`.

Записать данные в основную область флэш-памяти можно только после предварительного стирания этой

Таблица 1. Адресная организация флэш-памяти младшего семейства STM32

Блок	Имя	Базовый адрес	Размер, байт
Основная память	Страница 0	0x08000000 - 0x080003FF	1 КБ
	Страница 1	0x08000400 - 0x080007FF	1 КБ
	Страница 2	0x08000800 - 0x08000BFF	1 КБ
	Страница 3	0x08000C00 - 0x08000FFF	1 КБ
	...	...	...
	Страница 31	0x08007C00 - 0x08007FFF	1 КБ
Информационный блок	Системная память	0x1FFFF000 - 0x1FFFF7FF	2 КБ
	Опционные байты	0x1FFFF800 - 0x1FFFF80F	16
Регистры интерфейса с флэш-памятью	FLASH_ACR	0x40022000 - 0x40022003	4
	FLASH_KEYR	0x40022004 - 0x40022007	4
	FLASH_OPTKEYR	0x40022008 - 0x4002200B	4
	FLASH_SR	0x4002200C - 0x4002200F	4
	FLASH_CR	0x40022010 - 0x40022013	4
	FLASH_AR	0x40022014 - 0x40022017	4
	Резерв	0x40022018 - 0x4002201B	4
	FLASH_OBR	0x4002201C - 0x4002201F	4
FLASH_WRP	0x40022020 - 0x40022023	4	



**Листинг**

```

// Подключение библиотек
#include "stm32f10x.h"
#include "stm32f10x_gpio.h"
#include "stm32f10x_rcc.h"
// Объявление ключей разблокировки
#define FLASH_KEY1 ((uint32_t)0x45670123)
#define FLASH_KEY2 ((uint32_t)0xCDEF89AB)
// Объявление номера страницы
#define FLASH_PAGE ((uint8_t)0x7F)
// Функция анализа готовности флэш-памяти
uint8_t flash_ready(void)
{
    return !(FLASH->SR & FLASH_SR_BSY);
}
// Функция стирания страницы флэш-памяти,
// которой принадлежит определенный адрес
void flash_erase_page(uint32_t address)
{
    FLASH->CR |= FLASH_CR_PER;
    FLASH->AR = address;
    FLASH->CR |= FLASH_CR_STRT;
    while(!flash_ready()) {};
    FLASH->CR &= ~FLASH_CR_PER;
}
// Функция разблокировки флэш-памяти
void flash_unlock(void)
{
    FLASH->KEYR = FLASH_KEY1;
    FLASH->KEYR = FLASH_KEY2;
}
// Функция блокировки флэш-памяти
void flash_lock()
{
    FLASH->CR |= FLASH_CR_LOCK;
}
// Функция записи 32-разрядного слова во флэш-память
// по определенному адресу
void flash_write(uint32_t address, uint32_t data)
{
    FLASH->CR |= FLASH_CR_PG;
    while(!flash_ready()) {};
    *(__IO uint16_t*)address = (uint16_t)data;
    while(!flash_ready()) {};
    address+=2;
    data>>=16;
    *(__IO uint16_t*)address = (uint16_t)data;
    while(!flash_ready()) {};
    FLASH->CR &= ~(FLASH_CR_PG);
}
// Функция чтения флэш-памяти
uint32_t flash_read(uint32_t address)
{
    return *(__IO uint32_t*) address;
}
// Главный модуль программы
int main(void)
{
    uint32_t dwr=0x12345678; // Переменная для записи
    uint32_t drd=0x0; // Переменная для чтения
    // Стартовый адрес страницы
    uint32_t st_address = FLASH_BASE + FLASH_PAGE * 1024;
    flash_unlock(); // Разблокировать флэш-память
    flash_erase_page(st_address); // Стереть страницу
    flash_lock(); // Заблокировать флэш-память
    flash_unlock(); // Разблокировать флэш-память
    flash_write(st_address, dwr); // Записать данные
    flash_lock(); // Заблокировать флэш-память
    drd=flash_read(st_address); // Читать данные
    while(1)
    {
        // место для других команд
    }
}

```

Регистр FLASH\_SR содержит следующие специальные разряды:

- разряд 5 EOP (End of Operation) устанавливается аппаратно при успешном завершении операции записи или стирания флэш-памяти;
- разряд 4 WRPRERR (Write Protection Error) устанавливается при попытке программирования защищённой от записи области памяти;
- разряд 2 PGERR (Programming Error) устанавливается при программировании, если данные по заданному адресу отличны от значения 0xFFFF, то есть не стёрты перед программированием;
- разряд 0 BSY (Busy) устанавливается во время операций записи и стирания флэш-памяти. Он аппаратно сбрасывается по окончании операций записи или стирания, а также при возникновении ошибки во время этих операций.

Все эти разряды сбрасываются записью в них единичного значения.

Регистр FLASH\_CR имеет следующие разряды:

- разряд 12 EOPIE (End of Operation Interrupt Enable) разрешает генерацию прерывания после завершения операции записи или стирания данных во флэш-память;
- разряд 10 ERRIE (Error Interrupt Enable) разрешает генерацию прерывания при возникновении ошибок, то есть при установке бита PGERR или WRPRERR в регистре FLASH\_SR;
- разряд 9 OPTWRE (Option Bytes Write Enable) разрешает программирование опционных байт, устанавливается записью последовательности ключей в регистр FLASH\_OPTKEYR, а сбрасывается программно;
- разряд 7 LOCK блокирует запись и стирание флэш-памяти;
- разряд 6 STRT запускает процедуру стирания флэш-памяти;
- разряд 5 OPTER разрешает процедуру стирания опционных байт;
- разряд 4 OPTPG разрешает процедуру программирования опционных байт;
- разряд 2 MER (Mass Erase) разрешает процедуру полного стирания флэш-памяти;
- разряд 1 PER (Page Erase) разрешает процедуру стирания одной страницы флэш-памяти;
- разряд 0 PG (Programming) разрешает процедуру программирования флэш-памяти.

Регистр FLASH\_AR предназначен для задания адреса программируе-

мой флэш-памяти. В нём указывается адрес, по которому будут записаны данные при программировании. Для стирания определённой страницы флэш-памяти в этот регистр необходимо записать любой из адресов, входящих в эту страницу, причём не обязательно начальный.

Регистры FLASH\_OBR и FLASH\_WRPR в операциях записи и стирания не задействованы. Они предназначены для установки защиты от несанкционированного доступа к коду программы, а также для некоторых других настроек.

### ПРИМЕР ПРОГРАММЫ

Рассмотрим пример программы для стирания и программирования флэш-памяти непосредственно из работающего приложения, то есть в так называемом режиме IAP (In Application Programming).

Все процедуры стирания и записи флэш-памяти производятся с активным использованием рассмотренных выше регистров блока FPFC.

В листинге приведена программа, которая содержит все необходимые функции для работы с флэш-памятью, а так же главный модуль, осуществляющий стирание страницы флэш-памяти и запись в неё произвольных данных. Программа снабжена подробными комментариями, которые поясняют выполняемые процедуры.

На основе примера из листинга можно создать библиотечную функцию, которая будет эмулировать память EEPROM и может применяться по мере необходимости в любых программах для микроконтроллеров серии STM32.

Следует помнить о том, что для стирания и записи данных во флэш-память требуется определённое время, поэтому во время выполнения таких операций реакция микроконтроллера будет замедленной.

В качестве тренировки можно развить данную программу, дополнив её возможностью обмена данными через последовательный порт USART, который позволит производить нужные операции с флэш-памятью в режиме диалога с оператором.

### ЛИТЕРАТУРА

1. [www.st.com](http://www.st.com).
2. *Вальна О.* Современные 32-разрядные ARM-микроконтроллеры серии STM32: блок резервных данных ВКР. Современная электроника. № 3. 2014.

