

ARM-микроконтроллеры серии STM32: флэш-память

Олег Вальпа

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

ВВЕДЕНИЕ

Несмотря на все преимущества микроконтроллеров серии STM32 [1], они все же имеют определенный 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 все же существует. Например, можно подключить внешнюю память EEPROM через последовательный интерфейс I²C или SPI. Если же не хочется подключать дополнительную внешнюю микросхему, то в качестве энергонезависимой памяти в STM32 можно использовать энергонезависимые ячейки памяти BKP [2] или часть флэш-памяти, которая предназначена для хранения программ. Правда, запись в такую память со стороны процессора требует соблюдения определенных процедур. Тем не менее, это лучше, чем ничего.

Использование энергонезависимых ячеек памяти BKP 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 объем флэш-памяти, частичное ее использование для хранения энергонезависимых данных не причинит больших неудобств. Необходимо еще 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 поочередное 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 FPEC (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 поделен на страницы объемом несколько килобайт.

Информационный блок содержит в себе два раздела: системную память и опционные байты. В системной памяти хранится загрузчик, который позволяет

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

Информационный блок	Страница 31	0x08007C00 - 0x08007FFF	1 КБ
	Системная память	0x1FFFF000 - 0x1FFFF7FF	2 КБ
Регистры интерфейса с флэш-памятью	Опционные байты	0x1FFFF800 - 0x1FFFF80F	16
	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

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 области памяти. Кроме того, перед записью данных необходимо разрешить запись в специальном регистре блока FPFC.

Стирать флэш-память можно только кратно страницам, каждая из которых состоит из нескольких килобайт. При этом необходимо помнить, что во флэш-памяти записана и хранится программа микроконтроллера. Поэтому область для записи данных необходимо выбирать в свободном от кода программы пространстве флэш-памяти. Удобнее всего для этого использовать последние страницы памяти. Кроме того, необходимо регулярно следить, чтобы размер программы не перекрыл область поль-

зовательских данных, учитывая размер страниц микроконтроллера.

Все разряды данных очищенной области флэш-памяти принимают единичное значение. По этому признаку можно определить очищена память или нет.

Запись в основную область флэш-памяти можно производить по любому адресу. Вначале записываются два младших байта слова, а затем два старших байта.

Перед стиранием или записью в память необходимо снять ее блокировку, а после окончания процедуры стирания или записи установить блокировку. Для снятия блокировки флэш-памяти необходимо последовательно записать в регистр FLASH_KEYR два числа: 0x45670123 и 0xCDEF89AB. Если записать в этот регистр другие данные, то блокировку невозможно будет снять до полной перезагрузки микроконтроллера.

РЕГИСТРЫ БЛОКА FPFC

Карта регистров блока контроллера записи и стирания флэш-памяти FPFC приведена в таблице 2. Рассмотрим состав и назначение регистров данного блока, принимающих участие в работе с флэш-памятью.

В регистре FLASH_ACR используется лишь один третий разряд HLF CYA, который разрешает половинные циклы обращения к флэш-памяти, то есть 16-разрядными данными. Остальные биты этого регистра зарезервированы.

Регистр FLASH_KEYR используется для разблокировки флэш-памяти контроллера FPFC и обеспечивает сброс бита LOCK в регистре FLASH_CR. Разблокировка производится последовательной записью в этот регистр двух ключей: KEY1=0x45670123 и KEY2=0xCDEF89AB.

После записи ключа KEY1, а затем ключа KEY2 в данный регистр, бит LOCK будет сброшен, блок FPFC разблокирован, а регистр FLASH_CR станет доступным для записи.

После записи данных во флэш-память нужно программно установить бит LOCK. При необходимости снова записать данные, придется опять выполнить процедуру разблокировки с помощью ключей KEY1 и KEY2.

В случае ошибки при записи ключей, блок FPFC заблокируется, и повторная запись верных ключей не сможет разблокировать его. В таком случае поможет только сброс микроконтроллера, после чего снова будет возможна процедура разблокировки блока FPFC.

Таблица 2. Карта регистров блока FPFC

Сдвиг	Регистр	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0									
0x000	FLASH_ACR	Резерв																																		HLF	CYA	Резерв				
	Исх. значение	0																																								
0x004	FLASH_KEYR	FKEYR[31:0]																																								
	Исх. значение	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x									
0x008	FLASH_OPTKEYR	OPTKEYR[31:0]																																								
	Исх. значение	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x									
0x00C	FLASH_SR	Резерв																																		EOP	WRP	TERR	Резерв	PGERR	ERLYBSY	BSY
	Исх. значение	0																																		0			0	0	0	
0x010	FLASH_CR	Резерв																				EOPIE	Резерв	ERRIE	OPTWRE	Резерв	LOCK	STRT	OPTER	OPTPG	Резерв	MER	PER	PG								
	Исх. значение	0																						0	0		1	0	0	0		0	0	0								
0x014	FLASH_AR	FAR[31:0]																																								
	Исх. значение	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x									
0x018	Резерв	Резерв																																								
0x01C	FLASH_OBR	Резерв								Data1								Data2								Не используется								nRST_STDBY	nRST_STOP	WDG_SW	RDPR	OPTERR				
	Исх. значение									1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1					
0x020	FLASH_WRP	WRP[31:0]																																								
	Исх. значение	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1							

Регистр FLASH_OPTKEYR используется для получения доступа к установке и сбросу разрядов в блоке опционных байт, которые изначально доступны только для чтения. Для получения возможности изменения данных в области опционных байт необходимо записать в регистр FLASH_OPTKEYR такую же последовательность ключей, что и для регистра FLASH_KEYR. После записи правильных ключей в регистре FLASH_CR будет установлен бит OPTWRE, который разрешает запись данных в область опционных байт флэш-памяти.

Регистр FLASH_SR содержит следующие специальные разряды:

- разряд 5 EOP (End of Operation) устанавливается аппаратно при успешном завершении операции записи или стирании флэш-памяти;
- разряд 4 WRPRERR (Write Protection Error) устанавливается при попытке программирования защищенной от записи области памяти;
- разряд 2 PGERR (Programming Error) устанавливается при программировании, если данные по заданному адресу отличны от значения 0xFFFF, то есть не стерты 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;

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
    }
}
```

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

Регистры FLASH_OBR и FLASH_WRP 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 FPES.

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 нее 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 определенное время, поэтому во время выполнения таких операций реакция микроконтроллера будет замедленной.

В качестве тренировки можно развить данную программу, дополнив ее 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.
2. Вальпа О. Современные 32-разрядные ARM-микроконтроллеры серии STM32: блок резервных данных ВКР. Современная электроника. № 3. 2014. **СНУ**

* Статья перепечатана из журнала «Современная электроника», № 8, 2013 г., с разрешения редакции, тел. +7 (495) 232-00-87, www.soel.ru



АВТОРИЗОВАННЫЙ ДИСТРИБ'ЮТОР



Авторизований дистриб'ютор STMicroelectronics, NXP та Vishay в Україні

м. Київ, вул. Бориспільська, 9Д
тел. +38 (044) 567-44-48, (067) 219-27-86

+38 (044) 566-79-03
info@mastek.com.ua
www.mastek.com.ua