

Современные 32-разрядные ARM-микроконтроллеры серии STM32.

Дополнительная энергонезависимая память с интерфейсом I²C

Олег Вальпа (sandh@narod.ru)

В статье описана организация дополнительной энергонезависимой памяти с интерфейсом I²C для 32-разрядных ARM-микроконтроллеров серии STM32 от компании STMicroelectronics.

ВВЕДЕНИЕ

Память для любого микроконтроллера является важным обязательным компонентом. От объёма оперативной памяти зависит количество организуемых в программе переменных, массивов данных и стека подпрограмм. Однако оперативная память не способна хранить информацию без питания, т.е. при выключении устройства все данные теряются. Для постоянного хранения данных применяется энергонезависимая память. В качестве неё может выступать память программ, память внешних носителей или электрически перепрограммируемая постоянная память EEPROM. Микроконтроллеры серии STM32 [1] не имеют встроенной памяти EEPROM, тем не менее её можно организовать с помощью подключения внешней микросхемы энергонезависимой памяти.

ОБЗОР МИКРОСХЕМ ПАМЯТИ EEPROM

В настоящее время наиболее распространёнными компонентами дополнительной памяти EEPROM являются микросхемы серии 24Cxx с последовательным интерфейсом I²C, которые выпускаются многими производителями. Рассмотрим линейку таких микро-

Таблица 1. Типы микросхем памяти

Тип микросхемы	Объём памяти	
	Бит	Байт
AT24C01	1024	128
AT24C02	2048	256
AT24C04	4096	512
AT24C08	8192	1024
AT24C16	16 384	2048
AT24C32	32 768	4096
AT24C64	65 536	8192
AT24C128	131 072	16 384
AT24C256	262 144	32 768
AT24C512	524 288	65 536
AT24C1024	1 048 576	131 072

схем из семейства AT24Cxx от компании Atmel [2]. В таблице 1 приведена информация об объёме памяти микросхем данной серии. Доступ к памяти микросхем организован побайтно.

Структурная схема чипа памяти с описанием назначения внутренних узлов и связей между ними приведена на рисунке 1.

Данные микросхемы памяти EEPROM оптимизированы для использования во многих промышленных и коммерческих приложениях, где важны малая потребляемая мощность и работа при низком напряжении питания. Ниже приведены основные технические характеристики описываемых микросхем:

- напряжение питания в зависимости от исполнения – 1,8...5,5 В;
- ток потребления – не более 3 мА;
- двухпроводной последовательный интерфейс связи I²C;

- триггеры Шмитта на входах для подавления шума;
- двунаправленный протокол передачи данных;
- совместимость с частотами синхронизации: 1 МГц при 5 В, 400 кГц при 2,7 В и 100 кГц при 1,8 В;
- вход защиты от записи для аппаратной и программной защиты данных;
- режим записи 64-байтных страниц с возможностью частичной записи страницы;
- самосинхронизированный цикл записи длительностью не более 5 мс;
- высокая надёжность;
- количество циклов записи – не менее 1 млн;
- гарантированный срок хранения данных – не менее 40 лет;
- диапазон рабочих температур –40...+85°C;
- корпуса без содержания свинца и галогенных соединений;
- типы корпусов: 8-выводные JEDEC PDIP, JEDEC и EIAJ SOIC, MAP, TSSOP, SAP и dBG2.

Расположение выводов микросхем для всех типов корпусов показано на

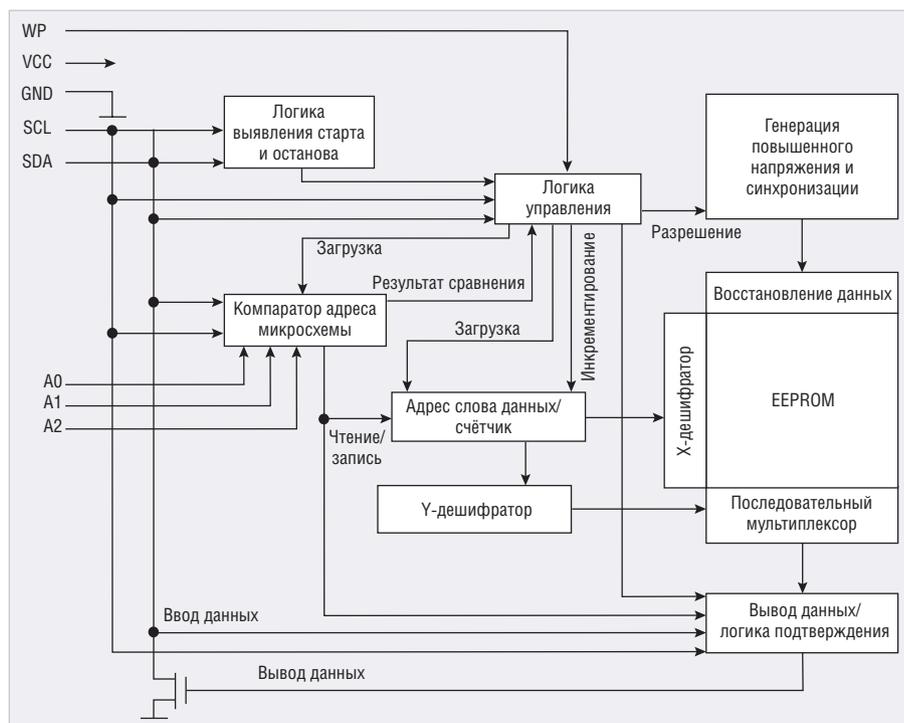


Рис. 1. Структурная схема чипа памяти

рисунке 2. В таблице 2 приведено описание выводов микросхем.

Микросхема содержит адресные входы, позволяющие назначать её адрес для интерфейса I²C. Таким образом, к одной последовательной шине можно подключить до 8 микросхем. Адресация микросхем памяти для интерфейса I²C приведена в таблице 3.

Младший разряд адреса микросхемы R/W является битом выбора операции чтения/записи. Операция чтения инициируется, когда данный разряд имеет высокий уровень; если же этот разряд имеет низкий уровень, то инициируется операция записи.

Вывод SDA используется для двунаправленной последовательной передачи данных. Он является выводом с открытым стоком и может быть соединён с несколькими аналогичными выводами с открытым стоком или коллатором.

Вход SCL используется для синхронизации последовательных данных посредством интерфейса I²C.

Линии SDA и SCL необходимо обязательно согласовать с напряжением источника питания с помощью резисторов. На частоте тактирования 100 кГц используются резисторы номиналом 10 кОм, а на частоте тактирования 400 кГц и выше – 2 кОм.

Вход WP позволяет блокировать запись данных при подключении этого входа к цепи питания VCC. При этом все команды записи в память игнорируются. Если данный вывод подключить к линии GND, то операции записи будут разрешены.

Схема подключения чипа памяти к микроконтроллеру типа STM32 приведена на рисунке 3.

ОПЕРАЦИИ ЗАПИСИ

Для выполнения операции записи данных в микросхему памяти предварительно передаются адрес микросхемы и два 8-разрядных слова адреса, приём которых подтверждается нулевым потенциалом на линии SDA со стороны микросхемы памяти. Затем при поступлении импульсов синхронизации микросхема принимает первый байт данных, после чего вновь формирует нулевой потенциал на SDA. Далее микроконтроллер прерывает последовательность записи передачей условия «СТОП», как показано на рисунке 4. В этот момент в микросхеме EEPROM активизируется внутренний синхронизируемый цикл записи

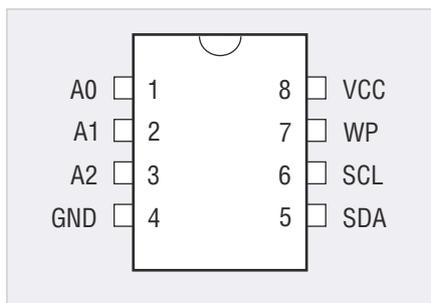


Рис. 2. Расположение выводов микросхем

Таблица 2. Описание выводов микросхем памяти

Вывод	Обозначение	Описание
1	A0	Адресный вход 0
2	A1	Адресный вход 1
3	A2	Адресный вход 2
4	GND	Общий
5	SDA	Сигнал данных
6	SCL	Сигнал синхронизации
7	WP	Вход защиты записи
8	VCC	Напряжение питания

Таблица 3. Адресация микросхем памяти

Микросхема	Разряды адреса							
	7	6	5	4	3	2	1	0
AT24CXX	1	0	1	0	A2	A1	A0	R/W

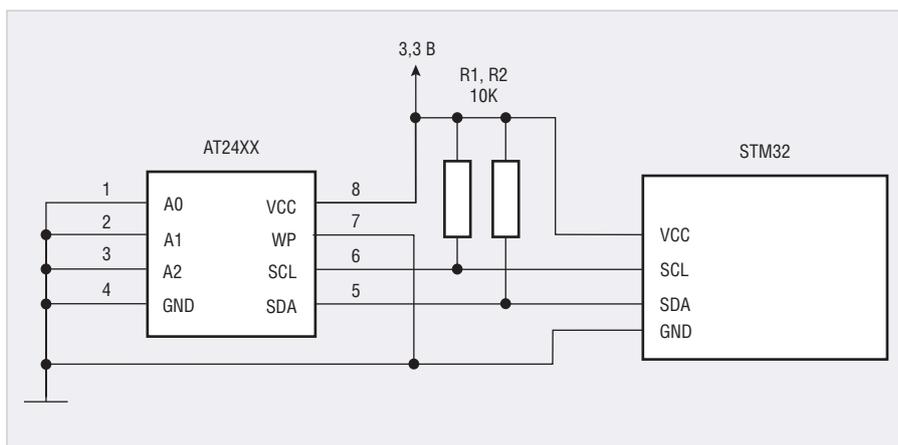


Рис. 3. Схема подключения чипа памяти

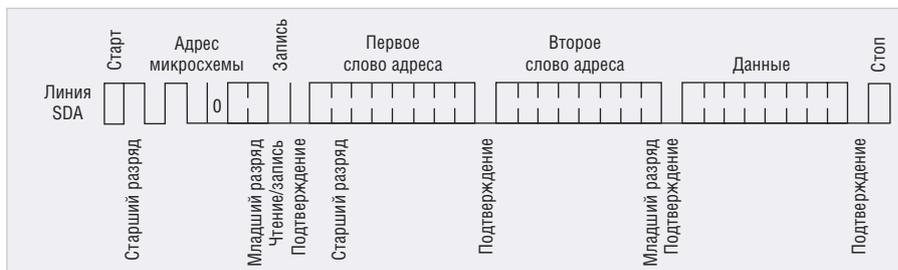


Рис. 4. Запись байта

данных в энергонезависимую память. В процессе выполнения цикла записи и до её завершения все входы микросхемы памяти отключаются и она не реагирует на внешние запросы.

Микросхемы EEPROM с объёмом памяти более 128 Кбит поддерживают запись 64-байтных страниц. Страничная запись инициируется таким же способом, что и побайтная, за исключением того, что микроконтроллер не отправляет условие останова после записи первого байта – вместо этого он может передавать дополнительно до 63 байт данных. После приёма каждого последующего байта данных микросхема подтверждает операцию нулевым потенци-

алом на линии SDA. Микроконтроллер прекращает последовательность страничной записи передачей условия останова, как показано на рисунке 5.

После приёма каждого байта данных в микросхеме памяти инкрементируются младшие 6 бит слова адреса. Старшие биты слова адреса не инкрементируются. Если в результате внутреннего инкрементирования слова адреса достигается его граничное значение, то следующий байт будет приниматься в начало текущей страницы. Если в микросхему EEPROM передаётся более 64 байт данных, то ранее переданные данные будут заменены вновь поступившими. При выполнении инкрементирования в про-

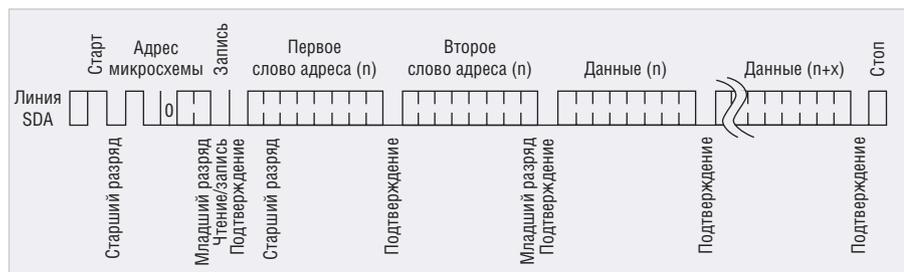


Рис. 5. Запись страницы

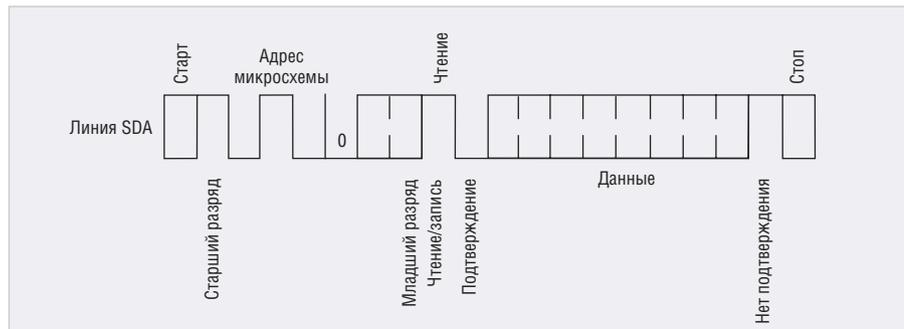


Рис. 6. Чтение по текущему адресу

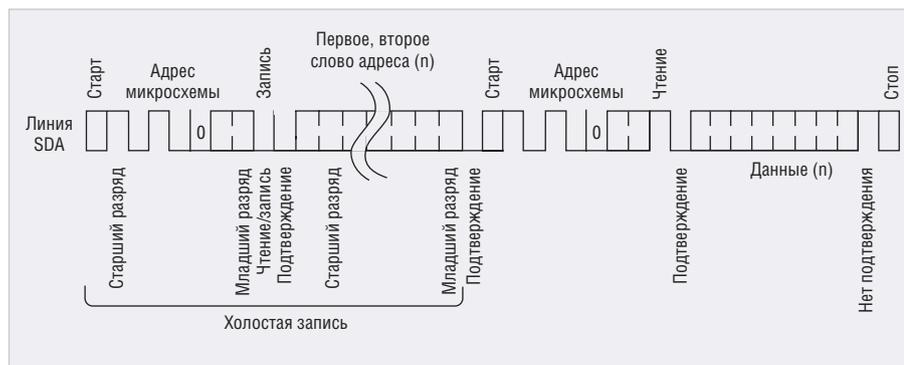


Рис. 7. Чтение по произвольному адресу

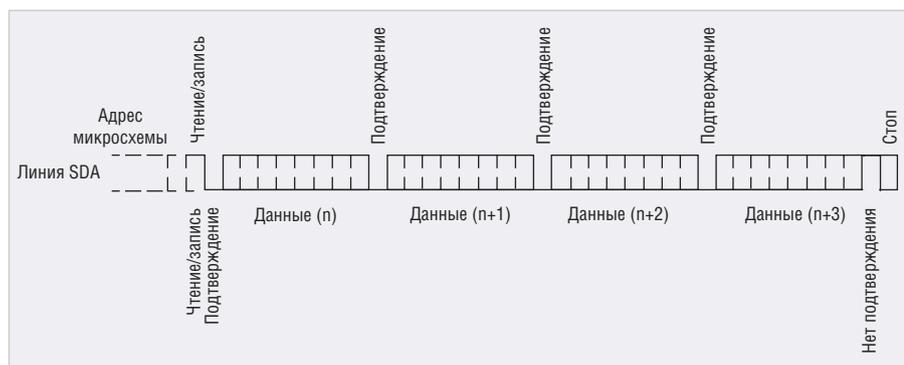


Рис. 8. Упорядоченное чтение

пессе записи последнего байта текущей страницы устанавливается адрес первого байта этой же страницы.

Сразу после инициирования внутреннего синхронизируемого цикла записи и отключения входов микросхемы EEPROM можно начать опрос подтверждения. Для этого необходимо отправить условие старта после адресного слова. Бит чтения/записи опреде-

ляет выполнение желаемой операции. Микросхема EEPROM отвечает нулевым потенциалом на SDA только при завершении внутреннего цикла записи, тем самым позволяя продолжить последовательность записи данных.

ОПЕРАЦИИ ЧТЕНИЯ

Операции чтения иницируются тем же способом, что и операции записи, за

исключением того, что бит выбора операции чтения/записи в адресном слове равен единице. Поддерживаются три вида операций чтения: чтение по текущему адресу, чтение по произвольному адресу и упорядоченное чтение.

При чтении по текущему адресу внутренний счётчик адреса слова данных хранит адрес, который использовался при последней операции чтения или записи, увеличенный на 1. Данный адрес остаётся действительным до тех пор, пока на микросхему подано питание. После чтения последнего байта последней страницы и автоматического инкрементирования адреса устанавливается адрес первого байта первой страницы.

Сразу после приёма адреса микросхемы с битом выбора чтения/записи, равным 1, и подтверждения приёма со стороны микросхемы EEPROM передаётся байт данных по текущему адресу. При этом микроконтроллер не отвечает микросхеме нулевым потенциалом на SDA, а передаёт условие останова, как показано на рисунке 6.

Для чтения по произвольному адресу необходимо выполнить «холостую» процедуру загрузки адреса микросхемы и данных. Как только микросхема примет адресное слово микросхемы, адресное слово данных и отправит подтверждение, микроконтроллер должен сгенерировать новое условие старта. После этого необходимо инициировать операцию чтения по текущему адресу путём отправки адреса микросхемы с установленным в единичное состояние битом выбора чтения/записи. Микросхема памяти подтвердит адрес микросхемы и последовательно передаст слово данных. При этом микроконтроллер не ответит нулевым потенциалом на SDA, а должен будет сгенерировать условие останова, как показано на рисунке 7.

Упорядоченное чтение иницируется после операции чтения по текущему адресу или после операции чтения по произвольному адресу. После приёма микроконтроллером байта данных он отвечает соответствующим подтверждением. Когда микросхема памяти принимает подтверждение, выполняется инкрементирование адреса слова данных и передаётся очередной байт данных. При достижении границы адресного пространства адрес данных переходит в начальное состояние и упорядоченное чтение продолжается. Упорядоченное чтение прекратится, когда микроконтроллер не ответит нулевым потенциалом

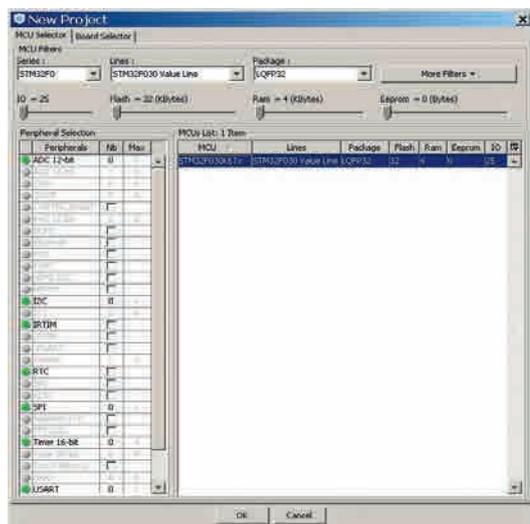


Рис. 9. Выбор микроконтроллера в STM32CubeMX

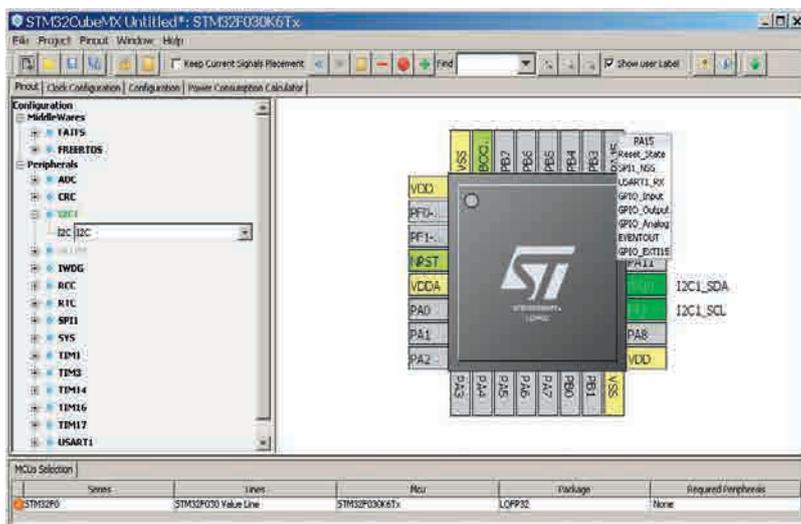


Рис. 10. Конфигурирование портов в STM32CubeMX

на линии SDA и генерирует условие останова, как показано на рисунке 8.

ПРИМЕРЫ ПРОГРАММ

Создать программу для организации дополнительной энергонезависимой памяти на микросхеме EEPROM посредством интерфейса I²C можно с помощью генератора кода STM32CubeMX компании STMicroelectronics [1]. После запуска генератора кода STM32CubeMX необходимо создать в нём новый проект и выбрать тип микроконтроллера, например STM32F030K6Tx, как показано на рисунке 9.

На вкладке *Pinout* в разделе *Peripherals* раздела *I2C1* необходимо выбрать режим шины I²C, при этом генератор кода STM32CubeMX автоматически задействует порт PA9 для сигнала SCL и порт PA10 для сигнала SDA согласно внутренней организации микроконтроллера. Далее следует настроить один порт, например PA15, как GPIO_Output для подключения к нему светодиода, индицирующего процессы обращения к микросхеме памяти. Делается это одиночным кликом по изображению вывода порта и соответствующему режиму из разворачиваемого списка, как показано на рисунке 10. Перейдя на вкладку *Configuration*, можно проверить параметры модуля *I2C1*.

Режим скорости выбран стандартным (Standard Mode) с частотой 100 кГц, что соответствует номиналу подключённых резисторов 10 кОм. Генератор кода STM32CubeMX позволяет также выбрать режим Fast Mode с частотой 400 кГц – в этом случае нужно применять резисторы номиналом 2 кОм. Микросхема памяти допускает работу на такой частоте при напряжении питания выше 2,5 В.

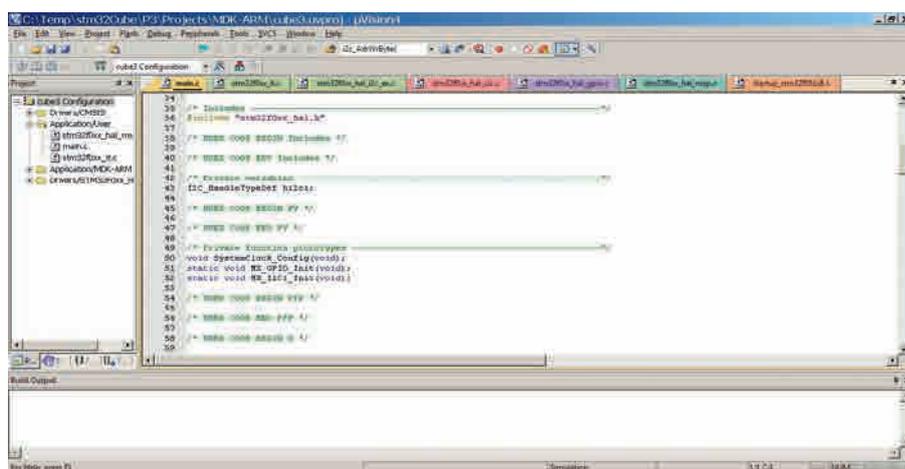


Рис. 11. Окно среды разработки KEIL

Остальные настройки можно оставить без изменений и сохранить проект в каталог с новым именем. Далее следует сгенерировать код программы, например для MDK-ARM KEIL [3], с помощью меню *Project*. После завершения процесса генератор кода STM32CubeMX предложит открыть полученный код в среде разработки кода с файлами проекта (см. рис. 11).

Все сгенерированные с помощью STM32CubeMX файлы записываются и хранятся в каталоге *Src* созданного проекта. Основной файл *main.c* содержит в себе функции инициализации и основное тело программы. Вспомогательный файл *stm32f0xx_it.c* служит для обработки прерываний. В файле *stm32f0xx_hal_msp.c* можно увидеть инициализацию портов PA9 и PA10 интерфейса I²C с помощью функций HAL.

ЗАПИСЬ И ЧТЕНИЕ ПАМЯТИ EEPROM

Чтобы дополнить сгенерированный файл программы *main.c* некоторыми

переменными и командами для организации обмена данными с микросхемой памяти EEPROM, в разделе пользовательских переменных следует инициализировать однобайтный буфер обмена *buf*, адрес микросхемы EEPROM *I2C1_DEV_ADR* со значением 0x50 для интерфейса I²C и ячейку памяти EEPROM *ADR* с адресом 0x01 (см. листинг 1).

Запись новых строк программы всегда нужно осуществлять в строго отведённых местах между строками `/* USER CODE BEGIN...*/` и `/* USER CODE END...*/`. Это позволит сохранить новые строки программы при повторной генерации кода с помощью STM32CubeMX.

Для записи и чтения одного байта памяти EEPROM в бесконечном цикле необходимо ввести в главную функцию *main* строки из листинга 2.

В результате работы написанной программы будет произведена запись одного байта в ячейку памяти EEPROM с последующим чтением этой же ячейки памяти через 5 с. Если прочитанный байт совпадёт с записанным байтом,

Листинг 1

```

/* main.c */
/* USER CODE BEGIN PV */
uint8_t buf[]='v'; // Сохранить в буфере символ 'v'
#define I2C1_DEV_ADR 0x50 // Адрес микросхемы EEPROM=0101 0000
A2=A1=A0=0
#define EEPROM_ADR 0x01 // Адрес ячейки памяти
/* USER CODE END PV */

```

Листинг 2

```

/* USER CODE BEGIN 3 */
/* Infinite loop */
while (1)
{
// Записать в EEPROM данные буфера по адресу ячейки памяти
HAL_I2C_Mem_Write(&hi2c1, (uint16_t) I2C1_DEV_ADR<<1, EEPROM_ADR, 1,
buf, 1, 5);
osDelay(10); // Ожидание записи
xBuffer[0]=0; // Обнулить буфер
osDelay(5000); // Задержка на 5 секунд
// Чтение памяти EEPROM по адресу ячейки памяти
HAL_I2C_Mem_Read(&hi2c1, (uint16_t) I2C1_DEV_ADR<<1, EEPROM_ADR, 1,
buf, 1, 5);
if(xBuffer[0] == 'v') // Если прочитанные данные совпали с записанными
{
HAL_GPIO_WritePin(GPIOC, GPIO_PIN_15, GPIO_PIN_SET); // Включить
светодиод
}
}
/* USER CODE END 3 */

```

будет произведено включение светодиода, подключённого к выводу PA15.

Функции `HAL_I2C_Mem_Write` и `HAL_I2C_Mem_Read` предназначены для работы с памятью, поэтому нет необходимости использовать общие функции

работы с шиной I²C `HAL_I2C_Master_Transmit` и `HAL_I2C_Master_Receive`. Адрес EEPROM передаётся в программе функции со смещением влево на 1 разряд `I2C1_DEVICE_ADDRESS<<1`, т.к. он является 7-битным, а младший раз-

ряд адреса устройства по спецификации I²C отвечает за операцию записи/чтения.

В функциях `HAL_I2C_Mem_Write` и `HAL_I2C_Mem_Read` последним аргументом является время ожидания, за которое микроконтроллер стабильно будет получать реакцию от EEPROM на его запросы. Задержка в 10 мс `osDelay(10)` после функции `HAL_I2C_Mem_Write` необходима для совершения операции записи в EEPROM. Несмотря на то что в описании этой микросхемы указано время записи 5 мс, его лучше увеличить, чтобы гарантировать стабильность операции.

Описанные выше функции записи и чтения данных в память EEPROM можно использовать для написания других программ.

В качестве готового устройства для проверки и отработки программ подойдёт любая отладочная плата, например из семейства Discovery.

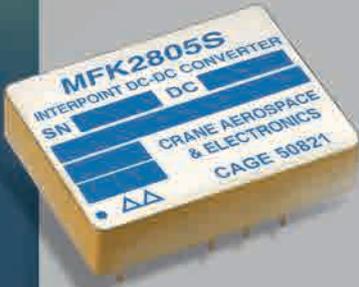
ЛИТЕРАТУРА

1. www.st.com
2. www.atmel.com
3. www.keil.com



НОВИНКА!

25-ваттные DC/DC-преобразователи Interpoint® MFK Series™



- Широкий диапазон входного напряжения от 16 до 50 В
- Удельная мощность до 2570 Вт/дм³
- 11 значений выходного напряжения от 1,8 до 28 В
- Одно- и двухканальные модели
- КПД до 87%
- Трансформаторная развязка в контуре обратной связи
- Диапазон рабочих температур от -55 до +125°C
- Обширный ряд сервисных функций



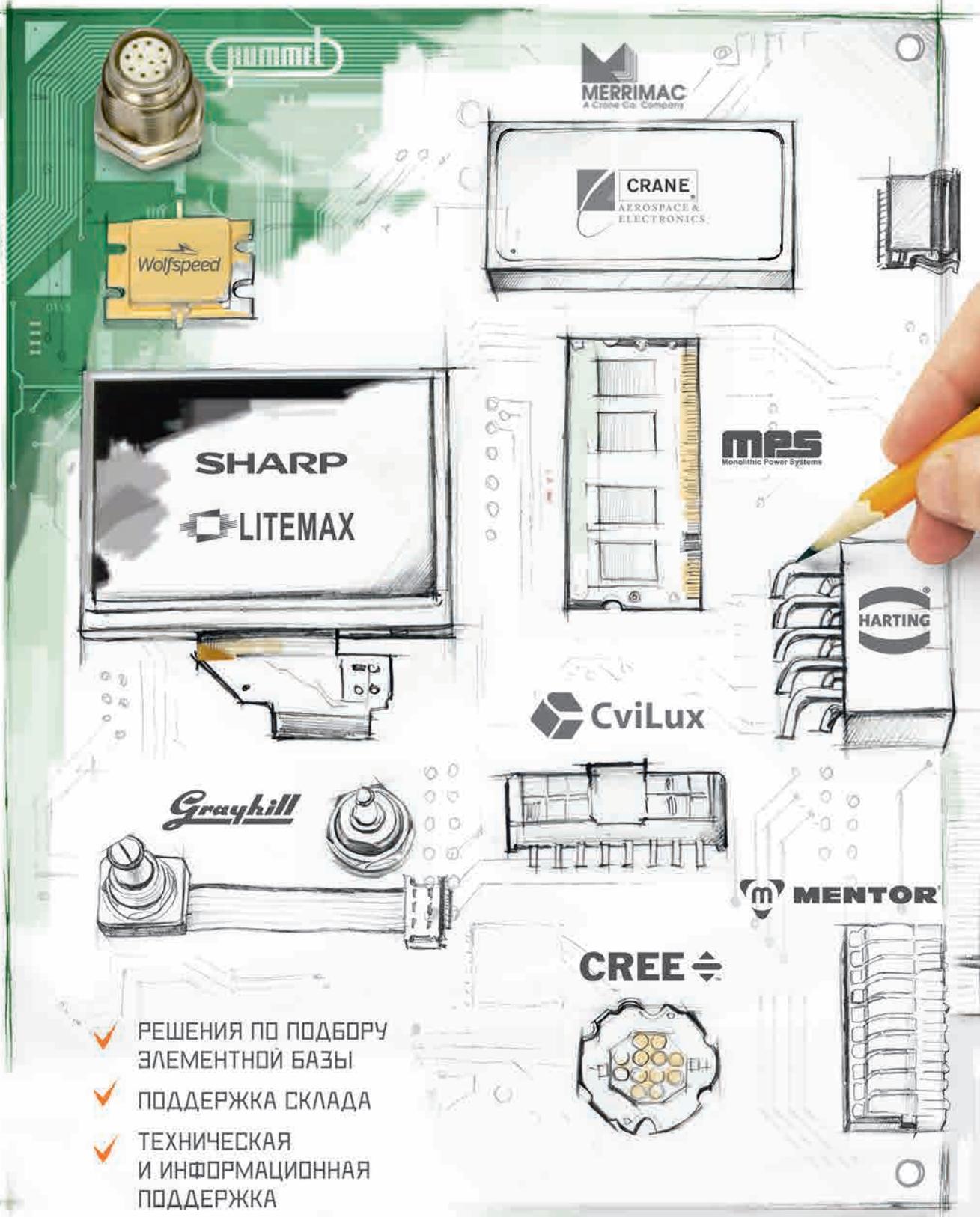
www.inter-pro.net

info@inter-pro.net

INTERPRO



Реклама



- ✓ РЕШЕНИЯ ПО ПОДБОРУ ЭЛЕМЕНТНОЙ БАЗЫ
- ✓ ПОДДЕРЖКА СКЛАДА
- ✓ ТЕХНИЧЕСКАЯ И ИНФОРМАЦИОННАЯ ПОДДЕРЖКА

