

# Российские микроконтроллеры с ядром Cortex M3 и пример реализации проекта

Мария ЕРМАК

Статья посвящена краткому обзору серии контроллеров 1986VE9x на ядре Cortex-M3 производства ЗАО «ПКК Миландр», средствам разработки и отладки для него.

## Введение

Популярность 32-битной микропроцессорной архитектуры с сокращенным набором команд (RISC) фирмы ARM Limited растет с каждым днем, поэтому все большее число компаний разрабатывают микроконтроллеры на этом ядре. Одна из последних версий ядра — Cortex-M3. Пока очень немногие из производителей МК освоили это ядро. В настоящее время на российском рынке в основном представлены микроконтроллеры с ядром ARM Cortex-M3 компаний STMicroelectronics (семейство STM32Fxxx) и NXP (семейство LPC17xx). И поэтому особенно приятно, что к этим зарубежным компаниям можно добавить отечественный дизайн-центр ЗАО «ПКК Миландр» [1, 2].

Следует отметить, что срок окончания ОКР по разработке МК серии 1986VE9x — сентябрь 2010 года, но уже сейчас заинтересованным компаниям предоставляются образцы МК, отладочные средства для них и полная техническая поддержка.

Стоит обратить внимание на тот факт, что МК были разработаны для аппаратуры специального назначения, требующей большого запаса по стойкости (то есть способности микросхемы работать в агрессивной окружающей среде), и с широким рабочим температурным диапазоном. Версии микроконтроллеров для гражданского рынка будут представлены компанией в ближайшее время.

## Основные характеристики микроконтроллера

Устройства серии 1986VE9x являются микроконтроллерами со встроенной Flash-памятью программ и построены на базе высокопроизводительного процессорного RISC-ядра ARM Cortex-M3 (производительность — 1,25 DMIPS/МГц (Dhrystone 2.1) при нулевой задержке между обращениями к памяти). Здесь необходимо добавить, что максимальная тактовая частота работы

рассматриваемого микроконтроллера — 80 МГц, в то время как максимальная частота выборки данных (в данном случае команд) из Flash-памяти составляет 28,6 МГц (соответствует 35 нс). Для обеспечения максимального быстродействия при существующих технологических возможностях необходимо применять аппаратные решения для ускорения процесса обращения к Flash-памяти. В ряде микроконтроллеров, например, тех, что производят фирмы ЗАО «ПКК Миландр» и STMicroelectronics, для этого реализован специальный буфер «шириной» 64 бита. Но и в него выборка 32-разрядных команд осуществляется фактически с частотой  $28,6 \times 2 = 57,2$  МГц. Таким образом, для работы при частоте свыше 57,2 МГц необходимо искусственно вводить задержку между обращениями к памяти. При частоте ниже 57,2 МГц задержка между считываниями из памяти является нулевой.

В ядре Cortex-M3 реализованы следующие функции:

- блок аппаратной защиты памяти от несанкционированного доступа;
- умножение за один цикл (табл. 1);
- аппаратная реализация деления (32 бита/32 бита).

Микроконтроллеры работают на тактовой частоте до 80 МГц и содержат 128 кбайт Flash-памяти программ и 32 кбайта ОЗУ. Контроллер внешней системной шины позволяет работать с внешними микросхемами статического ОЗУ и ПЗУ, NAND Flash памятью и другими периферийными устройствами [3].

**Таблица 1.** Зависимость количества циклов синхрочастоты, требуемых для выполнения операции умножения, при различных разрядностях операндов и результата

Источник	Назначение	Циклы
16 бит × 16 бит	32 бит	1
32 бит × 16 бит	32 бит	1
32 бит × 32 бит	32 бит	1
32 бит × 32 бит	64 бит	3–7

Встроенные RC-генераторы HSI (8 МГц) и LSI (40 кГц) и внешние генераторы HSE (2–16 МГц) и LSE (32 кГц), а также две схемы умножения тактовой частоты PLL для ядра и USB-интерфейса позволяют гибко настраивать скорость работы периферийных блоков микроконтроллеров.

Процессор Cortex-M3 выполнен по Гарвардской архитектуре, которая подразумевает использование отдельных шин данных и инструкций. Они называются шиной Dcode и Icode соответственно. Также имеется дополнительная системная шина, которая предоставляет доступ к области системного управления. У встроенной отладочной системы процессора Cortex имеется еще одна дополнительная шинная структура, которая называется локальной шиной устройств ввода/вывода. Системная шина и шина данных ядра подключаются к внешним (относительно ядра) блокам микроконтроллера через набор высокоскоростных шин, называемых матрицей шин. Для разрешения конфликтов при запросе на доступ к шине используются арбитраж. Блоки, подключенные к шине, могут быть активными датчиками шины — «мастерами». Матрица шин образует несколько параллельных соединений между шинами ядра Cortex и другими внешними шинными «мастерами», такими как каналы DMA, статическое ОЗУ и устройства ввода/вывода. Если два шинных «мастера» (например, ядро Cortex и канал DMA) предпринимают попытку доступа к одному и тому же устройству ввода/вывода, то вступит в действие внутренний арбитр, который разрешит конфликт, предоставив доступ к шине тому, кто имеет наивысший приоритет.

Таким образом, архитектура системы памяти за счет матрицы системных шин позволяет минимизировать возможные конфликты при работе системы и повысить общую производительность.

Контроллер DMA дает возможность ускорить обмен информацией между ОЗУ и периферией без участия процессорного ядра.

Встроенный регулятор для формирования питания внутреннего цифровой части форми-

рует напряжение 1,8 В и не требует дополнительных внешних элементов. Таким образом, для работы микроконтроллеров достаточно одного внешнего напряжения питания в диапазоне от 2,2 до 3,6 В. Также в микроконтроллерах реализована возможность работы кристалла от внешней батареи, что позволяет иметь резервированное питание, на которое микроконтроллер автоматически переключается при пропадании основного напряжения питания. При этом будут сохранены специальные флаги и не произойдет нарушения работы часов реального времени. Встроенные детекторы напряжения питания могут отслеживать уровень внешнего основного питания и уровень напряжения питания на батарее. Аппаратные схемы сброса по снижению уровня питания позволяют исключить некорректное выполнение кода программы и неправильное поведение микроконтроллера в целом при выходе уровня напряжения питания за допустимые пределы.

Для возможности реализации приложений, критичных к уровню энергопотребления, в микроконтроллерах существуют следующие режимы:

- Sleep;
- Deep sleep;
- Standby.

Микроконтроллеры обладают богатой периферией, набор которой зависит от модели МК (табл. 2).

Максимальным количеством периферийных устройств обладает МК 1986BE91T:

- Цифровые модули:
  - контроллер прямого доступа в память с функциями передачи периферия – память, память – память;
  - два контроллера CAN-интерфейса;
  - контроллер USB-интерфейса с режимами работы Device и Host;
  - контроллеры интерфейсов USART, SPI, I<sup>2</sup>C;
  - до 96 пользовательских линий ввода/вывода;

- три 16-разрядных таймера с 4 каналами схем захвата и ШИМ с функциями формирования «мертвой зоны» и аппаратной блокировки;
- системный 24-разрядный таймер;
- два сторожевых таймера.

Реализация двух сторожевых таймеров позволяет расширить возможности по определению сбоя в выполнении кода программы МК. Один из них — оконный сторожевой таймер, который необходимо обновлять с определенной частотой. Другой — независимый сторожевой таймер, который синхронизируется отдельным генератором, не связанным с основной системной синхронизацией.

Аналоговые модули:

- два 12-разрядных АЦП (до 16 каналов), измеряемый диапазон напряжений от 0 до 3,6 В;
- температурный сенсор;
- двухканальный 12-разрядный ЦАП;
- встроенный компаратор.

Для отладки устройств на базе МК 1986BE91T в них реализованы 2 интерфейса:

- последовательный отладочный интерфейс SWD;
- последовательный отладочный интерфейс JTAG.

### Средства разработки и отладки

Для разработки ПО для МК серии 1986BE9x можно использовать три разных пакета инструментальных средств:

- CodeMaster-ARM — интегрированная среда разработки компании «Фитон».
- Keil uVision — интегрированная среда разработки компании Keil.
- IAR Embedded Workbench — интегрированная среда разработки компании IAR Systems (поддержка в стадии разработки).

Программирование осуществляется либо с помощью стандартного программатора фирмы Keil (ULINK2), либо с помощью программатора, разработанного фирмой «Фитон».

Также для ознакомления с МК 1986BE91T разработана отладочная плата, позволяющая максимально использовать периферию и возможности МК [4]. Внешний вид платы показан на рис. 1. Описание элементов платы представлено на рис. 2.

Для любого разработчика устройств на базе МК среда Keil uVision не нуждается в представлении, так как она поддерживает огромное количество микроконтроллеров и существует уже давно. Наверняка многие из разработчиков с ней уже сталкивались. Намного больший интерес представляет среда CodeMaster-ARM (фирма «Фитон»).

### CodeMaster-ARM

CodeMaster-ARM — набор программно-аппаратных средств, предназначенный для разработки и отладки систем на базе микро-



Рис. 1. Внешний вид отладочной платы для МК серии 1986BE9x

контроллеров ARM7/ARM9/Cortex-Mx, в том числе систем на базе первых отечественных микроконтроллеров с ядром Cortex-M3 серии 1986BE9x, разработанных компанией ЗАО «ПКК Миландр» [5, 6, 7].

Среда разработки CodeMaster-ARM обладает всеми необходимыми средствами для реализации проектов на микроконтроллерах с ядром ARM. Особое внимание следует уделить тому факту, что среда разработана в России, поэтому пользователь в любой момент может получить качественную техническую поддержку и задать интересующие его вопросы. Неоспоримым достоинством также следует считать возможность выбора языка интерфейса между русским и английским.

Более подробная информация о среде CodeMaster-ARM, а также руководство пользователя могут быть получены на сайте фирмы «Фитон» [8].

### Пример реализации простого проекта в среде CodeMaster-ARM

В комплекте со средой CodeMaster-ARM поставляются примеры проектов для МК 1986BE9x, что несомненно облегчает разработку собственных проектов.

Рассмотрим пример создания простого проекта в среде CodeMaster-ARM для ознакомления с функциональными возможностями микроконтроллера 1986BE91T и с демонстрационной платой. Предположим, нужно оцифровывать некоторый сигнал и подавать его на выход ЦАП. Причем данные необходимо выводить на LCD-дисплей и посылать в соп-порт ПК. Общий алгоритм работы представлен на рис. 3.

Первое, с чего начинается работа МК, — это инициализация тактовых сигналов и выбор их для различной периферии.

На отладочной плате есть генератор 8 МГц, его и будем использовать как основной источник тактовой частоты. Согласно документации тактовая частота АЦП не должна превышать 24 МГц. Поэтому для его тактирования используем PLL с коэффициентом умножения 3. В то же время для работы с дисплеем будем использовать внешнюю системную шину, но время цикла чтения/записи для дисплея MT-12864J [9] должно быть не меньше 1 мкс, поэтому для его тактирования используем внутренний генератор

Таблица 2. Сводная таблица микроконтроллеров серии 1986BE9x

Периферия	1986BE91T	1986BE92U	1986BE93U
Корпус	4229.132-3 (132 вывода)	H18.64-1B (64 вывода)	H16.48-1B (48 выводов)
Ядро	ARM Cortex-M3		
ПЗУ	128 кбайт Flash		
ОЗУ	32 кбайт		
Питание	2,2–3,6 В		
Частота	80 МГц		
Температура	–60...+125 °С		
USER IO	96	45	31
USB	Device и Host FS (до 12 Мбит/с), встроенный PHY		
UART	2	2	2
CAN	2	2	1
SPI	2	2	1
I <sup>2</sup> C	1	1	Нет
ADC 12 разрядов 1 Мвыб/с	16 каналов	8 каналов	5 каналов
DAC 12 разрядов	2	1	1
Компаратор	3 входа	2 входа	2 входа
Внешняя шина	32 разряда	8 разрядов	8 разрядов

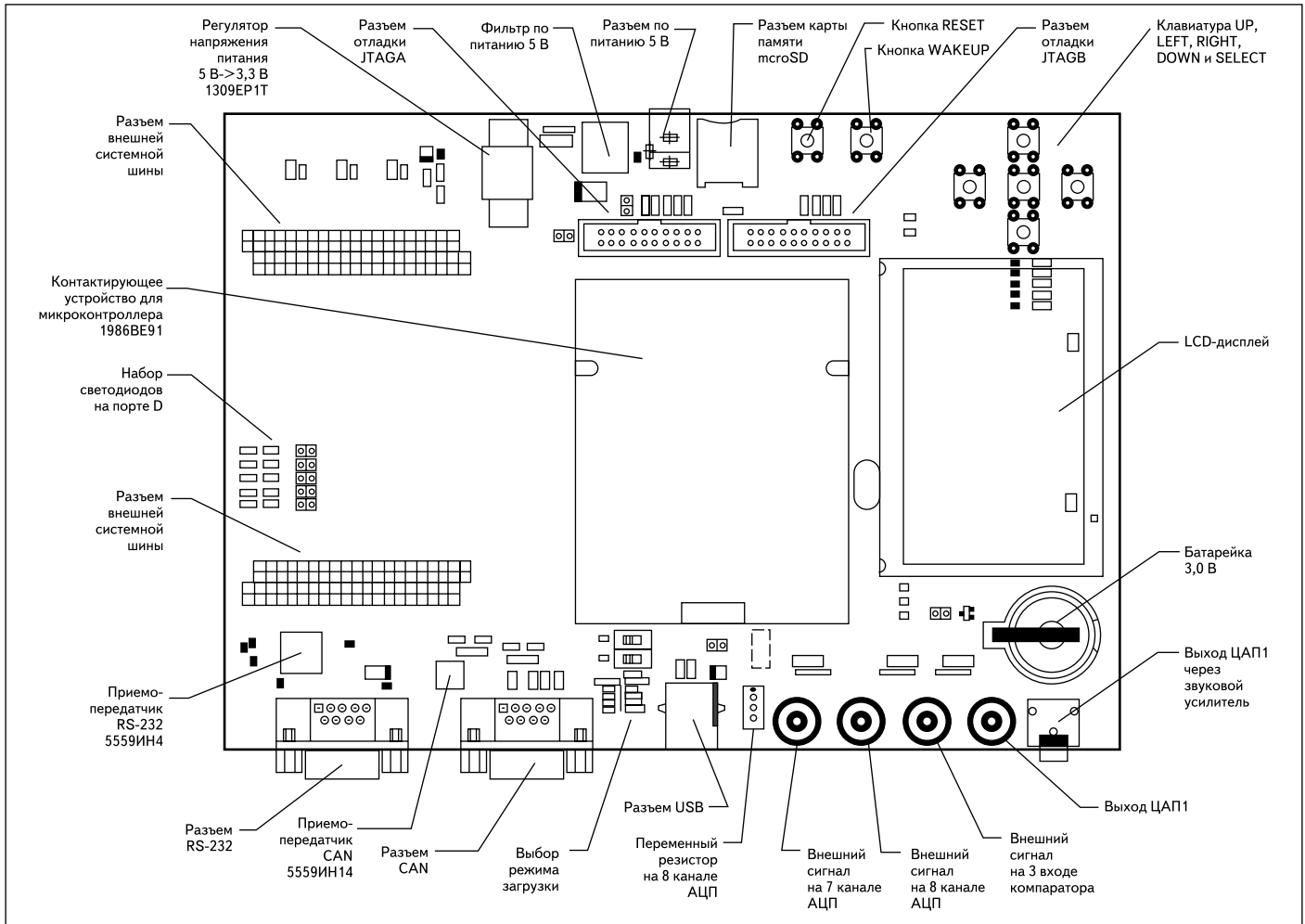


Рис. 2. Описание элементов отладочной платы

без умножения частоты. Для тактирования УСАПП также воспользуемся внутренним генератором.

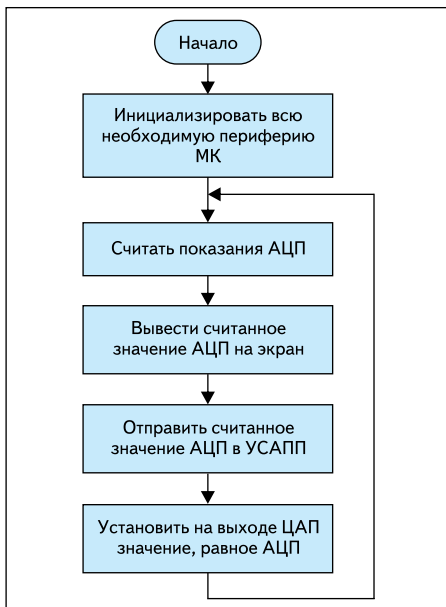


Рис. 3. Алгоритм работы программы примера для МК 1986BE91Т

Необходимо включить тактирование всех используемых блоков периферии. Таким образом, процедура инициализации тактовых частот будет выглядеть так:

```

void clk_init(void)
{
    // включаем внешний осциллятор в режиме осциллятора
    HS_CONTROL= HSE_ON;
    // устанавливаем параметры PLL:
    // коэффициент умножения CPU_PLL 3 при внешней частоте
    // 8 МГц, на выходе PLL 24 МГц
    // коэффициент умножения USB_PLL не задан
    // включение CPU_PLL - включено
    // включение USB_PLL - выключено
    // перезапуск CPU_PLL - включено
    // перезапуск USB_PLL - выключено
    PLL_CONTROL=(CPU_PLL_3)((CPU_PLL_on))(CPU_PLL_toggle);

    // ждем выхода на режим CPU_PLL
    // настройка тактовых сигналов CPU
    // Основной сигнал - HSE
    // HSE - на PLL, с PLL дальше
    // Коэффициент деления - 1
    // тактирование периферии HSI
    CPU_CLOCK=(HSE)(PLL_CPU0)((CPU_C2))(HCLK_HSI);
    // настройка тактовых сигналов ADC
    // тактирование АЦП 8 МГц (внеш)*3 (PLL) = 24 МГц (при максимуме 28 МГц)
    ADC_MCO_CLOCK=(ADC_CLK_EN)((ADC_C1_CPU_C2)
    (ADC_C2_C1))(ADC_C3);
    // раздача тактовых сигналов
    PER_CLOCK=0xFFFFFFFF;
}
  
```

Кроме того, необходимо инициализировать периферию. Не будем останавливаться

на этом подробно, в силу того, что это несложно сделать, используя документацию на МК.

Теперь осталось написать вспомогательные функции для вывода информации на дисплей. Они занимают достаточно много строчек кода и полностью взяты из примера, поставляемого со средой, поэтому в целях экономии места не будем здесь их приводить. Приведем код функций передачи/приема данных в/из УСАПП, получения данных из АЦП и для работы с ЦАП:

```

void SEND_UART2(unsigned char i)
{
    UART2DR = i; // помещаем посылаемый байт в регистр данных
}

unsigned char GET_UART2()
{
    unsigned char i;
    i = UART2DR; // считываем принятый байт из регистра данных
    return i;
}

unsigned int read_adc (unsigned char channel)
{
    unsigned int adc_res=10;
    // сброс предыдущего номера канала
    ADC2_CFG&= ~all_channel;

    // установка нового номера канала и запуск преобразования
    ADC2_CFG|= (channel<<4);
    ADC2_CFG|= GO;
    // ждем окончания преобразования
    while(ADC2_CFG&GO);
    // считываем только результат преобразования
    adc_res=(ADC2_RESULT&0x00000fff);

    return adc_res;
}
  
```

```

}
void set_dac (unsigned int code)
{
DAC1_DATA=code; // записываем цифровой код сигнала в регистр данных ЦАП
}

```

И самое основное — функция main:

```

int main(void)
{
unsigned int i=0, adc_result=0;
  clk_init();
  port_init();
  adc_init();
  dac_init();
  INIT_UART2();
  LCD_INIT();
  LCD_CLS();
  //----- основной цикл-----
  while(1)
  {
    adc_result=read_adc(8);
    SEND_UART2((unsigned char) (adc_result>>8));
    SEND_UART2((unsigned char) (adc_result));
    translate(adc_result);
    for (i=0;i<4; i++) LCD_PUTC(5+i*12, 0, num[i]);
    set_dac(adc_result);
  } // end of while(1)
}

```

## Заключение

Необходимо отметить, что сравнимых с МК серии 1986VE9x по функциональности, быстродействию, широте сферы применения микроконтроллеров отечественного производства просто не существует (а зарубежных аналогов очень немного). Также нельзя не упомянуть, что для этого МК существует широкий набор средств программной, аппаратной, а также технической поддержки, что облегчает его освоение [10].

## Литература

1. Шумилин С. Новая серия отечественных 32-разрядных высокопроизводительных микроконтроллеров семейства 1986 на базе процессорного ядра ARM Cortex\_M3 // Компоненты и технологии. 2008. № 10.
2. Шумилин С. Характеристики производительности микроконтроллеров на базе ядра ARM

Cortex-M3 // Электронные компоненты. 2009. № 8.

3. Предварительный вариант спецификации. Серия 1986VE9x высокопроизводительных 32-разрядных микроконтроллеров на базе процессорного ядра ARM Cortex-M3.
4. Руководство по эксплуатации отладочной платы для МК 1986VE91T.
5. Комплект инструментальных средств для микроконтроллеров ЗАО «ПКК Миландр» 1986VE91. Быстрый старт.
6. Пакет инструментальных средств CodeMaster-ARM для микроконтроллеров ПКК «Миландр» 1986VE91xx с ядром Cortex-M3.
7. Инструментальные средства для разработки и отладки систем на базе микроконтроллеров Cortex-M3/M1/M0, ARM7/ARM9. Руководство пользователя.
8. [www.phyton.ru](http://www.phyton.ru)
9. Спецификация. Жидкокристаллический модуль МТ-12864J.
10. [www.milandr.ru](http://www.milandr.ru)