

Микроконтроллеры 1886ВЕ3У/4У

Мария Ермак (Москва)

Статья содержит краткое описание микроконтроллеров типа 1886ВЕ3У/4У производства ЗАО «ПКК Миландр».

Описаны существующие средства аппаратной и программной поддержки разработок на этих МК. Рассматривается пример создания простого USB-устройства на базе МК типа 1886ВЕ4У.

ВВЕДЕНИЕ

Для обеспечения высокой скорости обмена данными в современных устройствах всё чаще используется интерфейс USB благодаря его универсальности, возможности подключения нескольких устройств к шине USB и поддержке Plug-and-Play. Существуют различные способы реализации USB-устройств: можно использовать специализированные микросхемы, преобразующие поток данных по шине USB к какому-либо стандартному интерфейсу (USART, SPI и др.), либо реализовывать интерфейс USB в микроконтроллере. В последнем случае возможна программная либо аппаратная реализация. До недавнего времени при выборе аппаратной реализации приходилось использовать исключительно зарубежные микроконтроллеры, но сейчас появилась возможность применения МК отечественного производства: ЗАО «ПКК Миландр» предлагает МК типа 1886ВЕ3У и 1886ВЕ4У. В данной статье мы рассмотрим МК типа 1886ВЕ4У, поскольку у него более полная реализация USB-интерфейса.

ОБЗОР МИКРОКОНТРОЛЛЕРА

Высокопроизводительный восьмиразрядный RISC-микроконтроллер с гарвардской архитектурой 1886ВЕ4У выполнен по КМОП-технологии 0,6 мкм. Он предназначен для однокристалльной реализации систем передачи, обработки и хранения данных. Микроконтроллер может использоваться для организации вычислительных систем невысокого быстродействия и в качестве преобразователя различных типов интерфейсов. Ядро микроконтроллера совместимо с PIC17C756.

Основные характеристики микроконтроллера:

- напряжение питания 4,5...5,5 В;
- тактовая частота до 33 МГц;
- минимальное время выполнения команды 121 нс;
- четырёхвекторный контроллер прерываний, поддерживающий 18 источников прерываний (внешних и внутренних);
- 16-уровневый аппаратный стек;
- возможность раздельной работы с внутренней и внешней памятью программ (режимы: микроконтроллер, расширенный микроконтроллер и микропроцессор);
- объём внутренней памяти данных 902 байта;
- все регистры специального назначения находятся в адресном пространстве памяти данных;
- объём адресуемой памяти программ до 64К × 16 бит;
- на кристалле содержится 32К × 6 бит памяти программ флэш-типа.

Микроконтроллеры отличаются числом пользовательских конечных точек (в 1886ВЕ3У их 2, а в 1886ВЕ4У – 4), наличием блока аппаратной поддержки шифрования по ГОСТ 28147-89 в приборе 1886ВЕ3У и наличием интерфейса SPI в микросхеме 1886ВЕ4У. Поэтому в дальнейшем будем рассматривать только МК 1886ВЕ4У [1, 2].

Характеристики периферийных модулей МК:

- до 28 универсальных линий ввода/вывода с индивидуальной настройкой направления;
- 16-разрядный таймер/счётчик с восьмиразрядным программируемым предделителем (таймер 0);
- один универсальный синхронно-асинхронный приёмопередатчик (USART) с программируемой скоростью передачи информации в синхронном режиме до 8 Мбит/с, в асинхронном режиме – до 1 Мбит/с;
- универсальный, аппаратно реализованный контроллер и аналого-

вый приёмопередатчик интерфейса USB 1.1 с 4 пользовательскими конечными точками и со скоростью передачи до 12 Мбит/с;

- последовательный синхронный порт с режимом последовательного периферийного интерфейса SPI;
- универсальный контроллер внешней памяти типа NAND FLASH;
- встроенная энергонезависимая память с электрическим стиранием (EEPROM) объёмом 256 байт.

Микроконтроллер выпускается серийно для работы в двух температурных диапазонах: –60...85°C (1886ВЕ4У) и 0...70°C (К1886ВЕ4АУ), с приёмкой «5» и «1» соответственно. Корпус 48-выводной Н16.48-1В. Внешний вид корпуса микроконтроллера и его условное графическое обозначение показаны на рисунке 1.

ПРОГРАММНАЯ И АППАРАТНАЯ ПОДДЕРЖКА

Для разработки ПО для МК типа 1886ВЕ4У существует следующий набор программных и аппаратных средств:

- для создания исходного кода программ можно использовать две среды программирования и отладки: IDE1886 (разработана ЗАО «ПКК Миландр») или MPLab фирмы Microchip (версии 6.xx и 7.xx);
- для компиляции программ необходим один из компиляторов ассемблера MPASMWIN.EXE фирмы Microchip, PASM-PIC.EXE фирмы Phytion или компилятор Си HI-TECH PICC фирмы HI-TECH Software [3];
- для программирования МК необходим программатор, разработанный ЗАО «ПКК Миландр», который подключается к ПК одновременно через LPT-порт и USB-порт;
- для работы программатора необходим драйвер *giveio.sys*;
- для работы программатора и работы с созданным на базе МК устройством USB необходим драйвер USBIO;
- также для программирования используется программа MicroProg 1886, разработанная ЗАО «ПКК Миландр»;
- для отладки программ можно использовать внутрисхемный эмуля-

тор, разработанный ЗАО «ПКК Миландр»;

- для быстрого освоения микроконтроллера 1886BE4Y можно использовать отладочную плату, разработанную ЗАО «ПКК Миландр» (см. рис. 2).

Более подробную информацию о программной и аппаратной поддержке разработок на МК типа 1886BE4Y можно получить из документации, размещённой на интернет-странице ЗАО «ПКК Миландр» [4].

РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ 1886BE4Y

Для создания USB-устройства на 1886BE4Y разработчику не потребуются глубокого знания интерфейса или длительного освоения, т.к. большинство необходимых для инициализации МК действий в качестве USB-устройства после его подключения к шине выполняются аппаратно (см. рис. 3).

Блок контроллера USB предназначен для приёма и передачи данных по USB-интерфейсу версии 1.1 между микроконтроллером (functional device) и хост-контроллером (USB host). Блок контроллера USB может функционировать в режимах Low Speed (1,5 Мбит/с) и Full Speed (12 Мбит/с). В составе блока реализовано пять конечных точек. Нулевая конечная точка предназначена для инициализации контроллера USB после подсоединения его к шине USB. Для обмена пользовательскими данными с хост-контроллером могут быть настроены четыре пользовательские конечные точки. Они могут быть сконфигурированы в режим IN (для передачи данных хост-контроллеру) или OUT (для приёма данных от хост-контроллера).

В зависимости от выбранного режима работы USB-модуля обеспечены соответствующие стандарту типы обмена пользовательских конечных точек с хост-контроллером. При работе в режиме Low Speed обеспечивается только тип обмена Interrupt Transfer. При работе в режиме Full Speed обеспечиваются следующие типы обмена: Interrupt Transfer, Bulk Transfer и Isochronous Transfer. Блок контроллера USB позволяет разработчику задать собственный дескриптор устройства.

Для автоматической инициализации USB-интерфейса при подключе-

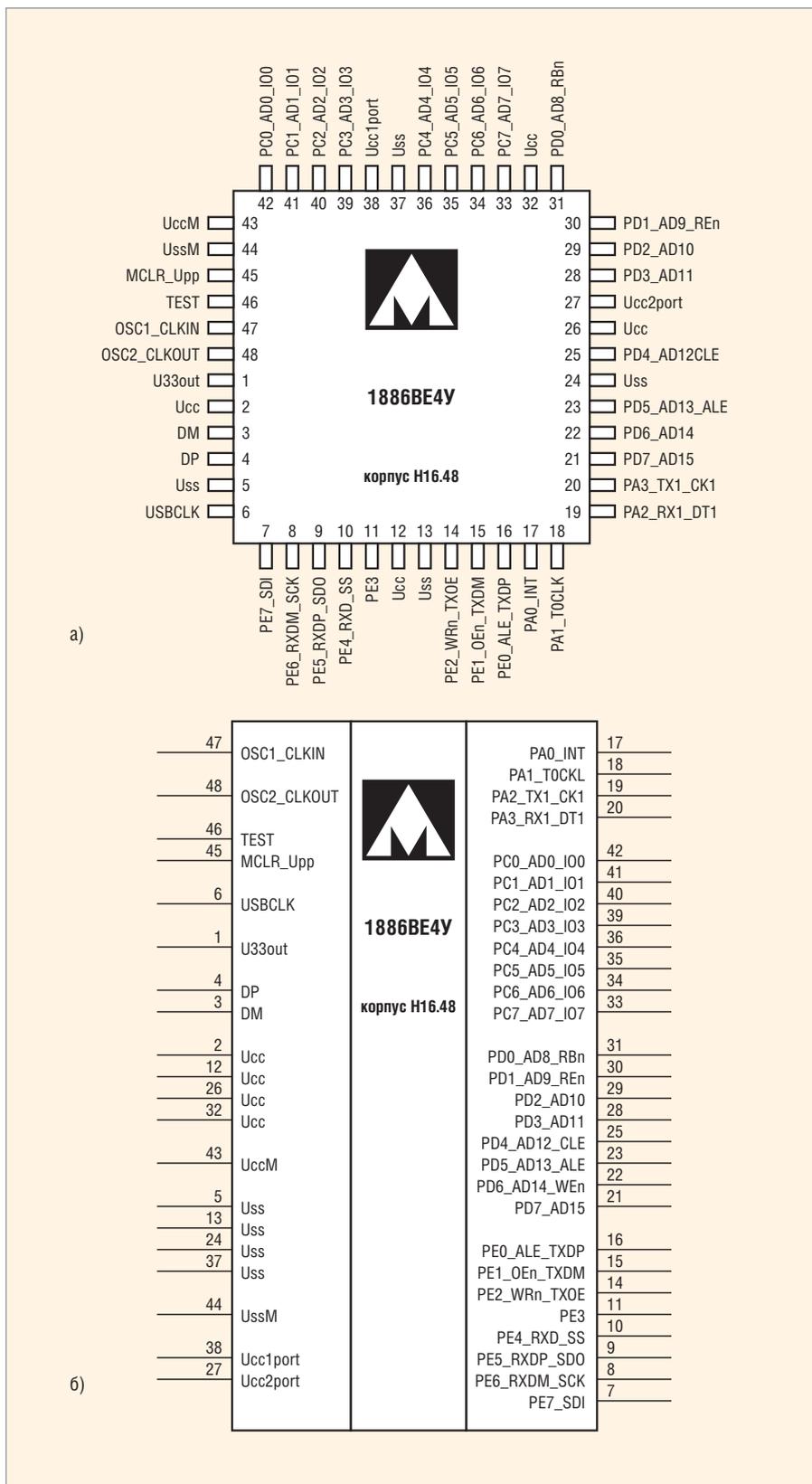


Рис. 1. Внешний вид корпуса микроконтроллера и его условное графическое обозначение

нии устройства к хост-контроллеру необходимо:

- выбрать конфигурацию конечных точек;
- заполнить дескриптор в соответствии с конфигурацией конечных точек;
- задать режим работы USB-устройства;

- разрешить работу блока USB.

Выдача хост-контроллеру информации о дескрипторе устройства, получение от хост-контроллера функционального адреса и конфигурирование устройства происходят автоматически. В ходе инициализации блок контроллера USB отражает текущее состоя-



Рис. 2. Внешний вид набора программных и аппаратных средств поддержки разработок на МК типа 1886BE4U

ние в регистрах статуса и ошибок. Программа, выполняемая на микроконтроллере, должна отслеживать состояние блока контроллера USB, а в случае возникновения ошибок принять меры по их исправлению.

С момента подсоединения микроконтроллера к хост-контроллеру до момента начала автоматической инициализации пройдёт не менее 500 мс. За это время должно включиться питание микроконтроллера, запуститься ядро и начаться выполнение программы (если микроконтроллер получает питание от шины USB). Программа должна успеть задать пользовательский дескриптор,

сконфигурировать конечные точки, задать режим работы блока и разрешить его работу.

После автоматической инициализации драйвер USB-устройства, работающий на стороне хост-контроллера, должен задать конфигурацию и интерфейс для блока контроллера USB.

В качестве примера рассмотрим реализацию простого USB-устройства на МК типа 1886BE4U на базе отладочной платы Eval9. Внешний вид платы изображён на рисунке 4. Для написания и отладки программы будем использовать среду MPLAB IDE v7.42 и компилятор HT_PICC.v8_05_pl2.

СТРУКТУРА ПРОГРАММЫ

Для примера рассмотрим реализацию простого USB-устройства на базе 1886BE4U. Устройство будет работать в режиме Full Speed, принимать от хоста однобайтовые числа, отображать их в двоичном виде на светодиодах и возвращать в хост. Для обмена данными со стороны хоста используется программа *USBIOApp.exe*.

Как и проект для любого другого микроконтроллера, написанный на Си, проект для МК типа 1886BE4U состоит из исполняемых модулей и за-

головочных файлов (см. рис. 5). Рассмотрим подробнее каждый из заголовочных файлов.

Файл *mil1886BE4.b* определяет регистры специального назначения для 1886BE4U, а также биты некоторых регистров для облегчения работы с ними и повышения читаемости кода.

Файл *leds.b* содержит макросы для управления светодиодами на плате Eval9. Светодиоды на плате объединены в матрицу с переключками, поэтому удобнее один раз определить все необходимые манипуляции для управления ими.

Файл *enumeration.b* содержит макросы для работы с USB-модулем, управления конечными точками, настройки USB-модуля, а также прототипы функций, описанных в *enumeration.c*.

Исполняемый код размещён в двух файлах:

- файл *enumeration.c* (см. сайт журнала) содержит функции, необходимые для инициализации USB-модуля: заполнение дескрипторов, конфигурирование конечных точек и общая инициализация. Алгоритм работы ПО для инициализации USB-модуля показан на рисунке 6;

- файл *main.c* содержит основную программу, которая инициализирует USB-модуль, затем ведёт непрерывный обмен данными с хост-контроллером через пользовательские конечные точки. Приём данных от хоста происходит через первую конечную точку, а отправка данных в хост – через вторую конечную точку. Также в основной программе осуществляется обработка полученных от хоста данных для вывода на светодиоды. Для проверки правильности приёма данные отправляются обратно в хост.

Не будем приводить заголовочные файлы полностью, рассмотрим лишь те макросы, которые используются в этом исполняемом модуле:

- *set_max_packet_size_ep1(ep_packet)* устанавливает максимальный размер передаваемого/принимаемого пакета для первой конечной точки путём записи соответствующего значения [1] в регистр P1_CFG1;
- *set_type_ep1(ep_type)* конфигурирует тип первой конечной точки путём записи в регистр EP1_CFG2 соответствующего значения [1];

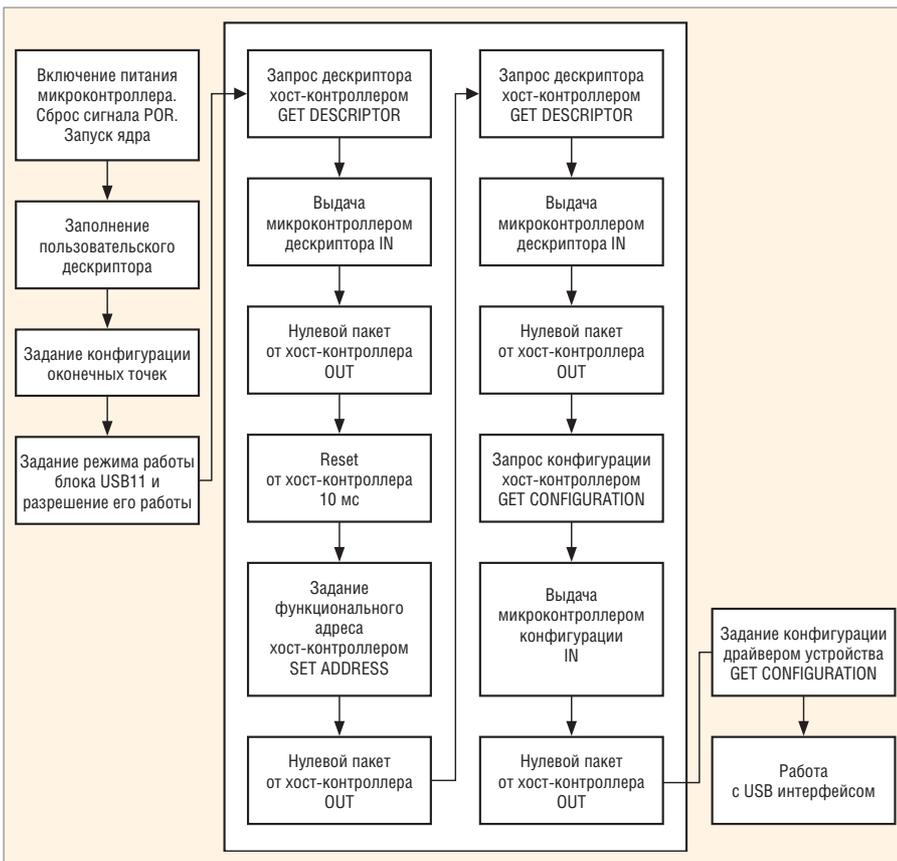


Рис. 3. Блок-схема работы микроконтроллера при запуске USB-интерфейса

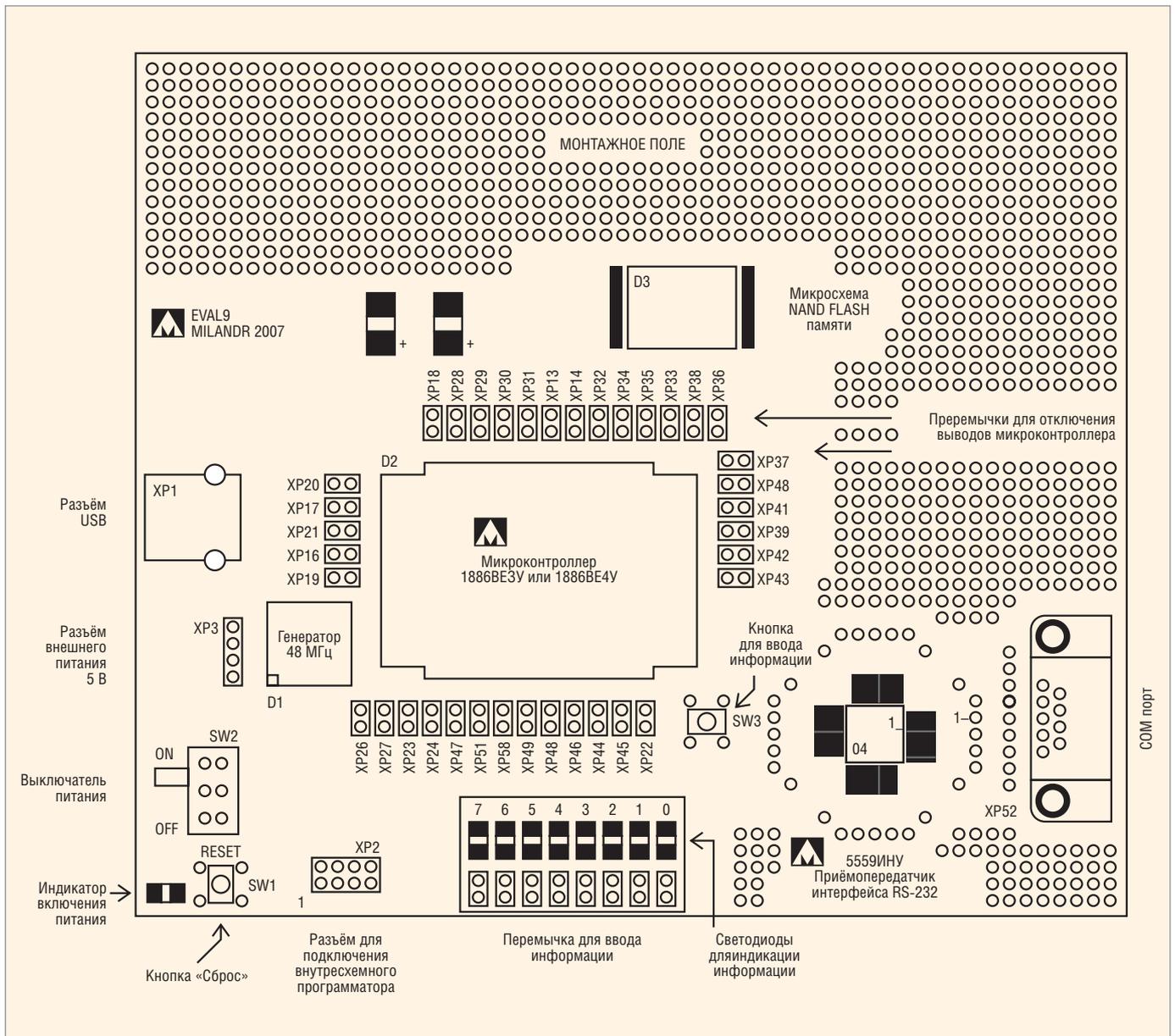


Рис. 4. Внешний вид демонстрационно-отладочной платы

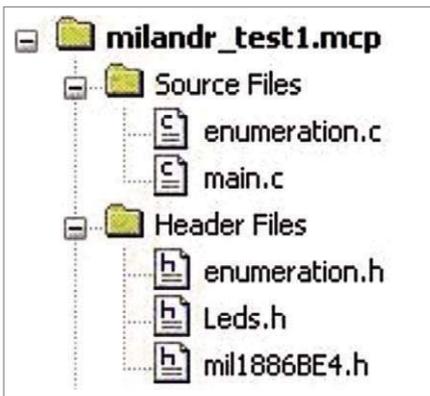


Рис. 5. Структура проекта

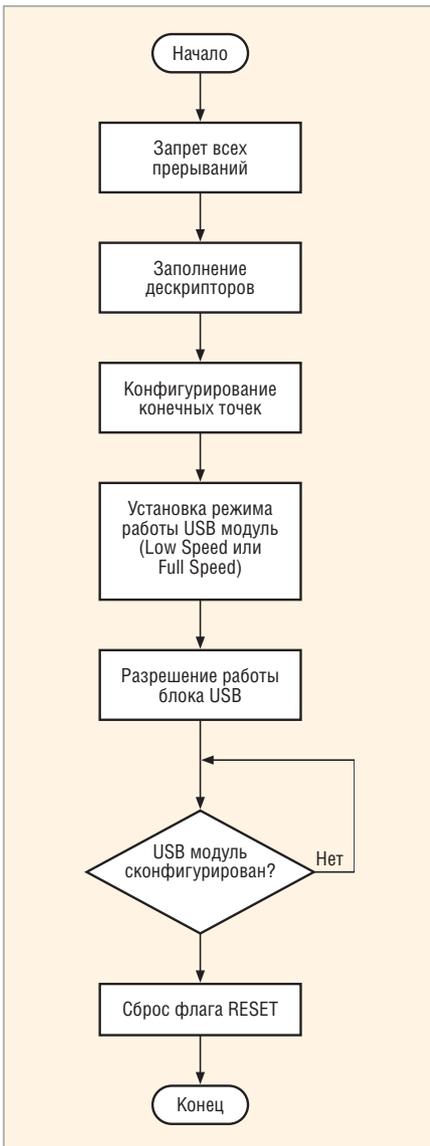


Рис. 6. Алгоритм работы ПО для инициализации USB-модуля

- *usb_mode_fullow()* устанавливает режим работы USB модуля: Full Speed или Low Speed;
- *usb_enable()* разрешает работу USB-модуля.

Файл *main.c*
`#include "mil1886BE4.h"`

```
#include "enumeration.h"
#include "Leds.h"
//посылка данных в хост через ep2
//прием данных из хоста через ep1
void main()
{
    unsigned char
data=0,new_data=0;
    DDRE=0;
    PORTE=0;
    Leds_enable();
    usb_init();
while(!configuration_ok());//ждем
готовности USB модуля
clear_usb_rst();//сброс флага
USB_RST
    while(1)
    {
        if(
!(fifo1_empty()) )
        {
            data=
usb_read_byte();
        }
        if(
(fifo2_empty())
){usb_write_byte(data);}
        Leds_0_3_off();
        Leds_4_7_on();
        if(data>0x0f)
        {
PORTE|=(data&0xf);
PORTE&=0x0f;
        }
        else
        {
PORTE&=0x0f;
        }
        Leds_4_7_off();
        Leds_0_3_on();
        PORTE|=(data<<4);
        PORTE&=0x0f;
    }
}
```

В основной программе используются следующие макросы:

- *configuration_ok()* проверяет выставление флага CONF_SET; если флаг выставлен, то USB модуль успешно сконфигурирован;
- *clear_usb_rst()* сбрасывает флаг получения от хост-контроллера команды RESET;
- *fifo1_empty()* проверяет наличие данных в fifo-буфере первой конечной точки;
- *usb_read_byte()* читает байт данных из fifo-буфера первой конечной точки;

- *usb_write_byte(data)* записывает в fifo-буфер второй конечной точки байт данных.

Из приведённого выше листинга видно, что программа предельно проста, и для инициализации USB-модуля необходимо всего несколько строк (не считая заполнения полей дескриптора).

ЗАКЛЮЧЕНИЕ

Как у любого микроконтроллера, у 1886BE4Y есть свои достоинства и недостатки. Поэтому необходимо выбирать USB-микроконтроллер, исходя из требований к создаваемому устройству. Если для обмена данными между разрабатываемым устройством и ПК не требуется скорость более 12 Мбит/с, если есть специалисты, знакомые со средой MPLab, но не программировавшие ранее для интерфейса USB, а время, отведённое на разработку, ограничено и необходимо применить элементную базу отечественного производства, то идеальным решением будет использование описанного выше прибора 1886BE4Y.

Очевидными достоинствами этого МК являются:

- простота написания программы;
 - для разработки ПО необходим минимум знаний об интерфейсе USB;
 - имеются средства программной и аппаратной отладки ПО и освоения МК;
 - возможность получения оперативной консультации специалиста как по электронной почте, так и по телефону.
- Недостатки:
- медленная загрузка ПО в МК и стирание ранее загруженного ПО (LPT-программатор);
 - недостаточная гибкость для реализации любого USB-устройства, т.к. структура дескрипторов жёстко задана.

ЛИТЕРАТУРА

1. Спецификация 1886BE4Y, K1886BE4Y Версия 2.1 от 21.03.2008.
2. Спецификация 1886BE3Y, 1886BE31Y, K1886BE3Y, K1886BE31Y, K1886BE3 Версия 2.1 от 20.03.2008.
3. MicroProg2006 – программирование внутренней флэш-памяти микроконтроллера 1886BE2Y. Руководство пользователя.
4. www.milandr.ru.

