

ИСПОЛЬЗОВАНИЕ ОТЛАДОЧНОЙ ПЛАТЫ STM32F3 DISCOVERY ДЛЯ ИЗУЧЕНИЯ МИКРОКОНТРОЛЛЕРОВ С ЯДРОМ ARM

Д.О. Макаров

dimakk444@gmail.com

SPIN-код: 2645-6840

А.С. Антонов

artemkauser@yandex.ru

SPIN-код: 1759-5370

МГТУ им. Н.Э. Баумана, Москва, Российская Федерация

Аннотация

Рассмотрены основы программирования семейства микроконтроллеров STM32F3 на базе ядра фирмы Advanced RISC Machine — ARM Cortex-M4 с архитектурой ARMv7M. Представлено описание отладочной платы Discovery kit with STM32F303 MCU. Перечислены ключевые моменты использования документации в процессе изучения микроконтроллера. Приведено детальное описание работы с периферийными устройствами микроконтроллера с помощью библиотеки изготовителя StdPeriph. Решена задача управления периферийными устройствами через удаленный терминал. В ходе решения настроены порты ввода-вывода и последовательный порт микроконтроллера. Для обмена информацией с компьютером использован стандартный интерфейс UART.

Ключевые слова

Микроконтроллер, STM32F303VC, управление, однокристалльный компьютер, отладочная плата, библиотека функций, Advanced RISC Machine, ядро Cortex-M4, UART

Поступила в редакцию 25.12.2017

© МГТУ им. Н.Э. Баумана, 2018

Микроконтроллер — это однокристалльная микросхема, способная управлять электронными устройствами. Микроконтроллеры используются в самых разных системах — умных домах, смартфонах, мехатронных модулях, интеллектуальных датчиках и небольших распределенных системах. Неотъемлемой частью их разработки является изучение возможностей кристалла и его программирование. В данной статье рассмотрены результаты исследования возможностей микроконтроллера и предложен один из подходов к их программированию.

Современной мировой тенденцией является использование микроконтроллеров на базе архитектур типа RISC. Наиболее популярными представителями этого типа являются архитектуры фирмы Advanced RISC Machine (ARM). Эти устройства имеют хорошее соотношение производительности на ватт потребляемой мощности. Разумеется, один даже самый мощный процессор с архитектурой ARM на сегодняшний день несопоставим по вычислительной мощности с популярными процессорами, такими как Intel Core i7. Но этот процессор нуждается в активной системе охлаждения, его тяжело разместить в модулях, к габаритным размерам которых предъявляются жесткие требования. Использование таких устройств оправдано на верхнем уровне системы управления ме-

хатронной системы [1], так как там требуется большая вычислительная мощность. В то же время применять их на нижнем уровне нецелесообразно, поскольку они обладают избыточной производительностью и имеют большую стоимость. Здесь как нельзя лучше подходят устройства ARM, изготовленные в виде одного кристалла (system on chip — SoC). Современные SoC включают контроллер оперативной памяти, встроенную flash-память, модули аналого-цифрового и цифроаналогового преобразования, различные коммуникационные интерфейсы, такие как USART, SPI, I2C, USB [2–4].

Для изучения возможностей микроконтроллера были приобретены отладочные печатные платы STM32F3 Discovery, на которых установлен чип STM32F303VC с ядром ARM Cortex M4. Данное устройство позволяет изучить возможности микроконтроллера, исследовать его периферийные модули и датчики, расположенные на самой печатной плате.

Описание платы STM32F3 Discovery. Устройство представляет собой отладочную плату (рис. 1), на которой установлен микроконтроллер STM32F303VC 1, программатор ST-LINK/V2, позволяющий записывать и отлаживать программу [5], два порта USB (порт программатора 2 и пользовательский), две кнопки, восемь подключенных к портам ввода-вывода светодиодов 3, MEMS гироскоп и компас [6]. Для подключения внешних устройств на плате имеются контактные штыри, с которыми соединены выводы портов микроконтроллера 4.

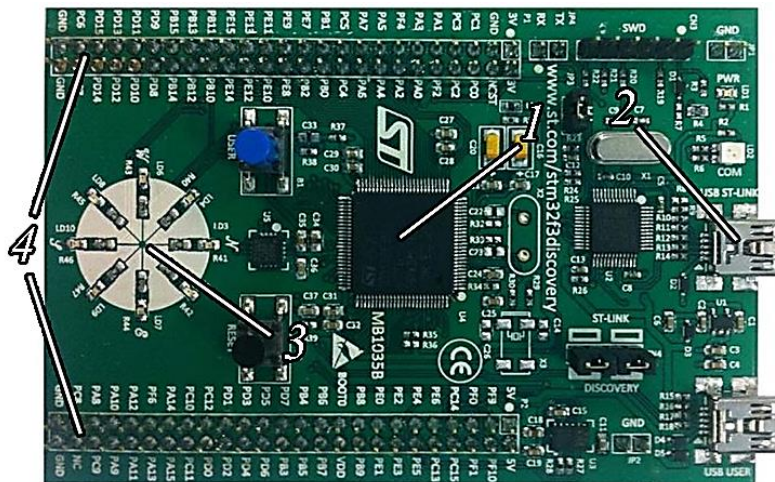


Рис. 1. STM32F3 Discovery

«Сердцем» микроконтроллера является 32-битное ядро ARM Cortex-M4. На кристалле также расположены 256 Кб flash-памяти, 40 Кб SRAM, и несколько периферийных устройств, таких как таймеры, АЦП, ЦАП, компараторы, часы реального времени. Микроконтроллер оснащен различными интерфейсами, включая UART, SPI, I2C, USB [7, 8]. Для снижения энергопотребления по умолчанию все устройства отключены, их питанием и тактированием управляет модуль RCC (reset and clock control).

Использование платы STM32F3 Discovery. Основной задачей микроконтроллера является управление какими-либо внешними устройствами по заданному алгоритму. В общем случае сигналы управления могут быть аналоговыми или цифровыми. Поскольку в современном мире преобладает цифровая техника, будем рассматривать цифровые порты ввода-вывода.

Как правило, для решения сложных задач, таких как управление роботом, вычислительных возможностей одного микроконтроллера недостаточно. Для их решения применяют системный подход, в ходе которого систему управления разбивают на иерархически распределенные уровни. На каждом уровне может быть один или несколько вычислительных модулей. Требуется обеспечить обмен данными между модулями внутри одного уровня и между разными уровнями. Для этого используют стандартные телекоммуникационные интерфейсы, такие как RS232 и RS485, которые являются последовательными и используют логику UART.

Обе рассмотренные задачи могут быть решены на примере управления светодиодами от внешнего компьютера. В реальных приложениях аналогом служит задача управления приводной системой робота от устройства верхнего уровня. Устройство верхнего уровня посылает сигналы задания скорости или положения, а нижний уровень обрабатывает их и подтверждает исполнение. В данном случае верхний уровень — это компьютер, а нижний уровень — это отладочная плата с микроконтроллером.

Реализуем управление светодиодами от внешнего компьютера посредством интерфейса UART. Эта задача требует настройки соответствующих периферийных модулей микроконтроллера и состоит из трех подзадач:

- настройка управления светодиодами;
- настройка порта UART на микроконтроллере и компьютере;
- обмен данными по выбранному порту между микроконтроллером и компьютером.

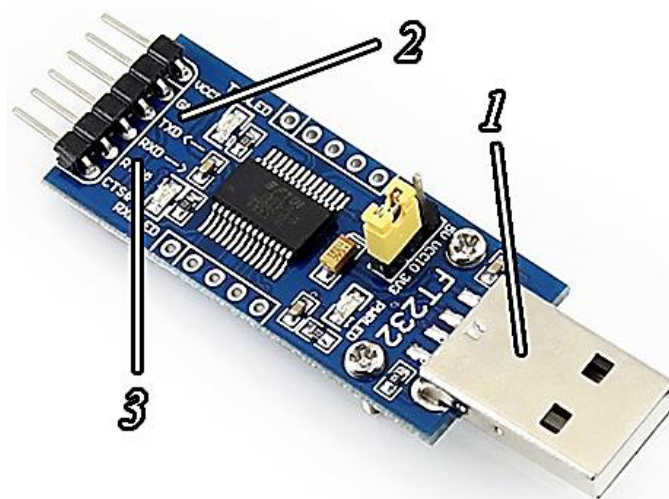


Рис. 2. Мост USB–UART

Для связи с компьютером был использован мост USB-UART (рис. 2), который подключается к компьютеру через USB-порт 1, а выводы Tx и Rx (позиции 2 и 3) соединяются с модулем UART на микроконтроллере.

Настройка периферийных модулей осуществляется путем установки бит соответствующих управляющих регистров, которую выполняют в несколько этапов:

- настройка тактирования периферийного устройства с помощью регистров управления RCC шины, к которой оно подключено;
- настройка регистров управления самого устройства.

Управление светодиодами с помощью портов ввода-вывода общего назначения. На плате имеется 8 светодиодов (по два каждого цвета), которые подключены к портам ввода-вывода общего назначения (general purpose input-output — GPIO). В примере рассмотрено управление красными светодиодами.

Схема подключения светодиодов к выводам микроконтроллера приведена на рис. 3, а также в документации к плате [6]. Оба красных светодиода подключены к выводам 9 и 13 порта E (на рис. 3 обозначены как PE9 и PE13 соответственно). Для доступа к ним необходимо включить тактирование порта и настроить порт E на выдачу данных.

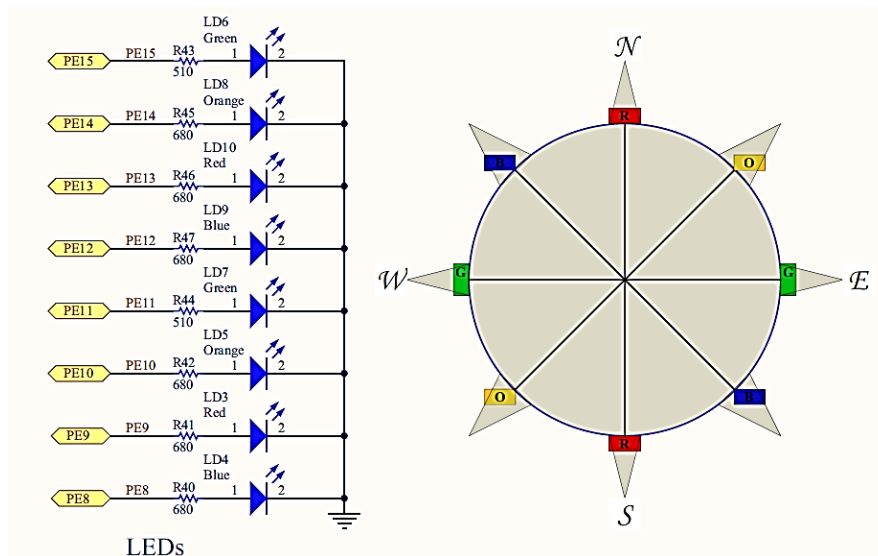


Рис. 3. Электрическая схема подключения светодиодов

Использование библиотеки StdPeriph Library. Изготовитель микроконтроллеров предоставляет набор стандартных библиотек, которые позволяют абстрагироваться от конкретного устройства и писать программы, используя набор готовых функций и битовых масок, а не заполнять регистры. Все режимы работы и параметры устройств описаны в виде структур в соответствующих файлах. Процесс настройки сводится к заполнению структуры и вызову функции инициализации устройства. Полное описание библиотеки представлено в [9].

Настройка порта на вывод данных осуществляется посредством инициализации экземпляра структуры `GPIO_InitTypeDef`, заполнением его полей, а затем вызовом функции инициализации порта. В поле `GPIO_Pin` указывают выводы `GPIO_Pin_9` и `GPIO_Pin_13`, к которым подключены светодиоды. Выбирают режим работы порта `GPIO_Mode` на выход — `GPIO_Mode_OUT`. Настройки записывают в регистры управления по команде `GPIO_Init`, где указывают имя порта.

Установка выходного значения порта, т. е. включение и выключение светодиода, осуществляется функциями `GPIO_SetBits` и `GPIO_ResetBits` соответственно.

Работа с интерфейсом UART. Рассматриваемый в данной работе микроконтроллер имеет три порта USART (1, 2, 3) и два порта UART (4, 5). При этом порты USART могут работать в режиме UART, поэтому для нашей задачи можно использовать любой порт. Настройку портов требуется выполнять в следующем порядке, рассмотренном на примере порта USART1:

- выбрать порт и физически соединить его с мостом USB–UART с помощью проводов;
- сконфигурировать выбранный порт на прием / передачу данных;
- запустить прием / передачу данных через порт.

Выводы микроконтроллера, к которым подсоединен передатчик, определяют с помощью [10]. Мост с микроконтроллером соединяют тремя проводами: GND-GND, TX-RX, RX-TX. В рассматриваемом примере выход TX моста подключен к выводу PA10, вход RX — к выводу PA9.

Инициализация порта выполняется в четыре этапа:

- инициализировать физический порт ввода-вывода GPIO;
- программно подключить выводы порта GPIO к модулю UART;
- настроить модуль UART;
- запустить модуль UART.

Инициализация порта ввода-вывода GPIO проводится аналогично тому, как описано в примере со светодиодами, а для задания режима работы выбирается «альтернативная функция».

Для подключения выводов микроконтроллера к блоку UART используется функция `GPIO_PinAFConfig`. В качестве параметров указываются: порт GPIO, имя вывода и номер альтернативной функции. В нашем случае для передатчика UART1 используются имена выводов `GPIO_PinSource9` и `GPIO_PinSource10`, номер альтернативной функции `GPIO_AF_7`.

Настройку порта UART выполняют аналогично настройке порта GPIO. Необходимо включить тактирование порта, заполнить поля структуры и вызвать функцию инициализации. Поля заполняют в соответствии с параметрами интерфейса, установленными на компьютере. Для запуска модуля UART необходимо вызвать функцию запуска порта `USART_Cmd` с параметром `ENABLE`.

Финальной частью программы является организация приема команды с компьютера для управления питанием светодиодов. Для приема данных существует функция `USART_ReceiveData`, возвращающая беззнаковое целое 16-битное число. Через порт UART данные передаются побайтно, поэтому

старшие 8 бит заполняются нулями, а младшие — значащими битами. О наличии данных в буфере приема сигнализирует флаг USART_FLAG_RXNE, состояние которого проверяют с помощью функции USART_GetFlagStatus. Таким образом, нужно постоянно проверять этот флаг, а при его установке выполнить прием данных. Затем осуществляется распознавание команды и выдается сигнал (посылка) на включение / выключение светодиодов. В данном случае светодиоды включаются при приеме команды 0x01 и выключаются по команде 0x00. Последнее действие — отправка команды обратно на компьютер с помощью функции USART_SendData (так называемый режим Echo) — означает успешный прием и обработку команды микроконтроллером. Итоговый листинг программы приведен ниже:

```
#include "stm32f30x.h"

int main(void)
{
    // переменная для хранения команды
    uint8_t data = 0;

    // структуры для настройки
    GPIO_InitTypeDef GPIO_InitStructure_LEDS;
    GPIO_InitTypeDef GPIO_InitStructure_USART1;
    USART_InitTypeDef USART_InitStructure_USART1;

    // настройка светодиодов
    RCC_AHBPeriphClockCmd(RCC_AHBPeriph_GPIOE, ENABLE);

    GPIO_InitStructure_LEDS.GPIO_Mode = GPIO_Mode_OUT;
    GPIO_InitStructure_LEDS.GPIO_Pin = GPIO_Pin_9 | GPIO_Pin_13;
    GPIO_Init(GPIOE, &GPIO_InitStructure_LEDS);

    // настройка UART
    RCC_AHBPeriphClockCmd(RCC_AHBPeriph_GPIOA, ENABLE);

    GPIO_InitStructure_USART1.GPIO_Mode = GPIO_Mode_AF;
    GPIO_InitStructure_USART1.GPIO_Pin = GPIO_Pin_9 | GPIO_Pin_10;

    GPIO_Init(GPIOA, &GPIO_InitStructure_USART1);
    GPIO_PinAFConfig(GPIOA, GPIO_PinSource9, GPIO_AF_7);
    GPIO_PinAFConfig(GPIOA, GPIO_PinSource10, GPIO_AF_7);

    RCC_APB2PeriphClockCmd(RCC_APB2Periph_USART1, ENABLE);

    USART_InitStructure_USART1.USART_BaudRate = 9600;
    USART_InitStructure_USART1.USART_HardwareFlowControl =
USART_HardwareFlowControl_None;
    USART_InitStructure_USART1.USART_Mode =
USART_Mode_Rx|USART_Mode_Tx;
    USART_InitStructure_USART1.USART_Parity = USART_Parity_No;
    USART_InitStructure_USART1.USART_StopBits = USART_StopBits_1;
    USART_InitStructure_USART1.USART_WordLength =
USART_WordLength_8b;
```

```
USART_Init(USART1, &USART_InitStruct_USART1);
USART_Cmd(USART1, ENABLE);

// основной цикл
while(1)
{
    // проверка на наличие данных в буфере
    if (USART_GetFlagStatus(USART1, USART_FLAG_RXNE) == SET)
    {
        // прием данных
        data = USART_ReceiveData(USART1);
        // распознавание данных
        if (data == 0x01)
            GPIO_SetBits(GPIOE, GPIO_Pin_9 | GPIO_Pin_13);
        if (data == 0x00)
            GPIO_ResetBits(GPIOE, (GPIO_Pin_9|GPIO_Pin_13));
        // отправка команды обратно на компьютер (ре-
        жим Echo)
        USART_SendData(USART1, data);
    }
}
```

Теперь при отправке через терминал компьютера команды 01 красные светодиоды загораются, в консоли появляется принятая команда 01(h). При отправке команды 00 светодиоды гаснут, в консоли появляется принятая команда 00(h).

Программа была написана в среде Keil uVision и загружена на плату, ее работа протестирована на реальном устройстве. Тесты подтвердили работоспособность данной программы.

Заключение. В процессе работы с микроконтроллером была решена задача управления цифровыми портами с внешнего устройства через коммуникационный интерфейс UART. Можно сделать вывод, что при программировании микроконтроллера семейства STM32F3 на ядре ARM наиболее ответственной частью является инициализация устройств периферии. Материалы данной статьи позволяют наглядно изучить способ разработки программного обеспечения для однокристальных систем, встраиваемых в мехатронные модули.

Литература

- [1] Карнаухов Н.Ф. *Электромеханические и мехатронные системы*. Ростов-на-Дону, Феникс, 2006, 320 с.
- [2] Огородников И.Н. *Микропроцессорная техника: введение в Cortex-M3*. Екатеринбург, Изд-во Уральского университета, 2015, 116 с.
- [3] Торгаев С.Н., Тригуб М.В., Мусоров И.С., Чертихина Д.С. *Практическое руководство по программированию STM-микроконтроллеров*. Томск, Изд-во ТПУ, 2015, 111 с.
- [4] Trevor M. *The insider's guide to the STM32 ARM based microcontroller*. Hexit, 2008, 89 p.

- [5] *ST-LINK/V2 in-circuit debugger/programmer for STM8 and STM32 UM1075*. STMicroelectronics, 2016, 19 p.
- [6] *Discovery kit with STM32F303VC MCU UM1570*. STMicroelectronics, 2016, 37 p.
- [7] ARM Infocenter. URL: <http://infocenter.arm.com/> (дата обращения 23.09.2017).
- [8] *STM32F303xB STM32F303xC*. STMicroelectronics, 2016, 148 p.
- [9] *Description of STM32F30xx/31xx Standart Peripheral Library UM1581*. STMicroelectronics, 2012, 584 p.
- [10] *STM32F303xB/C/D/E advanced ARM-based MCUs RM0316*. STMicroelectronics, 2017, 1141 p.

Макаров Дмитрий Олегович — студент кафедры «Робототехнические системы и мехатроника», МГТУ им. Н.Э. Баумана, Москва, Российская Федерация.

Антонов Артем Сергеевич — студент кафедры «Компьютерные системы автоматизации производства», МГТУ им. Н.Э. Баумана, Москва, Российская Федерация.

Научный руководитель — Михайлов Борис Борисович, кандидат технических наук, доцент кафедры «Робототехнические системы и мехатроника», МГТУ им. Н.Э. Баумана, Москва, Российская Федерация.

USING THE DEBUG BOARD STM32F3 DISCOVERY FOR STUDYING THE MICROCONTROLLERS WITH THE NUCLEUS ARM

D.O. Makarov

dimakk444@gmail.com

SPIN-code: 2645-6840

A.S. Antonov

artemkauser@yandex.ru

SPIN-code: 1759-5370

Bauman Moscow State Technical University, Moscow, Russian Federation

Abstract

The article considers the foundations of programming the microcontrollers STM32F3 family based on the Advanced RISC Machine nucleus ARM Cortex-M4 with the architecture ARMv7M. We introduce a description of the debug board Discoverykitwith STM32F303 MCU and list out the key points of using the documentation in the process of studying the microcontroller. The article describes in detail the microcontroller's operating with peripheral equipment through the use of the manufacturer's library StdPeriph. We have solved the problem of peripheral equipment control through the remote terminal. In the course of solution we have configured the input-output ports and the sequential port of the microcontroller. To exchange information with the computer we use standard interface UART.

Keywords

Microcontroller, STM32F303VC, control, single-chip computer, debug board, function library, Advanced RISC Machine, nucleus Cortex-M4, UART

© Bauman Moscow State Technical University, 2018

References

- [1] Karnaukhov N.F. Elektromekhanicheskie i mekhatronnye sistemy [Electromechanical and mechatronic systems]. Rostov-na-Donu, Feniks publ., 2006, 320 p.
- [2] Ogorodnikov I.N. Mikroprotsessornaya tekhnika: vvedenie v Cortex-M3 [Microprocessor engineering: introduction to Cortex-M3]. Ekaterinburg, Ural University publ., 2015, 116 p.
- [3] Torgaev S.N., Trigub M.V., Musorov I.S., Chertikhina D.S. Prakticheskoe rukovodstvo po programmirovaniyu STM-mikrokontrollerov [Practice guide to STM-microcontrollers programming]. Tomsk, TPU publ., 2015, 111 p.
- [4] Trevor M. The insider's guide to the STM32 ARM based microcontroller. Hexit, 2008, 89 p.
- [5] ST-LINK/V2 in-circuit debugger/programmer for STM8 and STM32 UM1075. STMicroelectronics, 2016, 19 p.
- [6] Discovery kit with STM32F303VC MCU UM1570. STMicroelectronics, 2016, 37 p.
- [7] ARM Infocenter. URL: <http://infocenter.arm.com/> (data obrashcheniya 23.09.2017).
- [8] STM32F303xB STM32F303xC. STMicroelectronics, 2016, 148 p.
- [9] Description of STM32F30xx/31xx Standart Peripheral Library UM1581. STMicroelectronics, 2012, 584 p.
- [10] STM32F303xB/C/D/E advanced ARM-based MCUs RM0316. STMicroelectronics, 2017, 1141 p.

Makarov D.O. — student, Department of Robotics and Mechatronics, Bauman Moscow State Technical University, Moscow, Russian Federation.

Antonov A.S. — student, Department of Computer Systems of Manufacturing Automation, Bauman Moscow State Technical University, Moscow, Russian Federation.

Scientific advisor — B.B. Mikhaylov, Cand. Sc. (Eng.), Assoc. Professor, Department of Robotics and Mechatronics, Bauman Moscow State Technical University, Moscow, Russian Federation.