МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**Национальный исследовательский ядерный университет «МИФИ»**

**Институт   
интеллектуальных кибернетических систем**

**Кафедра кибернетики (№ 22)**

Направление подготовки 09.03.04 Программная инженерия

**Курсовая работа**

**по дисциплине:**

Микропроцессорные системы

**на тему:**

Реализация управления диодом через терминал

|  |  |
| --- | --- |
| **Преподаватель:** | Кононов В.М. |
| **Группа:** | Б20-524 |
| **Студент:** | Косенко Н. И. |
| **Оценка:** |  |
| **Подпись:** |  |

**Москва 2022**

**Содержание**

[1. Постановка задачи 2](#_Toc123080605)

[2. Теоретические основы 3](#_Toc123080606)

[3. Результаты работ 5](#_Toc123080607)

[4. Выводы 7](#_Toc123080608)

[Список литературы 8](#_Toc123080609)

### Постановка задачи

В рамках работы за семестр была поставлена задача, включающая в себя изучение микроконтроллеров семейства STM32, методов работы с этими микроконтроллерами и написание программы позволяющая совершать мигания диодами на плате, контролируя их поведение через консоль, также для этого необходимо было настроить диалог между контроллером и пользователем.

В качестве исследуемого микроконтроллера был выдан STM32F407VET6, к которому в комплект шел программатор ST-LINK и переходник USB to TTL.

### Теоретические основы

Микроконтроллер – микросхема, позволяющая управлять электронными устройствами, на одном кристалле хранит функции процессора и периферийных устройств, ОЗУ (оперативно запоминающее устройство, оперативная память) и/или ПЗУ (постоянно запоминающее устройство).

Наш микроконтроллер построен на базе ядра Cortex-M4. Это можно найти в Programming Manual для платы на официальном сайте STM [1]. Там же можно найти описание данного ядра:

Процессор Cortex-M4 — это высокопроизводительный 32-разрядный процессор, разработанный для

рынок микроконтроллеров. Он предлагает значительные преимущества для разработчиков, в том числе:

* Выдающаяся производительность обработки в сочетании с быстрой обработкой прерываний;
* Улучшенная отладка системы с расширенными возможностями точек останова и трассировки;
* Эффективное ядро процессора, система и память;
* Сверхнизкое энергопотребление со встроенными спящими режимами;
* Надежность защиты платформы со встроенным блоком защиты памяти (MPU).

Arm® Cortex®-M4, способен работать на частоте до 168 МГц. Ядро Cortex-M4 оснащено модулем с плавающей запятой (FPU) одинарной точности, который поддерживает все инструкции и типы данных обработки данных Arm с одинарной точностью. Семейство STM32F407xx включает в себя высокоскоростную встроенную память (флеш-память до 1 Мбайт, до 192 Кбайт SRAM), до 4 Кбайт резервного SRAM, а также широкий спектр расширенных вводов-выводов и периферийных устройств, подключенных к двум APB шинам, трем шинам AHB и 32-битной матричной шине multi-AHB. Взято из источника [2].

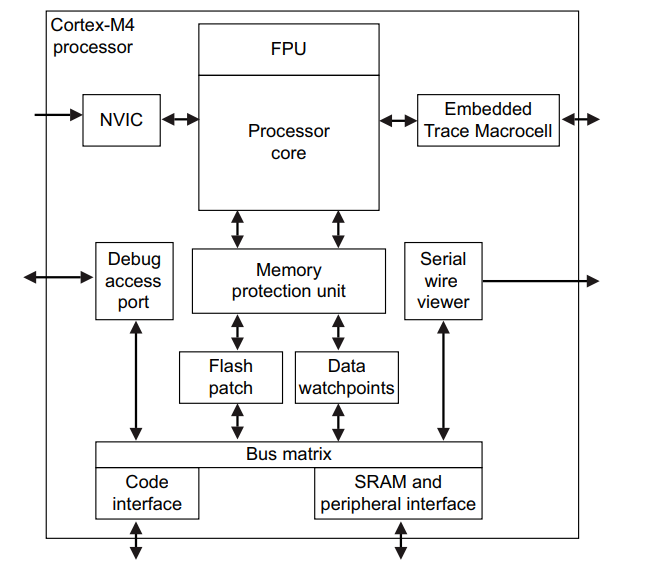


Рис.1 Реализация STM32 Cortex-M4 из источника [1]

В STM32 для некоторого устройства ассоциирован определенный участок памяти. Записывая в эти участки значения, можно управлять определенной периферией. При работе на Cortex-M4 c адреса 0х8000000 начинается Flash-память — это та память, которая будет использована для прошивки.

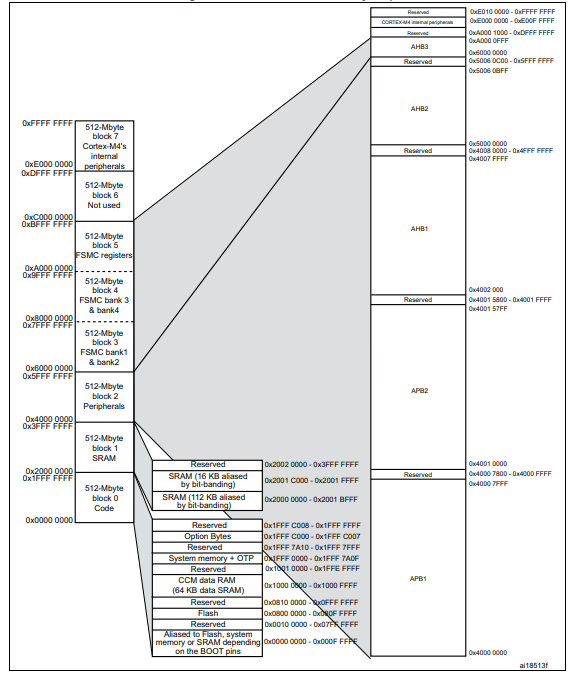


Рис.2 – Memory map(выделена Flash-память)

Также на рис.2 видно, что периферия настраивается по 4 шинам: AHB<1/2> или APB<1/2>.

На нашей плате присутствует 3 диода: D1, D2, D3. D1 используется для обозначения питания на плате, его использовать не получится. D2 и D3 подключены к пину PA6 и PA7 соответственно(рис.3).

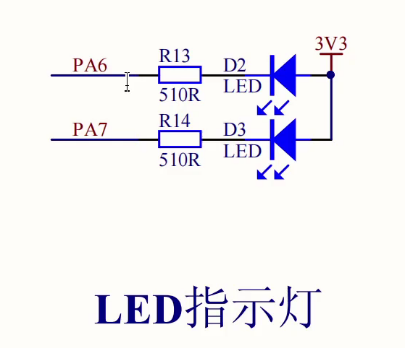


Рис. 3 – Схема диодов, источник [3]

Чтобы включить эти диоды, нужно включить тактирование на GPIO и направить сигнал на пин PA6 или PA7, в зависимости от выбора пользователя.

### Результаты работ

Чтобы включить тактирование на GPIO, необходимо посмотреть, как работает RCC\_AHB1ENR. В документации [4] видно смещение, по которому этот регистр располагается и так как необходим нулевой бит, записав туда единичку включиться тактирование для порта GPIOA.

Для данной работы был представлен примерный сценарий работы:

1. Ввести код операций:
   1. Включить диод.
   2. Выключить диод.
   3. Включить мигание диодом.
   4. Включить мигание диодом n раз.
   5. Задать частоту мигания.
2. Провести соответствующее действие в зависимости от введенного кода операции
3. Вывести в консоль сообщение о выполнении

Также для того, чтобы консоль на компьютере соединить с микроконтроллером, используется интерфейс USART.

В работе использовался STM32CubeIDE. Данная среда разработки позволяет настроить конфигурацию, написать программу и прошить ее для STM32 контроллера.

Весь проект доступен по ссылке: [https://gitlab.com/iliyvas/mps2022/-/tree/Maslakov/Coursework/MaslakofArtemИзображение выглядит как текст

Автоматически созданное описание](https://gitlab.com/iliyvas/mps2022/-/tree/Maslakov/Coursework/MaslakofArtem)

Рис. 4 – Настройка USART

На рис.4 показана конфигурация USART.

Далее был подключен RCC и настроен GPIO на выход пина PA6 (рис. 5).

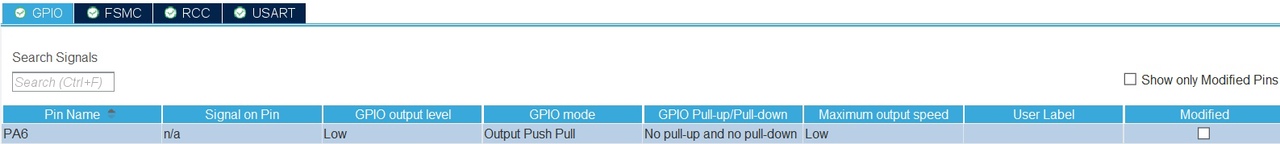


Рис. 5 – Конфигурация GPIO

После запуска программы в контроллер прошивается наша программа и к плате можно подключиться по консоли. Для реализации общения через консоль была выбрана консоль RealTerm.

Для корректной работы с консолью, надо ее настоить. Необходимо выбрать скорость обмена данными и порт через который будет происходить обмен.

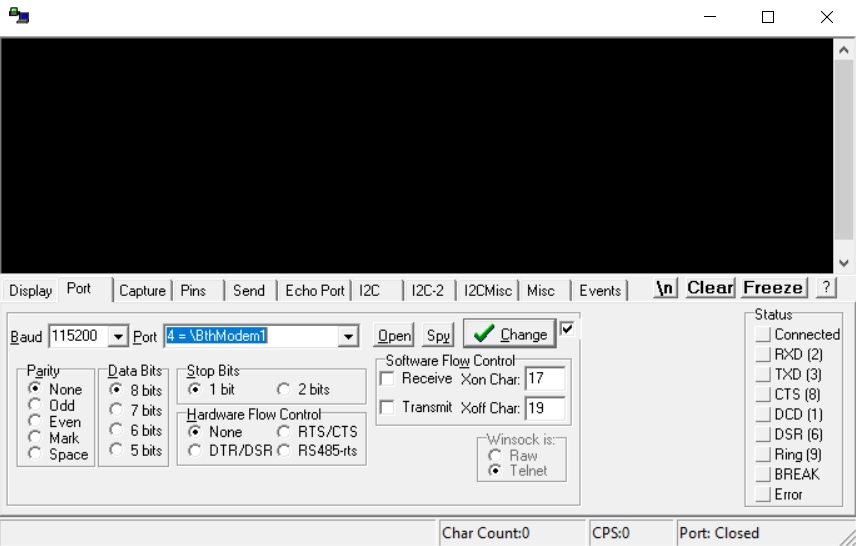


Рис. 6 – Настройка консоли

Чтобы ввести команду и отправить ее на исполнение используем поле ввода в консоли. Для корректной работы необходимо поставить галочки у параметров +CR и +LF. Так же нужно настроить консоль на постановку символа “\n” в конце запроса.

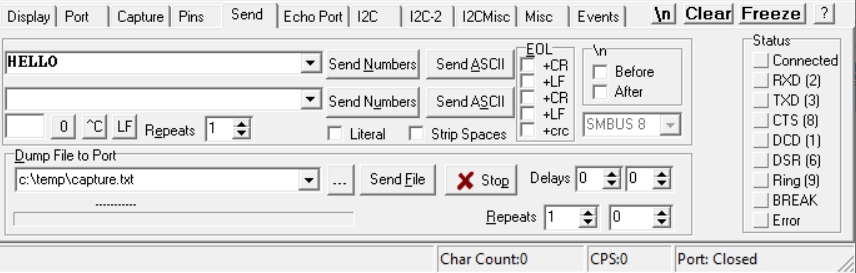


Рис. 7 – Настройка консоли и ввод команды

Чтобы команда отправилась на исполнение нужно нажать на кнопку Send ASCII.

Список команд:

1. On – Включение диода
2. Off – Выключение диода
3. Blink on – Включить контролируемое мигание диода
4. Blink off – Выключить контролируемое мигание диода
5. Flash – Включить мигание
6. Flash %d times – Мигать n раз
7. Impulse %d – изменить время горения диода.
8. Period %d – изменить период мигания диода

### Выводы

В ходе работы была изучена работа микроконтроллера семейства STM32 и проведены базовые операции по взаимодействию платы с компьютером посредством работы периферийных устройств. Также была изучена работа программы STM32CoreIDE.

### Список литературы

1. STM32 Cortex®-M4 MCUs and MPUs programming manual [Электронный ресурс] // URL: https://www.st.com/resource/en/programming\_manual/dm00046982-stm32-cortexm4-mcus-and-mpus-programming-manual-stmicroelectronics.pdf (дата обращения: 27.12.2022)
2. High-performance foundation line, Arm Cortex-M4 core with DSP and FPU, 512 Kbytes of Flash memory, 168 MHz CPU, ART Accelerator, Ethernet, FSMC [Электронный ресурс] // URL: https://www.st.com/en/microcontrollers-microprocessors/stm32f407ve.html (дата обращения: 27.12.2022)
3. Board-user manual [Электронный ресурс] // URL: https://stm32-base.org/assets/pdf/boards/original-schematic-STM32F407VET6-STM32\_F4VE\_V2.0.pdf (дата обращения: 27.12.2022)
4. Reference manual [Электронный ресурс] // URL: https://www.st.com/resource/en/reference\_manual/dm00031020-stm32f405-415-stm32f407-417-stm32f427-437-and-stm32f429-439-advanced-arm-based-32-bit-mcus-stmicroelectronics.pdf (дата обращения: 27.12.2022)