

Федеральное государственное автономное образовательное учреждение высшего
образования
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ

А.А. Ключарёв

Программирование микропроцессоров

методические указания по выполнению контрольной работы для студентов заочной
формы обучения направления 11030100 - Радиотехника

Санкт-Петербург 2020 г.

Дисциплина "Программирование микропроцессоров" направлена на получение студентами первичных знаний и умений в области использования современных микроконтроллеров при решении практических задач проектирования различных радиотехнических устройств.

В качестве базового устройства для изучения в курсе используется микроконтроллер 32 разрядный STM32F303VCTx, построенный на основе микропроцессора ARM Cortex M4. В теоретической части курса и лабораторном практикуме используются отладочный модуль STM32F3 Discovery, включающий этот микроконтроллер и оборудование, необходимое для его аппаратного программирования и отладки.

Для программирования микропроцессорных устройств в настоящее время широко используются различные инструментальные программные средства (среда разработки), ориентированные, как правило, на определенную группу микроконтроллеров. Часто эти инструментальные средства поддерживаются производителями соответствующих микроконтроллеров.

Общим для большинства современных сред разработки является преимущественное использование языка программирования C (C/C++), поэтому от специалиста, занимающегося программированием микроконтроллера знания этого языка является обязательным.

Цель выполнения **контрольной работы** состоит в самостоятельном освоении приемов программирования на языке C (C/C++) применительно к решению задач, характерных для программирования микроконтроллеров. Предполагается, что студент владеет базовыми знаниями и определенным опытом программирования, полученным при изучении дисциплины "Информатика", основами цифровой схемотехники и способен расширить эти знания при самостоятельной работе с литературой.

Выполнение контрольной работы не требует дополнительных технических средств, кроме персонального компьютера с операционной системой Windows версии 7 и выше.

Для написания, отладки и выполнения программ по индивидуальному заданию можно использовать любую среду программирования на языке C (C/C++), однако рекомендуется установить на компьютер программу STM32CubeIDE (<https://www.st.com/en/development-tools/stm32cubeide.html>), которая будет использоваться в лабораторном практикуме. Это позволит студенту, в том числе, познакомиться со средой программирования микроконтроллера, а если у студента есть доступ к отладочной плате, самостоятельно выполнить задания по лабораторным работам.

Список литературы, которая может использоваться для выполнения контрольной работы приведен в соответствующем разделе. Следует учитывать, что литературы по использованию языка C (C/C++) весьма много и выбор конкретной книги или интернет ресурса существенно зависит от начальной подготовки студента. Не рекомендуется использование изданий типа "..... для чайников". Следует обратить внимание на то, что подавляющее большинство доступной литературы ориентировано на объектно-ориентированный язык программирования C++ и предполагает либо знание языка C, либо ограниченное использование его возможностей при написании программ. Объектно-ориентированные технологии ограничено используются при программировании микроконтроллеров из-за стремления к получению наиболее эффективного кода программы, занимающей минимум памяти и обеспечивающего наибольшее быстродействие проектируемого устройства. Поэтому в рамках данного курса мы будем использовать стиль программирования языка C, являющегося по сути частью (подмножеством) языка C++. Часто в литературе это называют "старый стиль программирования". В этом плане полезна литература, имеющая в названии " C/C++". В ней обычно рассматривается "старый стиль программирования на C" как основа, затем выполняется переход на рассмотрение объектно-ориентированного программирования на C++. Вторую часть, как правило, при первом чтении можно опустить.

В приведенном ограниченном списке литературы по возможности дана характеристика издания применительно к задачам курса.

Язык программирования C (C/C++) обладает обширными возможностями, из которых в рамках данного курса следует обратить особое внимание на следующее:

- работа в заданной инструментальной среде программирования;
- базовые типы данных, используемые при описании переменных;
- базовые операции и выражения;
- операторы (в различной литературе могут называться "структуры управления", "управляющие конструкции");
- функции, правила их описания и использования;
- общие представления об указателях.

1. Рекомендации по выполнению контрольной работы

В рамках изучения дисциплины за основу взят современный микроконтроллер STM32F3, построенный на основе процессорного ядра ARM Cortex M4 [6,7]. Ознакомиться с приведенной литературой полезно, но объем материала существенно превышает объем, необходимый для изучения данной дисциплины. Имеется большое количество профессиональной литературы на английском языке и технической документации, представленной на сайте www.st.com.

STM32F3 представляет собой 32 разрядный процессор объединенный в одном кристалле с большим количеством различных по назначений программируемых периферийных устройств. Это позволяет проектировать на его основе самые разнообразные устройства, работающие, как правило, в реальном времени.

При работе микроконтроллер выдает и принимает сигналы через 16 разрядные порты ввода-вывода, обозначаемые как GPIO A, GPIO B и т.д.

При выполнении контрольной работы студент не имеет возможности использовать для программирования конкретный микроконтроллер. Это и не требуется, так как основная задача - освоить основные приемы программирования на языке C/C++. Поэтому выполняя индивидуальные задания необходимо объявить и использовать переменные **Port_A**, **Port_E** соответствующей разрядности. Эти переменные будут имитировать реальные порты ввода-вывода микроконтроллера. Операции ввода и вывода разрешается применять только к этим переменным.

Номера вариантов заданий для каждого студента заданы в соответствующем файле в личных кабинетах.

Рекомендации по настройке среды STM32CubeIDE применительно к выполнению задания приведены в приложении.

В лабораторном практикуме будет использоваться функция задержки вывода данных на заданное количество миллисекунд. В контрольной работе рекомендуется использовать аналог этой функции Sleep() в стандартном C/C++. Аргументом этой функции является число, указывающее на величину задержки, прототип функции описан в файле `#include <stdio.h>`.

Для ознакомления со средой разработки первоначально рекомендуется подготовить и выполнить простую программу, 10 раз выводящую на консоль фразу "!!!Hello World!!!" с задержкой 200 мс.:

```
#include <stdio.h>
#include <stdlib.h>
#include <windows.h>

int main(void) {
    int i = 0;
    while ( i <10)
    {
        Sleep(200);
        puts("!!!Hello World!!!"); /* prints !!!Hello World!!! */
        i++;
        Sleep(200);
    }
    return EXIT_SUCCESS;
}
```

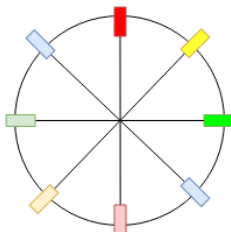
2. Варианты заданий

В основу вариантов заданий по контрольной работе положены соответствующие варианты для выполнения лабораторной работы №1. Предполагается, что при выполнении лабораторной работы студент сможет максимально воспользоваться результатами выполнения контрольной работы. На лабораторном макете используется пользовательская кнопка, подключенная к младшему разряду порта GPIOA и старшие разряды порта GPIOE - PE8 - PE15, к которому подключены 8 светодиодов, расположенных на макете по кругу.

В контрольной работе следует вместо портов ввода-вывода использовать без знаковые 16 разрядные переменные **Port_A**, **Port_E**. Ввод нечетного числа, код которого содержит "0" в младшем разряде, соответствует отжатому состоянию кнопки, нечетного - нажатому.

Вариант 1.

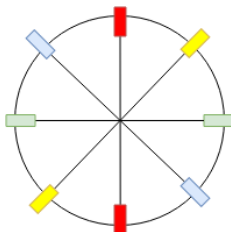
- 1) Включить 3 светодиода как показано ниже:



- 2) циклически сдвигать их с шагом 1 против часовой стрелки с задержкой 200 мс до нажатия кнопки.
- 3) циклически сдвинуть их с шагом 2 стрелке с задержкой 50 мс до отпущения кнопки.
- 4) непрерывно повторять шаги 2-3.

Вариант 2.

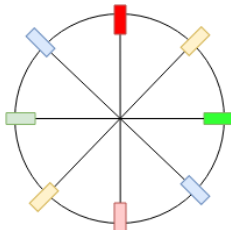
- 1) Включить 3 светодиода как показано ниже:



- 2) циклически сдвинуть их с шагом 2 по часовой стрелке с задержкой 150 мс до нажатия кнопки.
- 3) циклически сдвинуть их с шагом 1 по часовой стрелке с задержкой 50 мс до отпущения кнопки.
- 4) непрерывно повторять шаги 2-3.

Вариант 3.

- 1) Включить 2 светодиода как показано ниже:

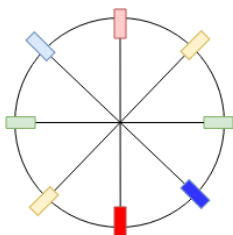


- 2) циклически сдвинуть их с шагом 4 против часовой стрелки с задержкой 300 мс до нажатия кнопки.
- 3) циклически сдвинуть их с шагом 1 против часовой стрелки с задержкой 50 мс до отпущения кнопки.

- 4) непрерывно повторять шаги 2-3.

Вариант 4.

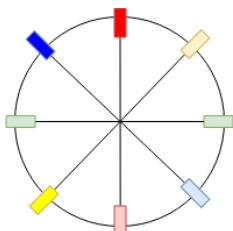
- 1) Включить 2 светодиода как показано ниже:



- 2) циклически сдвинуть их с шагом 2 против часовой стрелки с задержкой 200 мс до нажатия кнопки.
- 3) циклически сдвинуть их с шагом 1 по часовой стрелке с задержкой 70 мс до отпущения кнопки.
- 4) непрерывно повторять шаги 2-3.

Вариант 5.

- 1) Включить 2 светодиода как показано ниже:



- 2) циклически сдвинуть их с шагом 7 по часовой стрелке с задержкой 300 мс. При нажатии кнопки анимация останавливается.
- 3) При отпущении кнопки выполняется циклический сдвиг анимации с шагом 1 против часовой стрелке с задержкой 500 мс.
- 4) При нажатии кнопки на время до 0,2 с. анимация повторяется с шага 2.

Вариант 6.

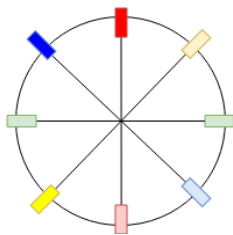
Реализовать на выходе порта GPIOE двоичный счетчик с коэффициентом счета 7, изменяющий свое состояние через каждые 30 мс. При переполнении счетчика счет останавливается, на линейке светодиодов загорается состояние счетчика, при нажатии на кнопку счет продолжается.

Вариант 7.

Реализовать на выходе порта GPIOE два последовательно соединенных двоично-десятичных четырехразрядных счетчиков. изменяющих свое состояние через каждые 50мс. При состоянии счетчика 88 (десятичное значение) счет останавливается. После нажатия на кнопку счетчик изменяет направление счета.

Вариант 8.

- 1) Включить 2 светодиода как показано ниже:



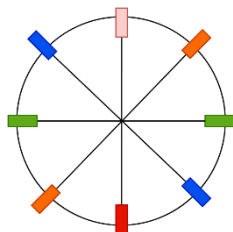
- 2) циклически сдвигать их с шагом 2 с задержкой 100 мс до нажатия на кнопку.
- 3) циклически сдвинуть их с шагом 1 против часовой стрелке с задержкой 200 мс.
- 4) непрерывно повторять шаги 2-3, при каждом повторении увеличивать задержку на 100 мс.
- 5) при достижении задержки на шаге 2 величины 1с, задать исходные значения задержек.

Вариант 9.

Реализовать на выходе порта GPIOE реверсивный двоичный счетчик, изменяющий свое состояние через каждые 30 мс. Вначале суммирующий режим. При нажатии на кнопку направление счета изменяется на противоположное.

Вариант 10.

- 1) Включить 7 светодиода как показано ниже:



- 2) реализовать режим, сдвигающий светодиоды, по часовой стрелке с задержкой 100 мс.
- 3) при нажатии кнопки, задержка должна возрасть до на 200 мс.
- 4) при достижении задержки в 1500 мс, ее необходимо сбросить обратно в 100 мс.

3. Требования к оформлению контрольной работы.

Контрольная работа оформляется в виде текстового файла и выкладывается в личный кабинет студента. Для зачета контрольной работы преподавателю обязательно должен быть представлен работающий программный проект и файл отчета. Требования к содержанию отчета:

- 1) титульный лист;
- 2) цель работы;
- 3) задание. В данном пункте должно содержаться: общие для всех заданий положения, и индивидуальное задание.
- 4) результаты выполнения: листинг кода. Достаточно привести только код функции main и других ваших вспомогательных функций;
- 5) выводы.

Если при выполнении работы у студента появились вопросы, следует обращаться по адресу ak@aanet.ru к доценту Ключарёву Александру Анатольевичу.

Рекомендуемая литература

1. Шмидский Я.К. программирование на языке C/C++. Самоучитель. : - М.: Издательский дом "Вильямс", 2004.-352 с.

Комментарии: Хорошая книга с четкими объяснениями основных положений и большим количеством примеров. Важно, что она написана как самоучитель и, соответственно, позволяет освоить материал при самостоятельном изучении.

2. Хабибулин И.Ш. Программирование на языке высокого уровня. C/C++. - СПб.: БХВ - Петербург, 2006.-512 с.

Комментарии: Учебник, предназначенный для студентов, начинающих изучать программирование, отличающийся достаточно большой глубиной изложения материала и методически продуманной его организацией.

3. Павловская Т. А. C/C++. Программирование на языке высокого уровня: Учебник. - СПб. : ПИТЕР, 2003. - 461 с.

Комментарии: Учебник, рекомендованный для студентов технических специальностей, рекомендуется студентам, имеющим небольшой опыт программирования. Учебник имеется в достаточном количестве в библиотеке ГУАП.

4. Кучин Н.В., Павлова М.М. Основы программирования на языке СИ:Учебное пособие/СПбГУАП.- СПб, 2001, 86 с.

Комментарии: Учебное пособие, подготовленное преподавателями ГУАП. Достаточно полно и кратко изложены основные аспекты написания программ на языке C.

5. Культин Николай. C/C++ в задачах и примерах, 2-е издание. – СПб.: БХВ-Петербург, 2003. – 288 с.: ил.

Комментарии: Сборник задач по программированию на C/C++ , с объяснениями решений. Полезен для самостоятельной работы студента. В книге приведены задачи направленные на освоение основных аспектов программирования и примеры их решения. В последней главе книги приведен достаточно полный справочник по стандартным функциям языка.

6. Мартин М. Инсайдерское руководство по STM32 (без выходных данных)

Комментарии: Книга не имеет выходных данных, но свободно доступна для скачивания в ИНТЕРНЕТ ресурсах. Достаточно полное руководство по основам микроконтроллеров STM32 и одна из немногих, представленных на русском языке.

7. Джозер Ю. Ядро Cjrtex-M3 компании ARM. Полное руководство. -М.: Додэка-XXI, 2012.- 552с.

Комментарии: Профессиональное издание для специалистов, занимающихся разработкой микропроцессорных устройств.