

## ЛАБОРАТОРНАЯ РАБОТА №1

### Программирование микроконтроллеров STM32F4 с использованием среды STM32CubeIDE

*Цель работы:* освоить программирование микроконтроллеров STM32 с применением конструктора кода CubeMX и высокоуровневых библиотек HAL; освоить работу с таймером и ЦАП микроконтроллера STM32F4.

Расчетная трудоемкость работы: 12 часов.

Предварительное задание: освежить знания о препроцессоре и синтаксисе языка C.

## 1. КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

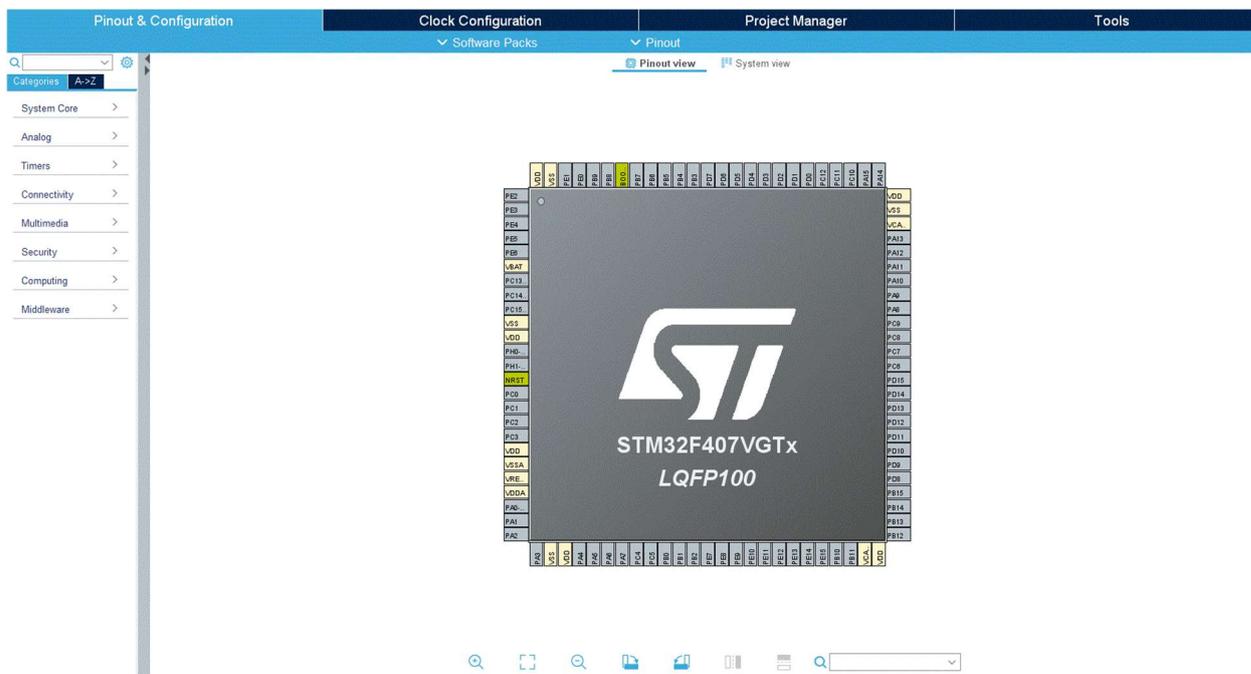
### *1.1 Создание проекта в STM32CubeIDE.*

В работе используется отладочный модуль STM32F4DISCOVERY с микроконтроллером STM32F407VG. Для разработки программного кода будем использовать среду STM32CubeIDE.

Для начала работы с программной средой запустите ее: *Пуск → Программы → САПР → STM32CubeIDE.*

При запуске программы укажите рабочее пространство. В каталоге *F:\CubeIDE* создайте папку со своей фамилией (все названия прописываем латиницей!). Создайте новый проект командой *File → New → STM32 Project.* В открывшемся диалоговом окне укажите тип контроллера (Part Number) STM32F407VG. На следующей странице нужно указать название проекта и его расположение. Создайте в указанном ранее рабочем пространстве папку с названием проекта. Название проекта должно совпадать с именем папки. В разделе *Options* выберите язык C, тип выходного файла Executable, тип проекта STM32Cube.

Перед вами откроется конструктор кода CubeMX.



### 1.2 Применение конструктора CubeMX

В среде программирования STM32CubeIDE встроен конструктор кода под названием CubeMX. Эта утилита позволяет перед созданием проекта сформировать код для всех необходимых настроек – задать схему тактового питания, настроить альтернативные функции выводов, выполнить настройки периферийных модулей на требуемый режим работы.

По умолчанию пользователь находится на вкладке Pinout&Configuration, которая позволяет выполнить необходимые настройки проекта и сгенерировать программный код, который сформирует эти настройки. Внимательно изучите все настройки, которые можно произвести на вкладках Pinout&Configuration и Clock Configuration.

В текущей работе нам понадобится цифро-аналоговый преобразователь микроконтроллера. Включите его и выберите для него электрическую линию. Изучите прочие возможные настройки.

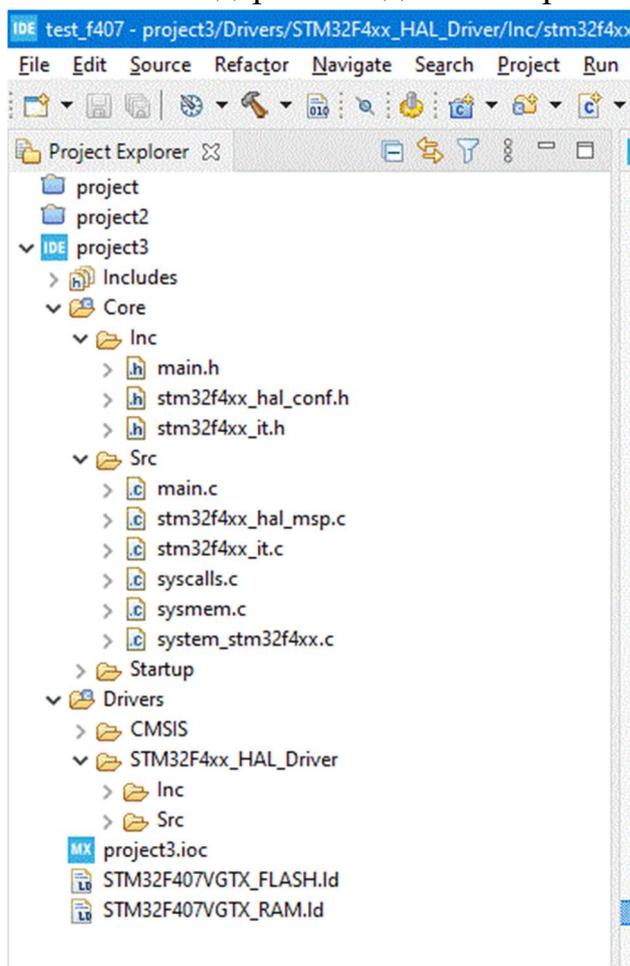
Для завершения работы конструктора выполните команду *Project* → *Generate Code*. Сформируется дерево проекта, в который будут добавлены необходимые библиотеки, а для настройки параметров будет сгенерирован программный код.

### 1.3 Применение библиотек HAL

Среда CubeIDE содержит встроенную библиотеку HAL, которая облегчает процесс разработки приложений. HAL (Hardware Abstraction Layer) – это высокоуровневая библиотека, предназначенная для разработки программного кода, не зависящего от марки конкретного микроконтроллера.

Это дает возможность разработчикам переносить написанный программный код в неизменном виде на другой вид контроллеров.

Рассмотрите внимательно дерево созданного проекта.



Во вкладке *Core* → *Src* находится основной файл *main.c*. В нем уже сгенерирован код для настройки цифро-аналогового преобразователя. Во вкладке *Drivers* → *STM32F4xx\_HAL\_Driver* в папках *Inc* и *Src* найдите файлы *stm32f4xx\_hal\_dac* с расширением *.h* и *.c* соответственно. Эти файлы представляют собой библиотеку HAL для работы с ЦАП микроконтроллера. Внимательно изучите функции, приведенные в этих файлах. Обратите внимание на то, как они структурировано их название, как они построены и какую функцию реализуют, какие переменные необходимо указать для выполнения этих функций, что они возвращают в качестве результата. При необходимости, просмотрите включенные заголовочные файлы. При возникновении вопросов относительно настройки и работы ЦАП, обратитесь к Reference manual микроконтроллеров STM32F4.

#### 1.4 Разработка проекта

После окончания подготовительных действий, описанных в пп.1.1-1.3, перед вами окажется созданный шаблон проекта, в котором уже выполнены настройки всей необходимой периферии микроконтроллера.

Разработайте проект согласно заданию. Код проекта пишете в файле *main.c* в основной функции проекта *main()*. Для компиляции проекта выполните команду *Project* → *Build All*.

### 1.5 Программирование отладочного модуля

После завершения разработки программного кода, его необходимо проверить на отладочном модуле. Для этого подключите отладочный модуль к порту USB компьютера, используя шнур mini-USB (не путать с micro-USB!). От этого порта модуль получает питание и прошивку одновременно.

Для подключения среды к модулю зайдите в меню *Run* → *Debug Configurations* и на вкладке *Debugger* выберите интерфейс SWD и программатор ST-LINK. Нажмите *Debug*.

Для прошивки проекта в отладочный модуль выберите меню *Run* → *Run*.

## 2. ПРОГРАММА РАБОТЫ

2.1 Создайте проект по схеме, приведенной в п.1.1.

2.2 Задайте в конфигураторе CubeMX настройки, как показано в п.1.2.

2.3 Откройте библиотеку HAL для ЦАП, изучите функции, приведенные в ней.

2.4 На выходе ЦАП реализуйте синусоиду с дискретностью 20 точек на период.

2.5 Реализуйте работу ЦАП по таймеру таким образом, чтобы частота выходной синусоиды составляла 1 кГц.

## 3. СОДЕРЖАНИЕ ОТЧЕТА

1. Название и программа работы.
2. Задание
3. Код программы с подробными комментариями
4. Выводы по работе

## 4. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Что представляют собой элементы библиотеки HAL? В виде каких файлов они содержатся в среде?
2. Как подключить необходимую библиотеку HAL для той или иной периферии микроконтроллера?
3. Как использовать функции библиотеки HAL в собственном проекте?
4. Как узнать, какие функции содержит данная библиотека HAL?
5. Как узнать, какие входные или выходные переменные необходимо указать для определенной функции библиотеки?