

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
ФГБОУ ВО АЛТАЙСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Физико-технический факультет

Кафедра вычислительной техники и электроники

**ИЗУЧЕНИЕ МИКРОКОНТРОЛЛЕРОВ ARM CORTEX-M СЕМЕЙСТВА
STM32 F0**

Методические указания по выполнению лабораторной работы

Барнаул 2018

Методические указания к выполнению лабораторной работы по курсу «Практикум по микропроцессорным системам» для студентов направления 09.03.01 «Информатика и вычислительная техника».

Данные методические указания созданы на основании требований рабочей программы дисциплины «Практикум по микропроцессорным системам».

Тема: Изучение микроконтроллеров ARM Cortex-M семейства STM32 F0.

Цель работы: изучение архитектуры и организации микроконтроллеров ARM семейства Cortex-M STM32. Получение навыков программирования, тестирования и отладки устройств на базе микроконтроллеров STM32F0 с использованием облачной среды разработки Mbed и отладочной платы Nucleo-L053R8.

Задачи: изучить архитектуру и организацию микроконтроллеров ARM Cortex-M семейства STM32. Изучить основы программирования микропроцессорных систем на базе микроконтроллера. В соответствии с заданием написать и отладить программы с помощью облачной среды программирования Mbed и произвести проверку работоспособности с помощью отладочной платы Nucleo-L053R8. Функционирование устройства проверить, используя измерительное и другое дополнительное оборудование.

Введение

Изучение программных средств микроконтроллеров невозможно без предварительного знакомства с их особенностями и аппаратными средствами. Прежде, чем приступить к программированию, следует четко представлять работу всех периферийных устройств МК, объем и структуру памяти, набор команд и т.п. В связи с этим сначала предлагается ознакомиться с микроконтроллерами ARM, семейством микроконтроллеров STM32F0 в целом и МК STM32L053R8 в частности.

1. Микроконтроллеры ARM

Архитектура ARM (от англ. Advanced RISC Machine - усовершенствованная RISC-машина; иногда – Acorn RISC Machine) – семейство лицензируемых 32-битных и 64-битных микропроцессорных ядер разработки компании ARM Limited. Среди лицензиатов - AMD, Apple, Analog Devices, Atmel, Xilinx, Altera, Cirrus Logic Intel, Marvell, NXP, STMicroelectronics, Samsung, LG, MediaTek, MStar, Qualcomm, Sony, Texas Instruments, nVidia, Freescale, Миландр, ЭЛВИС, HiSilicon. Первый процессор с архитектурой ARM был выпущен в 1985 году. В процессе развития и усовершенствования архитектуры фирма ARM представила пользователям целый ряд модификаций ядра, образующих семейства.

В наши дни 32-разрядная ARM — наиболее широко используемая архитектура процессоров для мобильных устройств, и наиболее популярная архитектура 32-разрядных процессоров для встраиваемых систем. Блок-схема архитектуры ARM приведена на рисунке 1.1.

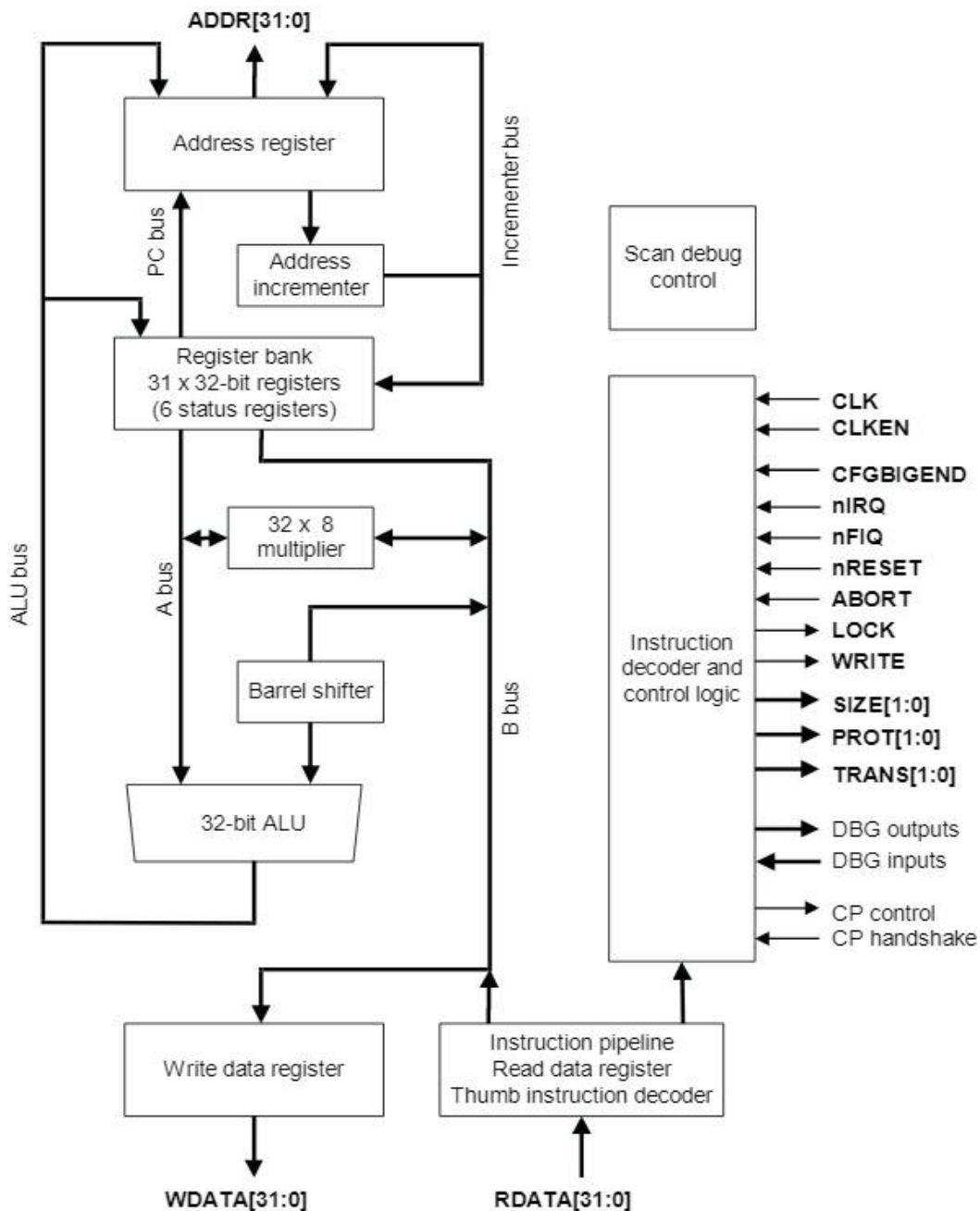


Рис.1.1 Блок-схема архитектуры ARM.

В настоящее время архитектуру ARM представляют два класса: «классические» процессоры, куда входят семейства ARM7, ARM9 и ARM11, и семейство Cortex, разделяемое на три «профиля»: Cortex-A, Cortex-R и Cortex-M. Основной задачей профиля A является достижение высокой вычислительной мощности. Изделия этой марки - Cortex-A - класси-

ческие микроконтроллеры (Snapdragon), которые являются дальнейшей эволюцией разработок ARM. Профиль R предназначен для использования во встраиваемых системах, поэтому эти процессоры обновляются для выполнения задач в реальном времени. Основной задачей профиля M является простота и низкая стоимость. Технически Cortex-M представляют сильно упрощенные версии более старших моделей. Однако даже такие упрощенные МК имеют вычислительную мощность, которая намного превышает другие аналоги. Также отличием от старших ARM стала поддержка битовых операций, необходимых в микроконтроллерах для работы с периферией.

В настоящее время распространение получили 3 варианта ядер: Cortex-M0, Cortex-M3, Cortex-M4. Сравнение ядер Cortex-M0, Cortex-M3, Cortex-M4 приведено на рисунке 1.2.

	Cortex-M0	Cortex-M3	Cortex-M4
Architecture Version	V6M	v7M	v7ME
Instruction set architecture	Thumb, Thumb-2 System Instructions	Thumb + Thumb-2	Thumb + Thumb-2, DSP, SIMD, FP
DMIPS/MHz	0.9	1.25	1.25
Bus interfaces	1	3	3
Integrated NVIC	Yes	Yes	Yes
Number interrupts	1-32 + NMI	1-240 + NMI	1-240 + NMI
Interrupt priorities	4	8-256	8-256
Breakpoints, Watchpoints	4/2/0, 2/1/0	8/4/0, 2/1/0	8/4/0, 2/1/0
Memory Protection Unit (MPU)	No	Yes (Option)	Yes (Option)
Integrated trace option (ETM)	No	Yes (Option)	Yes (Option)
Fault Robust Interface	No	Yes (Option)	No
Single Cycle Multiply	Yes (Option)	Yes	Yes
Hardware Divide	No	Yes	Yes
WIC Support	Yes	Yes	Yes
Bit banding support	No	Yes	Yes
Single cycle DSP/SIMD	No	No	Yes
Floating point hardware	No	No	Yes
Bus protocol	AHB Lite	AHB Lite, APB	AHB Lite, APB
CMSIS Support	Yes	Yes	Yes

Рис.1.2 Сравнительные характеристики ядер Cortex.

Микроконтроллеры Cortex-M являются одним из направлений развития микропроцессорных ядер, предлагаемых ARM. Компания STMicroelectronics (ST) является крупнейшим производителем микроконтроллеров в мире, при этом большая часть приходится на семейства STM32. Компания ST одной из первых (в 2007 г.) выпустила свои МК STM32 на основе ядра ARM Cortex-M0 и быстро стала лидирующим игроком на этом рынке. Продукция STMicroelectronics отличается от продукции других производителей полупроводников прекрасным соотношением цены и функциональности при сохранении самых высоких стандартов качества, и семейство STM32 — тому яркий пример.

Одним из базовых семейств от STMicroelectronics являются STM32F0, STM32F1, STM32F3. Данная группа включает семейства со сбалансированными характеристиками и компромиссным значением производительности/потребления/цены.

STM32F0 – наиболее маломощное с точки зрения производительности семейство. Оно построено на базе ядра Cortex®-M0 с рабочей частотой 48 МГц и имеет объем Flash-памяти 16...256 кбайт. Основное назначение этих микроконтроллеров – вытеснение 8- и 16-битных систем из традиционных бюджетных приложений. Стоимость отдельных представителей STM32F0 составляет всего 0,5\$. Архитектура одного из микроконтроллеров данного семейства показана на рисунке 1.3.

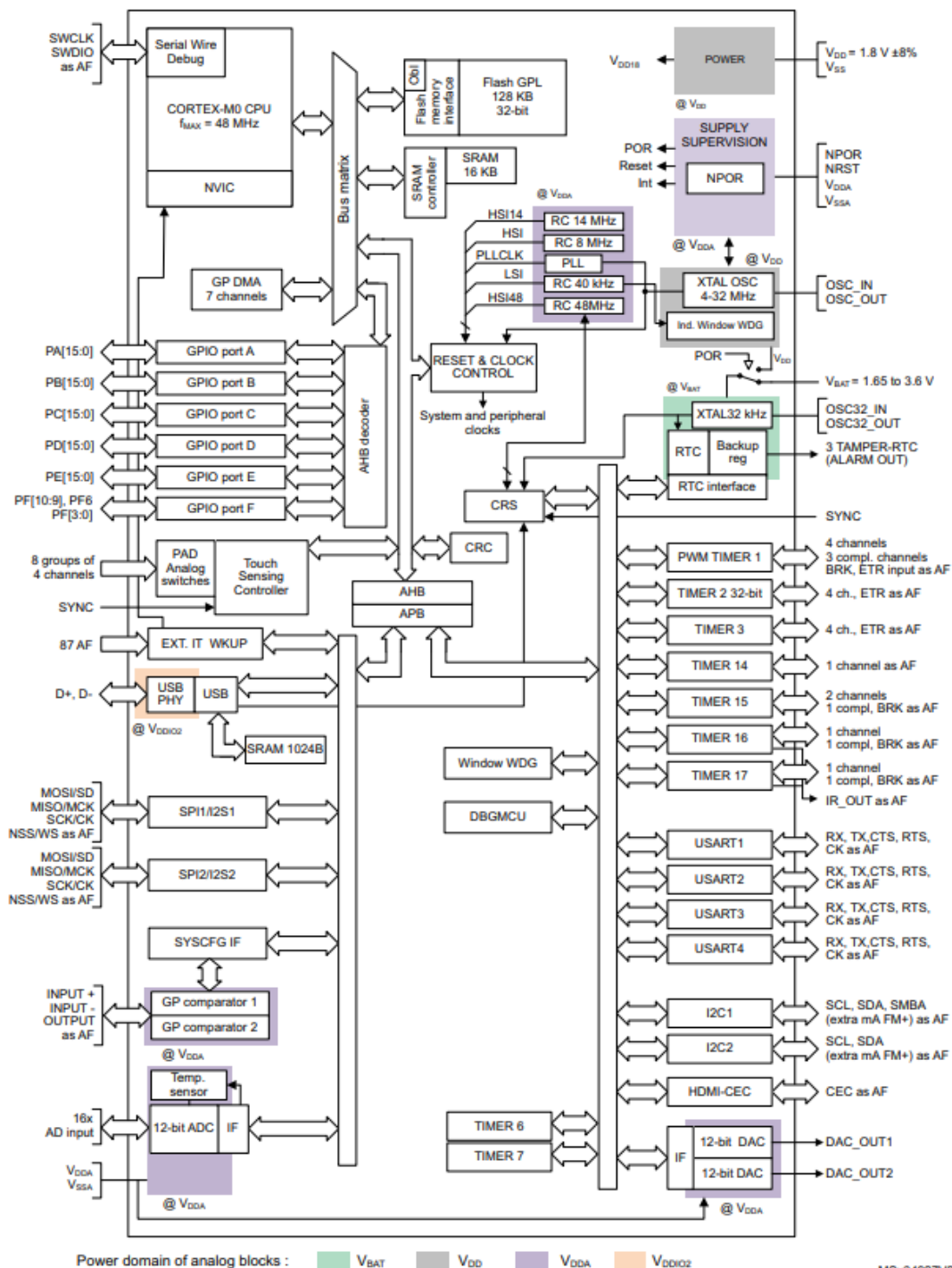


Рис. 1.3 STM32F078CB.

АЦП, сверхбыстродействующими АЦП последовательного приближения со скоростью преобразования до 5 миллионов выборок в секунду. На рисунке 1.5 показана архитектура микроконтроллера STM32F373xx.

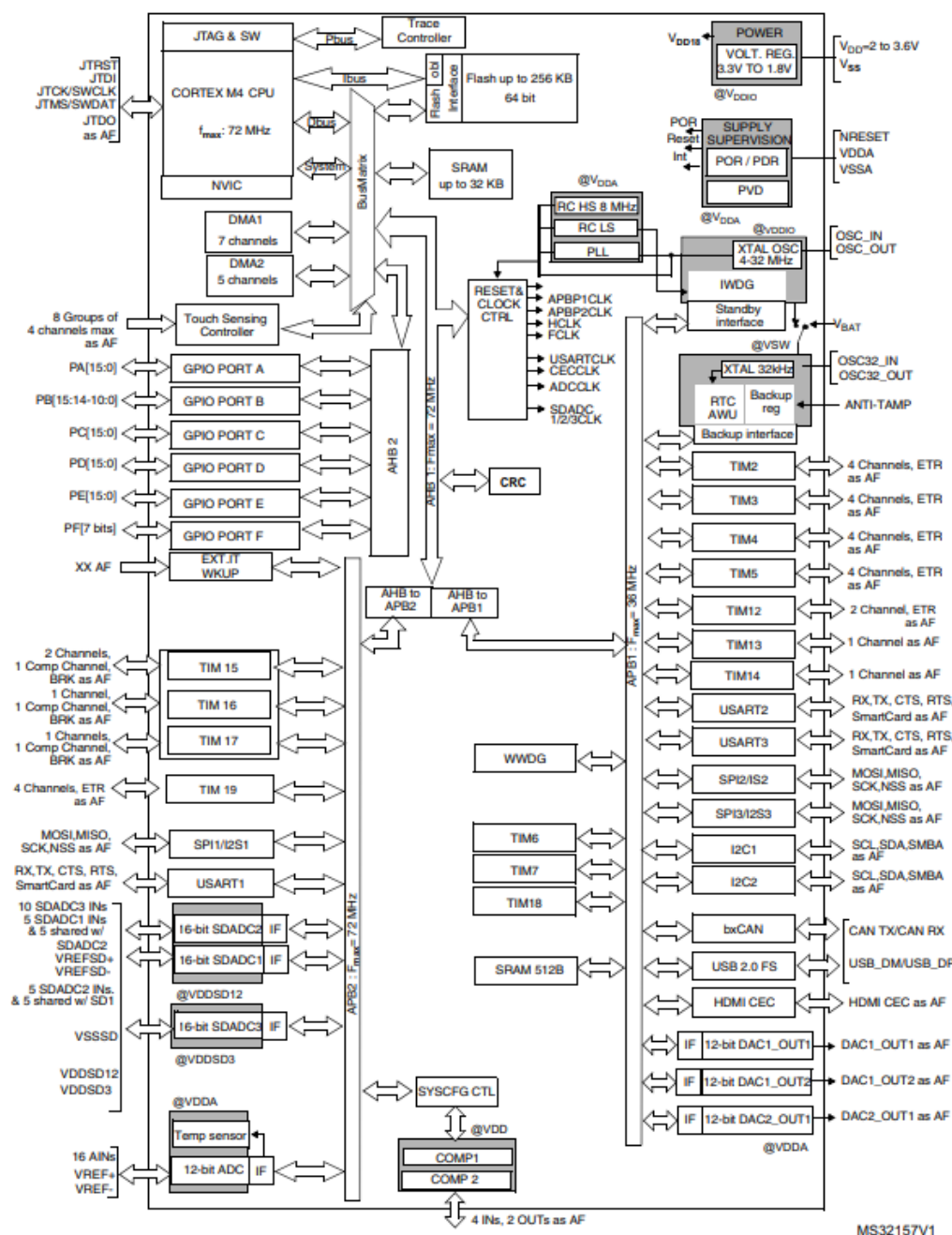


Рис.1.5 STM32F373xx.

Основная область применения 32-разрядных контроллеров:

- в вычислительной технике: контроллеры дисководов жестких и гибких дисков, CD и DVD, калькуляторах;

- электронике и разнообразных устройствах бытовой техники, в которой используется электронные системы управления — стиральных машинах, микроволновых печах, посудомоечных машинах, телефонах и современных приборах, различных роботах, системах «умный дом» и др.

Для выполнения лабораторных работ по курсу МПС предлагается использовать микроконтроллер фирмы STMicroelectronics семейства F0.

Контрольные вопросы

1. Опишите архитектуру процессоров ARM.
2. Сколько шин имеют процессоры ARM? Указать их особенности.
3. Сколько таймеров общего назначения, UART, портов ввода-вывода, каналов I2C, SPI, DAC, ADC, USB, CAN имеют различные представители семейства STM32F0, STM32F1, STM32F3?
4. Как устроен контроллер прерываний? Что такое вектор прерывания?
5. В чем отличие архитектуры ядер Cortex M0, Cortex M3, Cortex M4 фирмы STMicroelectronics (или архитектуры ARM)?
6. Назовите не менее 3х представителей каждого из семейств STM32F0, STM32F1, STM32F3.

2. Средства программирования

Чтобы научиться программировать микропроцессорную систему на базе микроконтроллера семейства STM32F0, необходимо знать архитектуру МК и языки высокого уровня, например, C/C++, а также изучить средства программирования и отладки.

В настоящее время вниманию разработчиков STM32 предлагается богатый выбор программных инструментов, в том числе такие мощные среды как ARM Keil или IAR. Они имеют широчайший функционал, позволяют создавать проекты и использовать огромное число готовых программных модулей (операционные системы, файловые системы, стеки популярных протоколов). Однако у них есть один большой недостаток – цена.

Компания STMicroelectronics предлагает несколько вариантов бесплатных программных сред разработки:

- ST MCU Finder (Рис. 2.1) – приложение для смартфонов, позволяющее на первом этапе выбрать оптимальный контроллер или отладочный набор.



Рис.2.1 ST MCU Finder.

- STM32CubeMX (Рис. 2.2)– графический редактор для конфигурирования микроконтроллеров STM32, позволяющий автоматически генерировать конфигурационный С-код и «скелет» прикладной программы с помощью набора визуальных утилит. Благодаря STM32CubeMX этап создания функций инициализации происходит очень быстро. При этом от программиста не требуется написание ни единой строки кода, хотя, конечно, алгоритмическую и смысловую части программы по-прежнему нужно создавать вручную.

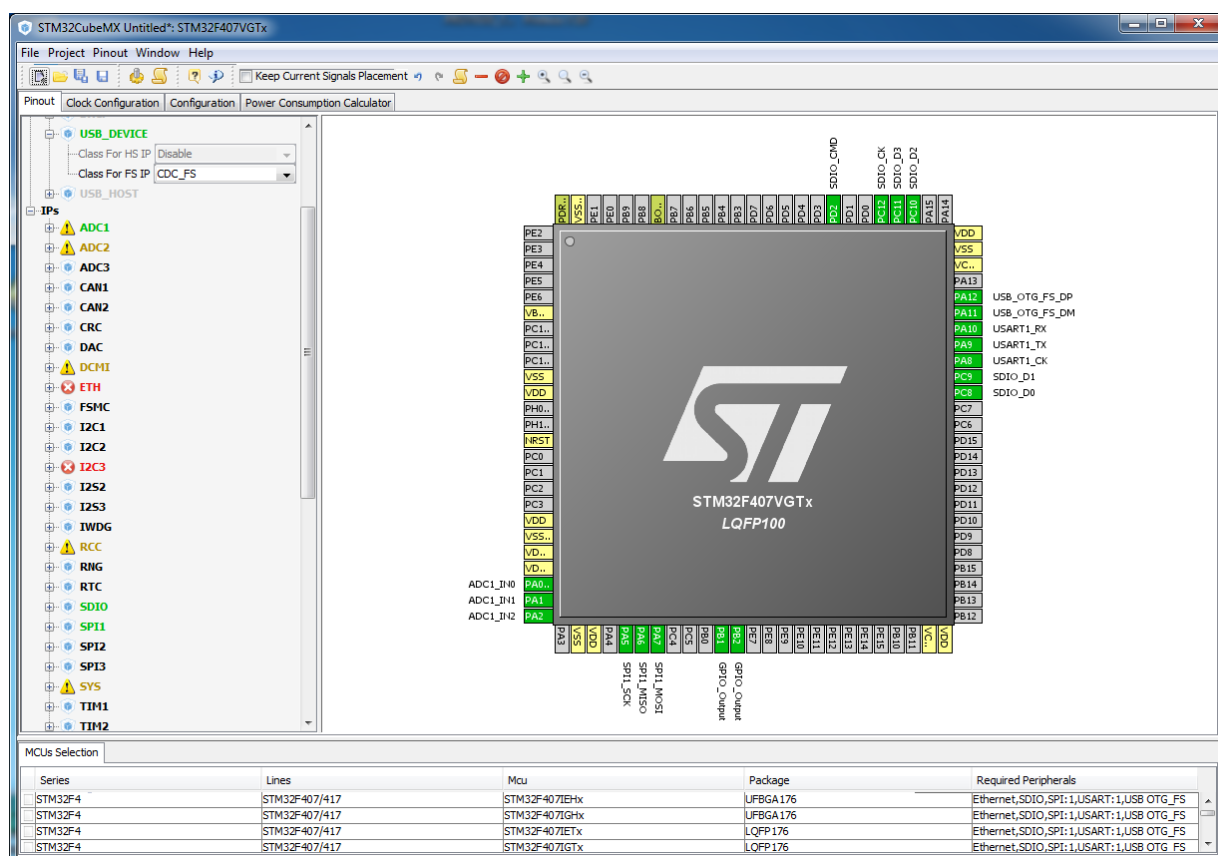


Рис.2.2 STM32CubeMX.

- SW4STM32 (Рис. 2.3) – интегрированная среда разработки встраиваемого ПО компании AC6 для микроконтроллеров STM32, позволяющая писать программы, загружать и отлаживать их.

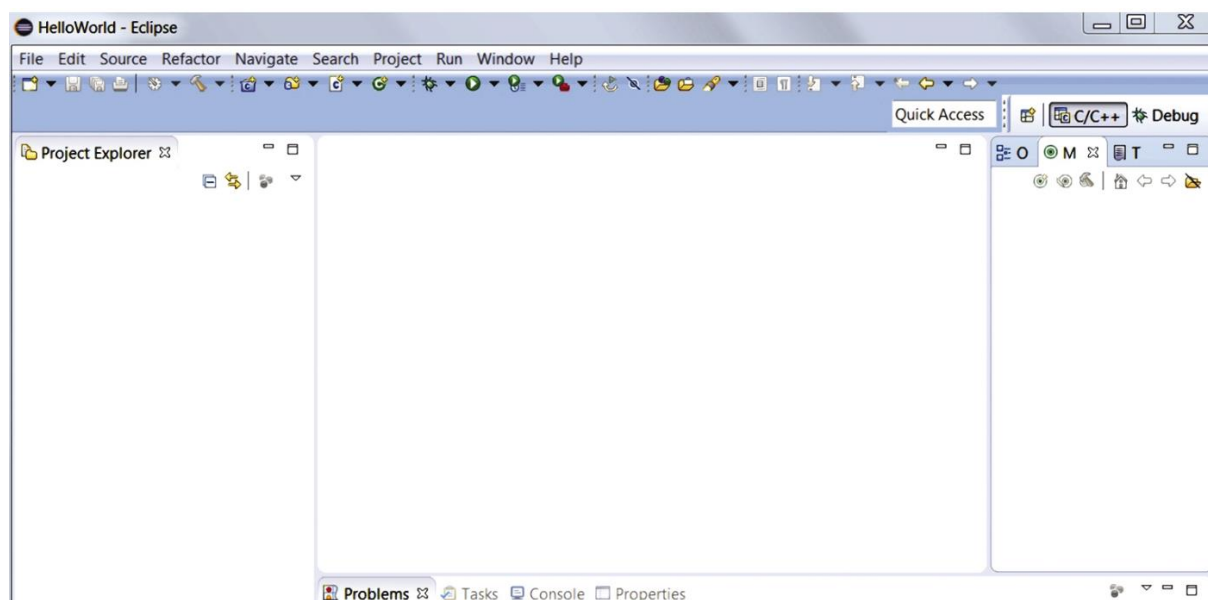


Рис.2.3 SW4STM32.

- STM Studio (Рис. 2.4) – фирменная утилита от STMicroelectronics, которая помогает отслеживать значения пользовательских переменных при выполнении кода в реальном времени.

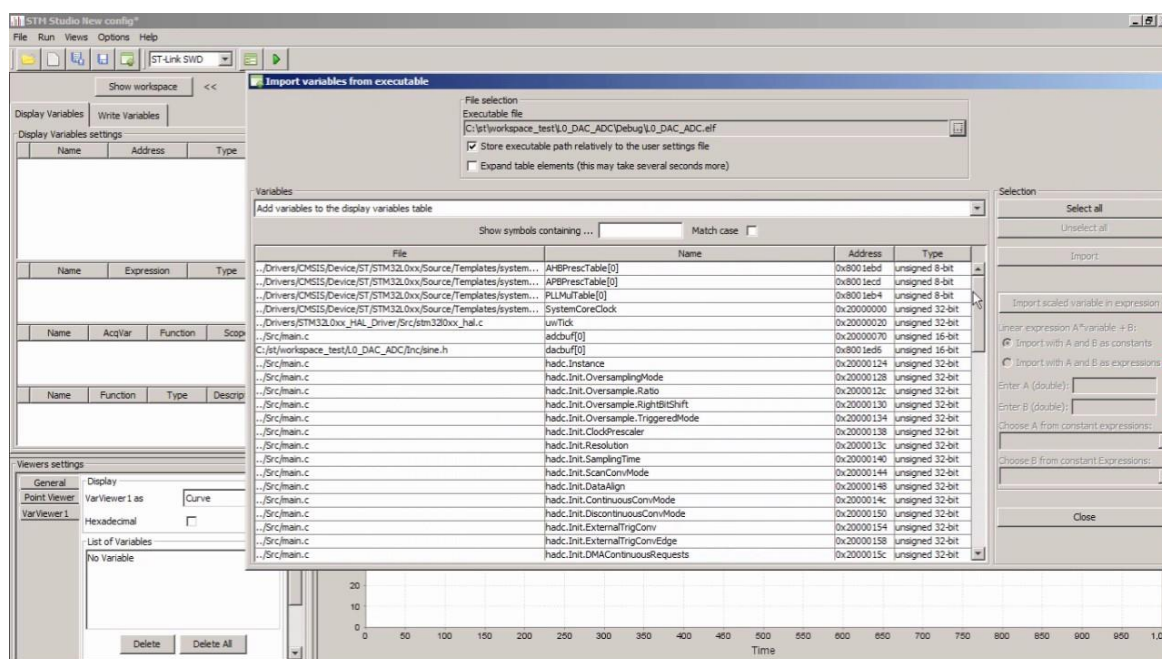


Рис.2.4 STM Studio.

Все эти среды доступны пользователям, но имеют один недостаток – они привязаны к одному ресурсу и располагаются локально, что не позволяет использовать среду в удаленном режиме или требует дополнительных ресурсов (например, подключения дополнительных утилит или внешних библиотек), поэтому было решено остановиться на использовании облачных технологий, которые сегодня активно развиваются и предлагаются ведущими фирмами-производителями как hardware, так и software. Крупнейшие мировые корпорации такие как Microsoft, Texas Instruments, Amazon, Google, STMicroelectronics давно начали развивать свои облачные платформы. С недавнего времени к ним присоединились и разработчики программного обеспечения для программирования 32-разрядных МК компании Keil.

Самые важные преимущества облачных технологий:

1. Затраты

Облачные вычисления позволяют избежать капитальных затрат на приобретение оборудования и программного обеспечения, настройку и эксплуатацию локальных центров обработки данных.

2. Скорость

Большинство облачных вычислительных служб предоставляются в режиме самообслуживания и по запросу, так что даже большие объемы вычислительных ресурсов можно подготовить за несколько минут, обычно всего за несколько щелчков кнопкой мыши.

3. Глобальный масштаб

Преимущества служб облачных вычислений включают возможность эластичного масштабирования. В контексте облачных служб это означает выделение необходимого объема ИТ-ресурсов (например, увеличение или уменьшение вычислительной мощности, объема хранилища или пропускной способности) тогда, когда это нужно, и в соответствующем географическом расположении.

4. Производительность

Для локальных центров обработки данных обычно требуются много стоек и серверов, а также настройка оборудования, обновление программного обеспечения и другая рутинная работа, которая отнимает много времени. Облачные вычисления позволяют избежать многих из этих задач.

5. Надежность

Облачные вычисления делают резервное копирование данных, аварийное восстановление и непрерывность бизнес-процессов более легкими и менее затратными, так как данные можно зеркалировать на нескольких дублирующих сайтах в сети поставщика облачных служб.

Подавляющее большинство технологий облачного формата подразделяются на 3: услуга инфраструктуры IaaS, услуга платформы PaaS и услуга обеспечения SaaS. Комплекс этих форм технологий называют информационным облачным стеком, поскольку они тесно связаны друг с другом.

Самой популярной средой для работы с микроконтроллерами является облачная среда разработки mbed. Компилятор mbed предоставляет легкую интерактивную C / C++ IDE, которая предварительно сконфигурирована, чтобы можно было быстро писать программы, компилировать и загружать их для запуска на имеющемся микроконтроллере. На самом деле это очень удобно, когда не нужно устанавливать или настраивать что-либо для работы с МК. Поскольку это веб-приложение, то можно входить в систему из любого места и продолжать работу с того места, где остановились. Еще одним плюсом использования данной среды является то, что можно работать в Windows, Mac, iOS, Android, Linux или во всех из них попеременно.

Именно поэтому выбрана облачная среда программирования Mbed – программно-аппаратная платформа и одноимённая операционная система для устройств на базе 32-разрядных микроконтроллеров семейства ARM Cortex-M. Платформа работает онлайн и содержит интегрированную среду разработки (IDE) включая текстовый редактор, компилятор, набор библиотек и примеры программного кода.

2.1. Запуск Mbed и создание файла проекта

Запуск программы Mbed осуществляется путем запуска любого браузера и перехода на сайт <https://os.mbed.com/>, после чего откроется окно среды разработки Mbed OS (Рис. 2.5).

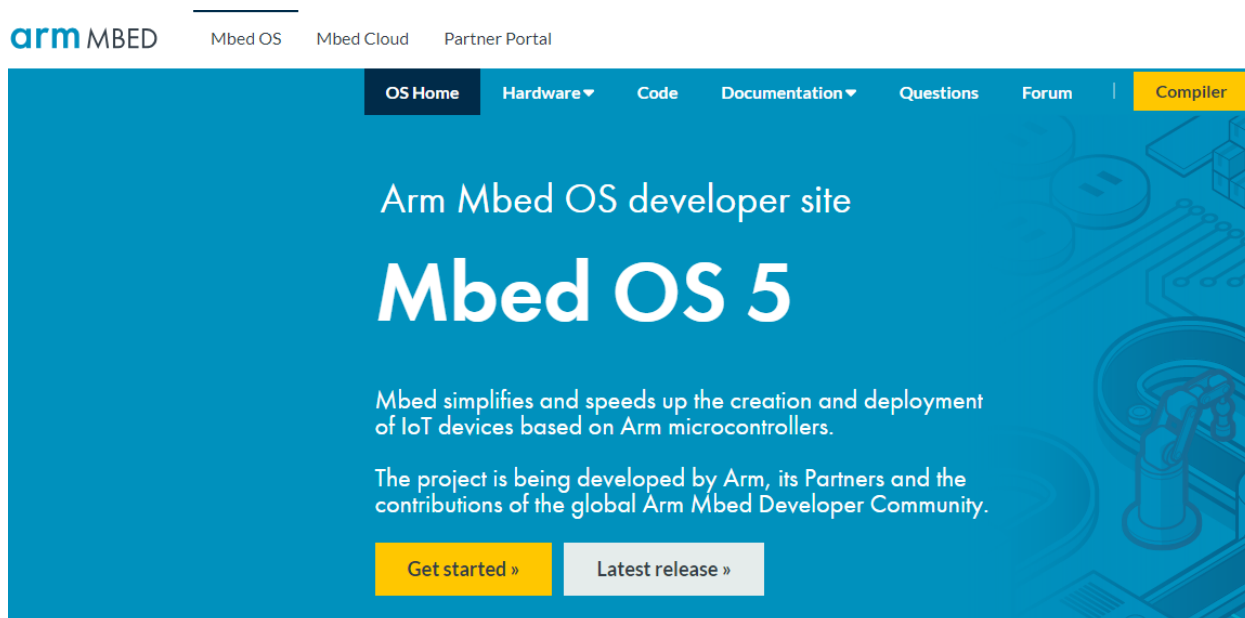


Рис. 2.5 Окно среды разработки Mbed.

Далее необходимо выбрать вкладку «Compiler» в верхнем правом углу (Рис. 2.6).

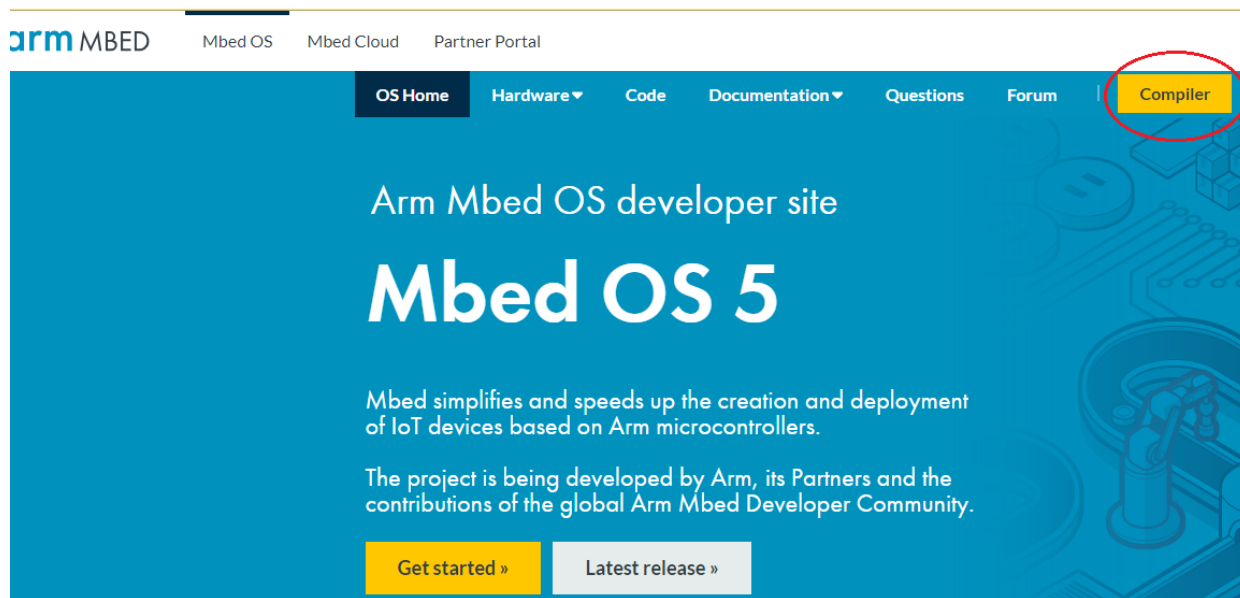


Рис. 2.6 Переход в режим разработки.

После перехода во вкладку компиляции откроется окно для создания личного аккаунта пользователя. Вводим свои данные (Рис. 2.7). Необходимо указать настоящий почтовый ящик. После окончания процедуры регистрации вам придёт письмо со ссылкой для подтверждения того, что введённый почтовый адрес действительно принадлежит вам.

Sign up

Create a free Mbed account to access Mbed tools and services and contribute to the developer community.

Email address:

Username:


Password (must be 6 characters or longer):

Confirm password:

First name:

Last name:

Country:

<input type="checkbox"/> I'm not a robot	 reCAPTCHA Privacy - Terms
--	---

Arm will process your information in accordance with our [Privacy Policy](#).

☐ By ticking this box you indicate your consent to receiving marketing communications from Arm in accordance with our Privacy Policy. Please visit our [Subscription Center](#) to manage your marketing preferences or unsubscribe from further communications.

☐ I agree to Arm's [terms and conditions of use](#). (required)

Sign up

Рис. 2.7 Окно регистрации

После регистрации откроется окно с приветствием. Необходимо войти в систему под вашим аккаунтом (Рис. 2.8).

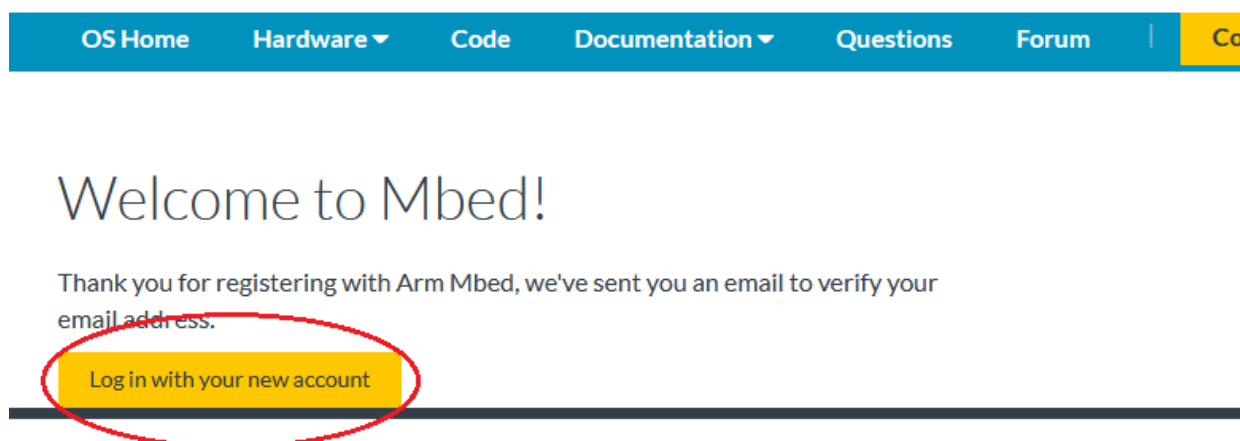


Рис. 2.8 Страница регистрации.

Окно авторизации созданного аккаунта пользователя показано на рисунке 2.9.

The image displays the login interface of the Arm Mbed platform. At the top center is the 'arm MBED' logo, with 'arm' in blue and 'MBED' in grey. Below the logo are two tabs: 'Log in' and 'Sign up'. The 'Log in' tab is selected and contains the following elements: a label 'Email or username' above a text input field with the placeholder 'Enter your email or username'; a blue link 'Forgotten your username?' below the input field; a label 'Password' above another text input field with the placeholder 'Enter your password'; and a blue link 'Forgotten your password?' below the second input field. At the bottom of the login form is a large, solid yellow button with the text 'Log In' in black.

Рис.2.9 Окно авторизации

После авторизации снова откроется окно платформы, но уже с аккаунтом пользователя. На рисунке 2.10 под номер 1 показан авторизованный пользователь. Далее необходимо перейти в режим разработки. Для этого в этом же окне необходимо выбрать Compiler (Рис. 2.10, номер 2).

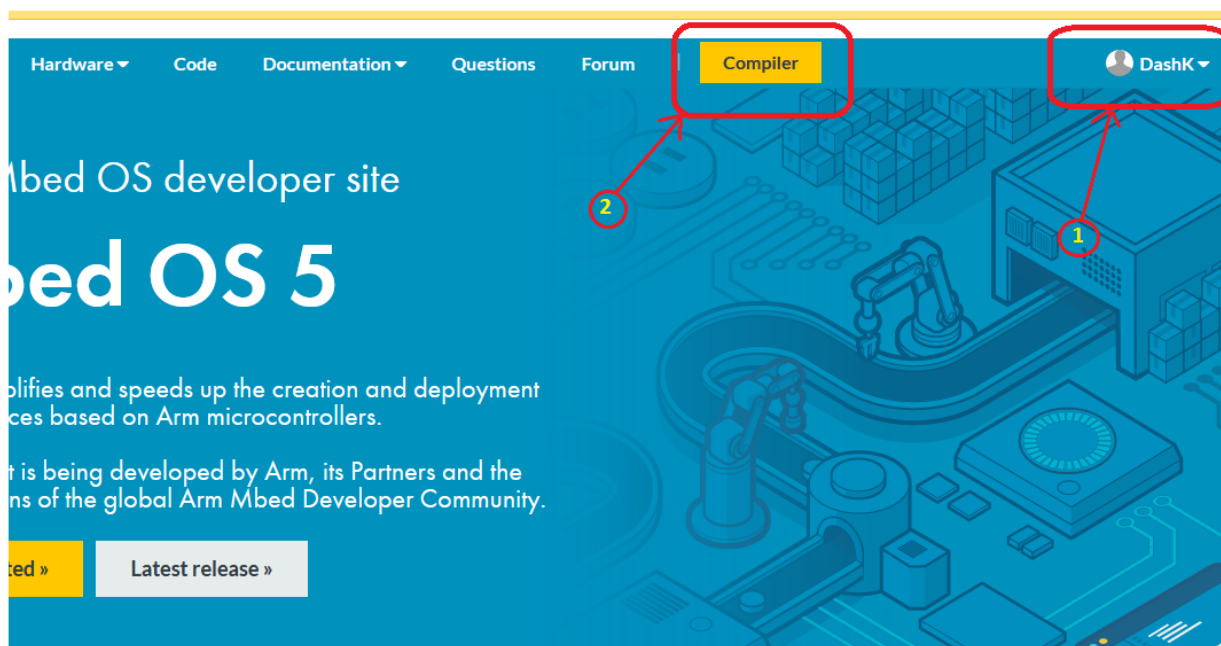


Рис.2.10 Окно платформы.

В открывшемся окне разработки можно начинать работу (Рис. 2.11).

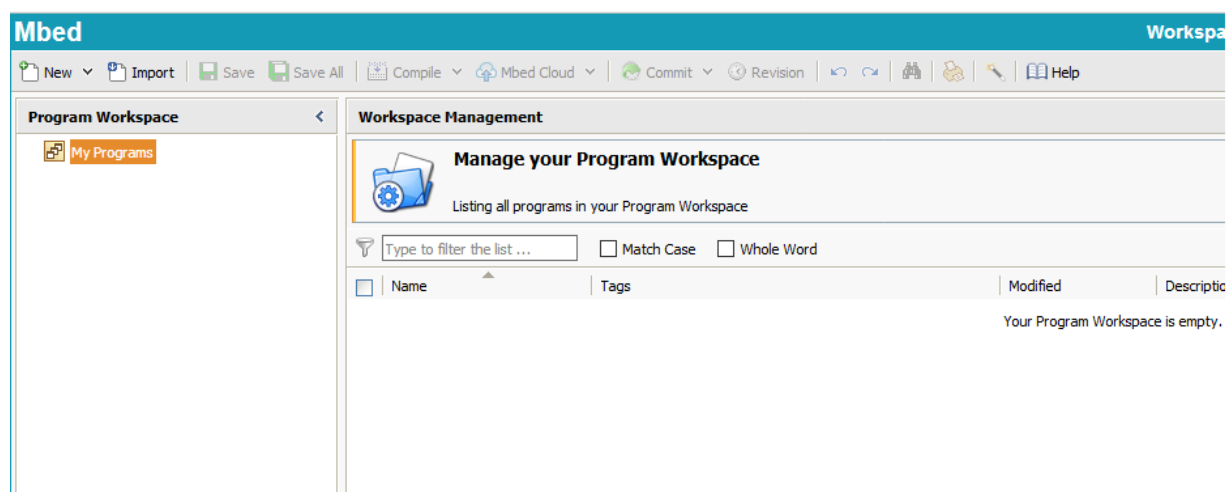


Рис. 2.11 Окно разработки.

Любая новая работа, как и во всех современных компиляторах, начинается с создания нового файла проекта. Файл проекта содержит имена всех исходных файлов, связанных с проектом, а также установки компиляции, трансляции и связывания файлов, чтобы генерировать выполняемую программу. IDE включает полный редактор кода, включающий подсветку синтаксиса, стандартные сочетания клавиш редактора, отмену / повтор, вырезание / копирование / вставку, вкладки, комментариев к блоку / строке и даже код автоматического форматирования.

Для того чтобы создать новый проект, необходимо выбрать New. Откроется диалоговое окно New Project. Здесь потребуется выбрать отладочную плату (Рис. 2.12). Для этого нужно кликнуть Add Platform в открывшемся окне.

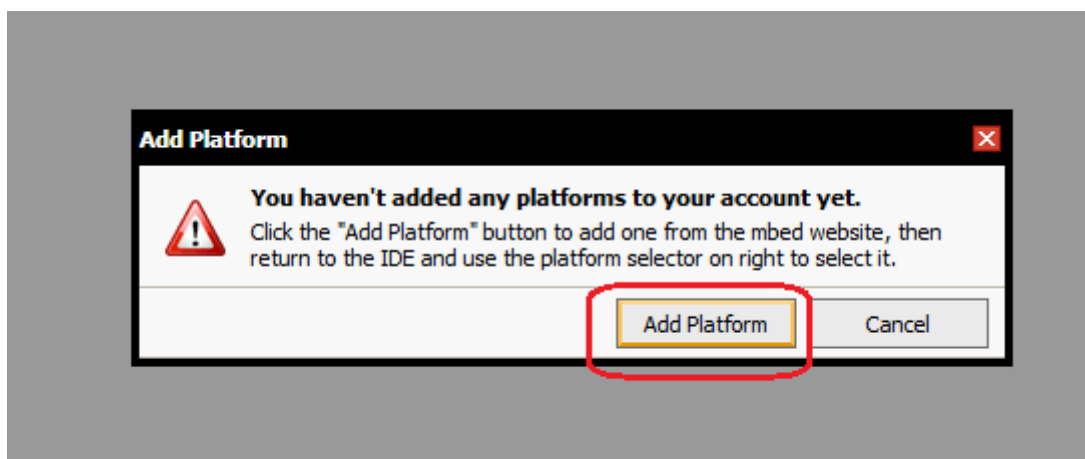


Рис. 2.12 Окно выбора платы.

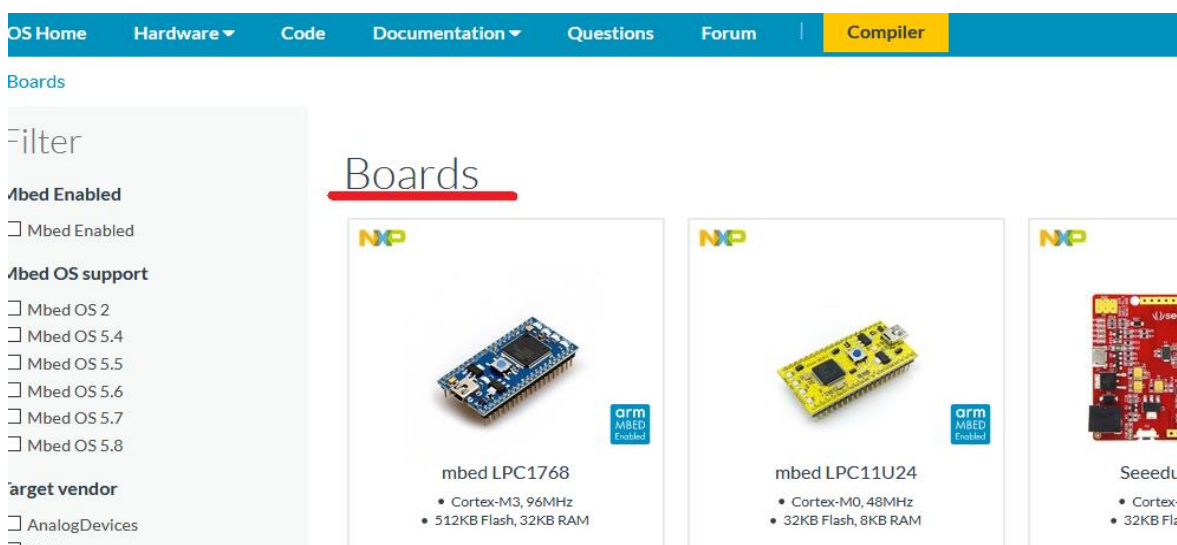


Рис. 2.13 Выбор отладочной платы.

В данной работе используется плата Nucleo-L053R8 фирмы STMicroelectronics. Чтобы быстрее найти нужную плату необходимо воспользоваться фильтром на странице списка отладочных плат. В разделе Platform Vendor нужно поставить галочку напротив STMicroelectronics (Рис. 2.14).

Platform vendor

- ☐ AnalogDevices
- ☐ ARM
- ☐ Avnet Silica
- ☐ BBC Make it Digital Campaign
- ☐ communitycontributors
- ☐ Delta
- ☐ Embedded Artists
- ☐ JKSoft
- ☐ L-Tek
- ☐ Maxim Integrated
- ☐ MultiTech
- ☐ Nordic Semiconductor ASA
- ☐ Nuvoton
- ☐ NXP Semiconductors
- ☐ Realtek
- ☐ RedBearLab
- ☐ Renesas
- ☐ SeeedStudio
- ☐ Semtech
- ☐ Silicon Labs
- ☒ STMicroelectronics
- ☐ Switch Science Inc.
- ☐ Toshiba

ST
live-augmented

ST
live-augmented

Рис. 2.14 Фильтрация отладочных плат.

В списке плат, которые определит фильтр необходимо выбрать отладочную плату Nucleo-L053R8 (Рис. 2.15).

Boards

Showing 42 of 138 ([Show all](#))

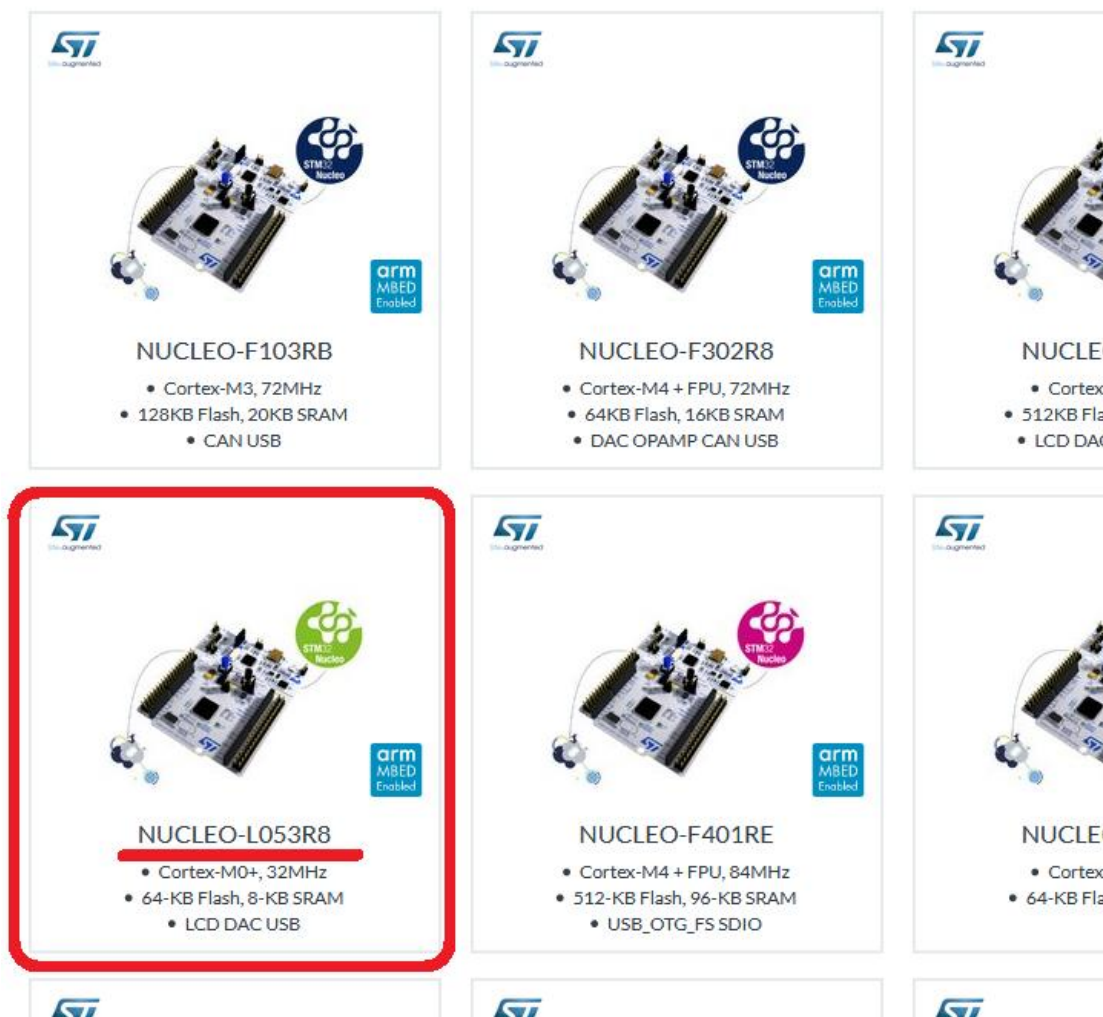


Рис.2.15 Выбор отладочной платы.

На открывшейся странице собрана вся необходимая информация по работе с платой Nucleo-L053R8. Чтобы добавить данную плату в основную среду разработки необходимо кликнуть Add to your Mbed Compiler (Рис. 2.16).

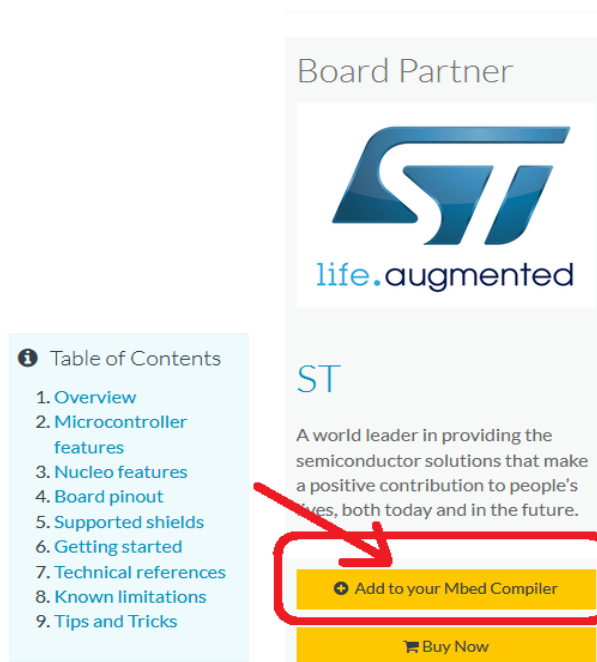


Рис. 2.16 Добавление платы в компилятор.

Для окончательного добавления платы в компилятор необходимо выбрать Open Mbed Compiler (Рис. 2.17).

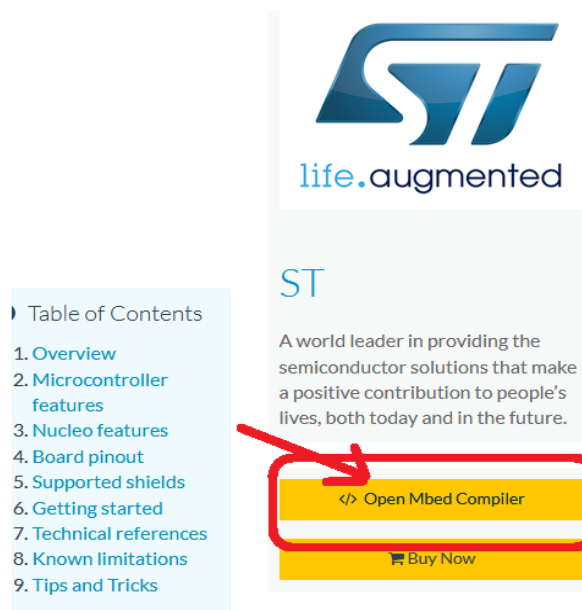


Рис. 2.17 Добавление платы в компилятор.

Когда менеджер проекта открывает файл проекта, окно проекта показывает включенные исходные файлы. В данном случае представлен набор готовых программ от STMicroelectronics для используемой отладочной платы Nucleo-L053R8 (Рис. 2.18).

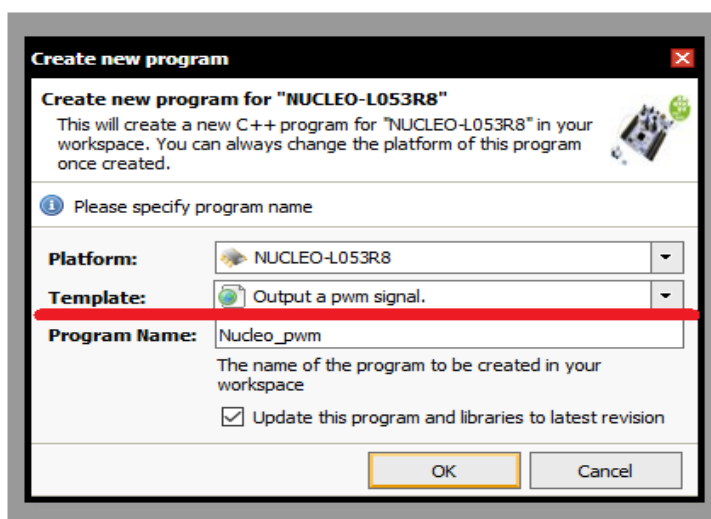


Рис. 2.18 Включенные исходные файлы.

2.2. Добавление файла с исходным текстом и его редактирование

Чтобы подключить файлы к проекту, нужно выбрать New Program из меню New (Рис. 2.19).

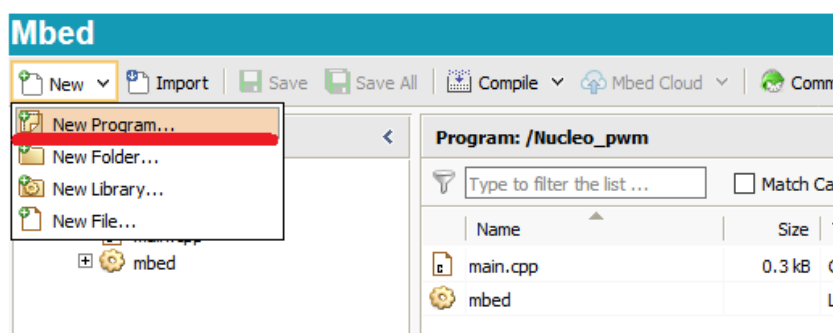


Рис. 2.19 Окно Program.

После чего откроется окно для выбора готовых решений (Рис. 2.20).

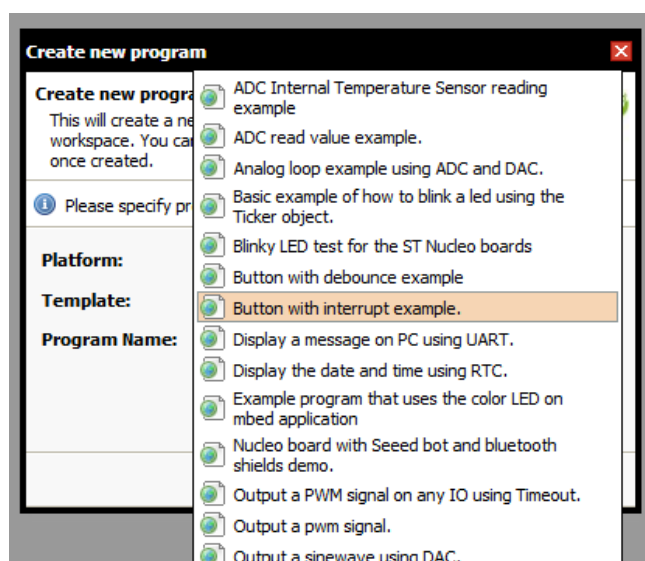


Рис. 2.20 Окно готовых решений.

Выбрать можно любую программу, после чего в окне компилятора появится и код самой программы (Рис. 2.21).

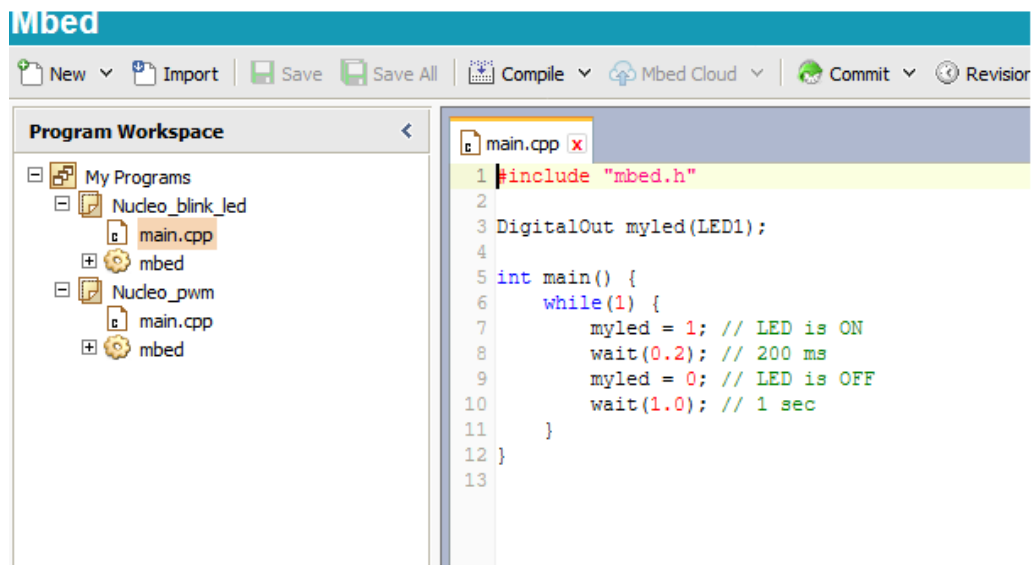


Рис. 2.21 Пример программы.

Теперь можно редактировать данный файл или компилировать и прошивать плату.

2.3.Компиляция и компоновка

Этот процесс компилирует, связывает исходные файлы с библиотеками и создает абсолютный объектный модуль, который можно проверить на отладочной плате. Достаточно просто подключить плату по mini USB кабелю к компьютеру и скачать на нее скомпилированный скетч как на обычную флэш-карту.

2.4.Тестирование и отладка

Чтобы запустить отладчик программы требуется выбрать пункт Compile (Рис. 2.22).

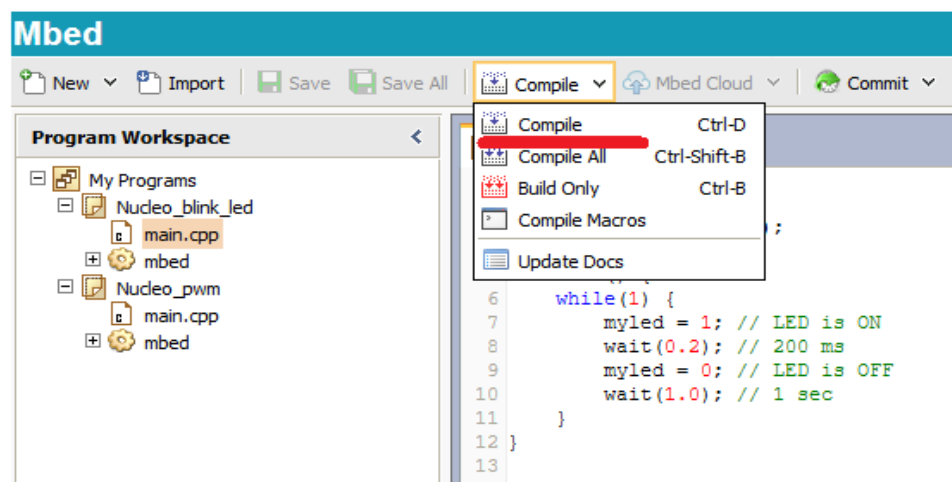


Рис. 2.22 Компиляция.

Если в процессе сборки проекта возникнут ошибки, то это будет видно в окне сообщений программы (Рис. 2.23). Также в этом окне выводится сообщение об успешном завершении компиляции программы.

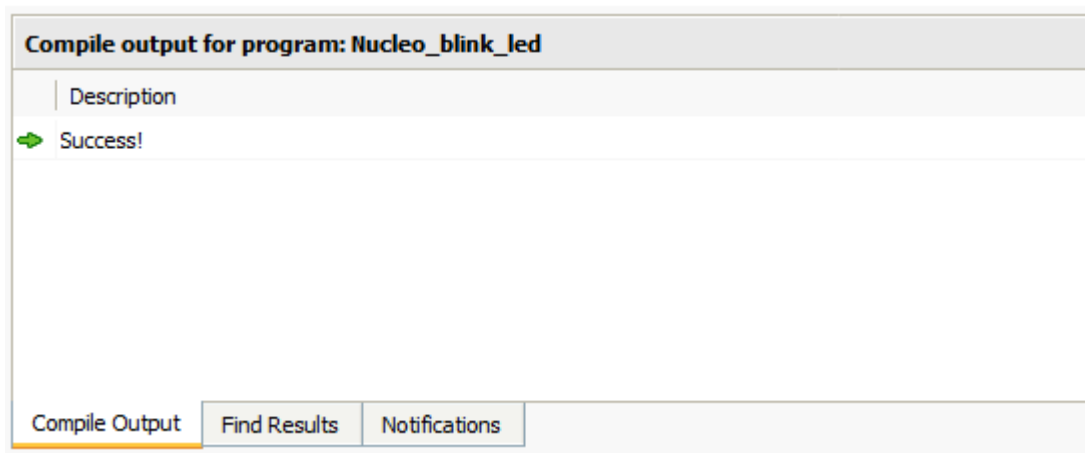


Рис. 2.23 Окно сообщений.

2.5. Подключение отладочной платы и загрузка проекта

Для соединения платы с компьютером необходимо подключить mini USB кабель (Рис. 2.24). Подключенная плата автоматически определится в среде Mbed (Рис. 2.25) и на компьютере (Рис. 2.26).

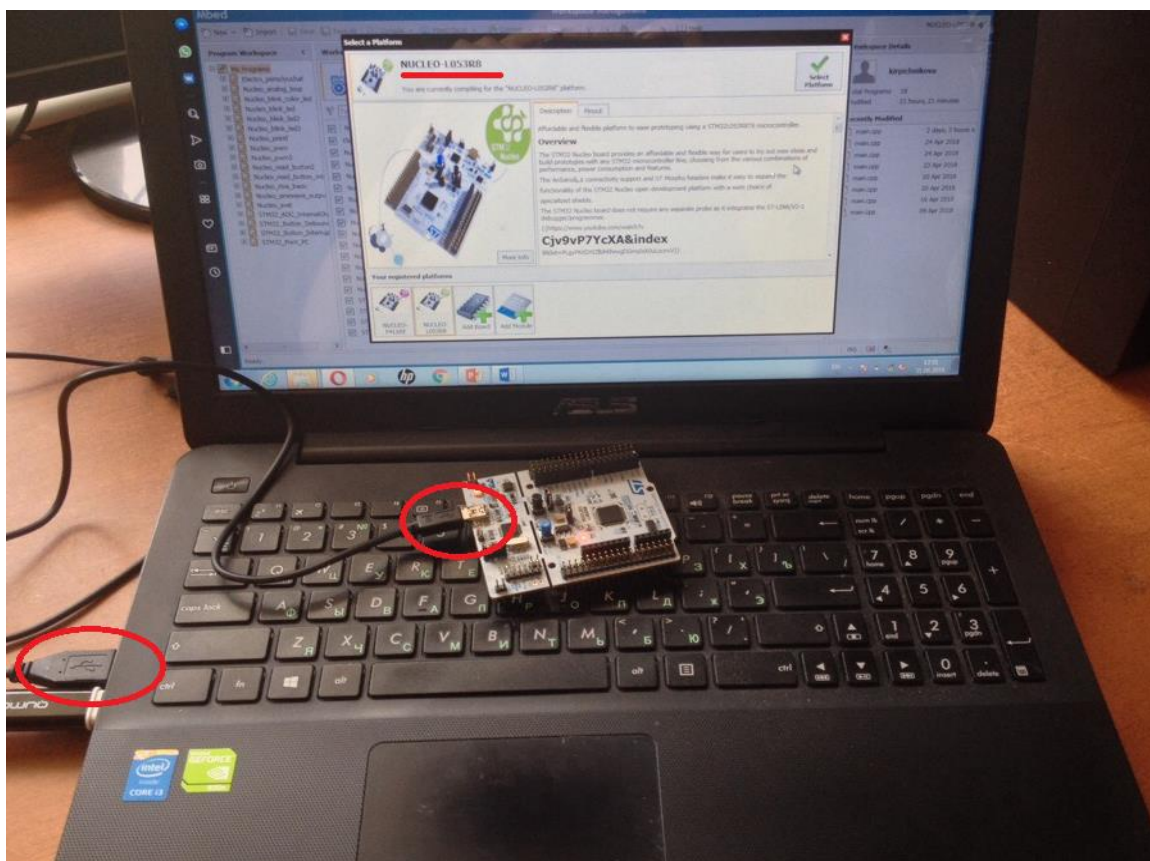


Рис. 2.24 Подключение платы к компьютеру.

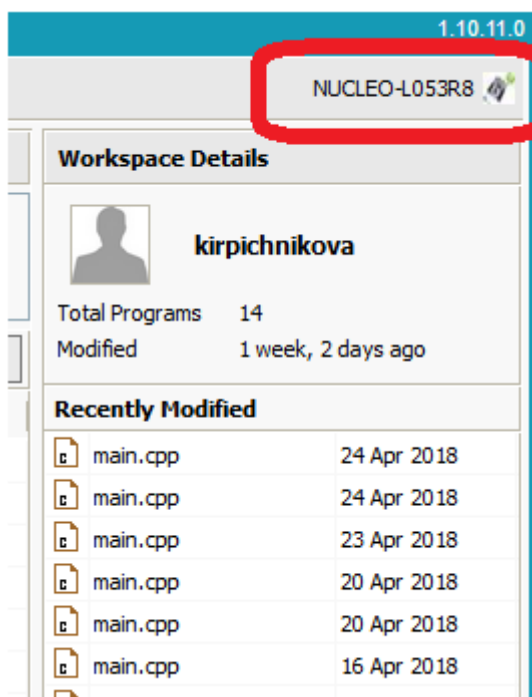


Рис. 2.25 Определение платы в Mbed.

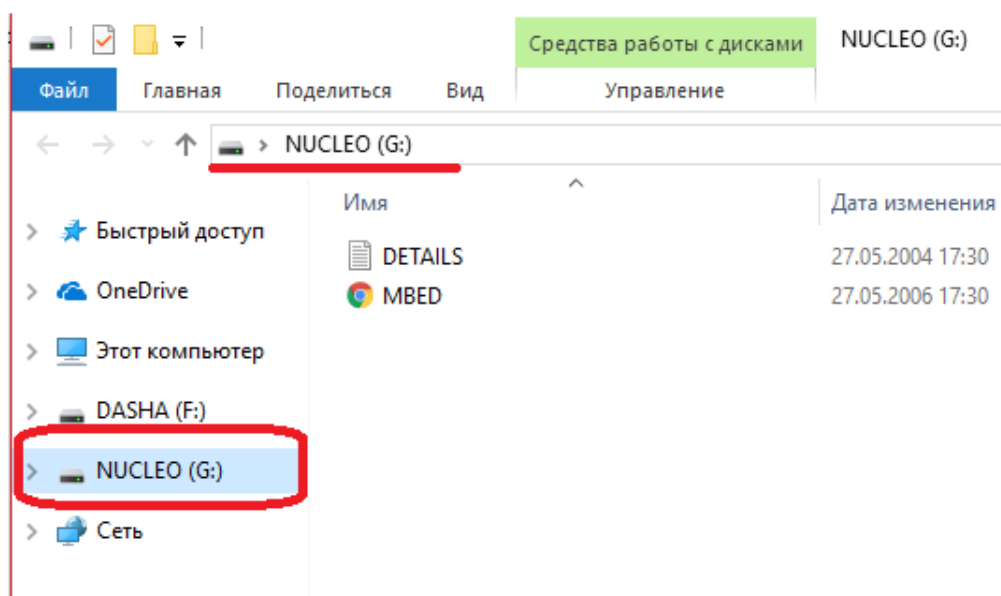


Рис. 2.26 Определение отладочной платы на устройстве.

2.6. Библиотеки Mbed

Все библиотеки хранятся в репозитории Mbed, который также находится на сервере платформы. Библиотеки можно импортировать из самого компилятора, а также прямо с её страницы на сайте, воспользовавшись поиском (Рис. 2.27).

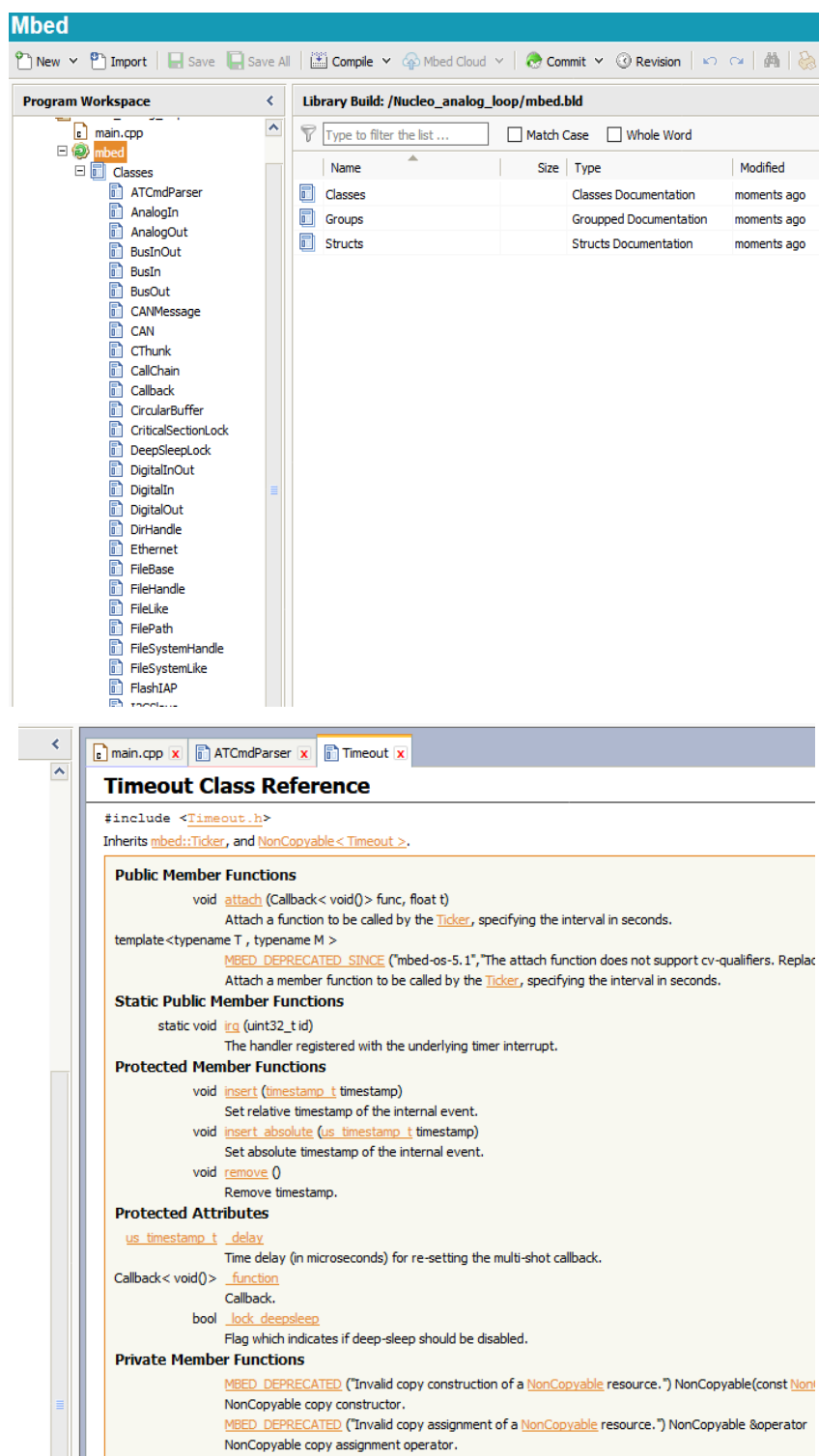


Рис. 2.27 Библиотеки Mbed.

Контрольные вопросы

1. Какие средства программирования МК вы знаете?
2. Что такое облачные технологии? Приведите примеры данных технологий.
3. Какие форматы облачных технологий существуют и в чем их различие? К какому формату относится облачная среда программирования Mbed?
4. Почему необходимо при регистрации указывать действующий адрес e-mail?
5. Как запустить компилятор Mbed?
6. Как посмотреть результаты компиляции и распределение памяти для компилируемой программы?
7. Как изменить используемую в Mbed отладочную плату?
8. Как осуществить возврат к предыдущей версии программы?
9. Для чего используется класс Serial? Какие параметры он использует?

3. Примеры выполнения заданий

Для выполнения лабораторной работы необходимо иметь следующие компоненты: персональный компьютер, кабель mini USB, отладочную плату Nucleo-L053R8 и доступ в Интернет.

Отладочная плата

Отладочная плата (рис. 3.1) используется для тестирования написанной программы и реализации конечного устройства. В данном случае это плата Nucleo-L053R8.

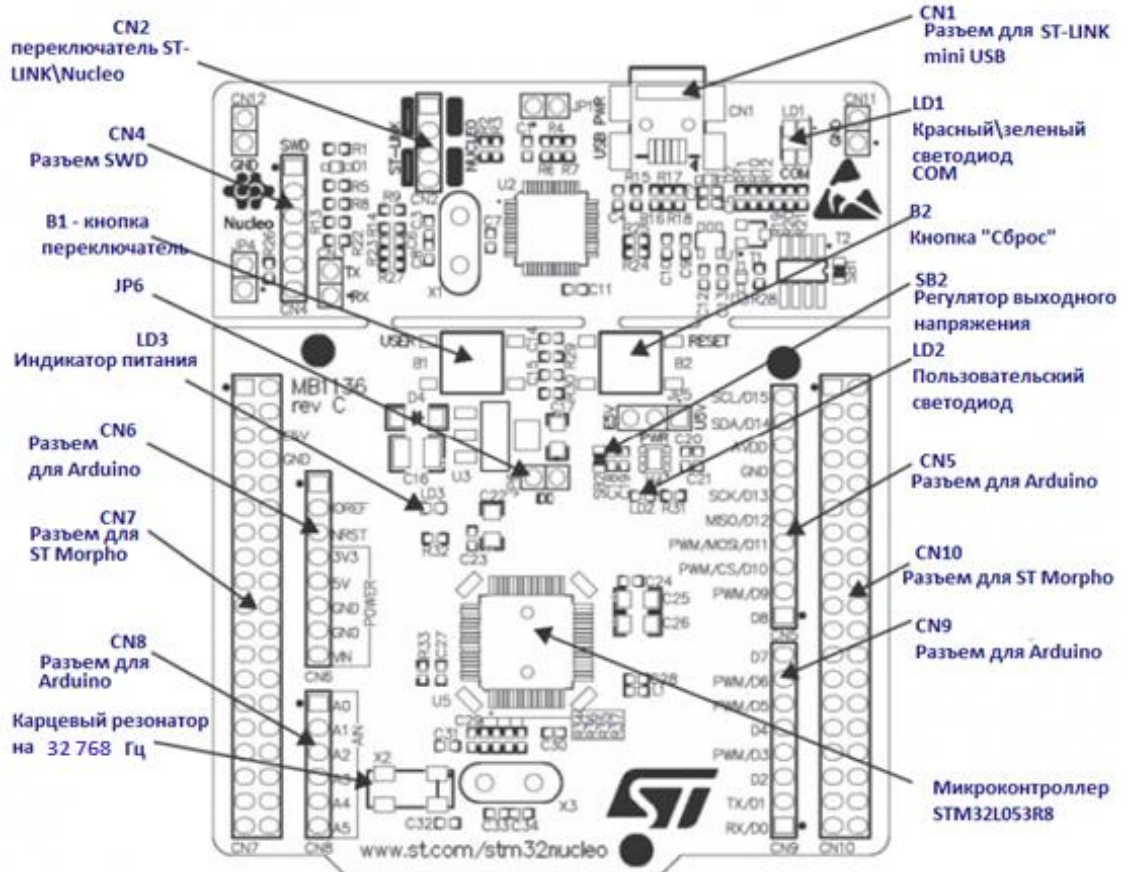


Рис.3.1 Отладочная плата Nucleo-L053R8.

Отладочная плата включает в себя:

- микроконтроллер STM32L053R8T6, который имеет 51 порт ввода-вывода. Ниже приведены блок-схема внутреннего устройства микроконтроллера STM32L053R8T6 (рис. 3.2) и конфигурация выводов (рис.3.3 и рис.3.4).

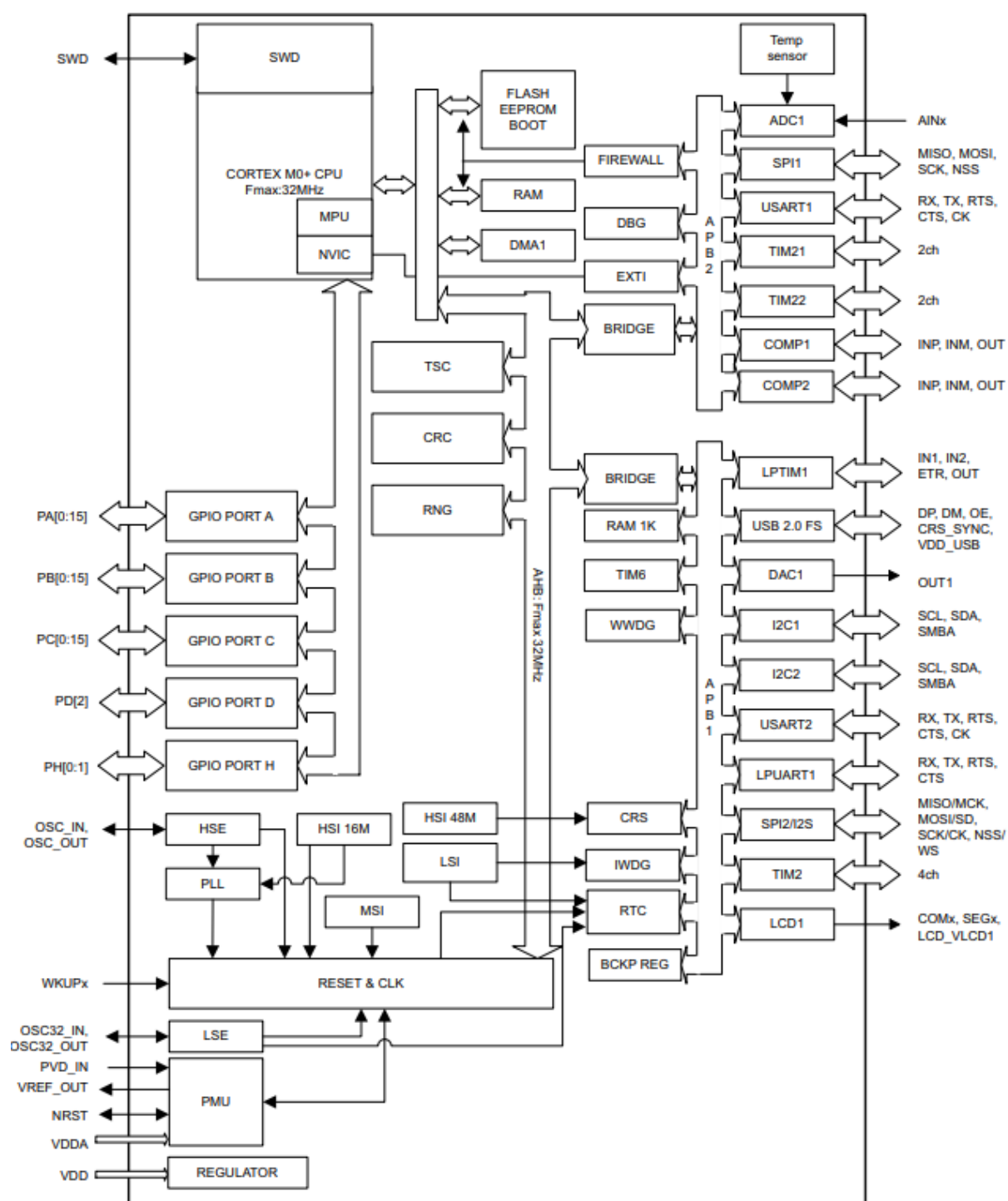


Рис. 3.2 Блок-схема внутреннего устройства МК STM32L053R8T6.

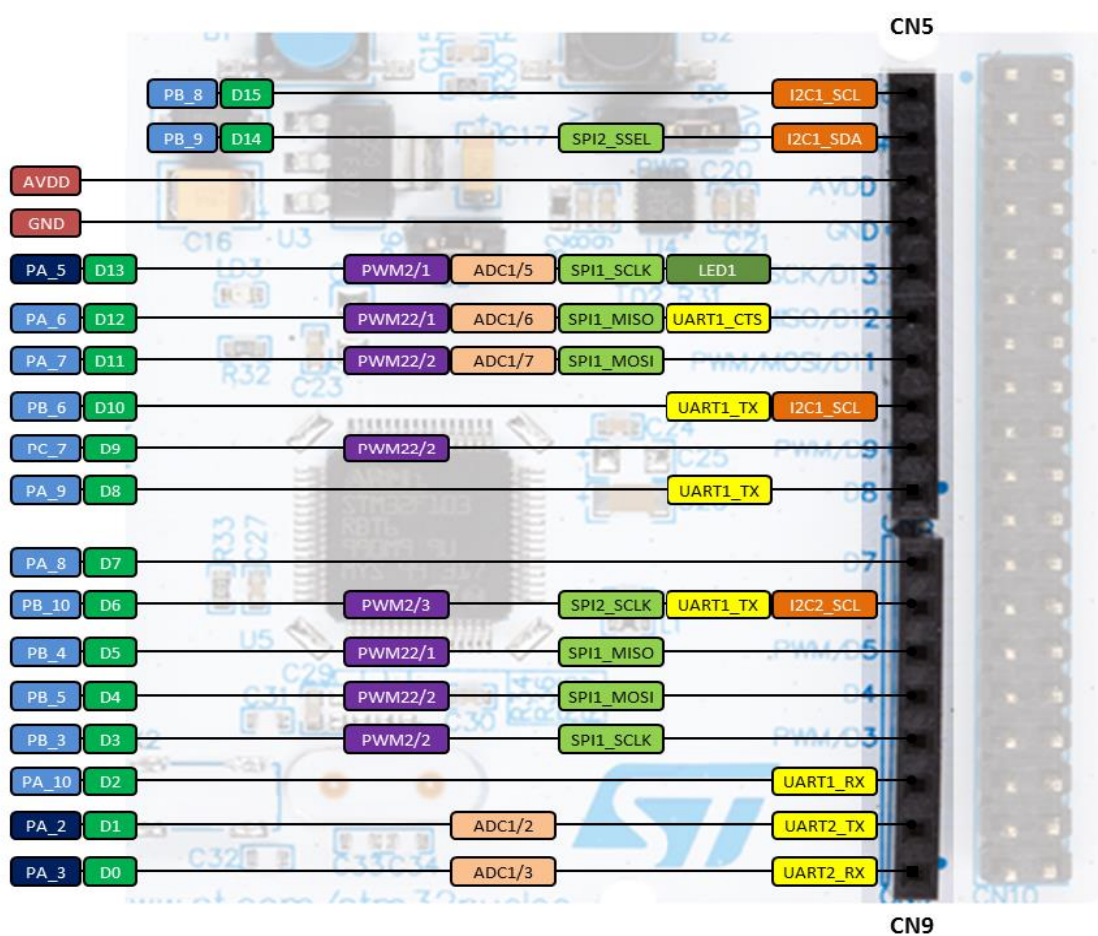
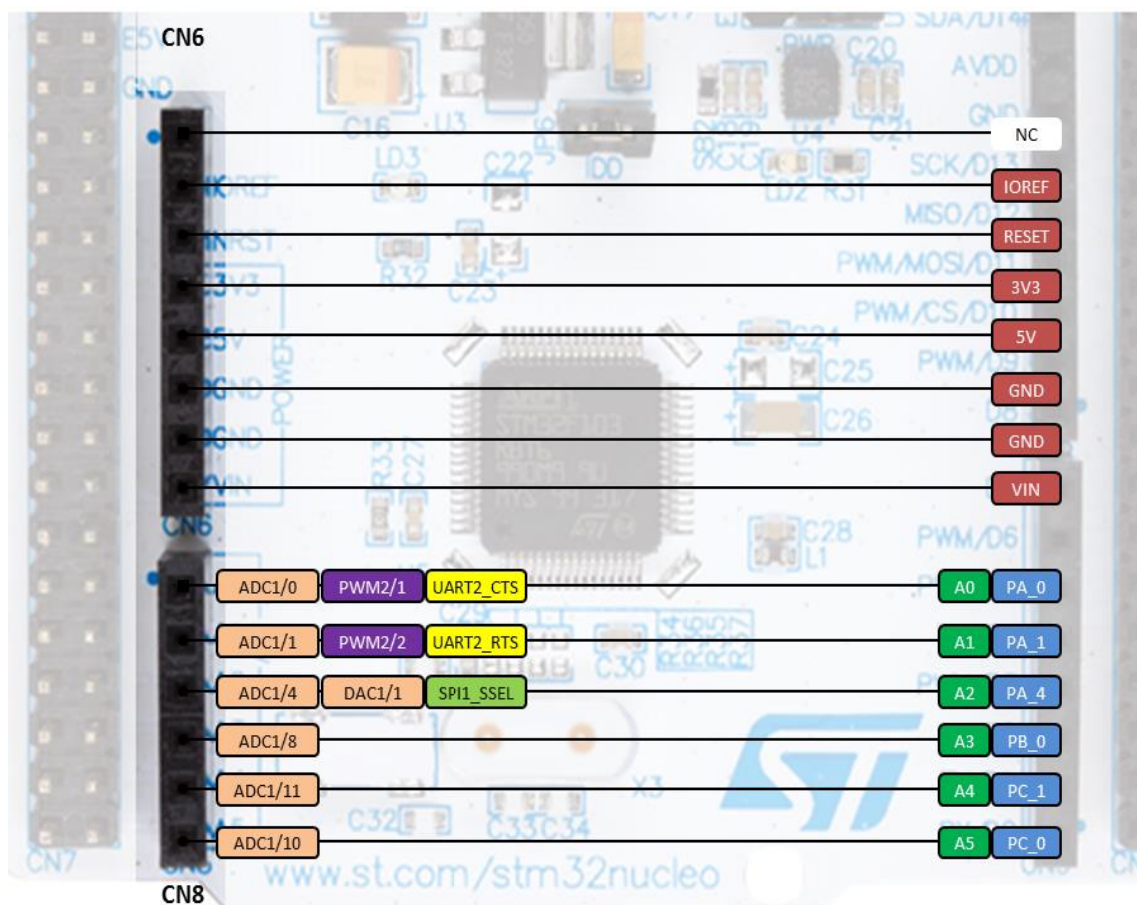


Рис. 3.3 Расположение портов ввода-вывода как у Arduino подобных плат.



– тактовая кнопка, находящаяся на линии порта РС13. Отдельно расположена кнопка «Сброс».

– 1 красный светодиод LD3 – индикатор питания, 1 зеленый светодиод LD2, который является пользовательским и 1 зеленый/красный светодиод LD1, расположенный рядом с USB портом, который показывает то, что в данный момент происходит с платой (программируется или готова к работе).

– кварцевый резонатор, частота которого составляет 32 768 Гц.

Начнем с выполнения задания от производителя.

Задача №1: программа для подключения терминала и вывода в него сообщения «Hello world».

Чтобы выполнить данную задачу необходимо ознакомиться с правилами подключения приложения Terminal (см. Приложение). В компиляторе Mbed необходимо открыть программу Display a message on PC using UART (Nucleo_printf). Теперь требуется реализовать блок-схему алгоритма (БСА) работы программы.

Блок-схема работы программы «Display a message on PC using UART» приведена на рис.3.5.



Рис. 3.5 Блок-схема программы Display a message on PC using UART.

На языке схем алгоритмов разработчик обычно описывает метод, выбранный для решения поставленной задачи. Довольно часто бывает, что одна и та же задача может быть решена различными методами. Способ решения задачи, выбранный на этапе ее инженерной интерпретации, на основе которого формируется БСА, определяет не только качество разрабатываемой прикладной программы, но и качественные показатели конечного изделия.

Далее необходимо скомпилировать программу (рис.3.6) и подключив отладочную плату, сохранить файл с расширением *.bin (см. главу 2).

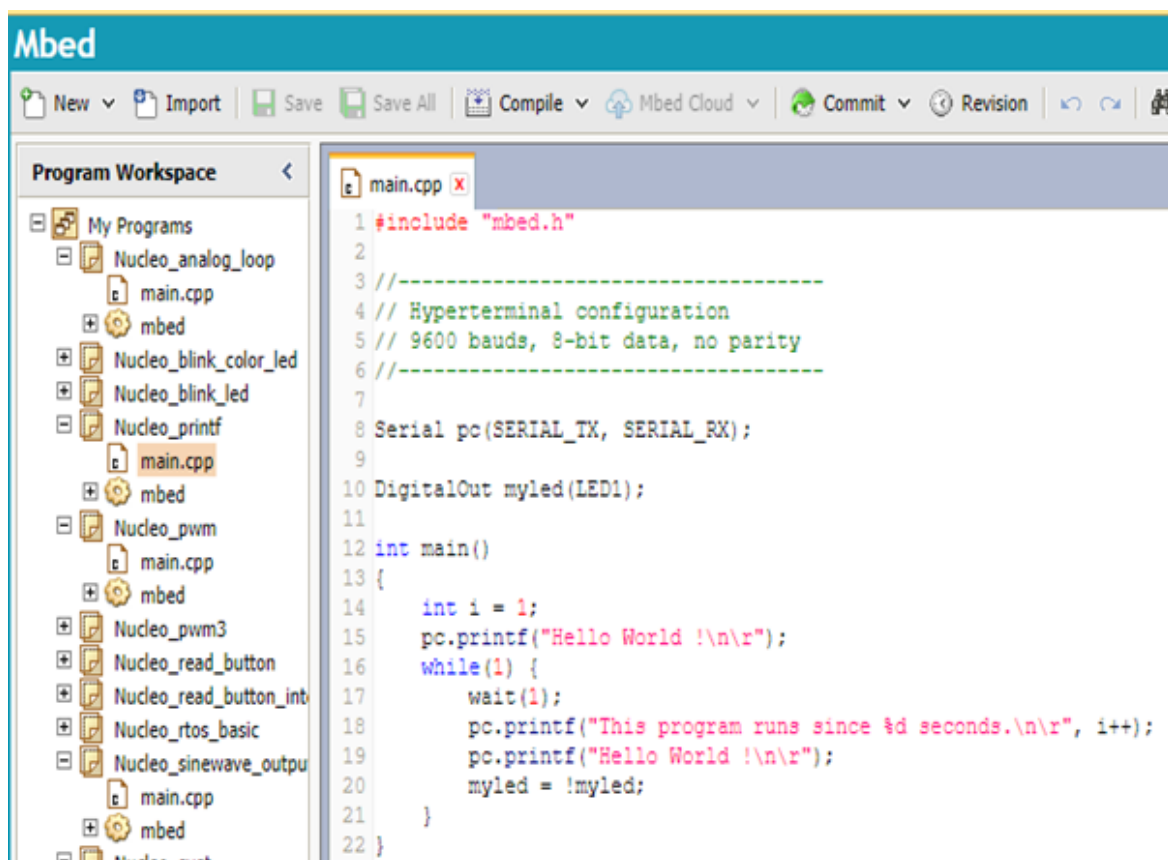


Рис. 3.6 Программа Display a message on PC using UART (Nucleo_printf).

После прошивки платы для обеспечения перезапуска процессора необходимо вытащить USB кабель из разъема и тут же снова подключить. Далее нужно запустить приложение Terminal (см. Приложение) и убедиться в корректности работы программы. На рисунке 3.7 показан результат работы данной программы.

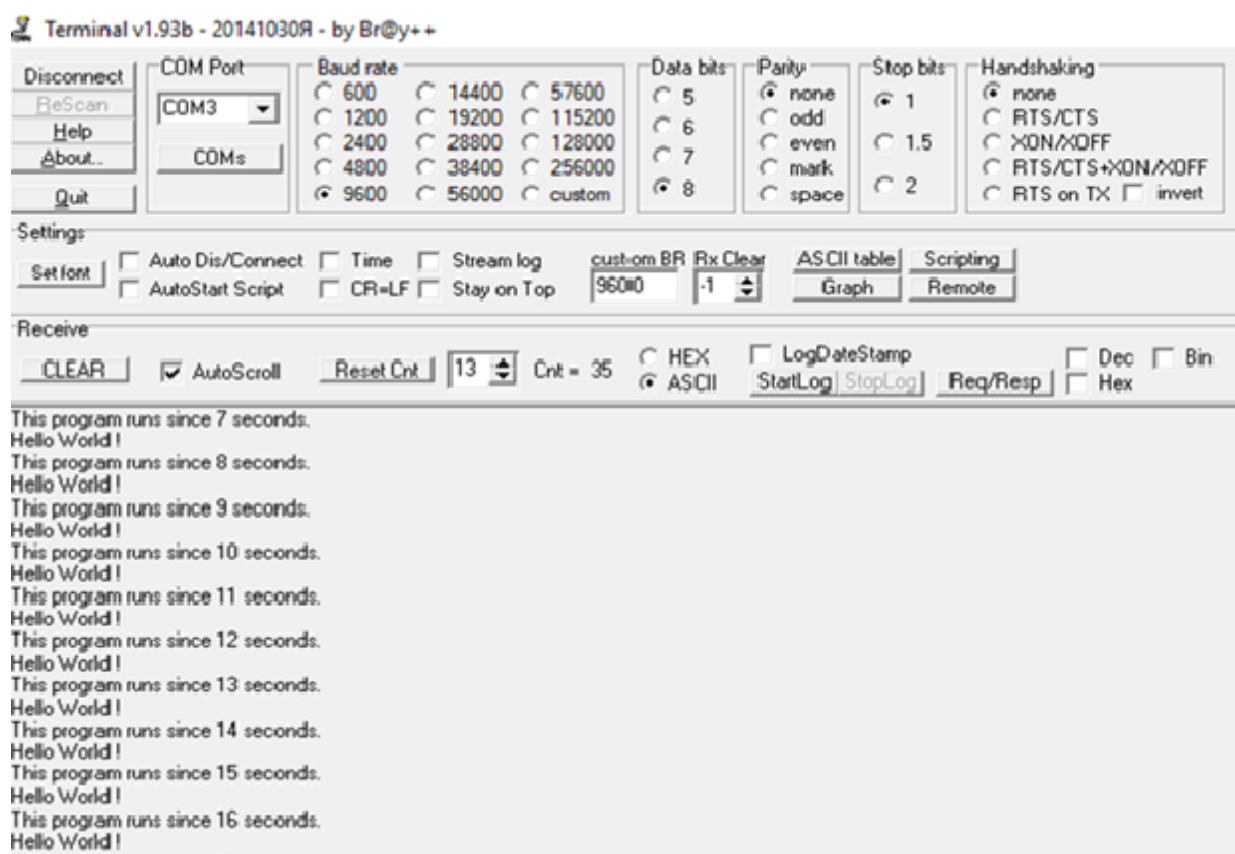


Рис. 3.7 Результат работы программы Display a message on PC using UART.

Приложение Terminal можно использовать и в других случаях, когда необходимо будет производить какой-либо контроль за действиями программ.

Задача №2: реализовать устройство «Электронный переключатель». В состав устройства входят кнопка пользователя В1 и один индикатор – пользовательский светодиод LD2, находящиеся на линиях портов микроконтроллера. В любой момент времени светится светодиод – индикатор питания LD3. По нажатию кнопки происходит увеличение скорости переключения индикатора.

При работе МК с датчиками, имеющими механические или электромеханические контакты (кнопки, клавиши, реле и клавиатуры), возникает явление, называемое дребезгом. Это явление заключается в том, что при замыкании контактов возможно появление отскока (BOUNCE) контактов, которое приводит к переходному процессу. При этом сигнал с контакта может быть прочитан МК как случайная последовательность нулей и единиц. Подавить это нежелательное явление можно программным путем.

Наибольшее распространение получили два программных способа ожидания установившегося значения:

1. подсчет заданного числа совпадающих значений сигнала;
2. временная задержка.

Будем использовать в программе первый способ.

Также требуется реализовать блок-схему алгоритма (БСА) работы программы.

Блок-схема работы программы «Электронный переключатель» приведена на рис.3.8.

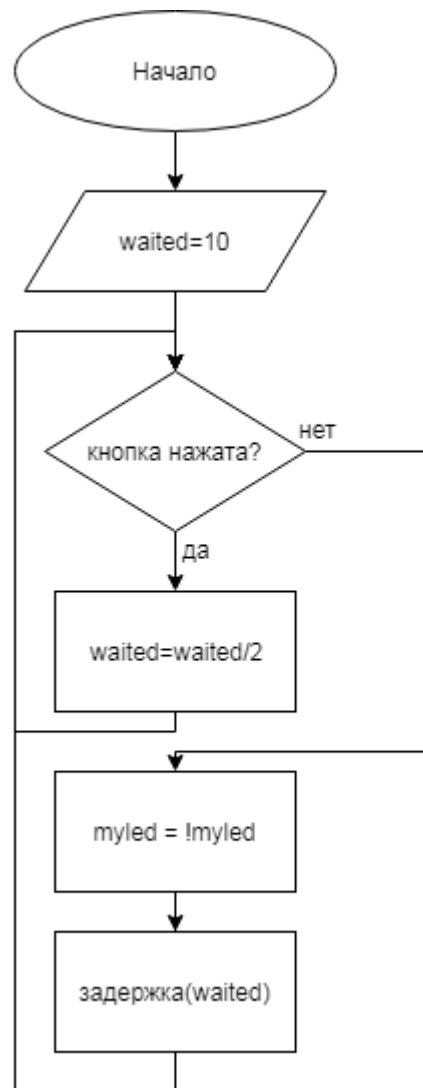


Рис. 3.8 Блок-схема программы «Электронный переключатель».

После того, как разработана блок-схема программы, можно приступить к написанию и отладке кода программы. Получаем следующий код:

```

#include "mbed.h"

InterruptIn mybutton(USER_BUTTON);

float waited=2.0;

DigitalOut myled(LED2);

void pressed()
{
    waited=waited/2.0;// 200 ms
}

int main() {

```

```

mybutton.fall(&pressed);
while(1) {
    myled = !myled; // Toggle the LED state
    wait(waited);
}
}

```

На этапе компиляции и отладки программы автоматически будет создан файл с расширением *.bin. Этот файл будет использоваться для программирования микроконтроллера.

На рис.3.9 изображен процесс отладки программы «Электронный переключатель».

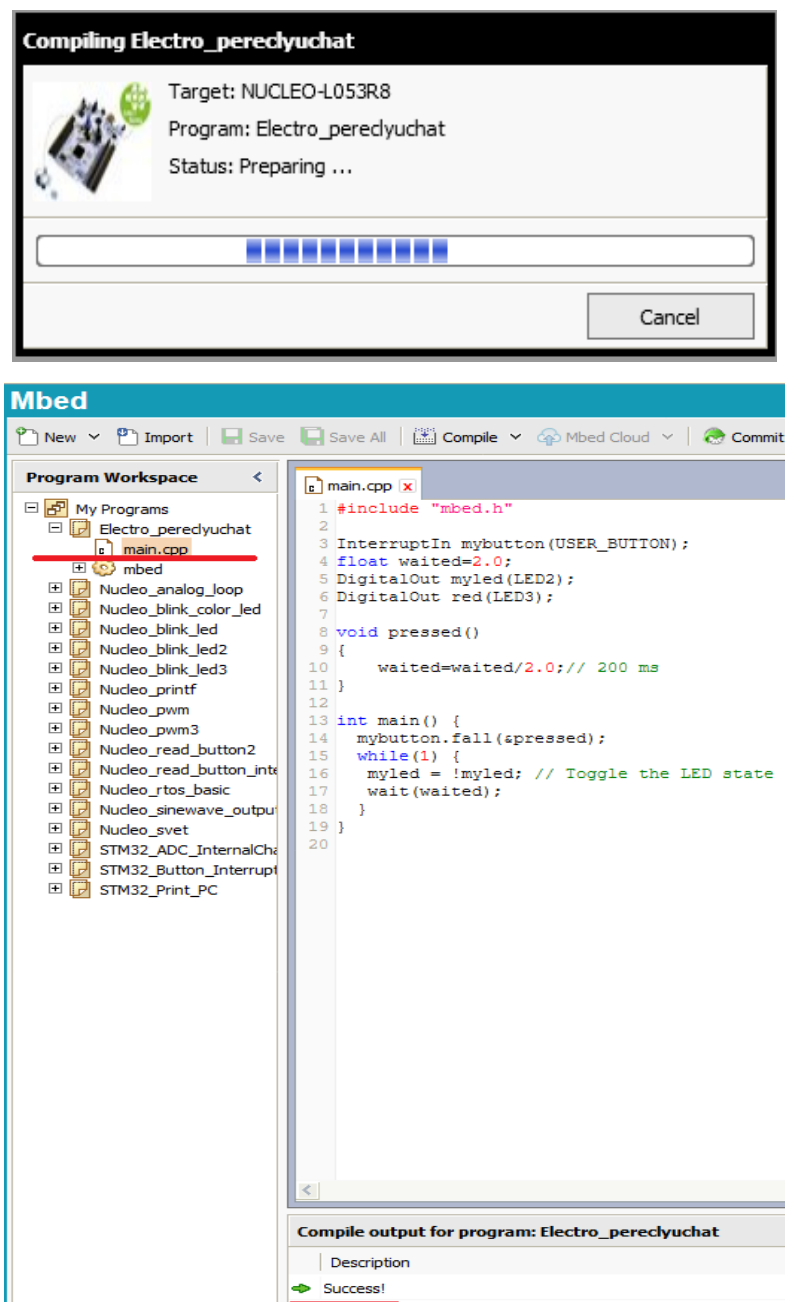


Рис. 3.9 Отладка программы «Электронный переключатель».

Файл с расширением *.bin необходимо будет сохранить в память микроконтроллера уже известным способом, а также убедиться в корректности работы программы, используя кнопки (элементы управления) и светодиоды (элементы индикации).

Контрольные вопросы

1. Опишите номенклатуру устройств, входящие в состав рабочего места студента.
2. Расскажите об устройстве отладочной платы.
3. В чем особенности организации микроконтроллера STM32L053R8? Назовите основные отличия от STM32F078CB и STM32F103.
4. Какова частота кварцевого резонатора, входящего в состав отладочной платы? Как изменить частоту тактирования микроконтроллера?
5. В чем отличие расположения контактов головных разъемов в режиме Arduino и Morpho?
6. Расскажите правила работы с терминалом. Как изменить количество стоп-бит в передаваемом символе?
7. Расскажите о назначении линий головного разъема D0 и D1. Почему их применение ограничено? Какие линии также имеют ограничения?

Литература

1. Магда Ю. Программирование и отладка C/C++ приложений для микроконтроллеров ARM. –: ДМК Пресс, 2013. – 168 с.
2. STM32 MCU Nucleo [Электронный ресурс]. – Режим доступа: https://www.st.com/content/st_com/en/products/evaluation-tools/product-evaluation-tools/mcu-eval-tools/stm32-mcu-eval-tools/stm32-mcu-nucleo.html?querycriteria=productId=LN1847, свободный. – Загл. с экрана.
3. Mbed OS 5 Help [Электронный ресурс]. – Режим доступа: <https://os.mbed.com/compiler/#nav:/>; свободный. – Загл. с экрана.
4. Козлов-Кононов Д. Процессорные ядра семейства Cortex. Сочетание высокой производительности и низкого энергопотребления / Д.Козлов-Кононов. // ЭЛЕКТРОНИКА: Наука, Технология, Бизнес. – 2010. – 8. – С. 16-25.

Задания

Раздел 1. В данном разделе необходимо загрузить в среду разработки указанный файл проекта, написать алгоритм работы, нарисовать блок-схему алгоритма, запустить проект на выполнение, показать результат преподавателю, если необходимо, подключить соответствующие измерительные устройства, и объяснить принцип работы проекта.

1. Пример программы управления временем свечения светодиода.
 - Файл проекта: Blinky LED test for the ST Nucleo boards.
 - Название программы: Nucleo_blink_led.
2. Пример программы обработки нажатия кнопки пользователя на плате Nucleo
 - Файл проекта: Read the user button state on the Nucleo board.
 - Название программы: Nucleo_read_button.
3. Пример программы обработки нажатия кнопки пользователя с помощью внешнего прерывания.
 - Файл проекта: Read the user button using external interrupt.
 - Название программы: Nucleo_read_button_interrupt.
4. Пример программы с использованием АЦП и ЦАП.
 - Файл проекта: Analog loop example using ADC and DAC.
 - Название программы: Nucleo_analog_loop.
5. Пример программы считывания внутренних каналов АЦП.
 - Файл проекта: ADC Internal Temperature Sensor reading example.
 - Название программы: STM32_ADC_internalchannels.

Раздел 2. В данном разделе необходимо загрузить в среду разработки указанный файл проекта, написать алгоритм работы, нарисовать блок-схему алгоритма, запустить проект на выполнение, показать результат преподавателю, если необходимо, подключить соответствующие измерительные устройства, и объяснить принцип работы проекта. По заданию преподавателя модифицировать проект, результаты отразить в отчете.

1. Пример программы синтеза ШИМ-сигнала. Требуется подключение осциллографа.
 - Файл проекта: Output a pwm signal.
 - Название программы: Nucleo_pwm.
2. Пример программы синтеза синусоидального сигнала с использованием ЦАП. Требуется подключение осциллографа.

- Файл проекта: Output a sinewave using DAC.
 - Название программы: Nucleo_sinewave_output.
3. Пример программы считывания значения напряжения на аналоговом входе с помощью АЦП. Требуется подключение дополнительных устройств.
- Файл проекта: ADC read value example.
 - Название программы: STM32_read_analog.
4. Пример программы пересылки данных с помощью SPI. Требуется подключение дополнительных устройств.
- Файл проекта: Send data with SPI
 - Название программы: Nucleo_spi_master
5. Пример программы изменения рабочего цикла выходного ШИМ-сигнала с помощью нажатия кнопки. Требуется подключение дополнительных устройств.
- Файл проекта: Output pwm signal with duty-cycle changed when pressing a bu.
 - Название программы: Nucleo_pwm2.

Раздел 3. В данном разделе необходимо написать алгоритм работы, нарисовать блок-схему алгоритма, самостоятельно создать проект и написать программу, запустить проект на выполнение, показать результат преподавателю, если необходимо, подключить соответствующие измерительные и дополнительные устройства, объяснить принцип работы проекта.

6. . Разработать индикатор количества нажатий кнопки в двоичном коде (двоичный счетчик). (Номер кнопки, направления счета и дополнительные устройства получить у преподавателя).
7. Написать программу «Светофор». Светофор управляется кнопкой, которая подключена к одной из линий порта А. В исходном состоянии горит зеленый для автомобилей и красный для пешеходов. После нажатия кнопки зеленый мигает 10 секунд, загорается желтый + красный для автомобилей, горит 10 секунд. Затем 20 секунд для автомобилей горит красный, а для пешеходов зеленый. Для пешеходов зеленый мигает 10 секунд, для автомобилей горит желтый. Затем для автомобилей загорается зеленый, а для пешеходов красный. После выполнения данного цикла светофор 10 секунд не реагирует на кнопку. (Дополнительные устройства получить у преподавателя).
8. Написать программу «Бегущие огни». Программа «Бегущие огни» должна выводить световые эффекты (не менее 16) с помощью 8 красных светодиодов.

Эффекты переключаются двумя кнопками. При достижении максимального или минимального номера эффекта дважды по 0.5с. должен засветиться зеленый светодиод. Другая пара кнопок должна регулировать скорость (не менее 16 скоростей) смены кадров в эффектах. При достижении минимальной или максимальной скорости должен засветиться зеленый светодиод на 1с. Если в течении 20 секунд не нажата ни одна кнопка, программа автоматически должна переключать эффекты в сторону увеличения по циклу с выдержкой каждого эффекта не менее 30 секунд. (Дополнительные устройства получить у преподавателя).

Требования к защите работы

1. Демонстрация написанных и полностью отлаженных программ на компьютере и на отладочной плате.
2. Отчет по лабораторной работе, оформленный в соответствии с правилами.
3. Готовность отвечать на вопросы преподавателя по теме работы.

Правила оформления отчета к работе

1. Отчет, является документом, отражающим результаты и ход выполнения лабораторной работы. Отчет должен содержать следующие пункты:
2. Титульный лист, содержащий тему и номер лабораторной работы, фамилии выполнявшего студента и проверявшего преподавателя.
3. Цель работы – указываются цели, которые необходимо достигнуть в результате выполнения лабораторной работы.
4. Задачи – указываются задачи, решаемые в ходе выполнения лабораторной работы.
5. Выполнение работы – указывается последовательность действий или алгоритм работы программы, приводятся выполненные расчеты.
6. Результаты работы – приводятся алгоритм, блок-схема алгоритма работы программы и текст работы программы с комментариями на русском языке.
7. Вывод по каждому пункту, в соответствии с заданием. Общий вывод по работе в соответствии с заявленными целями – пункт содержит перечень решенных в ходе работы задач и выполненных целей, а также выводы по достигнутым результатам.

Обратить внимание

1. Отчет принимается только в печатном виде на бумажном носителе. Т.е. отчет должен быть полностью набран на компьютере и распечатан. Отчеты в электронном виде рассматриваться не будут.
2. Листы отчета должны быть пронумерованы, скреплены между собой и помещены в папку.

ПРИЛОЖЕНИЕ

Использование программы Terminal

Программа Terminal используется для контроля за передачей по последовательному порту между компьютером и отладочной платой, а также для отображения результатов работы функции printf().

1. Приложение Terminal необходимо запустить из папки, которую находится на рабочем столе вашего учебного ПК.
2. После запуска необходимо установить параметры терминала как показано на Рис.1. Если к ПК подключено несколько устройств, то в строке COM-port необходимо указать тот, который используется для связи с отладочной платой Nucleo-L053R8.

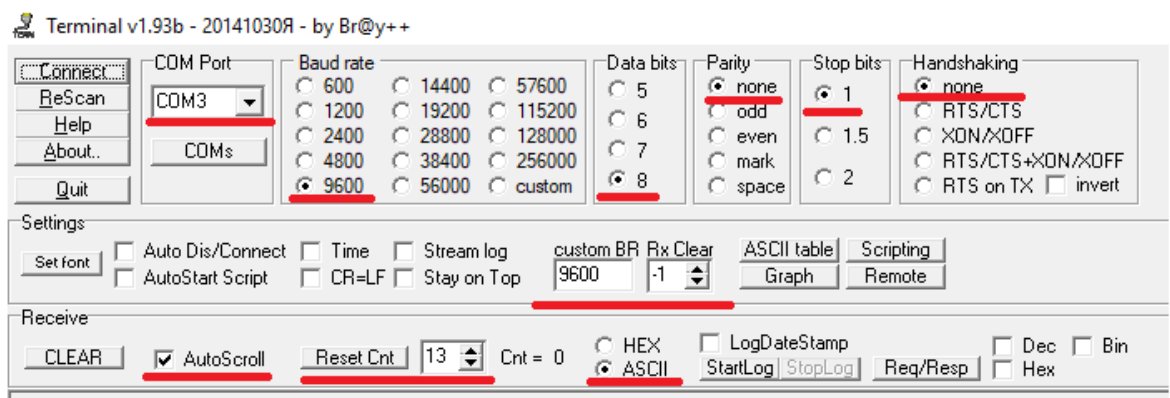


Рис.1 Настройка терминала.

3. После загрузки исполняемого кода в плату, при выполнении заданий, использующих класс Serial, необходимо в приложении Terminal создать соединение (Connect) между отладочной платой и ПК, по окончании выполнения задания необходимо разорвать (Disconnect) связь.