****

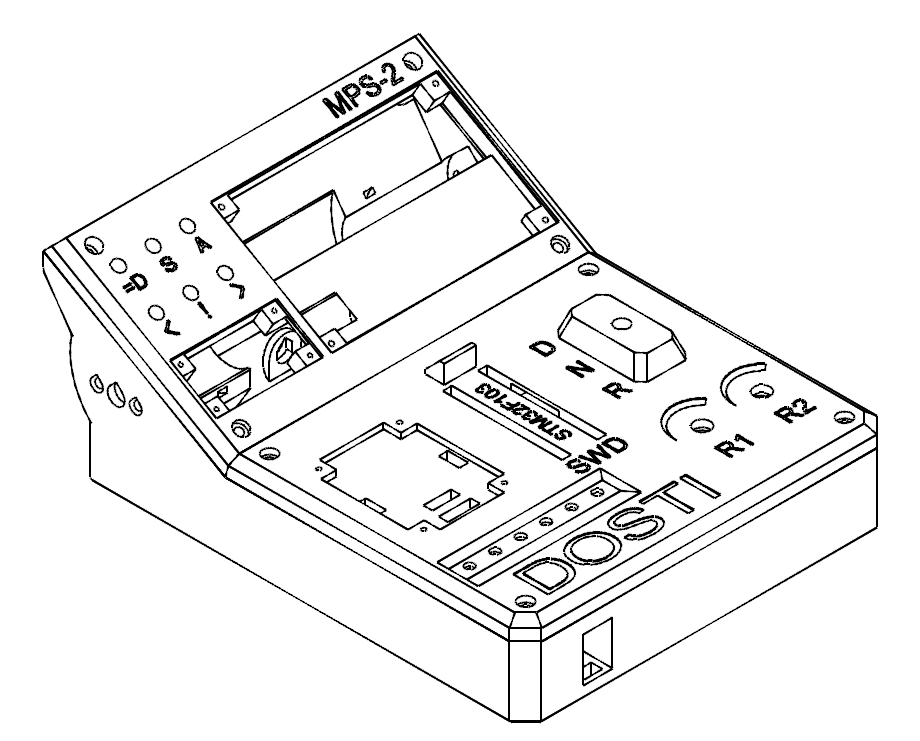
**MPS – 2 (Module Programmable STM32F103)**

**Стенд учебный программируемый**

**Д.Н.КАЛАЕВ, А.Р.ВАФАЕВ**

**РУКОВОДСТВО**

**ПОЛЬЗОВАТЕЛЯ**



**СОДЕРЖАНИЕ**

|  |  |
| --- | --- |
|  | стр. |
| **1 Описание устройства** | 3 |
| 1.1 Комплект поставки | 4 |
| 1.2 Технические характеристики | 5 |
| 1.3 Функциональные компоненты | 8 |
| **2 Программное обеспечение для работы с устройством** | 11 |
| 2.1 Установка и настройка ПО Keil uVision | 11 |
| 2.2 Установка и настройка ПО STM32 Cube MX | 16 |
| **3 Особенности работы в среде Keil uVision** | 23 |
| 3.1 Интерфейс Keil uVision | 23 |
| 3.2 Синтаксис программной среды | 24 |
| 3.3 Типы данных | 27 |
| 3.4 Операторы | 28 |
| **Приложение А** | 30 |
| **Приложение Б** | 31 |

Наименования устройств и названия компаний, упоминающиеся в данном руководстве, могут являться зарегистрированными торговыми марками или объектами авторского права соответствующих компаний и используются исключительно для идентификации или пояснений, а также для выгоды владельцев, без намерения нарушить их права.

1. **Описание устройства**

Микроконтроллеры STM32 являются популярной и востребованной платформой, позволяющей создавать профессиональные решения для автоматизации в различных областях, таких как медицинские приборы, средства измерений, системы безопасности и т.д.

С учетом этого необходимо обеспечить качественную подготовку специалистов в области программирования промышленных микроконтроллеров. Для обучения учащихся учебных заведений систем профессионального технического и высшего технического образования необходимо использовать интуитивно понятные и безопасные в использовании платформы.

Основываясь на этом в Инновационном центре «DOSTI» был разработан стенд учебный программируемый MPS-2 (Module Programmable STM32F103).

Устройство разработано по аналогии с органами управления автомобилем (для лёгкости усвоения материала на практике). Процесс программирования строится от простого к сложному. Есть возможность программирования устройства на работу только одной или несколько функций, либо создание программы работы всей системы одновременно. Сложность программирования различных функций также различна, от программирования работы светодиодов, до вывода на дисплей всех параметров и регулировки скорости вращения двигателей.

Устройство предназначено для:

* формирования понимания принципов работы микроконтроллеров;
* обучения программированию многофункционального микроконтроллера STM32;
* изучения принципов работы различных электронных и электромеханических устройств и их взаимодействия;
* использования для подготовки студентов к соревнованиям по профессиональному мастерству;
* использования в качестве необходимой базы для проведения профессиональных соревнований среди студентов;
* использования в качестве необходимой базы в учебном процессе учреждений профессионального технического и высшего технического образования.
  1. **Комплект поставки**

В комплект поставки входит:

1. Стенд учебный программируемый MPS-2 – 1 шт.
2. Блок питания 9В 3А – 1 шт.
3. Отладочная плата STM32F103C8T6 (запасной элемент) – 1 шт.
4. Драйвер двигателя L298N (запасной элемент) – 1 шт.
5. USB удлинитель Type A-A – 1 шт.

В состав MPS-2 входит:

1. Отладочная плата STM32F103C8T6 – 1 шт.
2. Драйвер двигателя L298N – 1 шт.
3. Символьный LCD дисплей 2004 (20 столбцов, 4 строки) – 1 шт.
4. Модуль 4-разрядного цифрового семисегментного индикатора (на базе TM1637) – 1 шт.
5. Двигатель постоянного тока с редуктором и встроенным энкодером – 2 шт.
6. Светодиод – 6 шт.
7. Тактовая кнопка – 6 шт.
8. Трёх позиционный переключатель – 1 шт.
9. Потенциометр – 2 шт.
10. Тумблер включения питания – 1 шт.
11. Колесо с переходником – 2 шт.
12. Печатная плата с элементами – 1 шт.
13. Бокс для аккумуляторов стандарта 18650 – 1 шт.
14. Аккумуляторы Li-ion 18650 – 2 шт.
15. Корпус PLA пластика – 1 шт.
16. Программатор ST-LINK/V2 – 1 шт.
    1. **Технические характеристики**

Стенд учебный программируемый MPS-2 представляет собой компактное устройство, выполненное из ударопрочного пластика включающее в себя большое количество различных модулей.

Устройство моделирует работу основных систем автомобиля. Стенд позволяет создавать различные конфигурации программ приближенных к используемым в транспортных средствах, а также в роботизированных системах.

Устройство обладает рядом достоинств:

* мобильность;
* возможность использования устройства в автономном режиме и от сети;
* защита от некорректных действий;
* изолированный ударопрочный корпус;
* простота замены отдельных модулей;
* возможность визуального наблюдения за работой модулей.

Таблица 1.1 - Технические характеристики MPS-2

|  |  |
| --- | --- |
| **Параметр** | **Значение** |
| Размеры (ШхВхГ) (без учета колес) | 160х83х210 мм |
| Размеры (ШхВхГ) (с учетом колес) | 220х83х210 мм |
| Масса | 800 гр |
| Рабочая температура | 10-35 С° |
| Рабочее напряжение | 7-10 В |
| Напряжение питания адаптера | 220 В |
| Максимальная скорость вращения колёс | 300 об/мин |
| Максимальный ток адаптера | 3 А |
| Напряжение на выходе адаптера | 9 В |
| Тип аккумуляторов | Li-Ion стандарта 18650 |
| Количество аккумуляторов | 2 шт. |
| Количество символов дисплея LCD2004 | 20х4 |
| Количество символов дисплея TM1637 | 4 |

Основой устройства является STM32F103C8T6 (рис.1.1) — это микроконтроллер, построенный на ядре ARM Cortex-M3. Данное ядро имеет много преимуществ, его основное преимущество на сегодняшний день — универсальность. За годы существования Cortex-M3 стал индустриальным стандартом. Об этом говорит количество производителей, присоединившихся к данной архитектуре. Все основные производители микроконтроллеров, которые присутствуют в СНГ, кроме Microchip, имеют или развивают решения на основе этой архитектуры. В 32-битном сегменте разработчики редко работают с ассемблером, в основном используются языки высокого уровня, например, язык C.

Таблица 1.2 – Технические характеристики микроконтроллера STM32F103C8T6

|  |  |
| --- | --- |
| **Параметр** | **Значение** |
| Число разрядов | 32 бита |
| Максимальная частота | 72 мГц |
| Объем памяти программ (FLASH) | 64 кБайт |
| Объем памяти данных (RAM) | 20 кБайт |
| Количество входов/выходов | 37 |
| Таймеры общего назначения | 3 |
| Расширенный таймер с ШИМ управления двигателем | 1 |
| Системный таймер | 1 |
| Сторожевые таймеры | 2 |
| UART | 3 |
| SPI | 2 |
| I2C | 2 |
| CAN | 1 |
| USB | 1 |
| Контроллеры прямого доступа к памяти | 7 |
| Наличие АЦП/ЦАП | АЦП 10x12b |
| Часы реального времени | есть |
| Аппаратный модуль расчета CRC | есть |
| Напряжение питания микроконтроллера | 2 … 3,6 В |
| Напряжение питания платы | 5 В |
| Ток потребления | до 50 мА |
| Размеры платы | 53 x 22,5 мм |

Семейство STM32 отличается от конкурентов хорошим поведением в температурном диапазоне от -40 до 85°С. Производительность ядра и периферии сохраняется полностью. В семействе STM32 есть ряд изделий, сертифицированных на расширенный температурный диапазон от -40 до 105°С.

Одна из причин мировой популярности семейства STM32 — максимальный комфорт разработчика. Если универсальность ядра STM32 позволяет менять производителя c минимальными затратами на программный код, то Pin-to-Pin совместимость внутри семейства STM32 позволяет менять объем памяти (флэш-память и ОЗУ) и периферию (Ethernet, USB, CAN, и т.д.), не трогая печатную плату. «Pin-to-Pin совместимость» означает, что для одного размера корпуса все сигналы сохраняются на тех же самых вводах/выводах для разных вариантов микроконтроллеров семейства.

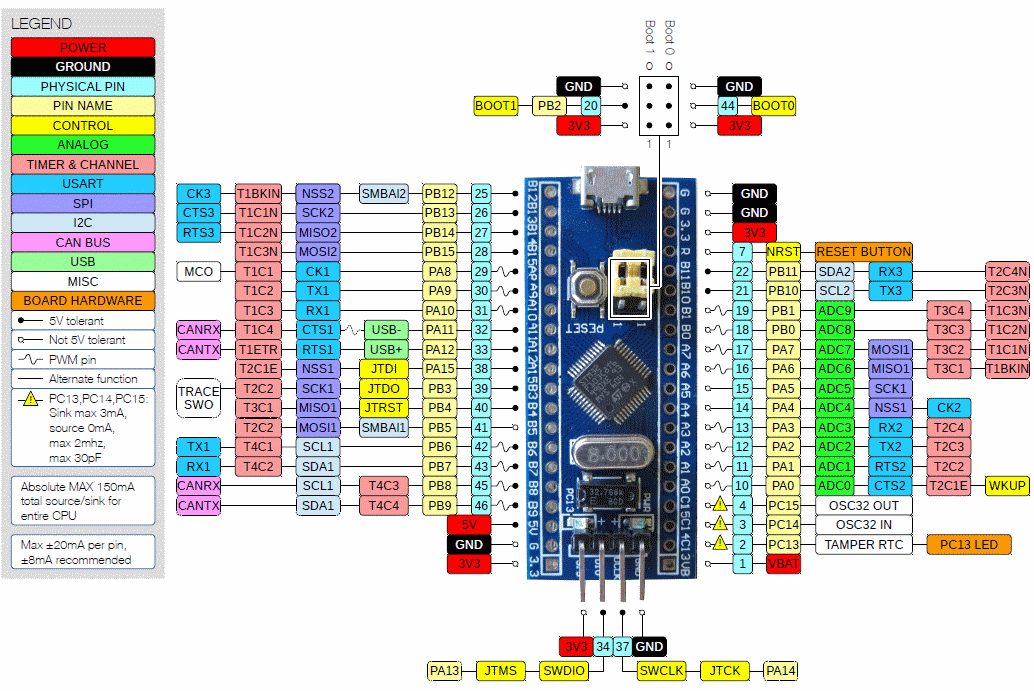


Рисунок 1.1 – Микроконтроллер STM32F103C8T6 и его распиновка

* 1. **Функциональные компоненты**

Основными компонентами устройства MPS-2, с которыми взаимодействует пользователь, являются:

1. Символьный LCD дисплей 2004 (20 столбцов, 4 строки);
2. Светодиоды;
3. Модуль 4-разрядного цифрового семисегментного индикатора (на базе TM1637);
4. Колеса, установленные на валу DC моторов с редукторами и встроенными энкодерами;
5. Трёхпозиционный переключатель;
6. Потенциометры R1, R2;
7. Драйвер двигателя L298N;
8. Отладочная плата STM32F103C8T6;
9. Кнопки;
10. Тумблер включения питания.



Рисунок 1.2 – Функциональные компоненты стенда учебного программируемого MPS-2

Символьный жидкокристаллический дисплей является аналогией приборной панели водителя. На приборной панели отображаются шкала мощности, пробег, скорость оборотов каждого колеса и положение рычага АКПП. Панель из светодиодов представляет собой внешнюю индикацию автомобиля (табл. 1.3) и напрямую зависит от органов управления в виде кнопок. Модуль цифрового индикатора является спидометром и выводит скорость автомобиля (м/мин). Регуляторы R1 и R2 в виде потенциометров являются аналогией педалей в автомобиле и отвечают за скорость и торможение колес установленных на валах двигателей постоянного тока (табл. 1.4). Трехпозиционный переключатель представляет собой коробку перемены передач (КПП) в автомобиле и отвечает за направление движения колес (табл. 1.5).

Таблица 1.3 – Функции и индикация кнопок

|  |  |  |  |
| --- | --- | --- | --- |
| **Функция** | **Орган управления** | **Индикация светодиодами** | **Индикация на дисплее** |
| Ручной тормоз | Кнопка 1 | 2-ой красный горит | STOP |
| Аварийный сигнал | Кнопка 2 | 4-ый желтый мигает  5-ый красный мигает  6-ой жёлтый мигает | <!> |
| Сигнал левого поворота | Кнопка 3 | 4-ый желтый мигает | < |
| Сигнал правого поворота | Кнопка 4 | 6-ой жёлтый мигает | > |
| Включение фар | Кнопка 5 | 1-ый белый горит | =D |
| ABS | Кнопка 6 | 3-ий зелёный горит | ABS |

Таблица 1.4 – Функции и индикация регуляторов R1 и R2   
(ГАЗ и ТОРМОЗ)

|  |  |  |  |
| --- | --- | --- | --- |
| **Функция** | **Описание** | **Орган управления** | **Индикация на дисплее** |
| ГАЗ | Управление разгоном колёс | Регулятор R1 | “Power”  █ █ █ █ █ █ █ |
| ТОРМОЗ | Управление торможением колёс | Регулятор R2 |  |

Таблица 1.5 – Функции и индикация АКПП

|  |  |  |  |
| --- | --- | --- | --- |
| **Функция** | **Описание** | **Орган управления** | **Индикация на дисплее** |
| Drive | Движение вперёд | Рычаг, положение D | D |
| Neutral | Нейтральное положение | Рычаг, положение N | N |
| Reverse | Движение назад | Рычаг, положение R | R |

Устройство MPS-2 позволяет обучиться работе с:

* тактированием процессора;
* АЦП (аналого-цифровой преобразователь);
* ШИМ (широтно-импульсная модуляция);
* таймерами;
* прерываниями (как аппаратными, так и программными);
* библиотеками;
* интерфейсами;
* настройкой портов;
* дисплеями;
* световой индикацией;
* тумблерами;
* кнопками;
* энкодерами;
* драйверами;
* регистрами;
* программаторами.

1. **Программное обеспечение для работы с устройством**
   1. **Установка и настройка ПО Keil uVision**

**Keil uVision** - среда разработки, представляющая собой набор утилит для выполнения полного комплекса мероприятий по написанию программного обеспечения для микроконтроллеров.

Программа позволяет работать с проектами любой степени сложности, начиная с введения и правки исходных текстов и заканчивая внутрисхемной отладкой кода и программированием ПЗУ микроконтроллера. От разработчика скрыта большая часть второстепенных функций, что сильно разгружает интерфейс и делает управление интуитивно понятным. Однако при возрастании сложности реализуемых задач, всегда можно задействовать весь потенциал модулей, функционирующих под управлением единой оболочки.

Для установки программного обеспечения на ПК необходимо перейти на сайт ARMKeil во вкладку «Download», далее выбрать пункт «Product Downloads». Также можно перейти по ссылке ниже:

<https://www.keil.com/download/product/>

После перехода по ссылке нам необходимо выбрать конкретное программное обеспечение, предназначенное для наших контроллеров. Находим и выбираем пункт «MDK-Arm» (рис. 2.1), как следует из описания (Development environment for Cortex and Arm devices) данное ПО предназначено для разработки программ для контроллеров на базе ядра Cortex, коим является контроллер STM32F103C8T6.

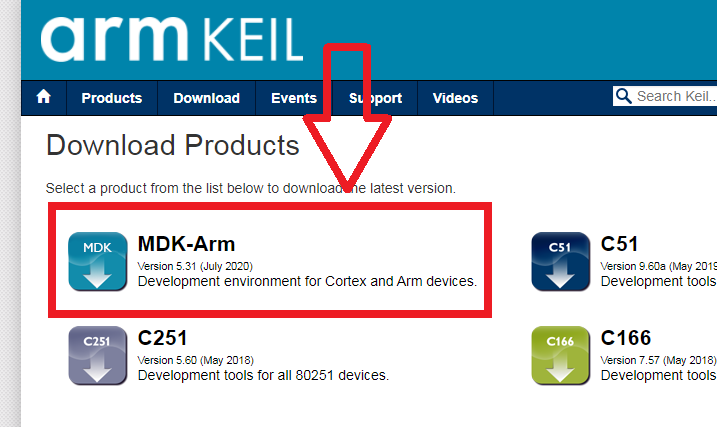


Рисунок 2.1 – Выбор программного обеспечения для скачивания

После выбора «MDK-Arm», необходимо указать контактные данные и нажать на «Submit». В случае верного ввода всех данных откроется доступ к ссылке (рис. 2.2) на скачивание программного обеспечения Keil uVision «MDK №версии программы.EXE».

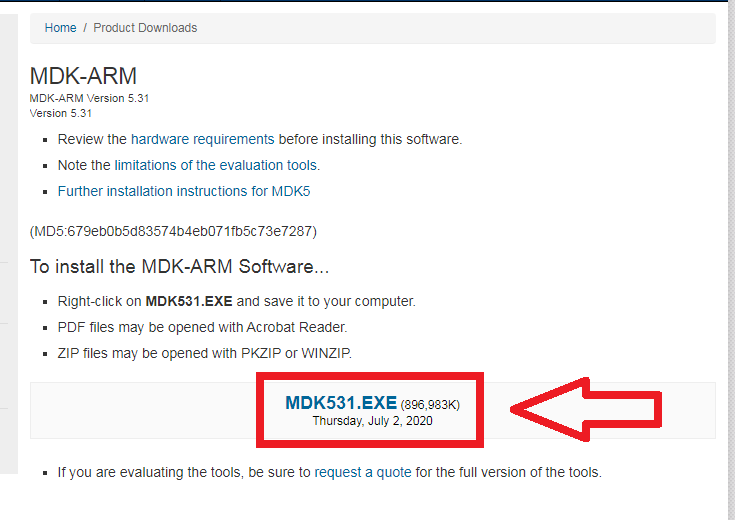


Рисунок 2.2 – Ссылка на скачивание Keil uVision

После окончания загрузки файла, необходимо запустить файл и назначить настройки на установку ПО Keil uVision (рис. 2.3).

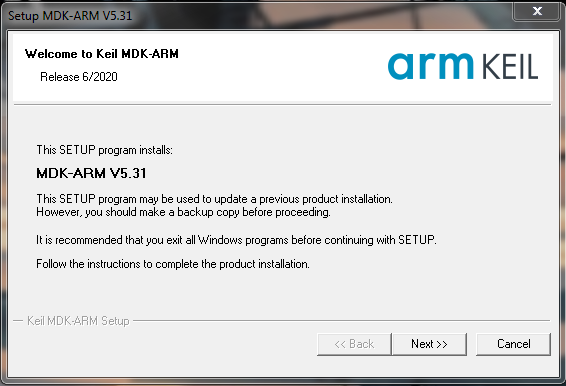


Рисунок 2.3 – Окно установки Keil uVision

Далее необходимо нажать "Next", согласиться с лицензионным соглашением, указать директорию для установки ПО, ввести информацию о пользователе, после начинается процесс установки программы на ПК.

В процессе установки (рис 2.4) необходимо разрешать устанавливать все необходимые драйверы и ПО связанные с ARM Ltd.

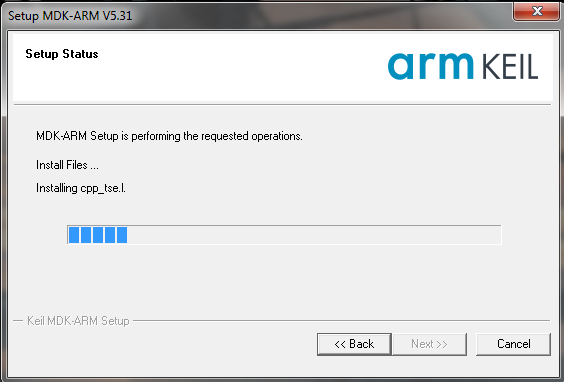


Рисунок 2.4 – Процесс установки программы Keil uVision

По окончании установки запускаем программу через автоматически созданный ярлык на Рабочем столе. При первом запуске будет открыто окно «Pack Installer» (рис. 2.5), в котором будет предложено скачать и установить драйвера и библиотеки на нужные микроконтроллеры.

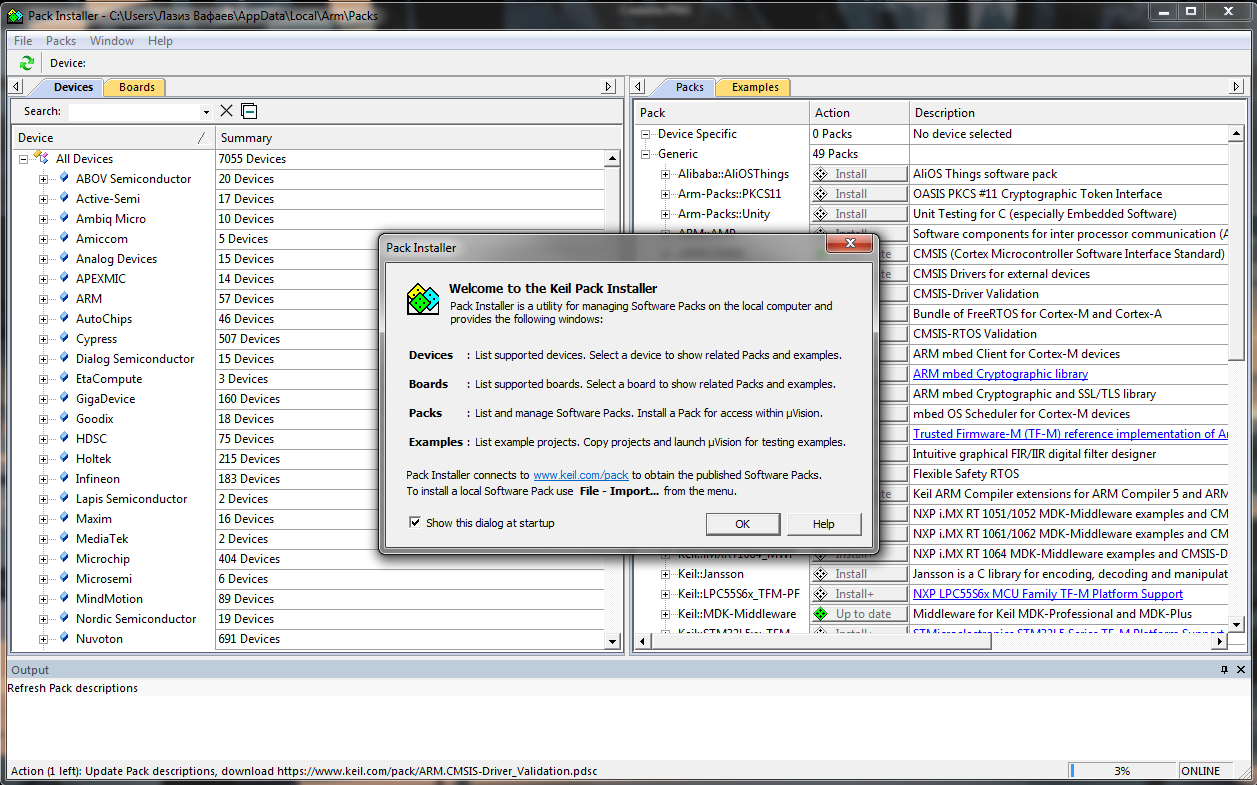


Рисунок 2.5 – Окно Pack Installer

Далее необходимо выбрать в списке устройств необходимый микроконтроллер «STMicroelectronics» >> «STM32F1 Series» >> «STM32F103» >> «STM32F103C8» (рис. 2.6).

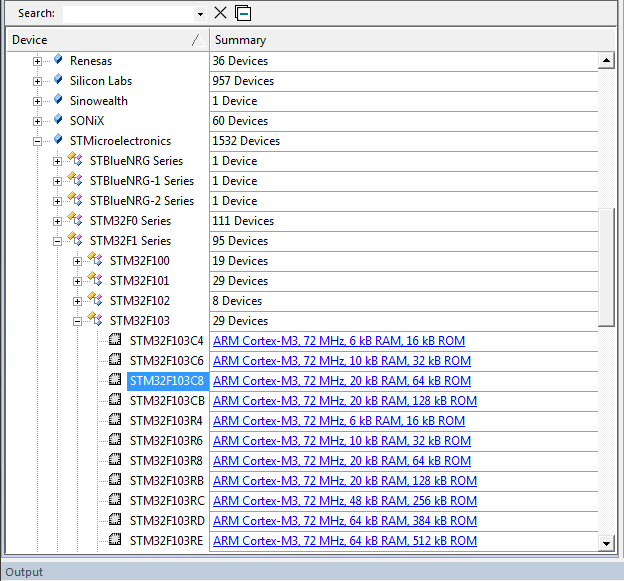


Рисунок 2.6 – Выбор контроллера STM32F103C8 в Pack Installer

После выбора нужного микроконтроллера справа будет доступен список актуальных драйверов и библиотек для выбранной платформы. В данном списке нужно нажать кнопки «Install» на необходимых пунктах (рис. 2.7), без которых работа с контроллером STM32F103C8T6 будет невозможна.

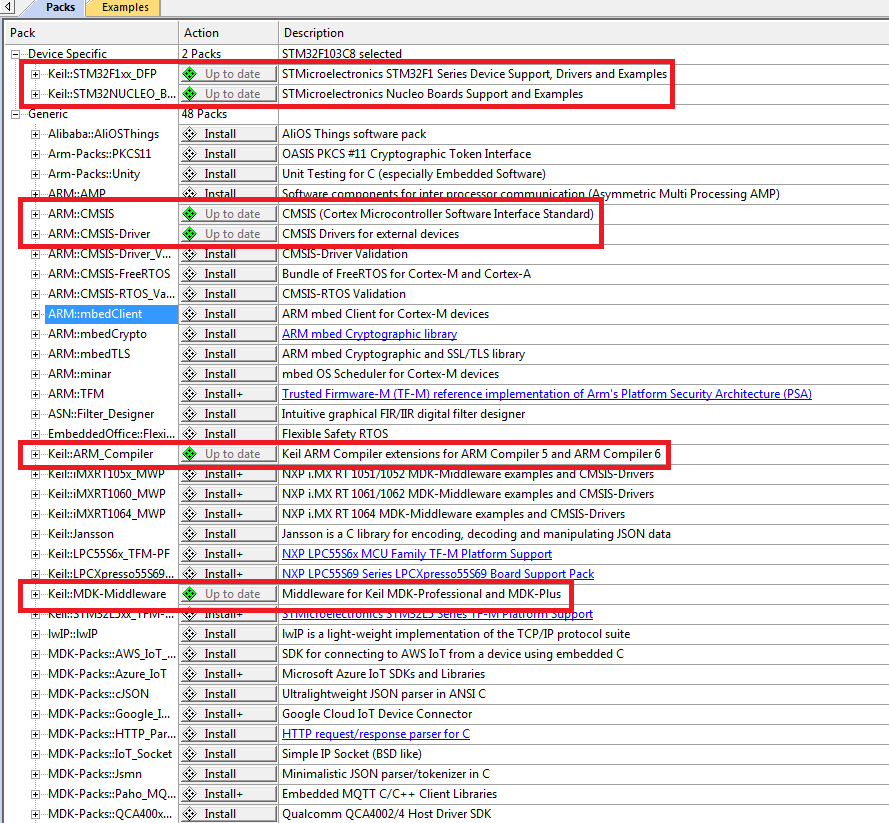


Рисунок 2.7 – Загрузка необходимых драйверов и библиотек

После установки необходимых драйверов и библиотек через Pack Installer закрываем окно и переходим непосредственно в среду разработки Keil uVision.

Далее можно создать проект, перейдя во вкладку «Project» и выбрав пункт «New uVision Project…». После этого действия откроется окно выбора имени и директории сохранения проекта. После сохранения проекта будет открыто окно «Select Device for Target» в котором нужно выбрать микроконтроллер, на который будет разрабатываться программа.

После выше перечисленных манипуляций можно приступить к созданию программы для контроллера STM32F103C8T6. Но так как рядовой пользователь может столкнуться со сложностями первичной настройки контроллера и работы его периферии необходимо установить программное обеспечение STM32 CubeMX, которое в свою очередь упрощает данный процесс при помощи графического интерфейса. После настройки контроллера все настройки можно импортировать в среду разработки   
Keil uVision.

* 1. **Установка и настройка ПО STM32 CubeMX**

**STM32CubeMX** – это генератор инициализационного кода для семейства микроконтроллеров STM32, позволяющий автоматически настроить всю периферию для данного кристалла или сгенерировать инициализационный код для какого-нибудь отладочного набора.

Встроенное программное обеспечение не только охватывает все микроконтроллеры STM32 с высокой переносимостью драйверов низкого уровня, но поставляется с набором компонентов middleware уровня, такие как RTOS, USB, TCP / IP, файловая система или графика. STM32 CubeMX помогает пользователю настроить микроконтроллер STM32 (распиновку, цепи тактирования и периферию) и программное обеспечение стека.

Для установки программного обеспечения на ПК необходимо перейти на сайт ST по ссылке ниже:

<https://www.st.com/en/development-tools/stm32cubemx.html#get-software>

После перехода по ссылке нам необходимо выбрать пункт «Get Software» (рис. 2.8), после нужно будет выбрать необходимую версию программного обеспечения (рис.2.9).

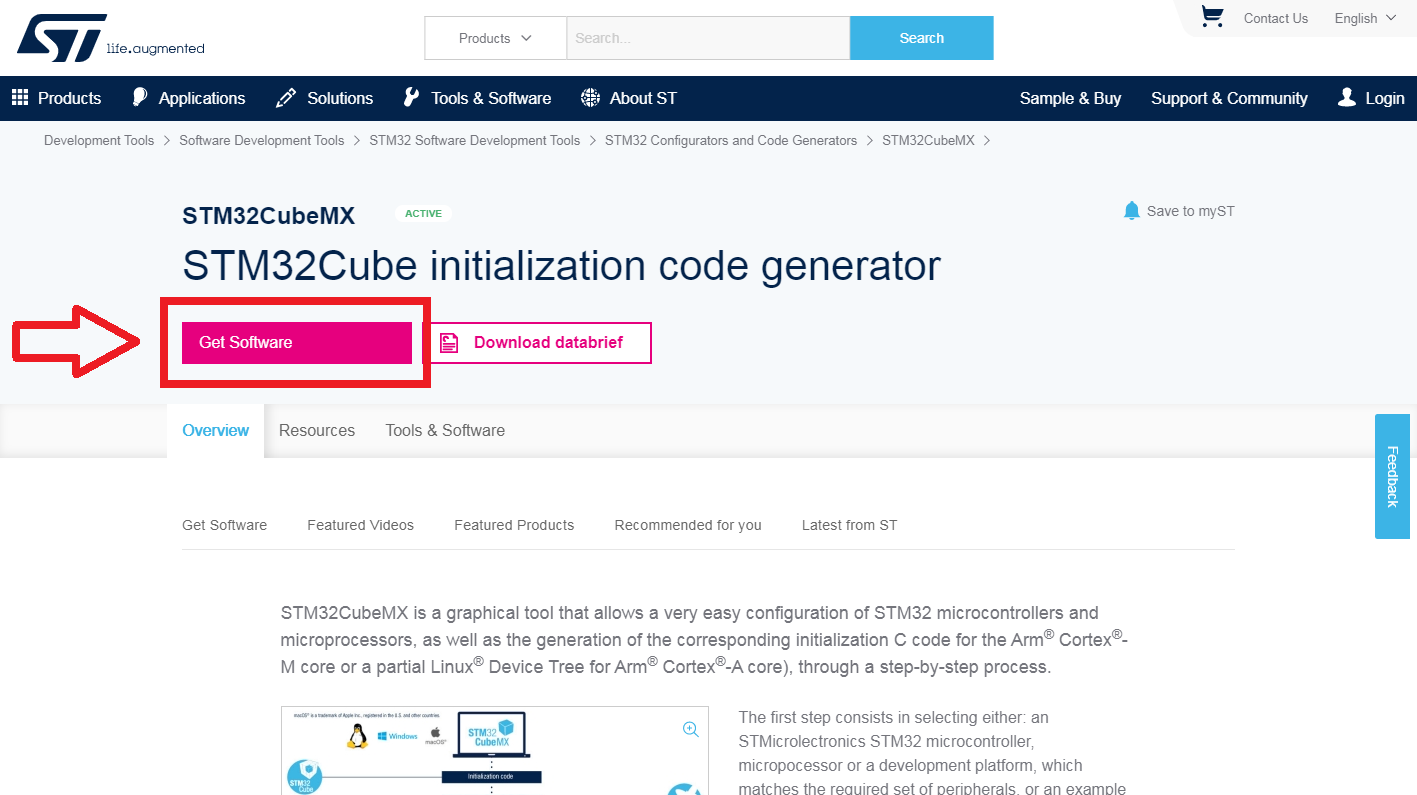


Рисунок 2.8 – Сайт для загрузки ПО STM32CubeMX

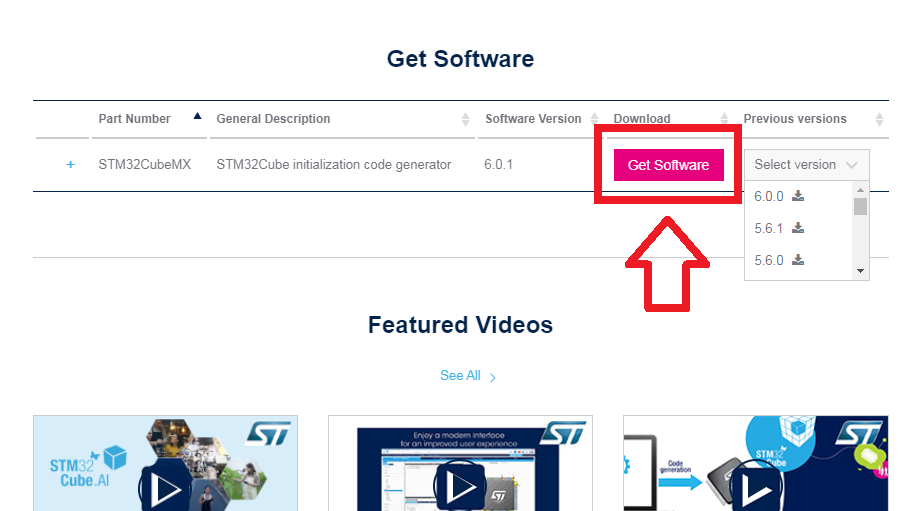


Рисунок 2.9 – Выбор версии ПО STM32CubeMX

Далее необходимо согласиться с лицензионным соглашением и корректно ввести контактные данные пользователя, одним из которых будет электронная почта, куда поступит письмо со ссылкой на скачивание STM32CubeMX. После чего нажать на пункт «Download», далее при помощи ссылки отправленной на электронную почту начнется загрузка программного обеспечения на ПК. После загрузки необходимо запустить исполняемый файл и начать установку (рис. 2.10).

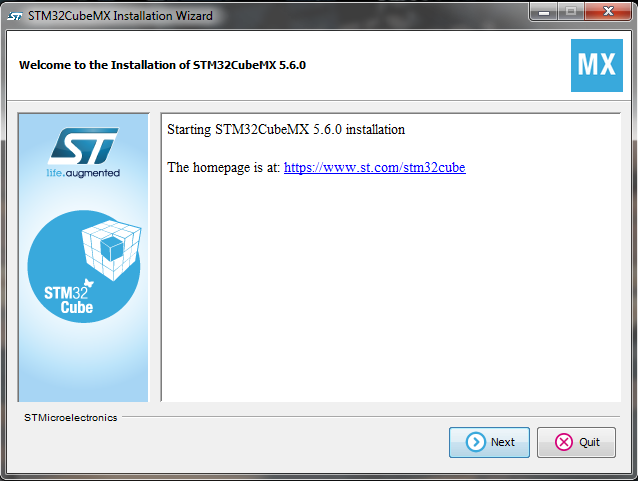


Рисунок 2.10 – Начало установки ПО STM32CubeMX

Далее необходимо выбрать директорию для установки программного обеспечения, согласиться со всеми лицензионными соглашениями и начать процесс установки ПО (рис. 2.11).

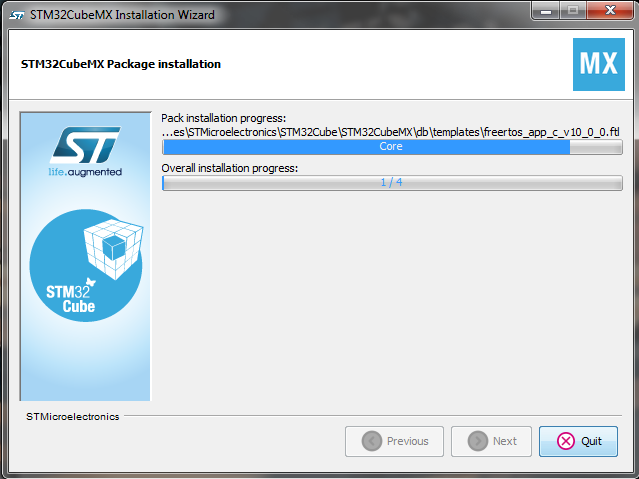


Рисунок 2.11 – Процесс установки ПО STM32CubeMX

После установки программного обеспечения, запускаем при помощи ярлыка созданного на рабочем столе. В начальном окне необходимо выбрать пункт «File» >> «New Project»(рис. 2.12).

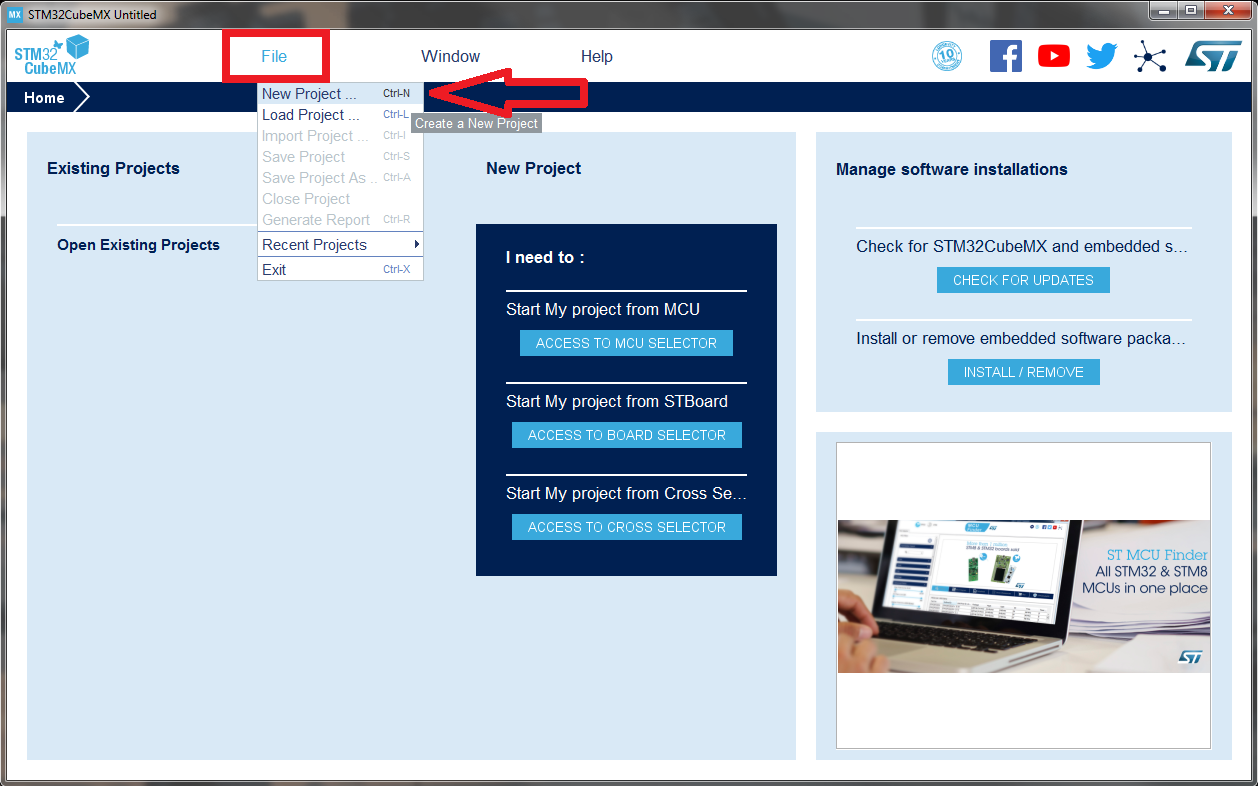


Рисунок 2.12 – Создание нового проекта в STM32CubeMX

В следующем окне необходимо через поиск найти микроконтроллер, с которым будет вестись работа, а именно STM32F103C8. После выбора микроконтроллера можно запускать проект через команду «Start Project» (рис. 2.13).

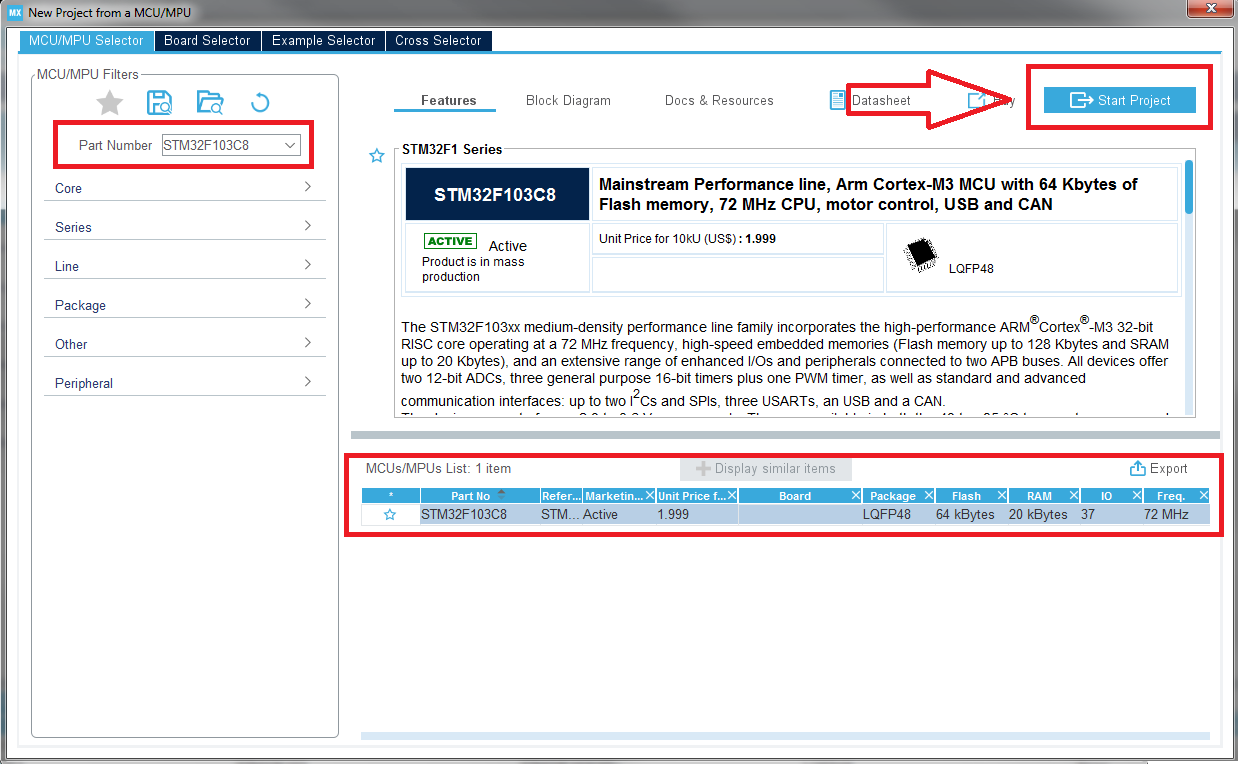


Рисунок 2.13 – Выбор целевого устройства и запуск проекта в STM32CubeMX

После запуска проекта необходимо провести конфигурацию ряда параметров. Слева во вкладке «System Core» находятся все необходимые настройки для первичного запуска проекта и импорта в среду разработки   
Keil uVision. Во вкладке «System Core» находим и выбираем пункт «RCC», в котором нужно провести настройку обработки информации. Используя пункт «High Speed Clock» выбираем «Crystal/Ceramic Resonator» (рис. 2.14). Далее переходим во вкладку «SYS» где определяем возможность отладки микроконтроллера, используя пункт «Debug» выбираем «Serial Wire»   
(рис. 2.15).

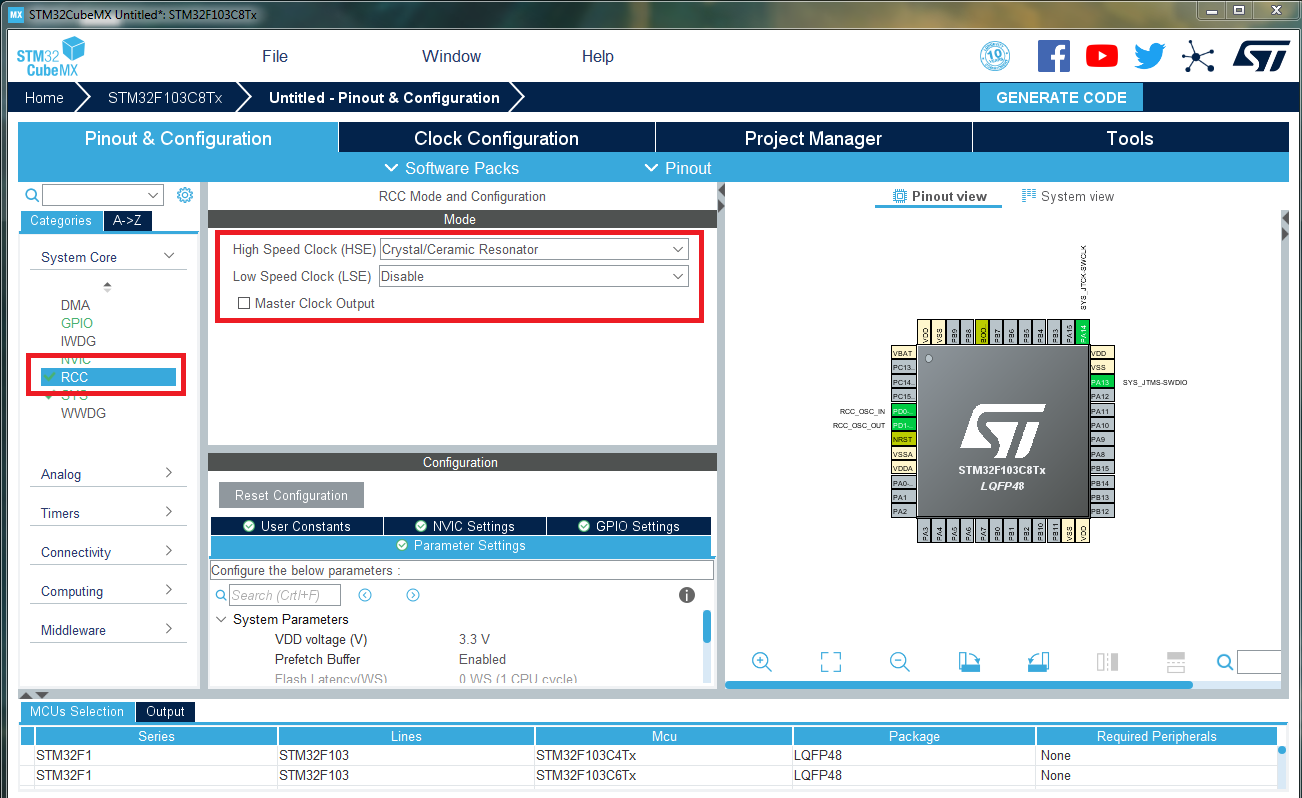


Рисунок 2.14 – Конфигурация параметров во вкладке «RCC»

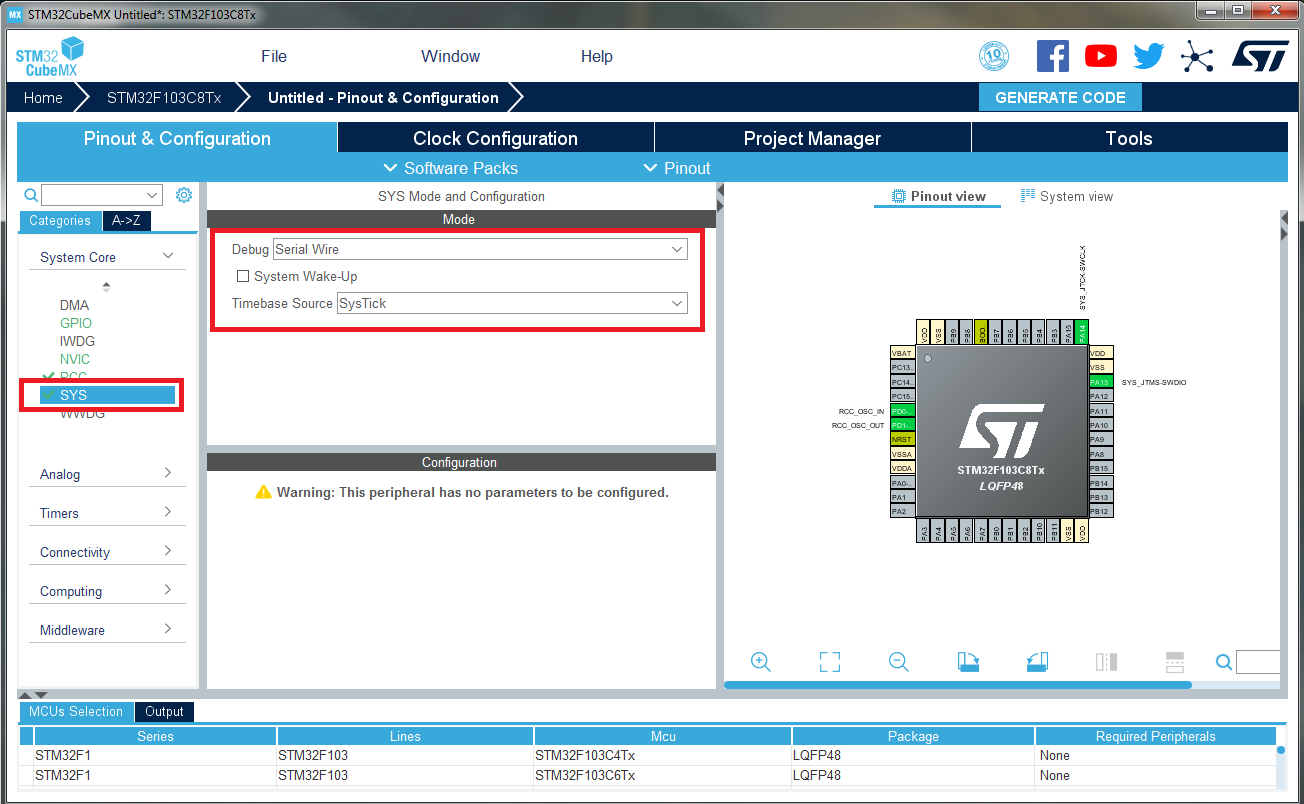


Рисунок 2.15 – Конфигурация параметров во вкладке «SYS»

Далее в панели «Project Manager» необходимо задать директорию и название проекта (в названии могут использоваться только английские буквы и цифры при условии, что цифры не будут в начале названия). Помимо этого нужно выбрать интегрированную среду разработки (IDE), в данном случае MDK-ARM V5 (рис. 2.16).

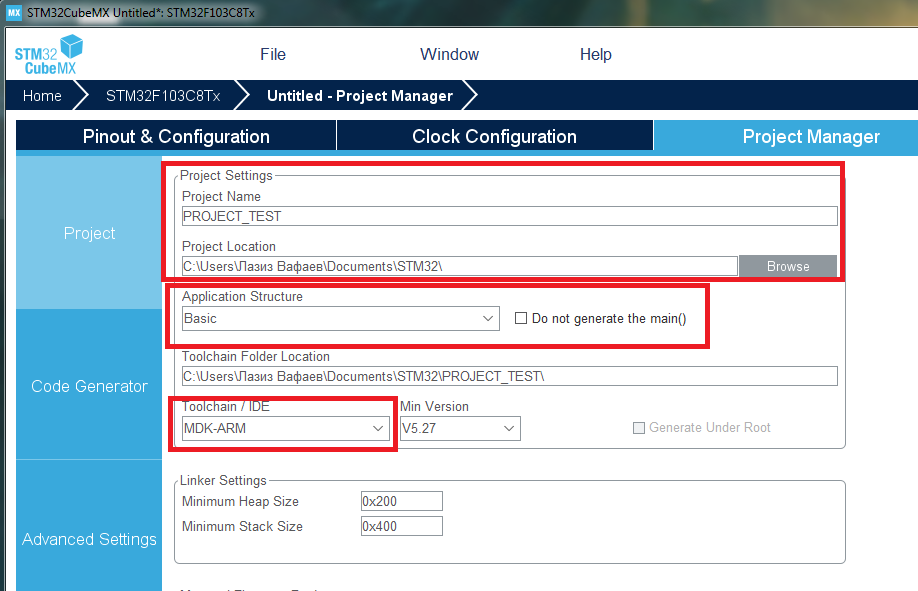


Рисунок 2.16 – Конфигурация параметров во вкладке «Project Manager»

Дополнительно во вкладке «Code Generator» выбираем пункт «Copy only necessary library files» (рис. 2.17) для того чтобы в проекте использовались только необходимые файлы библиотек. После этого при помощи пункта «Generate Code» в правом верхнем углу программы проводим импорт инициализационного кода в среду разработки Keil uVision.

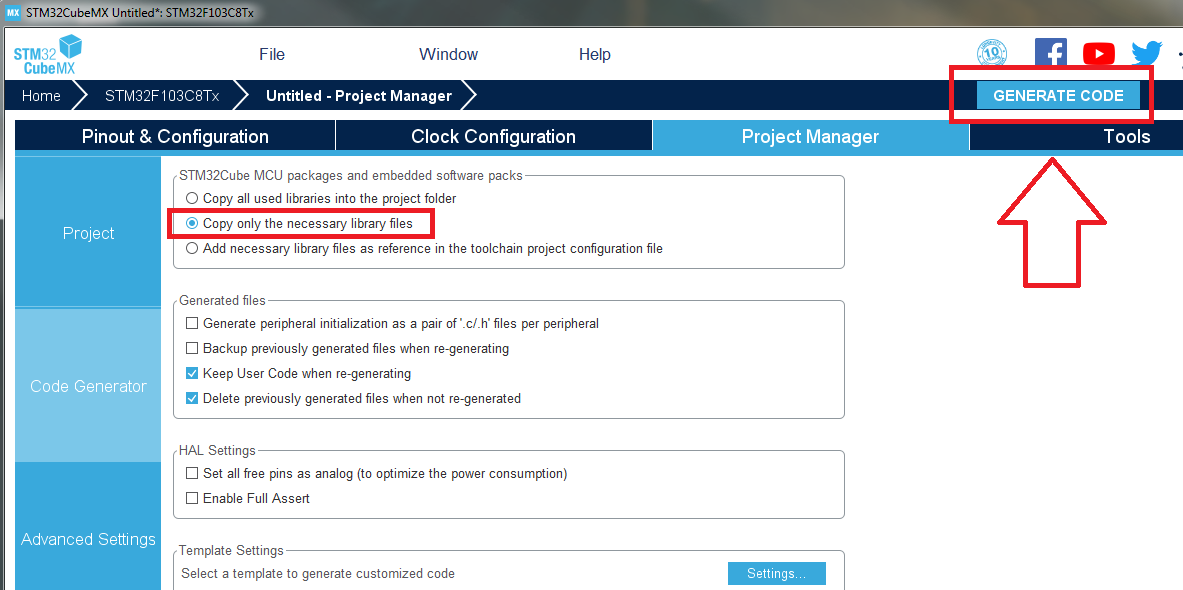


Рисунок 2.17 – Конфигурация параметров и запуск генерации кода

После импорта инициализационного кода из STM32 CubeMX в Keil uVision, можно непосредственно приступить к написанию программ для микроконтроллера STM32F103C8T6, подключив контроллер к ПК через программатор ST-LINK/V2 и проверив корректность установки драйверов для программатора для предотвращения ошибок при загрузке готового кода в микроконтроллер.

Все программное обеспечение является бесплатным (Keil uVision с некоторыми ограничениями), и может быть загружено с официальных сайтов. Очень важно: некоторые версии ПО STM32 CubeMX могут быть несовместимы или совместимы некорректно со средой Keil uVision, или наоборот, поэтому рекомендуем воспользоваться программным обеспечением по ссылке ниже:

<https://drive.google.com/drive/folders/1mNqGH9XPZZrZTV2538IpXE1apcMh7jCO?usp=sharing>

1. **Основы программирования в среде Keil uVision**
   1. **Интерфейс Keil uVision**

Основные компоненты интерфейса:

* Окно **Project** показывает исходные файлы проекта и выбранные программные компоненты. Внизу компонентов вы найдете файлы конфигурации и библиотеки;
* Проект поддерживает различные целевые устройства **Targets**. Они имеют простой менеджер конфигурации и могут быть использованы для отладки сборки или адаптации для разных аппаратных платформ;
* Окно **Manage Run-Time Environment** показывает все доступные программные компоненты, которые совместимы с выбранным устройством;
* **Configuration Wizard** - это утилита встроенного редактора для создания элементов управления конфигурацией, подобной GUI, в ассемблере, C/C ++ или файле инициализации;
* Окно **Functions** предоставляет быстрый доступ к функциям в каждом исходном C/C++ файле программного модуля;
* Автодополнение кода - **Code Completion** и список параметров функций – **Function Parameter** облегчает работу и позволяет лучше ориентироваться в коде;
* Автоматическая проверка синтаксиса - **Dynamic Syntax Checking**, проверяет синтаксис программы во время набора текста и выдает предупреждения в режиме реального времени о возможных ошибках в коде перед компиляцией.

Горячие клавиши:

* **F7**- сборка проекта
* **F8** - прошивка Flash
* **Ctrl + F5** - начать/завершить отладку
* **Ctrl + F**- поиск в файле
* **Ctrl + Shift + F** - поиск в файлах
* **Ctrl + S** – сохранить
* **Ctrl + O** – открыть
* **Ctrl + N** - новый файл
  1. **Синтаксис программной среды**

Основной код в программной среде Keil uVision пишется в созданном программой документе под названием «main.c».

Особенность при программировании контроллеров через Keil uVision является то, что код поделен на разные блоки. Определить каждый помогают автоматически созданные комментарии по их названию, они и разделяют блоки кода. К примеру, создание переменных проводится между комментариями «USER CODE BEGIN PV» и «USER CODE END PV» (рис. 3.1), в случае с присваиванием переменных началом является «USER CODE BEGIN 0» заканчиваясь на «USER CODE END 0».

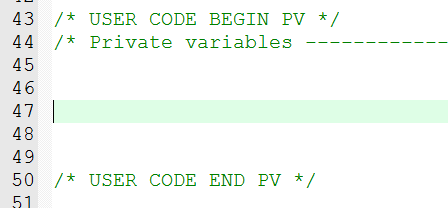


Рисунок 3.1 – Создание переменных

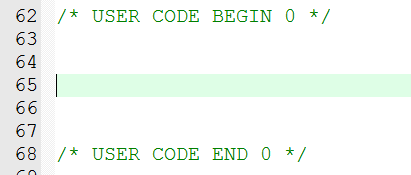


Рисунок 3.2 – Присваивание переменных

Комментарий «USER CODE BEGIN 1» является началом блока, где можно задать переменным значения находясь в основной функции (рис. 3.3), завершается блок соответственно на «USER CODE END 1».

«USER CODE BEGIN 2» является блоком для написания кода, который выполнится единожды при запуске программы на микроконтроллере (рис.3.4).

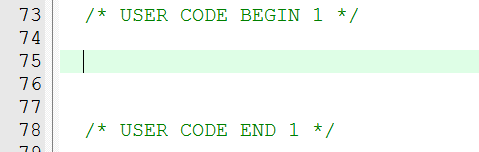


Рисунок 3.2 – Задание значений переменным

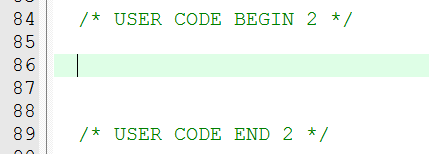


Рисунок 3.4 – Блок кода выполняющийся единожды

Основной же код записывается в цикле while и начинается с «USER CODE BEGIN WHILE» (рис. 3.5).

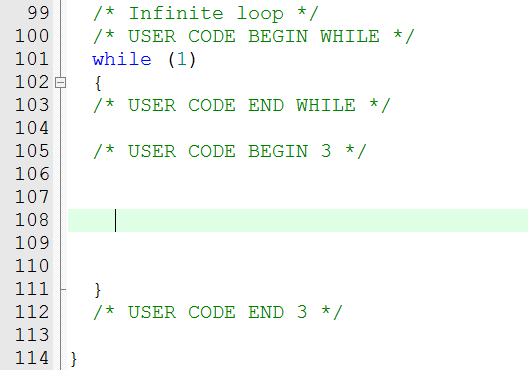


Рисунок 3.5 – Основной блок кода в цикле

* 1. **Типы данных**

Таблица 3.1 – Типы данных компилятора Keil uVision

|  |  |  |  |
| --- | --- | --- | --- |
| **Тип данных** | **Количество бит** | **Количество байт** | **Диапазон значений** |
| bit | 1 | — | 0, 1 |
| char | 8 | 1 | -128…+127 |
| unsigned char | 8 | 1 | 0…255 |
| enum | 8 или 16 | 1 или 2 | -128…+127  или -32768…+32767 |
| signed short | 16 | 2 | -32768…+32767 |
| unsigned short | 16 | 2 | 0…65535 |
| signed int | 16 | 2 | -32768…+32767 |
| unsigned int | 16 | 2 | 0…65535 |
| signed long | 32 | 4 | -2147483648… 2147483647 |
| unsigned long | 32 | 4 | 0…4294967295 |
| float | 32 | 4 | ±1.175494E-38…±3.402823E+38 |
| data \*, idata \*, pdata \* | 8 | 1 | 0x00...0xFF |
| code \*, xdata \* | 16 | 2 | 0x0000...0xFFFF |
| generic pointer | 24 | 3 | Memory type (1 байт); Offset (2 байта) 0...0xFFFF |

* 1. **Операторы**

Таблица 3.2 – Арифметические операторы

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Операция (выражение)** | **Оператор** | **Синтаксис выражения** | **Пример** | |
| **Член типа T** | **Определение вне класса** |
| [Присваивание](https://ru.wikipedia.org/wiki/%D0%9E%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D1%8F_%D0%BF%D1%80%D0%B8%D1%81%D0%B2%D0%B0%D0%B8%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F_%D0%B2_%D0%A1%2B%2B) | **=** | **a = b** | R& T::operator =(S b); | н/д |
| Сложение | **+** | **a + b** | R T::operator +(S b); | R operator +(T a, S b); |
| Вычитание | **-** | **a - b** | R T::operator -(S b); | R operator -(T a, S b); |
| Унарный плюс | **+** | **+a** | R T::operator +(); | R operator +(T a); |
| Унарный минус | **-** | **-a** | R T::operator -(); | R operator -(T a); |
| Умножение | **\*** | **a \* b** | R T::operator \*(S b); | R operator \*(T a, S b); |
| Деление | **/** | **a / b** | R T::operator /(S b); | R operator /(T a, S b); |
| Операция модуль ([остаток от деления](https://ru.wikipedia.org/wiki/%D0%94%D0%B5%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5_%D1%81_%D0%BE%D1%81%D1%82%D0%B0%D1%82%D0%BA%D0%BE%D0%BC) целых чисел) | **%** | **a % b** | R T::operator %(S b); | R operator %(T a, S b); |
| [Инкремент](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D0%BA%D1%80%D0%B5%D0%BC%D0%B5%D0%BD%D1%82) префиксный | **++** | **++a** | R& T::operator ++(); | R& operator ++(T a); |
| [Инкремент](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D0%BA%D1%80%D0%B5%D0%BC%D0%B5%D0%BD%D1%82) суффиксный | **++** | **a++** | R T::operator ++(int); | R operator ++(T a, int); |
| [Декремент](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D0%BA%D1%80%D0%B5%D0%BC%D0%B5%D0%BD%D1%82) префиксный | **--** | **--a** | R& T::operator --(); | R& operator --(T a); |
| [Декремент](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D0%BA%D1%80%D0%B5%D0%BC%D0%B5%D0%BD%D1%82) суффиксный | **--** | **a--** | R T::operator --(int); | R operator --(T a, int); |

Таблица 3.3 – Операторы сравнения

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Операция (выражение)** | **Оператор** | **Синтаксис выражения** | **Пример** | |
| **Член типа T** | **Определение вне класса** |
| Равенство | == | a == b | R T::operator ==(S b); | R operator ==(T a, S b); |
| Неравенство | != | a != b | R T::operator !=(S b); | R operator !=(T a, S b); |
| Больше | > | a > b | R T::operator >(S b); | R operator >(T a, S b); |
| Меньше | < | a < b | R T::operator <(S b); | R operator <(T a, S b); |
| Больше или равно | >= | a >= b | R T::operator >=(S b); | R operator >=(T a, S b); |
| Меньше или равно | <= | a <= b | R T::operator <=(S b); | R operator <=(T a, S b); |

Таблица 3.4 – Логические операторы

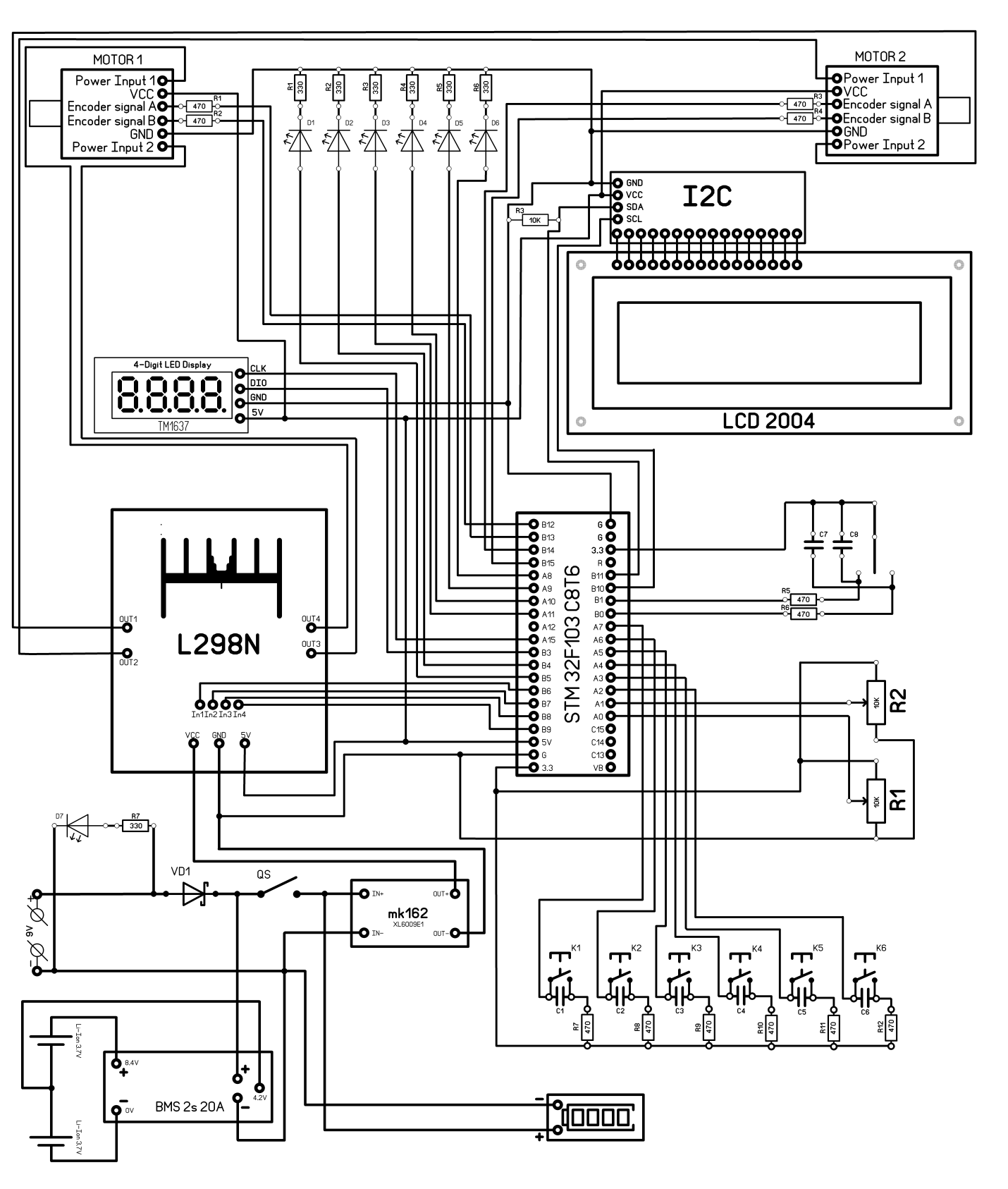
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Операция (выражение)** | **Оператор** | **Синтаксис выражения** | **Пример** | |
| **Член типа T** | **Определение вне класса** |
| Логическое отрицание, НЕ | ! | !a | R T::operator !(); | R operator !(T a); |
| Логическое умножение, И | && | a && b | R T::operator &&(S b); | R operator &&(T a, S b); |
| Логическое сложение, ИЛИ | || | a || b | R T::operator ||(S b); | R operator ||(T a, S b); |

Таблица 3.5 – Побитовые операторы

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Операция (выражение)** | **Оператор** | **Синтаксис выражения** | **Пример** | |
| **Член типа T** | **Определение вне класса** |
| Побитовая инверсия | ~ | ~a | R T::operator ~(); | R operator ~(T a); |
| Побитовое И | & | a & b | R T::operator &(S b); | R operator &(T a, S b); |
| Побитовое ИЛИ (or) | | | a | b | R T::operator |(S b); | R operator |(T a, S b); |

**ПРИЛОЖЕНИЕ А**

Электрическая принципиальная схема   
Стенда учебного программируемого MPS2   
(Module Programmable STM32F103)

****

**ПРИЛОЖЕНИЕ Б**

Функциональная распиновка   
микроконтроллера STM32F103C8T6

