Додаток А. **Встановлення та налаштування   
Keil μVision5**

1. **Завантаження середовища програмування мікроконтролерів STM32 та пошук документації для них[[1]](#footnote-1)**

Почнемо з того, що визначимо тип мікроконтролеру на якому створено оціночний модуль (*Evaluations Board STM32F401 NUCLEO абоSTM32F4 DISCOVERY* ). Це можна зробити за допомогою написів на коробці з ним (рис. 1) або наклейки на платі модулю (рис. 2).

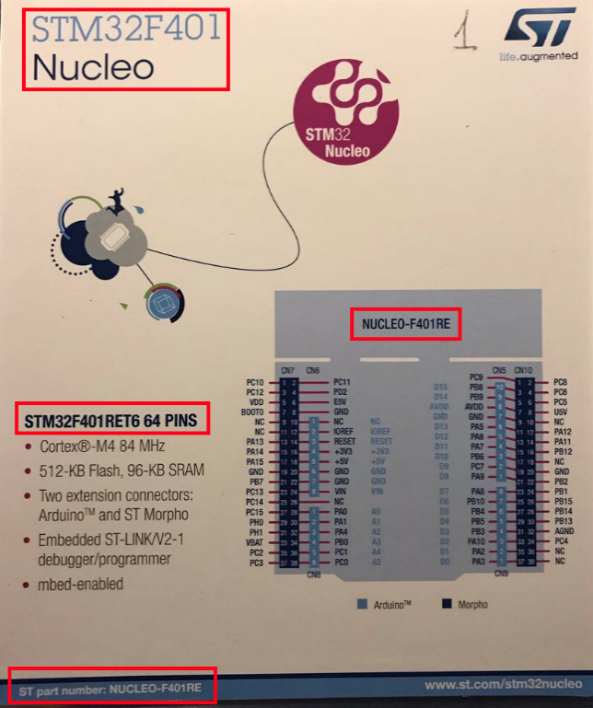


Рисунок 1 - Фото коробки мікроконтролера

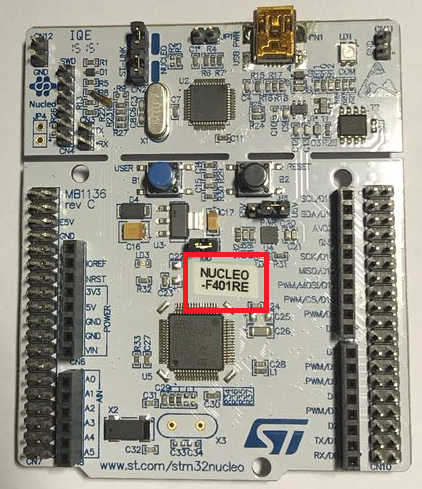


Рисунок 2 - Фото оціночної плати

Бачимо, що тут застосовано мікроконтролер компанії STMicroelectronics STM32F401RE.

Перейдемо на офіційний сайт розробника за адресою «[www.st.com](http://www.st.com/)». У вкладці «Products» обираємо пункт «Microcontrollers» (рис. 3).



Рисунок 3 - Сайт виробника мікроконтролерів

Далі обираємо тип мікроконтролеру, відповідно до того, що ми щойно визначили, у нашому випадку це «STM32» (рис. 4).

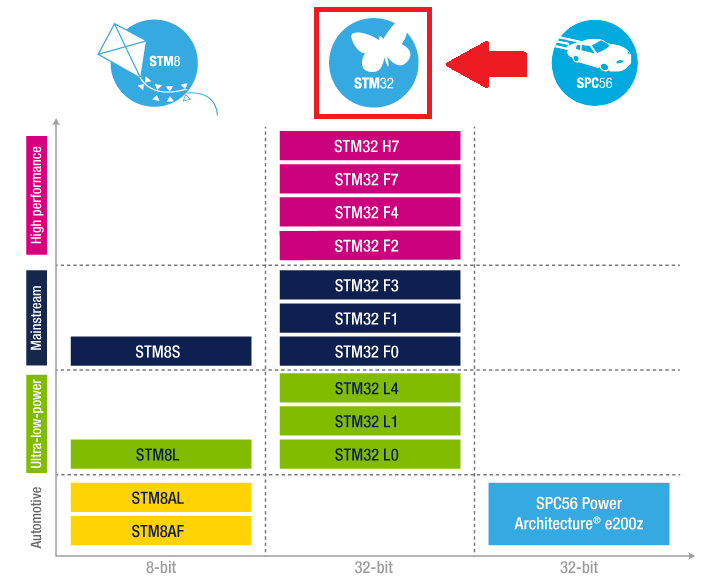


Рисунок 4 - Вибір типу мікроконтролеру

Тепер необхідно вибрати модель плати, на якій знаходиться мікроконтролер, у нашому випадку це «Nucleo», тому обираємо відповідний пункт «STM32 Nucleo development boards, …» (рис. 5).

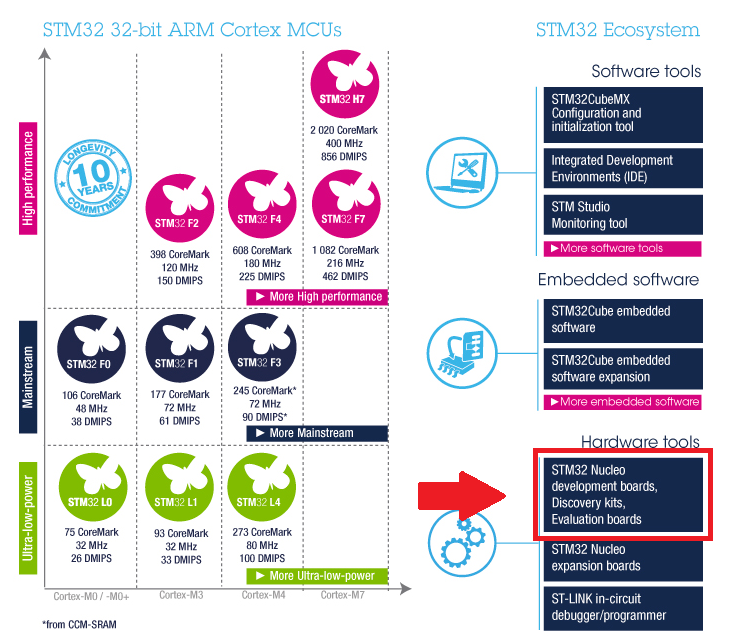


Рисунок 5 - Вибір моделі плати, на якій розміщено мікроконтролер

Відкриється сторінка зі списком плат, обираємо в ньому вкладку, що містить необхідну плату і шукаємо її там (рис. 6). Зверніть увагу на те, що список складається з декількох сторінок і під ним можна вибрати кожну з них. Звідси завантажимо «user manual» для «*STM32 Nucleo-64 board*» у форматі PDF (рис. 7). В кінці цього документу є схеми, за якими ми будемо орієнтуватися при роботі з виводами на платі.

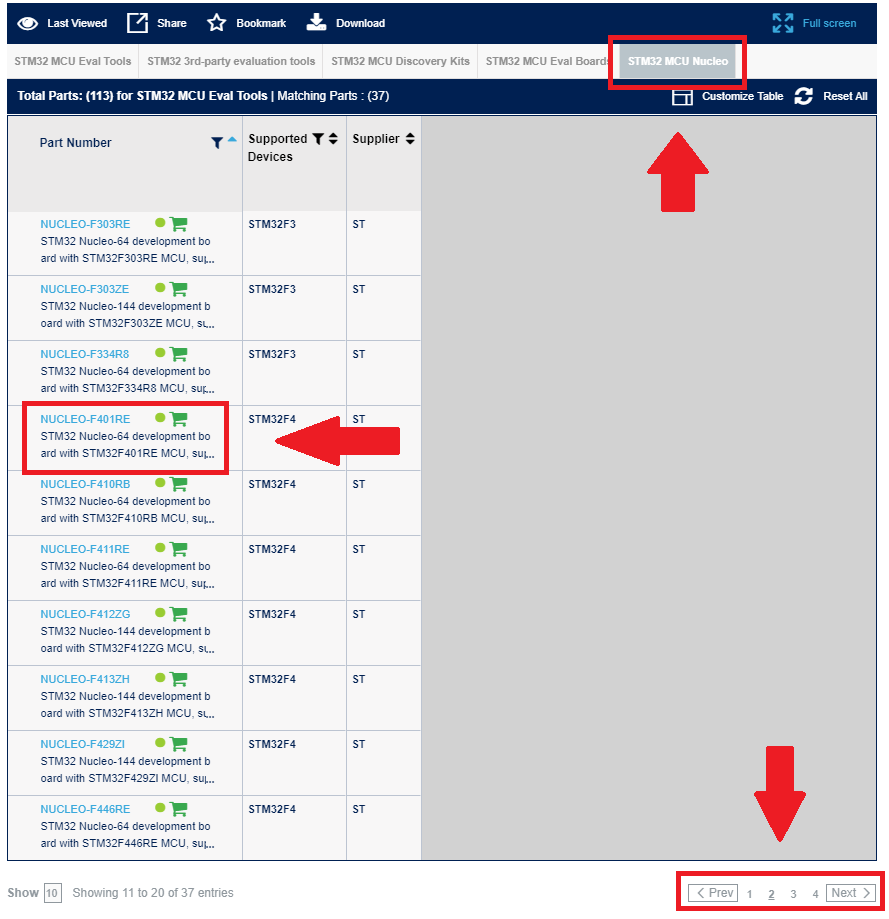


Рисунок 6 - Список плат даної моделі

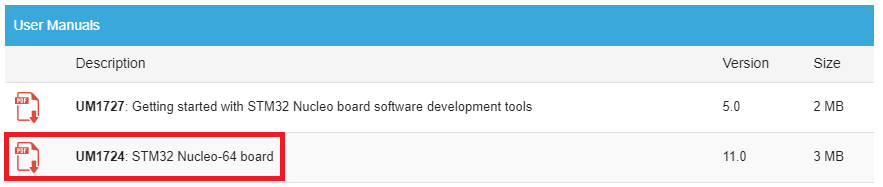


Рисунок 7 - Документ, що містить схеми плати

Повернемося на сторінку, де ми вибирали тип мікроконтролеру, але на цей раз оберемо конкретну його серію, у нашому випадку це «STM32 F4» (рис. 8).

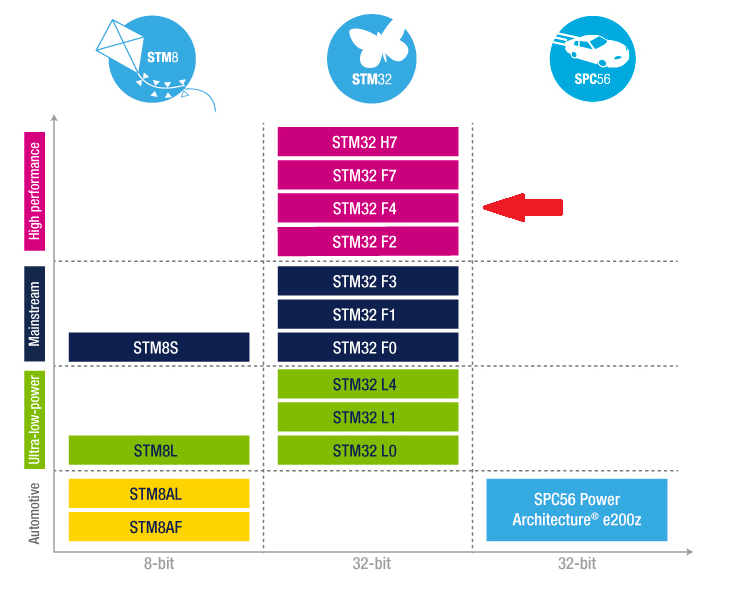


Рисунок 8 - Меню вибору серії мікроконтролеру

Тепер шукаємо у списку необхідну назву і натискаємо на неї, у нашому випадку це «STM32F401RE»   
(рис. 9).

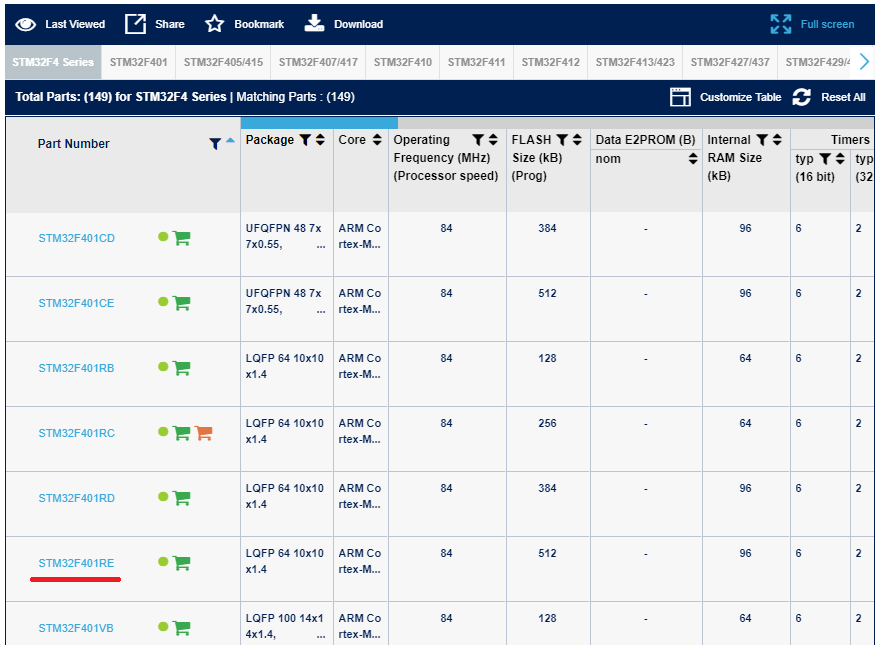


Рисунок 9 - Список всіх мікроконтролерів обраної серії

З цієї сторінки також раджу завантажити собі PDF файли, зображені на рис. 10. Верхній – це «reference manual», в ньому прописані регістри мікроконтролера, їх адреси в пам’яті, функції та багато чого ще. Нижній файл – це «programing manual», в ньому прописані інструкції, що підтримуються даним мікроконтролером і доступні програмісту, та деталі їх використання.

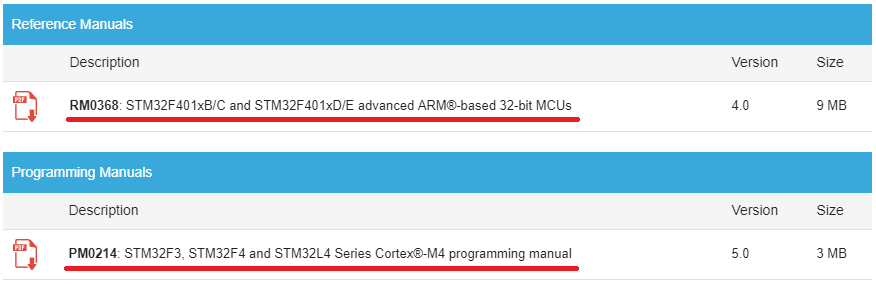


Рисунок 10 - Рекомендовані до завантаження документи

Нижче, на цій же сторінці, є перелік середовищ програмування (IDE) для обраного мікроконтролера та посилання на сайти розробників. Обираємо «MDK-ARM-STM32» від компанії «Keil» (рис. 11).

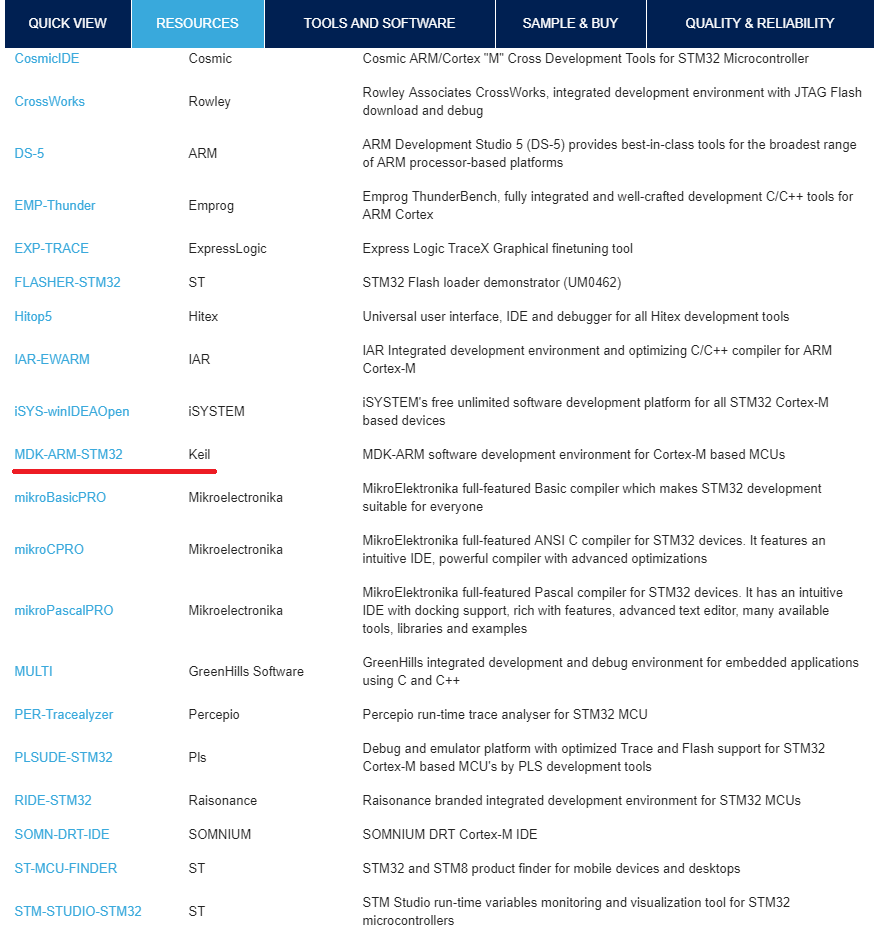


Рисунок 11 - Список IDE, що підтримують даний мікроконтролер

Перейшовши за адресою «[www2.keil.com/mdk5/install](http://www2.keil.com/mdk5/install)» ви потрапите на сайт для завантаження «MDK-Lite». Це безплатна версія, її найбільша відмінність від повноцінної платної версії в обмеженні розміру коду до 32Кб, однак цього достатньо для наших цілей. Натиснувши кнопку «Download MDK-Core» ви потрапите на сторінку, де необхідно буде заповнити форму анкети, після чого відправити її натиснувши на кнопку «Submit».

Якщо всі поля заповнені вірно, то відкриється сторінка завантаження «MDK-ARM» (див. рис. 15 на ст. 9). Розмір цього файлу 800 Мб, однак трохи пізніше нам доведеться ще завантажити додатковий матеріал, що забезпечить підтримку нашої плати даним IDE. Сумарний розмір після завантаження всіх компонентів та їх встановлення складатиме приблизно 4.8 Гб.

Нижче надано інший (більш простий) шлях закачування програми для завантаження MDK-ARM Keil µVision, але без посилань на файли документації.

1. **Прямий шлях завантаження програми KEIL µVision5**

Нижче надано стислі рекомендації по встановленню IDE Keil µVision5 безпосередньо з сайту Keil.

2.1. Встановлення IDE Keil µVision з MDK-ARM Keil.   
2.1.1 Скачати програму встановлення IDE можна з сайту її виробника (рис.12) за посиланням:  
<https://www.keil.com> або з відповідного розділу <https://www.keil.com/download/product/>

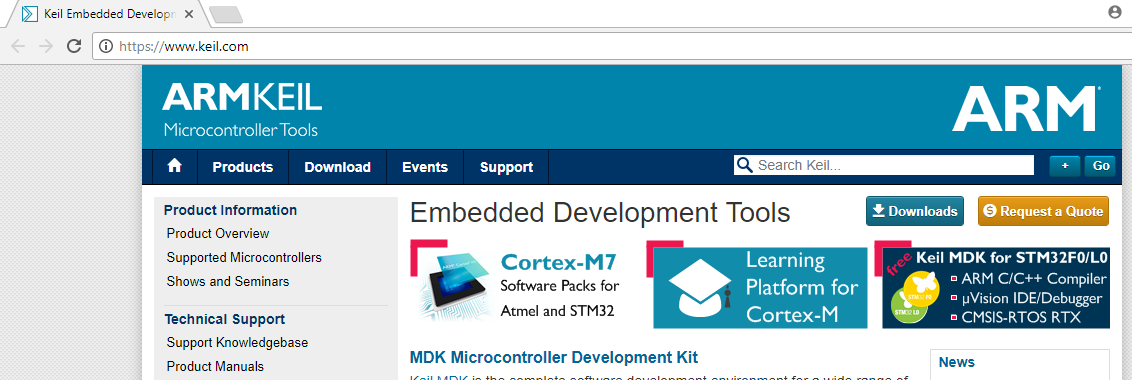


Рисунок12 - Вибір операції загрузки ПО

Тиснемо кнопку «Downloads» і на наступної вкладці (рис.13) вибираємо для загрузки пункт MDK-ARM останньої версії (на момент написання матеріалу такою була версія 5.24а).

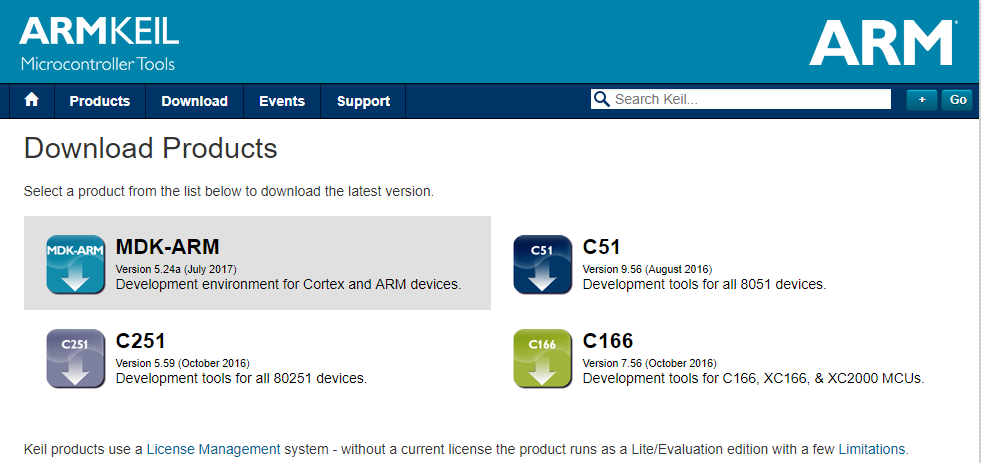


Рисунок13 - Вибір середовища розробки MDK-ARM

Далі ми попадаємо на сторінку, де необхідно ввести свої персональні данні (рис. 14). Заповніть анкету, і натисніть «Submit»[[2]](#footnote-2).

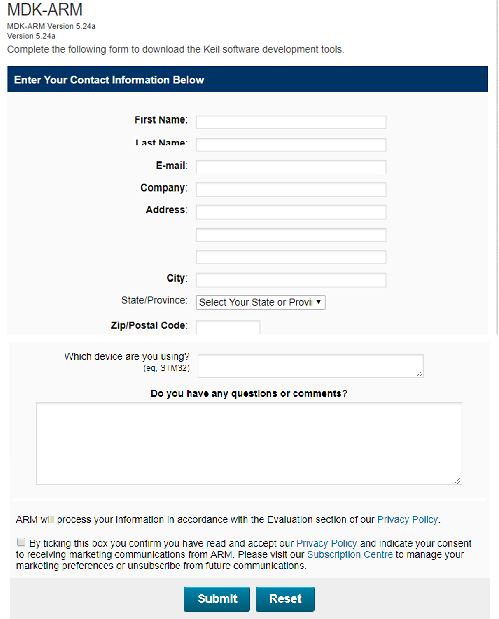


Рисунок14 – Анкета для реєстрації

Далі вибираємо актуальну на даний момент версію програми і тиснемо посилання (рис.15).

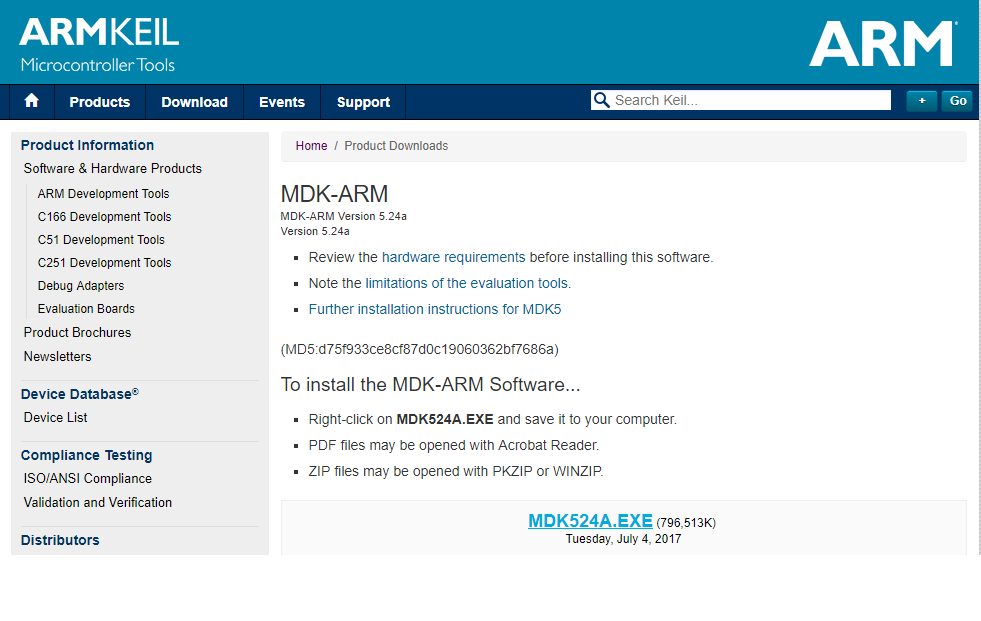


Рисунок 15 - Запуск процедури загрузки програми

Чекаємо поки програма загружчик буде записано на ваш комп’ютер!

1. **Встановлення середовища програмування мікроконтролерів**

Відкриваємо щойно завантажений файл **setupMDK-ARMv5.24a.exe** і починаємо встановлення програми «MDK-ARM» (рис. 16).

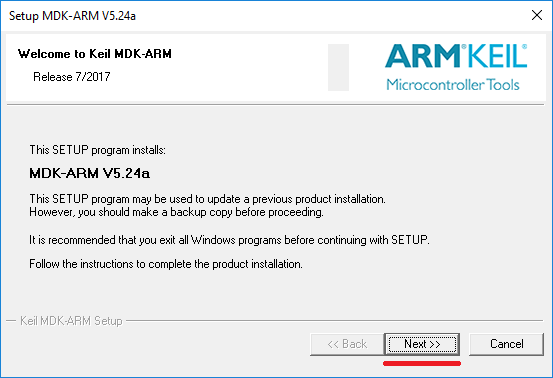


Рисунок 16 - Початок встановлення IDE

Погоджуємося з ліцензією (рис. 17).

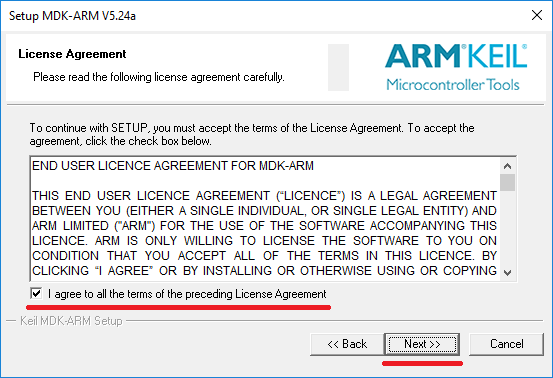


Рисунок 17 - Ліцензійний договір на використання IDE

Обираємо шлях на жорсткому диску для встановлення IDE (заповнюємо лише верхнє поле, нижнє заповниться автоматично відносно верхнього) (рис. 18).

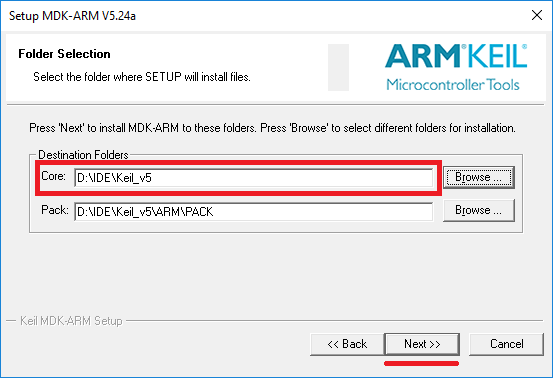


Рисунок 18 – Шлях встановлення IDE

Знову заповнюємо форму, однак ці дані можуть відрізнятися від тих, що ми вводили при реєстрації на сайті (рис. 14), після чого має початися процес встановлення (рис. 19, рис.20).

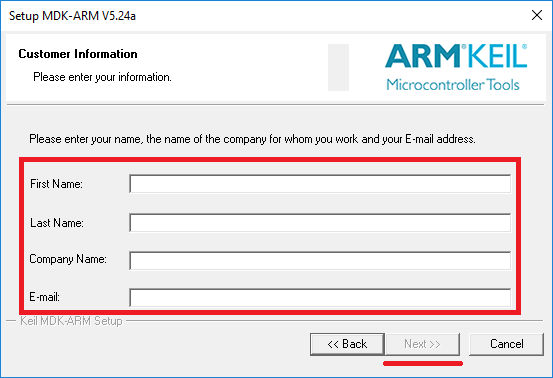


Рисунок 19 - Форма з інформацією про користувача

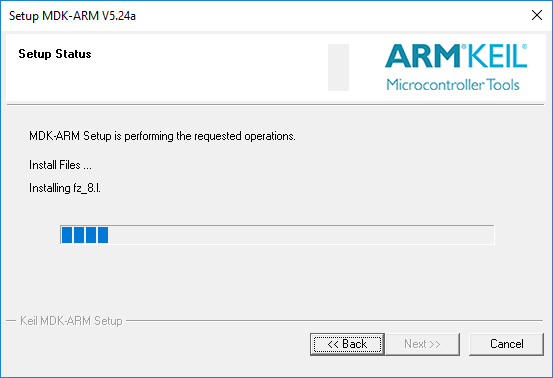


Рисунок 20 - Процес встановлення IDE

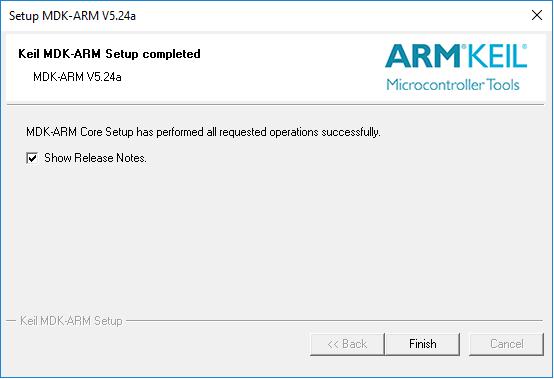


Рисунок 21 - Завершення процесу встановлення IDE

Після встановлення знімаємо прапорці з непотрібних пунктів і закриваємо програму встановлення.

Але, зараз встановлені лише базові компоненти середовища, яких не достатньо для того щоб почати програмувати наш мікроконтролер. Ми виправимо це в наступних пунктах.

1. **Встановлення Windows-драйверів загружчика STLink  
   для мікроконтролерів STM32**

Для забезпечення взаємодії IDE із платою по USB інтерфейсу потрібно встановити відповідні драйвери Windows. Для цього зайдемо в теку, яка була обрана для встановлення програми у попередньому пункті, далі в теку «ARM/STLink/USBDriver» після чого запукаємо програму «dpinst\_x86.exe», якщо система 32-бітна, або «dpinst\_amd64.exe», якщо система 64-бітна (рис. 22).

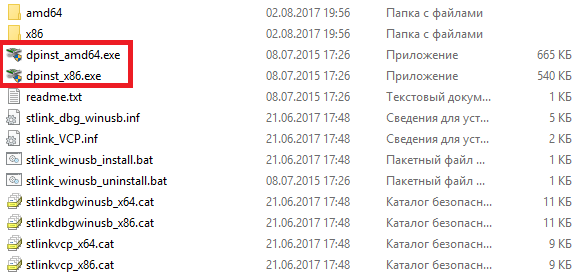


Рисунок 22 - Вибір драйверів у відповідності до операційної системи

Починаємо встановлення (рис. 23).

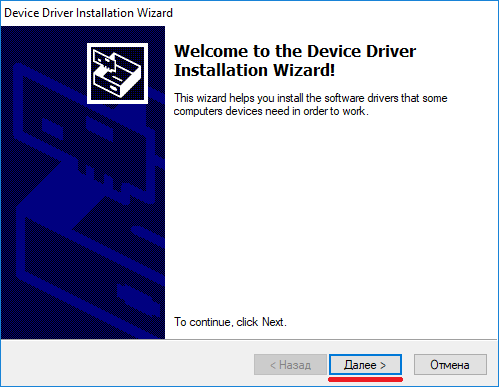


Рисунок 23 - Початок встановлення драйверів

Дозволяємо встановлення компонентів (рис. 24), після чого має з’явитися вікно зображене на рис. 25. Статус встановлення всіх драйверів має бути «Ready to use».

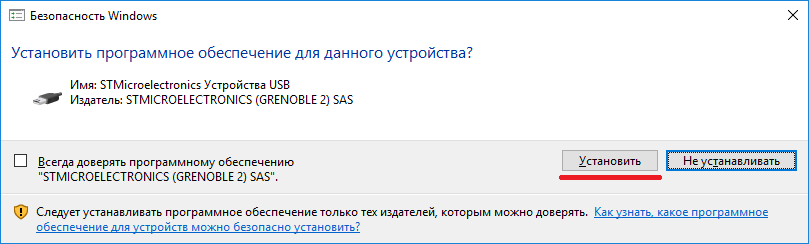


Рисунок 24 - Дозвіл на встановлення програмного забезпечення

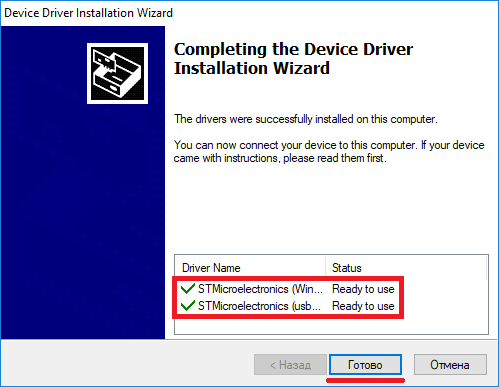


Рисунок 25 - Вікно зі статусом встановлення драйверів

1. **Налаштування середовища програмування мікроконтролерів STM32**

Почнемо налаштування щойно встановленого IDE зі встановлення необхідних пакетів, знову переходимо в теку, до якої ми встановлювали середовище програмування, далі в «UV4» і запускаємо програму «PackInstaller.exe» (рис. 26).[[3]](#footnote-3)

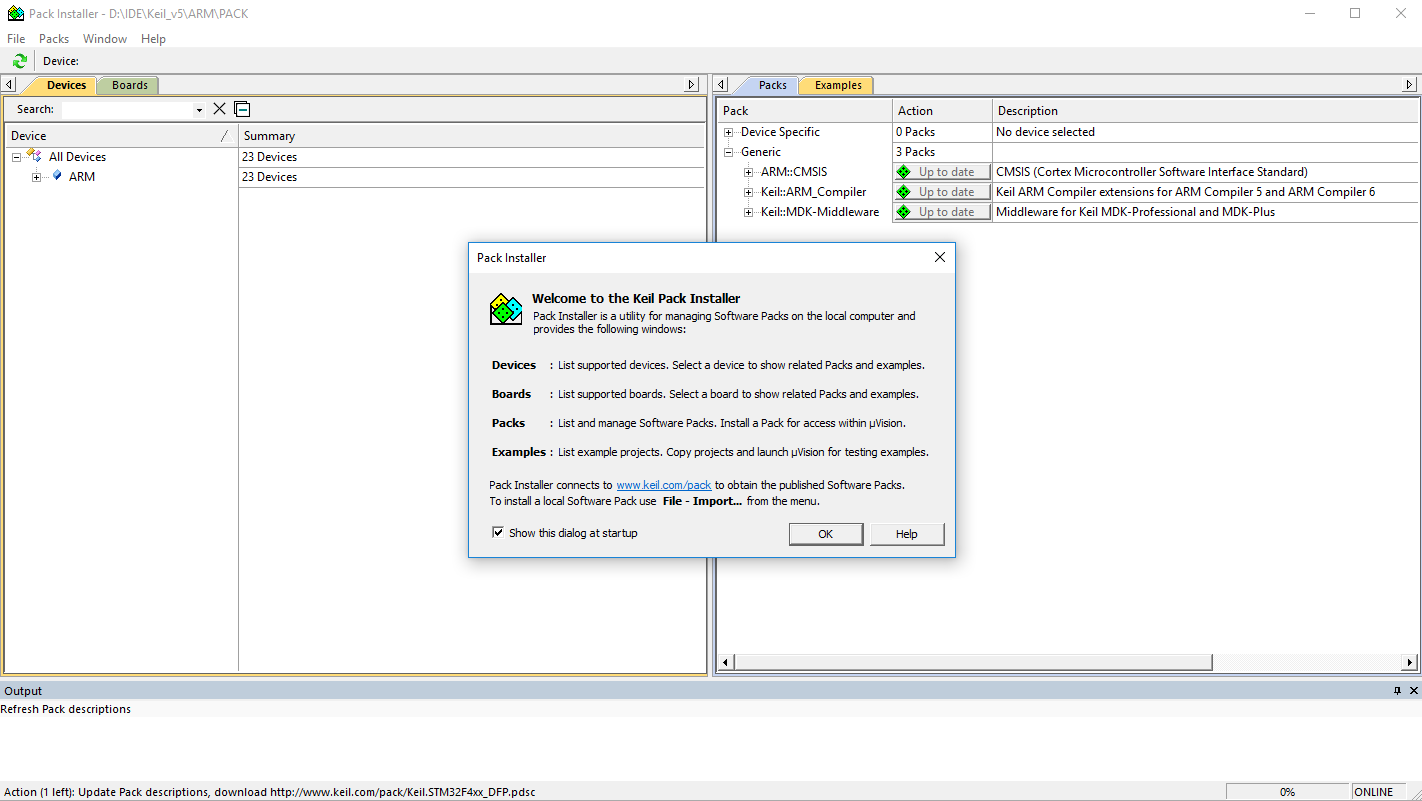


Рисунок 26 - Вікно менеджер пакетів середовища програмування

Однак можливо вона відкриється автоматично після встановлення IDE. Знадобиться підключення до інтернету для того, щоб дана програма завантажила необхідні ресурси. Після того, як в лівому нижньому куті вікна «PackInstaller» з’явиться напис «Ready», в лівій частині вікна, у вкладці «Devices», виберіть зі списку назву необхідного мікроконтролеру, в нашому випадку це «STM32F401RETx» (рис. 27), тепер в правій частині вікна з’явиться перелік пакетів доступних для обраного мікроконтролеру. Натискаємо на кнопку «Install» пакунків, позначених на рис. 27 і чекаємо закінчення їх встановлення, про що буде сигналізувати зміна напису на кнопках з «install» на «up to date».

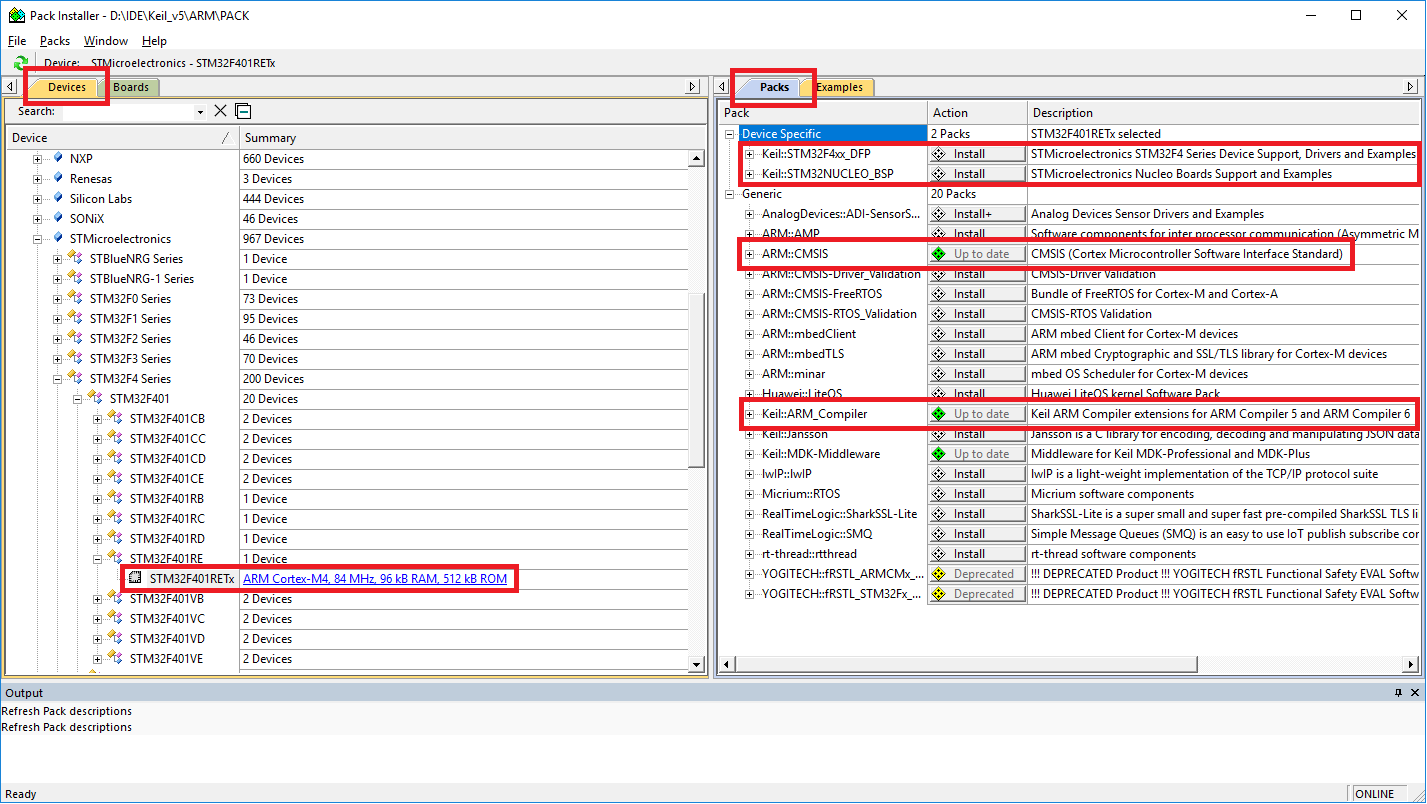


Рисунок 27 - Встановлення пакетів для обраного мікроконтролеру

Тепер цю програму можна закрити і в цій же теці відкриваємо програму «UV4.exe». З’явиться вікно зображене на рис. 28.

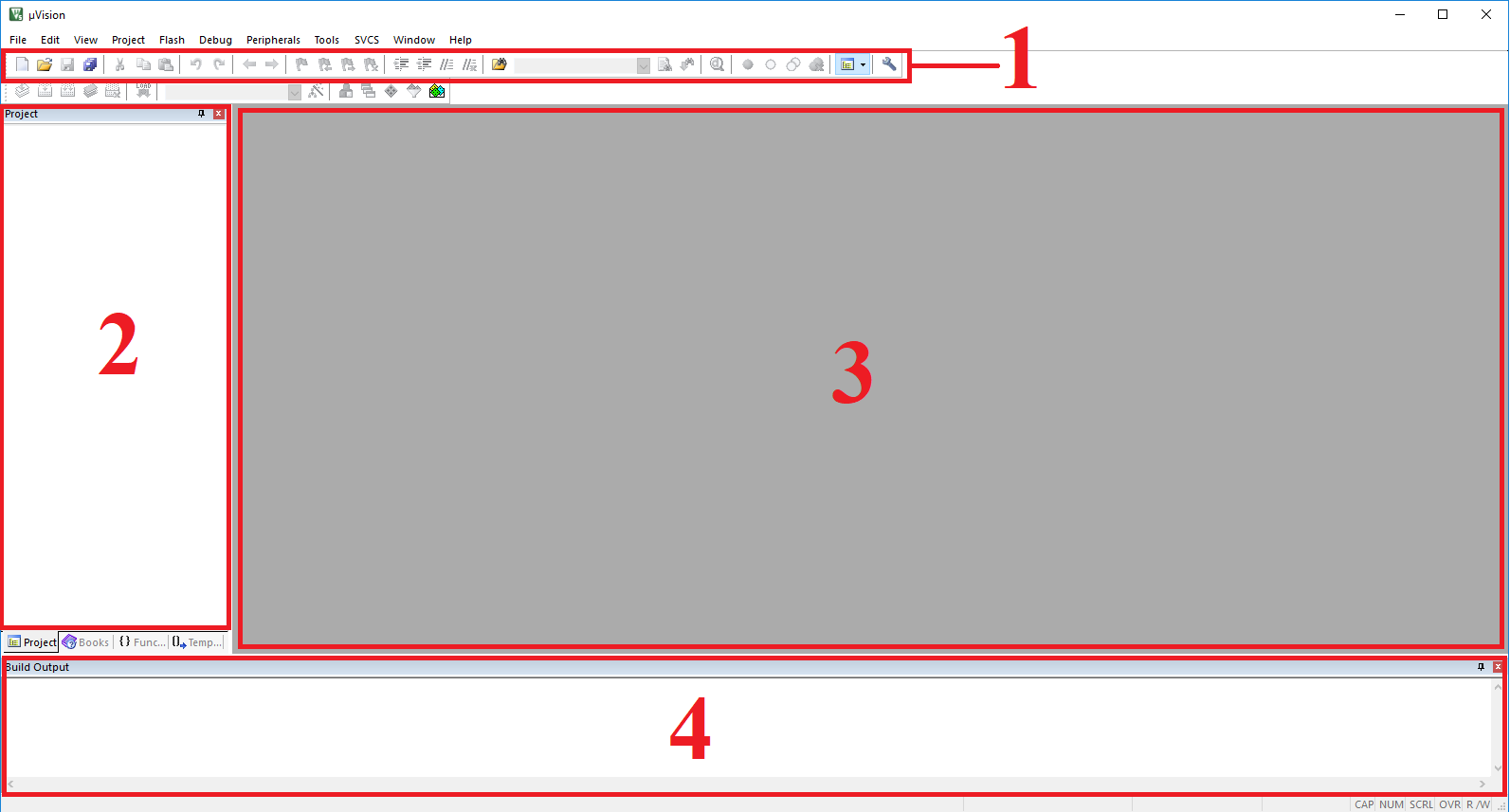


Рисунок 28- Робоча поверхня IDE

В області «1» знаходяться піктограми для швидкого доступу до таких функцій як новий файл, зберегти, відмінити останню дію і тд. В області «2» у вкладці «Project» буде відображатися ієрархія усіх тек і файлів, які входять у склад даного проекту. Текст самих файлів з кодом буде відображатися в області «3», а вся інформація про хід компіляції коду та можливі помилки в ньому будуть виводитися в область «4».

Для того щоб створити новий проект натисніть «Project -> New µVision Project» (рис. 29).

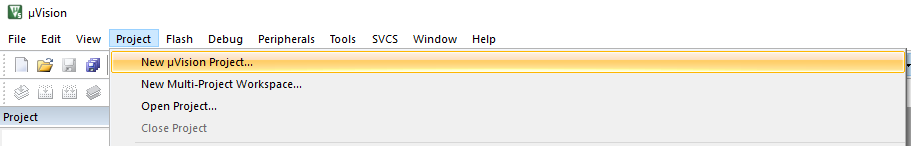


Рисунок 29 – Команда створення нового проекту

Оберіть місце розташування нового проекту та назву для нього (рис. 30). Краще для кожного нового проекту створювати окрему теку, що зробить подальшу роботу з проектом більш зручною.

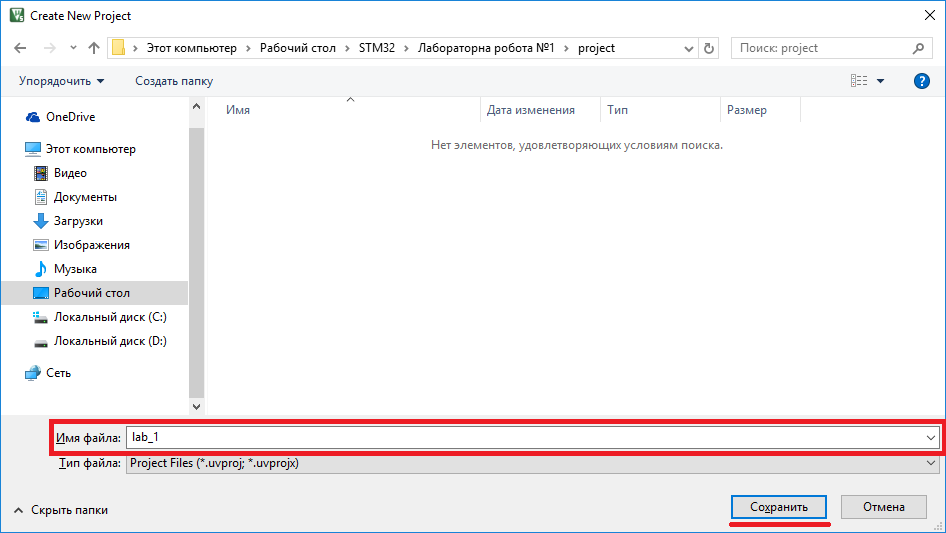


Рисунок 30 - Вибір місця розташування та назви нового проекту

Тепер необхідно вказати мікроконтролер, для якого створюється даний проект. У списку відображаються всі підтримувані мікроконтролери (рис. 31), якщо необхідного у списку немає, то треба повернутися в попередній розділ даної інструкції і встановити пакет, який включає його в себе.

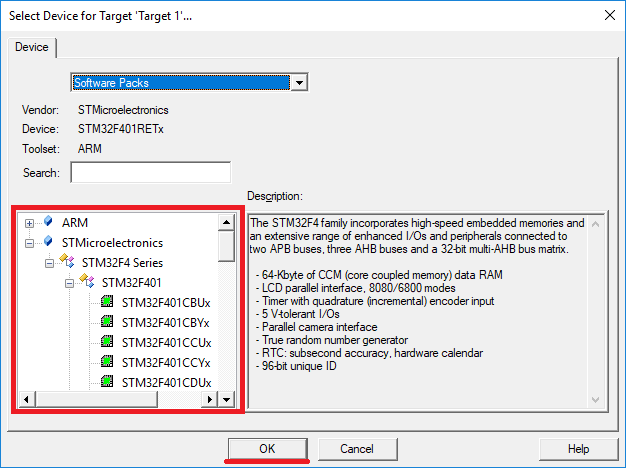


Рисунок 31- Вибір мікроконтролеру для цілі «Target 1»

Після цього перед нами з’явиться вікно (рис. 32) вибору бібліотек, що включають в себе файли з вже написаним, а інколи і збудованим, кодом, який полегшує використання компонентів мікроконтролера. Наприклад, замість написання великої кількості низькорівневих інструкцій для запалювання світлодіоду на платі можна буде викликати лише одну функцію.

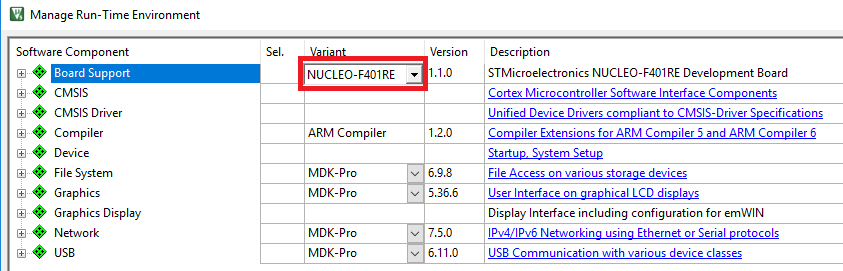


Рисунок 32 - Вибір бібліотек, що увійдуть до складу проекту

Параметри в цьому вікні можна міняти в будь-який момент відкривши його знову через «Project -> Manage -> Run-Time Environment…». Однак ці бібліотеки будуть розглянуті в наступних лабораторних роботах, а зараз ми лише вкажемо необхідну нам плату в якості підтримуваної у випадному меню, що зображено на рис. 32

Зараз в області «2» з’явилося ієрархічне дерево проекту (рис.33). Тут ми бачимо назву проекту «lab\_1», ціль проекту «Target 1», пусту групу файлів «Source Group 1».

Зверніть увагу нате, що це саме група файлів, вона видна лише в IDE, на жорсткому диску файли будуть в одній теці, також тут будуть відображатися всі бібліотеки, які підключені до проекту. Цілей в проекті може буди декілька. Вона включає в себе важливі настройки проекту, в тому числі вибір мікроконтролеру. Це потрібно для того щоб можна було зібрати програму з одних і тих самих файлів для різних мікроконтролерів. А групи необхідні для організації файлів у проекті, що забезпечує легкий та швидкий доступ до них. Назви цілей та груп можна змінити обравши їх у списку та натиснувши клавішу «F2». Змінемо ім’я групи «Source Group 1» на «src», що являється простим скороченням від англійського слова «source» - джерело, і будемо сюди додавати всі файли з вихідним кодом.

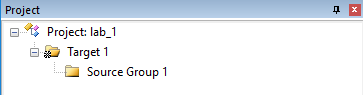


Рисунок 33 - Ієрархічне дерево проекту в області «2»

Налаштуємо «Target 1», натиснувши на нього правою кнопкою миші та обравши пункт «Options fot Target «Target 1»…. У цьому вікні можна налаштувати, наприклад, рівень оптимізації коду, задати попередньо задекларовані символи («#define»), шляхи включення заголовочних файлів для програм, що написані на мовах С/С++, але все це буде розглянуто в наступних роботах.

Зараз же перейдемо на вкладку «Debug» та оберемо «ST-Link Debugger» (рис. 34), для якого були встановлені драйвери. Як що цей пункт обрано, то налагодження програм буде відбуватися прямо на платі, тому її підключення обов’язкове. Якщо ж під рукою плати немає, тоді обираємо пункт «Use Simulator» у лівої половині вікна. Однак цим інструментом користуються досить нечасто. Також знімемо прапорець з «Run to main()», щоб при налагодженні бачити кожен «крок» мікроконтролеру, а не тільки починаючи з функції «main».

Після кожного прошивання мікроконтролеру треба буде натискати на ньому кнопку «Reset», інакше нова прошивка почне виконуватися з того місця, де зупинилася стара. Однак це можна не робити, якщо перейти до меню «Settings», правіше від обраного методу налагодження, потім на вкладку «Flash Download» і поставити прапорець біля «Reset and Run» (рис.35).

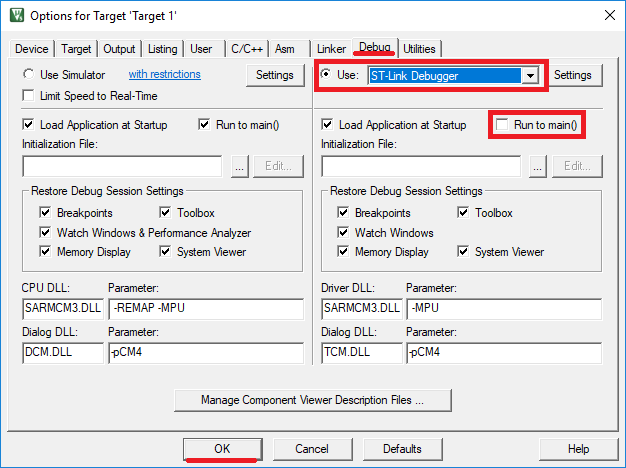


Рисунок 34 - Налаштування цілі «Target 1»

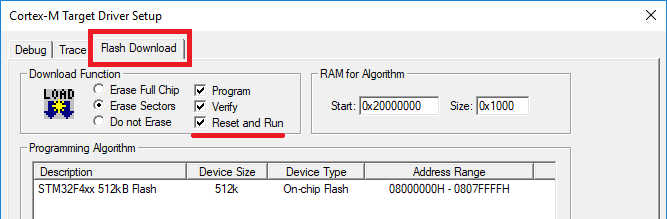


Рисунок 35 - Налагодження скидання мікроконтролеру після прошивання

Також налаштуємо текстовий редактор IDE, для цього переходимо в «Edit -> Configuration» зверху головного вікна або натискаємо піктограму гайкового ключа в області «1». Тут змінюємо кодування символів на «UTF-8», забороняємо вставляти пробіли замість відступів та робимо відступи в 4 символи для всіх типів файлів (рис. 36), хоча це вже на смак кожного. Не забуваємо підтвердити зміни натиснувши кнопку «ОК».

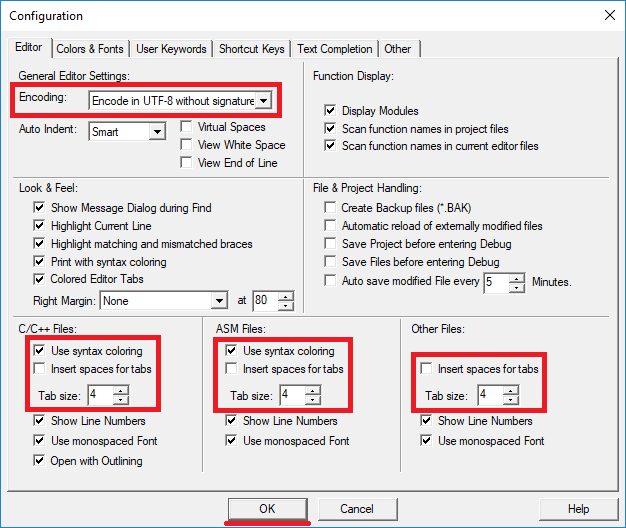


Рисунок 36 - Налаштування текстового редактору IDE

Щоб пересвідчитись у тому, що все виконано правильно створимо маленьку програму. Для цього натискаємо правою кнопкою миші по групі «src», до якою хочемо додати файл з кодом, у вікні структури проекту та обираємо пункт «Add New Item to Group ‘src’…» (рис. 37).

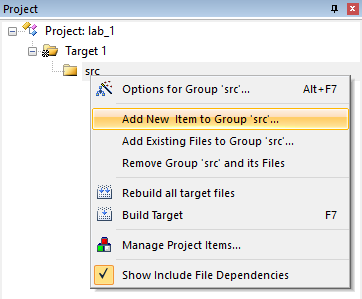


Рисунок 37 - Створення нового файлу в групі «src»

Обираємо тип файлу «Asm File» (рис. 38). За домовленістю між розробниками файли, що містять код на мові асемблеру мають розширення «.s», код на мові С – «.с», код на мові С++ – «.срр», а заголовочні файли для мов С/С++ – «.h», хоча також можна зустріти розширення «.hpp» для заголовочних файлів на мові С++. Вказуємо назву нового файлу та, за бажанням, місце його розташування.

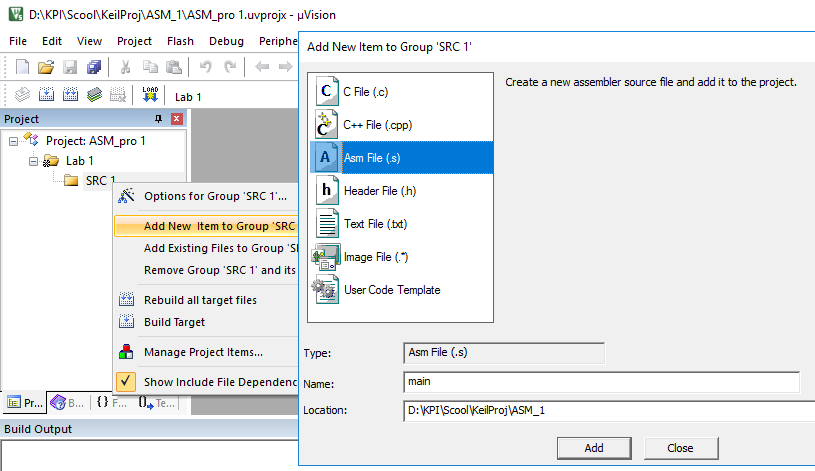


Рисунок 38 - Задання типу, назви та розташування нового файлу

У щойно створений файл запишіть код програми наведений на рис. 39 (див. також Додаток А.1 наприкінці цього файлу, або файл main.s у папці Source). Ця програма нічого не робить, але коректно запускається і може бути шаблоном для створення інших програм. Зверніть увагу на те, що всі інструкції починаються з відступу, а всі мітки з початку рядка. Це обов’язкове оформлення коду. Натиснувши кнопку «F7», або на відповідну піктограму зверху (рис. 40), почнемо побудову цілі, яка вибрана як основна в даний момент. В нашому випадку це «Lab 1». Якщо все було зроблено правильно, то в області «4» буде сповіщення про те, що помилки та попередження відсутні. Тепер підключаємо до комп’ютера плату і натискаємо кнопку «F8», або піктограму, і має початися процес завантаження збудованої цілі в пам’ять мікроконтролера. Результатом має бути текст у вікні виводу (область «4»), схожий з тим, що зображено на рис.41. Зараз мікроконтролер, прошитий цією програмою нічого корисного не робить, однак це тільки початок і найцікавіше попереду.

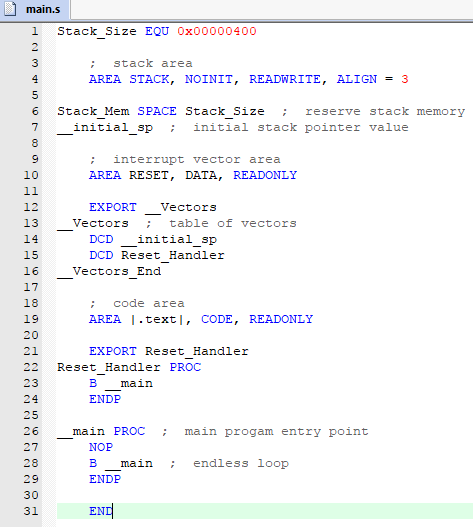


Рисунок 39 - Текст програми для мікроконтролеру

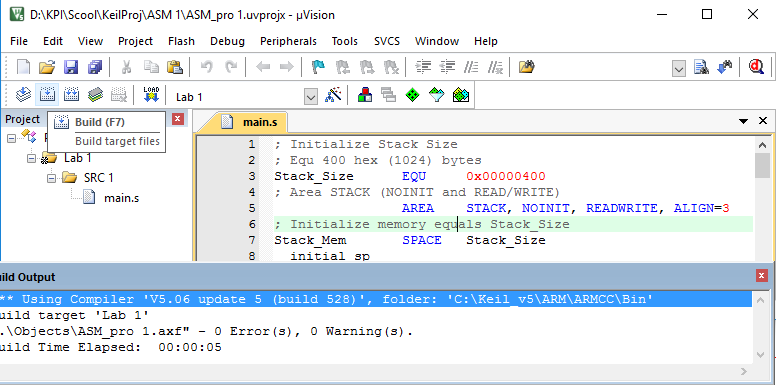


Рисунок 40 - Побудова проекту Lab 1

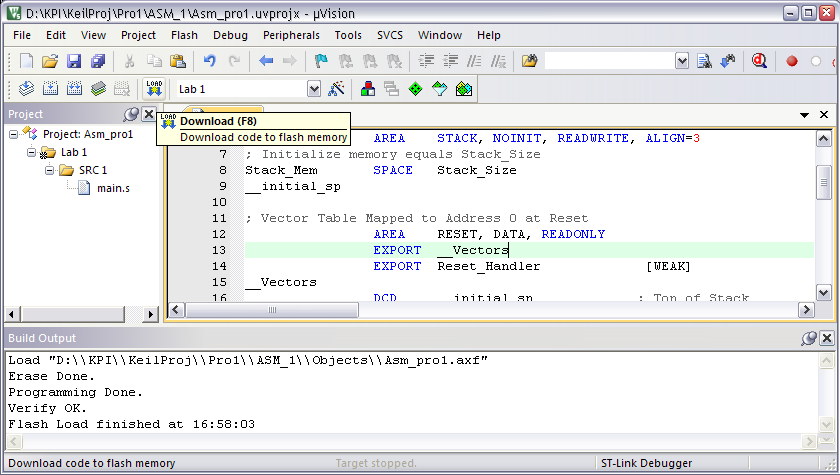


Рисунок 41 - Інформація про перебіг завантаження прошивки у мікроконтролер

На цьому вся процедура встановлення і налагодження середовища створення программ для мікроконтролерів STM32, а також створення проекту закінчено.

**Додаток А.1 Текст файлу main.s**

; Initialize Stack Size

; Equ 400 hex (1024) bytes

Stack\_Size EQU 0x00000400

; Area STACK (NOINIT and READ/WRITE)

AREA STACK, NOINIT, READWRITE, ALIGN=3

; Initialize memory equals Stack\_Size

Stack\_Mem SPACE Stack\_Size

\_\_initial\_sp

; Vector Table Mapped to Address 0 at Reset

AREA RESET, DATA, READONLY

EXPORT \_\_Vectors

EXPORT Reset\_Handler [WEAK]

\_\_Vectors

DCD \_\_initial\_sp ; Top of Stack

DCD Reset\_Handler ; Reset Handler

\_\_Vectors\_End

AREA |.text| , CODE, READONLY

Reset\_Handler PROC

B \_\_main

ENDP

\_\_main PROC

; place you initialization code here

\_\_mainloop

; place you application code here

B \_\_mainloop

ENDP

ALIGN

END

1. В п.2 надано прямий (більш короткий) шлях встановлення середовища Keil µVision5 безпосередньо з сайту ARM KEIL, але без вказівок з завантаження необхідної фірмовій документації. [↑](#footnote-ref-1)
2. Рекомендую указати вашу реальну email-адресу, на яку ви будете получати повідомлення про оновлення програми. [↑](#footnote-ref-2)
3. Для виконання завдань наступної лабораторної роботи №1, де вивчається процедура створення першого проекту на мові Assembler, два наступних пункти, що описані тут можна не виконувати. Але для програмування мікроконтролера на мові С/C++, чим ми й будемо займатися у більшості лабораторних робіт, потрібна бібліотека функцій, які підтримують роботу з ресурсами цього мікроконтролера. Тому можна завантажити їх у Keil заздалегідь.

   В інструкції до лабораторній роботі 1, де ви будете створювати і досліджувати ваш перший проект для STM32 ці дії будуть ще раз описані на прикладі плати STM32F4 Discovery з мікроконтролером STM32F407TVG. [↑](#footnote-ref-3)