**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"**

Інститут **КНІТ**

Кафедра **ПЗ**

**Лабораторна робота № 1**

**На тему:** *“* *Установка та дослідження середовища* **STM32CubeIDE** *та програмування плати STM32F4 Discovery”*

**З дисципліни:** *“Основи інтернету речей”*

**Львів – 2021**

**Тема роботи:** Установка та дослідження середовища **STM32CubeIDE** o та програмування плати STM32F4 Discovery.

**Мета роботи**: Установити середовище розробки, ознайомитися з основними компонентами середовища, створити проект та завантажити його на плату.

**Теоретичні відомості**

STM32 - сімейство 32-бітних мікроконтролерів виробництва STMicroelectronics. Чіпи STM32 групуються в серії, в рамках кожної з яких використовується один і той же 32-бітове ядро ARM, наприклад, Cortex-M7F, Cortex-M4F, Cortex-M3, Cortex-M0+ або Cortex-M0. Кожен мікроконтролер складається з ядра процесора, статичної RAM-пам'яті, флеш-пам'яті, отладочного і різних периферійних інтерфейсів.

Atollic TrueSTUDIO для STM32 - це комерційно розширений C / C ++ IDE на основі компонентів з відкритим кодом з потужними професійними розширеннями, функціями та утилітами.

Ключові характеристики плати:

Ядро: процесор Cortex-M4 з процесором ARM32 з FPU, адаптивний прискорювач у режимі реального часу (ART Accelerator), що дозволяє виконувати стан 0-очікування з флеш-пам’яті, частотою до 168 МГц, блоком захисту пам’яті, 210 DMIPS / 1,25 DMIPS / МГц (Dhrystone 2.1) та інструкції DSP

Пам’ять: 1 Мбайт флеш-пам'яті, 192 + 4 Кбайт оперативної пам’яті SRAM, включаючи 64 Кбайт оперативної пам'яті CCM (основна пам’ять)

Гнучкий контролер статичної пам’яті, що підтримує компактну пам’ять Flash, SRAM, PSRAM, NOR та NAND

**STM32CubeIDE**



STM32CubeIDE - це вдосконалена платформа розробки C/C++ з периферійною конфігурацією, генерацією коду, компіляцією коду та функціями налагодження для мікроконтролерів та мікропроцесорів STM32. Він базується на структурі Eclipse та наборі інструментів GCC для розробки та GDB для налагодження. Він дозволяє інтегрувати сотні існуючих плагінів, які доповнюють можливості Eclipse IDE.

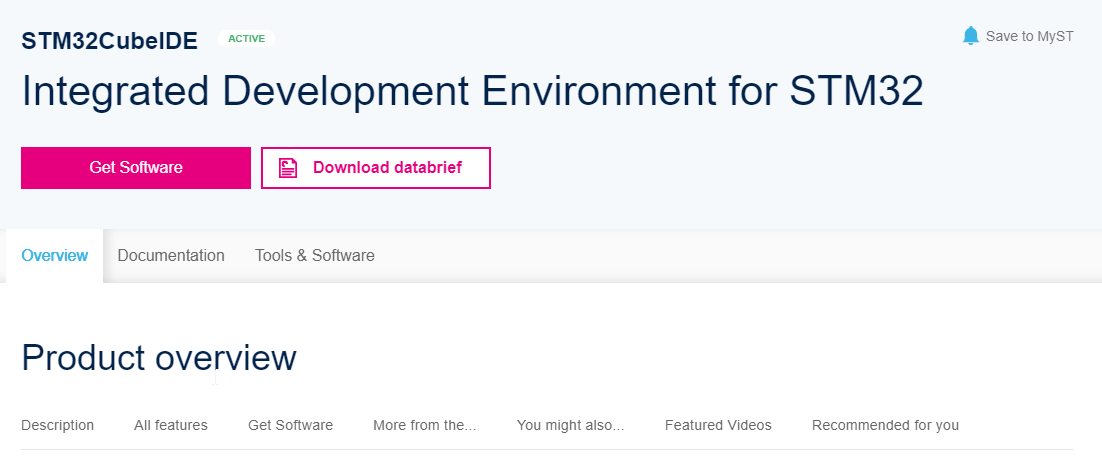
STM32CubeIDE інтегрує функції конфігурації та створення проектів STM32 від STM32CubeMX, щоб запропонувати універсальний інструмент та заощадити час встановлення та розробки. Після вибору порожнього MCU або MPU STM32, або попередньо налаштованого мікроконтролера або мікропроцесора з вибору плати або вибору прикладу, проект створюється та генерується код ініціалізації. У будь -який час під час розробки користувач може повернутися до ініціалізації та налаштування периферійних пристроїв або проміжного програмного забезпечення та відновити код ініціалізації без впливу на код користувача.

STM32CubeIDE містить аналізатори складання та стеку, які надають користувачеві корисну інформацію про стан проекту та вимоги до пам'яті.

STM32CubeIDE також включає стандартні та розширені функції налагодження, включаючи перегляди регістрів ядра процесора, пам'яті та периферійних реєстрів, а також функціональних змінних годинників, інтерфейсу перегляду послідовних проводів або аналізатора помилок.

**Установка середовища**

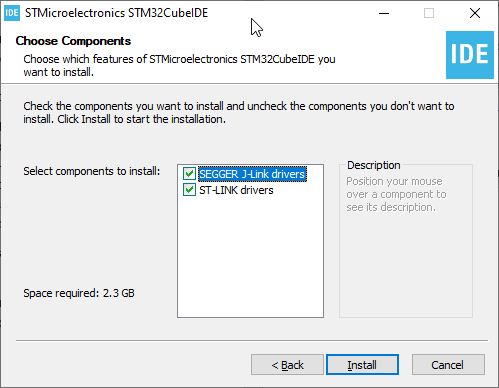
Середовище розробки STM32CubeIDE надається компанією STMicroelectronics. Завантажити середовище можна за посиланням: <https://www.st.com/en/development-tools/stm32cubeide.html> натиснувши на кнопку “Get software”.



Перед завантаженням сайт запросить вас авторизуватися, або зареєструватися. Компанія пропонує середовища розробки під такі операційні системи, як MacOS, Windows, Linux.

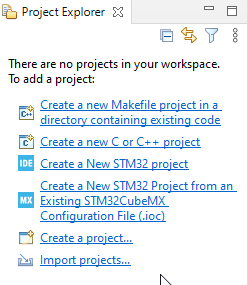


Процес установки середовища не потребує особливих налаштувань, окрім вибору драйверів програматора. Інсталятор пропонує Jlink та STlink. На борту STM32F4-DISC1 з коробки напаяний STLink, тому необхідно його відмітити обов’язково.

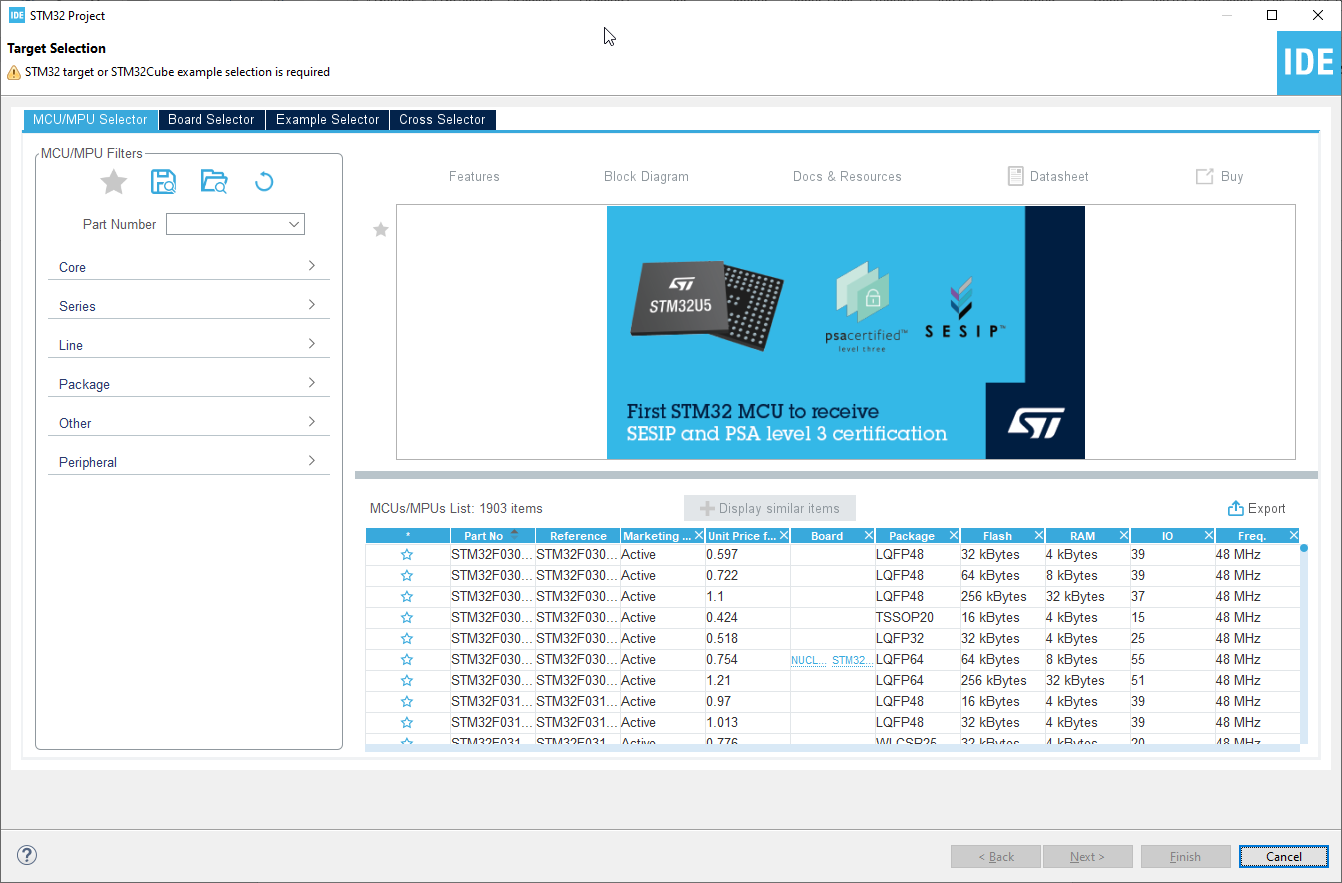


**Створення проекту для борди**

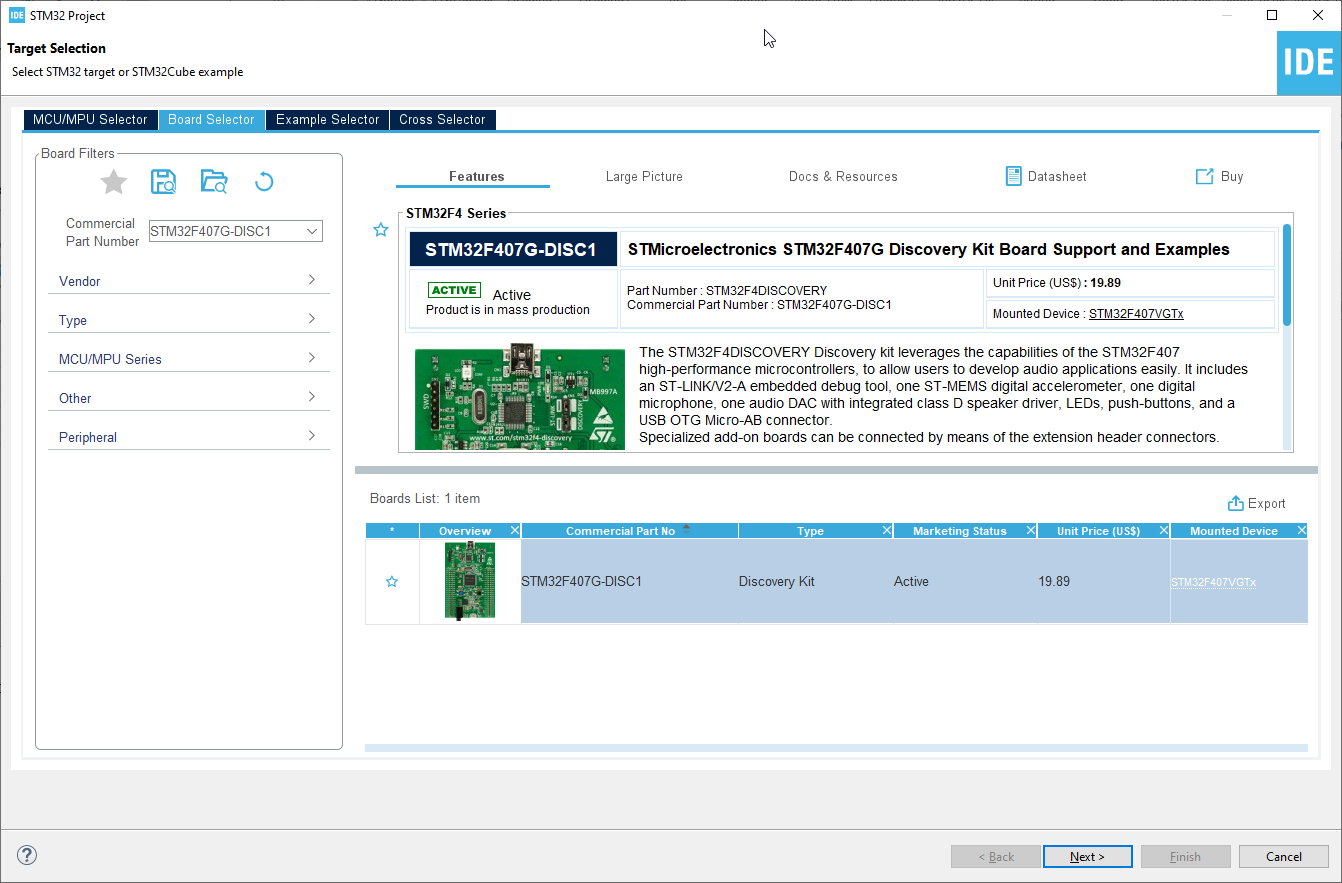
У відкритому середовищі необхідно створити проект. Для створення проекту необхідно обрати “Create New STM32 project”, або у меню “File->New-> STM32 project”. STM32 project це комплексний проект, який містить в собі файли бібліотеки для програмування мікроконтролерів STM32, файли конфігурації бордів та мікроконтролерів.



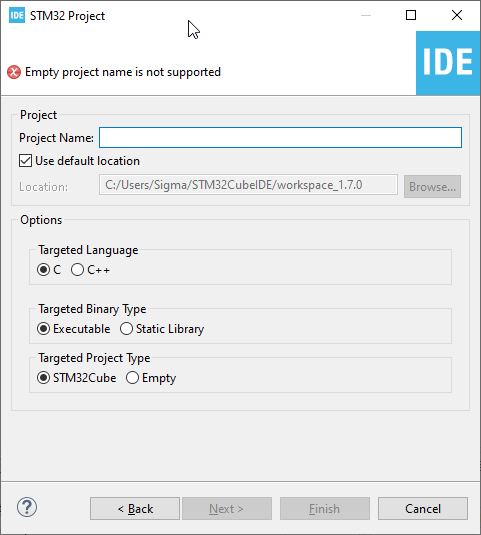
Після натиску ми попадаємо у вікно створення проекту STM32. Вікно виглядає так:



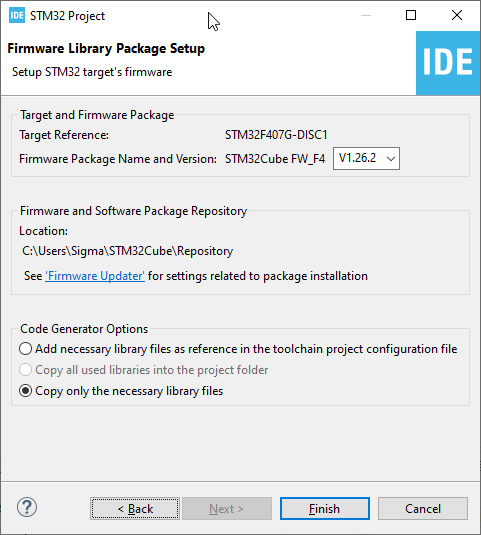
Обираємо вкладку “Board selector” і у полі “Part Number” вписуємо назву борди. У нашому випадку це “STM32F407G-DISC1”, та обираємо її в сусіднім вікні. Після натиску на плату кнопка “Next” стає активна.



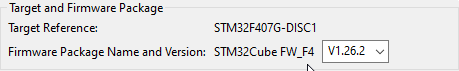
Після натиску на “Next” відкривається вікно налаштування проекту. В наступному вікні нам потрібно обрати мову програмування, назву проекту і т.д.

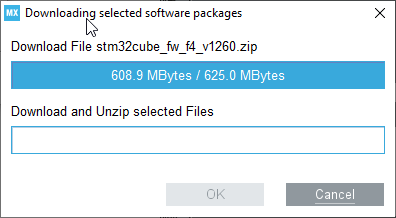


На наступному кроці налаштувань необхідно брати бібліотеку, яка буде використовується. Якщо це є перше створення проекту, тоді розпочнеться завантаження пакету апаратного забезпечення.

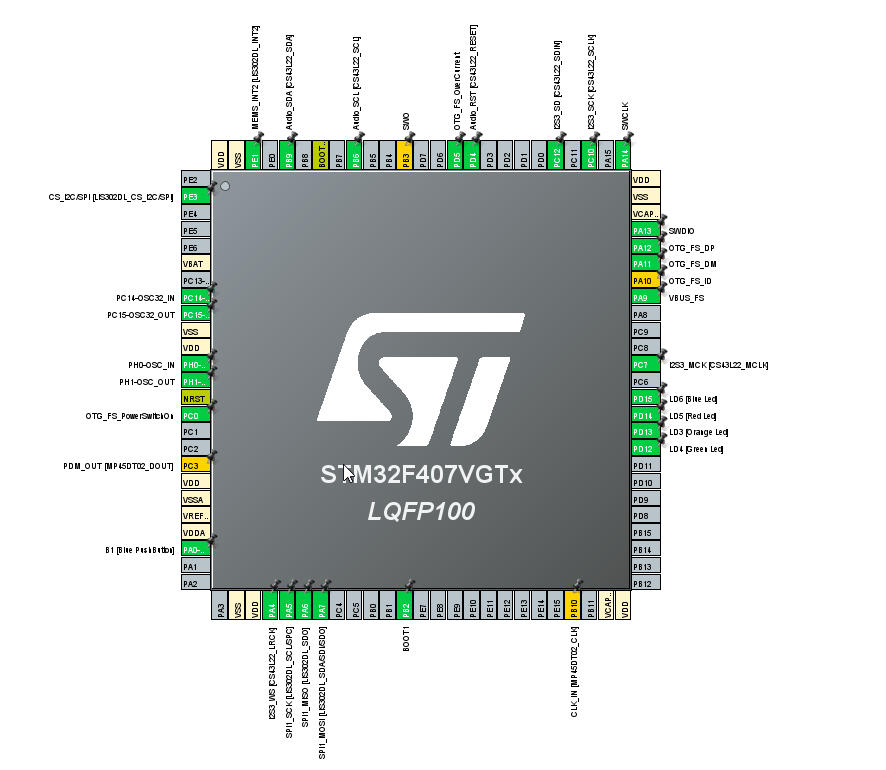


Також зауважте, якщо patch версія не є 0, тоді розпочнеться завантаження версії з patch 0, а потім оновлення patch.





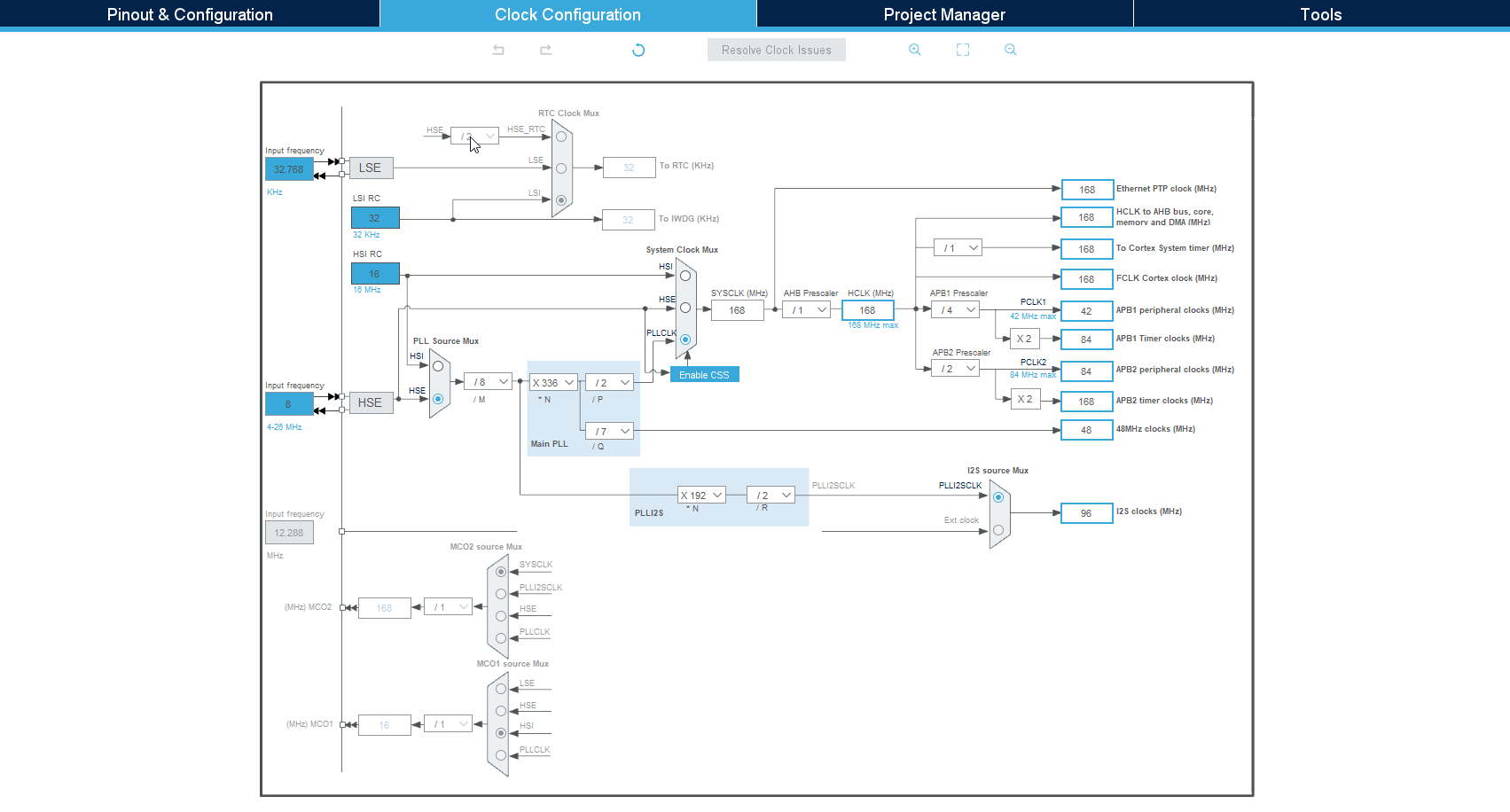
При створення проекту під борду ініціалізується уся наявна переферія на платі, тому схематичне зображення містить наступне зображення.



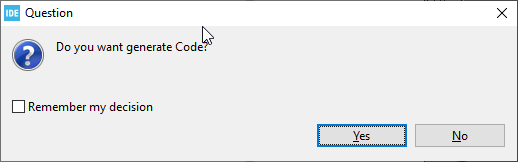
Що означають усі ці наступні кольори: сірий колір – нога мікроконтролера не має призначення, зелений – нога має призначення, жовтий – нога налаштована, проте має недоініціалізованість, чи конфлікт.

Оскільки ми вибрали конфігурацію борди, то середовище проініціалізувано нам піни під світлодіоди. Наступним кроком є налаштування тактування.

Аналогічно до попереднього випадку середовище само проініціалізувало плату на максимальну швидкість тактування. За допомогою зовнішнього кварцового резонатора.



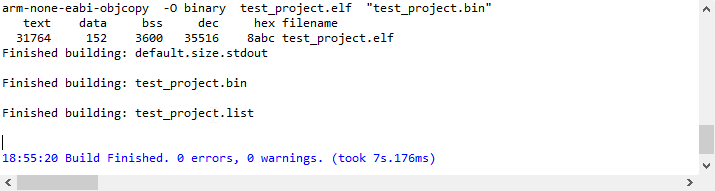
При збереження проекту середовище запитує нас, чи перегенерувати код відповідно до змін, які ми зробили у файлі конфігурацій. При натисненні на “Yes” середовище згенерує код відповідно до налаштувань.



Спробуємо зібрати проект натиснувши на “молоток” у верхній частині вікна.



Результат збору проекту відображатиметься у консольній області внизу екрана.

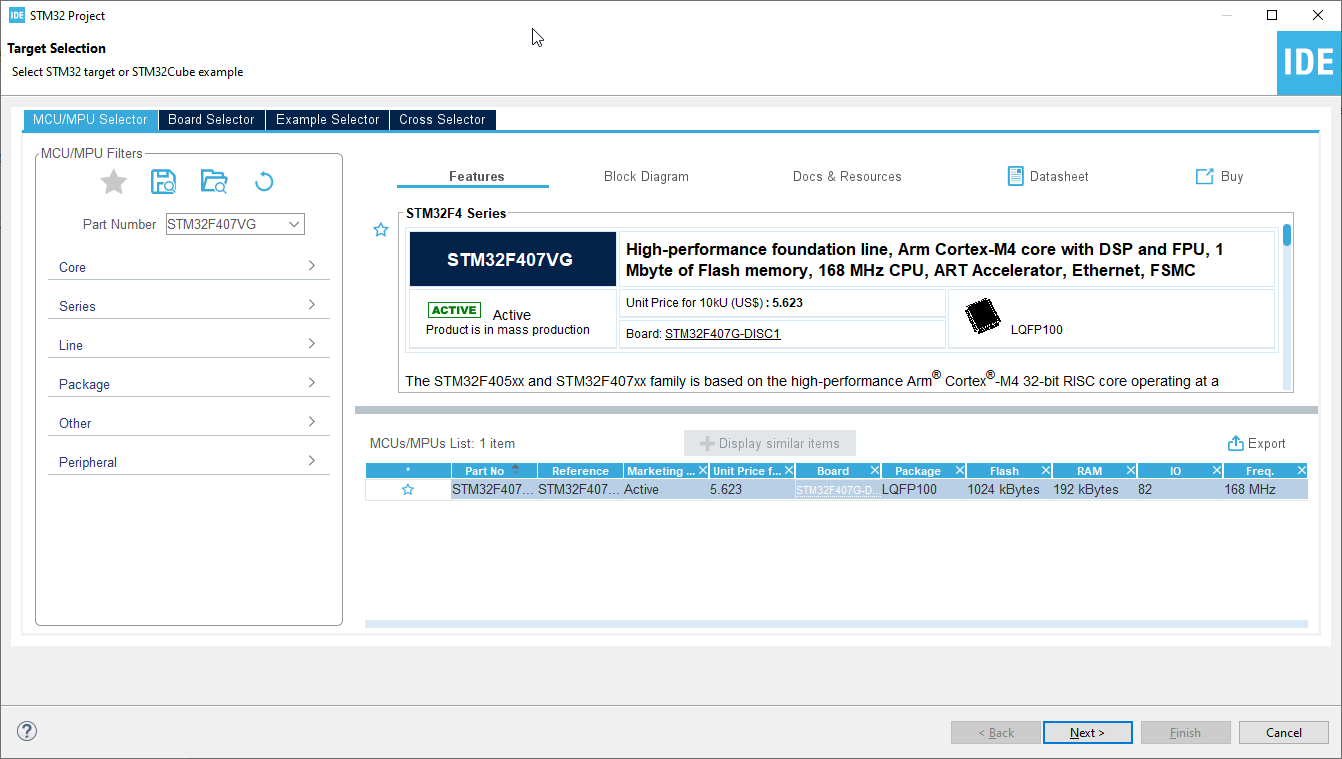


**Створення проекту під контролер**

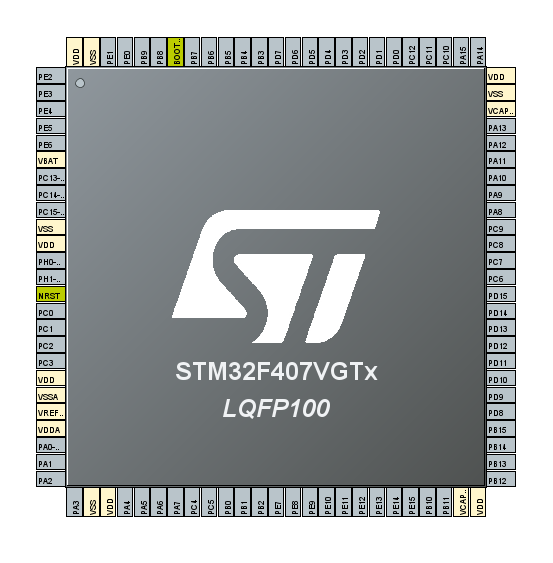
Згадаємо, чи коли небудь ми бачили пристрій в середині якого була б навчальна борда, як STM32F407G-DISC1. Звісно, що ні. Комерційні проекти використовують окремо мікроконтролер, до якого напаюють переферію, яку потрібно. В цьому випадку не потрібно створювати проект під цілу борду. Середовище містить можливість створення проекту під чисто мікроконтролер, в цьому випадку середовище не буде підказувати куди що підключено, а усе надається для конфігурацій розробником.

Створення є аналогічним до попереднього, проте у вікні створення проекту необхідно обрати не “Board selection”, а “MCU selection”.

В полі “Part number” необхідно вказати назву контролера. В нашому випадку на борді “STM32F407G-DISC1” встановлений мікроконтролер STM32F407VG.

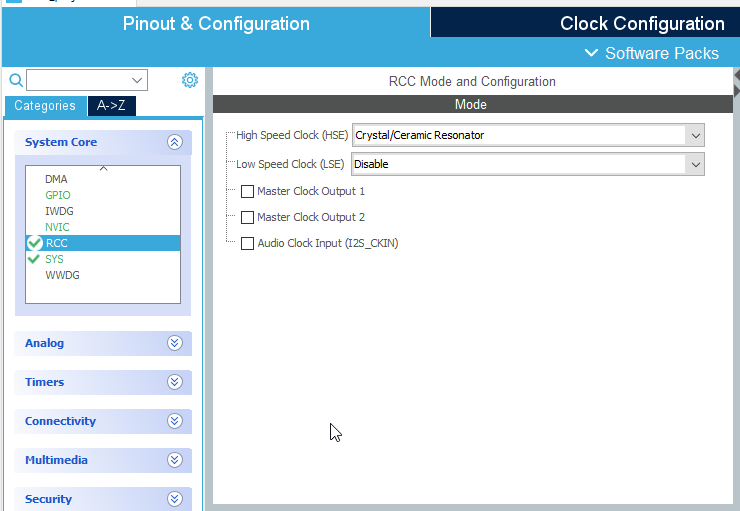


Наступні кроки налаштування є аналогічні до налаштування проєкту для борди. Проте в цьому випадку не будуть ініціалізовані ноги мікроконтролера.



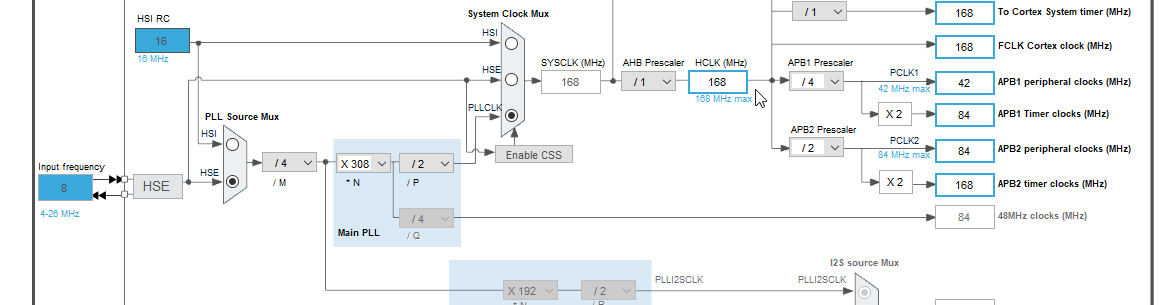
Для роботи програми мікроконтролера потрібне тактування. Можливі 2 види тактування: внутрішнє (HSI) і зовнішнє(HSE). Внутрішнє тактування можливе за частоти 16Mh, зовнішнє здійснюється за допомогою зовнішнього кварцового резонатора. На борді, яку ми використовуємо він присутній і його тактова частота є 8Mh. Питання є в тому, яка різниця між ними, якщо їх частоти є рівними. А різниця в тому, що чим менше роботи виконує мікроконтролер – тим довший ресурс його роботи.

Налаштування тактування виконується на вкладці “Clock configuration”. За допомогою різних множників і подільників тактова частота може збільшитися до 168Mh. Для ввімкнення зовнішнього тактування необхідно включити RCC High Speed Clock на зовнішній резонатор.

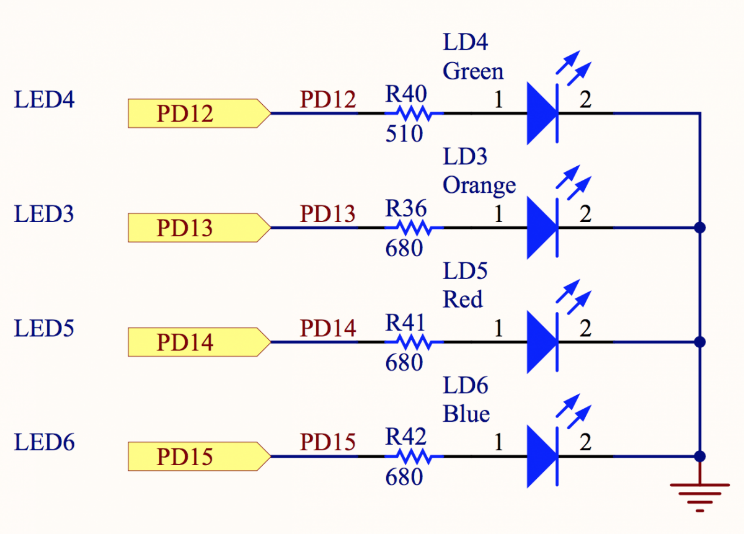
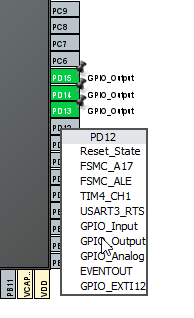


Після цього на вкладні налаштування тактування буде доступний HSE. Проте значення необхідно змінити з 25 на 8, оскільки в нас на платі є 8Mh кварц.

Основним полем тактування є HCLK, який визначає частоту тактування мікроконтролера. Середовище підтримує автоматичний підбір параметрів тактування. Необхідно просто ввести потрібні значення тактування в поля і обрати ресурс тактування. Наприклад, якщо ввести в поле HSE 8Mh, обрати його, як основне тактування і ввести в поле HCLK і натиснути Enter – середовище налаштує тактування на максимальну частоту за допомогою зовнішнього тактування.



Також необхідно налаштувати піни до яких підключені світлодіоди на вихід.



**Написання коду**

Середовище забезпечує автоматичне генерування коду ініціалізації, тому не потрібно писати окрему ініціалізацію. Середовище само її написало у функціях HAL\_init() та MX\_GPIO\_init().

Згенерований код підтримує пере генерування коду, тому весь код написаний розробником потрібно писати у областях:

/\* USER CODE BEGIN 0 \*/

/\* USER CODE END 0 \*/

Для затримки програми необхідно використовувати функцію HAL\_Delay(x), яка блокує роботу програми на x мілісекунд.

Для роботи з ножками мікроконтролера є функції:

HAL\_GPIO\_WritePin(GPIOx, GPIO\_Pin, PinState) – для встановлення певного стану піна

HAL\_GPIO\_TogglePin(GPIOx, GPIO\_Pin) – для зміни стану піна

GPIOx – це об’єкти портів мікроконтролера.

GPIO\_Pin – це дефениція, яка вказує на номер піна до якого застосовується функція.

PinState – це стан піна. Він може бути 1 для активного і 0 для пасивного

Наприклад, перший світлодіод в нас підключений до порта D, піна 12, тому щоб встановити на нього активний стан необхідно викликати наступну функцію:

HAL\_GPIO\_WritePin(GPIOD, GPIO\_PIN\_12, 1);

Програма мікроконтролера розпочинає свою роботу із функції main. Якщо зауважити, то функція містить безкінечний цикл, тому що програма не повинна завершуватися.

Далі наведено приклад програми мигання усіма світлодіодами щосекунди (коментарі середовища є опущені).

**int** **main**(**void**)

{

HAL\_Init();

SystemClock\_Config();

MX\_GPIO\_Init();

**while** (1)

{

HAL\_GPIO\_TogglePin(GPIOD, GPIO\_PIN\_12);

HAL\_GPIO\_TogglePin(GPIOD, GPIO\_PIN\_13);

HAL\_GPIO\_TogglePin(GPIOD, GPIO\_PIN\_14);

HAL\_GPIO\_TogglePin(GPIOD, GPIO\_PIN\_15);

HAL\_Delay(1000);

}

}

Для більшого ознайомлення з функціями бібліотеки існує документація, яку можна переглянути за посиланням: <https://www.st.com/resource/en/user_manual/dm00105879-description-of-stm32f4-hal-and-ll-drivers-stmicroelectronics.pdf>

**Завдання на лабораторну**

1. Встановити середовище **STM32CubeIDE**
2. Створити проект згідно індивідуального завдання для борду STM32F4 Discovery.
3. Створити проект згідно індивідуального завдання для мікроконтролера STM32F407VG.
4. Виконати індивідуальне завдання.

**Індивідуальні завдання**

Індивідуальне завдання обирається згідно порядкового номеру в списку підгрупи. У випадку коли порядковий номер у списку перевищує 12 студент обирає варіант за формулою **«порядковий номер» -12**

1.Засвічення червоного та синього світлодіодів, після цього – зеленого та жовтого з інтервалом у 1 с, циклічно

2. Вмикання/вимикання червоного та синього світлодіодів з інтервалом у 0.5 с циклічно, але не більше 20 ітерацій.

3. Вмикання по черзі червоного та синього світлодіодів, через 1 с – зеленого та жовтого світлодіоду одночасно.

4. Вмикання/вимикання червоного світлодіоду з інтервалом, що з кожною ітерацією збільшується на 0.5 с.

5. Вмикання/вимикання червоного та жовтого світлодіодів з інтервалом, що з кожною ітерацією збільшується на 0.5 с.

6. Почергове циклічне увімкнення жовтого, зеленого, синього та червоного світлодіодів.

7. Почергове циклічне увімкнення зеленого, жовтого, синього та червоного світлодіодів.

8. Почергове циклічне увімкнення синього, жовтого, зеленого, та червоного світлодіодів.

9. Почергове циклічне увімкнення червоного, жовтого, зеленого та синього світлодіодів.

10. Почергове увімкнення/вимкнення жовтого, зеленого, синього та червоного світлодіодів. Перед засвіченням кожного наступного світлодіоду, попередній слід вимкнути.

11. Почергове циклічне увімкнення зеленого, жовтого, синього та червоного світлодіодів. Перед засвіченням кожного наступного світлодіоду, попередній слід вимкнути.

12. Почергове циклічне увімкнення червоного, жовтого, зеленого та синього світлодіодів. Перед засвіченням кожного наступного світлодіоду, попередній слід вимкнути.