**ЛАБОРАТОРНА РОБОТА №1**

***Ознайомлення з програмою Proteus. Бібліотека HAL***

***Налаштування периферії за допомогою Cube MX. Апаратний ШІМ.***

Мета роботи: Симуляція роботи мікропроцесора в програмі Proteus. Генерація коду за допомогою програми Cube MX. Реалізація ШІМ. Створення і компіляція робочої програми на мові програмування С++.

***Теоретичні відомості***

*Proteus*

*Програмний пакет Proteus VSM дозволяє зібрати схему будь-якого електронного пристрою і симулювати його роботу, виявляючи помилки, допущені на стадії проектування . Програма складається з двох модулів. ISIS - редактор електронних схем з наступною імітацією їх роботи. ARES - редактор друкованих плат, оснащений вбудованим редактором бібліотек і автоматичною системою розміщення компонентів на платі. Крім цього ARES може створити тривимірну модель друкованої плати.*

*HAL:*

Бібліотека HAL - шар апаратної абстракції (англ. Hardware Abstraction Layer, HAL). Її основні завдання - скоротити час розробки і дозволити писати код на будь-який мікроконтролер сімейства STM32 (F0, F1 і т.д.).

*CUBE MX*

STM32CubeMx - програмний продукт, що дозволяє легко і невимушено за допомогою досить зрозумілого графічного інтерфейсу заново відрегулювати будь-якої наявної на борту мікроконтролера периферії. ST мають дуже різноманітну лінійку мікроконтролерів. Відповідно, постає питання про якомусь єдиному наборі бібліотек і єдиному інструменті для налаштування і конфігурації всього цього різноманіття. Ось для вирішення цих цілей і був випущений STM32CubeMx. Працює все це так - створюємо проект, вибираємо мікроконтролер і нам відразу ж пропонується велика схема з усіма виходами обраного нами контролера. Натискаючи на виходи і заходячи в різноманітні меню, ми легко налаштовуємо як периферію, так і режими роботи кожного конкретного виходу.

*ШІМ*

ШІМ або PWM (широтно-імпульсна модуляція, англ. pulse-width modulation) - це спосіб управління подачею потужності до навантаження. Управління полягає в зміні тривалості імпульсу при постійній частоті проходження імпульсів. Широтно-імпульсна модуляція буває аналоговою і цифровою

Застосування широтно-імпульсної модуляції дозволяє підвищити ККД електричних перетворювачів, особливо це стосується імпульсних перетворювачів, які сьогодні становлять основу вторинних джерел живлення різних електронних апаратів.

Широтно-імпульсна модуляція дозволяє регулювати яскравість підсвічування рідкокристалічних дисплеїв стільникових телефонів, смартфонів, ноутбуків. ШІМ реалізована в зварювальних апаратах, в автомобільних інверторах, в зарядних пристроях і т. д.

***Порядок виконання роботи***

*Завдання виконувати за допомогою бібліотеки HAL, програми STM32CubeMX. fбо будь якими іншими інструментами). Але демонстрація повинна бути в Proteus*

*Д*ля виконання поставленої задачі потрібно:

1. Ознайомитись з теоретичними відомостями по виконуваній роботі;
2. Ознайомитись з схемою мікроконтролера та всіма документами, які завантажені на гугл диск
3. Завантажити та встановити  *програми STM32CubeMX, Proteus, Cube IDE*
4. Cтворити новий проект в програмі *STM32CubeMX та згенерувати код*
5. Написати робочу програму на мові програмування С++ згідно свого завдання
6. Знайти папку своєї групи на гугл диску в папці з лабjраторними, якщо такої немає, створити таку
7. Створити папку зі своїм прізвищем в папці своєї групи
8. Створити папку «Лабораторна 1» та завантажте туди файли з вашою програмою та файл проекту Proteus.

***Завдання***

N – номер студента у списку, якщо номер > 15, то N = N – 15 (наприклад, якщо номер студента 16, то N = 16 –

15 = 1; 17 – 2; 18 – 3…)

N1 = N – 1

T1 = N / 10

Щоб отримати 60

1. Налаштувати тактову частоту мікроконтроллера (HCLK) на N mhz
2. Підключити 10 світлодіодів та 2 кнопки до будь яких вільних пінів.
3. Якщо перша кнопка натиснута, то «активні» перші 5 світлодіодів, якщо ні то з 6-10

Щоб отримати 85

1. Виконати всі завдання з «Щоб отримати 60»
2. Змінювати сигнали з 0 на 1 та з 1 на 0 на «активних» світлодіодах, за допомогою будь якого таймеру з швидкістю T1 разів за секунду.

Щоб отримати 100

1. Виконати всі завдання з «Щоб отримати 85»
2. Підключити віртуальний осцилограф та перевірити правильність виконання попереднього завдання
3. Налаштувати апаратний ШІМ на будь якому таймері та каналі використовуючи звичайний та комплементарний виводи.

**ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ**

1. Proteus.
2. *STM32CubeMX*
3. *CUBE IDE*