**ЛАБОРАТОРНА РОБОТА №2**

***Аналогово-цифрове перетворення. UART.***

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

***Теоретичні відомості***

*АЦП*

*Аналогово-цифровий перетворювач (АЦП) - один з найбільш важливих електронних компонентів в вимірювальному і тестовому обладнанні. АЦП перетворює напругу (аналоговий сигнал) в код, над яким мікропроцесор і програмне забезпечення виконують певні дії. Навіть якщо Ви працюєте тільки з цифровими сигналами, швидше за все Ви використовуєте АЦП в складі осцилографа, щоб дізнатися їх аналогові характеристики.*

*Існує кілька основних типів архітектури АЦП, хоча в межах кожного типу існує також безліч варіацій. Різні типи вимірювального обладнання використовують різні типи АЦП. Наприклад, в цифровому осцилографі використовується висока частота дискретизації, але не потрібно високий дозвіл. В цифрові мультиметри потрібно більшу роздільну здатність, але можна пожертвувати швидкістю вимірювання. Системи збору даних загального призначення за швидкістю дискретизації і роздільної здатності зазвичай займають місце між осцилографами і цифровий мультиметр. В обладнанні такого типу використовуються АЦП послідовного наближення або сигма-дельта АЦП. Існують також паралельні АЦП для додатків, що вимагають швидкісної обробки аналогових сигналів, і інтегрують АЦП з високими дозволом .*

*UART:*

UART (англ. universal asynchronous receiver/transmitter — універсальний асинхронний приймач/передавач) — тип асинхронного приймача-передавача, компонентів комп'ютерів та периферійних пристроїв, що передає дані між паралельною та послідовною формами. UART звичайно використовується спільно з іншими комунікаційними стандартами, такими як EIA RS-232. Під час асинхроної передачі UART телетайпного типу посилає стартовий біт, потім від п'яти до восьми бітів даних, перший — найменш значимий, потім опціональний біт парності, і потім один, півтора чи два стопових біти. Стартовий біт надсилається в зворотній полярності до звичайного незайнятого стану ліній зв'язку. Стоповий біт відповідає незайнятому стану лінії і забезпечує паузу перед наступною порцією даних. Це зветься асинхронною старт-стоповою передачею. В механічних телетайпах стоповий біт часто був розтягнутим вдвічі, щоб дати можливість механізму надрукувати символ. Розтягнутий стоповий біт також допомагав при ресинхронізації. Біт парності перевіряє кількість одиниць між стартовим і стоповим бітами або парним та непарним, або ж цей біт може бути відсутнім. Непарна перевірка надійніша, бо вона може засвідчити, що принаймні одна одиниця передалася, а це дозволяє багатьом UART пересинхронізуватися. В синхронній передачі частота тактового генератора відновлюється окремо з потоку даних і старт-стопові біти не використовуються. Це покращує ефективність каналу зв'язку для надійних ліній, також надсилається більше корисних даних. Асинхронна передача не посилає нічого, коли нема що передавати. Натомість синхронний інтерфейс має завжди посилати якісь дані, щоб підтримувати синхронізацію між передавачем і приймачем. Як заповнювач порожнечі часто використовується ASCII-символ «SYN», це робиться автоматично передавачем.

USART чипи мають синхроний та асинхроний режими.

***Порядок виконання роботи***

*Завдання виконувати за допомогою бібліотеки HAL, програми STM32CubeMX.*

*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 лабораторними, якщо такої немає, створити таку
7. Створити папку зі своїм прізвищем в папці своєї групи
8. Створити папку «Лабораторна 2» та завантажте туди файли з вашою програмою та файл проекту Proteus.

***Завдання***

Щоб отримати 60

1. Налаштувати АЦП
2. Налаштувати UART
3. Виводити в UART значення АЦП

Щоб отримати 85

1. Виконати всі завдання з «Щоб отримати 60»
2. Налаштувати ще один UART та реалізувати базові команди такі як: увімкнути / вимкнути світлодіод

Щоб отримати 100

1. Виконати всі завдання з «Щоб отримати 100»
2. Реалізувати команди по налаштуванні таймерів на певний час в секундах, а також ШІМ.

**ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ**

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