

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЧЕРНІВЕЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ЮРІЯ ФЕДЬКОВИЧА
Відокремлений структурний підрозділ «Фаховий коледж Чернівецького
національного університету імені Юрія Федьковича»

Природниче відділення
Циклова комісія комп'ютерної інженерії

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітньо-професійного ступеня «фаховий молодший бакалавр»
зі спеціальності 123 Комп'ютерна інженерія
підготовки за освітньо-професійною програмою «Комп'ютерна інженерія»
на тему: **«Система моніторингу та відображення ціни крипто
валютних акцій за допомогою ESP8266.»**

Виконав (ла):

студент (ка) 4-го курсу Мойсюк Михайло Вікторович

(прізвище, ім'я та по батькові)

_____ (підпис)

Керівник:

Букурос О.В.

(науковий ступінь, вчене звання, прізвище та ініціали)

_____ (підпис)

Рецензент:

Тащук О.Ю.

(науковий ступінь, вчене звання, прізвище та ініціали)

_____ (підпис)

До захисту допущено:

Протокол засідання циклової комісії № _

від „__” _____ 2024 р.

Голова циклової комісії _____ Олександр ТАЩУК

Чернівці, 2024

Завдання та календарний план роботи

Міністерство освіти і науки України
Чернівецький національний університету імені Юрія Федьковича»
Відокремлений структурний підрозділ «Фаховий коледж
Чернівецького національного університету імені Юрія Федьковича»

Затверджую
Голова циклової комісії,
Олександр ТАЩУК.

«___» _____ 2023 р.

Завдання

На кваліфікаційну роботу	Мойсюк Михайло Вікторович
Тема кваліфікаційної роботи	Система моніторингу та відображення ціни крипто валютних акцій за допомогою ESP32
Постановка завдання в короткій формі	Розробка Система моніторингу та відображення ціни крипто валютних акцій за допомогою ESP8266 та IoT технологій
Вихідні дані (2-3 адреси сайтів, матеріали яких рекомендує керівник кваліфікаційної роботи)	

Календарний план роботи

Дата отримання версії звіту керівником	Етап виконання роботи	Виконання (зазначає керівник)
10.10.23	Отримання завдання до кваліфікаційної роботи.	
08.11.23	Огляд літератури, присвяченої розробці систем відстеження криптовалют на основі IoT.	
21.12.23	Аналіз існуючих конструкторських рішень. Оформлення розділу 1.	
27.01.24	Вибір компонентної бази, узгодження модулів системи.	
29.02.24	Написання теоретичного матеріалу по використовуваних модулях.	
30.03.24	Проектування та розробка апаратної частини, підключення Оформлення розділу 2.	
28.04.24	Проектування та розробка програмної частини, підключення. Оформлення розділу 3.	

17.05.24	Оформлення та редагування кваліфікаційної роботи.	
31.05.24	Подання роботи на перевірку Інтернет-сервісом Unichesk.	
03.06.24	Подання роботи на рецензію	
06.06.24	Представлення роботи на засіданні Циклової комісії	
<i>за графіком</i>	Захист кваліфікаційної роботи	

Дата видачі завдання _____

Студент

Михайло МОЙСЮК

Керівник

Олеся БУКУРОС

АНОТАЦІЯ

Мойсюк Михайло Вікторович Система моніторингу та відображення ціни крипто валютних акцій за допомогою ESP32 Кваліфікаційна робота.

В роботі виконано моделювання та розробку Системи моніторингу та відображення ціни крипто валютних акцій за допомогою ESP8266 та IoT технологій

Актуальність роботи пояснюється тим, що Криптовалюти стали невід'ємною частиною сучасної фінансової системи, їхня популярність зростає завдяки децентралізації, анонімності та можливостям інвестування. Однак, нестабільність ринку криптовалют вимагає постійного моніторингу їх цін. Система моніторингу цін криптовалют у реальному часі за допомогою ESP8266 надає доступне та ефективне рішення для відстеження цих змін, що є важливим для інвесторів та трейдерів.

Робота складається з 3 розділів, в яких проаналізовано роботу, змодельовано, описано та виконано розробку Система моніторингу та відображення ціни крипто валютних акцій за допомогою ESP8266.

Кваліфікаційна робота містить 60 сторінок, 4 рисунки та 6 посилань на літературні джерела.

Ключові слова: *Інтернет речей IoT, система моніторингу ціни крипто валютних акцій, криптовалюта, Bitcoin, Ethereum, Litecoin, OLED-екран, ESP8266, хмарне сховище.*

ЗМІСТ

СПИСОК ТЕРМІНІВ, СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ.....	6
ВСТУП.....	7
1. ПОСТАНОВКА ЗАДАЧІ.....	8
1.1 Еволюція криптовалют та їх ринок.....	8
1.2 Технічні аспекти моніторингу криптовалют.....	9
1.3 Використання IoT у фінансових технологіях.....	14
2. ПРОЕКТУВАННЯ АПАРАТНОЇ СКЛАДОВОЇ СИСТЕМИ МОНІТОРИНГУ ТА ВІДОБРАЖЕННЯ ЦІНИ КРИПТО ВАЛЮТ.	16
2.1 Вибір основних компонент та їх характеристики.....	26
2.2. Принципи роботи з API для отримання цін криптовалют (CoinMarketCap, Coindesk).....	20
2.3. Налаштування апаратної частини: схеми підключення.....	22
3. ПРОЕКТУВАННЯ ПРОГРАМНОЇ СКЛАДОВОЇ СИСТЕМИ МОНІТОРИНГУ ЗДОРОВ'Я.....	24
3.1 Структура програмного забезпечення та хмарного сховища для системи IoT.....	24
3.2 Написання та відладка програмного коду.....	26
ВИСНОВКИ.....	31
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	32
ДОДАТКИ.....	33
Додаток А. Лістинг програми.....	33
Додаток Б. Архітектура систем моніторингу криптовалют.....	54
Додаток В. Мікроконтролер ESP8266.....	55
Додаток Г. Макетна схема Трекера криптовалюти.....	51
Додаток Д. Демонстраційні зображення роботи сисеми	52

СПИСОК ТЕРМІНІВ, СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ

IoT – Інтернет речей.

ОС – операційна система.

САПР – система автоматизованого проектування.

СП – система приводів.

Wi-Fi – технологія бездротової локальної мережі.

GPIO – інтерфейс для зв'язку між компонентами в системі комп'ютерів.

PCB – друкована плата

LCD- дисплей (Liquid Crystal) – рідкокристалічний дисплей.

ВСТУП

Криптовалюта стали невід'ємною частиною сучасної фінансової системи, їхня популярність зростає завдяки децентралізації, анонімності та можливостям інвестування. Однак, нестабільність ринку криптовалют вимагає постійного моніторингу їх цін. Система моніторингу цін криптовалют у реальному часі за допомогою ESP8266 надає можливість доступно та ефективно рішення для відстеження цих змін, що є важливим для інвесторів та трейдерів.

Метою роботи є розробка системи моніторингу та відображення цін криптовалют за допомогою ESP8266. Для досягнення поставленої мети використовуються наступні методи:

Методи дослідження Аналіз літературних джерел та існуючих рішень, проектування апаратної та програмної частини системи, експериментальні дослідження для перевірки працездатності системи, аналіз результатів та їх інтерпретація.

Основні завдання включають: Розробку апаратної частини на основі ESP8266 та OLED дисплею. Створення програмного забезпечення для отримання даних з API криптовалют та їх відображення. Тестування та оцінка ефективності розробленої системи.

Об'єкт дослідження Система моніторингу криптовалют, яка використовує мікроконтролер ESP8266.

Предмет дослідження Процеси та методи збору, обробки та відображення даних про ціни криптовалют.

Практична цінність полягає у розробці та реалізації ефективної системи моніторингу цін криптовалют на базі ESP8266. А також у створенні доступного інструменту для інвесторів та трейдерів, що дозволяє оперативно реагувати на зміни ринку криптовалют. Розроблена система може бути використана як основа для подальших досліджень та розробок у сфері IoT та фінансових технологій.

ПОСТАНОВКА ЗАДАЧІ

1.1 Еволюція криптовалют та їх ринок

Криптовалюти з'явилися відносно недавно, на рубежі двох десятиліть і за цей час пройшли значний шлях розвитку та зазнали значних змін та еволюції, що вплинуло на світові фінансові системи та економіку. Першою криптовалютою, яка привернула увагу суспільства та викликала справжню революцію у фінансах, став Bitcoin, представлений анонімним розробником (або групою розробників) під псевдонімом Сатоші Накамото [1].

Bitcoin (BTC) був вперше представлений світу в жовтні 2008 року через білу книгу під назвою "*Bitcoin: A Peer-to-Peer Electronic Cash System*". Накамото запропонував систему, що дозволяє здійснювати електронні транзакції без потреби в центральному фінансовому посереднику.

Технологія, яка лежить в основі Bitcoin, називається блокчейн. Це децентралізована база даних, що зберігає інформацію про всі транзакції в мережі і захищає її за допомогою криптографічних методів.

Блокчейн забезпечує прозорість, незмінність та безпеку транзакцій, що робить можливим створення системи довіри без участі третьої сторони. Перша транзакція з використанням Bitcoin відбулася в січні 2009 року, коли Сатоші Накамото відправив 10 BTC Халу Фінні, одному з перших прихильників ідеї криптовалют [2].

Успіх Bitcoin сприяв появі альтернативних криптовалют, або так званих "altcoins". Одним з перших значних altcoins став *Litecoin*, створений у 2011 році Чарлі Лі. *Litecoin* був задуманий як "срібло" до "золота" Bitcoin. Він пропонував швидші транзакції та інші технічні покращення, такі як інший алгоритм хешування (Scrypt замість SHA-256).

Ethereum, запущений у 2015 році став новим етапом у розвитку криптовалют. *Ethereum* не тільки дозволяв здійснювати фінансові транзакції, але й підтримував смарт-контракти - самовиконувані контракти з умовами, написаними в коді. Це відкривало можливості для створення децентралізованих додатків (DApps) та розширювало сферу використання

блокчейну [4].

З середини 2010-х років ринок криптовалют зазнав вибухового зростання. Bitcoin досяг значних цінових піків, що привертало увагу інвесторів та спекулянтів. У грудні 2017 року Bitcoin досяг своєї найвищої вартості на той момент - близько \$20,000 за одиницю. Це сприяло зростанню популярності інших криптовалют, таких як Ripple (XRP), Cardano (ADA), Binance Coin (BNB) та Polkadot (DOT).

Важливою особливістю ринку криптовалют є його волатильність. Ціни на криптовалюти можуть значно коливатися за короткий період часу, що створює як можливості для високих прибутків, так і ризики великих втрат. Волатильність обумовлена декількома факторами, включаючи регуляторні новини, ринкові спекуляції та технологічні нововведення .

Ринок криптовалют стикається з численними регуляторними викликами. Деякі країни, такі як Японія та Швейцарія, прийняли криптовалюти та розробили сприятливе правове середовище для їх розвитку. Інші країни, такі як Китай, навпаки, заборонили або жорстко регулюють використання криптовалют.

Майбутнє криптовалют значною мірою залежить від того, як регуляторні органи по всьому світу будуть адаптуватися до цих нових технологій. З одного боку, криптовалюти пропонують численні переваги, такі як швидкість та дешевизна міжнародних транзакцій, а також доступ до фінансових послуг для незабезпечених населення. З іншого боку, існують значні ризики, пов'язані з відмиванням грошей, ухиленням від сплати податків та фінансуванням тероризму .

1.2 Технічні аспекти моніторингу криптовалют

Системи моніторингу криптовалют стали важливим інструментом для трейдерів, інвесторів і ентузіастів, які бажають слідкувати за ринковими змінами в режимі реального часу. Ці системи надають можливість отримувати актуальну інформацію про ціни, обсяги торгів, ринкову

капіталізацію та інші важливі метрики.

Моніторинг криптовалют вимагає використання різних технічних засобів, що забезпечують отримання, обробку та відображення даних про ціни в реальному часі. Основними компонентами таких систем є API (інтерфейси програмування додатків), технології обробки даних та платформи для відображення інформації [5].

API є основним інструментом для отримання даних про курси криптовалют. Вони забезпечують можливість отримувати актуальні дані у реальному часі через прості HTTP-запити, що повертають дані у форматі JSON або XML. Відомі платформи, такі як: CoinMarketCap яка надає API дані про ціни, обсяги торгів, ринкову капіталізацію та інші метрики для тисяч криптовалют, що дає можливість інтеграції з іншими системами та додатками. CoinGecko пропонує широкий спектр даних про криптовалюту, а також інформацію про розробницьку активність, соціальні медіа та інші фактори, що можуть впливати на її вартість. API CoinGecko є безкоштовним для більшості випадків використання. Binance є однією з найбільших криптовалютних бірж у світі та забезпечує потужний API для доступу до даних про торги, ордери та ринкові котирування. API Binance підтримує як спотові, так і ф'ючерсні ринки, що робить його цінним інструментом для професійних трейдерів та розробників алгоритмічних систем торгівлі. надають API, які дозволяють отримувати інформацію про ціни, обсяги торгів, ринкову капіталізацію та інші метрики.

Архітектура систем моніторингу криптовалют може включати наступні компоненти Рис.1.1. :

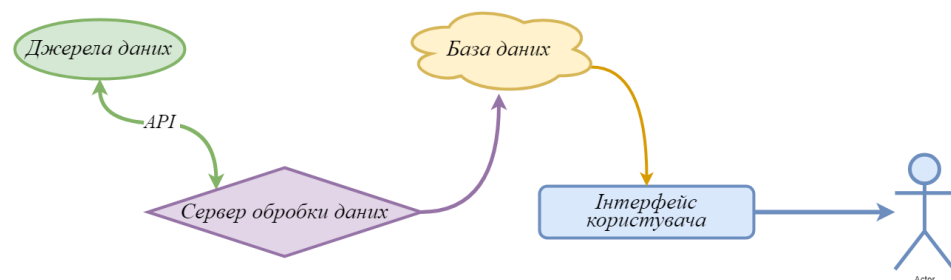


Рисунок 1.1. Архітектура систем моніторингу криптовалют

Джерела даних: Основою будь-якої системи моніторингу криптовалют є джерела даних, які забезпечують актуальну інформацію про ринок. Найчастіше для цього використовуються API (інтерфейси програмування додатків) криптовалютних бірж, таких як Binance, Coinbase, Bitfinex, а також спеціалізованих сервісів, як-от CoinMarketCap і CoinGecko. API надають можливість отримувати дані у форматі JSON або XML через HTTP-запити.

Сервер обробки даних: Відіграє ключову роль у забезпеченні швидкої та точної обробки інформації. Отримані дані потребують обробки та збереження. Для цього використовується серверна інфраструктура, яка може бути побудована на базі популярних мов програмування, таких як Python або JavaScript (Node.js). Бібліотеки на зразок Pandas (для Python) дозволяють обробляти великі обсяги даних та виконувати необхідні аналітичні операції

База даних: Дозволяє зберігати дані та забезпечують доступ до них у будь-який час. Для збереження оброблених даних використовуються реляційні (MySQL, PostgreSQL) та нереляційні (MongoDB, Redis) бази даних. Вибір конкретного типу бази даних залежить від обсягу даних, частоти оновлень та вимог до швидкості доступу до інформації.

Інтерфейс користувача: Дані повинні бути представлені користувачам у зручному та зрозумілому вигляді. Це може бути веб-інтерфейс, мобільний додаток або апаратний інтерфейс на базі мікроконтролерів. Веб-інтерфейси зазвичай реалізуються за допомогою фреймворків Angular, React або Vue.js, а мобільні додатки - за допомогою фреймворків Flutter, React Native або Swift.

Принципи роботи систем моніторингу криптовалют

Збір даних: Система здійснює регулярні запити до API для отримання актуальної інформації про ціни криптовалют, обсяги торгів та інші метрики. Частота запитів може варіюватися в залежності від вимог користувачів та технічних можливостей API.

Обробка даних: Отримані дані обробляються на сервері для видалення дублюючих записів, перевірки їх достовірності та перетворення у потрібний формат. Також можуть виконуватися аналітичні операції, такі як розрахунок

середніх значень, виявлення трендів та інші.

Збереження даних: Оброблені дані зберігаються у базі даних для подальшого аналізу та доступу користувачами. Це дозволяє зберігати історичні дані та забезпечувати доступ до них у будь-який час.

Відображення даних: Дані відображаються користувачам через інтерфейс, який може бути адаптований під конкретні потреби. Наприклад, для трейдерів важливо мати доступ до графіків змін цін у реальному часі, тоді як для інвесторів можуть бути важливими агреговані показники та аналітичні звіти.

Існуючі апаратні інтерфейси систем моніторингу криптовалют

Системи моніторингу криптовалют використовують різні апаратні інтерфейси для забезпечення зручності користування та отримання актуальної інформації у реальному часі. Ці інтерфейси можуть включати як стандартні комп'ютерні платформи, так і спеціалізовані мікроконтролери та IoT-пристрої, кожен з інтерфейсів має свої переваги та недоліки. Вибір конкретного рішення залежить від потреб користувачів, вимог до продуктивності, гнучкості та безпеки.

– Мікроконтролери та IoT-пристрої

ESP8266 є одним з найпопулярніших мікроконтролерів для створення недорогих та ефективних IoT-рішень. Завдяки вбудованому Wi-Fi модулю, він ідеально підходить для систем моніторингу криптовалют. ESP8266 може підключатися до інтернету, здійснювати запити до API криптовалютних бірж і відображати отримані дані на дисплеї. Основними перевагами ESP8266 є їх низька вартість, простота у використанні, широка підтримка з боку спільноти розробників, велика кількість бібліотек та прикладів для швидкого початку роботи [5].

Raspberry Pi — це недорогий одноплатний комп'ютер, який може використовуватися для створення більш потужних систем моніторингу криптовалют. Завдяки повноцінній операційній системі (наприклад, Raspbian), *Raspberry Pi* дозволяє використовувати стандартні інструменти

розробки та забезпечує більшу гнучкість у порівнянні з мікроконтролерами.

Перевагами Raspberry Pi є висока продуктивність, можливість використання повноцінної ОС, підтримка різноманітних периферійних пристроїв, можливість підключення до мережі через Ethernet або Wi-Fi.

Arduino — це платформа з відкритим вихідним кодом, яка широко використовується для створення прототипів різних електронних пристроїв. Для моніторингу криптовалют *Arduino* може використовуватися разом з Wi-Fi модулями (наприклад, ESP8266) для отримання даних з інтернету та відображення їх на екрані [6]. *Arduino* володіє простотою у використанні та налаштуванні, великою кількістю бібліотек та прикладів, підтримкою різноманітних датчиків та модулів.

– *Комп'ютерні платформи*

Персональні комп'ютери (ПК) використовуються для більш комплексних систем моніторингу криптовалют. ПК дозволяють використовувати потужне програмне забезпечення для аналізу та візуалізації даних, а також забезпечують високий рівень гнучкості та масштабованості.

Основні переваги ПК Висока продуктивність та швидкість обробки даних, можливість використання складного програмного забезпечення, підтримка великої кількості периферійних пристроїв, можливість інтеграції з іншими системами та сервісами.

Ноутбуки та планшети можуть використовуватися для мобільного моніторингу криптовалют. Вони забезпечують високу продуктивність та зручність у використанні завдяки своїй портативності. На таких пристроях можуть запускатися спеціалізовані додатки для моніторингу криптовалют, які забезпечують доступ до актуальної інформації у будь-який час та в будь-якому місці. Ноутбуки та планшети мають низку переваг портативність, висока продуктивність, можливість використання широкого спектра програмного забезпечення, інтеграція з мобільними додатками та сервісами.

– *Спеціалізовані апаратні пристрої*

Апаратні криптовалютні гаманці, деякі апаратні криптовалютні

гаманці, такі як Ledger Nano S та Trezor, також можуть забезпечувати функції моніторингу криптовалют. Вони дозволяють користувачам безпечно зберігати свої криптовалютні активи та відстежувати їх вартість у реальному часі через підключення до комп'ютера або мобільного додатку. Вони мають такі переваги: Високий рівень безпеки, захист від хакерських атак та шкідливого програмного забезпечення, можливість зберігання різних криптовалют, інтеграція з програмним забезпеченням для моніторингу, інтерактивні екрани та інформаційні панелі.

Для візуалізації даних про криптовалюту можуть використовуватися інтерактивні екрани та інформаційні панелі. Вони дозволяють відображати графіки, таблиці та іншу інформацію у зручному для користувачів форматі. Такі пристрої можуть бути інтегровані у торгові платформи або використовуватися в офісах та інших громадських місцях для інформування про стан ринку криптовалют. Інтерактивні екрани та інформаційні панелі зручні у використанні, можуть інтегруватися з різними системами, володіють високою якістю відображення інформації і них існує можливість налаштування під конкретні потреби користувачів.

1.3 Використання IoT у фінансових технологіях

Інтернет речей (IoT) має великий потенціал для трансформації фінансових технологій (FinTech). Завдяки підключенню різноманітних пристроїв до інтернету, IoT дозволяє автоматизувати процеси, збирати та аналізувати великі обсяги даних у реальному часі, а також забезпечувати новий рівень інтерактивності та зручності для користувачів.

Основними напрямки використання IoT у FinTech є автоматизація фінансових процесів, IoT дозволяє автоматизувати численні фінансові процеси, що значно підвищує їх ефективність та знижує витрати. Наприклад, розумні банкомати можуть самостійно моніторити свій стан і запас готівки, повідомляти про необхідність інкасації або технічного обслуговування. Це дозволяє знизити ризики простоїв та забезпечити безперебійну роботу

банкоматів.

IoT-пристрої можуть використовуватися для моніторингу та управління активами в реальному часі. Наприклад, страхові компанії можуть використовувати сенсори для моніторингу стану автомобілів або нерухомості та автоматично повідомляти про події, що можуть призвести до страхових випадків. Це дозволяє оперативно реагувати на проблеми та знижувати ризики.

Також IoT у поєднанні з блокчейн-технологіями може забезпечити високу безпеку та прозорість фінансових транзакцій. Розумні контракти, що виконуються автоматично при виконанні певних умов, можуть значно спростити фінансові операції та знизити витрати на їх здійснення. Наприклад, розумні контракти можуть автоматично здійснювати платежі за комунальні послуги або керувати інвестиціями на основі встановлених параметрів.

Пристрої на основі системи інтернету речей можуть використовуватися для збору великої кількості даних, які можуть бути проаналізовані для виявлення тенденцій та патернів у фінансовій діяльності. Це може включати моніторинг витрат, прогнозування грошових потоків, аналіз ризиків та інші види фінансової аналітики. Використання IoT дозволяє збирати дані у реальному часі, що підвищує точність та своєчасність аналізу.

Smart пристрої можуть використовуватися для здійснення автоматичних платежів та управління фінансами. Наприклад, розумні пристрої у домогосподарствах можуть автоматично здійснювати платежі за комунальні послуги, виходячи з поточних показників споживання. Це дозволяє знизити адміністративні витрати та підвищити зручність для користувачів.

IoT-пристрої можуть використовуватися для забезпечення безпеки та зручності інтернет-банкінгу та мобільних платежів. Наприклад, біометричні сенсори можуть використовуватися для аутентифікації користувачів, що забезпечує високий рівень безпеки транзакцій. Також можливе використання

розумних пристроїв для здійснення платежів за допомогою NFC-технології або QR-кодів.

Використання IoT у FinTech дозволяє автоматизувати численні фінансові процеси, що підвищує їх ефективність та знижує витрати. Наприклад, розумні банкомати можуть самостійно моніторити свій стан та запаси готівки, що дозволяє знизити витрати на їх обслуговування та забезпечити безперебійну роботу. Дозволяє створювати нові сервіси та покращувати існуючі, що підвищує зручність та якість обслуговування клієнтів. Наприклад, автоматичні платежі та персоналізовані страхові поліси на основі реального використання дозволяють забезпечити більш точне ціноутворення та знижувати витрати клієнтів.

Використання IoT у поєднанні з блокчейн-технологіями дозволяє забезпечити високий рівень безпеки та прозорості фінансових транзакцій. Розумні контракти, що виконуються автоматично при виконанні певних умов, можуть значно спростити фінансові операції та знизити витрати на їх здійснення. Це дозволяє знизити ризики шахрайства та підвищити довіру до фінансових систем. Також IoT дозволяє збирати великі обсяги даних у реальному часі, що підвищує точність та своєчасність фінансового аналізу.

Інтернет речей (IoT) відкриває нові можливості для фінансових технологій, забезпечуючи автоматизацію процесів, підвищення ефективності та зручності для користувачів, а також покращення безпеки та прозорості фінансових транзакцій. Завдяки використанню IoT, фінансові установи можуть створювати нові сервіси, покращувати обслуговування клієнтів та знижувати витрати, що сприяє розвитку інновацій у сфері FinTech.

ПРОЕКТУВАННЯ АПАРАТНОЇ СКЛАДОВОЇ СИСТЕМИ МОНІТОРИНГУ ТА ВІДОБРАЖЕННЯ ЦІНИ КРИПТО ВАЛЮТ

2.1 Вибір основних компонент та їх характеристики

Для розробки системи моніторингу та відображення цін криптовалют на базі ESP8266 важливо обрати відповідні компоненти, які забезпечать надійну та ефективну роботу системи.

ESP8266 є ключовим компонентом у цій системі. Це потужний мікроконтролер з вбудованим модулем Wi-Fi, який використовується для створення недорогих та ефективних систем моніторингу криптовалют він забезпечує підключення до інтернету для отримання даних про ціни криптовалют у режимі реального часу та відображати їх на OLED-екрані або іншому дисплеї.

ESP8266 отримав широке поширення як економічно ефективне рішення для пристроїв із підтримкою Інтернету речей і Wi-Fi. ESP8266 був розроблений компанією Espressif Systems із Шанхаю як послідовний (UART) Wi-Fi SoC (система на кристалі) на базі DPU Tensilica Xtensa LX3. Ця крихітна мікросхема включає радіочастотний передній кінець, оперативну пам'ять і (зазвичай) вбудований стек TCP/IP, який дозволяє їй підключатися до найближчої точки доступу, діяти як сама точка доступу або обидва.

Вона забезпечує економічно ефективне рішення для швидко зростаючого ринку підключених до Інтернету проектів і пристроїв, тому він став однією з найпопулярніших платформ розробки за останні роки Рис. 2.1.

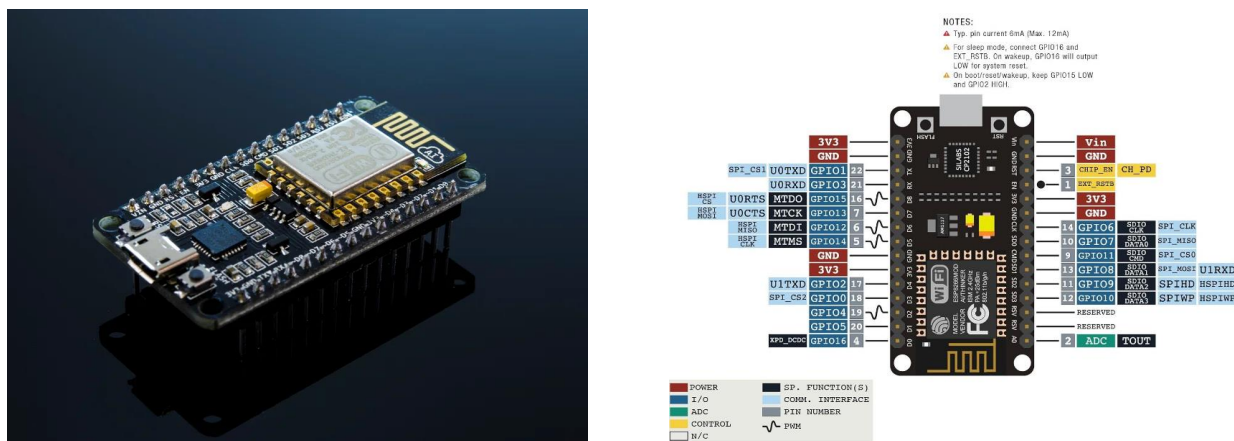


Рисунок 2.1. Мікроконтролер ESP8266

ESP8266 — це плата розробки, інтегрована з IoT WiFi, TCP/IP, 32-розрядним мікроконтролером, 10-розрядним АЦП і кількома інтерфейсами, такими як HSPI, UART, PWM, I2C і I2S. У DTIM10 повне енергоспоживання для підтримки з'єднання WiFi досягло 1,2 мВт. Оснащений 16 МБ зовнішньої флеш-пам'яті SPI, ESP8266 доступний для зберігання програм і прошивки.

Сумісність із програмуванням Arduino дозволяє Firebeetle ESP8266 знизити бар'єр програмування. Оператор може впроваджувати коди програмування Arduino безпосередньо на ESP8266, щоб зменшити складність роботи та підвищити стабільність плати esp8266.

Таблиця 1.1. Характеристики ESP8266

<i>№</i>	<i>Параметр</i>	<i>Значення</i>
1.	SKU	DFR0489
2.	Мікроконтролер	ESP8266
3.	Інтерфейс джерела живлення	USB або 3,7 В Lipo
4.	USB живлення або зовнішній (V)	3,3 - 5,0
5.	Робоча напруга (В)	3,3 В
6.	Частота ЦП (МГц)	160
7.	Спалах (М)	16
8.	SRAM (КБ)	50
9.	Аналогові штифти	1
10.	Цифрові шпильки	10
11.	Протокол Wi-Fi	802.11b/g/n
12.	Діапазон частот	2,4 - 2,5 ГГц
13.	UART	1
14.	I2C	1
15.	I2S	1
16.	SPI	1
17.	Сумісна IDE	Arduino IDE 1.6+
18.	Режим завантаження	Micro USB
19.	Сумісність з Arduino UNO	×
20.	Підтримка зарядного пристрою Lipo	√
21.	Розмір (мм)	58*29
22.	Вага (г)	24 г
23.	Ключові особливості	Підтримка програмування Arduino, RTOS, microPython. Вбудований 32-розрядний MCU Tensilica L106 і 10-розрядний АЦП.

Враховуючи головну мету мати пристрій малого форм-фактора, який міг би розміститися на наших столах, першим вибором було використання 0,96-дюймового OLED-екрану для відображення даних про ціну. Рис.2.2.



Рисунок 2.2. OLED-екран

Ці дисплеї невеликі, лише близько 1 дюйма в діаметрі, але добре читаються завдяки високій контрастності OLED-дисплеїв. Цей дисплей складається з окремих пікселів OLED 128x64, кожен з яких вмикається або вимикається чіпом контролера. Верхні 16 рядків складаються з жовтих пікселів, а нижні 48 — синіх. Оскільки дисплей освітлюється самостійно, підсвічування не потрібне. Це зменшує енергоспоживання, необхідне для роботи OLED, і тому дисплей має такий високий контраст;

Цей роз'єм можна використовувати з інтерфейсом SPI або I2C, який можна вибрати, припаявши дві перемички на задній панелі. Конструкція повністю готова до 5 В, з бортовим регулятором і вбудованим підвищуючим перетворювачем. Підключатися безпосередньо до мікроконтролера 3,3 В або 5 В або одноплатного комп'ютера простіше, ніж будь-коли ними дуже легко керувати через i2c, просто потрібно підключити 4 дроти до нашого мікроконтролера або одноплатного комп'ютера: 2 для живлення та заземлення, а 2 для даних і годинника лінії

Таблиця 2.2. Характеристики OLED-екран

№	Параметр	Значення
1.	Розмір	0,96 дюйма
2.	Роздільна здатність	128 64 пікселів
3.	Кут огляду	більше 160 град
4.	Потужність	0,04 Вт (режим низького споживання)
4.	Вхідна напруга	3,3 В ~ 5 В постійного струму
6.	Робоча температура	–30~80°C

7.	Розмір	27 мм × 27 мм × 4,1 мм
8.	Драйвер IC	SSD1306
9.	Зв'язок	I2C
10.	Підсвічування	OLED з самопідсвічуванням (без підсвічування)
11.	Колір пікселя	жовтий (16 рядків) і синій

2.2. Принципи роботи з API для отримання цін криптовалют (CoinMarketCap, Coindesk)

Використання API (інтерфейсів програмування додатків) є ключовим аспектом для систем моніторингу криптовалют. API дозволяють отримувати актуальні дані про їх ціни у реальному часі, що є критично важливим для трейдерів та інвесторів. Розглянемо принципи роботи з API від CoinMarketCap та Coindesk, а також їх основні характеристики та приклади використання.

CoinMarketCap (CMC) є одним з найпопулярніших ресурсів для отримання даних про криптовалюту. Їх API дозволяє отримувати інформацію про ціни, ринкову капіталізацію, обсяги торгів та інші важливі метрики.

Основні характеристики CoinMarketCap API:

Latest Listings: Отримання поточних цін на криптовалюту.

Historical Data: Дані про ціни за певний період часу.

Global Metrics: Загальні показники ринку, такі як загальна ринкова капіталізація.

HTTP-запити, що повертають дані у форматі JSON.

Підтримка GET-запитів для отримання інформації.

Використання API-ключа для доступу до даних. API-ключ можна отримати після реєстрації на сайті CoinMarketCap.

Різні рівні доступу (безкоштовний та платний), які обмежують кількість запитів на одиницю часу. Безкоштовний план дозволяє до 10,000 запитів на місяць.

```
import requests

url="https://pro-
api.coinmarketcap.com/v1/cryptocurrency/listings/latest"
```

```

parameters = {
    'start': '1',
    'limit': '10',
    'convert': 'USD'
}
headers = {
    'Accepts': 'application/json',
    'X-CMC_PRO_API_KEY': 'your_api_key_here',
}

response = requests.get(url, headers=headers,
params=parameters)
data = response.json()

for currency in data['data']:
    print(f"{currency['name']}:
${currency['quote']['USD']['price']:.2f}")

```

Цей код отримує поточні ціни на 10 найпопулярніших криптовалют і виводить їх у форматі "Назва: \$Ціна".

Coindesk також є популярним ресурсом для отримання даних про криптовалюту, особливо для цінових індексів та новин.

Основні характеристики Coindesk API:

Ендпоінти: Current Price: Поточні ціни на Bitcoin. Historical Data: Історичні дані про ціни Bitcoin. News: Останні новини та аналітика.

Метод доступу: HTTP-запити, що повертають дані у форматі JSON., Підтримка GET-запитів для отримання інформації.

Автентифікація: Деякі ендпоінти можуть вимагати використання API-ключа, однак базові запити зазвичай доступні без нього.

Планування запитів: Безкоштовний доступ до більшості даних.

Приклад використання Coindesk API:

```

import requests

url = "https://api.coindesk.com/v1/bpi/currentprice.json"
response = requests.get(url)
data = response.json()

print("Bitcoin Price Index:")
print(f"USD: ${data['bpi']['USD']['rate']}")
print(f"GBP: £{data['bpi']['GBP']['rate']}")
print(f"EUR: €{data['bpi']['EUR']['rate']}")

```

Цей код отримує поточні ціни Bitcoin у доларах США, фунтах стерлінгів та євро і виводить їх у зручному форматі.

Переваги використання API

Актуальність даних: API дозволяють отримувати дані в режимі реального часу.

Автоматизація: Використання API дозволяє автоматизувати процес збору та обробки даних, що знижує ймовірність помилок та підвищує ефективність.

Гнучкість: Різні ендпоінти API надають доступ до широкого спектра даних, що дозволяє створювати індивідуальні рішення для різних потреб.

API від CoinMarketCap та Coindesk є потужними інструментами для отримання даних про криптовалюти. Вони забезпечують доступ до актуальної інформації, що є необхідним для прийняття обґрунтованих рішень у світі криптовалют. Використання API дозволяє автоматизувати процеси та отримувати дані у зручному форматі, що значно спрощує роботу з криптовалютами.

API від CoinMarketCap та Coindesk є потужними інструментами для отримання даних про криптовалюти. Вони забезпечують доступ до актуальної інформації, що є необхідним для прийняття обґрунтованих рішень у світі криптовалют. Використання API дозволяє автоматизувати процеси та отримувати дані у зручному форматі, що значно спрощує роботу з криптовалютами.

2.3. Налаштування апаратної частини: схеми підключення

Плата ESP8266 включає в себе все необхідне, включаючи адаптер USB-Serial, світлодіод для поверхневого монтажу та регулятор напруги. Крім того, його легко встановити на макетну плату для розробки власних IoT-проектів.

Для програмування та живлення цього проекту потрібен кабель мікро-USB,

У середовищі Fritzing створимо модель підключення екрану до ESP8266 Рис.2.3.

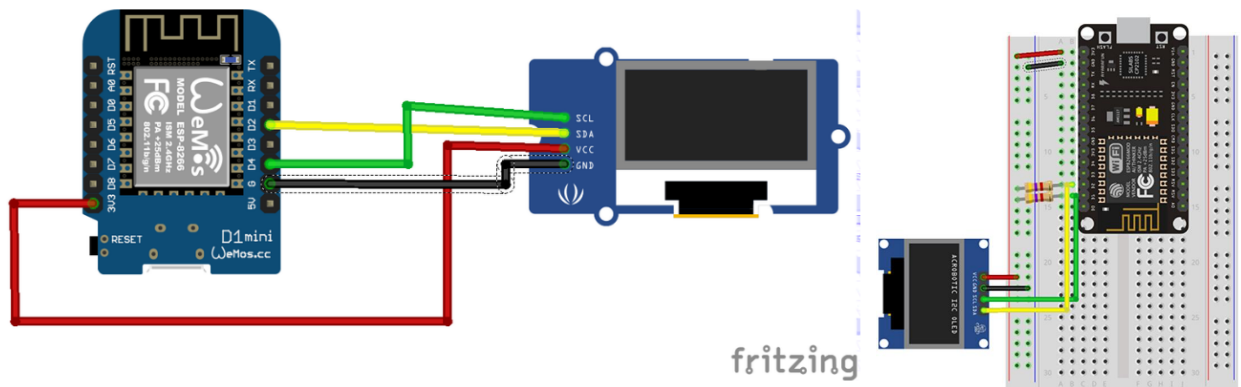


Рисунок 2.3. Макетна схема Трекера криптовалюти

Підключення здійснюється наступним чином:

Використовуються 2 резистори по 4,7 кОм, щоб підтягнути лінії I2C. Використання контактів I2C за замовчуванням для ESP8266 дозволяє нам просто використовувати вбудовану бібліотеку Wire для зв'язку I2C.

SCL дисплея OLED під'єднується до D2 ESP8266, SDA дисплея до D4, VCC до 3,3 В, GND до землі

ПРОЕКТУВАННЯ ПРОГРАМНОЇ СКЛАДОВОЇ СИСТЕМИ МОНІТОРИНГУ ЗДОРОВ'Я

3.1 Структура програмного забезпечення та хмарного сховища для системи IoT

IBM Cloud: *IBM Cloud* - це комплексна платформа для хмарних обчислень, що надає широкий спектр послуг та інструментів для розробників, вона пропонує широкий спектр послуг, включаючи обчислення, зберігання даних, мережі, аналітику даних, штучний інтелект та блокчейн. *IBM Cloud* відома своєю безпекою та надійністю, вона включає інфраструктуру (IaaS), платформу (PaaS) та програмне забезпечення (SaaS).

IBM Cloud: *IBM Cloud* володіє такими ресурсами:

– *Обчислювальні ресурси*

IBM Cloud Virtual Servers: Надає масштабовані віртуальні сервери, які можуть бути розгорнуті та налаштовані за кілька хвилин, підтримують різні операційні системи та конфігурації, що дозволяє користувачам вибирати потрібні ресурси відповідно до своїх потреб.

Bare Metal Servers: Виділені сервери з високою продуктивністю для робочих навантажень, що потребують значних обчислювальних потужностей, забезпечують повний контроль над апаратним забезпеченням та його налаштуванням.

Serverless Computing (IBM Cloud Functions): Підтримка безсерверної архітектури, що дозволяє розробникам виконувати код у відповідь на події без необхідності управління інфраструктурою, оплата лише за фактичне використання ресурсів, що дозволяє знизити витрати.

– *Збереження даних*

Object Storage: Масштабоване сховище для зберігання неструктурованих даних з підтримкою API для легкого доступу та інтеграції, забезпечує високу доступність та безпеку даних.

Block Storage: Високопродуктивне блочне сховище для баз даних та додатків, що потребують швидкого доступу до даних, підтримує

налаштування продуктивності та об'єму сховища відповідно до потреб користувачів.

File Storage: Сховище для збереження файлів з підтримкою протоколів NFS та SMB, Підходить для спільної роботи та зберігання великих обсягів даних.

– *Мережа*

Content Delivery Network (CDN): Глобальна мережа серверів для прискорення доставки контенту користувачам, знижує затримки та покращує продуктивність веб-додатків.

Direct Link: Приватні з'єднання між інфраструктурою користувача та IBM Cloud для забезпечення безпечної та надійної передачі даних, підвищує безпеку та знижує затримки при передачі даних.

– *Інструменти для розробників*

IBM Cloud Continuous Delivery: Інструменти для безперервної інтеграції та доставки (CI/CD), що дозволяють автоматизувати процес розгортання додатків, підтримка різних середовищ розробки, включаючи GitHub та Jenkins.

IBM Cloud CLI: Командний інтерфейс для управління ресурсами IBM Cloud. Дозволяє розробникам автоматизувати розгортання та управління інфраструктурою.

IBM Cloud Pak: Інтегровані рішення для розробки, управління та захисту додатків. Підтримка контейнерних технологій, таких як Kubernetes та Docker, для масштабованого розгортання додатків.

– *Штучний інтелект та аналітика*

IBM Watson: Набір інструментів для розробки додатків на основі штучного інтелекту, підтримує технологій машинного навчання, обробки природної мови, комп'ютерного зору та інших AI-технологій.

IBM Cloud Databases: Керовані бази даних, що включають підтримку SQL, NoSQL та in-memory баз даних, масштабування, високої доступності та безпеки даних.

IBM Cloud Analytics: Інструменти для аналізу даних та бізнес-аналітики, підтримка візуалізації даних, аналітичних панелей та звітності.

IBM Cloud пропонує широкий спектр послуг для різних потреб бізнесу та розробників. Від обчислювальних ресурсів та зберігання даних до інструментів для розробників та технологій штучного інтелекту - IBM Cloud забезпечує все необхідне для створення, розгортання та управління сучасними додатками. Ці функції дозволяють підприємствам масштабувати свої операції, підвищувати продуктивність та знижувати витрати, що робить IBM Cloud потужним інструментом для сучасних IT-рішень.

3.2 Написання та відладка програмного коду

Програмування системи передбачає збір, відображення та передачу даних про ціну криптовалюти на основі модуля ESP 8266 та відображення результатів на OLED екрані.

Щоб отримати дані, спочатку знаходимо їх у вільному доступі. Пошук Google показує 2 популярні служби, якими можна користуватися, CoinMarketCap і Coindesk. Перегляд їхніх API дає нам знати, що ми можемо отримати доступ до поточної ціни для різних криптовалют, надіславши запит GET на:

CoinMarketCap за допомогою <https://api.coinmarketcap.com/v1/ticker/bitcoin/>

Coindesk за допомогою <https://api.coindesk.com/v1/bpi/currentprice.json>

Якщо ми просто наведемо наш браузер на одну з URL-адрес, ми побачимо дані, які ми шукаємо. Однак, щоб зробити це з ESP8266, нам потрібно спочатку підключитися до Інтернету, а потім використовувати веб-клієнт. Незважаючи на те, що обидва сервіси працюють чудово, у цьому проєкті ми зупинимося на Coindesk.

– Підключення до Інтернету

На основі бібліотеки ESP8266WiFi.h, ми використовуємо метод begin об'єкта WiFi з парою масивів символів як аргументів, які містять облікові дані для доступу до мережі.

```

const char* ssid = "ВАШ_SSID";
const char* password = "ВАШ_ПАРОЛЬ";

WiFi.begin(ssid, пароль);

while (WiFi.status() != WL_CONNECTED)
{
    затримка (500); Serial.print(".");
}

```

Це виконується всередині функції `setup()`. Крім того, також використовується цикл `while()` для моніторингу стану з'єднання. Цикл виконується, доки з'єднання не буде успішним, хоча це сліпо припускає, що мережа доступна. Щоб зробити код більш надійним, можна обробляти випадки, коли з'єднання не вдається.

– Використання веб-клієнта

Використовуючи вбудовану бібліотеку `ESP8266WiFi.h`, створюється екземпляр класу `WiFiClient`, налаштовуємо запит, який будемо надсилати на сервер `Coindesk`, і оголошуємо буфер, у якому буде зберігатися відповідь сервера.

Ми також створюємо запит, який збираємося надіслати на сервер, вказуючи мінімальний набір необхідних полів заголовка.

```

#define CD_API "/v1/bpi/currentprice.json"
#define CD_URL "api.coindesk.com"

клієнт WiFiClient;
static char respBuffer[4096];

const char запит [] =
    "ОТРИМАТИ " CD_API " HTTP/1.1\r\n"
    "Агент користувача: ESP8266/0.1\r\n"
    "Прийняти: */*\r\n"
    "Хост: " CD_URL "\r\n"
    "Підключення: закрити\r\n"
    "\r\n";

```

Далі ми намагаємося підключитися до хост-сайту за допомогою методу `connect()` об'єкта `WiFiClient`. Якщо підключення не вдалось, ми

виконуємо базову обробку помилок.

Використовуючи метод `print()` об'єкта `WiFiClient`, ми надсилаємо запит. Потім ми закриваємо та очищаємо з'єднання, а також чекаємо довільний, але розумний проміжок часу (1 секунда), поки сервер відповість.

Коли сервер відповідає, використовуємо цикл `while()`, щоб завантажити відповідь у буфер символів. Для цього ми використовуємо метод `read()` об'єкта `WiFiClient`.

```
if (!client.connect(CD_URL, 80))
{<br> Serial.println("Помилка підключення");
  повернення;
}
client.print(request);<br>client.flush();                                затримка
(1000);<br>індекс          uint16_t          =          0;
while(client.connected())<br>{<br>    if(client.available())<br>{<br>    respBuffer[index++] = client.read();<br>    delay(1);<br>    }
}<br>}
```

Завдяки дизайну API, до якого ми надсилаємо запит, ми знаємо, що відповідь від сервера буде відформатовано в JSON, роздруковуючи буфер символів, ми помічаємо, що у відповіді сервера є додаткові символи. З цієї причини ми використовуємо метод `strchr()`, щоб обрізати буфер (створюючи новий) аж до першої відкриваючої фігурної дужки '{', де ми знаємо, перевібивши, що починаються дані у форматі JSON.

```
char * json = strchr(respBuffer, '{');

DynamicJsonBuffer jBuffer;
JsonObject& root = jBuffer.parseObject(json);
Serial.println("JsonObject: ");
root.prettyPrintTo(Serial);
Serial.println();

JsonObject& bpi = root["bpi"];
JsonObject& usd = bpi["USD"];
Рядок tmp = usd["rate_float"];
дані = tmp;

Serial.print("BTC/USD:");<br>Serial.println(дані);
```

Використовуючи фантастичну бібліотеку `ArduinoJson`, ми можемо

аналізувати дані. На основі створення статичного або динамічного об'єкта JsonBuffer (ми вибрали пізніший) і викликавши його відповідний метод parseData(). Потім об'єкт(и) JsonBuffer дозволяє нам вибрати потрібні дані за допомогою знайомої нотації масиву з ключами.

Наразі ми можемо візуалізувати дані в Serial Monitor за допомогою методу print() об'єкта Serial (не забудьте ініціалізувати його у функції setup()) наступним кроком є передача його на OLED-екран.

Хоча існує кілька бібліотек для роботи з екранами OLED, ми вирішили слідувати в автономній демонстрації OLED: Ai_Demos_ESP8266/tree/master/i2c_oled/hello_oled Таким чином, ми створили допоміжний файл util.h, який містить кілька констант, включаючи дані шрифту.

Потім створимо кілька допоміжних функцій низького рівня для налаштування OLED-екрану та спілкування з ним. І одна функція високого рівня для друку потрібних нам даних:

```
    setTextXY(0,0); // Встановити курсор 0-ї сторінки (рядка),
0-го стовпця
    displayString("BTC/USD:");

void printData (рядкові дані)
{
    setTextXY(2,0);
    displayString("
    setTextXY(0,0); // Встановити курсор 0-ї сторінки (рядка),
0-го стовпця
    displayString("BTC/USD:");

void printData (рядкові дані)
{
    setTextXY(2,0);
    displayString("$");
    setTextXY(2,1);
    char __data[sizeof(data)];
    data.toCharArray(__data, sizeof(__data));
    displayString(__data);
}

    setTextXY(2,1);
    char __data[sizeof(data)];
```

```
data.toCharArray(__data, sizeof(__data));  
displayString(__data);  
}
```

Також потрібно інкапсулювали збір і аналіз даних у функцію `getData()`, а дані, що відображаються на OLED- екрані, – у функцію `prinData()`. Ці дві функції викликаються зі стандартної функції `loop()` кожні 10 секунд.

ВИСНОВКИ

Розробка системи моніторингу та відображення ціни криптовалют за допомогою мікроконтролера ESP8266 є важливим кроком у створенні доступних та ефективних рішень для вивчення та використання технологій Інтернету речей (IoT) у фінансових додатках. У цьому проєкті було реалізовано декілька ключових компонентів, що дозволяють отримувати та відображати актуальну інформацію про ціни криптовалют у реальному часі. Цей висновок узагальнює основні досягнення, виклики та перспективи подальшого розвитку системи.

Використання мікроконтролера ESP8266 дозволило створити компактну та енергоефективну систему. Підключення до інтернету через вбудований Wi-Fi модуль забезпечило можливість отримання даних у реальному часі. OLED-дисплей був обраний як засіб відображення інформації, що забезпечило високу чіткість та контрастність зображення, необхідні для зручності користувачів.

Розроблено програмне забезпечення для отримання даних про ціни криптовалют з використанням API CoinMarketCap та Coindesk. Було забезпечено обробку та парсинг JSON-даних, що дозволило відображати актуальну інформацію на дисплеї.

Створено базову програму для ESP8266, яка включає функції підключення до Wi-Fi, здійснення HTTP-запитів та відображення отриманих даних.

Використання безсерверної архітектури забезпечило гнучкість та масштабованість системи. Це дозволяє легко інтегрувати додаткові функції та сервіси без необхідності значних змін у основному коді.

Забезпечено базові рівні безпеки даних, такі як шифрування з'єднання через HTTPS та обробка помилок у разі втрати з'єднання або невдалих запитів до API.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Nakamoto, S. "Bitcoin: A Peer-to-Peer Electronic Cash System". Retrieved from. (2008). Bitcoin.org.
2. Vigna P., & Casey M. J. "The Age of Cryptocurrency: How Bitcoin and Digital Money Are Challenging the Global Economic Order". St. Martin's Press. (2015).
3. Islam, S. M. The Internet of Things for Health Care: A Comprehensive Survey. R., Kwak, D., Kabir, M. H., Hossain, M., & Kwak, K.-S. (2015). IEEE Access, 3, 678–708.
4. Wood, G. (2014). "Ethereum: A Secure Decentralised Generalised Transaction Ledger". Retrieved from Ethereum.org.
5. Bailey, J. (2017). "Getting Started with ESP8266: A Beginner's Guide". Retrieved from Instructables.
6. Programming Digest. "IOT Water Quality Monitoring using Arduino, pH Sensor, ESP8266 & Blynk". Retrieved from Programming Digest. (2023).

ДОДАТКИ

Додаток А

Лістинг програми

```
* ESP8266 Crypto
*
* A program to display various cryptocurrency and stock
* prices on an OLED display using an ESP8266
*
* Conor Walsh 2021-02-27
* conor.engineer
*/

#include <Arduino.h>
#include <ESP8266WiFi.h>
#include <ESP8266WiFiMulti.h>
#include <ESP8266HTTPClient.h>
#include <WiFiClientSecureBearSSL.h>

/**** Setup ****/

// Uncomment the next line if you are using an ESP-01
// #define ESP01

// WIFI Network name and password
#define WIFI_SSID "YOUR_NETWORK"
#define WIFI_PASS "YOUR_PASSWORD"

// Setup time in seconds
#define SETUP_TIME 4

// Update screen every __ seconds
#define UPDATE_TIME 10

// Currency for crypto, tested: eur usd gbp
#define CURRENCY_CODE "eur"
// Currency symbol for crypto, tested: letters and $
#define CURRENCY_SYM 'E'
```

					Система моніторингу та відображення ціни крипто валютних акцій за допомогою ESP8266					Лім.		Маса		Масштаб		
Змн.	Арк.	№ докум.	Підпис	Дата												
Розроб.		Мойсюк М. В.														
Перевір.		Букурос О.В.														
Т. Контр.																
Реценз.										Арк.		Аркушів		1		
Н. Контр.										Циклова комісія Комп'ютерної інженерії						
Затверд.																

```

// Fingerprint for api.cryptonator.com - expires 6 Feb 2022
const uint8_t fingerprint_crypto[20] = {0x10, 0x76, 0x19, 0x6B,
0xE9, 0xE5, 0x87, 0x5A, 0x26, 0x12, 0x15, 0xDE, 0x9F, 0x7D,
0x3B, 0x92, 0x9A, 0x7F, 0x30, 0x13};
// Fingerprint for finnhub.io - expires 10 Aug 2022
const uint8_t fingerprint_stock[20] = {0x95, 0xE6, 0xC6, 0x0D,
0xD3, 0x9E, 0xC0, 0xB2, 0x40, 0x7C, 0x47, 0x66, 0xB5, 0x89,
0x8F, 0x51, 0x72, 0x81, 0x27, 0xA6};

// API Key for finnhub.io, get your free API key at finnhub.io
const String stocksApiKey = "XXXXXXXXXXXXXXXXXXXX";

/** End Setup */

ESP8266WiFiMulti WiFiMulti;

#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

// Screen Definitions
#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
#define OLED_RESET -1
#define SCREEN_ADDRESS 0x3C
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire,
OLED_RESET);

// All asset logos must be 48x48
#define LOGO_WIDTH 48
#define LOGO_HEIGHT 48

// Define isCrypto flags
#define CRYPTO true
#define STOCK false

```


```
// Struct to store an asset
struct asset {
    bool isCrypto;
    String assetName;
    String url;
    const unsigned char logo [288];
};

// Declare all assests used
// To add a new asset
// 1: Add a new setup similar to one shown in the examples
// 2: Update the name and isCrypto flag for the asset (either
CRYPTO or STOCK)
// 3: Find the api.cryptonator.com or finnhub.io api url for
the asset
// 4: Create a 48x48 bitmap logo for the asset using
http://javl.github.io/image2cpp/
asset assets[] = {
    // Doge setup
    {CRYPTO, "Doge",
    "https://api.cryptonator.com/api/ticker/doge-
"+String(CURRENCY_CODE),
    {0x03, 0x00, 0x00, 0x01, 0xe0, 0x00, 0x07, 0xc0, 0x00,
0x03, 0xf8, 0x00, 0x0f, 0xe0, 0x00, 0x07,
0xf8, 0x00, 0x0f, 0xe0, 0x00, 0x0f, 0xfc, 0x00, 0x1f,
0xf0, 0x00, 0x3f, 0x3c, 0x00, 0x1e, 0xf8,
0x00, 0x7e, 0x38, 0x00, 0x1e, 0x78, 0x00, 0xfc, 0x38,
0x00, 0x1c, 0x3c, 0x03, 0xff, 0x38, 0x00,
0x3c, 0x3f, 0xff, 0xff, 0xf8, 0x00, 0x3c, 0x7f, 0xff,
0xff, 0xf8, 0x00, 0x3c, 0xff, 0xff, 0x83,
0xf8, 0x00, 0x38, 0x78, 0x04, 0x01, 0xfc, 0x00, 0x78,
0x20, 0x00, 0x00, 0xfe, 0x00, 0x78, 0x00,
0x00, 0x00, 0x3f, 0x00, 0x78, 0x00, 0x00, 0x00, 0x1f,
0x80, 0x78, 0x00, 0x00, 0x00, 0x07, 0xc0,
0x78, 0x00, 0x00, 0x01, 0xe3, 0xe0, 0x78, 0x00, 0x00,
0x03, 0xe1, 0xe0, 0x78, 0x00, 0x78, 0x03,
0xf1, 0xf0, 0x78, 0x00, 0x7c, 0x03, 0xe0, 0xf0, 0x78,
0x00, 0xfc, 0x01, 0xe0, 0x78, 0xf0, 0x00,
0x7c, 0x00, 0x00, 0x78, 0xf0, 0x00, 0x38, 0x00, 0x00,
0x3c, 0xf0, 0x00, 0x00, 0x00, 0x00, 0x3c,
```

					Система моніторингу та відображення ціни крипто валютних акцій за допомогою ESP8266	Лім.			Маса	Масштаб
Змн.	Арк.	№ докум.	Підпис	Дата						
Розроб.		Мойсюк М. В.								
Перевір.		Букурос О.В.								
Т. Контр.						Арк.			Аркушів 1	
Реценз.						Циклова комісія Комп'ютерної інженерії				
Н. Контр.										
Затверд.										

```

0xe0, 0x00, 0x00, 0x00, 0x00, 0x1c, 0xe0, 0x00, 0x00,
0x00, 0xf8, 0x1e, 0xe0, 0x00, 0x00, 0x03,
0xf8, 0x1e, 0xe0, 0x00, 0x00, 0x03, 0xfc, 0x1e, 0xe0,
0x00, 0x00, 0x01, 0xf8, 0x1e, 0xe0, 0x00,
0x00, 0x00, 0xf8, 0x1e, 0xf0, 0x00, 0x00, 0x00, 0x70,
0x1e, 0xf0, 0x00, 0x00, 0x00, 0x00, 0x1e,
0x78, 0x00, 0x00, 0x00, 0x00, 0x1e, 0x78, 0x00, 0x00,
0x00, 0x30, 0x1e, 0x7c, 0x00, 0x00, 0x38,
0xf8, 0x1e, 0x3e, 0x00, 0x00, 0x3f, 0xf8, 0x1e, 0x1e,
0x00, 0x00, 0x7f, 0xf0, 0x3c, 0x1f, 0x00,
0x00, 0x3f, 0xc0, 0x3c, 0x0f, 0xc0, 0x00, 0x00, 0x00,
0x78, 0x07, 0xe0, 0x00, 0x00, 0x00, 0x78,
0x03, 0xf0, 0x00, 0x00, 0x00, 0xf0, 0x01, 0xfc, 0x00,
0x00, 0x01, 0xf0, 0x00, 0x7f, 0x00, 0x00,
0x07, 0xe0, 0x00, 0x3f, 0xf0, 0x00, 0x1f, 0xc0, 0x00,
0x0f, 0xff, 0xff, 0xff, 0x80, 0x00, 0x03,
0xff, 0xff, 0xff, 0x00, 0x00, 0x00, 0xff, 0xff, 0xfc,
0x00, 0x00, 0x00, 0x03, 0xff, 0xc0, 0x00}
},
// BTC setup
{CRYPTO, "BTC",
"https://api.cryptonator.com/api/ticker/btc-
"+String(CURRENCY_CODE),
{0x00, 0x00, 0x0f, 0xf0, 0x00, 0x00, 0x00, 0x00, 0xff,
0xff, 0x00, 0x00, 0x00, 0x03, 0xff, 0xff,
0xc0, 0x00, 0x00, 0x0f, 0xff, 0xff, 0xf0, 0x00, 0x00,
0x3f, 0xff, 0xff, 0xfc, 0x00, 0x00, 0x7f,
0xff, 0xff, 0xfe, 0x00, 0x00, 0xff, 0xff, 0xff, 0xff,
0x00, 0x01, 0xff, 0xff, 0xff, 0xff, 0x80,
0x03, 0xff, 0xff, 0xff, 0xff, 0xc0, 0x07, 0xff, 0xfe,
0x3f, 0xff, 0xe0, 0x0f, 0xff, 0xfe, 0x73,
0xff, 0xf0, 0x0f, 0xff, 0xfc, 0x63, 0xff, 0xf0, 0x1f,
0xff, 0x0c, 0x63, 0xff, 0xf8, 0x1f, 0xff,
0x00, 0x63, 0xff, 0xfc, 0x3f, 0xff, 0x80, 0x07, 0xff,
0xfc, 0x3f, 0xff, 0xe0, 0x01, 0xff, 0xfc,
0x7f, 0xff, 0xe0, 0x80, 0x7f, 0xfe, 0x7f, 0xff, 0xe0,
0xf0, 0x3f, 0xfe, 0x7f, 0xff, 0xe1, 0xf8,
0x3f, 0xfe, 0xff, 0xff, 0xc1, 0xf8, 0x1f, 0xff, 0xff,
0xff, 0xc1, 0xf8, 0x1f, 0xff, 0xff, 0xff,
0xc1, 0xf8, 0x3f, 0xff, 0xff, 0xff, 0xc0, 0x00, 0x3f,
0xff, 0xff, 0xff, 0x80, 0x00, 0x7f, 0xff,

```

					Система моніторингу та відображення ціни крипто валютних акцій за допомогою ESP8266				Лім.		Маса	Масштаб	
Змн.	Арк.	№ докум.	Підпис	Дата									
Розроб.	Мойсюк М. В.												
Перевір.	Букурос О.В.												
Т. Контр.									Арк.		Аркушів 1		
Реценз.									Циклова комісія Комп'ютерної інженерії				
Н. Контр.													
Затверд.													

```

0xff, 0xff, 0x82, 0x00, 0xff, 0xff, 0xff, 0xff, 0x83,
0xc0, 0x7f, 0xff, 0xff, 0xff, 0x87, 0xf0,
0x3f, 0xff, 0xff, 0xff, 0x07, 0xf8, 0x3f, 0xff, 0xff,
0xfb, 0x07, 0xf8, 0x3f, 0xff, 0x7f, 0xf8,
0x07, 0xf0, 0x3f, 0xfe, 0x7f, 0xf0, 0x03, 0xf0, 0x3f,
0xfe, 0x7f, 0xf8, 0x00, 0x00, 0x7f, 0xfe,
0x3f, 0xff, 0x80, 0x00, 0x7f, 0xfc, 0x3f, 0xff, 0x88,
0x00, 0xff, 0xfc, 0x3f, 0xff, 0x9c, 0x43,
0xff, 0xfc, 0x1f, 0xff, 0x18, 0xff, 0xff, 0xf8, 0x0f,
0xff, 0x18, 0xff, 0xff, 0xf8, 0x0f, 0xff,
0xf8, 0xff, 0xff, 0xf0, 0x07, 0xff, 0xfd, 0xff, 0xff,
0xe0, 0x03, 0xff, 0xff, 0xff, 0xff, 0xc0,
0x01, 0xff, 0xff, 0xff, 0xff, 0x80, 0x00, 0xff, 0xff,
0xff, 0xff, 0x00, 0x00, 0x7f, 0xff, 0xff,
0xfe, 0x00, 0x00, 0x3f, 0xff, 0xff, 0xfc, 0x00, 0x00,
0x0f, 0xff, 0xff, 0xf0, 0x00, 0x00, 0x07,
0xff, 0xff, 0xe0, 0x00, 0x00, 0x00, 0xff, 0xff, 0x00,
0x00, 0x00, 0x00, 0x1f, 0xf8, 0x00, 0x00}
},
// ETH setup
{CRYPTO, "ETH",
"https://api.cryptonator.com/api/ticker/eth-
"+String(CURRENCY_CODE),
{0x00, 0x00, 0x07, 0xf0, 0x00, 0x00, 0x00, 0x00, 0xff,
0xff, 0x00, 0x00, 0x00, 0x03, 0xff, 0xff,
0xc0, 0x00, 0x00, 0x0f, 0xfe, 0xff, 0xf0, 0x00, 0x00,
0x1f, 0xfc, 0x7f, 0xfc, 0x00, 0x00, 0x7f,
0xfc, 0x7f, 0xfe, 0x00, 0x00, 0xff, 0xf8, 0x3f, 0xff,
0x00, 0x01, 0xff, 0xf0, 0x1f, 0xff, 0x80,
0x03, 0xff, 0xf0, 0x1f, 0xff, 0xc0, 0x07, 0xff, 0xe0,
0x0f, 0xff, 0xe0, 0x07, 0xff, 0xe0, 0x07,
0xff, 0xf0, 0x0f, 0xff, 0xc0, 0x07, 0xff, 0xf0, 0x1f,
0xff, 0x80, 0x03, 0xff, 0xf8, 0x1f, 0xff,
0x80, 0x03, 0xff, 0xf8, 0x3f, 0xff, 0x00, 0x01, 0xff,
0xfc, 0x3f, 0xff, 0x00, 0x00, 0xff, 0xfc,
0x7f, 0xfe, 0x00, 0x00, 0xff, 0xfe, 0x7f, 0xfc, 0x00,
0x00, 0x7f, 0xfe, 0x7f, 0xfc, 0x00, 0x00,
0x7f, 0xfe, 0x7f, 0xf8, 0x00, 0x00, 0x3f, 0xfe, 0xff,
0xf8, 0x00, 0x00, 0x3f, 0xff, 0xff, 0xf0,
0x00, 0x00, 0x1f, 0xff, 0xff, 0xe0, 0x00, 0x00, 0x0f,
0xff, 0xff, 0xe0, 0x00, 0x00, 0x0f, 0xff,

```

					Система моніторингу та відображення ціни крипто валютних акцій за допомогою ESP8266				Лім.		Маса	Масштаб	
Змн.	Арк.	№ докум.	Підпис	Дата									
Розроб.	Мойсюк М. В.												
Перевір.	Букурос О.В.												
Т. Контр.									Арк.		Аркушів 1		
Реценз.									Циклова комісія Комп'ютерної інженерії				
Н. Контр.													
Затверд.													

```

0xff, 0xe0, 0x00, 0x00, 0x0f, 0xff, 0xff, 0xf8, 0x00,
0x00, 0x3f, 0xff, 0xff, 0xfe, 0x00, 0x00,
0x7f, 0xff, 0x7f, 0xe7, 0x00, 0x01, 0xcf, 0xff, 0x7f,
0xf3, 0xc0, 0x07, 0x9f, 0xff, 0x7f, 0xf8,
0xf0, 0x0e, 0x3f, 0xfe, 0x7f, 0xf8, 0x38, 0x3c, 0x3f,
0xfe, 0x7f, 0xfc, 0x1e, 0xf0, 0x7f, 0xfe,
0x3f, 0xfe, 0x07, 0xc0, 0xff, 0xfe, 0x3f, 0xff, 0x03,
0x80, 0xff, 0xfc, 0x1f, 0xff, 0x00, 0x01,
0xff, 0xfc, 0x1f, 0xff, 0x80, 0x03, 0xff, 0xf8, 0x0f,
0xff, 0xc0, 0x07, 0xff, 0xf0, 0x0f, 0xff,
0xc0, 0x07, 0xff, 0xf0, 0x07, 0xff, 0xe0, 0x0f, 0xff,
0xe0, 0x03, 0xff, 0xf0, 0x1f, 0xff, 0xc0,
0x01, 0xff, 0xf0, 0x1f, 0xff, 0xc0, 0x00, 0xff, 0xf8,
0x3f, 0xff, 0x00, 0x00, 0x7f, 0xfc, 0x7f,
0xfe, 0x00, 0x00, 0x3f, 0xfe, 0x7f, 0xfc, 0x00, 0x00,
0x0f, 0xff, 0xff, 0xf8, 0x00, 0x00, 0x07,
0xff, 0xff, 0xe0, 0x00, 0x00, 0x00, 0xff, 0xff, 0x80,
0x00, 0x00, 0x00, 0x1f, 0xf0, 0x00, 0x00}
},
// XLM setup
{CRYPTO, "Stellar",
"https://api.cryptonator.com/api/ticker/xlm-
"+String(CURRENCY_CODE),
{0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x03, 0xc0, 0x00, 0x00, 0x00,
0x00, 0x3f, 0xfc, 0x00, 0x00, 0x00, 0x00,
0xff, 0xff, 0x00, 0x00, 0x00, 0x03, 0xff, 0xff, 0xc0,
0x03, 0x00, 0x0f, 0xfc, 0x3f, 0xf0, 0x0f,
0x00, 0x1f, 0xc0, 0x03, 0xc0, 0x3f, 0x00, 0x3f, 0x00,
0x00, 0x00, 0xff, 0x00, 0x7c, 0x00, 0x00,
0x03, 0xfc, 0x00, 0xf8, 0x00, 0x00, 0x0f, 0xf0, 0x01,
0xf0, 0x00, 0x00, 0x3f, 0xe0, 0x01, 0xe0,
0x00, 0x00, 0xff, 0x80, 0x03, 0xc0, 0x00, 0x03, 0xfe,
0x01, 0x03, 0xc0, 0x00, 0x07, 0xf8, 0x07,
0x07, 0x80, 0x00, 0x3f, 0xe0, 0x1f, 0x07, 0x80, 0x00,
0x7f, 0x80, 0x7f, 0x0f, 0x00, 0x01, 0xfe,
0x01, 0xfe, 0x0f, 0x00, 0x07, 0xf8, 0x07, 0xf8, 0x0f,
0x00, 0x1f, 0xe0, 0x1f, 0xf8, 0x0f, 0x00,
0x7f, 0x80, 0x7f, 0xf0, 0x0e, 0x01, 0xfe, 0x01, 0xfe,
0x70, 0x0e, 0x07, 0xf8, 0x07, 0xf8, 0x70,

```

					Система моніторингу та відображення ціни крипто валютних акцій за допомогою ESP8266					Лім.		Маса		Масштаб		
Змн.	Арк.	№ докум.	Підпис	Дата												
Розроб.		Мойсюк М. В.														
Перевір.		Букурос О.В.														
Т. Контр.																
Реценз.										Арк.		Аркушів		1		
Н. Контр.										Циклова комісія Комп'ютерної інженерії						
Затверд.																

```

0x0e, 0x1f, 0xe0, 0x1f, 0xe0, 0x70, 0x0e, 0x7f, 0x80,
0x7f, 0x80, 0x70, 0x0f, 0xfe, 0x01, 0xfe,
0x00, 0xf0, 0x1f, 0xf8, 0x07, 0xf8, 0x00, 0xf0, 0x1f,
0xe0, 0x1f, 0xe0, 0x00, 0xf0, 0x7f, 0x80,
0x7f, 0x80, 0x00, 0xf0, 0xfe, 0x01, 0xfe, 0x00, 0x01,
0xe0, 0xf8, 0x07, 0xfc, 0x00, 0x01, 0xe0,
0xe0, 0x1f, 0xe0, 0x00, 0x03, 0xc0, 0x80, 0x7f, 0xc0,
0x00, 0x03, 0xc0, 0x01, 0xff, 0x00, 0x00,
0x07, 0x80, 0x07, 0xfc, 0x00, 0x00, 0x0f, 0x80, 0x0f,
0xf0, 0x00, 0x00, 0x1f, 0x00, 0x3f, 0xc0,
0x00, 0x00, 0x3e, 0x00, 0xff, 0x00, 0x00, 0x00, 0xfc,
0x00, 0xfc, 0x03, 0xc0, 0x03, 0xf8, 0x00,
0xf0, 0x0f, 0xfc, 0x3f, 0xf0, 0x00, 0xc0, 0x03, 0xff,
0xff, 0xc0, 0x00, 0x00, 0x00, 0xff, 0xff,
0x00, 0x00, 0x00, 0x00, 0x3f, 0xfc, 0x00, 0x00, 0x00,
0x00, 0x03, 0xc0, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00}
},
// INTC setup
{STOCK, "Intel",

```

```

"https://finnhub.io/api/v1/quote?symbol=INTC&token="+stocksApiKey,

```

```

{0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x0f, 0xf0, 0x00, 0x07, 0x80, 0x00, 0x0f,
0xf0, 0x00, 0x07, 0x80, 0x00, 0x0f, 0xf0, 0x00, 0x07,
0x80, 0x00, 0x0f, 0x00, 0x00, 0x07, 0x80,
0x00, 0x0f, 0x00, 0x00, 0x07, 0x80, 0x00, 0x0f, 0xf3,
0x9f, 0x07, 0xe0, 0x7c, 0x0f, 0xf3, 0xbf,

```

					Система моніторингу та відображення ціни крипто валютних акцій за допомогою ESP8266	Лім.			Маса	Масштаб
Змн.	Арк.	№ докум.	Підпис	Дата						
Розроб.	Мойсюк М. В.									
Перевір.	Букурос О.В.									
Т. Контр.						Арк.			Аркушів 1	
Реценз.						Циклова комісія Комп'ютерної інженерії				
Н. Контр.										
Затверд.										


```

0x7f, 0xff, 0xff, 0xff, 0xff, 0xfe, 0x7f, 0xff, 0xff,
0xff, 0xff, 0xfe, 0x7f, 0xff, 0xff, 0xff,
0xff, 0xfe, 0xff, 0xe0, 0xe1, 0xc3, 0xf0, 0x7f, 0xff,
0x80, 0xc0, 0x01, 0xc0, 0x3f, 0xfe, 0x00,
0x80, 0x00, 0xc6, 0x1f, 0xfc, 0x3c, 0x1e, 0x3c, 0x8f,
0x9f, 0xfc, 0x7c, 0x1e, 0x3c, 0x9f, 0xff,
0xf8, 0xfc, 0x1e, 0x3c, 0x9f, 0xff, 0xf9, 0xfc, 0x1e,
0x3c, 0x8f, 0x9f, 0xf0, 0x0c, 0x1e, 0x3c,
0xc6, 0x1f, 0xf0, 0x04, 0x1e, 0x3c, 0xc0, 0x3f, 0xf0,
0x0c, 0x9f, 0x7c, 0xf0, 0x7f, 0x7f, 0xff,
0xff, 0xff, 0xff, 0xfe, 0x7f, 0xff, 0xff, 0xff, 0xff,
0xfe, 0x7f, 0xff, 0xff, 0xff, 0xff, 0xfe,
0x3f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xfc, 0x3f, 0xff, 0xff,
0xff, 0xff, 0xfc, 0x3f, 0xff, 0xff, 0xff,
0xff, 0xfc, 0x1f, 0xff, 0xff, 0xff, 0xff, 0xf8, 0x0f,
0xff, 0xff, 0xff, 0xff, 0xf0, 0x0f, 0xff,
0xff, 0xff, 0xff, 0xf0, 0x07, 0xff, 0xff, 0xff, 0xff,
0xe0, 0x03, 0xff, 0xff, 0xff, 0xff, 0xc0,
0x01, 0xff, 0xff, 0xff, 0xff, 0x80, 0x00, 0xff, 0xff,
0xff, 0xff, 0x00, 0x00, 0x7f, 0xff, 0xff,
0xfe, 0x00, 0x00, 0x3f, 0xff, 0xff, 0xfc, 0x00, 0x00,
0x0f, 0xff, 0xff, 0xf8, 0x00, 0x00, 0x07,
0xff, 0xff, 0xe0, 0x00, 0x00, 0x00, 0xff, 0xff, 0x00,
0x00, 0x00, 0x00, 0x1f, 0xf8, 0x00, 0x00}
},
// Apple setup
{STOCK, "Apple",

```

```

"https://finnhub.io/api/v1/quote?symbol=AAPL&token="+stocksApiKey,

```

```

{0x00, 0x00, 0x00, 0x01, 0x80, 0x00, 0x00, 0x00, 0x00,
0x07, 0x80, 0x00, 0x00, 0x00, 0x00, 0x0f,
0x80, 0x00, 0x00, 0x00, 0x00, 0x1f, 0x80, 0x00, 0x00,
0x00, 0x00, 0x3f, 0x80, 0x00, 0x00, 0x00,
0x00, 0x7f, 0x00, 0x00, 0x00, 0x00, 0x00, 0x7f, 0x00,
0x00, 0x00, 0x00, 0x00, 0xfe, 0x00, 0x00,
0x00, 0x00, 0x00, 0xfc, 0x00, 0x00, 0x00, 0x00, 0x00,
0xf8, 0x00, 0x00, 0x00, 0x00, 0x00, 0xf0,
0x00, 0x00, 0x00, 0x0f, 0xe0, 0x83, 0xf8, 0x00, 0x00,
0x3f, 0xfc, 0x1f, 0xfe, 0x00, 0x00, 0x7f,

```

					Система моніторингу та відображення ціни крипто валютних акцій за допомогою ESP8266	Лім.			Маса	Масштаб
Змн.	Арк.	№ докум.	Підпис	Дата						
Розроб.		Мойсюк М. В.								
Перевір.		Букурос О.В.								
Т. Контр.						Арк.			Аркушів 1	
Реценз.						Циклова комісія Комп'ютерної інженерії				
Н. Контр.										
Затверд.										

```

0xff, 0xff, 0xff, 0x80, 0x00, 0xff, 0xff, 0xff, 0xff,
0xc0, 0x01, 0xff, 0xff, 0xff, 0xff, 0xc0,
0x03, 0xff, 0xff, 0xff, 0xff, 0xc0, 0x03, 0xff, 0xff,
0xff, 0xff, 0x80, 0x07, 0xff, 0xff, 0xff,
0xff, 0x00, 0x07, 0xff, 0xff, 0xff, 0xfe, 0x00, 0x07,
0xff, 0xff, 0xff, 0xfe, 0x00, 0x0f, 0xff,
0xff, 0xff, 0xfc, 0x00, 0x0f, 0xff, 0xff, 0xff, 0xfc,
0x00, 0x0f, 0xff, 0xff, 0xff, 0xfc, 0x00,
0x0f, 0xff, 0xff, 0xff, 0xfc, 0x00, 0x0f, 0xff, 0xff,
0xff, 0xfc, 0x00, 0x0f, 0xff, 0xff, 0xff,
0xfc, 0x00, 0x0f, 0xff, 0xff, 0xff, 0xfc, 0x00, 0x0f,
0xff, 0xff, 0xff, 0xfc, 0x00, 0x0f, 0xff,
0xff, 0xff, 0xfc, 0x00, 0x0f, 0xff, 0xff, 0xff, 0xfe,
0x00, 0x07, 0xff, 0xff, 0xff, 0xff, 0x00,
0x07, 0xff, 0xff, 0xff, 0xff, 0x00, 0x07, 0xff, 0xff,
0xff, 0xff, 0x80, 0x07, 0xff, 0xff, 0xff,
0xff, 0xe0, 0x03, 0xff, 0xff, 0xff, 0xff, 0xf0, 0x03,
0xff, 0xff, 0xff, 0xff, 0xf0, 0x01, 0xff,
0xff, 0xff, 0xff, 0xe0, 0x01, 0xff, 0xff, 0xff, 0xff,
0xe0, 0x00, 0xff, 0xff, 0xff, 0xff, 0xc0,
0x00, 0xff, 0xff, 0xff, 0xff, 0xc0, 0x00, 0x7f, 0xff,
0xff, 0xff, 0x80, 0x00, 0x7f, 0xff, 0xff,
0xff, 0x00, 0x00, 0x3f, 0xff, 0xff, 0xff, 0x00, 0x00,
0x1f, 0xff, 0xff, 0xfe, 0x00, 0x00, 0x0f,
0xff, 0xff, 0xfc, 0x00, 0x00, 0x07, 0xfc, 0x0f, 0xf8,
0x00, 0x00, 0x01, 0xe0, 0x01, 0xe0, 0x00}
},
// Tesla setup
{STOCK, "Tesla",

```

```

"https://finnhub.io/api/v1/quote?symbol=TESLA&token="+stocksApiKey,

```

```

{0x00, 0x00, 0x7f, 0xfe, 0x00, 0x00, 0x00, 0x1f, 0xff,
0xff, 0xf8, 0x00, 0x01, 0xff, 0xff, 0xff,
0xff, 0x80, 0x0f, 0xff, 0x00, 0x00, 0xff, 0xf0, 0x7f,
0xc0, 0x00, 0x00, 0x03, 0xfe, 0xfe, 0x03,
0xf0, 0x0f, 0xc0, 0x7f, 0x70, 0x7f, 0xf0, 0x0f, 0xfe,
0x0e, 0x03, 0xff, 0xf8, 0x1f, 0xff, 0xc0,
0x1f, 0xff, 0xfc, 0x3f, 0xff, 0xf8, 0x3f, 0xff, 0xfe,
0x7f, 0xff, 0xfc, 0x1f, 0xf8, 0x7e, 0x7e,

```

					Система моніторингу та відображення ціни крипто валютних акцій за допомогою ESP8266				Лім.		Маса	Масштаб	
Змн.	Арк.	№ докум.	Підпис	Дата									
Розроб.	Мойсюк М. В.												
Перевір.	Букурос О.В.												
Т. Контр.									Арк.		Аркушів 1		
Реценз.									Циклова комісія Комп'ютерної інженерії				
Н. Контр.													
Затверд.													

```

0x1f, 0xf8, 0x0f, 0xc0, 0x7f, 0xfc, 0x03, 0xf0, 0x03,
0x80, 0x3f, 0xfc, 0x01, 0xc0, 0x01, 0x00,
0x3f, 0xfc, 0x00, 0x80, 0x00, 0x00, 0x3f, 0xfc, 0x00,
0x00, 0x00, 0x00, 0x3f, 0xfc, 0x00, 0x00,
0x00, 0x00, 0x3f, 0xfc, 0x00, 0x00, 0x00, 0x00, 0x1f,
0xf8, 0x00, 0x00, 0x00, 0x00, 0x1f, 0xf8,
0x00, 0x00, 0x00, 0x00, 0x1f, 0xf8, 0x00, 0x00, 0x00,
0x00, 0x1f, 0xf8, 0x00, 0x00, 0x00, 0x00,
0x1f, 0xf8, 0x00, 0x00, 0x00, 0x00, 0x1f, 0xf8, 0x00,
0x00, 0x00, 0x00, 0x0f, 0xf0, 0x00, 0x00,
0x00, 0x00, 0x0f, 0xf0, 0x00, 0x00, 0x00, 0x00, 0x0f,
0xf0, 0x00, 0x00, 0x00, 0x00, 0x0f, 0xf0,
0x00, 0x00, 0x00, 0x00, 0x0f, 0xf0, 0x00, 0x00, 0x00,
0x00, 0x07, 0xe0, 0x00, 0x00, 0x00, 0x00,
0x07, 0xe0, 0x00, 0x00, 0x00, 0x00, 0x07, 0xe0, 0x00,
0x00, 0x00, 0x00, 0x07, 0xe0, 0x00, 0x00,
0x00, 0x00, 0x07, 0xe0, 0x00, 0x00, 0x00, 0x00, 0x07,
0xe0, 0x00, 0x00, 0x00, 0x00, 0x03, 0xc0,
0x00, 0x00, 0x00, 0x00, 0x03, 0xc0, 0x00, 0x00, 0x00,
0x00, 0x03, 0xc0, 0x00, 0x00, 0x00, 0x00,
0x03, 0xc0, 0x00, 0x00, 0x00, 0x00, 0x03, 0xc0, 0x00,
0x00, 0x00, 0x00, 0x03, 0xc0, 0x00, 0x00,
0x00, 0x00, 0x01, 0x80, 0x00, 0x00, 0x00, 0x00, 0x01,
0x80, 0x00, 0x00, 0x00, 0x00, 0x01, 0x80, 0x00, 0x00, 0x00,
0x00, 0x01, 0x80, 0x00, 0x00, 0x00, 0x00,
0x01, 0x80, 0x00, 0x00, 0x00, 0x00, 0x01, 0x80, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00}
}
};

// Stonks and not stonks logos are both 70x30
#define STONKS_WIDTH 70
#define STONKS_HEIGHT 30

```

					Система моніторингу та відображення ціни крипто валютних акцій за допомогою ESP8266	Лім.			Маса	Масштаб
Змн.	Арк.	№ докум.	Підпис	Дата	Циклова комісія Комп'ютерної інженерії	Арк.			Аркушів 1	
Розроб.		Мойсюк М. В.								
Перевір.		Букурос О.В.								
Т. Контр.										
Реценз.										
Н. Контр.										
Затверд.										

```
// Stonks symbol as bitmap
const unsigned char stonks [] PROGMEM = {
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x06, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x07,
    0xfc, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x03, 0xff,
    0xf0, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x01, 0xff, 0xfc, 0x00, 0x00, 0x00, 0x01, 0x80, 0x00,
    0x01, 0xff, 0xf8, 0x00, 0x00, 0x00,
    0x07, 0x80, 0x00, 0x00, 0xff, 0xf8, 0x00, 0x00, 0x00, 0x0f,
    0xc0, 0x00, 0x00, 0xff, 0xf8, 0x00,
    0x00, 0x00, 0x3f, 0xc0, 0x00, 0x01, 0xff, 0xf0, 0x00, 0x00,
    0x00, 0xff, 0xe0, 0x00, 0x07, 0xff,
    0xf0, 0x00, 0x00, 0x01, 0xff, 0xf0, 0x00, 0x1f, 0xff, 0xf0,
    0x00, 0x00, 0x07, 0xff, 0xf0, 0x00,
    0x3f, 0xff, 0xe0, 0x00, 0x00, 0x1f, 0xff, 0xf8, 0x00, 0xff,
    0xff, 0xe0, 0x00, 0x00, 0x3f, 0xff,
    0xf8, 0x03, 0xff, 0xff, 0xc0, 0x00, 0x00, 0xff, 0xff, 0xfc,
    0x07, 0xff, 0xe7, 0xc0, 0x00, 0x03,
    0xff, 0xff, 0xfe, 0x1f, 0xff, 0xc7, 0xc0, 0x00, 0x07, 0xff,
    0xf3, 0xfe, 0x7f, 0xff, 0x03, 0x80,
    0x00, 0x1f, 0xff, 0xc1, 0xff, 0xff, 0xfe, 0x03, 0x80, 0x00,
    0x7f, 0xff, 0x01, 0xff, 0xff, 0xf8,
    0x01, 0x80, 0x00, 0xff, 0xfe, 0x00, 0xff, 0xff, 0xe0, 0x00,
    0x00, 0x03, 0xff, 0xf8, 0x00, 0xff,
    0xff, 0xc0, 0x00, 0x00, 0x0f, 0xff, 0xe0, 0x00, 0x7f, 0xff,
    0x00, 0x00, 0x00, 0x1f, 0xff, 0xc0,
    0x00, 0x3f, 0xfc, 0x00, 0x00, 0x00, 0x7f, 0xff, 0x00, 0x00,
    0x3f, 0xf8, 0x00, 0x00, 0x00, 0xff,
    0xfc, 0x00, 0x00, 0x1f, 0xe0, 0x00, 0x00, 0x00, 0x7f, 0xf8,
    0x00, 0x00, 0x1f, 0x80, 0x00, 0x00,
    0x00, 0x3f, 0xe0, 0x00, 0x00, 0x0f, 0x00, 0x00, 0x00, 0x00,
    0x3f, 0x80, 0x00, 0x00, 0x04, 0x00,
    0x00, 0x00, 0x00, 0x1f, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x1c, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x1c, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00
};
```

Змн.	Арк.	№ докум.	Підпис	Дата	Система моніторингу та відображення ціни крипто валютних акцій за допомогою ESP8266	Лім.			Маса		Масштаб	
Розроб.		Мойсюк М. В.										
Перевір.		Букурос О.В.										
Т. Контр.						Арк.			Аркушів 1			
Реценз.						Циклова комісія Комп'ютерної інженерії						
Н. Контр.												
Затверд.												

```
// Not stonks symbol as bitmap
const unsigned char notstonks [] PROGMEM = {
    0x08, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x1c,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x1f, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x3f, 0xc0, 0x00, 0x00, 0x0c,
    0x00, 0x00, 0x00, 0x00, 0x7f, 0xf0, 0x00, 0x00, 0x0f, 0x00,
    0x00, 0x00, 0x00, 0x7f, 0xf8, 0x00,
    0x00, 0x1f, 0xc0, 0x00, 0x00, 0x00, 0xff, 0xfe, 0x00, 0x00,
    0x1f, 0xe0, 0x00, 0x00, 0x00, 0x7f,
    0xff, 0x80, 0x00, 0x3f, 0xf8, 0x00, 0x00, 0x00, 0x1f, 0xff,
    0xc0, 0x00, 0x7f, 0xfe, 0x00, 0x00,
    0x00, 0x07, 0xff, 0xf0, 0x00, 0x7f, 0xff, 0x00, 0x00, 0x00,
    0x03, 0xff, 0xfc, 0x00, 0xff, 0xff,
    0xc0, 0x00, 0x00, 0x00, 0xff, 0xfe, 0x00, 0xff, 0xff, 0xf0,
    0x01, 0x80, 0x00, 0x3f, 0xff, 0x81,
    0xff, 0xff, 0xf8, 0x01, 0x80, 0x00, 0x1f, 0xff, 0xe3, 0xff,
    0xff, 0xfe, 0x03, 0x80, 0x00, 0x07,
    0xff, 0xf3, 0xfe, 0x3f, 0xff, 0x83, 0xc0, 0x00, 0x01, 0xff,
    0xff, 0xfc, 0x1f, 0xff, 0xc7, 0xc0,
    0x00, 0x00, 0xff, 0xff, 0xfc, 0x07, 0xff, 0xff, 0xc0, 0x00,
    0x00, 0x3f, 0xff, 0xf8, 0x01, 0xff,
    0xff, 0xe0, 0x00, 0x00, 0x0f, 0xff, 0xf8, 0x00, 0xff, 0xff,
    0xe0, 0x00, 0x00, 0x03, 0xff, 0xf0,
    0x00, 0x3f, 0xff, 0xe0, 0x00, 0x00, 0x01, 0xff, 0xe0, 0x00,
    0x0f, 0xff, 0xf0, 0x00, 0x00, 0x00,
    0x7f, 0xe0, 0x00, 0x07, 0xff, 0xf0, 0x00, 0x00, 0x00, 0x1f,
    0xc0, 0x00, 0x01, 0xff, 0xf0, 0x00,
    0x00, 0x00, 0x0f, 0xc0, 0x00, 0x00, 0xff, 0xf8, 0x00, 0x00,
    0x00, 0x03, 0x80, 0x00, 0x00, 0xff,
    0xf8, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0xff, 0xfc,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x03, 0xff, 0xfc, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x03,
    0xff, 0xc0, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x07, 0xe0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x0f, 0xc0, 0x00
};
```

																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																				</
--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	----

```

// Variable to store what asset should be displayed
int currentAsset = 0;
// Const to store how many assets are configured
const int maxAssets = sizeof(assets)/sizeof(assets[0]);

void setup() {

    Serial.begin(115200);

    #ifdef ESP01
        // I2C for ESP-01 is different
        Wire.begin(2, 0);
    #endif

    // Try to start display if that doesn't work halt program and
    inform user
    if(!display.begin(SSD1306_SWITCHCAPVCC, SCREEN_ADDRESS)){
        Serial.println(F("HALT: OLED allocation failed"));
        while(1);
    }

    // Immediately clear display to remove Adafruit logo
    display.clearDisplay();
    // Set the text color to white this will turn on pixels when
    set
    display.setTextColor(SSD1306_WHITE);
    // Set starting text size to 1 for connecting message
    display.setTextSize(1);

```

Змн.	Арк.	№ докум.	Підпис	Дата	Система моніторингу та відображення ціни крипто валютних акцій за допомогою ESP8266	Лім.			Маса		Масштаб	
Розроб.		Мойсюк М. В.										
Перевір.		Букурос О.В.										
Т. Контр.						Арк.			Аркушів 1			
Реценз.												
Н. Контр.						Циклова комісія Комп'ютерної інженерії						
Затверд.												

```

// Allow esp wifi time to setup before attempting connection
for (uint8_t t = SETUP_TIME; t > 0; t--) {
    // Print info to serial
    Serial.printf("[SETUP] WAIT %d...\n", t);
    Serial.flush();
    // and print info to OLED
    display.clearDisplay();
    display.setCursor(0, display.height()/4);
    display.print(F("Setup please wait . . . "));
    display.print(t);
    display.display();
    delay(1000);
}

// Use font size 2 for the rest of the program
display.setTextSize(2);

// Add Wifi network to ESP
WiFi.mode(WIFI_STA);
WiFiMulti.addAP(WIFI_SSID, WIFI_PASS);
}

void loop() {
    // Wait for WiFi connection
    if ((WiFiMulti.run() == WL_CONNECTED)) {
        // Set the correct fingerprint
        uint8_t fingerprint[20];
        if(assets[currentAsset].isCrypto)
            // In C a memcpy needs to be used as an array can't be
            // directly assigned to another array
            memcpy(fingerprint, fingerprint_crypto, 20);
        else
            memcpy(fingerprint, fingerprint_stock, 20);

        // Setup a https client
        std::unique_ptr<BearSSL::WiFiClientSecure>client(new
        BearSSL::WiFiClientSecure);
        client->setFingerprint(fingerprint);
        HTTPClient https;
    }
}

```



```

// Connect to API
Serial.print("HTTPS begin...\n");
if (https.begin(*client, assets[currentAsset].url)) { //
HTTPS

    // start connection and send HTTP header
    Serial.print("HTTPS GET...\n");
    int httpCode = https.GET();

    // httpCode will be negative on error
    if (httpCode > 0) {
        // HTTP header has been send and Server response header
        has been handled
        Serial.printf("HTTPS GET... code: %d\n", httpCode);

        // If the HTTP code is valid process and display the
        data
        if (httpCode == HTTP_CODE_OK || httpCode ==
        HTTP_CODE_MOVED_PERMANENTLY) {
            // Store the API payload in a string
            String payload = https.getString();
            // Calculate the price, change and percentage change
            float price;
            float change;
            if(assets[currentAsset].isCrypto){
                price =
                payload.substring(payload.indexOf("price")+8,payload.indexOf("vo
                lume")-3).toFloat();
                change =
                payload.substring(payload.indexOf("change")+9,payload.indexOf("t
                imESTAMP")-4).toFloat();
            }
            else{
                price =
                payload.substring(payload.indexOf("c")+3,payload.indexOf("h")-
                2).toFloat();
                change = price -
                payload.substring(payload.indexOf("pc")+4,payload.indexOf("times
                tamp")-2).toFloat();
            }

```


```

float changePercent = change/price*100;
// Print payload to serial
Serial.println(assets[currentAsset].assetName);
Serial.println(payload);
// Print info to display
display.clearDisplay();
drawLogo(assets[currentAsset].logo);
drawStonks(change);
printPrice(price);
printChange(changePercent);

// Move to the next coin
currentAsset++;
if(currentAsset>=maxAssets){
    currentAsset = 0;
}
} else {
    // Warn user using serial and OLED that call failed
    Serial.printf("[HTTPS] GET... failed, error: %s\n",
https.errorToString(httpCode).c_str());
    display.clearDisplay();
    display.setTextSize(1);
    display.setCursor(0, display.height()/4);
    display.println(F("HTTPS GET failed"));
    display.print(F("Will retry in a few seconds"));
    display.setTextSize(2);
}

// Must close the https connection
https.end();
}

```


```

else {
    // Warn user using serial and OLED that call failed
    Serial.println("Unable to connect to API");
    display.clearDisplay();
    display.setTextSize(1);
    display.setCursor(0, display.height()/4);
    display.println(F("Unable to connect"));
    display.print(F("Will retry in a few seconds"));
    display.setTextSize(2);
}
}

Serial.print("Wait a few seconds before next round...\n\n");
delay(UPDATE_TIME*1000);
}

// Draw the asset logo on the OLED
void drawLogo(const unsigned char logo []) {
    display.drawBitmap(0,
        (display.height() + (display.height()/4) - LOGO_HEIGHT) / 2,
        logo, LOGO_WIDTH, LOGO_HEIGHT, 1);
    display.display();
}

// Draw the stonks or not stonks logo on the OLED
void drawStonks(float change) {
    if(change>=0){
        display.drawBitmap(display.width()-STONKS_WIDTH,
            display.height()-STONKS_HEIGHT, stonks, STONKS_WIDTH,
            STONKS_HEIGHT, 1);
    }
    else{
        display.drawBitmap(display.width()-STONKS_WIDTH,
            display.height()-STONKS_HEIGHT, notstonks, STONKS_WIDTH,
            STONKS_HEIGHT, 1);
    }
    display.display();
}

```

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

```

// Print the price on the OLED
void printPrice(float price){
    char priceBuf[20];
    // Get correct currency symbol
    char currencySymbol = '$';
    if(assets[currentAsset].isCrypto)
        currencySymbol = CURRENCY_SYM;
    // Construct the price string
    if(price<10){
        sprintf(priceBuf, "%c%.5f", currencySymbol, price);
    }
    else {
        sprintf(priceBuf, "%c%.3f", currencySymbol, price);
    }
    // To center the price we need to get the size it will be on
the OLED
    int16_t x1, y1;
    uint16_t w, h;
    display.getTextBounds(priceBuf, 0, 0, &x1, &y1, &w, &h);
    // Set the cursor to the starting location that will center
the text
    display.setCursor((display.width() - w) / 2, 0);
    // Print the string to the OLED
    display.print(priceBuf);
    display.display();
}

// Print the price change percentage to the OLED
void printChange(float change) {
    display.setCursor(display.width()-STONKS_WIDTH,
display.height()/4);
    // Manually print a + for positive changes
    if(change>=0){
        display.print(F("+"));
    }
    // If the change is less than 10 use a decimal point if not
dont
    if(abs(change)<10){
        display.print(change,1);
    }
}

```

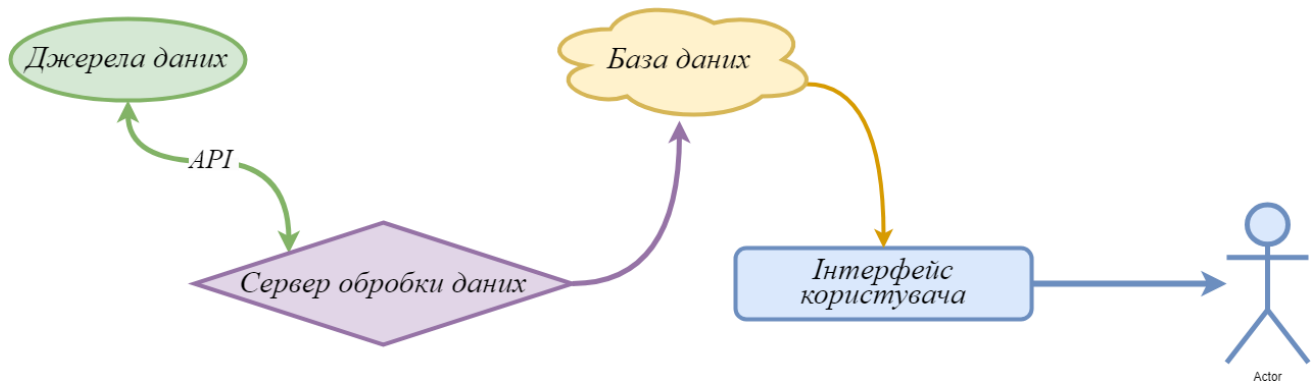

```

else{
    display.print(change,0);
}
display.print(F("%"));
display.display();
}

```

					Система моніторингу та відображення ціни крипто валютних акцій за допомогою ESP8266	Лім.			Маса		Масштаб	
Змн.	Арк.	№ докум.	Підпис	Дата								
Розроб.		Мойсюк М. В.										
Перевір.		Букурос О.В.										
Т. Контр.												
Реценз.						Арк.			Аркушів			1
Н. Контр.						Циклова комісія Комп'ютерної інженерії						
Затверд.												

Архітектура систем моніторингу криптовалют



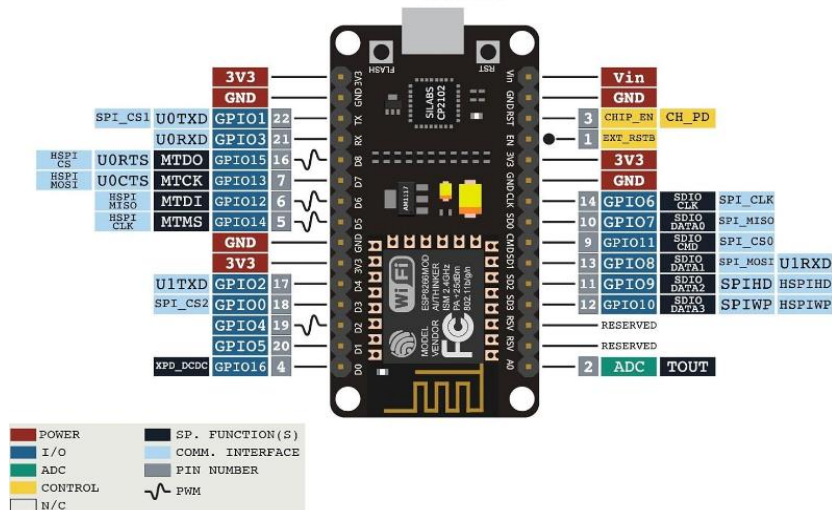
					Система моніторингу та відображення ціни крипто валютних акцій за допомогою ESP8266					Лім.		Маса		Масштаб		
Змн.	Арк.	№ докум.	Підпис	Дата												
Розроб.		Мойсюк М. В.														
Перевір.		Букурос О.В.														
Т. Контр.																
Реценз.								Арк.		Аркушів		1				
Н. Контр.								Циклова комісія Комп'ютерної інженерії								
Затверд.																

Мікроконтролер ESP8266



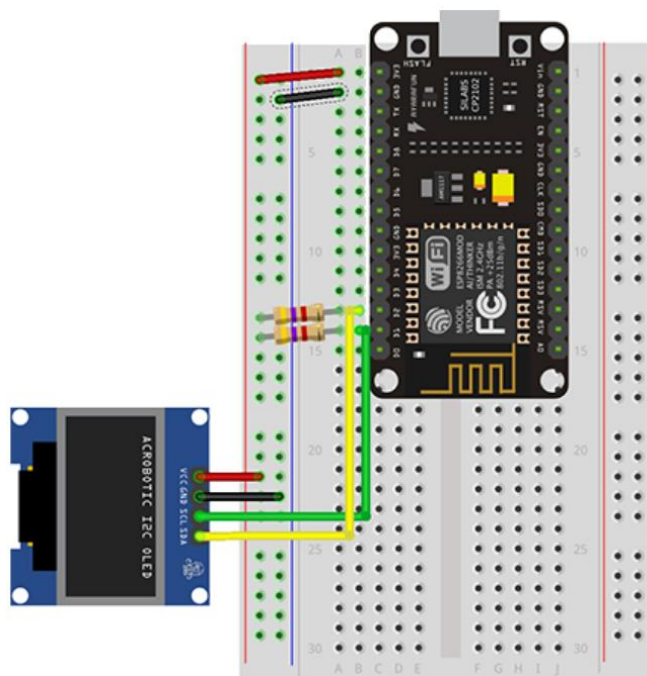
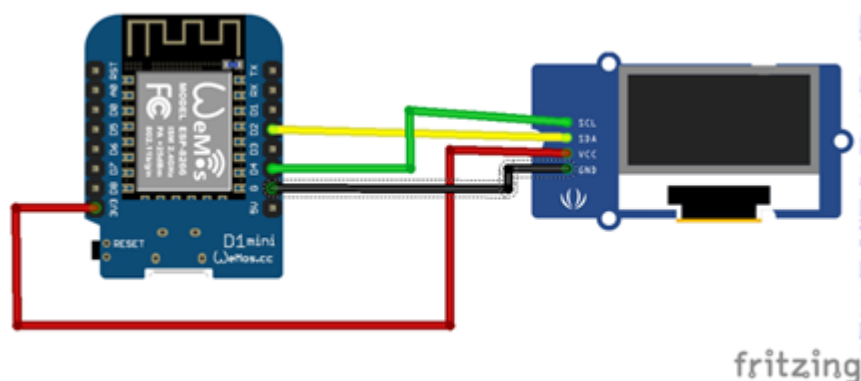
NOTES:

- ▲ Typ. pin current 6mA (Max. 12mA)
- ▲ For sleep mode, connect GPIO16 and EXT_RSTB. On wakeup, GPIO16 will output LOW for system reset.
- ▲ On boot/reset/wakeup, keep GPIO15 LOW and GPIO2 HIGH.



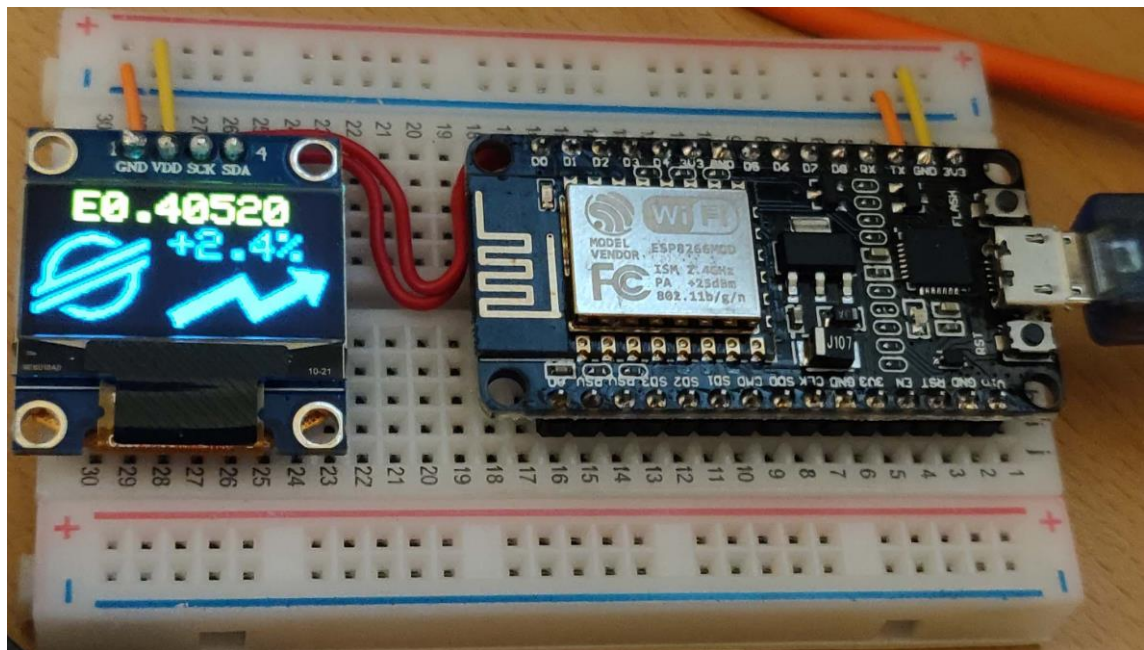
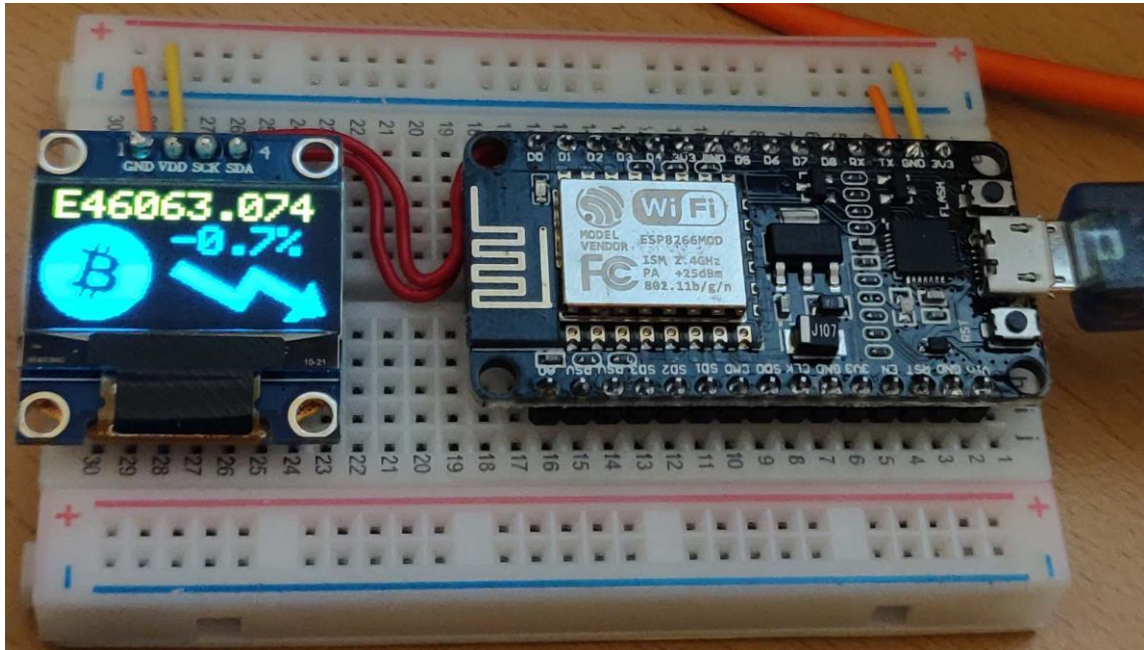
					Система моніторингу та відображення ціни крипто валютних акцій за допомогою ESP8266	Лім.			Маса	Масштаб
Змн.	Арк.	№ докум.	Підпис	Дата						
Розроб.		Мойсюк М. В.								
Перевір.		Букурос О.В.								
Т. Контр.						Арк.			Аркушів 1	
Реценз.					Циклова комісія Комп'ютерної інженерії					
Н. Контр.										
Затверд.										

Макетна схема Трекера криптовалюти

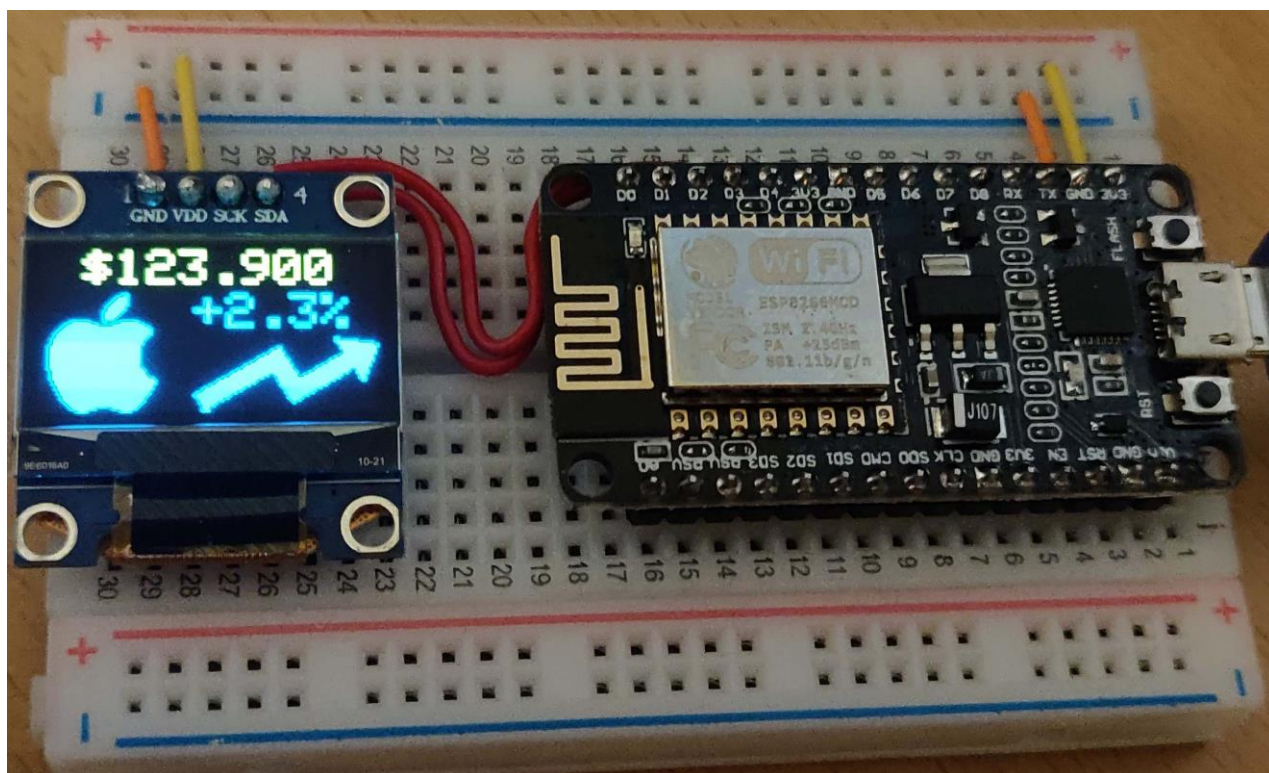
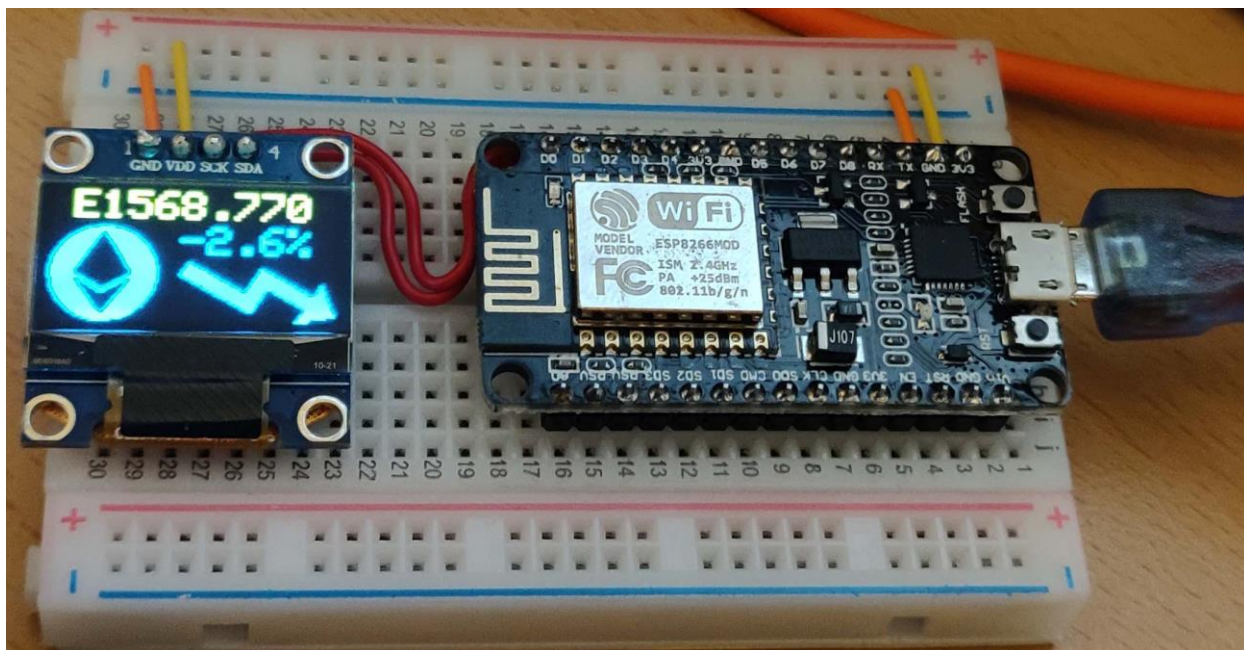


					Система моніторингу та відображення ціни крипто валютних акцій за допомогою ESP8266	Лім.	Маса	Масштаб
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Мойсюк М. В.						
Перевір.		Букурос О.В.						
Т. Контр.						Арк.	Аркушів	1
Реценз.						Циклова комісія Комп'ютерної інженерії		
Н. Контр.								
Затверд.								

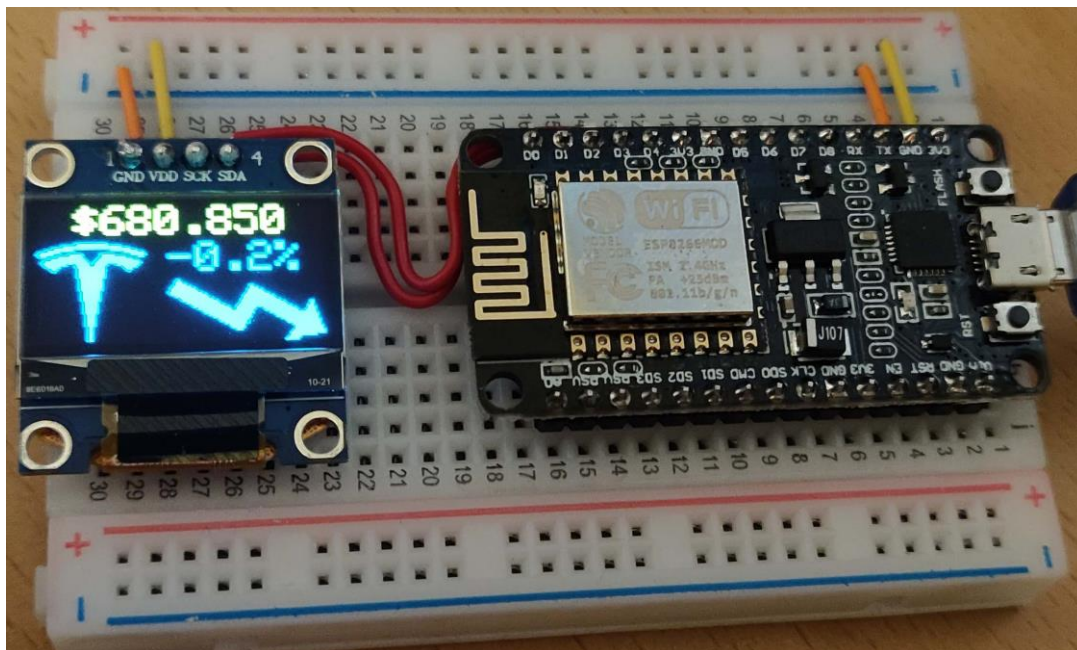
Демонстраційні зображення роботи системи



					Система моніторингу та відображення ціни крипто валютних акцій за допомогою ESP8266	Лім.	Маса	Масштаб
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Мойсюк М. В.						
Перевір.		Букурис О.В.						
Т. Контр.						Арк.	Аркушів	1
Реценз.					Циклова комісія Комп'ютерної інженерії			
Н. Контр.								
Затверд.								



					Система моніторингу та відображення ціни крипто валютних акцій за допомогою ESP8266	Лім.	Маса	Масштаб
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Мойсюк М. В.						
Перевір.		Букурос О.В.						
Т. Контр.						Арк.	Аркуші	1
Реценз.					Циклова комісія Комп'ютерної інженерії			
Н. Контр.								
Затверд.								



					Система моніторингу та відображення ціни крипто валютних акцій за допомогою ESP8266			
Змн.	Арк.	№ докум.	Підпис	Дата	Система моніторингу та відображення ціни крипто валютних акцій за допомогою ESP8266	Лім.		
Розроб.	Мойсюк М. В.					Маса		
Перевір.	Букурос О.В.					Масштаб		
Т. Контр.								
Реценз.								
Н. Контр.					Циклова комісія Комп'ютерної інженерії	Арк.		
Затверд.						Аркушів 1		