

Вінницький національний технічний університет  
(повне найменування вищого навчального закладу)

Факультет інформаційних технологій та комп'ютерної інженерії  
(повне найменування інституту, назва факультету (відділення))

Кафедра обчислювальної техніки  
(повна назва кафедри (предметної, циклової комісії))

## Пояснювальна записка

до дипломного проекту

магістр

(освітньо-кваліфікаційний рівень)

на тему Технологія автоматизації процесів і об'єктів за допомогою  
мікропроцесорних WiFi модулів

Виконав: студент 2 курсу, групи 1КІ-18м  
напряму підготовки (спеціальності)

123 Комп'ютерна інженерія

(шифр і назва напряму підготовки, спеціальності)

(прізвище та ініціали)

Керівник Кадук О.В.

(прізвище та ініціали)

Рецензент Поплавський А. В.

(прізвище та ініціали)

## ВСТУП

Інтернет речей (Internet of Things, IoT) - це міжмережева взаємодія фізичних пристроїв, транспортних засобів, які називають «підключеними» або «розумними» пристроями, будівель та інших предметів, забезпечених електронікою, програмним забезпеченням, датчиками і підключенням до мережі, які дозволяють цим об'єктам збирати і обмінюватися даними.

Стрімкий розвиток технологій Інтернету речей обумовлює збільшення числа об'єктів і потоків інформації, які потребують опрацювання. Управління та розмежування взаємодії комп'ютерних систем і їх ресурсів є одним з важливих аспектів, що може бути реалізовано за рахунок створення новітніх технологій апаратно-програмних засобів. Останнім часом все більша увага науковців в галузі Інтернету речей приділяється підходам побудови таких засобів на автономних модулях, зокрема ESP8266, а тому тема є актуальною.

Об'єктом дослідження є процеси формування, перетворення та опрацювання цифрової інформації для Інтернету речей.

Предметом дослідження є методи і засоби опрацювання цифрової інформації в пристроях Інтернету речей.

Мета і задачі дослідження.

Постановка мети даної магістерської кваліфікаційної роботи базується на використанні апаратно-програмних засобів Інтернету речей.

Мета роботи полягає у автоматизації об'єктів і процесів за допомогою Wi-Fi модулів сімейства ESP8266.

Методи дослідження.

Дослідження, виконані під час роботи над кваліфікаційною магістерською роботою, ґрунтуються на:

- сучасних підходах і принципах формування, перетворення, опрацювання та передачі цифрової інформації;
- структурно-функціональному проектуванні – для реалізації структурних і функціональних схем системи;

- структурному проектуванні програмного забезпечення – для реалізації основного та допоміжного програмного забезпечення;
- методах роботи із електронними компонентами – для реалізації електричних принципових схем і макетів системи.

Для досягнення поставленої мети наукового дослідження необхідно вирішити такі задачі:

- здійснити огляд і аналіз сфер застосування Інтернету речей;
- здійснити класифікацію методів реалізації Інтернету речей;
- дослідити способи та методи побудови апаратно-програмних засобів;
- здійснити проектування структурно-функціональної та електричної принципової схем, а також здійснити вибір електронних компонентів;
- здійснити розробку та відлагодження програмного забезпечення.

#### **Наукова новизна одержаних результатів.**

У даній кваліфікаційній магістерській роботі вдосконалена технологія, в якій здійснюється передавання даних і керування об'єктами, що дозволило розширити сферу застосування інтернету речей.

#### **Практичне значення одержаних результатів.**

Практичне значення одержаних у результаті даного магістерського наукового дослідження полягає у розробці програмного забезпечення мікроконтролера ESP8266 з підтримкою бездротової мережі Wi-Fi, та протоколів обміну з системою управління, протоколу MQTT. Об'єкти можуть входити в загальну мережу «інтернету речей» (IoT) з можливістю автоматичного розпізнавання і початкового налаштування, а також мати веб-інтерфейс для остаточного налаштування.

Особистий внесок здобувача. Основні положення й результати магістерської роботи отримані автором.

## 1 ТЕХНІКО–ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ ДОЦІЛЬНОСТІ ТЕХНОЛОГІЇ АВТОМАТИЗАЦІЇ ОБ'ЄКТІВ І ПРОЦЕСІВ

### 1.1 Огляд існуючих технологій автоматизації об'єктів і процесів

Одне з таких рішень – це система «Розумний будинок» HomeKit компанії Apple. Завдяки HomeKit можна підключити, об'єднати і контролювати зі смартфона всю «розумну» техніку в будинку, на якій є мітка «Works with Apple HomeKit» (Сумісно з Apple HomeKit). Такими аксесуарами можуть бути світильники, замки, термостати, інтелектуальні розетки і багато іншого.

Все сумісні з HomeKit пристрої можуть бути об'єднані і працювати разом по заданому користувачем сценарією або за допомогою інтелектуального помічника Siri. Управління приладами відбувається з пристроїв iOS. Перевага Apple HomeKit в тому, що простота установки і настройки системи не вимагають від власника спеціальних знань і навичок [12].

#### Розумний офіс

Офіс Microsoft, а точніше, технологічний центр (Microsoft Technology Center), створений для демонстрації і тестування передових технологій, є реальним досвідом проектування та реалізації проекту «Розумного офісу». Роботу інфраструктури будівлі контролюють сотні датчиків і виконавчих механізмів, які забезпечують управління такими параметрами як:

- 1) температура і вологість у приміщеннях;
- 2) електропостачання;
- 3) охолодження;
- 4) відеоспостереження;
- 5) освітлення;
- 6) аудіо-відео та презентації обладнання;
- 7) розклад, конфігурація і завантаження приміщень;
- 8) клімат;

9) система доступу;

10) підвищення бізнес-продуктивності і об'єднаних комунікацій.

Система має безліч цікавих функцій, які, наприклад, дозволяють ідентифікувати співробітників або гостей офісу по смарт-карткам або за допомогою NFC в телефоні, забезпечує навігацію по офісу для гостей, має розумного електронного секретаря, вмє керувати проектором, освітленням та іншим обладнанням в переговорній кімнаті голосом зі смартфона і багато іншого [8].

#### Розумне місто

Поширення Інтернету речей не обмежується тільки масштабами будинків і окремих офісів. Цілі міста, з успіхом впроваджують в свою інфраструктуру «розумні сервіси».

#### Іннополіс

Це масштабний інфраструктурний проект федерального значення, спеціалізацією якого є сфера високих інформаційних технологій. Основною метою проекту є створення унікальної міської екосистеми, що забезпечує якісну підготовку, високий рівень життя і ефективну роботу кваліфікованих фахівців в сфері ІТ.

Всі існуючі інфраструктурні системи міста (будівлі, транспортні об'єкти, інженерні комунікації і т. д.) об'єднані в єдину «розумну» мережу «Smart + Connected Communities, S + CC», розроблену компанією Cisco. Запроваджене рішення передбачає єдину міську мережу Wi-Fi і відеоспостереження, контроль міського середовища та екологічної обстановки, інтелектуальну транспортну систему (комплекс датчиків транспортних потоків, центр обробки і керовані світлофори) і багато іншого. Вся інформація, отримана від усіх смарт-об'єктів міста, обробляється в центрах обробки даних Іннопіліса. На їх підставі системи життєзабезпечення міста оптимізує роботу всієї міської інфраструктури, що дозволяє місту ефективніше використовувати ресурси і скоротити фінансові витрати [11].

### Розумне страхування

Страхова компанія INTOUCH дає можливість повернути частину грошей за поліс КАСКО, завдяки програмі «Розумне страхування».

Суть програми в тому, що протягом півроку моніторинговий пристрій, встановлений в автомобілі автовласника, збирає інформацію про якість його водіння (перевищення швидкості, різкі прискорення і гальмування та ін.) І передає дані на сервер INTOUCH. На підставі цих даних система розраховує рівень акуратності водія, його стиль водіння і поведінки за кермом. За результатами оцінки якості, автовласникові повертається частина суми, яку він заплатив за поліс. Всі дані, що фіксуються, доступні в додатку на смартфоні, а також в особистому кабінеті на сайті компанії, що дозволяє відслідковувати зміну оцінки [9].

### Розумна медицина

«Інтернет речей» відкриває для медицини безпрецедентні можливості. Американська компанія Proteus Digital Health розробила продукт Proteus Discover, який дозволяє лікарям оцінити здоров'я пацієнта і контролювати курс лікування хворого. Proteus Discover – це система, яка складається з датчиків і програмного забезпечення.

Суть роботи системи в тому, що пацієнт приймає ліки, наприклад, у вигляді таблетки, на якій встановлений маленький, розміром з піщинку, датчик-сенсор. Другий датчик у вигляді пластиру клеїться на пацієнта і фіксує сигнали від проковтнутого сенсора. Коли датчик-сенсор потрапляє в шлунок, він посилає сигнал другому датчику про те, коли пацієнт прийняв ліки. Далі, в міру взаємодії сенсора з навколишнім середовищем, він передає інформацію про її стан і в якій області організму в даний момент знаходиться. Датчик-приймач оснащений бездротовим модулем зв'язку, який передає отримані дані на сервер. Через додаток на мобільному телефоні або на порталі Discover Portal лікуючий лікар може простежити, приймав, або не приймав його підопічний ліки вчасно, де знаходиться таблетка в даний момент, і оцінити стан здоров'я пацієнта. Крім цього,

Proteus Discover має детектор якості сну і датчик рівня фізичної активності користувача, що дозволяє виміряти додаткові параметри організму. Система нагадає пацієнтові через додаток на смартфоні, коли йому необхідно прийняти чергову порцію ліків згідно з курсом лікування [10].

Компанія REDMOND перша на ринку представила лінійку унікальних побутових приладів з технологією дистанційного керування Ready for Sky. Прилади з технологією R4S управляються одним зручним мобільним додатком. Для передачі даних між приладами використовуються сучасні технології, які істотно скорочують енерговитрати. Створення та впровадження революційного стандарту управління технікою R4S стало можливим завдяки роботі міжнародної команди hardware & software-розробників.

Зручний і інтуїтивно зрозумілий мобільний додаток Ready for Sky для пристроїв на платформах iOS і Android об'єднує всі необхідні користувачу дані, сортуючи їх по вигляду. Так, в додатку міститься більше 3000 рецептів для мультиварки, калькулятори підрахунку харчової цінності продуктів для кухонних ваг, лічильник кількості кроків або витрачених калорій для фітнес-браслета, варіанти дієт і графіки зміни ваги для підлогових ваг і багато іншого. Крім того, завдяки додатку R4S можна зберігати будь-які рецепти, що сподобалися, і ділитися ними з іншими користувачами.

Для управління побутовою технікою з R4S досить підключити прилади до маршрутизатора SkyBox або станції SkyTVbox, і встановити додаток на свій смартфон. І тоді, де б користувач ні перебував і чим би не займався, він завжди буде підтримувати невлотимий зв'язок зі своїм будинком. Управління розумною технікою Redmond безпосередньо зі смартфона або планшета можливо на відстані до 50 метрів (максимальна «далекобійність» Bluetooth 4.0), в залежності від товщини стін та інших перешкод на шляху сигналу відстань може варіюватися. На жаль, техніка Redmond Smart Home оснащена тільки Bluetooth-модулями. Так що для контролю через інтернет знадобиться ще один смартфон або планшет, який

буде виступати в якості «гейтвея». Апарат-передавач весь час повинен залишатися вдома, адже він буде приймати ваші команди через Інтернет і транслювати їх по Bluetooth вашим чайникам, праскам та іншій розумній техніці. Для перетворення гаджета в «Гейтвей» необхідно встановити на нього інший додаток – R4S Gateway. Вимоги до смартфону-передавача невисокі: Android 4.3 +, підтримка Bluetooth 4.0 і доступ в інтернет (Wi-Fi, 4G / 3G або хоча б 2G / EDGE). Програма R4S Gateway працює у фоновому режимі, займає мало пам'яті, не заважає працювати іншим додаткам і вкрай ощадливо витрачає трафік. Додаток Ready for Sky використовується не тільки для управління і контролю за розумною побутовою технікою, з його допомогою можна оновлювати ПО пристроїв. В розумних мультиварках і прасках може оновлюватися прошивка точно так же, як в смартфонах і планшетах. Причому з оновлення не просто виправляються якісь баги і недоробки – апарати набувають нові функції і режими роботи [13].

## 1.2 Огляд існуючих аналогів

Технологія бездротової передачі даних в локальних мережах Wi-Fi з'явилася в 1998 році завдяки інженеру австралійської лабораторії радіоастрономії CSIRO Джону О'Саллівану. Перший стандарт безпроводного протоколу обміну даними IEEE 802.11n був затверджений в 2009 році.

За час свого розвитку технологія Wi-Fi придбала широку популярність, перш за все через відсутність необхідності використання проводів при підключенні до мережі. І якщо спочатку технологія Wi-Fi застосовувалася для підключення переносних комп'ютерів, то в даний час ця технологія проникла і в фотоапарати, і в побутову техніку, і в мультимедійні пристрої, і в пристрої управління. Широко застосовуються і бездротові датчики різних фізичних величин - температури, тиску, вологості і т.п. Пристрої контролю і управління «розумним будинком», оснащені Wi-Fi модулями, можуть здійснювати свої функції з будь-якої точки, де доступна локальна бездротова



мережа, а в разі, якщо ця мережа через роутер має можливість виходу в глобальну мережу – і з будь-якої точки, де є Інтернет.

Стандартна схема Wi-Fi мережі містить, як мінімум, одну точку доступу, яка формує бездротову мережу з відомим ідентифікатором (SSID) і параметрами шифрування, до якої підключений, як мінімум, один клієнт. Точкою доступу може служити, як спеціалізований прилад, так і підключений до глобальної мережі роутер, оснащений бездротовим радіомодулем. Також, наприклад, точкою доступу можуть виступати ноутбук або смартфон, оснащені Wi-Fi модулями, і підключеними до мережі за допомогою кабелю або технології GSM відповідно.

З існуючої кількості аналогів, огляду підлягають модулі, що використовують лише бездротову технологію Wi-Fi. Деякі модулі призначені для використання в проектах DIY, поставляються у вигляді друкованої плати з компонентами і не мають корпусів, інші ж виконані у вигляді закінчених пристроїв і призначені відразу для використання. Слід врахувати, що кожне з розглянутих пристроїв є клієнтом бездротової мережі, отже, для підключення їх до мережі потрібно визначити точку доступу. Також потрібно звертати увагу на те, з якою мережею пристрій з'єднується – локальною або глобальною. Багато Wi-Fi пристроїв використовують портали, розміщені в глобальній мережі для зв'язку з іншими такими пристроями та обміну інформацією. Такий спосіб зв'язку дозволяє спростити з'єднання.

Почне наш огляд з DIY-модулів, що використовують технологію Wi-Fi.

#### 1. MP3509 - Wi-Fi реле з термометром і двома реле.

Основою пристрою служить, що набув широкого поширення, Wi-Fi модуль ESP8266. Модуль являє собою мікроконтролер, оснащений бездротовим інтерфейсом. Він підтримує стандарти IEEE 802.11 b / g / n, з шифруванням WEP і WPA / WPA2. Також модуль має 11 доступних для користувача портів введення / виведення і інтерфейси дротового зв'язку SPI, I2C, I2S, UART і 10-розрядний АЦП. Є кілька вільно розповсюджуваних

комплектів розробника (SDK) з компілятором і бібліотеками, що дозволяють ефективно використовувати можливості ESP8266.



Рисунок 1.1 – Зовнішній вигляд модуля MP3509

Все це дозволило створити сучасний пристрій для мобільного управління різними електроприладами за допомогою смартфона або планшета.

Wi-Fi реле MP3509 розраховане на роботу в локальній Wi-Fi мережі.

Крім дистанційного керування двома реле, призначеними для комутації навантаження до по 2000 Вт кожен, модуль забезпечує прийом і передачу в мережу показань підключених до нього двох датчиків температури типу DS18B20, датчика вологості DHT11 або DHT22, аналогових датчиків з використанням вбудованого АЦП. Модуль має відкриту систему команд для інтеграції в автоматизовані системи типу «розумний дім».

Модуль може використовуватися як окремо, з управлінням за допомогою програми для Андроїд, так і в складі, наприклад, системи автоматизації OpenHAB.

Як вже було зазначено, модуль MP3509 розрахований для роботи в локальних мережах. Якщо необхідно використовувати доступ до глобальної мережі, то в цьому допоможе наступний прилад.

2. MP3509 MQTT – Інтернет реле з термометром і двома реле, що використовує сервер MQTT.



Рисунок 1.2 – Зовнішній вигляд модуля MP3509 MQTT

Мережевий протокол MQTT (Message Queue Telemetry Transport) є спрощеним протоколом передачі даних між пристроями та працює поверх протоколу TCP/IP. Цей протокол використовує поведінковий шаблон проектування передачі повідомлень, відомий як «видавець-передплатник», вельми простий у використанні і адмініструванні, не створює великих навантажень на канали зв'язку і успішно працює при наявності проблем в цих каналах, а також не накладає обмежень на формат переданих даних. MQTT розроблений в розрахунку на малопотужні вбудовані пристрої, тому для його реалізації потрібні мінімальні обчислювальні потужності, з якими справляються мікроконтролери. Таким чином, протокол MQTT є, поряд з

деякими іншими аналогічними протоколами, наприклад MODBUS або RS-485, відмінним засобом для реалізації функцій «інтернету речей» - IoT.

Модуль MP3509 MQTT має ті ж технічні можливості, що і MP3509 :

- дистанційне керування двома реле по 2000 Вт кожен;
- прийом і передача в мережу показань підключаються до нього двох датчиків температури типу DS18B20;
- прийом і передача показань датчиків вологості DHT11 або DHT22, аналогових датчиків з використанням вбудованого АЦП.

Але, крім цього, він забезпечує зчитування даних з датчиків і управління вбудованими реле через Інтернет в будь-якій точці, тобто є підключення до глобальної мережі. При роботі в глобальній мережі використовується безкоштовний MQTT сервер, за замовчуванням [iot.eclipse.org](http://iot.eclipse.org), але можна використовувати і інший. Даний має дещо вищу вартість, у порівнянні з попереднім.

3. Якщо попередні два пристрої використовують можливості мікроконтролера ESP8266, то модуль VM8034 має в своєму складі власний більш потужний мікроконтролер серії STM8 і являє собою пристрій збору і передачі даних по Wi-Fi.

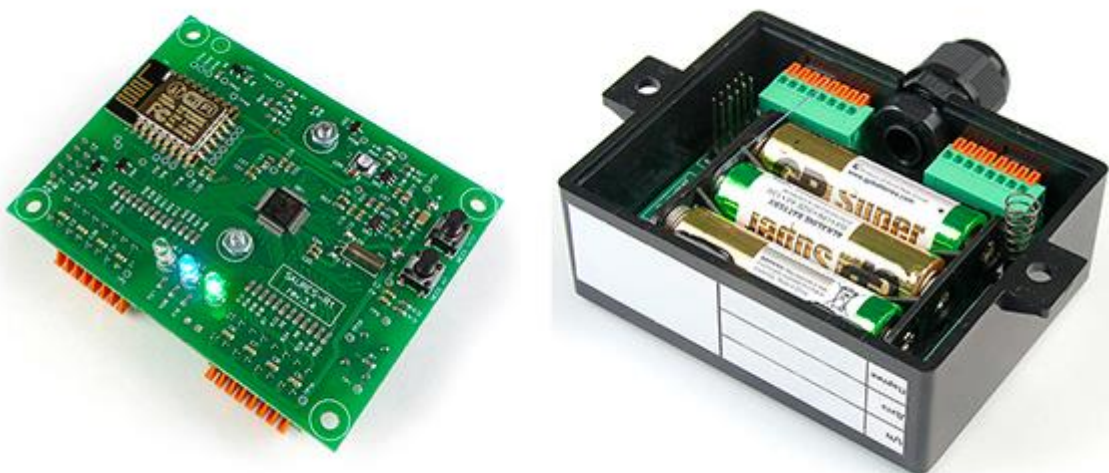


Рисунок 1.3 – Зовнішній вигляд модуля VM8034

Застосування мікроконтролера STM8 дозволило реалізувати в невеликому обсязі вельми широкий функціонал. Модуль є з одного боку закінченим пристроєм для збору даних про споживаних побутових ресурсах, таких, як вода, тепло, газ, електроенергія, а з іншого боку - багатофункціональним пристроєм контролю та управління виконавчими модулями і механізмами.

Прилад збирає дані з підключених до нього датчиків і лічильників за встановленим розкладом і передає ці дані на сервер, для подальшої обробки та використання.

До одного приладу може бути підключено до 8 будь-яких пристроїв в будь-якій комбінації:

- лічильники води;
- лічильники газу;
- лічильники електрики (при установці додаткового модуля інтерфейсу CAN або RS-485);
- датчики температури, наприклад, VM8034temp ;
- датчики протікання води, наприклад, VM8034water ;
- датчики рівня рідини;
- датчики витоку газу;
- виконавчі пристрої (запірно-регулююча арматура з електроприводом), наприклад, кульовий кран з електроприводом VM8034valve .

При використанні додаткових вбудованих модулів інтерфейсів RS-485 або CAN до одного модулю, крім 8 пристроїв, перерахованих вище в описі, можна підключити до 8 лічильників електроенергії типу Меркурій.

Модулі можна об'єднувати для збільшення кількості обслуговуваних каналів збору інформації.

Прилад збирає дані з лічильників і датчиків і за встановленим в настройках розкладом і передає їх на сервер. У штатному режимі дані відправляються раз на добу з погодинною деталізацією. Якщо буде виявлено

аварійна ситуація (відключився лічильник води, виникло протікання, сідає батарея і т.д.) прилад виходить на зв'язок негайно і повідомляє про це власника з використанням PUSH або E-MAIL повідомлень. На сервері організований особистий кабінет кожного користувача.

В особистому кабінеті можна вказати, в який день і годину сервер буде щомісяця автоматично відправляти показання. Показання можуть відправлятися наступними способами: у вигляді PUSH повідомлення, у вигляді E-MAIL, безпосередньо на портал.

Також ви можете просто дивитися показання і графіки на вашому мобільному телефоні, планшеті або комп'ютері, використовуючи браузер або мобільний додаток для iOS і Android.

Живлення модуля проводиться від трьох лужних (Alkaline) батарейок типорозміру AA, тривалість автономної роботи не менше 3 років. При зниженні рівня заряду нижче 10% користувачеві буде відправлено PUSH або E-MAIL повідомлення.

Пристрій зв'язується з сервером, використовуючи вихід в Інтернет через приватну чи публічну мережу Wi-Fi з шифруванням. Можна використовуватися дві мережі: основну і резервну. У відсутності зв'язку прилад збирає і зберігає погодинний журнал протягом 1 місяця, при його поновленні передає дані на сервер. У будь-якій позаштатній ситуації (прилад не виходить на зв'язок, відбулось протікання, обрив в лінії зв'язку до датчика або лічильника і т.п.) сервер відправить вам PUSH або E-MAIL повідомлення.

Таким чином, модуль VM8034 є відмінною основою для побудови систем для збору і контролю споживання побутових ресурсів в житлових будинках, будинках, невеликих селищах і т.п.

Розглянь тепер кілька закінчених корпусних пристроїв.

4. Розумна Wi-Fi розетка MA0201 з лічильником споживання електроенергії дозволяє управляти підключеними до неї побутовими електроприладами за допомогою смартфона з операційними системами

Android і iOS, а також виводити на екран смартфона інформацію про поточне споживання електроенергії, а також про сумарному споживанні за день, тиждень, місяць і рік.



Рисунок 1.4 – Зовнішній вигляд модуля MA0201

Управління здійснюється як вручну, так і за заздалегідь запрограмованим сценаріями. Для цього використовується безкоштовний додаток e-Control, в якому зберігаються всі зроблені настройки.

Пристрій може визначити, чи перебуваєте ви вдома, і виконати відповідний сценарій, включаючи або вимикаючи електроприлади.

Особливістю модуля є можливість спільної роботи з іншими модулями серії MA02xx.

## 5. MA0202 - це Wi-Fi модуль управління електроприладами.



Рисунок 1.5 – Зовнішній вигляд модуля MA0202

По суті це та ж розумна розетка, але виконана в конструктиві прихованої проводки для прямого підключення до електромережі та керованого пристрою. Таке рішення буде зручно для управління стельовими світильниками, кондиціонерами, теплою підлогою і т.п.

Управління також проводиться через додаток e-Control.

Особливість пристрою – наявність зворотного зв'язку. Інформація про поточний стан навантаження відображається на екрані смартфона. Також модуль може включати і вимикати навантаження за розкладом. Така функція може бути корисна для створення ефекту «присутності господарів», коли вдома нікого немає – одного із способів відвадити любителів поживитися чужим добром.

Прилад оснащений світлодіодним підсвічуванням і кнопкою вмикання/вимикання на корпусі. Корпус має можливість настінного кріплення.



6. Виконаний у футуристичному дизайні прилад MA0203 є станцією контролю якості повітря і навколишнього середовища.



Рисунок 1.6 – Зовнішній вигляд модуля MA0203

Станція оснащена датчиками, які в режимі реального часу визначають температуру і вологість повітря, освітленість, рівень шуму, а також наявність в повітрі летючих органічних речовин і вуглекислого газу. При цьому проводиться інтегральна оцінка якості повітря на основі заводських налаштувань.

Вся зібрана приладом інформація може бути відображена на екрані смартфона.

Пристрій може служити універсальним датчиком «розумного будинку». Залежно від показань датчиків, при наявності інших підключених до цієї ж мережі пристроїв серії MA02xx, можна налаштувати сценарії

автоматизації певних дій. Наприклад, MA0203 може включити зволожувач повітря при зниженні вологості до заданого рівня, освітлення при настанні темряви або витяжну вентиляцію при перевищенні рівня CO<sub>2</sub>.

Пристрій має на корпусі три USB-роз'єму, один з яких використовується для підключення джерела живлення, а два інших для під'єднання додаткових датчиків.

Підсвічування, вмонтована в корпус, наочно підкаже вам ступінь комфорту в приміщенні: червоний, синій і зелений кольори відповідають поганий, гарній і ідеальній обстановці.

7. До складу «розумного будинку» може органічно увійти система охорони MA0204 .



Рисунок 1.5 – Зовнішній вигляд модуля MA0204

Датчики системи безперервно стежать за охоронюваним приміщенням, і в разі небажаного проникнення в нього на ваш смартфон буде негайно

відправлено PUSH-повідомлення, яке миттєво буде відображено на екрані разом зі звуковим сигналом.

Крім блоку управління, в комплект системи входять:

- пульт для зручного оперативного управління;
- датчик відкриття дверей;
- PIR-датчик руху, який виявляє переміщення в інфрачервоному діапазоні, тобто реагує на рух теплих предметів, таких, як людина або тварина.

Після нескладної настройки система буде надсилати на смартфон повідомлення про те, який датчик спрацював в приміщенні. Також систему можна налаштувати таким чином, що при будь-яких змінах станів датчиків інформація про це буде передаватися іншим підключеним до мережі пристроїв і змушувати їх виконувати заздалегідь визначені дії. Таким чином, система MA0204 може бути центром попередження і реагування на всі події в контрольованому просторі події.

Брелок-пульт дистанційного керування спрощує оперативну роботу з системою, дозволяє однією кнопкою включати настройки безпеки будинку або, навпаки, знімати їх, а також подавати сигнал «SOS» в надзвичайних ситуаціях. Режим «AUTO» заздалегідь підготує комфортні умови в будинку до вашого приходу. Виїжджаючи в напрямку будинку, ви заздалегідь можете подати команду запуску підключених пристроїв для включення опалення, світла, вентиляції і т.д.

### 1.3 Оцінювання комерційного потенціалу розробки

Метою проведення технологічного аудиту є оцінювання комерційного потенціалу розробки. Для проведення технологічного аудиту було залучено 2-х незалежних експертів.

Здійснюємо оцінювання комерційного потенціалу розробки за 12-ма критеріями за 5-ти бальною шкалою.

Таблиця 1.1 – Критерії оцінювання комерційного потенціалу розробки

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
Критерій	0	1	2	3	4
Технічна здійсненність концепції					
1	Достовірність концепції не підтверджена	Концепція підтверджена на розрахунках	Концепція підтверджена на розрахунках	Концепція перевірена на практиці	Перевірено роботоздатність
Ринкові переваги (недоліки)					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижча за ціни аналогів
Ринкові переваги (недоліки)					
4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів

Таблиця 1.1 – Критерії оцінювання комерційного потенціалу розробки (продовження 1)

Ринкові перспективи					
6	Ринок малий і не має позитивної динаміки	Ринок малий але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Активна конкуренція	Незначна конкуренція	Конкурентів немає
Практична здійсненність					
8	Відсутні фахівці як з технічної так і з комерційної реалізації	Необхідно наймати фахівців або витратити значні кошти та час на навчання	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
Практична здійсненність					
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування ідеї відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні дешеві та досяжні матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві

Таблиця 1.1 – Критерії оцінювання комерційного потенціалу розробки (продовження 2)

11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10 років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

Результати оцінювання комерційного потенціалу розробки наведено в таблиці 1.2.

Таблиця 1.2 – Результати оцінювання комерційного потенціалу розробки

Критерії	Експерт 1	Експерт 2
	Бали, виставлені експертами:	
1	3	3
2	4	5
3	3	4
4	5	4
5	4	3
6	3	4

Таблиця 1.2 – Результати оцінювання комерційного потенціалу розробки (продовження 1)

7	3	3
8	4	4
9	4	3
10	4	4
11	3	5
12	3	4
Сума балів	СБ <sub>1</sub> = 43	СБ <sub>2</sub> = 47
Середньоарифметична сума балів $\overline{СБ}$	$\overline{СБ} = \frac{\sum_1^3 СБ_i}{2} = 45$	

Отже, з отриманих даних таблиці 1.1 видно, що нова розробка має високий рівень комерційного потенціалу.

#### 1.4 Потенційні ринки збуту та прогнозування попиту на інноваційне рішення

Аналіз попиту на нову продукцію – один з найважливіших напрямів діяльності інноваційних компаній. Комерційним компаніям немає сенсу вкладати свої кошти в НДДКР, якщо кінцевий результат себе не окупить.

Тому необхідним є розрахунок ємності ринку. Для цього необхідно спрогнозувати наступні дані:

- середня кількість споживачів, які використовують товари аналогічні розроблюваному (П);
- середній відсоток споживачів, які зацікавлені придбанням інноваційного продукту (П<sub>п</sub>);
- середній термін заміни інноваційного продукту (Т);
- середній відсоток споживачів, що захочуть придбати розроблюваний товар повторно (С<sub>п</sub>).

Визначившись із вихідними даними можна визначити:

- потребу в інноваційному продукті:

$$Pi = \frac{P \cdot P_n \%}{100\%}; \quad (1.1)$$

$$Pi = \frac{12000 \cdot 50\%}{100\%} = 6000;$$

- оптимістичний прогноз попиту на інноваційне рішення:

$$OP = \frac{Pi}{T}; \quad (1.2)$$

$$OP = \frac{6000}{2} = 3000;$$

- песимістичний прогноз попиту на інноваційне рішення:

$$ПП = \frac{OP \cdot C_n}{100\%}; \quad (1.3)$$

$$ПП = \frac{3000 \cdot 20}{100\%} = 600;$$

- реалістичний прогноз попиту на інноваційне рішення:

$$RP = \frac{OP + ПП}{2}. \quad (1.4)$$

$$RP = \frac{3000 + 600}{2} = 1800 \text{ шт.}$$



## 1.5 Висновок

Огляд існуючих технологій автоматизації об'єктів і процесів та їх сфер застосування показує важливість і вагомість у різних галузях. Аналіз позитивних та негативних складових дозволило зрозуміти вирішені та невирішені задачі, які потребують інноваційних підходів та рішень.

Проведено оцінювання комерційного потенціалу розробки та визначено потенційні ринки збуту і здійснено прогнозування попиту на інноваційне рішення.

Розробка технології автоматизації об'єктів і процесів за допомогою Wi-Fi модулів є актуальною задачею, технічно та економічно доцільною.

## 2 ОГЛЯД І АНАЛІЗ ПІДХОДІВ АВТОМАТИЗАЦІЇ ОБ'ЄКТІВ І ПРОЦЕСІВ ЗА ДОПОМОГОЮ WI-FI МОДУЛІВ

### 2.1 Мікроконтролер ESP8266

Компанія Espressif Systems постійно здійснює науково-дослідницьку і проектну діяльність в області Wi-Fi і Bluetooth-технологій. В результаті чого, продукція Espressif широко відома в усьому світі і використовується в мобільних пристроях, побутової техніки та промислових додатках.

Високоінтегровані Espressif Wi-Fi чіпсети і модулі створені таким чином, щоб займати мінімальний місце на платі, універсальні в застосуванні, економічні по споживанню і вигідні за ціною.

Приблизно в серпні 2014 року на торговому майданчику Aliexpress з'явилися дешеві (близько 4 \$) WI-FI модулі ESP8266 китайського розробника. Модуль продається із завантаженою прошивкою, яка утворює Wi-Fi-- UART міст для підключення до іншого мікроконтролера, в тому числі і до Arduino. Налагодження та обмін даними відбувається за допомогою AT команд [17].

Технічні характеристики ESP8266

Процесор: однопоточний Tensilica L106 частотою до 160 MHz;

Стандарти WI-FI: 802.11 b / g / n;

Типи шифрування: WEP, WPA, WPA2;

Режими роботи: Клієнт (STA), Точка доступу (AP), Клієнт + Точка доступу (STA + AP);

Напруга живлення: 1.7..3.6 В;

Струм: до 215мА в залежності від режиму роботи;

Кількість GPIO: 16 (фактично до 11). Доступно на модулях: ESP-01 - 4, ESP-03 - 7 + 1, включаючи UART. Існують і інші варіанти модулів;

Інтерфейси: 1 ADC, I2C. UART, SPI, PWM;

Зовнішня Flash пам'ять може бути встановлена від 512кб до 4Мб; RAM даних 80 кб, RAM інструкцій - 64 кб.

Структурна схема чіпа показана на рисунку 2.1. Вона складається всього з декількох елементів. Відповідно менше елементів – менше ціна компонентів, менше вартість пайки, менша площа розміщення, менше вартість друкованої плати. Управляє всією цією розширена версія 32-бітного процесора Tensilica's L Diamond series [18].

Енергоспоживання – одна з найважливіших характеристик інтернету речей. Споживачі не готові міняти батареї в датчиках кожні два-три місяці. Тому «вихід в мережу» доводилося забезпечувати мостами, підключеним до постійного живлення. І тепер з появою цього модуля, Wi-Fi можна використовувати навіть в автономних датчиках, що працюють на невеликих батареях. Завдяки використанню просунутих механізмів управління енергоспоживанням. Якщо подивитися на характеристики споживання чіпа, 215mA в режимі передачі, близько 60mA в режимі глибокого сну (з працюючими годинами реального часу) і менше 1.0mA (DTIM = 3) або менше 0.5mA (DTIM = 10) в режимі підтримки зв'язку з точкою доступу Wi-Fi [18].

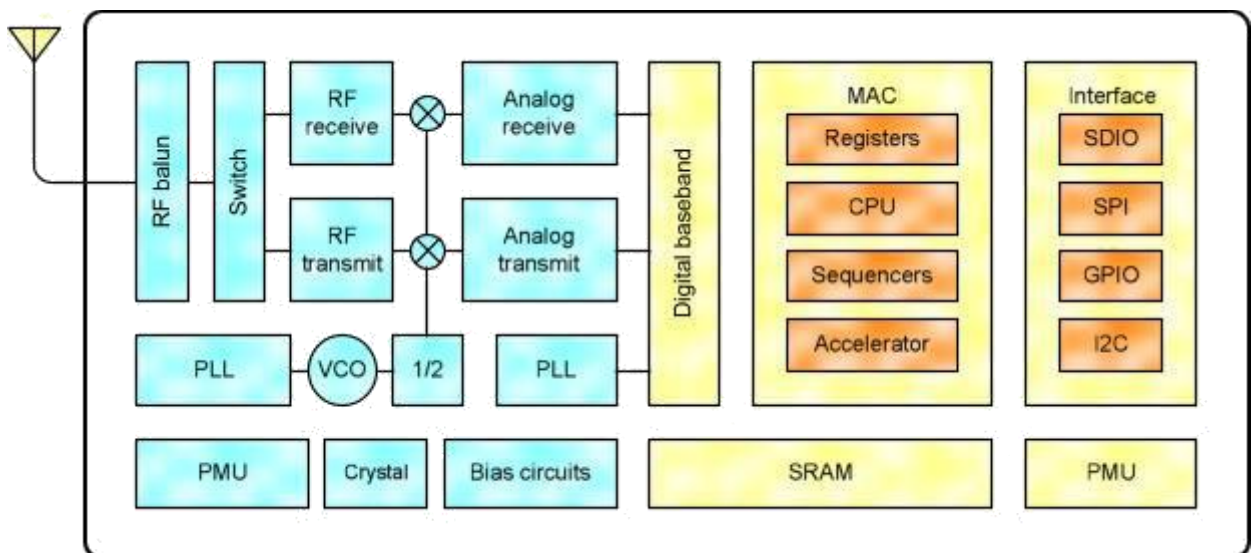


Рисунок 2.1 – Структурна схема чіпів сімейства ESP8266

## 2.2 Різновиди ESP8266

Зараз у продажу можна знайти готові модулі 12 видів: з підключенням зовнішньої антени, з керамічною антеною, з антеною з доріжок, екранування в корпусі. Так само буває виведено різну кількість GPIO.

Модулі ESP8266: ESP-01, ESP-02, ESP-03, ESP-04, ESP-05, ESP-06, ESP-07, ESP-08, ESP-09, ESP-10, ESP-11, ESP-12, ESD-12, ESP-13, WROOM, WROOM-02. Мікроконтролери, які найбільше користуються попитом, представлені на рисунку 2.2.

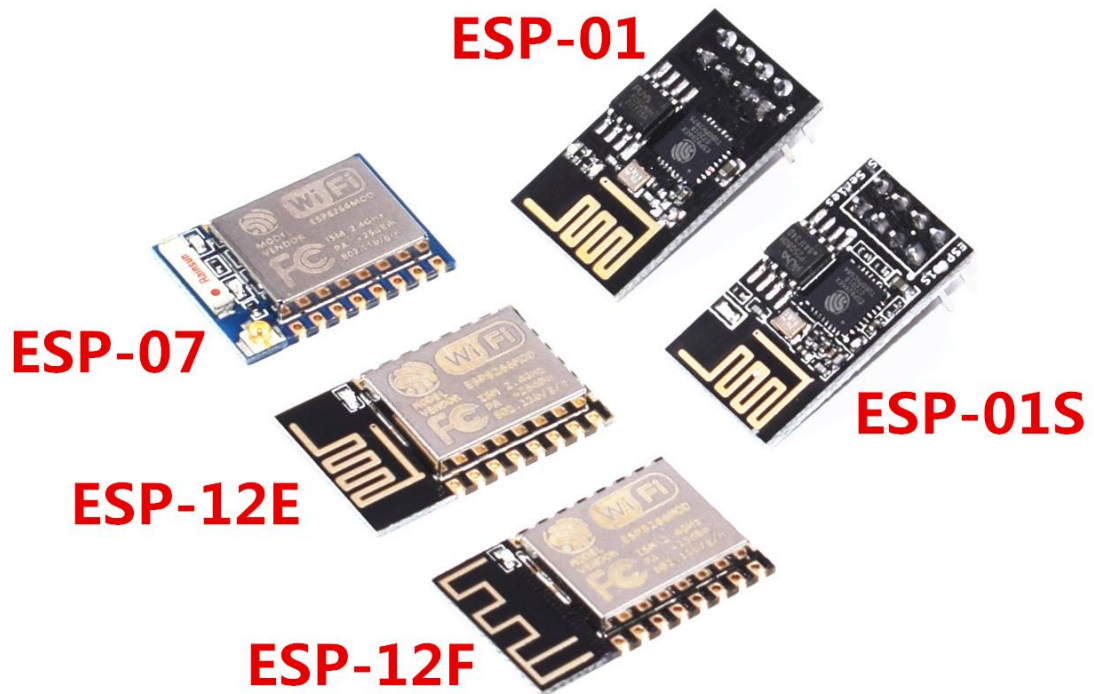


Рисунок 2.2 – Зовнішній вигляд модулів сімейства ESP8266

Мікроконтролери не мають на кристалі користувальницької незалежної пам'яті. Виконання програми ведеться з зовнішньої SPI ПЗП шляхом динамічного підвантаження необхідних ділянок програми в кеш інструкцій. Підтримується до 16 МБ зовнішньої пам'яті програм. Можливий Standard, Dual або Quad SPI інтерфейс.

Чіп ESP8266 є одним з найбільш високоінтегрованих рішень для роботи з Wi-Fi [19].

### 2.3 Аналіз та вибір оптимального рішення

Аналіз та вибір оптимального рішення, неминує призводить до порівняння між собою ESP8266-сумісних чіпів і модулів, встановлених на існуючих платах. Порівнювати досить складно не стільки тому, що подібних плат безліч, тому, що під різними іменами випускаються одні і ті ж плати.

Що таке NodeMCU і ESP8266

На жаль, є плутанина в позначенні версій і поколінь плат.

Таблиця 2.1 – Типи плат NodeMCU

Generation (покоління)	Версія NodeMCU	Common Name
1	0.9	V1
2	1.0	V2
2	1.0	V3

Вміст третього стовпчика (Common Name) цієї таблиці найчастіше зустрічається в маркуванні виробів, доступних у продажу в китайських Інтернет-магазинах, таких як Banggood, AliExpress і dx.com. Причому на AliExpress часто трапляються плати з маркуванням V3, хоча вони виглядають точно так само, як плати V2.

Додатково підсилює плутанину іменувань той факт, що апаратура відкрита (open hardware), і фактично будь-яке підприємство може виробляти і випускати на ринок свої власні NodeMCU-сумісні плати. На даний момент є 3 головних виробника таких плат: Amica [3], DOIT / SmartArduino [4], and LoLin / WeMos [5].

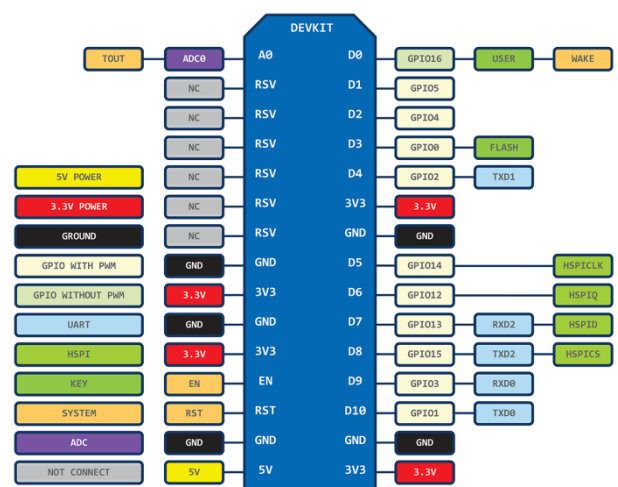
Плати 1 і 2 покоління зручно відокремлювати один від одного, тому їх розмір відрізняється. Обидва цих покоління використовують чіпи ESP-12 з пам'яттю 4MB flash, проте 2 покоління використовує більш новий покращений чіп ESP-12E.

## 1 покоління / v0.9 / V1

Оригінальний і вже застарілий dev kit зазвичай продається у вигляді плати з жовтої маскою розміром 47 x 31 мм. По краях модуля розташовані Штиркові коннектори зі стандартним кроком 2.54 мм, однак ряди розташовані один від одного на великій відстані, що робить незручним установку цих модулів в звичайні плати макетування bread board. Огляд [6] дає дуже гарне уявлення про ці плати.



а)



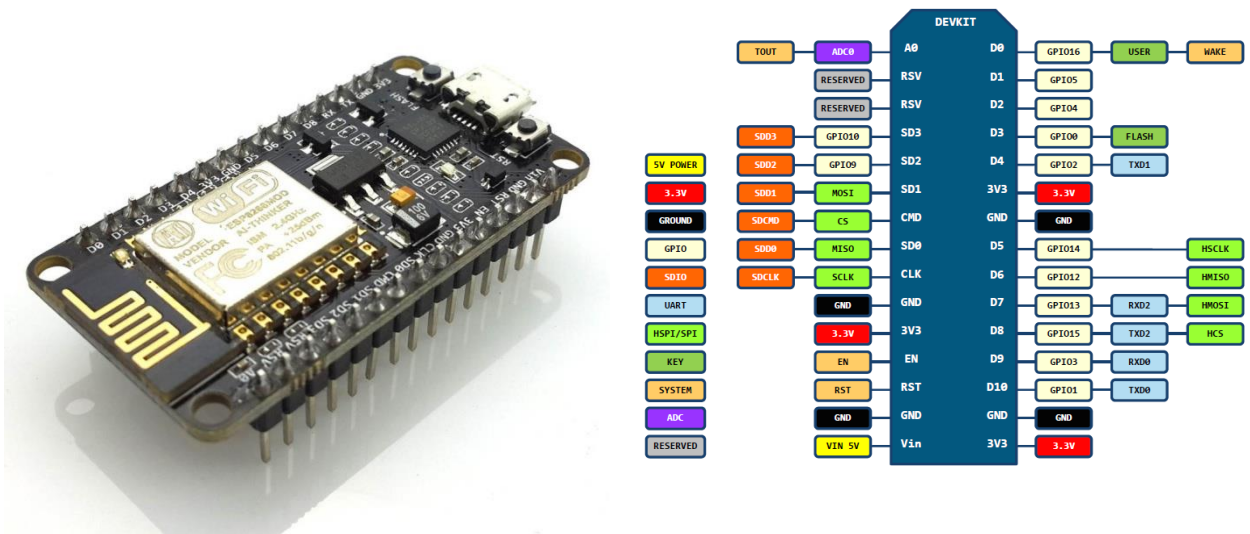
б)

Рисунок 2.3 – Плата NoduMCU 1 покоління: а) зовнішній вигляд;  
б) призначення виводів.

Плати поставляються з модулем ESP-12 і flash-пам'яттю 4МБ.

## 2 покоління / v1.0 / V2

Ці модулі швидко замінили 1 покоління, і тому їх зручніше встановлювати в плату bread board. Модуль ESP-12 був замінений на більш новий ESP-12E.



а)

б)

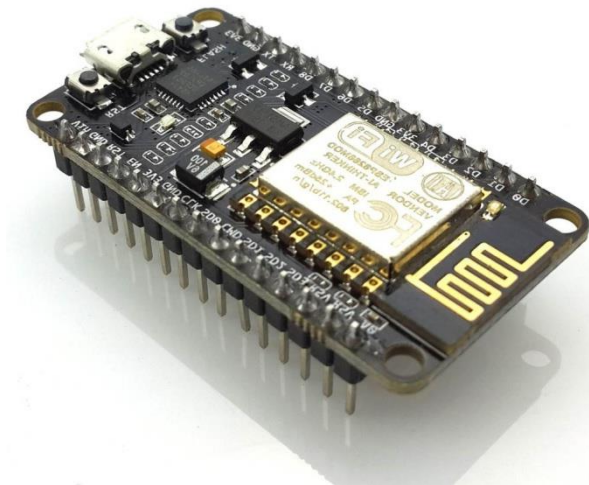
Рисунок 2.4 – Плата NoduMCU 2 покоління: а) зовнішній вигляд;  
б) призначення виводів.

3 покоління / v1.0 / V3

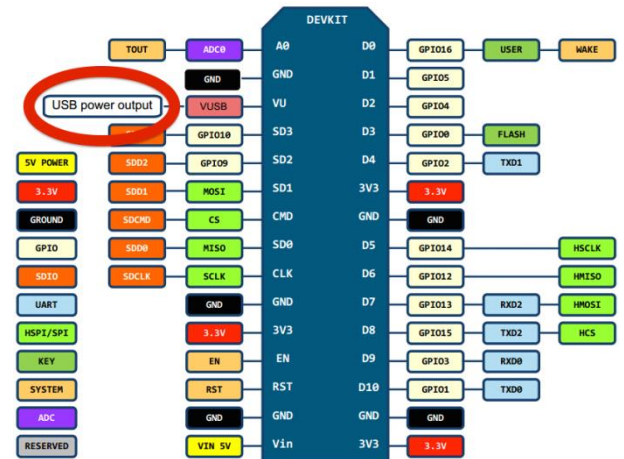
Поки що проект NodeMCU не випустив нової специфікації, тому офіційно немає ніяких плат 3-го покоління. Таким чином, V3 це всього лише "версія", винайдена виробником LoLin, щоб показати незначні поліпшення плат V2. Серед інших поліпшень стверджується, що порт USB став більш надійним.

Якщо порівнювати цоколевку плат V2 і V3, то знайдеться дуже мало відмінностей. Компанія LoLin прийняла рішення використовувати один з двох зарезервованих висновків для виведення напруги живлення USB, і другий використовує як додатковий контакт GND.

Варто звернути увагу на відмінності в розмірах, ця плата LoLin значно більше плат Amica і DOIT V2 (дтв. рис. 2.4).

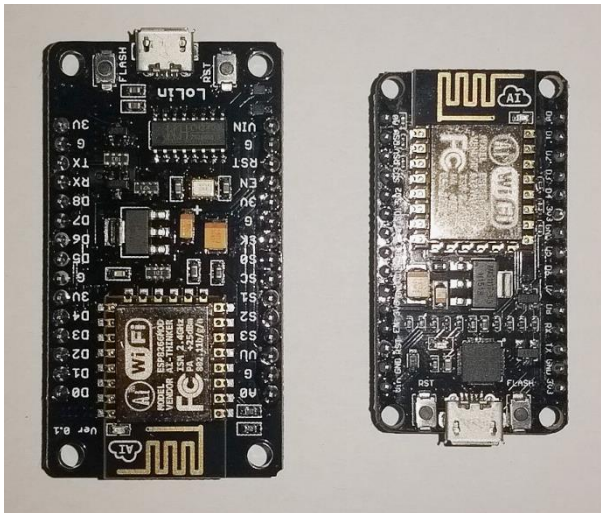


а)



б)

Рисунок 2.5 – Плата NodeMCU 3 покоління: а) зовнішній вигляд; б) призначення виводів.



а)



б)

Рисунок 2.6 – Зовнішній вигляд плат Amica и LoLin для порівняння: а) з боку деталей; б) зі зворотної сторони.

На ринку існує наявність "офіційних" плат та "неофіційних" Їстотних відмінностей немає. Компанія Amica виглядає як єдиний виробник плат, на 100% сумісних зі специфікацією апаратури V2 NodeMCU. Такі плати можна знайти на AliExpress або у Seed Studio [7]. Майте на увазі, що зараз безліч плат мають маркування "NodeMCU V2", хоча вони не відповідають цій специфікації.



## WeMos D1 mini

У наприкінці 2015 року стала доступною цікава плата WeMos D1 mini. Вона має приблизно ту ж ширину (25.6 мм), що і V2 NodeMCU devkit, але майже на третину коротше (довжина 34.2 мм). Плата працює на ESP-8266EX MCU, і надає 4MB flash. 9 виводів GPIO роблять плату D1 mini підходящою для безлічі цільових застосувань IoT. Плата підтримується середовищами розробки Arduino і NodeMCU.

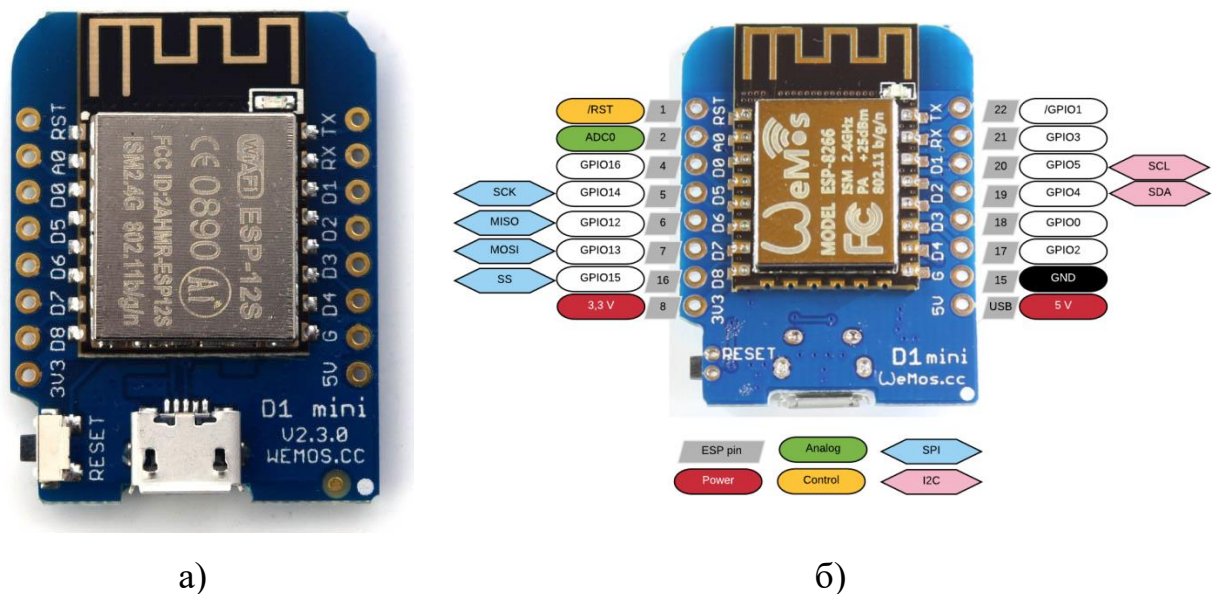
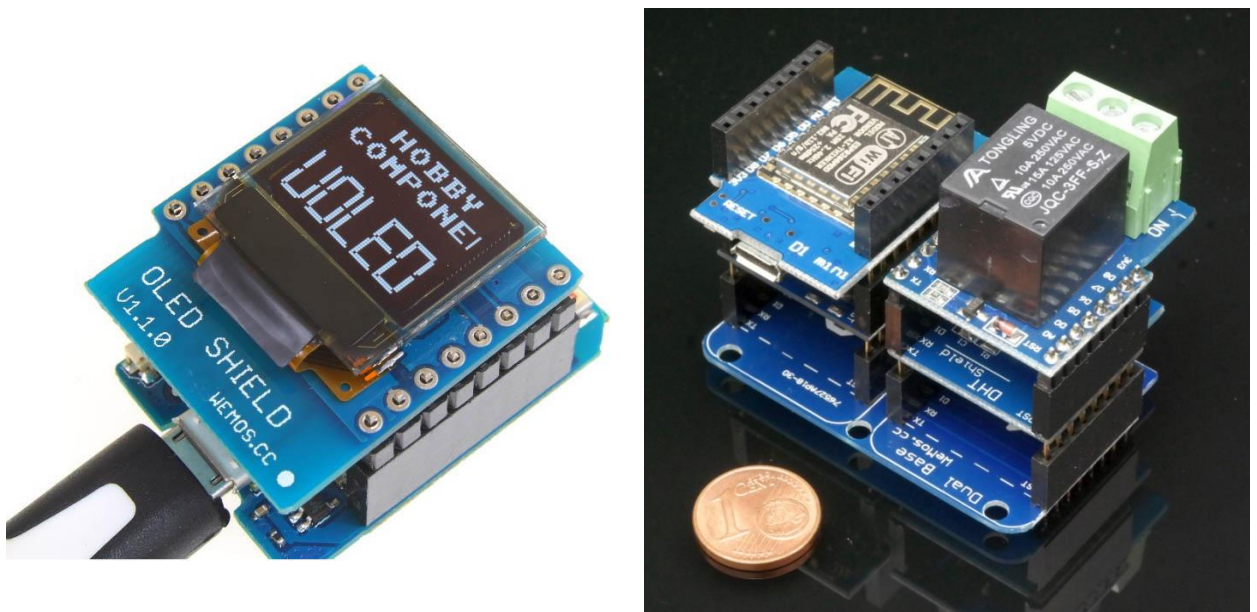


Рисунок 2.7 – Плата WeMos D1 mini: а) зовнішній вигляд;  
б) призначення виводів.

Компанія WeMos також продає кілька ШІлд для плати D1 mini, які добре стикуються, в результаті виходять відмінні малогабаритні пристрої.

Невеликий недолік в тому, що припаювати роз'єми доведеться самому. Деякі користувачі скаржаться, що буває складно отримати коректно працюючі драйвера для мікросхеми перетворювача CH34x USB-UART [8], яка встановлена на D1 mini. Така ж мікросхема стоїть на дешевих клонах плат Arduino.



а)

б)

Рисунок 2.8 – Плати розширення (Шілд) WeMos D1 mini:

а) однорівневі; б) багаторівневі.

Також очевидно, що у D1 mini трохи менше контактів, ніж у звичайній платі NodeMCU. Зрозуміло, що не було іншого способу зменшити розміри плати, крім зменшення кількості контактів на роз'ємах. Але все-таки контакт з напругою 5V був залишений, в цьому плата схожа на LoLin V3.

Радують невисока ціна на плату D1 mini, її можна знайти на AliExpress всього лише за \$ 4.

### WiFiMCU

У DOIT / SmartArduino, виробника плат V2, також є альтернативний dev kit з чіпом Cortex-M4 [9].

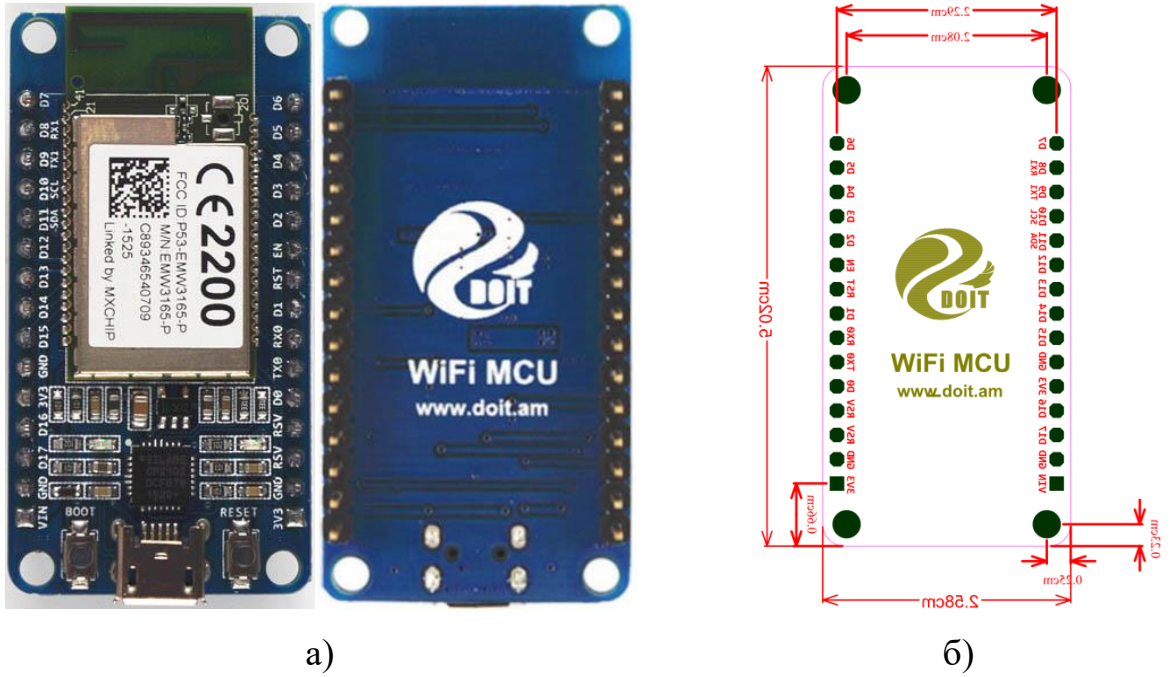


Рисунок 2.9 – Плата WiFiMCU: а) зовнішній вигляд; б) габаритні розміри.

Adafruit / SparkFun

Ще швидко з'явилися дві альтернативи - SparkFun ESP8266 Thing [10] і Adafruit HUZZAH ESP8266 Breakout [11].

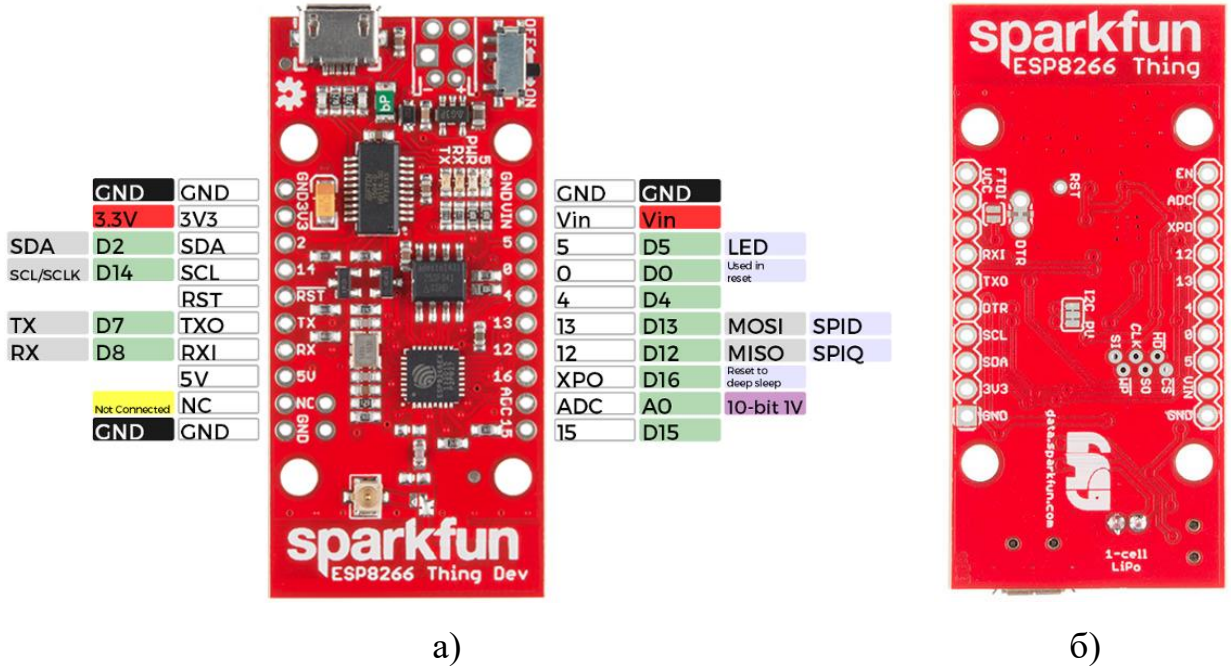


Рисунок 2.10 – Плата SparkFun ESP8266 Thing:  
а) вигляд зі сторони деталей; б) зі зворотної сторони.

Плата Adafruit виглядає досить привабливо, вона трохи менше, ніж NodeMCU dev kit, проте для її підключення потрібен окремий перехідник USB - TTL UART замість стандартного кабелю USB.

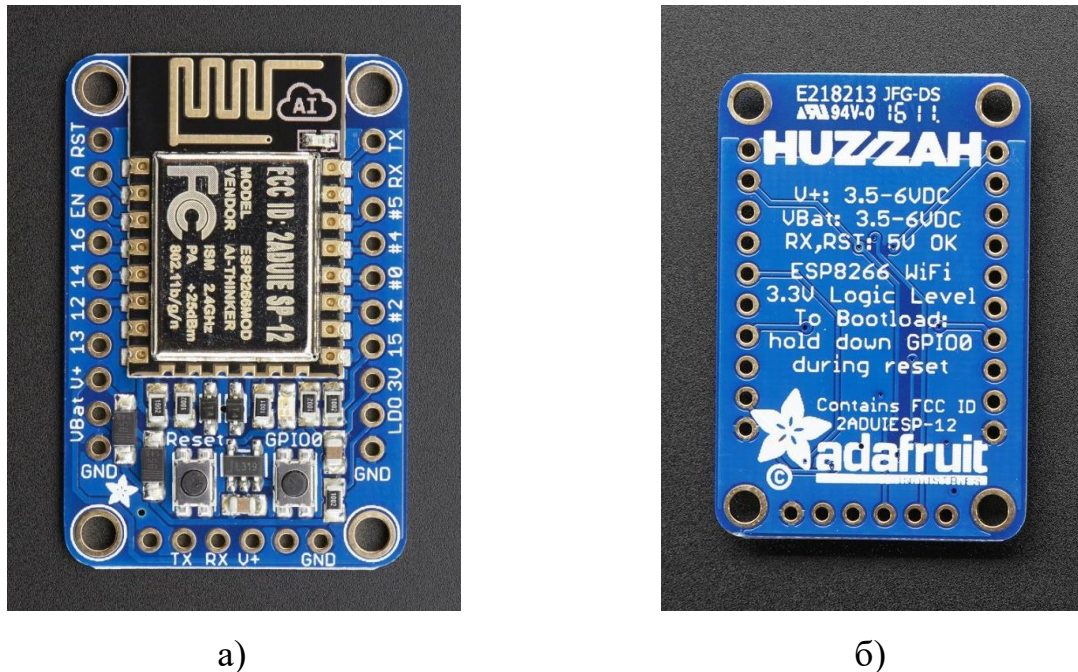


Рисунок 2.11 – Adafruit HUZAZH ESP8266 Breakout:  
а) вигляд зі сторони деталей; б) зі зворотної сторони.

### Adafruit HUZAZH ESP8266 Breakout

Також Adafruit продає повноформатну плату, подібну NodeMCU v2, на якій є додатковий коннектор для підключення акумулятора LiPo [12].

### Плата ESP8285

У статті [13] описується, як використовувати firmware NodeMCU на маленькій платі ESP8285 Tindie . Цей кристал SoC (System On Chip) дуже схожий на ESP8266, відмінність тільки в тому, що в ньому є пам'ять 1 MB SPI flash, вбудована прямо в SoC.

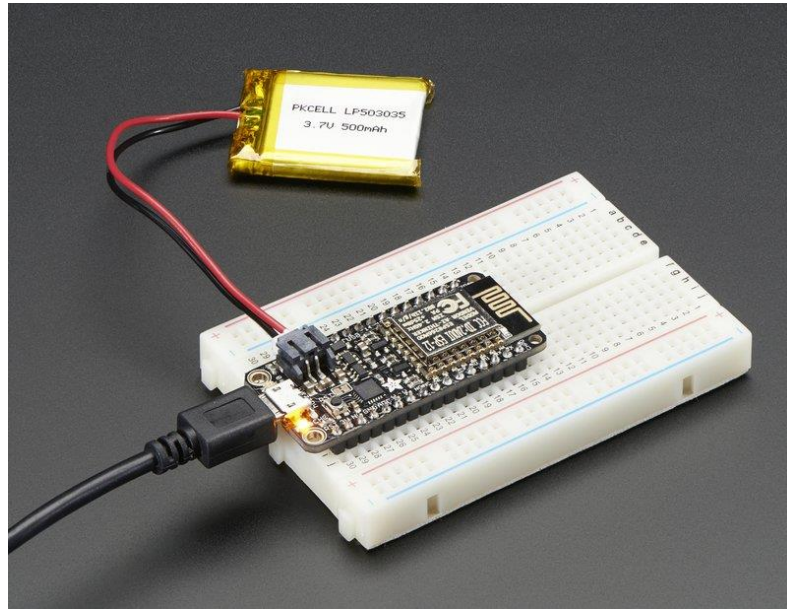


Рисунок 2.12 – Плата Adafruit з можливістю підключення LiPo акумулятора

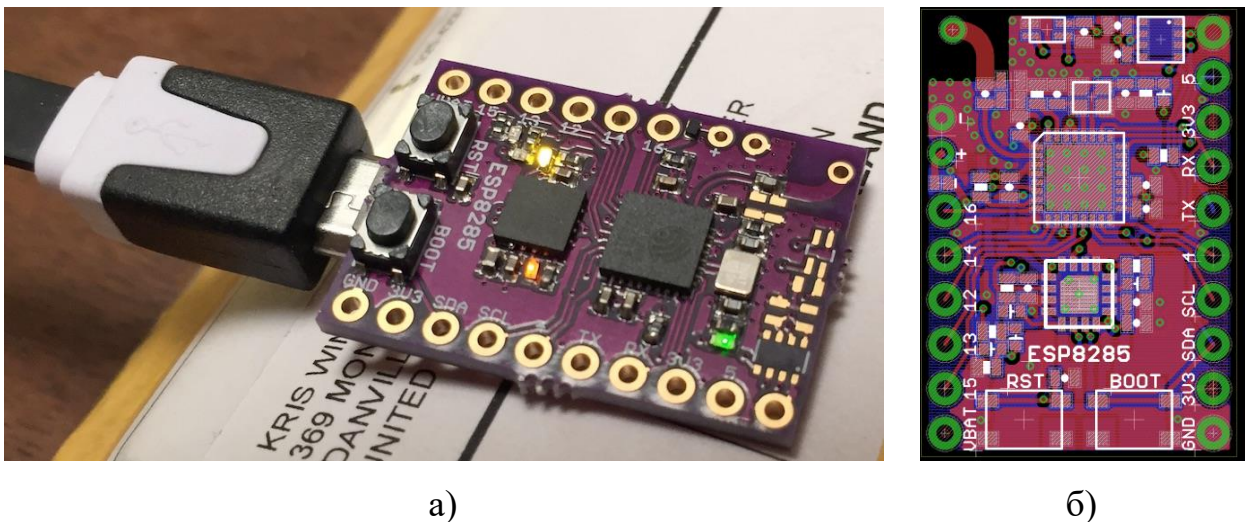


Рисунок 2.13 – ESP8285 Development Board:

а) вигляд зі сторони деталей; б) зі зворотної сторони.

#### 2.4 Огляд прошивок модуля ESP8266

Щоб спростити використання мікроконтролера в типових проектах можливе використання готових бінарних файлів, придатних до прямої заливки в ПЗП модулів (так званих прошивок). Готові прошивки можна розділити на кілька груп відповідно до концепції їх використання:

а) Прошивки для роботи під керуванням зовнішнього контролера. У цих прошивках реалізований якийсь функціонал, якому зовнішній контролер

через UART задає параметри роботи. До таких прошивки відноситься - прошивка з керуванням AT-командами з SDK Espressif.

б) Прошивки з вбудованими інтерпретаторами різноманітних мов високого рівня. Ці прошивки дозволяють довантажувати через UART і виконувати скрипти розробника пристрою.

NodeMCU – проект на основі скриптованої мови LUA. Прошивка вміє виконувати LUA-скрипти як з UART (аналогічно AT-командам) так і з внутрішньої flash пам'яті. Для завантаження скриптів в flash пам'ять підтримується файлова система SPIffs. З боку Wi-Fi є вбудовані MQTT протокол і HTTP сервер. Вбудована графічна оболонка, що дозволяє підключати до ESP8266 графічні індикатори.

в) Прошивки для Інтернету речей. Цей клас прошивок дозволяє з одного боку підключити до ESP8266 набір датчиків і виконавчих пристроїв, а з іншого боку надає необхідний мережевий функціонал для роботи в інфраструктурі Internet of Things.

ESP Easy – прошивка для домашньої автоматизації. Виконана як скетч Arduino. Підтримує безліч датчиків і виконавчих пристроїв, підтримка яких є в Arduino (наприклад, датчики температури, вологості, освітленості, дисплеї, реле і т. п.). Є клієнт MQTT.

Wi-Fi-iot.com/ - онлайн конструктор IoT-прошивок для ESP8266 для проекту домашньої автоматизації. Дозволяє прямо на сайті створити прошивку ESP8266 для роботи з однією з безлічі підтримуваних IoT систем верхнього рівня. Варіант необхідної прошивки створюється шляхом вибору опцій конфігурації. Проект комерційний, але є безкоштовні з обмеженими функціональними можливостями [19].

#### 2.4.1 Початкове завантаження і оновлення прошивки

Для управління процесом оновлення прошивки є безліч утиліт:

XTCOM – консольна утиліта з Espressif SDK.

NodeMCU-Flasher – віконна утиліта під Win.

esptool ck – консольна утиліта, написана на Cі.

esp\_tool – консольна утиліта, написана на C ++.

esptool.py – консольна утиліта, написана на Python[19].

## 2.5 Програмні засоби розробки

Програмний комплект розробника SDK складається з:

- компілятора. Оскільки компілятор має відкриті вихідні тексти, то в різних SDK можуть міститися різні збірки цього компілятора, які трохи відрізняються опціями, що підтримуються;
- бібліотек для роботи з периферією контролера, стеків протоколів;
- засобів завантаження файлу в пам'ять програм мікроконтролера;
- опціональної IDE.

Espressif вільно поширює свій комплект розробника. У цей комплект входить компілятор GCC, бібліотеки Espressif і завантажувальна утиліта XTCOM. Бібліотеки поставляються в вигляді скомпільованих бібліотек, без вихідних текстів.

Espressif підтримує дві версії SDK: одна на основі RTOS, інша на основі зворотних викликів (callback) [19].

### 2.5.1 Мова програмування LUA

LUA є типовою процедурною мовою програмування. Він надає широкі можливості для об'єктно-орієнтованої та функціональної розробки. LUA створювалася як потужна і проста мова, що володіє всіма необхідними виразними засобами. Бібліотека функцій мови LUA написання на мові C. LUA не має поняття "main" програми: вона працює в середовищі виконання, скорочено званої хост. Хост-програма дозволяє запускати частини коду, написані на LUA, модифікувати змінні LUA і реєструвати C-функції для використання безпосередньо в коді LUA. Завдяки можливості розширення за допомогою C-функцій, LUA може застосовуватися для вирішення широкого кола завдань. Таким чином, можемо створювати спеціалізовані бібліотеки,

використання яких нічим не відрізняється від використання стандартних засобів мови. Іменами (ідентифікаторами) в LUA можуть бути будь-які рядки з букв, цифр і символу підкреслення, що не починаються з цифри. Це правило типово для більшості мов програмування. (Поняття літери залежить від поточної локалі: будь-який символ з алфавіту поточної локалі може бути використаний у складі ідентифікатора). Ідентифікатори використовуються для іменування змінних і таблиць значень. LUA є мовою, чутливою до регістру символів. LUA є мовою з динамічним визначенням типів даних. Змінна мови може містити значення будь-якого типу. Можливості визначення призначених для користувача типів даних відсутні. Всі значення в LUA можуть зберігатися в змінних, використовуватися в якості аргументів при виклику функцій і повертатися у вигляді результату їх виконання. У LUA можна використовувати функції, написані на LUA і на C [15].

## 2.6 Типове використання модуля ESP8266

ESP8266 можна використовувати в розумних розетках, mesh-мережах, IP-камерах, бездротових сенсорах, що носить електроніці і так далі.

Передбачено два варіанти використання чіпа:

- у вигляді моста UART-WI-FI, коли модуль на базі ESP8266 підключається до існуючого рішення на базі будь-якого іншого мікроконтролера і управляється AT-командами, забезпечуючи зв'язок рішення з інфраструктурою Wi-Fi;
- реалізації нового рішення, яке використовує сам чіп ESP8266 в якості керуючого мікроконтролера [18].

Основне застосування ESP8266 знаходить в управлінні різноманітними побутовими приладами через бездротові мережі. Існує кілька популярних реалізацій концепції IoT в плані обміну даними по мережі:

### HTML сервер на ESP8266

Контроль і управління пристроєм ведеться через браузер. Рішення, підходить для автономних пристроїв автоматки.



AllJoyn – набирає популярність відкритий IoT протокол великого альянсу виробників цифрової техніки «Allseen».

### MQTT

Це простий протокол поверх TCP / IP. Дуже популярне рішення. Існує велика кількість IoT додатків верхнього рівня для Android, iOS і інших платформ, що підтримують цей протокол.

### SNMP

Розширюваний протокол управління мережевими пристроями. Основний недолік в тому що в більшості мереж файрволли блокують проходження SNMP.

ModBus і інші протоколи промислової автоматизації.

Модулі ESP8266 – це дуже дешеве рішення для побудови мережі розумного будинку та іншої домашньої автоматизації з використанням Wi-Fi. Але для роботи в якості самостійного контролера ESP8266 має замало ресурсів. В цілому ESP8266 мають великі перспективи [19].

## 2.7 Протоколи і стандарти зв'язку

1-Wire – основною магістраллю для передачі даних тут виступає двохнаправлена шина, яка виглядає як двожильний провід. Один провід використовується для живлення і передачі даних, інший - для заземлення. Топологія мережі - загальна шина. Головна перевага даного стандарту – дешевизна і невибагливість.

X-10 – для передачі сигналу використовується електропроводка будівлі. Також передбачена можливість використання трансиверів, які ловлять радіосигнал від бездротових пристроїв, перетворюють його в потрібний формат і передають в електричну мережу. Призначений для взаємодії з датчиками і пультами дистанційного керування. Серйозним недоліком X10 є низька швидкість передачі даних.

KNX – дорогий варіант, який користується популярністю в Європі. Як середовище для передачі даних протокол KNX може використовувати шину

(кручену пару), електричну мережу або радіоканал. Стандарт передбачає різні варіанти топології мережі. Система зобов'язана мати власне джерело живлення, але при цьому в ній може бути відсутнім центральний контролер, так, як KNX дозволяє створювати децентралізовані рішення, в яких сенсори і виконавчі модулі взаємодіють безпосередньо.

WI-FI – стандарт протоколу IEEE 802.11 на обладнання для широкосмугового радіозв'язку, призначеної для організації локальних бездротових мереж.

ZigBee – протокол зв'язку по радіоканалу. Стандарт дозволяє створювати датчики з низьким енергоспоживанням і чудовою чуйністю. ZigBee підтримує комірчасту топологію мережі, при якій окремі компоненти можуть виступати в якості посередника, що передає сигнал від одного пристрою до іншого. Mesh-мережі також дозволяє істотно збільшити зону покриття бездротової мережі.

Z-Wave – протокол бездротового зв'язку, який багато в чому схожий з ZigBee – тут і низьке енергоспоживання, і підтримка комірчастої топології мережі. Ці стандарти також можна порівняти за вартістю обладнання. Z-Wave працює в діапазоні частот до 1 ГГц і оптимізована для передачі простих керуючих команд з малими затримками (наприклад, включити / виключити, змінити гучність, яскравість і т.д.)

## 2.8 Протокол MQTT

MQTT (Message Queuing Telemetry Transport) легкий і простий протокол обміну повідомленнями, який реалізує модель «публікація / підписка» (publish / subscribe) і призначений для зв'язку комп'ютеризованих пристроїв, підключених до локальної або глобальної мережі, між собою і різними громадськими чи приватними веб-сервісами. Схема роботи протоколу MQTT представлена на рисунку 2.3.

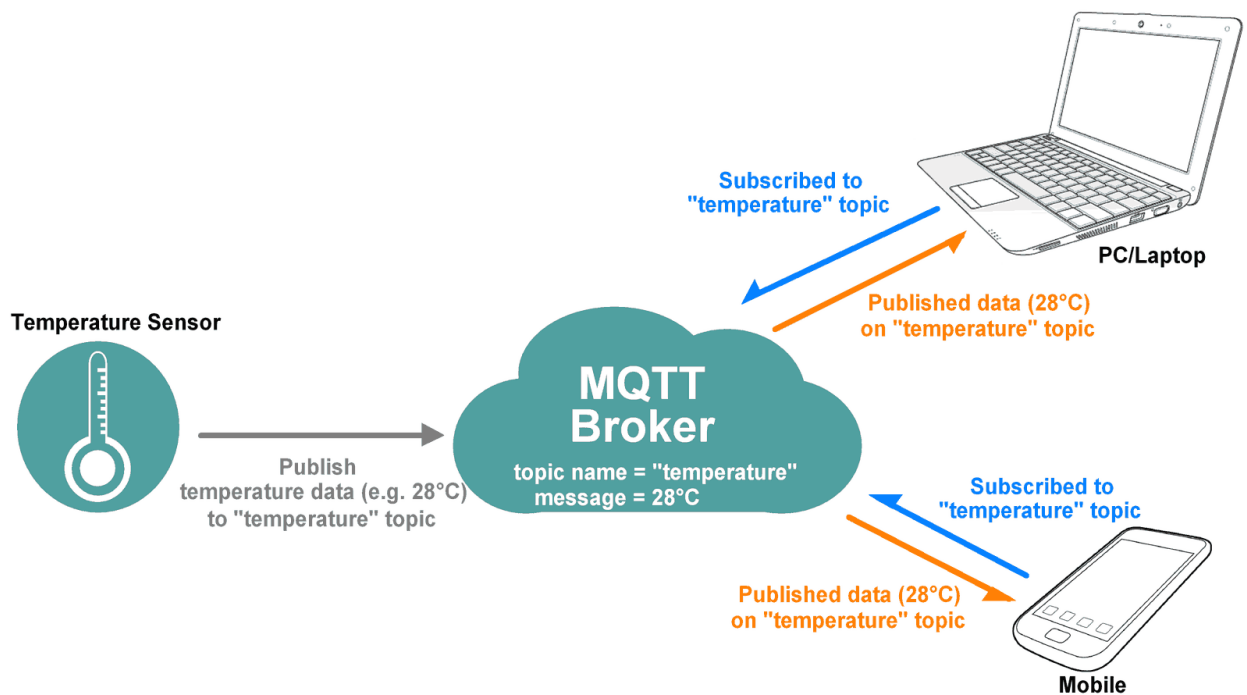


Рисунок 2.14 – Схема роботи протоколу MQTT

Протокол MQTT спочатку був створений для датчиків, які відстежують стан труб, проте пізніше сфера його діяльності була розширена і він знайшов своє застосування в безлічі вбудованих рішень, в тому числі в смартфонах. Так соціальна мережа Facebook застосовує цей протокол для обміну повідомленнями (Facebook Messenger).

У мережі на базі протоколу MQTT розрізняють 3 об'єкти:

1) видавець (publisher) -MQTT-клієнт, який при виникненні певних подій передає інформацію про неї в брокер;

2) брокер (broker) -MQTT-сервер, який приймає інформацію від видавців і передає її відповідним передплатникам, в складних системах може виконувати також різні операції, пов'язані з аналізом і обробкою даних, що надійшли;

Брокери можуть бути налаштовані мостом як показано на рисунку 2.4. Наприклад брокер А може перенаправляти повідомлення, що прийшли на певний канал, брокеру В. Таким чином можна масштабувати мережу управління.

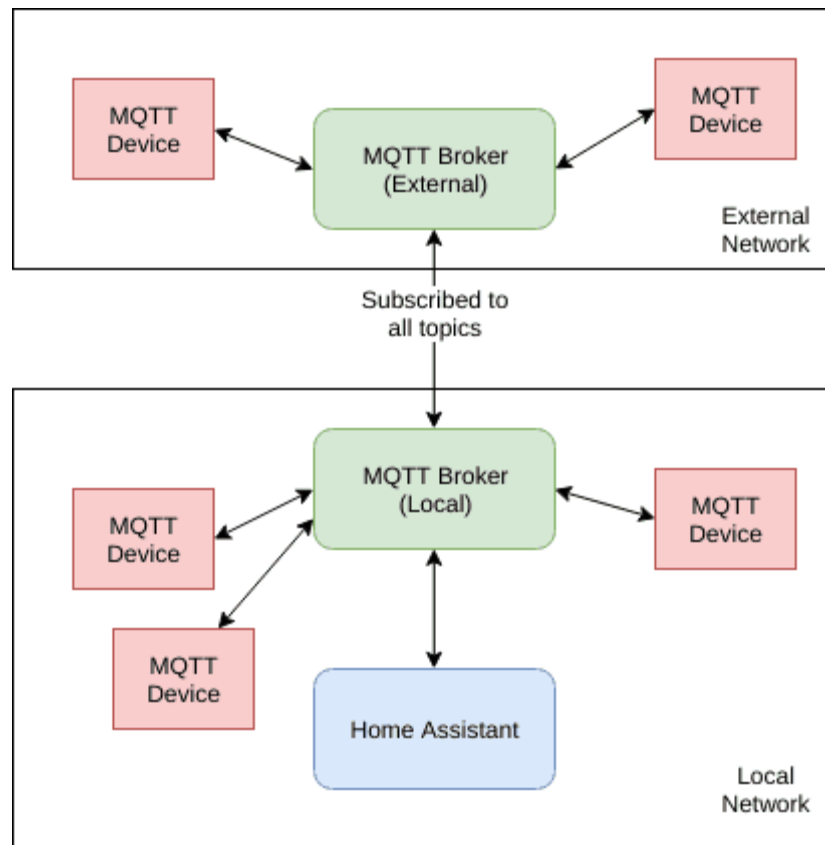


Рисунок 2.15 – З'єднання у вигляді моста

3) передплатник (subscriber) -MQTT-клієнт, який після підписки у відповідного брокера більшу частину часу «слухає» його і постійно готовий до прийому і обробки вхідного повідомлення від брокера.

Специфікація MQTT відкрита і доступна в Інтернеті. В даний час є два варіанти специфікації MQTT:

- MQTT v3.1 - основна специфікація для мереж на базі TCP / IP;
- MQTT-S v1.2 для датчиків і вбудованих пристроїв в мережах, відмінних від TCP / IP, наприклад ZigBee.

MQTT вже використовується в роботі супутників, а також в медицині та деяких промислових сферах.

Основні переваги протоколу MQTT:

- невеликі накладні витрати на транспортному рівні (заголовок фіксованого розміру довжиною 2 байт);

- протокол обміну зведений до мінімуму для зменшення мережевого трафіку;
- вбудований механізм контролю з'єднання.

Протокол MQTT має ряд переваг, порівняно з протоколом HTTP:

- менші накладні витрати на передачу даних і менша смуга пропускання;
- для своєї роботи він не вимагає постійного з'єднання між клієнтом і сервером (як у випадку HTTP).

MQTT також добре адаптований до роботи по каналах зв'язку з низькою пропускнуою здатністю [14].

## 2.9 Висновок

Здійснено огляд і аналіз підходів автоматизації об'єктів і процесів за допомогою Wi-Fi модулів, а саме сімейства мікроконтролера ESP8266 та їх прошивок. Розглянуто початкового завантаження і оновлення прошивки, існуючих програмних засобів розробки, зокрема мови програмування LUA. Детально розглянуто типове використання модуля ESP8266. Також оглянуто протоколи і стандарти зв'язку, зокрема протокол MQTT.

### 3 РЕАЛІЗАЦІЯ ТЕХНОЛОГІЇ АВТОМАТИЗАЦІЇ ОБ'ЄКТІВ І ПРОЦЕСІВ ЗА ДОПОМОГОЮ WI-FI МОДУЛІВ

Модуль ESP8266 може використовуватися як елемент управління об'єктами і процесами. Дистанційно вмикати або вимикати будь-якої електросистеми зі смартфона, ноутбука, ПК, планшета. А також використовувати для опрацювання сигналів із датчиків, опрацьовувати та перевати у канал з'єднання. З'єднання з пристроями здійснюється по бездротовому зв'язку стандарту Wi-Fi.

#### 3.1 Розробка структурно-функціональної схеми та вибір електронних компонентів

Розглянемо структурно-функціональну схему, яку показано на рисунку 3.1.

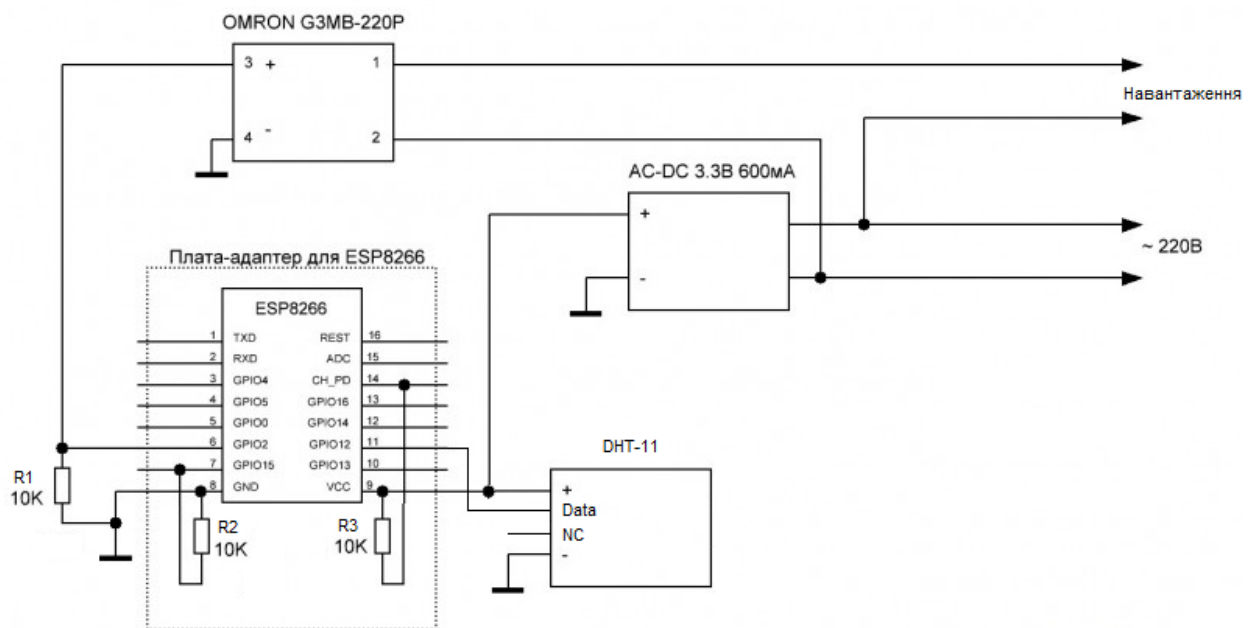


Рисунок 3.1 – Структурно-функціональна схема системи

У якості навантаження може виступати будь-який електроприлад, потужність якого не перевищує максимальне навантаження обраного реле. Реле можна використовувати як контактне так і твердотільне. Перевага

твердотільного реле в тому, що воно має компактні розміри в порівнянні з контактним.

Система складається з:

- Перетворювача напруги (що складається з понижуючого трансформатора і випрямляча з виходом напруги на виході 3.3 вольт, для живлення модуля ESP);
- Твердотільного реле Omron G3MB-220P (має компактні розміри, максимальний струм навантаження 2А);
- Датчика температури і вологості DHT11;
- Самого модуля ESP8266;
- Резисторів R1, R2, R3 (резистори R1, R3 - можуть бути номіналом від 4,7 кОм до 50 кОм, а резистор R2 - номіналом до 10 кОм.) Підключення цих резисторів необхідна вимога виробника для забезпечення стабільної роботи пристрою.

Конфігурація виводів модуля може бути різною, все залежить від того які завдання буде виконувати даний модуль.

У системі модуль ESP8266 налаштований таким чином - виводи GPIO0, GPIO2, GPIO4, GPIO5, GPIO14, GPIO16 будуть призначені для підключення реле а виводи GPIO12, GPIO13, GPIO15 - для підключення датчиків DHT. На схемі реле підключено до висновку GPIO2 модуля. Аналогічно можна підключити реле до інших виводів. Датчик DHT11 підключено до виводу GPIO12.

Вибір електронних компонентів, орієнтовано на технічні характеристики та може бути змінено, залежно від задач. У реалізованому випадку було обрані наступні компоненти:

DHT – цифровий датчик температури і вологості, який показано на рисунку 3.2, містить калібрований цифровий вихідний сигнал з показаннями температури і вологості. Датчик включає в себе резистивний сенсор вологості і компоненти NTC структури для вимірювання температури. Датчик DHT складається з ємнісного датчика вологості і термістора. Також,

датчик містить в собі простенький АЦП для перетворення аналогових значень вологості і температури.

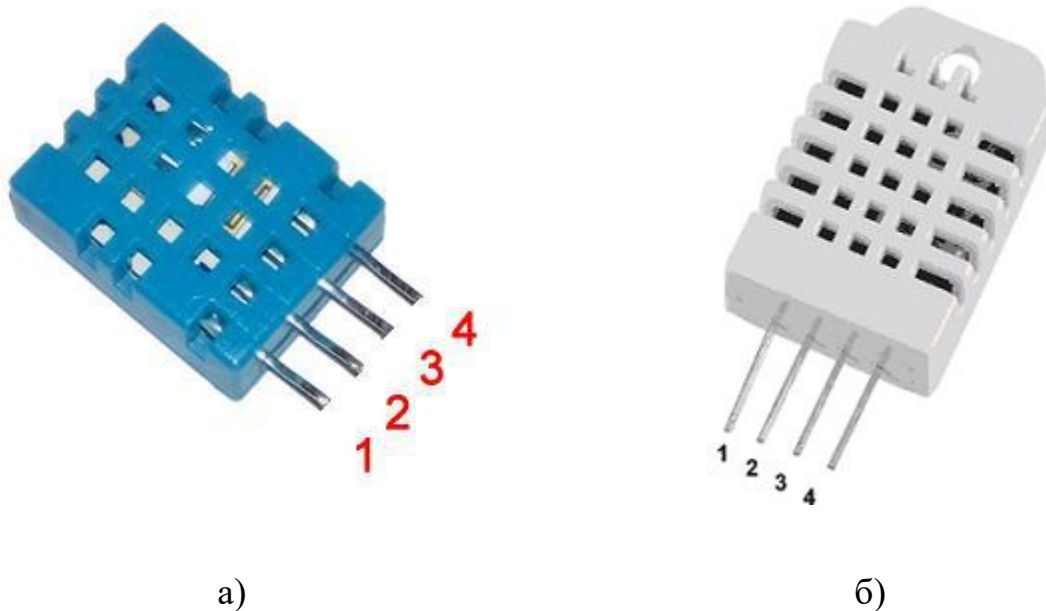


Рисунок 3.2 – Датчик температури і вологості:  
а) DHT11; б) DHT22.

- Vcc - живлення +3,5 ~ 5,5 В;
- Data - передача даних;
- NC - не використовується;
- GND - земля(-).

Розглянемо 2 види недорогих цифрових датчика температури і вологості DHT11 і DHT22.

По розпіновці дані датчики ідентичні, проте трохи розрізняються характеристиками.

DHT11:

- Низька вартість;
- Живлення I/O 3-5 В;
- Визначення вологості 20-80% з 5% точністю;
- Визначення температури 0-50 град. з 2% точністю;



- Частота опитування не більше 1 Гц (не більше одного разу в 1 с);
- Розміри 15.5мм x 12мм x 5.5мм;
- 4 виводи з відстанню між ніжками 0.1"

#### DHT22:

- Живлення і I/O 3-5 В;
- Визначення вологості 0-100% з 2-5% точністю;
- Визначення температури -40 - 125 град. з точністю  $\pm 0.5$  град.;
- Частота опитування не більше 0.5 Гц (не більше одного разу в 2 с);
- Розміри 15.1мм x 25мм x 7.7мм;
- 4 виводи з відстанню між ніжками 0.1 "

Сенсор DHT22 має більш кращі характеристики. Обидва датчика повільні і зчитувати з них інформацію можливо не частіше, ніж раз в 1 або 2 с. [16].

В якості комутуючого елемента використано твердотільні реле Omron зовнішній вигляд якого показано на рисунку 3.3.

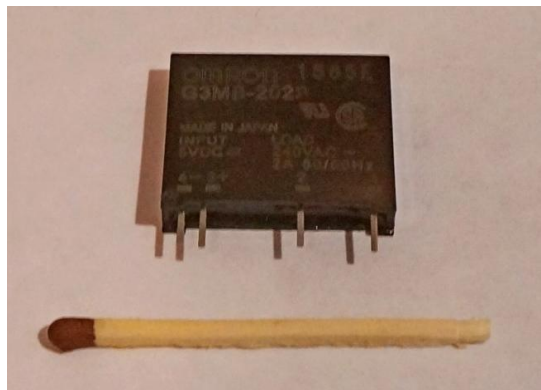


Рисунок 3.3 – Твердотільне малогабаритне реле Omron

#### Характеристики:

- Серія: G3MB;
- Канали: SPST-NO (1 Form A);
- Тип виходу: AC, Zero Cross;

- Струм навантаження: 2 А;
- Напруга керуючого входу: 4 ~ 6 VDC;
- Напруга - навантаження: 75 ~ 264 V;

Також можна використовувати контактна реле, наприклад SRD-05VDC-SL-C , зовнішній вигляд якого показано на рисунку 3.4.



Рисунок 3.4 – Контактна реле SRD-05VDC-SL-C

Характеристики:

- Контакти 1 x DT;
- Номінальна напруга котушки 5 В;
- Номінальний струм котушки 71,4 мА;
- Максимальна напруга комутації (AC) 250 В;
- Максимальний комутований струм 10 А.

## 3.2 Розробка програмного забезпечення

### 3.2.1 Вибір варіанту реалізації програмного забезпечення і середовища розробки

Існує 3 варіанти реалізації:

1. Розробка програмного забезпечення модуля на мові C ++ з використанням бібліотек SDK фірми Espressif System і бібліотек сторонніх розробників.

2. Розробка програмного забезпечення на скриптовій мовою Lua з використанням готової операційної системи NodeMSU, яка має свою

файлову систему SPIFFS, вбудований інтерпретатор і вмiє виконувати Lua скрипти з пам'ятi пристрою.

3. Розробка програмного забезпечення модуля на мовi C ++ з використанням багатозадачної операцiйної системи реального часу FreeRTOS написаної на мовi C, з відкритим вихідним кодом. Дане рiшення дозволяє створювати кiлька процесiв виконують свої завдання.

Обрано варіант з використанням операцiйної системи NodeMcu і мови програмування Lua. Так ,як цей варіант є найбільш швидким і простим у плані реалізації поставленого завдання. Але у нього є свої мiнуси. Він не підійде для реалізації проектiв в яких потрібно забезпечити багатозадачність даного модуля, через нехватку пам'ятi.

Виконання поставлених завдань передбачає розробку внутрішнього програмного забезпечення контролера з підтримкою протоколу MQTT. Для реалізації обміну даними між кінцевим WI-FI пристроєм і сервером.

Найбільш оптимальною і зручною базою для розробки ПО є скриптова мова LUA. Необхідно створити ПО для цього мікроконтролера складається з скриптiв написаних на мовi LUA. Але для того щоб ці скрипти виконувалися потрібно прошити модуль прошивкою з вбудованим інтерпретатором мов високого рівня. Така прошивка дозволяє довантажувати через UART і виконувати скрипти розробника пристрою.

В даному проекті використовується прошивка NodeMCU. Прошивка вмiє виконувати Lua-скрипти як з UART (аналогічно AT-командам) так і з внутрішньої flash пам'ятi. Для завантаження скриптiв в flash пам'ять підтримується файлова система spiffs. Можна створити безліч скриптiв на LUA і записати їх в файлову систему. Після чого через WI-FI можна викликати потрібні скрипти на виконання.

Основний недолік в тому, що VM LUA виконує LUA скрипти розміщення в оперативній пам'ятi кристала. А цієї пам'ятi для скриптiв всього 20 Кбайт. Цього обсягу пам'ятi вистачає на виконання скрипта

приблизно в 110 рядків. Тому створення порівняно великих скриптів для даного модуля має свою специфіку:

- Необхідно алгоритм розділити на лінійні блоки;
- Записати ці блоки в окремі файли файлової системи модуля;
- Виконувати ці модулі за допомогою оператора `do file`.

При написанні модулів треба дотримуватися наступних правил:

- В кінці кожного модуля потрібно викликати збирач сміття;
- Для обміну даними між модулями використовувати глобальні змінні, а для обчислень всередині модулів – локальні.

Прийом і передача даних від модуля до сервера здійснюється за рахунок мережі WI-FI по протоколу MQTT.

Основні функції, які повинен виконувати модуль:

1. Ініціалізація пристрою.
2. Прийом команд від сервера (брокера).
3. Подача сигналу на відповідні виводи модуля для управління елементами;
4. Читання даних, отриманих з кінцевих пристроїв;
5. Передача даних (стану) на сервер (брокер);
6. Періодичне інформування сервера (брокера) про функціонування пристрою.

### 3.2.2 Алгоритм роботи пристрою

При включенні живлення відбувається ініціалізація пристрою. На цьому етапі здійснюється підключення модуля до існуючої WI-FI мережі, запускається `mqtt` - клієнт, який здійснює пінг брокера, відсилення стану виводів з інтервалом часу 2 сек., і прослушку порту 1884 для прийому повідомлень від брокера. Інтервал часу 2с виставлений як тестовий, надалі його можна змінити в залежності від розв'язуваної задачі.

При появі команди від брокера відбувається її виконання і пристрій повертається в режим очікування команди. Якщо команди немає пристрій продовжує знаходитися в режимі очікування.

### 3.2.3 Опис режимів роботи

Контролер може працювати як самостійний пристрій при використанні вбудованого веб-сервера, який розгортається в пам'яті пристрою. Управління виводами здійснюється через веб-інтерфейс по протоколу HTTP, за допомогою get-запитів, підключившись безпосередньо до самого модулю за його IP-адресою. При підключенні до модулю відкривається веб сторінка на якій відображені найменування виводів контролера а на проти з права розташовуються 2 кнопки ON і OFF відповідно включити і вимкнути.

Або контролер може бути елементом системи управління в якій прийом і передача даних здійснюється через MQTT-брокер по протоколу MQTT. У цьому випадку модуль постійно в заданий інтервал часу відсилає на сервер свій ідентифікаційний номер присвоєний йому виробником, стан виводів (у вигляді 1 або 0 вкл або викл), і інформацію з датчика температури і вологості в форматі об'єкта json. І здійснює прослушку порту 1884 для прийому повідомлень від сервера. Управління здійснюється з системи управління через веб інтерфейс, команда посилається через брокер у вигляді повідомлення в гілку / NodeMsu / 4ON або 4OFF (4 - номер виводу, ON / OFF - вкл. Викл.), Так і з додатка під андроїд зі смартфона або планшета.

### 3.2.4 Опис модулів програмного забезпечення контролера

ПО модуля складається з 8 файлів (скриптів):

Найголовніший файл прошивки – це файл ініціалізації init.lua (Додаток А). Він починає виконуватися після подачі живлення або рестарту модуля. У ньому описуються основні дії, які треба зробити перед виконанням основного блоку і, відповідно, передається управління на основний блок. Інші файли можуть бути названі довільно.

config.lua - конфігураційний файл в якому вказується SSID точки доступу її пароль, IP адреса брокера, порт брокера, гілки MQTT куди посилаємо і з яких зчитуємо інформацію, вказуються виводи до яких підключаємо реле і виводи до яких підключаємо датчики ДНТ (Додаток Б).

`application.lua` – модуль, що виконує основну логіку програми і MQTT протокол. Посилає пінг і свій ID брокера, стан виводів контролера, дані з датчиків температури і вологості, чекає повідомлень від брокера (Додаток В).

`Wi-Fi_connect.lua` – модуль який здійснює підключення до існуючої Wi-Fi мережі (Додаток Г).

`setup.lua` – першочергове налаштування виходів модуля при включенні. Всі виходи знаходяться в режимі включено (Додаток Г).

`webserver.lua` – модуль який розгортає веб-сервер на базі мікроконтролера для можливості керувати виводами через веб інтерфейс по протоколу HTTP за допомогою get запитів (Додаток Д).

`io_module.lua` – модуль управління цифровими входами, тобто модуль який здійснює включення або виключення виводів контролера (Додаток Е)

`dht11.lua` – модуль роботи з датчиком. В даному модулі проводиться опитування датчика Датчик DHT11 працює по інтерфейсу 1-wire (Додаток Є).

### 3.2.5 Опис конфігурації модуля

Конфігурація виводів контролера налаштована таким чином.

Піни для реле:

`module.IO0 = 0 - GPIO16`

`module.IO1 = 1 - GPIO5`

`module.IO2 = 2 - GPIO4`

`module.IO3 = 3 - GPIO0`

`module.IO4 = 4 - GPIO2`

`module.IO5 = 5 - GPIO14`

Піни для датчиків DHT11

`module.IO6 = 6 - GPIO12`

`module.IO7 = 7 - GPIO13`

GPIO0-GPIO16 - позначення виводів контролера, а цифри від 0 до 7 - призначення номера виводу для управління. (Наприклад позначення виведення 0 відповідає маркуванню виведення контролера GPIO16 і т.д.) При

включенні модуля або рестарті за замовчуванням всі виводи знаходяться в стані включено.

### 3.2.6 Опис протоколу взаємодії з системою управління

Інформація з датчика DHT і стан виводів контролера відсилається у форматі об'єкта json.

Датчик DHT11 працює по інтерфейсу 1-wire (тобто одна лінія для передачі даних в ту і в іншу сторону за часом.)

Формат передачі даних який розсилає датчик:

- 8 біт – вологість ціла частина.
- 8 біт – вологість дрібна частина.
- 8 біт – температура ціла частина.
- 8 біт – температура дрібна частина.
- 8 біт контрольна сума.

Таким чином одна посилка даних з датчика - це 40 біт. Потім їх читаємо по порядку, перевіряємо контрольну суму, яка вважається як сума перших 4х байт, тобто: вологість ціла + вологість дробова + температура ціла + температура дробова і порівнюється з останнім байтом контрольної суми.

Формат об'єкта json виглядає наступним чином:

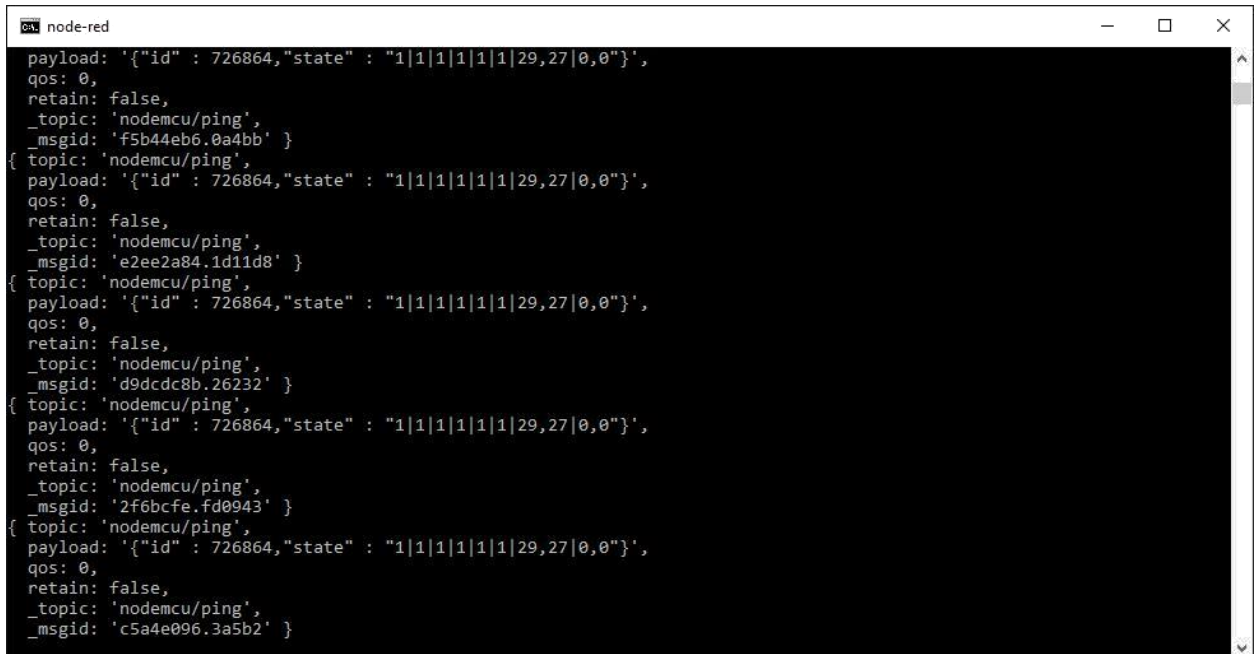
```
{ "id": 726864, "state ":" 1 | 1 | 1 | 1 | 1 | 1 | 29,27 | 0,0 }
```

Де "id": 726864 – постійний ідентифікаційний номер пристрою присвоєний йому виробником.

1 | 1 | 1 | 1 | 1 | 1 | – стан виводів контролера сконфігурованих під реле, нумерація виводів здійснюється з ліва на право тобто 0,1,2,3,4,5 (1 - означає ввімкнено, 0 - вимкнено). Решта 2 виходи 6 і 7 сконфігуровані під прийом інформації з датчиків температури і вологості DHT.

| 29,27 | – 29 це відображення температури, 27 – показник вологості з датчика DHT. На 7 виводів відображаються нулі тому, що до цього виходу не підключений датчик.

На рисунку 3.5 показаний приклад прийому повідомлень від контролера mqtt клієнтом node-red, який є обробником даних формату json, системи управління.



```

node-red
payload: '{"id" : 726864,"state" : "1|1|1|1|1|1|29,27|0,0"}',
qos: 0,
retain: false,
_topic: 'nodemcu/ping',
_msgid: 'f5b44eb6.0a4bb' }
{
  topic: 'nodemcu/ping',
  payload: '{"id" : 726864,"state" : "1|1|1|1|1|1|29,27|0,0"}',
  qos: 0,
  retain: false,
  _topic: 'nodemcu/ping',
  _msgid: 'e2ee2a84.1d11d8' }
{
  topic: 'nodemcu/ping',
  payload: '{"id" : 726864,"state" : "1|1|1|1|1|1|29,27|0,0"}',
  qos: 0,
  retain: false,
  _topic: 'nodemcu/ping',
  _msgid: 'd9dcdc8b.26232' }
{
  topic: 'nodemcu/ping',
  payload: '{"id" : 726864,"state" : "1|1|1|1|1|1|29,27|0,0"}',
  qos: 0,
  retain: false,
  _topic: 'nodemcu/ping',
  _msgid: '2f6bcfe.fd0943' }
{
  topic: 'nodemcu/ping',
  payload: '{"id" : 726864,"state" : "1|1|1|1|1|1|29,27|0,0"}',
  qos: 0,
  retain: false,
  _topic: 'nodemcu/ping',
  _msgid: 'c5a4e096.3a5b2' }

```

Рисунок 3.5 – Прийом повідомлень від контролера

### 3.3 Підготовка модуля до роботи

Порядок прошивки модуля наступний:

1. Необхідно завантажити прошивку NodeMCU.
2. Завантажити програму для зміни прошивки XTCOM\_UTIL або ESP8266Flasher.
3. Підключити модуль через перетворювач USB-COM до ПК.

Для заливки прошивки вивід GPIO0 повинен бути «підтягнутим» до GND.

Схема підключення показана на рисунку 3.6.

4. Запускаємо програму XTCOM\_UTILITY, в меню Tools вибираємо Config Device. Вибираємо COM порт і швидкість 115200. Програма підтримує максимально COM6. Якщо перетворювач визначився як COM9 або 8, то треба його замінити в диспетчері пристроїв. Диспетчер пристроїв - Параметри порту - Додатково.



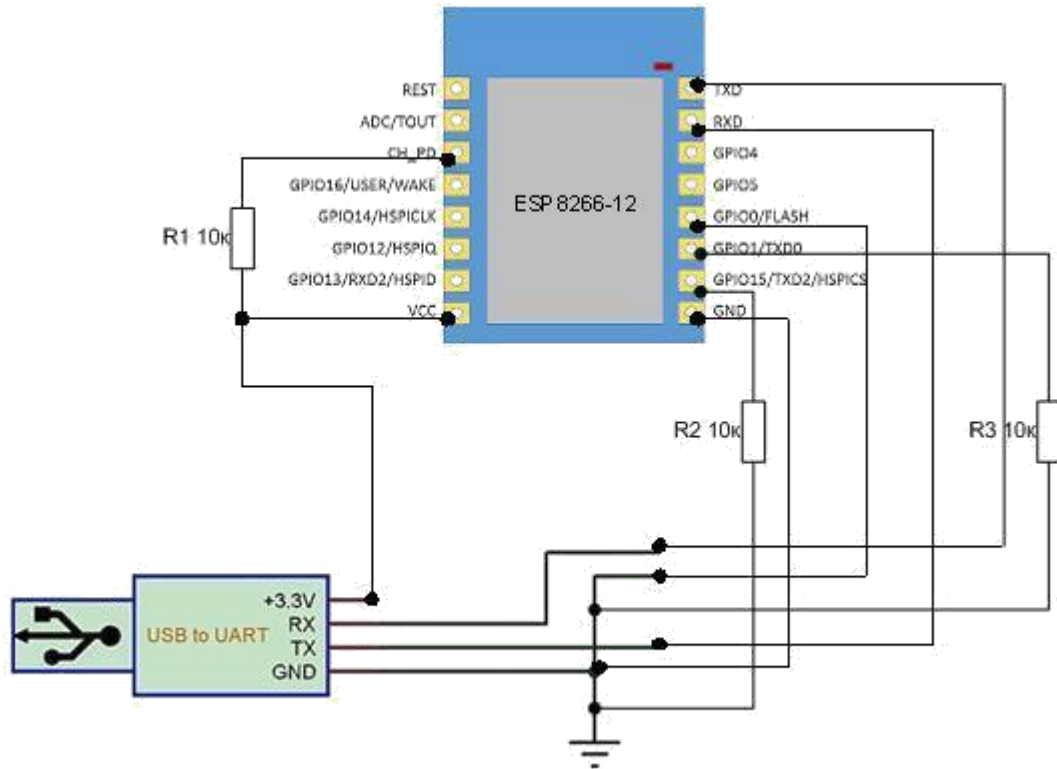


Рисунок 3.6 – Схема підключення модуля до USB to UART при прошивці

Успішне з'єднання з контролером показано на рисунку 3.7.

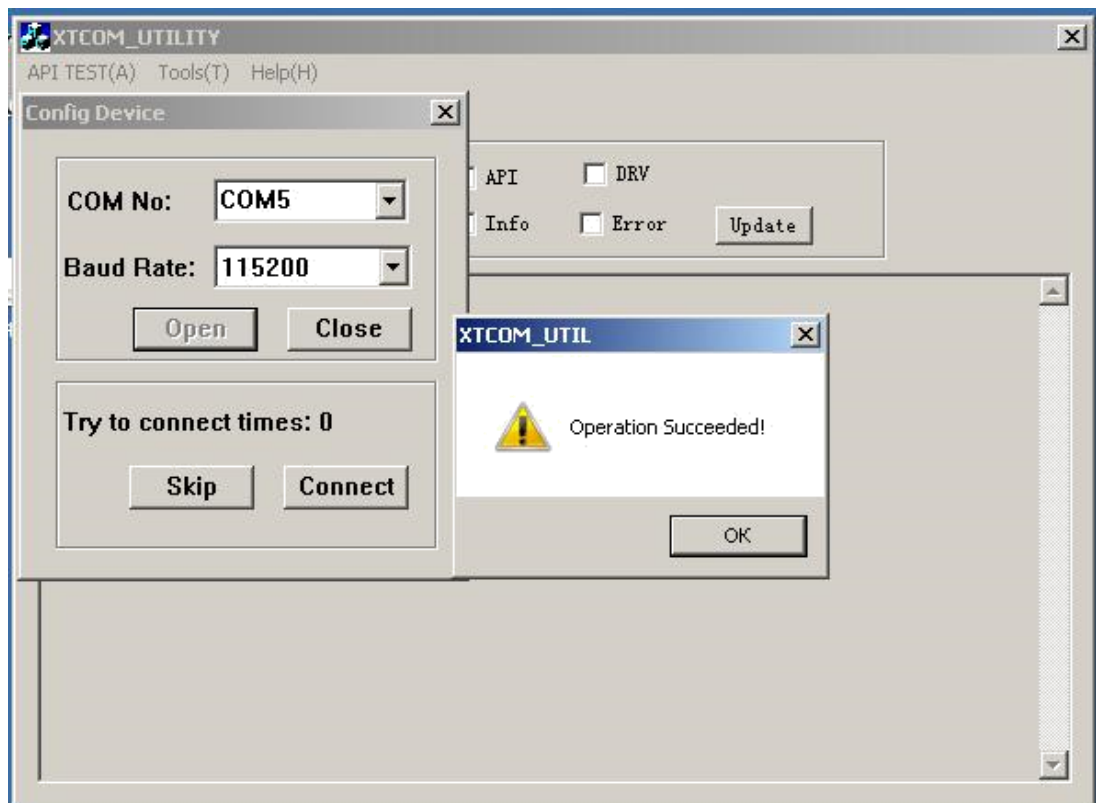


Рисунок 3.7 – Успішне з'єднання з контролером

5. Тепер в API TEST (A) вибираємо (4) Flash Image Download. Вказуємо шлях до файлу прошивки, адресу залишаємо 0x00000, тиснемо Download і чекаємо закінчення процесу прошивки (рисунок 3.8).

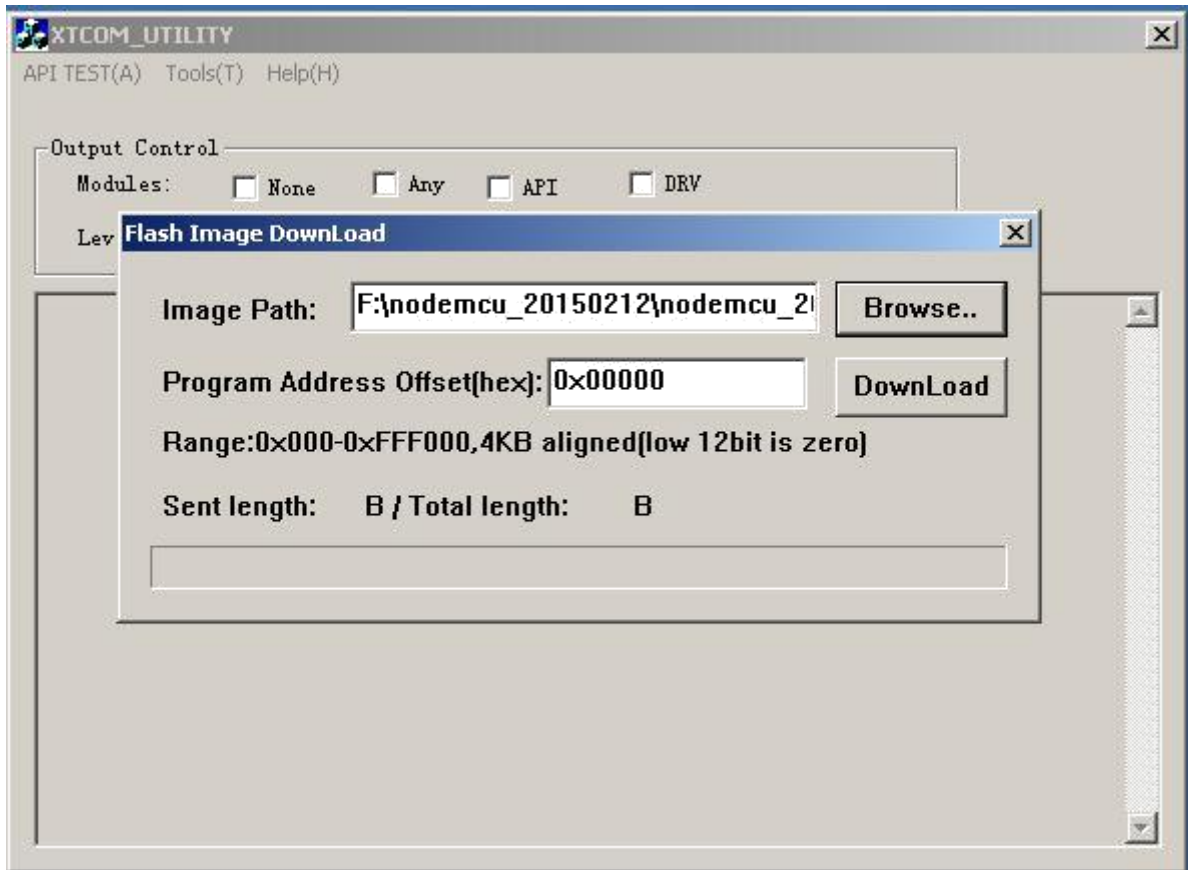


Рисунок 3.8 – Початок прошивки

Тепер відключимо GPIO0 від GND. Перевантажимо модуль живлення. Далі можна приступити до написання скриптів.

Середовище розробки може бути звичайний блокнот, FAR менеджер, але набагато зручніше використовувати спеціальну програму ESPloader, що дозволяє писати скрипти і завантажувати їх на флеш пам'ять контролера. Але перед її використанням потрібно встановити JAVA, тому що програма працює з використанням даного середовища. Інтерфейс роботи програми представлений на рисунку 3.9.

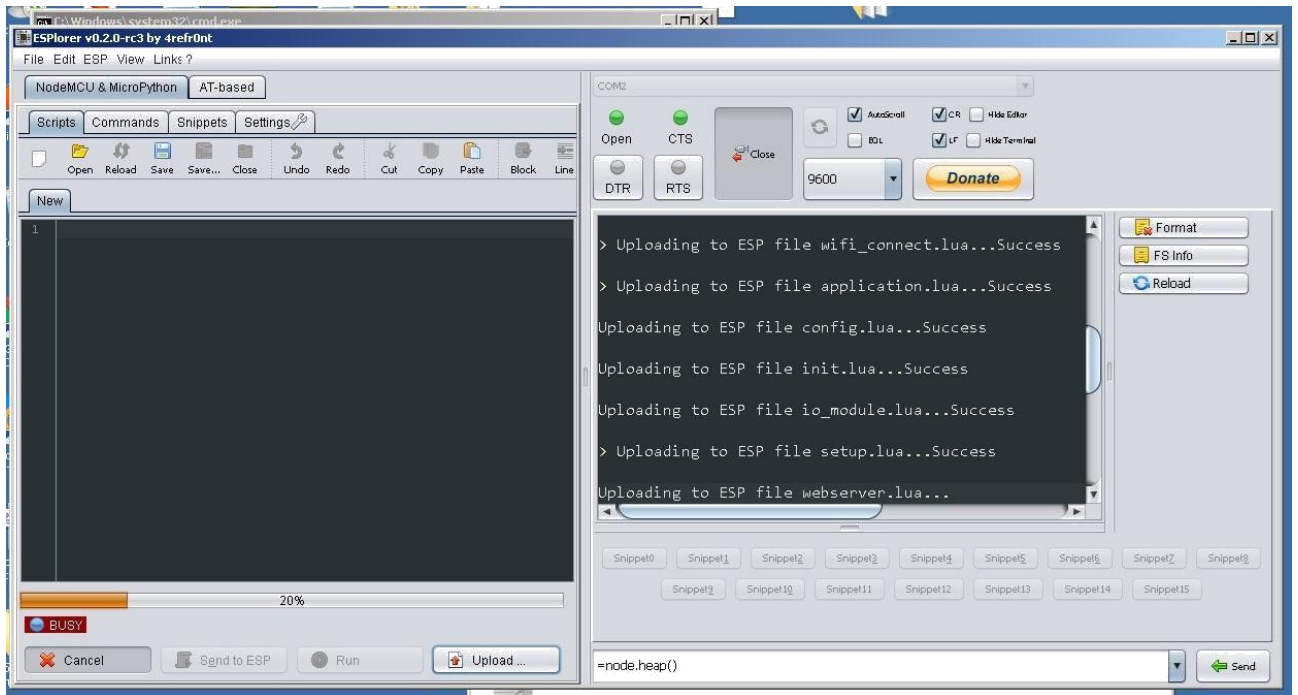


Рисунок 3.9 – Завантаження скриптів в пам'ять мікроконтролера за допомогою програми ESPlorer

Справа зверху вибираємо COM порт і натискаємо Open. Нижче знаходиться монітор порту, де видно команди, що посилаються в модуль і його відповіді. Ще нижче список деяких команд, які можна вибрати і послати в модуль кнопкою Send. Зліва закладки з файлами прошивки. Їх можна додавати, видаляти, коригувати. Під файлами розташовані кнопки для управління модулем.

### 3.4 Загальний опис роботи і взаємодія програмних модулів

При подачі живлення на модуль починає виконуватися файл ініціалізації `init.lua`, в якому описані основні дії, які потрібно виконати перед виконанням основного блоку. Цей скрипт реалізує сценарій включення віддалених файлів. Далі виконується скрипт `setup.lua` в якому описана початкова конфігурація виходів GPIO контролера, потім стартує модуль підключення до Wi-Fi мережі. Здійснюється підключення до існуючої Wi-Fi мережі відповідно до параметрів заданими в файлі конфігурації.

В файлі конфігурації вказані параметри точки доступу до якої підключається модуль (ssid і пароль), а також налаштування для підключення до брокера (його IP-адреса, порт і гілки за якими він буде відправляти і отримувати повідомлення від брокера).

Далі паралельно стартують веб-сервер і додаток, що виконує основну логіку програми і MQTT протоколу application.lua – (запускається MQTT – клієнт, створюється дві гілки прийому і передачі даних, модуль починає пінгувати MQTT – брокер за IP адресою вказаною у файлі конфігурації і відсилати йому свій ідентифікаційний номер ID, інформацію про стан виходів і інформацію отриману з датчика DHT в форматі json, а друга гілка переходить в режим прослуховування порту 1884, для прийому повідомлень від брокера).

Команда від системи управління надходить в форматі повідомлення на гілку (топик) /NODEMSU /

Формат повідомлення такої /NODEMSU/2ON або /NODEMSU/2OFF Цифра означає номер виводу контролера, ON/OFF – ввімкнути/ вимкнути.

Робота з модулем через Web-сервер.

В адресному рядку браузера набираємо IP-адресу модуля, відкривається HTML сторінка на якій відображені всі сконфігуровані виводи контролера. Відобразатися вони можуть по різному все залежить від того як їх названо. Праворуч від назви виведення дві кнопки ON і OFF ввімкнути і вимкнути. Дизайн, фон сторінки і оформлення кнопок при необхідності можна змінити у файлі конфігурації Web-сервера.

Web-сервер передбачений на той випадок якщо неможливо з якихось причин підключитися через систему управління.

### 3.5 Висновок

Здійснено реалізацію технології автоматизації об'єктів і процесів за допомогою Wi-Fi модулів, для чого розроблено структурно-функціональну схему та обрано електронні компоненти. Також розроблено програмне

забезпечення на основі обраного варіанту реалізації і середовища розробки, відповідно до алгоритму роботи. Здійснено опис режимів роботи модулів програмного забезпечення контролера, його конфігурації та протоколу взаємодії з системою управління. Здійснено підготовку модуля до роботи та описано роботу принцип роботи і взаємодія програмних частин.

## 4 ЕКОНОМІЧНА ЧАСТИНА

### 4.1. Оцінювання комерційного потенціалу розробки.

Метою проведення технологічного аудиту є оцінювання комерційного потенціалу розробки, створеної в результаті науково-технічної діяльності [22].

Результатом магістерської кваліфікаційної роботи «Технологія автоматизації об'єктів та процесів за допомогою мікропроцесорних WiFi модулів» є розробка апаратно-програмного забезпечення для реалізації технології автоматизації об'єктів та процесів за допомогою мікропроцесорних WiFi модулів. Для проведення технологічного аудиту залучено трьох незалежних експертів: Кадук Олександр Володимирович (к.т.н., доцент каф. комп'ютерної інженерії ВНТУ), Богомолів Сергій Віталійович (к.т.н., доцент каф. комп'ютерної інженерії ВНТУ) та Гарнага Володимир Анатолійович (к.т.н., доцент каф. комп'ютерної інженерії ВНТУ).

Оцінювання комерційного потенціалу буде здійснене за критеріями, що наведені в таблиці 4.1.

Таблиця 4.1 - Критерії оцінювання комерційного потенціалу розробки  
бальна оцінка

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
Кри- тері й	0	1	2	3	4
Технічна здійсненність концепції:					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено роботоздатність продукту в реальних умовах
Ринкові переваги (недоліки):					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни

			аналогів		аналогів
--	--	--	----------	--	----------

## Продовження таблиці 4.1

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
Кри-тер.	0	1	2	3	4
4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
Ринкові перспективи					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкуренція немає
Практична здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витратити значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві

Продовження таблиці 4.1

11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

Результати оцінювання комерційного потенціалу експертами розробки зведено в таблицю 4.2.

Таблиця 4.2 - Результати оцінювання комерційного потенціалу розробки

Критерії	Прізвище, ініціали, посада експерта		
	1 – Кадук	2 – Богомолів	3 – Гарнага
	Бали, виставлені експертами:		
1	4	3	4
Ринкові переваги (недоліки):			
2	3	2	2
3	4	4	4
4	4	3	4
5	3	4	3
Ринкові перспективи			
6	2	2	2
7	3	4	3
Практична здійсненність			
8	4	3	4
9	3	2	3
10	4	4	4
11	3	4	3
12	3	3	3
Сума балів	СБ <sub>1</sub> =40	СБ <sub>2</sub> =38	СБ <sub>3</sub> =39
Середньоарифметична сума балів $\overline{СБ}$	39		



За даними таблиці 4.2 можна зробити висновок, щодо рівня комерційного потенціалу розробки. Зважимо на результат й порівняємо його з рівнями комерційного потенціалу розробки, що представлено в таблиці 4.3.

Таблиця 4.3 – Рівні комерційного потенціалу розробки

Середньоарифметична сума балів $\overline{СБ}$ , розрахована на основі висновків експертів	Рівень комерційного потенціалу розробки
0 – 10	Низький
11 – 20	Нижче середнього
21 – 30	Середній
31 – 40	Вище середнього
41 – 48	Високий

Рівень комерційного потенціалу розробки, становить 39 балів, що відповідає рівню «вище середнього».

Актуальність розробки полягає у необхідності реалізації новітніх технологій для побудови апаратно-програмного забезпечення засобів на автономних WiFi модулях.

Головними конкурентами на ринку є розробники мікропроцесорних систем (МПС) віддаленого керування (Arduino, MediaTek), а також мікрокомп'ютерів Raspberry, Odroid.

Перевагами МПС систем є низька вартість та компактні розміри, недоліками – обмежені ресурси процесора та пам'яті. Перевагами мікрокомп'ютерних систем є наявність потужного ядра, великої кількості пам'яті та додаткової периферії, недоліками – дещо завеликі розміри, висока вартість та складність використання. Ціни на МПС – 500-1000 грн, Мікрокомп'ютерів – 1000-10000 грн.

В якості аналога для розробки було обрано систему на базі Arduino Uno у поєднанні з Wi-Fi та GSM шилдом.

Недоліком є обмежені ресурси процесора та пам'яті, великі габаритні розміри, у випадку використання GSM, необхідність поповнення рахунку SIM карти мобільного оператора. Також недоліком є те, що при потребі

розширення функціональних можливостей необхідно в придбанні додаткової периферії.

У таблиці 4.4 наведені основні технічні показники аналога і нового програмного продукту

Таблиця 4.4 - Основні технічні показники аналога і нового програмного продукту

Показники	Аналог	Нова розробка	Відношення параметрів нової розробки до параметрів аналога
Функціональність	3	6	2
Надійність	0,5	0,95	1,9
Сумісність	50	100	2
Супровід	2	8	4
Економія ресурсів і часу	ні	так	+
Простота використання	ні	так	+

Технологія автоматизації об'єктів та процесів за допомогою мікропроцесорних WiFi модулів є новим технічним рішенням, яке реалізовано на базі сучасного апаратного обладнання та програмного забезпечення, що реалізує новітні алгоритми роботи. Легкість використання та встановлення, зрозумілість інтерфейсу, мобільність та віддалений доступ зробить його конкурентоспроможним на ринку.

Вона може бути застосована в різних галузях господарської діяльності, де потребується віддалене керування об'єктами і процесами, а також у побуті та повсякденному житті. Таким чином, споживачами є широке коло зацікавлених осіб, які будуть використовувати продукт у своїх цілях, відповідно до своєї сфери діяльності.

Розробка значно спрощує повсякденне життя будь-якої людини. Надає можливість дистанційно керувати різними приладами. Технологія екологічно безпечна і дешева. Підходить для всіх верств населення.

Рівень здійсненості розробки (технічна готовність) 100%, існує дослідний зразок.

З метою комерціалізації розробки необхідно провести переговори з потенційними партнерами.

Комерціалізація розробки потребує незначних фінансових ресурсів, більшість матеріалів для її реалізації відомі і давно використовуються. На ринку праці наявні фахівці відповідної кваліфікації для обслуговування та підтримки програмного продукту, регламентні обмеження відсутні і немає необхідності отримання дозвільних документів.

Планується розробити власний web-сайт для рекламування та продажу продукту через мережу Інтернет. На сайті буде доступний зворотній зв'язок з розробником, куди можна буде писати зауваження та побажання стосовно розробки.

Також необхідно буде передбачити надання послуг з технічної підтримки продукту, якій буде поставлятися безпосереднім споживачам. Новому товару погрожує жорстка конкуренція, тому необхідно адресуватися до широкого кола споживачів, для того щоб довести переваги нової розробки і зацікавити споживачів придбати саме цей продукт. Для цього необхідно максимально використовувати рекламу в мережі Інтернет.

#### **4.2 Прогнозування витрат на виконання науково-дослідної, дослідно-конструкторської та конструкторсько-технологічної роботи**

Проведемо прогнозування витрат на виконання науково-дослідної, дослідно-конструкторської та конструкторсько-технологічної роботи для розробки програмного забезпечення, яке складається з таких етапів:

1-й етап: розрахунок витрат, які безпосередньо стосуються виконавців даного розділу роботи;

2-й етап: розрахунок загальних витрат на виконання даної роботи;

3-й етап: прогнозування загальних витрат на виконання та впровадження результатів даної роботи.

Виконаємо розрахунок витрат, які безпосередньо стосуються виконавця даного розділу роботи, приймаючи до уваги те, що для розробки програми було залучено одного розробника.

Основна заробітна плата розробника (дослідника)  $Z_o$ :

$$Z_o = \frac{M}{T_p} \cdot t \text{ [грн]}, \quad (4.1)$$

де  $M$  – місячний посадовий оклад розробника, 5000 грн.

$T_p$  – число робочих днів в місяці; приблизно  $T_p = (21 \dots 23)$  дні;

$t$  – число робочих днів роботи розробника (дослідника); розробка програмного забезпечення триває 3 місяців, тому кількість робочих днів - 66.

$$Z_o = \frac{5000}{22} \cdot 66 = 15000 \text{ (грн)}.$$

Додаткова заробітна плата  $Z_d$  розробника, розраховується як 10 % від суми основної заробітної, тобто:

$$Z_d = (0,1 \dots 0,12) \cdot Z_o \text{ [грн]}. \quad (4.2)$$

$$Z_d = 0,1 \cdot 15000 = 1500 \text{ (грн)}.$$

Нарахування на заробітну плату  $H_{zn}$  розробника становлять 22 % і розраховуються за формулою:

$$H_{zn} = (Z_o + Z_d) \cdot \frac{\beta}{100} \text{ [грн]}, \quad (4.3)$$

де  $Z_o$  – основна заробітна плата розробників, грн;  
 $Z_p$  – основна заробітна плата робітників, грн;  
 $Z_d$  – додаткова заробітна плата всіх розробників та робітників, грн;  
 $\beta$  – ставка єдиного соціального внеску, 22%.

$$H_{зп} = (15000 + 1500) \cdot 0,22 = 3630 \text{ (грн)}.$$

Амортизацію обладнання, комп'ютерів та приміщень, які використовувались під час виконання даного етапу роботи не розраховуємо, оскільки використовувався власний комп'ютер розробника, а приміщення було надано безкоштовно.

Витрати на силову електроенергію  $V_e$  розраховуються за формулою:

$$V_e = V \cdot П \cdot \Phi \cdot K_{п} \text{ [грн]}, \quad (4.4)$$

де  $V$  – вартість 1 кВт-год. електроенергії, 2 грн/кВт;  
 $П$  – установлена потужність обладнання, кВт;  
 $\Phi$  – фактична кількість годин роботи обладнання, годин.

На проведення дослідження та розробку програмного забезпечення витрачено загалом 3 місяці, то з урахуванням кількості робочих днів та робочого часу - 8 годин, маємо  $8 \cdot 22 \cdot 3 = 528$  годин роботи комп'ютера.

$K_{п}$  – коефіцієнт використання потужності;  $K_{п} < 1$ ; потужність комп'ютера становить 0,3.

$$V_e = 2 \cdot 0,35 \cdot 528 \cdot 0,3 = 110,88 \text{ (грн)}.$$

Під час розробки програмного продукту використовували безкоштовні програмні продукти.

Послуги інтернет 300 грн.

Витрати на канцтовари становлять 200 грн.

Інші витрати  $V_{ін}$  можна приймаємо як 100% від суми основної заробітної плати розробника тобто:

$$V_{ін} = 100\% \cdot Z_o [грн]. \quad (4.5)$$

$$V_{ін} = 1 \cdot 15000 = 15000 \text{ (грн)}.$$

Сума всіх попередніх статей витрат дає витрати на виконання даної частини (розділу, етапу) роботи –  $V$ :

$$V = 15000 + 1500 + 3630 + 110,88 + 200 + 300 + 15000 = 35740,88 \text{ (грн)}.$$

Проведемо прогнозування загальних витрат  $ZB$  на виконання та впровадження результатів виконаної наукової роботи за формулою:

$$ZB = \frac{V_{заг}}{\beta} [грн], \quad (4.6)$$

де  $\beta$  – коефіцієнт, який характеризує етап (стадію) виконання даної роботи. Так, як розробка знаходиться на стадії дослідного зразка, то  $\beta \approx 0,7$ .

$V_{заг}$  - загальна вартість всієї наукової роботи.

Оскільки наукова робота завершена,  $V_{заг} = 35740,88$  (грн).

$$ZB = \frac{35740,88}{0,7} = 51058,40 \text{ (грн)}.$$

Отже, прогноз загальних витрат на виконання та впровадження результатів становить 51058,40 грн.

### 4.3 Прогнозування комерційних ефектів від реалізації результатів розробки.

Збільшення чистого прибутку підприємства  $\Delta\Pi_i$  для кожного із років, протягом яких очікується отримання позитивних результатів від впровадження розробки, розраховується за формулою:

$$\Delta\Pi_i = \sum_i^n (\Delta\Pi_o \cdot N + C_o \cdot \Delta N) \cdot \lambda \cdot \rho \cdot \left(1 - \frac{v}{100}\right) [\text{грн}], \quad (4.7)$$

де  $\Delta\Pi_o$  – покращення основного оціночного показника від впровадження результатів розробки у даному році. Зазвичай таким показником може бути ціна одиниці нової розробки;

$N$  – основний кількісний показник, який визначає діяльність підприємства у даному році до впровадження результатів наукової розробки;

$\Delta N$  – покращення основного кількісного показника діяльності підприємства від впровадження результатів розробки;

$C_o$  – основний оціночний показник, який визначає діяльність підприємства у даному році після впровадження результатів наукової розробки;

$n$  – кількість років, протягом яких очікується отримання позитивних результатів від впровадження розробки;

$\lambda$  – коефіцієнт, який враховує сплату податку на додану вартість.

$\rho$  – коефіцієнт, який враховує рентабельність продукту.

Рекомендується приймати  $\rho = 0,25$ ;

$v$  – ставка податку на прибуток (18%).

Реалізація продукції до впровадження результатів наукової розробки не здійснювалась, тому приймаємо 1. Можливі обсяги продажу нового

продукту 500 одиниць в перший рік. Ціна, яку встановимо за 1 ліцензію – 500 грн.

В результаті впровадження результатів наукової розробки покращується якість продукту, що дозволяє підвищити ціну його реалізації на 500 грн.

Кількість одиниць реалізованої продукції також збільшиться: протягом першого року – на 500 шт., протягом другого року – ще на 200 шт., протягом третього року – ще на 300 шт.

Спрогнозуємо збільшення чистого прибутку від впровадження результатів наукової розробки у кожному році відносно базового.

Збільшення чистого прибутку підприємства  $\Delta \Pi_1$  протягом першого року складе:

$$\Delta \Pi_1 = [500 \cdot 1 + 1000 \cdot 500] \cdot 0,88 \cdot 0,25 \cdot \left(1 - \frac{18}{100}\right) = 90090,00 \text{ (грн)}.$$

Збільшення чистого прибутку підприємства  $\Delta \Pi_2$  протягом другого року (відносно базового року, тобто року до впровадження результатів наукової розробки) складе:

$$\Delta \Pi_2 = [500 \cdot 1 + 1000 \cdot 700] \cdot 0,88 \cdot 0,25 \cdot \left(1 - \frac{18}{100}\right) = 126090,00 \text{ (грн)}.$$

Збільшення чистого прибутку підприємства  $\Delta \Pi_3$  протягом третього року (відносно базового року, тобто року до впровадження результатів наукової розробки) складе:

$$\Delta \Pi_3 = [500 \cdot 1 + 1000 \cdot 1000] \cdot 0,88 \cdot 0,25 \cdot \left(1 - \frac{18}{100}\right) = 180090,00 \text{ (грн)}.$$



#### 4.4 Розрахунок ефективності вкладених інвестицій та періоду їх окупності.

Основними показниками, які визначають доцільність фінансування наукової розробки певним інвестором, є абсолютна і відносна ефективність вкладених інвестицій та термін їх окупності.

Розрахунок ефективності вкладених інвестицій передбачає:

*1-й крок.* Розрахунок теперішньої вартості інвестицій  $PV$ , що вкладаються в наукову розробку. Такою вартістю ми можемо вважати прогнозовану величину загальних витрат  $ЗВ$  на виконання та впровадження результатів НДДКР,  $ЗВ = PV = 51058,40$  (грн).

*2-й крок.* Розраховуємо очікуване збільшення прибутку  $\Delta\Pi_i$ , що його отримає підприємство (організація) від впровадження результатів наукової розробки, для кожного із років, починаючи з першого року впровадження.

*3-й крок.* Будуємо вісь часу, на якій відображаємо всі платежі (інвестиції та прибутки), що мають місце під час виконання науково-дослідної роботи та впровадження її результатів.

Рисунок 4.1 характеризує рух платежів (інвестицій та додаткових прибутків).

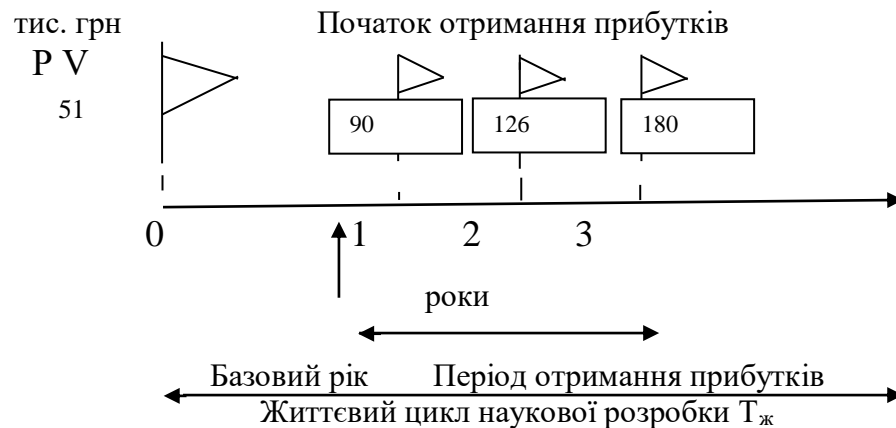


Рисунок 4.1 – Вісь часу з фіксацією платежів, що мають місце під час розробки та впровадження результатів НДДКР

4-й крок. Розраховуємо абсолютну ефективність вкладених інвестицій  $E_{абс}$  за формулою:

$$E_{абс} = (ПП - PV) \text{ [грн]}, \quad (4.8)$$

де ПП – приведена вартість всіх чистих прибутків, що їх отримає підприємство (організація) від реалізації результатів наукової розробки, грн;

PV – теперішня вартість інвестицій  $PV = ЗВ$ , грн.

Приведена вартість всіх чистих прибутків ПП розраховується за формулою:

$$ПП = \sum_1^t \frac{\Delta\Pi_i}{(1 + \tau)^t}, \quad (4.9)$$

де  $\Delta\Pi_i$  – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої НДДКР, грн;

t – період часу, протягом якого виявляються результати впровадженої НДДКР, роки;

$\tau$  – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні - 0,1;

t – період часу (в роках) від моменту отримання чистого прибутку до точки „0”;

$$ПП = \frac{90090,00}{(1 + 0,1)^1} + \frac{126090,00}{(1 + 0,1)^2} + \frac{180090,00}{(1 + 0,1)^3} = 323579,88 \text{ (грн)}.$$

$$E_{абс} = 323579,88 - 51058,40 = 272521,48 \text{ (грн)}.$$

Оскільки  $E_{abc} > 0$ , результат від проведення наукових досліджень щодо розробки програмного продукту та їх впровадження принесе прибуток, тобто є доцільним, але це ще не свідчить про те, що інвестор буде зацікавлений у фінансуванні даної програми.

5-й крок. Розраховують відносну (щорічну) ефективність вкладених в наукову розробку інвестицій  $E_B$  за формулою:

$$E_B = \sqrt[T_{ж}]{1 + \frac{E_{abc}}{PV}} - 1, \quad (4.10)$$

де  $E_{abc}$  – абсолютна ефективність вкладених інвестицій, грн;

$PV$  – теперішня вартість інвестицій  $PV = 3B$ , грн;

$T_{ж}$  – життєвий цикл наукової розробки, роки.

$$E_B = \sqrt[3]{1 + \frac{272521,48}{51058,40}} - 1 = \sqrt[3]{6,33} - 1 = 0,85 \text{ або } 85\%$$

Порівняємо  $E_B$  з мінімальною (бар'єрною) ставкою дисконтування  $\tau_{\text{мін}}$ , яка визначає ту мінімальну дохідність, нижче за яку інвестиції вкладатися не будуть.

Спрогнозуємо величину  $\tau_{\text{мін}}$ . У загальному вигляді мінімальна (бар'єрна) ставка дисконтування  $\tau_{\text{мін}}$  визначається за формулою:

$$\tau = d + f, \quad (4.11)$$

де  $d$  – середньозважена ставка за депозитними операціями в комерційних банках;  $d = 0,2$ ;

$f$  – показник, що характеризує ризикованість вкладень; величина  $f = 0,05$ .

$$\tau = 0,2 + 0,05 = 0,25$$

Оскільки  $E_B = 85\% > \tau_{\text{мін}} = 25\%$ , то у інвестора є потенційна зацікавленість у фінансуванні даної наукової розробки.

*б-й крок.* Розраховують термін окупності вкладених у реалізацію наукового проекту інвестицій  $T_{\text{ок}}$  за формулою:

$$T_{\text{ок}} = \frac{1}{E_e} \text{ грн} \quad (4.12)$$

$$T_{\text{ок}} = \frac{1}{1,58} = 1,17 \text{ (року)}.$$

Оскільки термін окупності вкладених у реалізацію наукового проекту інвестицій менше трьох років ( $T_{\text{ок}} < 3$  років), то фінансування нової розробки є доцільним.

#### **4.5 Висновок.**

В даному розділі було здійснено оцінювання комерційного потенціалу розробки. Проведено технологічний аудит з залученням трьох експертів. Аналіз експертних даних показав, що рівень комерційного потенціалу розробки вище середнього. Дослідження комерційного потенціалу розробки показав, що програмний продукт за своїми характеристиками випереджає аналогічні програмні продукти і є перспективною розробкою. Він має кращі функціональні показники, а тому є конкурентоспроможним товаром на ринку. Існуючі переваги нової розробки дозволять швидко її поширити та популяризувати.

Згідно із розрахунками всіх статей витрат на виконання науково-дослідної, дослідно-конструкторської та конструкторсько-технологічної

роботи загальна вартість витрат на розробку і впровадження складає 51058,40 грн.

Розрахована абсолютна ефективність вкладених інвестицій в сумі 272521,48 свідчить про отримання прибутку інвестором від впровадження програмного продукту у діяльність підприємства.

Щорічна ефективність вкладених в наукову розробку інвестицій складає 85 %, що вище за мінімальну бар'єрну ставку дисконтування, яка складає 25 %. Це означає потенційну зацікавленість інвесторів у фінансуванні розробки.

Термін окупності складає 1,17 року, що також свідчить про доцільність її фінансування.

## ВИСНОВКИ

У магістерській кваліфікаційній роботі розроблено технологію автоматизації об'єктів і процесів за допомогою Wi-Fi модулів.

Під час виконання даної роботи були вирішені всі поставлені завдання.

Результатом даної роботи є нові схемотехнічні рішення та внутрішнє програмне забезпечення модуля ESP8266 написаного на скриптовій мовою LUA .

ПО було протестовано з використанням локального брокера розгорнутого на локальному ПК (MQTT- брокер Mosquitto), MQTT – клієнта у вигляді стандартної програми під Android (MyMqtt) встановленої на смартфон, самого модуля ESP8266, датчика температури і вологості DHT11, контактного реле управління навантаженням.

Все ж для реалізації більш складних і багатозадачних рішень необхідно використовувати варіанти розробки програмного забезпечення на мові C ++ з використанням бібліотек SDK, або операційної системи з відкритим вихідним кодом FreeRTOS.

Технологія забезпечує керування об'єктами та процесами через мережу Інтернет, та базується на новітньому підходів і може бути застосована широким колом користувачів. Практичні дослідження підтверджують її актуальність та працездатність.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Wi-Fi [Електронний ресурс]: стаття. Режим доступу: вільний. URL: <https://en.wikipedia.org/wiki/Wi-Fi>
2. Wi-Fi [Електронний ресурс]: стаття. Режим доступу: вільний. URL: <http://ccm.net/faq/9081-Wi-Fi-description-and-explanations>
3. Гарі Корнелл, Кей С. Хорстманн. Core Java Volume II: Advanced Features, 2004. 1088 стор.
4. Джон Хант. Key Java, 2013. 331 стр.
5. Geektimes [Електронний ресурс]: Робота з ESP8266: Збираємо компілятор і пишемо першу прошивку, [2014] - Режим доступу: <http://geektimes.ru/post/241842/>
6. ESP8266 Спільнота розробників [Електронний ресурс]: ESP8266 - підключення і оновлення прошивки, [2015] - Режим доступу: <http://esp8266.ru/esp8266-podkluchenie-obnovlenie-proshivki/>
7. ESP8266 Спільнота розробників [Електронний ресурс]: ESPlorer, [2015] - Режим доступу: <http://esp8266.ru/esplorer/>
8. ESP8266 Спільнота розробників [Електронний ресурс]: Завантажити файли документації, datasheet, прошивки, утиліти, [2015] - Режим доступу: <http://esp8266.ru/downloads/>
9. ESP8266 Спільнота розробників [Електронний ресурс]: Терморегулятори ESP8266, різні модифікації модулів на базі ESP8266, [2015] - Режим доступу: <http://esp8266.ru/modules-esp8266/>
10. Журнал «Компоненти і технології» [Електронний ресурс]: Конфігуровані процесори Xtensa фірми Tensilica для вбудованих систем, [2019] - Режим доступу: [http://www.kite.ru/articles/elcomp/2009\\_10\\_66.php](http://www.kite.ru/articles/elcomp/2009_10_66.php)
11. Новини Microsoft [Електронний ресурс]: Розумний офіс Microsoft, [2015] - Режим доступу: <https://news.microsoft.com/ru-ru/features/smart-office-microsoft-russia/>

12. Intouch-страхування [Електронний ресурс]: Розумне страхування, [2015] - Режим доступу: <http://www.in-touch.ru/land/cascosmart/>
- 13.10. Proteus Digital Health [Електронний ресурс]: Система, яка складається з датчиків і програмного забезпечення Proteus Discover, [2016] – Режим доступу: <http://www.proteus.com/>
14. Місто високих технологій [Електронний ресурс]: Місто високих технологій, [2015] - Режим доступу: <http://innopolis.ru/>
15. IOS9 - HomeKit -Apple [Електронний ресурс]: HomeKit Надійний контроль над будинком. У ваших дбайливих руках, [2016] - Режим доступу: Ready for Sky [Електронний ресурс]: Чим ми займаємося, [2016] - Режим доступу: [http://readyforsky.com/what\\_we\\_do/](http://readyforsky.com/what_we_do/)
16. Інтернет речей. [Електронний ресурс]: Протокол MQTT, [2015] - Режим доступу: <http://iot.ru/protokol-mqtt/>
17. Довідник з мови Lua [Електронний ресурс]: Довідник з мови Lua - Режим доступу: <http://www.lua.ru/doc/1.html>
18. Блоги по електроніці [Електронний ресурс]: Датчики температури і вологості DHT11 і DHT22, [2011] - Режим доступу: <http://electronics-lab.ru/blog/mcu/46.html>
19. Homes Smart Домашня автоматизація [Електронний ресурс]: Наддешевий WI-FI модуль ESP8266, [2015] - Режим доступу: <http://homes-smart.ru/index.php/oborudovanie/bez-provodov-wi-fi/sverkhdeshevyj-wi-fi-modul-esp8266>
20. Хабрахабр [Електронний ресурс]: Що в нутрії «народного WIFI» ?, [2015] - Режим доступу: <https://habrahabr.ru/company/coolrf/blog/238443/>
21. Вікіпедія [Електронний ресурс]: ESP 8266, [2016] - Режим доступу: <https://ru.wikipedia.org/wiki/ESP8266>
22. Методичні вказівки до виконання студентами-магістрантами економічної частини магістерських кваліфікаційних робіт / Уклад. В. О. Козловський – Вінниця: ВНТУ, 2012. – 22 с.



## ДОДАТКИ

## ДОДАТОК А

Міністерство освіти та науки України  
Вінницький національний технічний університет  
Інститут інформаційних технологій та комп'ютерної інженерії

ЗАТВЕРДЖУЮ

Завідувач кафедри ОТ

д.т.н., професор Мартинюк Т. Б. \_\_\_\_\_

(наук. ст., вч. зв., ініц. та прізви.)

(підпис)

“ \_\_\_\_ ” \_\_\_\_\_ 20\_\_р.

## ТЕХНІЧНЕ ЗАВДАННЯ

на виконання магістерської кваліфікаційної роботи

Технологія автоматизації об'єктів та процесів за допомогоюмікропроцесорних Wi-Fi модулів08–23.МКР.013.00.000.ТЗ

Науковий керівник: к.т.н., доц. Кадук О.В.

\_\_\_\_\_  
(підпис)студент групи КІ-18м\_\_\_\_\_  
Сташевський С.Б.

(підпис)

Вінниця 2019р.

## 1. Підстава для виконання магістерської кваліфікаційної роботи (МКР)

- а) актуальність досліджень;
- б) наказ про затвердження теми дипломної роботи.

## 2. Мета і призначення МКР

а) мета – полягає автоматизації об'єктів і процесів за допомогою Wi-Fi модулів сімейства ESP8266;

б) призначення розробки – виконання магістерської кваліфікаційної роботи, виконання організаційно – технологічних та наукових досліджень.

## 3. Вихідні дані для виконання МКР

- технічні характеристики мікроконтролерів сімейства ESP8266, їх різновидів та прошивок програмного забезпечення;
- опис програмних засобів розробки, мова LUA, протоколів і стандартів зв'язку, протокол MQTT.

## 4. Вимоги до виконання МКР

- здійснити огляд існуючих технологій автоматизації об'єктів і процесів;
- здійснити класифікацію методів реалізації Інтернету речей;
- дослідити способи та методи побудови апаратно-програмних засобів;
- здійснити проектування структурно-функціональної схеми та вибір електронних компонентів;
- здійснити розробку та відлагодження програмного забезпечення.

## 5. Етапи МКР та очікувані результати

№ етапу	Назва етапу	Термін виконання		Очікувані результати
		початок	кінець	
1	Пошук та огляд інформаційних джерел	02.09.19р.	08.09.19р.	Розділ 1
2	Дослідження способів автоматизації об'єктів і процесів за допомогою Wi-Fi модулів	09.09.19р.	22.09.19р.	Розділ 2
3	Реалізація технологія автоматизації об'єктів та процесів за допомогою мікропроцесорних Wi-Fi модулів	23.09.19р.	22.10.19р.	Розділ 3
4	Економічна частина	23.10.19р.	12.11.19р.	Розділ 4

#### 6. Матеріали, що подаються до захисту МКР

Пояснювальна записка МКР, графічні і ілюстративні матеріали, протокол попереднього захисту МКР на кафедрі, відзив наукового керівника, відзив опонента, протоколи складання державних екзаменів, анотації до МКР українською та іноземною мовами, нормоконтроль про відповідність оформлення МКР діючим вимогам.

#### 7. Порядок контролю виконання та захисту МКР

Виконання етапів графічної та розрахункової документації МКР контролюється науковим керівником згідно зі встановленими термінами. Захист МКР відбувається на засіданні Державної екзаменаційної комісії, затвердженою наказом ректора.

#### 8. Вимоги до оформлення МКР

Вимоги викладені в МЕТОДИЧНИХ ВКАЗІВКАХ до дипломного проектування, ДСТУ\_ 3008-95, ДСТУ 3974-2000 «Правила виконання дослідно-конструкторських робіт. Загальні положення» та діючого ГОСТ 2.114-95 ЕСКД.

#### 9. Вимоги щодо технічного захисту інформації в МКР з обмеженим доступом

Відсутні.

Додаток Б  
Модуль ініціалізації

```
- file: init.lua  
app = require ( "application")  
config = require ( "config")  
setup = require ( "setup")  
webserver = require ( "webserver")  
wifi_connect = require ( "wifi_connect")  
io_module = require ( "io_module")  
dht11 = require ( "dht11")  
  
setup.start ()  
  
print ( "Hi! NODEMCU START!")
```

Додаток В  
Модуль конфігурації контролера

```
-- file: config.lua
local module = {}
-- назва та пароль Wi-Fi мережі
module.SSID = {}
  module.SSID [ "ACORP" ] = "UXGA4R6P"

-- ір-адреса і порт MQTT-брокера
module.HOST = "192.168.1.3"
module.PORT = 1884

-- Гілки MQTT куди посилаємо і з яких зчитуємо інформацію
module.ID = node.chipid ()
module.MQTT_PUBLISH = "message /"
module.ENDPOINT = "nodemcu /"

- Піни контролера NODEMCU
module.IO0 = 0
module.IO1 = 1
module.IO2 = 2
module.IO3 = 3
module.IO4 = 4
module.IO5 = 5

- зверху реле, внизу датчики dht11
module.IO6 = 6          - GPIO12
module.IO7 = 7          - GPIO13
module.IO8 = 8          - GPIO15
return module
```

## Додаток Г

Модуль, що виконує основну логіку програми

```
-- file: application.lua
local module = {}
m = nil

-- Відсилаємо інформацію брокеру кожні 2 сек.
-- функція ping'a
local function send_ping ()
    -- Читаємо стан пінів контролера
    local state_byte = 0;
    local state_str = "" .. gpio.read (config.IO0);

    state_str = state_str .. "|" .. gpio.read (config.IO1);
    state_str = state_str .. "|" .. gpio.read (config.IO2);
    state_str = state_str .. "|" .. gpio.read (config.IO3);
    state_str = state_str .. "|" .. gpio.read (config.IO4);
    state_str = state_str .. "|" .. gpio.read (config.IO5);

    -- читаємо температуру

    dht11.read (config.IO6)
    state_str = state_str .. "|" .. dht11.getTemperature ()
    state_str = state_str .. "," .. dht11.getHumidity ()

    dht11.read (config.IO7)
    state_str = state_str .. "|" .. dht11.getTemperature ()
    state_str = state_str .. "," .. dht11.getHumidity ()
```

```

    m: publish (config.ENDPOINT .. "ping", "{\ id \ ":" .. config.ID .. ", \ state \
": \ " .. state_str .. \ }" , 0,0)

```

```

collectgarbage ();

```

```

end

```

```

-- посилаємо брокеру MQTT свій ID для реєстрації

```

```

local function register_myself ()

```

```

    m: subscribe (config.ENDPOINT .. config.MQTT_PUBLISH ..
config.ID, 0, function (conn)

```

```

--    m: subscribe (config.ENDPOINT .. config.ID, 0, function
    (conn) print ( "Successfully subscribed to data endpoint")
    end)

```

```

end

```

```

local function mqtt_start ()

```

```

    m = mqtt.Client (config.ID, 120)

```

```

-- обробчкі одержання повідомлення в даній MQTT-гілці

```

```

m: on ( "message", function (conn, topic, data)

```

```

    if data ~= nil then

```

```

        print (topic .. ":" .. data)
    end

```

```

-- Прийняли повідомлення в свою MQTT-гілку. Обробляємо його.

```

```

io_module.set_io (data)

```

```

    end

```

```

end)

```

```

--з'єднуємося з брокером

```

```

m: connect (config.HOST, config.PORT, 0, 1, function (con)

```

```

    register_myself ()

```



```
-- Надсилаємо інформацію про свою присутність кожні 4 секунди
tmr.stop (6)
tmr.alarm (6, 2000., 1, send_ping)
end)

end

-- функція початку роботи модуля
function module.start ()
    mqtt_start ()
end

return module
```

## Додаток Г

Модуль, що здійснює підключення до мережі WiFi

```

-- file:
wifi_connect.lua
-- local module = {}

local function wifi_wait_ip ()
if wifi.sta.getip () == nil then
    print ( "IP unavailable, Waiting ...")
else
    tmr.stop (1)
    print ( "\ n =====")
    print ( "ESP8266 mode is:" .. wifi.getmode ())
    print ( "MAC address is:" .. wifi.ap.getmac ())
    print ( "IP is" ..wifi.sta.getip ())
    print ( "=====")

--          - З'являється Wi-Fi коли отримали IP-адреса. Запускаємо наш
MQTT-додаток

app.start ()
    -- webservice.start()
    -- end
end

local function wifi_start (list_aps)
    if list_aps then
        for key, value in pairs (list_aps) do
            if config.SSID and config.SSID [key] then

```

```
wifi.setmode (wifi.STATION);
wifi.sta.config (key, config.SSID [key])
wifi.sta.connect ()
print ( "Connecting to" .. key .. "...")
--config.SSID = nil - can save memory
-- Законектились до Wi-Fi. чекаємо IP-адреса
tmr.alarm (1, 2500, 1, wifi_wait_ip)
end
end
else
print ( "Error getting AP list")
end
end

function module.start ()

-- Намагаємося законектитися до wi-
fi print ( "Trying connect to WiFi")
wifi.setmode (wifi.STATION);
wifi.sta.getap (wifi_start)
end

return module
```

## Додаток Д

Модуль, що визначає налаштування виводів

```
-- file: setup.lua
local module = {}

function module.start ()
    print ( "Config PINs ..." )

    -- налаштовуємо ПІНи
    gpio.mode (config.IO0, gpio.OUTPUT)
    gpio.mode (config.IO1, gpio.OUTPUT)
    gpio.mode (config.IO2, gpio.OUTPUT)
    gpio.mode (config.IO3, gpio.OUTPUT)
    gpio.mode (config.IO4, gpio.OUTPUT)
    gpio.mode (config.IO5, gpio.OUTPUT)
    gpio.mode (config.IO6, gpio.OUTPUT)
    gpio.mode (config.IO7, gpio.OUTPUT)
    gpio.mode (config.IO8, gpio.OUTPUT)

    gpio.write (config.IO0, gpio.HIGH)
    gpio.write (config.IO1, gpio.HIGH)
    gpio.write (config.IO2, gpio.HIGH)
    gpio.write (config.IO3, gpio.HIGH)
    gpio.write (config.IO4, gpio.HIGH)
    gpio.write (config.IO5, gpio.HIGH)
    gpio.write (config.IO6, gpio.HIGH)
    gpio.write (config.IO7, gpio.HIGH)
    gpio.write (config.IO8, gpio.HIGH)
```

```
-- конфігуруємо wifi
  -- print ( "Configuring WiFi ...")
  -- as WiFi access point
  -- wifi_accesspoint.start ()
  --
  -- connect to wifi network
  wifi_connect.start ()

end

return module
```

Додаток Е  
Модуль створення Web-сервера

```

-- file: webservice.lua
local module = {}
-- srv = nil
function module.start () print (
"Start webservice") srv =
net.createServer (net.TCP) srv:
listen (80, function (conn)
    conn: on ( "receive", function (client,
        request) local buf = "";
        local _, _, method, path, vars = string.find (request, "([AZ] +) (. +)? (. +)
HTTP");
        if (method == nil) then
            _, _, Method, path = string.find (request, "([AZ] +) (. +) HTTP");
        end
        local _GET = {} if
        (vars ~ = nil) then
            for k, v in string.gmatch (vars, "(% w +) = (% w +) & *")
                do _GET [k] = v
            end
        end
        buf = buf .. "<h1> ESP8266 Web Server </ h1>";
        buf = buf .. "<p> GPIO1 <a
href = \ "? pin = 1ON \ "> <button> ON </ button> </a> & nbsp; <a
href = \ "? pin = 1OFF \ "> <button> OFF </ button> </a>
</ p> "; buf = buf .. "<p> GPIO2 <a
href = \ "? pin = 2ON \ "> <button> ON </ button> </a> & nbsp; <a

```

```

href = \ "? pin = 2OFF \"> <button> OFF </ button> </a> </ p> ";
    buf = buf .. "<p> GPIO3 <a
href = \ "? pin = 3ON \"> <button> ON </ button> </a> & nbsp; <a
href = \ "? pin = 3OFF \"> <button> OFF </ button> </a> </ p> ";
    buf = buf .. "<p> GPIO4 <a
href = \ "? pin = 4ON \"> <button> ON </ button> </a> & nbsp; <a
href = \ "? pin = 4OFF \"> <button> OFF </ button> </a> </ p> ";
    buf = buf .. "<p> GPIO5 <a
href = \ "? pin = 5ON \"> <button> ON </ button> </a> & nbsp; <a
href = \ "? pin = 5OFF \"> <button> OFF </ button> </a> </ p> ";
    buf = buf .. "<p> GPIO6 <a
href = \ "? pin = 6ON \"> <button> ON </ button> </a> & nbsp; <a
href = \ "? pin = 6OFF \"> <button> OFF </ button> </a> </ p> ";
    buf = buf .. "<p> GPIO7 <a
href = \ "? pin = 7ON \"> <button> ON </ button> </a> & nbsp; <a
href = \ "? pin = 7OFF \"> <button> OFF </ button> </a> </ p> ";
    buf = buf .. "<p> GPIO8 <a
href = \ "? pin = 8ON \"> <button> ON </ button> </a> & nbsp; <a
href = \ "? pin = 8OFF \"> <button> OFF </ button> </a> </ p> ";
    local _on, _off = "", ""
        local set_pin = _GET.pin;
        io_module.set_io (set_pin)
    client: send (buf);
    client: close ();
    collectgarbage ();
end)
end)
end
return module

```

Додаток Є  
Модуль управління входами

```
-- file: io.lua local
module = {}

function module.set_io (set_pin)

if (set_pin == "0ON") then
    gpio.write (config.IO0, gpio.HIGH);
elseif (set_pin == "0OFF") then
    gpio.write (config.IO0, gpio.LOW);

    elseif (set_pin == "1ON") then
    gpio.write (config.IO1, gpio.HIGH);
elseif (set_pin == "1OFF") then
    gpio.write (config.IO1, gpio.LOW);

        elseif (set_pin == "2ON") then
    gpio.write (config.IO2, gpio.HIGH);
elseif (set_pin == "2OFF") then
    gpio.write (config.IO2, gpio.LOW);

            elseif (set_pin == "3ON") then
    gpio.write (config.IO3, gpio.HIGH);
elseif (set_pin == "3OFF") then
    gpio.write (config.IO3, gpio.LOW);

                elseif (set_pin == "4ON") then
    gpio.write (config.IO4, gpio.HIGH);
```



```
elseif (set_pin == "4OFF") then
    gpio.write (config.IO4, gpio.LOW);

        elseif (set_pin == "5ON") then
            gpio.write (config.IO5, gpio.HIGH);
elseif (set_pin == "5OFF") then
    gpio.write (config.IO5, gpio.LOW);

        elseif (set_pin == "6ON") then
            gpio.write (config.IO6, gpio.HIGH);
elseif (set_pin == "6OFF") then
    gpio.write (config.IO6, gpio.LOW);

        elseif (set_pin == "7ON") then
            gpio.write (config.IO7, gpio.HIGH);
elseif (set_pin == "7OFF") then
    gpio.write (config.IO7, gpio.LOW);

        elseif (set_pin == "8ON") then
            gpio.write (config.IO8, gpio.HIGH);
elseif (set_pin == "8OFF") then
    gpio.write (config.IO8, gpio.LOW);

end

end

function module.start ()
end

return module
```

Додаток И  
Модуль роботи з датчиком dht11

```
local module = { }
```

```
local humidity
```

```
local temperature
```

```
local checksum
```

```
local checksumTest
```

```
local humiditTenth
```

```
local temperatureTenth
```

```
function module.read (pin)
```

```
    humidity = 0
```

```
    temperature = 0
```

```
    checksum = 0
```

```
    humiditTenth = 0
```

```
    temperatureTenth = 0
```

```
    gpio_read = gpio.read
```

```
    gpio_write = gpio.write
```

```
    bitStream = { }
```

```
    for j = 1, 40, 1 do
```

```
        bitStream [j] = 0
```

```
    end
```

```
    bitlength = 0
```

```
-- Крок 1: посилаємо початковий сигнал в DHT11
```

```

gpio.mode (pin, gpio.OUTPUT)
gpio.write (pin, gpio.LOW)
tmr.delay (20000)

```

```

gpio_read = gpio.read
gpio_write = gpio.write

```

```

gpio.mode (pin, gpio.INPUT)

```

```

-- Крок 2: DHT11 датчик посилає початковий відповідний сигнал

```

```

-- Чекаємо коли датчик підніме і опустить шину даних

```

```

c = 0

```

```

while (gpio_read (pin) == 0 and c <800) do c = c + 1 end

```

```

c = 0

```

```

while (gpio_read (pin) == 1 and c <100) do c = c + 1 end

```

```

-- bus will always let up eventually, do not bother with

```

```

timeout c = 0

```

```

while (gpio_read (pin) == 0 and c <800) do c = c + 1 end

```

```

c = 0

```

```

while (gpio_read (pin) == 1 and c <100) do c = c + 1 end

```

```

-- Крок 3: DHT11 посилає дані. 40 біт - тобто 5 байт. 2 байта
температури - цілу і дробову частину. 2 байта вологість. і байт
контрольної суми

```

```

for j = 1, 40, 1 do

```

```

    while (gpio_read (pin) == 1 and bitlength <10) do

```

```

        bitlength = bitlength + 1

```

```

    end

```

```

    bitStream [j] = bitlength

```

```

bitlength = 0
c = 0
while (gpio_read (pin) == 0 and c <800) do c = c + 1 end
end

--Обробка даних.
for i = 1, 8, 1 do
  if (bitStream [i + 0]> 2) then
    humidity = humidity + 2 ^ (8-i)
  end
  if (bitStream [i + 8]> 2) then
    humiditTenth = humiditTenth + 2 ^ (8-i)
  end
  if (bitStream [i + 16]> 2) then
    temperature = temperature + 2 ^ (8-i)
  end
  if (bitStream [i + 24]> 2) then
    temperatureTenth = temperatureTenth + 2 ^ (8-i)
  end
  if (bitStream [i + 32]> 2) then
    checksum = checksum + 2 ^ (8-i)
  end
end
end

checksumTest = (humidity + humiditTenth + temperature +
temperatureTenth)% 256
if checksum ~ = checksumTest then
  humidity = -1
  - temperature = -1
End

```

```
collectgarbage ();  
end  
  
-- Функція отримати температуру  
function module.getTemperature ()  
    return temperature  
end  
  
-- Функція отримати вологість  
function module.getHumidity ()  
    return humidity  
end  
  
return module
```