

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет імені Петра Могили
Факультет комп'ютерних наук
Кафедра комп'ютерної інженерії

ДОПУЩЕНО ДО ЗАХИСТУ

Завідувач кафедри,

д-р техн. наук, проф.

_____ І. М. Журавська

«__» _____ 2023 р.

КВАЛІФІКАЦІЙНА МАГІСТЕРСЬКА РОБОТА
Програмно-апаратний комплекс диференційованого годування
тварин на базі ESP8266 (IoT)

Спеціальність «Комп'ютерна інженерія»

123 – КМР.1 – 605.21710532

Студент

_____ Д. О. Чумаченко

підпис

«__» _____ 2023 р.

Керівник канд. техн. наук, доцент

_____ В. Ю. Савінов

підпис

«__» _____ 2023 р.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ	4
ВСТУП	5
1 АНАЛІТИЧНА ЧАСТИНА. ОГЛЯД ІСНУЮЧИХ РІШЕНЬ СИСТЕМИ, ЩО РОЗРОБЛЯЄТЬСЯ. ФОРМУВАННЯ ВИМОГ ДО АПАРАТНО-ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	7
1.1 Огляд існуючих рішень	9
1.2 Постановка задачі та формування вимог	17
1.3 Дослідження технологій для створення годівниці	18
Висновок до розділу 1	23
2 МАТЕМАТИЧНІ МЕТОДИ ТА ПРОЄКТУВАННЯ СИСТЕМИ, ЩО РОЗРОБЛЯЄТЬСЯ	24
2.1 Розрахунок добового використання корму та об'єму годівниці	24
2.2 Дослідження впливу автоматичної годівниці для домашніх тварин і частого годування їжею для зниження ваги в домогосподарствах з кількома котами	26
2.3 Результати проведеного дослідження	31
Висновок до розділу 2	35
3 АПАРАТНО-ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ	36
3.1 Вибір апаратної платформи	36
3.2 Детальний опис обраної апаратної платформи та її компонентів	43
3.3 Побудова макетної та принципової схеми	51
3.4 Вибір мови програмування та середовища розробки	56
Висновок до розділу 3	64
4 ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ. АНАЛІЗ РЕЗУЛЬТАТІВ ОБЧИСЛЕНЬ	65
4.1 Виконання налаштувань	65
4.2 Випробування роботи програмно-апаратного комплексу	70
Висновок до розділу 4	71
ВИСНОВКИ	72
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	73

Кафедра комп'ютерної інженерії	3
Програмно-апаратний комплекс диференційованого годування тварин на базі ESP8266	
ДОДАТОК А Код ПРОГРАМИ	76
ДОДАТОК Б МАТЕРІАЛИ АПРОБАЦІЇ	79

ПЕРЕЛІК СКОРОЧЕНЬ

АЦП	- аналого-цифровий перетворювач
ЛОМ	- локальна обчислювальна мережа
ПЗ	- програмне забезпечення
ІоТ	- Інтернет речей
ІТ	- Інтернет технології
і/о	- Інформаційні технології
LCD	- Вхід/вихід
MVC	- Рідкокристалічний дисплей
RAM	- Model-View-Controller
ROM	- Random Access Memory
SPI	- Serial Peripheral Interface;

ВСТУП

Актуальність теми: підвищення якості життя у суспільстві за останні десятиріччя дає людині змогу задовольняти не тільки свої природні потреби, а й подбати про моральний стан та те, що раніше було не доступно. Збільшення житлової площі та матеріального забезпечення приводить до того що люди можуть заводити домашніх улюбленців у більшій кількості, але у круговороті справ не завжди знаходиться час подбати про нового члена сім'ї. Якщо відсутність ласки та тренувань жива істота може винести, то відсутність годування може призвести до серйозних наслідків. Тож дана робота присвячена автоматизації процесу годування тварин за умов відсутності власника протягом тривалого часу.

Мета магістерської роботи: розробити апаратно-програмний комплекс автоматизації процесу годування тварин без прямого втручання власника у процес.

Об'єкт дослідження магістерської роботи: Годування домашніх улюбленців за допомогою модуля ESP 8266 та мікроконтролера Arduino.

Предмет дослідження магістерської роботи: повноцінна робота програмно-апаратного комплексу для автоматизації процесу в умовах цілодобової роботи у енергозберігаючому режимі .

Для досягнення поставленої мети необхідно вирішити такі **завдання:**

- підібрати засоби з бібліотек Arduino для досягнення поставлених цілей з автоматизації;
- розробити апаратний комплекс, який би забезпечував безперервну роботу пристрою в реальному часі;
- підібрати програмний комплекс для забезпечення правильного функціонування усієї системи цілком та максимальної автоматизації усіх процесів.

Наукова новизна: в Україні ще немає патентів на корисну модель енергозберігаючої годівниці.

Практичне значення отриманих результатів: розроблений програмно-апаратний комплекс може бути застосований, як більш дешевий, але не менш функціональний аналог дорогих аналогів представлених на ринку в даний момент в умовах зменшеного енергоспоживання відносно інших пристроїв, що використовуються для тих же цілей.

Наукова новизна полягає в реалізації програмно-апаратного комплексу для годування тварин, який відрізняється простотою програмної частини, мінімального використання модулів, що зменшить затрати на побудову комплексу.

Робота пройшла апробацію на Всеукраїнській науково-практичній конференції молодих вчених, аспірантів і студентів «Інформаційні технології та інженерія» (Миколаїв, 07–10 лютого 2023 р.) [4].

1 АНАЛІТИЧНА ЧАСТИНА. ОГЛЯД ІСНУЮЧИХ РІШЕНЬ СИСТЕМИ, ЩО РОЗРОБЛЯЄТЬСЯ. ФОРМУВАННЯ ВИМОГ ДО АПАРАТНО-ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Model-View-Controller (MVC) – схема поділу даних програми, призначеного для користувача інтерфейсу і керуючої логіки на три окремих компоненти: модель, уявлення і контролер – таким чином, що модифікація кожного компонента може здійснюватися незалежно.

Python – високорівнева мова програмування загального призначення з динамічної строгою типізацією і автоматичним управлінням пам'яттю, орієнтований на підвищення продуктивності розробника, читання коду і його якості, а також на забезпечення переносимості написаних на ньому програм.

Штучний інтелект – властивість інтелектуальних систем виконувати творчі функції, які традиційно вважаються прерогативою людини; наука і технологія створення інтелектуальних машин, особливо інтелектуальних комп'ютерних програм.

Lua – це легка, високорівнева багатопарадигмальна мова програмування, розроблена переважно для вбудованого використання в програмах. Lua є кросплатформною, оскільки інтерпретатор скомпільованого байт-коду написаний на ANSI C, а Lua має відносно простий C API для вбудовування його в програми.

Lua виникла в 1993 році як мова для розширення програмних застосунків для задоволення зростаючого попиту на налаштування в той час. Він надавав базові можливості більшості процедурних мов програмування, але не було включено більш складних або предметно-спеціальних функцій; скоріше, він включав механізми для розширення мови, що дозволило програмістам реалізувати такі функції. Оскільки Lua мала бути загальною мовою розширення, що вбудовується, розробники Lua зосередилися на

покращенні її швидкості, переносимості, розширюваності та простоти використання під час розробки.

C/C ++ – компільований, статично типізований мова програмування загального призначення. Підтримує такі парадигми програмування, як процедурне програмування, об'єктно-орієнтоване програмування, узагальнене програмування.

Локальна обчислювальна мережа (ЛОМ) – комп'ютерна мережа, що покриває зазвичай відносно невелику територію або невелику групу будівель. Термін LAN також використовується в кіберспорті і означає, що гравці знаходяться в одному приміщенні і пов'язані локальною мережею, а не грають з дому через інтернет.

Мікроконтролер – мікросхема, призначена для управління електронними пристроями. Типовий мікроконтролер поєднує на одному кристалі функції процесора і периферійних пристроїв, містить ОЗУ і ПЗУ. По суті, це однокристальний комп'ютер, здатний виконувати досить прості завдання.

STM32 – сімейство 32-бітних мікроконтролерів виробництва STMicroelectronics. Чіпи STM32 групуються в серії, в рамках кожної з яких використовується один і той же 32-бітне ядро ARM, наприклад, Cortex-M7F, Cortex-M4F, Cortex-M3, Cortex-M0 + або Cortex-M0.

Internet of Things (IoT) – це концепція, за якою об'єкти з навколишнього середовища, які зазвичай не пов'язані з інформаційними технологіями, можуть бути підключені до Інтернету та взаємодіяти один з одним за допомогою датчиків, даних та програмного забезпечення.

У простішому варіанті, IoT описує підключення різноманітних пристроїв до Інтернету, що дозволяє збирати, обробляти та аналізувати дані з цих пристроїв. Це включає в себе взаємодію з пристроями, які мають вбудовані сенсори, які можуть вимірювати різноманітні параметри, такі як

температура, вологість, рівень освітленості, рівень шуму тощо. Інформація, зібрана з цих пристроїв, може бути використана для вирішення різноманітних проблем, які виникають у різних сферах життя, включаючи енергетику, транспорт, медицину, промисловість та бізнес.

Застосування IoT можуть включати в себе: віддалене керування пристроями та системами, моніторинг стану обладнання та інфраструктури, автоматизацію виробничих процесів, створення інтелектуальних будинків та міст, покращення управління транспортними системами та багато іншого.

В період з 05.12.2022 – 17.12.2022 було поставлено ціль – розглянути сучасні методи та технології розробки автоматизованих годівниць для задоволення потреб домашніх улюбленців.

1.1 Огляд існуючих рішень

Почавши інтернет-серфінг в пошуках аналогів автоматичних годівниць, які можуть запропонувати необхідний функціонал із збереженням пропорції ціна-якість, було знайдено велику кількість пристроїв з низькими цінами, але якість та відгуки про них були нижче середнього (рис. 1.1).



Рисунок 1.1 – Приклад представнику найдешевшого сегменту ринку

Такий варіант має лише дуже умовну автоматизацію так, як тварина може спорожнити контейнер зберігання навіть за один раз (виходячі з розміру деяких представників тваринного світу та відсутність розуміння ними концепції зберігання в деяких умовах). Що робить такі варіанти і задовільними в плані ціни, але зовсім нефункціональними.

Також на ринку було знайдено так званий “преміум” сегмент пристроїв, який може задовільнити високою якістю зборки, розширеним функціоналом та надійним брендом (рис. 1.2).



Рисунок 1.2 – Годівниця Xiaomi Petoneer Visual Feeder.

Цей пристрій обладнаний модулем Wi-Fi, камерою та динаміком для додаткової взаємодії зі своїм улюбленцем в умові відсутності власника безпосередньо в місці його проживання. Автоматична годівниця управляється через програму та хмарний сервіс Petoneer. Налаштуйте розмір порцій та встановіть відповідний графік годування. Програма надсилає push-

повідомлення про кожного, а також попереджає про те, що корм у резервуарі закінчується.

Але покупка такого пристрою стане великим ударом по кишені бо він обійдеться у $\approx 5\ 629$ €, тому було прийняту рішення продовжити пошук у меншому ціновому діапазоні.

Надалі було розглянуто та розсортовано ще декілька прямих аналогів у порядку зменшення ціни.

Petkit Smart Fresh Element 3 (рис. 1.3) – завдяки вбудованому Wi-Fi й застосунку Petkit, годування перетворюється процес, який не вимагає від власника особистої присутності. Налаштувавши необхідні параметри в застосунку, розумна система подавання корму відміряє точну кількість. Крім того, програма показує, скільки їжі видається і скільки грамів вихованець з'їдає щодня або в інший визначений проміжок часу. А ще ця автоматична годівниця сама перевіряє рівень корму та за потреби надсилає повідомлення на смартфон. Ще одна цікава функція – можливість записати голосове повідомлення, яке слугуватиме закликком до годування. Щоразу, коли пристрій видаватиме порцію їжі, він вмикатиме запис, щоб улюбленець знав, що пора поїсти. Такий функціонал обійдеться у 5 499 €, що виводить його у дорогий сегмент ринку.



Рисунок 1.3 - Petkit Smart Fresh Element 3

CHINAPST FD-BL3 (рис. 1.4) – більш дешевий пристрій, аналог розглянутого вище пристрою, його функціонал також включає в себе автоматичне керування, а також користувач може використовувати годівницю як охоронний пристрій. Коли він у від'їзді та домашній улюбленець з ним, а в будинку нікого немає, то в застосунку можна активувати режим "детектор руху" і користувач отримує повідомлення і фото на телефон про тривожні рухи у вашому будинку. Разом з цим у застосунку зберігаються всі записи та фото

нестандартних ситуацій у домі , що вплинуло на його ціну $\approx 4100\text{€}$.



Рисунок 1.4 – Годівниця CHINAPST FD-BL3

PETRUST PP005 (рис. 1.5) – пристрій, що як і попередні аналоги має свій застосунок та можливість записати голосове повідомлення до 10 секунд. Зовнішній вигляд контейнеру представлений з прозорого пластику, що дає змогу наглядно бачити кількість залишку корму, щоб не зіткнутись з повідомленням про його закінчення у самий незручний для цього момент. Ціна такого пристрою вар'юється у межах 3200 ₴.



Рисунок 1.5 – Годівниця PETRUST PP005

Petrust PP002 (рис. 1.6) – пристрій, що як і попередні розглянуті аналоги має можливість годування за розкладом, власник також може записувати голосові повідомлення тривалістю до 10 секунд. В годівницю вбудовано додаткові батареї, які забезпечать його роботу при відключенні світла. Також присутній контроль залишку корму, ці дані відображаються на дисплеї. Ціна як і у попереднього аналога приблизно 3200 ₴.



Рисунок 1.6 – Petrust PP002

Найбільш підійшов пристрій ELF Automatic Pet Feeder (рис. 1.7).

У цьому пристрої відсутнє віддалене керування, яке замінено на кнопкову панель на самому корпусі. Також присутній дисплей та прозорий контейнер. Ціна являється найбільш задовільною із усіх, але все ж доволі висока $\approx 2200\text{€}$.



Рисунок 1.7 – Годівниця ELF Automatic Pet Feeder

Найдешевшим знайденим на ринку варіантом який має функцію автоматизації є Trixie TX2 (рис. 1.8). Пристрій працює від батарейок і має вбудований таймер, але основним недоліком є його об'єм 300 мл і те, що тварина може з'їсти весь корм за один раз.



Рисунок 1.8 – Trixie TX2

Після дослідження ринку на предмет аналогів, що потенційно могли б задовольнити потреби, були виділені критерії, за якими слід обрати продукт:

- пристрій повинен виконувати роль свою роль не гірше аналогів;
- ціна повинна не перевищувати мінімум який потрібен для створення приладу власноруч;
- якість повинна задовольняти санітарні норми, а функціонал не повинен мати непотрібних функцій для збереження задовільної ціни.

Оцінивши усі варіанти за критеріями, не було знайдено аналогу, що повністю б задовольняв їх. Було прийнято рішення створити власну годівницю, що мала би усі критерії описані вище.

1.2 Постановка задачі та формування вимог

Основним недоліком існуючих зараз автономних годівниць є недостатня енергоефективність да ціна. Тож годівниця має бути зроблена таким чином, щоб максимально економно використовувати енергію при її виробництві та використанні. Крім того, так як у наш час на перше місце вийшла якість товарів та здоров'я тварин, годівниця має бути розроблена з

матеріалів, які витримують умови експлуатації та не шкодять здоров'ю тварин та бути зручною і не завдавати шкоди тваринам механічно. Вибір правильних матеріалів гарантує міцність та зносостійкість.

1.3 Дослідження технологій для створення годівниці

Для виконання поставленої задачі необхідні дві складові сервісу:

- апаратна частина для безпосереднього зберігання та дозованої видачі корму;
- програмна частина, яка оброблює отриману від датчиків інформацію про стан наповнення та видає оброблені дані.

Для створення апаратної частини було вирішено використати модуль Arduino та ESP8266 через варіативність модельного ряду та повністю відкриту архітектуру системи, що дає можливість вибрати найбільш зручний для конкретної задачі варіант навіть з неофіційного модельного ряду компанії.

1.3.1 Модель модулю

Після того, як було обрано апаратну платформу потрібно вибрати модель яка би задовольняла потреби виставлені для цієї роботи. Найбільш суттєвим є питання компактності, так як великі габарити не дадуть оптимізувати компактність пристрою та призведуть до збільшення місця яке буде займати годівниця. Таким чином вибір пав на модель ESP8266 (рис. 1.6) на базі процесора Tensilica, яка має найменший розмір серед представленого модельного ряду.

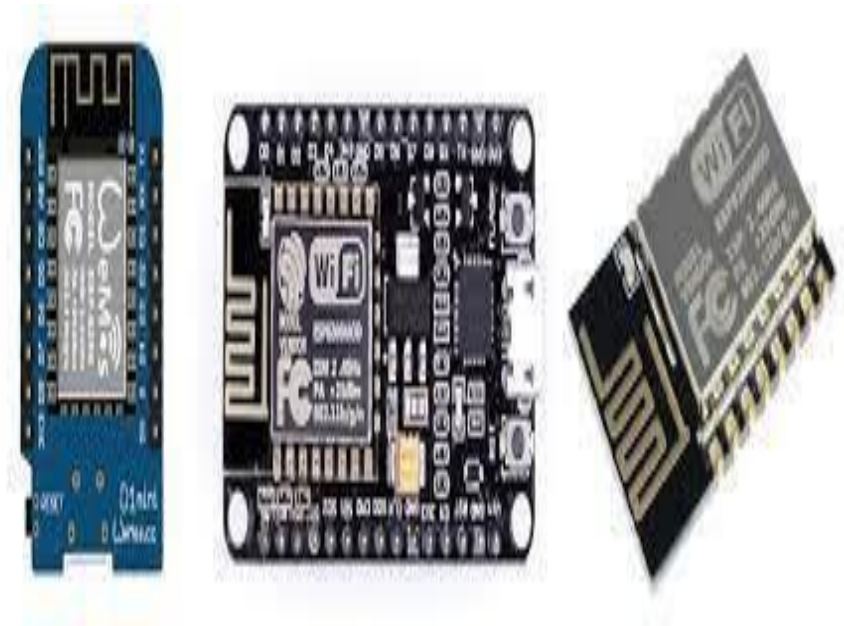


Рисунок 1.6 – ESP8266

1.3.2 Середина програмування та мова програмування

Модуль ESP8266 є однією із найзручніших платформ для створення найрізноманітніших проектів у сфері IoT.

ESP8266 – це мікроконтролер, розроблений у 2014 році і що випускається компанією Espressif Systems – китайською компанією з Шанхаю. Він являє собою мережеве рішення з WiFi-трансівером на борту плюс можливість виконання застосунків, що записуються в його пам'ять. Існує безліч модифікацій плат, які називаються зазвичай від ESP-01 до ESP-12. Сьогодні вже з'явилися ще інші назви плат від сторонніх розробників. Відмінності в платах полягає в основному в портах введення-виводу, кількості флеш-пам'яті, виду конекторів і т. п. Процесор – той самий, отже з погляду програмування немає значення яку плату програмувати. Специфікація ESP8266 наведена у табл. 1.1.

Таблиця 1.1 – Специфікації ESP8266

Protocols	802.11 b/g/n (HT20)
Frequency Range	2.4 GHz ~ 2.5 GHz (2400 MHz ~ 2483.5 MHz)
TX Power	802.11 b: +20 dBm 802.11 g: +17 dBm 802.11 n: +14 dBm
Rx Sensitivity	802.11 b: -91 dbm (11 Mbps) 802.11 g: -75 dbm (54 Mbps) 802.11 n: -72 dbm (MCS7)
Antenna	PCB Trace, External, IPEX Connector, Ceramic Chip
CPU	Tensilica L106 32-bit processor
Peripheral Interface	UART/SDIO/SPI/I2C/I2S/IR Remote Control GPIO/ADC/PWM/LED Light & Button
Operating Voltage	2.5 V ~ 3.6 V
Operating Current	Average value: 80 mA
Operating Temperature Range	-40 °C ~ 125 °C
Package Size	QFN32-pin (5 mm x 5 mm)
Wi-Fi Mode	Station/SoftAP/SoftAP+Station
Security	WPA/WPA2
Encryption	WEP/TKIP/AES
Firmware Upgrade	UART Download / OTA (via network)
Software Development	Supports Cloud Server Development / Firmware and SDK for fast on-chip programming
Network Protocols	IPv4, TCP/UDP/HTTP

User Configuration	AT Instruction Set, Cloud Server, Android/iOS App
--------------------	---

Зі специфікації видно, що питання як довго працюватиме ESP8266 від батарейок не може бути легко визначений. Споживання енергії змінюється в широкому діапазоні – при передачі на повній потужності воно становить 170 міліампер, а в режимі сні – всього 10 мікроампер.

Хоча модуль має різні моделі і зборки у своїй суті його розпіновка залишається майже однаковою (рис. 1.7).

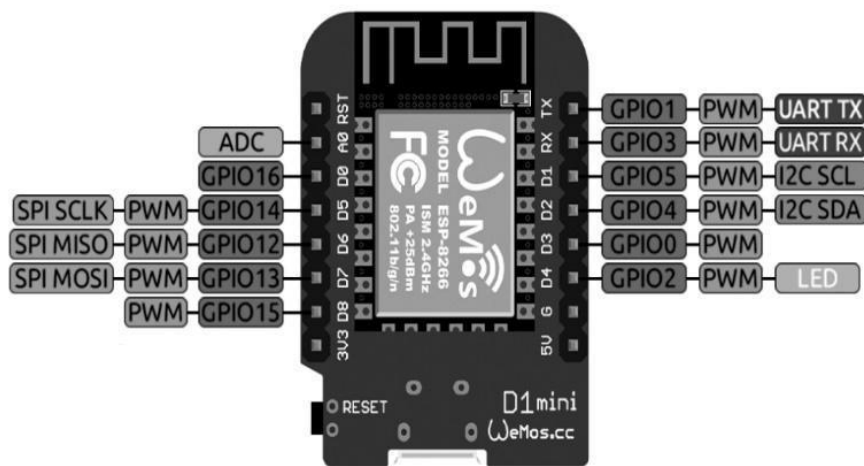


Рисунок 1.7 – Опис та розташування пінів

Типове застосування ESP8266 як апаратної основи Internet of Things найчастіше передбачає встановлення у будинках чи офісах. При цьому мережне підключення здійснюється до домашньої/офісної локальної мережі з виходом в інтернет через роутер. Користувач пристрою може контролювати його за допомогою планшета або комп’ютера через свою локальну мережу або віддалено через Інтернет.

Програми для модуля ESP8266 пишуться зазвичай мовою C. Перед завантаженням програми в модуль її слід скомпілювати з тексту в машинні коди. Редагувати текст програми найзручніше в якомусь редакторі, який має підсвічування синтаксису, вбудовану довідку та інші корисні функції, і ще

краще – в інтегрованому середовищі розробки (Integrated Development Enviroment). Працюючи в такому середовищі, ви можете написати текст програми, скомпілювати її і відразу ж завантажити в модуль. Широко відомі такі середовища розробки як Eclipse та Arduino IDE.

Отже легкість у використанні та невелика ціна (до 3\$) робить його одним із кращих компонентів при створенні будь-якого проекту IoT.

1.3.3 Середа програмування та мова програмування

Наступною частиною є вибір програмного комплексу за допомогою якого буде створено програмну частину. Так як апаратна частина проекту створена на базі модулю Arduino то і у якості середи програмування було вибрано Arduino IDE(рис. 1.8). Ця середа являється кроссплатформним застосунком основною метою якого є створення та завантаження програмного коду на плати компанії Arduino. Ця IDE дає змогу створювати програмний код який зветься скетч за допомогою мови програмування з назвою Arduino c, що по факту являється мовою C++ с фреймворком Wiring.



Рисунок 1.8 – Інтерфейс Arduino IDE

Сама C++ є статично типізованою мовою програмування з підтримкою парадигм програмування таких як : процедурне програмування, Об'єктно-орієнтоване програмування та узагальнене програмування. Існує велика кількість як платник так і безкоштовних реалізацій цієї мови. Це обумовлено великою популярністю та широкою сферою використання, а саме створення драйверів, операційних систем, застосунків, ігор, та навіть серверів. Навіть Arduino IDE створена за допомогою цієї мови.

Висновок до розділу 1

Було проаналізовано існуючі рішення їхній основний недолік, виконано розгляд існуючих аналогів та способів реалізації годівниці. Серед розглянутих технологій було обрано ті, що більше підходять для виконання цієї роботи. Було виділено таку базу технологій для розробки сучасного рішення:

- 1) Arduino – як основний компонент апаратної частини з можливістю проводити потрібні розрахунки;
- 2) Arduino IDE – було виділено як найбільш відповідну середу розробки для програмного забезпечення;
- 3) Arduino C – як найзручнішу мову програмування для створення скетчів як програмної частини на базі відомої C++.

2 МАТЕМАТИЧНІ МЕТОДИ ТА ПРОЄКТУВАННЯ СИСТЕМИ, ЩО РОЗРОБЛЯЄТЬСЯ

2.1 Розрахунок добового використання корму та об'єму годівниці

Для коректної роботи пристрою потрібно розрахувати кількість корму який буде зберігатись, залежно від типу тварини та її віку.

Наприклад для дорослого кота масою 3 кг потрібно приблизно 30 грам сухого корму на добу. Зі збільшенням маси тварини збільшується і кількість корму який використовується приблизно до 15 г на 1 кг маси тварини.

Для виконання роботи був обраний контейнер для корму об'ємом 10 літрів.

Щільність корму становить приблизно 400 г/л.

За таких даних ми можемо розрахувати що у наш посуд можна буде вмістити $400 * 10 = 4000$.

Тобто у контейнер можна насипати 4 кг сухого корму. Для домашнього улюбленця цього повинно вистачити на деякий період. А саме виходячі із усієї отриманої інформації це можна розрахувати помноживши масу корму який використовується за один сеанс прийому їжі на кількість годувань на день і розділивши на вагу доступного корму. Багато котів страждають від зайвої ваги. З цього приводу проводились дослідження. За їх результатами було встановлено, що коти які годувались за допомогою автоматичних годівниць змогли скинути зайву вагу краще ніж ті що годувались звичайним шляхом. Також установлено, що кількість годувань на день не дуже впливала на результати, але краще всього себе показала група котру годували два рази на день. Тобто оптимальною кількістю годувань домашнього улюбленця буде 2-3 рази на день. Відповідно цим даним для розрахунків у цій роботі будуть братись 3 годування на добу.

$$4000 / (15 * 3) = 88,8.$$

Розрахувавши рівняння можна побачити що корму повинно вистачити на 88 дні.

Встановивши датчики у середину контейнеру можна слідкувати за кількістю корму що залишилась. Це можна зробити використавши звичайну формулу обчислення об'єму(рис.2.1): $V = h*d$ де h – висота контейнера і d – його діаметр у прикладі з круглим контейнером.

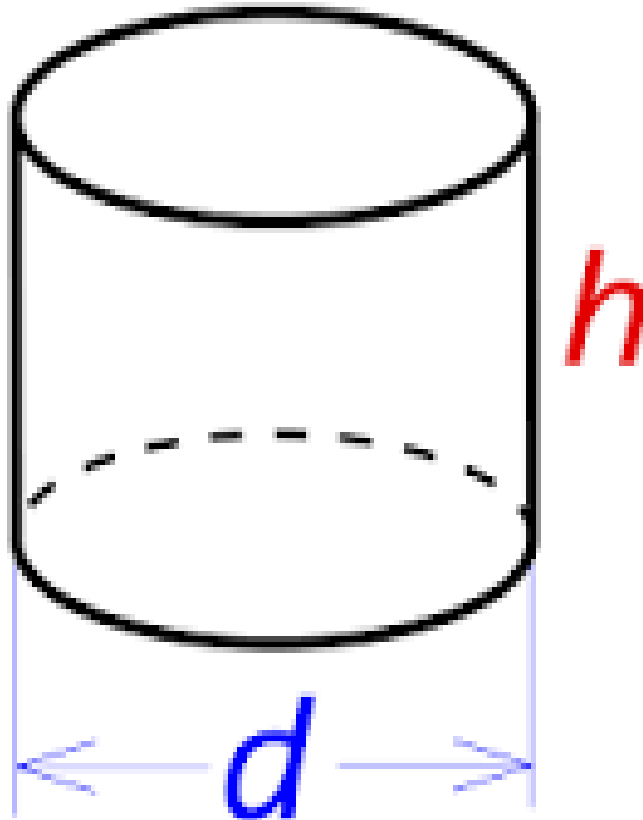


Рисунок 2.1 – Обчислення об'єму

Отже, встановивши датчик на нижній частині контейнеру, ми зможемо отримати інформацію коли потрібно знов засипати корм.

Наприклад якщо потрібно дізнаватись інформацію про залишок 20 % корму при контейнері розміром 21 см у висоту та 30 см діаметру потрібно помістити датчик на висоту $21/5 = 4,2$ см.

2.2 Дослідження впливу автоматичної годівниці для домашніх тварин і частого годування їжею для зниження ваги в домогосподарствах з кількома котами

Також було проведено невелике дослідження.

Цілі цього дослідження були подвійними: визначити, чи система годівниці, що забезпечує індивідуальний доступ до їжі та автоматичний розподіл їжі, полегшує програми зниження ваги в будинках з кількома котами для власників котів; та оцінити, чи шість разів на день знижує пошук їжі та збільшує потребу в калоріях порівняно з дворазовим харчуванням.

Ожиріння є одним із найпоширеніших захворювань домашніх котів, приблизно 25–67% котів мають надлишкову вагу або ожиріння. 1–4 котячі програми схуднення можуть бути складними. У дослідженні, що оцінювало терапевтичну дієту для схуднення, лише 14% котів досягли своєї ідеальної ваги в кінці 6 місяців. У довгостроковому дослідженні оцінки відновлення у кішок, які завершили програму схуднення, майже половина набрала вагу, а понад 25% відновили понад 50% втраченої ваги. Результати виявили дві основні проблеми, пов'язані з програмами схуднення котів: прихильність власника та поведінка, що просить. У цьому дослідженні 32 котів із зайвою вагою, які брали участь у 2-річній програмі схуднення, 22% були звільнені достроково через ці проблеми.

Вживання та перетравлення їжі потребує енергії, і дослідники припустили, що часті прийоми їжі можуть сприяти втраті ваги.

Хоча аналіз дійсно виявив негативну кореляцію між частотою прийому їжі та ожирінням, результати одного дослідження виявились спотвореними, і більшість оцінених досліджень не виявили різниці. Домашні коти еволюціонували як поодинокі мисливці, які харчувалися дрібною здобиччю приблизно 8–12 частими годуваннями на день. Надання котам необмеженого доступу до їжі для випасу протягом дня схиляє до переїдання та ожиріння. Власники кішок зазвичай забезпечують лише 1-2 прийоми їжі на день. Коли рідкісні прийоми їжі поєднуються з обмеженням енергії, це може призвести

до небажаного пошуку уваги або поведінки, пов'язаної з голодом. У недавньому невеликому дослідженні за участю 12 пацієнтів не було виявлено різниці в самооцінках голоду у пацієнтів, які дотримувалися ізокалорійної дієти, розділеної на три або вісім прийомів їжі на день. Вплив невеликих, частих прийомів їжі на почуття голоду у котів потребує подальшого вивчення. Одна з цілей поточного проекту полягала в тому, щоб визначити, чи виявляють власники більшу поведінку жебракування, коли котів годують два рази проти шести разів на день.

Хоча програми зниження ваги можуть бути складними для котів та їхніх власників, додавання кількох котів з різними дієтичними потребами в одному домі створює ще більше перешкод. Власникам часто важко відокремити котів із нормальною вагою, які віддають перевагу харчуватись протягом дня, від котів із надмірною вагою, які потребують обмеження енергії. Завдяки розвитку технологій тепер доступні автоматичні годівниці, які видають заздалегідь визначені порції їжі та контролюють небажаний доступ домашніх тварин. Деякі годівниці сканують підшкірні мікрочіпи, щоб дозволити тваринам доступ до годівниці, тоді як інші використовують бирки або нашійники, щоб дозволити або заборонити доступ до їжі.

У нещодавньому дослідженні, у якому використовувалися кілька режимів технології (цифрові ваги, автоматичні годівниці, монітори активності та камери для ласощів для домашніх тварин) під час програми зниження ваги, власники котів повідомили, що домашні ваги та годівниці є цінними інструментами.¹⁴ Основна мета цього дослідження полягала в тому, щоб визначити, чи покращилася поведінка котів, які пройшли курс схуднення, як повідомляли власники, коли котів годували з автоматичної годівниці замість того, щоб власник роздавав їжу в миску. У цьому дослідженні також оцінювався вплив частоти прийому їжі на поведінку, що вимагає жебрацтва, і потребу в калоріях для схуднення.

2.1.1 Умови

Щоб відповідати критеріям реєстрації, коти, що належать клієнтам, повинні бути старше 1 року, мати показники стану тіла (BCS) $\geq 7/9$, 15і не мати захворювань, які впливають на апетит або швидкість метаболізму або перешкоджають використанню досліджуваної дієти (наприклад, цукровий діабет, гіпертиреоз, хронічна хвороба нирок або хронічна ентеропатія). Коти також повинні були жити вдома, ділити будинок принаймні з ще одним котом, який живе в основному вдома, і бути готовими їсти виключно сухий корм. На тижні 0 коти пройшли медичний огляд, повний аналіз крові, хімічний аналіз сироватки, аналіз гормонів щитовидної залози та аналіз сечі, щоб виявити захворювання, які виключають реєстрацію. Використовуючи метод випадкової випадкової перестановки, коти були випадковим чином розподілені в одну з трьох груп: двічі на день годували з традиційної миски для їжі (B2); дворазове годування з автогодівниці (A2); або шість щоденних годувань з автоматичної годівниці (A6; PortionPro Rx; Vet Innovations). Автоматична годівниця використовує технологію радіочастотної ідентифікації, вбудовану в нашійник, щоб унікально ідентифікувати окремих тварин у годівниці. Годівниця розпізнала бирку нашійника, щоб дозволити або заборонити доступ до тарілки з їжею шляхом розміщення пластикового щитка над мискою.

Щоб запобігти спільному харчуванню, усі домашні коти груп A2 та A6 були забезпечені автоматичними годівницями. У домогосподарствах B2 не використовувалися автоматичні годівниці, і власникам було сказано зменшити розподіл їжі за допомогою традиційних методів, таких як розділення кімнат під час годування їжею. Обсяги годування (з кроком 1/8 склянки) і частоту прийому їжі (два або шість разів на день) були запрограмовані в годівниці навченим ветеринарним техніком під час першого візиту.

2.1.2 Методи дослідження

Дорослі коти з балами стану тіла (BCSs) 7/9 або вище проходили план зниження ваги на термін до 6 місяців, використовуючи традиційну миску для годування з дворазовим харчуванням (B2) або автоматичну годівницю з дворазовим харчуванням (A2) або шестиразове харчування (A6) на день. Котів зважували на тижнях 0, 2, 4, 6, 10, 14, 18, 22 і 26, і споживання їжі регулювали для підтримки темпів втрати ваги між 0,5% і 1,5% маси тіла на тиждень. Анкети для власників заповнювалися під час кожного візиту.

Один дослідник, який не мав інформації про призначення лікування, виконав початкову оцінку BCS та індексу жиру в організмі (BFI). BFI та BCS використовували для оцінки відсотка жиру в організмі, а ідеальну масу тіла (IBW) розраховували за такою формулою: $100 - \% \text{ жиру} = \% \text{ м'язової тканини}$; $[\text{поточна маса тіла (BW)} \times (\% \text{ безжирова маса}/100)]/0,8 = \text{IBW}$ при 20% жиру. Усіх котів годували однаковою терапевтичною дієтою для схуднення (сухий корм Hill's Prescription Diet Feline Metabolic; Hill's Pet Nutrition), а рекомендації щодо калорій базувалися на IBW із застосуванням рівняння $[70 \times (\text{IBW кг } 0,75)] \times 0,8$. Котів зважували на тижнях 0, 2, 4, 6, 10, 14, 18, 22 і 26, і споживання їжі регулювали для підтримки темпів втрати ваги між 0,5% і 1,5% BW на тиждень. Автоматизована годівниця дозволяла б лише 1/8 склянки поступових коригувань (приблизно 37 ккал при досліджуваному раціоні).

Щоб підтримувати узгодженість коригувань у групах, зміни калорій як у мисці, так і в групі з автоматичною годівницею базувалися на прирості 1/8 чашки. Онлайн-анкети власників заповнювалися під час кожного відвідування дослідження (рис. 2.2). Хоча питання не підтверджені, вони стосувалися змін загальної активності, голоду та якості життя. Власники зафіксували зміни в поведінці, пов'язані з голодом і стресом, такі як неналежне усунення, агресія по відношенню до людей або інших тварин, посилення вокалізації та/або випрошування їжі. Власників також попросили оцінити задоволення та легкість програми схуднення.

Програмно-апаратний комплекс диференційованого годування тварин на базі ESP8266

In answering these questions, please think about the cat participating in the study during the past 2 days.

- 1) How hungry is your cat? Select one.
 - Extremely hungry throughout the day (1)
 - Moderately hungry throughout the day (2)
 - Only hungry as meal times approach (3)
 - Never hungry (4)
- 2) How active is your cat? Select one.
 - Very active, plays a lot of the day (1)
 - Moderately active (2)
 - Not very active (3)
 - Not at all active, usually only gets up for food, water or to use the litter box (4)
- 3) Did your cat display any of the following negative behaviors? Check all that apply.
 - Urinate or defecate outside of the litter box (1)
 - Sprays urine indoors (soil a vertical surface with urine) (2)
 - Signs of aggression to people (hissing, biting, or scratching) (3)
 - Destructive behaviors, such as scratching or chewing objects in the home (4)
 - Problematic interactions with other cats or pets in the household (5)
 - Frequent meowing (6)
- 4) How would you describe your cat's quality of life? Select one.
 - Excellent (1)
 - Good (2)
 - Fair (3)
 - Poor (4)
- 5) During the last month, is your cat:
 - More affectionate (1)
 - Less affectionate (2)
 - No difference (3)
- 6) Grooms him/herself:
 - More often (1)
 - Less often (2)
 - No difference (3)
- 7) Sleeps:
 - More often (1)
 - Less often (2)
 - No difference (3)
- 8) Plays:
 - More often (1)
 - Less often (2)
 - No difference (3)
- 9) Uses the litter box shortly after eating:
 - More often (1)
 - Less often (2)
 - No difference (3)
- 10) Waking you up at night:
 - More often (1)
 - Less often (2)
 - No difference (3)
- 11) How satisfied are you with your cat's current weight loss plan(s)?
 - Extremely satisfied (1)
 - Somewhat satisfied (2)
 - Neither satisfied nor dissatisfied (3)
 - Somewhat dissatisfied (4)
 - Extremely dissatisfied (5)
- 12) How easy or difficult was this weight loss plan?
 - Extremely easy (5)
 - Somewhat easy (4)
 - Neither easy nor difficult (3)
 - Somewhat difficult (2)
 - Extremely difficult (1)

13) Since last filling out the survey, did your cat display any of the following behaviors at or just prior to the cat's usual mealtimes? Check as many as apply.

	More (1)	Less (2)	Same (3)	Not Applicable (4)
Biting the person feeding him/her (1)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Scratching the person feeding him/her (2)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Stood at the food bowl and begged (3)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Stood where the food is stored and begged (4)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Pawed and scratched where the food is stored and begged (5)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Approached food bowl even if no food is present (6)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Paced around the food bowl (7)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Follow you prior to feeding him/her (8)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Rubbing his/her body and tail against you (9)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Rubbing his/her cheek against you (10)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Paw/bat at you prior to feeding him/her (11)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Scratching furniture/carpet prior to feeding him/her (12)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Рисунок 2.2 - Онлайн-анкета, заповнена власниками котів на 2, 4, 6, 10, 14, 18, 22 і 26 тижнях

2.1.3 Статистичні методи

Дисперсійний аналіз ANOVA використовувався для перевірки групових відмінностей у початковій масі тіла та приблизному відсотку жиру в організмі, а також загальному відсотку втрати ваги під час дослідження. Тести Краскела–Уолліса використовували для перевірки групових відмінностей у середньому віці та початковому BCS. Пірсона χ^2

використовувався тест, щоб визначити, чи відрізняються стать або кількість кішок, які досягли розрахункової IBW, залежно від групи. Дисперсійний аналіз змішаної моделі повторних вимірювань із структурою коваріації AR(1) використовувався для перевірки групових і часових відмінностей у результатах (відсоток втрати маси тіла за тиждень і споживання калорій на кг). Відповіді на запитання опитування власників (зміни в загальній активності, відчуття голоду, якість життя, задоволення та легкість програми зниження ваги) були усереднені протягом періоду лікування. Поведінка пошуку їжі була перекодована як кумулятивна сума. Через ненормальний розподіл відповідей на запитання опитування групи порівнювали за допомогою тестів Краскела–Уолліса. Статистичний аналіз проводили за допомогою SPSS 28 (IBM). Для всіх аналізів значення $P < 0,05$ вважалося значущим.

2.3 Результати проведеного дослідження

Двадцять чотири коти були спочатку обстежені, але одного кота піддали евтаназії на 2-му тижні через непов'язане захворювання. Тому в дослідження були включені дані 23 котів. Кішки були випадковим чином розподілені до групи B2 ($n = 6$), A2 ($n = 10$) або A6 ($n = 7$). Один кіт був спочатку призначений до A6, але відмовився використовувати автоматичну годівницю та був переведений після 2-го тижня в групу B2. Дані 2-го тижня для цього кота були виключені з аналізу, а всі інші дані були проаналізовані з групою B2. Остаточний розподіл груп був B2 ($n = 7$), A2 ($n = 10$) і A6 ($n = 6$). Середній вік (3 роки) і BCS (9/9) не відрізнялися залежно від групи. Коти в групі B2 мали чисельно більший відсоток жиру в організмі (50,0 %) порівняно з A2 (42,7 %) і A6 (40,5 %); однак суттєвої різниці виявлено не було, ймовірно, через малий розмір вибірки ($P = 0,06$). Початкова маса тіла кішок у B2 була вищою ($P = 0,035$), але ця група також мала більшу частку самців (86 %), ніж дві інші групи (50 %). Усі коти мали нормальний стан м'язів протягом усього дослідження.

Власники повідомили, що коти споживали всю призначену їм кількість їжі, якщо не вказано інше. Вважалось, що коти досягли цільового рівня втрати ваги, якщо вони досягли BCS 4–6/9. Цей діапазон використовувався для врахування суб'єктивності та мінливості оцінки жиру в організмі за BCS та BFI. 16

Три кішки не завершили 26 тижнів дослідження або не досягли BCS $\leq 6/9$ (двоє в B2 і один в A2). Вилучення за вибором власника (один у B2), переміщення власника (один у A2) і неможливість розділити котів під час годування (один у B2) були причинами незавершення. Отже, п'ять кішок у B2, дев'ять у A2 і шість у A6 завершили дослідження. Групи відрізнялися за досягненням ідеального стану тіла ($P = 0,006$): 83,2% котів у A6, 40% котів у A2 і жодна кішка в B2 не досягли BCS $\leq 6/9$. Загальний відсоток втрати маси тіла був чисельно нижчим у групі B2 (9,7%) порівняно з A2 (15,2%) і A6 (17,1%), але це не було статистично значущим, і ретроспективний аналіз визначив, що знадобиться набагато більший розмір вибірки, щоб виявити різницю. Середній \pm SD відсоток втрати маси тіла за тиждень становив $0,68 \pm 0,086$ і не відрізнявся залежно від групи чи тижня. Подібним чином, середнє споживання калорій на кг (B2: $32,6 \pm 5,3$; A2: $37,6 \pm 8,1$; A6: $38,0 \pm 4,2$) не відрізнялося між групами або тижнями. Трьом котам у дослідженні (по одному з кожної групи) потрібно було обмежити споживання калорій нижче початкової встановленої кількості. Цим котам було потрібно лише одне зменшення на 1/8 склянки, щоб підтримувати швидкість втрати ваги між 0,5% і 1,5% BW на тиждень. Шість кішок (один у B2, три в A2 і два в A6) у дослідженні потребували однократного збільшення калорій на 1/8 склянки, щоб підтримувати встановлену швидкість втрати ваги.

Через відсутність мінливості в різні моменти часу відповіді на запитання 1–13 в опитуванні власників були усереднені в різних точках часу. Не було виявлено суттєвих відмінностей між групами для запитань 1–10 щодо відчуття котячого голоду та активності котів. Задоволеність між групами не відрізнялася ($P = 0,309$). Групові відмінності були виявлені у

питанні 12 («Наскільки легким чи важким був цей план схуднення?»); $P = 0,011$).

Попарне порівняння показало, що оцінка для В2 (4,0) була статистично значно нижчою, ніж для А2 (4,7; $P = 0,025$) і А6 (5,0; $P = 0,004$), без відмінностей між двома групами А ($P = 0,352$).

Поведінка пошуку їжі (питання 13) порівнювалася між трьома групами за допомогою тестів Краскела–Уолліса через ненормальний розподіл даних. Групові відмінності були виявлені за «стояв біля миски з їжею та благав» ($P = 0,030$), «Пішов за вами, перш ніж погодувати його/її» ($P = 0,005$) та «лапати/бити на вас, перш ніж погодувати його/її» ($P = 0,024$). Попарні порівняння показали, що група В2 демонструвала таку поведінку більше, ніж А2 і А6, без відмінностей між групами з автоматичною годівлею.

Програми зниження ваги в домогосподарствах з кількома котами є складними. Кішки з негативним енергетичним балансом часто проявляють небажану поведінку, пов'язану з пошуком їжі, і роздільне харчування тварин може бути обтяжливим. Для боротьби з епідемією котячого ожиріння необхідна розробка інструментів і стратегій для полегшення планів схуднення. Це дослідження демонструє, що система годівниці, яка забезпечувала індивідуальний доступ до їжі та автоматизоване порціонування їжі, полегшила програму схуднення для власників котів. Автоматизована годівниця також зменшила випадки певної поведінки пошуку їжі, як повідомляють власники. Це говорить про те, що коти почали відмежовувати власника тварини від годування їжею. Однак використання автоматичних годівниць, як виявилось, не покращило загальне почуття ситості або не зменшило всі види поведінки, пов'язані з пошуком їжі.

Автоматична годівниця в цьому дослідженні забезпечила власникам котів дві зручності: можливість розділяти котів під час їжі та точне порціонування їжі. Не оцінювалося, чи вважали власники програму схуднення за допомогою автоматичної годівниці легшою через розділення котів чи порціонування їжі. Таким чином, результати цього дослідження не

можуть бути безпосередньо перенесені на системи годівниці, які лише відокремлюють котів або забезпечують лише розподіл їжі. Більшість кішок добре звикли носити нашийники та бирки та використовувати автоматичну годівницю, лише один кіт відмовився використовувати автоматичну годівницю.

Кішки, які їли з автоматичної годівниці, мали більше шансів досягти свого ідеального стану, ніж коти, які їли з мисок. Жодна кішка в групі миски не змогла досягти свого ідеального стану тіла, незважаючи на те, що споживання калорій було подібне до котів з автоматичної годівниці. Розділення домашніх котів у групі В2 могло бути складнішим, що призвело до крадіжки їжі. Незважаючи на статистично суттєві відмінності, коти в групі В2 також мали вищий середній відсоток жиру в організмі на початку дослідження, що збільшило б час, необхідний для досягнення ідеального стану тіла, і, ймовірно, сприяло нижчому відсотку успіху в цій групі.

Харчування невеликими частими прийомами їжі вважається методом покращення насичення та збільшення витрат енергії через термічний ефект травлення. У цьому дослідженні порівнювали котів, які їли ту саму їжу шість разів або двічі на день ізокалорійним способом. Коти, які харчувалися шість разів на день (А6), мали подібні темпи втрати ваги та споживання калорій, як і коти, які їли два рази на день (А2). Власники з двох груп автоматичних годівниць також повідомили про подібні випадки поведінки, пов'язаної з пошуком їжі. Ці результати свідчать про те, що збільшення частоти прийому їжі мало клінічно впливає на втрату ваги у котів. Однак розмір вибірки дослідження був невеликим, і для повного визначення впливу частого прийому їжі може знадобитися більше котів.

Харчування шість разів замість двох на день не призвело до швидшої втрати ваги або потреби збільшити споживання калорій для підтримки відповідних темпів втрати ваги. Більш часте харчування також не зменшило частоту поведінки, пов'язаної з голодом. Власники кішок, які використовують автоматичні годівниці, які порційно розподіляють їжу та

відокремлюють тварин, описують програму схуднення як простішу, ніж ті, хто використовує традиційну миску для годування. Кішки, які їли з мисок, також мали менше шансів досягти ідеального стану тіла. Використання цих типів годівниць може покращити успіх програм зниження ваги в домогосподарствах з кількома котами.

Висновок до розділу 2

У розділі було проаналізовано добову норму корму для домашніх тварин на прикладі звичайного кота та розраховано кількість корму, який він споживає. Відповідно до цього отримано дані щодо часу до спустошення контейнеру та способів моніторингу спустошення.

Крім того, проведено дослідження щодо впливу автоматизації наповнення годівниці та кількості разів годування котів на їх стан та можливість схуднення.

3 АПАРАТНО-ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ

3.1 Вибір апаратної платформи

Головною та керуючою частиною проекту є мікроконтролер. Це виконаний у вигляді мікросхеми спеціалізований комп'ютер, який включає у себе мікропроцесор, оперативну(RAM) та постійну пам'ять(ROM) для збереження виконуваного коду програм і даних, також порти вводу-виводу(I/O) і блоки зі спеціальними функціями (лічильники, компаратори, АЦП та інші).

По суті, це – однокристальний комп'ютер, здатний виконувати прості завдання, який використовується для керування електронними пристроями. Використання однієї мікросхеми значно знижує розміри, енергоспоживання і вартість пристроїв, побудованих на базі мікроконтролерів, що і є однією із головних частин роботи.

Найбільш відомими мікроконтролерами можна назвати лінійку Raspberry PI, Arduino та мікроконтроллери ESP.

Найбільш дешевими з названих є мікроконтроллери ESP, що є дуже важливим фактором при виборі пристрою, так як одним із основних завдань є максимально зменшити затрати на виробництво у порівнянні з іншими гідівницями представленими на ринку України.

Судячи з вимог до апаратної платформи описаних у розділі 1, а саме: невеликий розмір компактного застосування, зручність та зрозумілість у використанні, низька ціна у порівнянні з аналогами, було виокремлено декілька можливих апаратних платформ.

Для виконання поставленої задачі необхідні дві складові сервісу:

- апаратна частина для безпосереднього контролю зберігання та дозованої видачі корму;

- програмна частина, яка оброблює отриману від датчиків інформацію про стан наповнення та видає оброблені дані.

3.1.1 Порівняння апаратних платформ

На рис. 3.1 наведено порівняння можливих апаратних платформ.

	Huzzah ESP8266	NodeMCU v2	Generic ESP	imp001	Edison	Core	Photon	WiPy	Oak
Company	Adafruit	NodeMCU	Espressif	Electric Imp	Intel	Particle (Spark)	Particle	WiPy	Digistump
Microcontroller	Xtensa lx106 micro 32-bit	Xtensa lx106 micro 32-bit	Xtensa lx106 micro 32-bit	Cortex M3	Atom Z3400 + Quark	STM32F103 Cortex M3	STM32F205 Cortex M3	CC3200 onboard Cortex M4	Xtensa lx106 micro 32-bit
Wifi Module	ESP-12	ESP-12E	ESP-12	Unknown	BCM43340	TI CC3000	BCM43362	TI CC3200	ESP8266EX
Wifi Standards	802.11 b/g/n	802.11 b/g/n	802.11 b/g/n	802.11 b/g/n	802.11 a/b/g/n	802.11 b/g	802.11 b/g/n	802.11 b/g/n	802.11 b/g/n
Concurrent Sockets	5	5	5	1 to cloud	4	4	5	8	5
Access Point Mode	WiFi Direct, Soft-AP	WiFi Direct, Soft-AP	WiFi Direct, Soft-AP	None	WiFi Direct	No	Soft-AP	AP, Station, WiFi Direct	WiFi Direct, Soft-AP
True SSL (HTTPS)	No	No	No	Yes	Yes with wolfSSL	No	No	Yes (2 concurrent)	No
Arduino IDE Programmable	Yes	Yes	Yes	No	Yes	No	No	No	No
Manufacturer Cloud	No	No	No	Yes	No	Yes	Yes	No	Yes
Size	25 x 38 mm	49 x 24.5 mm	24 x 16 mm	24 x 32 mm	35.5 x 25.0 mm	36.58 x 20.3 mm	36.58 x 20.32 mm	25 x 45 mm	23.4 x 30 mm
Transmit Current	~250 mA	~250 mA	215 mA (typ)	400 ma (max)	120 mA unofficial	300 mA (max)	430 mA (peak)	229 mA	220 mA (max)
Digital Pins	9	13	9	6 shared	20	10	18	25	11
Analog Pins	1	1	1	6 in & out, shared	6	8	10	3	1
Antenna Type	PCB	PCB	PCB	PCB	Chip or uFL	Chip or uFL	Chip and uFL	Chip and uFL	Chip
Flash	4 MB	4 MB	Can Vary	not stated	4 GB	2MB + 128 KB	1 MB	2 MB	1 MB
RAM	128 KB	128 KB	128 KB	128 KB	1 GB	20 KB	128 KB	256 KB	64 KB
Supply Voltage	3 to 12 volts	5 volts	3.3 volts	2.7 to 3.3 volts	7 to 15 volts	3.6 to 6 volts	3.6 to 5.5 volts	3.6 to 5.5 volts	3.3 to 4.5 volts
Signal Voltage	3.3 volts	3.3 volts	3.3 volts	3.3 volts	1.8 volts	3.3 volts	3.3 volts	3.3 volts	3.3 volts
Cost (US dollars)	\$9.95	\$12.95	\$3.37 to \$6.95 +	29.95 +	49.95 +	\$39.00	\$19.00	29 Euro (About \$33)	\$10.00

Рисунок 3.1 – Порівняння апаратних платформ

Arduino Nano – це функціональний аналог Arduino UNO, який має значно менший форм-фактор.

Крім розміру відмінність Nano полягає в тому, що плата не має власного роз'єму для зовнішнього живлення та використовує Mini-USB роз'єм для підключення до комп'ютера, а також штиркові контакти, що дозволяє легко встановлювати її на макетну плату.

Живлення плати можливе через USB або з використанням зовнішнього джерела живлення (пін VIN), у цьому випадку напруга живлення може становити 6–12 В, оскільки плата має вбудований стабілізатор або 5 В стабільного зовнішнього живлення (пін "5V").

Програмується за допомогою програмного забезпечення ArduinoIDE.

Технічні характеристики Arduino Nano представлені у табл. 3.1.

Таблиця 3.1 – Технічні характеристики Arduino Nano

Мікроконтролер	ATmega328P
Робоча напруга	5 В
Тактова частота	16 МГц
Вхідна напруга живлення	6-12 В
Цифрові входи/виходи	14 (6 з яких ШІМ)
Аналогові входи	6
Максимальний струм одного висновку	40 мА
Максимальний струм одного висновку	(3.3 В): 50 мА
Інтерфейси	I2C/TWI, SPI, UART, PWM
USB-перетворювач	CH340G
Flash-пам'ять	32 кбайт
Оперативна пам'ять (SRAM)	2 кбайт
EEPROM-пам'ять	1 кбайт
Розмір	44,6 мм x 17,8 мм x 7 мм
Вага	6 г

Raspberry Pi 4 model b (распберрі) (рис. 3.2) – остання версія популярного у всьому світі мікрокомп'ютера, у новій моделі є великі зміни щодо попередників Raspberry Pi 3 та Raspberry Pi 2. Остання версія працює на потужному процесорі Broadcom BCM2711 із частотою 1,5 ГГц, має 4 Гбайт оперативну пам'яті і може підтримувати відразу два монітори з роздільною здатністю 4К.

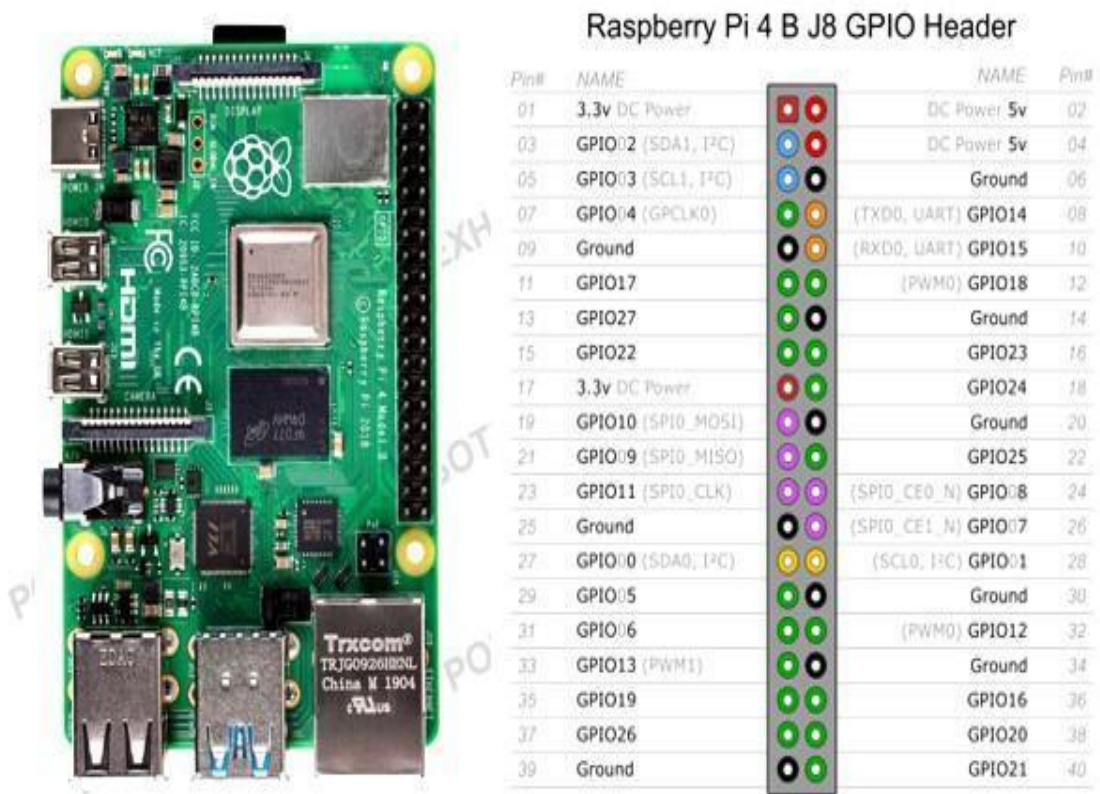


Рисунок 3.2 – Raspberry Pi 4 B

На верхній стороні плати знаходиться 40-контактний роз'єм GPIO, виходи MIPI DSI та MIPI CSI для дисплеїв та камер. Поруч із роз'ємами GPIO розміщений екранований модуль для Wi-Fi та Bluetooth 5.0. Слот для microSD розміщений унизу плати. Поруч із процесором знаходиться ОЗУ DDR4, що замінила модуль DDR2. На довгій стороні розміщено аудіовиход 3,5 мм і два входи microHDMI, на короткій стороні розміщено USB 2.0 та два USB 3.0. Корпус мікрокомп'ютера має продуманий дизайн, складається з пластикових панелей, які збираються на клямки та гвинти з комплекту.

Вентилятор слід встановити на корпусі безпосередньо над процесором та підключити для живлення до роз'ємів GPIO – 4 та 6.

У стандартну комплектацію входять такі елементи: материнська плата, блок живлення 5В з USB, пластиковий корпус з вентилятором, кабель CAT.5e, кабель microHDMI-HDMI (довжина 2 м), картка пам'яті microSD 1 Гбайт із встановленою операційною системою Raspbian. Таким чином вам не потрібно турбуватися про пошук сумісного джерела живлення, про встановлення та налаштування ОС та додаткові аксесуари. Характеристики Raspberry Pi 4 В винесені у табл. 3.2.

Таблиця 3.2 - Характеристики Raspberry Pi 4 В

Частота	1500 МГц, Broadcom BCM2711, Cortex-A72
Кількість ядер	Чотири
Оперативна пам'ять	1 ГБ, 2 ГБ, 4 ГБ, LPDDR4, SDRAM
Інтегрована графіка	VideoCore VI 500 МГц
Контролер звуку	Інтегрований
Інтерфейс	Wi-Fi 802.11ac, Bluetooth 5.0 LE, гігабітовий Ethernet RJ-45
Швидкість передачі даних через LAN	1000 Мбайт Ethernet RJ-45
Порти	microHDMI, UART, SPI, I2C, аудіовиход 3.5 мм, USB 3.0, USB 2.0
Постійна пам'ять	microSD
Живлення	5 В, 3 А через microUSB
Інтерфейси взаємодії	DSI, CSI, 40-pin GPIO

Операційна система, встановлена за замовчуванням на карту пам'яті, Raspbian заснована на популярному та стабільному програмному забезпеченні Linux Debian 10, яка використовується на серверах. Для менш продуктивних комп'ютерів Raspbian використовується полегшене середовище робочого столу. Все керування здійснюється, як у Microsoft Windows за допомогою лівого та правого клацання кнопок миші. У Raspbian панель завдань знаходиться вгорі, в ньому розміщені попередньо встановлені програми, включаючи Libre Office (аналог MS Office), браузер Chromium, інструменти для програмування та прості аксесуари – все це знайоме для користувачів, які раніше працювали в Linux. За потреби програми можна встановити за допомогою утиліти «Встановлення та видалення програмного забезпечення».

Кого не влаштує інтерфейс Raspbian, можна встановити операційну систему на Raspberry Pi 4 b від інших постачальників програмного забезпечення — Windows 10 IoT Core (для звиклих до середовища Microsoft) або OSMC (для створення мультимедійного центру). Можна протестувати й інші ОС, наприклад LibreELEC, Ubuntu MATE або RISC OS, які підходять для виконання різних завдань. Найбільшим же недоліком цього мікроконтролера є його ціна від 4 тисяч гривень, що взагалі не підходить до вимог роботи.

Після аналізу всіх згаданих платформ було вирішено обрати для виконання кваліфікаційної роботи Generic ESP8266 від виробника Espressif Systems, оскільки вона виділяється перш за все своєю низькою ціною, достатньою продуктивністю для виконання поставлених завдань, невеликим розміром, наявністю необхідних бездротових технологій для передачі інформації користувачу та доступністю на ринку, а варіативність модельного ряду та повністю відкриту архітектуру системи, що дає можливість вибирати найбільш зручний для конкретної задачі варіант навіть з неофіційного модельного ряду компанії.

Модуль ESP8266 є однією із найзручніших платформ для створення найрізноманітніших проектів у сфері IoT.

ESP8266 - це мікроконтролер, розроблений у 2014 році, що випускається компанією Espressif Systems - китайською компанією з Шанхаю. Він являє собою мережеве рішення з WiFi-трансівером на борту плюс можливість виконання застосунків, що записуються в його пам'ять. Існує безліч модифікацій плат, які називаються зазвичай від ESP-01 до ESP-12. Сьогодні вже з'явилися ще інші назви плат від сторонніх розробників. Відмінності в платах полягає в основному в портах введення-виводу, кількості флеш-пам'яті, виду конекторів і т. п. Процесор той самий, отже з погляду програмування немає значення яку плату програмувати.

Специфікація ESP8266:

- напруга живлення: 3,3 В
- енергоспоживання: 10 мкА ... 170 мА
- флеш-пам'ять: до 16 Мбайт максимум (зазвичай 512 кбайт)
- процесор: Tensilica L106, 32 біта
- швидкість процесора: 80...160 МГц
- ОЗП: 32 кбайт + 80 кбайт
- порти введення-виведення загального призначення: 17
(мультиплексовані з іншими функціями)
- АЦП: 1 введення з роздільною здатністю 1024
- підтримка 802.11: b/g/n/d/e/i/k/r
- максимальна кількість підключень TCP: 5

Зі специфікації видно, що питання як довго працюватиме ESP8266 від батарейок не може бути легко визначений. Споживання енергії змінюється в

широкому діапазоні - при передачі на повній потужності воно становить 170 міліампер, а в режимі сні - всього 10 мікроампер.

Хоча модуль має різні моделі і зборки у своїй суті його розпіновка залишається майже однаковою (рис. 3.3).

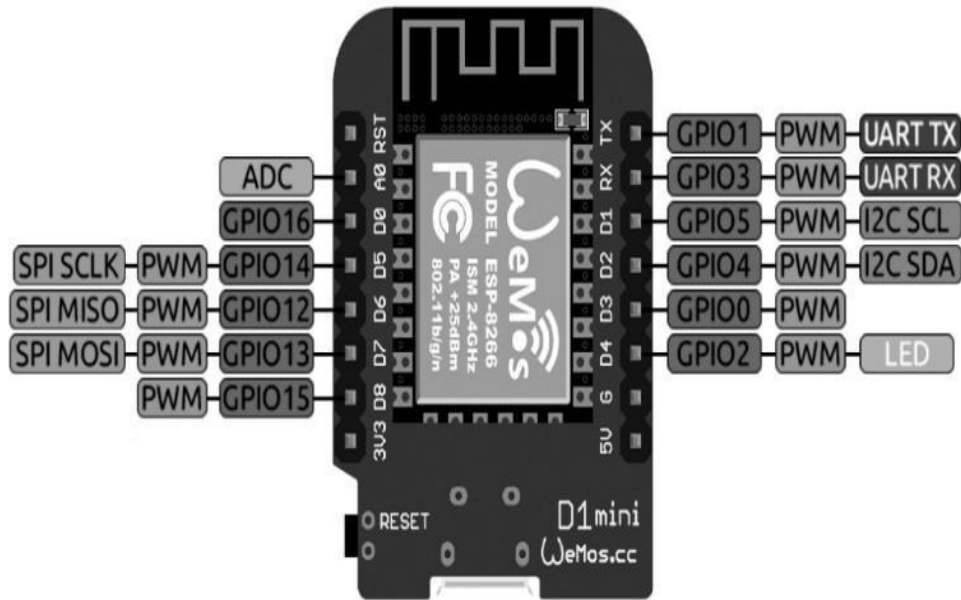


Рисунок 3.3 – Опис та розташування пінів

Типове застосування ESP8266 як апаратної основи Internet of Things найчастіше передбачає встановлення у будинках чи офісах. При цьому мережне підключення здійснюється до домашньої/офісної локальної мережі з виходом в інтернет через роутер. Користувач пристрою може контролювати його за допомогою планшета або комп'ютера через свою локальну мережу або віддалено через Інтернет.

Програми для модуля ESP8266 пишуться зазвичай мовою C. Перед завантаженням програми в модуль її слід скомпілювати з тексту в машинні коди. Редагувати текст програми найзручніше в якомусь редакторі, який має підсвічування синтаксису, вбудовану довідку та інші корисні функції, і ще краще – в інтегрованому середовищі розробки (Integrated Development

Environment). Працюючи в такому середовищі, ви можете написати текст програми, скомпілювати її і відразу ж завантажити в модуль. Широко відомі такі середовища розробки як Eclipse та Arduino IDE.

Отже легкість у використанні та невелика ціна (до 3\$) робить його одним із кращих компонентів при створенні будь-якого проекту IoT.

3.2 Детальний опис обраної апаратної платформи та її компонентів

3.2.1 Підбір сервомотору

Для коректного функціонування розроблюємого пристрою потрібно знайти механізм який буде фізично регулювати подачу корму для тварин. Таким чином було прийнято рішення використовувати сервопривід MG995 (рис. 3.4).



Рисунок 3.4 – Сервопривід MG995

Живлення сервоприводу MG995 здійснюється через зовнішній адаптер. Дворядний шарикопідшипник і металеві передачі сервоприводу MG995 забезпечують стабільну і надійну роботу. Характеристики сервомотору представлені у таблиці (табл. 3.3).

Таблиця 3.3 – Характеристика сервоприводу MG995

Довжина проводу	300 мм
Розмір сервоприводу:	40 мм × 19 мм × 43 мм
Вага сервоприводу:	55 г
Швидкість роботи (4.8 V без навантаження)	0,17 с / 60 градусів
Швидкість роботи (V 6.0 без навантаження)	0,13 с / 60 градусів
Пусковий момент: 4,8 В	13 кг-см (180,5 унцій-в)
Пусковий момент: у 6В	15 кг-см (208,3 унцій-в)
Робоча напруга:	4,8 – 7,2 В
Сумісність роз'ємів	Futaba, Hitec, Sanwa, GWS і т. д.

3.2.2 Підбір акумулятору

Так як пристрою необхідне постійне хоча й невелике живлення було вирішено використовувати акумулятор бо використання звичайних батарейок несе в собі додаткові затрати на їх постійне покупку, що призводить до додаткових витрат та зайвим витратам часу яких можна запобігти.

Після пошуку було обрано декілька варіантів:

Перший з них акумулятор BAILONG 18650 (рис. 3.5):

- типорозмір батареї: 18650

- напруга: 3,7 V
- тип батареї: Li-Ion
- ємність: 8800 mAh



Рисунок 3.5 – Акумулятор BAILONG 18650

Другим варіантом був розглянутий акумулятор Акумулятор 18650 SAMSUNG INR18650 (рис. 3.6).



Рисунок 3.6 – Акумулятор 18650 SAMSUNG INR18650

Характеристики акумулятора 18650:

- Типоразмер: 18650
- Напряжение номинальное: 3,6В
- Полный разряд: 2,65В
- Полный заряд: 4,2В
- Максимальный непрерывный разрядный ток - 8А, не постоянный - до 13А
- Максимальный зарядный ток - 2А
- Температура заряда: 0 to 45С
- Температура эксплуатации: -10 to 60С
- Размеры: 18,55 x 65,25мм
- Вес: макс. 50гр

Ще одним варіантом було обрано Акумулятор 18650 SAMSUNG

INR18650 29E 2900 mAh (рис. 3.8).



Рисунок 3.8 – Акумулятор 18650 SAMSUNG INR18650 29E

Усі параметри акумулятора були взяті з офіційного даташиту акумулятора і записані в табл. 3.4.

Таблиця 3.4 – Специфікація акумулятору 18650 SAMSUNG INR18650 29E

Item	Specification
3.1. Nominal Capacity	2.850mAh (0.2C, 2.50V discharge)
3.1.1. Typical Capacity	2.850mAh (0.2C, 2.50V discharge)
3.1.2. Minimum Capacity	2.750mAh (0.2C, 2.50V discharge)
3.2. Charging Voltage	4.20 ± 0.05 V
3.3. Nominal Voltage	3.65V (0.2C discharge)
3.4. Charging Method	CC-CV(constant voltage with limited current)
3.5. Charging Current	Standard charge: 1.375mA
3.6. Charging Time	Standard charge: 3hours
3.7. Max. Charge Current	2750mA (not for cyclelife)

Item	Specification
3.8. Max. Discharge Current	2.750mAh (continuous discharge) 8250mAh (not for continuous discharge)
3.9. Discharge Cut-off Voltage	2.50V
3.10. Cell Weight (max. (g))	48g
3.11. Cell Dimension	Diameter(max.) : 18.40 mm Height(max) : 65.00 mm
3.12. Operating Temperature (Cell Surface Temperature)	Charge: 0 to 45°C Discharge: □20 to 60°C

3.2.3 Підбір дисплею

Додатково було вирішено приєднати до пристрою дисплей для додаткової зручності у користуванні то виводу їм корисної та потрібної інформації.

Зважаючи на усі потреби було обрано чотирирядковий символний індикатор LCD2004A (рис. 3.9).



Рисунок 3.9 – Чотирирядковий символний індикатор LCD2004A

Індикатор LCM 2004A виконаний за технологією STN.

STN (super twisted nematic) – рідкий кристал із поворотом від 180° до 270°.

Індикатор має світлодіодне (LED) підсвічування. Достоїнствами цього підсвічування є його довговічність, відсутність необхідності в додатковому джерелі живлення. Характеристики:

- тип дисплею: Символьний;
- вбудований контролер: KS0066;
- текстова роздільна здатність: 20 символів 4 рядки;
- технологія: STN (позитив);
- тип підсвічування: LED;
- колір крапки: Білий;
- колір тла: Синій;
- розмір: 77 мм × 46,7 мм × 11,5 мм;
- робоча температура: – 20 ... + 70;

Плата дисплея має 16 цифрових виходів, з яких 6 є обов'язковими для підключення (табл. 3.5).

Таблиця 3.5 – Виходи LCD2004A

№ з/п	ПК-дисплей	Arduino	Опис контактів
1	VSS	GND	“земля”
2	VDD	5V	Напруга живлення дисплея
3	VO	A0(Потенціометра)	Контрактність екрану
4	RS	D12	Вибір регістру

№ з/п	ПК-дисплей	Arduino	Опис контактів
5	RW	GND	Читання та запис даних (при підключені до “землі” вмикається режим запису)
6	EN	D11	Включення
7	D0	-	цифровий контакт (не використовується при 4- бітному режимі)
8	D1	-	цифровий контакт (не використовується при 4- бітному режимі)
9	D2	-	цифровий контакт (не використовується при 4- бітному режимі)
10	D3	-	цифровий контакт (не використовується при 4- бітному режимі)
11	D4	D5	цифровий контакт
12	D5	D4	цифровий контакт

№ з/п	ПК-дисплей	Arduino	Опис контактів
13	D6	D3	цифровий контакт
14	D7	D2	цифровий контакт
15	A	5V	напруга живлення підсвічування
16	K	GND	загальний провід підсвічування

3.3 Побудова макетної та принципової схеми

Для побудови макетної та принципової схеми було обрано програмне забезпечення Fritzing (рис. 3.11).

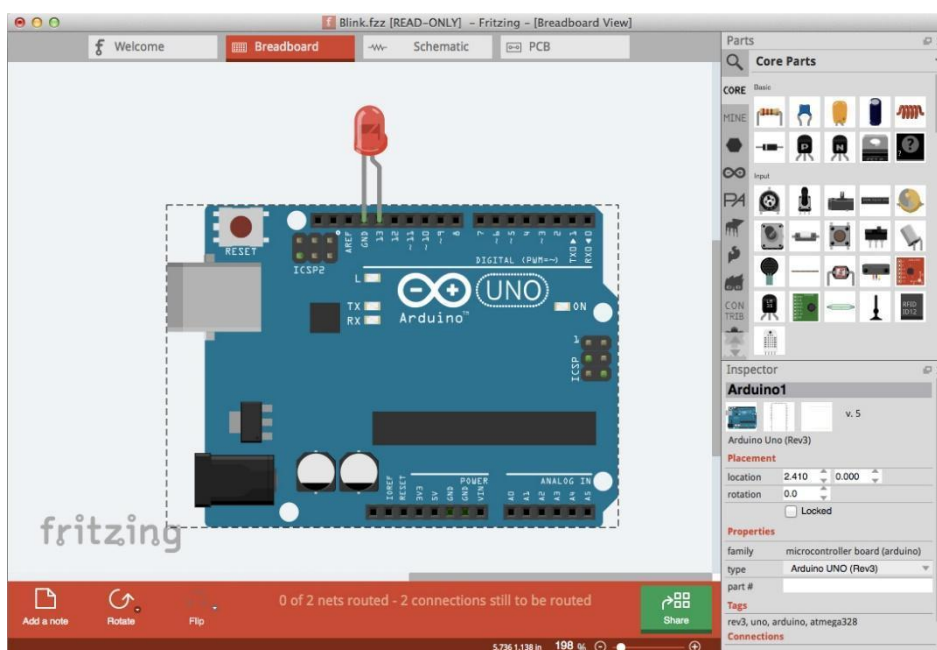


Рисунок 3.11 – Зовнішній вигляд інтерфейсу Fritzing

Fritzing – це вільне та відкрите програмне забезпечення для розробки електронних схем та проектів на мікроконтролерах. Fritzing має простий та інтуїтивно

зрозумілий інтерфейс, що робить його доступним навіть для початківців.

Програмне забезпечення було створено за мотивами мови програмування Processing і мікроконтролера Arduino і дозволяє дизайнеру, художнику, досліднику чи любителю документувати свій прототип на базі Arduino та створювати макет друкованої плати для виробництва. Пов'язаний веб-сайт допомагає користувачам обмінюватися та обговорювати проекти та досвід, а також зменшувати витрати на виробництво. Крім того важливим є можливість створювати принципові схеми, що зберігає час та прибирає потребу в додатковому софті (рис. 3.12).

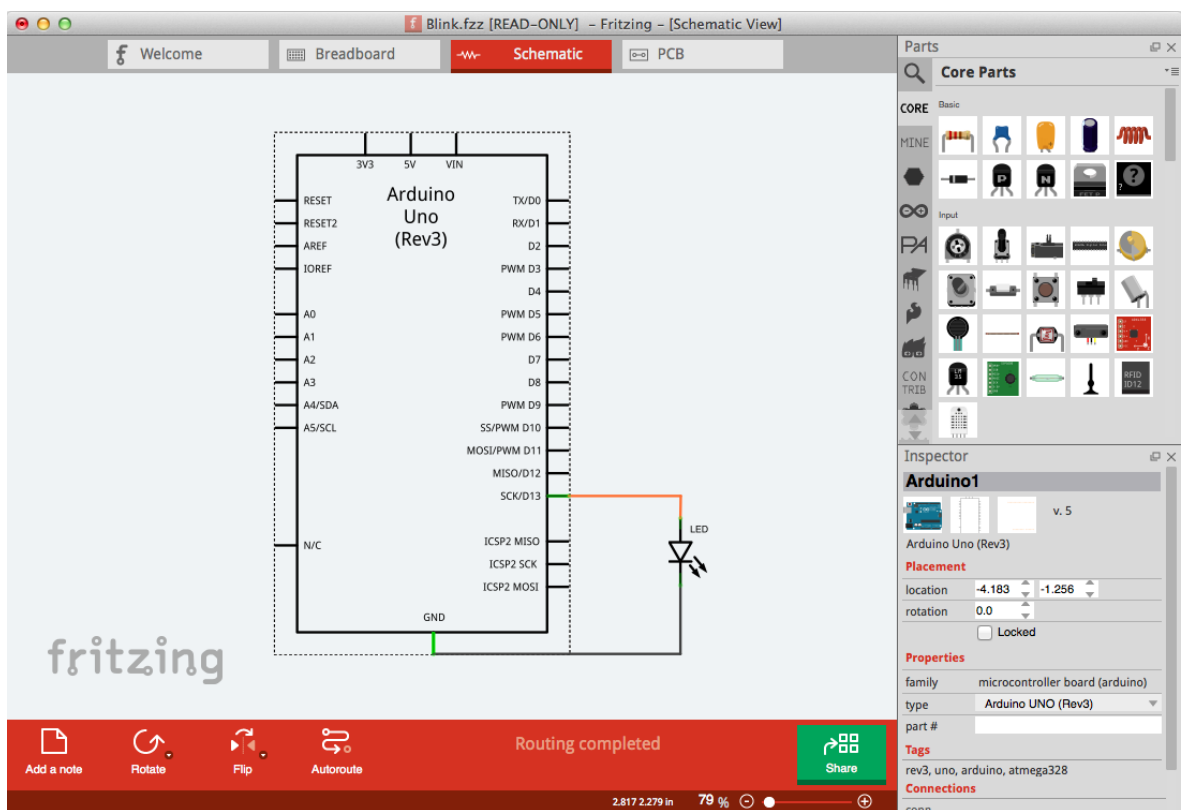


Рисунок 3.12 – Інтерфейс створення принципової схеми

Основна функціональність Fritzing включає:

- розробка електричних схем з можливістю вставки елементів з бібліотеки компонентів;
- розміщення компонентів на платі, створення розмітки та друк схеми;

- створення 3D-моделей плат, що дає можливість зробити візуальний прототип вашого проекту;

- імпорт / експорт проектів у різних форматах.

Деякі переваги Fritzing перед іншими аналогами:

- простий та зрозумілий інтерфейс, що зробиє його доступним для початківців;

- бібліотека компонентів містить більше 1000 елементів;

- велика кількість різноманітних налаштувань, що дозволяє налаштовувати різні параметри проекту;

- вбудований редактор коду, що дозволяє швидко перевірити роботу проекту.

Однак, важливо зазначити, що Fritzing не є ідеальним та має деякі обмеження, такі як обмежену функціональність порівняно з професійними програмними засобами, такими як Altium Designer (рис. 3.13) та KiCAD. Також, Fritzing може мати проблеми з рендерингом та зображенням деяких складних елементів на схемі.

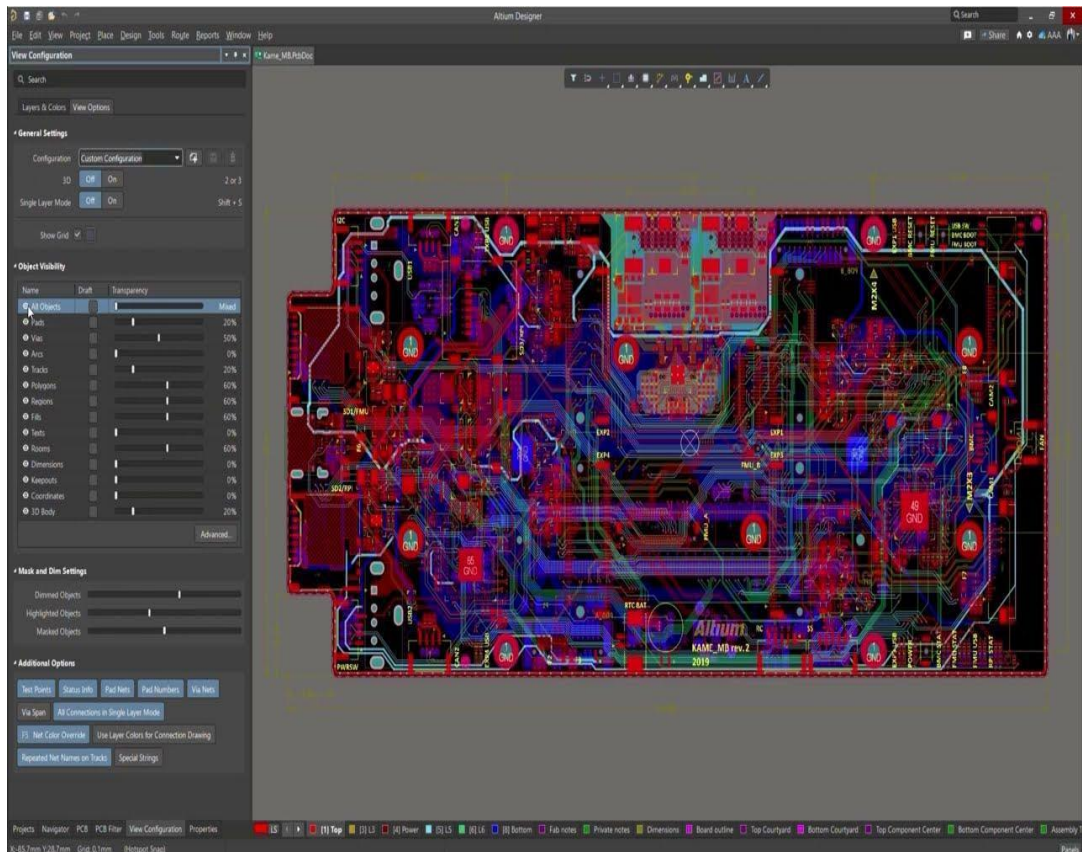


Рисунок 3.13 – Інтерфейс Altium Designer

Отже, це ПЗ є прекрасним вибором для початківців та для швидкого створення прототипу електронних проектів. В той же час, якщо вам потрібні більш продвинуті можливості, ви можете розглянути інші програмні засоби.

Для розробки макетної схеми (рис. 3.14) плати було використано стандартні елементи з бібліотеки ПЗ.

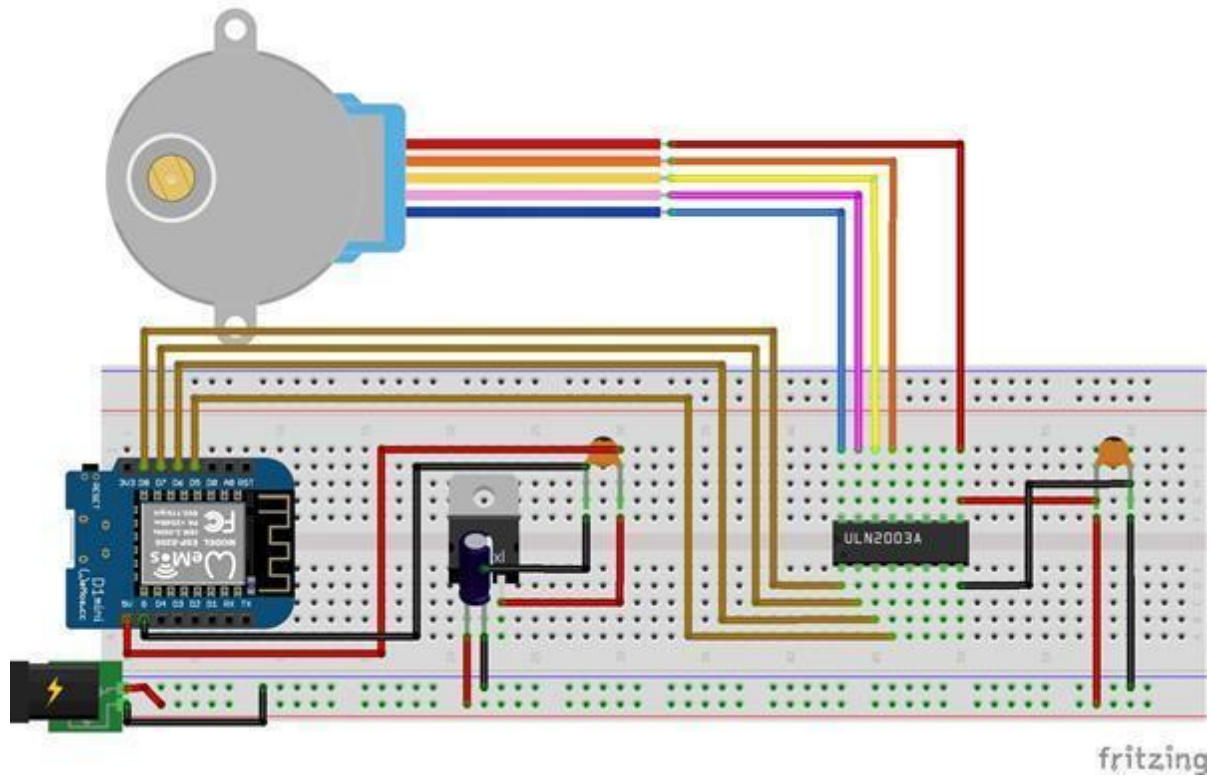


Рисунок 3.14 – Макетна схема

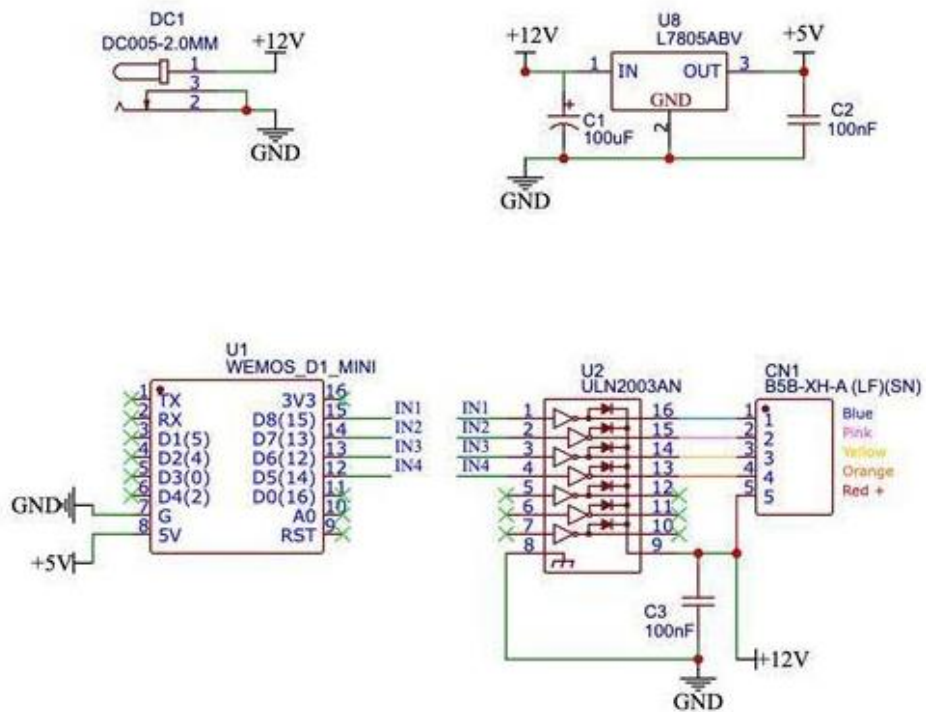


Рисунок 3.15 – Принципова схема

3.4 Вибір мови програмування та середовища розробки

3.4.1 Вибір мови програмування

ESP8266 підтримує код на великій кількості мов програмування.

Програмування може проходити на мовах Lua, Python, C/C++, Java. Під час роботи над пристроєм як можливі варіанти було розглянуто різні мови.

Lua є легкою, високорівневою, багатопарадигмальною мовою програмування, розробленою в основному для вбудованого використання в програмах. Lua є кросплатформною, оскільки інтерпретатор скомпільованого байт-коду написаний мовою ANSI C, а Lua має відносно простий C API для вбудовування його в програми. Lua зазвичай описують як «мультипарадигмальну» мову, що забезпечує невеликий набір загальних функцій, які можна розширити, щоб відповідати різним типам проблем.

Lua не містить явної підтримки успадкування, але дозволяє реалізувати його за допомогою метатаблиць. Так само Lua дозволяє програмістам реалізовувати простори імен, класи та інші пов'язані функції, використовуючи реалізацію однієї таблиці; першокласні функції дозволяють використовувати багато прийомів функціонального програмування; а повне лексичне визначення дозволяє приховувати детальну інформацію для забезпечення дотримання принципу найменших привілеїв. Загалом, Lua прагне надати прості, гнучкі метафункції, які можна розширити за потреби, а не надавати набір функцій, специфічний для однієї парадигми програмування. Як наслідок, базова мова легка – **повний довідковий інтерпретатор скомпільовано лише близько 247 кбайт – і легко адаптується до широкого діапазону програм.**

Python – це мова програмування високого рівня загального призначення. Його філософія дизайну наголошує на читабельності коду з використанням значних відступів. Python динамічно типізується та збирає сміття. Він підтримує кілька парадигм програмування, включаючи структуроване (зокрема процедурне), об'єктно-орієнтоване та функціональне

програмування. Його часто описують як мову «батареї включені» через його повну стандартну бібліотеку.

Python є багатопарадигмальною мовою програмування. Об'єктно-орієнтоване програмування та структуроване програмування повністю підтримуються, і багато їхніх функцій підтримують функціональне та аспектно-орієнтоване програмування (включаючи метапрограмування та метаоб'єкти). Багато інших парадигм підтримуються через розширення, включаючи проектування за контрактом і логічне програмування.

Python використовує динамічну типізацію та комбінацію підрахунку посилань і збирача сміття, що визначає цикл, для керування пам'яттю. Він використовує динамічне розпізнавання імен (пізні зв'язування), яке зв'язує імена методів і змінних під час виконання програми.

Його дизайн пропонує певну підтримку функціонального програмування в традиції Lisp. Він має функції фільтра, відображення та зменшення; розуміння списків, словники, набори та вирази генератора. Стандартна бібліотека має два модулі (itertools і functools), які реалізують функціональні інструменти, запозичені з Haskell і Standard ML.

C++ це мова програмування високого рівня загального призначення, створена датським комп'ютерним науковцем Б'ярне Страуструпом і вперше випущена в 1985 році як розширення мови програмування C або «C with Classes». Мова значно розширилася з часом, і сучасний C++ тепер має об'єктно-орієнтовані, загальні та функціональні функції на застосунок до можливостей низькорівневого маніпулювання пам'яттю. Він майже завжди реалізується як скомпільована мова, і багато постачальників надають компілятори C++, включаючи Free Software Foundation, LLVM, Microsoft, Intel, Embarcadero, Oracle і IBM, тому він доступний на багатьох платформах.

C++ було розроблено з урахуванням системного програмування та вбудованого програмного забезпечення з обмеженими ресурсами та великих систем, причому продуктивність, ефективність і гнучкість використання були основними рисами його дизайну. C++ також було визнано корисним у

багатьох інших контекстах, головними перевагами яких є інфраструктура програмного забезпечення та програми з обмеженими ресурсами, включно з настільними програмами, відеоіграми, серверами (наприклад, електронної комерції, веб-пошуку чи баз даних) і критично важливими для продуктивності програмами (наприклад, телефонні комутатори або космічні зонди).

C++ стандартизовано Міжнародною організацією стандартизації (ISO), остання версія стандарту ратифікована та опублікована ISO

Мова C++ має два основних компоненти: пряме відображення функцій апаратного забезпечення, що забезпечується в основному підмножиною C, і абстракції без накладних витрат, засновані на цих відображеннях. Страуструп описує C++ як «легку абстрактну мову програмування [створену] для побудови та використання ефективних і елегантних абстракцій»; і «пропонувати як апаратний доступ, так і абстракцію є основою C++. Ефективне виконання — це те, що відрізняє її від інших мов. »

Після огляду представлених мов програмування було прийнято рішення використовувати мову C++ при розробці програмного забезпечення для кваліфікаційної роботи.

3.4.2 Вибір середовища розробки

Середовищем розробки для кваліфікаційної роботи було обрано середовище Arduino IDE.

Arduino IDE(рис. 3.9) – це інтегроване середовище розробки програмного забезпечення (ІСРЗ) для мікроконтролерів Arduino. Це безкоштовне програмне забезпечення, яке може бути встановлено на різних операційних системах, включаючи Windows, MacOS та Linux.



Рисунок 3.9 – Інтерфейс Arduino IDE

Arduino IDE є популярним інструментом для розробки програмного забезпечення для мікроконтролерів Arduino. Це інтегроване середовище розробки (IDE), яке забезпечує зручну розробку коду, компіляцію та завантаження на мікроконтролер.

Arduino IDE проста у використанні та має зрозумілий інтерфейс користувача, що дозволяє навіть початківцям швидко розпочати програмування мікроконтролерів. Інтерфейс містить всі необхідні інструменти для програмування, від текстового редактора до панелі серійного моніторингу.

Один з головних переваг Arduino IDE полягає у тому, що воно має велику кількість бібліотек та прикладів коду, що значно полегшує розробку проектів. Крім того, Arduino IDE підтримує багато типів мікроконтролерів, що робить його досить універсальним інструментом для розробки.

Іншою перевагою Arduino IDE є те, що воно є відкритим програмним забезпеченням з відкритим кодом. Це означає, що його можна легко модифікувати та налаштувати під свої потреби. Крім того, Arduino IDE має велику спільноту користувачів, що дозволяє знайти відповіді на будь-які

запитання та ділитися досвідом з іншими програмістами. Ця спільнота активно розвивається та надає безкоштовну підтримку та оновлення програмного забезпечення.

3.4.3 Бібліотеки Arduino IDE

Інтегроване середовище розробки програмного забезпечення Arduino (IDE) дає можливість підключати до проекту додаткові бібліотеки.

Бібліотеки - це набір підготовлених до використання функцій та коду, які можна використовувати в програмі для мікроконтролера. У випадку Arduino IDE, бібліотеки дозволяють розширювати функціональність платформи Arduino та швидко виконувати складні завдання, які можуть бути важкі для програміста, який створює програму з нуля.

Бібліотеки містять готові функції, які можуть бути використані у програмах для Arduino. Наприклад, якщо ви хочете працювати з сенсором температури та вологості, можна використати бібліотеку DHT11, яка містить функції для зчитування даних з цього сенсора. Використовуючи бібліотеку, ви можете зчитати дані з сенсора за декілька рядків коду, використовуючи вже написаний код в бібліотеці. Це дозволяє вам зосередитися на розробці основної логіки програми, замість того, щоб займатися написанням коду для кожного сенсора окремо.

Використання бібліотек дозволяє зменшити час, який потрібно на розробку програми, а також зменшити кількість помилок, які можуть бути допущені при написанні коду з нуля. Бібліотеки також забезпечують більшу зручність в роботі з платформою Arduino, адже вони дозволяють використовувати готові рішення для багатьох типових задач.

У загальному, використання бібліотек у Arduino дозволяє прискорити розробку програм для мікроконтролера, забезпечити більшу точність та надійність коду, а також спростити процес програмування.

3.4.3.1 Бібліотека Time

Бібліотека `time` – це стандартна бібліотека для Arduino, яка дозволяє взаємодіяти зі складовою годинника/календаря в реальному часі (Real Time Clock, RTC). Ця бібліотека містить функції для зчитування та запису часу та дати, а також для обробки та форматування даних, пов'язаних з часом.

Основні можливості бібліотеки `time` включають:

- отримання поточного часу та дати. Бібліотека дозволяє отримати час та дату з RTC-модуля із точністю до секунди;
- налаштування часу та дати. Бібліотека дозволяє задати час та дату в RTC-модулі;
- робота з форматами часу. Бібліотека дозволяє конвертувати формати часу та дати, наприклад, перетворювати час з формату часу Unix у звичайний формат годин:хвилин:секунд;
- робота зі змінними типу часу. Бібліотека містить вбудовані типи змінних, такі як `time_t`, які дозволяють зручно працювати з часовими значеннями;
- робота з таймерами. Бібліотека дозволяє створювати таймери, які дозволяють виконувати певний код через певний інтервал часу.

3.4.3.2 Бібліотека ESP8266WiFi

Бібліотека `ESP8266WiFi` (рис. 3.10) - це стандартна бібліотека для Arduino, яка дозволяє взаємодіяти з WiFi-модулем ESP8266, який є популярним рішенням для Інтернету речей (IoT). Ця бібліотека містить функції для налаштування та управління підключенням до WiFi-мереж, а також для відправки та отримання даних через Інтернет.

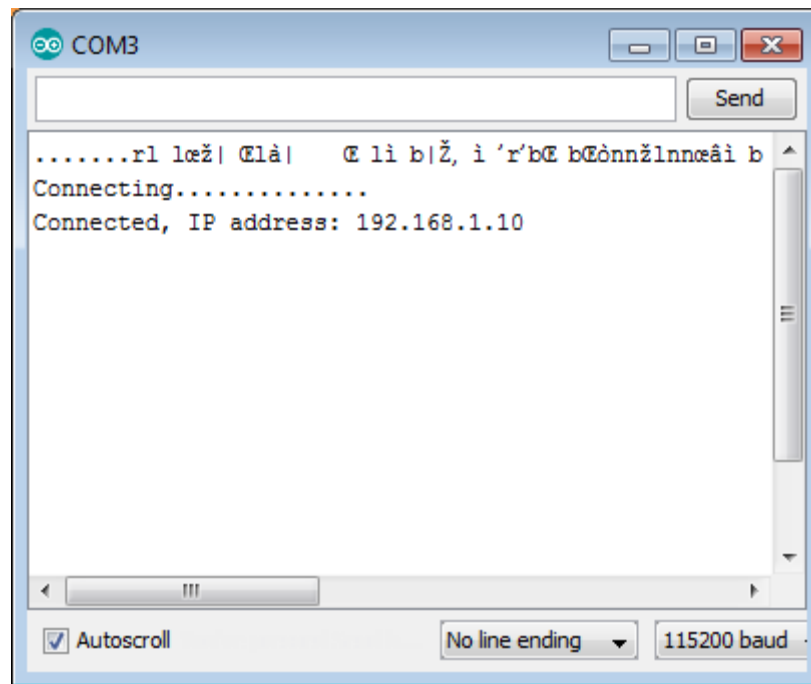


Рисунок 3.10 – Приклад роботи бібліотеки ESP8266WiFi

Основні можливості бібліотеки ESP8266WiFi включають:

- підключення до WiFi-мережі. Бібліотека дозволяє налаштувати параметри підключення до WiFi-мережі, такі як ім'я мережі та пароль.
- управління підключенням до WiFi-мережі. Бібліотека дозволяє перевіряти стан підключення до WiFi-мережі, змінювати параметри підключення та відключати з'єднання.
- отримання та відправлення даних через Інтернет. Бібліотека дозволяє відправляти та отримувати дані через Інтернет, такі як веб-сторінки, API-запити, електронні листи та інше.
- робота з серверами. Бібліотека дозволяє створювати сервери, які можуть оброблювати запити від клієнтів та відправляти відповіді.
- захист підключення. Бібліотека підтримує різні методи захисту підключення, такі як WPA, WPA2 та інші.

3.4.3.3 Бібліотека Servo

Бібліотека Servo є вбудованою бібліотекою в Arduino IDE та призначена для керування сервоприводами. Сервоприводи - це механізми, що

використовуються для точного кутового керування, зазвичай від 0 до 180 градусів. Їх використовують у різних пристроях, таких як роботи, моделі літаків, двері гаражів і багато іншого.

Ця бібліотека дозволяє легко керувати сервоприводами з Arduino. Вона містить функції для встановлення кута сервопривода, зчитування поточного кута сервопривода, налаштування мінімального і максимального кутів, і зміни частоти PWM, що використовується для керування сервоприводом.

Щоб використовувати бібліотеку Servo, необхідно підключити сервопривод до Arduino, підключивши його до відповідних виводів мікроконтролера. Після цього можна створити об'єкт класу Servo, використовуючи конструктор Servo, та налаштувати його за допомогою функцій бібліотеки Servo. Наприклад, можна встановити кут сервопривода за допомогою функції `write()`, і зчитати його поточний кут за допомогою функції `read()`.

Крім того, бібліотека Servo також має додаткові функції, які дозволяють керувати кількома сервоприводами одночасно. Наприклад, можна використовувати функцію `attach()`, щоб підключити сервопривод до певного виводу мікроконтролера, і використовувати функцію `writeMicroseconds()` для встановлення кута сервопривода за допомогою мікросекунд.

Загалом, бібліотека Servo дозволяє легко і точно керувати сервоприводами з Arduino, що робить її корисним інструментом для роботи.

3.4.3.4 Бібліотека LiquidCrystal

Бібліотека LiquidCrystal є однією з найбільш поширених бібліотек для роботи з дисплеєм символьного типу у середовищі Arduino IDE. Вона дозволяє контролювати дисплей LCD, підключений до мікроконтролера Arduino або сумісного з ним.

Завдяки бібліотеці LiquidCrystal можна виводити на дисплей текст, цифри, спеціальні символи та графічні елементи. Для цього бібліотека містить набір функцій, які дозволяють змінювати положення курсора на

екрані, записувати символи на певні координати, стирати вміст дисплею та змінювати його параметри.

Ця бібліотека підтримує різні типи дисплеїв LCD, включаючи дисплеї з різною кількістю символів у рядку та різною кількістю рядків. Для коректної роботи з дисплеєм, необхідно вказати правильні параметри конфігурації дисплею, такі як кількість символів у рядку, кількість рядків, тип зв'язку (наприклад, 4-бітний або 8-бітний), та номери пінів, на яких дисплей підключений до плати Arduino.

Отже бібліотека LiquidCrystal значно спрощує роботу з символьним дисплеєм у проектах з використанням Arduino, дозволяючи швидко та легко виводити інформацію на дисплей і контролювати його параметри.

Висновок до розділу 3

У третьому розділі було детально розглянуто, проаналізовано та вибрано апаратну частину в якості якої було вибрано ESP8266. Крім того схожий процес був проведений і для програмної частини, а саме мови програмування результатом цього процесу став вибір мови C++.

Вибрано забезпечення для написання програмного коду Arduino IDE та проаналізовано потрібні бібліотеки.

Для подальшого процесу було створено принципову та макетну схеми розроблюваного продукту.

4 ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ. АНАЛІЗ РЕЗУЛЬТАТІВ ОБЧИСЛЕНЬ

4.1 Виконання налаштувань

Для коректної роботи усієї системи перед внесенням коду потрібно провести деякі налаштування.

Перш за все потрібно налаштувати середовище програмування.

Щоб установити плату ESP8266 у Arduino IDE, потрібно виконати наступні дії:

У Arduino IDE перейти у меню «Файл» > «Параметри» (рис. 4.1).

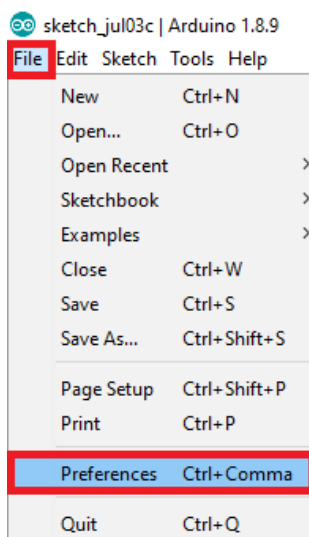


Рисунок 4.1 – Перехід у меню налаштувань.

Введіть http://arduino.esp8266.com/stable/package_esp8266com_index.json у поле «Additional Boards Manager URLs», як показано нижче (рис. 4.2). Потім натисніть кнопку «ОК»

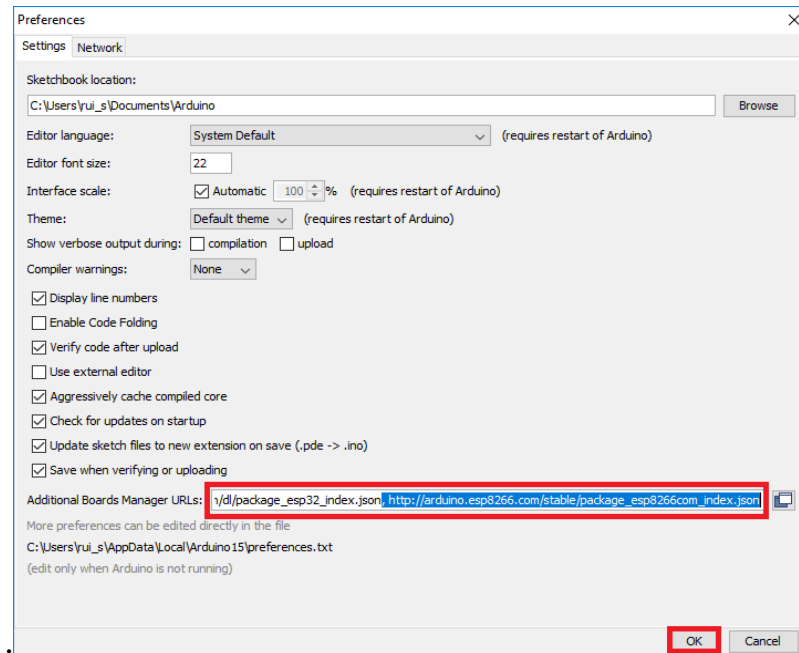


Рисунок 4.2 – Меню налаштувань

Знайдіть ESP8266 і натисніть кнопку встановлення для «ESP8266 від спільноти ESP8266», встановіть надбудову плати ESP8266 у пошуку Arduino IDE ESP8266 (рис.4.3).

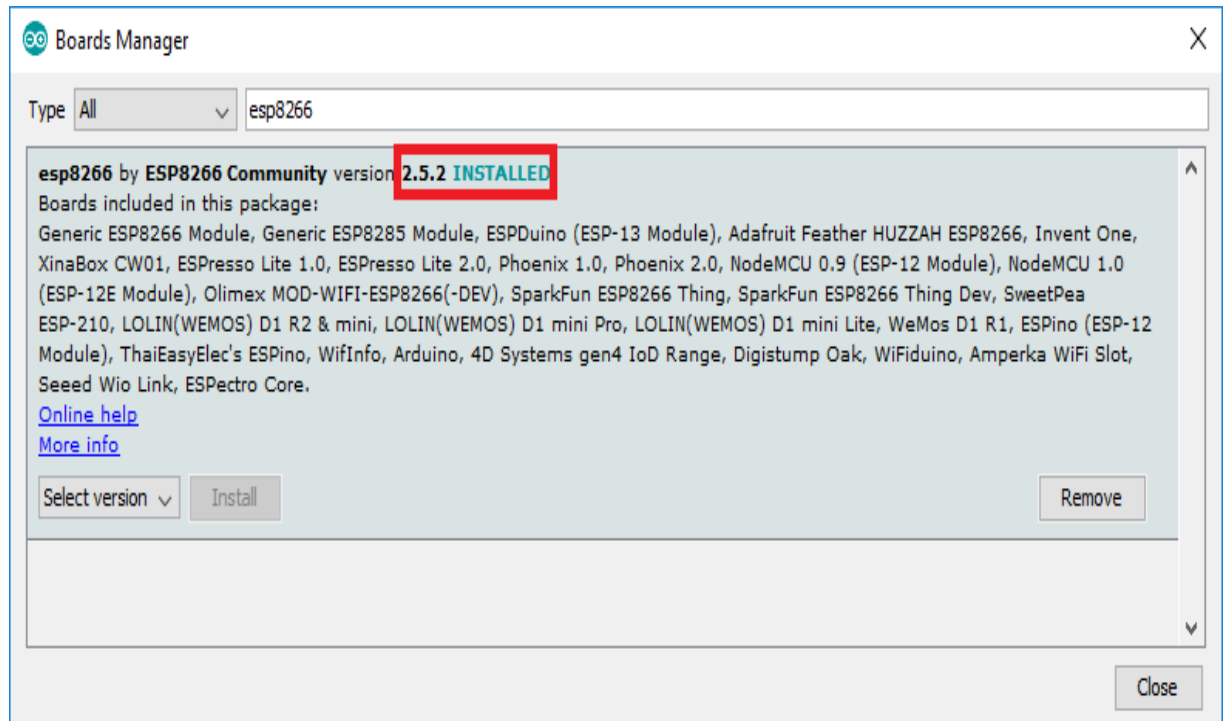


Рисунок 4.3 – Вибір потрібної плати

Крім того потрібно коректно налаштувати маршрутизацію.

4.1.1 Налаштування маршрутизатора D-Link DIR615

Щоб зайти у налаштування маршрутизатора, потрібно у веб-браузері відкрити нове вікно перейти по наступній локальній адресі: 192.168.0.1 та ввести ім'я та пароль від акаунту адміністратора (рис. 4.4).

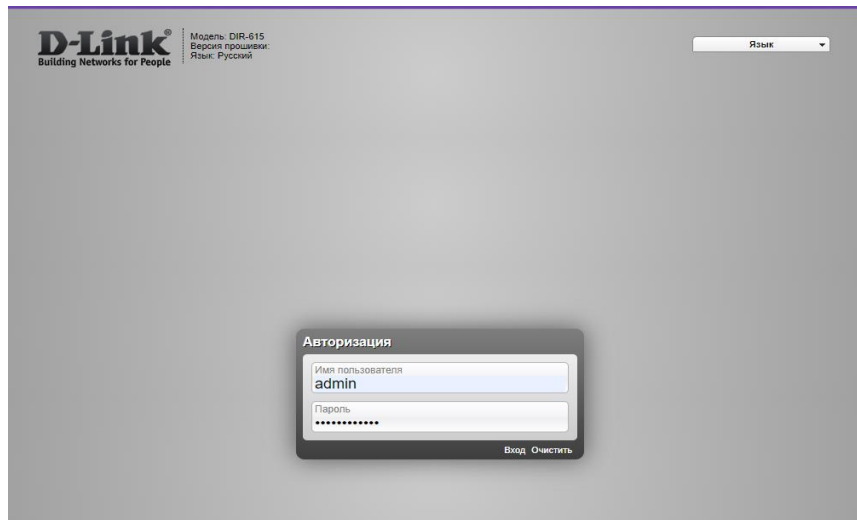


Рисунок 4.4 – Сторінка авторизації

Після успішної авторизації, потрібно перейти в меню розширених налаштувань (рис. 4.5) у вкладці «Бездротовий режим» – «Основні налаштування» можна знайти SSID, або ім'я бездротової мережі. У моєму випадку це Explorer (рис. 4.6).

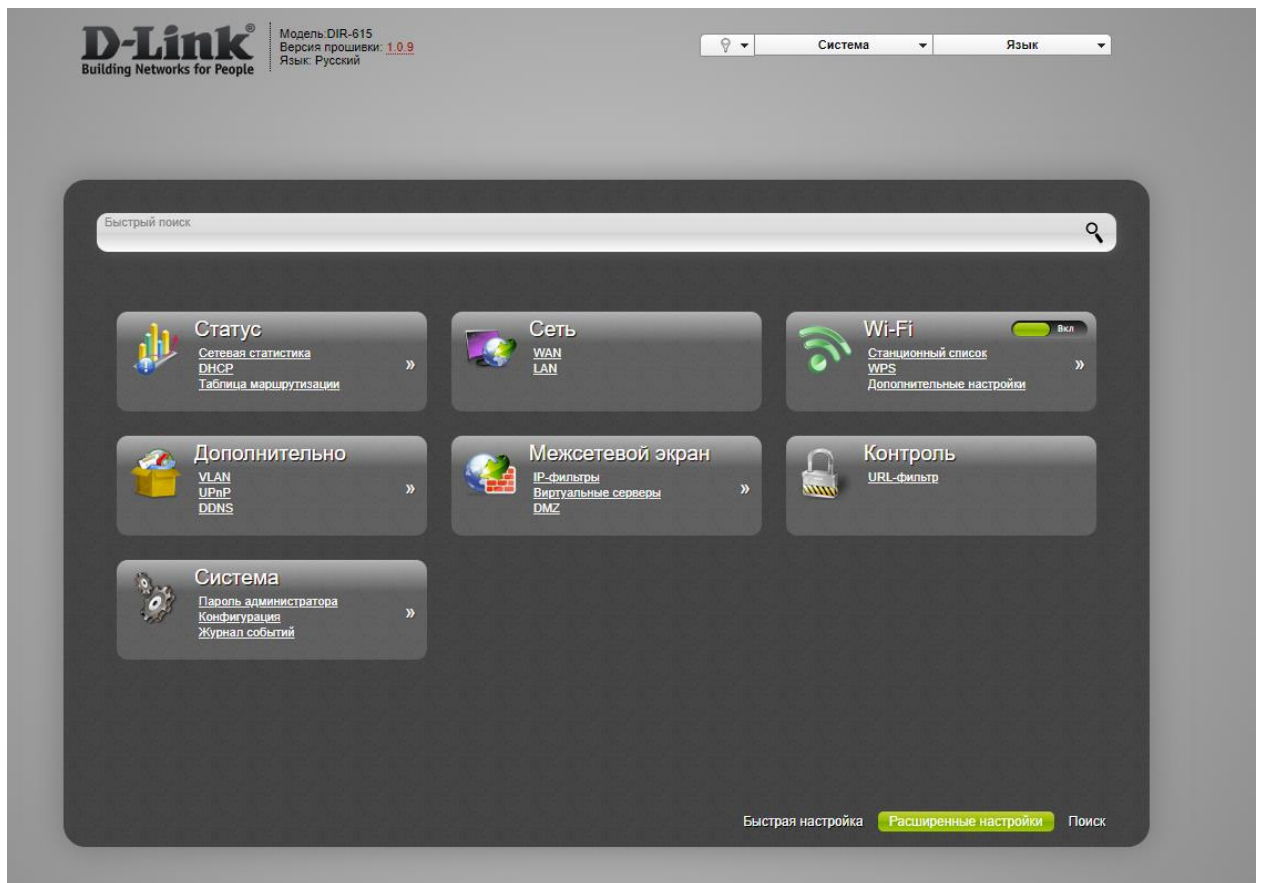


Рисунок 4.5 – Меню розширених налаштувань

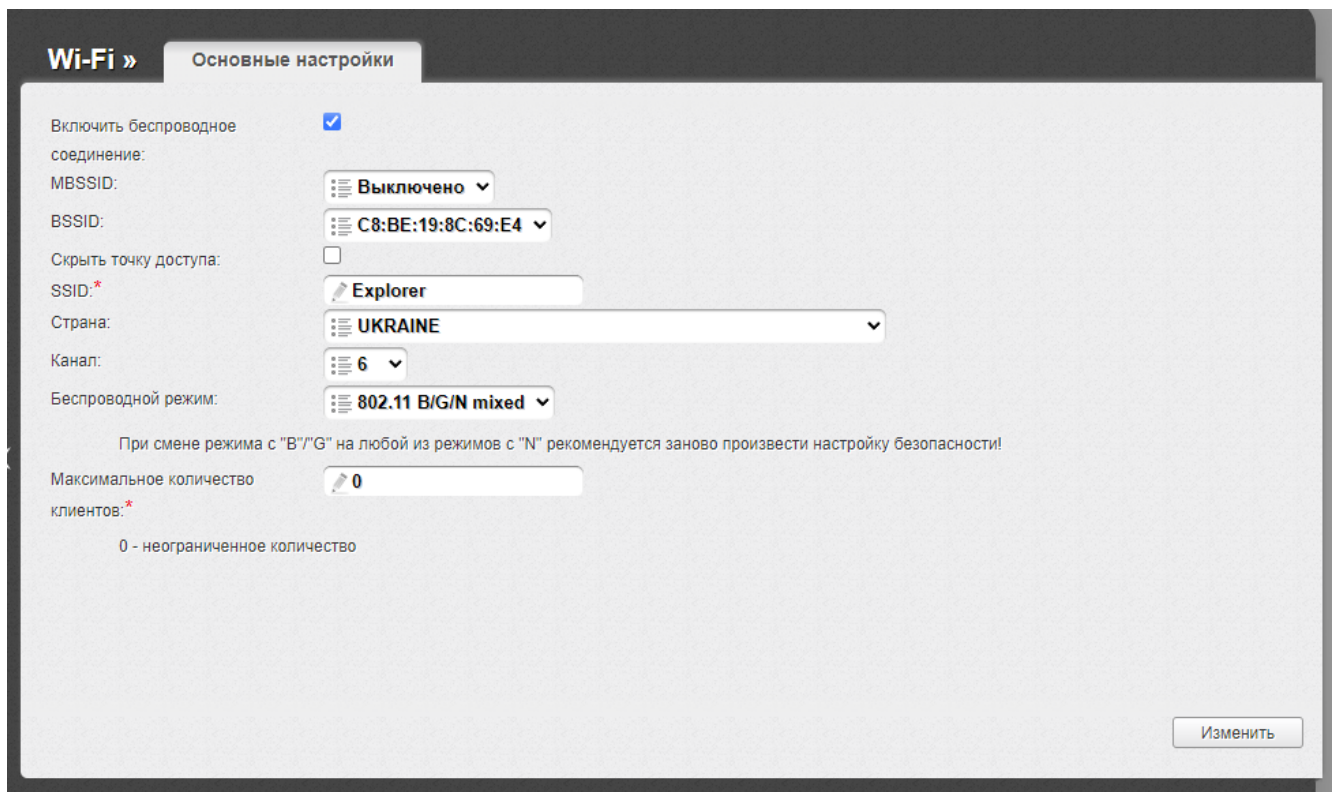


Рисунок 4.6 – Основні налаштування бездротового режиму

У вкладці «Налаштування безпеки» можна знайти пароль від мережі. У моєму випадку це M0d26cl4s0 (рис. 4.7).

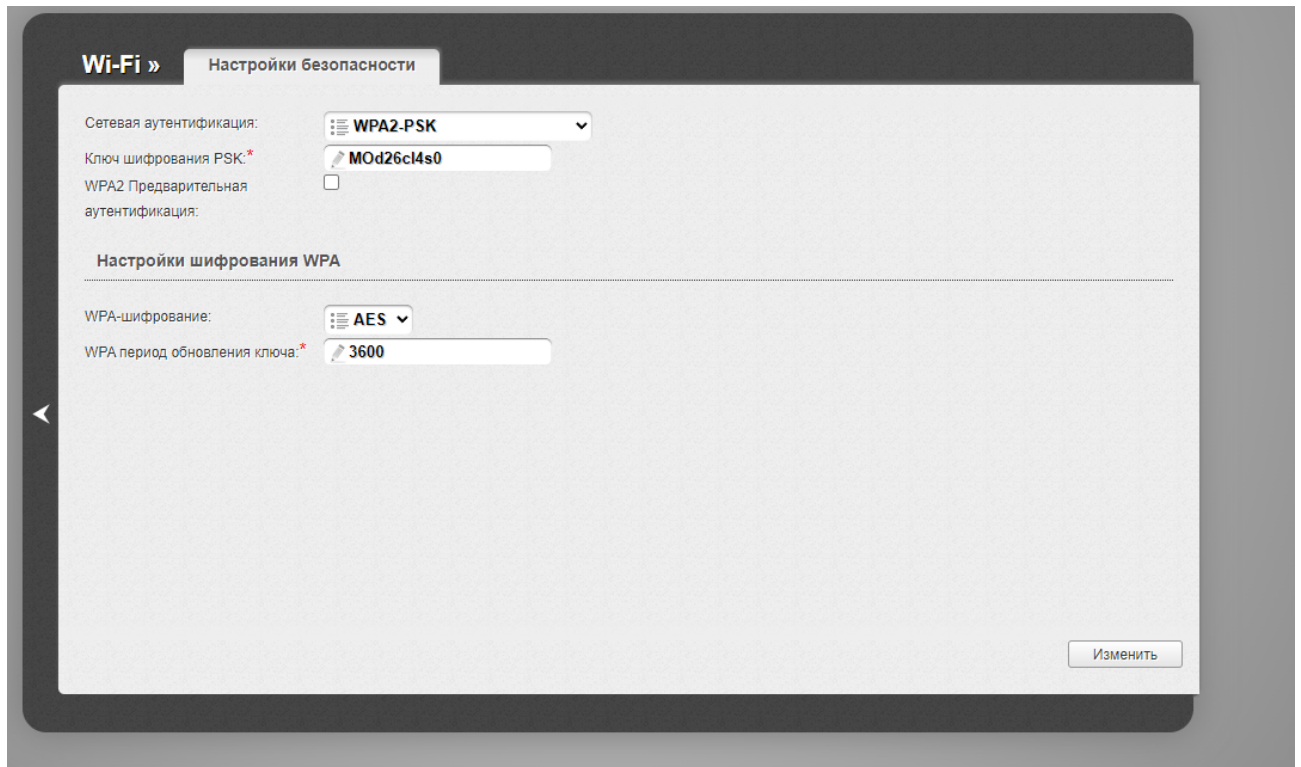


Рисунок 4.7 – Сторінка налаштувань безпеки

Після того як потрібні дані були знайдені, у програмній частині були внесені зміни до наступних рядків: ssid та password (рис. 4.8)

```
const char* ssid = "YOUR_WiFi_NAME";  
const char* password = "Wi-Fi_PASSWORD";
```

Рисунок 4.8 – Зміни у програмному коді

Після завершення налаштувань можна приступати до апаратної частини, а саме створення частин яких не вистачає. У цьому випадку потрібно змоделювати та створити частину пристрою, яка буде проштовхувати корм (рис. 4.9).

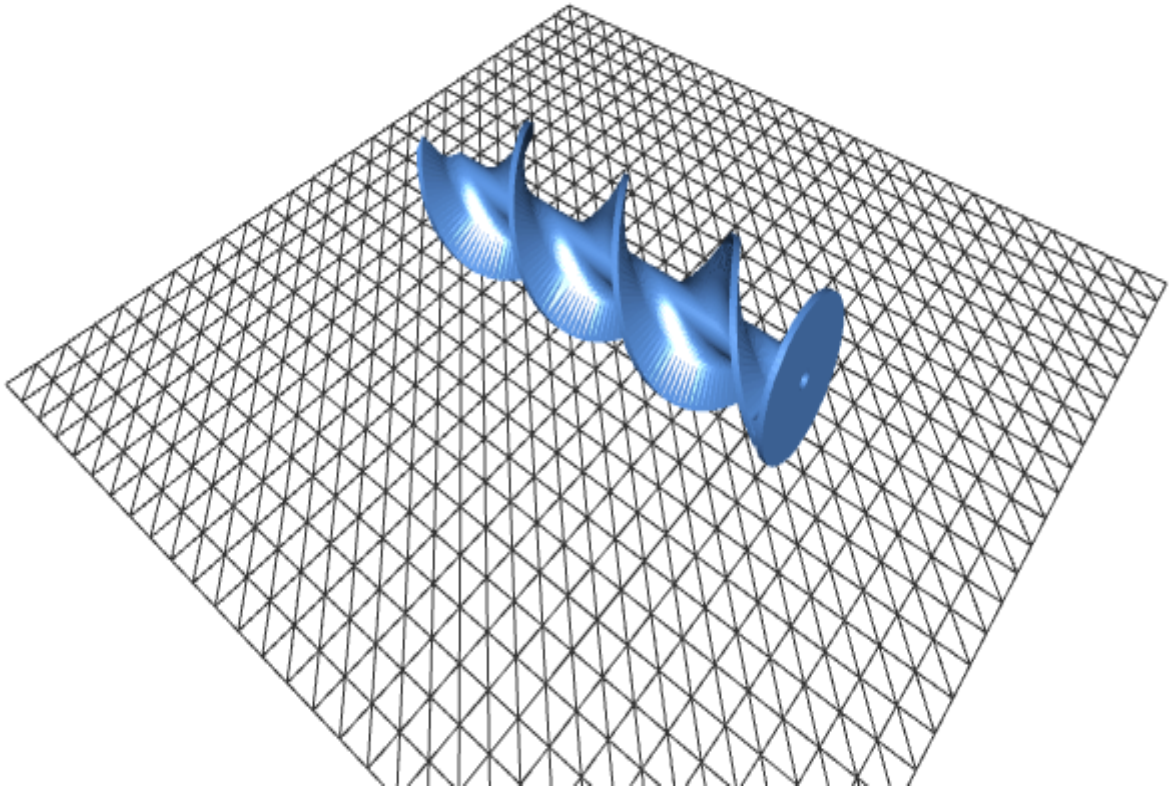


Рисунок 4.9 – 3D-модель потрібних частин

4.2 Випробування роботи програмно-апаратного комплексу

Для того, щоб почати випробування початку випробувань, треба було встановити зв'язок з модулем ESP8266 через бездротову мережу. Цей процес повинен відбутись автоматично після запуску програмного коду.

Останнім кроком є запуск і перевірка коректності роботи з мобільного пристрою (рис. 4.10).

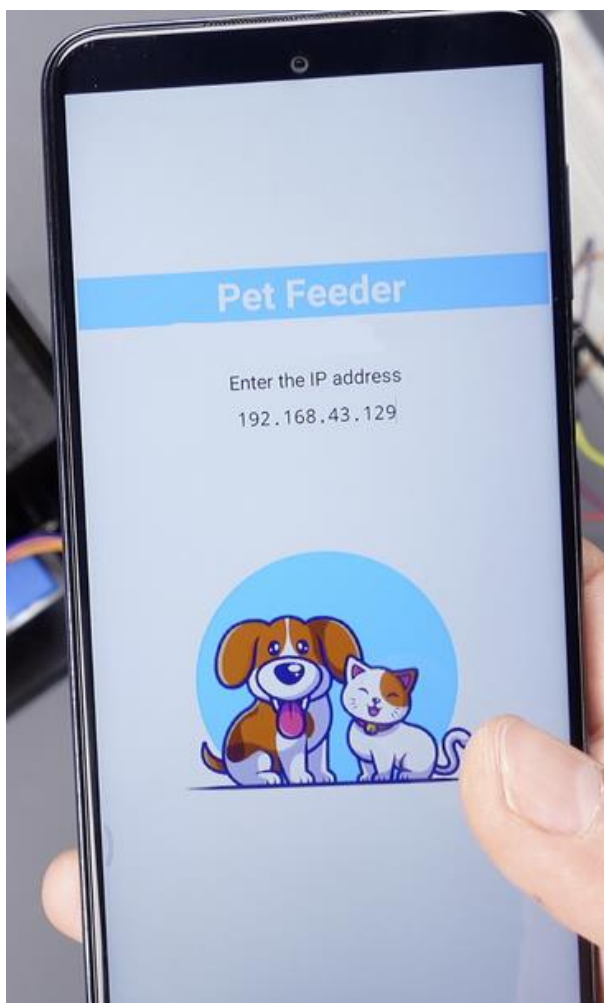


Рисунок 4.10 – Інтерфейс застосунку

Крім того було розраховано приблизний строк дії пристрою при роботі від сервоприводу.

Висновок до розділу 4

Було проведено налаштування бездротової мережі та маршрутизатора. Відредаговано програмний код для коректної роботи.

Змодельовано та створено потрібні частини пристрою та перевірено працездатність пристрою.

ВИСНОВКИ

У ході виконання кваліфікаційної роботи було проаналізовано існуючі аналоги автоматичної годівниці для тварин та виявлено що їх недоліком являється висока ціна.

Під час підготовки до реалізації було проаналізовано мови програмування, середовища розробки та можливі варіанти апаратної частини. Розроблено апаратний комплекс на базі ESP8266 та сервоприводу.

Результатом виконання став пристрій, який здатний кормити домашнього улюбленця без додаткового втручання зі сторони користувача.

Недоліком являється обмежений заряд акумулятора що потребує підзарядки час від часу, у подальшому для вирішення цього недоліка є модернізація пристрою для роботи від мережі.

Результати виконання роботи можуть бути рекомендовані для використання у повсякденному житті для годування домашнього улюбленця .

В процесі виконання також було виконано частину з охорони праці. Були проаналізовані умови в приміщенні підприємства в якому проводились дослідження. Було запропоновано доцільні заходи щодо його покращення.

Практичне значення отриманих результатів: розроблений програмно-апаратний комплекс може бути застосований, як більш дешевий, але не менш функціональний аналог дорогих аналогів представлених на ринку в даний момент в умовах зменшеного енергоспоживання відносно інших пристроїв, що використовуються для тих же цілей.

Робота пройшла апробацію на всеукраїнській науково-практичній конференції молодих вчених, аспірантів і студентів «Інформаційні технології та інженерія» [4].

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Богдан О. О. Програмування мікроконтролерів на мові C++ з використанням Arduino IDE. Київ : Видавництво "Підручники і посібники", 2019.
2. Дмитрієв В. І. Основи програмування мікроконтролерів ESP8266 з використанням мови C++. Одеса : Видавництво "Одеський національний університет імені І. І. Мечникова", 2018.
3. Жуковський В. О. Розробка програмного забезпечення з використанням мови C++ для мікроконтролерів ESP8266. Львів : Видавництво Львівської політехніки, 2020.
4. Чумаченко Д. О., Савінов В. Ю. ESP 8266 як платформа для проєктів IoT. *Інформаційні технології та інженерія: тези доп. Всеукр. наук.-практ. конф. Миколаїв, 7–10 лют. 2023 р. Миколаїв : Чорном. нац. ун-т ім. Петра Могили, 2023. С.88–89.*
5. Witzel-Rollins A. , Murphy M. Springer Cary M., Tamberlyn D Moyers Evaluation of a pet-separating automatic feeder and high-frequency meal feeding for weight loss in multi-cat households Publ: 28/06/22 The Journal of Feline Medicine and Surgery DOI: 10.1177/1098612X221105046
6. Новак І. І. Комп'ютерна інженерія : навчальний посібник. Київ : Видавництво "Київський університет", 2021.
7. Синявський В. І. Програмування мікроконтролерів ESP8266 з використанням мови C++. Київ : Видавництво "Нова книга", 2019.
8. Ткаченко Л. М. Розробка програмного забезпечення з використанням мови C++ для мікроконтролерів ESP8266. Харків : Видавництво "Харківський національний університет радіоелектроніки", 2020.
9. Banzi, M., & Shiloh, M. (2014). Getting started with Arduino: The open source electronics prototyping platform. Maker Media, Inc.

10. Dorhea, S. (2016). ESP8266 Arduino IDE Guide: Internet of Things with ESP8266(NodeMCU). Amazon Digital Services LLC.
11. Khan, R. A. (2018). IoT with ESP8266: Build exciting internet of things projects by interfacing ESP8266 with your smartphone. Packt Publishing Ltd.
12. Liu, X. (2019). ESP8266 Development Using Arduino IDE. Apress.
13. Marzetta, K. (2018). IoT Automation: Arrow Electronics IoT Internet of Things IoT Arduino ESP8266 ESP32. Amazon Digital Services LLC.
14. Thorén, D. (2016). ESP8266 Home Automation Projects: Leverage the power of this tiny WiFi chip to build exciting smart home projects. Packt Publishing Ltd.
15. Gupta, B., & Singh, R. (2019). IoT and Arduino: A beginner's guide to learning about the Internet of Things. Apress.
16. Lewis, R. (2019). IoT with Arduino: Build Internet of Things with Arduino, and explore practical use cases and examples. Packt Publishing Ltd.
17. Madhumitha, M., & Kaviya, M. (2020). IoT with ESP8266: Interfacing of ESP8266 with Arduino Uno using IOT application. International Journal of Engineering Research and Technology, 9(4), 561-567.
18. Malik, M. A., & Singh, N. P. (2018). ESP8266 based smart irrigation system using IOT. International Journal of Pure and Applied Mathematics, 119(12), 6095-6107.
19. Sharma, S., & Goyal, D. (2019). IoT based smart home automation using ESP8266. International Journal of Engineering Research and Technology, 12(2), 156-163.
20. Upadhyay, D., & Tiwari, S. (2019). ESP8266 based home automation using IOT. International Journal of Engineering Research and Technology, 12(5), 650-654.
21. Banerjee, T. (2019). Internet of Things with Python: Build exciting IoT projects with the Raspberry Pi and Python. Packt Publishing Ltd.

22. Bass, M. (2019). *Beginning Arduino: A comprehensive guide to getting started with Arduino, including Arduino basics, Arduino hardware and software, and Arduino projects.* Apress.

23. Esposito, M. (2018). *MicroPython Cookbook: Over 110 practical recipes for programming embedded systems and microcontrollers with Python.* Packt Publishing Ltd.

24. Martino, R. (2019). *Python Programming for Arduino: Learn how to program and use Arduino boards with Python by exploring different IoT projects.* Packt Publishing Ltd.

25. Monk, S. (2019). *Programming Arduino Next Steps: Going Further with Sketches.* McGraw-Hill Education.

26. Ojha, S., & Kaushik, A. (2021). *IoT with ESP8266 using Arduino IDE: Getting Started with Arduino IoT Projects.* Wiley.

27. Rosebrock, A. (2019). *Deep Learning for Computer Vision with Python.* PyImageSearch.

28. Simon, M., & Wachowiak, R. (2019). *Python Machine Learning for IoT: Build advanced machine learning models using Python and IoT devices.* Packt Publishing Ltd.

29. Yalamanchi, S. (2020). *IoT using Python on Microcontrollers: Exploring CircuitPython, MicroPython and Zerynth.* Packt Publishing Ltd.

30. Zambon, G. (2020). *Python for Arduino: A Hands-on Approach.* Springer.

ДОДАТОК А КОД ПРОГРАМИ

```
#include <ESP8266WiFi.h>

#define IN1a 15
#define IN2a 13
#define IN3a 12
#define IN4a 14

int delayTime = 2;

WiFiClient client;
WiFiServer server(80);

const char* ssid = "YOUR_WiFi_NAME";
const char* password = "WiFi_PASSWORD";

String data = "";

void setup() {
  pinMode(IN1a, OUTPUT);
  pinMode(IN2a, OUTPUT);
  pinMode(IN3a, OUTPUT);
  pinMode(IN4a, OUTPUT);

  Serial.begin(115200);
  connectWiFi();
  server.begin();
}

void loop() {
  client = server.available();
  if (!client) return;
  data = checkClient ();

  if (data == "feed") {
    for (int adim = 0; adim < 200; adim++) {
      Serial.println("FORWARD");
      forwardMotor();
    }

    delay(200);
  }
}
```

```
    for (int adim = 0; adim < 200; adim++) {
        Serial.println("BACKWARD");
        backwardMotor();
    }
    stopMotor();
}
}

void connectWiFi()
{
    Serial.println("Connecting to WIFI");
    WiFi.begin(ssid, password);
    while (!(WiFi.status() == WL_CONNECTED))
    {
        delay(300);
        Serial.print("..");
    }
    Serial.println("");
    Serial.println("WiFi connected");
    Serial.println("Wemos Local IP is : ");
    Serial.print((WiFi.localIP()));
    Serial.print("");
    Serial.println("");
}

String checkClient (void)
{
    while (!client.available()) delay(1);
    String request = client.readStringUntil('\r');
    request.remove(0, 5);
    request.remove(request.length() - 9, 9);
    return request;
}

void forwardMotor(void) {
    digitalWrite(IN4a, HIGH);
    digitalWrite(IN3a, LOW);
    digitalWrite(IN2a, LOW);
    digitalWrite(IN1a, LOW);
    delay(delayTime);
    digitalWrite(IN4a, LOW);
    digitalWrite(IN3a, HIGH);
    digitalWrite(IN2a, LOW);
    digitalWrite(IN1a, LOW);
    delay(delayTime);
}
```

```
digitalWrite(IN4a, LOW);
digitalWrite(IN3a, LOW);
digitalWrite(IN2a, HIGH);
digitalWrite(IN1a, LOW);
delay(delayTime);
digitalWrite(IN4a, LOW);
digitalWrite(IN3a, LOW);
digitalWrite(IN2a, LOW);
digitalWrite(IN1a, HIGH);
delay(delayTime);
}

void backwardMotor(void) {
digitalWrite(IN4a, LOW);
digitalWrite(IN3a, LOW);
digitalWrite(IN2a, LOW);
digitalWrite(IN1a, HIGH);
delay(delayTime);
digitalWrite(IN4a, LOW);
digitalWrite(IN3a, LOW);
digitalWrite(IN2a, HIGH);
digitalWrite(IN1a, LOW);
delay(delayTime);
digitalWrite(IN4a, LOW);
digitalWrite(IN3a, HIGH);
digitalWrite(IN2a, LOW);
digitalWrite(IN1a, LOW);
delay(delayTime);
digitalWrite(IN4a, HIGH);
digitalWrite(IN3a, LOW);
digitalWrite(IN2a, LOW);
digitalWrite(IN1a, LOW);
delay(delayTime);
}

void stopMotor(void) {
digitalWrite(IN4a, LOW);
digitalWrite(IN3a, LOW);
digitalWrite(IN2a, LOW);
digitalWrite(IN1a, LOW);
}
```

ДОДАТОК Б

МАТЕРІАЛИ АПРОБАЦІЇ

Міністерство освіти і науки України
 Чорноморський національний університет
 імені Петра Могили



«Інформаційні технології та інженерія»

*Всеукраїнська науково-практична конференція
 молодих вчених, аспірантів і студентів*

ТЕЗИ

7–10 лютого 2023 року

<i>Джаман С. В., Крайник Я. М.</i> Застосування алгоритму QO1 у стисненні зображень.....	70
<i>Коваленко І. О., Шаріпова І. В., Левченко А. О., Стукалов С. А.</i> Генерація тестових даних за допомогою мікроконтролерів задля тестування системи «розумний будинок».....	72
<i>Кравцов К. В., Пузіров С. В.</i> Кластерна система доставки контенту (CDN) на базі Raspberry Pi.....	74
<i>Ахундула В. І., Савінов В. Ю.</i> Апаратно-програмний комплекс розпізнавання пожежі на базі Arduino та сповіщення користувача через мобільний застосунок.....	76
<i>Кушнір С. Ю., Пузіров С. В.</i> Кластерна серверна система на базі Raspberry Pi та Terraform.....	78
<i>Рибченко С. С., Крайник Я. М.</i> Застосування MQTT в IoT системах.....	79
<i>Савенко Б. О.</i> Модель архітектури частково розподілених систем та їх компонентів в комп'ютерних мережах.....	81
<i>Саломовський Б. Г., Могіла В. Р.</i> Система збору та аналізу інформації про радіаційний фон.....	83
<i>Серета М. О., Пузіров С. В.</i> Побудова сенсорної IoT-мережі для моніторингу якості повітря.....	85
<i>Старченко В. В., Шкрянда В. О.</i> Система моніторингу територіальної активності.....	86
<i>Чумаченко Д. О., Савінов В. Ю.</i> ESP 8266 як платформа для проєктів IoT.....	88
Методи і засоби програмної інженерії	
<i>Афонін Ю. С., Журавська І. М.</i> Мобільний застосунок для моніторингу здоров'я у вигляді казуальної гри на платформі Unity.....	90
<i>Безруб С. С., Бойко А. П.</i> Програмне забезпечення симуляції біополувації на основі нейронної мережі та ігрового рушія.....	92
<i>Губарев М. О., Бойко А. П.</i> Навігація транспортних засобів на основі аналізу трафіку з використанням комп'ютерного зору.....	94
<i>Кірей К. О., Горбаль Г. В.</i> Вдосконалення системи контролю, обліку та доступу в інтернет абонентів оператора телекомунікацій.....	97
<i>Коваль О. В., Гаєришко Є. В., Лобода П. П., Старовій І. С.</i> Структура бази даних та знаєць цифрового двійника нового безпечного конфайнменту ЧАЕС.....	100

Миколай – 2023

У результаті попередніх випробувань протоколу системи була досягнута точність локалізації місця пострілу біля 10 м при розміщенні сенсорів на відкритій місцевості більш ніж на 800 м.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. 2022: Global Terrorism Index. URL: <https://ladamlaw.com/medium.com/global-terrorism-index-for-2022-sitemap-ce12ac2ff5c9> (Last accessed: 17.01.2023).
2. Антитерористичний центр при СБУ. URL: <https://ssu.gov.ua/baluyetogustyslamuy-tsemp-ryu-ssu> (дата звернення: 17.01.2023).
3. Almeida E., Bose P., Sano P., Silveira R. Flips in higher order Delaunay triangulations. *Lecture Notes in Computer Science Book Series (LNCS)*. 2020. Vol. 12118. P. 223–234. DOI: 10.1007/978-3-030-61792-9_18.

УДК 004.057.4

*Чумаченко Д. О., Савинов В. Ю.
 Черноморський національний університет ім. Петра Могили,
 Миколаїв, Україна*

ESP 8266 ЯК ПЛАТФОРМА ДЛЯ ПРОЄКТІВ ІОТ

Модуль ESP8266 є однією із найвруччіших платформ для створення найрізноманітніших проєктів у сфері ІоТ.

ESP8266 – це мікроконтролер, розроблений у 2014 році і що випускається компанією Espressif Systems – китайською компанією з Шанхаю. Він являє собою мережеве рішення з Wi-Fi-трансивром на борту плюс можливість виконання застосунків, що записуються в його пам'ять. Існує безліч модифікацій плат, які називаються зазвичай від ESP-01 до ESP-12. Сьогодні вже з'явилися ще інші назви плат від портативних розробників. Відмінності в платах полягають в основному в потужності введення-виведення, кількості флеш-пам'яті, виду конекторів тощо. Процесор - той самий, отже з погляду програмування немає значення яку плату програмувати.

- Специфікація ESP8266:
- Напруга живлення: 3.3 В
 - Енергоспоживання: 10 мкА ... 170 мА
 - Флеш-пам'ять: до 16 мб максимум (зазвичай 512 кб)
 - Процесор: Tensilica L106, 32 біта
 - Швидкість процесора: 80...160 МГц

- ОЗП: 32 кб + 80 кб
- Порти введення-виведення загального призначення: 17 (мультиплексовані з іншими функціями)

- АЦП: 1 введення з роздільною здатністю 1024
 - Підтримка 802.11: b/g/n/d/s/b/t/r
 - Максимальна кількість підключень TCP: 5
- Зі специфікації видно, як довго працюватиме живлення ESP8266 від батарейок не може бути легко визначено. Споживання енергії змінюється в широкому діапазоні – при передачі на повній потужності воно становить 170 міліампер, а в режимі сну - всього 10 мікроампер.

Хоча модуль має різні моделі і збірняк, у своїй суті його розпізнавка залишається майже однаковою (рис.1)

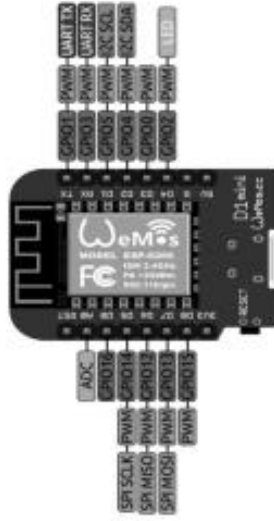


Рисунок 1 – Опис та розташування пінів

Типове застосування ESP8266 як апаратної основи Internet of Things найчастіше передбачає встановлення у будинках чи офісах. При цьому мережеве підключення здійснюється до домашньої/офісної локальної мережі з виходом в інтернет через роутер. Користувач пристрою може контролювати його за допомогою планшета або комп'ютера через свою локальну мережу або віддалено через Інтернет.

Програми для модуля ESP8266 пишуться зазвичай мовою С. Перед завантаженням програми в модуль її слід скомпілювати з тексту в машинні коди. Редагувати текст програми найвруччіше в якомусь редакторі, який має підтримку синтаксису, вбудовану довідку та інші корисні функції, і ще краще - в інтегрованому середовищі розробки (Integrated Development Environment). Працюючи в такому середовищі, ви можете написати текст програми, скомпілювати її і

відразу ж завантажити в модуль. Широко відомі такі середовища розробки як Espressif та Arduino IDE.

Отже легкість у використанні та невелика ціна (до 3\$) робить його одним із кращих компонентів при створенні будь-якого проекту IoT.

Методи і засоби програмної інженерії

УДК 004.42

Афонін Ю. С., Журавська І. М.

*Черноморський національний університет ім. Петра Могили,
Миколаїв, Україна*

МОБІЛЬНИЙ ЗАСТОСУНОК ДЛЯ МОНИТОРИНГУ ЗДОРОВ'Я У ВИГЛЯДІ КАЗУАЛЬНОЇ ГРИ НА ПЛАТФОРМІ UNITY

Підтримання і моніторинг здоров'я безумовно повинні бути першими пріоритетами людини, однак через зайнятість, роботу, стрес, інші чинники більшість із нас пріоритизують йому надто мало часу. Пропонуємо розглянути який чинник інтерактивний мобільний застосунок-гра може моніторити показники здоров'я та мотивувати користувачів до здійснення фізичної активності.

За мету було поставлено проаналізувати, які рішення у галузі програмного забезпечення використовувались та використовуються сьогодні, та яким чином мобільні ігри можуть спонукати людей до фізичної активності.

В недалекому минулому питання підтримання форми та моніторингу здоров'я знайшло своє вирішення у масовому ринковому попиті на фітнес-браслети та розумні годинники, які, містячи в собі певні датчики активності (пульсометри, пульсоксиметри, акселометри тощо), почали мотивувати людей до фізичної активності [1]. Це стало поштовхом до створення спеціалізованого мобільного програмного забезпечення (Google Fit, Zepp Life, Samsung Health тощо) від компаній-виробників таких пристроїв. Дане програмне забезпечення пропонує користувачу аналітику основних показників (пульс, тиск, рівень стресу, сон тощо) та програми тренувань.

З розвитком потреб сьогодні у Google Play, Apple Store та інших крамничках з'явилося безліч спортивних та фітнес застосунків призначених як для спортсменів, так і звичайних людей, які прагнуть приділяти власному здоров'ю більше уваги.