

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра обчислювальної техніки

БАКАЛАВРСЬКА ДИПЛОМНА РОБОТА


на тему:

SMART — система управління термоустановкою на базі модуля ESP 8266

ПОЯСНЮВАЛЬНА ЗАПИСКА

08—23.БДР.007.00.000 ПЗ

Виконав студент 2 курсу, групи ІКІ—20мсз
спеціальності 123 — Комп'ютерна інженерія

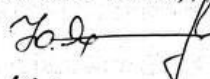
 Мазур В.Ю.

Керівник роботи к.т.н., проф. каф. ОТ

 Томчук М.А.

" 13 " червня 2022 р.

Опонент к.т.н., доц. каф. ЗІ

 Яремчук Ю. Є.

" 16 " червня 2022 р.

Допущено до захисту
д.т.н., проф. Азаров О.Д.

" 17 " червня 2022 р.

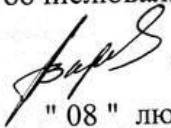


ВІННИЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра обчислювальної техніки
Освітній рівень — бакалавр
Спеціальність — 123 Комп'ютерна інженерія

ЗАТВЕРДЖУЮ

Завідувач кафедри обчислювальної техніки

 О.Д. Азаров
" 08 " лютого 2022 р.

ЗАВДАННЯ НА ДИПЛОМНУ РОБОТУ СТУДЕНТУ

Мазуру Володимиру Юрійовичу

1 Тема роботи «SMART — система управління термоустановкою на базі модуля ESP 8266».

Керівник роботи Томчук Микола Антонович , к.т.н., доц., затверджені наказом вищого навчального закладу від "09" березня 2022року №66

2 Строк подання студентом роботи 14.06.22.

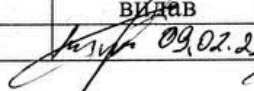

3 Вихідні дані до роботи: технічний опис програмного застосунку, мова програмування C++, Android застосунок, середовище розробки Arduino IDE.

4 Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити):вступ, аналіз сучасних методів розробки,варіантний вибір засобів для розробки програмного забезпечення, програмна реалізація, тестування розробленого програмного забезпечення.

5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень):блок-схема алгоритму роботи застосунку, функціональна електрична схема апаратної частини установки.

6 Консультанти розділів роботи наведені в таблиці 1.

Таблиця 1 — Консультанти роботи

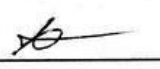
Розділ	Прізвище, ініціали та посада Консультанта	Підпис, дата	
		завдання видав	завданн прийма
1-3	Томчук М.А., к.т.н., доц.		09.02.22 

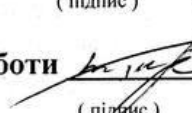
7 Дата видачі завдання 09.02.22.

8 Календарний план наведено в таблиці 2.

Таблиця 2 — Календарний план

№ з/п	Назва етапів дипломної Роботи	Строк виконання етапів проекту (роботи)	Прим ка
1	Постановка задачі роботи	09.02.22	Ви
2	Пошук матеріалів по технологіям розробки Andriod IoT застосунків	13.02.22 — 25.02.22	Ви
3	Структурне проектування установки	27.02.22 — 10.03.22	Ви
4	Обґрунтування та вибір засобів реалізації системи	29.03.22 — 04.04.22	Ви
5	Розробка Andriod застосунку	05.04.22 — 29.04.22	Ви
6	Підготовка матеріалів тестування та розробка інструкції користувача	30.04.22 — 27.05.22	Ви
7	Оформлення пояснювальної записки та ілюстративного матеріалу	28.05.212 — 06.06.22	Ви
8	Перевірка якості виконання бакалаврської роботи та усунення недоліків	07.06.22	Ви

Студент  Мазур В.Ю.
(підпис) (прізвище та ініціали)

Керівник роботи  Томчук М.А.
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Бакалаврська дипломна робота: 88 с., 2 табл., 40 рис., 21 джерел.

Проаналізовано існуючі у світі ІТ-рішення для реалізації завдання на проект, та вибрано оптимальні з них за критеріями ціна/ефективність. Розроблено нову інформаційну модель інформаційної системи керування термоустановкою з використанням сучасних ІоТ-технологій, яка, на відміну від існуючих, дозволяє автоматизувати та віддалено керувати мобільною термоустановкою. Розроблено систему віддаленого керування. Протестовано працездатність та доведена її ефективність.

Ключові слова: термоустановка, arduino ide, iot система, c++, android застосунок, смартфон, холодильна камера, датчик температури.

ABSTRACT

Bachelor's thesis: 88 pages, 2 tables, 40 figures, 21 sources.

The existing IT solutions in the world for the implementation of the project task are analyzed, and the optimal ones are selected according to the criteria of price / efficiency. A new information model of the information system of thermal installation control has been developed with the use of modern IOT technologies, which, unlike the existing ones, allows to automate and remotely control the mobile thermal installation. A remote control system has been developed. Performance tested and its effectiveness proved.

Keywords: heat installation, arduino ide, iot system, c ++, android valve, smartphone, refrigerator, temperature sensor.

ЗМІСТ

ВСТУП	8
1 ОГЛЯД СУЧАСНИХ МОБІЛЬНИХ ХОЛОДИЛЬНИХ СИСТЕМ ТА ЗАСОБІВ РОЗРОБКИ ІОТ СИСТЕМ	10
1.1 Аналіз існуючих холодильних установок	10
1.2 Поняття інтернету речей та його технічна складова	15
1.3 Аналіз сучасних сервісів розробки ІоТ систем	18
1.4 Аналітичний огляд обладнання для створення ІоТ систем	26
1.5 Обґрунтування вибору мови програмування та середовища	34
2 ПРОЕКТУВАННЯ АПАРАТНОЇ ЧАСТИНИ	38
2.1 Розробка структурної схеми апаратної системи	38
2.2 Визначення моделей компонентів системи	39
2.3 Макетування системи	47
3 ПРОЕКТУВАННЯ ПРОГРАМНОЇ ЧАСТИНИ	49
3.1 Розробка основного алгоритму роботи.	49
3.2 Проектування роботи сервера	51
3.3 Програмування мікроконтролера	54
3.4 Тестування програмного застосунку	60
3.5 Розробка інструкції користувача	69
ВИСНОВКИ	70
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	71
ДОДАТОК А Технічне завдання	73
ДОДАТОК Б Лістинг програми застосунку	76
ДОДАТОК В Протокол перевірки кваліфікаційної роботи на наявність текстових запозичень	88

					<i>08-23.БДР.007.00.000 ПЗ</i>			
<i>Змн.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	SMART — система управління термоустановкою на базі модуля ESP8266	<i>Лім.</i>	<i>Арк.</i>	<i>Аргументів</i>
<i>Розроб.</i>		<i>Мазур В.Ю</i>						
<i>Перевір.</i>		<i>Томчук М.А</i>					7	88
<i>Реценз.</i>		<i>Яремчук Є.Ю</i>				ВНТУ, гр. 1КІ-20мсз		
<i>Н. Контр.</i>		<i>Швець С.І</i>						
<i>Затверд.</i>		<i>Азаров О.Д</i>						

ВСТУП

Сучасні технології і техніка розвиваються дуже високими темпами, завдяки чому в нашому повсякденному житті користуємось такими пристроями, про які раніше могли тільки уявляти і мріяти. Наприклад, поява мобільної телефонії, яка прийшла на зміну дротового зв'язку, або безпілотних автомобілів, які вже активно використовуються в різних країнах Європи і можуть проїхати велику відстань без участі людини. Останні напрямки розвитку суспільства свідчать про те, що воно є інформаційним. Інформаційне суспільство є наступним кроком розвитку людства, коли інформація та знання визначають його стан.

Розвиток інтелектуальних мехатронних модулів обумовлений появою недорогих мікропроцесорів і контролерів на їх базі і спрямований на інтелектуалізацію процесів, і в першу чергу — процесів управління функціональними перетвореннями і роботою машин та агрегатів. В даний час існує декілька платформ для управління фізичними процесами стосовно мехатронних модулів. Однак реалізацію апаратної платформи необхідно здійснювати на базі перспективних структурних рішень, що відкриті для розвитку і мають ієрархічну структуру, таких, як платформа Arduino.

Тому платформа Arduino стає основним елементом для рішення задач в галузі мехатроніки.

IoT системи набирають все більшої популярності серед користувачів. Зараз дуже важко уявити сучасні будинки і підприємства, в яких прокладені десятки кілометрів проводів для зв'язку між собою. Зараз цей напрямок стрімко розвивається так як стикається з великою кількістю труднощів: тип і засіб зв'язку, організація бездротових сенсорних мереж.

З цього випливає що вивчення в даному напрямку є актуальним і представляє інтерес для користувачів і розробників подібних пристроїв.

Базуючись на розглянутому, розробка, яка полягає у необхідності вирішення проблеми можливості керування побутовою мобільною холодильною

установкою, яка дозволить вирішити її, та спроектувати технічно інформаційну систему і отримати застосунок для її керування, на основі якого буде можливість коректно організувати та оптимізувати роботу холодильної установки є **актуальною задачею**.

Метою роботи є розробка програмного забезпечення віддаленого керування холодильною установкою в середовищі Arduino IDE.

Задачі дослідження бакалаврської роботи:

- проаналізувати методи та технології розробки додатків IoT;
- проаналізувати існуючі аналоги;
- проаналізувати та обрати засоби реалізації системи;
- виконати моделювання;
- розробити застосунок за допомогою сервісу RemoteXY;
- створити систему що буде аналізувати зняті значення температури та керувати установкою;
- протестувати розроблений додаток.

Об'єкт дослідження — процеси, що відбуваються у SMART-системах управління термоустановками на базі модуля ESP 8266.

Предмет дослідження — термоустановка на базі модуля ESP 8266 та застосунок до неї.

Методи дослідження бакалаврської роботи: у роботі використано принципи та методи IoT побудови систем та мови C++ для вправдження запропонованого підходу.

Апробація: участь у LI науково-технічній конференції факультету інформаційних технологій та комп'ютерної інженерії 2022 з доповіддю "SMART-система управління термоустановкою на базі модулі ESP8266". Тези опубліковано 12.06.2022.

Практичне значення отриманих результатів полягає в можливості використання розробленої системи для побутових потреб.

1 ОГЛЯД СУЧАСНИХ МОБІЛЬНИХ ХОЛОДИЛЬНИХ УСТАНОВОК ТА ЗАСОБІВ РОЗРОБКИ ІОТ СИСТЕМ

1.1 Аналіз існуючих холодильних установок

Головною ідеєю роботи є створення «розумної» системи керування побутовою мобільною холодильною системою, на прикладі автохолодильника. За звичай це термобокс з системою охолодження на основі елемента Пельтьє без будь-кого керування, що може працювати від мережі автомобіля чи побутової.

Так наприклад деякі автохолодильники обладнанні термостатом, но їх вартість набагато вища та не передбачене віддалене керування. Розглянемо мобільні холодильні системи та подібні прототипи до проєктованої холодильної установки.

Найблищим прототипом для розробки є автомобільний холодильник Mirabelle E-24 12/230 В із вбудованим адаптером 230 В і процесором, що регулює споживану потужність залежно від режиму роботи (рисунок. 1.1). Процесор сам вибере потрібний режим охолодження й переведе автохолодильник у режим зі зниженим споживанням живлення, але в ньому не передбачене віддалене керування [1].



Рисунок 1.1 — Автомобільний холодильник Mirabelle E-24 12/230 В

У довгій подорожі, коли ваш автохолодильник під'єднаний і працює від бортової мережі автомобіля, процесор керує системою охолодження й регулює споживану потужність автоматично. Автохолодильник термоелектричний,

працює за принципом Пельтьє, не містить компресора й озоноруйнівних речовин; система енергоощадження EEI, мануальний контроль (потужності) за допомогою регулятора, у трьох режимах: Eco-Min-Max-контроль (економія електроенергії до 70% у разі під'єднання від побутової мережі 220 В). Відсік для зберігання шнура живлення і штепсельної вилки (12 В) вмонтований у кришку. Кришка знімна — для зручності чищення й доступу до вмісту; має силіконовий ущільнювач для мінімізації конденсату та втрат температури. Холодильник можна під'єднувати до будь-якого джерела живлення усередині машини, що має напругу 12 В (до прикурювача) або за допомогою адаптера до побутової мережі 230 В. Режим роботи 12/230 В. Матеріал — ударостійкий пластик + внутрішній шар термоізоляційної піни PU; внутр. шар — пластик, не містить ПВХ. Вбудований вентилятор забезпечує постійний ефект охолодження. Корисний об'єм 21.75 л. 3200 грн. [1].

Автохолодильник Ezetil E-3000 12V/24/230V AES/LCD SSBF (рисунок 1.2) має ефективну систему охолодження, не потребує обслуговування, гарантує оптимальний режим охолодження. Внутрішній вентилятор забезпечує швидкий і постійний ефект охолодження. Оснащений захистом від розрядки акумуляторної батареї та температурним дисплеєм. Холодильник можна підключити до будь-якого джерела живлення всередині машини, що має напругу 12 В або 24 В [1]. Вартість 6700 грн.



Рисунок 1.2 — Ezetil E-3000 12V/24/230V AES/LCD SSBF

Опис автохолодильника термоелектричного Vitol VBL-122A 12V/220V 45/55W 22л. Переносний автохолодильник Vitol VBL-122A буде дуже корисний у спекотні літні дні. Відмінно підходить для активного відпочинку або автоподорожей, так як у ньому не використовується компресор і йому не потрібен холодоагент. Працює від мережі 12V і 220V. Охолоджує продукти на 18°C нижче температури навколишнього середовища, але не нижче 0°C.

Характеристики:

- об'єм 22 л;
- споживана потужність при 12V — 45W;
- при 220V — 55W;
- матеріал — поліпропілен, алюміній;
- система охолодження — термоелектричні панелі;
- термоізоляція — ізоляційний пінополіуретан, вартість 1600 грн.

Холодильник термоелектричний для використання в автомобілі, на природі, дачі або на рибалці ThermoMix BL-219-19L. Холодильник BL-219 — 19L відрізняється оптимальним обсягом холодильної камери, яка становить 30 літрів. Холодильник дозволяє охолоджувати продукти на 18 градусів, але не нижче 0 градусів. Охолодження продуктів відбувається швидко, так як холодильник забезпечений двома системами термоелектричних пластин працюють на основі принципу Пельтьє. Холодильник може працювати від мережі постійного струму 12 і 24 вольт, а так само від мережі змінного струму 220 вольт.

Лицьова панель BL-219 — 19L дозволяє не тільки переключатися між режимами обігріву та охолодження, але і за допомогою регулятора настроїти температуру на вимогу. На вбудованому цифровому дисплеї відображається поточна температура холодильної камери (рисунок 1.3). Вартість 5200 грн [2].

Опис автохолодильника термоелектричного EZetil Mirabelle E-24 12 V (рисунок 1.4). Має швидке і якісне охолодження. Живлення можливе від мережі або використовуючи прикурювач автомобіля. Даний автохолодильник оснащений гумовим ущільнювачем, який допоможе запобігти виникненню

конденсату, користувач зможе отримати завжди свіжі і охолоджені продукти і напої.



Рисунок 1.3 — BL-219-19L

Крім того, тут облаштована продуктивна і ефективна охолоджувальна система, яка сприяє отриманню оптимально низької і стабільної температури в холодильнику. Більше ефекту від охолодження забезпечується також через наявність спеціального вентилятора [2].



Рисунок 1.4 — EZetil Mirabelle E-24 12 V

В результаті, користувачеві надається оптимально вигідна температура всередині пристрою — на 14 градусів нижче від температури зовнішнього середовища. Для запобігання перегину шнура в автомобільному холодильнику передбачений відсік для його зберігання. Завдяки спеціальній конструкції

холодильний пристрій герметично закривається, а зручна ручка надає комфортне і просте перенесення з одного місця на інше. Вартість 1650 грн. [4].

Ізотермічний контейнер Adriatic 22л (рисунок 1.5) Особливості: Матеріал ударостійкий пластик, не містить бісфенолу А; термоізоляція — цілісний полістирол у корпусі, що повторює форму контейнера без розривів У кришці шару полістиролу немає. Виріб призначений для збереження температурного режиму: до 12 год. (Із застосуванням акумуляторів температури). Безпечний для здоров'я й довкілля. Вартість 650 грн. [1].



Рисунок 1.5 — Ізотермічний контейнер Adriatic

На основі аналізу подібних холодильних установок на сайтах онлайн магазинів (<https://baza-ua.com>, <https://130.com.ua>, <https://rozetka.com.ua>, <https://epicentrik.ua> та інших) всі термоелектричні холодильники можна поділити на 3 групи:

- мінімальна комплектація, лише термоелемент і вентилятор, цінова категорія від 1200грн до 2500грн;
- з терморегулятором від 2500грн до 5000грн;
- з процесором від 3200грн до 12000грн.

На основі вище поданого найоптимальнішою буде розробка системи на основі звичайного ізотермічного контейнера, або автохолодильника з мінімальною комплектацією.

1.2 Поняття інтернету речей та його технічна складова

Інтернет речей (англ. Internet of Things, IoT) — концепція мережі, що складається із взаємозв'язаних фізичних пристроїв, які мають вбудовані датчики, а також програмне забезпечення, що дозволяє здійснювати передачу й обмін даними між фізичним світом і комп'ютерними системами за допомогою використання стандартних протоколів зв'язку. Окрім датчиків, мережа може мати виконавчі пристрої, вбудовані у фізичні об'єкти і пов'язані між собою через дротові чи бездротові мережі [3].

Основною ідеєю інтернету речей є можливість підключення різноманітних об'єктів (речей), які людина може використовувати в повсякденному житті, наприклад: холодильник, кондиціонер, автомобіль, велосипед та чайник, до мережі Інтернет. Всі ці об'єкти (речі) повинні бути оснащені вбудованими датчиками, які мають можливість обробляти інформацію, що надходить з навколишнього середовища, та передавачами, щоб мати можливість обмінюватися інформацією та виконувати різні дії в залежності від отриманої інформації. Прикладом впровадження такої концепції є система “розумний будинок”. Ця система аналізує дані навколишнього середовища і в залежності від показників може регулювати різні параметри: освітлення, температуру, вмикати чи вимикати систему сигналізації тощо. Для бездротової передачі даних особливо важливу роль в побудові Інтернету речей відіграють такі характеристики, як ефективність, відмовостійкість, адаптивність, можливість самоорганізації. Згідно авторів [4] основне зацікавлення в цьому сенсі представляє стандарт IEEE 802.15.4, що управляє доступом для організації енергоефективних персональних мереж, і є основою для таких протоколів, як ZigBee, Wi-Fi, Bluetooth, 6LoWPAN.

Інтернет речей — поняття, що охоплює велику кількість пристроїв, які сильно відрізняються одне від одного за своїми технічними складовими та завданнями, які вони вирішують. Проте, можна виділити загальні риси, які притаманні переважній більшості пристроїв IoT.

На апаратному рівні пристрої Інтернету речей, зазвичай, можуть мати:

- сенсори (оптичний сенсор, сенсор температури та вологості, сенсор диму чи газу тощо);
- електромеханічні елементи (актуатори, реле);
- мікроконтролери чи одноплатні комп'ютери, які є “мозком” пристрою;
- модулі зв'язку та передачі даних.

Також багато виробників IoT пристроїв створюють мобільні або веб-додатки, за допомогою яких може відбуватися взаємодія користувачів з пристроями.

До контролерів можна підключити різноманітні датчики. Датчик — це окремий пристрій, який здатний вимірювати певну фізичну величину або реагувати на фізичне явище і передає інформацію про це в вигляді електричного сигналу [3]. Вимірювальне перетворення — операція перетворення вхідного сигналу у вихідний, інформативний, параметр якого з заданою точністю функціонально пов'язаний з інформативним параметром вхідного сигналу. Вимірювальний перетворювач (ВП) — це технічний засіб, що реалізує вимірювальне перетворення. Залежність між інформативними параметрами вхідного і вихідного сигналів ВП встановлюється номінальною статичною характеристикою перетворення, яка може виражатися у вигляді рівняння, графіка або таблиці. Велика частина природних залежностей між фізичними сигналами має нелінійний характер [4]. Чутливий елемент — це вимірювальний перетворювач, що забезпечує перетворення вимірюваної величини в інформативний сигнал, зручний для подальшого використання (передачі, обробки, зберігання тощо). Будь-який первинний вимірювальний перетворювач чутливий до різних фізичних величин. Наприклад, п'єзоелемент формує електричний заряд під впливом таких величин як сила, температура, вологість, оптичне випромінювання [5, 7]. Датчики розрізняються по тому, що вони фіксують. Існують датчики для вимірювання температури, відстані, вібрації, звуку, вологості, руху, магнітного поля та багато інших. Також вони розрізняються за типом сигналу. Деякі передають сигнал у вигляді змінної

напруги (аналогові), деякі у вигляді послідовності низької і високої напруги (цифрові), інші змінюють власний опір. Тип сигналу визначає як фізичні показання проектуються на електричні величини придатні для зчитування. Далі, сенсори розрізняються по протоколу. Він визначає як приймаюча сторона (наприклад ESP 8266) повинна інтерпретувати сигнал, щоб отримати значення вимірюваної величини. Напівпровідникові датчики температури призначені для вимірювання температури від мінус 55°C до плюс 150°C. В цей діапазон потрапляє величезна кількість завдань, як в побутових, так і в промислових додатках. Завдяки високим характеристикам, простоті застосування і низькій вартості напівпровідникові датчики температури виявляються дуже привабливими для застосування в мікропроцесорних пристроях вимірювання та автоматики. Наприклад датчик на основі мікросхеми DS18B20 для перетворення температури в цифрове значення безпосередньо в корпусі датчика дає переваги у вигляді високої точності і стабільності показань, відсутності впливу з'єднувальних проводів на показання датчика, легкості підключення до мікроконтролерів і ПК, можливість роботи декількох датчиків на одному шлейфі [8, 9].

У більшості випадків “розумні” речі використовують бездротовий зв’язок використовуючи технології Wi-Fi, Bluetooth Low Energy, RFID, ZigBee тощо. Використання цих відкритих та достатньо розповсюджених технологій дозволяє цим пристроям працювати разом з великою кількістю інших пристроїв, зберегти кошти виробника через відсутність необхідності впровадження власного протоколу зв’язку, та забезпечити високий рівень комфорту для користувача, смартфони та планшети якого вже мають можливість роботи за вищезазначеними технологіями. На відміну від інших мереж Wi-Fi має більшу швидкість передачі даних, також має можливість передавати данні на відстань до 300 метрів, що є більше ніж у Bluetooth. Мережа Wi-Fi може працювати в діапазоні таких частот від 2,4 ГГц до 5 ГГц — це дає можливість обирати частоту, для запобігання виникнення помилок або накладання сигналів, тому як частота 2,4 ГГц є розповсюдженою серед більшої кількості пристроїв.

На програмному рівні пристрої IoT можуть використовувати велику кількість протоколів, які вибирають виробники в залежності від покладеної на конкретний пристрій задачі, умов роботи пристрою та його апаратних можливостей. Найбільш розповсюдженими протоколами роботи “розумних” пристроїв є: MQTT, XMPP, HTTP/HTTPS, SOAP, RTSP, STOMP та DDS. Також, деякі пристрої можуть використовувати пропрієтарні протоколи від виробника. В такому випадку взаємодія користувача з цими пристроями може відбуватися через спеціальні додатки, розроблені виробником [3].

1.3 Аналіз сучасних сервісів розробки IoT систем

Проекти для інтернету речей реалізуються у багатьох сферах, включаючи охорону здоров'я, промисловість і бізнес. Існує незліченна кількість рішень для розробки в області IoT, тому нерідко спантеличує вибір найбільш відповідного інструменту. Розглянемо короткий огляд найбільш актуальних платформ для побудови IoT систем [10, 11].

Eclipse IoT — це один з найбільш корисних інструментів розробки для IoT. Він є продуктом спільних зусиль декількох організацій з різних галузей. З його допомогою можна створювати фреймворки, отримувати реалізації з відкритим вихідним кодом, надавати сервіси, що допомагають у використанні IoT-рішень, а також створювати інструменти для самих розробників в цій області. Він зарекомендував себе як інструмент, що надає високоякісні послуги доступним і прогнозованим чином [10].

IBM Watson використовується компаніями при впровадженні штучного інтелекту, виявляючись також корисним для тих, хто тільки починає свій бізнес або працює в банківській, виробничій та інших сферах. З його допомогою ви зможете надати клієнтам індивідуальний досвід використання продукту, мінімізувати ризики і стимулювати інновації. В даному випадку всі сервіси надаються експертами, що прискорює роботу і надає гарантії надійності. Ця програма дозволить прогнозувати майбутні зміни і вносити необхідні відповідні коригування.

Arduino є одним з популярних засобів розробки для IoT. Він легкий у використанні і дозволяє налаштувати управління зручним вам способом. Мікроконтролери Arduino проєктуються для виконання найрізноманітніших завдань. На введення цих пристроїв можна подавати інформацію, керуючи таким чином різними пристроями, а за допомогою виведення можна відображати і зберігати дані, наприклад на ПК.

Node-Red — це засіб який завойовує все більшу популярність у світі технологій та інтернету речей, зокрема. Node-Red — це дуже гнучкий і корисний візуальний редактор з відкритим кодом, який дозволяє розробникам будь-якого рівня налагоджувати взаємозв'язок між хмарними системами, базами даних, API і не тільки. Працювати він може в тому числі і на низькобюджетних машинах, включаючи недорогі хмарні рішення і Raspberry Pi. Складається даний інструмент з 225,000 модулів, що полегшує розширення палітри вузлів для додавання нових можливостей. Node-Red розроблений IBM, тому за допомогою даного редактора можна створювати Java-функції, які в підсумку можна зберігати для подальшого використання, так само як шаблони і потоки [9].

Particle — це платформа яка була запущена в 2013 році, надаючи полегшену взаємодію з IoT як розробникам, так і бізнесменам. Крім того, потрапивши в число швидкозростаючих інноваційних компаній, вона була відзначена в доповіді Gartner серед передових рішень в сфері IoT. Дана платформа характеризується надійністю і захищеністю, дозволяючи програмістам розробляти, підключати, об'єднувати і адмініструвати IoT-рішення швидко і без особливих зусиль. Багато інженерів з різних країн використовують цей сервіс для організації і управління розробкою IoT-продуктів [9].

Каа відноситься до числа найбільш гнучких і розширюваних платформ, що дозволяють як створювати IoT-рішення, так і управляти підключеними пристроями. Вона пропонує багатий набір можливостей, включаючи збір і обробку даних, аналітику, управління конфігурацією, візуалізацію даних і багато іншого. Каа підтримує відкриті протоколи IoT і забезпечує повноцінну безпеку для користувачької інформації. Вона дає вам повний контроль над пристроями,

при цьому ви можете вибирати зручний для вас варіант хостингу, будь то публічна хмара або приватні локальні сервери. З її допомогою можна підключати і управляти пристроями, використовуючи графічний UPI або REST API. Крім того, вона проста у використанні і має високу сумісність [9].

ThingsBoard — ця IoT-платформа допоможе організувати для проєктів збір та обробку даних, управління пристроями і візуалізацію. При її використанні вивід продукту на ринок займає мінімум часу, а можливі технічні складності вирішуються дуже ефективно. Вона дозволяє відстежувати тисячі і навіть мільйони пристроїв, здійснюючи взаємозв'язок за допомогою таких протоколів як HTTP, MQTT і ряду інших. Використовуючи ThingsBoard, можна моніторити стек технологій і вільно вибирати провайдерів хмарних послуг. Цей сервіс дозволяє моделювати файли проєктів, налаштовувати пристрої, а також отримувати на основі їх даних потрібну інформацію, роблячи ці процеси більш ефективними [10].

Blynk IoT Platform — досить популярна IoT-платформа, що відрізняється рядом унікальних можливостей, таких як відображення, зберігання і візуалізація даних. За допомогою бібліотеки Blynk можна підключати більше 400 моделей обладнання, організовуючи з'єднання через Wi-Fi, Ethernet, 2G, 3G, 4G, LTE і т.д. Вся платформа поділяється на три головні компоненти: додаток Blynk надає віджети, за допомогою яких ви можете створювати інтерфейси для своїх проєктів, сервер Blynk дозволяє управляти тисячами пристроїв, а також налагоджувати зв'язок між обладнанням і смартфонами, бібліотеки Blynk забезпечують взаємодію з сервером і обробку команд [9].

Це відкрита платформа, здатна працювати в будь — якому середовищі. Для неї характерна швидкість, надійність, а також можливість обробки мільярдів інформаційних потоків і запитів, одержуваних з пристроїв.

Device Hive — є відкритим інструментом, що допомагає підключати до вашого додатку пристрої і додавати в нього об'єкти. Підключення пристроїв здійснюється через WebSocket, REST API або MQTT. Платформа підтримує кілька мов програмування, що робить її універсальною для всіх пристроїв. Крім

іншого, вона надає учасникам можливість обговорення розробки або бізнесу, а також пропонує багатообіцяючі підходи до створення ПЗ в комерційних цілях. Підключення пристрою можна контролювати без використання мережі конфігурації [9].

ThingWorx — це чудова платформа, за допомогою якої як програмісти, так і інші клієнти створюють незліченні рішення для інтелектуальних підключених операцій (smart connected operations) і продуктів в сфері медицини, промисловості і виробництва. Цей ресурс надає провідні в області IoT можливості для промислових організацій, які дозволяють ефективніше використовувати бізнес-моделі і нарощувати сам бізнес [9].

RemoteXY — це система розробки і використання мобільних графічних інтерфейсів для управління контролерами зі смартфона або планшета. До складу системи входять: редактор мобільних графічних інтерфейсів для контролерів, розміщений на сайті remotexy.com, мобільний додаток RemoteXY, що дозволяє підключатися до контролера і відображати графічні інтерфейси. Підтримувані IDE: Arduino IDE; FLProg IDE; MPIDE. Підтримувані мобільні операційні системи: Android; iOS [12].

Для розробки системи обрано платформи Arduino та RemoteXY, оскільки дана термоустановка є побутовою, не потребує великої точності та надійності, тому вартісні контролери чи інші системи є недоцільними, а отже дані платформи є дешевими, простими в проектуванні, оптимальними саме для цієї розробки [12].

Платформа Arduino може бути використана для розробки систем, що керують датчиками та перемикачами. Такі системи, у свою чергу, можуть керувати роботою широкого діапазону індикаторів, двигунів та інших пристроїв. Модулі на базі Arduino можуть бути як автономними, так і взаємодіяти з програмним забезпеченням, що працює на персональному комп'ютері. Будь-яку плату Arduino можна зібрати власноручно, або купити готовий модуль. Середовище розробки для програмування такої плати безкоштовне та має відкритий вихідний код. Платформа Arduino спрощує процес роботи з

мікроконтролерами та надає низку переваг для інженерів, студентів та радіолюбителів. Основні характеристики платформи [13, 14]:

- низька ціна, порівняно зі подібними платформами, плати Arduino мають порівняно низьку вартість (від 10 \$), а можливість зібрати плату вручну дозволяє максимально зменшити затрати та отримати Arduino за мінімальну ціну [14];

- кросплатформеність, програмне забезпечення Arduino працює на операційних системах Windows, OSX та Linux, у той час як більшість аналогів орієнтовані на роботу тільки у Windows;

- просте та зручне середовище програмування, середовище програмування підходить як для початківців, так і для досвідчених програмістів та інженерів, воно ґрунтується на середовищі програмування Processing — відкритій мові програмування, що ґрунтується на Java і є зручним та легким в освоєнні інструментарієм для тих, хто бажає програмувати анімацію та інтерфейси, завдяки цьому, студенти, які освоюють ази програмування в середовищі Processing, зможуть без проблем працювати з Arduino;

- можливість розширювати програмне забезпечення, програмне забезпечення Arduino випускається як інструмент, який можуть доповнити досвідчені користувачі, зокрема, дана мова може бути доповнена бібліотеками C++ [14];

- можливість розширювати апаратне забезпечення, велику популярність плата Arduino отримала не тільки завдяки низькій вартості, простій розробці та програмуванню, але, головним чином, через наявність плат розширення (shields), що дають Arduino додаткову функціональність.

Шилди (крім маленьких модулів та плати LilyPad) підключаються до Arduino за допомогою наявних на них штирових роз'ємів. Існує широкий асортимент різних за функціональністю shields — від найпростіших, призначених для макетування, до складних, що являють собою окремі багатофункціональні прилади. На відміну від модулів, які можна підключити до

довільних виходів Arduino, виходи shields, в більшості випадків, прив'язані до виходів Arduino.

Arduino — це комбінація апаратної і програмної частин для простої розробки електроніки. Апаратна частина включає в себе велику кількість видів плат Arduino з вбудованими програмованими мікроконтролерами, а так само додаткові модулі. Програмна частина складається з середовища розробки (програми для написання скетчів і прошивки мікроконтролерів Arduino), спрощеного мови програмування, величезної кількості готових функцій і бібліотек. Arduino створювалося викладачами для більшого залучення студентів в електротехніку. Ідея мала величезний успіх і Arduino пішло набагато далі. Завдяки відкритій архітектурі виробляти дані мікроконтролери, доповнювати модельний ряд, писати програми може хто завгодно. Всі схеми і вихідний код програм є у відкритому доступі [11, 13].

Arduino був створений для навчання студентів та школярів електротехніці, програмуванню, електроніці та системам автоматизації. За допомогою мікроконтролера ви можете не тільки виконувати навчальні проекти, а й виготовляти справжнє корисне обладнання. Arduino створює проекти автоматизації, розумні домашні пристрої, портативні метеостанції, роботи-маніпулятори та багато інших корисних пристроїв. З технічної точки зору, Arduino вміє приймати і відправляти сигнали відповідно до інструкцій в прошивці. Звучить досить скромно, але на практиці це дозволяє отримувати і обробляти інформацію з сенсорів і передавати команди виконавчим механізмам або інших пристроїв. Наприклад: мікроконтролер може отримувати дані з датчиків температури, тиску, вологості і виводити зведену інформацію на дисплей. Цих можливостей вистачає для реалізації складних пристроїв, таких як безпілотні літальні апарати, 3D — принтери, роботизований маніпулятор, радіокеровані машинки, човни, всюдиходи та т. д.

Переваги:

— проста мова програмування Wiring, яку без проблем опановують школярі та діти дошкільного віку;

— Wiring є надбудовою над C++, тому в Arduino можна застосовувати весь інструментарій C та C++ [11].

Для початку роботи з Arduino вам знадобиться спеціальне програмне забезпечення. Це середовище для розробки прошивок Arduino IDE. У цій програмі легко і зручно писати скетчі і завантажувати їх на ваш мікроконтролер Arduino. У середовищі розробки вже попередньо встановлено велику кількість прикладів і додаткових бібліотек (рисунок 1.6).

```

sketch_apr07a $
int a=13;
int b=12;
int c=11;
void setup () {
  pinMode(a, OUTPUT);
  pinMode(b, OUTPUT);
  pinMode(c, OUTPUT);
}
void loop() {
  digitalWrite (a, LOW);
  digitalWrite (a, HIGH);
  digitalWrite (b, LOW);
  digitalWrite (c, LOW);
  digitalWrite (c, HIGH);
  delay(10000);
}
Компілювання виконано.
Скетч використовує 944 байтів (2%) місця зберігання для
Глобальні змінні використовують 9 байтів (0%) динамічної

```

Рисунок 1.6 — Зовнішній вигляд Arduino IDE

Отже, Arduino — це зручна платформа для реалізації проектів різної складності. Вона прийнятна як початківцям, які ще не мають навичок у сфері робототехніки, так і досвідченим користувачам. Для платформи Arduino наявна велика кількість матеріалів для розробки, починаючи від бібліотек, які можна використовувати для спрощення програмування, закінчуючи використанням уже готових проектів, що можуть надихнути на створення нових. Більш потужні плати Arduino (TRE, YUN, DUE) можуть бути застосовані для розв’язання більш складних задач, зв’язаних з розробкою великих проектів.

RemoteXY — це система розробки і використання мобільних графічних інтерфейсів для управління контролерами зі смартфона або планшета. До складу системи входять: редактор мобільних графічних інтерфейсів для контролерів,

розміщений на сайті remotexy.com, мобільний додаток RemoteXY, що дозволяє підключатися до контролера і відображати графічні інтерфейси [12].

Розглянемо особливості системи RemoteXY. Конфігурація графічного інтерфейсу зберігається у контролері. При підключенні немає жодної взаємодії з сторонніми серверами для того, щоб завантажити графічно інтерфейс. Конфігурація графічного інтерфейсу завантажується на мобільний додаток з контролера. З одного мобільного додатка ви можете керувати всіма своїми пристроями. Кількість пристроїв не обмежена.

Підтримуються такі способи зв'язку між контролером та мобільним пристроєм:

- інтернет із будь-якого місця через хмарний сервер;
- WiFi в режимі клієнта та точки доступу;
- Bluetooth;
- Ethernet за IP-адресою або URL;
- USB OTG (доступно лише для Android із підтримкою USB OTG).

Генератор вихідного коду інтерфейсу підтримує такі контролери:

- Arduino UNO, MEGA, Leonardo, Pro Mini, Nano, MICRO та сумісні AVR контролери;
- контролери на ESP8266;
- контролери на ESP32;
- ChipKIT UNO32, ChipKIT uC32, ChipKIT Max32.

Підтримувані IDE:

- Arduino IDE;
- FLProg IDE;
- MPIDE.

Мобільні операційні системи, що підтримуються:

- Android;
- iOS [12].

1.4 Аналітичний огляд обладнання для створення IoT систем

Для розробки IoT систем можуть бути використанні контролери, одноплатні компютери з засобами передачі інформації: WiFi модулі, Bluetooth модулі, Ethernet модулі, GSM/GPRS/GNSS модулі, радіомодулі та інше. Для створення побутової системи керування холодильною установкою самим оптимальним та найдешевшим варіантом є плата NodeMCU, яка являє собою і контролер і WiFi модуль.

NodeMCU — це недорога платформа IoT з відкритим кодом [15].

Спочатку вона містила в собі прошивку, яка працювала на ESP8266 Wi-Fi SoC від Espressif Systems, та апаратне забезпечення, яке базувалося на модулі ESP12. Пізніше була додана підтримка 32-бітного MCU ESP32. Назва "NodeMCU" поєднує в собі "node""вузол" і "MCU" (мікроконтролерний блок) [16].

Мікроконтролер ESP32 випущений влітку 2016 року фірмою Espressif. Це потужний та недорогий мікроконтролер. Пристрій являє собою систему на кристалі, побудовану за технологією TSMC 40 нм, з Wi-Fi і Bluetooth контролерами. Він оснащений двоядерним 32-бітовим процесором, який працює на частотах 80, 160 або 240 МГц. Також в систему інтегровані антенні комутатори, радіочастотні компоненти, фільтри, підсилювачі, модулі управління живленням. Підключається ESP32 до комп'ютера за допомогою USB роз'єму [15].

Технічні характеристики ESP32:

- двох або одноядерний 32-бітний процесор Tensilica Xtensa LX6;
- тактова частота 160МГц або 240 МГц;
- 520 Кб SRAM;
- максимальна потреба 260 мА, в режимі 10 мА;
- стандарти бездротового зв'язку - Wi-Fi: 802.11 b / g / N, Bluetooth: v4.2 BR / EDR та BLE;
- наявність датчиків температури, Холла, тач-сенсорів;

- інфрачервоне дистанційне управління;
- наявна можливість підключення двигунів та світлодіодів через ШІМ роз'єм;
- стандарт IEEE 802.11 з підтримкою WFA, WPA / WPA2 та WAPI;
- можливість безпечного завантаження;
- шифрування флеш-диску;
- вбудоване керування енергоживленням;
- програмування модуля ESP32 можливе на різноманітних платформах та середовищах [16].

Хоча мікроконтролери ESP8266 та ESP32 виготовила одна фірма Espressif, вони відрізняються один від одного. Модуль ESP32 відрізняється від свого попередника за рахунок вдосконалення обладнання, збільшення функціоналу, кількості виходів GPIO та об'єму пам'яті. Однак вартість ESP32 також дві рази більша, ніж вартість мікроконтролера ESP8266.

Перевагою мікроконтролера ESP32 є більш швидкий Wi-Fi та Bluetooth. Встановлений в ESP32 більш потужний процесор легко реалізує складні проекти. Для більш дешевих розробок використовується мікропроцесор ESP8266 [15, 16].

Об'єм пам'яті в модулі ESP32 збільшено до 512 Кб порівняно з 160 Кб в мікропроцесора ESP8266. До декількох контактів на ESP32 прикріплені ємнісні сенсорні датчики та датчик температури. ESP32 має вісімнадцять 12-бітних АЦП каналів порівняно з одним 10-бітним каналом у його попередника.

До недоліків ESP32 на сучасному етапі можна віднести відсутність бібліотек для підтримуючих сенсорів і малу кількість драйверів [8], які з часом будуть подолані. Тому наш вибір зупинився на використанні ESP8266.

ESP8266 представляє собою мікроконтролер з інтегрованим інтерфейсом Wi-Fi виробництва компанії Espressif Systems. Окрім вбудованого модуля Wi-Fi, мікроконтролер характеризується відсутністю флеш-пам'яті в SoC, тому мікрокод контролера виконуються із зовнішньої флеш-пам'яті, підключеної за інтерфейсом SPI [17]. Мікроконтролер набув значної популярності в 2014 році разом з виходом перших продуктів на його основі, а також внаслідок низької

вартості. На початку 2016 року розпочалося виробництво контролера ESP8285, що поєднував у собі ESP8266 і флеш-пам'ять на 1 МБайт. Разом із цим Espressif представила логічне продовження лінійки цього сімейства контролерів — мікросхему ESP32 і модулі на її основі. Як було зазначено вище, мікроконтролер не має на кристалі вбудованої пам'яті.

Мікросхема ESP8266 — один з найпопулярніших інструментів для організації бездротового зв'язку. Мережеве рішення з Wi-Fi та можливість виконання записаних в його пам'ять додатків, які працюють в зв'язці з іншими МК. Однак вбудованих можливостей МК серії ESP досить для реалізації великої кількості проектів.

У більшості випадків цей МК застосовують в системах автоматизації побуту і IoT (Internet of Things — Інтернет речі). Якщо МК буде застосовуватися там, куди не дістає WiFi-мережа, або її там просто і не повинно бути, то ESP8266 може працювати в режимі точки доступу.

Існує безліч модифікацій плат та іменованих зазвичай від ESP01 до ESP12. Зараз вже з'явилися ще інші найменування плат від сторонніх розробників. Відмінності в платах полягає в основному в портах введення-виведення, кількості флеш-пам'яті, виду конекторів і т.п. Процесор — один і той самий, так що з точки зору програмування не має значення яку плату програмувати.

Важливою особливістю є відсутність користувальницької незалежної пам'яті на кристалі. Програма виконується від зовнішньої SPI постійно записуючому пристрої (ПЗП), за допомогою динамічного завантаження необхідних елементів програми.

ESP8266 розроблений так, що він може використовувати підключений до нього модуль пам'яті і це зазвичай флеш-пам'ять [16].

Має такі характеристики:

— споживання енергії змінюється в дуже широкому діапазоні, при передачі на повній потужності воно становить 170 міліампер, а в режимі сну всього 10 мікроампер;

- підтримує Wi-Fi протоколи 802.11 b / g / n з WEP, WPA, WPA2;
- володіє 14 портами введення і виведення, SPI, I2C, UART, 10 біт АЦП;
- підтримує зовнішню пам'ять до 16 МБ;
- наявність 4 Мб флеш-пам'яті;
- будовані кнопки для перезавантаження і перепрошивці.

Існує різна кількість модулів ESP-8266. Найпоширеніші з них:

- EPS-01;
- ESP-03;
- ESP-07;
- WeMos D1 mini.

Модуль ESP-01— це мініатюрний мікроконтролер з Wi-Fi передавачем, який дозволяє працювати автономно, без додаткових плат. На ESP-01 розташовані два основних чіпа — це мікро контролером ESP8266 і flash пам'ять на 1 Мб. Основною перевагою і недоліком є надрукована антена, так як це дає малі розміри плати, але також і можливість передавати на менші відстані ніж через звичайну антену. На платі встановлено два світлодіоди, червоний і блакитний. Червоний світлодіод, світиться, коли на модулі є живлення, а синій блимає при виконанні команд. Для підключення модуля ESP-01, передбачено вісім виводів, два з готових є цифровим входами-виходом, що підтримують широтно — імпульсну модуляцію [16]. Модуль зображений на рис. 1.7.

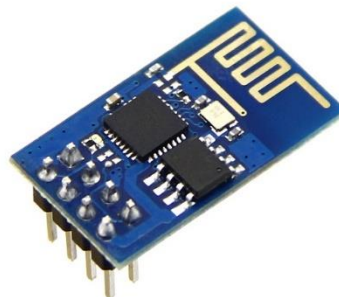


Рисунок 1.7 — Зовнішній вигляд модулю ESP-01

Модуль ESP-03 є аналогічним пристроєм ESP-01, однак має одну зміну у вигляді керамічної антени, що зменшує розмір плати, а також не багато покращує дальність використання Wi-Fi. Друкована антена цілком придатна до застосування в робочих пристроях на базі ESP8266, за умови, що вона нормально узгоджена з платою, однак трохи поступається керамічній. Також даний вид антени не рекомендований в умовах підвищеної вологості, так як антена може розмокнути і втратити свої робочі властивості, що є не припустимим [16]. Даний модуль зображений на рис 1.8.

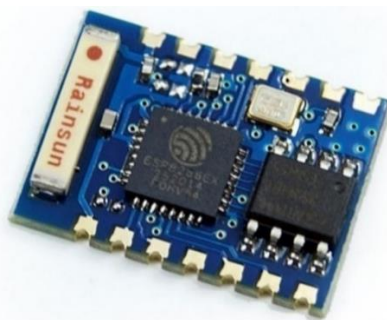


Рисунок 1.8 — Зовнішній вигляд модулю ESP-03

Модуль ESP-07 — це версії модуля з'являється екранування, для захисту від зовнішніх впливів. Також з'явилася можливість підключення зовнішньої антени, але за замовчуванням модуль поставляється з вбудованою керамічною антеною. На даній платі з'явилися можливості підключення додаткових виводів, яких складає 9 штук і вивід під АЦП, що є перевагою над попередніми версіями, цей модуль зображений на рис 1.9 [15, 17].



Рисунок 1.9 — Зовнішній вигляд модулю ESP-07

Модуль WeMos D1 mini — це плата WeMos D1 mini, на основі Wi-Fi модуля ESP-12F, на чіпі ESP8266EX. На платі є 11 цифрових GPIO, аналоговий вхід.

WeMos D1 mini побудована на базі 32 розрядного мікроконтролера ESP8266 з інтегрованим Wi-Fi модулем (802.11 b / g / n 2.4 ГГц). Так само на платі присутні стабілізатор напруги на 3,3 В, роз'єм USB типу Micro-B і USB-UART перетворювач на базі чіпа CH340G. Мікроконтролер ESP8266 працює на тактовій частоті 80 МГц і володіє оперативною пам'яттю RAM даних на 80 кБ, і пам'яттю RAM інструкцій на 32 кБ. Програми зберігаються в flash пам'яті об'ємом 4 Мб, цей модуль зображен на рис. 1.10.



Рисунок 1.10 – Зовнішній вигляд модулю WeMos D1 mini

Модуль WeMos D1 mini має великі габаритні розміри ніж плати серії ESP, проте він більше підходить для роботи з іншими платами необхідними для побудови вузла літаючої сенсорної мережі. Це обумовлено тим, що ESP 8266 дуже чутливий до робочій напруги, що складає 3,3 В, на даній платі це проблема вирішена шляхом використання стабілізатора напруги, тому спрощується використання різних джерел живлення і на макеті, можливість використання живлення від USB адаптера. Основною перевагою даного рішення, є можливість використання плати CC2530, так як дана плата має живлення в 5 В [16].

ZigBee — трансивер CC2530. Мікросхема CC2530 підтримує стандарт IEEE 802.15.4, призначена для організації мереж стандарту ZigBee Pro, засобів дистанційного керування на базі ZigBee RF4CE і обладнання стандарту Smart

Energy. Інтегральна схема CC2530 об'єднує в своєму кристалі радіочастотний трансивер і мікроконтролер, ядро якого сумісно зі стандартним ядром 8051 і відрізняється від нього поліпшеним швидкодією.

Трансивер CC2530 є ідеальним вибором для конструювання пристрою, він зображений на рис. 1.11 [16].

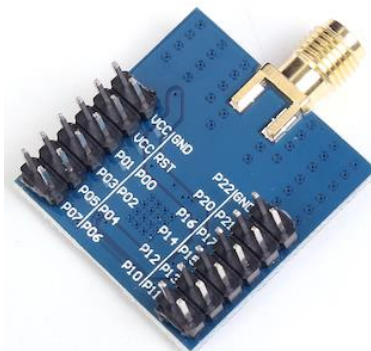


Рисунок 1.11 — Зовнішній вигляд ZigBee — трансивера CC2530

Трансивер CC2530 має відповідні характеристики для використання в пристроях літаючої сенсорної мережі:

- робоча частота 2,4 ГГц;
- режим роботи у вигляді приймача, передавача;
- швидкість передачі даних до 250 кбіт / с;
- напруга живлення від 2 В до 3,6 В, що відмінно вписується в робоче напруги ESP8266;
- низький рівень перешкод;
- дальність зв'язку до 100 м;
- струм, що споживається у вигляді приймача 27 мА, а у вигляді передавача 40 мА;
- 8 каналів АЦП;
- робоча температура: від мінус 40 °С до плюс 125 °С.

Перевагами даного пристрій:

- малі масогабаритні параметри;

- можливість дистанційної настройки пристрою;
- мале споживання енергії;
- можливість підключення великої кількості пристроїв.

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 — трансівером на борту плюс можливість виконання додатків, що записуються в його пам'ять [15].

На основі ESP8266 створена плата NodeMCU V2 та подібна їй NodeMCU V3, яку і використаємо в даній роботі (рисунок 1.12). NodeMCU * (Lolin) являє собою плату розробника на базі чіпа ESP8266 (версія ESP12E), який є UART Wi-Fi модуль з ультра низьким споживанням. Сам чіп проєктувався для пристроїв зі світу інтернет речей, а дана плата дозволяє спростити розробку, тому що на ній вже реалізовано підключення по USB, регулятор живлення і всі виводи чіпа розведені на гребінки зі стандартним кроком 2.54 мм, що дозволяє вставити його в макетную плату і створити прототип навіть не включаючи паяльник. Крім цього плата поставляється з прошивкою NodeMCU, що дозволяє програмувати її за допомогою мови Lua або за допомогою Arduino IDE [16].



Рисунок 1.12 — Зовнішній вигляд мікроконтролера NodeMCU V3

Основні характеристики NodeMCU V3:

- WiFi 802.11 b/g/n;
- підтримка STA/AP/STA+AP режимів;
- вбудований стек протоколів TCP/IP з підтримкою множинних клієнтських підключень (до 5);

- D0 ~ D8, SD1 ~ SD3, можуть бути використані як GPIO, PWM, IIC, тощо;
- струм на виході 15 мА;
- живлення від 4.5В до 9В (10В максимум);
- споживання ~ 70 мА (200 мА максимум);
- очікування <200 мкА;
- швидкість передачі від 110 б/сек до 460800 б/сек;
- підтримка UART / GPIO інтерфейсів передачі даних;
- перепрошивка з хмари або через USB;
- відстань між контактними пінами: 28 мм;
- діапазон робочих температур: від мінус 40° С до плюс 125 ° С;
- вага 18 г [16].

1.5 Обґрунтування вибору мови програмування та середовища

Перш ніж приступити до розробки програмного забезпечення проведемо огляд доступних програмних засобів і мов програмування та відібрати такі, що найкраще підходять для розробки даної програми. Оскільки на вибір мови програмування і середовища програмування також впливає вибір контролера, тому розглянемо основні мови і середовища, що досить легко співпрацюють з контролером NodeMCU V3 на основі ESP 8266.

Для розробки буде проведемо огляд таких мов програмування, що найчастіше використовуються для програмування мікроконтролерів, як C++, Python, Java та Асемблер, Wiring, Lua.

Wiring це діалект мови C++ з додатковими спеціальними бібліотеками. C++ — це компільована мова програмування високого рівня, розроблена Б'ярне Страуструпом. Перша версія мови була випущена у 1979 році. Найактуальніша версія мови C++ 17 вийшла у 2017 році. C++ успадкувала синтаксис і основні аспекти мови C. По суті, C++ є надмножиною мови C, на що також вказує початкова назва — «C with classes» — «C з класами». Мова C++ підтримує парадигму об'єктно-орієнтованого програмування, характеризується високою

швидкодією за рахунок того, що код програми компілюється у машинний код, а також має велику кількість сторонніх бібліотек, що спрощують процес розробки. До недоліків мови відноситься відсутність автоматичного керування пам'яттю («збір сміття» — «garbage collection»), що є потенційно небезпечним, адже допускає існування помилок, що призводять до втрат пам'яті (явище, коли частина пам'яті, виділеної для виконання програми, не вивільняється після завершення її використання). Натомість, ручне керування пам'яттю надає розробнику більше контролю над роботою програми.

Java — мова програмування високого рівня, випущена компанією «Sun Microsystems» у 1995 році. Найактуальніша версія мови Java Standard Edition 10 вийшла 20 березня 2018 року. Має C-подібний синтаксис. Підтримує парадигму об'єктно-орієнтованого програмування. Має дещо гіршу швидкодію у порівнянні з мовами C++ та C#. Характеризується наявністю великої кількості сторонніх бібліотек. Java використовує автоматичний збирач сміття для керування пам'яттю під час життєвого циклу об'єкта. Програміст вирішує, коли створювати об'єкти, а віртуальна машина відповідає за звільнення пам'яті після того, як об'єкт стає непотрібним. Коли до певного об'єкта вже не залишається посилань, збирач сміття може автоматично прибирати його із пам'яті.

Python — інтерпретована об'єктно-орієнтована мова програмування високого рівня зі строгою динамічною типізацією. Розроблена в 1990 році Гвідо ван Россумом. Структури даних високого рівня разом із динамічною семантикою та динамічним зв'язуванням роблять її привабливою для швидкої розробки програм, а також як засіб поєднання наявних компонентів. Python підтримує модулі та пакети модулів, що сприяє модульності та повторному використанню коду. В мові програмування Python підтримується кілька парадигм програмування, зокрема: об'єктно-орієнтована, процедурна, функціональна та аспектно-орієнтована. Python має ефективні структури даних високого рівня та простий, але ефективний підхід до об'єктно-орієнтованого програмування. Те, що це інтерпретована мова з динамічною обробкою типів, робить її ідеальною для написання скриптів та швидкої розробки прикладних програм, але самі

прикладні програми будуть дещо повільніше працювати, аніж якби вони були написані на компільованій мові програмування.

Асемблер (assembler) — система програмування, яка включає мову асемблера та транслятор з цієї мови. Асемблер є мовою програмування низького рівня. Чим нижчий рівень мови програмування, тим ближча специфіка роботи програми до самого процесора, для якого вона й була написана. Вважається, що мови низького рівня складніші й потребують більш вузької спеціалізації програміста, оскільки програма написана на асемблері для одного типу процесорів виявиться не завжди придатною для роботи з іншими процесорами. З іншого боку програми написані на асемблері компактні та швидкі, що теж є немаловажливим [21].

У простому випадку асемблер переводить одну пропозицію початкової програми в один об'єкт (команду, константу) модуля завантаження (трансляція «один в один»). При цьому взаємне розташування об'єктів в модулі завантаження і, зрештою, в пам'яті машини визначається порядком пропозицій в початковій програмі на автокоді і повністю залежить від програміста. Асемблер виконує і допоміжні функції, такі, як підготовка до друку документів необхідної форми, реєстрація зв'язків даної програми з іншими програмами і т. д. Для цієї мети в автокодах передбачаються команди асемблера, які не породжують об'єктів в робочій програмі і призначені тільки для вказівки допоміжних дій асемблера.

Lua — швидка і компактна скриптова мова програмування, розроблена підрозділом Tecgraf Католицького університету Ріо-де-Жанейро (Computer Graphics Technology Group of Pontifical Catholic University of Rio de Janeiro in Brazil). Інтерпретатор мови є вільно поширюваним, з відкритим сирцевим кодом на мові C [10].

За можливостями, ідеологією і реалізацією, мова найближча до JavaScript, проте Lua відрізняється могутнішими і набагато гнучкішими конструкціями, спроектованими з метою «не плодити сутності понад необхідне». Хоча Lua не містить поняття класу і об'єкта в явному вигляді, механізми об'єктно-орієнтованого програмування (ООП) з підтримкою прототипів (включаючи

множинне успадкування) легко реалізуються з використанням метатаблиць, які також дозволяють перевантаження операцій, тощо. Реалізована модель ООП (як і в JavaScript) — прототипна.

У Lua використовується динамічна типізація, мовні конструкції перетворюються на байт-код, який виконується поверх реєстрової віртуальної машини з автоматичним збирачем сміття. Сам інтерпретатор оформлений у вигляді бібліотеки, легко інтегрованої в проекти на мовах C та C++. Код інтерпретатора Lua написаний на мові C і розповсюджується під ліцензією MIT.

Для більш зручної розробки програмних засобів використовуються IDE (інтегровані середовища розробки). Інтегроване середовище розробки — це комплекс програмних засобів для розробки, що складається з текстового редактора коду, інструментів для компіляції та відлагодження проектів.

Для розробки програм для мікроконтролерів можна використовувати наступні середовища розробки: Proteus, Microsoft Visual Studio, HaikuVM, Eclipse, pyfirmata, pyserial, ArduinoPyApi, Arduino IDE; або будь який текстовий редактор: Notepad, Sublime Text, Notepad++, RemoteXY. Так середовища Microsoft Visual Studio та Eclipse є мультимовними.

У результаті проведеного огляду для розробки було обрано мову C++, що є основною в середовищі Arduino IDE. Це обгрунтовано потужністю цієї мови програмування саме для мікроконтролерів, великою кількістю бібліотек та зручністю роботи з контролерами і швидкістю розробки.

Найпростішим і найоптимальнішим варіантом вибору є середовище Arduino IDE та мова програмування Wiring яку підтримує воно і яка є діалектом мови C++, а також сервіс RemoteXY для створення програмних застосунків для смартфонів і айфонів. Детальніше описувались Arduino IDE та RemoteXY в підрозділі 1.3.

2 ПРОЕКТУВАННЯ АПАРАТНОЇ ЧАСТИНИ

2.1 Розробка структурної схеми SMART-системи управління термоустановкою

Основне призначення апаратної складової це керування потужністю, а отже кількістю теплоти, що виділяється, на пластинах модулів Пельтьє. Загальна електрична функціональна схема апаратної частини подана на рисунку 2.1. Керує всією цією схемою мікроконтролер NodeMCU V3, загальні відомості про якого наведені в пункті 1.4.

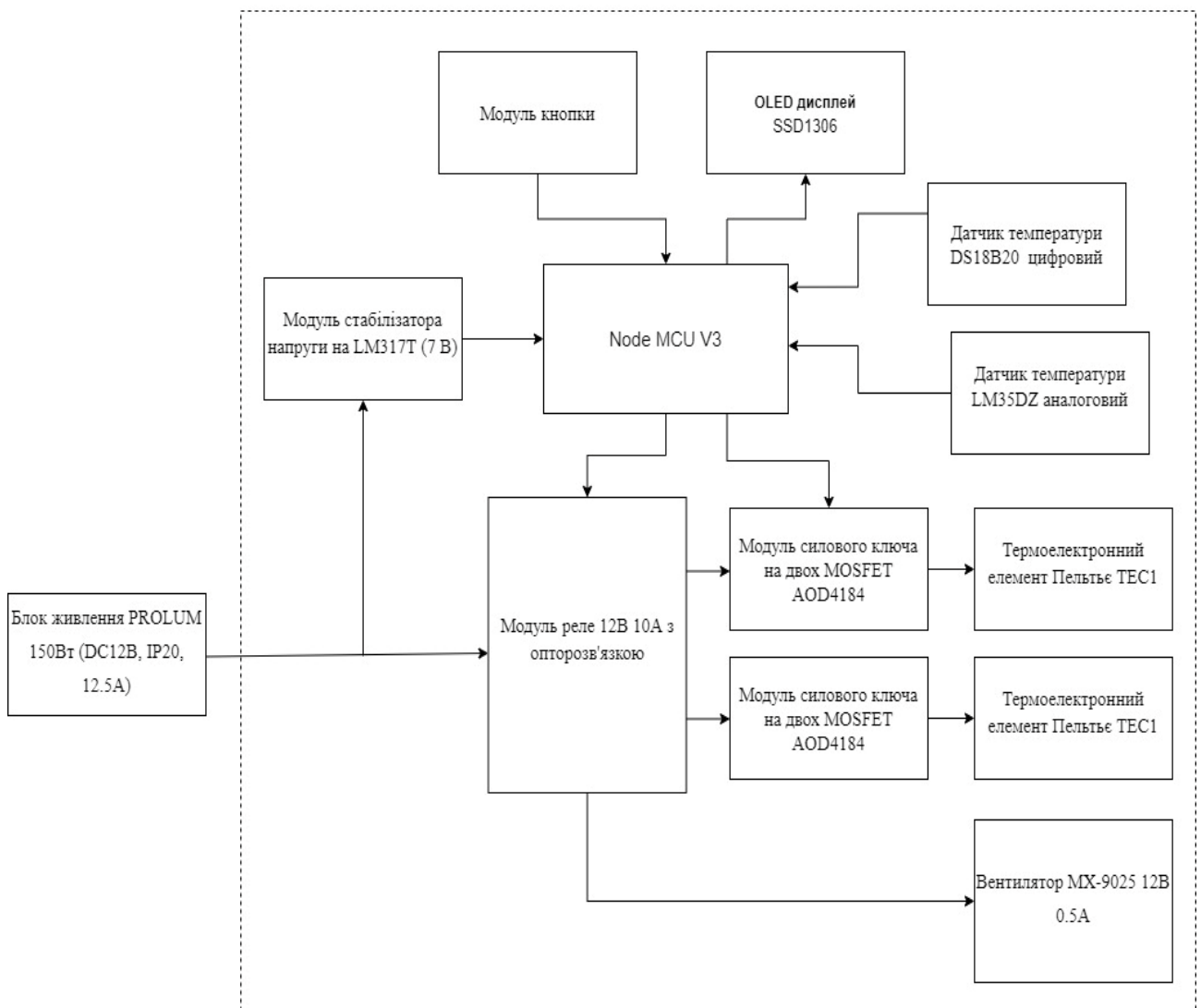


Рисунок 2.1 — Структурна схема апаратної частини

2.2 Визначення моделей компонентів системи

Розглянемо характеристики основних елементів цієї схеми, що живиться від джерела 12 В.

Модуль Пельтьє TEC1-12706. Модуль використовується як охолоджувач чи нагрівач в автомобілях, переносних міні холодильниках, автомобільних міні холодильниках, термобоксах, термокулерах, осушувачах повітря, портативних холодильниках також для охолодження процесорів, відеокарт та чіпів(рисунок 2.2) [17].



Рисунок 2.2 — Модуль Пельтьє TEC1-12706

Також може використовуватись як генератор енергії: якщо на двох сторонах елемента різна температура, ця різниця перетворюється в енергію на виходах. При використанні елемента для охолодження, варто пам'ятати про обмежену можливість радіатора гарячої частини віддавати тепло в навколишнє середовище. Для кращого ефекту на холодній частині в більшості випадків треба зменшувати потужність живлення до такої, яку може розсіяти радіатор гарячої сторони.

Характеристики:

- живлення від 0 В до 15 В;
- струм від 0 А до 6 А;
- потужність 60 Вт (Макс. 70 Вт);
- робочий діапазон температур від мінус 30 °С до плюс 70 °С;

- джерело живлення: постійної напруги або акумулятор;
- максимальна температура корпусу модуля: до +90 градусів;
- різниця температур на стінках модуля: може досягати 25 градусів (Не плутати з температурою всередині самого холодильника);
- матеріал: керамічні пластини білого кольору;
- кількість рп переході 127 штук;
- розмір 40мм x 40мм x 4 мм;
- маса близько 23 грам.

Полярність підключення може бути будь-якою. Якщо підключати червоний до плюса, чорний до мінуса, холодитиме з боку напису, а з іншого боку, елемент буде нагріватися, обов'язково має використовуватися термопаста.

Залежно від якості, термомодуль Tec1-12706 може замінювати на Tec1-12703, Tec1-12704, Tec1-12705, Tec1-12708, Tec1-12710, Tec1-12715 [17].

Міні-модуль потужного реле 5В 30А з опторозв'язкою. Компактний модуль потужного реле з можливістю вибору рівня керуючого сигналу (рисунок 2.3). При невеликих розмірах модуль дозволяє комутувати навантаження зі струмом до 30А. При цьому для управління реле досить 5 мА струму керуючого сигналу. Для захисту входу і обмеження струму через оптрон гальванічної розв'язки на платі встановлений струмообмежувальний резистор.

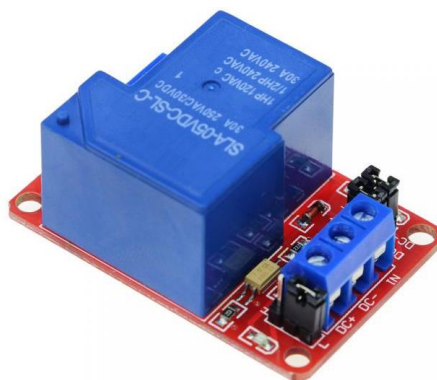


Рисунок 2.3 — Міні-модуль потужного реле 5В 30А з опторозв'язкою

Контакти модуля:

- «DC →» — вхід негативного виводу джерела живлення;

- «DC+» — вхід позитивного виводу джерела живлення;
- «HI-LO» — перемикач вибору рівня керуючого сигналу;
- «IN1» — вхід керуючого сигналу.

Для керування модулем сигналом високого логічного рівня необхідно переставити перемикач в положення H x (2-3 Піни). Для керування сигналом низького логічного рівня необхідно переставити перемикач в положення L x (1- 2 Піни).

Також на платі присутні дві перемикачі (JD — DC+ і JD — DC-) повної гальванічної ізоляції модуля.

Характеристики:

- напруга живлення 5В;
- максимальний струм навантаження реле 30А;
- струм сигналу керування 5мА;
- розмір модуля 50 мм x 33 мм x 24 мм [18].

Стабілізатор напруги LM317T. Модуль живлення з регульованим стабілізатором напруги призначений для побудови трансформаторних джерел живлення зі стабільною вихідною напругою (рисунок 2.4). Завдяки використанню стабілізатора напруги LM317 вихідна напруга стабілізована з високою точністю. При використанні модуля необхідно враховувати, що різниця напруг вхідного і вихідного розсіюється у вигляді тепла, яке розсіюється радіатором. Так як розміри радіатора невеликі то і вихідний струм, при великій різниці напруг, повинен бути невеликий. При необхідності підключення більш потужного навантаження необхідно використовувати радіатор більшого розміру і забезпечити його гарне охолодження [19].

Характеристики:

- стабілізатор напруги: LM317T;
- вхідна напруга: від 12В до 24В постійного або змінного струму;
- вихідна напруга: 9В;
- максимальний вихідний струм: 1А (при використанні додаткового охолодження);

— розміри плати: 49мм x 26 мм.

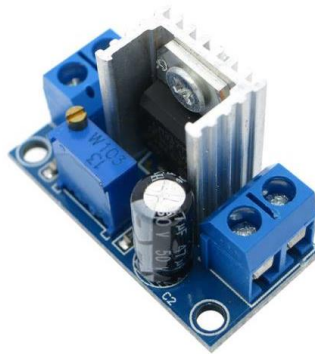


Рисунок 2.4 — Стабілізатор напруги LM317T

Модуль силового ключа на двох MOSFET AOD4184. Силовий модуль на двох польових транзисторах AOD4184 призначений для комутації навантаження з максимальною напругою до 36В і струмом до 15А при кімнатній температурі (рисунок 2.5). Для струмів навантаження більших 15А і напрузі керування менше 4В необхідне додаткове примусове охолодження модуля [20].



Рисунок 2.5 — Модуль силового ключа на двох MOSFET AOD4184

Модуль розрахований на управління навантаженням за допомогою ШІМ регулювання. Для зручності контролю сигналу управління на модулі встановлено індикаторний світлодіод [19].

При керуванні соленоїдами і електромагнітами потрібна обов'язкова установка захисного діода паралельно обмотці електромагніту в зустрічному включенні. Напруга управління подається на вхід TRIG / PWM силового модуля.

Характеристики:

- транзистор модуля AOD418;
- кількість транзисторів модуля: 2;
- максимальна напруга живлення: 40В;
- максимальний струм транзистора: 50А;
- опір відкритого каналу: при напрузі на затворі 4,5 В — менше 11МОм, при напрузі на затворі 11В — менше 8МОм.
- максимальна частота ШІМ управління: 20КГц;
- максимальна напруга управління: 20В;
- максимальна потужність комутації: 400Вт;
- робочий діапазон температур: від мінус 40 до плюс 85 градусів;
- розміри модуля: 34мм x 17мм x 17 мм.

Датчик температури DS18B20. Датчик температури DS18B20 в захисному водонепроникному корпусі з пиловологозахистом IP67 (рисунок 2.6). Діапазон температур, вимірюваних датчиком знаходиться в межі від мінус 55° С до плюс 125° С. Але так як захисна оболонка датчика зроблена з ПВХ, то рекомендується верхній діапазон виміру обмежити ста градусами. Сам вимірювальний елемент DS18B20 розміщений в герметичному пиловолого захищеному корпусі, що забезпечує максимальний ступінь захисту датчика і дозволяє проводити вимірювання температури в будь-яких умовах вологості, запиленості, а також при повному зануренні датчика в рідину.

Характеристики:

- червоний дріт — VCC (живлення);
- зелений (синій, жовтий) провід — DATA (дані);
- жовтий (чорний, чорний) провід — GND (земля);
- робоча напруга дані/живлення від 3В до 5.5В;
- точність ± 0.5 ° С в діапазоні від мінус 10 ° С до плюс 85 ° С;
- робочий діапазон температур від мінус 55 до плюс 125 ° С;
- вибір 9 чи 12 бітної розрядності;
- інтерфейс 1 — Wire;

Рисунок 2.7 — Датчик температури LM35DZ аналоговий

Яскравий, економічний, дуже контрастний OLED дисплей, контрастність дисплея дозволяє впевнено зчитувати з нього інформацію навіть при дуже яскравому світлі (рисунок 2.8). Просте підключення і поширений I2C інтерфейс дозволяє підключити декілька дисплеїв до будь — якого мікроконтролера або міні — комп'ютера.

Характеристики:

- розмір екрану 0.91мм;
- тип екрану OLED;
- роз'єм 4 pin;
- напруга живлення від 3.3В до 6В;
- роздільна здатність дисплея 128мм x 32мм;
- товщина 12 мм (з роз'ємом);
- ширина 12 мм;
- довжина 38 мм;
- вага 3.0 грам.



Рисунок 2.8 — OLED дисплей 0.91 "I2C 128x32

Специфікація:

- драйвер OLED модуля: SSD1306;
- кут огляду: > 160 градусів;
- роз'єм: 4 pin;

- напруга живлення: від 3.3В до 6В;
- рівні входних сигналів: 3.3В / 5В.

Цоколювання роз'єму:

- GND — загальний;
- VCC — напруга живлення;
- SDA — шина даних;
- SCL — шина тактування даних.

Модуль тактової кнопки з підтягуючим резистором та контактним роз'ємом (рисунок 2.9). Більш зручний, ніж просто кнопка для макетування, зважаючи на більший розмір кнопки та ковпачка, але при цьому зберігає місце на макетній платі.



Рисунок 2.9 — Модуль кнопки

Характеристики:

- діапазон робочих температур від мінус 20°C до плюс 60°C;
- опір контактів — 50 мОм;
- максимальна напруга — 100В;
- ліміт натискань — 10 000.

Вентилятор MX-9025. Невеликий але продуктивний кулер (вентилятор) типорозміру 6015 та напругою живлення 12В (рисунок 2.10). Відрізняється непоганою продуктивністю при відносно низькому рівні шуму, здвоєний (два вентилятори на одному валу, один для охолодження та відтоку тепла, інший для швидшої конвекції холодних мас у камері).

Характеристики:

- вентилятор корпусний;

- тип вентилятора: здвоєний шарикопідшипник;
- швидкість вентилятора 5200 об / хв;
- сила струму: 0.5А;
- потужність (Вт): 5 Вт;
- шум: 28.7дВА;
- розміри вентилятора: 92мм x 92мм x 25 мм;
- роз'єм: 2 пін;
- живлення: 12 В.



Рисунок 2.10 — Вентилятор МХ-9025

2.3 Макетування системи

Після вибору відповідних елементів апаратної частини почався монтаж елементів Пельтьє на радіатор та подальше підключення спочатку на безпайній платі, згодом в корпус самої установки (рисунок 2.11).

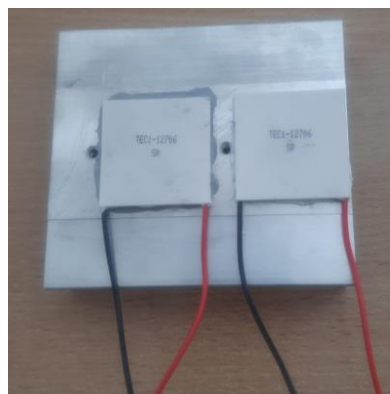


Рисунок 2.11 — Монтаж елементів Пельтьє на радіатор

Пробне підключення елементів апаратної частини на основі безпайної плати (рисунок 2.12).

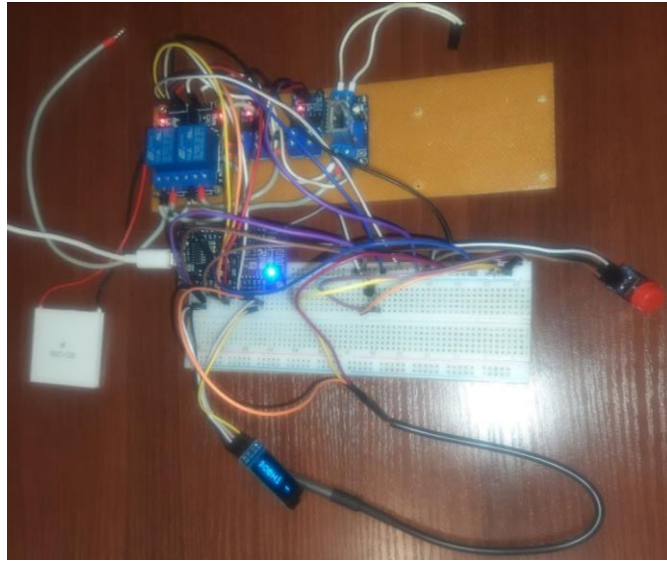


Рисунок 2.12 — Пробне підключення елементів апаратної частини на основі безпайної плати

Монтаж елементів у корпус кришки термоустановки (рисунок 2.13).



Рисунок 2.13 — Монтаж елементів у корпус кришки термоустановки

Після збору усієї установки можна приступати до тестового підключення та розробки програмного забезпечення.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ПЛАТФОРМИ

3.1 Розробка основного алгоритму роботи

Керування установкою здійснюється вручну (кнопкою) або за допомогою Android застосунку (рисунок 3.1). Застосунок містить 3 сторінки, які в свою чергу містять елементи керування та мітки (надписи).

Програмне забезпечення, що керує установкою завантажується в контролер NodeMCU V3, та використовує 7 зовнішніх бібліотек, які частково пов'язані між собою (рисунок 3.2). Поля структури RemoteXY, що реалізують елементи інтерфейсу користувача застосунку (рисунок 3.3).

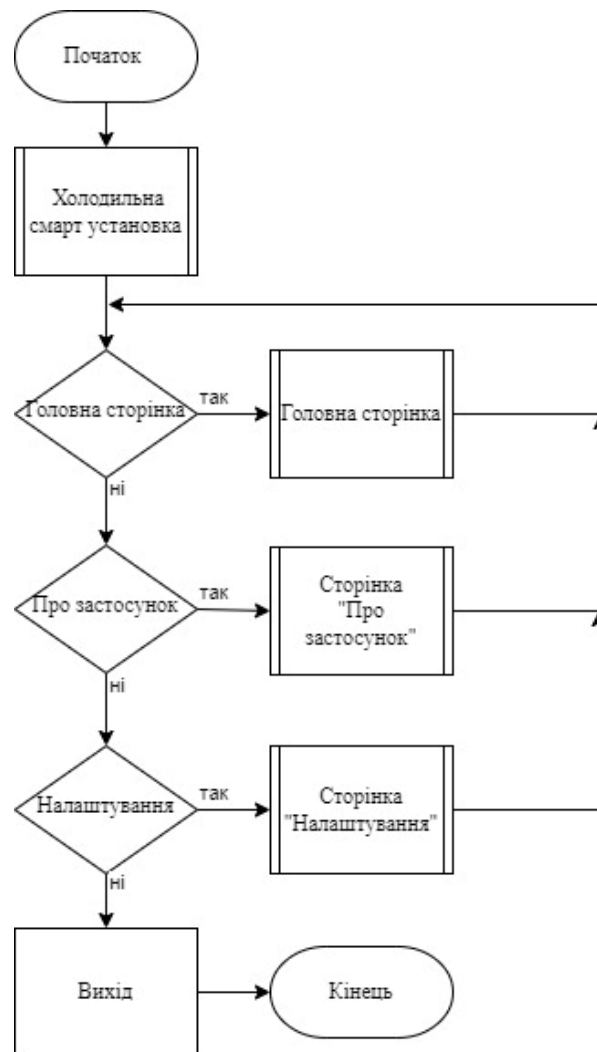


Рисунок 3.1 — Структура роботи Android застосунку

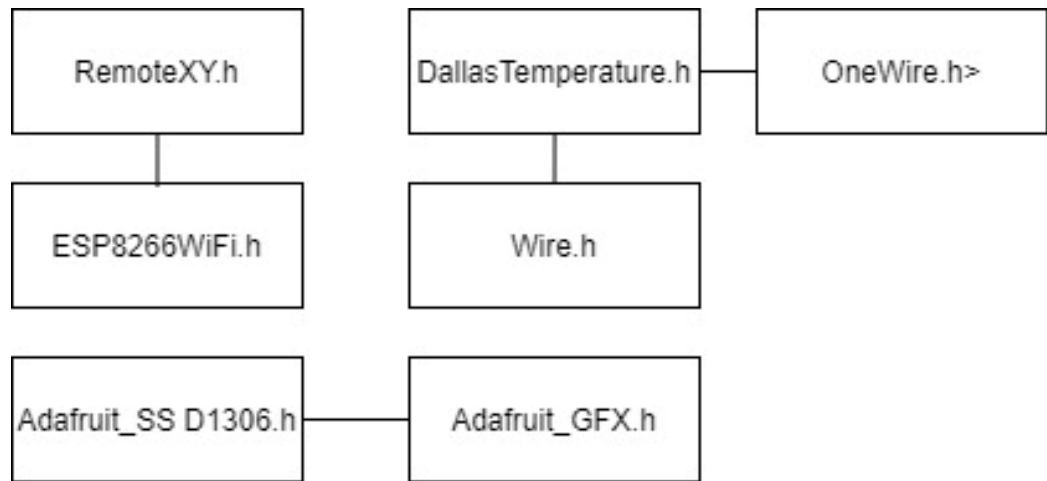


Рисунок 3.3 — Зовнішні бібліотеки та їх зв'язки

RemoteXY
+ slider_11 : int8_t
+ switch_11 : uint8_t
+ slider_21 : int8_t
+ slider_22 : int8_t
+ select_21 : uint8_t
+ select_22 : uint8_t
+ pushSwitch_21 : uint8_t
+ onlineGraph_1_var1 : float
+ onlineGraph_1_var2 : float
+ text_11 : char[11]
+ text_12 : char[11]
+ text_14 : char[11]
+ text_21 : char[16]

Рисунок 3.3 — Діаграма структури RemoteXY

Структурна схема програмного забезпечення застосунку, містить 5 функцій (рисунок 3.4). Так функція `String utf8rus(String source)` відповідає за зміну нумерації символів, оскільки таблиці символів, екрану і середовища програмування різні. Точкою входу у програму є функція `void setup()`, де встановлюються всі початкові налаштування. Після чого керування програмою передається функції `void loop()`, яка працює в безкінечному циклі і з якої викликаються функція `void buton(int pin)` — обробка нажимання кнопки, та функція `void display()`, що відповідає за вивід інформації на екран.

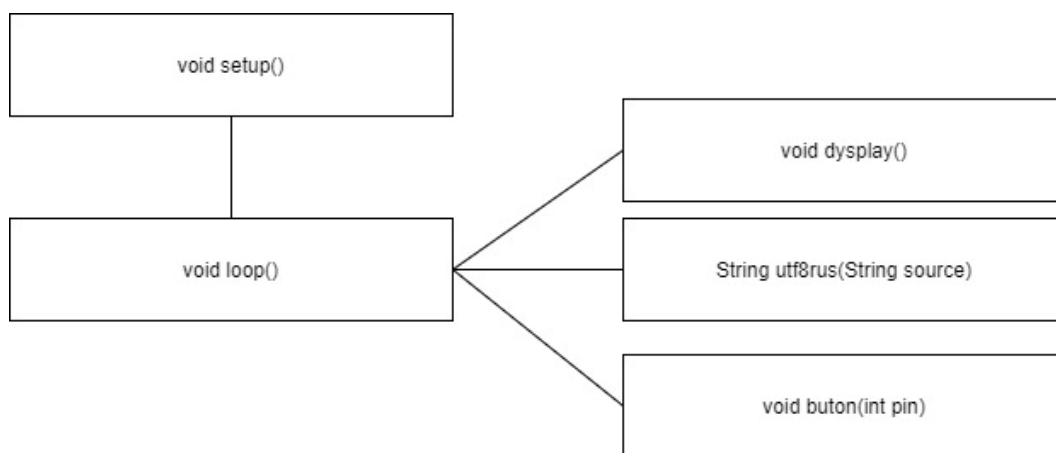


Рисунок 3.4 — Структурна схема програмного забезпечення застосунку

3.2 Проектування роботи сервера

Для ефективної розробки програмного забезпечення розділимо його на наступні складові:

- складання схеми апаратної частини та підключення всіх елементів до контролера;
- моделювання інтерфейсу користувача програмного застосунку для Android в сервісі RemoteXY;
- розробка програми в середовищі Arduino IDE на основі програмного коду сервісу RemoteXY.

Щодо підключення контролера, спеціально на самий початок скетча винесено декларації опису констант для підключення пінів NodeMCU до інших пристроїв системи з відповідними коментарями.

```

// Підключення дисплею SSD1306 через I2C (SDA, SCL pins) D1,D2
#define PIN_T1 A0 // пін аналогового датчика температури (зовнішня)
#define PIN_T2 D3 // пін цифрового датчика температури (внутрішня)
#define PIN_P1 D5 // пін регулювання потужності 1 ЕП (елемента
Пельтьє)
#define PIN_P2 D6 // пін регулювання потужності 2 ЕП (елемента
Пельтьє)
#define PIN_Rele1 D7 // реле1
  
```

```
#define PIN_K D8 // кнопка
```

Після основних налаштувань спроектовано головну сторінку застосунку, копію вікна браузера з сервісом RemoteXY (рисунок 3.1).

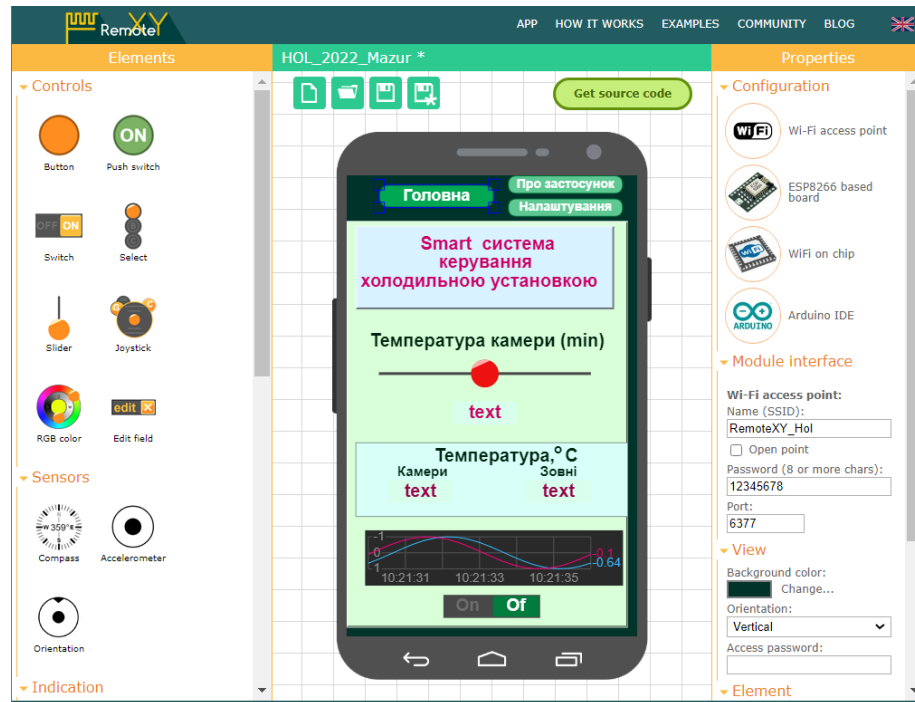


Рисунок 3.5 — Проектування головної сторінки застосунку в сервісі RemoteXY

Наступним етапом розробки є створення дизайну застосунку за допомогою сервісу RemoteXY. Спочатку налаштовуються конфігурація з'єднання смартфона з NodeMCU. Ми налаштували роботу NodeMCU як точки доступу WIFI, можна було змінити налаштування зв'язку на одну із інших, наприклад робота на сервері у хмарі (рисунок 3.2).

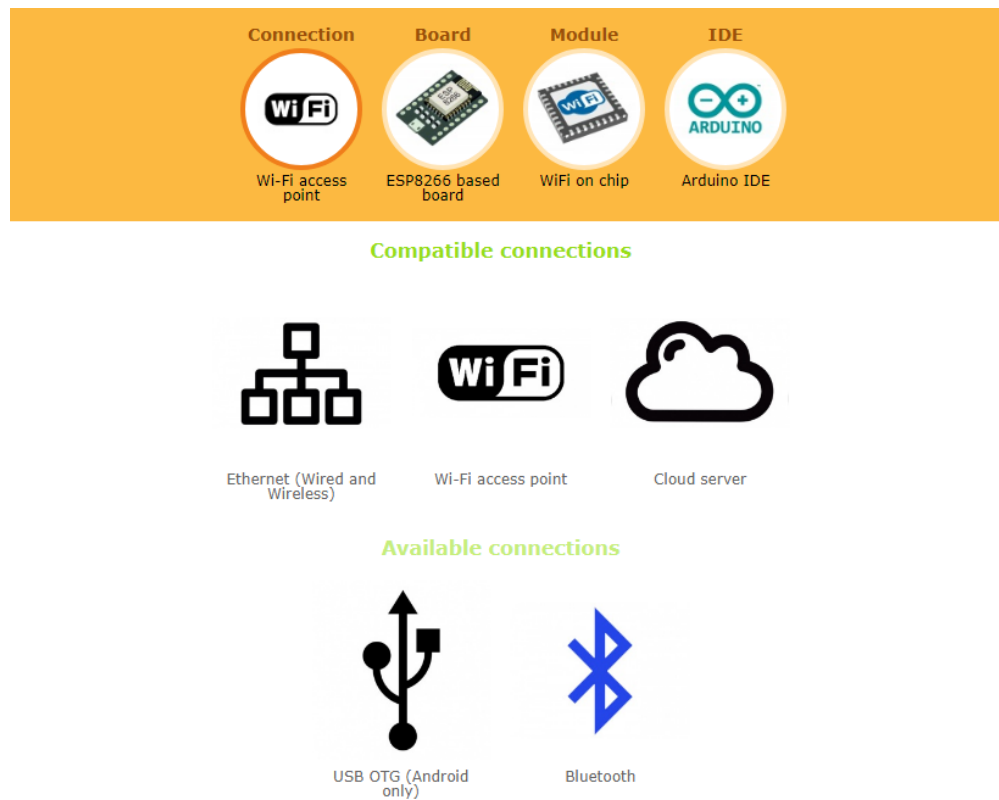


Рисунок 3.6 — Налаштування для комутації застосунку в сервісі RemoteXY

Основні налаштування для комутації відображені у програмному коді так:

- `#define REMOTEXY_WIFI_SSID "RemoteXY_Hol";`
- `#define REMOTEXY_WIFI_PASSWORD "12345678";`
- `#define REMOTEXY_SERVER_PORT 6377.`

По завершенні проектування можна отримати заготовку програмного коду з масивом даних про інтерфейс користувача клацнувши кнопку «Get Source Code» (рисунок 3.3).

Source code of project: HOL_2022_Mazur

1. Download the source code of the program, open it in the Arduino IDE.
2. Install **RemoteXY library** for Arduino IDE.
3. Compile the source code and upload it to the ESP8266 board using the Arduino IDE.
4. Install the mobile app **RemoteXY ver.4.11.1** for smartphone/tablet.
5. Connect to ESP8266 using mobile app.

[project.ino](#) [Download code](#) [Download library](#)

```

/*
  -- HOL_2022_Mazur --

  This source code of graphical user interface
  has been generated automatically by RemoteXY editor.
  To compile this code using RemoteXY library 3.1.8 or later version
  download by link http://remotexy.com/en/library/
  To connect using RemoteXY mobile app by link http://remotexy.com/en/download/
  - for ANDROID 4.11.1 or later version;
  - for iOS 1.9.1 or later version;

  This source code is free software; you can redistribute it and/or
  modify it under the terms of the GNU Lesser General Public
  License as published by the Free Software Foundation; either
  version 2.1 of the License, or (at your option) any later version.
*/

////////////////////////////////////
// RemoteXY include library //
////////////////////////////////////

```

Рисунок 3.7 — Отримання заготовки програмного коду в сервісі RemoteXY

3.3 Програмування мікроконтролера

Отримавши програмний код та завантаживши його в Arduino IDE маємо основну структуру «RemoteXY», що описує елементи інтерфейсу.

```

struct {
  // input variables
  int8_t slider_11; // =-100..100 slider position
  uint8_t switch_11; // =1 if switch ON and =0 if OFF
  int8_t slider_21; // =0..100 slider position
  int8_t slider_22; // =0..100 slider position
  uint8_t select_21; // =0 if select position A, =1 if position B, =2 if position C, ...
  uint8_t select_22; // =0 if select position A, =1 if position B, =2 if position C, ...
  uint8_t pushSwitch_21; // =1 if state is ON, else =0

```

```

// output variables
float onlineGraph_1_var1;
float onlineGraph_1_var2;
char text_11[11]; // string UTF8 end zero
char text_12[11]; // string UTF8 end zero
char text_14[11]; // string UTF8 end zero
char text_21[16]; // string UTF8 end zero
char text_22[31]; // string UTF8 end zero
// other variable
uint8_t connect_flag; // =1 if wire connected, else =0
} RemoteXY;

```

Оголошуємо основні об'єкти класів (що описані в зовнішніх бібліотеках), що відповідатимуть за роботу з цифровим датчиком температури і дисплею SSD1306 відповідно.

```

OneWire oneWire_out(PIN_T2);
DallasTemperature sensor_outhouse(&oneWire_out);
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -
1);

```

Для керування і збору та виводу інформації в застосунок Android та на екран використовуються також змінні.

```

double tempoutDOUBLE = 0; //змінна з температурою для 18B20
float temp1, temp2; // значення температури всередині та зовні камери
float t; // граничні значення температури
int p1, p2; // значення потужності елементів Пельтьє №1 та №2
int r1 = 0, r2 = 1, r3 = 1, r11 = 1;
int buttonState; // стан кнопки
int lastButtonState = LOW; // попередній стан кнопки
unsigned long lastDebounceTime = 0; // час нажаття кнопки
unsigned long debounceDelay = 50; // Час затримки для перевірки стану

```

КНОПКИ

int n = 0 , m = 140; // режим, номер символу (біжучий рядок) ,

Функція utf8rus перекодування символів кирилиці з UTF-8 в Win-1251, без якої всі символи кирилиці будуть на екрані висвітлюватись іншими знаками.

```
String utf8rus(String source)
```

```
{ int i, k; String target; unsigned char n;
```

```
char m[2] = { '0', '\0' };
```

```
k = source.length(); i = 0;
```

```
while (i < k) { n = source[i]; i++;
```

```
if (n >= 0xC0) {
```

```
switch (n) { case 0xD0: { n = source[i]; i++;
```

```
if (n == 0x81) { n = 0xA8; break; }
```

```
if (n >= 0x90 && n <= 0xBF) n = n + 0x30; break; }
```

```
case 0xD1: { n = source[i]; i++;
```

```
if (n == 0x91) { n = 0xB8; break; }
```

```
if (n >= 0x80 && n <= 0x8F) n = n + 0x70; break; }
```

```
} } m[0] = n; target = target + String(m); }
```

```
return target;
```

```
}
```

Функція що програмно знімає бразкіт контактів ,я а також визначає стан нажимання на кнопку є функція void buton(int pin).

```
void buton(int pin)
```

```
{int reading = digitalRead(pin);
```

```
if (reading != lastButtonState) { lastDebounceTime = millis(); }
```

```
if ((millis() - lastDebounceTime) > debounceDelay) {
```

```
if (reading != buttonState) { buttonState = reading;
```

```
if (buttonState == HIGH) { r3++; if (r3 > 4) r3 = 1; } } }
```

```
lastButtonState = reading;
```

```
}
```

Для виводу інформації на екран та керування екраном, що може працювати в 3 режимах слугує функція void dysplay().

```

void display()
{ char ss1[5]; char ss2[5]; char ss3[3];
  dtostrf(temp2, 4, 1, ss1); dtostrf(temp1, 4, 1, ss2);  dtostrf(r11, 0, 0, ss3);
  if (r2 == 0) // реалізація виводу тексту у 3 рядки
  { display.clearDisplay();
    display.setTextColor(WHITE);
    display.setTextSize(1);  display.setCursor(8, 0);
    display.print(utf8rus("Темп. камери-")); display.println(ss1);
    display.setCursor(10, 10);
    display.print(utf8rus("Темп. ззовні-")); display.println(ss2);
    display.setCursor(35, 20);
    display.print(utf8rus(" Режим-")); display.println(ss3);
    display.display();
  }
  if (r2 == 1) // реалізація виводу тексту у формі біжучого рядка
  { ... }
  if (r2 == 2) // очистка екрану
  { display.clearDisplay();  display.display(); }
}

```

Точкою входу програми і встановлення початкових значень та налаштувань є функція `setup()`, після виконання якої керування програмою переходить до функції `void loop()`

```

void setup()
{ RemoteXY_Init ();
  Serial.begin(9600);
  pinMode (PIN_P1, OUTPUT);
  pinMode (PIN_P2, OUTPUT);
  pinMode (PIN_Rele1, OUTPUT);
  pinMode (PIN_K, OUTPUT);
}

```



```

    currentTime = millis();          // считываем время, прошедшее с момента
запуска программы

```

```

    loopTime = currentTime; sensor_outhouse.begin();
    Serial.begin(115200);
    if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
// Address 0x3D for 128x64
        Serial.println(F("SSD1306 allocation failed"));
    }
    display.cp437(true); display.clearDisplay();
    temp1 = temp2 = t = 20;
}

```

Функція `void loop()` працює в безкінечному циклі та з якої викликаються інші функції, що описані вище. Її можна поділити на 4 фрагменти:

а) виклик зовнішніх функцій;

```

void loop()
{
    RemoteXY_Handler ();
    buton(PIN_K); //// кнопка ////
    display();
    ....
}

```

б) кожні 5 с відбувається збір інформації з датчиків і автоматичне керування модулем реле;

```

if (currentTime >= (loopTime + 5000))
{
    sensor_outhouse.requestTemperatures();
    temp2 = sensor_outhouse.getTempCByIndex(0);
    temp1 = analogRead(PIN_T1);
    temp1 = (temp1 / 1023.0) * 3.0 * 1000 / 10;
    if (( t > temp2 ))// && (RemoteXY.switch_11))
    { digitalWrite(PIN_Rele1, LOW); }
}

```

```

if (( t < temp2 ))// && (RemoteXY.switch_11))
{   digitalWrite(PIN_Rele1, HIGH);   }
if (RemoteXY.switch_11)
{   digitalWrite(PIN_Rele1, LOW);   }
////////////////////////////////////
loopTime = currentTime;
}
if (RemoteXY.switch_11)
{   digitalWrite(PIN_Rele1, LOW);   }
if ((!RemoteXY.switch_11)&& ( t < temp2 ))
{   digitalWrite(PIN_Rele1, HIGH);   }

```

в) вивід інформації в додаток на смартфон;

```

RemoteXY.onlineGraph_1_var1 = (float) temp2;
RemoteXY.onlineGraph_1_var2 = (float) temp1;
t = map(RemoteXY.slider_11, -100, 100, 0, 25);
dtostrf(t, 0, 0, RemoteXY.text_14);
dtostrf(temp2, 0, 2, RemoteXY.text_11);
dtostrf(temp1, 0, 2, RemoteXY.text_12); //temp2
r1 = RemoteXY.select_21;
dtostrf(r1, 0, 0, RemoteXY.text_21);
r2 = RemoteXY.select_22;
switch (r2) {
    case 0: sprintf(RemoteXY.text_22, "Статичний текст"); break;
    case 1: sprintf(RemoteXY.text_22, "Біжучий рядок"); break;
    case 2: sprintf(RemoteXY.text_22, "Вимкнений екран"); break;
}

```

Г) Програмне керування потужністю модулів силового ключа на двох MOSFET AOD4184 в залежності від положення повзунків slider_21 та slider_22.

```

if (RemoteXY.pushSwitch_21 == 1)
{

```

```
p1 = 255 - map( RemoteXY.slider_21, 0, 100, 0, 255);  
p2 = 255 - map( RemoteXY.slider_22, 0, 100, 0, 255);  
analogWrite(PIN_P1, map( p1, 0, 100, 0, 255));  
analogWrite(PIN_P2, map( p2, 0, 100, 0, 255));  
}
```

3.4 Тестування програмного застосунку

Тестування являє собою невід’ємну частину розробки сучасних програмних продуктів. Для IoT сервісів, які використовують апаратну частину (датчики і модулі), цей етап є ще важливішим. Перевірка коректності отримання даних, їх обробка і реакція керуючих модулів являють основну задачу при тестуванні. Ціллю процесу тестування є забезпечення якості ПЗ та наявності найбільш малої кількості багів, адже їх критична наявність може привести до виходу з ладу всієї системи.

Методика тестування також включає в себе процес пошуку дефектів, помилок, несправностей. Також це є випробуванням програмних складових з метою оцінити готовність програмного продукту до використання. Результат тестування оцінюється за наступними критеріями:

- відповідність вимогам, які надавалися розробником та проектувальником;
- відповідність вихідних даних;
- прийнятний час виконання функцій;
- практичність;
- відповідність вимогам замовника.

Кількість тестів навіть для простих програмних компонентів може бути ледь не нескінченним, тому тактика тестування має полягати в тому, що будуть проведені тільки необхідні тести з урахуванням доступного часу та ресурсів. Як результат, програмні засоби тестуються стандартним виконанням програми з метою виявлення багів, помилок або інших дефектів.

Тестування системи — це виконання програмного забезпечення в його остаточній конфігурації, інтегрованого з іншими програмними та апаратними системами.

Є дві основних методики тестування: тестування «білої скриньки» та тестування «чорної скриньки».

Методика «білої скриньки», Об'єктом тестування тут є не зовнішня, а внутрішня поведінка програми. Перевіряється коректність побудови всіх елементів програми та правильність їхньої взаємодії один з одним. Зазвичай аналізуються керуючі зв'язки елементів, рідше—інформаційні зв'язки. Тестування за принципом «білої скриньки» характеризується ступенем, в якому тести виконують або покривають логіку (вихідний текст) програми.

Одним із способів вивчення поставленого питання є дослідження методики «чорної скриньки», Основне місце програми тестів «чорної скриньки» це інтерфейс ПЗ. Ці тести демонструють:

- виконання функції програми;
- приймання вихідних даних;
- демонстрування результату;
- зберігання цілісності зовнішньої інформації.

При тестуванні «чорної скриньки» розглядаються системні характеристик и програм, ігнорується їхня внутрішня логічна структура. Вичерпне тестування, як правило, неможливе. Наприклад, якщо в програмі 10 вхідних величин і кожна приймає по 10 значень, то кількість тестових варіантів становитиме 1010. Тестування «чорної скриньки» не реагує на багато особливостей програмних помилок.

Спочатку протестуємо нашу установку саме методом «білої скриньки», тобто ми маємо протестувати чи робота всіх датчиків і керуючих модулів є правильною, такою як проектувалась.

При ввімкненні системи всі елементи працюють як передбачалось. Ввімкнена установка, що живиться від блока живлення імпульсного 12В 200Вт. Подано фото екрану ввімкненої установки на початку тестування (рисунок 3.8).



Рисунок 3.8 — Фото ввімкненої установки на початку тестування

Режим 1 визначає $\frac{1}{4}$ потужності установки, виставлено по замовчуванню (рисунок 3.5). Показано копії екрану смартфона при тестуванні охолодження установки, зчитуванні даних з датчиків та побудови графіків, як видно з графіків і температури камери з часом температура холодильної камери опускається, тобто проєктована система працює як і передбачалось (рисунок 3.6).

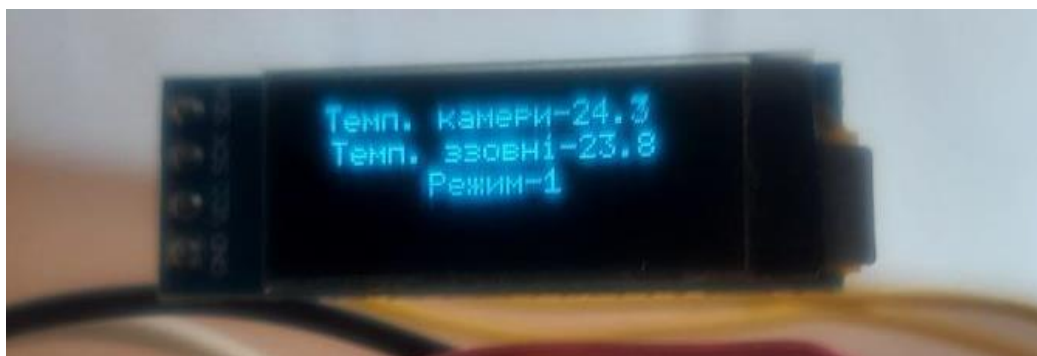


Рисунок 3.9 — Фото екрану ввімкненої установки на початку тестування

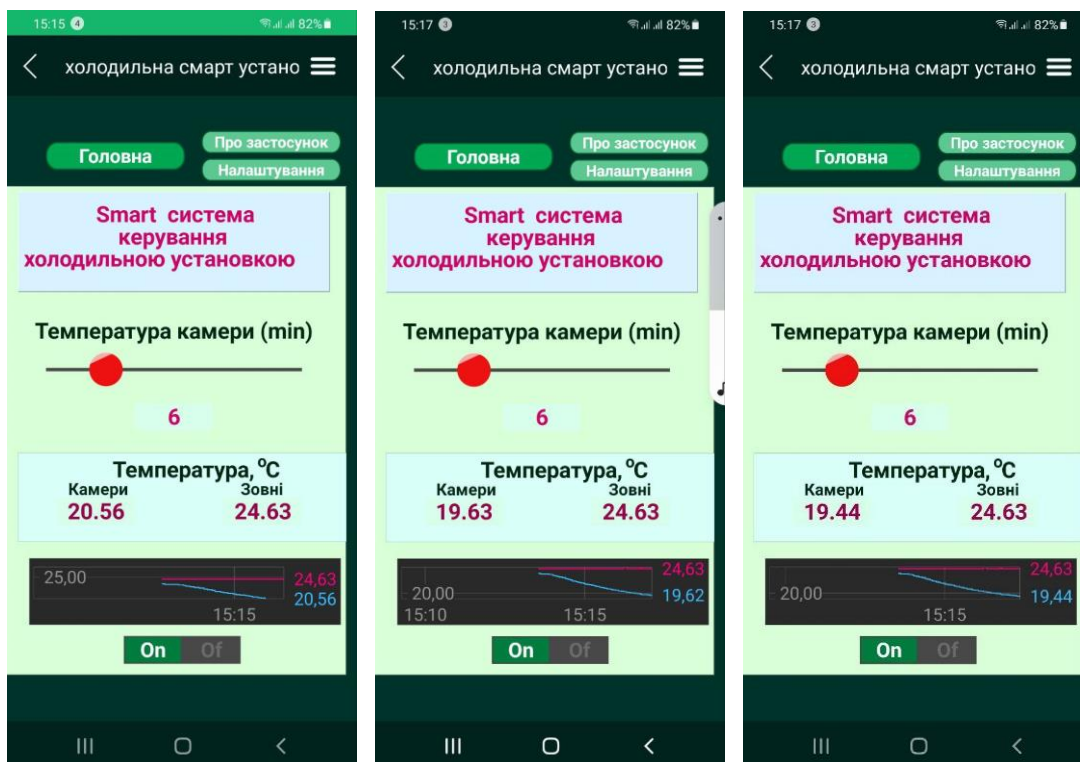


Рисунок 3.10 — Копії екрану смартфона при тестуванні охолодження установки в режимі 1 та зчитуванні даних з датчиків і побудови графіків

Подано фото екрану де температура охолодженої частини опустилась до $21,1^{\circ}\text{C}$, була 24°C (рисунок 3.11). Також для швидшого охолодження було обрано режим 4, що відповідає потужності 100 %, тобто ввімкнено обидва елементи Пельтьє на повну потужність.



Рисунок 3.11 — фото екрану при ввімкненні режиму 4 кнопкою та опусканні температури до $21,1^{\circ}\text{C}$, була 24°C .

Подано копії екрану смартфона при тестуванні режимів охолодження установки, коли вона керується з застосунку Android та режимів роботи екрану (рисунок 3.8). Так режимів роботи установки є 5:

- 1 — керування кнопкою;
- 2 — 25% потужності (перший модуль Пельтьє 50%, другий вимкнений);
- 3 — 50% потужності (перший модуль Пельтьє 100%, другий вимкнений);
- 4 — 75% потужності (перший модуль Пельтьє 100%, другий 50%);
- 5 — 100% потужності (перший модуль Пельтьє 100%, другий 100%).

При зміні режимів роботи екрану інформація вивидиться в залежності від режиму:

- 1 — статичний текст в 3 рядки;
- 2 — біжучий рядок, при цьому текстова інформація виводиться на екран в один рядок з розміром в 3 рази більшим;
- 3 — вимкнений екран.

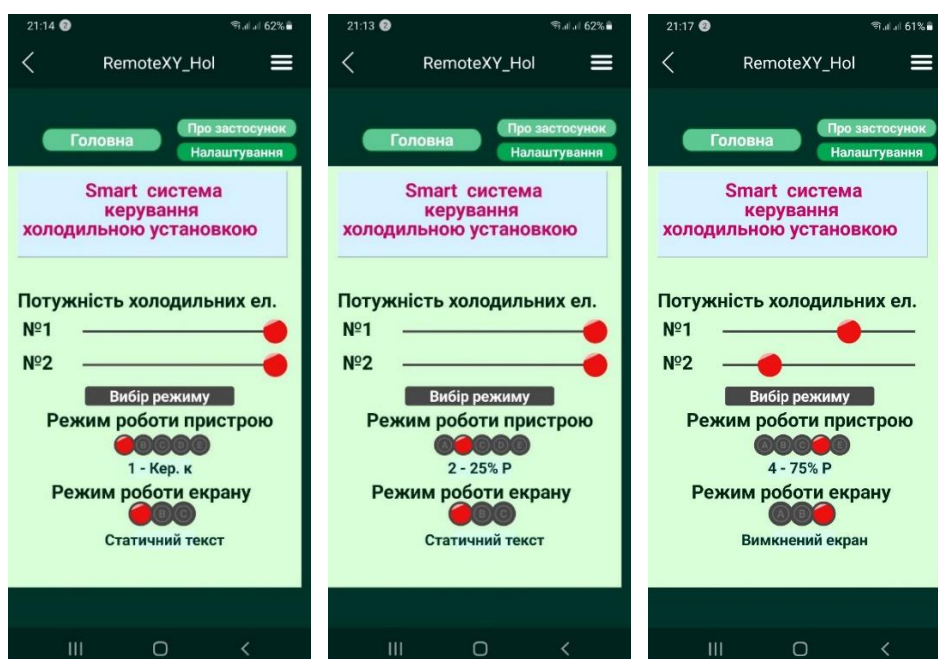


Рисунок 3.12 — Копії екрану смартфона при тестуванні режимів охолодження установки при керуванні з застосунку Android

Також показано тестування довільної потужності на кожному з елементів Пельтьє за допомогою повзунків (рисунок 3.13). Копія екрана смартфона при виборі сторінки «Про застосунок» (рисунок 3.14).

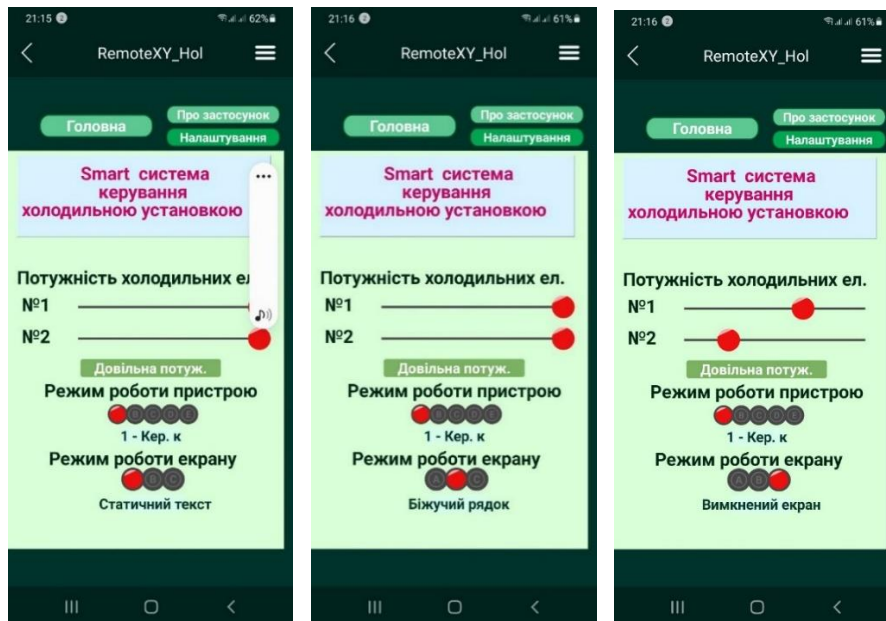


Рисунок 3.13 — тестування довільної потужності



Рисунок 3.14 — Копія екрана смартфона при виборі сторінки «Про застосунок»

Тестування програмних засобів буде доречно також провести використовуючи методику «чорної скриньки» (рисунок 3.15). Вона базується на використанні шаблонів тестування, або ж тест-кейсів. Це означає, що буде створено декілько ситуацій у яких перевірається працездатність системи, коректності її основних функцій.



Рисунок 3.15 — Фото зібраної установки при тестуванні методом чорної скриньки

Розроблені тест-кейси описано в таблиці 3.1.

Таблиця 3.1 — Тест-кейси

Код тест-кейса	Опис тест-кейса		
	Хід тестування		Очікуваний результат
	Дата тестування	Результат	Примітка
001	Перевірка ввімкнення та вимкнення установки		
	Ввімкнення установки та запуск застосунку на смартфоні. Ввімкнення/вимкнення установки за допомогою застосунку. Установка вмикається /вимикається, реле спрацьовує.	Коректне відображення значень температури на екрані смартфона та на OLED екрані. Система запустилась у режимі 1. Установка вмикається /вимикається, реле спрацьовує як і планувалось.	

Продовження таблиці 3.1

	23.05.2022	Пройдено	-
002	Зміна режиму роботи установки за допомогою кнопки на установці		
	1. Ввімкнення установки та запуск застосунку на смартфоні. 2. Послідовна установка режимів 2,3,4 за допомогою кнопки. 3. Перевірити роботу установки у режимі-4. Короткотривале охолодження.	Система запустилась у режимі 1. Система успішно переходить з одного режиму до наступного. Перехід в режим 4 (на повну потужність) На протязі 5хв. Температура камери опистилась на 1,5 °С, зовнішня не змінилась.	
	23.05.2022	Пройдено	-
003	Перевірка керування установкою за допомогою елементів керування сторінки «Головна»		
	1. При зміні мінімальної температури відносно температури зовнішньої, установка вмикається та вимикається 2. На графіку залежності температури камери і зовнішньої від часу відображаються 2 графіки. 3. Вивід значень температури з датчиків на смартфоні є коректним. 4. Кнопки «On» «Of» вмикають та вимикають установку.	При установці мінімальної температури камери нище зовнішньої установка ввімкнена, в іншому разі вимикається На графіку залежності температури камери і зовнішньої від часу відображаються синьою лінією графік температури камери і червоною графік залежності зовнішньої температури від часу. Коректний текстовий вивід значень температури. Установка вмикається /вимикається, реле спрацьовує.	

Продовження таблиці 3.1

	23.05.2022	Пройдено	-
004	Перевірка керування термоустановкою за допомогою елементів керування сторінки «Налаштування»		
	1. Ввімкнення кнопки «Довільна потужність», зміна значень повзунків №1 та №2.	Після переведення установки у стан «Довільна потужність» можна змінювати потужність установки за допомогою повзунків.	
	2. Ввімкнення кнопки «Вибір режиму», зміна значень повзунків №1 та №2 нідочого не призводить, вони вимкнулись. Зміна режиму перемикається за допомогою елемента інтерфейсу, що підписаний як «Режим роботи пристрою» 3. Преремикання режиму роботи екрану за допомогою перемикача «Режим роботи екрану» -положення 1; - положення 2; - положення 3.	При зменшенні загальної потужності до 0, установка вимикається, при збільшенні загальної потужності до 100% установка відчутно виробляє «холод» (знижує температуру однієї поверхні елементів Пельтьє). При зміні режиму за допомогою перемикача режимів відбувається перехід у відповідні режими: 1- керування кнопкою, 2-25% потужності, 3-50% потужності, 4-75% потужності, 5-100% потужності. Перехід від одного до наступного режиму відчутний по звуку кулера так і по кількості теплоти що виділяється зовні камери і кількості теплоти, що поглинається к камері. Зміна режиму роботи екрану здійснюється перемикачем «Режим роботи екрану» Статичний текст на екрані 3 рядки. Біжучий рядок на екрані біжучий рядок) Вимкнений екран. Екран не працює.	

3.5 Розробка інструкції користувача

Для початку роботи потрібно запуснути установку. Перед користувачем буде інтерфейс який складається з закладок 3 сторінок «Головна», «Налаштування», «Про програму». Інтерфейс є інтуїтивно зрозумілим.

«Головна сторінка дозволяє вмикати/вимикати установку та керувати встановленням мінімальної температури камери. На графіку відображаються значення температури камери та зовнішньої температури.

Сторінка «Налаштування дозволяє плавно змінювати потужність кожного елемента від мінімальної до максимальної та вибирати наперед визначенні режими потужності 25%, 50%, 75%, 100%. Також на сторінці «Налаштування» є можливість змінювати режим роботи екрану на: статичний текст, біжучий рядок, вимкнений екран.

ВИСНОВКИ

У ході виконання бакалаврської дипломної роботи було проведено розгляд мобільних систем охолодження. Розглянуто поняття інтернету речей та його складові, сучасні сервіси розробки IoT систем. Обґрунтовано вибір мови C++ і середовища програмування Arduino IDE.

Було розроблено структурну апаратної та програмної частин. Розглянуто характеристики усіх елементів апаратної частини. Проведено монтаж усіх елементів в корпус установки.

Було розроблено структурну апаратної та програмної частин. Розглянуто характеристики усіх елементів апаратної частини. Проведено монтаж усіх елементів в корпус установки.

Описано програмну реалізацію системи. Створено Android застосунок який надає користувачу змогу керувати установкою, також передбачено керування установкою за допомогою кнопки на корпусі.

Було розглянуто основні методики тестування, виконано тестування за методиками «білої» та «чорної скриньки», складено перелік тест-кейсів для тестування, а також розроблено інструкцію користувача.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Термопродукція. Електронний ресурс. Режим доступу: <https://rozetka.com.ua/ua/tourizm-freezers/c82682/>
2. Автохолодильники. Електронний ресурс. Режим доступу: <https://epicentrk.ua/ua/shop/avtokholodilniki/>
3. Технології інтернету речей. Навчальний посібник [Електронний ресурс]: навч. посіб. для студ. спеціальності 126 «Інформаційні системи та технології», спеціалізація «Інформаційне забезпечення робототехнічних систем» / Б. Ю. Жураковський, І.О. Зенів; КПІ ім. Ігоря Сікорського. – Електронні текстові дані (1 файл: 12,5 Мбайт). – Київ: КПІ ім. Ігоря Сікорського, 2021. – 271 с.
4. Програмно-апаратна технологія для навчання технологіям Інтернету Речей: навчальний посібник / [А.В. Пархоменко, А.В. Туленков, О.В. Соколянський, Я.І. Залюбовський]. – Запоріжжя: Дике Поле, 2017 – 120 с.
5. Q. Liu, X. Yang, L. Deng. An IBeacon-Based Location System for Smart Home Control // Sensors. – 2018.
6. IT-Enterprise. [Електронний ресурс]. Режим доступу: <https://www.it.ua/knowledge-base/technology-innovation/promyshlennyj-internet-veschej/>
7. Arduino + датчик DHT11 (влажність и температура). Електронний ресурс. Режим доступу: <http://smart.home.te.ua/?s=DHT11&submit=Search&lang=uk>. Дата доступу 11.04.2020.
8. Напівпровідникові датчики температури. Електронний ресурс. Режим доступу: <http://mcucpu.ru/index.php/pdevices/datchiki/108-poluprovodnikovye-datchiki-temperatury>. Дата доступу 04.04.2020
9. Топ-10 інструментів IoT-розробки. <https://www.telesphera.net/news/top-10-instrumentiv-iot-rozrobki.html>

10. Розанов І. Є., Сергієнко С. П., Чернов Д. В. Р 64 Методичні вказівки до виконання лабораторних робіт з курсу Інтернет речей. Вінниця : ДонНУ імені Василя Стуса, 2019. 60 с.
11. Петин В.А. Проекты с использованием контроллера Arduino. – БХВПетербург, 2014. – 400 с.
12. RemoteXY. Електронний ресурс. Режим доступу: <https://remotexy.com/ru/help/>.
13. Соммер У. Программирование микроконтроллерных плат Arduino/Freeduino – Санкт-Петербург.: БХВ-Петербург, 2014, 398с.
14. Огляд Arduino. Електронний ресурс. Режим доступу: <http://arduino.ua/ru/about/>.
15. D. A. Aziz. Webserver Based Smart Monitoring System Using ESP8266 Node MCUModule // International Journal Of Scientific & Engineering Research. 2018. – V. 10, № 6. – P. 801–807.
16. Засоби розробки на основі ESP8266 та ESP 32. <https://arduino.ua/cat122-esp8266>.
17. Термоелектронний кулер елемент Пельтье TEC1. Електронний ресурс. Режим доступу: <https://arduino.ua/prod1357-termoelektronnii-kyler-element-pelte-tec1-12706-12v>.
18. Міні-модуль потужного реле 5В 30А з опторозв'язкою. Електронний ресурс. Режим доступу: <https://arduino.ua/prod2577-mini-modyl-moshhnogo-rele-5v-30a-s-optorazvyazkoi>.
19. Перетворювачі напруги, стабілізатори, димери. Електронний ресурс. Режим доступу: <https://arduino.ua/cat362-preobrazovateli-napryajeniya-stabilizatori>.
20. Датчики. Температура, вологість. Електронний ресурс. Режим доступу: <https://arduino.ua/cat38-temperatyravljajnost>
21. Асемблер. Електронний ресурс. Режим доступу: <http://www.znannya.org/?view=asm>

ДОДАТОК А

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

ЗАТВЕРДЖУЮ

Завідувач кафедри ОТ ВНТУ

д.т.н., проф.

_____ О.Д. Азаров

“__” _____ 2022 р.

ТЕХНІЧНЕ ЗАВДАННЯ

на виконання бакалаврської дипломної роботи
«SMART- система управління термоустановкою на базі модуля ESP
8266»
08–23.БДР.007.00.000 ТЗ

Науковий керівник к.т.н. доц. каф. ОТ

_____ Томчук М.А.

Студент групи КІ–20мсз

_____ Мазур В.Ю.

Вінниця 2022

1 Підстави для виконання бакалаврської дипломної роботи (БДР) наступні:

— актуальність розробки, яка полягає у необхідності вирішення проблеми збільшення потужності та організації віддаленого керування мобільною холодильною установкою, що дозволять отримати додаток для вирішення проблеми, на основі якого буде можливість віддалено керувати установкою та її потужністю;

— наказ про затвердження теми бакалаврської дипломної роботи.

2 Мета і призначення БДР наступні:

— метою БДР є розробка програмного забезпечення для віддаленого керування холодильною установкою.

— призначення розробки полягає у виконанні бакалаврської дипломної роботи із подальшим впровадженням та розвитком.

3 Вихідні дані для виконання БДР наступні:

— технічний опис програмного застосунку;

— мова програмування C++;

— Android додаток;

— сервіс RemoteXY;

— середовище розробки Arduino IDE.

4 Вимога до виконання БДР наступна:

— створення Android додатку;

— можливість віддалено керувати холодильною установкою.

5 Етапи БДР та очікувані результати. Робота виконується за вісім етапів, таблиця А.1.

6 Матеріали, що подаються до захисту БДР, наступні: пояснювальна записка БДР, графічні і ілюстративні матеріали, протокол попереднього захисту БДР на кафедрі, відгук наукового керівника, відгук опонента, протоколи складання державних екзаменів, анотації до БДР українською та іноземною мовами, нормоконтроль про відповідність оформлення ДР діючим вимогам.

Таблиця А.1 — Етапи виконання роботи

№ етапу	Назва етапу	Термін виконання		Очікувані результати
		початок	кінець	
1	Постановка задачі роботи	09.03.22	09.03.22	Вступ
2	Пошук матеріалів по технологіям розробки віконних додатків	10.03.22	25.03.22	Розділ 1
3	Структурне проектування додатку організації робочого процесу	26.03.22	28.03.22	Розділ 2
4	Обґрунтування засобів реалізації системи	29.03.22	04.04.22	Розділ 2
5	Розробка головного вікна, методів створення календарю та подій	05.04.22	29.04.22	Розділ 3
6	Підготовка матеріалів та розробка алгоритму збереження та виведення списку подій	30.04.22	27.05.22	Розділ 3, Працююча система
7	Оформлення пояснювальної записки та ілюстративного матеріалу	28.05.22	06.06.22	Пояснювальна записка
8	Перевірка якості виконання бакалаврської роботи та усунення недоліків	07.06.22	07.06.22	Презентація

7 Порядок контролю виконання та захисту БДР, виконання етапів графічної та розрахункової документації контролюється науковим керівником згідно зі встановленими термінами. Захист відбувається на засіданні Державної екзаменаційної комісії, затвердженою наказом ректора.

8 При оформлюванні БДР використовуються:

— ДСТУ 3008 : 2015 «Звіти в сфері науки і техніки. Структура та правила оформлювання»;

— ДСТУ 8302 : 2015 «Бібліографічні посилання. Загальні положення та правила складання»;

— ГОСТ 2.104-2006 «Єдина система конструкторської документації. Основні написи»;

— документами на які посилаються у вище вказаних.

Технічне завдання до виконання отримав _____ Мазур В.Ю.

ДОДАТОК Б

Лістинг програми застосунку

Файл Hol_ESP_2022.ino

```
// Declaration for an SSD1306 display connected to I2C (SDA, SCL pins) D1,D2
```

```
#define PIN_T1 A0 // пін аналогового датчика температури (зовнішня)
```

```
#define PIN_T2 D3 // пін цифрового датчика температури (внутрішня)
```

```
#define PIN_P1 D5 // пін регулювання потужності 1 ЕП (елемента
```

Пельтьє)

```
#define PIN_P2 D6 // пін регулювання потужності 2 ЕП (елемента
```

Пельтьє)

```
#define PIN_Rele1 D7 // реле1
```

```
#define PIN_K D8 // кнопка
```

```
////////////////////////////////////
```

```
// RemoteXY include library //
```

```
////////////////////////////////////
```

```
// режим з'єднання RemoteXY
```

```
#define REMOTEXY_MODE__ESP8266WIFI_LIB_POINT
```

```
#include <ESP8266WiFi.h>
```

```
#include <RemoteXY.h>
```

```
#define REMOTEXY_WIFI_SSID "RemoteXY_Hol"
```

```
#define REMOTEXY_WIFI_PASSWORD "12345678"
```

```
#define REMOTEXY_SERVER_PORT 6377
```

```
// конфігурація інтерфейсу
```

```
#pragma pack(push, 1)
```

```
uint8_t RemoteXY_CONF[] = // 944 bytes
```

```
{ 255, 7, 0, 88, 0, 169, 3, 16, 158, 5, 130, 1, 0, 10, 61, 88, 0, 139, 130, 1,
```

```
2, 11, 56, 18, 0, 181, 129, 0, 20, 17, 21, 4, 0, 245, 208, 186, 208, 181, 209, 128,
```

```
209, 131, 208, 178, 208, 176, 208, 189, 208, 189, 209, 143, 32, 0, 131, 1, 7, 2,
```

```
25, 5, 1, 147, 31, 208, 147, 208, 190, 208, 187, 208, 190, 208, 178, 208, 189, 208, 176,
```

0, 68, 18, 4, 77, 56, 12, 1, 8, 178, 246, 4, 160, 4, 40, 52, 6, 1, 1, 26, 131, 0, 35, 0, 25, 4, 2, 147, 31, 208, 159, 209, 128, 208, 190, 32, 208, 183, 208, 176, 209, 129, 209, 130, 208, 190, 209, 129, 209, 131, 208, 189, 208, 190, 208, 186, 0, 131, 0, 35, 5, 25, 4, 3, 147, 31, 208, 157, 208, 176, 208, 187, 208, 176, 209, 136, 209, 130, 209, 131, 208, 178, 208, 176, 208, 189, 208, 189, 209, 143, 0, 129, 0, 5, 34, 51, 4, 1, 144, 208, 162, 208, 181, 208, 188, 208, 191, 208, 181, 209, 128, 208, 176, 209, 130, 209, 131, 209, 128, 208, 176, 32, 208, 186, 208, 176, 208, 188, 208, 181, 209, 128, 208, 184, 32, 40, 109, 105, 110, 41, 0, 2, 0, 21, 91, 21, 5, 1, 146, 26, 31, 31, 79, 102, 0, 79, 110, 0, 130, 0, 2, 58, 59, 16, 1, 167, 67, 5, 9, 66, 14, 5, 1, 244, 153, 11, 67, 5, 39, 66, 14, 5, 1, 244, 153, 11, 129, 0, 11, 63, 11, 3, 1, 144, 208, 154, 208, 176, 208, 188, 208, 181, 209, 128, 208, 184, 0, 129, 0, 19, 59, 25, 4, 1, 144, 208, 162, 208, 181, 208, 188, 208, 191, 208, 181, 209, 128, 208, 176, 209, 130, 209, 131, 209, 128, 208, 176, 44, 32, 32, 208, 161, 0, 129, 0, 2, 36, 56, 4, 3, 144, 208, 159, 208, 190, 209, 130, 209, 131, 208, 182, 208, 189, 209, 150, 209, 129, 209, 130, 209, 140, 32, 209, 133, 208, 190, 208, 187, 208, 190, 208, 180, 208, 184, 208, 187, 209, 140, 208, 189, 208, 184, 209, 133, 32, 208, 181, 208, 187, 46, 0, 4, 192, 13, 42, 45, 5, 3, 1, 26, 4, 192, 13, 49, 45, 5, 3, 1, 26, 129, 0, 3, 42, 7, 4, 3, 144, 226, 132, 150, 49, 0, 129, 0, 3, 49, 7, 4, 3, 144, 226, 132, 150, 176, 32, 0, 129, 0, 3, 21, 51, 4, 0, 245, 209, 133, 208, 190, 208, 187, 208, 190, 208, 180, 208, 184, 208, 187, 209, 140, 208, 189, 208, 190, 209, 142, 32, 209, 131, 209, 129, 209, 130, 208, 176, 208, 189, 208, 190, 208, 178, 208, 186, 208, 190, 209, 142, 0, 129, 0, 12, 68, 40, 4, 2, 203, 208, 160, 208, 190, 208, 183, 209, 128, 208, 190, 208, 177, 208, 189, 208, 184, 208, 186, 58, 32, 208, 156, 208, 176, 208, 183, 209, 131, 209, 128, 32, 208, 161, 46, 0, 129, 0, 11, 41, 42, 3, 2, 203, 208, 161, 208, 184, 209, 129, 209, 130, 208, 181, 208, 188, 208, 176, 32, 208, 176, 208, 178, 209, 130, 208, 190, 208, 188, 208, 176, 209, 130, 208, 184, 208, 183, 208, 190, 208, 178, 208, 176, 208, 189, 208, 189, 208, 190, 208, 179, 208, 190, 0, 129, 0, 16, 45, 34, 3, 2, 203, 208, 178, 209, 150, 208, 180, 208, 180, 208, 176, 208, 187, 208, 181, 208, 189, 208, 190, 208, 179, 208, 190, 32, 208, 186, 208, 181, 209, 128, 209, 131, 208, 178, 208, 176, 208, 189, 208, 189, 209, 143, 0, 129, 0, 13, 49, 39, 3, 2, 203, 209, 133, 208, 190, 208, 187, 208, 190, 208, 180, 208, 184, 208, 187, 209, 140, 208, 189, 208, 190, 209, 142, 32,

```
209, 131, 209, 129, 209, 130, 208, 176, 208, 189, 208, 190, 208, 178, 208, 186, 208,
190, 209, 142, 0, 3, 133, 208, 146, 208, 184, 208, 177, 209, 150, 209, 128, 32, 209,
128, 208, 181, 208, 182, 208, 184, 208, 188, 209, 131, 0};
```

```
// структура що визначає всі змінні і події інтерфейсу керування
```

```
struct {
```

```
    // input variables
```

```
    int8_t slider_11; // =-100..100 slider position
```

```
    uint8_t switch_11; // =1 if switch ON and =0 if OFF
```

```
    int8_t slider_21; // =0..100 slider position
```

```
    int8_t slider_22; // =0..100 slider position
```

```
    uint8_t select_21; // =0 if select position A, =1 if position B, =2 if position C,
```

```
    ... uint8_t select_22; // =0 if select position A, =1 if position B, =2 if position
    C, ..
```

```
    uint8_t pushSwitch_21; // =1 if state is ON, else =0
```

```
    // output variables
```

```
    float onlineGraph_1_var1;
```

```
    float onlineGraph_1_var2;
```

```
    char text_11[11]; // string UTF8 end zero
```

```
    char text_12[11]; // string UTF8 end zero
```

```
    char text_14[11]; // string UTF8 end zero
```

```
    char text_21[16]; // string UTF8 end zero
```

```
    char text_22[31]; // string UTF8 end zero
```

```
    // other variable
```

```
    uint8_t connect_flag; // =1 if wire connected, else =0
```

```
} RemoteXY;
```

```
#pragma pack(pop)
```

```
////////////////////////////////////
```

```
//      END RemoteXY include      //
```

```
////////////////////////////////////
```

```
#include <OneWire.h>
```

```

#include <DallasTemperature.h>
OneWire oneWire_out(PIN_T2);
DallasTemperature sensor_outhouse(&oneWire_out);
double tempoutDOUBLE = 0;//переменная с температурой для 18B20
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#define SCREEN_WIDTH 128 // OLED display width, in pixels
#define SCREEN_HEIGHT 32 // OLED display height, in pixels
// Declaration for an SSD1306 display connected to I2C (SDA, SCL pins)
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -
1);
unsigned long currentTime;
unsigned long loopTime;
float temp1, temp2; // ЗНАЧЕННЯ ТЕМПЕРАТУРИ ВСЕРЕДИНИ ТА
ЗОВНІ КАМЕРИ
float t; // ГРАНИЧНІ ЗНАЧЕННЯ ТЕМПЕРАТУРИ
int p1, p2; // ЗНАЧЕННЯ ПОТУЖНОСТІ ЕЛЕМЕНТІВ ПЕЛЬТЬЄ №1 ТА
№2
int r1 = 0, r2 = 1, r3 = 1, r11 = 1;
/* Функція перекодування кирилиці з UTF-8 в Win-1251 */
String utf8rus(String source)
{
int i, k;
String target;
unsigned char n;
char m[2] = { '0', '\0' };
k = source.length(); i = 0;
while (i < k) {
n = source[i]; i++;

```

```
if (n >= 0xC0) {
  switch (n) {
    case 0xD0: {
      n = source[i]; i++;
      if (n == 0x81) {
        n = 0xA8;
        break;
      }
      if (n >= 0x90 && n <= 0xBF) n = n + 0x30;
      break;
    }
    case 0xD1: {
      n = source[i]; i++;
      if (n == 0x91) {
        n = 0xB8;
        break;
      }
      if (n >= 0x80 && n <= 0x8F) n = n + 0x70;
      break;
    }
  }
}
m[0] = n; target = target + String(m);
}
return target;
}
void setup()
{
  RemoteXY_Init ();
  Serial.begin(9600);
```

```

// pinMode (PIN_T2, INPUT);
pinMode (PIN_P1, OUTPUT);
pinMode (PIN_P2, OUTPUT);
pinMode (PIN_Rele1, OUTPUT);
pinMode (PIN_K, OUTPUT);
currentTime = millis();
loopTime = currentTime;
sensor_outhouse.begin();
Serial.begin(115200);
if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) { // Address 0x3D for
128x64
    Serial.println(F("SSD1306 allocation failed"));
}
// delay(2000);
display.cp437(true);
display.clearDisplay();
temp1 = temp2 = t = 20;
}
int buttonState;          // стан кнопки
int lastButtonState = LOW; // Попередній стан кнопки
unsigned long lastDebounceTime = 0; // Час нажиманн кнопки
unsigned long debounceDelay = 50; // Час затримки для перевірки стану
КНОПКИ

int n = 0 , m = 140, t1 = 1;
////////////////////////////////////
void buton(int pin)
{
    int reading = digitalRead(pin);
    if (reading != lastButtonState) {

```



```

    lastDebounceTime = millis();
}
if ((millis() - lastDebounceTime) > debounceDelay) {
    if (reading != buttonState) {
        buttonState = reading;
        if (buttonState == HIGH) {
            r3++;
            if (r3 > 4) r3 = 1;
        }
    }
}
lastButtonState = reading;
}
void dysplay()
{
    char ss1[5];
    char ss2[5];
    char ss3[3];
    dtostrf(temp2, 4, 1, ss1);
    dtostrf(temp1, 4, 1, ss2);
    dtostrf(r11, 0, 0, ss3);
    if (r2 == 0)
    { display.clearDisplay();
      display.setTextColor(WHITE);
      display.setTextSize(1);
      display.setCursor(8, 0);
      display.print(utf8rus("Темп. камери-")); display.println(ss1);
      display.setCursor(10, 10);
      display.print(utf8rus("Темп. ззовні-")); display.println(ss2);
      display.setCursor(35, 20);

```

```

    display.print(utf8rus(" Режим-")); display.println(ss3);
    display.display();
}
if (r2 == 1)
{ display.clearDisplay();
  display.setTextColor(WHITE);
  display.setTextSize(3);
  display.setCursor(m, 10);
  display.print(utf8rus("Температура камери -")); display.print(ss1);
  display.print(utf8rus("; Температура ззовні -")); display.print(ss2);
  display.print(utf8rus(";          Режим          -"));          display.print(ss3);
display.print(utf8rus(";"));
  display.display();
}
if (r2 == 2)
{ display.clearDisplay();
  display.display();
}
}

void loop()
{
  RemoteXY_Handler ();
  currentTime = millis();          // считываем время, прошедшее с момента
запуска программы
  m = m - 1;
  if (m <= -1100) m = 130;
  buton(PIN_K); //// кнопка //////////
  if (currentTime >= (loopTime + 5000))
  {

```

```

sensor_outhouse.requestTemperatures();
temp2 = sensor_outhouse.getTempCByIndex(0);
temp1 = analogRead(PIN_T1);
temp1 = (temp1 / 1023.0 ) * 3.0 * 1000 / 10;
if (( t > temp2 ))// && (RemoteXY.switch_11))
{
    digitalWrite(PIN_Rele1, LOW);
}
if (( t < temp2 ))// && (RemoteXY.switch_11))
{
    digitalWrite(PIN_Rele1, HIGH);
}
if (RemoteXY.switch_11)
{
    digitalWrite(PIN_Rele1, LOW);
}
loopTime = currentTime;
}

////////////////////////////////////

RemoteXY.onlineGraph_1_var1 = (float) temp2;
RemoteXY.onlineGraph_1_var2 = (float) temp1;
t = map(RemoteXY.slider_11, -100, 100, 0, 25);
dtostrf(t, 0, 0, RemoteXY.text_14);
dtostrf(temp2, 0, 2, RemoteXY.text_11);
dtostrf(temp1, 0, 2, RemoteXY.text_12); //temp2
r1 = RemoteXY.select_21;
dtostrf(r1, 0, 0, RemoteXY.text_21);
r2 = RemoteXY.select_22;
switch (r1) {

```

```
case 0: {
    sprintf(RemoteXY.text_21, "1 - Кер. кноп.");
    r11 = r3;
} break;
case 1: {
    sprintf(RemoteXY.text_21, "2 - 25%% P");
    r11 = 1;
} break;
case 2: {
    sprintf(RemoteXY.text_21, "3 - 50%% P");
    r11 = 2;
} break;
case 3: {
    sprintf(RemoteXY.text_21, "4 - 75%% P");
    r11 = 3;
} break;
case 4: {
    sprintf(RemoteXY.text_21, "5 - 100%% P");
    r11 = 4;
} break;
}
if (RemoteXY.pushSwitch_21 == 0)
{
    if (r11 == 1) {
        p1 = 50;
        p2 = 0;
    }
    if (r11 == 2) {
        p1 = 50;
        p2 = 50;
    }
}
```

```

}
if (r11 == 3) {
  p1 = 100;
  p2 = 50;
}
if (r11 == 4) {
  p1 = 100;
  p2 = 100;
}
analogWrite(PIN_P1, map( p1, 0, 100, 0, 255));
analogWrite(PIN_P2, map( p2, 0, 100, 0, 255));
}
switch (r2) {
  case 0: sprintf(RemoteXY.text_22, "Статичний текст"); break;
  case 1: sprintf(RemoteXY.text_22, "Біжучий рядок"); break;
  case 2: sprintf(RemoteXY.text_22, "Вимкнений екран"); break;
}
display();
if (RemoteXY.pushSwitch_21 == 1)
{
  p1 = 255 - map( RemoteXY.slider_21, 0, 100, 0, 255);
  p2 = 255 - map( RemoteXY.slider_22, 0, 100, 0, 255);
  analogWrite(PIN_P1, map( p1, 0, 100, 0, 255));
  analogWrite(PIN_P2, map( p2, 0, 100, 0, 255));
}
////////////////////////////////////
if (RemoteXY.switch_11)
{
  digitalWrite(PIN_Rele1, LOW);
}

```

```
if ((!RemoteXY.switch_11)&& ( t < temp2 ))
{
  digitalWrite(PIN_Rele1, HIGH);
}
}
////////////////////////////////////
```

ДОДАТОК В
ПРОТОКОЛ
ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ
НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ

Назва роботи: SMART — система управління термоустановкою на базі модуля ESP 8266

Тип роботи: бакалаврська дипломна робота
(БДР, МКР)

Підрозділ кафедра обчислювальної техніки
(кафедра, факультет)

Показники звіту подібності Unichек

Оригінальність 71,5 Схожість 28,5

Аналіз звіту подібності (відмітити потрібне):

- Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.
- Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її виконання автором. Роботу направити на розгляд експертної комісії кафедри.
- Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Особа, відповідальна за перевірку Захарченко С.М.

Ознайомлені з повним звітом подібності, який був згенерований системою Unichек щодо роботи.

Автор роботи Мазур В.Ю.

Керівник роботи Томчук М.А.