

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

МАТЕМАТИЧНИЙ ФАКУЛЬТЕТ

Кафедра програмної інженерії

**КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА**

на тему: «РОЗРОБКА МОБІЛЬНОГО ДОДАТКУ  
УПРАВЛІННЯ СВІТЛОМ З ВИКОРИСТАННЯМ  
NODEMCU»

Виконав: студент  2  курсу, групи  8.1211-1іпз   
спеціальності  121 інженерія програмного забезпечення   
(шифр і назва спеціальності)  
освітньої програми  інженерія програмного забезпечення   
(назва освітньої програми)

К. Барау

(ініціали та прізвище)

Керівник  доцент кафедри програмної інженерії, доцент,   
 к.т.н., Мухін В.В.   
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Рецензент  завідувач кафедри фундаментальної та   
 прикладної математики, професор, д.т.н.   
 Гребенюк С.М.   
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Запоріжжя  
2022

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**

Факультет математичний

Кафедра програмної інженерії

Рівень вищої освіти магістр

Спеціальність 121 інженерія програмного забезпечення

(шифр і назва)

Освітня програма інженерія програмного забезпечення

**ЗАТВЕРДЖУЮ**

Завідувач кафедри програмної  
інженерії, к.ф.-м.н., доцент

Лісняк А.О.

(підпис)

« \_\_\_\_ » \_\_\_\_\_ 2022 р.

**З А В Д А Н Н Я**

**НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТОВІ(СТУДЕНТЦІ)**

Барау Каріму

(прізвище, ім'я та по-батькові)

1. Тема роботи (проекту) Розробка мобільного додатку управління  
світлом з використанням podemcu

керівник роботи (проекту) Мухін Віталій Вікторович, к.т.н., доцент

(прізвище, ім'я та по-батькові, науковий ступінь, вчене звання)

затверджені наказом ЗНУ від « \_\_\_\_ » \_\_\_\_\_ 2022 року № \_\_\_\_\_

2. Строк подання студентом роботи \_\_\_\_\_

3. Вихідні дані до роботи 1. Постановка задачі.

2. Перелік літератури.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Постановка задачі.

2. Проектування управління світлом.

3. Розробка додатку управління світлом за допомогою App Inventor

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) \_\_\_\_\_

Презентація

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання \_\_\_\_\_

**КАЛЕНДАРНИЙ ПЛАН**

№	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Розробка плану роботи.		
2.	Збір вихідних даних.		
3.	Обробка методичних та теоретичних джерел.		
4.	Розробка першого та другого розділу.		
5.	Розробка третього розділу.		
6.	Оформлення та нормоконтроль кваліфікаційної роботи.		
7.	Підготовка доповіді та презентації.		
8.	Захист кваліфікаційної роботи.		

Студент \_\_\_\_\_  
(підпис)

К. Барау \_\_\_\_\_  
(ініціали та прізвище)

Керівник роботи \_\_\_\_\_  
(підпис)

В. В. Мухін \_\_\_\_\_  
(ініціали та прізвище)

**Нормоконтроль пройдено**

Нормоконтролер \_\_\_\_\_  
(підпис)

О. В. Кудін \_\_\_\_\_  
(ініціали та прізвище)

## РЕФЕРАТ

Кваліфікаційна робота магістра «Розробка мобільного додатку управління світлом з використанням nodemcu»: 58 с., 38 рис., 26 джерела, 2 додатки.

APP INVECTOR, ARDUINO, NODEMCU, RELAY, VOLTAGE DETECTOR.

Об'єкт дослідження – технологія використання nodemcu.

Мета роботи: розробка мобільного додатку управління світлом з використанням nodemcu.

Метод дослідження – синтез.

У кваліфікаційній роботі розглядаються основні додатки для управління освітленням в Україні, а також технологію застосування NodeMcu на основі модуля ESP8266 для управління світлом. розроблено скетч для прошивки плати написаний в середовищі Arduino IDE, створено андроїд додаток, який демонструє роботу включення (виключення) світла в кімнаті. Андроїд додаток реалізовано за допомогою хмарного середовища візуальної розробки Android додатків – App Inventor.

## SUMMARY

Master`s qualifying paper « Development of the Mobile Light Control Application using Nodemcu »: 58 pages, 38 figures, 26 references, 2 supplements.

APP INVECTOR, ARDUINO, NODEMCU, RELAY, VOLTAGE DETECTOR.

The object of the research is technology of using the nodemcu.

The aim of the research is to construct the Android program for remote light control by using the nodemcu.

The method of research is synthesis.

The main proposals applications for lighting control in Ukraine are considered, as well as the technology of NodeMcu board based on the module ESP8266 for remote light control are considered in the qualifying paper. The sketch for the firmware board written in the Arduino IDE is developed. The android application that demonstrates the operation of remote on (off) lights in the room is created. The android application is implemented using a cloud environment for visual development of Android applications – App Inventor.

## ЗМІСТ

Завдання на кваліфікаційну роботу.....	2
Реферат.....	4
Summary.....	5
Вступ.....	7
1 Огляд додатків для керування світлом .....	9
2 Апаратні засоби управління світлом.....	16
2.1 Платформа NodeMcu .....	16
2.2 Електромагнітне реле.....	21
2.3 Прохідний вимикач .....	23
2.4 Детектор напруги з оптопарою.....	24
2.5 Датчик вологості та температури DHT11.....	25
2.6 Підключення NodeMCU до комп'ютера .....	26
2.7 Управління світлом .....	30
3 Android додаток включення (виключення) світла.....	33
Висновки.....	42
Перелік посилань.....	43
Додаток А Скетч для прошивки плати на основі модуля ESP8266.....	46
Додаток Б Додаток, який демонструє роботу включення (виключення) світла в кімнаті.....	53

## ВСТУП

Ми звикли вмикати та вимикати світло за допомогою звичайних вимикачів. У системах «розумний дім» вже давно використовуються прогресивні способи – пульти дистанційного керування та панелі. Вони дозволяють контролювати освітлення одразу у всьому будинку з однієї точки. З появою функції контролю світла через Wi-Fi власники будинків та квартир можуть керувати освітленням поза домом.

Причин, щоб керувати світлом через смартфон, декілька:

- можна використовувати власний смартфон усередині будинку, не вдаючись до пошуку пульта дистанційного керування;
- створювати «ефект присутності» в будинку, перебуваючи за його межами – з метою охорони;
- налаштовувати потрібний сценарій світла, йдучи з роботи, щоби прийти в освітлений будинок.

Така функція особливо зручна тим, хто має обмеження у пересуванні, пов'язані зі станом здоров'я. Також це забезпечить комфорт батькам – наприклад, коли маленька дитина прокинулася вночі та злякалася темряви.

Контроль за освітленням у цій схемі виконується через Wi-Fi сигнал. Щоб налаштувати керування світлом через смартфон, потрібні пристрої:

- смартфон, в який встановлюється спеціальний додаток;
- виконавчий пристрій – до нього підключаються світильники;
- wi-fi роутер, який здійснює передачу даних між смартфоном та виконавчим пристроєм.

В ідеалі мінімальний набір функцій, якими повинно бути наділено «розумне світло» повинен складатися з можливості включати і вимикати світло як з вимикача, розташованого на стіні, так і віддалено за допомогою смартфона.

Отже, метою кваліфікаційної роботи є розробка мобільного додатку управління світлом.

Для досягнення поставленої мети в роботі поставлені наступні задачі:

- провести аналіз основних додатків для керування світлом;
- ознайомитись з платою розширення NodeMcu на основі модуля ESP8266;
- розробити мобільний додаток управління світлом.



## 1 ОГЛЯД ДОДАТКІВ ДЛЯ КЕРУВАННЯ СВІТЛОМ

Розглянемо основні додатки для управління освітленням доступних сьогодні на ринку.

**Додаток Smart Life:** Додаток Smart Life є універсальним додатком, який можна використовувати для керування безліччю різних типів пристроїв, включаючи освітлення [1, 2] (рис. 1.1).



Рисунок 1.1 – Додаток Smart Life

Додаток Smart Life робить управління розумним будинком максимально зручним і комфортним [3]. Так користувач може одним натисканням включати будь-який розумний прилад або створювати розумні сценарії. До того ж, можна здійснювати управління пристроями за допомогою розумних годин.

**Додаток Philips Hue:** Додаток Philips Hue доступний для телефонів iOS та Android, iPad та планшетів [4]. Philips Hue дозволяє легко розмістити окремі світильники у кімнатах. Ви можете вмикати / вимикати кімнати, змінювати колір або яскравість всіх джерел світла або створювати свою власну індивідуальну атмосферу освітлення (рис. 1.2).

На жаль, для керування освітленням нам потрібен не лише додаток Philips Hue, а також Philips Hue Smart Hub, який необхідно підключити до роутера. Таким чином він може спілкуватися з лампами Philips Hue через Wi-Fi.

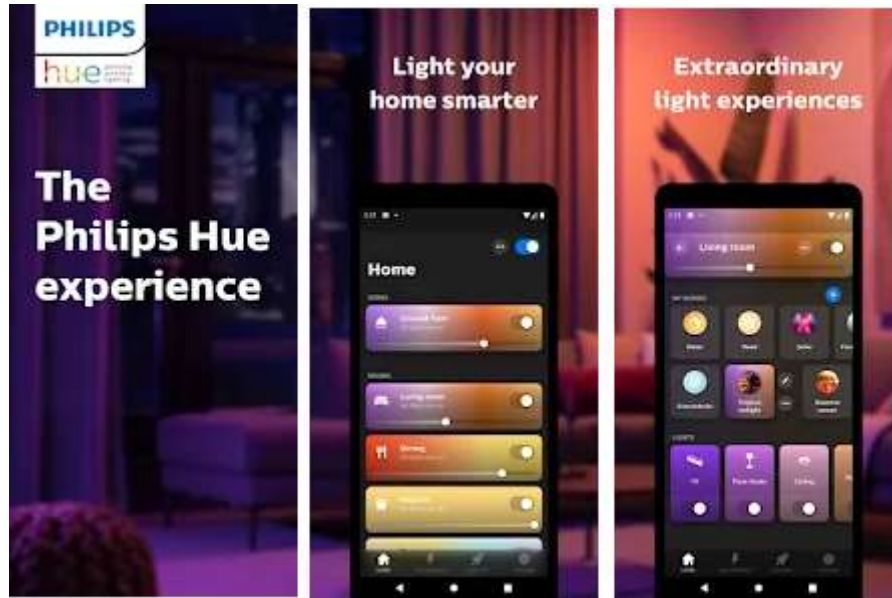


Рисунок 1.2 – Додаток Philips Hue

**Додаток Mi Home:** Додаток Mi Home для управління усією екосистемою розумного будинку від компанії Xiaomi [5] (рис. 1.3). У ній можна отримати контроль не лише над світлом, а й кожним розумним приладом, що встановлений у вашому будинку. Проте їм підтримується як фірмова продукція, а й стороння.

**Додаток Ecobee:** Додаток Ecobee доступний як для користувачів iOS, так і для Android [6] (рис. 1.4). Ecobee дозволяє віддалено контролювати температуру в будинку та налаштування освітлення. Це також дозволяє вам перевірити, чи залишили ви світло увімкненим, навіть коли ви знаходитесь вдалині від будинку.

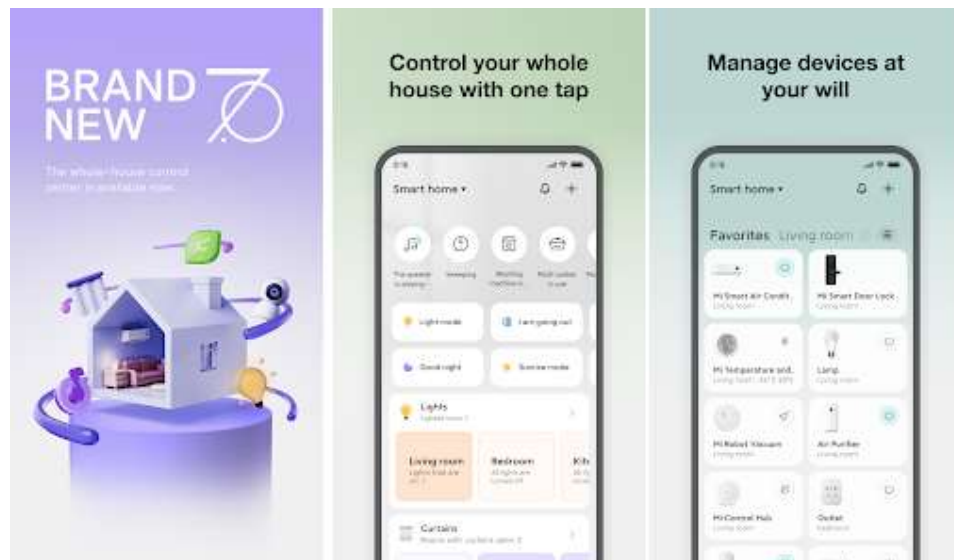


Рисунок 1.3 – Додаток Philips Hue

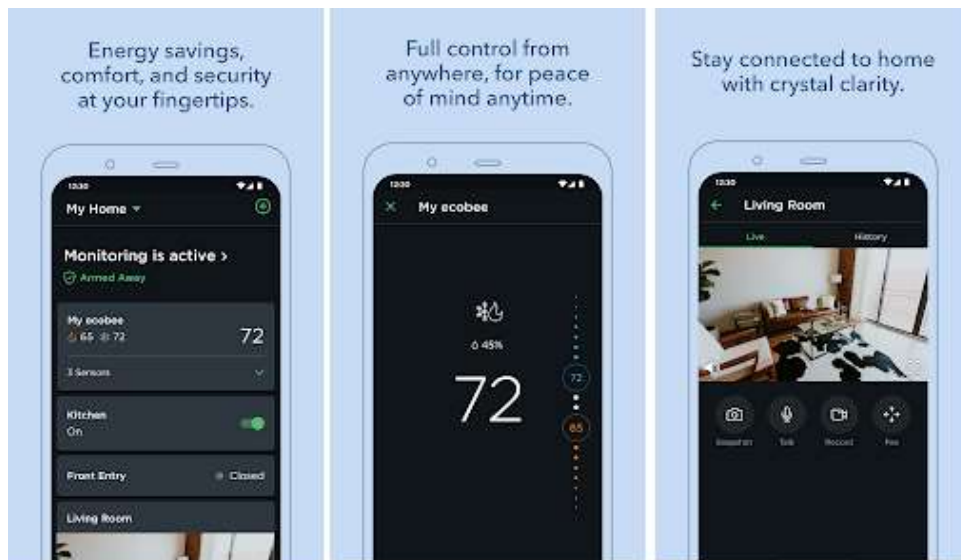


Рисунок 1.4 – Додаток Ecobee

**Додаток Wemo:** Wemo також доступний для користувачів Android та iOS. Wemo управляє серією пристроїв домашньої автоматизації, включаючи інтелектуальні перемикачі, розетки та міст Wi-Fi [7] (рис. 1.5). Додаток Wemo дозволяє керувати всім освітленням та побутовою технікою у вашому будинку, автоматично вмикати світло, коли ви приходите додому, і випадковим чином вмикати та вимикати світло в різних кімнатах з метою безпеки.

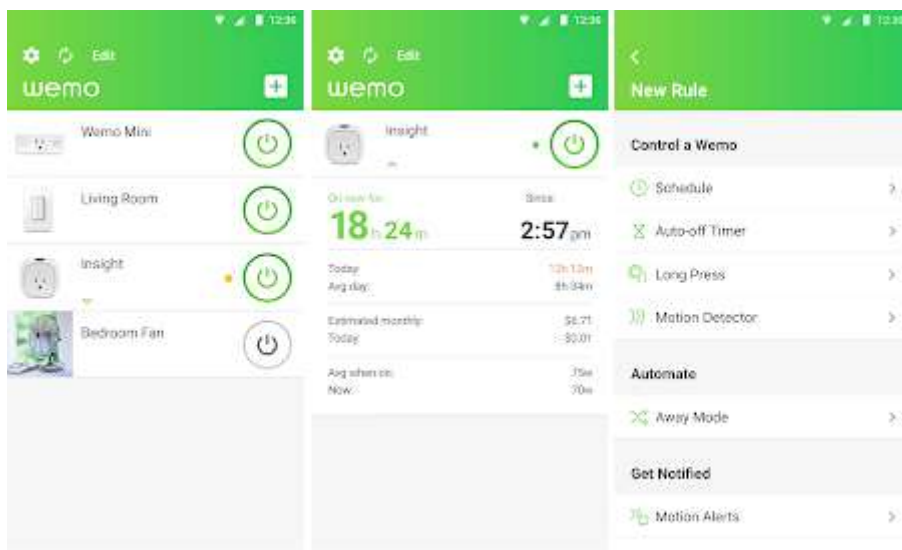


Рисунок 1.5 – Додаток Wemo

**Додаток Yonomi:** Yonomi доступна як для Android, так і для iOS. Yonomi можна використовувати для керування безліччю гаджетів розумного будинку [8] (рис. 1.6). Він дозволяє легко регулювати яскравість і відтінок підключених джерел світла, будь то лампа, підключена до GE Link, LIFX A19 або Philips Hue. Він підтримує сторонніх помічників, таких як Google Assistant та Amazon Alexa, для голосових команд користувача.

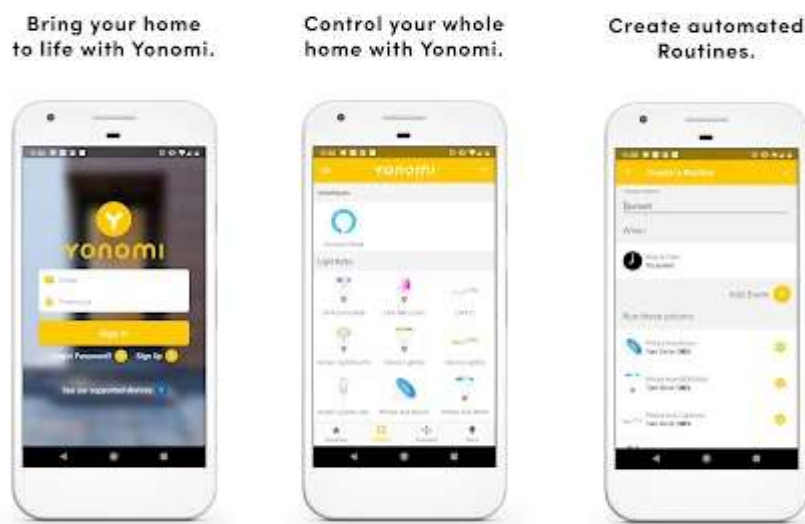


Рисунок 1.6 – Додаток Yonomi

**Додаток ELARI SmartHome:** ELARI SmartHome дозволяє керувати не лише компонентами безпеки, а й освітленням у розумному будинку [9] (рис. 1.7). Підтримка чотирьох різних видів ламп, які поставляються компанією. Обумовлено це тим, що використовувати їх можна в різних частинах інтер'єру: стелі, світильниках тощо. Якщо ж говорити про налаштування освітлення, то тут все стандартно. Можна налаштувати за таймером, місцезнаходженням та іншими умовами, які встановить сам власник. Основна відмінність – це та різноманітність, яку можна привнести в інтер'єр при грамотному використанні.

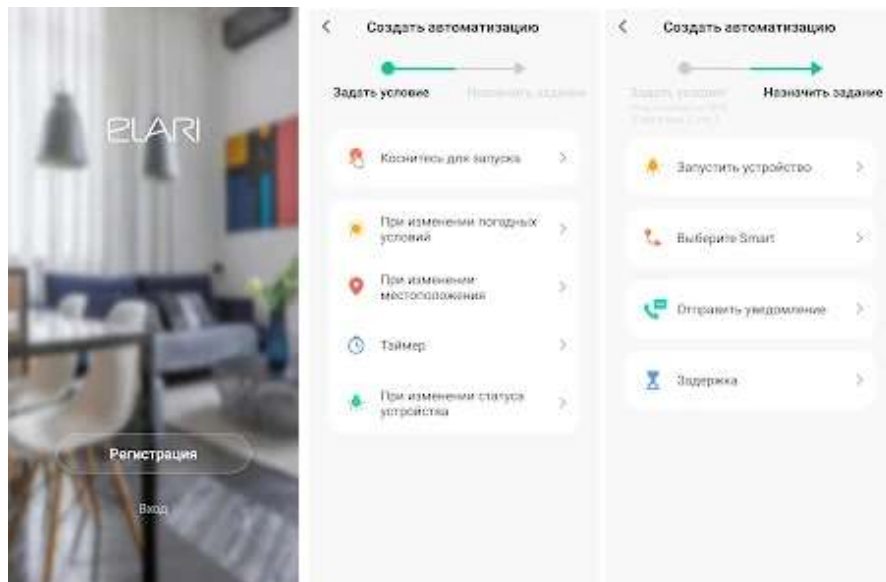


Рисунок 1.7 – Додаток ELARI SmartHome

**Додаток Livicom:** Система домашньої безпеки і комфорту Livicom зробить зі звичного житла розумний будинок, яким можна керувати з будь-якої точки світу [10] (рис. 1.8). Система складається з хаба, радіодатчиків і пристроїв управління Livi, а також сумісного обладнання (відеокамер, що підключаються сторонніх виробників). Додаток Livicom дозволяє дистанційно керувати пристроями розумного будинку, надавати доступ членам сім'ї, створювати сценарії і розкладу для автоматизації роботи приладів і інженерних систем.



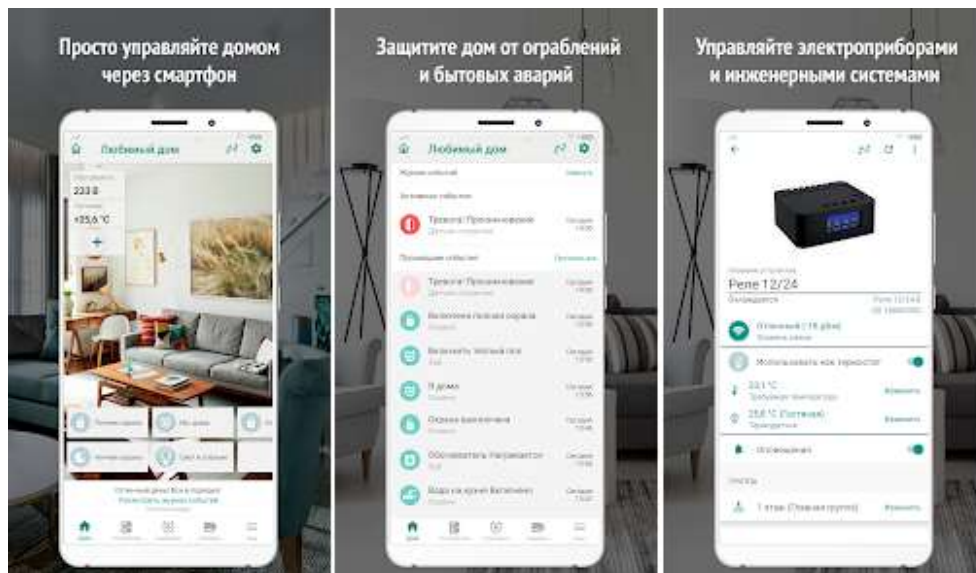


Рисунок 1.8 – Додаток Livicom

**Додаток Eufy Clean.** Eufy Clean дозволяє керувати своїми пристроями розумного будинку Eufy із будь-якої точки світу [11] (рис. 1.9). Компанія не відрізняється сценаріями використання — вони приблизно скрізь однакові, але пропонує незвичайні світлові рішення. У неї можна знайти нічні лампи, які працюють безпосередньо з розетки, звичайні лампи, світильники, і «струни», які світяться, що нагадують LED-стрічки. Програма компанії підтримує Андроїд і дозволяє групувати список пристроїв, щоб керування залежало від кімнати.

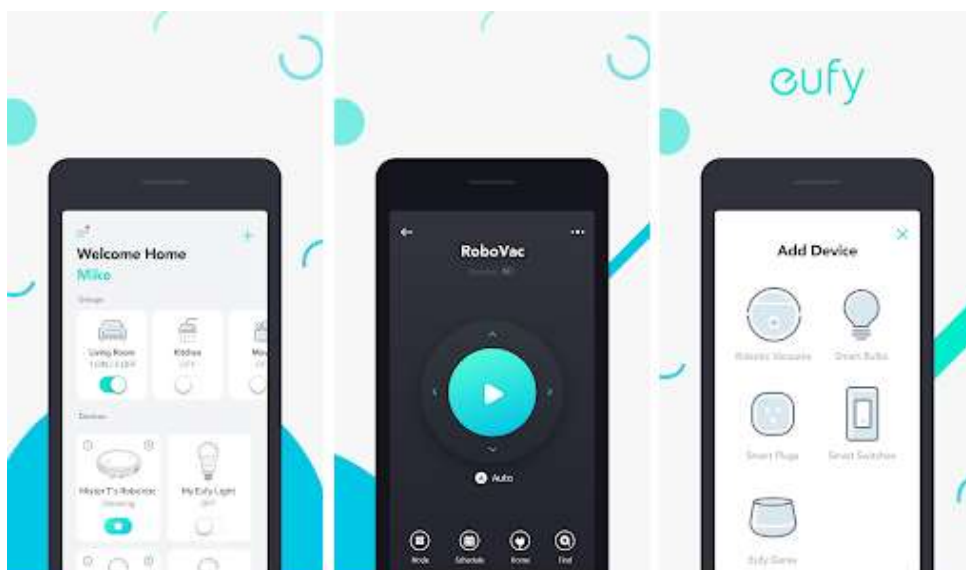


Рисунок 1.9 – Додаток Eufy Clean

**Додаток Life Control 2.0:** Доступна та зрозуміла система розумного будинку LifeControl [12] (рис. 1.10). З системою Розумний дім ви отримуєте не тільки всеосяжну інформацію про важливі показники та стан об'єкта, але й можливість будь-якої миті вплинути на ситуацію. Функціонал, який підтримується:

- відеоспостереження та зберігання важливих відеозаписів у хмарі;
- відстеження розташування близьких людей та автомобіля;
- управління розумними пристроями (світло, ворота, жалюзі, насоси тощо);
- контроль будинку через датчики: диму, протікання, якості повітря, датчик руху, датчики відкриття дверей та вікон та інших;
- контроль стану ваших рослин через датчик догляду за рослинами.



Рисунок 1.10 – Додаток Life Control 2.0

Список додатків не вичерпний, тому що виробників розумних пристроїв на ринку зараз надмірно багато. Проте це основні гравці, які точно зможуть надати якісну продукцію та сервіс. Крім, деякі з продуктів можуть керуватися як фірмовими, а й сторонніми програмами.

## 2 АПАРАТНІ ЗАСОБИ УПРАВЛІННЯ СВІТЛОМ

### 2.1 Платформа NodeMcu

NodeMCU — це платформа на основі модуля ESP8266 [13]. Плата призначена для зручного управління різними схемами на відстані за допомогою передачі сигналу в локальну мережу або інтернет через Wi-Fi. На базі Node MCU можна створити «розумний будинок», налаштувавши управління світлом через телефон, реєстрацію показань датчиків і багато іншого.

Плата оснащена модулем ESP-12E, що містить мікросхему ESP8266 з мікропроцесором RISC Tensilica Xtensa® 32-bit LX106, який працює з регульованою тактовою частотою від 80 до 160 МГц і підтримує RTOS (рис. 2.1) [14].

#### Характеристики ESP-12E

- 32-розрядний LX106 від Tensilica Xtensa®;
- Тактова частота від 80 до 160 МГц;
- 128 КБ вбудованої оперативної пам'яті;
- 4 МБ зовнішньої зовнішньої флеш-пам'яті;
- Приймач Wi-Fi 802.11b/g/n.

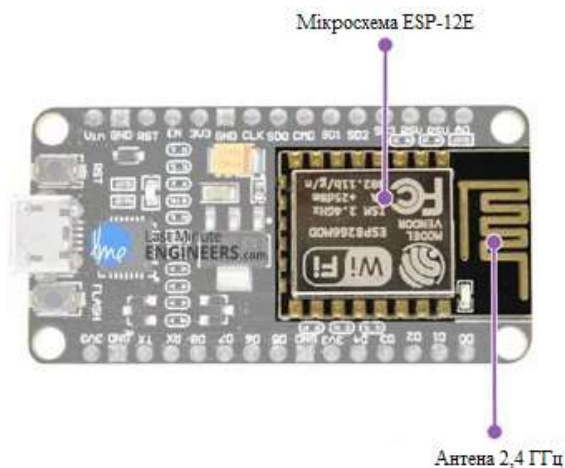


Рисунок 2.1 – ESP8266 NodeMCU



Також даний модуль має 128 КБ ОЗУ та 4 МБ флеш-пам'яті (для зберігання програм та даних), достатніх, щоб впоратися з великими рядками, які становлять веб-сторінки, даними в JSON/XML та всім, що ми сьогодні додаємо на пристрої IoT.

ESP8266 містить вбудований приймач Wi-Fi 802.11b/g/n HT40, тому він може не тільки підключатися до мережі Wi-Fi і взаємодіяти з інтернетом, але і встановлювати власну мережу, дозволяючи іншим пристроям підключатися безпосередньо до нього. Це робить ESP8266 NodeMCU ще більш універсальним.

Оскільки діапазон робочої напруги ESP8266 становить від 3 до 3,6 В, дана плата для підтримки постійної напруги на рівні 3,3 В поставляється з LDO стабілізатором напруги. Він може надійно забезпечувати струм до 600 мА, що має бути більш ніж достатньо, оскільки ESP8266 під час радіочастотних передач споживає до 80 мА. Вихід стабілізатора також виводиться на виводи на сторонах плати та позначений як 3V3. Ці виводи можна використовувати для подачі живлення на зовнішні компоненти (рис. 2.2).

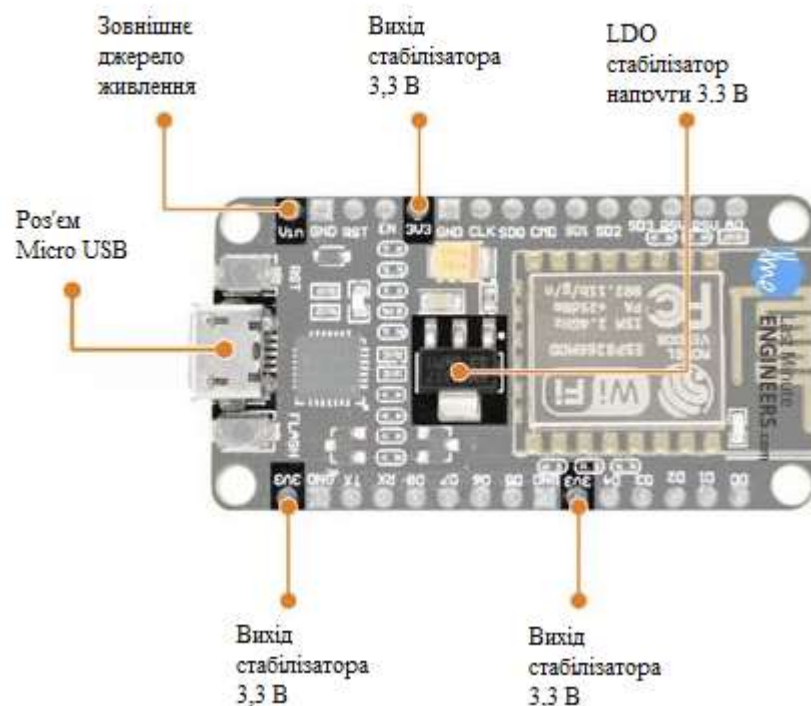


Рисунок 2.2 – Елементи живлення ESP8266 NodeMCU

Вимоги до живлення

- робоча напруга: від 2,5 до 3,6 В;
- вбудований стабілізатор: 3,3 В, 600 мА;
- робочий струм: 80 мА;
- споживання в режимі сну: 20 мкА.

Живлення до ESP8266 NodeMCU подається через вбудований USB-роз'єм MicroB. Як альтернатива, якщо у вас є стабілізоване джерело напруги 5 В, можна використовувати вивод VIN для безпосереднього живлення ESP8266 та його периферії.

ESP8266 NodeMCU має загалом 17 виводів GPIO, виведених на роз'єми з обох сторін плати. Ці виводи можуть використовуватися для виконання різних периферійних завдань, зокрема:

- вхід АЦП – канал 10-розрядного АЦП;
- інтерфейс UART – інтерфейс UART використовується для завантаження коду послідовного зв'язку;
- виходи ШІМ – виводи ШІМ можуть використовуватися для регулювання яскравості світлодіодів або керування двигунами;
- інтерфейси SPI, I2C – інтерфейси використовуються SPI та I2C для підключення всіляких датчиків та периферійних пристроїв;
- інтерфейс I2S – інтерфейс I2S використовується для цифрової передачі звуку.

Мультиплексовані виводи введення/виводу (рис. 2.3)

- канал АЦП
- інтерфейси UART
- виходи ШІМ
- інтерфейси SPI, I2C та I2S

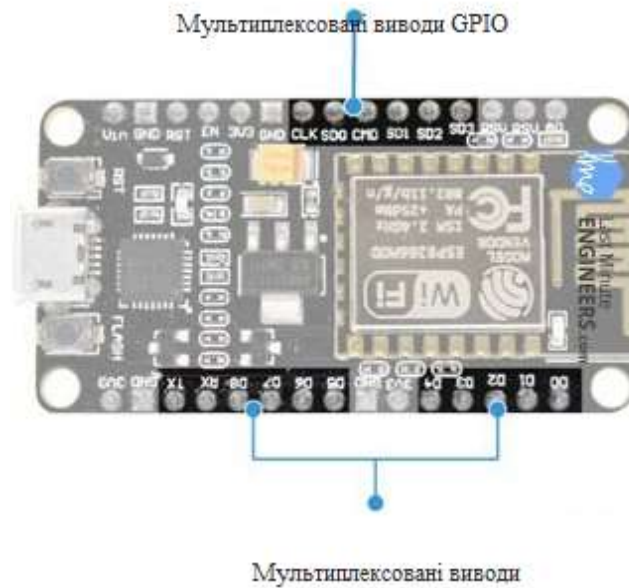


Рисунок 2.3 – Мультиплексовані виводи GPIO плати ESP8266 NodeMCU

ESP8266 використовується функція мультиплексування виводів (кілька периферійних пристроїв мультиплексуються на один вивід GPIO). Це означає, що один вивід GPIO може діяти як PWM/UART/SPI.

ESP8266 NodeMCU має дві кнопки. Одна, позначена як RST, розташована у верхньому лівому кутку, це кнопка скидання, яка, звичайно ж, використовується для скидання чіпа ESP8266. Інша кнопка FLASH в нижньому лівому куті — це кнопка завантаження, яка використовується при оновленні прошивки.

ESP8266 NodeMCU має всього 30 контактів (рис. 2.4).

Призначення контактів:

- вивід VIN – вивід для підключення зовнішнього джерела живлення;
- виводи 3V3 та GND – вихід зі внутрішнього стабілізатора напруги, можна використовувати для подачі живлення на зовнішні пристрої;
- RST – вивід RST використовується для скидання мікросхеми ESP8266;
- EN – коли на виводі висока напруга мікросхема ESP8266 включена, коли низька в режимі енергозбереження;

– CLK (GPIO6), SDO (GPIO7), CMD (GPIO11), SD1 (GPIO8), SD2 (GPIO9) та SD1 (GPIO10) – вивода підключені до Flash чіпу в модулі ESP-12E, не рекомендується використовувати;

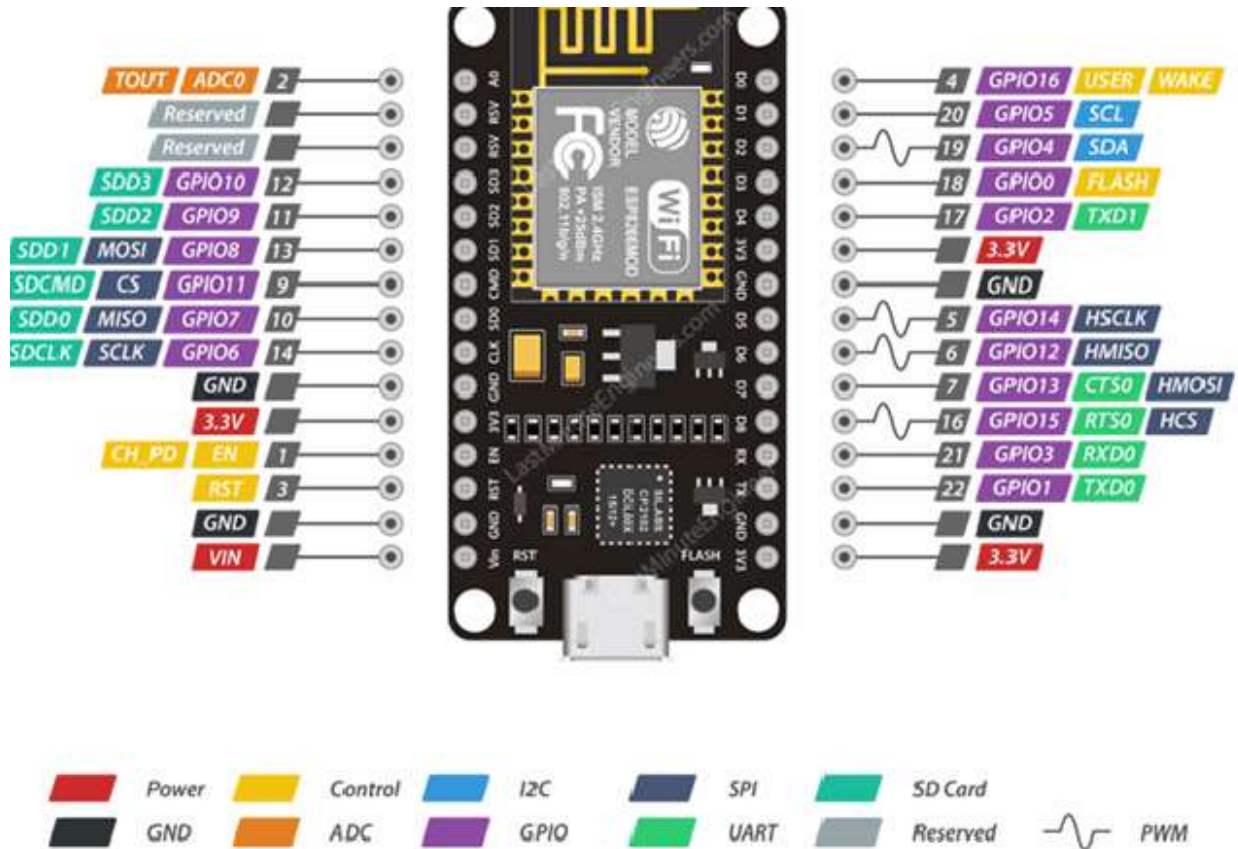


Рисунок 2.4 – Призначення контактів NodeMCU ESP8266

– PSW – зарезервовані контакти;  
 – A0 – вивід вбудованого 10-розрядного АЦП;  
 – TX (GPIO1), RX (GPIO3) – контакти для завантаження прошивки і зв'язку з комп'ютером, не рекомендується використовувати;  
 – D0 (GPIO16), D1 (GPIO5), D2 (GPIO4), D3 (GPIO0), D4 (GPIO2), D5 (GPIO14), D6 (GPIO15), D7 (GPIO13), D8 (GPIO15) – вивода для підключення зовнішніх пристроїв.

Так само, деякі вивода GPIO мають додаткові функції.

Інтерфейс SPI:

– GPIO12: MOSI;

- GPIO13: MISO;
- GPIO14: SCLK;
- GPIO15: CS.

Інтерфейс I2C (Так як ESP8266 не має апаратних виводів I2C, але можна реалізувати програмно). Зазвичай в якості контактів I2C використовуються наступні вивода:

- GPIO5: SCL;
- GPIO4: SDA;

– *ШИМ* (PWM), ESP8266 підтримує програмний ШІМ на всіх виводах введення / виведення: від GPIO0 до GPIO16.

## 2.2 Електромагнітне реле

Для керування електричними приладами будемо використовувати електромагнітне реле. Ми не можемо безпосередньо підключити до плати NodeMcu навіть звичайну лампочку – плата не призначена для такого навантаження і не працюватиме.

Реле – це електрично керований, механічний перемикач [15]. Усередині цього корпусу є електромагніт, при попаданні на нього електричного струму, він спрацьовує, внаслідок чого якір притягується до електромагніту і контактна група замикає або розмикає ланцюг живлення навантаження.

Простими словами: подаючи певний струм на електроланцюг з реле, воно замикає (розмикає) інший електроланцюг.

Реле має 2 роздільні ланцюги: ланцюг управління (контакти A1, A2) та керований ланцюг (контакти 1, 2, 3). Ланцюги ніяк не пов'язані між собою. Між контактами A1 і A2 встановлено металевий сердечник, при протіканні струму яким до нього притягується рухомий якір (2). Контакти ж 1 та 3 нерухомі. Якір підпружинний і поки ми не пропустимо струм через сердечник, якір буде утримується притиснутим до контакту 3. При подачі струму, як уже

говорилося, сердечник перетворюється на електромагніт і притягується до контакту 1. При знеструмленні пружина знову повертає яркір до контакту 3 (рис. 2.5).

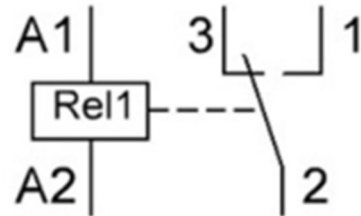


Рисунок 2.5 – Схема реле

Для керування різними виконавчими пристроями, комутації ланцюгів, керування приладами в електроніці активно застосовується електромагнітне реле.

Пристрій реле досить простий. Його основою є котушка, що складається з великої кількості витків ізолюваного дроту.

Всередину котушки встановлюється стрижень із м'якого заліза. В результаті виходить електромагніт. Також у конструкції реле присутні яркір. Він закріплений на пружному контакті. Сам пружний контакт закріплений на ярмі. Разом зі стрижнем і ярком ярмо утворює магнітопровід.

Якщо котушку підключити до джерела струму, то магнітне поле, яке утворилося, намагнічує сердечник. Він своєю чергою притягує яркір. Яркір укріплений на пружному контакті. Далі пружний контакт замикається з іншим нерухомим контактом. Залежно від конструкції реле, яркір може по-різному механічно керувати контактами (рис. 2.6).

Реле включає або вимикає зовнішні пристрої, певним чином замикаючи або розмикаючи окрему електричну мережу, в яку вони підключені. За допомогою NodeMcu та реле (рис.2.7) ми керуємо процесом включення або вимикання так само, як вмикаємо або вимикаємо світло вдома – подаючи команду на замикання або розмикання. NodeMcu подає сигнал, саме

замикання чи розмикання робитиме реле через спеціальні внутрішні механізми.

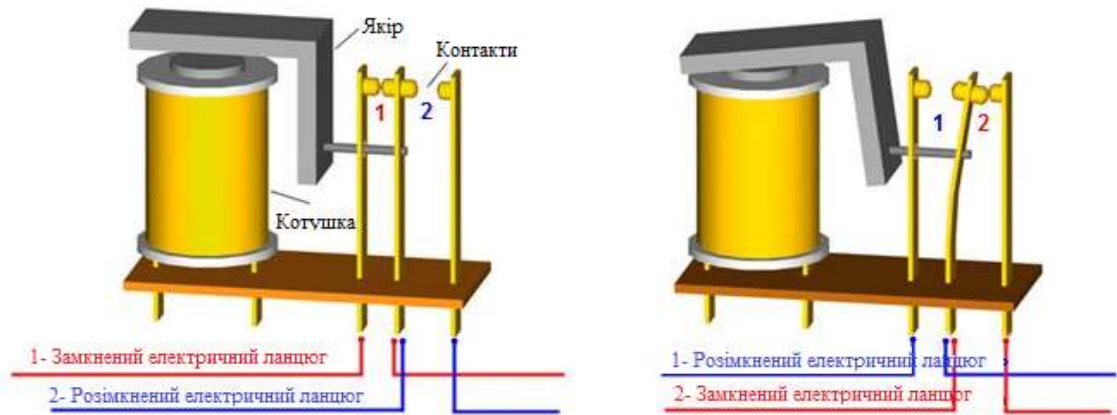


Рисунок 2.6 – Електромагнітне реле



Рисунок 2.7 – Електромагнітне реле Arduino

### 2.3 Прохідний вимикач

Для здійснення включення і виключення освітлення з одного боку будемо використовувати прохідний вимикач а з іншого електромагнітне реле, яке керується платформою NodeMcu.

Прохідним вимикачем називають вимикач, у якому, на відміну від традиційного вимикача, задіяно не пару контактів, а три контакти; себто, пристрій, який являє собою побутовий перемикач [16].

Вимикач прохідний можна використовувати скрізь, де є потреба маніпулювати одним джерелом світла з кількох місць.

Принцип роботи прохідного вимикача не дуже складний. Наприклад, якщо ви хочете керувати освітленням на сходах, що з'єднує кілька поверхів, то один вимикач встановлюється на початок сходового маршу, а інший в кінці, таким чином ви можете вмикати світло на вході на сходи і вимикати, коли заходите на поверх, і навпаки.

Для установки вам знадобиться щонайменше два вимикачі. Перемикання одного з них викликає зміну стану всього електричного ланцюга на протилежне (розімкнуто/замкнуто). Це означає, що, увімкнувши світло першим вимикачем, ви зможете вимкнути його другим.

На рисунку 2.8 наведена найпростіша схема, у якій навантаженням буде необхідна кількість паралельно підключених ламп, і задіяна пара прохідних вимикачів [17].

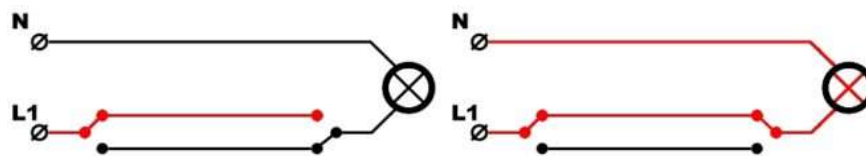


Рисунок 2.8 – Схема роботи прохідного вимикача

## 2.4 Детектор напруги з оптопарою

Для визначення наявності напруги 220В будемо використовувати детектор напруги.

Детектор напруги з оптопарою [18] є одним з найбільш якісних і ефективних рішень такого типу (рис. 2.9). Рівень вихідного TTL-сигналу можна визначити за допомогою мікроконтролера. Пристрій має оптогальванічну розв'язку, це дає можливість підключати модуль безпосередньо до виводу мікроконтролера. Модуль корисний для проектів



користувачів на базі мікроконтролерів. Датчик можна задіяти в інших системах, наприклад в розумному будинку для перетворення величини змінної напруги і її подальшого вимірювання. За наявності змінної напруги на вході, вихід OUT матиме логічний нуль.



Рисунок 2.9 – Детектор напруги 220В з опт опарою

## 2.5 Датчик вологості та температури DHT11

Для вимірювання вологості та температури в приміщенні будемо використовувати датчик DHT11 (рис. 2.10) [19-21] .



Рисунок 2.10 – Датчик вологості і температури DHT11

Датчик DHT11 – це цифровий датчик температури і вологості, що дозволяє калібрувати цифровий сигнал на виході. Складається з ємнісного датчика вологості і термістора. Також, датчик містить в собі АЦП для перетворення аналогових значень вологості і температури. Живлення і I/O 3.5-5.5 В.

Характеристики:

- визначення вологості : 20-90% RH  $\pm$  5% (макс.);
- визначення температури : 0-50 °C  $\pm$  2% (макс.);
- частота опитування : не більше 1 Гц;

- розміри  $15.5 \times 12 \times 5.5$  мм;
- 4 виводи з відстанню між контактами 0.1

Виводи:

- VCC (3-5В живлення);
- Data Out – виведення даних;
- NC – не використовується;
- загальний.

## 2.6 Підключення NodeMCU до комп'ютера

Для початку роботи з NodeMCu потрібно підключити плату до комп'ютера [22-23]. Першим кроком буде установка драйвера CH340 і відкриття Arduino IDE. Потім потрібно вибрати пункт меню «Файл» – «Налаштування» і в вікно «додаткові посилання для менеджера плат» вставити посилання [http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json) (рис. 2.11).

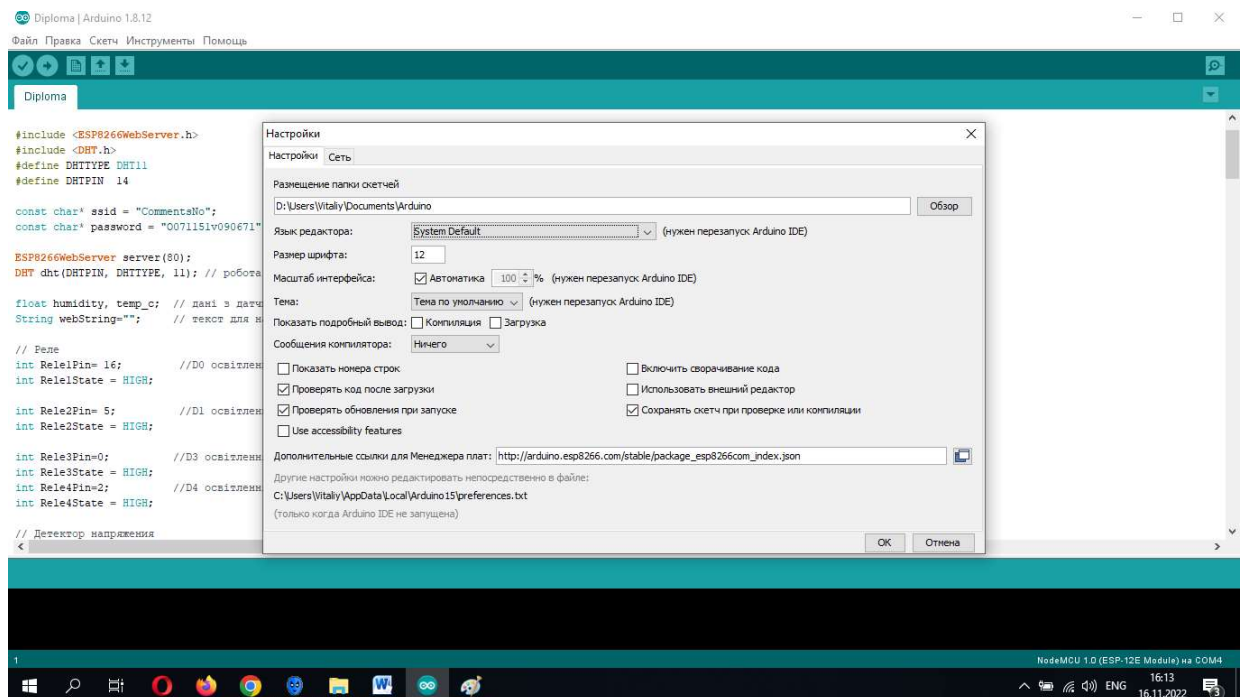


Рисунок 2.11 – «Налаштування»

Переходимо у вікно менеджера плат (рис. 2.12):

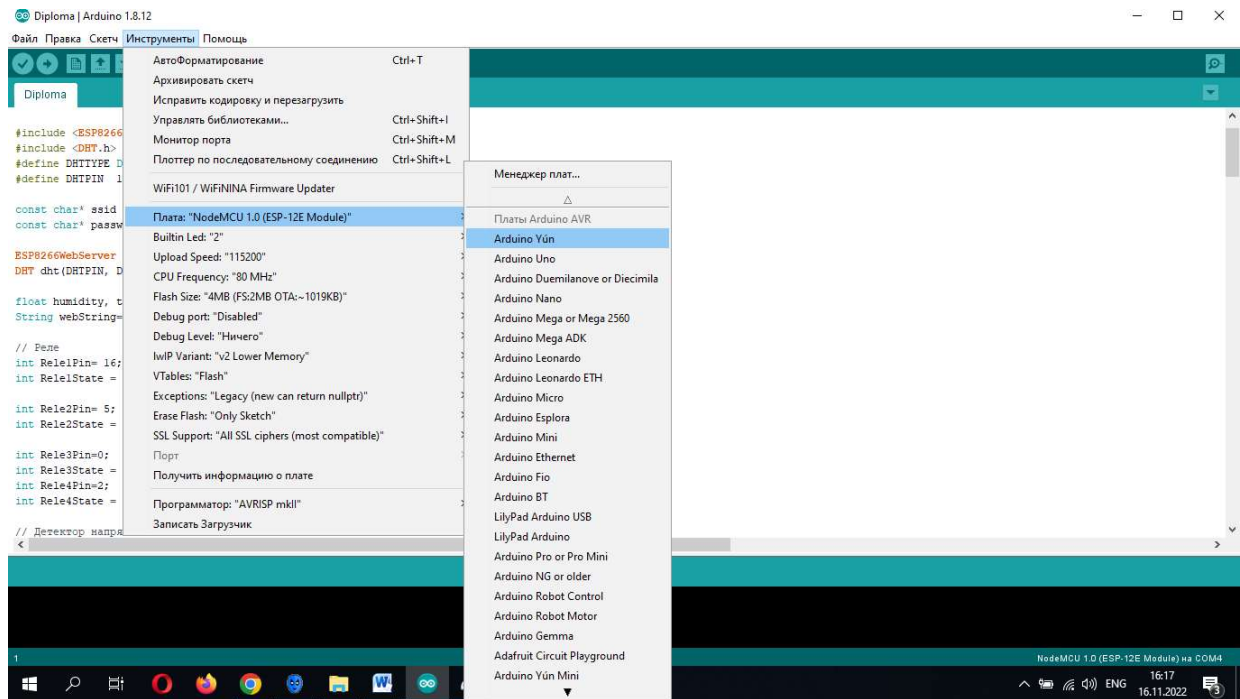


Рисунок 2.12 – Вікно менеджера плат

В рядку пошуку менеджера плат пишемо «esp» (рис. 2.13).

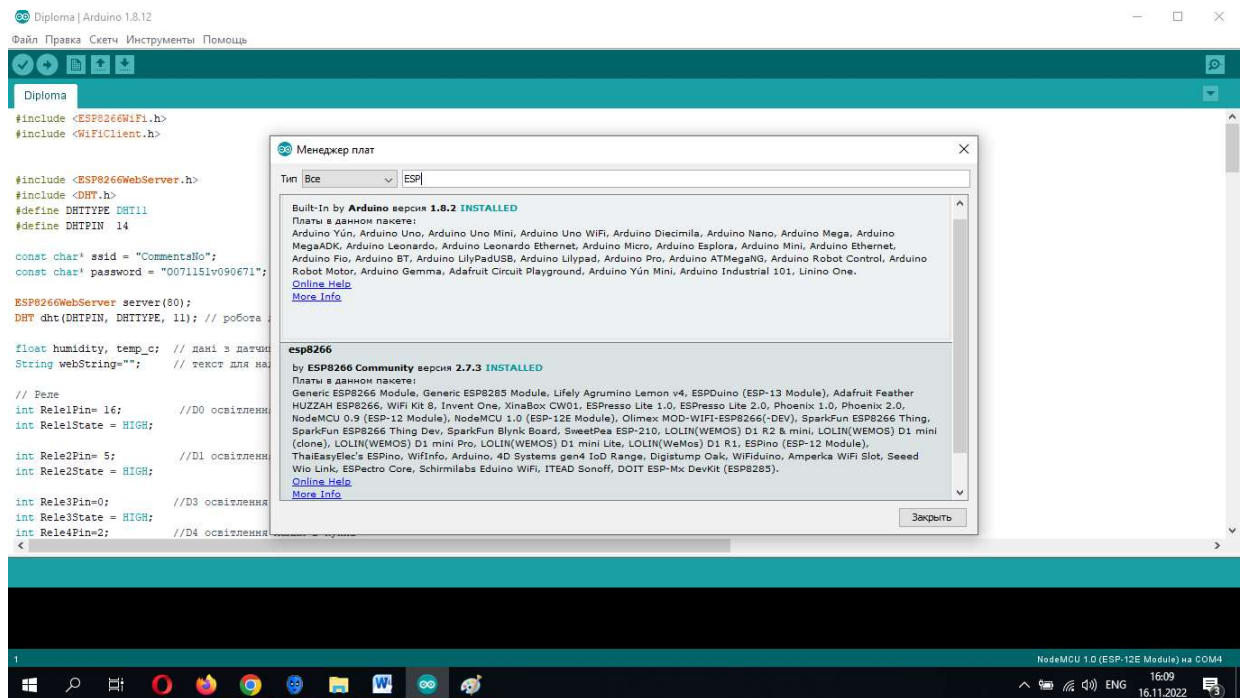


Рисунок 2.13 – Esp8266 by ESP8266 Community версія 2.7.3

У самому низу буде потрібний нам пакет «esp8266 by ESP8266 Community». Вибираємо його і тиснемо кнопку «Установка».

Через деякий час пакет скопіює необхідні файли і в Arduino IDE можна буде вибрати потрібну нам плату (рис. 2.14).

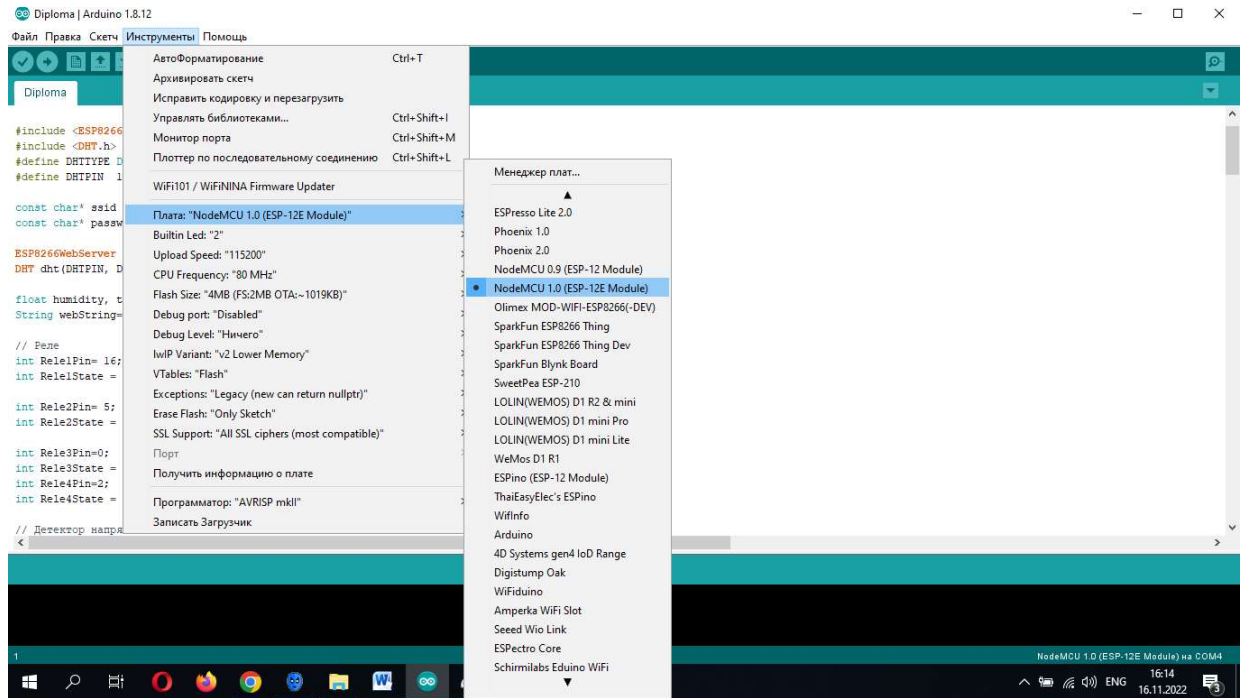


Рисунок 2.14 – NodeMCU 1.0 (ESP-12E Module)

Необхідно в тому ж розділі зайти в розділ «порт» і вибрати той, в який підключена плата (рис. 2.15).

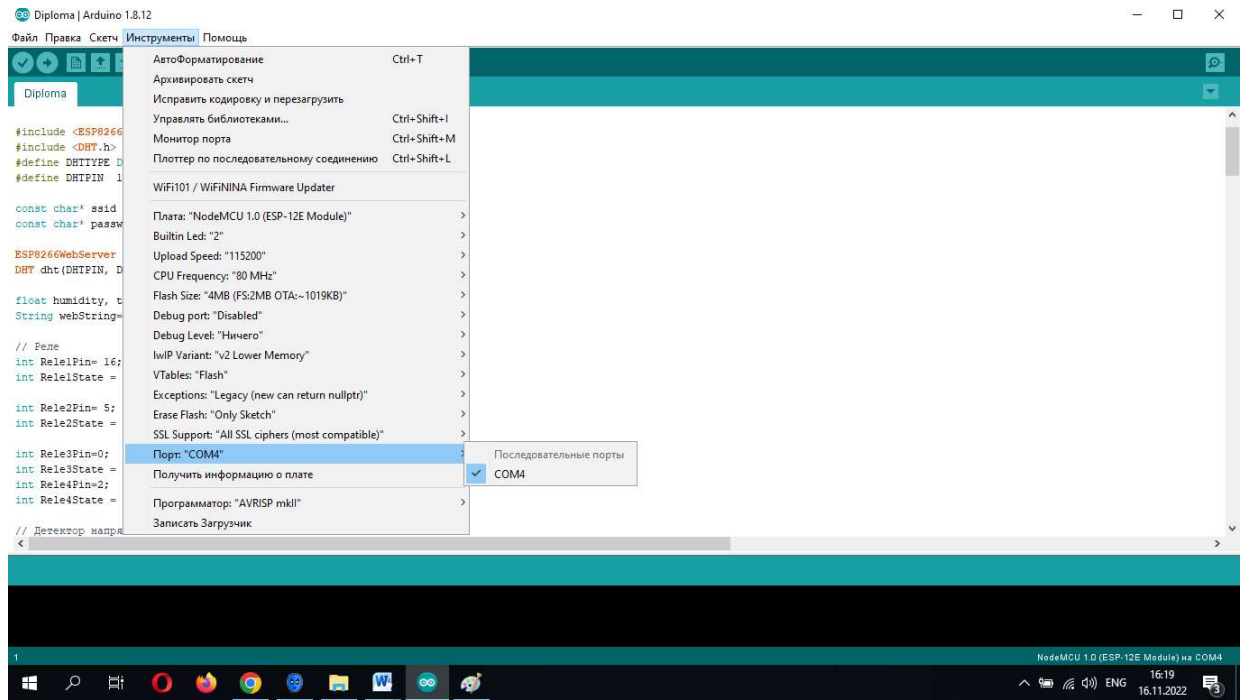


Рисунок 2.15 – Вибір «порта»

Далі встановимо в розділі інструменти: Upload speed (115200 bouad) (рис. 2.16).

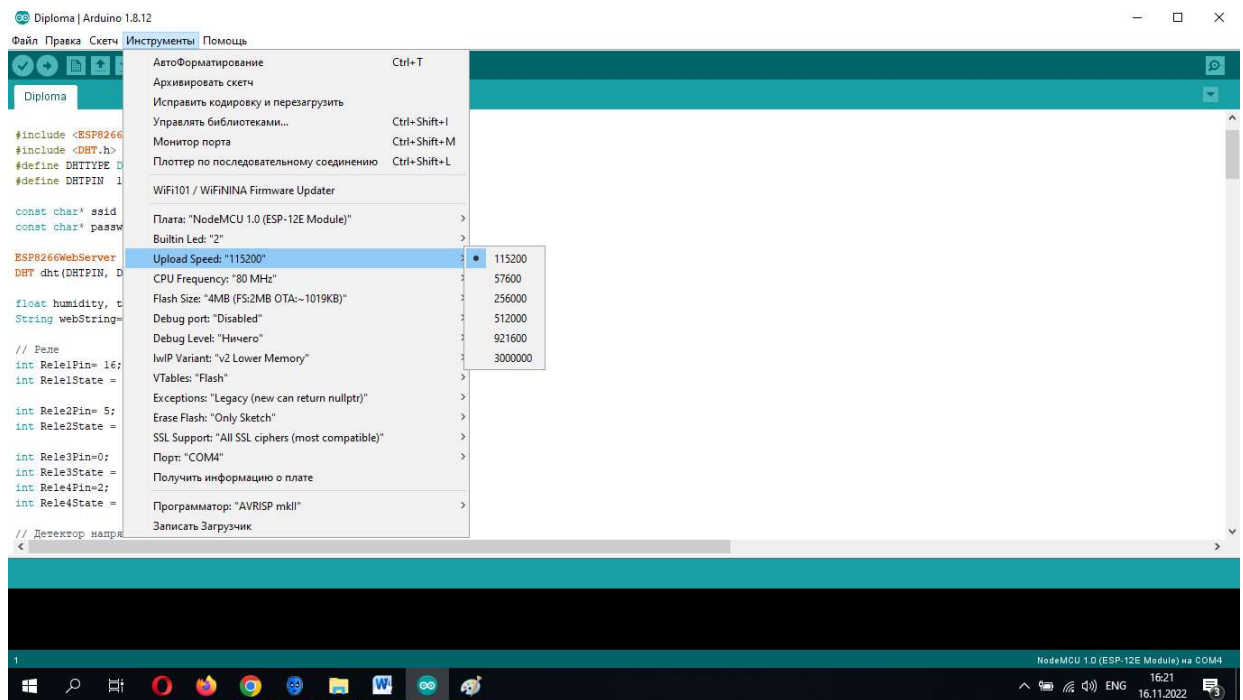


Рисунок 2.16 – Upload speed

## 2.7 Управління світлом

Перше, про що замислюється людина, що бажає додати елементи «розумного будинку» – це керування освітленням [24-26]. В ідеалі мінімальний набір функцій, якими повинно бути наділено «розумне світло» повинен складатися з можливості включати і вимикати світло як з вимикача, розташованого на стіні, так і віддалено за допомогою смартфона.

Таким чином, мінімальний функціонал:

- а) включення / вимикання світла з вимикача;
- б) включення / вимикання світла за допомогою Wi – Fi.

Вмикати і вимикати світло можна будь-яким з перерахованих вище способів, при цьому завжди можна дізнатися стан освітлення (увімкнене) перебуваючи не тільки в будинку, а з будь-якої точки земної кулі, де є Інтернет.

Для реалізації функціоналу мені знадобилися такі елементи:

- NodeMCU – плата на основі модуля ESP8266;
- реле, для управління навантаженням;
- світильник;
- детектор напруги – для визначення наявності напруги;
- прохідний вимикач.
- додатково для вимірювання вологості та температури в ванній кімнаті – датчик DHT11.

Схема підключених пристроїв представлена на рисунку 2.17 – 2.18.

Програму писатимемо в середовищі Arduino IDE (додаток А).



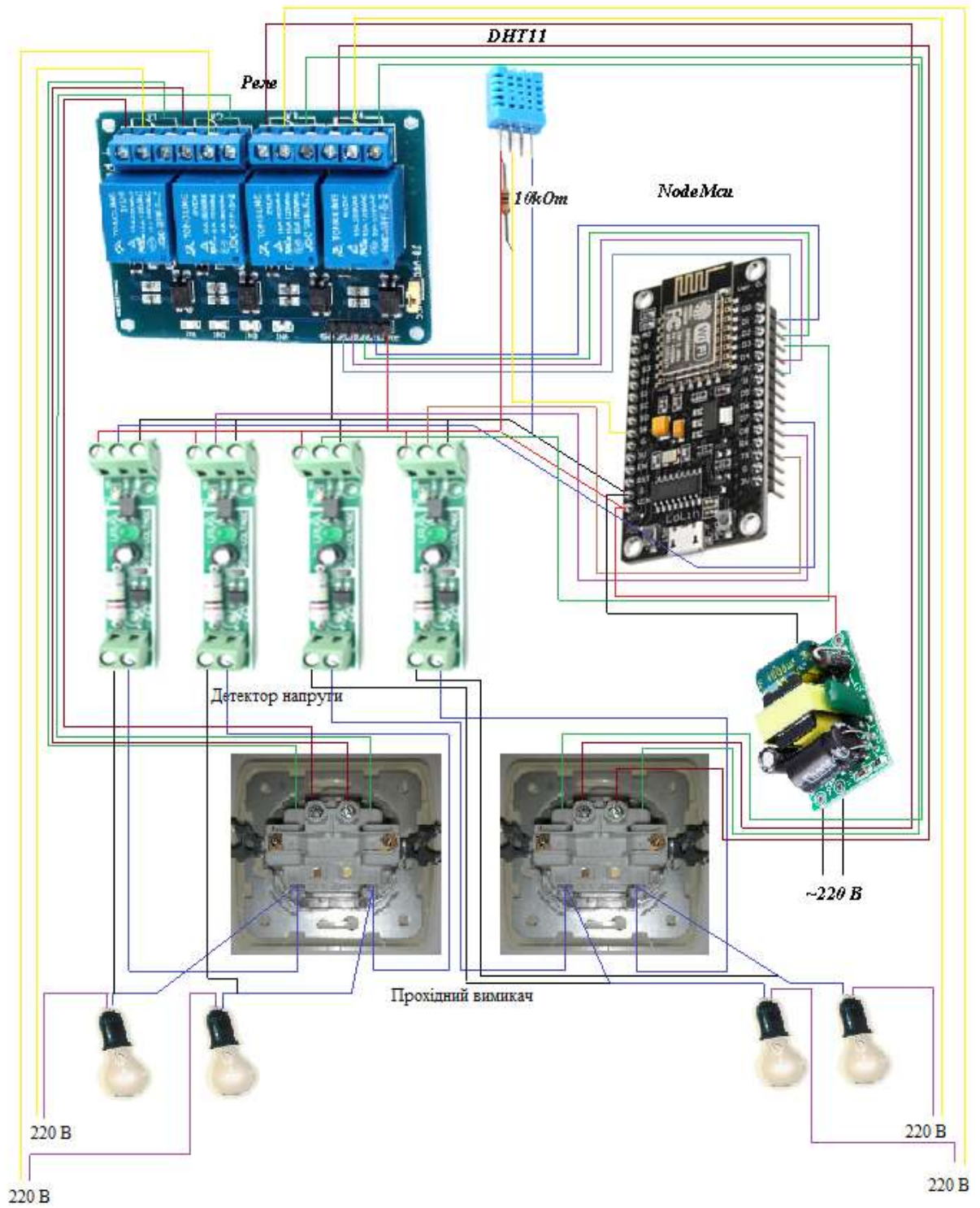


Рисунок 2.17 – Схема підключених пристроїв



Рисунок 2.18 – Макет підключених пристроїв



### 3 АНДРОЇД ДОДАТОК ВКЛЮЧЕННЯ (ВИКЛЮЧЕННЯ) СВІТЛА

Для розробки андроїд додатку який демонструє роботу включення (виключення) світла і відображення температури та вологості в кімнаті будемо використовувати хмарне середовище візуальної розробки App Inventor.

Розробка мобільного додатку в MIT App Inventor <http://ai2.appinventor.mit.edu/> відбувається у 2 етапи. Перший етап – проектування інтерфейсу користувача, другий – програмування компонент додатку.

Використовуючи графічний редактор Design Editor, створюємо зовнішнє оформлення додатку (рис. 3.1), а також визначаємося з опціями та даними, які будуть у додатку.

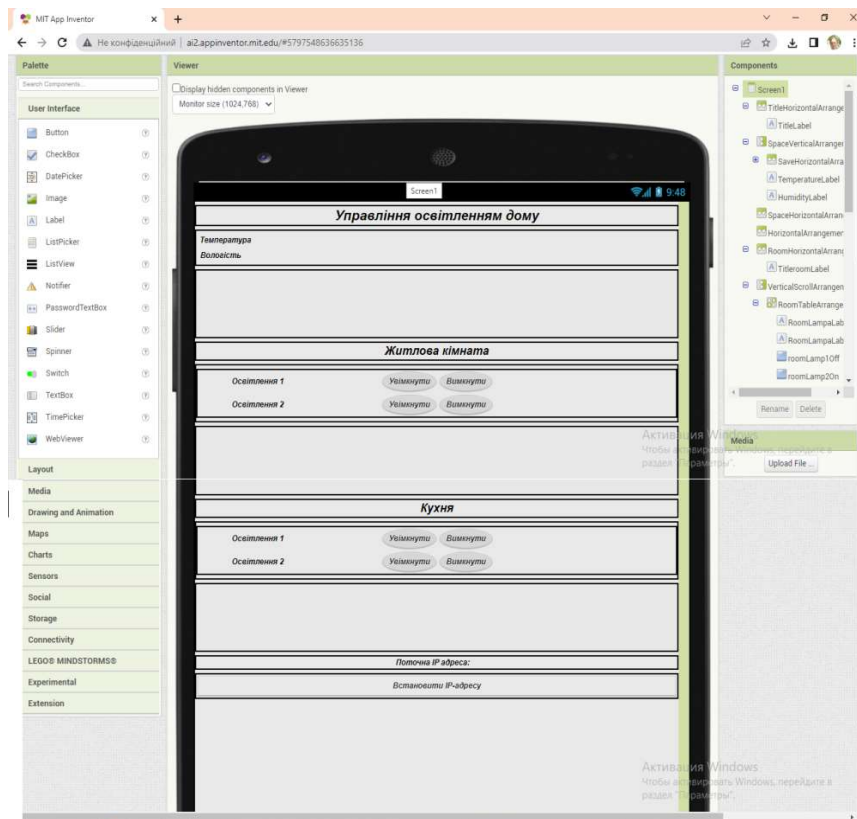


Рисунок 3.1 – Додаток включення (виключення) світла

На рисунку 3.2 представлений блок ініціалізації глобальних змінних: для перевірки підключення змінна `on` рівна «0»; список елементів `list` рівний « » а також змінна помилок.



Рисунок 3.2 – Ініціалізації глобальних змінних

На рисунку 3.3 показані блоки задання Ір-адреси. Якщо натиснута кнопка (`SetIpButton`), значенню змінної `esp_ip` присвоюється значення поля `TextBox1`, а значення змінної `on` стає рівним 1.

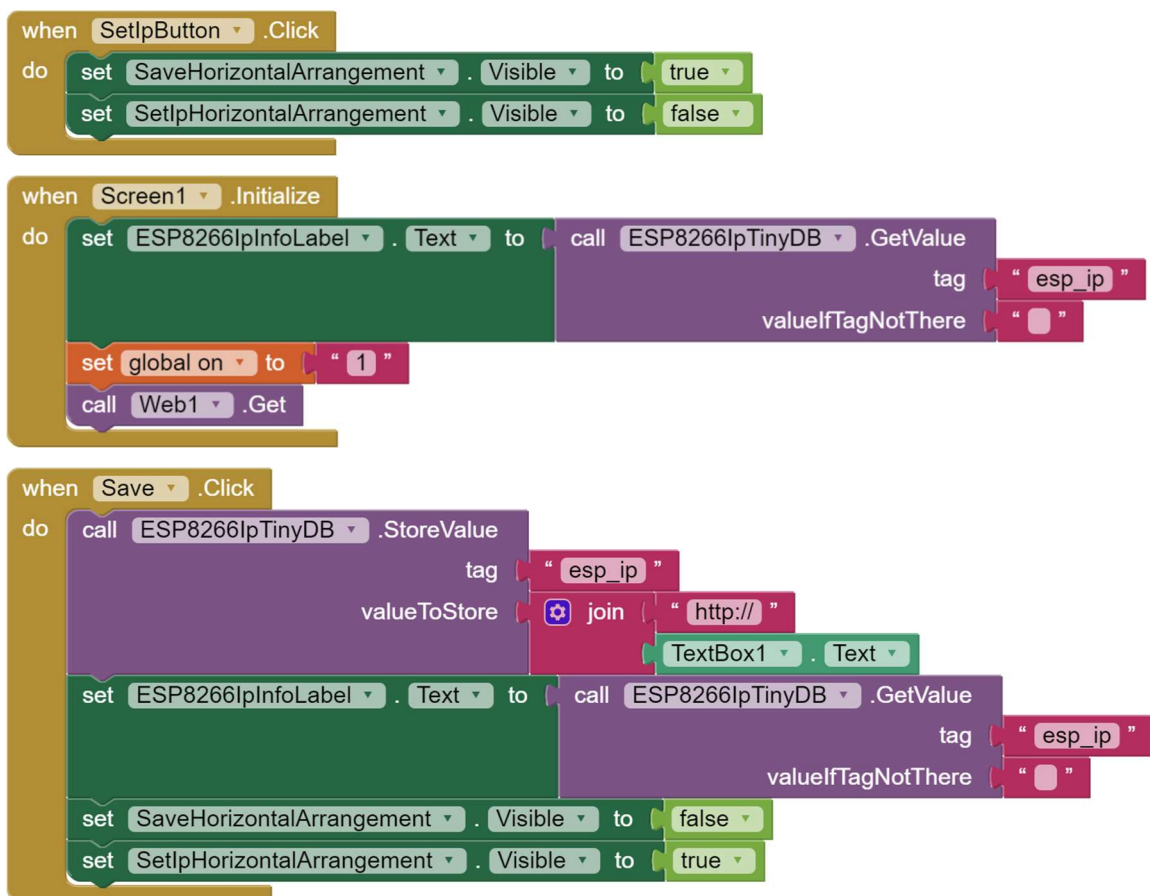


Рисунок 3.3 – Отримання Ір – адреси

Для збереження параметрів додатку (IP-адреси) на Android пристрої використовується компонент TinyDB. Приклад задання налаштувань додатку представлено на рисунку 3.4.



Рисунок 3.4 – Задання Ір-адреси

Наступний блок представлений на рисунку 3.5 виконує функцію будильника Clock.Timer. Clock.TimerInterval визначає, як часто буде викликатись подія Clock.Timer. За замовчуванням для TimerInterval встановлено 1000 мілісекунд (1 секунда).

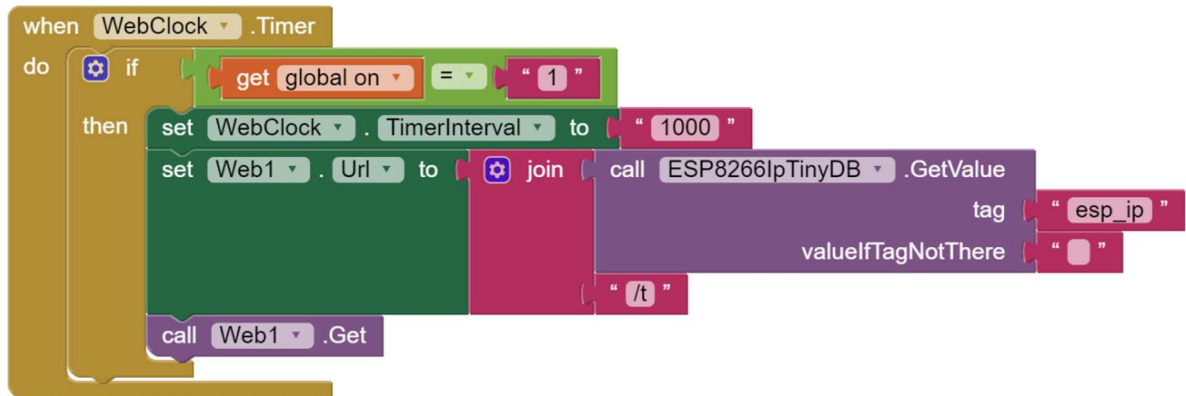


Рисунок 3.5 – Функція Clock.Timer

Блок зображений на рисунку 3.6 – отримання даних температури та вологості, а також стан: включено або вимкнено світло.

Дані температури і вологості, а також стан реле: включено (виключено), значення детектора напруги ми отримуємо у вигляді тексту з роздільником коми. Потім, завдяки роздільнику, ми розбиваємо текст на список з десяти частин. Перша з яких – це температура, друга – вологість, третя, п’ята, сьома, дев’ята – стан реле, четверта, шоста, восьма, десята – значення детектора напруги. Температуру і вологість відображаємо у вигляді напису (рис. 3.7).

Розглянемо визначення стану світла на прикладі однієї лампи. Світло є включеним, якщо виконується умова, що елемент списку з індексом 4 дорівнює 0, тобто на світильнику присутня змінна напруга, відповідно детектор напруги згідно зі схеми розглянутої в попередньому розділі повертає логічний нуль.

В цьому випадку текст “Увімкнути”, який має чорний колір кнопки roomLamp1On змінюється на “Увімкнено” та виводиться зеленим кольором. Текст червоного кольору “Вимкнено” кнопки roomLamp1Off змінюється на текст “Вимкнути” та відображається чорним кольором.

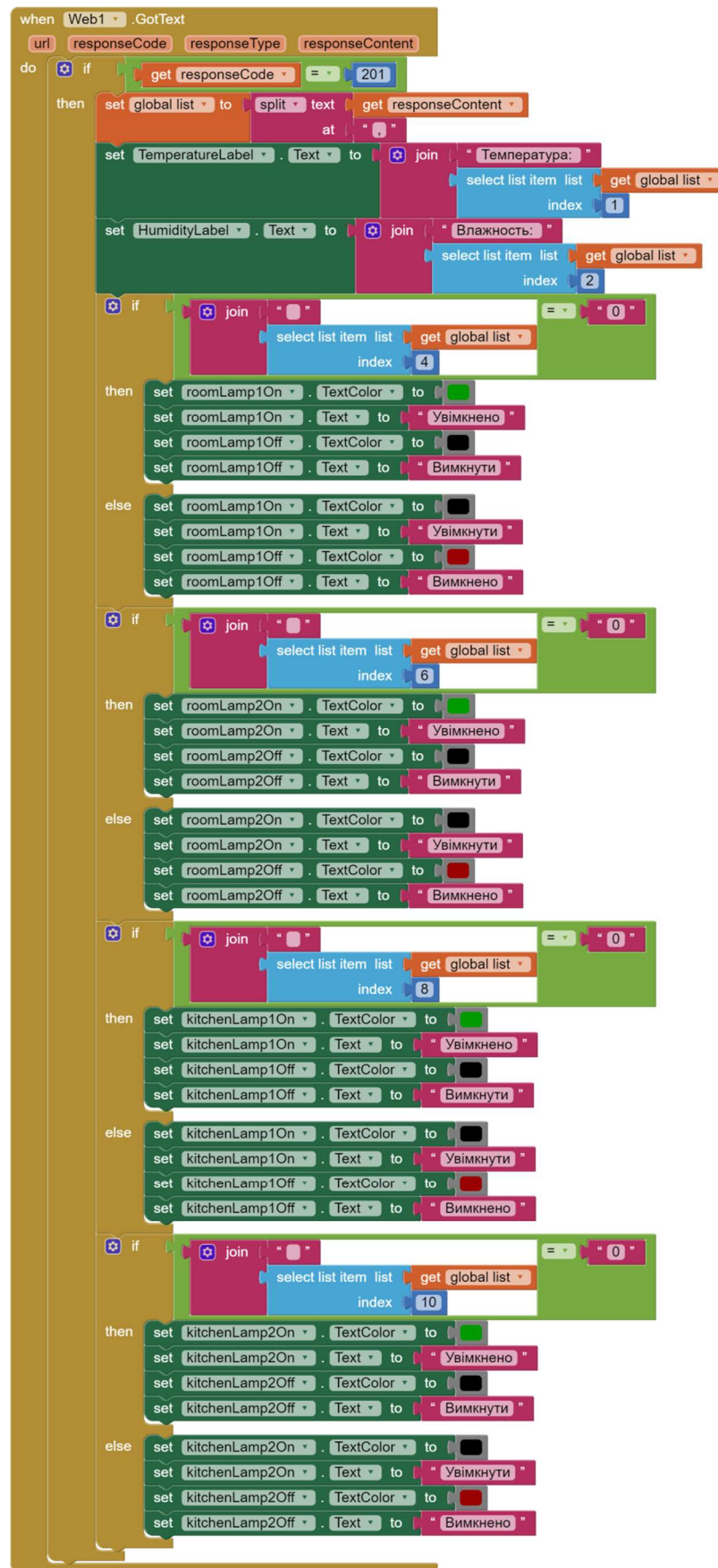


Рисунок 3.6 – Отримання даних температури та вологості, а також стан: включено або вимкнено світло



Рисунок 3.7 – Дані температури та вологості, а також стан: включено або вимкнуто світло

Якщо світло є вимкненим, то текст “Увімкнути” кнопки roomLamp1On виводиться чорним, а текст “Вимкнено” кнопки roomLamp1Off червоним. Відображення даних датчика температури та вологості, а також стан світла здійснюється з інтервалом 1 сек.

На рисунку 3.8 зображено приклад блоку кнопки включення світла.

Визначимо поведінку кнопки включення.

Якщо світло вимкнене:

а) реле знаходиться в стані вимкнено (логічна 1), детектор напруги повертає логічну 1 (немає змінної напруги на лампі), то для того щоб включити світло, нам необхідно відправити get запит включити реле /16on. В результаті реле включено (логічний 0), детектор напруги повертає теж логічний 0;

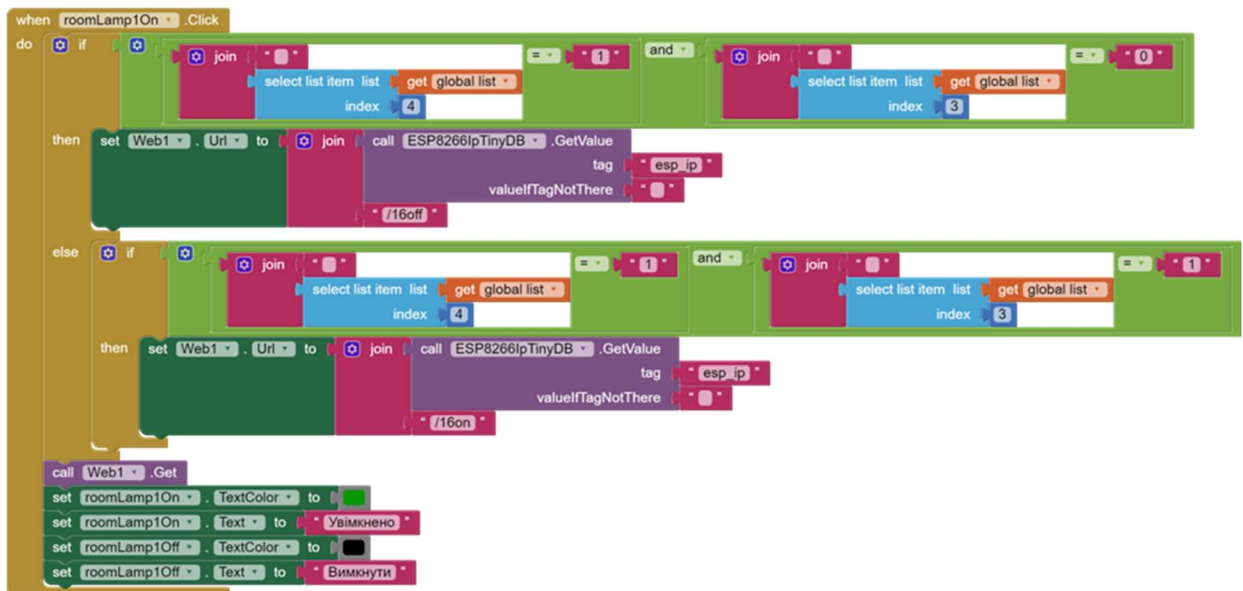


Рисунок 3.8 – Кнопка включення світла

б) реле знаходиться в стані включено (логічний 0), а детектор напруги повертає логічну 1 (вимкнено). (Світло було вимкнено за допомогою прохідного вимикача). Для того, щоб включити світло за допомогою смартфона, нам необхідно надіслати get запит вимкнути реле /16off. Статус реле придбає стан вимкнено (логічна 1), детектор напруги поверне логічний 0 (включено).

Текст “Увімкнути” кнопки roomLamp1On змінюється на “Увімкнено” та виводиться зеленим кольором, а текст “Вимкнено” кнопки roomLamp1Off змінюється на “Вимкнути” і має чорний колір.

На рисунку 3.9 зображено приклад блоку кнопки вимкнення світла.

Визначимо поведінку кнопки вимкнення світла.

Якщо світло включено:

а) реле знаходиться в стані включено (логічний 0), детектор напруги повертає логічний 0 (ε змінна напруга на лампі), то для того щоб вимкнути світло, нам необхідно відправити get запит виключити реле /16off. В результаті реле вимкнено (логічна 1), детектор напруги повертає теж логічну 1;



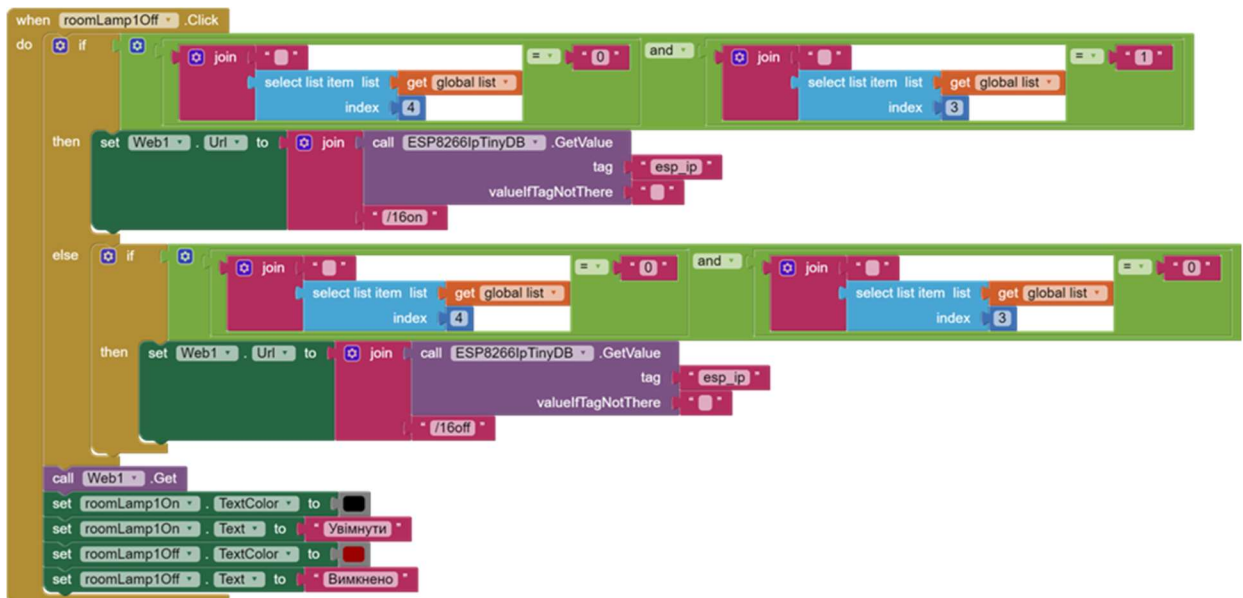


Рисунок 3.9 – Кнопка вимкнення світла

б) реле знаходиться в стані вимкнено (логічна 1), а детектор напруги повертає логічний 0 (включено). (Світло було включено за допомогою прохідного вимикача). Для того, щоб вимкнути світло за допомогою смартфона, нам необхідно надіслати get запит включити реле /16on. Статус реле придбає стан включено (логічний 0), детектор напруги поверне логічну 1 (вимкнено).

Текст “Увімкнено” кнопки roomLamp1On змінюється на “Увімнути” та виводиться чорним кольором, а текст “Вимкнути” кнопки roomLamp1Off змінюється на “Вимкнути” і відображається червоним кольором.

Розроблений додаток який демонструє роботу включення (виключення) світла та відображення температури та вологості в кімнаті відображений на рисунку 3.10.





Рисунок 3.10 – Android додаток включення (виключення) світла та відображення температури та вологості в кімнаті

Android програма управління світлом наведена в додатку Б.

## ВИСНОВКИ

В результаті виконання кваліфікаційної роботи мною були отримані наступні результати:

- провести аналіз основних додатків для керування світлом в Україні;
- розроблено скетч для прошивки плати на основі модуля ESP8266 написаний в середовищі Arduino IDE;
- створено андроїд додаток, який демонструє роботу включення (виключення) світла в кімнаті. Андроїд додаток реалізовано за допомогою хмарного середовища візуальної розробки Android додатків – App Inventor.

## ПЕРЕЛІК ПОСИЛАНЬ

1. Smart Life. URL: <https://play.google.com/store/apps/details?id=com.tuya.smartlife&hl=ru&gl=US> (дата звернення: 03.09.2022).
2. Короткий посібник з керування додатком Smart Life. URL: <https://tcl-aircon.ua/upload/iblock/c70/mgksdlxgei0ke033j2c54x330txsri3v.pdf> (дата звернення: 11.07.2022).
3. Додаток Smart Life для розумного будинку. URL: [https://bcvolt.com/index.php?route=blog/article&article\\_id=120](https://bcvolt.com/index.php?route=blog/article&article_id=120) (дата звернення: 23.08.2022).
4. Додаток Philips Hue. URL: <https://play.google.com/store/apps/details?id=com.philips.lighting.hue2&hl=ru&gl=US> (дата звернення: 05.10.2022).
5. Додаток Mi Home. URL: <https://play.google.com/store/apps/details?id=com.xiaomi.smarthome&hl=uk&gl=US> (дата звернення: 05.10.2022).
6. Додаток Ecobee. URL: <https://play.google.com/store/apps/details?id=com.ecobee.athenamobile&hl=uk&gl=US> (дата звернення: 05.10.2022).
7. Додаток Wemo. URL: <https://play.google.com/store/apps/details?id=com.belkin.wemoandroid&hl=ru&gl=US> (дата звернення: 05.10.2022).
8. Додаток Yonomi. URL: <https://play.google.com/store/apps/details?id=com.yonomi&hl=ru&gl=US> (дата звернення: 15.10.2022).
9. Додаток ELARI SmartHome. URL: <https://play.google.com/store/apps/details?id=com.elari.smarthome&hl=uk&gl=US> (дата звернення: 15.10.2022).
10. Додаток Livicom. URL: <https://play.google.com/store/apps/details?id=ru.livicom&hl=uk&gl=US> (дата звернення: 15.10.2022).
11. Додаток Eufy Clean. URL: <https://play.google.com/store/apps/details?id=com.eufylife.smarthome&hl=ru&gl=US> (дата звернення: 15.10.2022).
12. Додаток Life Control 2.0 URL: <https://apps.apple.com/ua/app/life-control-2-0/id1460169096?l=ru> (дата звернення: 15.10.2022).

13. Інтернет-радіоринок "Електроніка". URL: <https://electronica.in.ua/ua/p1577526733-modul-nodemcu-esp8266.html> (дата звернення: 25.08.2022).
14. Insight Into ESP8266 NodeMCU Features & Using It With Arduino IDE. URL: <https://lastminuteengineers.com/esp8266-nodemcu-arduino-tutorial/> (дата звернення: 16.07.2022).
15. Реле. Устройство и принцип работы. Электромагнитное и твердотельное реле. URL: <https://ampermarket.kz/base/relay/> (дата звернення: 11.10.2022).
16. Прохідний вимикач. URL: [https://uk.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%85%D1%96%D0%B4%D0%BD%D0%B8%D0%B9\\_%D0%B2%D0%B8%D0%BC%D0%B8%D0%BA%D0%B0%D1%87](https://uk.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%85%D1%96%D0%B4%D0%BD%D0%B8%D0%B9_%D0%B2%D0%B8%D0%BC%D0%B8%D0%BA%D0%B0%D1%87) (дата звернення: 08.11.2022).
17. Амперок. URL: <https://amperok.com.ua/scho-take-prohidnij-vimikach/> (дата звернення: 21.08.2022).
18. Arduino.ua. URL: <https://arduino.ua/prod2141-detektor-napryajeniya-220v> (дата звернення: 21.08.2022).
19. Урок 16. Подключение датчика DHT11 к Arduino UNO. Выводим температуру и влажность на LCD 1602 I2C дисплей. URL: <https://lesson.iarduino.ru/page/podklyuchenie-datchika-dht11-k-arduino-uno-vyvodim-temperaturu-i-vlazhnost-na-lcd-1602-i2c-displey/> (дата звернення: 12.09.2022).
20. Датчик температуры и влажности Ардуино DHT11. URL: <https://xn--18-6kcdusowgbt1a4b.xn--p1ai/%D0%B4%D0%B0%D1%82%D1%87%D0%B8%D0%BA-%D1%82%D0%B5%D0%BC%D0%BF%D0%B5%D1%80%D0%B0%D1%82%D1%83%D1%80%D1%8B-%D0%B8-%D0%B2%D0%BB%D0%B0%D0%B6%D0%BD%D0%BE%D1%81%D1%82%D0%B8/> (дата звернення: 12.09.2022).
21. Подключение датчика DHT11 или DHT22 к Ардуино. URL: <https://arduinomaster.ru/datchiki-arduino/datchiki-temperatury-i-vlazhnosti-dht11-dht22/> (дата звернення: 21.08.2022).

22. NodeMCU (ESP8266) для начинающих: что такое, как подключить.  
URL: <http://edurobots.ru/2017/04/nodemcu-esp8266/> (дата  
звернения: 12.09.2022)

23. Esp8266 nodemcu v3 подключение датчиков. URL:  
<https://vemiru.ru/info/esp8266-nodemcu-v3-podkljuchenie-datchikov/> (дата  
звернения: 12.09.2022)

24. Макаров С. Л. Arduino Uno и Raspberry Pi 3. От схемотехники к  
интернету вещей. ДМК Пресс, 2019. 202 с.

25. Шварц Марко. Интернет вещей с ESP8266. Санкт-Петербург:  
БХВ-Петербург, 2018. 192 с.

26. Петин В. А Создание Умного дома на базе Arduino. Litres, 2019.  
180 с.

## ДОДАТОК А

Скетч для прошивки плати на основі модуля ESP8266

```

#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266WebServer.h>
#include <DHT.h>
#define DHTTYPE DHT11
#define DHTPIN 14
const char* ssid = "*****";
const char* password = "*****";
ESP8266WebServer server(80);
DHT dht(DHTPIN, DHTTYPE, 11); // робота датчика з esp8266
float humidity, temp_c; // дані з датчика
String webString=""; // текст для надсилання
// Реле
int Rele1Pin= 16; //D0 освітлення лампи 1 житлової кімнати
int Rele1State = HIGH;
int Rele2Pin= 5; //D1 освітлення лампи 2 житлової кімнати
int Rele2State = HIGH;
int Rele3Pin=0; //D3 освітлення лампи 1 кухні
int Rele3State = HIGH;
int Rele4Pin=2; //D4 освітлення лампи 2 кухні
int Rele4State = HIGH;
// Детектор напруги
int Voltage_Sensor_1=LOW;
int Voltage_Sensor_Pin1=12; //D6 освітлення лампи 1 житлової кімнати
int Voltage_Sensor_2=LOW;

```

```

int Voltage_Sensor_Pin2=13; //D7 освітлення лампи 2 житлової кімнати
int Voltage_Sensor_3=LOW;
int Voltage_Sensor_Pin3=4; //D2 освітлення лампи 1 кухні
int Voltage_Sensor_4=LOW;
int Voltage_Sensor_Pin4=1; //Tx освітлення лампи 2 кухні
unsigned long previousMillis = 0; // час останнього читання
const long interval = 2000; // інтервал читання з датчика
void setup() {
Serial.begin(9600);
dht.begin(); // ініціалізуємо датчик температури
// готуємо GPI для реле
pinMode(Voltage_Sensor_Pin1,INPUT); //Налаштовує цей контакт на вхід
// (12-D6) освітлення лампи 1 житлової кімнати
digitalWrite(Voltage_Sensor_Pin1,Voltage_Sensor_1); // 0 В – LOW
pinMode(Voltage_Sensor_Pin2,INPUT); //Налаштовує цей контакт на вхід
// (13-D7) освітлення лампи 2 житлової кімнати
digitalWrite(Voltage_Sensor_Pin2,Voltage_Sensor_2); // 0 В – LOW
pinMode(Voltage_Sensor_Pin3,INPUT); //Налаштовує цей контакт на вхід
// (4-D2) освітлення лампи 1 кухні
digitalWrite(Voltage_Sensor_Pin3,Voltage_Sensor_3); // 0 В – LOW
pinMode(Voltage_Sensor_Pin4,INPUT); //Налаштовує цей контакт на вхід
// (1-Tx) освітлення лампи 2 кухні
digitalWrite(Voltage_Sensor_Pin4,Voltage_Sensor_4); // 0 В – LOW
pinMode(Rele1Pin, OUTPUT); //D0 освітлення лампи 1 житлової кімнати,
// налаштовує цей контакт на вихід (16-D0)
digitalWrite(Rele1Pin,Rele1State); // 5 В – HIGH
pinMode(Rele2Pin, OUTPUT); //D1 освітлення лампи 2 житлової кімнати,
// налаштовує цей контакт на вихід (5-D1)
digitalWrite(Rele2Pin,Rele2State); // 5 В – HIGH
pinMode(Rele3Pin, OUTPUT); //D3 освітлення лампи 1 кухні, налаштовує цей

```

```

        // контакт на вихід (0-D3)
digitalWrite(Rele3Pin,Rele3State); // 5 В – HIGH
pinMode(Rele4Pin, OUTPUT); //D4 освітлення лампи 2 кухні, налаштовує цей
        // контакт на вихід (2-D4)
digitalWrite(Rele4Pin,Rele4State);
Serial.println();
Serial.println();
Serial.print("Connecting to ");
Serial.println(ssid);
WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) {
delay(500);
Serial.print(".");
}
Serial.println("");
Serial.println("WiFi connected");

// стартує сервер
server.begin();
Serial.println("Server started");
// виводимо отриманий IP
Serial.print("IP address: ");
Serial.println(WiFi.localIP());
server.on("/16on", rel1_on);
server.on("/16off", rel1_off);
server.on("/5on", rel2_on);
server.on("/5off", rel2_off);
server.on("/0on", rel3_on);
server.on("/0off", rel3_off);
server.on("/2on", rel4_on);

```



```

server.on("/2off", rel4_off);
server.on("/t", data);
}
void loop() {
server.handleClient();
Rel1State= digitalRead(Rel1Pin); //Зчитує значення із зазначеного
// цифрового виводу чи , HIGH чи LOW 16-D0.
// Освітлення лампи 1 житлової кімнати
digitalWrite(Rel1Pin,Rel1State); // Запишіть HIGH або LOW значення на
// цифровий контакт.

Rel2State= digitalRead(Rel2Pin); //Зчитує значення із зазначеного
// цифрового виводу чи , HIGH чи LOW 5-D1.
// Освітлення лампи 2 житлової кімнати
digitalWrite(Rel2Pin,Rel2State);
Rel3State= digitalRead(Rel3Pin); //Зчитує значення із зазначеного
// цифрового виводу чи , HIGH чи LOW 0-D3.
// Освітлення лампи 1 кухні
digitalWrite(Rel3Pin,Rel3State);
Rel4State= digitalRead(Rel4Pin); //Зчитує значення із зазначеного
// цифрового виводу чи , HIGH чи LOW 2-D4.
// Освітлення лампи 2 кухні
digitalWrite(Rel4Pin,Rel4State);
Voltage_Sensor_1 = digitalRead(Voltage_Sensor_Pin1); //перевіряємо за
// допомогою датчика наявність напруги 12-D6.
// Освітлення лампи 1 житлової кімнати
digitalWrite(Voltage_Sensor_Pin1,Voltage_Sensor_1);
Voltage_Sensor_2 = digitalRead(Voltage_Sensor_Pin2); //перевіряємо за
// допомогою датчика наявність напруги 13-D7.
// Освітлення лампи 2 житлової кімнати

```



```
}  
void rel1_off() {  
digitalWrite(Rele1Pin,HIGH ); // вимкнути освітлення 1 житлової кімнати  
server.send(200, "text/html","gpio16 off");  
}  
void rel2_on() {  
digitalWrite(Rele2Pin, LOW); // увімкнути освітлення 2 житлової кімнати  
server.send(200, "text/html","gpio5 on");  
}  
void rel2_off() {  
digitalWrite(Rele2Pin, HIGH); // вимкнути освітлення 2 житлової кімнати  
server.send(200, "text/html","gpio5 off");  
}  
void rel3_on() {  
digitalWrite(Rele3Pin, LOW);  
server.send(200, "text/html","gpio0 on"); // увімкнути освітлення 1 кухні  
}  
void rel3_off() {  
digitalWrite(Rele3Pin, HIGH);  
server.send(200, "text/html","gpio0 off"); // вимкнути освітлення 1 кухні  
}  
void rel4_on() {  
digitalWrite(Rele4Pin, LOW);  
server.send(200, "text/html","gpio2 on"); // увімкнути освітлення 2 кухні  
}  
void rel4_off() {  
digitalWrite(Rele4Pin, HIGH);  
server.send(200, "text/html","gpio2 off"); // вимкнути освітлення 2 кухні  
}  
//строка значень датчиків і стану реле
```

```
void data() {  
  webString = String((int)temp_c)+" "+String((int)humidity)+" "+  
  String((int)Rele1State) +" "+ String((int)  
  Voltage_Sensor_1)+" "+String((int)Rele2State) +" "+ String((int)  
  Voltage_Sensor_2)+" "+ String((int)Rele3State) +" "+ String((int)  
  Voltage_Sensor_3)+" "+ String((int)Rele4State) +" "+ String((int)  
  Voltage_Sensor_4);  
  Serial.print (String((int)temp_c)+" "+String((int)humidity)+" "+String((int)  
  Rele1State) +" "+String((int) Voltage_Sensor_1)+" "+String((int)Rele2State)  
  +" "+String((int) Voltage_Sensor_2)+" "+String((int) Rele3State) +" "+String((int)  
  Voltage_Sensor_3)+" "+String((int) Rele4State) +" "+String((int)  
  Voltage_Sensor_4));  
  server.send(201, "text/plain", webString);  
}
```

## ДОДАТОК Б

Додаток, який демонструє роботу включення (виключення) світла в кімнаті

```

initialize global on to " 0 "
initialize global list to " "
initialize global err to " "

when SetIpButton .Click
do
  set SaveHorizontalArrangement . Visible to true
  set SetIpHorizontalArrangement . Visible to false

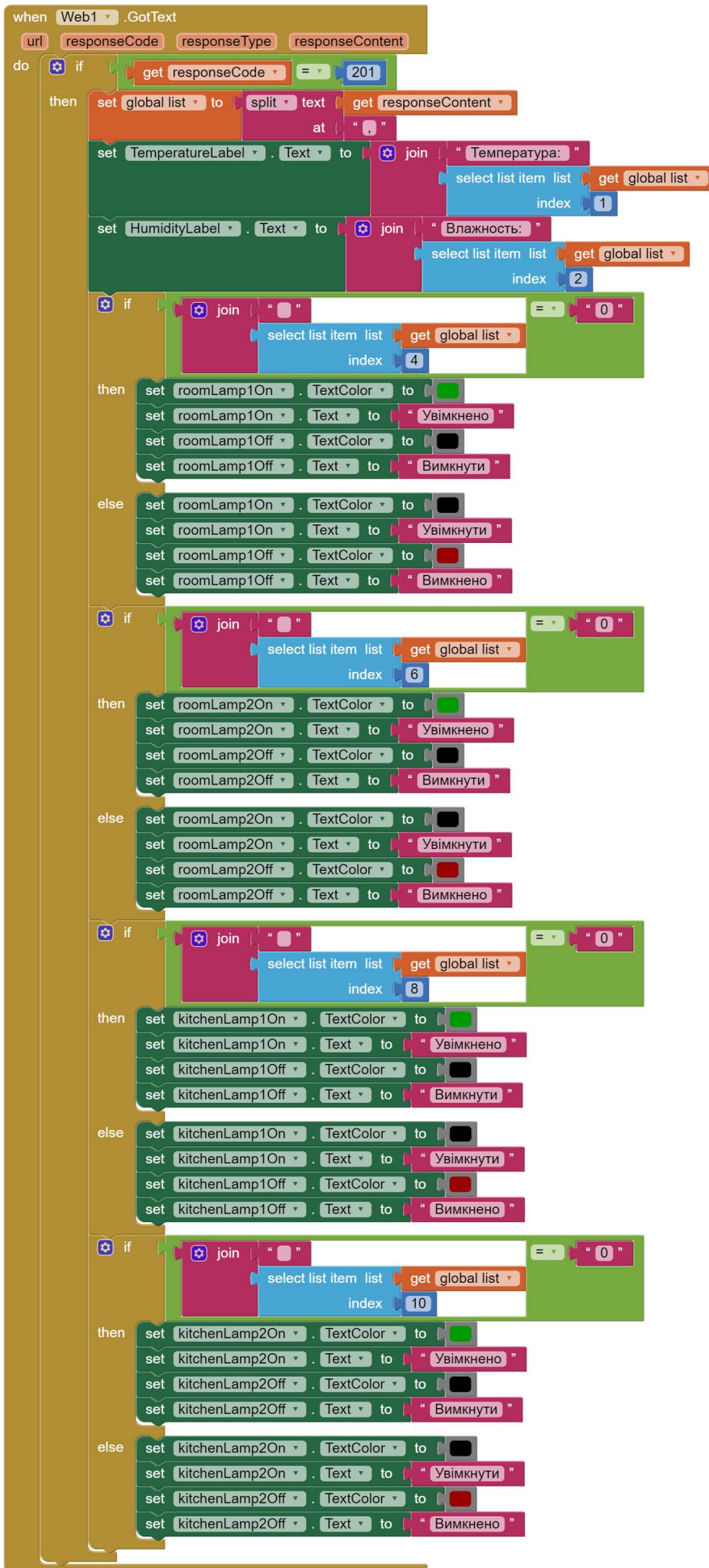
when Screen1 .Initialize
do
  set ESP8266IpInfoLabel . Text to call ESP8266IpTinyDB .GetValue
  tag " esp_ip "
  valueIfTagNotThere " "
  set global on to " 1 "
  call Web1 .Get

when Screen1 .ErrorOccurred
component functionName errorNumber message
do
  set global err to join
  get component
  get functionName
  get errorNumber
  get message

when Save .Click
do
  call ESP8266IpTinyDB .StoreValue
  tag " esp_ip "
  valueToStore join " http://"
  TextBox1 . Text
  set ESP8266IpInfoLabel . Text to call ESP8266IpTinyDB .GetValue
  tag " esp_ip "
  valueIfTagNotThere " "
  set SaveHorizontalArrangement . Visible to false
  set SetIpHorizontalArrangement . Visible to true

```

```
when WebClock .Timer
do
  if
  then
    set WebClock . TimerInterval to "1000"
    set Web1 . Url to join
      call ESP8266IpTinyDB .GetValue
      tag "esp_ip"
      valueIfTagNotThere ""
      "/t"
    call Web1 .Get
```



when roomLamp1On .Click

```

do
  if
    join " " select list item list get global list index 4 = "1" and join " " select list item list get global list index 3 = "0"
  then
    set Web1 . Uri to join call ESP8266pTinyDB .GetValue tag "esp_ip" valueIfTagNotThere "/16off"
  else
    if
      join " " select list item list get global list index 4 = "1" and join " " select list item list get global list index 3 = "1"
    then
      set Web1 . Uri to join call ESP8266pTinyDB .GetValue tag "esp_ip" valueIfTagNotThere "/16on"
    call Web1 .Get
    set roomLamp1On . TextColor to
    set roomLamp1On . Text to "Увімкнено"
    set roomLamp1Off . TextColor to
    set roomLamp1Off . Text to "Вимкнути"
  
```

when roomLamp1Off .Click

```

do
  if
    join " " select list item list get global list index 4 = "0" and join " " select list item list get global list index 3 = "1"
  then
    set Web1 . Uri to join call ESP8266pTinyDB .GetValue tag "esp_ip" valueIfTagNotThere "/16on"
  else
    if
      join " " select list item list get global list index 4 = "0" and join " " select list item list get global list index 3 = "0"
    then
      set Web1 . Uri to join call ESP8266pTinyDB .GetValue tag "esp_ip" valueIfTagNotThere "/16off"
    call Web1 .Get
    set roomLamp1On . TextColor to
    set roomLamp1On . Text to "Увімнути"
    set roomLamp1Off . TextColor to
    set roomLamp1Off . Text to "Вимкнено"
  
```

when roomLamp2On .Click

```

do
  if
    join " " select list item list get global list index 6 = "1" and join " " select list item list get global list index 5 = "0"
  then
    set Web1 . Uri to join call ESP8266pTinyDB .GetValue tag "esp_ip" valueIfTagNotThere "/5off"
  else
    if
      join " " select list item list get global list index 6 = "1" and join " " select list item list get global list index 5 = "1"
    then
      set Web1 . Uri to join call ESP8266pTinyDB .GetValue tag "esp_ip" valueIfTagNotThere "/5on"
    call Web1 .Get
    set roomLamp2On . TextColor to
    set roomLamp2On . Text to "Увімкнено"
    set roomLamp2Off . TextColor to
    set roomLamp2Off . Text to "Вимкнути"
  
```



