

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ ТА ТЕХНОЛОГІЙ**

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Пристрій з використанням ESP для дистанційного керування приладами
в мережах IoT»

на здобуття освітнього ступеня бакалавра
зі спеціальності 126 Інформаційні системи та технології
(код, найменування спеціальності)

освітньо-професійної програми Інформаційні системи та технології
(назва)

*Кваліфікаційна робота містить результати власних досліджень.
Використання ідей, результатів і текстів інших авторів мають посилання на
відповідне джерело*

(підпис)

Іван УЖАНОВ
Ім'я, ПРІЗВИЩЕ здобувача

Виконав: здобувач вищої освіти гр. ІСД-41
Іван УЖАНОВ

Ім'я, ПРІЗВИЩЕ

Керівник: Катерина НЕСТЕРЕНКО
Доктор технічних наук, Ім'я, ПРІЗВИЩЕ
професор

Рецензент: _____
науковий ступінь, Ім'я, ПРІЗВИЩЕ
вчене звання

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
Навчально-науковий інститут інформаційних технологій**

Кафедра Інформаційні системи та технології

Ступінь вищої освіти Бакалавр

Спеціальність Інформаційні системи та технології

Освітньо-професійна програма Інформаційні системи та технології

ЗАТВЕРДЖУЮ

Завідувач кафедру

_____ Каміла СТОРЧАК

«___» _____ 20__ р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

Ужанов Іван Олегович

(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи: Пристрій з використанням ESP для дистанційного керування приладами в мережах IoT

керівник кваліфікаційної роботи Катерина НЕСТЕРЕНКО, док. тех.наук, професор,
(Ім'я, ПРІЗВИЩЕ, науковий ступінь, вчене звання)

затверджені наказом Державного університету інформаційно-комунікаційних технологій від «24» Лютого 2025 р. № 56

2. Строк подання кваліфікаційної роботи «30» травня 2025 р.

3. Вихідні дані до кваліфікаційної роботи:

Актуальні аналітичні та статистичні дані про розвиток ринку Інтернету речей (IoT) за 2023–2025 роки від компаній The Business Research Company, Transforma Insights, Statista.

Технічна документація на мікроконтролер ESP8266 та модуль NodeMCU версії Lolin

Специфікації і документації крокового двигуна 28BYJ-48 і драйвера ULN2003

Документація середовища Blynk

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

4.1. Аналіз сучасного стану та перспектив розвитку технологій Інтернету речей (IoT), сфер їх застосування, обсягів ринку та динаміки поширення приладів.

4.2. Виготовлення та опис реалізації апаратної частини пристрою пристрою

4.3. Опис розробки програмного забезпечення пристрою

4.4 Реалізація дистанційного керування пристрою через застосунок Blynk

5. Перелік ілюстративного матеріалу: презентація
- 5.1 Титульний слайд
- 5.2 Мета, Предмет та Об'єкт роботи
- 5.3 Глобальний ринок інтернету речей
- 5.4 Реалії розповсюдженості девайсів інтернету речей
- 5.5 Складові дистанційного перемикача
- 5.6 Схема підключення компонентів
- 5.7 Корпус пристрою
- 5.8 Алгоритм роботи програмного забезпечення пристрою
- 5.9 Код програмного забезпечення пристрою
- 5.10 Висновки
- 5.11 Реалізовані тези
6. Дата видачі завдання «24» лютого 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Підбір науково-технічної літератури	25.02.2025	Виконано
2	Дослідження розповсюдженості пристроїв концепції IoT	01.03.2025	Виконано
3	Вивчення технологічного середовища для розробки приладу	06.03.2025	Виконано
4	Розробка та реалізація апаратної складової приладу	14.03.2025	Виконано
5	Розробка та реалізація програмного забезпечення приладу	23.03.2025	Виконано
6	Розробка та реалізація інтерфейсу для середовища Blynk	26.03.2025	Виконано
7	Написання трьох розділів кваліфікаційної роботи	15.04.2025	Виконано
8	Формулювання та написання реферату, вступу, висновків та переліку використаних джерел	22.04.2025	Виконано
9	Створення демонстраційних матеріалів	23.04.2025	Виконано

Здобувач вищої освіти

(підпис)

Іван УЖАНОВ

(Ім'я, ПРІЗВИЩЕ)

Керівник

кваліфікаційної роботи

(підпис)

Катерина НЕСТЕРЕНКО

(Ім'я, ПРІЗВИЩЕ)

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня бакалавра: 54 стор., 39 рис., 3 табл., 22 джерел.

Мета роботи – дослідження особливостей впровадження технологій Інтернету речей (IoT) для автоматизації побутових пристроїв та розробка функціонального макета розумного дистанційного перемикача з використанням мікроконтролера ESP та крокового двигуна.

Об'єкт дослідження – процес автоматизації існуючих приладів і техніки, що не відносяться до сфери IoT.

Предмет дослідження – способи розробки та реалізації дистанційного натискача кнопок з підтримкою бездротового управління, для розширення можливостей пристроїв без можливості віддаленого керування.

Методи дослідження – аналіз та синтез сучасного стану тенденцій розвитку інтернету речей, порівняння типів електродвигунів, мікроконтролерів ESP та варіантів механізмів; метод технічного моделювання схеми та алгоритму роботи пристрою; метод програмного моделювання програмного забезпечення пристрою.

Короткий зміст роботи: У роботі проаналізовано ринок пристроїв IoT, наведено огляд їх розповсюдженості, сфер застосування та перспектив розвитку. Визначено особливості мікроконтролера ESP8266 та обґрунтовано його вибір. Розглянуто варіанти електродвигунів, серед яких обрано модель 28BYJ-48 та драйвер ULN2003. Реалізовано апаратну частину пристрою, корпус з механізмом перетворення обертального руху на лінійний кривошипно-шатунного типу, програмне забезпечення, інтерфейс користувача у застосунку Blynk. Також реалізовано два режими роботи пристрою і можливість відкладеного спрацювання.

КЛЮЧОВІ СЛОВА: ІНТЕРНЕТ РЕЧЕЙ, ІОТ, ESP8266, АВТОМАТИЗАЦІЯ, BLYNK, КРОКОВИЙ ДВИГУН, C#, РОЗУМНИЙ ПЕРЕМИКАЧ, ДИСТАНЦІЙНЕ КЕРУВАННЯ, АВТОМАТИЗАЦІЯ, БЕЗДРотові ТЕХНОЛОГІЇ, SMART HOME, МІКРОКОНТРОЛЕР, NODEMCU, ESP.

ЗМІСТ

ВСТУП.....	8
1 АНАЛІЗ ПОПУЛЯРНОСТІ ОБРАНОЇ СФЕРИ ТА ТЕОРЕТИЧНІ ОСНОВИ РЕАЛІЗАЦІЇ ПРОЕКТУ	11
1.1 Аналіз глобального ринку інтернету речей.....	11
1.2 Реалії розповсюдженості девайсів інтернету речей	13
1.3 Сфери застосування IoT	14
1.4 Основні види електродвигунів.....	23
1.5 Мікроконтролер ESP та модуль NodeMCU версії Lolin	26
2 АПАРАТНА СКЛАДОВА ДИСТАНЦІЙНОГО ПЕРЕМИКАЧА	30
2.1 Принцип та алгоритм роботи пристрою	30
2.2 Компоненти та схема їх з'єднання	31
2.3 Механічна складова пристрою та корпус	37
3 ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДИСТАНЦІЙНОГО ПЕРЕМИКАЧА НА БАЗІ ESP	41
3.1 Налаштування середовища розробки.....	41
3.2 Під'єднання до сервісу Blynk	45
3.3 Алгоритм роботи програмної складової	52
3.4 Лістинг коду програми	53
ВИСНОВОК.....	60
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	62
ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ	65

ВСТУП

Вибір теми для кваліфікаційної роботи зумовлений високою вартістю та складністю автоматизації побутового життя для людини без наявних необхідних технічних знань, та необхідністю створення навчально-демонстраційного макету для поширення інформації про особливості виготовлення приладів для локальної автоматизації уже наявної побутової техніки.

У світі існують багато частково або повністю автоматизованих промислових виробництв, розумних будинків, медичних приладів та інші. Але не так багато девайсів можна використовувати для виконання різних задач, наразі наявні прилади на кшталт голосових помічників, взаємодіють у більшості випадків лише зі своїм сертифікованим обладнанням, тим самим ускладнюючи або повністю унеможливаючи перехід звичайних уже наявних пристроїв до автоматизації без значних перепланувань та грошових витрат.

Інтернет речей (IoT) – це концепція мережі у складі якої відбувається взаємодія взаємозв'язаних фізичних пристроїв із програмним забезпеченням, котрі дають можливість здійснювати передачі та обмін даними між фізичним світом та комп'ютерними системами у автоматичному режимі із використанням базових протоколів зв'язку. В основному вони поділяються на два типи: сенсори(давачі) та виконавці і можуть бути пов'язані між собою за допомогою дротової або бездротової мережі. Мій пристрій відноситься до виконавця із пультом віддаленого керування через застосунок Blynk.

Згідно статистичних прогнозів ринку інтернет речей від компанії The Business Research Company за 2023 рік, з кожним роком обсяг даного ринку збільшується, а разом із ним попит та зацікавленість людей у цій сфері, це у свою чергу впливає на кількість досліджень та розвиток даної сфери. Згідно даної інформації автоматизація процесів стає масовим трендом.

Інформаційною базою для написання кваліфікаційної роботи були використані публічно доступні матеріали досліджень та аналітики, статистичні данні, електронні ресурси наукова та спеціалізована література та інші джерела,

серед котрих положення про кваліфікаційні роботи здобувачів Вищої освіти у державному університеті Інформаційно-комунікаційних технологій.

Об'єктом дослідження кваліфікаційної роботи є процес автоматизації існуючих приладів, техніки та вимикачів, які не відносяться до інтернету речей (IoT) для збільшення та покращення можливостей оптимізації життя.

Предметом дослідження роботи є способи розробки та використання віддалених перемикачів, котрі можна інтегрувати на застарілі девайси .

Метою дослідження є дослідження розповсюдженості автоматизації інтернету речей та розробка концепції розумного дистанційного перемикача для поліпшення можливостей доступності дистанційного керування для застарілих приладів.

Згідно мети кваліфікаційної роботи було поставлено наступні задачі:

1. Дослідити сферу інтернету речей та зробити висновки про її розвиток сферу застосування та популярність серед користувачів.
2. Обрати необхідні компоненти та застосунки для реалізації проекту, вивчити основні поняття.
3. Ознайомитися із документацією та особливостями роботи обраних компонентів.
4. На базі теоретичної інформації розробити та реалізувати концепцію дистанційного натискача кнопок на основі мікроконтролера ESP та шагового двигуна.

Методика дослідження включає у себе два етапи: теоретичний та практичний. Відповідно до теоретичної частини виконується вивчення поставленої задачі, проводиться пошук рішень та аналіз їх ефективності та спроможності до реалізації. Після чого на основі отриманих даних виконується перехід до практичного етапу. Практичний етап відповідає за безпосередньо розробку пристрою – дистанційного приладу для натискання кнопок – його апаратної частини, корпусу та програмного забезпечення. А також налаштування інтерфейсу користувача на базі застосунку Blynk.

Наукова новизна дослідження полягає у тому що концепція даного девайсу є досить новою і не зустрічається у великому різноманітті, також для цих цілей майже не використовувалися мікроконтролери ESP з використанням шагового двигуна, задача цього дослідження полягає саме у реалізації такої системи із компонентами що описані вище .

Практична значимість результатів дослідження полягає у можливості застосування створеного пристрою в рамках дистанційного або автоматичного ввімкнення або вимкнення приладів із кнопками незалежно від середовища застосування, наприклад: кухонні плити, газові котли, стаціонарні комп'ютери, перемикачі світла і т.д.

1 АНАЛІЗ ПОПУЛЯРНОСТІ ОБРАНОЇ СФЕРИ ТА ТЕОРЕТИЧНІ ОСНОВИ РЕАЛІЗАЦІЇ ПРОЕКТУ

1.1 Аналіз глобального ринку інтернету речей

Концепція терміну IoT (інтернет речей) бере свій початок ще з 1999 року, де вона була введена британським інженером Кевіном Ештоном під час його роботи над «Procter & Gamble», задля того щоб мати можливість для опису системи у котрій фізичні об'єкти могли бути пов'язані із давачами та Інтернет мережею. Насамперед Ештон створив цей термін задля поточнення та демонстрації можливостей RFID (радіочастотної ідентифікації), яка в свою чергу застосовувалась у системах поставок, а саме у відстеженні товарів і проведенню необхідних обчислень без необхідності втручання людської робочої сили. Іншими словами кажучи, задля автоматизації необхідних процесів.

Сьогодні, інтернет речей – це всесвітньо відомий та популярний термін котрий охоплює безліч сфер застосування, таких як: розумні домівки, промисловість, розумні міста, сільське господарство, охорона довкілля, енергетика, торгівля, автомобільне будівництво, освіта, транспорт та логістика, туризм, медицина, охорона праці та безпека а також розважальна індустрія.

Відповідно до загального ринкового дослідження британської компанії «The Business Research Company» за січень 2025 року серед таких країн як Австралія, Бразилія, Китай, Німеччина, Індія, Франція, США, Канада, Іспанія, Індонезія, Японія, Італія, росія, Північна Корея та Великобританія, а також додатково враховано такі країни як Сінгапур, Тайланд, Філіппіни, Бангладеш, В'єтнам, Малайзія, Гонконг, Новозеландія, Чилі, Аргентина, Колумбія, Фінляндія, Швеція, Швейцарія, Австрія, Бельгія, Ірландія, Туреччина, Єгипет, Польща, Україна, Ізраїль, Румунія, Нігерія та інші. Розміри ринку інтернету речей із кожним роком зростають приблизно на 17,5% за рік. У 2024 році загальна вартість ринку становила \$535,97 мільярдів, наразі ж вона зросла до \$629,5 мільярда, це свідчить про ріст популярності та ефективності використання даного напрямку у світі

IoT Global Market Report 2025

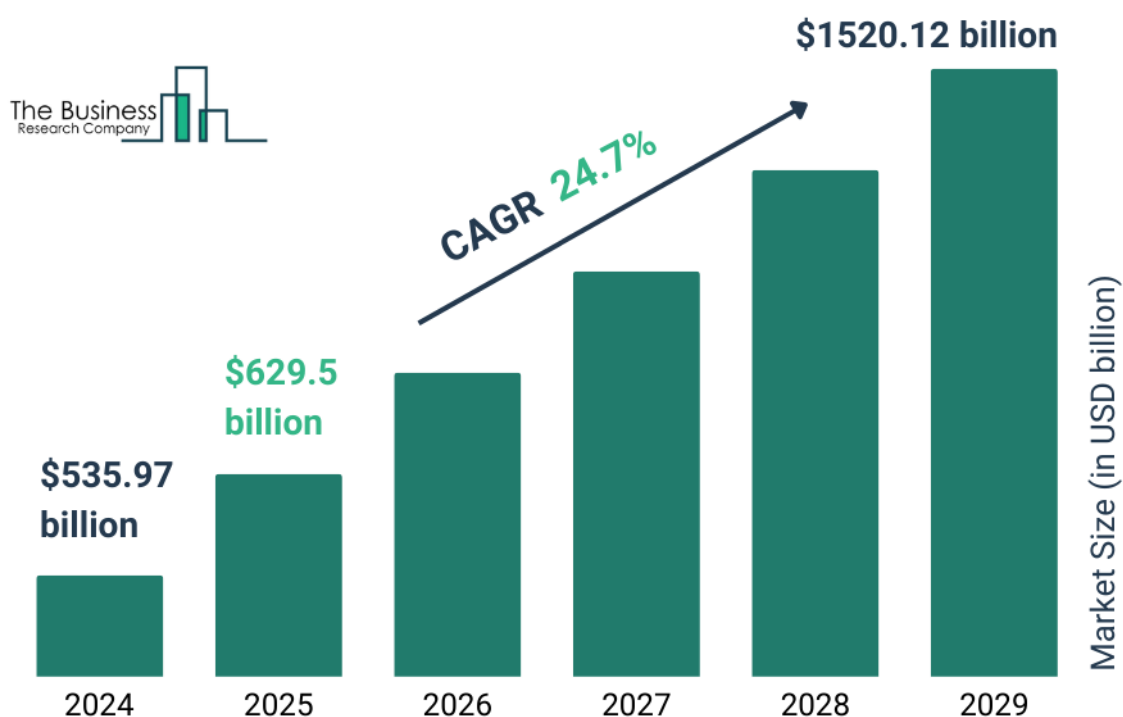


Рис. 1.1 Припущення тенденції розвитку ринку IoT

Відповідно до виявлених тенденцій компанія припускає що до 2029 року ринок інтернету речей зможе сягнути неймовірних \$1520,12 мільярдів при загально річному темпі зростання у 24,7 %. Їхні обрахунки базуються на набираючих оберти світових тенденціях та технологіях таких як розширення варіантів девайсів та технологій інтернету речей, розгортанням та використанням нового варіанту мережі 5G, занепокоєністю людей у екологічності та енергоефективності, а також довговічності наявних технологій. Також важливий внесок у цьому відіграє достатньо нова та прогресивна технологія AI(штучний інтелект), котрий виводить можливості автоматизації процесів на новий рівень, оскільки після проведення певного періоду навчання, штучний інтелект набуває здатності самостійно приймати рішення на основі опрацьованої бази даних, та на підставі набутих навичок та знань здатен виконувати подальші задачі та адаптуватися до нових виникаючих складностей.

1.2 Реалії розповсюдженості девайсів інтернету речей

Оскільки з кожним роком ринок індустрії інтернету речей, давайте проаналізуємо загально статистичні дані щодо розповсюдженості у світі приладів котрі притримуються даної концепції, відповідно до досліджень та прогнозів компанії «Transforma Insights» за 2023 рік, а також статистичних даних «Exploding topics» та «Statista» Статистична добірка припущень розрахована на період з 2023 року по 2025 рік.

Насамперед, відповідно до наявних за 2023 рік даних у світі існує приблизно 15,01 мільярд фізичних приладів інтернету речей. З них ~ 5,04 мільярда знаходяться на території Китаю, це становить майже 33,6% від усіх наявних девайсів світу, на другому місці знаходиться Європа із ~ 3,48 мільярдами (23,1%), третє місце посідає Північна Америка з рахунком ~ 3,22 мільярда (21,5%).

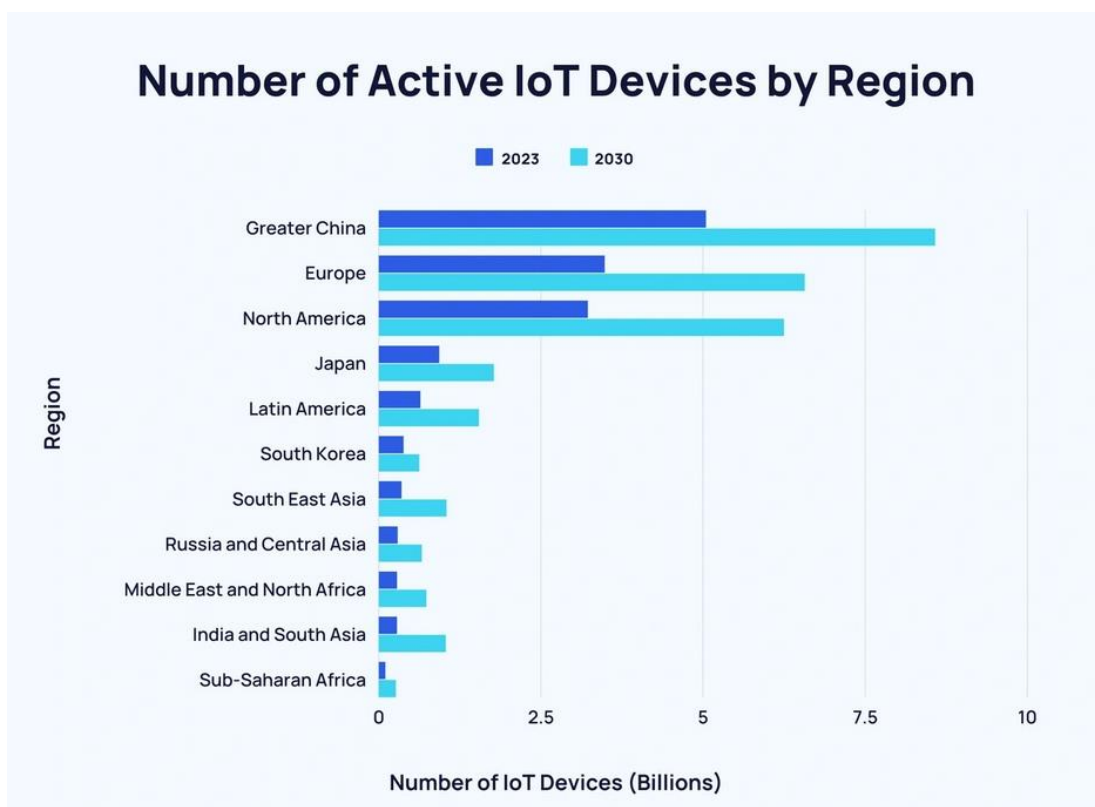


Рис. 1.2 Порівняння кількості IoT девайсів та прогноз на 2030 рік

Подальші країни, які не займають великий відсоток від загальної кількості, але розглядалися також: Японія ~ 930 мільйонів (6,2%), Латинська Америка ~ 640 мільйонів (4,3%), Південна Корея ~ 380 мільйонів (2,5%), Південно-східна Азія ~ 350 мільйонів (2,3%), росія та Центральна Азія ~ 290 мільйонів (1,9%), Центрально-

східна і Північна Африка ~ 287 мільйонів (1,9%), Індія та Південна Азія ~ 280 мільйонів (1,9%), Південна Африка ~ 100 мільйонів (0,7%).

Ми дізналися скільки приблизно на нашій планеті вже знаходиться пристроїв що відповідають концепції IoT, відповідно до припущень на 2025 рік, загальна кількість зросла приблизно до 20,1 мільярда одиниць, це у котрий раз підтверджує світову зацікавленість у покращені та автоматизації процесів. Згідно цих же статистичних даних на 2025 рік у світі існує приблизно 10,3 мільярди девайсів, які не відносяться до інтернету речей це складає приблизно 33,6%(Загалом 30,4 мільярдів) .

1.3 Сфери застосування IoT

Наразі IoT використовується майже у всіх сферах нашого життя, таких як:

Сільське господарство, одна із найбільш важливих сфер нашого життя. Скотарі сьогодення використовують саме такі рішення як розумні бирки для худоби, що відслідковують стан здоров'я, місцеположення за допомогою технології GPS, а також зберігають у собі інформацію про дату народження, підвид та ім'я тварини, уся ця інформація доступна через спеціальні застосунки для скотарів. Одна із основоположних компаній що розпочала виробництво подібних приладів це «HerdDogg» Також увагою не обділені фермери, для тепличного варіанту використовуються автоматичні гідропоніки, що у свою чергу крім того що скорочують розміри необхідного для росту ґрунту, так ще й замірюють параметри вдобреності землі, а також степінь поливу в наслідку чого відпадає необхідність ручного поливу, оскільки майже всі процеси відбуваються автоматично. Ще є варіанти у яких навіть не використовується земля, усі необхідні добрива одразу замішуються до розчину котрим поливають культури, але мінусом такого способу є втрата смаку вирощених таким чином продуктів. Поля теж мають певні інновації, починаючи із автоматичного поливу або сповіщення про його необхідність і закінчуючи автоматичними комбайнами, котрі можуть збирати врожай без необхідності у людській силі, звісно це не розповсюджується на всі відомі нам овочі та фрукти. Гарним прикладом можна вважати компанію «AGCO» Але не дивлячись на усі переваги доля від загального обсягу складає приблизно 3-4%, або

ж ~1,5 – 1,7 мільярда пристроїв. Лідером серед виробників та споживачів у кількості приладів цієї сфери вважається Китай.

Медицина теж зазнала втручання IoT. У стаціонарних клініках зараз можна зустріти базові датчики наповненості палат по кількості пацієнтів, ця система працює за принципом чи є якась вага на ліжку, усі данні посилаються на реєстрацію, де медсестри співвідносять ці дані із даними потерпілих, це допомагає відслідковувати статистику наповненості тої чи іншої палати або відділення, та порахувати загальну кількість постраждалих. У цій системі також є можливість переглянути лікаря за яким закріплено пацієнта, посилення на історію хвороби та точний час початку лікування, також там згодом з'являються орієнтовні терміни завершення реабілітації. Ще існують досить старі, але все ще ефективні датчики рівню цукру в крові, зустрічаються варіанти які не лише відправляють необхідну інформацію у відповідний додаток, а й такі що автоматично вводять необхідну кількість інсуліну згідно заданих параметрів тіла користувача. Аналогічно до датчиків рівня цукру існують мікродатчики для відстеження ризиків виникнення інфарктів та інсультів, переважно більшу кількість із них виготовляє компанія «Endotronix's Cordella». Взагалі існує велика кількість різноманітних датчиків що тим чи іншим чином відстежують поточний стан пацієнтів та інформують них або медичний персонал без необхідності втручання їх носія. Також подібні девайси зустрічаються у лікарнях спрямованих на огляд психічнохворих людей, в більшості випадків вони допомагають відстежити поточне місцезоположення пацієнта, це необхідно для того щоб зменшити шанси на втечу для особливо небезпечних із них. Відносно загального обсягу частини сфери охорони здоров'я складає приблизно 8-10%(2-2,4 мільярда пристроїв). Найбільше їх у США та країнах Європи

Також інтернет речей починає набирати популярність у роздрібній торгівлі та складських цілях. Наприклад уже існують такі ресурси як «Engage3», котрі збирають статистичні данні згідно конкретних тематик за конкретним регіоном у світі, тим самими дозволяючи продавцям виставляти оптимальні ціни на власну продукцію для продажу у межах своєї країни або закордон, вони працюють за принципом машинного навчання на базі вебскрейпінга та точних даних про

продажі в конкретному регіоні. За подібною технологією існують безліч ресурсів котрі тим чи іншим чином збирають статистичні данні та передають їх для підприємців а керівників У складських приміщеннях наразі все частіше і частіше можна зустріти розумні стелажі, котрі завдяки набору датчиків та свого програмного забезпечення пришвидшують роботу персоналу за рахунок можливості перевірки конкретної позиції на наявність або відсутність, та ознайомитися із написаними заздалегідь даними про той чи інший товар. Ця технологія зустрічається не лише у складах для товарів неїстівного характеру, ресторани та кафе також використовують схожі технології для відстеження строків зберігання рідких інгредієнтів та вчасному вживанні їх. Також важливо зазначити що вже існують повністю розумні магазини, які використовують технологію «Zippin's Checkout-free» і за допомогою камер та екосистеми сенсорів фіксують, який конкретний товар бере той чи інший покупець і вносить його до чеку, таким чином знімаючи необхідність у застосуванні касирів та консультантів. Важливо зазначити що клієнт має бути зареєстрований у застосунку цього магазину, де він має завантажити зображення свого обличчя та персональні дані своїх розрахункових реквізитів. Ця сфера також тільки починає набирати обороти, загалом проблеми у ній зумовлені відхиленням менталітету в тих чи інших країнах, багато людей не можуть перебороти своє бажання безкоштовної наживи, власне через це дана тенденція має дуже невелику частку розповсюдженості, всього приблизно 5% від усіх IoT пристроїв або ж приблизно 2 мільярди одиниць.

Транспорт та вантажні перевезення - одні із перших сфер, що застали модифікацію за рахунок технологій інтернету речей. Напевно найкращім прикладом застосування можна вважати перевозки великого обсягу вантажів на великі відстані, далекобійники окрім заданого місця у яке необхідно доставити вантаж ще мають цілу систему зв'язку, як із подібними до себе водіями, задля того щоб точніше передбачати свій маршрут та поточний стан дороги, так і систему відслідковування місцеположення, вона у свою чергу необхідна для контролю доставки тих чи інших вантажів, а також для збору статистичних даних про завантаженість руху на певних ділянках роботи. Основним плюсом цієї системи є

можливість отримувати наближений до точного час виконання поїздки. Технологічні інновації і не оминули ринок автомобілів для персонального користування, компанія «Waymo» вважається першою в світі, оскільки вони змогли реалізувати та запустити перші безпілотні автомобілі таксі у кінці 2018 року. Робота над ними велася не один рік, оскільки система працює на основі нейронної мережі, що аналізує зображення дороги, а також своє місце положення у світі, в наслідку чого намагається слідувати правилам дорожнього руху країни в якій знаходиться. Не дивлячись на значний вік цієї технології, вона все ще не вважається завершеною. Ну і не можна не згадати допоміжні та вбудовані девайси та сенсори, які взаємодіють зі своїми застосунками та збирають інформацію про стан автомобільного транспорту, а також додають ряд нових фіч, таких як система анти уgonу та інше. Загалом наразі існує приблизно 3 мільярди пристроїв цієї сфери у світі, це складає приблизно 12-13%. Данні технології приблизно однаково розповсюджені по світу за виключенням країн Африки та деяких острівних країн(крім Японії).

Застосування IoT у промисловій індустрії, власне і є основою та початком автоматизації інших сфер, оскільки більшість наразі відомих нам приладів виробляються на автоматичній конвеєрній лінії. Компанія «Amper» займається виготовленням моніторингових систем, котрі використовують машинне навчання, наукову інформацію та датчики інтернету речей для створення та покращення виробничого процесу. В основному датчики фіксують цілу низку факторів, таких як: необхідний рівень електричного напруження для роботи системи, час простою або технічних робіт, на основі цих даних власники заводів і керівники можуть створювати плани графіків, зменшувати затрати і виділяти найбільш ефективні сегменти фабрики. Також існують окремі компанії на кшталт «StrongArm Technologies», вони створюють девайси, які зв'язуються із інтернетом речей и сприяють підвищенню безпеки для робітників. Вони збирають інформацію про час роботи, метрики пересування на базі датчиків та камер відеоспостереження, задля формування оцінки безпеки. Таким чином вони можуть виявити слабкі та ненадійні місця і проінформувати керівництво задля їх переналаштування. Загалом подібних

до описаних вище систем та приладів у світі на сьогоднішній налічується приблизно 4-5 мільярдів пристроїв (22-23% від загальної кількості) і ця цифра росте з кожним днем. Ці технології розповсюджені майже у всіх наразі існуючих країнах.

Розумні будинки – це наймасштабніша сфера застосування пристроїв інтернету речей із титанічними ~ 5,5-6 мільярдами одиниць і ця цифра зростає з кожним днем (приблизно 25-27% від загальної кількості). Прилади інтернету речей щільно вросли у наше повсякденне життя, вони в більшості випадків є автоматизованими аналогами побутових пристроїв або надають поточну інформацію відповідно до окремих необхідних на поточний момент параметрів. Напевно одними із самих розповсюджених приладів є розумні термостати, оскільки вони непогано інтегруються з уже існуючими пристроями, а також за замовченням є в наявності у всіх розумних будинках. За допомогою датчиків вони роблять заміри температури в необхідних кімнатах та підганяють її до встановленої у разі якщо та відрізняється, також розумний термостат дозволяє більш економно використовувати електроенергію, оскільки працює цілодобово і може вимикати нагрівачі прилади або охолоджуючі прилади. Лідерами серед термостатів прийнято вважати «Nest Learning Thermostat», які були розроблені компанією «Google». Але лідерство серед розповсюдженості та доступності за правом належить розумному освітленню. Воно стало доступне звичайним користувачам найперше, наразі його можна зустріти будь-де: на рекламних біл-бордах, у складських приміщеннях, в садах і домах людей і тд. Найпростіші з їх представників просто вмикаються коли датчики світла не фіксують необхідного рівня освітленості, такі варіації досить часто мають вбудовані міні сонячні панелі, що подовжує строк їх роботи, а також оптимізує процес зарядки ліхтарика. Далі йде розумне освітлення що може керуватися через застосунки із телефону або персонального комп'ютера, вони використовують технологію Wi-Fi або Bluetooth, щоб мати змогу приймати сигнали від керуючого приладу, але відповідно до отриманого контролю потребують кращих джерел живлення таких як акумулятори, батарейки або ж безпосереднє живлення від розетки. Залежно від приладу та застосунка можна створювати власні правила на ввімкнення та вимкнення відносно

часу або дня тижня. Наприклад лампи «Hue» від компанії «Philips». Також важливо згадати розумні замки, котрі забезпечують безпечний доступ до обраного приміщення чи елементу меблів типу шухляди. Більшість із них не потребують фізичного ключа і відкриваються відносно таких способів: безпосередньо від наближення авторизованого користувача через сканування обличчя або говоріння конкретної фрази, через мобільний застосунок, через введення пін-коду на відповідній панелі, за допомогою біометричних даних (відбиток пальця). Також до цієї категорії можна віднести замки що використовують технологію RFID, де за допомогою однієї картки можна налаштувати відкриття конкретних дверей для кожної з них, таку технологію використовують у різних наукових лабораторіях, або промислових зонах, оскільки це допомагає запобігти витоку даних. На власному досвіді я стикався лише із замками від компанії «Yale Approach Lock». Також наразі досить популярними є роботи-прибиральники, а саме роботи-пилососи, оскільки вони майже повністю знімають необхідність прибирання підлоги у власному будинку або в квартирі. Ці пристрої знатні проводити як сухе так і вологе прибирання без втручання людини, але параметри якості, швидкості та ефективності роботи напряду залежать від виробника та вартості, оскільки основними параметрами є місткість баку для сміття, швидкість роботи та датчики розпізнавання маршруту. Логіка роботи досить інтуїтивний, при першому запуску пристрій починає шукати мобільний додаток користувача, після чого починає досліджувати обсяг роботи (будувати карту доступної території), та будувати найбільш ефективний маршрут для пересування. Через застосунок можна обмежувати зони роботи, або ж вибирати окремі кімнати, також можна налаштувати графік роботи, питання щодо підзарядки наразі не виникає оскільки всі наявні варіанти з заводу мають програму яка запам'ятовує теперішнє місце відносно створеного маршруту та у разі розрядки акумулятора повертає робота у хаб для дозарядки, розрахунки цього явища відбуваються на постійній основі, через це ситуація у якій прилад не доїде до зарядної станції зводиться до нуля. Найбільш доступними серед усіх виробників є роботи-пилососи від компанії «Xiaomi», а саме моделі типу «Roborock».

Але все ж таки найголовнішою сферою застосування приладів є розумні системи безпеки, тому що більшість приладів в разі виникнення специфічних ситуацій можна досить просто перепрошити на вимоги нового користувача, саме тому це одна із перших і найдорожчих систем для встановлення. Більшість із них нам добре відомі оскільки зустрічаються переважно у всіх публічних місцях. Камери відеоспостереження, досить прості у розміщенні та керуванні, зустрічаються як сталі варіації із можливістю запису відео для подальшого пересилання інформації у хмарне середовище так і більш складні, з детектором руху, вони зберігають лише фрагменти відео у моменти розпізнавання пересування. Взагалі камери із датчиками руху можна виділити у окрему категорію, тому що вони також можуть використовуватися для контролю доступу до певних територій, де за допомогою нейронної мережі проводиться збір даних із фотографій обличчя конкретного персоналу, ця технологія може співпрацювати із деякими варіантами розумних замків. У разі виникнення будь-якого порушення, згідно встановлених правил при розсортуванні системи безпеки, спрацьовує сигналізація, вони теж поділяються на кілька видів, але найбільш поширеним із них є беззвучний із автоматичним викликом служби безпеки, або поліції. Компанія «ADT», досить популярна у цій сфері оскільки допомагає у проектуванні систем безпеки переважно для публічних закладів, а для сфери розумних будинків більш підходить компанія «SimpliSafe» завдяки більш автоматизованому та бездротовому підходу до створення системи.

Мною було описано основні сфери застосування IoT, а також базові приклади технологій і компаній що до них відносяться, але це лише пристрої давачі та приймачі, бо задля досягнення автоматизації, крім них ще необхідно мати середу у якій вони могли би взаємодіяти один з одним, обробляти результати. Для цього у більшості сфер використовуються технології хмарного обчислення. Вони дозволяють використовувати уже готові рішення або у разі бажання мають гнучкі можливості для автоматизації незвичних процесів, більш детально дана тема описана у моїх тезах за другу всеукраїнську науково-технічну конференцію(18 листопада 2024 року). Для кращого розуміння доступних варіантів я звернусь до

своїх тез, де мною було розібрано три найбільш відомих і популярних середовища хмарних послуг, а також наведено приклади застосування:

« Amazon Web Services (AWS) пропонує рішення для промислового IoT, такі як AWS IoT Greengrass і AWS IoT Core, які інтегрують хмарну інфраструктуру з локальними обчисленнями на виробничих підприємствах. Вони дозволяють збирати, обробляти та аналізувати дані з підключених пристроїв у реальному часі, покращуючи продуктивність і моніторинг обладнання. AWS Lambda допомагає автоматизувати обробку подій та створювати безсерверні додатки для керування.

Microsoft Azure для цифрової трансформації Azure IoT Hub та Azure Machine Learning є ключовими інструментами для цифровізації виробництва. Azure IoT Hub дозволяє інтегрувати пристрої IoT з хмарною інфраструктурою, забезпечуючи моніторинг і віддалене керування в реальному часі. Azure Digital Twins допомагає створювати цифрові моделі виробничих процесів, що дозволяє проводити аналіз та оптимізацію на основі симуляцій. Це значно знижує кількість помилок, виявляючи потенційні проблеми які можуть виникнути.

Google Cloud для виробництва Google Cloud IoT Core надає можливості для збору, обробки і візуалізації даних з IoT-пристроїв у реальному часі, дозволяючи підприємствам отримувати аналітику для прийняття рішень. Використовуючи Google Cloud BigQuery, виробничі компанії можуть обробляти великі набори даних і отримувати аналітичні висновки швидко і без необхідності налаштування складної інфраструктури.

Попри очевидні переваги, хмарні обчислення стикаються з певними викликами у виробничій сфері. Одним з основних є безпека даних, оскільки переміщення виробничих даних до хмари потребує додаткових заходів для захисту конфіденційної інформації, особливо в галузях з високими вимогами до безпеки. Також важливою проблемою є затримки в передачі даних: віддалений доступ до хмарних сервісів може викликати затримки в обробці критично важливої інформації, що є неприйнятним для багатьох виробничих процесів у реальному часі. Ще одним викликом є інтеграція з існуючими системами, оскільки багато

підприємств стикаються з труднощами при впровадженні хмарних рішень у свої застарілі ІТ-системи.».

Після сфер описаних вище нарешті можна перейти до апофеозу сфер застосування пристроїв інтернету речей, а саме до концепції розумних міст. Розумне місто відноситься до урбаністичної концепції у якій інформаційні та комунікаційні технології містяться майже у всіх сферах життя та слугують задля підвищення якості життя для жителів, збільшення ефективності міських послуг та інфраструктури та для більш стабільного розвитку в цілому. Основні характеристики, що дають можливості звичайному місту почати вважатися розумним це впровадження одного, кількох або у ідеальному випадку усіх наступних критеріїв. З інфраструктури: використання IoT для моніторингу дорожнього та громадського трафіку з можливістю коригування напрямів за допомогою AI, розумне вуличне освітлення, автоматизована система вивозу, сортування та переробки сміття, система оптимізації енергоспоживання, система автоматичного прибирання вулиць та паркових зон, розумні переходи і тд. З безпеки та охорони здоров'я: використання камер відеоспостереження для перевірки та аналітики жителів задля запобігання злочинності, використання тої ж системи для ідентифікації стану громадян та виклику швидкої у разі необхідності, система забезпечуюча доступність усіх основних методів лікування та діагностики і тп. З транспорту: гнучкі інтелектуальні системи керування рухом транспорту, безпілотний громадський транспорт, розвинена система різновидів громадського транспорту, розумні переходи і тд. З екології: моніторинг якості та ступеня забруднення повітря, автоматичні водоочисні споруди, система ефективного управління ресурсами і тд. Наразі не існує однозначно розумного міста, але вже існують кілька міст що намагаються наблизитись до цієї урбаністичної концепції. Напевно найкращім прикладом реалізації даної ідеї є Сінгапур, станом на 2023 рік з моменту запуску ініціативи «Smart nation»(у 2014 році) у місті реалізована система безконтактних платежів на усі види громадського транспорту, також там ввели цифрову систему для людей пенсійного віку, завдяки якій пенсіонери можуть отримати консультацію лікаря через онлайн дзвінок, також влада Сінгапуру робила

заяву про створення нового житлового району, котрий буде вбудовано у 700 гектарний ліс із автоматизованими симбіотичними системами контролю якості повітря, рівня фізичного та світлового забруднення та спорудами для фільтрації води.

1.4 Основні види електродвигунів

Електродвигуни – це механізми що перетворюють електроенергію на механічний рух шляхом обертаючогося магнітного поля. Процес відбувається при пропусканні електричного струму через певну кількість котушок(в більшості випадків зроблених із мідного дроту) тим самим створюючи електромагнітне поле, котре у свою чергу взаємодіє зі сталими магнітами двигуна. Результатом цієї взаємодії є обертальний рух ротора із певної швидкістю, ці параметри можна коригувати за рахунок змінення струму джерела живлення або напруги. При використанні у сфері робототехніки для взаємодії із різними платами електродвигуни у більшості випадків вимагають окремий драйвер для коректного налаштування і правильності їх роботи.

Приблизна частка розповсюдженості використання електродвигунів у світі



Рис. 1.3 Приблизна частка розповсюдженості електродвигунів

Загалом електродвигуни поділяють на 4 типи, за принципом роботи, а саме:

Електродвигуни змінного струму – це двигуни, що працюють за рахунок створення магнітного поля, яке обертається в стартері та індукує струм у роторі

тим самим викликаючи оберти. Завдяки своїй ефективності, надійності та простоти експлуатації вони застосовуються в низці галузей починаючи з промислового виробництва й завершуючи побутовою технікою. Відповідно до назви, живлення двигуна виконується за рахунок змінного струму (постійно змінює напрям руху) за рахунок чого швидкість обертів залежить від самої частоти змін струму. У свою чергу ці двигуни поділяються на два типи: синхронний та асинхронний двигуни. Перший працює лише з фіксованою швидкістю обертів та застосовується переважно для видобутку електроенергії та у системах точного керування. Другий – працює на меншій швидкості, але зустрічається у різних варіаціях, а саме однофазні та трифазні електродвигуни, їхня різниця переважно полягає у кількості вольт необхідній їм для роботи (220 та 380-660), це впливає на те у якій конкретно галузі вони будуть використані. Загалом двигуни змінного струму є одними із найбільш популярних та розповсюджених варіацій у світі.

Електродвигуни постійного струму – це двигуни, які працюють за схожим принципом до типу на змінному струмі, вони мають магнітне поле, котре взаємодіє зі струмом для утворення обертового моменту, котрий у свою чергу використовується для приведення у дію певних механічних приладів, наприклад конвеєрів або насосів. На відміну від перемінних, швидкість постійних можна контролювати шляхом зміни напруги або струму, що живить двигун. Саме цей фактор і робить цей тип адаптуємим для різних механізмів та девайсів. Переважно такі двигуни знаходять своє застосування у комерційних та промислових сферах. Ці двигуни теж поділяються на кілька типів: щіткові та безщіткові. Щітками називають контакти між комутатором та землею, вони виготовляються з вугілля і слугують перемикачами полярностей електричного струму в обмотках двигуна, що власне і приводить ротор у обертання. Безщіткові ж використовують електронну комутацію для тих же цілей. Основна відмінність між ними це по-перше принцип роботи, по-друге щіткові банально простіші у керуванні та налаштуванні, а безщіткові довговічніші. Саме тому обидва цих варіанти до сих пір зустрічаються у більшості знайомих для нас приладах, таких як : 3D принтери, безпілотники, принтери, кондиціонери і тп.

Серводвигуни –двигуни що працюють на змінному типі струму , також в них знаходиться механізм зворотного зв'язку, який у свою чергу дозволяє налаштовувати швидкість обороту та виставлятися на певне положення від заводського нуля. Ці можливості стали доступні завдяки сервоприводам, завдяки ряду датчиків котрі отримують необхідні параметри завдяки енкодерам або потенціометрам, він збирає дані та керує мотором. Це досить новий тип, саме тому і кількість сфер його застосування ще тільки набирає оберти у прикладах застосування, наразі він знаходиться в основі робототехніки, верстатів з ЧПУ , точних 3D принтерах та інших різних автоматизованих виробничих системах. Як і попередні варіанти вони поділяються на типи: лінійні – серводвигуни що здатні здійснювати поступальні рухи; поворотні – виконують обертання навколо осі, у більшості випадків обмежені кутом повороту у 180 градусів, але зустрічаються і варіанти із повним обсягом обертання (360 градусів); лінійно-поворотні – самий рідкісний тип, зустрічається переважно у точних виробках, поєднує у собі можливості двох попередніх типів та виконує обертання або переміщення у двох просторах; також ще виділяють щіткові та безщіткові сервоприводи, які окрім своїх нововведень ще мають особливості, котрі були описані мною вище.

Крокові двигуни - це безщітковий електродвигун працюючий від постійного струму, який поділяє повний оберт (360 градусів) на кроки, не блокуючи можливість робити повні оберти у всі сторони та точні на конкретну кількість кроків. Ще одною перевагою є їхня можливість підтримувати конкретне положення ротору без споживання електроенергії, тип самим зменшуючи температуру виділяему при використанні та збільшуючи строк їхньої служби . Вони поділяються на дві категорії із своїми варіаціями. До першої категорії входять біполярні та уніполярні крокові двигуни. Де біполярні вимагають складнішої схеми управління, але мають вищий обертальний момент і більшу точність обертуту, а другі (уніполярні) навпаки простіші у користуванні та мають відповідно нижчі характеристики. Друга ж категорія поділяється на : однокрокові – «базові», найпростіші у керуванні та найрозповсюдженіші; двокрокові – покращена версія однокрокових із більшою точністю та контрольованістю роботи; мікрокрокові – апогей точності та контролю

серед крокових двигунів, але саме тому вони є найважчими для освоєння та використання. Сфери їх застосування майже на 100% збігається із сервоприводами, але вони все ще відрізняються як принципом роботи так і можливостями у плані налаштування.

1.5 Мікроконтролер ESP та модуль NodeMCU версії Lolin

ESP – лінійка мікроконтролерів китайської компанії «Espressif Systems» з Wi-Fi інтерфейсом, характерною особливістю якої є у переважній більшості робота з системами на 32 біта та мережами з частотою 2,4 гігагерц. До неї входять такі серії мікроконтролерів ESP як : «ESP32-S3», «ESP32-S2», «ESP32-C6», «ESP32-C61», «ESP32-C5», «ESP32-C3», «ESP32-C2», «ESP32-H2», «ESP32» та «ESP8266». Кожна серія має свій ряд переваг та недоліків, детальніше таблиця 1. А саме, починаючи з «ESP8266», з'явившогося ще у 2014 році, котрий своєю ціною та ефективністю сколихнув ринок розумних товарів, важливо зазначити що серед подальших моделей він є еталоном простоти та доступності. Після дебюту 8266 компанія випустила оновлену модель «ESP32» з покращеними характеристиками, включаючи двоядерний процесор, підтримку технології Bluetooth 4.2 та розширену периферію. Наступні моделі в основному були спрямовані на розширення функціоналу та покращення рівня безпеки пристрою. Можна виділити моделі серії «ESP32-C3», оскільки вони отримали можливість сумісної роботи з «ESP8266». Починаючи з «ESP32-C5»(включно) наступні серії отримали дводіапазонний Wi-Fi 6 MCU, а також заручилися підтримкою стандарту IEEE 802.15.4(Zigbee 3.0 ; Thread 1.3).Подальші новіші серії по типу «ESP32-C6», «ESP32-C61» та «ESP32-H2» були спрямовані на покращення безпеки мікроконтролера та зменшенні енергоспоживання , а також підтримці протоколу прикладного рівня Matter.

Більш детальна інформація щодо відмінностей між мікроконтролерами різних серій згідно наступних параметрів: архітектура, частота роботи, пам'ять (SRAM), частоту Wi-Fi, Bluetooth та GPIO, наведено у таблиці 1.1. Важливо зазначити що були опущені найновіші серії оскільки наразі вони існують лише у

вигляді технічної документації та переліку можливостей, без наявного фізичного прототипа або експериментального екземпляра.

Порівняння серій контролерів ESP

Таблиця 1.1

Серія	Архітектура	Частота роботи	Пам'ять (SRAM)	Частота Wi-Fi	Bluetooth	GPIO
ESP8266	Tensilica Xtensa LX106	80-160 МГц	64 КБ	2,4 ГГц	Немає	16
ESP32	2x Xtensa LX7	До 240 МГц	520 КБ	2,4 ГГц	4.2 (BR/EDR + LE)	до 34
ESP32-S2	1x Xtensa LX7	До 240 МГц	320 КБ	2,4 ГГц	Немає	до 43
ESP32-S3	2x Xtensa LX7	До 240 МГц	512 КБ	2,4 ГГц	5.0(LE)	до 45
ESP32-C3	1x RISC-V (32 bit)	До 160 МГц	400 КБ	2,4 ГГц	5.0(LE)	до 22
ESP32-C6	1x RISC-V (32 bit)	До 160 МГц	400 КБ	2,4 ГГц	5.0(LE)	до 22
ESP32-C2	1x RISC-V (32 bit)	До 160 МГц	272 КБ	2,4 ГГц	5.0(LE)	до 20
ESP32-C5	1x RISC-V (32 bit)	До 240 МГц	400 КБ	2,4 ГГц 5 ГГц	5.0(LE)	до 22
ESP32-H2	1x RISC-V (32 bit)	До 96 МГц	256 КБ	Немає	5.3(LE)	до 26

Для створення цього пристрою я буду використовувати плату NodeMCU версії Lolin, на базі мікроконтролеру «ESP8266», оскільки її потужностей достатньо для виконання цих завдань, важливо зазначити що ця плата

використовує особливий драйвер “CH340G” для з’єднання USB з портом Serial та живиться від 4.5 V- 10 V, при поданні більшої напруги(оскільки заявлено, що плата може працювати при 15 V) необхідно використовувати зовнішнє охолодження. Важливо зазначити розпіновку цього модулю для подальшого розуміння етапу збірки девайсу та його подальшого налаштування. Схема модуля Lolin NodeMCU на базі «ESP8266» зображена на рисунку 1.4

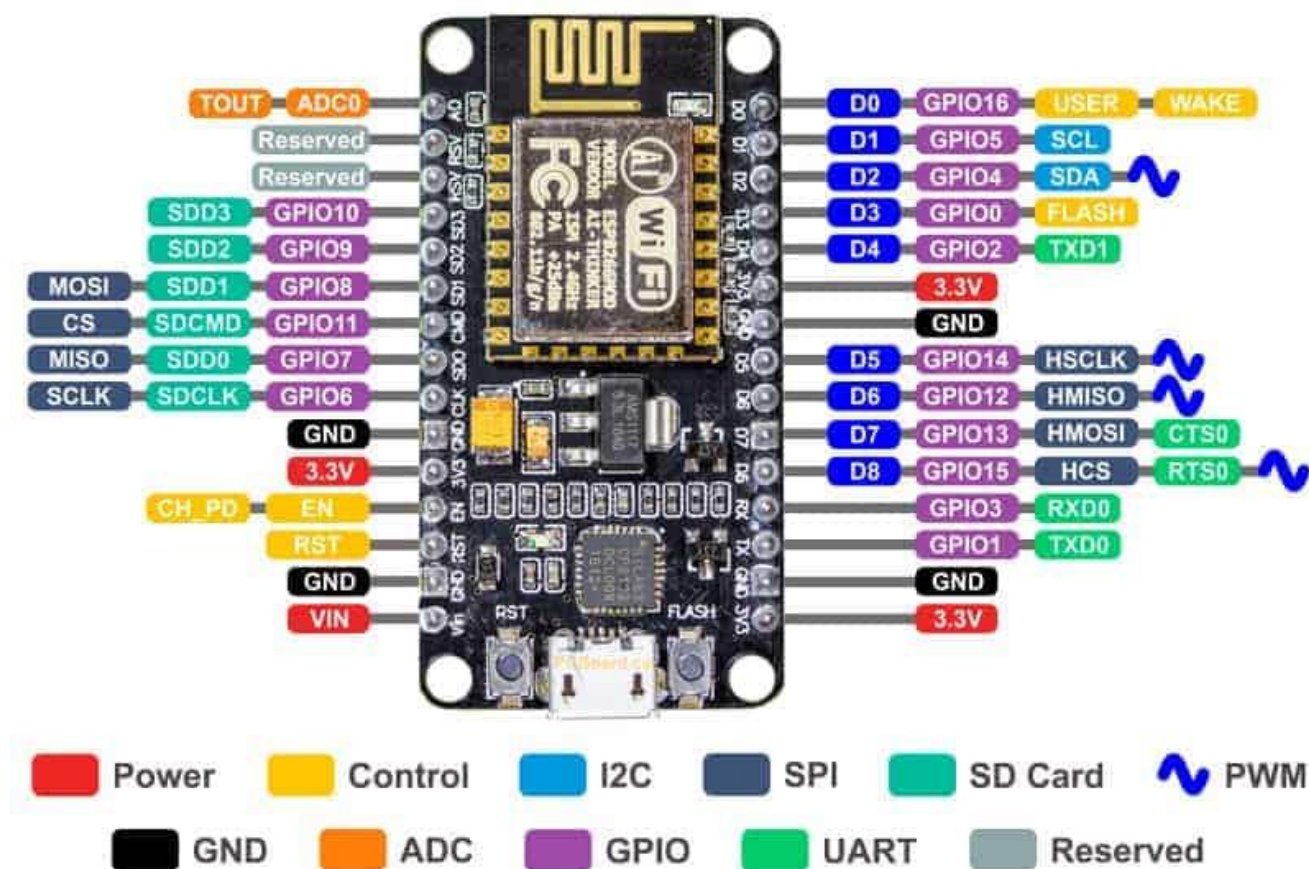


Рис. 1.4 Розпіновка та пояснення функцій

Живлення (Power) :

Vin – пін для підключення джерела живлення у діапазоні від 4,5 V до 10 V (15 V*), саме завдяки стабілізатору AMS1117-3,3(при використанні живлення від 10 V до 15 V необхідно додаткове охолодження).

3.3 V – контакти, на які подається вихідна напруга в 3.3 V, може бути використана для живлення датчиків і тп.

Земля(GND від англ. Ground) – заземлення, використовуються для замикання електричного кола та спільної опорної напруги між підключеними пристроями

Піни (GPIO - General Purpose Input/Output) – плата має 17 універсальних цифрових входів/виходів, що можуть використовуватися для наступних функцій I2C, I2S, UART, PWM, (IR) інфрачервоний пульт, керування світлодіодами або кнопками.

I2C піни («SCL», «SDA») – призначені для датчиків та периферійних пристроїв з підтримкою I2C (Inter-Integrated Circuit - міжінтегральна шина). NodeMCU з ESP8266 підтримує обидва режими роботи, як ведучий так і підлеглий, дана функція реалізується програмно, з максимальною частотою сигналу 100 КГц.

ADC канал – на платі є 10-бітний SAR ADC (Successive Approximation Register Analog-to-Digital Converter – аналого-цифровий перетворювач) що може використовуватися для заміри напруги на вході «TOUT» або для напруги живлення із виводу «VDD3P3».

UART піни – на платі знаходиться два універсальні асинхронні приймачі-передавачі (UART – Universal Asynchronous Receiver-Transmitter) що працюють на форматах RS232 та RS485, зі швидкістю до 4,5 Мб/сек. А саме: UART0 («TXD0», «RXD0», «RTS0», «CTS0»), котрий використовується для повноцінного обміну даними та UART1 («TXD1»), який лише передає дані.

SPI піни – (SPI- Serial Peripheral Interface) також є два інтерфейси: SPI («MOSI», «CS», «MISO», «SCLK») та HSPI («HCLK», «HMISO», «HMOSI», «HCS»), котрі мають 4 режими обміну даними, підтримують частоти до 80 МГц, мають буфер обсягом до 64 байтів та підтримують обидва режими роботи.

SD піни («SDD3», «SDD2», «SDD1», «SDCMD», «SDD0», «SDCLK») – дозволяють пряму приєднати SD-карти за підтримки інтерфейсу SDIO (Secure Digital Input/Output), а саме версій 1,1 (4 біти, 25 МГц) та 2 (4 біти, 50МГц).

Керуючі піни :

EN – для увімкнення чипа, при отриманні одиниці чип активується, при отриманні значення нуля переходить у режим мінімального енергоспоживання

RST – для перезапуску ESP8266

WAKE – для виводу чипу із режиму глибокого сну

2 АПАРАТНА СКЛАДОВА ДИСТАНЦІЙНОГО ПЕРЕМИКАЧА

2.1 Принцип та алгоритм роботи пристрою

У межах цієї роботи було поставлено задачу розробити електричний пристрій, котрий зможе виконувати механічне дистанційне натискання на кнопку, у кількох режимах роботи, на базі контролера ESP. Мною було обрано контролер NodeMCU на ESP8266 (варіант плати Lolin).

Пристрій планується для використання у домашніх умовах задля автоматизації приладів, які не відносяться до пристроїв підтримуючих концепцію інтернету речей.

Дистанційний натискач має:

1. Мати можливість для дистанційного керування.
2. Працювати у кількох режимах роботи
3. Підтримувати можливість відкладеного натиску через певний проміжок часу

У зв'язку з тим що мною буде створено демонстраційний варіант пристрою для фізичної демонстрації виконаної роботи, він має відповідати певному ряду вимог для подальшого транспортування та використання , а саме:

1. Безпечність експлуатації
2. Надійність роботи
3. Можливість заміни компонентів у разі виходу пристрою з ладу
4. Міцність та адаптивність корпусу

Для початку потрібно проаналізувати алгоритм роботи пристрою, для кращого розуміння подальших етапів таких, а саме збірка та написання програмного коду пристрою. Отже при ввімкненні пристрою починається пошук та під'єднання до мережі, котра заздалегідь була описана у програмній частині приладу (у разі технічних перебоїв або вимкнення необхідної мережі пошук не припиняється), після чого необхідно перейти у додаток для перевірки підключення та початку використання функціоналу пристрою. Сам девайс при старті одразу переходить у режим очікування команди та у разі її отримання, відразу виконує після чого повертається знову до режиму очікування (Рисунок 2.1)

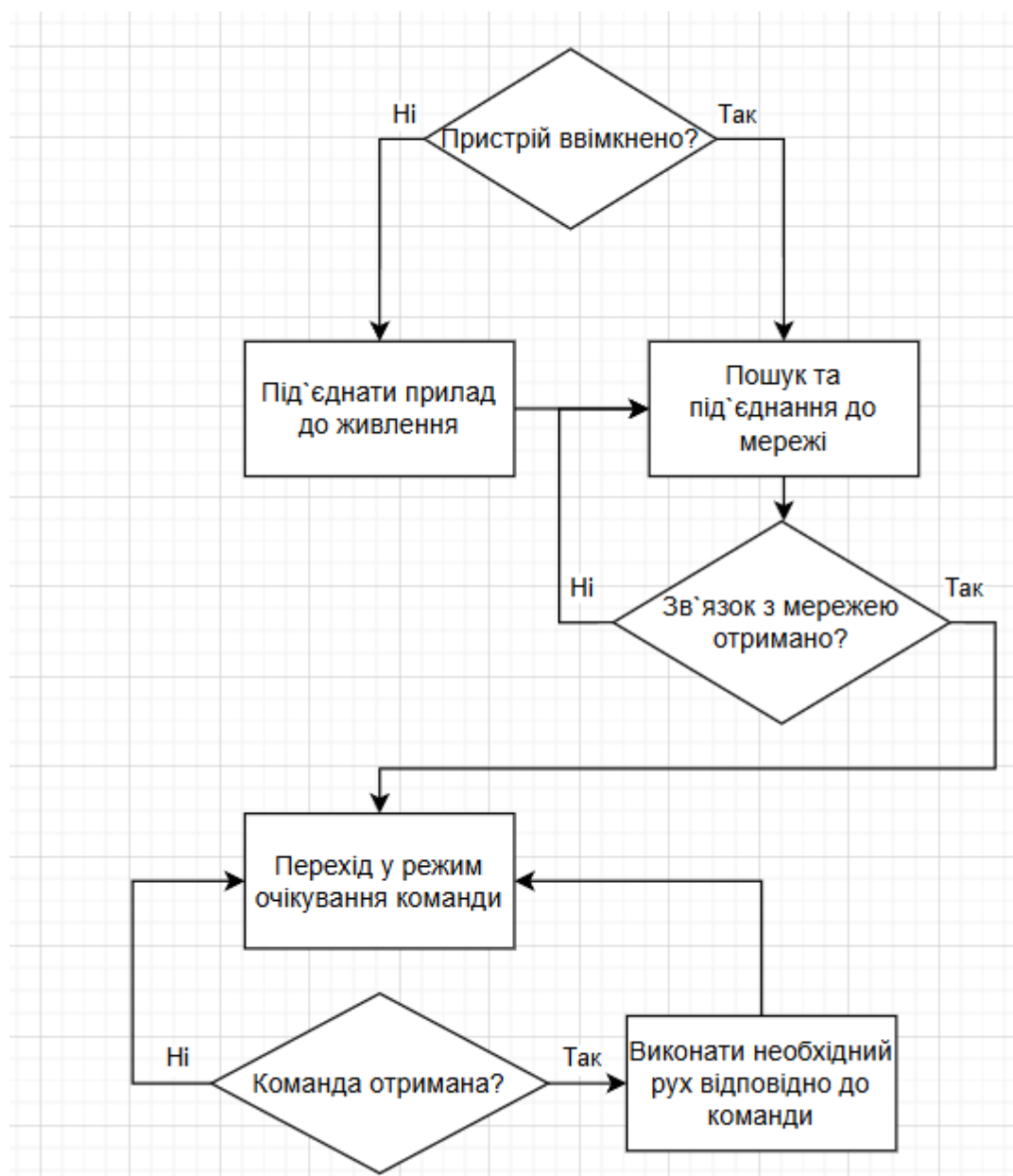


Рис. 2.1. Алгоритм роботи пристрою

2.2 Компоненти та схема їх з'єднання

Відповідно до поставленої задачі, функцій та вимог приладу, що були прописані раніше, для реалізації усіх поставлених задач пристрій має складатися із наступних елементів, а саме:

1. Кроковий двигун на 5V моделі 28BYJ-48
2. Драйвер для крокового двигуна ULN2003
3. Плата для керування NodeMCU на базі ESP8266
4. Живлення на 5 V

Модуль NodeMCU (ESP8266) буде використано у проєкті через необхідність у віддаленому керуванні пристроєм з використанням технології Wi-Fi, у моєму випадку буде використовуватися застосунок Blink, оскільки в нього зручний інтерфейс та широкий безкоштовний план якого достатньо для реалізації функціоналу цього девайсу. Детальніша інформація щодо NodeMCU була описана у кінці попереднього розділу із особливостями модулів ESP та детальним описом пінів та їх призначенням. Вибір ESP версії 8266 зумовлено меншою вартістю ніж модулі інших серій, високою надійністю роботи та майже усім необхідним функціоналом для виконання поставлених задач. У перспективі можна замінити ESP8266 на ESP32-C5, оскільки ця версія здатна працювати із 5 ГГц Wi-Fi, котрий почав набирати популярність останнім часом, але данні модулі наразі існують лише у вигляді прототипів та документацій (Згідно інформації на офіційному сайті)

Модуль буде приймати команди із застосунку і давати сигнал кроковому двигуну для обертання на певну кількість градусів, необхідну для натискання на майже будь-яку кнопку або клавішу .

Перебравши більшість доступних варіантів крокових двигунів, мною було обрано двигун моделі 28BYJ-48 на 5V, оскільки співвідношення його вартості до крутного моменту $\sim 34,3$ МН/м є майже ідеальними для цих цілей. Лінійні сервоприводи було пропущено через їхню високу вартість та складність адаптації для виконання конкретно натискального руху з можливістю зміни насадок для розширення списку можливих для перемикання кнопок. Деталі розмірів двигуна знаходяться на рисунку 2.2. Ключовими параметрами крокового двигуна є:

Номінальна напруга для живлення складає 5 V(постійного струму) на 1 Ватт

Загальна кількість кроків та мікрокроків, яка впливає на точність обертання ротору двигуна : 64 кр на 4096 МКр (важливо зазначити що кількість мікрокроків округлено, оригінальна кількість становить 4075.7728395 МКр і задається у коді, у разі використання інших даних двигун буде працювати некоректно)

Рівень шуму при роботі складає < 40 ДБ, при частому використанні пристрій майже не буде створювати зайвий шум.

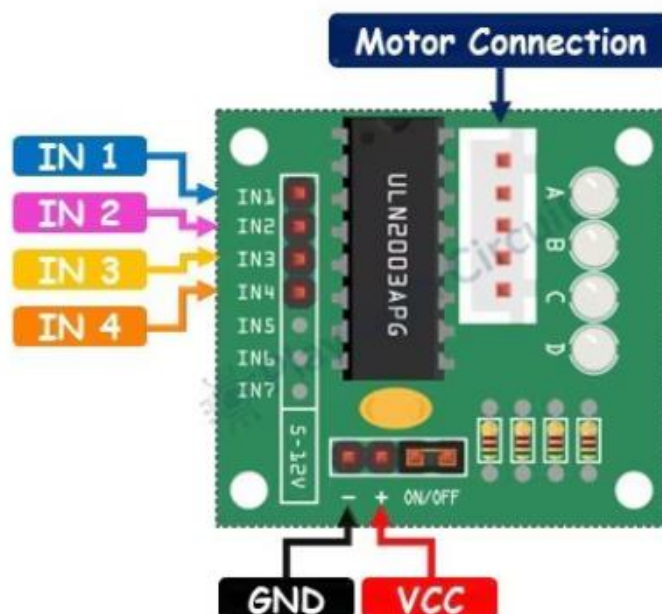


Рис. 2.3. Розпіновка мікросхеми драйверу ULN2003

Найважливішим елементом схеми є джерело живлення, оскільки прототип девайсу створюється у демонстраційних цілях, мною було обрано модуль для живлення макетних плат 3,3 V – 5 V з кабелем USB Type A – USB Type A. Оскільки для роботи кожного із компонентів необхідно саме 5 V. Принцип роботи досить простий, при подачі живлення (через DC роз'єм або USB порт) необхідно впевнитися що модуль включено за допомогою кнопки на платі, після чого плата подає обрану напругу на обидві пари пінів із обох сторін. Вибір напруги здійснюється за допомогою спеціальних перемикачів, які замикають певні піни на платі, всього доступно три варіанти: 3,3 V , 5 V або вимкнути (OFF). Загалом модуль має наступні компоненти та піни (Рисунок 2.4) :

Роз'єм DC : для підключення зовнішнього джерела живлення на 6,5V -12V (для зазначених 15 V бажано додаткове охолодження для запобігання перегріву)

USB порт: для альтернативного живлення на 5V (У моєму випадку основного)

Два перемикачі для вибору вихідної напруги для кожної шини макетної плати окремо

Кнопка увімкнення/вимкнення подачі живлення на модуль

Світлодіод індикації живлення, світиться при подачі та гасне при вимкненні плати або позбавленні джерела живлення

Додаткові піни-виводи для 3,3 V , 5 V та GND для підключення зовнішніх пристроїв.

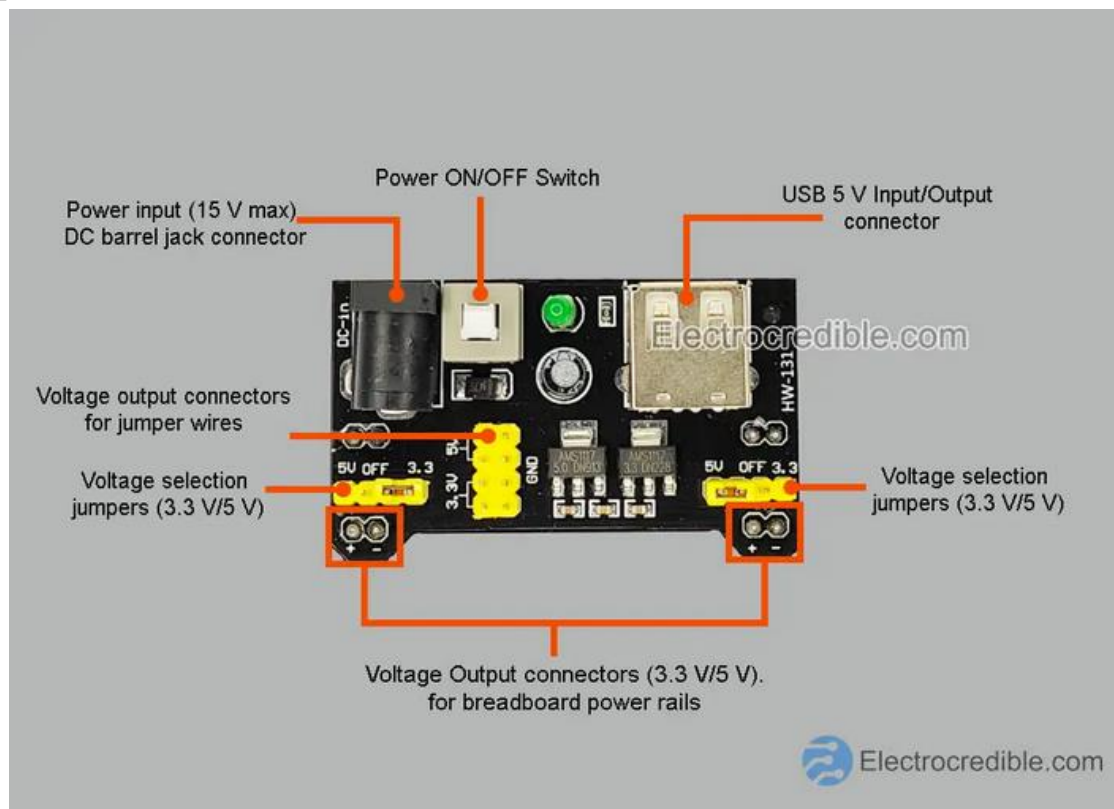


Рис. 2.4. Розпіновка модуля живлення

Отже ми переглянули основні компоненти пристрою, тепер необхідно зібрати їх у систему задля подальшої розробки коду та реалізації механізму передавання обертального руху до поступального. Схема з'єднання компонентів зображена на рисунку 3.6. Для початку необхідно під'єднати плату NodeMCU версії Lolin з ESP8266 та драйвер ULN2003 до модуля живлення, важливо, встановити обидві перемички на обох сторонах на пункт 5 V. Після чого вставляємо штекер від крокового двигуна 28BYJ-48 до спеціального роз'єму драйверу, важливо упевнитися що пази для центрування та вірного напрямку співпадають оскільки компоненти мають певні зазори. Після цього під'єднуємо вільні піни драйверу до NodeMCU. Більш детальна інформація щодо з'єднання подана у таблиці 2.1. Якщо все зібрано правильно то при ввімкненні кабелю USB до ноутбука або павербанки загориться зелений світлодіод, якщо він не загорівся то

це може бути спричинено кількома факторами: кабель, що використовується дефектний; (у варіанті ноутбуку) не розпізнається підключення, можливо некоректно працює роз'єм; потрібно перевірити чи ввімкнений модуль живлення.

Піни підключення компонентів

Таблиця 2.1

NodeMCU ESP8266	Плата живлення	ULN2003
D1	=>	IN1
D2	=>	IN2
D3	=>	IN3
D4	=>	IN4
GND	GND	5V (-)
VIN	5 V	12V (+)

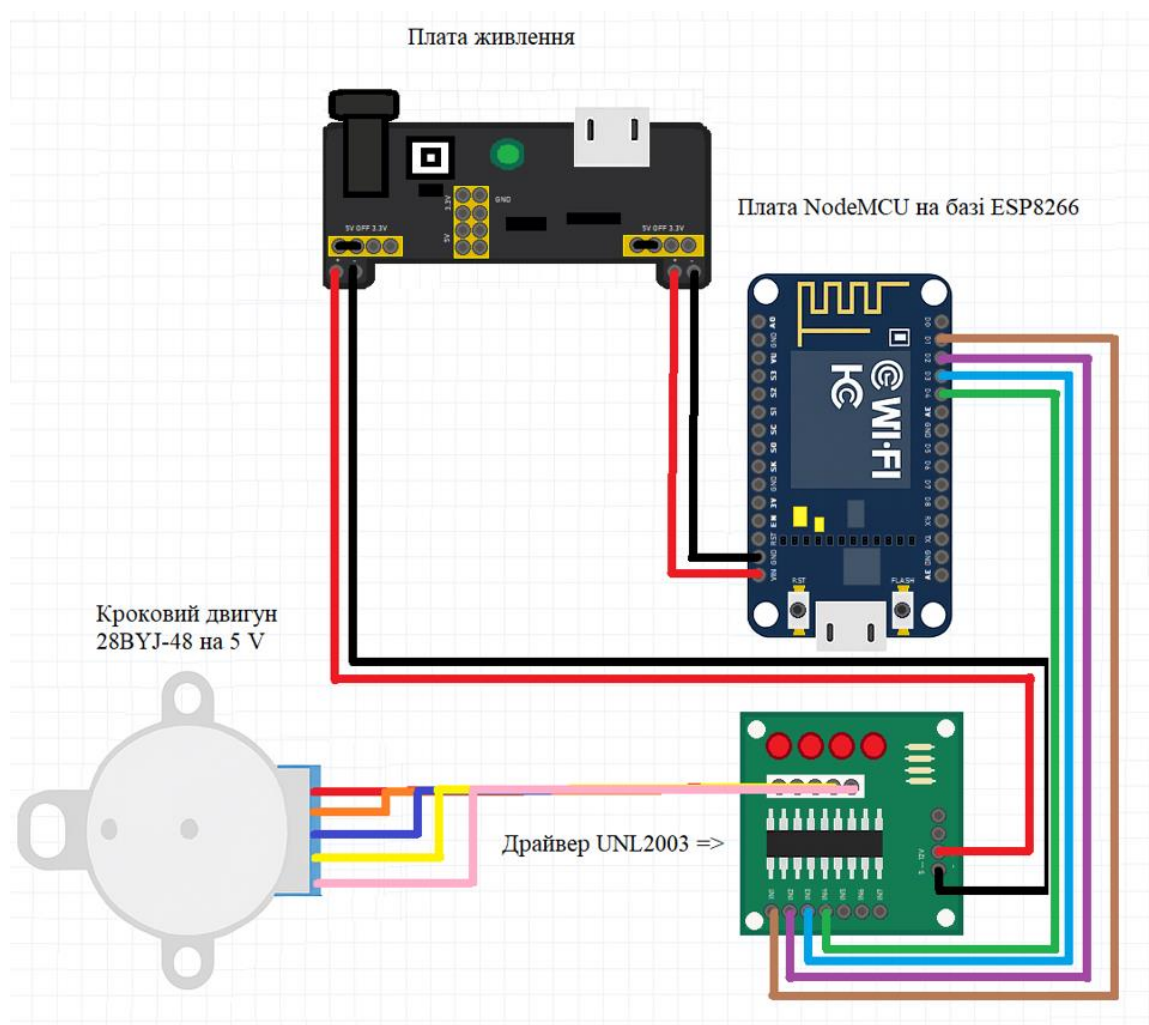


Рис. 2.5. Схема з'єднання компонентів

2.3 Механічна складова пристрою та корпус

Оскільки ротор крокового двигуна виконує хоч і точні, але виключно обертальні рухи, необхідно продумати оптимальний механізм, що буде відповідати за перетворення. Існує багато механізмів для подібного перетворення руху, але не всі з них підходять для виконання даної задачі. Наприклад механізм Ліпкіна-Посельє, перший у світі плаский шарнірний механізм, що був винайдений ще у 1864 році (Рисунок 2.6.), працює за принципом обертання у точці із синьою лінією, за умови що всі частини однакового кольору мають однаковий розмір та з'єднані вільними точками.

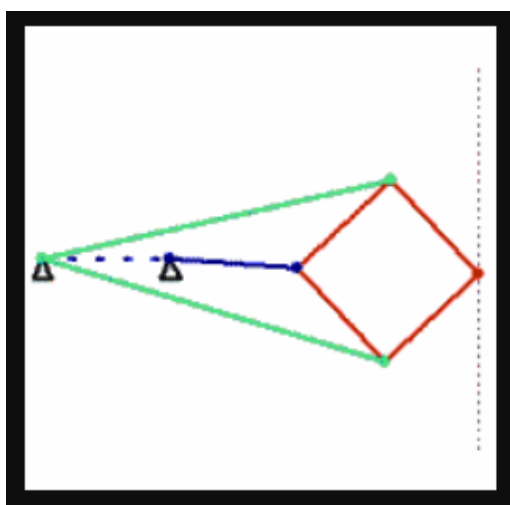


Рис. 2.6. Механізм Ліпкіна-Посельє

Проблема цього механізму у його розмірі та принципі передавання сили, відповідно до чого цей механізм не є оптимальним оскільки не надає достатньо сили для натиску кнопки, також особливість конструкції буде заважати при використанні. Також варто згадати досить базовий механізм «Рейка та шестерня» (Рисунок 2.7).

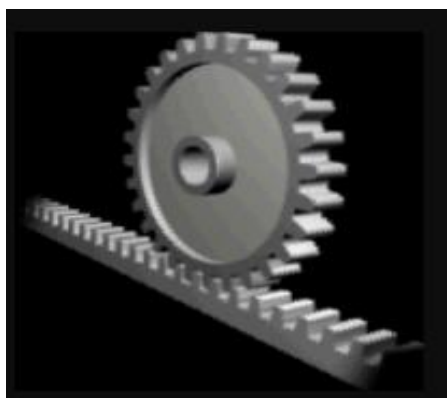


Рис. 2.7. «Механізм Рейка та шестерня»

Принцип роботи механізму простий, при обертанні шестерні вона рухає прилягаючу до себе рейку, залежно від напрямку обертання, напрям руху рейки змінюється. У цього варіанта цілий ряд переваг таких як: компактність, надійність та достатня сила передачі. Але для коректної роботи даного механізму потрібні точно підібрані деталі з достатньо січних матеріалів, оскільки при частому використанні (за умови використання пластмасових деталей) вони будуть стиратися та досить швидко виводити механізм із робочого стану, також важливо зазначити що швидкість спрацювання буде дуже малою, що негативно впливає на фіксацію настику певними варіантами кнопок.

Оптимальним варіантом для виконання цієї задачі є «Кривошипно-шатунний механізм». Цей варіант напевно один із найбільш розповсюджених механізмів у світі, адже саме він використовується у автомобільних двигунах. (Рисунок 2.8)

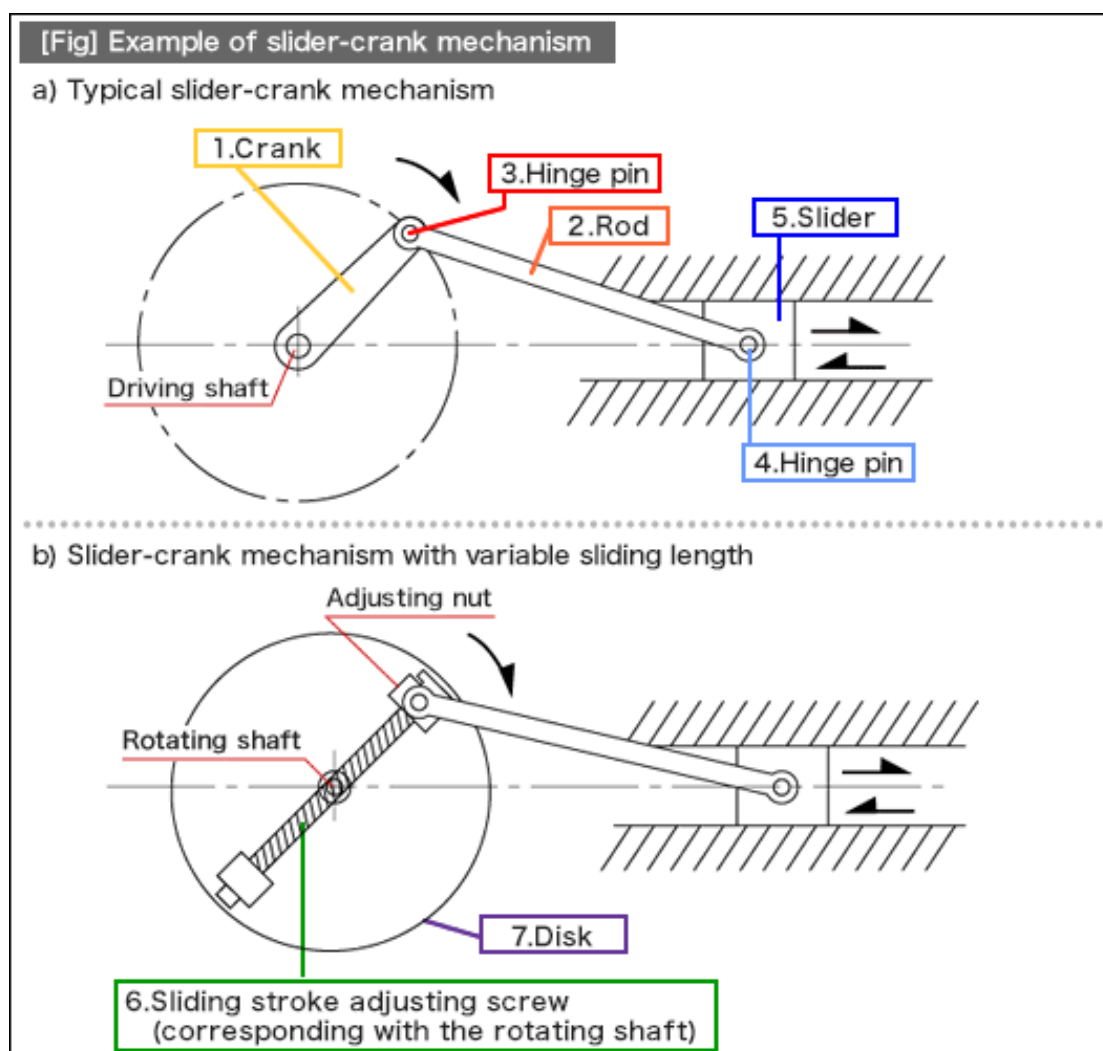


Рис. 2.8 Кривошипно-шатунний механізм

Принцип роботи трохи простіший ніж у механізмі Ліпкіна-Посельє, ротор обертає вал що з'єднано вільно із передатчиком, котрий у свою чергу тим же самим з'єднанням зв'язано із частиною що буде виконувати лінійний рух, важливо зазначити, що заради коректної роботи механізму обов'язково необхідна направляюча. Даний механізм може працювати у обидві сторони та має достатньо: сили натиску, обсягу вільного місця конструкції для комфортного використання та налаштування, швидкості роботи. Тому саме механізм цього принципу буде використаний у фізичному прототипі приладу.

Для створення корпусу я використав залізний конструктор, оскільки він має велику варіативність використання та модифікації, а також є достатньо міцним та важким, що буде корисно при транспортуванні та виконанні своєї прямої задачі – натискання кнопок. Реалізація приладу була розпочата зі створення самого механізму передачі обертання ротору у поступальний рух, а саме припасування круга на ротор крокового двигуна, під'єднання рухомої «лапки» та приєднання її до направляючої, у моєму випадку залізної палички. (Рисунок 2.9)

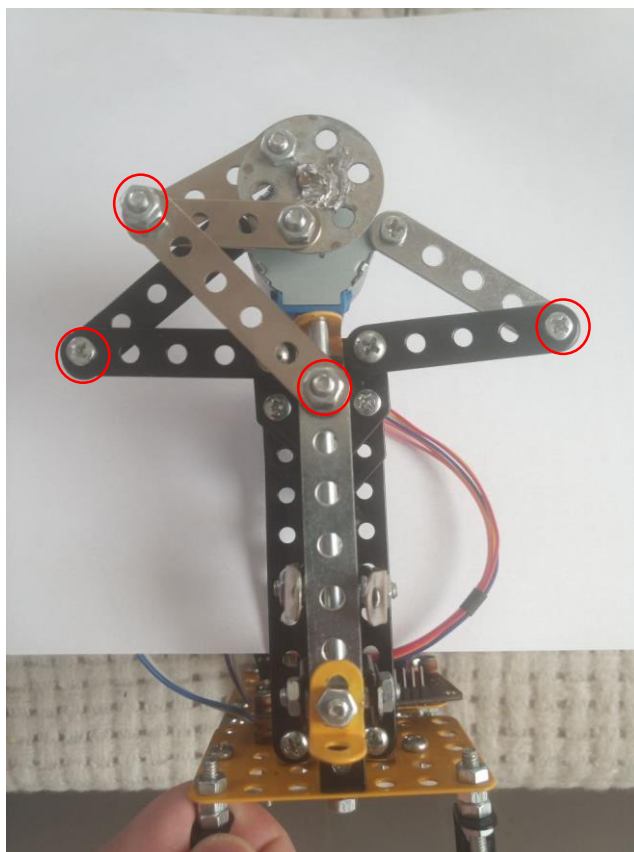


Рис. 2.9 Механізм передачі обертального руху у лінійний

Оскільки пази кріплення крокового двигуна трохи не співпадають із стандартним кроком конструктора, було додано додатково по дві деталі з обох сторін для кращого центрування двигуна відносно направляючої. На зображенні усі кріплення крім тих що виділені є сталими, без можливості люфту, виділені ж мають можливість вільно обертатися у обидві сторони. Також варто зазначити що сама «ніжка»(жовта деталь на кінці лапки) може бути замінена під конкретні варіації кнопок або перемикачів, у тому числі і сенсорних їх варіацій.

Після створення основної механічної складової приладу необхідно створити корпус для противаги самого механізму, щоб стабілізувати роботу та мати можливість налаштувати необхідну висоту роботи. Важливо зазначити що без подання напруги сервомотор можна обертати руками, для відладки перемикання певної кнопки необхідно повністю опустити ніжку щоб вона натискала на кнопку, у разі недостатньої висоти можна збільшити або зменшити розмір стійок самого корпусу. Детальніше зображено на рисунку 2.10.

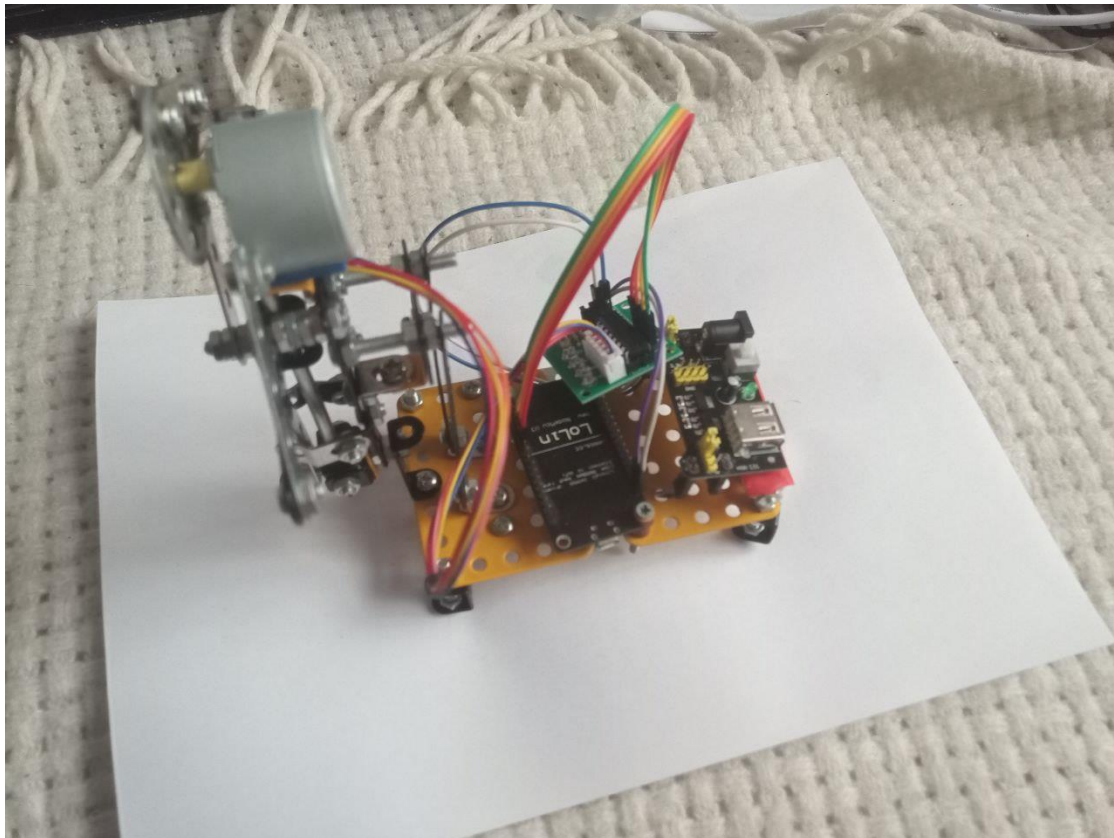


Рис. 2.10 – Демонстраційна модель приладу

3 ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДИСТАНЦІЙНОГО ПЕРЕМИКАЧА НА БАЗІ ESP

3.1 Налаштування середовища розробки

Після збірки схеми та корпусу пристрою слід підготувати робоче середовище для написання програмного коду. Для роботи було обрано середовище «Arduino IDE». Це застосунок з відкритим кодом, що дозволяє досить комфортно писати код на мові програмування «C#». Але у базовій комплектації ця програма не бачить ESP8266 тому необхідно у налаштуваннях встановити пакет параметрів для плат : « http://arduino.esp8266.com/stable/package_esp8266com_index.json » у поле « URL менеджерів додаткових плат » (Рисунок 3.1)

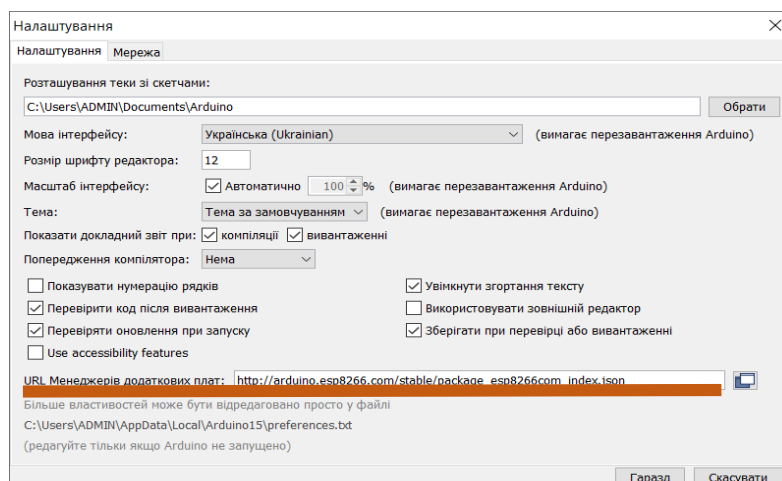


Рис. 3.1 – Налаштування додаткових плат

Далі необхідно через вкладку «Інструменти» → «Плата:» → «Менеджер плат» встановити «ESP8266 Community», це пак з переліком усіх можливих видів ESP плат на їхніх налаштувань(Рисунок 3.2)

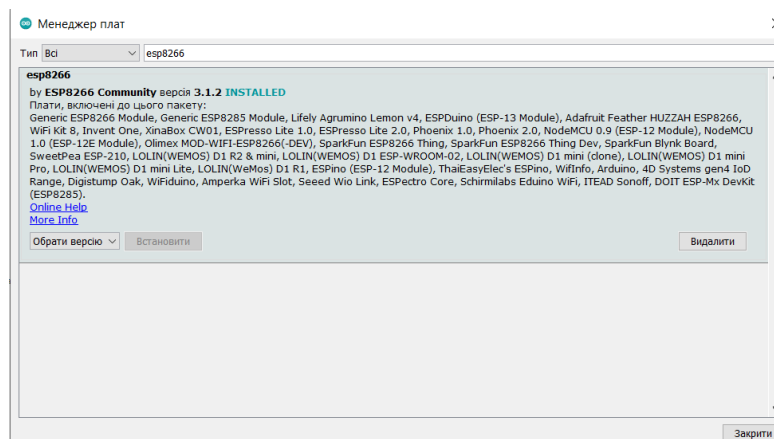


Рис. 3.2 – Встановлення плат ESP

Після цього треба перейти знову до вкладки « Плата: » та обрати варіант, який я використовую, а саме « Node MCU v1.0 (ESP 12 – E Module) » (Рисунок 3.3)

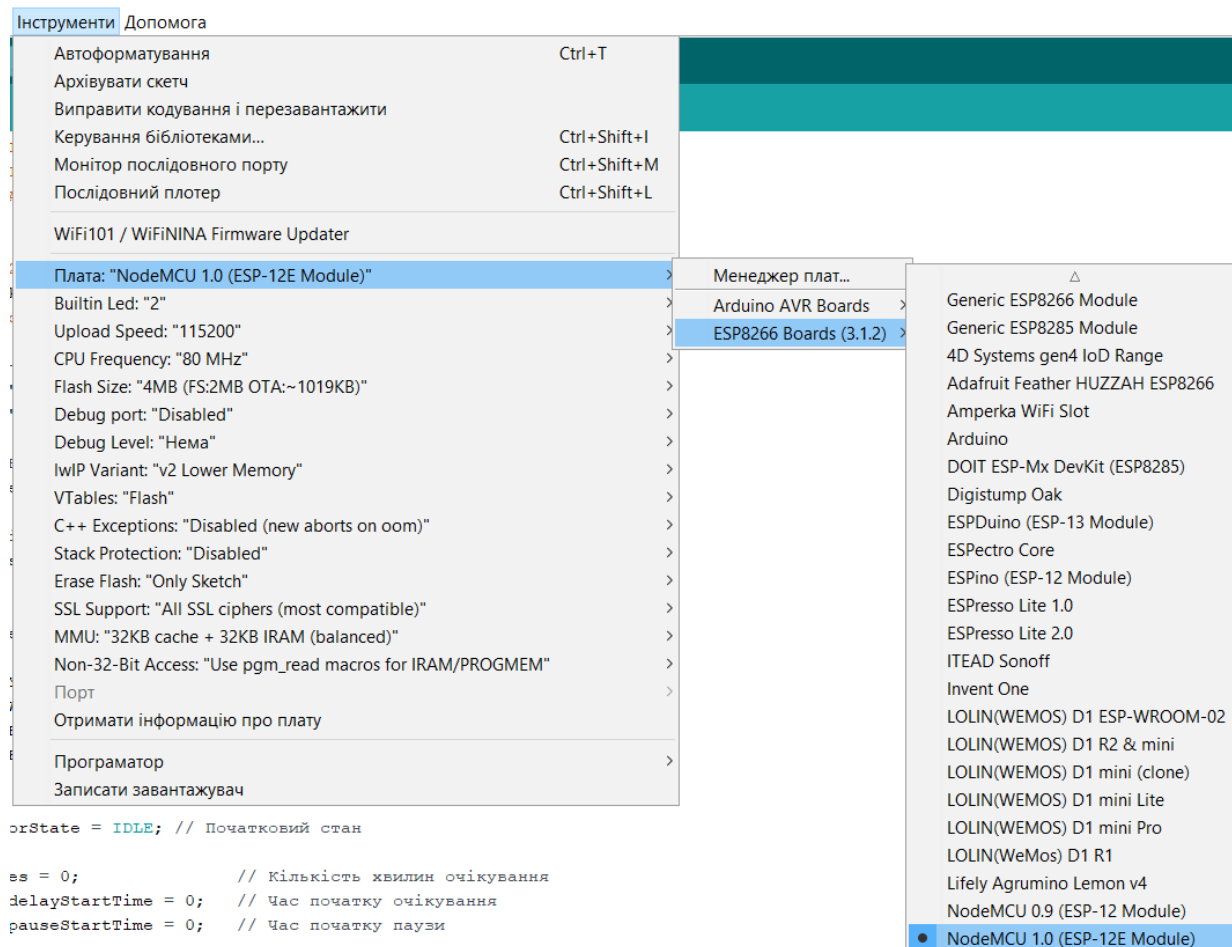


Рис. 3.3 – Вибір плати

Середовище підготовано, але ще потрібно завантажити драйвер «CH340G», оскільки без нього комп'ютер буде сприймати під'єднану плату некоректно, у наслідку чого буде неможливо прошити саму плату NodeMCU на ESP. Цей процес необхідний також для зміни налаштованої мережі для під'єднання пристрою. Для цього необхідно перейти до «Диспетчеру пристроїв», де знайти вкладку «Порти (COM та LPT)», якщо драйвер вже було встановлено то там буде знаходитися «USB-SERIAL CH340G (назва порту)», у такому випадку все працює коректно і можна переходити до наступного кроку. (Рисунок 3.4) Але, у разі відсутності його необхідно перейти до налаштувань «Прилади та принтери», це найшвидший спосіб відслідкувати прилад, що некоректно сприймається системою. Там у пункті інші девайси буде знаходитись щось на кшталт «Unknown device», на який треба

натиснути правою кнопкою миші, обрати «Властивості». У вікні що відкрилося потрібно перейти до вкладки «Обладнання» та знову натиснути на кнопку «Властивості». У третьому вікні переходимо до вкладки «Драйвер» та обираємо опцію «Оновлення драйвера», після чого з'являється вікно вибору варіанту оновлення (Рисунок 3.5), де ми обираємо встановити з комп'ютера та обираємо папку драйвера(її потрібно завантажити з мережі інтернет). Після цього плата почне розпізнаватися комп'ютером коректно.

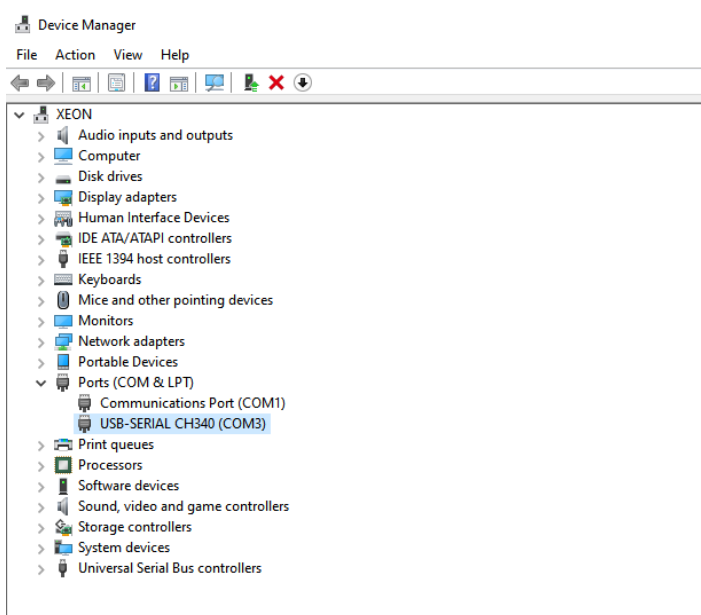


Рис. 3.4 Драйвер CH360G у наявності

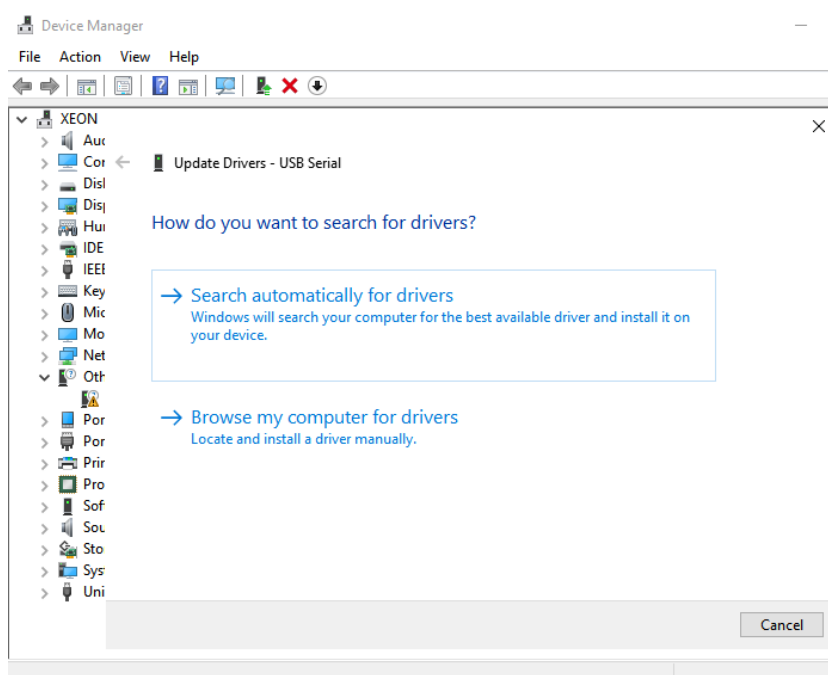


Рис. 3.5 – Встановлення драйвера

Важливою складовою налаштування є під'єднання або завантаження конкретних бібліотек для написання коду у моєму випадку для виконання поставлених мною задач необхідно завантажити бібліотеки «Blynk» та «CustomStepper». Власне саме бібліотеку «Blynk» можна завантажити через саме середовище ардуїно, а саме потрібно відкрити «Скетч» → «Додати бібліотеку» → «Керування бібліотеками». У менеджері бібліотек потрібно встановити наступний варіант бібліотеки(Рисунок 3.6).

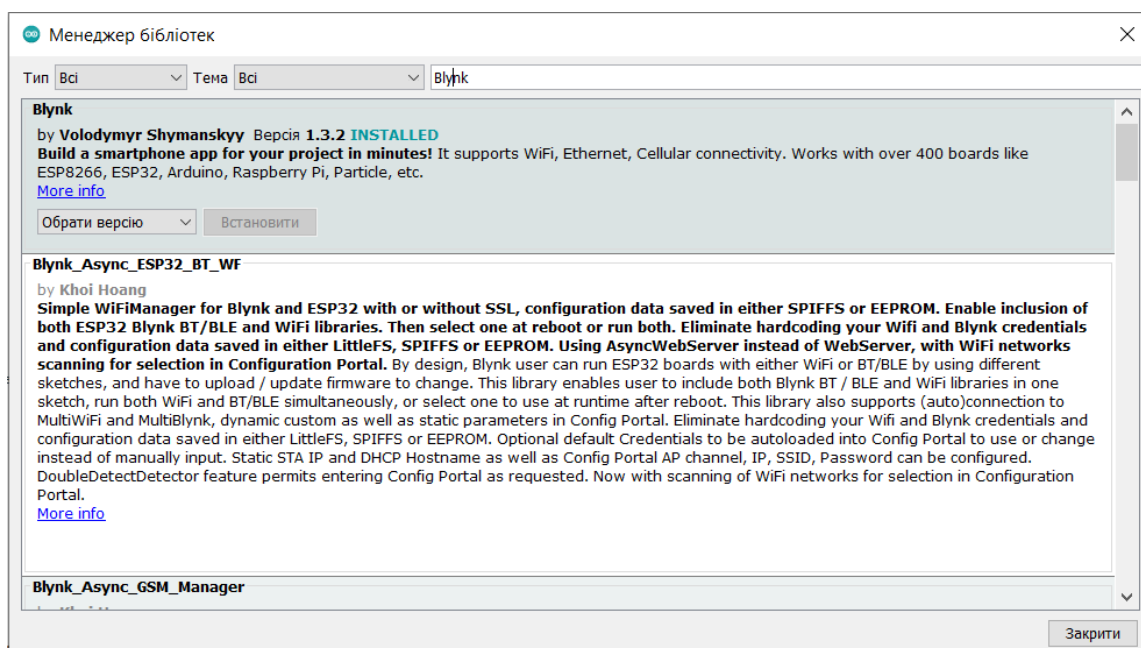


Рис. 3.6 Завантаження бібліотеки Blynk

А от бібліотеку «CustomStepper» завантажити через менеджер бібліотек не вийде, оскільки вона є сторонньою. Для керування кроковими двигунами у ардуїно передбачено бібліотека «Stepper», але вона недостатньо оптимально прописана, наслідком цього є проблеми та перебої роботи двигуна при роботі із нею, саме тому була обрана ця стороння бібліотека. Принцип її завантаження теж відрізняється, для початку необхідно завантажити архів із нею за наступним посиланням «<https://forum.arduino.cc/uploads/short-url/9mPCGLiO6eqbDcwjjG45UnWeD1v.zip>». Після чого потрібно виконати майже ті самі кроки, а саме «Скетч» → «Додати бібліотеку» → «Додати ZIP бібліотеку», та у з'явившомуся вікні задати шлях до цієї бібліотеки. (Рисунок 3.7)

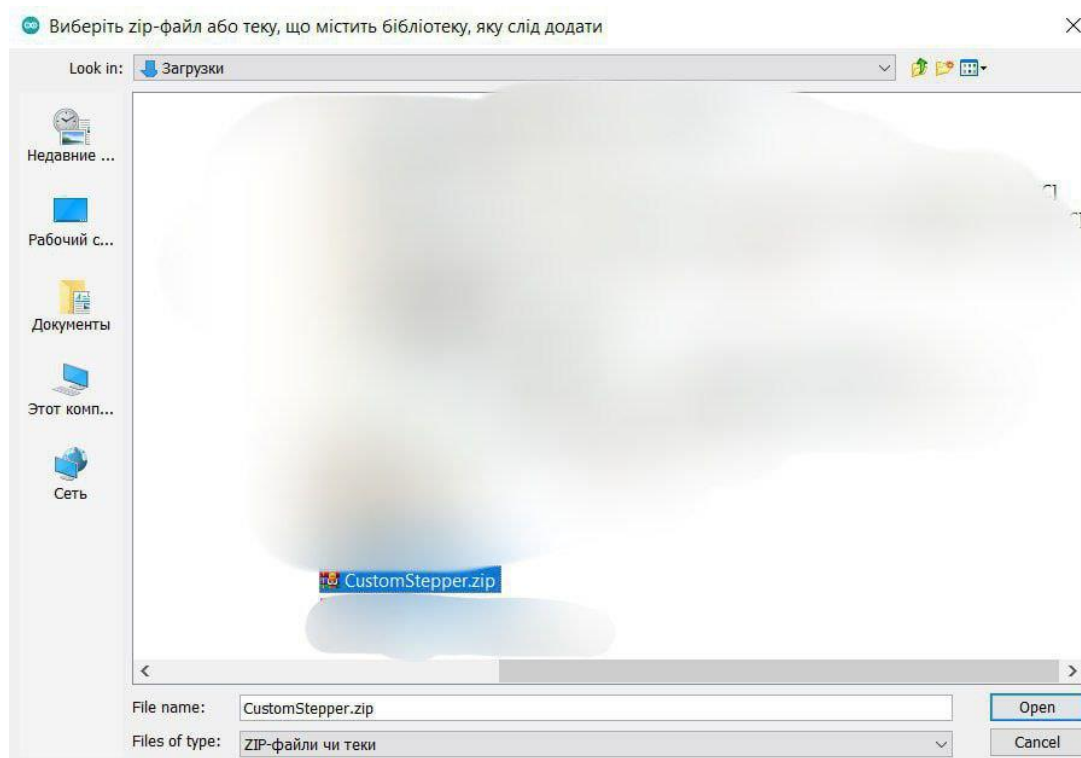


Рис. 3.7 Завантаження бібліотеки CustomStepper

3.2 Під'єднання до сервісу Blynk

Для реалізації можливості дистанційного керування через Wi-Fi я буду використовувати сервіс Blynk. Це комплекс який складається із застосунка, сервера та набору бібліотек що дозволяє створювати додатки для приладів інтернету речей, задля дистанційного керування. Робота у застосунку передбачає вибір плану доступності функцій, за замовченням доступний безкоштовний план із лімітованим часом роботи, що виділяється для серверу та трохи обмеженим переліком доступних віджетів для створення інтерфейсу. Важливо зазначити що у безкоштовній версії також наявний ліміт на їхню кількість, через що реалізація великих проектів не є можливою, але на виконання цього проекту даного функціоналу буде достатньо. Загалом цей застосунок працює з майже усіма самими популярними платами : «ESP», «Arduino», «Raspberry Pi» та інші.

Спочатку необхідно перевірити з'єднання керуючої плати (NodeMCU на ESP8266) з серверами Blynk, для цього переходимо та реєструємося на офіційному сайті «<https://blynk.io/>». Після чого переходимо до вкладки «Devices» (При

створенні нового акаунту вона буде порожня), де натискаємо на кнопку «Help» та обираємо опцію «QuickStart»(Рисунок 3.8)

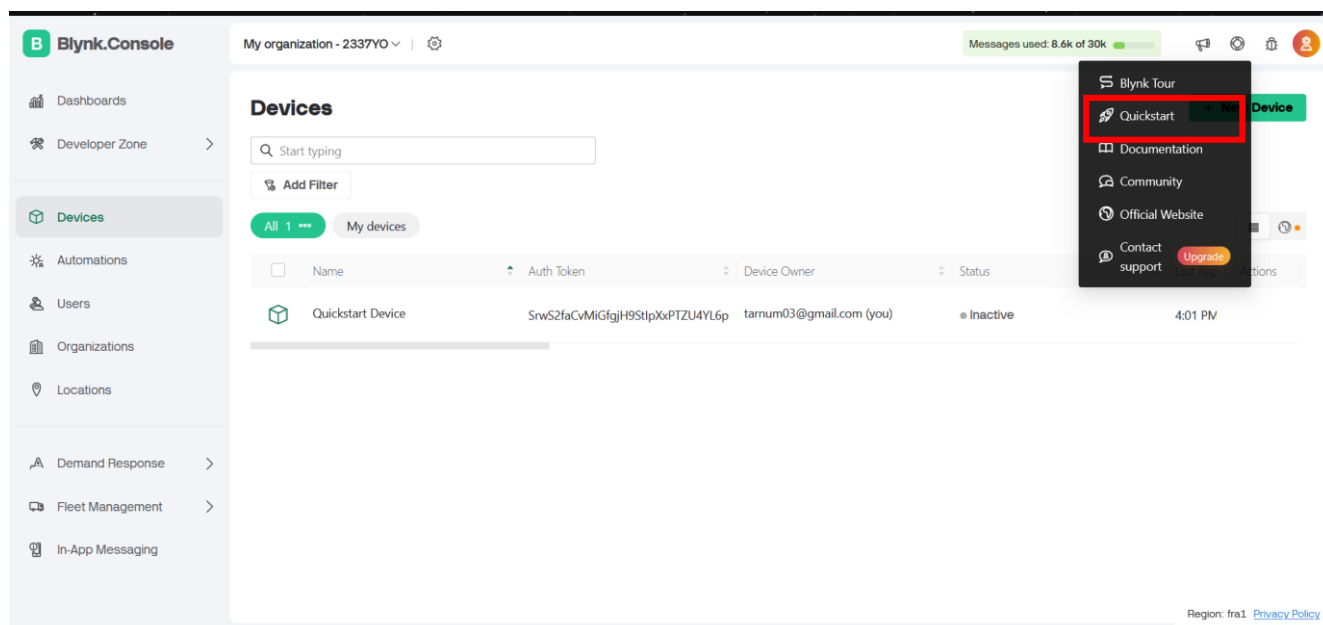


Рис.д 3.8 Створення нового проекту

Після чого проходимо короткий гайд на перший запуск та під'єднання нового девайсу до серверів Blynk, де вказуємо що конкретно за плата використовується та за якою технологією, після чого йде вибір середовища розробки проекту (Рисунок 3.9), за необхідності у цьому пункті є посилання для безпечного завантаження середовища у разі його відсутності, далі йде інформація про необхідність завантаження до середовища роботи саму бібліотеку Blynk(Про це детальніше було описано у пункті «Налаштування середовища розробки»), далі нам дають базову версію коду та два поля для заповнення назви мережі та паролю, після введення інформації необхідно завантажити цей код на плату і нарешті перейти до останнього кроку – активації девайсу (Рисунок 3.10). Після цього ми отримуємо шаблон для створення проекту

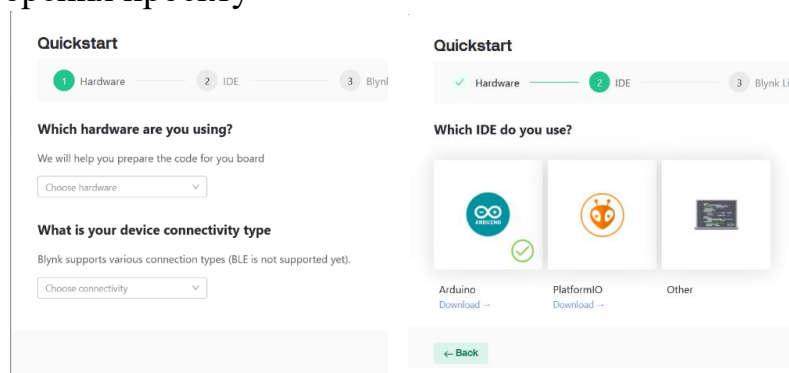


Рис. 3.9 Перші кроки для створення проекту

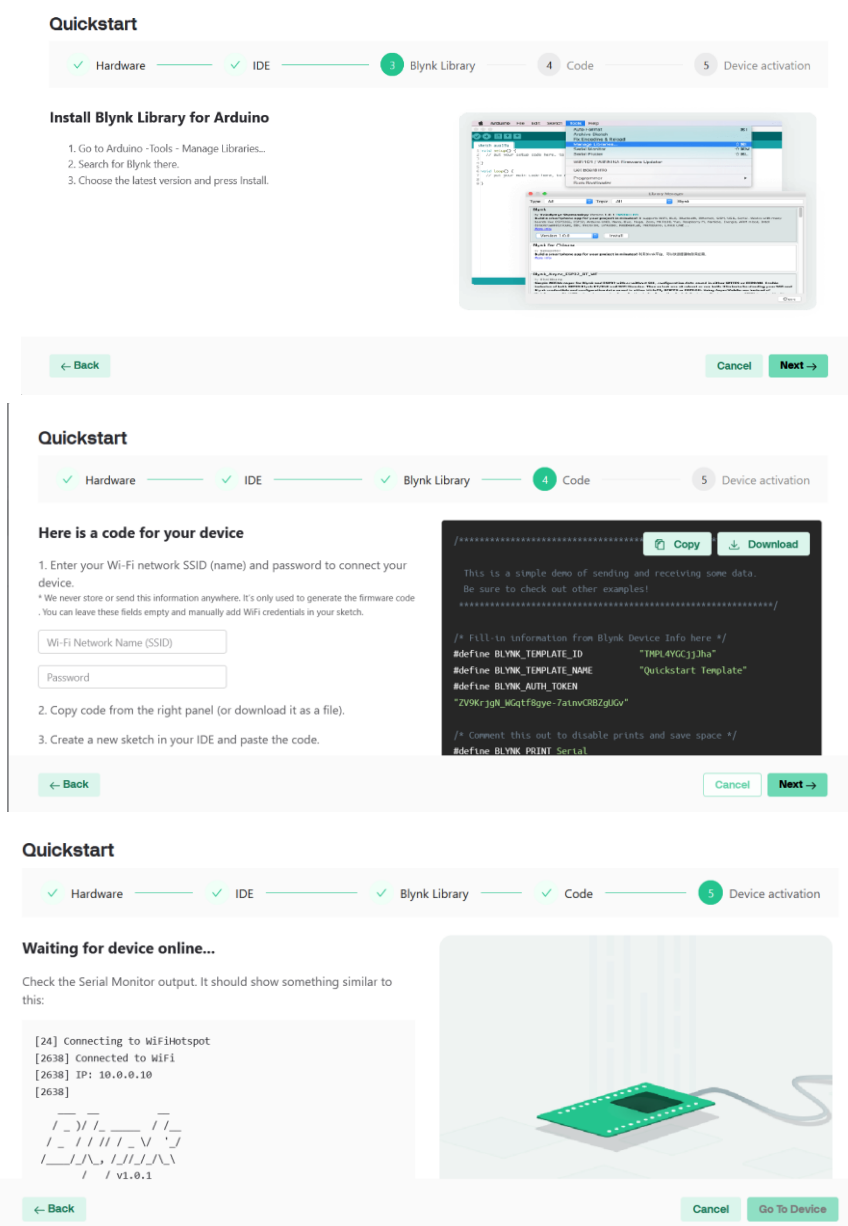


Рис. 3.10 Кроки технічного налаштування для створення шаблону

Також важливо зазначити що інструкція не до кінця вірна, оскільки опускає ряд моментів що необхідні для роботи коду із сервером, а саме необхідно додати такі параметри як назву та ID проекту, за шаблоном пропонується ввести лише токен, який можна скопіювати із вкладки «Devices». Щоб знайти їх потрібно перейти до проекту, зайти у його налаштування та прогортати у низ до «Firmware info», звідки ми і скопіюємо цю інформацію для нашого коду (Рисунок 3.11). Важливо зазначити що ID проекту можна знайти тільки тут у той час як інші два параметри можна побачити на загальній вкладці.

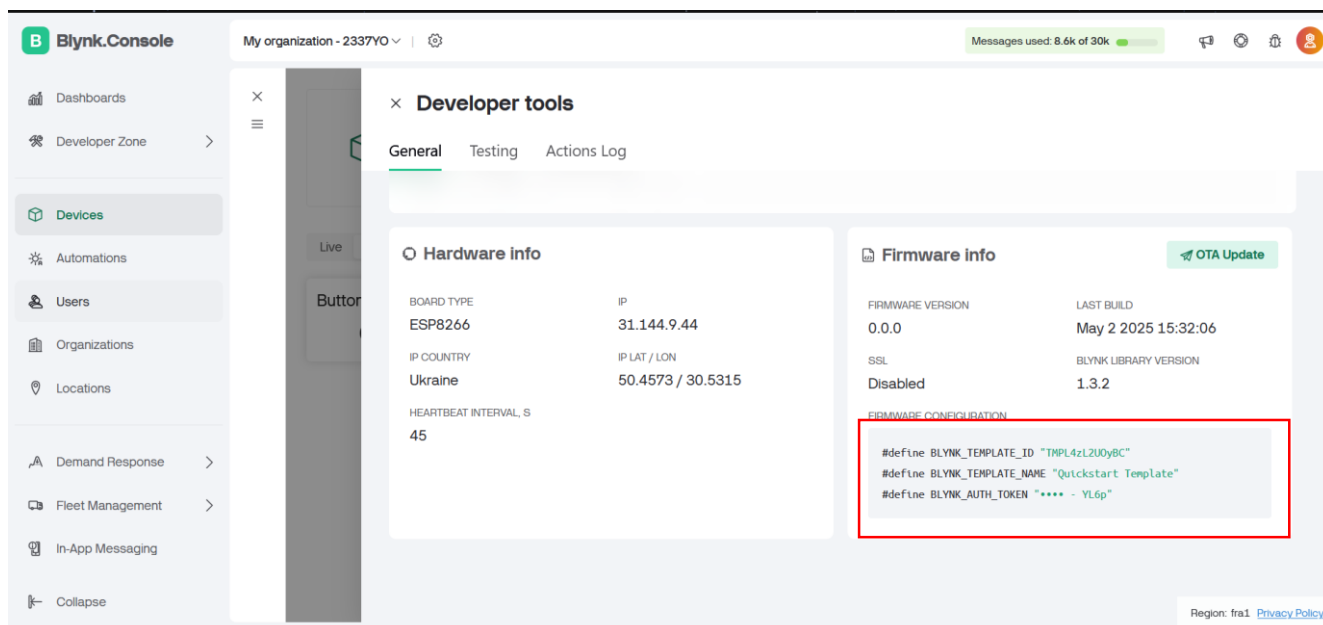


Рис. 3.11 Необхідні параметри для роботи коду

Тепер можна переходити до застосунку для мобільного телефону Blynk, по-перше через те що там більш зручний інтерфейс для створення «пульта керування для пристрою», а по друге через те що для демонстрації буде використовуватися саме варіант керування через телефон. Спочатку необхідно увійти із акануту що використовувався на сайті Blynk у браузері, після чого нас зустріне наступний інтерфейс (Рисунок 3.12)

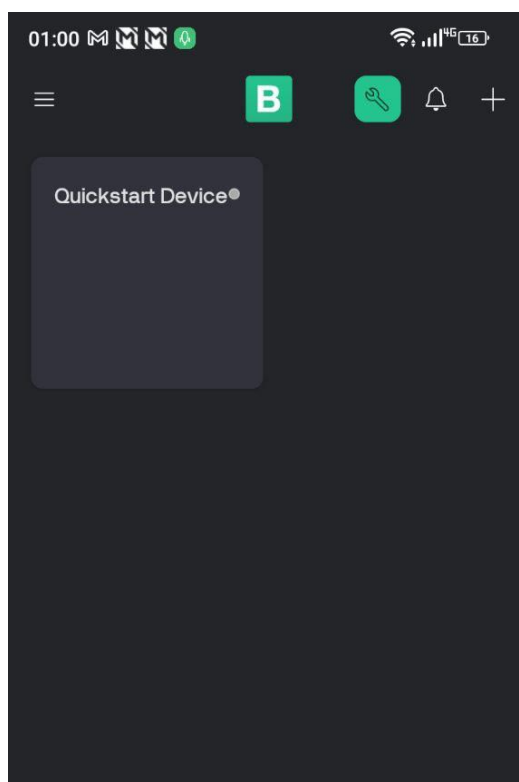


Рис. 3.12 Додаток Blynk

Тут ми обираємо наш створений проект та починаємо створювати його компоненти, редактор проекту представляє собою набір віджетів що можна додавати до робочої області, вони можуть виступати у ролі як простих текстових написів, так і інтерактивних варіантів що змінюються відносно певних подій, також можна додавати кнопки, перемикачі, слайдери і багато чого іншого (Рисунок 3.13), але не всі опції доступні у безкоштовній версії.

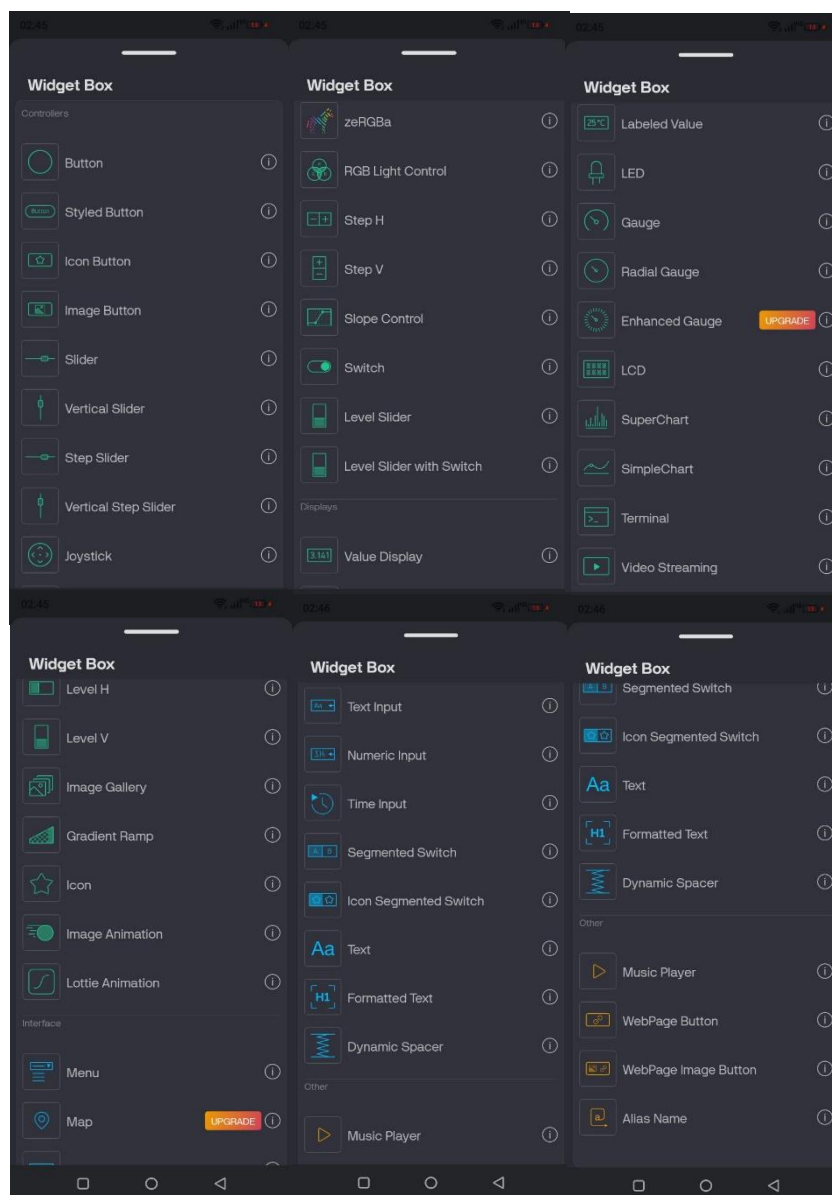


Рис. 3.13 Варіанти віджетів

Також важливо розуміти що для пов'язання створюємих компонентів додатку із кодовою складовою потрібні віртуальні піни та перемінні що пов'язані із ними, вони автоматично створюються для певних об'єктів, та для деяких їх потрібно створювати власноруч. Важливо зазначити що назви змінних потрібні

лише для кращого розуміння за що і де вони використані у самому застосунку, до кодової частини налаштування ESP вони доступні за назвами віртуальних пінів. Всього для мого проекту було використано 4 елементи: кнопка V0 зі змінною «Switch Control», числовий ввід із кроком у одну одиницю V7 зі змінною «Timer», блок меню V8 зі змінною «Working Mode», та показчик перемінної «Seconds» V2.(Рисунок 3.14).

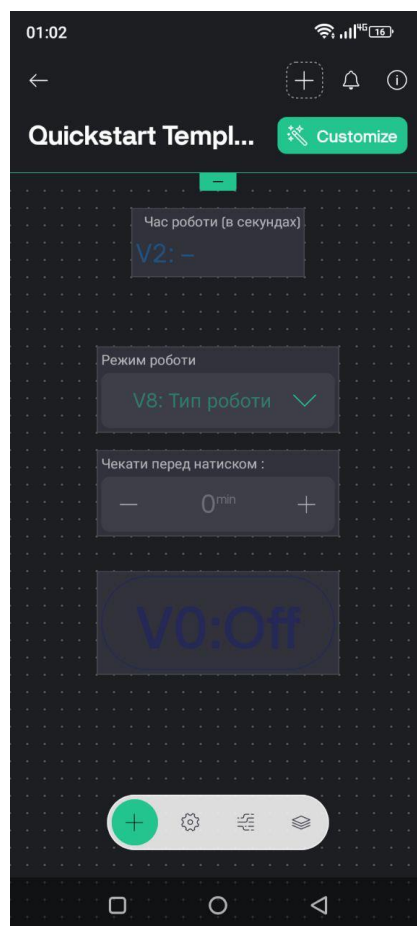


Рис. 3.14. Середя налаштування компонентів

Показчик перемінної «Seconds» це базовий компонент що демонструє у секундах час роботи пристрою, дана функція необхідна для того щоб відслідковувати скільки часу працює прилад, тому що у безкоштовній версії також лімітований час з'єднання з серверами, усього доступно 8 варіантів: одна година, 6 годин, один день, одна неділя, один місяць, 3 місяця, 6 місяців та один рік. У безкоштовній версії максимальний доступний термін роботи це одна неділя. Для кращого розуміння слід зазначити що час лімітовано на один запуск, тобто можливо використовувати прилад на безкоштовному плані, але буде потрібно

кожну неділю його вимикати та вмикати знову. Мій пристрій фігурує скоріше як варіативний інструмент, а отже даних часових лімітів буде достатньо. За замовченням використовується ліміт у одну годину, власне саме через це і необхідний даний таймер.

Поле для числового вводу змінної «Timer», ця перемінна використовується для створення затримки перед будь-якою дією. Де при значенні «0», обраний варіант виконується миттєво, при інших значеннях відбувається затримка на введену кількість хвилин. Це необхідно для збільшення варіативності використання пристрою для автоматичного ввімкнення якогось приладу через конкретний проміжок часу, наприклад для ввімкнення кавоварної машини щоб зранку не чекати доки вона розігріється і тд.

Блок меню зі змінною «Working Mode», це ще одне налаштування направлене на варіативність використання пристрою. Усього доступно два режими: натискання та затискання. У першому режимі при активації кнопки прилад виконає рух та повернеться у початкове положення. Другий режим складається із двох частин, при першому кліку по віртуальній кнопці лапка опуститься та буде тримати кнопку доки не відбудеться другий клік. Важливо зазначити що у момент очікування другої частини руху перемикавання режиму не буде сприйматися до завершення команди. Також можна задати затримку як на початок затискання так і на другу частину, тим самим розширюючи спектр можливих для виконання задач

Кнопка зі змінною «Switch Control», це основний елемент усього проекту, за допомогою цієї кнопки спрацьовує саме натискання на фізичному девайсі, відповідно до обраного режиму та встановленої затримки. Важливо зазначити, що був обраний тип кнопки «Push», тобто при натисканні значення перемінної стає дорівнювати одиниці, а при прибиранні пальця значення знову стає нулем, це важливо оскільки при написанні коду логіка його роботи відштовхується саме від цих параметрів. Також безкоштовна версія підтримує тип «Switch», але він не підходить до заданого функціоналу девайсу. Його можна виставити у самому додатку, але це лише додасть зайві кліки та погіршить процес використання.

Після усіх налаштувань у середовищі Blynk маємо наступний фінальний вигляд «пульта» дистанційного керування для приладу (Рисунок 3.15).

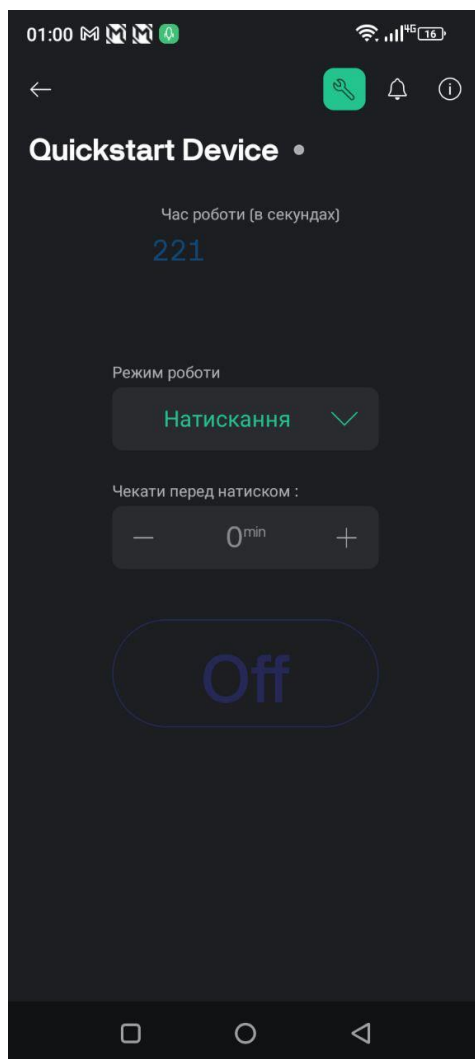


Рис. 3.15 Фінальний вигляд пульта дистанційного керування

3.3 Алгоритм роботи програмної складової

При поданні живлення на пристрій він одразу переходить у стан очікування команди. Після отримання сигналу він приступає до перевірки параметрів команди, спочатку йде перевірка типу команд: «Натискання» чи «Затискання». Це важливо оскільки перший режим складається лише із одного етапу, в той час як другий після першого кліку затискає кнопку і чекає другого щоб припинити процес затискання. (При використанні другого режиму повторний натиск завжди виконує завершення, навіть при прямому переключенні команди). Після чого йде перевірка наявності затримки (в хвилинах), якщо вона є то пристрій чекає виставлений час, а якщо затримки немає то спрацьовує миттєво. Після завершення виконання

команди пристрій переходить у режим очікування нової команди. Алгоритм роботи наведено на рисунку 3.16.

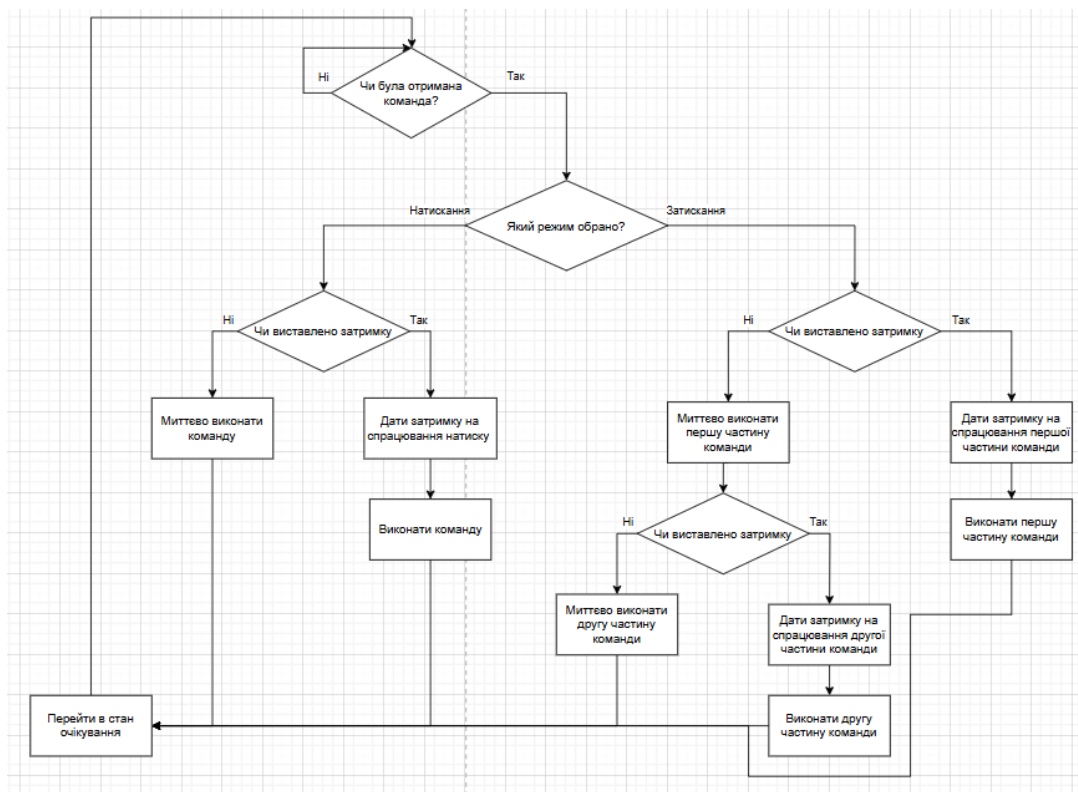


Рис. 3.16 Алгоритм роботи програмного забезпечення пристрою.

3.4 Лістинг коду програми

Для написання програмної складової у середовищі Arduino IDE буде використовуватися мова програмування C#. C# – це універсальна об’єктно-орієнтована мова програмування, створена компанією «Microsoft», що переважно використовується для створення десктопних, мобільних та веб застосунків. Однак через розвиток спеціалізованих фреймворків, а саме «.NET nanoFramework» та «.NET Micro Framework», ця мова програмування знайшла застосування у сфері інтернету речей, у тому числі включаючи мікроконтролери ESP8266 та ESP32.

C# не єдина мова програмування що використовується для подібних проектів, але вона є найбільш популярною, основні переваги це: швидкість розробки, оскільки високо рівневі абстракції дозволяють швидше створювати та тестувати прототипи; безпечність управління пам’яттю, бо завдяки технології «Garbage Collection» зменшується ризик витоків пам’яті; модульність, об’єктно-орієнтований підхід сприяє створенню модульного та масштабованого коду.

Отже для початку потрібно включити усі необхідні бібліотеки, а саме «ESP8266WiFi», «BlynkSimpleEsp8266» та «CustomStepper» (Рисунок 3.17)

```
// Підключення бібліотек
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <CustomStepper.h>
```

Рис. 3.17 Підключення бібліотек

Необхідно також задати параметри для підключення мікроконтролера до серверів Blynk (Рисунок 3.18) та дані для підключення до Wi-Fi (Рисунок 3.19)

```
#define BLYNK_TEMPLATE_ID "TMPL4zL2UOyBC"
#define BLYNK_TEMPLATE_NAME "Quickstart Template"
#define BLYNK_AUTH_TOKEN "SrWS2faCvMiGfgjH9StIpXxPTZU4YL6p"
```

Рисунок 3.18 Параметри, що необхідні для сервера Blynk

```
// Дані для Wi-Fi
char ssid[] = "WebPrayer";
char pass[] = "s0mepass@3w";
```

Рис. 3.19 Дані для з'єднання Wi-Fi

Після цього потрібно ініціалізувати таймер для функції із затримкою спрацювання та сам кроковий двигун (Рисунок 3.20)

```
// Таймер для Blynk
BlynkTimer timer;

// Ініціалізація двигуна на пінах D1-D4
CustomStepper stepper(D1, D2, D3, D4);
```

Рис. 3.20 Ініціалізація необхідних елементів

Далі йде перелік станів машини для коригування рухів(обертів) та задання поточного положення ротору за початкове, оскільки крокові двигуни не мають початкової градації (Рисунок 3.21) та ініціалізація інших основних параметрів, що прописані детально та будуть використані у роботі. (Рисунок 3.22)

```
// Стан машини для контролю послідовності рухів двигуна
enum MotorState {
    IDLE,           // Очікування
    WAITING_DELAY,  // Очікування перед стартом
    ROTATING_FORWARD, // Обертання вперед
    WAITING_PAUSE,  // Коротка пауза
    ROTATING_BACK   // Повернення назад
};

MotorState motorState = IDLE; // Початковий стан
```

Рис. 3.21 Задання станів для роботи крокового двигуна

```

int delayMinutes = 0;           // Кількість хвилин очікування
unsigned long delayStartTime = 0; // Час початку очікування
unsigned long pauseStartTime = 0; // Час початку паузи

int mode = 0; // 0 — натискання (стандартний режим), 1 — затискання (покроковий)
bool halfTurned = false; // Прапорець для пошагового режиму

```

Рис. 3.22 Ініціалізація основних параметрів

Для обробки отриманих сигналів із середовища Blynk для редагування типу виконуємої команди використовується наступний фрагмент, важливо зазначити що для кожної віртуальної змінної потрібно використовувати окрему функцію (Рисунок 3.23).

```

// Обробка натискання кнопки V0
BLYNK_WRITE(V0) {
    int buttonValue = param.asInt();

    // Реакція лише на натискання
    if (buttonValue == 1 && motorState == IDLE) {
        if (delayMinutes > 0) {
            // Якщо задана затримка — починаємо таймер
            delayStartTime = millis();
            motorState = WAITING_DELAY;
        } else {
            // Інакше запускаємо рух негайно
            startMotorAction();
        }
    }
}

// Обробка введення часу затримки з поля V7
BLYNK_WRITE(V7) {
    delayMinutes = param.asInt();
}

// Обробка вибору режиму з поля V8
BLYNK_WRITE(V8) {
    mode = param.asInt();
}

```

Рис. 3.23 Обробка даних із віртуальних параметрів

Для створення режимів роботи було створено функцію «startMotorAction()», у якій описано послідовність обертів для виконання кожного з режимів (Натискання або Затискання), також тут заданий захист від некоректного спрацювання завдяки параметру «halfTurned» (Рисунок 3.24)

```
// функція запуску дії двигуна залежно від обраного режиму
void startMotorAction() {
    if (mode == 0) {
        // Режим натискання — повний цикл туди-назад
        stepper.setDirection(CCW);           // Задання напрямку обертання
        stepper.rotateDegrees(90);           // Поворот на 45°
        motorState = ROTATING_FORWARD;       // Переходимо до наступного стану
    } else if (mode == 1) {
        // Режим затискання — крок вперед або назад
        if (!halfTurned) {
            stepper.setDirection(CCW);       // Поворот вперед
            stepper.rotateDegrees(90);
            halfTurned = true;
        } else {
            stepper.setDirection(CW);        // Поворот назад
            stepper.rotateDegrees(90);
            halfTurned = false;
        }
        motorState = ROTATING_BACK;         // Чекаємо завершення обертання
    }
}
```

Рис. 3.24 Функція запуску двигуна залежно від режиму роботи

Функція setup() викликається лише один раз, коли програма починає роботу та із відключенням живлення що подається на плату NodeMCU, та буде одноразово спрацьовувати кожного разу. Код функції setup() наведено на рисунку 3.24, а лістинг у таблиці 3.1.

```
void setup() {
    Serial.begin(115200);
    Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);

    stepper.setRPM(12);                     // Швидкість обертання
    stepper.setSPR(4075.7728395);           // Кількість кроків на 360° (точне значення для 28BYJ-48)

    // Відправлення часу роботи на V2
    timer.setInterval(1000L, []() {
        Blynk.virtualWrite(V2, millis() / 1000);
    });
}
```

Рис. 3.24 Код функції setup()

Лістинг коду функції setup()

Таблиця 3.1

Рядок коду	Призначення
void setup() {	Початок функції
Serial.begin(115200);	Ініціалізація серійного монітору
Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);	Запуск з'єднання з Blynk через Wi-Fi

Продовження таблиці 3.1. Лістинг коду функції setup()

stepper.setRPM(12);	Встановлення швидкості у обертах на хвилину
stepper.setSPR(4075.7728395);	Встановлення кроків на оберт (точне значення для 28BYJ-48)
timer.setInterval(1000L, []() {	Періодична відправка часу роботи пристрою на пін V2
Blynk.virtualWrite(V2, millis() / 1000);	Відправка секунд з моменту старту
});	Завершення команди timer.setInterval
}	Завершення функції

У setup() задані основні постійні налаштування для монітору порту, крокового двигуна та таймера початку роботи пристрою для відслідкування часу використання через особливості доступності безкоштовного плану Blynk.

Після реалізації функції setup() у якій встановлюються початкові та загальні значення створюється функція loop(), що постійно викликається до поки на плату подається живлення. У цій функції усі операції, що мають відбуватися циклічно, а саме запуск на виконання отриманих команд для крокового двигуна, запуск таймеру на затримку, реакція на введені параметри та обробка поточного стану роботи двигуна. Код функції loop() наведено на рисунку 3.25, лістинг у таблиці 3.2.

```

void loop() {
  Blynk.run(); // Обробка Blynk-подій
  timer.run(); // Запуск таймера
  stepper.run(); // Обов'язковий виклик для роботи CustomStepper.
  // Обробка поточного стану
  switch (motorState) {
    case WAITING_DELAY:
      // Перевірка, чи завершилася затримка
      if (millis() - delayStartTime >= delayMinutes * 60000UL) {
        startMotorAction(); // Запуск руху після очікування
      }
      break;
    case ROTATING_FORWARD:
      // Якщо завершено поворот вперед
      if (stepper.isDone()) {
        pauseStartTime = millis(); // фіксуємо час паузи
        motorState = WAITING_PAUSE; // Переходимо до стану паузи
      }
      break;
    case WAITING_PAUSE:
      // Пауза між обертаннями
      if (millis() - pauseStartTime >= 50) {
        stepper.setDirection(CW); // Напрямок назад
        stepper.rotateDegrees(90); // Поворот назад
        motorState = ROTATING_BACK; // Переходимо до останнього стану
      }
      break;
    case ROTATING_BACK:
      // Очікуємо завершення обертання назад
      if (stepper.isDone()) {
        stepper.setDirection(STOP); // Зупинка мотора
        motorState = IDLE; // Повертаємось у початковий стан
      }
      break;
    default:
      break;
  }
}

```

Рис. 3.25 Код функції loop()

Лістинг коду функції loop()

Таблиця 3.2

Рядок коду	Призначення
<code>void loop() {</code>	Початок функції
<code>Blynk.run();</code>	Обробка Blynk-подій
<code>timer.run();</code>	Запуск таймерів
<code>stepper.run();</code>	Обов'язковий виклик для роботи CustomStepper
<code>switch (motorState) {</code>	Обробка поточного стану
<code>case WAITING_DELAY:</code>	Стан очікування затримки
<code>if (millis() - delayStartTime >= delayMinutes * 60000UL){</code>	Перевірка, чи завершилася затримка
<code>startMotorAction();</code>	Запуск руху після очікування
<code>}</code>	Завершення умови
<code>break;</code>	Завершення стану
<code>case ROTATING_FORWARD:</code>	Стан початку обертання
<code>if (stepper.isDone()) {</code>	Якщо завершено поворот вперед
<code>pauseStartTime = millis();</code>	Фіксуємо час паузи
<code>motorState = WAITING_PAUSE;</code>	Переходимо до стану паузи
<code>}</code>	Завершення умови
<code>break;</code>	Завершення стану
<code>case WAITING_PAUSE:</code>	Стан очікування

Продовження таблиці 3.2. Лістинг коду функції loop()

<code>if (millis() - pauseStartTime >= 50) {</code>	Пауза між обертаннями
<code>stepper.setDirection(CW);</code>	Напрямок за часовою
<code>stepper.rotateDegrees(90);</code>	Оберт на 90 градусів
<code>motorState = ROTATING_BACK;</code>	Переходимо до останнього стану
<code>}</code>	Завершення умови
<code>break;</code>	Завершення стану
<code>case ROTATING_BACK:</code>	Стан для повернення у початкове положення
<code>if (stepper.isDone()) {</code>	Очікуємо завершення обертання назад
<code>stepper.setDirection(STOP);</code>	Зупинка мотора
<code>motorState = IDLE;</code>	Повертаємось у початковий стан
<code>}</code>	Завершення умови
<code>break;</code>	Завершення стану
<code>default:</code>	Перехід до звичайного стану
<code>break;</code>	Завершення стану
<code>}</code>	Завершення обробки поточного стану
<code>}</code>	Завершення функції

Програмне забезпечення дистанційного натискача забезпечує необхідний функціонал для коректної роботи приладу та виконання його основної функції, а саме дистанційного перемикачкн кнопок наявних пристроїв.

ВИСНОВОК

Дослідження розповсюдженості приладів інтернету речей показало, що починаючи з 2023 року і до сьогоднішнього загальна кількість розумних пристроїв в світі набирає все більші тенденції росту, та все ж важливо розуміти що наряду із ними збільшується кількість приладів, що не відносяться до цієї концепції. Загалом основною проблемою, що сприяє поширенню сфери IoT є розповсюдженість та популярність пристроїв без вбудованої можливості для автоматизації та дистанційного керування. Причин для цього безліч, але основною є вартість розумних приладів та необхідність додавання додаткових систем для координації їх роботи.

Особлива увага приділялася сферам використання пристроїв та систем інтернету речей. У результаті проведення аналізу знайдених статистичних даних, серед усіх можливих сфер застосування найпоширенішим став напрям розумних домів та міст (25-27% від загальної кількості). Загалом цей напрям стрімко збільшує кількість пристроїв та систем керування та автоматизації, хоча це переважно відбувається у межах країн, що є передовими лідерами розвитку ринку даного напрямку, а саме США, Китай та деякі країни ЄС.

Основною ідеєю проекту стало розширення можливостей пристроїв, які керуються фізичними кнопками, без необхідності глибокої інтеграції в електронні схеми самих приладів. У ході виконання дипломної роботи було розроблено, реалізовано та протестовано прилад інтернету речей з основною функцією віддаленого керування кнопковими інтерфейсами побутових або промислових приладів. За основу проекту взято мікроконтролер плати NodeMCU на базі ESP8266, який завдяки своїм компактним розмірам, підтримці бездротового з'єднання Wi-Fi, відкритому середовищу розробки та широкій підтримці спільноти став ідеальним рішенням для реалізації пристрою. Також було реалізовано механізм кривошипно-шатунного типу для фізичного натискання кнопок за допомогою мініатюрного електроприводу (зокрема, крокового двигуна 28BYJ-48 у парі з драйвером ULN2003). Для гнучкого управління було передбачено два режими

роботи — «натискання» та «затискання», що дозволяє використовувати систему в різних приладах із широкою варіативністю видів кнопок і перемикачів.

На програмному рівні для управління пристроєм було використано мобільне середовище Blynk, яке надає зручний інтерфейс для зв'язку з контролером через інтернет. Користувач має змогу задавати час затримки перед виконанням дії та обирати режим роботи.

Уся логіка обробки команд, контролю положення мотора та обробки подій реалізована на об'єктно-орієнтованій мові програмування C# у середовищі Arduino IDE із використанням сторонньої бібліотеки CustomStepper, що забезпечує плавність та точність роботи механічної складової приладу. Програмна складова розроблена з урахуванням більшості можливих варіантів неполадок. Алгоритм роботи програми досить простий та універсальний: зчитування обраного режиму та затримка і виконання поставленої задачі, у разі використання режиму «затискання», що складається із двох етапів, другий етап буде виконуватися завжди не звертаючи увагу на зміни режиму, єдине що можна відкоригувати це час затримки на відпускання.

Демонстраційний корпус пристрою складається з металевих елементів, для більшої інформативності принципу роботи приладу та ефективнішого розсіювання тепла від працюючих компонентів у разі довгострокового використання. У перспективі оптимізації та покращення не демонстраційного прототипу девайсу можливо виконати наступні кроки: корпус та механізм передачі можна надрукувати на 3D принтері та замінити макетну плату живлення на акумуляторний блок .

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

- 1 SpringerOpen: “Internet of Things is a revolutionary approach for future technology enhancement: a review” by Sachin Kumar, Prayag Tiwari, Mikhail Zumbler (09 December 2019) [Електронний ресурс] – Режим доступу до ресурсу: <https://journalofbigdata.springeropen.com/articles/10.1186/s40537-019-0268-2>
- 2 The Business Research Company: “IoT Global Market Report 2025” [Електронний ресурс] – Режим доступу до ресурсу: <https://www.thebusinessresearchcompany.com/report/iot-global-market-report>
- 3 Koroglutech: “Статистика та факти про IoT на 2023 рік” [Електронний ресурс] – Режим доступу до ресурсу: <https://koroglutech.com/uk/articles/31601-iot-statistics--facts-for-2023--10-year-forecast>
- 4 Exploding Topics: “Number of IoT Devices” by Fabio Duarte (February 19, 2024) [Електронний ресурс] – Режим доступу до ресурсу: <https://explodingtopics.com/blog/number-of-iot-devices>
- 5 Transforma Insights: “Current IoT Forecast Highlights” [Електронний ресурс] – Режим доступу до ресурсу: <https://transformainsights.com/research/forecast/highlights>
- 6 Builtin: “30 Internet of Things Examples You Should Know” by Mike Thomas (March 19, 2024) [Електронний ресурс] – Режим доступу до ресурсу: <https://builtin.com/articles/iot-examples>
- 7 TechTarget: “Definition smart home” by Kinza Yasar Technical Writer, Sharon Shea Executive Editor (August 2023) [Електронний ресурс] – Режим доступу до ресурсу: <https://www.techtarget.com/iotagenda/definition/smart-home-or-building>
- 8 Blue Ridge: “A Guide to Smart Home Security Devices” by Blue Ridge (November 25, 2024) [Електронний ресурс] – Режим доступу до ресурсу: <https://www.brctv.com/blog/guide-smart-home-security-devices>
- 9 II Всеукраїнська науково-технічна конференція: “Використання хмарних обчислень у цифровій трансформації виробництва” Ужанов І.О. (18

Листопада 2024) [Електронний ресурс] – Режим доступу до ресурсу: https://duikt.edu.ua/uploads/p_2661_83654085.pdf (ст. 189)

10 Earth.Org: “Top 7 Smart Cities in the World in 2024” by Olivia Lai (March 5 2024) [Електронний ресурс] – Режим доступу до ресурсу: <https://earth.org/top-7-smart-cities-in-the-world/>

11 ТМ Мотор: “Типи електродвигунів та їх використання” (3 Травня 2023) [Електронний ресурс] – Режим доступу до ресурсу: <https://tmmotor.ua/ua/news/tipy-elektrovdigatelj-i-ih-ispolzovanie>

12 Espressif. Official site: “Modules” [Електронний ресурс] – Режим доступу до ресурсу: <https://www.espressif.com/en/products/modules>

13 Make-it: “NodeMCU ESP8266 Detailed Review” [Електронний ресурс] – Режим доступу до ресурсу: <https://www.make-it.ca/nodemcu-details-specifications/>

14 Arduino.ua: “Кроковий двигун 5В 28BYJ-48” [Електронний ресурс] – Режим доступу до ресурсу: <https://arduino.ua/prod216-shagovii-dvigatel-5v-28byj-48>

15 Element14: “How to Control 28BYJ-48 Stepper Motor with ULN2003 Driver and Arduino” by rachna34(5 July 2024)[Електронний ресурс]–Режим доступу до ресурсу:<https://community.element14.com/products/arduino/arduino-projects/b/blog/posts/how-to-control-28byj-48-stepper-motor-with-uln2003-driver-and-arduino>

16 Electrocredible: “How to Use a Breadboard Power Supply Module” by Abhilekh Das (November 10, 2024) [Електронний ресурс] – Режим доступу до ресурсу: <https://electrocredible.com/how-to-use-breadboard-power-supply-module/>

17 Wiley Online Library: “Optimize the Synthesis Error in an Eight-Bar Peaucellier–Lipkin Mechanism Using an Objective Function Maximization Approach and Application to Load Lifting” by Abdel Axis Bodie Nguemiengo, Frédéric MBA, Alban Fabrice Lionel Epee, Claude Valery Ngayihi Abbe, Charles Hubert Kom (14 January 2025) [Електронний ресурс] – Режим доступу до ресурсу: <https://onlinelibrary.wiley.com/doi/full/10.1002/eng2.13084>

18 TestBook: “Rack and Pinion Steering System: Learn Diagram, Working, Advantages, and Applications” (12 August 2023) [Електронний ресурс] – Режим

доступу до ресурсу: <https://testbook.com/mechanical-engineering/rack-and-pinion-steering-system>

19 Misumi: “Low Cost Automation Tutorial” [Электронный ресурс] – Режим доступа до ресурсу: <https://www.misumi-techcentral.com/tt/en/lca/2016/08/251-basic-element-of-clever-mechanism-rotary-linear-motion-conversion-mechanism-1.html>

20 Arduined: “CH340 Windows 10 driver download” [Электронный ресурс] – Режим доступа до ресурсу: <https://www.arduined.eu/ch340-windows-10-driver-download/>

21 Arduino Forum: “CustomStepper – A library for stepper motors” (July 2019) [Электронный ресурс] – Режим доступа до ресурсу: <https://forum.arduino.cc/t/customstepper-a-library-for-stepper-motors/110233/1>

22 Oxorona: “Сервіс Blynk” (27.01.2022) [Электронный ресурс] – Режим доступа до ресурсу: <https://oxorona.com/blynk/>

23 CSA Engineering: “Using C# on embedded devices too – we show how it works and the benefits with nanoFramework” [Электронный ресурс] – Режим доступа до ресурсу: <https://www.csa.ch/en/blog/using-c-on-embedded-devices-too-we-show-how-it-works>

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ

Пристрій з використанням ESP для дистанційного керування приладами в мережах IoT

Презентація до захисту бакалаврської роботи
Студента групи ІСД-41
Спеціальності (126) Інформаційні системи та технології
Ужанов Іван Олегович
Керівник кваліфікаційної роботи
док. тех.наук, професор Нестеренко К.С.

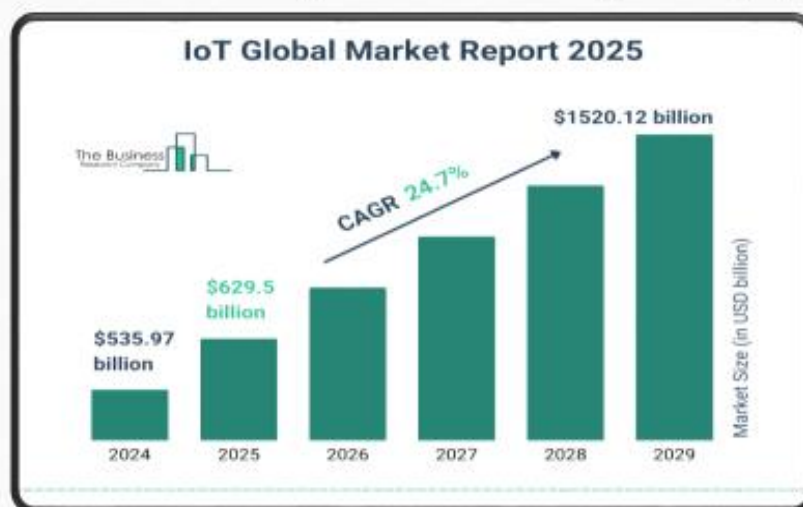
КИЇВ 2025

Мета, Предмет та Об'єкт роботи

- Мета роботи – розробка функціонального макета розумного дистанційного перемикача з використанням мікроконтролера ESP та крокового двигуна.
- Предмет дослідження – способи розробки та реалізації дистанційного натискача кнопок з підтримкою бездротового управління, для розширення можливостей пристроїв без можливості віддаленого керування.
- Об'єкт дослідження – процес автоматизації існуючих приладів і техніки, що не відносяться до сфери IoT.

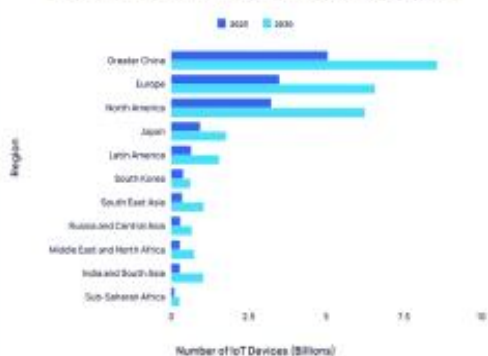


Глобальний ринок інтернету речей

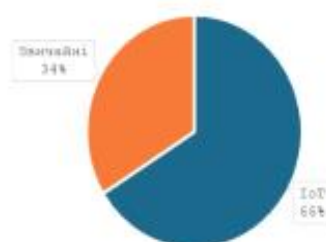


Реалії розповсюдженості девайсів інтернету речей

Number of Active IoT Devices by Region



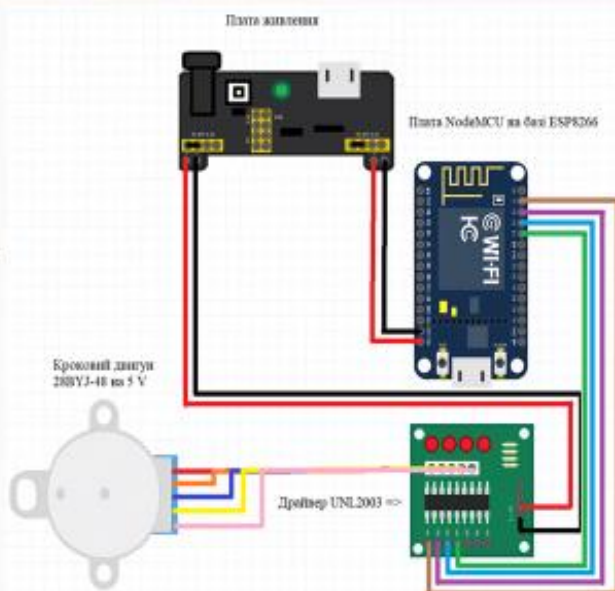
Приблизні підрахунки загальної кількості приладів у світі (2025)



Складові дистанційного перемикача



Схема підключення компонентів



Реалізовані тези

- II ВСЕУКРАЇНСЬКА НАУКОВО-ТЕХНІЧНА КОНФЕРЕНЦІЯ:
- «ТЕХНОЛОГІЧНІ ГОРИЗОНТИ: ДОСЛІДЖЕННЯ ТА ЗАСТОСУВАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ДЛЯ ТЕХНОЛОГІЧНОГО ПРОГРЕСУ УКРАЇНИ І СВІТУ»
- ТЕЗИ «ВИКОРИСТАННЯ ХМАРНИХ ОБЧИСЛЕНЬ У ЦИФРОВІЙ ТРАНСФОРМАЦІЇ ВИРОБНИЦТВА»



Дякую за увагу