

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: Комп'ютеризована система керування паркінгом

Виконав(ла): студент(ка) IV курсу, групи СІ-41
спеціальності 123 «Комп'ютерна інженерія»

(шифр і назва спеціальності)

Пакон О.Д.

(підпис)

(прізвище та ініціали)

Керівник

(підпис)

Жаровський Р.О.

(прізвище та ініціали)

Нормоконтроль

(підпис)

Тиш Є.В.

(прізвище та ініціали)

Завідувач
кафедри

(підпис)

Осухівська Г.М.

(прізвище та ініціали)

Рецензент

(підпис)

Муж В.В.

(прізвище та ініціали)

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних систем та мереж
(повна назва кафедри)

ЗАТВЕРДЖУЮ

Завідувач кафедри

Осухівська Г.М.

(підпис)

(прізвище та ініціали)

« » 2023р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

на здобуття освітнього
ступеня

бакалавр

(назва освітнього ступеня)

за спеціальністю 123 «Комп'ютерна інженерія»

(шифр і назва спеціальності)

студенту

Пакон Олег Дмитрович

(прізвище, ім'я, по батькові)

1. Тема роботи Комп'ютеризована система керування паркінгом

Керівник роботи

Жаровський Руслан Олегович к.т.н., ст. викладач

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

Затверджені наказом ректора від «28» 02 2023 року № 4/7-238

2. Термін подання студентом завершеної роботи 19.06.2023

3. Вихідні дані до роботи Технічне завдання

4. Зміст роботи (перелік питань, які потрібно розробити)

Вступ

1. Аналіз технічного завдання

2. Проектна частина

3. Практична частина

4. Безпеки життєдіяльності, основи охорони праці

Висновок

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

1. Алгоритм роботи телеграм бота

2. Алгоритм роботи функції резервування місця

3. Алгоритм роботи програми мікроконтролера

4. Схема електрона принципова

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
<i>Безпека життєдіяльності, основи охорони праці</i>	<i>Пилипець М.І., д.т.н., професор кафедри МТ</i>		

7. Дата видачі завдання 02.03.2023 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	<i>Розробка та затвердження технічного завдання</i>	<i>29.02.2023–06.03.2023</i>	<i>Виконано</i>
2	<i>Аналіз можливих рішень поставленого завдання</i>	<i>06.03.2023–12.04.2023</i>	<i>Виконано</i>
3	<i>Розробка структурної та функціональної схеми</i>	<i>12.04.2023–18.04.2023</i>	<i>Виконано</i>
4	<i>Розробка схеми електричної принципової, вибір елементної бази</i>	<i>18.04.2023–26.04.2023</i>	<i>Виконано</i>
5	<i>Вибір програмного забезпечення для проектування комп'ютеризованої системи керування паркінгом</i>	<i>26.04.2023–17.05.2023</i>	<i>Виконано</i>
6	<i>Реалізація проєктних рішень</i>	<i>17.05.2023–24.05.2023</i>	<i>Виконано</i>
7	<i>Оформлення пояснювальної записки кваліфікаційної роботи</i>	<i>24.05.2023–02.06.2023</i>	<i>Виконано</i>
8	<i>Оформлення графічної частини</i>	<i>02.06.2023–08.06.2023</i>	<i>Виконано</i>
9	<i>Безпека життєдіяльності, основи охорони праці</i>	<i>08.06.2023–11.06.2023</i>	<i>Виконано</i>
10	<i>Підготовка до захисту</i>	<i>11.05.2023–22.05.2023</i>	<i>Виконано</i>

Студент

(підпис)

Пакон О.Д.

(прізвище та ініціали)

Керівник роботи

(підпис)

Жаровський Р.О.

(прізвище та ініціали)

АНОТАЦІЯ

Комп'ютеризована система керування паркінгом. // Кваліфікаційна робота бакалавра // Пакон Олег Дмитрович// Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра комп'ютерних систем та мереж, група СІ-41 // Тернопіль, 2023 // с. – 71, рис. – 27 , табл. – 0 , кресл. – 4, додат. – 4 , бібліогр. –17 .

Ключові слова: комп'ютеризована система керування паркінгом, ESP 32, інфрачервоний датчик обходу перешкод, мікроконтролер, РК-дисплей.

В кваліфікаційній роботі бакалавра розроблено комп'ютеризовану систему керування паркінгом на базі мікроконтролера ESP32.

Кваліфікаційна робота складається з чотирьох розділів.

Перший розділ присвячений аналізу технічного завдання та перегляду можливих рішень завдання.

Другий розділ описує розробку системи та вибір інструментів для її реалізації.

В третьому розділі приводиться розробка алгоритму програмного забезпечення та реалізація проектних рішень. Також цей розділ включає тестування готового ПЗ.

Четвертий розділ описує безпеку життєдіяльності та основи охорони праці.

ABSTRACT

Computerized parking management system. // Bachelor's qualification work // Oleh Dmytrovych Pakon// Ternopil National Technical University named after Ivan Pulyu, Faculty of Computer and Information Systems and Software Engineering, Department of Computer Systems and Networks, Group CI-41 // Ternopil, 2023 // p. - 71 , fig. - 27 , tab. - 0, sheets A1. - 4 , appendix. – 4, bibliography – 17.

Keywords: computerized parking management system, ESP 32, infrared obstacle avoidance sensor, microcontroller, LCD display.

A computerized parking management system based on an ESP32 microcontroller was developed in the bachelor's qualification work.

The qualification work consists of four sections.

The first section is devoted to the analysis of the technical task and review of possible solutions to the task.

The second section describes the development of the system and the selection of tools for its implementation.

The third section describes the development of the software algorithm and the implementation of project solutions. Also, this section includes testing of finished software.

The fourth chapter describes life safety and the basics of labor protection.

ЗМІСТ

СПИСОК СКОРОЧЕНЬ.....	7
ВСТУП.....	8
РОЗДІЛ 1 АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ.....	10
1.1 Аналіз вимог до комп'ютеризованої системи керування паркінгом.....	10
1.2 Аналіз можливих рішень поставленого завдання.....	11
РОЗДІЛ 2 ПРОЄКТНА ЧАСТИНА.....	15
2.1 Розробка узагальненої структури комп'ютеризованої системи керування паркінгом.....	15
2.2 Обґрунтування вибору апаратного забезпечення для проектування комп'ютеризованої системи керування паркінгом.....	17
2.2.1 Обґрунтування вибору мікроконтролера.....	17
2.2.2 I2C LCD дисплей.....	21
2.2.3 Інфрачервоний датчик обходу перешкод.....	23
2.2.4 Мембрана клавіатура.....	25
2.2.5 Серводвигун.....	27
2.3 Обґрунтування вибору програмного забезпечення для проектування комп'ютеризованої системи керування паркінгом.....	29
2.3.1 Вибір середовища для написання ПЗ для мікроконтролера.....	29
2.3.2 Вибір середовища для написання ПЗ для телеграм бота.....	30
2.4 Проектування комп'ютеризованої системи керування паркінгом.....	32
РОЗДІЛ 3 ПРАКТИЧНА ЧАСТИНА.....	35
3.1 Розробка алгоритму роботи комп'ютеризованої системи керування паркінгом.....	35
3.1.1 Розробка алгоритму для програмного забезпечення мікроконтролера.....	35
3.1.2 Розробка алгоритму для програмного забезпечення телеграм бота.....	36

					КС КРБ 123.051.00.00 ПЗ					
Змн.										
Розроб.	Пакон О.Д..				Комп'ютеризована система керування паркінгом		Літ.		Аркуші	
Перевір.	Жаровський Р.О.						6	68		
Реценз.	Муж В.В.						ТНТУ, каф. КС, гр. СІ-41			
Н. контр.	Тиш Є.В.									
Затверд.	Осухівська Г.М.									

3.2 Реалізація проектних рішень.....	40
3.2.1 Реалізація проектних рішень програмного забезпечення мікроконтролера.....	40
3.2.2 Реалізація проектних рішень програмного забезпечення телеграм бот.....	43
3.3 Тестування.....	47
РОЗДІЛ 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ХОРОНИ ПРАЦІ.....	49
4.1 Долікарська допомога при ушкодженні м'яких тканин, суглобів і кісток.....	49
4.2 Вимоги пожежної безпеки при гасінні електроустановок.....	52
ВИСНОВОК.....	55
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	56
Додаток А Технічне завдання.....	57
Додаток Б Перелік елементів.....	63
Додаток В Алгоритм роботи телеграм бота.....	65
Додаток Ж Лістинг програми.....	67

					КС КРБ 123.051.00.00 ПЗ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

СПИСОК СКОРОЧЕНЬ

КСКП – комп'ютеризована система керування паркінгом.

IR датчик – інфрачервоний датчик.

GPS – система глобального позиціювання.

RIFD – радіочастотна ідентифікація.

ІД-ШКІ – штрих код ІД користувача.

					КС КРБ 123.051.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		8

ВСТУП

Зараз паркінги є необхідною частиною міст та місць відпочинку, що призводить до зростання попиту на них. Однак, недостатність місць на парковках та неефективне їх використання можуть призвести до заторів, збільшення кількості аварій, забруднення навколишнього середовища тощо. Тому потрібна система, яка б дозволила ефективно керувати парковкою та забезпечувати її максимальне використання.

Система управління паркуванням відноситься до інноваційних технологій, що пропонують рішення в галузі паркування. Ця система допомагає людям, компаніям і організаціям керувати своїми місцями для паркування.

Управління автостоянками – непросте завдання для компаній і організацій, оскільки існує багато непередбачуваних частин, включаючи трафік і наявність місць. Використання системи керування паркуванням може допомогти зменшити адміністративні витрати компанії на паркування та оптимізувати використання паркувальних місць на паркінгах.

Програмне забезпечення для паркування використовується в навчальних закладах, муніципалітетах, офісах, підприємствах і корпоративних організаціях.

Завдяки такій системі зникнуть буденні поняття привласнювати конкретне місце під конкретну особу, на заміну до цього поняття приходять гнучкий і зручний спосіб впорядкування автомобілів на паркінгах. Враховуючи темп життя сьогодення, резервувати певне місце під конкретного користувача на постійній основі є недоречним і не вигідним, оскільки під час одного циклу користування, таке місце може обслужити в пару разів більше користувачів, замість того щоб воно простоювало і чекало свого власника [1].

Система керування паркінгом складається із двох частин:

- телеграм бот;
- система створена на базі ESP32.

					КС КРБ 123.051.00.00 ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 1 АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ

1.1 Аналіз вимог до комп'ютеризованої системи керування паркінгом

Система керування паркінгом має включати в себе такі основні функції:

- робота в реальному часі;
- комунікація із користувачем через LCD екран;
- реєстрація та контроль в'їзду/виїзду автомобілів;
- бути енергоефективною.

Також окремо можна виділити вимоги до комунікаційної частини системи:

- можливість користувачем зарезервувати місце;
- повідомлення користувачів про стан парковки та вільні місця;
- моніторинг вільних місць на парковці.

Додатково, система повинна мати інтерфейс для адміністраторів, що дозволить виконувати наступні функції:

- керування списком користувачів та їх правами доступу до системи;
- перегляд та аналіз статистичної інформації про роботу парковки;
- налаштування параметрів системи;
- реагування на виниклі проблеми та забезпечення технічної підтримки користувачів.

КСКПІ повинна відповідати заданим вимогам технічного завдання, яке показане в Додатку А.

					КС КРБ 123.051.00.00 ПЗ		
Змн.							
Розроб.	Пакон О.Д.				Комп'ютеризована система керування паркінгом	Літ.	Аркушів
Перевір.	Жаровський Р.О.					6	68
Реценз.	Муж В.В.					ТНТУ, каф. КС, гр. СІ-41	
Н. контр.	Тиш Є.В.						
Затверд.	Осухівська Г.М.						

1.2 Аналіз можливих рішень поставленого завдання

Зважаючи на те, що технології керування паркінгом зародилися ще в 1920–х роках, сьогодні на ринку можна знайти широкий вибір різноманітних систем, які базуються на різних принципах. При цьому відомі і такі рішення, які ще не втілилися в життя, проте на їх базі можливо будувати нові системи керування паркінгом.

Серед існуючих рішень можна виділити системи з використанням датчиків, які визначають наявність вільних паркувальних місць на паркінгу, або з системами, які використовують технології розпізнавання номерних знаків автомобілів для ідентифікації транспортних засобів. Деякі системи базуються на технологіях GPS і радіочастотного ідентифікаційного (RFID) зчитування.

Деякі сучасні системи використовують мережу камер для відеоспостереження та обробки даних, що дозволяє точніше визначати наявність вільних місць та реєструвати в'їзд і виїзд транспортних засобів.

Окрім цього, на ринку присутні і такі інноваційні рішення, як системи автоматизованого паркування, які дозволяють автоматично паркувати автомобілі без участі водія.

Компанії, які пропонують рішення в області керування паркінгом, використовують різні технології та підходи. Ось декілька з них:

- ParkHelp: ця компанія пропонує рішення для керування паркінгом на основі камер відеоспостереження, які дозволяють відстежувати рух автомобілів на парковці та надсилати інформацію про вільні місця на спеціальний монітор водієві.
- Parkeon: ця компанія пропонує рішення, яке використовує сенсори для визначення наявності вільних місць для паркування та механізми оплати за паркування через мобільний додаток.
- Amano McGann: ця компанія пропонує рішення для керування паркінгом, яке включає в себе систему визначення наявності вільних місць, механізми оплати за паркування та засоби контролю доступу на парковці.

					КС КРБ 123.051.00.00 ПЗ	Арк.
						11
Змн.	Арк.	№ докум.	Підпис	Дата		

ParkHelp – це компанія, що спеціалізується на розробці технологій для керування паркінгом. Їхня система базується на використанні камер відеоспостереження та аналізу зображення, що дозволяє відстежувати рух автомобілів на парковці та надсилати інформацію про вільні місця на спеціальний монітор водієві.

Основні переваги системи ParkHelp полягають у високій точності визначення наявності вільних місць на парковці, можливості аналізу даних щодо користування парковкою та гнучкості в налаштуванні рішення під конкретні потреби клієнта.

Загалом, система ParkHelp є досить комплексною та потужною системою керування паркінгом, яка дозволяє оптимізувати використання ресурсів та поліпшити якість обслуговування користувачів парковки [2].

Parkeon – це ще одна компанія, яка пропонує рішення для керування паркінгом. Вона займається розробкою та виробництвом паркоматів, систем автоматизованого керування парковкою та різноманітних мобільних додатків для водіїв.

Основні переваги системи Parkeon полягають у високій надійності та функціональності їхніх продуктів, що дозволяє вирішувати широкий спектр завдань, пов'язаних з керуванням паркінгом.

Крім того, Parkeon відома своєю високою якістю обслуговування та підтримки клієнтів.

Система Parkeon є дуже потужною та комплексною системою керування паркінгом, яка дозволяє забезпечувати ефективну та зручну обслуговування користувачів парковок, а також контролювати використання місць [3].

Amano McGann – це провідна компанія в галузі паркувальних систем. Компанія спеціалізується на розробці та виробництві систем керування парковкою, автоматичних паркоматів, систем доступу та безпеки.

Amano McGann має довгий і успішний досвід у галузі паркувальних систем. Компанія була заснована в 1969 році в США і з тих пір вона розширилася на багато країн світу.

					КС КРБ 123.051.00.00 ПЗ	Арк.
						12
Змн.	Арк.	№ докум.	Підпис	Дата		

Основні переваги системи Amano McGann полягають у високій надійності та безпеці продуктів, які дозволяють вирішувати широкий спектр завдань, пов'язаних з керуванням паркінгом [4].

Рішення цих трьох компаній мають схожі компоненти, а саме:

- паркомати: це автоматичні машини, які дозволяють водіям оплачувати паркування, використовуючи монети, кредитні картки або смартфони.
- системи автоматизованого керування парковкою: вони включають в себе різноманітні датчики, камери та програмне забезпечення для відстеження руху автомобілів на парковці та контролю доступу.
- мобільні додатки: додатки дозволяють водіям бронювати місця на парковці, оплачувати паркування, отримувати повідомлення про статус їхніх бронювань та інше.
- аналітика: система збирає та аналізує дані про користування парковкою, що дозволяє управляти ресурсами більш ефективно та приймати обґрунтовані рішення.

При виборі системи керування паркінгом слід ретельно проаналізувати всі існуючі рішення та вибрати те, яке найбільш підходить для поставленого завдання з урахуванням бюджету та вимог до системи. Важливо враховувати переваги та недоліки кожної системи, їх вартість та можливості у майбутньому розширювати функціональні можливості системи.

В результаті аналізу можливих рішень для системи керування паркінгом були визначені декілька можливих рішень для реалізації системи. Комбінація використання датчиків для фіксування в'їзду або виїзду автомобіля із паркінгу та контролю і менеджменту місць на паркінгу дозволить зменшити ціну системи, відчутно не втративши ефективність та надійність. Також досить важливим рішенням в сучасній КСКП є використання мобільних додатків для взаємодії користувача із системою, щоб найбільш ефективно використовувати як час клієнтів так і ресурси паркінгу.

Це поєднання рішень дозволить розробити систему в реальному часі, котра зможе комунікувати із користувачами та ефективно розпоряджатися простором паркінгу.

					КС КРБ 123.051.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		13

Таким чином мінімізується часу пошуку вільних місць для водіїв та відбувається постійна циркуляція авто на паркінгах, тим самим зменшуючи затори.

Завдяки аналізу різних можливих рішень було зроблено обґрунтований вибір технології та підходу до реалізації комп'ютеризованої системи керування паркінгом. Дане рішення буде в основі подальшої розробки та впровадження системи керування паркінгом. На основі проведеного аналізу можливих рішень будуть визначені конкретні кроки щодо проектування архітектури системи, вибору необхідного обладнання та програмного забезпечення, розробки алгоритмів обробки даних, а також тестування та налагодження системи.

Крім того, передбачається проведення експериментального впровадження системи на обмеженому масштабі для перевірки її ефективності та коректності роботи. Збирання фідбеку та впровадження внесених коректив допоможе удосконалити систему перед її повним впровадженням та використанням на реальних паркінгах.

					КС КРБ 123.051.00.00 ПЗ	Арк.
						14
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 2 ПРОЄКТНА ЧАСТИНА

2.1 Розробка узагальненої структури комп'ютеризованої системи керування паркінгом

Щоб краще зрозуміти роботу КСКП розглянемо функціональну схему, яка зображена на рис. 2.1.



Рисунок 2.1 – Функціональна схема комп'ютеризованої системи керування паркінгом

					КС КРБ 123.051.00.00 ПЗ				
Змн.					РОЗДІЛ 2 ПРОЄКТНА ЧАСТИНА				
Розроб.	Пакон О.Д.			Літ.					Акрушіє
Перевір.	Жаровський Р.О.							15	20
Реценз.	Муж В.В.			ТНТУ, каф. КС, гр. СІ-41					
Н. контр.	Тиш Є.В.								
Затверд.	Осухівська Г.М.								

Із наведеної вище функціональної схеми, можемо зрозуміти, що система основана на мікроконтролеру ESP 32, отримавши інформацію із клавіатури та датчиків, він надсилає запит до сервера і в залежності від відповіді виводить певну інформацію на екран та змінює положення серводвигуна.

Детальніше з роботою керуючого модуля системи можна за допомогою структурної схеми, що зображена на рис. 2.2.

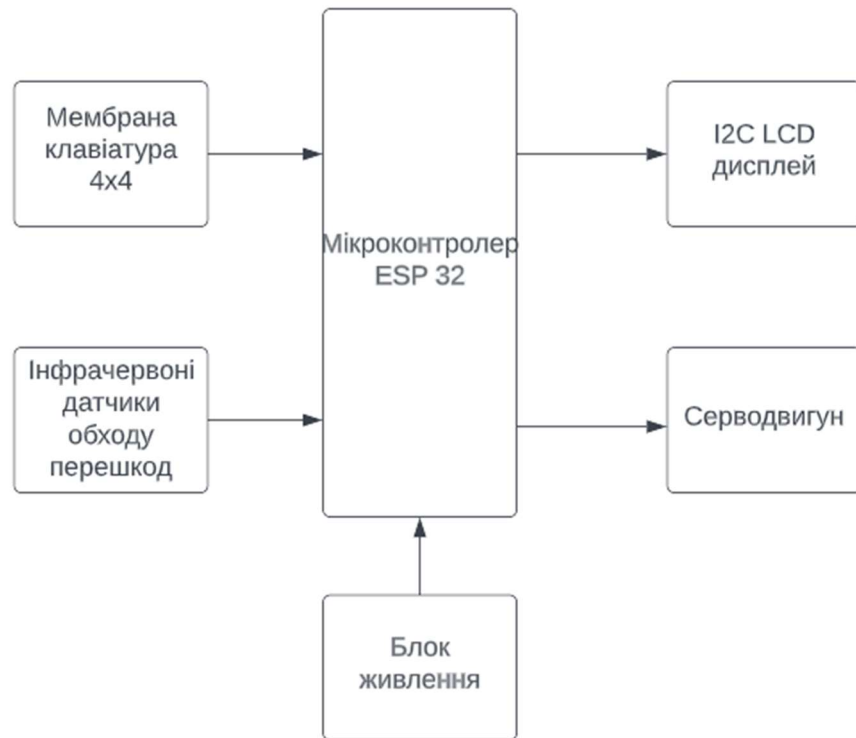


Рисунок 2.2 – Структурна схема комп’ютеризованої системи керування паркінгом

КСКП складається із наступних елементів

- мікроконтролер ESP 32;
- мембранна клавіатура 4x4;
- інфрачервоний датчик обходу перешкод;
- I2C LCD дисплей 16x2;
- сервопривід.

2.2 Обґрунтування вибору апаратного забезпечення для проектування комп'ютеризованої системи керування паркінгом

2.2.1 Обґрунтування вибору мікроконтролера

КСКП основана на базі мікроконтролера ESP32, а саме на розробній платі ESP32-DevKitC V4.

ESP32 – це мікроконтролер, розроблений компанією Espressif Systems, який поєднує в собі високу продуктивність, низьке споживання енергії та багатий набір функцій. Він є наступником популярного ESP8266 і є доступним у варіантах з одним або двома ядрами.

Основою ESP32 є 32-бітний мікропроцесор Tensilica Xtensa LX6 з тактовою частотою до 240 МГц. Цей мікропроцесор забезпечує високу продуктивність і швидкодію виконання програм. ESP32 також має велику кількість вбудованої пам'яті, включаючи 520 КБ SRAM для зберігання даних та 448 КБ ROM для програмного коду (рис. 2.3).

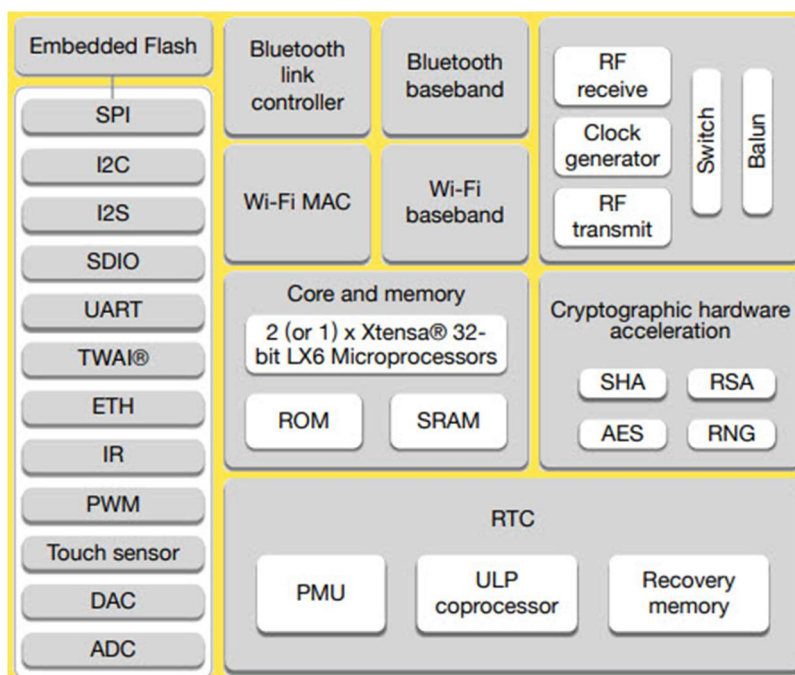


Рисунок 2.3 – Структурна схема будови ESP32

Однією з ключових особливостей ESP32 є його вбудована підтримка Wi-Fi і Bluetooth. Це дозволяє підключати мікроконтролер до бездротових мереж і взаємодіяти з іншими пристроями через Bluetooth зв'язок. Завдяки цьому ESP32 може бути використаний у різних застосуваннях, які потребують зв'язку в мережі або з іншими пристроями.

ESP32 має вбудовані GPIO (загального призначення введення–виведення), які дозволяють підключати різні зовнішні пристрої і сенсори. Кількість доступних GPIO пінів залежить від конкретної моделі ESP32, але вона зазвичай становить близько 34 піни. Це надає гнучкість при розробці проектів і дозволяє підключати різні периферійні пристрої.

ESP32 також має вбудовані аналогово–цифрові перетворювачі (АЦП) і цифро–аналогові перетворювачі (ЦАП). Це дозволяє зчитувати аналогові сигнали з датчиків і керувати аналоговими пристроями. Кількість доступних АЦП і ЦАП залежить від конкретної моделі ESP32, але зазвичай їх кількість становить до 18 каналів для АЦП і 2 канали для ЦАП.

Окрім того, ESP32 підтримує послідовне підключення до різних пристроїв, таких як SPI, I2C та UART. Це дозволяє зв'язувати ESP32 з різними пристроями через цифрові інтерфейси і обмінюватися даними з ними.

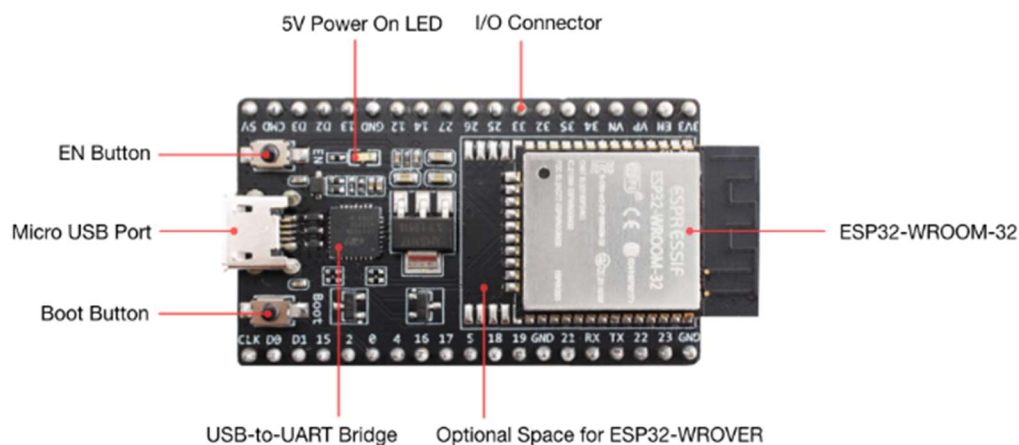
Щоб розробляти програми для ESP32, існує кілька популярних середовищ програмування. Наприклад, Arduino IDE – це інтегроване середовище розробки зі зрозумілою мовою програмування Arduino та багатьма готовими бібліотеками. Є також PlatformIO IDE, яке підтримує ESP32 та інші мікроконтролери і дозволяє програмувати їх за допомогою мови C/C++. Для легкого програмування є підтримка LUA та MicroPython.

ESP32 є потужним мікроконтролером з великим набором функцій, що дозволяє розробникам створювати різноманітні проекти, які вимагають зв'язку в мережі, низького споживання енергії та великої гнучкості. Його можна використовувати у сферах Інтернету речей (IoT), автоматизації, робототехніки та багатьох інших [5].

					КС КРБ 123.051.00.00 ПЗ	Арк.
						18
Змн.	Арк.	№ докум.	Підпис	Дата		

ESP32-DevKitC V4 є компактною розробною платою, що базується на мікроконтролері ESP32 та виготовлена компанією Espressif. Більшість вводу/виводу контактів розташовані по обидва боки плати, щоб забезпечити зручну взаємодію. Розробники можуть підключати зовнішні пристрої за допомогою перемичок або закріпити ESP32-DevKitC V4 на макетній платі.

На рис. 2.4 зображено ESP32-DevKitC V4 та описано ключові компоненти.



Компонент	Опис
ESP32-WROOM-32	Модуль з основою ESP32.
EN	Кнопка скидання.
Boot	Кнопка завантаження. Утримуючи натиснутою кнопку Boot, а потім натиснувши EN, запускається режим завантаження мікропрограми для завантаження мікропрограми через послідовний порт.
USB-to-UART Bridge	Чіп моста USB-UART забезпечує швидкість передачі даних до 3 Мбіт/с.
Micro USB Port	Інтерфейс USB. Блок живлення плати, а також інтерфейс зв'язку між комп'ютером і модулем ESP32-WROOM-32.
5V Power On LED	Вмикається при підключенні до плати USB або зовнішнього джерела живлення 5В.
I/O	Більшість контактів на модулі ESP розбиті на контактні роз'єми на платі. Можуть бути запрограмовані ESP32 для активації кількох функцій, таких як ШІМ, АЦП, ЦАП, I2C, I2S, SPI тощо.

Рисунок 2.4 – Зображення мікроконтролера та опис ключових компонентів.

Розглянемо детальніше контакти мікроконтролера, на рис. 2.5 та 2.6 зображено назви та функції контактів введення/виведення з обох боків плати.

No.	Name	Type ¹	Function
1	3V3	P	3.3 V power supply
2	EN	I	CHIP_PU, Reset
3	VP	I	GPIO36, ADC1_CH0, S_VP
4	VN	I	GPIO39, ADC1_CH3, S_VN
5	IO34	I	GPIO34, ADC1_CH6, VDET_1
6	IO35	I	GPIO35, ADC1_CH7, VDET_2
7	IO32	I/O	GPIO32, ADC1_CH4, TOUCH_CH9, XTAL_32K_P
8	IO33	I/O	GPIO33, ADC1_CH5, TOUCH_CH8, XTAL_32K_N
9	IO25	I/O	GPIO25, ADC1_CH8, DAC_1
10	IO26	I/O	GPIO26, ADC2_CH9, DAC_2
11	IO27	I/O	GPIO27, ADC2_CH7, TOUCH_CH7
12	IO14	I/O	GPIO14, ADC2_CH6, TOUCH_CH6, MTMS
13	IO12	I/O	GPIO12, ADC2_CH5, TOUCH_CH5, MTDI
14	GND	G	Ground
15	IO13	I/O	GPIO13, ADC2_CH4, TOUCH_CH4, MTCK
16	D2	I/O	GPIO9, D2 ²
17	D3	I/O	GPIO10, D3 ²
18	CMD	I/O	GPIO11, CMD ²
19	5V	P	5 V power supply

Рисунок 2.5 – Перелік контактів мікроконтролера та їхні функції

No.	Name	Type ¹	Function
1	GND	G	Ground
2	IO23	I/O	GPIO23
3	IO22	I/O	GPIO22
4	TX	I/O	GPIO1, U0TXD
5	RX	I/O	GPIO3, U0RXD
6	IO21	I/O	GPIO21
7	GND	G	Ground
8	IO19	I/O	GPIO19
9	IO18	I/O	GPIO18
10	IO5	I/O	GPIO5
11	IO17	I/O	GPIO17 ³
12	IO16	I/O	GPIO16 ³
13	IO4	I/O	GPIO4, ADC2_CH0, TOUCH_CH0
14	IO0	I/O	GPIO0, ADC2_CH1, TOUCH_CH1, Boot
15	IO2	I/O	GPIO2, ADC2_CH2, TOUCH_CH2
16	IO15	I/O	GPIO15, ADC2_CH3, TOUCH_CH3, MTDO
17	D1	I/O	GPIO8, D1 ²
18	D0	I/O	GPIO7, D0 ²
19	CLK	I/O	GPIO6, CLK ²

Рисунок 2.6 – Перелік контактів мікроконтролера та їхні функції

Звернувши увагу на тип контакту, можна зрозуміти для чого він призначений, а саме:

- P – живлення;
- I/O – контакти введення або виведення.

Також варто звернути увагу, що в ESP32 існує внутрішнє з'єднання між флеш–пам'яттю SPI та ESP32, яке здійснюється через виводи D0, D1, D2, D3, CMD і CLK. Ці виводи знаходяться біля роз'єму USB. Рекомендується уникати використання цих контактів, оскільки це може призвести до проблем з доступом до флеш–пам'яті SPI/RAM SPI [6].

2.2.2 I2C LCD дисплей

Рідкокристалічні дисплеї (РК) чудово підходять для відображення рядка слів або даних датчиків. Як видно з назви, символні РК–дисплеї ідеально підходять для відображення символів. Наприклад, дисплей розміром 16×2 символи, який зображений на рис. 2.7, може вмістити 32 символи ASCII у двох рядках. При ретельному розгляді можна побачити маленькі прямокутники для кожного символу на екрані, складені з пікселів. Кожен з цих прямокутників складається з сітки розміром 5×8 пікселів.

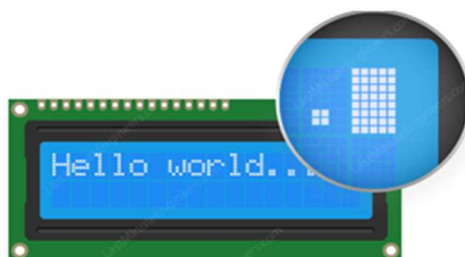


Рисунок 2.7 – Спосіб зображення символів на РК–дисплеї

Символьні РК–дисплеї доступні у різних розмірах і кольорах, включаючи 16×1 , 16×4 , 20×4 , білий текст на синьому фоні, чорний текст на зеленому фоні та багато інших варіацій.

Однією з переваг використання цих дисплеїв у проекті є їхня "замінність". Можна легко замінити один дисплей іншим розміром або кольором. Це принесе невеликі зміни у коді, але підключення залишиться незмінним [7].

I2C РК–дисплей оперує лише двома ввідно–вивідними контактами, які не входять до набору цифрових ввідно–вивідних контактів і можуть використовуватися одночасно з іншими пристроями, сумісними з I2C.

					КС КРБ 123.051.00.00 ПЗ	Арк.
						21
Змн.	Арк.	№ докум.	Підпис	Дата		

Адаптер базується на 8-розрядному мікросхемі розширення вводу/виводу з назвою PCF8574. Ця мікросхема перетворює I2C дані з Arduino на паралельні дані, необхідні для роботи з РК-дисплеєм. На платі також присутній невеликий потенціометр для точного налаштування контрастності дисплея. Крім того, на платі є перемичка, яка забезпечує живлення підсвічування. Якщо потрібно контролювати інтенсивність підсвічування, можна зняти перемичку і підключити зовнішню напругу до роз'єму, позначеного як "LED" (рис. 2.8) [8].

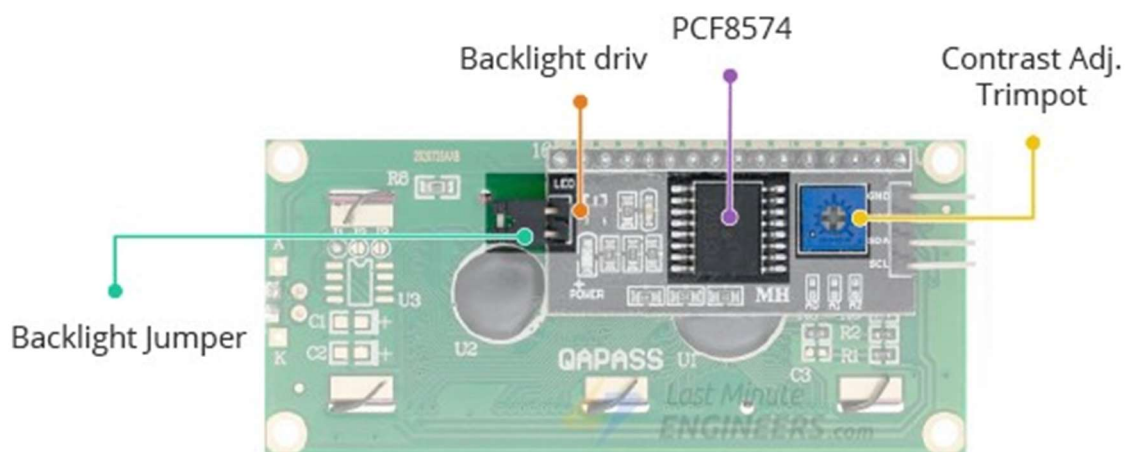


Рисунок 2.8 – Зображення адаптера

Тепер розглянемо перелік контактів дисплею рис. 2.9.

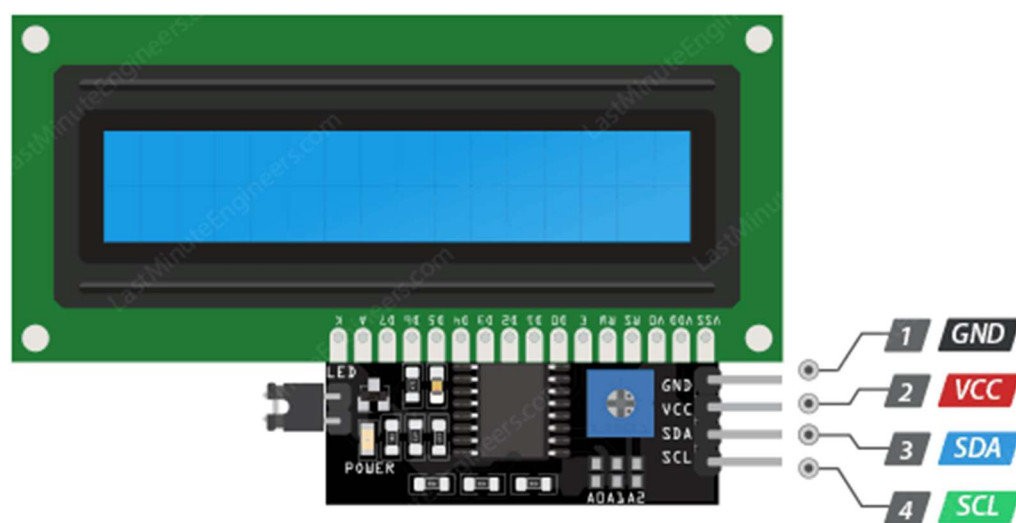


Рисунок 2.9 – Контакти підключення дисплею

РК–дисплей можна підключити за допомогою чотирьох контактів, а саме :

- контакту заземлення GND;
- контакту живлення VCC;
- контакту даних SDA;
- контакту синхронізації SCL.

2.2.3 Інфрачервоний датчик обходу перешкод

Інфрачервоний датчик обходу перешкод є пристроєм, який використовує інфрачервоне випромінювання для виявлення перешкод у своєму навколишньому середовищі. Цей тип датчика знаходить широке застосування в різних галузях, включаючи робототехніку та системи безпеки, де важливо виявляти перешкоди на шляху руху.

Принцип роботи інфрачервоного датчика обходу перешкод полягає у відбиванні інфрачервоного випромінювання від перешкоди та його отриманні за допомогою фотодіода або фототранзистора. Датчик випромінює інфрачервоний сигнал в напрямку перешкоди і очікує його відбиття. Якщо сигнал повертається, це свідчить про наявність перешкоди у зоні виявлення (рис. 2.10).

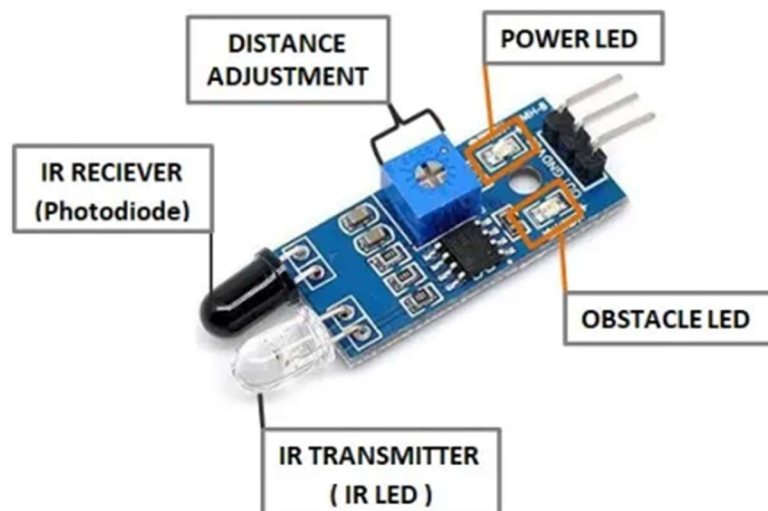


Рисунок 2.10 – Зовнішній вигляд датчика та призначення його компонентів

Зазвичай, інфрачервоні датчики обходу перешкод складаються з передавача і приймача. Передавач випромінює інфрачервоне випромінювання, а приймач отримує його після його відбиття від перешкоди. При отриманні сигналу, приймач активує певні дії, такі як зупинка руху, зміна напрямку або спрацювання сигналу тривоги.

Інфрачервоні датчики обходу перешкод мають кілька переваг, таких як працездатність в широкому діапазоні освітленості та здатність виявляти різні типи перешкод. Вони знадобляться для виявлення стін, стовпів, меблів та інших перешкод. Деякі моделі можуть навіть виявляти перешкоди з різних напрямків та визначати їхні розміри та відстані.

Конфігурація контактів для підключення інфрачервоного датчика обходу перешкод може варіюватися залежно від конкретної моделі датчика та системи, до якої він підключається. Проте, зазвичай датчик має кілька основних контактів, які використовуються для забезпечення комунікації з системою. Ці контакти можуть включати живлення (VCC) для подачі електропостачання, землю (GND) для заземлення, вихідний сигнал (Output), який передає інформацію про виявлення перешкоди, а також можуть бути присутні додаткові контакти для налаштування параметрів датчика (рис. 2.11) [11].

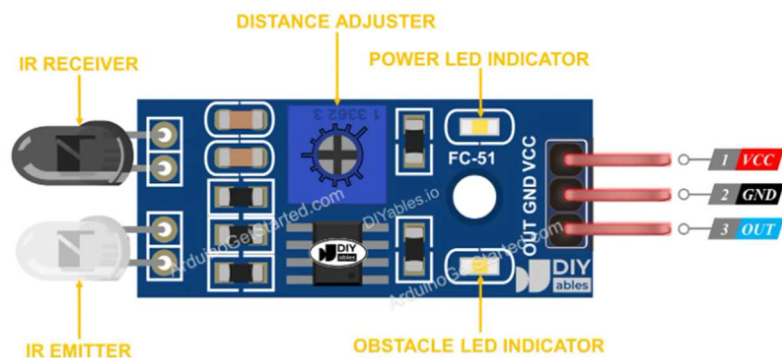


Рисунок 2.11 – Контакти підключення інфрачервоного датчик обходу перешкод

					КС КРБ 123.051.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		24

Загалом, інфрачервоний датчик обходу перешкод є корисним пристроєм для виявлення фізичних перешкод у навколишньому середовищі. Він знаходить широке застосування у робототехніці, системах безпеки та інших галузях, де важливо уникнути зіткнень з перешкодами та забезпечити безпечний рух.

2.2.4 Мембрана клавіатура

Мембранні клавіатури доступні у різних розмірах, і найпоширенішими серед них є клавіатура з розміром 4×3 (12 клавiш) і клавіатура з розміром 4×4 (16 клавiш). Вони мають розкладку, що схожа на стандартну телефонну клавіатуру, що робить їх легкими у використанні для будь-якого користувача (рис. 2.12)..

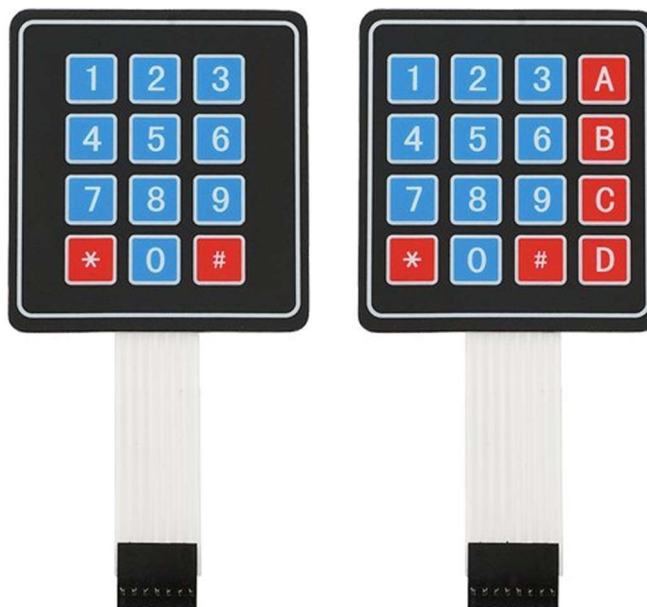


Рисунок 2.12 – Типи мембранних клавіатур

Мембранні клавіатури виготовляються з тонкого та гнучкого матеріалу, і зазвичай складаються з шести шарів, які мають різні функції:

- графічний шар: Цей шар містить графічні накладки, які часто виготовляються з поліестеру, оскільки він має довгий термін експлуатації порівняно з полікарбонатом.
- металеві куполи: Цей шар включає металеві куполи або полікуполи, які надають тактильний відгук при натисканні клавiш.

- верхній шар схеми: Зазвичай це поліефірний друкований шар, на якому знаходяться електропровідні чорнила, що містять срібло. Цей шар має гнучкий хвіст, який з'єднує клавіатуру з зовнішнім світом.
- прокладка: Цей шар розділяє верхній і нижній контакти, дозволяючи перемикачу залишатися відкритим, поки не натиснута клавіша.
- нижній шар ланцюга: Зазвичай це поліефірний друкований шар, що містить електропровідні фарби з додаванням срібла. Цей шар також має гнучкий хвіст.
- задній клейкий шар: Цей шар використовується для приклеювання клавіатури практично до будь-якої підкладки або поверхні.

Кожен з цих шарів виконує свою функцію і допомагає забезпечити правильну роботу мембранної клавіатури (рис. 2.13).

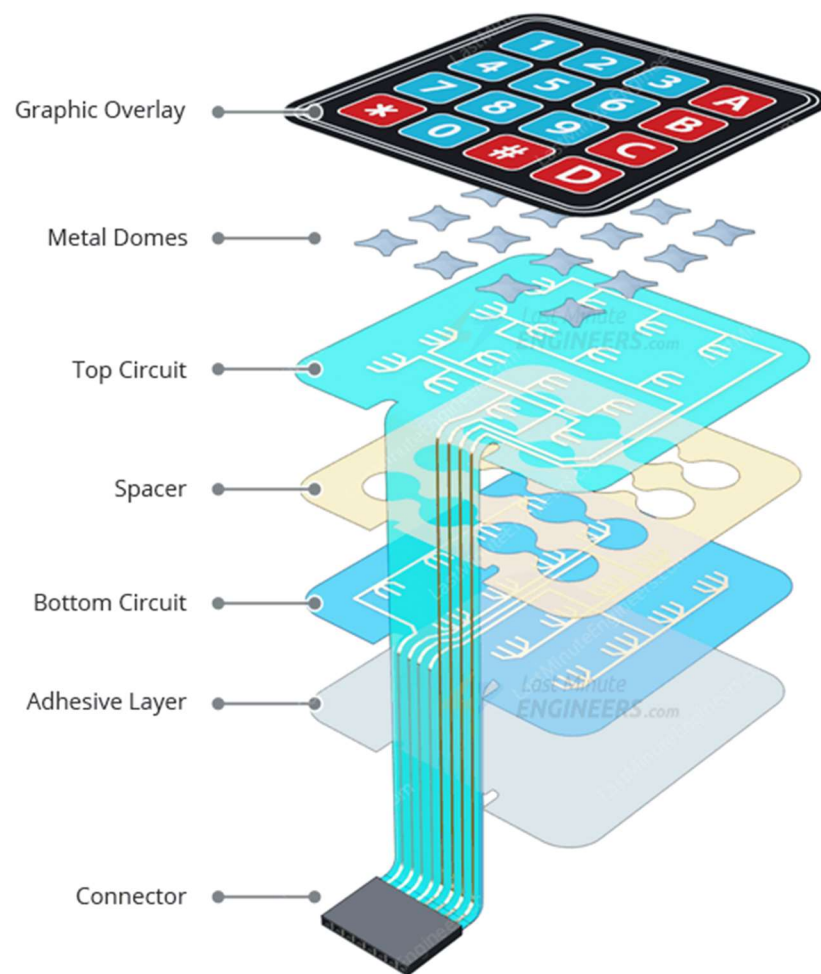


Рисунок 2.13 – Складові шари мембранної клавіатури

Кожен рядок мембранної клавіатури підключений до вхідного контакту, тоді як кожен стовпець підключений до вихідного контакту.

Вхідні контакти забезпечуються напругою HIGH за допомогою внутрішніх навантажувальних резисторів. Після цього мікроконтролер послідовно встановлює LOW на виводі кожного стовпця і перевіряє, чи є виводи рядків у стані LOW. Завдяки використанню підтягуючих резисторів, рядки залишаються HIGH, якщо жодна кнопка не натиснута [9].

Якщо вивід рядка переходить у стан LOW, це означає, що кнопка на цьому рядку та стовпці була натиснута. Після цього мікроконтролер очікує звільнення кнопки. Потім він шукає в масиві розкладок клавіш символ, який відповідає натиснутій кнопці (рис. 2.14).

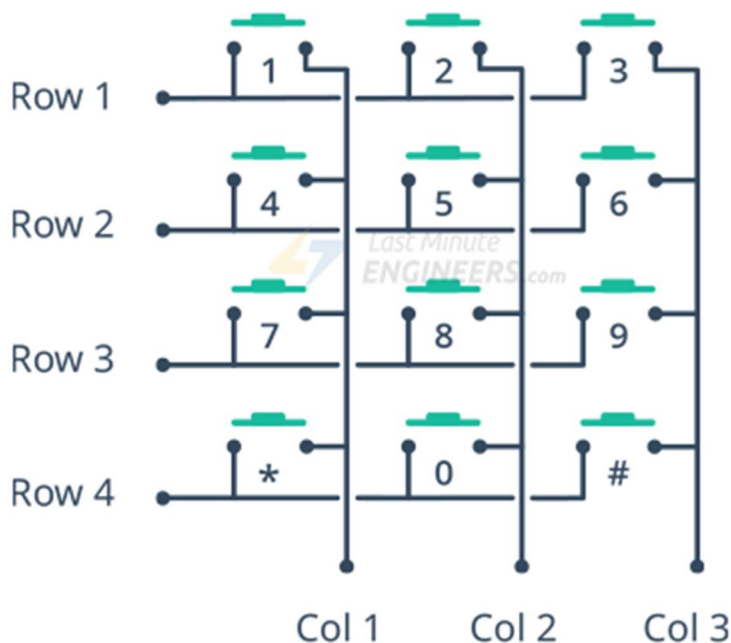


Рисунок 2.14 – Схема роботи мембранної клавіатури

2.2.5 Серводвигун

Сервоприводи представляють собою тип двигунів, які дозволяють точно контролювати фізичні рухи, оскільки їх рух зазвичай обмежується певним положенням, а не безперервним обертанням. Вони легкі у підключенні та керуванні, оскільки мають вбудований драйвер двигуна.

2.3 Обґрунтування вибору програмного забезпечення для проєктування комп'ютеризованої системи керування паркінгом

2.3.1 Вибір середовища для написання ПЗ для мікроконтролера

Написання програмного забезпечення буде відбуватись за допомогою середовища Arduino IDE. Це інтегроване середовище розробки, призначене для програмування платформи Arduino. Це програмне забезпечення надає зручний і простий у використанні інтерфейс для написання, завантаження і відлагодження програм для плат Arduino. Arduino IDE має багато корисних функцій, що полегшують розробку проєктів. Воно надає можливість вибрати правильну платформу Arduino та встановити необхідні бібліотеки для роботи з різними пристроями і компонентами. Також в ньому є редактор коду з підсвічуванням синтаксису, автодоповненням і можливістю перевірки синтаксичних помилок.

Мова програмування, яку використовують в Arduino IDE, називається мовою Arduino. Ця мова є спеціалізованою версією мови C++, адаптованою для роботи з платформою Arduino. Мова Arduino має свої особливості та бібліотеки, які спрощують програмування для взаємодії з різними компонентами та модулями, що підтримуються платформою Arduino.

Використовуючи мову Arduino, можна створювати програми, які керують різними функціями та пристроями, такими як світлодіоди, сенсори, дисплеї, мотори та багато інших. Мова Arduino проста у вивченні та використанні, особливо для початківців у програмуванні, оскільки має зрозумілу синтаксичну структуру та велику кількість прикладів та документації.

Окрім мови Arduino, в Arduino IDE також можна використовувати мови програмування C та C++. Це дає можливість розробникам з великим досвідом в цих мовах використовувати їх повну потужність при розробці проєктів на платформі Arduino [16].

Узагальнюючи, мова програмування в Arduino IDE – це мова Arduino, яка базується на мові C++ та має свої особливості та бібліотеки для спрощення розробки проєктів на платформі Arduino.

					КС КРБ 123.051.00.00 ПЗ	Арк.
						29
Змн.	Арк.	№ докум.	Підпис	Дата		

Після написання коду в Arduino IDE можна завантажити програму безпосередньо на плату Arduino за допомогою USB-підключення. Інтегрована функція відлагодження дозволяє контролювати виконання програми та перевіряти значення змінних в режимі реального часу.

Крім того, Arduino IDE підтримує велику спільноту користувачів, що дозволяє легко знайти документацію, приклади коду та отримати допомогу вирішення проблем. Це робить його популярним інструментом для початківців і досвідчених розробників, які працюють з платформою Arduino.

На рис. 2.16 зображено зовнішній вигляд написання коду в Arduino IDE.

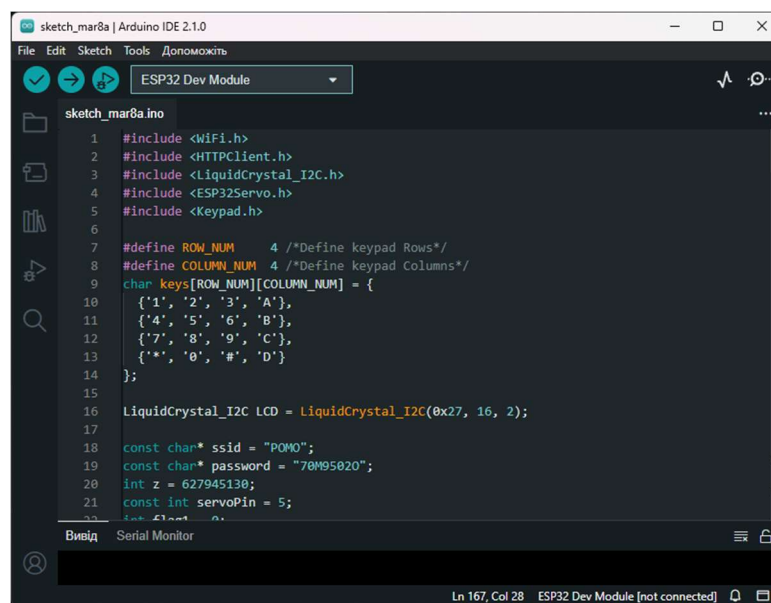


Рисунок 2.16 – Зовнішній вигляд документу в Arduino IDE

2.3.2 Вибір середовища для написання ПЗ для телеграм бота

Оскільки дані користувачів будуть зберігатися в Google sheets, то як середовище для написання коду бота буде використовуватись Apps script.

Google Apps Script – це середовище розробки, що дозволяє створювати скрипти та розширювати функціональність різних продуктів Google, таких як Google Sheets, Google Docs, Google Slides та інших. Він базується на JavaScript, але має спеціальні бібліотеки та об'єкти, які спрощують доступ до функцій та даних Google.

					КС КРБ 123.051.00.00 ПЗ	Арк.
						30
Змн.	Арк.	№ докум.	Підпис	Дата		

За допомогою цього середовища можна автоматизувати рутинні завдання, створювати власні функції та додаткові можливості для роботи з документами Google. Це дозволяє створювати зв'язки між різними продуктами Google, реалізовувати користувацькі інтерфейси, отримувати та обробляти дані з зовнішніх джерел, надавати додаткові можливості для спільної роботи та багато іншого.

Apps Script працює в хмарі Google, тому скрипти запускаються на серверах Google, а результати можуть бути доступні для користувачів з будь-якого пристрою з підключенням до Інтернету. Його можна використовувати як для особистих потреб, так і для розробки додатків для командної роботи або широкомасштабних бізнес-процесів.

Крім того, Google Apps Script надає можливість створювати спеціалізовані веб-додатки, що працюють в середовищі Google. За допомогою HTML, CSS та JavaScript можна створювати власні користувацькі інтерфейси, забезпечувати взаємодію з користувачем та зберігати дані на серверах Google. Такі додатки можуть бути розповсюджені серед обмеженого кола користувачів або публічно опубліковані для загального доступу.

Apps Script також підтримує інтеграцію з іншими сервісами та API, що дозволяє обмінюватися даними зовнішніми сервісами та створювати рішення, які поєднують різні платформи. За допомогою стандартних бібліотек та сторонніх пакетів можна легко взаємодіяти з базами даних, веб-сервісами, електронною поштою та багатьма іншими сервісами [15].

Загалом, Google Apps Script є потужним інструментом для автоматизації та розширення функціональності сервісів Google. Він дозволяє розробникам створювати різноманітні скрипти, додатки та інтеграції, що полегшують роботу з даними, автоматизують процеси та забезпечують персоналізовані рішення для користувачів. З його допомогою можна створювати потужні інструменти для роботи з документами, аналізу даних, спільної роботи та багато іншого.

					КС КРБ 123.051.00.00 ПЗ	Арк.
						31
Змн.	Арк.	№ докум.	Підпис	Дата		

2.4 Проектування комп'ютеризованої системи керування паркінгом

В основі схеми лежить плата ESP32 (рис.2.17), котра керує всіма модулями системи.

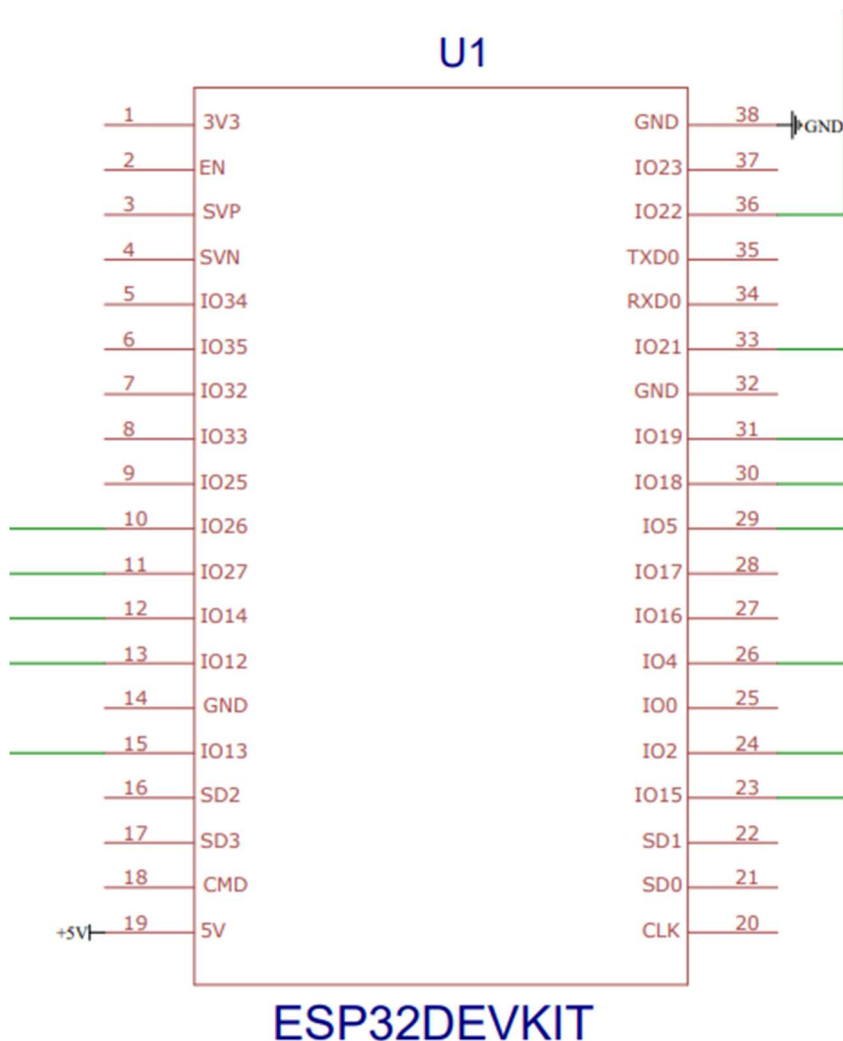


Рисунок 2.17 – Електрична принципова схема плати

На наступному рис. 2.18 зображено спосіб підключення до мікроконтролера серводвигуна та двох інфрачервоних датчиків обходу перешкоди. Всі три пристрої однакові за способом підключення, контакти V_{cc} під'єднані до контакту живлення на мікроконтролері, контакти GND під'єднані до землі, контакти даних під'єднані до 19, 18 і 5 контактів плати ESP32.

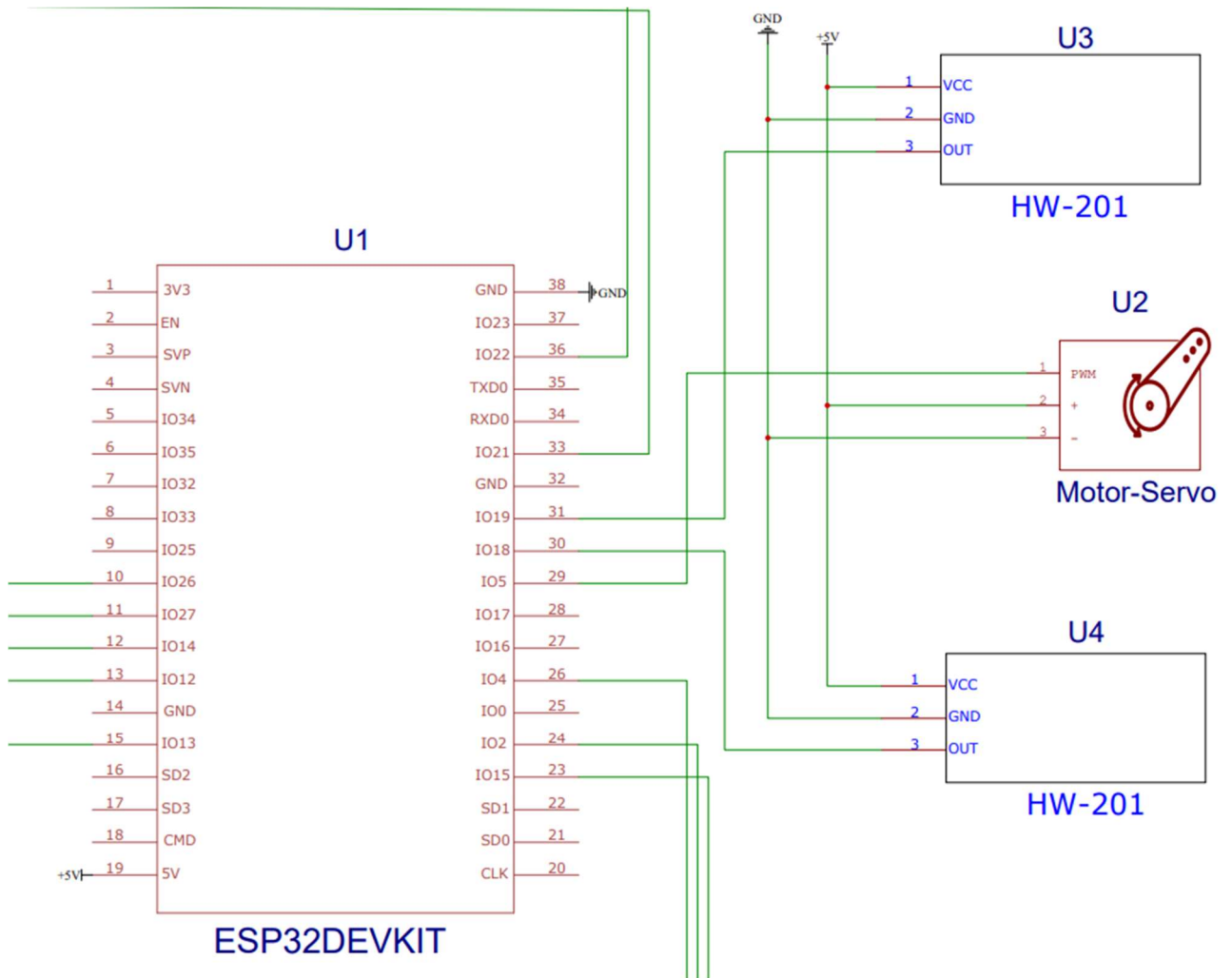


Рисунок 2.18 – Електрична принципова схема підключення сервопривода та датчиків до мікроконтролера

Рис. 2.19 зображує підключення I2C РК-дисплею та мембранної клавіатури 4x4 до ESP32. LCD дисплей має 4 контакти для підключення: живлення, заземлення, дані та контакт що відповідає за синхронізацію. Тож живлення екрану підключаємо до контакту 5 В мікроконтролера, заземлення дисплею підключаємо до відповідного контакту плати, а контакти даних та синхронізації підключаємо до контактів ESP32 під номером 21 та 22 відповідно.

Мембранна клавіатура хоч і не потребує живлення та заземлення, зате вміщує вісім контактів даних, а саме 4 контакти для стовбців та 4 контакти для рядків. Отже підключаємо ці контакти до 2, 4, 12, 13, 14, 15, 26, 27 контактів мікроконтролера.

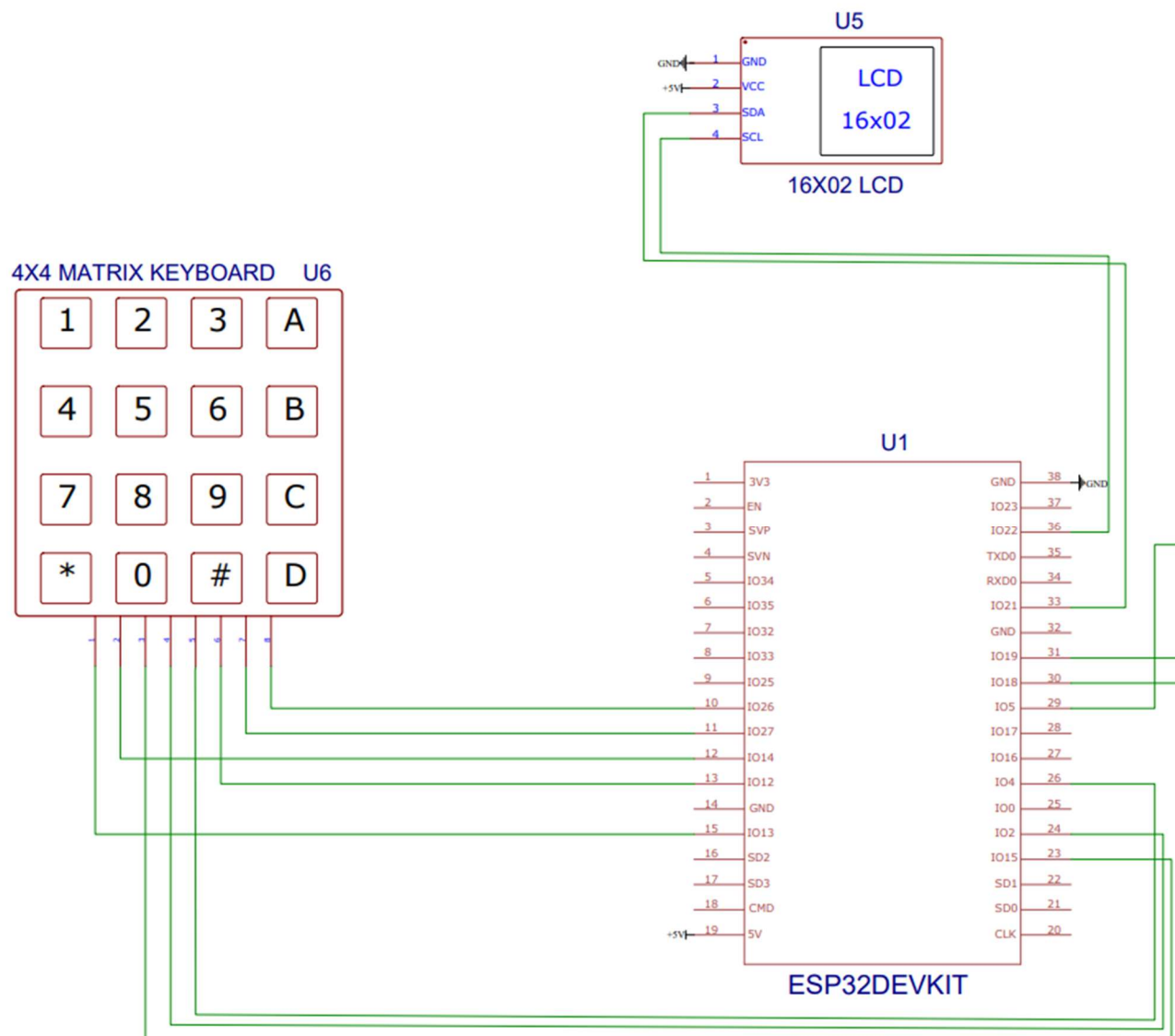


Рисунок 2.19 – Електрична принципова схема підключення дисплею та клавіатури до плати ESP32

Використання ESP32 саме у формі плати розробника дозволяє легко отримати доступ до інтернету та підключити всі потрібні модулі напряму. Адаптер PCF8574 дає змогу підключити РК-дисплей за допомогою всього чотирьох контактів, цим самим збільшуючи кількість вільних контактів для інших пристроїв. Загалом система вийшла енергоефективною завдяки мікроконтролеру та є можливість підключення додаткових пристроїв в процесі розширення системи або модифікації.

Спочатку програма виводить повідомлення про очікування введення ідентифікатора. Коли розпочався процес вводу ІД-коду, програма аналізує кожен символ отриманий з клавіатури і якщо введений символ знак «#», то стирається останній знак в рядку введення, якщо введений символ це «A», то програма завершує процес зчитування і переходить на наступну стадію виконання програми, якщо вищевказані символи не були задіяні при вводі, то новий символ додається в рядок введення.

Коли процес зчитування ідентифікатора завершено, програма отримує значення ІR датчиків обходу перешкод, один розташований на в'їзді, інший на виїзді паркінгу. Якщо якийсь з цих двох датчиків активний, то відправляється НТТР запит із значенням ідентифікатора користувача і очікується відповідь від сервера, якщо довжина відповіді дорівнює трьом, то виводиться інформація про те, що користувач не зарезервував місце. В іншому випадку на дисплей виводяться дані резервування користувача і змінюється значення серводвигуна, що дозволяє користувачу проїхати на паркінг.

3.1.2 Розробка алгоритму для програмного забезпечення телеграм бота

Рис. В.1 та В.2 (Додаток В), зображує алгоритм роботи телеграм бота для комунікації із користувачем.

Оскільки кожне повідомлення від користувача перезапускає виконання програми, потрібно якимось чином відокремлювати та маркувати певні повідомлення щоб організувати нормальну роботу алгоритму. Саме тому в боті використовується меню взаємодії та певні тригери для впорядкування потоку повідомлень від користувача.

Спершу взаємодія з будь-яким телеграм ботом починається із команди «/start», тоді програма реагує на це і відправляє початкове повідомлення і дає доступ користувачу до кнопок меню взаємодії.

					КС КРБ 123.051.00.00 ПЗ	Арк.
						36
Змн.	Арк.	№ докум.	Підпис	Дата		

Основне меню взаємодії включає такі пункти:

- зарезервувати місце на парковці;
- статус;
- розташування паркінгу;
- адміністрування бота.

Коли користувач вибирає перший пункт меню, спочатку програма перевіряє чи користувач на даний момент зарезервував місце, якщо ні то сканує наявність вільних місць на паркінгу, якщо місця є починається процес резервування місця.

Цей процес розпочинається із перевірки ІД користувача на наявність минулих резервувань, якщо такі є, то бот підтягує дані останнього резервування та пропонує зарезервувати місце для автомобіля вказаного в них, якщо користувач погоджується, то в наступних повідомленнях він обирає місце паркування та завершує резервування місця. Якщо користувач відмовляється від пропозиції, тоді він має ввести дані автомобіля для якого резервує місце, а саме марку та номерний знак, надіславши цю інформацію, користувач як в попередньому варіанті обирає місце і завершує процес резервування. Враховуючи, що кожне повідомлення перезапускає процес виконання коду, для збору інформації резервування був використаний лічильний, котрий рахує кількість надісланих користувачем повідомлень, цим самим врегульовуючи процес отримання інформації (рис.3.2).

					КС КРБ 123.051.00.00 ПЗ	Арк.
						37
Змн.	Арк.	№ докум.	Підпис	Дата		

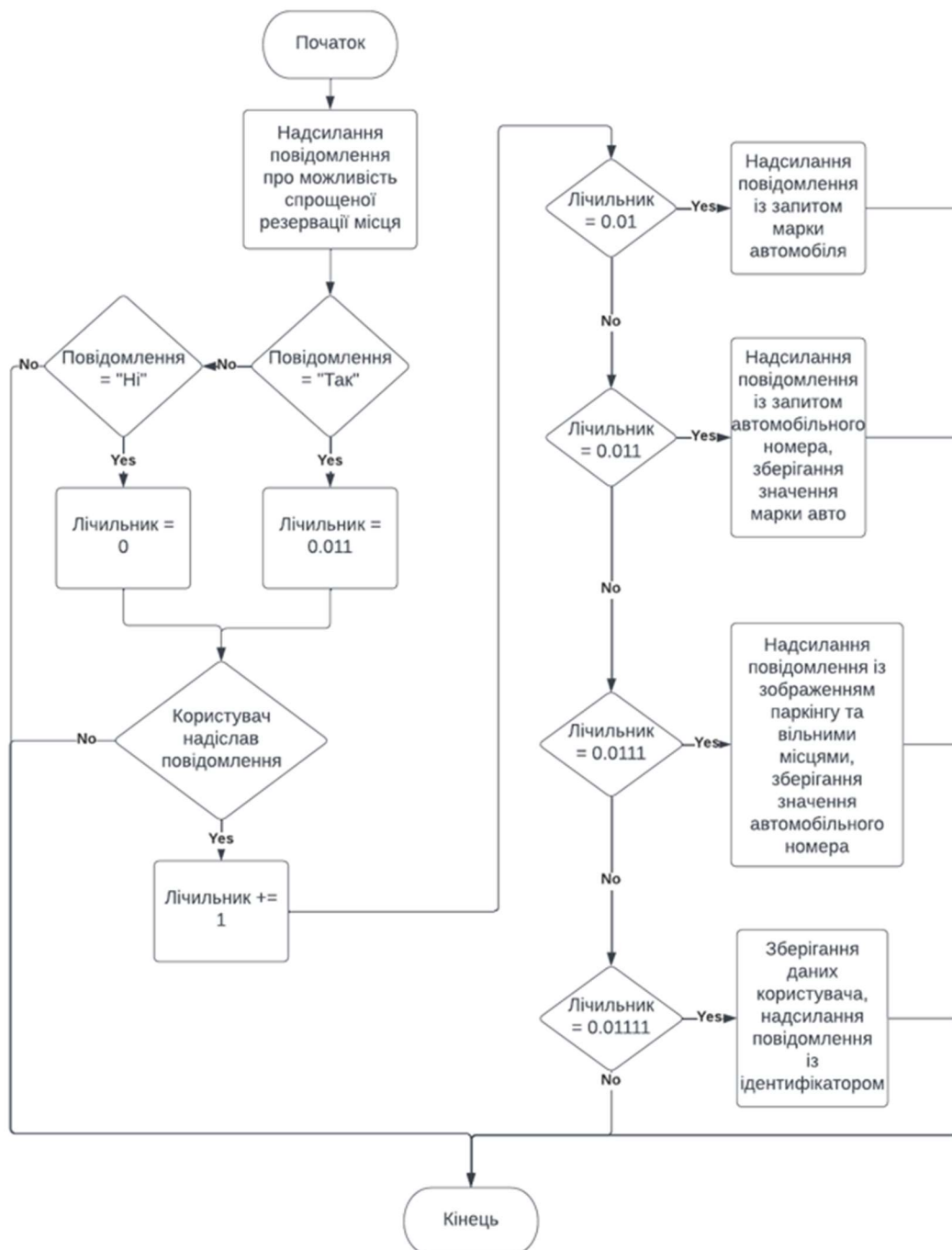


Рисунок 3.2 – Блок–схема алгоритму роботи функції резервування місця

При виборі другого пункту меню, бот перевіряє наявність ІД користувача в списку резерву, якщо таке є, то надсилається повідомлення із інформацією резервування, а саме: дата, марка авто, номерний знак та місце на паркінгу. Також користувачу надається можливість користуватись другорядним меню, воно включає такі пункти: ІД–ШКІ допуск, розташування місць на парковці, стерти резервування та назад.

Змн.	Арк.	№ докум.	Підпис	Дата

Перший пункт другорядного меню надсилає ШКІ код із ІД користувача для ідентифікації вже на самому паркінгу, другий пункт надсилає зображення всіх місць на паркінгу та текст із вибраним місцем користувачем, для того щоб користувач міг зорієнтуватись куди йому їхати. При виборі третього пункту відбувається стирання вже зарезервованого місця, за умови, якщо користувач не припаркував автомобіль на паркінгу. Останній пункт повертає користувача до основного меню.

Вибравши третій пункт основного меню, бот надсилає розташування паркінгу, яке переадресовує користувача в Google Maps, для зручного прокладання маршруту.

При виборі четвертого пункту меню, в користувача є змога зв'язатись з адміністратором, написавши повідомлення в якому вказується проблема та контакти для зв'язку.

П'ятий пункт видимий тільки для адміністраторів, отже звичайний користувач його не побачить, хоч після виклику цього пункту відбувається перевірка чи ІД користувача має відповідні права. За допомогою кнопки «Адміністрування бота» адміністратор може як додавати нових адміністраторів так і видаляти старих. Також є можливість отримати список всіх зарезервованих місць та переглянути їх статус, також є змога оновити дані про місця парковки, а саме оновити кількість загальнодоступних місць та змінити зображення всіх місць. Ще однією стадією захисту від ненавмисних дій або помилкове надання статусу адміністратора звичайному користувачеві є спосіб введення повідомлення для зміни параметрів бота. Кожна дія має свій ідентифікатор, який починається із символу «@». Для того щоб отримати список всіх зарезервованих місць, користувач має ввести «@L», для того щоб додати нову ІД адміністратора потрібно почати повідомлення із символів «@A» та ввести ІД, яку потрібно додати в список адміністраторів. Майже аналогічні дії треба зробити і для того щоб видалити ІД зі списку адміністраторів, тільки повідомлення потрібно почати із символів «@D». Для оновлення даних про місця паркінгу використовується пара символів «@P».

					КС КРБ 123.051.00.00 ПЗ	Арк.
						39
Змн.	Арк.	№ докум.	Підпис	Дата		

3.2 Реалізація проектних рішень

3.2.1 Реалізація проектних рішень програмного забезпечення мікроконтролера

Як завжди на початку кожної програми Arduino, спочатку відбувається оголошення бібліотек, далі вказуємо кількість стовпців та рядів мембранної клавіатури та вказуємо контакти підключення. Потім відбувається оголошення РК-дисплею, в якому вказується розмір дисплею та адреса на яку йдуть дані. Також відбувається оголошення контактів інфрачервоних датчиків обходу перешкод та серводвигуна, вказується інформація про WiFi до якого має підключатися система для комунікації із сервером. І ще в цій частині оголошуються певні лічильники, які застосовуються в коді (лістинг 3.3).

Лістинг 3.3 - оголошення бібліотек та всіх змінних

```
#include <WiFi.h>
#include <HTTPClient.h>
#include <LiquidCrystal_I2C.h>
#include <ESP32Servo.h>
#include <Keypad.h>
#define ROW_NUM      4
#define COLUMN_NUM   4
char keys[ROW_NUM][COLUMN_NUM] = {
    {'1', '2', '3', 'A'},
    {'4', '5', '6', 'B'},
    {'7', '8', '9', 'C'},
    {'*', '0', '#', 'D'}
};
LiquidCrystal_I2C LCD = LiquidCrystal_I2C(0x27, 16, 2);
const char* ssid = "*****";
const char* password = "*****";
const int servoPin = 5;
int flag1 = 0;
int flag2 = 0;
int flag3 = 0;
int cou = 1;
int inputPin1 = 18;
int inputPin2 = 19;
Servo servo;
byte pin_rows[ROW_NUM] = {2, 4, 27, 26};
byte pin_column[COLUMN_NUM] = {12, 13, 14, 15};
Keypad keypad = Keypad( makeKeymap(keys), pin_rows,
pin_column, ROW_NUM, COLUMN_NUM );
String input_password;
```

					КС КРБ 123.051.00.00 ПЗ	Арк.
						40
Змн.	Арк.	№ докум.	Підпис	Дата		

Лістинг 3.4 зображує функцію `setup()`, в ній налаштовуються всі датчики та відбувається підключення до інтернету.

Лістинг 3.4 – функція `setup()`

```
void setup() {  
  
    servo.attach(servoPin);  
    servo.write(100);  
    pinMode(inputPin1, INPUT);  
    pinMode(inputPin2, INPUT);  
  
    Serial.begin(115200);  
    WiFi.begin(ssid, password);  
  
    LCD.init();  
    LCD.backlight();  
    LCD.setCursor(0, 0);  
    LCD.print("Le GO!");  
    LCD.setCursor(0, 1);  
    LCD.print("WiFi ");  
    Serial.print("Connecting to WiFi");  
    while (WiFi.status() != WL_CONNECTED) {  
        delay(100);  
        Serial.print(".");  
    }  
  
    Serial.print("OK! IP=");  
    Serial.println(WiFi.localIP());  
    LCD.print("OK! IP=");  
    LCD.print(WiFi.localIP());  
  
    input_password.reserve(32);  
}
```

Лістинг 3.5 зображує частину коду, де відбувається аналіз введеного знаку із клавіатури а також формування посилання для HTTP запиту. Тут застосовуються лічильники, які були раніше оголошені, для того щоб контролювати положення серводвигуна в залежності від сигналів ІР датчиків. Це зроблено для того, щоб риг сервоприводу не спричинив пошкодження автомобілю.

					КС КРБ 123.051.00.00 ПЗ	Арк.
						41
Змн.	Арк.	№ докум.	Підпис	Дата		

Лістинг 3.5 – Програмний код, який відповідає за аналіз символів із клавіатури та формування посилання HTTP запитів

```
if (key) {
  Serial.println(key);
  if(key == '#')
  {
    input_password.remove(input_password.length() - 1,1);
    LCD.setCursor(3, 1);
    LCD.print("          ");
    LCD.setCursor(3, 1);
    LCD.print(input_password);
  }
  else{
    LCD.setCursor(3, 1);
    LCD.print(input_password + key);
  }
  if (key == 'A') {
    LCD.clear();
    LCD.setCursor(1, 0);
    LCD.print("Wait");
    if(digitalRead (inputPin1) == LOW && flag1==0){
      flag1=1;
    }
    if(flag2==0){
      String url = "https://url/exec?datIn=" + input_password;
      Serial.print("Fetching " + url + "... ");
      sendData(url);
      delay(300);
    }
  }
  if(digitalRead (inputPin2) == LOW && flag2==0){
    flag2=1;
  }
  if(flag1==0){
    String url = "https://url/exec?datOut=" + input_password;
    Serial.print("Fetching " + url + "... ");
    sendData(url);
    delay(300);
  }
}
```

Функція sendData() зображена в лістингу 3.6. В ній відбувається сам процес запиту та очікування відповіді від сервера, також там аналізується відповідь від сервера за її довжиною і в залежності від цього виводиться повідомлення на дисплей і змінюється положення сервопривода.

					КС КРБ 123.051.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		42

Лістинг 3.6 – Програмний код, який відповідає за відправку HTTP запиту та отримання і аналіз відповіді від серверу

```
void sendData(String url) {
    HTTPClient http;
    http.begin(url);
    http.setFollowRedirects(HTTPC_STRICT_FOLLOW_REDIRECTS);

    int httpResponseCode = http.GET();
    if (httpResponseCode > 0) {
        Serial.print("HTTP ");
        Serial.println(httpResponseCode);
        String payload = http.getString();
        input_password = "";
        Serial.println();
        if(payload.length() == 3){
            LCD.clear();
            String res = "You don`t resrvea place";

            for (int line = 0; line < 2; line++) {
                LCD.setCursor(0, line);
                LCD.print(res.substring(line * 16, (line + 1) * 16));
            }
            cou = 1;
            flag1=0, flag2=0;
            delay(5000);
        }
        else{
            LCD.clear();
            String name = getValue(payload, '|', 0);
            LCD.setCursor(0, 0);
            LCD.print(name);
            String place = getValue(payload, '|', 1);
            LCD.setCursor(0, 1);
            LCD.print(place);
            servo.write(0);
            flag3 = 1;
            Serial.println(payload);

            delay(1000);
        }
    }
}
```

3.2.2 Реалізація проектних рішень програмного забезпечення телеграм бота

На початку програми ми вказуємо особливий ідентифікатор створеного бота, код Google таблиці в якій зберігаються дані та оголошуємо спеціальні сценарії, які дозволяють зберігати значення (лістинг 3.7).

					КС КРБ 123.051.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		43

Лістинг 3.7 – Програмний код де прописуються ідентифікатори

```
var token = "";  
var telegramUrl = "https://api.telegram.org/bot" + token;  
var webAppUrl = "";  
var ssId = "";  
var valuePrp = PropertiesService.getDocumentProperties();  
var scriptPrp = PropertiesService.getScriptProperties();
```

На лістингу 3.8 зображено функції, які дозволяють надсилати повідомлення користувачу, а саме в текстовому варіанті, зображенням та розташування місця.

Лістинг 3.8 – Програмний код функцій надсилання повідомлень

```
function sendText(id,text) {  
    var url = telegramUrl + "/sendMessage?chat_id=" + id + "&text=" +  
        text ;  
    var response = UrlFetchApp.fetch(url);  
    Logger.log(response.getContentText());  
}  
function sendPhoto(id,text,photo) {  
    var url = telegramUrl + "/sendPhoto?caption=" + text +  
        "&photo=" + photo + "&chat_id=" + id + "&parse_mode=HTML";  
    var response = UrlFetchApp.fetch(url);  
    Logger.log(response.getContentText());  
}  
  
function sendLocation(id,text) {  
    var url = telegramUrl + "/sendlocation?chat_id=" + id + "&" +  
        text ;  
    var response = UrlFetchApp.fetch(url);  
    Logger.log(response.getContentText());  
}
```

Далі розпочинається основна частина коду бота, функція doPost(e). В ній ми отримуємо всі потрібні дані для спілкування з користувачем, а саме текст повідомлення від користувача, ІД користувача та ім'я користувача. Також тут оголошуються лічильники, які потрібні для впорядкування роботи телеграм бота та деякі змінні які потрібні в подальшій роботі (лістинг 3.9).

					КС КРБ 123.051.00.00 ПЗ	Арк.
						44
Змн.	Арк.	№ докум.	Підпис	Дата		

Лістинг 3.9 – Програмний код початку основної частини програми

```
function doPost(e) {
    var dat = JSON.parse(e.postData.contents);
    var text = dat.message.text;
    var id = dat.message.chat.id;
    var name = dat.message.chat.first_name;
    var ucon = id + "a";
    var upla = id + "b";
    var bar ="https://%20png%26style=197%26code=" + id + "A";
```

Лістинг 3.10 зображує частину коду, де описано початок роботи програми при виклику користувачем першого пункту меню. Спершу за допомогою функції searchId() програма перевіряє чи користувача немає в списку резерву, далі за допомогою функції plaCE() відбувається перевірка на наявність вільних місць. Якщо є вільні місця, функція lastAu() сканує чи користувач раніше резервував місце і якщо так, то пропонує йому зарезервувати для тих даних. За допомогою кнопок «Так» і «Ні» користувач обирає, далі запускається лічильник повідомлень та в залежності від вибору або підтягуються дані попередньої резервації, або користувач вводить нову інформацію.

Лістинг 3.10 – Програмний код роботи першого пункту меню

```
if(text == "📄 Зарезервувати місце на парковці"){
    var naz = "Rezerv";
    if(searchId(id,naz)){
    if(plaCE()== 0){
var qer = "Вибачте,але зараз доступних місць немає,спробуйте"
        sendKeyboard(id,qer,keyBoard); }
    else{
        var rres = lastAu(id,ucon,upla);
        if(rres){
            var qeo = "\nБажаєте зарезервувати місце для цього авто?";
            sendKeyboard(id,qeo,ikeyBoard);}
            else{
                valuePrp.setProperty(id, c);
                sendKeyboard(id,answer,keyDelete);}}}
    else{
        var erqw = " Ви вже зарезервували місце на праковці";
        sendKeyboard(id,erqw,keyBoard);}}
    if(text == "🚗 Так"){
        var c = 0.011;
        valuePrp.setProperty(id, c);}
    if(text == "✍ Ні"){
        valuePrp.setProperty(id, c);
        sendKeyboard(id,answer,keyDelete);}
```

					КС КРБ 123.051.00.00 ПЗ	Арк.
						45
Змн.	Арк.	№ докум.	Підпис	Дата		

На основі алгоритму дій зображеного вище, так працюють більшість пунктів меню. Код програми розбитий на велику різноманітність малих функцій, починаючи із функцій видалення інформації з таблиць, закінчуючи функціями котрі дозволяють отримати всю інформацію зарезервованих місць. Тож розглянемо декілька з них.

Функція `delety()` дозволяє видалити один рядок із таблиці з даними (лістинг 3.11).

Лістинг 3.11 – Програмний код функції `delety()`

```
function delety(value,naz){
var dat = SpreadsheetApp.openById(ssId) .
getSheetByName(naz) .getDataRange() .getValues() .slice(1);
dat.forEach((row,i)=>{
    if(row[1] == value){
SpreadsheetApp.openById(ssId) .getSheetByName(naz) .deleteRow(i+=2);
    }
})
}
```

Функція зображена в лістингу 3.12 використовується для пошуку вказаного значення в таблиці.

Лістинг 3.12 – Програмний код для пошуку даних

```
function searchId(value, naz){
var dat = SpreadsheetApp.openById(ssId) .
getSheetByName(naz) .getDataRange() .getValues() .slice(1);
let shSk = false;
dat.forEach((row)=>{
    if(shSk){
        return;
    }
    if(row[1] == value){
        result = false;
        shSk = true;
        return;
    }
    else{
        result = true;
    }
})
return result;
}
```

3.3 Тестування

Пройдемо шлях звичайного користувача. Спершу скористаємось ботом, щоб зареєструвати місце та перевірити роботу інших пунктів меню (рис. 3.13).

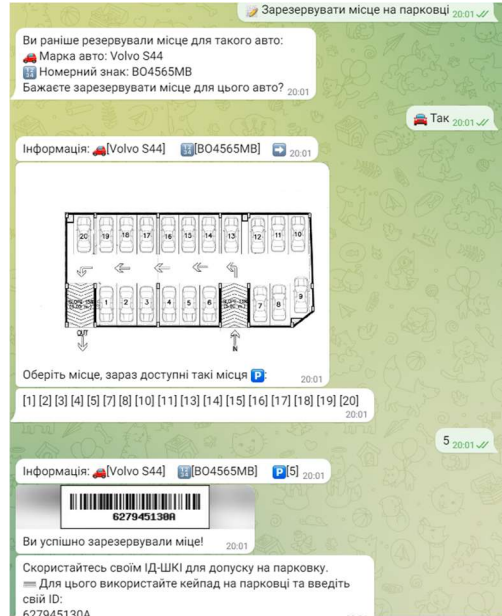


Рисунок 3.13 – Результат роботи першого пункту меню

Далі перевіримо роботу пункту «Статус» та його другорядного меню (рис. 3.14).

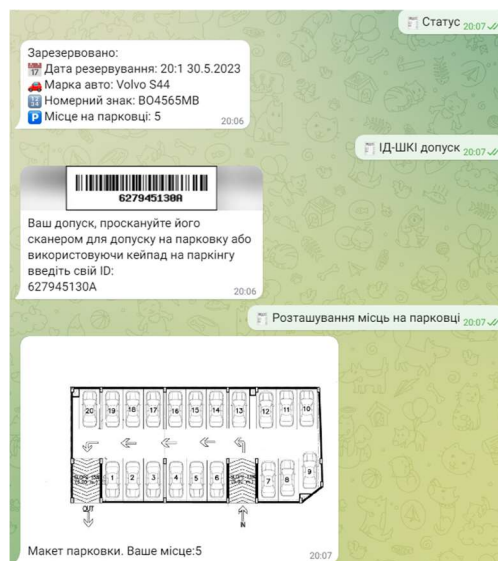


Рисунок 3.14 – Результат роботи другого пункту основного меню

Змн.	Арк.	№ докум.	Підпис	Дата

Зарезервувавши місце та перевіривши свій статус, протестуємо роботу схеми на базі мікроконтролера ESP32, яка стоїть на паркінгу. Коли користувач приїде на паркінг зреагує IR датчик на в'їзді і система попросить його ввести ІД (рис. 3.15).



Рисунок 3.15 – Результати вводу ідентифікатора користувача

Ввівши код, система перевіряє його наявність в таблиці і якщо все добре то відкриває шлагбаум та виводить інформацію резервування на дисплей (рис. 3.16).



Рисунок 3.16 – Результати роботи системи

Провівши тестування, можемо впевнитись, що система працює справно і виконує заявлені вимоги.

Змн.	Арк.	№ докум.	Підпис	Дата

РОЗДІЛ 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ХОРОНИ ПРАЦІ

4.1 Долікарська допомога при ушкодженні м'яких тканин, суглобів і кісток.

Розрізняють декілька видів ушкоджень, ось декілька з них: вивихи, розтягнення або розриви зв'язок, стискання.

Вивихи – ушкодження, при яких зміщуються суглобні кінці кісток в суглобах.

Розтягнення і розрив зв'язок – припухлість, рух у невласивому для суглоба напрямку.

Стискання – наслідок тиску великої ваги на орган людини, розчавлення м'язів, підшкірної жирової клітковини, судин і нервів.

Поранення хребта. Ушкодження хребта виникають при надмірному згинанні або розгинанні, падінні з висоти, пірнанні в неглибокому місці, під час автомобільних катастроф і обвалів.

Переломи можуть бути відкритими і закритими, з ушкодженням і без ушкодження спинного мозку.

Характерні ознаки: припухлість, локальна різка болісність, іноді утворення гематоми, можлива деформація хребта.

При ушкодженні спинного мозку настає часткове або повне переривання спинного мозку та його оболонок з розходженням кінців. Внаслідок порушення провідності спинного мозку розвивається втрата чутливості нижче місця ушкодження, паралічі, розлади функцій тазових органів.

Перша допомога: потерпілого обережно транспортують на ношах з щитом в положенні на спині. Під плечі і голову підкладають м'які валики або подушку.

Піднімати хворого з місця аварії за допомогою не менше 3-х осіб.

Ушкодження таза включають травми м'яких тканин, переломи кісток і ушкодження тазових органів (сечового міхура й уретри, прямої кишки).

					КС КРБ 123.051.00.00 ПЗ		
Змн.	Арк.	№ докум.	Підпис	Дата			
Розроб.		Пакон О.Д.			Літ.	Арк.	Аркуші
Перевір.		Жаровський Р.О.				49	6
Н. Контроль		Тиш Є.В.			ТНТУ, каф. КС, гр. СІс-43		
Консультант		Пилипець М.І.					
Затверд.		Осухівська Г.М.					

Переломи кісток таза виникають внаслідок тяжких травм, які супроводжуються здавленням тазу, здебільшого при падінні з великої висоти, обвалах, транспортних аваріях.

Розрізняють неускладнені переломи - без ушкодження внутрішніх органів, і ускладнені - з ушкодженням тазових органів.

В умовах Мирного часу, як правило, спостерігаються закриті переломи. Закриті переломи таза можуть супроводжуватись руйнуванням вертилюжної западини і центральним вивихом стегна.

При вдавленні тазового кільця в сагітальному напрямі виникають подвійні переломи Мальгєня; спереду перелом обох гілок лобкових кісток, ззаду - перелом кубової кістки.

Аналіз ставиться на підставі вивчення анамнезу, механізму травми і обстеження потерпілого.

Характерні ознаки: локальна болісність, крововилив, деформація в ділянці перелому, потерпілий перебуває в положенні «жаби»: нижні кінцівки трохи розведені і зігнуті в колінних суглобах, потерпілий самостійно не може підняти нижні кінцівки (симптом прилиплої п'ятки).

Струс головного мозку виникає при травмах черепа.

Характерні ознаки:

- втрата пам'яті, запаморочення, нудота, блювота,
- шкірні покриви бліді, холодні,
- зіниці нормальні або розширені,
- дихання та пульс сповільнені.

Важкий струс мозку супроводжується довготривалою втратою пам'яті, шоковим станом, розладом дихання та серцевої діяльності.

Перша допомога:

- потерпілого покласти на носі, голову набік;
- слідкувати, щоб не запав язик, не виникли задуха і блювання;
- транспортувати обережно.

					КС КРБ 123.051.00.00 ПЗ	Арк.
						50
Змн.	Арк.	№ докум.	Підпис	Дата		

Здавлювання мозку. Важка травма черепа, що супроводжується крововиливом в порожнину черепа і призводить до здавлювання мозку.

Характерні ознаки: симптоми струсу мозку, головний біль, порушення свідомості, нерідко коматозний стан та судоми.

Перша допомога: потерпілого покласти на носилки, голову набік; слідкувати, щоб не запав язик, не виникли задуха і блювання; транспортувати обережно.

Серед пошкоджень грудної клітки трапляються забій, струс та стиснення грудної клітки. Такі травми найчастіше виникають при катастрофах на транспорті, при обвалах і супроводжуються шоком з порушенням дихання.

Розриви легеневої тканини найчастіше виникають при переломах ребер, що супроводжуються стисненням грудної клітки. При цьому відламками ребер травмується плевра і легенева тканина. Внаслідок пошкодження легеневої тканини розвивається пневмоторакс та підшкірна емфізема. Між листками плеври від'ємний тиск, тому при розриві легенів повітря вільно виходить в порожнину плеври, що призводить до спадання легенів.

Характерні ознаки: сильний біль в місці перелому, дихання поверхневе, задуха, синюшність шкірних покривів.

Перша допомога:

- накласти тугу пов'язку у фазі максимального видиху;
- при наростанні задухи провести пункцію плевральної порожнини і відсмоктати повітря і кров;
- госпіталізація в напівсидячому положенні [12].

При використанні комп'ютеризованої системи керування паркінгом користувач повинен знати та вміти надавати долікарську допомогу при ушкодженні м'яких тканин, суглобів і кісток.

					КС КРБ 123.051.00.00 ПЗ	Арк.
						51
Змн.	Арк.	№ докум.	Підпис	Дата		

4.2 Вимоги пожежної безпеки при гасінні електроустановок

Електроустановки є потенційно небезпечними місцями для виникнення пожеж, так як вони містять велику кількість горючих матеріалів і речовин (ізоляційні матеріали, масла) і потенційні джерела займання (коротке замикання, скачки напруги, перевантаження, іскри). Таке поєднання пожежонебезпечних факторів призводить до того, що саме суворе дотримання норм безпеки не може повністю усунути можливість виникнення пожежі [14].

Основними причинами виникнення вогнищ горіння або задимлення в електроустановках є:

- аварійні ситуації, пов'язані з перевантаженням в електромережі при відсутності захисту необхідного рівня;
- коротке замикання через пошкодження обладнання або ліній електропередач;
- несправності технологічного обладнання;
- ушкодження допоміжних електромереж;
- порушення правил експлуатації і людський фактор.

При виникненні пожежі можна виділити два методи, які застосовуються для гасіння електроустановок:

- гасіння електроустановок відведених від напруги мережі;
- гасіння електроустановок, які знаходяться під напругою.

Електроустановки повинні бути під'єднані до заземлення з гнучкого мідного голого провідника з поперечним перерізом не менше 25 мм², який підключається до заземлених конструкцій. Місця підключення до заземлених конструкцій, які визначаються фахівцями енергетичних об'єктів разом з представниками гарнізону пожежної охорони, вносяться до графічної частини плану пожежогогасіння та позначаються знаком заземлення.

У разі виникнення пожежі на електроустановці особа, яка першою виявила факт загорання, повинна негайно повідомити відповідальних за пожежну безпеку осіб та керівника для уникнення подальшого загорання.

					КС КРБ 123.051.00.00 ПЗ	Арк.
						52
Змн.	Арк.	№ докум.	Підпис	Дата		

Гасіння електрообладнання під напругою із застосуванням ручних стволів повинне виконуватися за умови :

- застосування ефективних прийомів і способів подачі в зону горіння вогнегасних речовин;
- дотримання електробезпечних відстаней від електрообладнання, яке знаходиться під напругою, до пожежників, які використовують ручні пожежні стволи;
- забезпечення надійного заземлення пожежних автомобілів і стволів;
- використання індивідуальних ізолюючих електрозахисних засобів під час гасіння пожежі електроустановок без зняття напруги.

В якості вогнегасних речовин під час гасіння електроустановок, які знаходяться під напругою, доцільно застосовувати: розпилені порошкові суміші й інертні гази, струмені води, комбіновану суміш, яка являє собою розпилену воду з порошком. Застосування усіх видів піни під час гасіння електроустановок під напругою за участю людей ручними засобами забороняється, через те що піна й розчин піноутворювача мають підвищену електропровідність в порівнянні з розпиленою водою. При гасінні пожежі на електроустановках під напругою потрібно застосовувати прийоми та засоби подачі в зону горіння вогнегасних речовин, що забезпечують ефективне гасіння пожежі і безпечну роботу пожежників [14].

Правила пожежної безпеки України регламентують використання первинних засобів пожежогасіння на електроустановках. Згідно цих правил для гасіння електроустановок, які не знаходяться під напругою можна використовувати пісок, воду і вогнегасники всіх марок. Якщо електроустановка перебуває під напругою до 1000В - дозволено використовувати для придушення осередків займання або задимлення тільки вогнегасники порошкового, аерозольного або вуглекислотного типів з дотриманням всіх правил безпеки [13].

При гасінні пожежі на електроустановках, які перебувають під напругою до 220 кВ включно, тривалість перебування пожежників на бойових позиціях не лімітується .

					КС КРБ 123.051.00.00 ПЗ	Арк.
						53
Змн.	Арк.	№ докум.	Підпис	Дата		

Місця під'єднання до заземлених конструкцій повинні затверджуватися спеціалістами енергетичного об'єкта, позначатися відповідними знаками заземлення і вноситись у графічну частину плану пожежогасіння. Ручні пожежні насоси й стволи пожежних автомобілів необхідно заземлювати окремо. В процесі подачі води від внутрішнього водопроводу заземлюються лише стволи.

Індивідуальні електрозахисні ізолюючі засоби (діелектричні калоші, рукавиці) потрібно застосовувати для електробезпечності пожежників та персоналу, який безпосередньо бере участь у гасінні пожежі на електроустановках, які знаходяться під напругою.

Необхідна кількість індивідуальних ізолюючих захисних засобів на енергооб'єктах, у тому числі для пожежних підрозділів, що залучаються до гасіння пожеж з інших частин, затверджується під час розробки планів пожежогасіння.

При пожежі на електроустановках, які знаходяться під напругою, обслуговуючий персонал зобов'язаний у першу чергу повідомити про пожежу начальника зміни (диспетчера, чергового) й пожежну охорону, а потім вжити усіх необхідних заходів відповідно до плану пожежогасіння (картки пожежогасіння).

За необхідності гасіння пожежі повітряно-механічною піною, з об'ємним заповненням приміщення піною виконується попереднє закріплення піногенераторів, їх заземлення, а також заземлення насосів пожежних машин. Водій пожежної машини повинен працювати в діелектричному взутті і рукавицях [14].

При використанні комп'ютеризованої системи керування паркінгом користувач повинен знати вимоги пожежної безпеки при гасінні електроустановок.

ВИСНОВОК

Зважаючи на проведені дослідження та розробку комп'ютеризованої системи керування паркінгом, можна зробити наступні висновки:

- розроблена система на базі ESP32, яка включає датчики на в'їзді та виїзді, клавіатуру, сервер та телеграм-бот, успішно виконує функції збору даних, ідентифікації користувачів та передачі інформації на сервер.

- використання ESP32 у системі забезпечує надійність, швидкість та ефективність збору та обробки даних. Інтеграція з клавіатурою та телеграм-ботом забезпечує зручний і простий спосіб взаємодії для користувачів.

- проведені тести підтверджують, що розроблена система демонструє високу точність та швидкість реакції при ідентифікації користувачів та обробці даних. Вона здатна ефективно керувати паркінгом, забезпечуючи швидкий доступ та резервування місць.

- використання телеграм-боту як інтерфейсу для користувачів дозволяє забезпечити зручну та легку реєстрацію, резервування місць та спрощену взаємодію з системою.

На основі проведених досліджень рекомендується подальше вдосконалення системи, зокрема покращення масштабованості для більших зон паркінгу, вдосконалення системи ідентифікації користувачів та додавання функцій аналізу даних.

Усе це сприятиме подальшому розвитку системи керування паркінгом, забезпечуючи більш зручну та ефективну взаємодію для користувачів та оптимізацію управління ресурсами паркінгу.

					КС КРБ 123.051.00.00 ПЗ	Арк.
						55
Змн.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Комп'ютеризована система керування паркінгом. URL: <https://wayleadr.com/blog/parking-management-system/> (дата звернення: 06.03.2023).
2. Компанія ParkHelp. URL: <https://www.parkhelp.com/en/> (дата звернення: 08.03.2023).
3. Компанія Parkeon. URL: <https://www.flowbird.group/> (дата звернення: 08.03.2023).
4. Компанія Amano McGann. URL: <https://amanomcgann.com> (дата звернення: 08.03.2023).
5. ESP32. URL: <https://www.electronicshub.org/getting-started-with-esp32/> (дата звернення: 18.04.2023).
6. ESP32-DevKitC V4. URL: <https://docs.espressif.com/projects/espressif/en/latest/esp32/hw-reference/esp32/get-started-devkitc.html> (дата звернення: 20.04.2023).
7. РК-дисплей. URL: <https://lastminuteengineers.com/arduino-1602-character-lcd-tutorial/> (дата звернення: 22.04.2023).
8. I2C LCD дисплей. URL: <https://lastminuteengineers.com/i2c-lcd-arduino-tutorial/> (дата звернення: 22.04.2023).
9. Мембрана клавіатура 4x4. URL: <https://lastminuteengineers.com/arduino-keypad-tutorial/> (дата звернення: 23.04.2023).
10. Серводвигун. URL: <https://lastminuteengineers.com/servo-motor-arduino-tutorial/> (дата звернення: 24.04.2023).
11. Інфрочервоний датчик обходу перешкод. URL: <https://medium.com/@kekreaditya/ir-infrared-obstacle-avoidance-sensor-with-arduino-714837ad9ef5> (дата звернення: 25.04.2023).
12. Долікарська медична допомога. URL: <https://studfile.net/preview/6343723/page:7/> (дата звернення: 09.06.2023).
13. Інструкції з гасіння пожеж на енергетичних об'єктах України. URL: <https://zakon.rada.gov.ua/laws/show/z0013-12#Text> (дата звернення: 10.06.2023).

					КС КРБ 123.051.00.00 ПЗ	Арк.
						56
Змн.	Арк.	№ докум.	Підпис	Дата		

14. Особливості гасіння пожежі в електроустановках. URL: <https://euroservis.com.ua/ua/tushenie-pozharov-v-elektrostanovkakh-do-1000v.-ognetushiteli-dlya-elektrostanovok/> (дата звернення: 10.06.2023).

15. Google Apps. URL: <https://workspace.google.com/products/apps-script/> (дата звернення: 04.05.2023).

16. Arduino IDE. URL: <https://andprof.com/tools/what-is-arduino-software-ide-and-how-use-it/> (дата звернення: 13.05.2023).

17. Осухівська Г.М., Тиш Є.В., Луцик Н.С., Паламар А.М. Методичні вказівки до виконання кваліфікаційних робіт здобувачів першого (бакалаврського) рівня вищої освіти спеціальності 123 «Комп'ютерна інженерія» усіх форм навчання. Тернопіль, ТНТУ. 2022. 28 с.

					КС КРБ 123.051.00.00 ПЗ	Арк.
						57
Змн.	Арк.	№ докум.	Підпис	Дата		

Додаток А
Технічне завдання

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Тернопільський національний технічний університет імені Іван Пулюя
Факультет комп'ютерно-інформаційних систем і програмної інженерії

Кафедра комп'ютерних систем та мереж

«ЗАТВЕРДЖУЮ»

Завідувач кафедри КС

_____Осухівська Г.М.

«__» _____ 2023 р

КОМП'ЮТЕРИЗОВАНА СИСТЕМА КЕРУВАННЯ
ПАРКІНГОМ

ТЕХНІЧНЕ ЗАВДАННЯ

на _листках

Вид робіт: Кваліфікаційна робота

На здобуття освітнього рівня «Бакалавр»

Спеціальність 123 Комп'ютерна інженерія

«УЗГОДЖЕНО»

Керівник кваліфікаційної роботи

«ВИКОНАВЕЦЬ»

Студент групи СІ-41

_____к.т.н., ст.викладач Жаровський Р.О.

«__» _____ 2023 р.

_____Пакон О.Д.

«__» _____ 2023 р.

Тернопіль 2023

1. Назва та підстава для виконання роботи

1.1 Комп'ютеризована система керування паркінгом

1.2. Підставою для виконання кваліфікаційної роботи бакалавра (КРБ) є Наказ по Університету (№ 4/7–238 від 23.02.2023 р.).

2. Виконавець

2.1. Студент групи СІ–41 кафедри КС

Тернопільського національного технічного університету ім. І. Пулюя

Пакон Олег Дмитрович

3. Мета роботи

3.1. Метою роботи є розробити структуру та програмне забезпечення для комп'ютеризованої системи керування паркінгом

4. Склад виробу

4.1. До складу КС повинні входити:

1) Плата ESP32 DEVKIT.

2) Текстовий екран 16x2. LCD дисплей.

3) Інфрачервоний датчик обходу перешкод HW-201.

4) Мембрана клавіатура 4x4.

5) Серводвигун.

5. Технічні вимоги

5.1. Вимоги по призначенню.

5.1.1. Комп'ютеризована система повинна мати наступні параметри:

– Охоплення відстані: 0,5м;

– Швидкість відповіді: 2с;

– Висока частота сигналу;

5.2. Вимоги до умов експлуатації:

5.2.1. По умовам експлуатації виріб повинен відповідати вимогам ДСТУ 4158-2003

5.2.2. Працювати при кліматичних умовах -20°C до $+60^{\circ}\text{C}$.

5.3. Конструктивні вимоги.

5.3.1. Конструювання корпусу приладу в КРБ не передбачено.

5.3.2. Для побудови системи мають бути використані сучасні компоненти

5.3.3. При побудові системи необхідно передбачити розміщення роз'ємів живлення і обміну даними та наявність доступу до інтернету.

5.3.4. Габаритні розміри при макетуванні, мм, не більше:

– довжина 200мм

– ширина 100мм

– висота 100мм

5.3.5. Маса макету, кг, не більше 2

5.3.6. Конструкція макету повинна забезпечувати доступ до всіх комплектуючих виробів при тестуванні.

5.4. Вимоги до надійності.

5.4.1. Система повинна відповідати вимогам ДСТУ 2862-94.

5.4.2. Наробка на відмову, не менше 6000 год.

5.5. Вимоги метрології.

5.5.1. Вимірювання параметрів системи при моделюванні повинно виконуватись на універсальних вимірювальних приладах.

6. Економічні показники

6.1. Собівартість КС не повинна перевищувати 1000 грн.

7. Вимоги до документації

7.1. Конструкторська документація повинна відповідати вимогам ЄСКД, ДСТУ та ГОСТ.

8. Стадії та етапи розробки КРБ

8.1 Стадії та етапи виконання КРБ наведенні в табл. 1.

Таблиця 1 – Стадії та етапи виконання КРБ

№ п/п	Назва етапу	Строки виконання	
		початок	кінець
1	Технічне завдання	–	29.02.2023
2	Розділ 1	29.02.2023	12.04.2023
3	Розділ 2	12.04.2023	17.05.2023
4	Розділ 3	17.05.2023	08.06.2023
5	Розділ охорона праці	08.06.2023	11.06.2023
6	Нормоконтроль	11.06.2023	21.06.2023
7	Захист	з 22.06.2023	–

9. В дане ТЗ можуть вноситись зміни по узгодженню сторін

Додаток Б
Перелік елементів

Додаток В
Алгоритм роботи телеграм бота

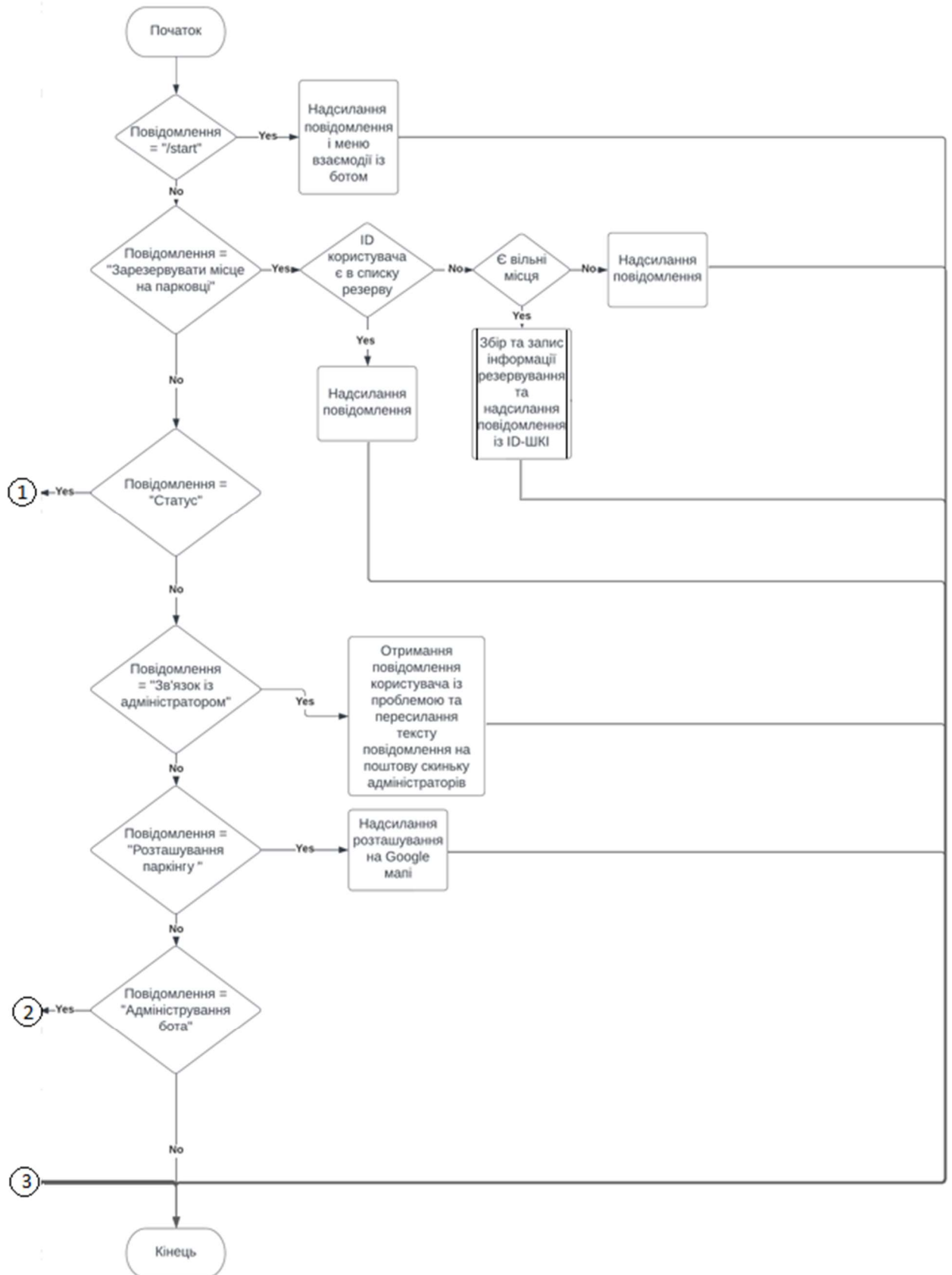


Рисунок В.1 – Блок-схема алгоритму роботи телеграм бота

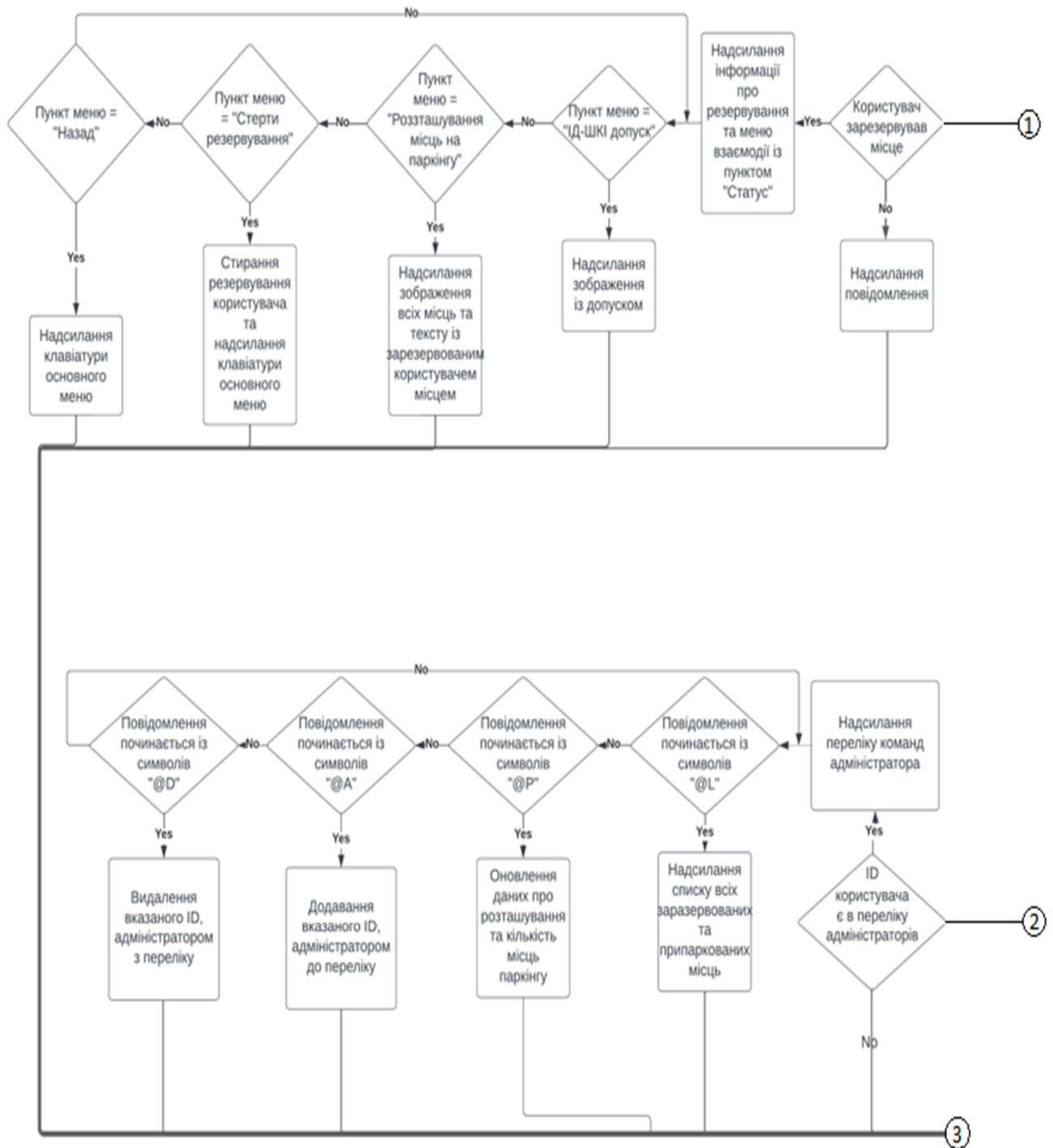


Рисунок В.2 – Блок-схема алгоритму роботи телеграм бота (продовження)

Додаток Ж

Лістинг програми

```
#include <WiFi.h>
#include <HTTPClient.h>
#include <LiquidCrystal_I2C.h>
#include <ESP32Servo.h>
#include <Keypad.h>

#define ROW_NUM 4 /*Define keypad Rows*/
#define COLUMN_NUM 4 /*Define keypad Columns*/
char keys[ROW_NUM][COLUMN_NUM] = {
    {'1', '2', '3', 'A'},
    {'4', '5', '6', 'B'},
    {'7', '8', '9', 'C'},
    {'*', '0', '#', 'D'}
};

LiquidCrystal_I2C LCD = LiquidCrystal_I2C(0x27, 16, 2);

const char* ssid = "POMO";
const char* password = "70M9502O";
const int servoPin = 5;
int flag1 = 0;
int flag2 = 0;
int flag3 = 0;
int cou = 1;
int inputPin1 = 18;
int inputPin2 = 19;
Servo servo;
byte pin_rows[ROW_NUM] = {2, 4, 27, 26}; /*Initialized ESP32 Pins for Rows*/
byte pin_column[COLUMN_NUM] = {12, 13, 14, 15}; /*Initialized ESP32 Pins for
Columns*/

Keypad keypad = Keypad( makeKeymap(keys), pin_rows, pin_column, ROW_NUM,
    COLUMN_NUM );

String input_password;

void setup() {
    servo.attach(servoPin);
    servo.write(100);
    pinMode(inputPin1, INPUT);
    pinMode(inputPin2, INPUT);
    Serial.begin(115200);
```

```

WiFi.begin(ssid, password);
LCD.init();
LCD.backlight();
LCD.setCursor(0, 0);
LCD.print("Le GO!");
LCD.setCursor(0, 1);
LCD.print("WiFi ");
Serial.print("Connecting to WiFi");
while (WiFi.status() != WL_CONNECTED) {
  delay(100);
  Serial.print(".");
}

Serial.print("OK! IP=");
Serial.println(WiFi.localIP());
LCD.print("OK! IP=");
LCD.print(WiFi.localIP());

input_password.reserve(32); // maximum input characters is 33, change if needed
}

void loop() {
if(cou == 1){
  LCD.clear();
  LCD.setCursor(1, 0);
  LCD.print("Enter your ID:");
  cou = 0;
}
char key = keypad.getKey();

if (key) {
  Serial.println(key);
  if(key == '#')
  {
    input_password.remove(input_password.length() - 1,1);
    LCD.setCursor(3, 1);
    LCD.print("      ");
    LCD.setCursor(3, 1);
    LCD.print(input_password);
  }
  else{
    LCD.setCursor(3, 1);
    LCD.print(input_password + key);
  }
}
if (key == 'A') {
  LCD.clear();
  LCD.setCursor(1, 0);

```

```

LCD.print("Wait");

if(digitalRead (inputPin1) == LOW && flag1==0){
  flag1=1;
if(flag2==0){

  String url =
    "https://script.google.com/macros/s/AKfycbxlACNXVLKzePCs7c6E_DFJq0H
    B_PGd-i9DuzLPvFFNTqjL83aQQoQy1YQVZ1GDZ8uB/exec?temperature="
    + input_password;
  Serial.print("Fetching " + url + "... ");
  sendData(url);
  delay(300);

}
}
if(digitalRead (inputPin2) == LOW && flag2==0){
  flag2=1;
if(flag1==0){
  String url
    ="https://script.google.com/macros/s/AKfycbxlACNXVLKzePCs7c6E_DFJq0
    HB_PGd-i9DuzLPvFFNTqjL83aQQoQy1YQVZ1GDZ8uB/exec?humidity="
    + input_password;
  Serial.print("Fetching " + url + "... ");
  sendData(url);
  delay(300);

}
}
}
else if(key == '#')
{

}
else {
  input_password += key; // append new character to input password string
}
}
if(digitalRead (inputPin1) == LOW && flag1==0 && flag3==1){
  flag1=1;
}
if(digitalRead (inputPin2) == LOW && flag2==0 && flag3==1){
  flag2=1;
}
if(flag1==1 && flag2==1){
  delay (1000);
  servo.write(100);
}

```

```
flag1=0, flag2=0, flag3=0;
cou = 1;
delay(1000);
}
}
```

```
void sendData(String url) {
  HTTPClient http;
  http.begin(url);
  http.setFollowRedirects(HTTPC_STRICT_FOLLOW_REDIRECTS);
```

```
  int httpResponseCode = http.GET();
  if (httpResponseCode > 0) {
    Serial.print("HTTP ");
    Serial.println(httpResponseCode);
    String payload = http.getString();
    input_password = "";
    Serial.println();
    if(payload.length() == 3){
      LCD.clear();
      String res = "You don `t resrvea place";
```

```
    for (int line = 0; line < 2; line++) {
      LCD.setCursor(0, line);
      LCD.print(res.substring(line * 16, (line + 1) * 16));
    }
    cou = 1;
    flag1=0, flag2=0;
    delay(5000);
  }
```

```
  else{
    LCD.clear();
    String name = getValue(payload, '|', 0);
    LCD.setCursor(0, 0);
    LCD.print(name);
    String place = getValue(payload, '|', 1);
    LCD.setCursor(0, 1);
    LCD.print(place);
    servo.write(0);
    flag3 = 1;
    Serial.println(payload);
```

```
    delay(1000);
```

```
  }
}
}
```

```
String getValue(String data, char separator, int index)
{
    int found = 0;
    int strIndex[] = { 0, -1 };
    int maxIndex = data.length() - 1;

    for (int i = 0; i <= maxIndex && found <= index; i++) {
        if (data.charAt(i) == separator || i == maxIndex) {
            found++;
            strIndex[0] = strIndex[1] + 1;
            strIndex[1] = (i == maxIndex) ? i+1 : i;
        }
    }
    return found > index ? data.substring(strIndex[0], strIndex[1]) : "";
}
```