

Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»

Інститут електроенергетики

(інститут)

Факультет інформаційних технологій

(факультет)

Кафедра програмного забезпечення комп'ютерних систем

(повна назва)

ПОЯСНЮВАЛЬНА ЗАПИСКА
кваліфікаційної роботи ступеня
магістра

(назва освітньо-кваліфікаційного рівня)

студенту	<i>Єгорову Владиславу Миколайовичу</i>
	(ПІБ)
академічної групи	<i>122М-21-2</i>
	(шифр)
спеціальності	<i>122 Комп'ютерні науки</i>
	(код і назва спеціальності)
освітньої програми	<i>Комп'ютерні науки</i>
на тему:	<i>Розробка роботизованого модуля на базі Arduino для визначення рівня радіаційного забруднення</i>

В.М. Єгоров

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
		рейтин говою	інституційною	
розділів кваліфікаційної роботи				
спеціальний	<i>Проф. Корнієнко В.І.</i>			

Рецензент	<i>Доц. Шедловський І.А.</i>			
-----------	------------------------------	--	--	--

Нормоконтролер	<i>Проф. Лактіонов І.С.</i>			
----------------	-----------------------------	--	--	--

Дніпро
2022

Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»

ЗАТВЕРДЖЕНО:

Завідувач кафедри

Програмного забезпечення комп'ютерних систем

(повна назва)

М.О. Алексєєв

(прізвище, ініціали)

(підпис)

« » ____

20 22 року

ЗАВДАННЯ

на виконання кваліфікаційної роботи магістра

спеціальності

122 Комп'ютерні науки

(код і назва спеціальності)

студенту

122М-21-1

(група)

Єгорову Владиславу Миколайовичу

(прізвище та ініціали)

Тема роботи: Розробка роботизованого модуля на базі Arduino для визначення рівня радіаційного забруднення

2 МЕТА ТА ВИХІДНІ ДАНІ ДЛЯ ПРОВЕДЕННЯ РОБІТ

Об'єктом дослідження є процес знаходження витоків радіації на певних об'єктах. Предметом дослідження є web-додаток аналізу знаходження витоків радіації на певних об'єктах. Метою кваліфікаційної роботи є розроблення web-додатку, що буде використовуватися зацікавленими особами, для аналізу збитків від техногенних або природніх катастроф.

3 ОЧІКУВАНІ НАУКОВІ РЕЗУЛЬТАТИ

Для вирішення задач пов'язаних з радіаційним забрудненням, в даному проєкті буде розглянуто розробку роботизованого модуля на базі Arduino. Він буде складатись із мікроконтролера Arduino, драйвера двигунів, який керує чотирма моторами та лічильника Гейгера. Роботизований модуль може керуватися за допомогою телефона на ОС Android, планшета або іншого сумісного пристрою по Bluetooth-з'єднанню.

4 ЕТАПИ ВИКОНАННЯ РОБІТ

Найменування етапів робіт	Строки виконання робіт (початок – кінець)
Аналіз існуючих рішень та постановка задачі роботи	12.09.2022-30.09.2022
Обґрунтування вибору методів аналізу знаходження витоків радіації на певних об'єктах	01.10.2022-31.10.2022
Розробка програмного забезпечення та дослідження ефективності його використання	01.11.2022-10.12.2022

Завдання видав

(підпис)

Корнієнко В.І.

(прізвище, ініціали)

Завдання прийняв до виконання

(підпис)

Єгоров В.М.

(прізвище, ініціали)

Дата видачі завдання: 10.09.2022 р.

Термін подання до ЕК 20.12.2022 р.

РЕФЕРАТ

Пояснювальна записка: 50 стор., 13 рис., 2 таблиці, 2 додатка, 26 джерел.

Об'єктом дослідження є процес знаходження витоків радіації на певних об'єктах. Предметом дослідження є web-додаток аналізу знаходження витоків радіації на певних об'єктах. Метою кваліфікаційної роботи є розроблення web-додатку, що буде використовуватися зацікавленими особами, для аналізу збитків від техногенних або природніх катастроф.

Для вирішення задач пов'язаних з радіаційним забрудненням, в даному проекті буде розглянуто розробку роботизованого модуля на базі Arduino. Він буде складатись із мікроконтролера Arduino, драйвера двигунів, який керує чотирма моторами та лічильника Гейгера. Роботизований модуль може керуватися за допомогою телефону на ОС Android, планшета або іншого сумісного пристрою по Bluetooth-з'єднанню.

В роботі представлена розробка модуля для використання в області громадської безпеки, та знаходження витоків радіації на певних об'єктах. Розроблено програмний додаток, що дозволяє оператору легко керувати модулем. Завдяки дружньому інтерфейсу, розробленому для програми, може реалізувати взаємодію людини і машини через FPV-камеру і передові датчики. Система показує свою здатність уникати і підніматися на перешкоди під час руху. Як тільки виявляє радіаційний витік, активується звуковий сигнал і попереджувальне повідомлення. Останнє включає в себе інформацію про положення витоків. Задовольняє двом поведінковим властивостям: корисності та надійності.

СПИСОК КЛЮЧОВИХ СЛІВ: ПРОГНОЗУВАННЯ, WEB-ДОДАТОК, АНАЛІЗ, МОДЕЛЮВАННЯ, БАЗА ДАНИХ, МОДУЛЬ, ТЕСТУВАННЯ.

ABSTRACT

Explanatory note: 50 pages, 13 figures, 2 tables, 2 appendices, 26 sources.

The object of research is the process of finding sources of radiation at certain objects. The subject of the study is a web-application for the analysis of radiation sources at certain objects. The purpose of the qualification work is to develop a web application that will be used by interested persons to analyze losses from man-made or natural disasters.

To solve problems related to radiation pollution, this project will consider the development of a robotic module based on Arduino. It will consist of an Arduino microcontroller, a motor driver that controls four motors, and a Geiger counter. The robotic module can be controlled using an Android phone, tablet or other compatible device via Bluetooth.

The work presents the development of a module for use in the field of public safety, and finding sources of radiation at certain objects. A software application has been developed that allows the operator to easily control the module. Thanks to the friendly interface designed for the program, it can realize human-machine interaction through FPV camera and advanced sensors. The system shows its ability to avoid and climb obstacles while driving. As soon as it detects a radiation leak, a sound signal and a warning message are activated. The latter includes information about the position of the leak. Satisfies two behavioral properties: utility and reliability.

LIST OF KEYWORDS: FORECASTING, WEB APPLICATION, ANALYSIS, SIMULATION, DATABASE, MODULE, TESTING.

ЗМІСТ

РЕФЕРАТ.....	4
ABSTRACT.....	5
ВСТУП.....	7
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ РОБОТИ.....	8
1.1. Огляд останніх досліджень і публікацій.....	8
1.2. Дистанційно керовані роботизовані системи.....	11
1.3. Аналіз існуючих аналогів.....	13
1.4. Вибір середовища розробки.....	14
1.5. Постановка задачі роботи.....	17
РОЗДІЛ 2. ПРОЕКТУВАННЯ ДОДАТКУ ДЛЯ АНАЛІЗУ РІВНЯ РАДІАЦІЙНОГО ЗАБРУДНЕННЯ	18
2.1. Основні блоки модуля	18
2.2. Мобільна платформа 4WD.....	19
2.3. Використання бібліотек Arduino	20
2.4. Проектування моделі бази даних.....	21
РОЗДІЛ 3 РОЗРОБКА WEB-ДОДАТКУ АНАЛІЗУ ЗБИТКІВ ВІД ТЕХНОГЕННИХ ТА ПРИРОДНИХ КАТАСТРОФ.....	24
3.1. Архітектура програмного додатку	24
3.2. Вбудована програма Arduino.....	29
3.3. Програмування додатку.....	32
ВИСНОВКИ.....	36
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	37
Додаток А. Код програми.....	39
Додаток Б. Перелік файлів на диску.....	51

ВСТУП

Забезпечення радіаційного контролю повинно здійснюватися у всіх напрямках професійної діяльності, умов життя, впливу на навколишнє середовище та людину. Для цього потрібно реалізовувати інструменти, які будуть забезпечувати вимірювання рівню радіаційного забруднення.

Інструменти які використовуються для вимірювання радіоактивного забруднення навколишнього середовища називаються дозиметричними приладами, завдання цих приладів це оцінка іонізуючого випромінювання. Існує чотири типи дозиметричних приладів за призначенням: індикатори, рентгенметри, радіометри та дозиметри. Вони використовують методи фотографічного, хімічного, сцинтиляційного то іонізаційного дослідження.

Основна сфера їхнього використання буде полягати в розвідці радіаційної обстановки, яка може утворитись в результаті радіаційного забруднення, на території адміністративного району, населеного пункту, господарського об'єкту або при аваріях на АЕС та інших радіаційно – небезпечних об'єктах. При оцінюванні радіаційної обстановки повинні виконуватись всі необхідні дії, для надання захисту населенню та робочим групам ліквідаторів та інших формувань з надзвичайних ситуацій, в умовах виникнення радіоактивного забруднення. Існує два етапи оцінки радіаційної обстановки, це є виявлення та оцінка обстановки. Перший етап включає в себе збір інформації про рівень радіації на території АЕС, або інших заражених територіях. На основі отриманих даних, другий етап полягає в тому щоб безпосередньо оцінити рівень радіаційного забруднення. Оцінка радіаційної обстановки вимірюється у відносному часі, за початок відліку береться, час коли сталася аварія або інша надзвичайна ситуація.

Для вирішення задач пов'язаних з радіаційним забрудненням, в даному дипломному проекті буде розглянуто розробку роботизованого модуля на базі Arduino. Він буде складатись із мікроконтролера Arduino, драйвера двигунів, який керує чотирма моторами та лічильника Гейгера. Роботизований модуль може керуватись за допомогою телефона на ОС Android, планшета або іншого сумісного пристрою по Bluetooth-з'єднанню.

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ РОБОТИ

1.1. Огляд останніх досліджень і публікацій

Для вимірювання радіоактивного забруднення, різних об'єктів, речовин та місцевості використовують пристрої радіаційної розвідки. Визначення доз поглинутого радіаційного опромінення відбувається за допомогою пристроїв дозиметричного контролю, який являє собою комплекс заходів, для визначення доз опромінення людей. Дозиметричний контроль дає змогу правильно оцінити вплив іонізуючого випромінювання на людину [1].

Умовно виділяють три групи пристроїв для вимірювання іонізуючих випромінювань, сюди відносяться радіометри, спектрометри і дозиметри (Рисунок 1.1).

Радіометри – використовуються для вимірювання активності в радіоактивних речовинах, також вони дозволяють виміряти ступінь забруднення різних поверхонь або об'єктів рослинного і тваринного походження. В основу радіометра покладено газорозрядні датчки і сцинтиляційні лічильники.

Спектрометри - оптичні пристрої, які призначені для накопичення спектру його кількісної обробки та аналізу. Для отримання аналізованого спектру використовується рентгенівське, лазерне або іскрове випромінювання. Реєстрація таких випромінювань відбувається шляхом флуоресценції. Зазвичай в дослідженнях вимірюють інтенсивність випромінювання та його довжину, хвилю або частоту, також можуть бути визначені додаткові параметри. Діапазоном роботи приладів є довжина хвиль від гамма до інфрачервоного випромінювання [5].

Дозиметри – пристрої які вимірюють дозу радіаційного опромінення, яка накопичується за певний проміжок часу в умовах перебування в зонах підвищених радіаційних навантажень. Такі пристрої використовуються для оперативного вимірювання рівня радіації в ручному режимі, або в якості попереджувальних індикаторів радіаційної небезпеки

Серед радіометрів існує дві основні групи, стаціонарні і переносні. Стаціонарні радіометри розрізняються за електричними і експлуатаційними

параметрами, і за своєю конструкцією, але у них схожа модель-схема пристрою, яка складається з імпульсивного підсилювача, перерахункового приладу, реєструючого пристрою, який візуально показує результати отриманого вимірювання, також такому пристрою потрібний елемент джерела високої напруги для живлення детектора. Роботу такого приладу забезпечує мережа змінного струму. Для виявлення радіаційного випромінювання та перетворення енергії від випромінювання в інші види енергії використовують детектор. Зазвичай амплітуда сигналів які поступають від детектора є не достатньою для їхньої реєстрації, тому в більшості приладів які призначені для вимірювання іонізуючого випромінювання використовують імпульсні підсилювачі. За допомогою підсилювача ці сигнали посилюються, формуються і відправляються на перерахунковий прилад. Перерахунковий прилад є основним компонентом радіометра. Задача цих приладів полягає у вимірюванні імпульсів в числовому значенні, які надійшли за окремий проміжок часу, по-іншому такі прилади називаються лічильниками імпульсів.

Переносні, лабораторні і польові радіометри являють собою невеликі пристрої, живлення яких може бути автономним або бути від мережі. Такі пристрої використовують для виявлення радіоактивних речовин, та для визначення їх гамма- чи бета-випромінювання. В їхній основі застосовується простий електричний пристрій який дає змогу отримувати інформацію за допомогою цифрової, світлової чи звукової індикації випромінювання.

Польові дозиметри - широко застосовуються для пошуку радіоактивних речовин, та рівня гамма-випромінювання. Такі дозиметри є невеликі за своїми розмірами, інформація яку вони отримують відображається на шкалі стрілочного приладу, також можуть бути присутні світлові чи звукові індикатори. Детекторами в таких дозиметрах виступають газорозрядні і сцинтиляційні лічильники. Індивідуальні дозиметри – призначені для вимірювання індивідуальної еквівалентної дози рентгенівського та гамма-випромінювання. Принцип дії побудований на підрахунку числа імпульсів, які поступають на вхід детектора який обраховує еквівалентну отриману дозу за спеціальним алгоритмом. При цьому час за який буде виміряно потужність дози вибирається автоматично, але враховується статистична похибка [6].



Рис. 1.1. Пристрої для вимірювання радіаційного забруднення

Робота з приладами які вимірюють радіаційне забруднення, вимагає певної підготовки, знать та навичок. Знаходження в зоні інтенсивного опромінення може згубно відобразитись на здоров'ї людини, при опроміненні критичного об'єму клітин порушується функціонування тканин і органів, такий ефект називається реакцією тканин і проявляється незабаром після отриманого опромінення в залежності від отриманої дози можуть виникати різні патологічні утворення, як приклад це може бути безпліддя, еритема, помутніння кришталика ока. Гостре опромінення у великих дозах, серйозно пошкоджує життєво важливі органи, що може викликати променеву хворобу або навіть смерть.

Основне завдання радіаційної безпеки полягає в обмеженні шкоди завданої для здоров'я людини. Перед тим як розпочати роботу з засобами радіаційної розвідки, оператор повинен впевнитись у власному захисті від іонізуючого опромінення. Практично цього можна досягнути при правильній роботі з самим джерелом зараження, та відповідною організацією людської діяльності.

Існує три основних принципи радіаційної безпеки. Перший принцип полягає у тому, що всі рішення пов'язані з радіаційною небезпекою повинні бути відповідно обгрунтовані, щоб уникнути завдання ще більшої шкоди. Наступний принцип оптимізації, він ставить перед собою основну задачу, утримувати настільки низький рівень опромінення, наскільки це взагалі можливо. Останній принцип нормування, полягає в тому, що при будь-якому плановому опроміненні, повинні дотримуватись граничні значення доз встановлені законодавством. Областями регулювання радіаційної безпеки є об'єкти які можуть створювати дозу опромінення небезпечну для людини або певної кількості людей. Ними

можуть бути безпосередньо природний фон або пристрої які містять радіоактивні матеріали. Захист від зовнішнього іонізованого опромінення перш за все повинен бути направлений на захист від шкідливих факторів.

Завдання дозиметрії базується на вимірюванні характеристик радіаційного поля, щоб кількісно оцінити ефект від опромінення на організм. Оператор повинен бути захищений за допомогою бар'єрів часу і відстані, це є найраціональніший спосіб уникнути опромінення. При проведенні радіаційної розвідки в умовах випромінювання отримана доза прямо пропорційна часу проведеного в зоні дії випромінювання і обернено пропорційна квадрату відстані від джерела. Якщо не вдалось уникнути зараження радіоактивними елементами вони потрапляють в організм людини і він стає джерелом внутрішнього опромінення. В подальшому практично не можливо вплинути на джерело внутрішнього опромінення, тому захист в такому випадку направлений на запобігання подальшому розповсюдженню зараження. Ефективність заходів безпеки повинна визначатись при обстеженні людини та навколишнього середовища. Для своєчасного виявлення зараження організму оператор пристроїв радіаційної розвідки в обов'язковому порядку повинен проходити медичне обстеження [2].

1.2. Дистанційно керовані роботизовані системи

Робототехніка є практичною галуззю, яка займається розробкою роботів та автоматизованих систем. Сьогодення показує, що це є одним з найперспективніших напрямків у сфері інтернет-технологій. Процес створення роботизованої системи також включає в себе деякі перспективні напрямки, наприклад створення систем штучного інтелекту або дистанційного-керування та інших. В умовах коли безпосередня участь людини, в тих чи інших процесах, може бути небезпечною для її здоров'я, потрібно забезпечувати виконання робіт різного напрямку і ступеню складності за допомогою дистанційно-керованих роботизованих систем [7].

Блочно-ієрархічний підхід заснований на виділенні різних ієрархічних рівнів робота, наприклад рівень планування руху, рівень управління, рівень виконавчих механізмів. При цьому на верхньому рівні дається загальний опис

робота, а на наступних рівнях цей опис деталізується. Для ієрархічної структури характерний зв'язок елементів із сусідніми рівнями, а міжелементні зв'язки на одному рівні відсутні. Найбільш часто розрізняють три рівні проектування. На верхньому рівні зазвичай розробляють структурні схеми та загальний вигляд робота. на середньому рівні відбувається розробка окремих пристроїв, в результаті чого з'являються функціональні і принципові схеми. На нижньому рівні проектуються окремі деталі і елементи роботів. Зазвичай проектування роботів є багатокроковим етапом, що вимагає проведення науково-дослідних робіт, потім слідує етап ескізного проекту, технічного проекту, етап випробувань.



Рис. 1.2. Види дистанційно керованих роботів за призначенням

Кожний з приведених етапів може бути розділений на різні частини, які називаються процедурами. Основні сфери використання дистанційно-керованих систем в умовах екстремальних середовищ та місцях які є важко доступними або небезпечними для людини, серед таких сфер потрібно відзначити космічні та підводні простори, виробництва і лабораторії з високим ризиком радіаційного випромінювання, це також можуть бути зони обслуговування термоядерних енергетичних установок, або контроль радіаційного рівня на територіях АЕС як показано (Рисунок 1.2). Велике значення надається використанню таких систем пов'язаних з різними галузями людської промисловості, побуту та медицини. Новим важливим етапом є створення рухомих, а також крокуючих апаратів, - це важливий етап при вивченні місяця, марсу та інших планет. Такі роботизовані платформи

розширюють можливості космічних досліджень, та дозволяють за допомогою одного апарату зібрати різну кількість даних з поверхні. Освоєння Місяця в майбутньому не можливо уявити без автоматичних та дистанційно-керованих засобів. Можна також створювати великогабаритні планетоходи, для перевезення вантажів та астронавтів, такі машини будуть обладнані оснащенням різного виду, в тому числі і маніпуляторними пристроями.

1.3 Аналіз існуючих аналогів

В світі розробляють різноманітні роботизовані системи для вимірювання радіаційного фону. Великі проектні системи розробляють в основному спеціалізовані інститути і компанії, та такий проект може коштувати значну суму грошей і людських ресурсів. Проекти менш затратної складності також мають право на життя, але їхня якість буде значно нижчою. Багато країн розробляють таких роботів, за для власних проблем. Особливо після наслідків аварії на Чорнобильській АЕС, коли стало зрозуміло що їхнє використання зможе зберегти людські життя.

В Україні, за роки що минули після аварії Національною академією наук України. Проектувались дистанційно керовані агрегати. Велися роботи по створенню і використанню всередині об'єкта "укриття" діагностичних дистанційно керованих пристроїв. Було виготовлено сімейство роботів різного призначення для вирішення поставлених завдань.

Китайські науковці також розробляють свого першого робота. До складу його оснащення буде входити маніпулятор і сенсорний апарат, здатний проводити різноманітні дослідження в умовах радіаційної обстановки. За своїми розмірами він буде не достатньо великий, відомо що його вага буде близько 30 кілограмів, а в висота його становитиме 88 см. Такий модуль зможе долати різні перешкоди, а також перетинати не великі водойми.

Робот зміїного типу був використаний в реакторі Ringhals 1 в Швеції для ремонту витоків, що надходить з труб, які є частиною механізму приводу штока управління, розташованого під корпусом реактора. Для цієї мети була розроблена точна копія, яка гарантує, що робот-маніпулятор "змія" зможе успішно виконати свою місію. Здатність такого робота досягати і ремонтувати області, які повинні були бути

недоступні протягом усього життєвого циклу об'єкта, представляє інтерес для його майбутніх застосувань.

Компанія Guan, розробили двоногу модель з п'ятьма ступенями руху, робот-скалолаз, що використовує спеціальні присоски. Він оснащений камерою, та дозиметром, а також датчиком вологості і бездротовим відеопередавачем. Навігаційна система використовує сигнали від датчика відстані, інклінометра і датчика тиску. Вбудована система управління включає в себе Мікроконтролер ST(STM32F103) на базі ядра ARM. Маса робота становить близько одного кылограма, а його розміри є достатньо маленькими, він був добре оцінений за його навігаційні здібності, та вміння долати перешкоди.

Зовсім недавно Dongetal, представили водонепроникний гусеничний колісний робот, який включає в себе можливість підйому по сходах за допомогою додаткових поворотних важільних гусениць. В основному корпусі знаходяться акумулятор, двигуни та обладнання бездротового зв'язку. Останній розміщений в окремому сегменті корпусу, щоб уникнути електромагнітних перешкод через силового пристрою. Радіаційний захист забезпечується за допомогою кадмієво-селенової оболонки на камері робота, а захисної вольфрамової та на внутрішніх електричних компонентах корпусу і посрібленого паперу на проводці. Управління положенням і рухом поворотної руки включено в людино-машинний інтерфейс. Автори включають в себе розробку алгоритму управління рухом робота. Ніякої інформації про датчики, що використовуються в системі, не наводиться.

1.4. Вибір середовища розробки

Платформи Arduino бувають різних сортів. Більшість з них базуються на мікросхемах AVR - серії, які випускає Atmel. У той час як саме ці чіпи забезпечують всю реальну функціональність, плата Arduino просто полегшує їх використання. Можна стверджувати, що для досвідчених користувачів електроніки справжня цінність Arduino походить не від плат, а від всієї екосистеми з відкритим кодом, яка виросла навколо них, і всіх бібліотек тав спільнот, які вони надають.

Arduino був обраний для цього проекту, тому що він зручний для початківців, простий в розробці і має безліч прикладів і навчальних посібників в Інтернеті.

Arduino shields-це плати доповнення, які можуть бути покладені поверх плати Arduino для подальшого розширення її можливостей. На ринку існують сотні різних видів щитів, як офіційних, так і неофіційних. Вони надають такі функції, як бездротовий зв'язок у вигляді Wi-Fi або радіо, простіший зв'язок з двигунами і сервоприводами, додаткові датчики і багато іншого. Плати розширення є основними елементами для розробки цього проекту, далі розглянуто деякі основні плати Arduino Shields.

Сенсорний модуль є хорошим доповненням для будь-якого проекту Arduino. РК-екран із сенсорним екраном TFT- це простий спосіб розпочати роботу з екранами такого типу. Існують модулі різного розміру та 2,4-дюймовий РК-дисплей є досить популярним, цей РК-дисплей підтримує роздільну здатність 320x240 пікселів. Існує багато РК-контролерів, але SPFD5408 є один із найвідоміших, він також містить слот, що підтримує карти microSD.

Управління роботами і радіокерованими автомобілями за допомогою джойстика – це доволі зручний процес. Модуль джойстика є досить складним, так як він не підходить для макетної плати. Таким чином, використання плати джойстика поверх плати Arduino повністю змінює відчуття. Існує кілька типів щитів джойстика, але той, який представлений тут, являє собою повний набір. Він містить 2-осьовий модуль джойстика, 6 Кнопок, роз'єм послідовного інтерфейсу, роз'єм інтерфейсу I2C і спеціальний роз'єм модуля nRF25L01.

Arduino IDE-це мультиплатформенна платформа розробки для створення вбудованих програм, які працюють на багатьох різних мікроконтролерах Arduino.В порівнянні з багатьма іншими сучасними IDE він є досить базовим за своєю функціональністю та іншими можливостями. Розгортання коду на мікроконтролері в Arduino IDE досить не складне. Потрібно просто підключити USB-кабель до USB-порту Arduino, вибрати правильну модель плати з IDE і завантажити код. Весь Arduino IDE з його численними службовими бібліотеками насправді є просто зручною оболонкою навколо компілятора AVR-GCC, який обробляє компіляцію вихідного коду у формат, який може зрозуміти вбудований ATmega Arduino. Після того як код був перевірений на наявність помилок і скомпільований в двійковий файл, інша утиліта під назвою AVRDUDE (скорочення від AVR Downloader/UploADer) обробляє завантаження і збереження

програми на самому Arduino. Багато просунутих користувачів вважають за краще обходити відносну громіздкість Arduino IDE за допомогою свого кращого текстового редактора / IDE і просто використовувати the AVR-GCC і AVRDUDE в якості автономних програм.

Функції `setup` і `loop` є єдиними двома необхідними функціями, які завжди повинні існувати в дійсній програмі Arduino. Функція Налаштування `setup` викликається один раз при першому включенні мікроконтролера на початку програми. Потім програма продовжує працювати постійно всередині функції `loop`. Далі приведено модель життєвого циклу для програм на Arduino (Рисунок 1.3.).



Рис. 1.3. Модель життєвого циклу для програм Arduino

Android – це дуже популярна операційна система на базі Linux, спеціально розроблена для смартфонів і планшетів. За оцінками в даний час існує більше одного мільярда пристроїв запущених на Android у світі. Крім планшетів і телефонів, Android також присутній наприклад в телевізорах, наручних годинниках і карта. Будучи заснованою на ядрі Linux, Android є технологією з відкритим вихідним кодом. Android спочатку був розроблений Android Inc., однак компанія була придбана Google в 2005 році, який в даний час підтримує Android.

Незважаючи на те, що Android технічно є технологією з відкритим кодом, більшість виробників включають принаймні деякі власні драйвери з закритим вихідним кодом, залежно від моделі. Це має небажаний побічний ефект багатьох пристроїв Android, які важко оновити, оскільки Google не може просто випустити нову версію операційної системи. Через це Android піддавався критиці за те, що

має багато фрагментації щодо його різних версій. Це робить його важким для розробників Android, оскільки їм, доводиться розробляти програмне забезпечення для багатьох сильно відмінних версій Android. Для цього проекту Android був обраний в якості розробки аплікації, для керування роботизованим модулем.

Git-це розподілена система управління версіями з відкритим вихідним кодом. В основному Git працює шляхом збереження і відправки знімків проекту в віддалений репозиторій. Інші люди можуть потім фіксувати і витягувати зміни з цього репозиторію. Будучи розподіленим, Git немає термінової необхідності в розміщенні служби управління версіями в інтернеті, тому що кожен репозиторій Git містить всю незакращену історію проекту. Git був спочатку створений Лінусом Торвальдсом, щоб допомогти в розробці ядра Linux . Незважаючи на те, що це традиційно програма командного рядка, позбавлена графічного інтерфейсу, багато таких сторонніх програм з'явилися з моменту git's створення Git.

1.5 Постановка завдання

Мета даного проекту полягає в створенні такої системи, яка здатна забезпечувати радіаційну розвідку в умовах підвищеного радіаційного фону. Створення цієї системи ставить перед собою наступні задачі:

- Дослідження теми проекту, визначення необхідних завдання, що зможуть покращити роботу на небезпечних ділянках.
- Вибір компонентів для створення модуля, яким можна керувати дистанційно.
- Написання програмних додатків, та тестування системи.

РОЗДІЛ 2

ПРОЕКТУВАННЯ ДОДАТКУ ДЛЯ АНАЛІЗУ РІВНЯ РАДІАЦІЙНОГО ЗАБРУДНЕННЯ

2.1. Основні блоки модуля

У даній роботі розроблена робототехнічна система для досягнення поставлених завдань що до забезпечення громадської радіаційної безпеки на АЕС, лабораторіях чи інших підприємствах, на яких є ризик радіаційного випромінювання. Система ділиться на п'ять основних блоків наступним чином:

1. Блок руху: він відповідає за переміщення робота в чотирьох напрямках. Цей блок включає мотор і 4 приводи мотора прикріплених до пластикових колес. Блок дозволяє пристрою рухатися вперед і назад, повертатися вліво і вправо і обертатися на 360°.

2. Блок управління: це основна електронна плата, яка склеює інші компоненти. Він закріплений по-середині корпусу керованого роботу. Було вирішено використати в системі мікроконтролерну плату Arduino UNO R3. Цей блок полегшує взаємодію між різними платами розширеннями роботу.

3. Чутливий блок: цей блок включає в себе в основному датчик випромінювання, який використовується для виявлення та вимірювання поточного рівня випромінювання в місці розташування робота. Для цифрового вимірювання рівня випромінювання використовується лічильник Гейгера для оповіщення про підвищений рівень радіаційного випромінювання на платі є пезо-динамік.

4. Блок бездротового зв'язку: цей блок служить для зв'язку між роботом і керуючим середовищем. Керуюче середовище, що включає програмне забезпечення, за допомогою якого здійснюється управління і моніторинг робота, буде детально розглянута в наступних розділах. Центральною частиною даного пристрою є бездротовий комутатор, який кріпиться до корпусу модуля. Цей блок включає в себе GPS-контролер, який використовується для визначення поточного місця розташування (його географічних координат, широти і довготи) робота при виникненні ситуації витоку радіації. GSM-контролер забезпечений SIM-картою для відправки відповідних SMS-повідомлень оператору в разі виникнення тривоги.

5. Блок зору: система роботу буде оснащена FPV камерою, яка діє як очі оператора

робота. Він підключений до Arduino UNO R3 і закріплений поверх платформи. Працює камера в денному режимі, так що бар'єри, з якими стикається під час свого руху платформа, будуть оминатися.

Модуль розроблений таким чином, щоб забезпечити високу гнучку мобільність, високий ступінь точності і надійності. Наприклад, рух, обертання і рівні енергоспоживання робота повинні бути в гармонії для швидкого руху і адаптивних функцій. Таким чином, точні результати можуть бути досягнуті з низькими витратами.

2.2. Мобільна платформа 4WD

В якості платформи для розміщення елементів роботизованої системи була обрана мобільна платформа 4WD для Arduino. Включає в себе 4-кратні приводні двигуни, 4-кратні колеса і повне шасі з монтажним обладнанням. Включений другий рівень дозволяє додати більше електроніки і визначити модульну компоновку. Прямокутний отвір дозволяє прикріпити сервопривід і використовувати його для декількох цілей, таких як рухома камера або роботизована рука. Високоміцний матеріал корпусу з алюмінієвого сплаву забезпечує гнучкість при швидкому переміщенні, особливо на відкритій траві, гравії, піску або похилій поверхні. Збільшений термін служби електроніки і зубчастих коліс завдяки структурі дослідження chassis.

Мобільна платформа Pirate – 4WD для Arduino також підходить для змагань роботів і дослідницьких проектів. Різні датчики можуть бути легко додані і захищені від зовнішніх пошкоджень. Нижній міцний корпус захищає двигуни і акумуляторну батарею, щоб уникнути пошкоджень при падінні. Якщо ви раніше будували з врятованих деталей, ви можете відрізнити їх від високоміцного алюмінієвого сплаву, який утримує деталі разом і призначений для датчиків і електронної збірки.

2.3 Використання бібліотек Arduino.

Однією з найкращих особливостей проекту Arduino є можливість додавання попередньо створених бібліотек, які забезпечують апаратну підтримку. Таких бібліотек існує дуже багато, і можна обрати необхідну, для того щоб встановити. Вони завантажуються лише тоді, коли необхідно працювати з обраним елементом, або групою елементів, без написання нового коду. Здебільшого можна

просто завантажити та сховати їх для подальшого використання.

Ескізи часто залежать від бібліотек, для того щоб побачити, що вони собою представляють, потрібно подивитись у верхню частину ескізу.

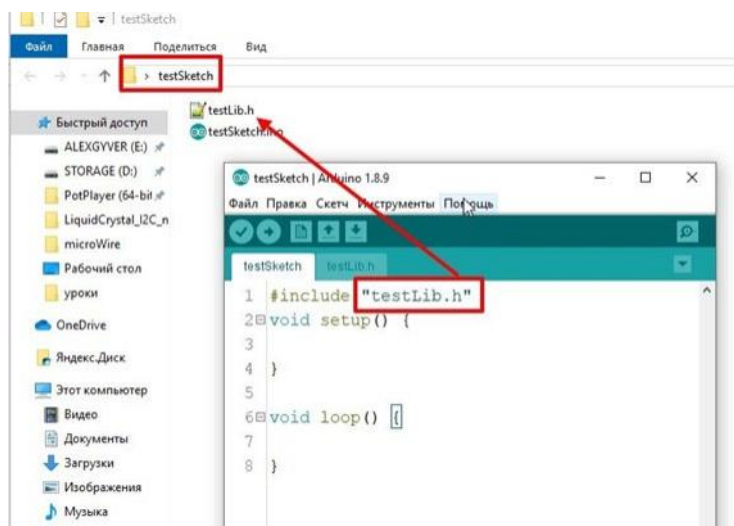


Рис. 2.1. Підключення бібліотеки в Arduino IDE

В бібліотеці знаходиться папка з деякими файлами в ній, файли будуть закінчуватися на .cpp (файл коду C++) і .h (файл заголовка c++). Також можуть бути деякі файли .o. Файли .o є скомпільованими об'єктами c++. Якщо при роботі над бібліотекою відбувається її зміна, потрібно обов'язково видалити файли .O, так як це змусить Arduino IDE перекомпілювати змінені файли .cpp в нові .o.

У Arduino v16 та попередніх версіях бібліотеки зберігалися в папці ArduinoInstallDirectory/hardware/libraries, яка також містила всі вбудовані бібліотеки (наприклад, дротові та послідовні).

У версії v17 і вище Користувальницькі бібліотеки тепер зберігаються в папці ArduinoSketchDirectory/libraries. Тому тепер, не доведеться переміщати та перевстановлювати свої бібліотеки кожного разу, коли оновлюється програмне забезпечення.

Збірка роботизованого модуля розроблена таким чином, щоб забезпечити високу гнучку мобільність, високий ступінь точності і надійності. Наприклад, рух, обертання і рівень енергоспоживання повинні працювати в гармонії для швидкого руху і адаптивних функцій. Таким чином, точні результати можуть бути досягнуті з низькими витратами.

При проектуванні системи слід враховувати три істотні характеристики:

1. Плавний рух і швидкість роботи: швидкість і рух роботи є важливими характеристиками для реалізації плавного виконання і точної роботи. У деяких середовищах як піщані або кам'янисті землі, робот повинен перекривати умови заносу і залипання коліс. Крім того, погодні умови, такі як дощ, сніг, бурі і сильний вітер виступає в якості негативних факторів. Для цього система повинна бути оснащена двигунами високої потужності і гнучким шасі з інтенсифікованого пластику для того, щоб уникнути раніше згаданих перешкод і працювати точно в вище згаданих умовах. Земля і Строби напруги струму відрегульовані для того щоб рушити основний модуль робота вперед і назад. Робот рухається з швидкістю за замовчуванням, яка коливається від 100 ШІМ (широтно-імпульсна модуляція) до 255 ШІМ. Плата моторного приводу з'єднана з двигуном постійного струму, який запрограмований на те, щоб уникати перешкод, автоматично і поступово збільшуючи швидкість системи при кожному зіткненні з перешкодами.
2. Швидкість відгуку сигналів дистанційного керування на обертання і рух: стійкість над землею повинна бути гарантована під час руху робота. Таким чином, чотири колеса мають завжди бути на зв'язку з землею. У випадку, якщо роботизований модуль рухається вперед або у зворотному напрямку кожен двигун, пов'язаний з колесом, повинен працювати одночасно з усіма іншими двигунами. При повороті вправо і вліво два з цих двигунів будуть вимкнені автоматично. В результаті швидкість відгуку збільшується на кривій, а загальна продуктивність поліпшується. Двигун постійного струму з'єднаний з блоком руху, схема підключення електродвигуна постійного струму до плати електродвигуна і КУ разом з лініями введення/виведення живлення проілюстрована нижче. Така конструкція служить і полегшує управління рухом транспортного засобу під час обертання. Крім того, він дозволяє роботу рухатися по круговим траєкторіям.
3. Приводні двигуни і вимоги до споживання енергії: Система робота призначена для виконання інших допоміжних і додаткових функцій, таких як моніторинг і передача даних. Тому споживання енергії є

проблемою; модуль повинен споживати стільки енергії скільки потрібно для керування всіма елементами системи. Вимоги до живлення як правило, повинні бути для руху на високих і низьких швидкостях, та з нормальними швидкостями руху, близькими до 2 км/ч. Робот призначений для адаптації низьких і високих швидкостей. У разі, якщо оператор має намір збільшити швидкість, напруга моторного приводу збільшується на 3 В. Максимальна напруга моторного приводу становить 12 В, а мінімальне-6 В згідно. Що стосується розподілу енергії батареї, показано, як енергія батарей розподіляється на інші компоненти системи.

Система робота, або ж роботизована композиція, побудована на акриловій пластині з розмірами 22 см в довжину, 15 см в ширину і 4 мм см в товщину. 4-колісні приводи (двигуни постійного струму) шириною 1.5 см і довжиною 5 см закріплені в нижній частині цієї пластини разом з їх пластиковим шасі. Решта компонентів розташовані поверх цієї пластини.

Збірка системи розпочинається з складання блоку руху зображено нижче (Рисунок 2.2).

- Першим кроком необхідно закріпити винт в двигуні постійного току - Далі додаємо кронштейн до двигуна та зажимаємо гайкою
- Наступним кроком закріпимо двигун постійного струму до платформи - Приєднуємо шасі до двигуна

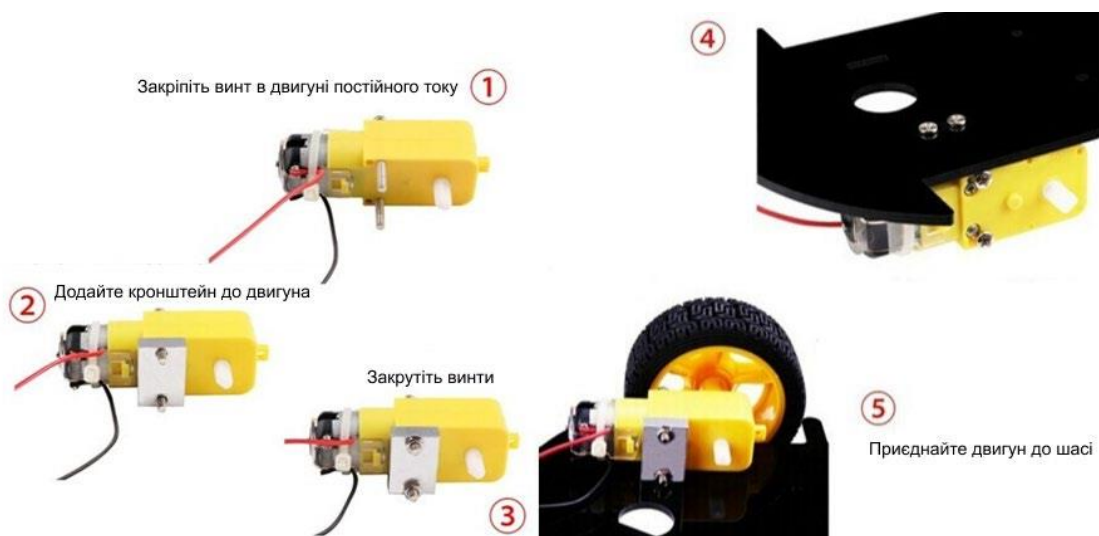


Рис. 2.2. Збірка двигунів і платформи

В даному розділу було розглянуто основні елементи модуля, блок руху, блок управління, чутливий блок, блок бездротового зв'язку. Ці елементи забезпечать технічні завдання системи, для виконання поставлених цілей. Далі було розглянуто більш детально підбір, кількість та характеристики елементів для кожного блоку. Описані елементи та методики є найбільш відповідними та сумісними, для розробленої системи. Слід враховувати що кожна комплектуюча потребує відповідних технічних знань по експлуатації, тому перед тим як перейти до безпосередньої реалізації проекту потрібно ознайомитись з усіма аспектами, відповідних деталей, за для уникнення їх пошкодження, всі матеріали доступні на офіційних сайтах виробників. Використання бібліотек Arduino, також є значною перевагою це допомагає в розробці більш обширних систем за можливостями, та дозволяє уникнути написання відповідного програмного коду.

РОЗДІЛ 3

РОЗРОБКА WEB-ДОДАТКУ АНАЛІЗУ ЗБИТКІВ ВІД ТЕХНОГЕННИХ ТА ПРИРОДНИХ КАТАСТРОФ

3.1 Архітектура програмного додатку

Набір команд, який використовується для керування рухом, обмежений: вперед, назад і поворот. Крім того, встановлюється з'єднання з камерою для передачі живого відеопотоку на екран операторського пульта. Клієнтський додаток (пристрій на ОС Android) підключається до плати управління по бездротовій мережі Bluetooth. Для використання передачі даних між виконувачим елементом робота і керуючим пристроєм використовується послідовний порт зв'язку

Підключення модуля Bluetooth та FPV камери зображено на (Рисунок 3.1) Піни на модулі потрібно з'єднати як показано на схемі, а камеру підключити через конденсатор на 470 uF, це дозволить запобігти пошкодженню камери від напруги.

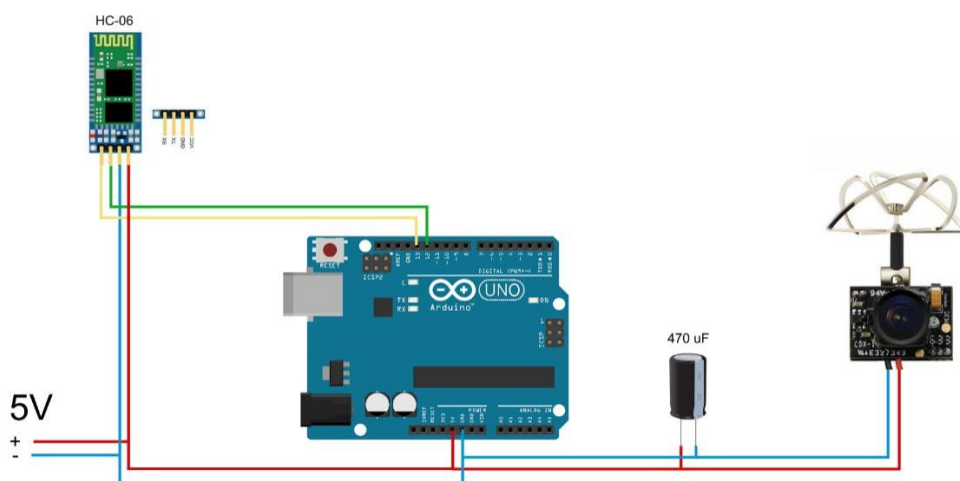


Рис. 3.1. Схема підключення Bluetooth та FPV камери

Що стосується датчиків, то модуль поставляється з двома типами датчиків, перший датчик називається датчиком випромінювання, а другий-лічильником Гейгера. Перший виявляє рівні ядерного випромінювання (тобто Альфа (α), бета (β) і гамма (γ)). Останній передає значення рівня радіаційного випромінювання.

Як тільки робот починає працювати, і в разі, якщо виявлене значення випромінювання виявляється більше допустимого стандартного значення (80 відліків в хвилину (CPM) на лічильнику Гейгера), активується GPS-сервіс для

визначення місця витоку випромінювання і поточного місця розташування робота. Місце розташування географічно описується значеннями координат широти і довготи. Крім того, плата GSM активується для визначення Google Map певного місця розташування і відправки SMS-повідомлення на заздалегідь визначений номер екстреного телефону. Однак значення, яке зчитується лічильником Гейгера, може перевищувати порогове значення, згадане вище, до великих значень (класифікуються як шкідливі для людини) в залежності від відстані між модулем і джерелом витоку випромінювання. Зображена на (Рисунок 3.2) схема демонструє підключення плат розширення до мікроконтролера Arduino

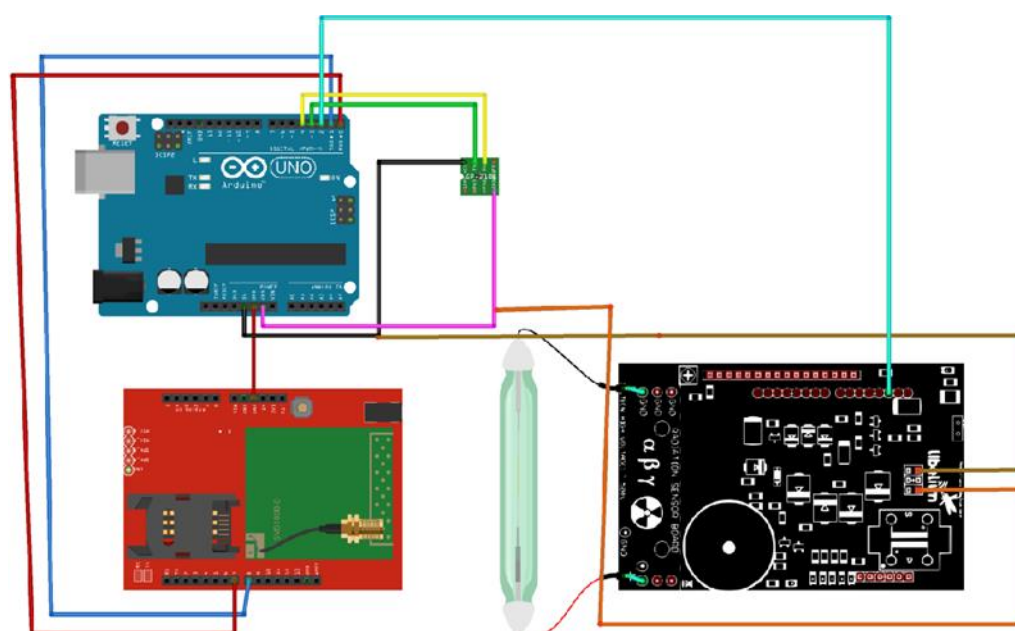


Рис. 3.2. Схема для підключення датчиків до плати GPS, GSM і плати Arduino.

Алгоритм роботи мікроконтролерної системи контролю радіаційного фону (Рисунок 3.3). При запуску програмного забезпечення мікроконтролера (пз) він виконує свою початкову настройку і ініціалізацію параметрів дозиметра-радіометра, послідовного інтерфейсу і/портів введення-виводу плати Arduino Uno. Наступний крок про систему стримувань в батареї напругою живлення. Якщо напруга нормально перевірена на аноді напруги через трубки, на СБМ-20 лічильник Гейгера-Мюллера в таймераі дає дію почати. В системі зараховується кількість імпульсів за 1 хв. Потім, в системі процесів на результати вимірювань і виводить їх на дисплей через послідовний інтерфейс на ПК. Система виводить чення потужності

радіоактивного випромінювання в $\mu\text{мкЗв} / \text{год}$ і кількість імпульсів в хвилину.

Кожну хвилину таймер скидається і перезапускається для наступного вимірювання. При перевищенні допустимої потужності випромінювання система, що має звукову сигналізацію, включає сигнальний світлодіод. При "Start/першому натисканні кнопки" Пуск / Стоп " система переходить в режим дозиметра. У режимі дозиметра починає вимірювати отриману дозу опромінення. Тоді за допомогою GSM модулю передаються значення дози опромінення. При повторному натисканні кнопки "Пуск / Стоп" режим дозиметра відключається, і система повертається в режим радіометра.

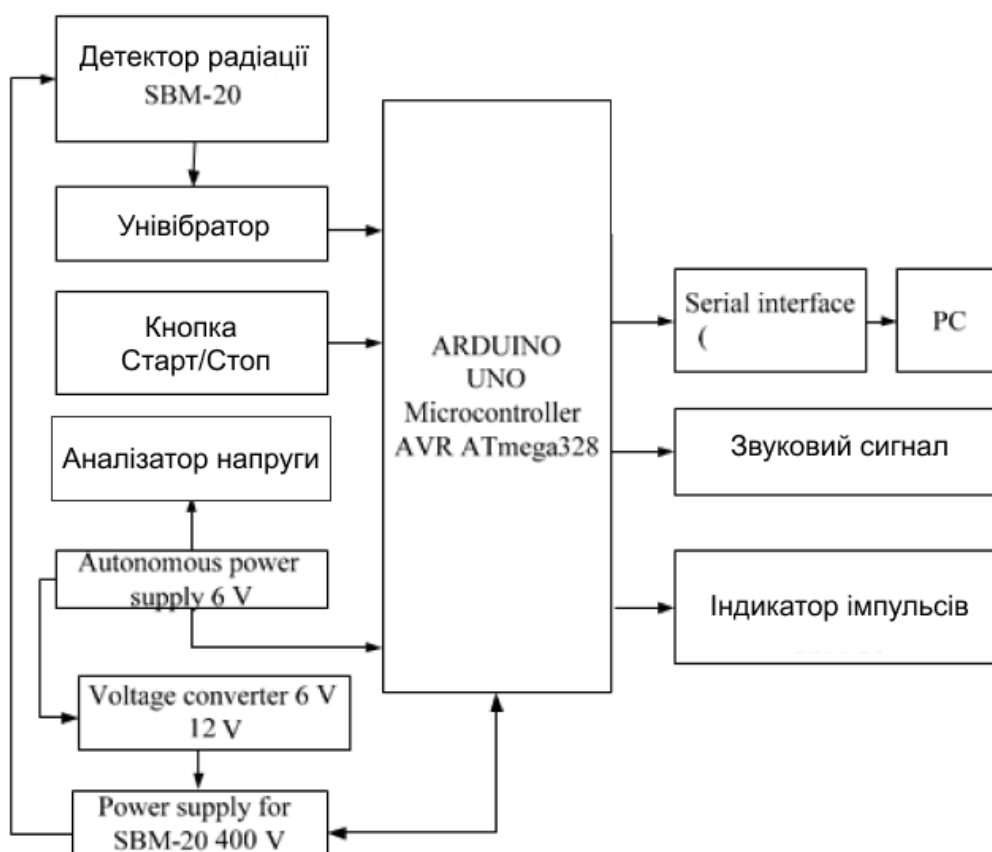


Рис. 3.3. Структурна схема мікроконтролерного радіаційного моніторингу, побудованого на платі Arduino UNO

Приведена схема (Рисунок 3.4) відображає алгоритм роботи мікроконтролерної системи контролю радіаційного фону, на схемі ми можемо бачити основний принцип роботи, та аналізу, пристрою в радіаційному середовищі

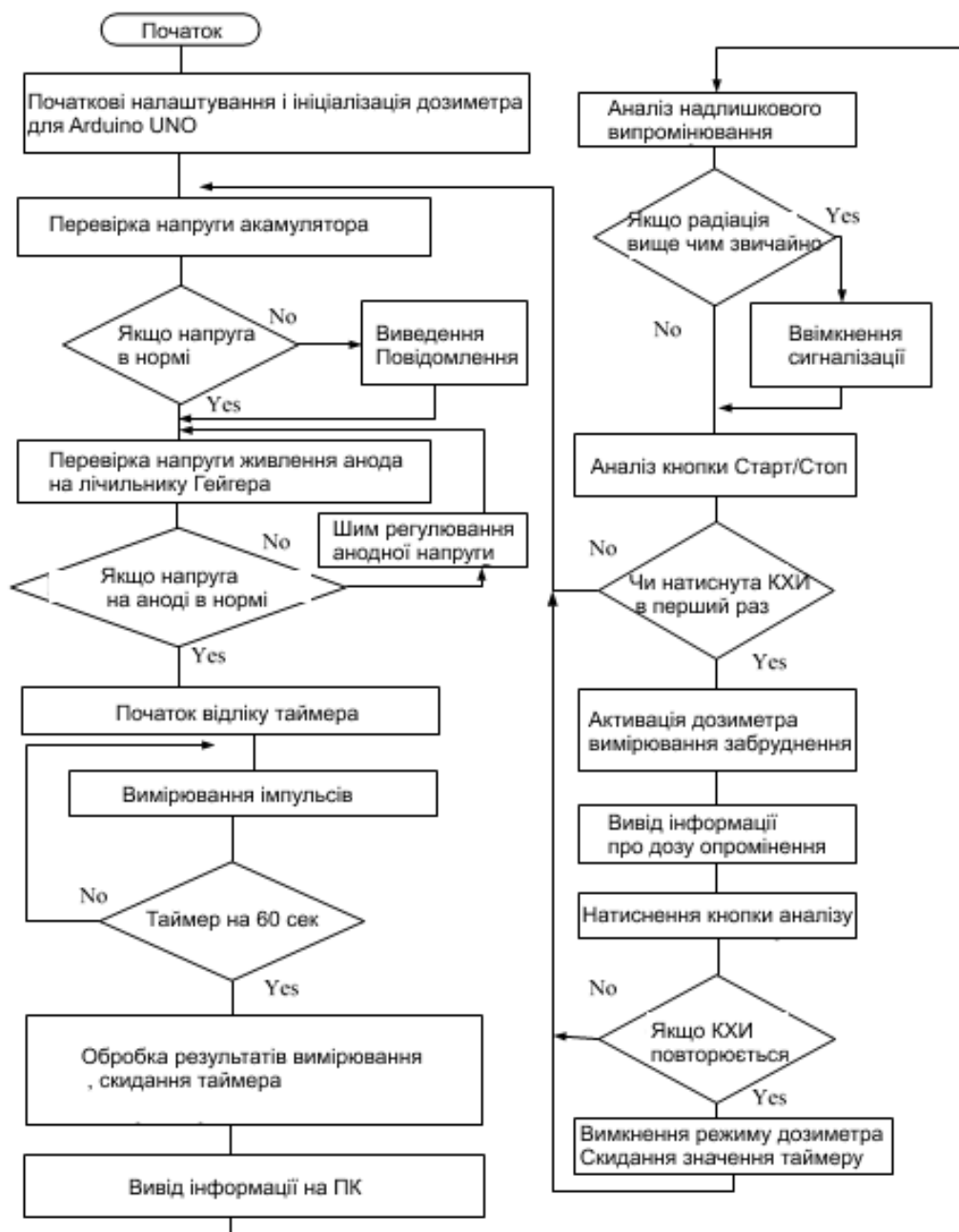


Рис. 3.4. Алгоритм роботи мікроконтролерної системи контролю радіаційного фону

Точність вимірювання залежить від ряду налаштувань компонентів проєктованої системи. Зокрема, на рис Далі представлена структурна схема алгоритму налаштування анодної напруги лічильника Гейгера-Мюллера (Рисунок 3.5), що є одним з основних компонентів системи.

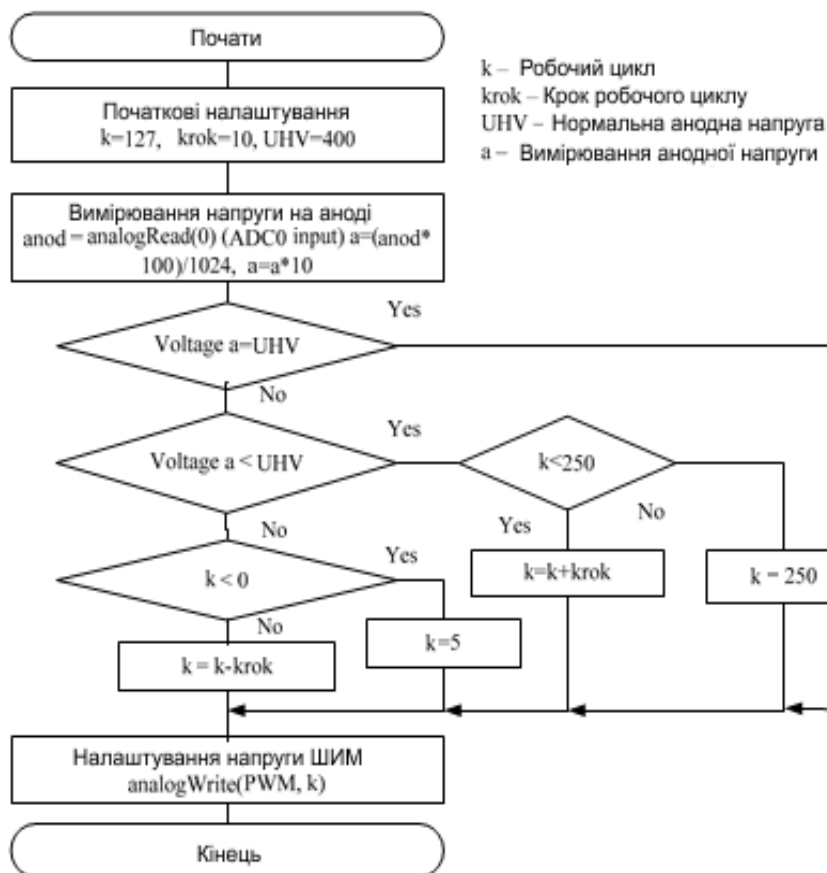


Рис. 3.5. Алгоритм регулювання анодної напруги лічильника Гейгера-Мюллера

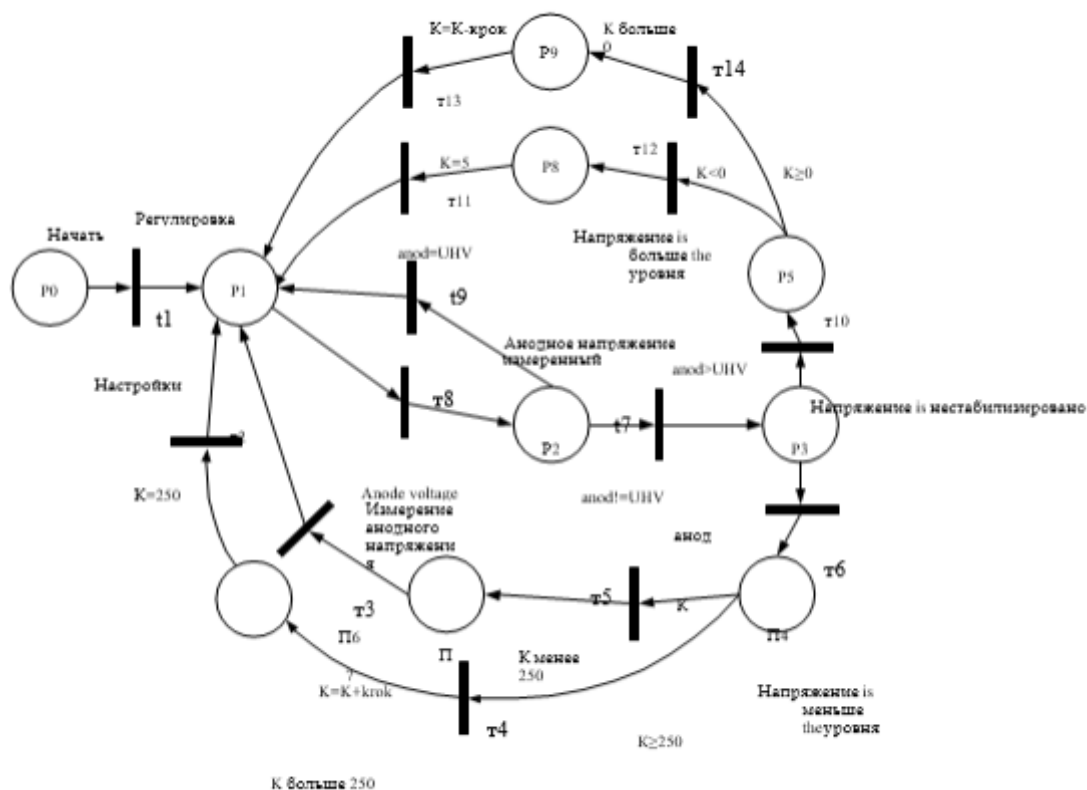


Рис. 3.6. Схематичний вигляд моделі на основі мереж Петрі для дослідження динаміки процесу регулювання анодної напруги лічильника Гейгера-Мюллера

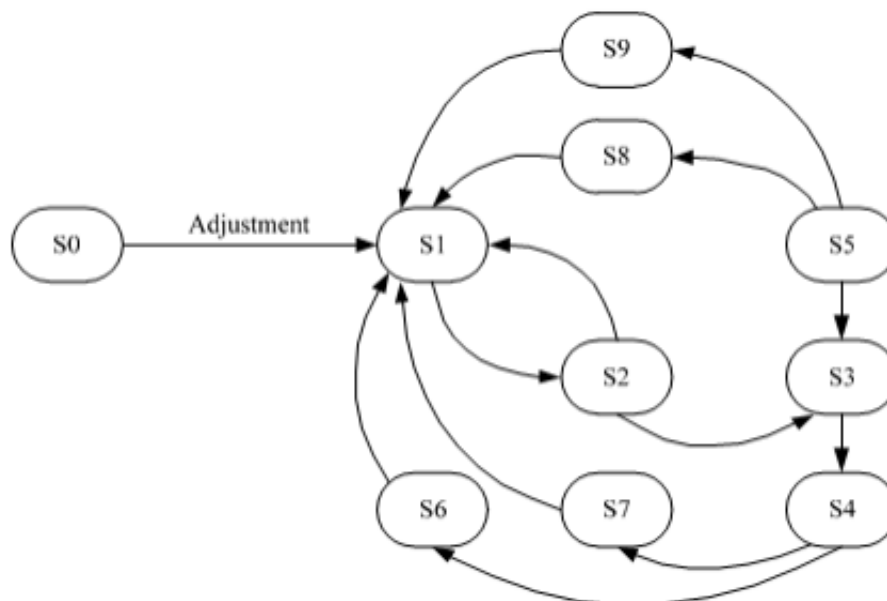


Рис. 3.7. Графік досяжності станів для дослідження динаміки процесу налаштування анодної напруги лічильника Гейгера-Мюллера

3.2. Вбудована програма Arduino

По суті, вся вбудована програма, що працює на Arduino, - це просто великий кінцевий автомат. Дві основні завдання вбудованої програми - це робота з лічильником Гейгера, а також зчитування даних з модуля Bluetooth і відповідна обробка цих даних.

Як згадувалося раніше, всі програми Arduino складаються з функції налаштування, яка запускається один раз, і основного циклу, який запускається нескінченно після налаштування. У цій конкретній програмі функція налаштування керує ініціалізацією різних компонентів, які разом складають роботу. Наприклад, він запускає послідовне з'єднання з лічильником Гейгера, GPS модулем і модулем Bluetooth, встановлює правильну налаштування різних компонентів і направляє двигуни в правильне вихідне положення, щоб робот наприклад міг повернутись обличчям вперед.

Основний цикл перевіряє і зчитує байти в послідовному буфері модуля Bluetooth. Він також керує елементом для вимірювання радіаційного забруднення. Кількість часу, протягом якого повинно відбуватись замірювання. Розраховується також команди отримані з модуля Bluetooth.

Так як Arduino немає нічого схожого на операційну систему, у нього є концепція, схожа на потоки. Через це ми використовуємо систему таймера, де на кожній

ітерації отримуємо поточний час з моменту запуску мікроконтролера з функцією `Millis` стандартної бібліотеки `Arduino`. Потім для кожного окремого таймера ми обчислюємо час, що минув з моменту останньої дії. Якщо цей час досягає певного порогу, ми виконуємо дію і скидаємо таймер. По суті, це дозволяє нам імітувати в `Arduino`, що виконує кілька дій кожні X мілісекунд. Загалом, у робота є три можливих стани. Ці стани-руху, вимірювання і спокою. За винятком `NOTHING-state`, ці стани безпосередньо відповідають за різних види команд від `Android application`, які може відправляти - додаток. Стан спокою не означає, що програма в даний момент простоює і чекає подальших команд від `Android-Додатка`.

Якщо робот знаходиться в моторному стані, це означає, що наступні дані, які він отримає, будуть вказувати різні швидкості і стану, в яких він повинен переміщати свої колеса і сервоприводи. Аналогічно, якщо робот знаходиться в стані вимірювання, це означає, що всі вхідні дані повинні бути передані на ПК після того як закінчиться робота таймера, що відповідає за вимірювання радіаційного фону. Дані отримані комп'ютером повинні отримуватись через `GPS/GSM` плату.

Кожна команда, відправлена на мікроконтролер, відправляється і зчитується у вигляді масивів байтів, як було описано в попередньому розділі. Для всіх намірів і цілей в контексті цього додатка а байт можна розглядати як числове значення між 0 і 255.

Однак, оскільки байти в мові `Java` призначені для того, щоб бути від -128 до 127, а байти в `C++` від 0 до 255, було виявлено, що вони не можуть бути передані безпосередньо. Щоб подолати цю проблему, значення, що надсилаються з програми `Android`, становлять від 0 до 127. Вони помножуються на 2 на стороні `Arduino`, щоб отримати правильні значення. і роботом.

Перше число-це простий роздільник, який дозволяє мікроконтролеру знати, що наступні п'ять символів повинні інтерпретуватися як команда руху, а не як що-небудь інше. На відміну від інших команд, немає необхідності в кінцевому роздільнику так як сам Мікроконтролер знає, що довжина команди робота становить всього шість байт.

Наступний байт вказує напрямок, в якому повинен рухатися самий лівий двигун. Цей байт відповідає або символу `ASCII F` для прямого, зворотного або

(82) для звільнення, що означає, що колесо повинно залишатися на місці. Третій байт-це швидкість, яку повинен розвивати самий лівий двигун, від 0 до 127. У цьому прикладі лівому двигуну наказано обертатися вперед на половині своєї максимальної швидкості.

Четверте і п'яте числа слідує тій же логіці, що і останні два, за винятком того, що вони для правого двигуна. У цьому випадку дві цифри говорять двигуну почати рух назад на повній швидкості.

Останній символ пов'язаний з позиціонуванням двох сервоприводів. Це число від 0 до 9, яке вказує двом сервоприводів, в якому напрямку вони повинні рухатися. Цей рух відбувається щодо їх поточного положення. Можна уявити їх як вісім сторін світу плюс ап додаткове значення, яке означає, що обидва сервоприводу повинні просто залишатися нерухомим.

Після планування та розробки кінцеве мобільне додаток, що буде використовуватися для керування самим роботом, має досить простим за своєю функціональністю.

Сам додаток складається всього з двох основних видів діяльності. У всіх сенсах і цілях в контексті програмування Android можна думати про діяльність як про різних "екрани", що складають цілий мобільний додаток. Кожна дія має окремий файл макета на основі XML і свій власний пов'язаний файл Java, який містить всю логіку і код, пов'язані з розглянутим дією .

Коли користувач запускає додаток, його спочатку зустрічає екран налаштувань. Після того як користувач з'єднав свій телефон з модулем Bluetooth робота, він може перейти до водіння автомобіля. Тут можна керувати роботом за допомогою інтерфейсу джойстика.

Одна з цілей нової версії Програми для Андроїд полягала в тому, щоб його можна було використовувати на планшетах Андроїд. Розробка на планшетних пристроях дуже схожа на розробку на мобільних телефонах. Єдиним серйозним винятком є загальна відсутність функцій, дуже характерних для мобільних телефонів, таких як телефонний дзвінок або відправка СМС. Більш важливим для цього проекту, зокрема, є той факт, що Планшети часто мають набагато більші екрани. Замість звичайних пікселів, в контексті Програмування на Android, розмір екрану частіше представлений в незалежних від щільності пікселях, або ДП для

стислості. Один ДП офіційно був визначений як еквівалент одного фізичного пікселя на 160 на екрані з роздільною здатністю 160 точок на дюйм. (Android.com).

Щоб допомогти розробникам забезпечити послідовний і приємний користувацький досвід незалежно від розміру їх пристрою, Android пропонує безліч способів. Для максимального контролю ви можете надати окремі файли макета для різних розмірів екрану. На практиці потрібно просто створити нову підпапку в папці ресурсів свого проекту андроїда і назвати її відповідним чином. Наприклад, якщо ви хочете спеціально орієнтуватися на великі планшетні пристрої, вони можуть створити папку під назвою "макет-планшета" і помістити туди свої файли макета певного розміру файли.

У невеликих додатках, таких як це, надання різних файлів макета для різних пристроїв насправді не є необхідним. При ретельному плануванні і стилізації елементів можна створювати макети, які добре працюють як на мобільних, так і на планшетних пристроях. Крім використання вищезгаданих незалежних від щільності пікселів для вказівки ширини і висоти графічних елементів, Android API також пропонує способи визначення розміру щодо елемент батьківського контейнера елемента. Для наприкладу ширина з двох кнопок написи "знайти пристрої" і "Вперед!" визначаються значенням "fill_parent". Це призводить до того, що кнопки стають настільки ж широкими, як і їх батьківський елемент, який в даному випадку є самою активністю. Через це кнопки виглядають абсолютно однаково незалежно від розміру екрану.

3.3. Програмування додатку.

Основна функціональність коду, що лежить в основі дії налаштувань, полягає в тому, щоб просто тримати вкладки з різними типами вхідних полів, які використовуються для налаштування іншої частини програми. Ці налаштування зберігаються і завантажуються з SharedPreferences Android, що є простим способом то зберігання значень in між application сеансами програми.

Іншою основною функціональністю цієї конкретної діяльності, яка заслуговує на згадку, є сполучення Bluetooth пристроїв, яке обробляється в цьому поданні. користувач може натиснути кнопку, щоб ініціювати пошук нових,

невдомих пристроїв. Якщо вони знайдені, вони додаються до кінця списку. Натискання на елемент у списку ініціює сполучення з пристроєм Bluetooth. Потім пристрій Bluetooth з'єднується з телефоном, і вихідний потік передається в BluetoothStreamManager, і користувач може перейти до потокової активності.

Робота BluetoothStreamManager полягає в забезпеченні правильної передачі даних роботу. Він пропонує простий інтерфейс для спільних дій, пов'язаних з Bluetooth, таких як підключення до пристрою, отримання міцності і стану з'єднання Bluetooth і передача даних. Інші дії програми можуть потім використовувати цей клас для відправки команд на Arduino, які потім захоплюються на виході пристрою.

BluetoothGattService-це клас, який представляє службу GATT. GATT-це аббревіатура, що позначає профіль універсального атрибута. Служба Bluetooth Gatt містить один або кілька зумовлених форматів даних, які називаються характеристиками. Кожна служба ідентифікується унікальним ідентифікатором, званним UUID. У цьому випадку ми просто хочемо записати дані в модуль Bluetooth LE, тому ми отримуємо відповідну характеристику за зумовленим UUID, встановлюємо її корисне навантаження в байтах для відправки, а потім записуємо характеристику в модуль Bluetooth.

Був створений досить великий список офіційних стандартизованих послуг Bluetooth онлайн, від відстеження артеріального тиску до складу тіла. Ці характеристики можна знайти на офіційному порталі розробників Bluetooth. Важливо відзначити, що така специфікація не існує для серійного UART з будь-якої причини. Існує написаний власний неофіційний користувальницький сервер UART GATT, який працює на Arduino і може використовуватися для простого послідовного зв'язку.

BluetoothStreamManager також містить посилання на поточну активність, тому він може відображати діалогове вікно помилки, якщо він втрачає з'єднання з модулем Bluetooth. Це діалогове вікно пропонує користувачеві повернутися до дії Налаштування і спробувати відновити пристрій.

В даному додатку окремий потік буде передавати а потім записувати дані з черги в вихідний потік. Вся ця система черг може бути досить перевантажена для цього конкретного випадку використання, так як насправді між різними діями

передавалося не так багато даних.

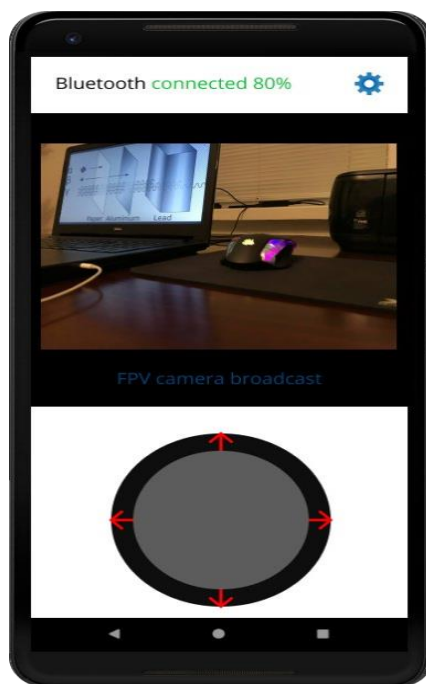


Рис. 3.8. Макет програмного додатку для Android

Як видно на зображенні нижче більша частина активності припадає на відеопотік. Внизу розташований елемент управління роботом, а в верхній частині ми бачимо кнопку налаштувань. Чим далі від центру джойстика ви можете потягнути, тим швидше буде рухатися робот.

Є також простий індикатор з'єднання в лівому верхньому кутку для міцності з'єднання Bluetooth. Рівень сигналу надходить від модуля у вигляді RSSI (received signal strength indicator), який потім обчислюється і відображається користувачеві у вигляді більш інтуїтивного процентного значення.

Найбільш важливим завданням логіки, що лежить в основі активності каналу, є відправка даних в Arduino кожні X мілісекунд, де X-значення швидкості оновлення, вказане на екрані налаштувань. Дані передаються у вигляді масиву байтів, причому потім інтерпретуються з боку вбудованої програми. Більшу частину часу цей байтовий масив містить значення, які говорять роботіві, що він повинен робити з точки зору руху. Джойстик на екрані використовують для отримання користувальницького введення, з якого потім обчислюються значення руху для двигунів.

Кнопка налаштувань, яка відкриває діалогове вікно, в якому користувач

може. Обрати потрібні налаштування. Вона також містить реалізацію класу `AsyncTask`, який використовується для асинхронного завантаження відеопотоку з веб-камери, сервера, за умови, що користувач раніше включив цю конкретну функцію в дії налаштування.

Компонента джойстика в нижній частині екрана має функцію, оновлення руху у вигляді джойстика, і ви можете вибирати кут, потужність і напрямок керуючи ним. Цей конкретний компонент спочатку розроблений, з деякими модифікаціями, зробленими, щоб забезпечити найкращу підгонку під проект. Ці зміни в основному пов'язані зі швидкістю оновлення подій джойстика.

ВИСНОВКИ

Початкові вимоги, як функціональні, так і нефункціональні, були дещо складними для цього проекту; однак, враховуючи всі обставини, кінцевим результатом цієї дипломної роботи є виконання початкового завдання. Розробка самого додатка і вбудованого коду дали змогу розібратись в деяких аспектах як програмування так і конструювання. Тако ж за темою проекту слід зауважити що було опрацьовано великий об'єм інформації стосовно пристроїв і засобів радіаційного вимірювання. Багато в чому це було пов'язано з тим, що більша частина роботи на стадії планування, уже була знайомою.

Найбільш значні труднощі виникли з використанням Bluetooth на стороні Android і розробка самого додатку, оскільки, не було достатньо досвіду для роботи в екосистемі Android. Це призвело до того, що потрібно було переглянути велику кількість документації, прикладів і тому подібного. Однак, врешті-решт, це не було великою проблемою.

Використання Bluetooth для зв'язку між додатком і роботом було цікавим досвідом. Однак Bluetooth в основному призначений для пристроїв які вимагають передачу даних на хост-пристрій відносно рідко, що не відноситься до даного конкретного проекту, оскільки він повинен передавати дані кожні 200 мілісекунд або близько того, щоб підтримувати прийнятну затримку з точки зору користувача. Крім того, як впливає з назви "Low Energy", Bluetooth призначений для вбудованих пристроїв з досить обмежувальними вимогами до живлення, таких як серцеві трекери та інші носяться технології. Споживана потужність модуля насправді не є проблемою в цьому випадку, так як різниця між Bluetooth Classic і Bluetooth все ще вимірюється десятками міліампер, які насправді не мають помітної різниці, наприклад, в порівнянні зі світлодіодами. Існує також питання вартості і простоти реалізації, так як модулі Bluetooth Classic можна придбати досить дешево в Інтернеті, і існує безліч документації по їх використанню.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. In: Proceedings of the 26th Conference on Neural Information Processing Systems; 2012 Dec 3–6; Lake Tahoe, NV, USA; 2012. p. 1097–105.
2. Cho K, van Merriënboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation. 2014. arXiv:1406.1078.
3. Silver D, Huang A, Maddison CJ, Guez A, Sifre L, van den Driessche G, et al. Mastering the game of Go with deep neural networks and tree search. Nature 2016;529(7587):484–9.
4. <https://science.house.gov/imo/media/doc/Schmidt%20Testimony%20Attachment.pdf>
5. https://keenlab.tencent.com/en/whitepapers/Experimental_Security_Research_of_Tesla_Autopilot.pdf
6. <https://www.science.org/doi/10.1126/science.aaw4399> 7. <https://arxiv.org/pdf/1707.08945.pdf>
8. <https://arxiv.org/abs/1812.00151>
9. <https://arxiv.org/pdf/2004.00622.pdf>
10. <https://thenextweb.com/news/adversarial-attacks-are-a-ticking-time-bomb-but-no-one-cares-syndication>
11. <https://venturebeat.com/2020/04/30/resistant-ai-raises-2-75-million-to-protect-algorithms-from-adversarial-attacks/>
12. <https://arxiv.org/pdf/1909.11764.pdf>
13. <https://www.microsoft.com/security/blog/2020/10/22/cyberattacks-against-machine-learning-systems-are-more-common-than-you-think/>
14. C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, “Intriguing properties of neural networks,” arXiv preprint arXiv:1312.6199, 2013.
15. I. J. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and harnessing adversarial examples (2014),” arXiv preprint arXiv:1412.6572.
16. X. Yuan, P. He, Q. Zhu, R. R. Bhat, and X. Li, “Adversarial examples: Attacks and

defenses for deep learning,” arXiv preprint arXiv:1712.07107, 2017.

- 17.A. Kurakin, I. Goodfellow, and S. Bengio, “Adversarial examples in the physical world,” arXiv preprint arXiv:1607.02533, 2016.
- 18.S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, “Deepfool: a simple and accurate method to fool deep neural networks,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2574–2582, 2016.
- 19.U. Jang, X. Wu, and S. Jha, “Objective metrics and gradient descent algorithms for adversarial examples in machine learning,” in Proceedings of the 33rd Annual Computer Security Applications Conference, pp. 262–277, 2017.
- 20.N. Carlini and D. Wagner, “Towards evaluating the robustness of neural networks,” in 2017 IEEE Symposium on Security and Privacy (SP), pp. 39–57, IEEE, 2017.
- 21.A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, “Towards deep learning models resistant to adversarial attacks,” arXiv preprint arXiv:1706.06083, 2017.
- 22.F. Croce and M. Hein, “Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks,” arXiv preprint arXiv:2003.01690, 2020.
- 23.S. Sabour, Y. Cao, F. Faghri, and D. J. Fleet, “Adversarial manipulation of deep representations,” arXiv preprint arXiv:1511.05122, 2015.
24. Praveen Balda et al, *International Journal of Computer Science and Mobile Computing*, Vol.4 Issue.4, April 2015, pg. 761-767
25. Verissimo, Paulo; Rodrigues, Luis. *Distributed systems for system architects*. Springer Science & Business Media, 2012.
26. Fleischmann, Albert. *Distributed systems: software design and implementation*. Springer Science & Business Media, 2012.

Додаток А**Лістинг програми**

Лістинг програмного коду для мікроконтролера Arduino

```
/*===== Налаштування =====*/

#define MOTOR_MAX 255

#define JOY_MAX 40

#define minDuty 0

#define RIGHT_MOTOR_DIRECTION NORMAL

#define LEFT_MOTOR_DIRECTION NORMAL

#define RIGHT_MOTOR_MODE HIGH

#define LEFT_MOTOR_MODE HIGH

/*===== Піни =====*/

#define RIGHT_MOTOR_D 10

#define RIGHT_MOTOR_PWM 11

#define LEFT_MOTOR_D 2

#define LEFT_MOTOR_PWM 3

#define BT_TX 13

#define BT_RX 12

#include "GyverMotor.h"

GMotor motorR(DRIVER2WIRE, RIGHT_MOTOR_D, RIGHT_MOTOR_PWM,
RIGHT_MOTOR_MODE);
GMotor motorL(DRIVER2WIRE, LEFT_MOTOR_D, LEFT_MOTOR_PWM,
LEFT_MOTOR_MODE);
```



```
#include <SoftwareSerial.h>

SoftwareSerial BTserial(BT_TX, BT_RX);

boolean doneParsing, startParsing;

int dataX, dataY;

String string_convert;

void setup() {

  #if (LEFT_MOTOR_PWM == 3 || LEFT_MOTOR_PWM == 11 ||
RIGHT_MOTOR_PWM == 3 || RIGHT_MOTOR_PWM == 11)

    TCCR2B = 0b00000001;

    TCCR2A = 0b00000011;

  #elif (LEFT_MOTOR_PWM == 9 || LEFT_MOTOR_PWM == 10 ||
RIGHT_MOTOR_PWM == 9 || RIGHT_MOTOR_PWM == 10)

    TCCR1A = 0b00000001;

    TCCR1B = 0b00001001;

  #endif

  BTserial.begin(9600);

  motorL.setMode(AUTO);

  motorR.setMinDuty(minDuty);

  motorL.setMinDuty(minDuty);

  motorR.setDirection(RIGHT_MOTOR_DIRECTION);

  motorL.setDirection(LEFT_MOTOR_DIRECTION);
```

```
}
```

```
void loop() {
```

```
  parsing();
```

```
  if (doneParsing) {
```

```
    doneParsing = false;
```

```
    int joystickX = map((dataX), -JOY_MAX, JOY_MAX, -MOTOR_MAX / 2,  
MOTOR_MAX / 2);
```

```
    int joystickY = map((dataY), -JOY_MAX, JOY_MAX, -MOTOR_MAX,  
MOTOR_MAX);
```

```
    int dutyR = joystickY + joystickX;
```

```
    int dutyL = joystickY - joystickX;
```

```
    motorL.setMode(AUTO);
```

```
    motorR.setMinDuty(minDuty);
```

```
    motorL.setMinDuty(minDuty);
```

```
    motorR.setDirection(RIGHT_MOTOR_DIRECTION);
```

```
    motorL.setDirection(LEFT_MOTOR_DIRECTION);
```

```
}
```

```
void loop() {
```

```
  parsing();
```

```
  if (doneParsing) {
```

```
doneParsing = false;
```

```
int joystickX = map((dataX), -JOY_MAX, JOY_MAX, -MOTOR_MAX / 2,  
MOTOR_MAX / 2);
```

```
int joystickY = map((dataY), -JOY_MAX, JOY_MAX, -MOTOR_MAX,  
MOTOR_MAX);
```

```
int dutyR = joystickY + joystickX;
```

```
int dutyL = joystickY - joystickX;
```

```
motorL.setMode(AUTO);
```

```
motorR.setMinDuty(minDuty);
```

```
motorL.setMinDuty(minDuty);
```

```
motorR.setDirection(RIGHT_MOTOR_DIRECTION);
```

```
motorL.setDirection(LEFT_MOTOR_DIRECTION);
```

```
}
```

```
void loop() {
```

```
  parsing();
```

```
  if (doneParsing) {
```

```
    doneParsing = false;
```

```
    int joystickX = map((dataX), -JOY_MAX, JOY_MAX, -MOTOR_MAX / 2,  
MOTOR_MAX / 2);
```

```
    int joystickY = map((dataY), -JOY_MAX, JOY_MAX, -MOTOR_MAX,  
MOTOR_MAX);
```

```
int dutyR = joystickY + joystickX;

int dutyL = joystickY - joystickX;

motorL.setMode(AUTO);

motorR.setMinDuty(minDuty);

motorL.setMinDuty(minDuty);

motorR.setDirection(RIGHT_MOTOR_DIRECTION);

motorL.setDirection(LEFT_MOTOR_DIRECTION);

}

void loop() {

  parsing();

  if (doneParsing) {

    doneParsing = false;

    int joystickX = map((dataX), -JOY_MAX, JOY_MAX, -MOTOR_MAX / 2,
MOTOR_MAX / 2);

    int joystickY = map((dataY), -JOY_MAX, JOY_MAX, -MOTOR_MAX,
MOTOR_MAX);

    int dutyR = joystickY + joystickX;

    int dutyL = joystickY - joystickX;

    motorL.setMode(AUTO);

    motorR.setMinDuty(minDuty);
```

```
motorL.setMinDuty(minDuty);

motorR.setDirection(RIGHT_MOTOR_DIRECTION);

motorL.setDirection(LEFT_MOTOR_DIRECTION);

}

void loop() {

  parsing();

  if (doneParsing) {

    doneParsing = false;

    int joystickX = map((dataX), -JOY_MAX, JOY_MAX, -MOTOR_MAX / 2,
MOTOR_MAX / 2);

    int joystickY = map((dataY), -JOY_MAX, JOY_MAX, -MOTOR_MAX,
MOTOR_MAX);

    int dutyR = joystickY + joystickX;

    int dutyL = joystickY - joystickX;
```

Лістинг програмного коду бібліотеки motor_control

```
#define DEADZONE 60

#define IN1 2

#define IN2 3

#define POT 0

#include "GyverMotor.h"

GMotor motor1(IN1, IN2);
```

```
int left_min = 512 - DEADZONE;

int right_min = 512 + DEADZONE;

void setup() {

}

void loop() {

    int potent = analogRead(POT);

    motor1.setMode(STOP);

} else if (potent > right_min) {

    motor1.setMode(FORWARD);

    byte duty = map(potent, right_min, 1023, 0, 255);

    motor1.setSpeed(duty);

} else if (potent < left_min) {

    motor1.setMode(BACKWARD);

    byte duty = map(potent, left_min, 0, 0, 255);

    motor1.setSpeed(duty);
```

Лістинг програмного коду для плати розширення Лічильника Гейгера

```
#include <LiquidCrystal_I2C.h>

#include <Wire.h>

unsigned long counts; //variable for GM Tube events

unsigned long previousMillis; //variable for measuring time

float averageCPM;

float sdCPM;
```

```
int currentCPM;

float calcCPM;

LiquidCrystal_I2C lcd(0x27, 20, 4);

float CPMArray[100];

#define LOG_PERIOD 30000 // count rate (in milliseconds)

void setup() { //setup

    counts = 0;

    currentCPM = 0;

    averageCPM = 0;

    calcCPM = 0;

    lcd.init();

    lcd.backlight();

    Serial.begin(9600);

    pinMode(2, INPUT);

    attachInterrupt(digitalPinToInterrupt(2), impulse, FALLING); //define external interrupts
}

void loop() { //main cycle

    lcd.setCursor(0,2);

    lcd.print("CPM Count: ");

    lcd.print(counts);

    unsigned long currentMillis = millis();

    if (currentMillis - previousMillis > LOG_PERIOD) {

        previousMillis = currentMillis;
```

```
CPMArray[currentCPM] = counts * 2;

lcd.clear();

lcd.setCursor(0,0);

lcd.print("uSv/hr: ");

lcd.print(outputSieverts(CPMArray[currentCPM]));

counts = 0;

calcCPM = 0;

lcd.init();

lcd.backlight();

Serial.begin(9600);

pinMode(2, INPUT);

attachInterrupt(digitalPinToInterrupt(2), impulse, FALLING); //define external interrupts
}

void loop() { //main cycle

  lcd.setCursor(0,2);

  lcd.print("CPM Count: ");

  lcd.print(counts);

  unsigned long currentMillis = millis();

  if (currentMillis - previousMillis > LOG_PERIOD) {

    previousMillis = currentMillis;

    CPMArray[currentCPM] = counts * 2;

    lcd.clear();

    lcd.setCursor(0,0);
```



```
lcd.print("uSv/hr: ");

lcd.print(outputSieverts(CPMArray[currentCPM]));

counts = 0;
calcCPM = 0;

lcd.init();

lcd.backlight();

Serial.begin(9600);

pinMode(2, INPUT);

attachInterrupt(digitalPinToInterrupt(2), impulse, FALLING); //define external interrupts
}

void loop() { //main cycle

lcd.setCursor(0,2);

lcd.print("CPM Count: ");

lcd.print(counts);

unsigned long currentMillis = millis();

if (currentMillis - previousMillis > LOG_PERIOD) {

previousMillis = currentMillis;

CPMArray[currentCPM] = counts * 2;

lcd.clear();

lcd.setCursor(0,0);

lcd.print("uSv/hr: ");

lcd.print(outputSieverts(CPMArray[currentCPM]));

counts = 0;
calcCPM = 0;

lcd.init();
```

```
lcd.backlight();

Serial.begin(9600);

pinMode(2, INPUT);

attachInterrupt(digitalPinToInterrupt(2), impulse, FALLING); //define external interrupts
}

void loop() { //main cycle

  lcd.setCursor(0,2);

  lcd.print("CPM Count: ");

  lcd.print(counts);

  unsigned long currentMillis = millis();

  if (currentMillis - previousMillis > LOG_PERIOD) {

    previousMillis = currentMillis;

    CPMArray[currentCPM] = counts * 2;

    lcd.clear();

    lcd.setCursor(0,0);

    lcd.print("uSv/hr: ");

    lcd.print(outputSieverts(CPMArray[currentCPM]));

    counts = 0;
```

ПЕРЕЛІК ДОКУМЕНТІВ НА ОПТИЧНОМУ НОСІЇ

Ім'я файла	Опис
Пояснювальні документи	
Кваліфікаційна робота.doc	Пояснювальна записка. Документ Word.
Кваліфікаційна робота.pdf	Пояснювальна в форматі PDF
Програма	
Program.rar	Архів. Містить коди програми і откомпільовану програму
Презентація	
Презентація.ppt	Презентація