

ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ І. І. МЕЧНИКОВА

(повне найменування закладу вищої освіти)

Факультет математики, фізики та інформаційних технологій

(повне найменування факультету)

Кафедра інформаційних технологій

(повна назва кафедри)

Кваліфікаційна робота

на здобуття ступеня вищої освіти «Бакалавр»

**«Розробка системи віддаленого моніторингу за домашнім
улюбленцем за допомогою Arduino»**

(тема кваліфікаційної роботи українською мовою)

**«Development of a Remote Monitoring System for Pets Using
Arduino»**

(тема кваліфікаційної роботи англійською мовою)

Виконав: здобувач денної форми навчання
спеціальності 122 Комп'ютерні науки

(код, назва спеціальності)

Освітня програма Комп'ютерні науки

(назва)

Ткаченко Ілля Андрійович

(прізвище, ім'я, по-батькові здобувача)

Керівник асистент Гадяцький І.А.

(науковий ступінь, вчене звання, прізвище, ініціали)


(підпис)

Рецензент к.т.н., доцент Перелигін Б.В.

(науковий ступінь, вчене звання, прізвище, ініціали)

Рекомендовано до захисту:
Протокол засідання кафедри
Інформаційних технологій

№ 1 від 09 червня 2024 р.

Завідувачка кафедри


(підпис) КАЗАКОВА Надія
(прізвище, ім'я)

Захищено на засіданні ЕК № 13,
протокол № 27 від 21 червня 2024 р.

Оцінка добре / С / 80
(за національною шкалою/шкалою ECTS/ бали)

Голова ЕК


(підпис) КОПИЧЕНКО Іван
(прізвище, ім'я)

Одеса 2024

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ	5
ВСТУП	6
1.АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАВДАННЯ.....	7
1.1 Основні поняття платформи Arduino	7
1.2 Програмування для платформи Arduino	14
1.3 Аргументи на користь роботи з платформою Arduino	20
1.4 Постановка завдання.	22
2.ПОШУК СПОСОБІВ РЕАЛІЗАЦІЇ ПРОЕКТУ	25
2.1 Вибір апаратного забезпечення.....	25
2.2 Вибір програмного забезпечення.....	36
2.3 Вибір системи управління базами даних.	40
3.РОБОТА З КОМПОНЕНТАМИ	46
3.1 Побудова схеми та розробка прошивки для Arduino	46
3.2 Розробка веб-додатку	53
ВИСНОВКИ.....	55
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	56

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

GPS – Global Positioning System – американська система глобального позиціонування.

БД – база даних.

OSM – Open Street Map – відкриті карти вулиць.

АЦП – аналого-цифрового перетворювача.

ICSP – In-Circuit Serial Programming – внутрішньосхемне послідовне програмування.

ОС – операційна система.

ШИМ – широко-імпульсна модуляція.

BMS – Battery Management System – система керування акумулятором.

СУБД – система управління реляційними базами даних.

SQL – Structured Query Language – мова структурованих запитів.

PHP – Hypertext Preprocessor – препроцесор гіпертексту.

MySQLi – MySQL Improved – покращена MySQL.

PDO – PHP Data Objects – PHP об'єкти даних.

NMEA – National Marine Electronics Association – національна асоціація морської електроніки.

ВСТУП

Arduino – це проста, та дуже гнучка платформа для створення різних проектів, різної складності, як зі сторони апаратної частини, так і зі сторони програмної реалізації. Величезна кількість деталей та бібліотек дозволяє рядовому користувачу, що не має глибоких знань у програмуванні, та апаратних тонкощах, доторкнутися до процесу створення проектів з нуля. Малий поріг входу, та відносна дешевизна робить платформу Arduino самим оптимальним варіантом.

Розробляємий проект – це система віддаленого моніторингу за домашнім улюбленцем. Ця система представляє з себе моніторинг руху у реальному часі, що буде відбуватися завдяки GPS трекеру. Трекер повинен отримувати супутникові дані, обробляти їх та відправляти на веб-сервер, який буде зберігати координати у базі даних. Користувач зможе легко переглянути ці дані у веб-додатку, що буде брати актуальні дані із БД, та відображати їх у вигляді координат на мапі. GPS трекер дозволяє власникам швидко відстежувати місцезнаходження свого улюбленця у випадку, якщо він загубився, або швидко реагувати в разі крадіжки тварини. Це може врятувати час та зусилля, спротививши процес пошуку. Трекери доволі розповсюджені, та в мережі Інтернет можна зустріти велику кількість подібних пристроїв. Але, здебільшого, їх проблема полягає у великій ціні, або у невеликому діапазоні дії. Тому було вирішено розробити свій варіант пристрою, який був би за меншу вартість і у той же час задовільняв потребу у якісному моніторингу. А саме, трекер дозволить точно і швидко відображати своє місцезнаходження у реальному часі, та матиме великий діапазон дії.

Диплома робота містить в собі 58 сторінок, 23 зображень, 23 таблиць, 11 джерел посилання.

1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАВДАННЯ

1.1 Основні поняття платформи Arduino

Комп'ютери вже давно стали невід'ємною частиною нашого повсякденного та професійного життя. Та іноді ми забуваємо, що крім них, нас оточують дуже маленькі комп'ютери, і мова йде не про телефони, а про мікроконтролери. Мікроконтролерні системи зараз повсюди, вони у пральних машинах, автомобілях, холодильниках та інших повсякденних речах. Також вони часто використовуються у різних інноваційних пристроях, таких як квадрокоптер, таких систем, як «Розумний будинок» чи, певно, найцікавіших з них – роботах. Наразі існують певні системи зв'язку з цими пристроями, але більшість з них можуть використовуватись тільки напряму людиною, наприклад, натискаючи кнопки. Не вибравши мікроконтролер, не можна починати писати керуючу програму. Схему і друковану плату теж не можна починати робити, бо незрозуміло навіть, яке живлення потрібно і який розмір контролера, і скільки у нього ніжок для підключення.

Arduino (Ардуіно) – апаратна обчислювальна платформа для аматорського конструювання, основними компонентами якої є плата мікроконтролера з елементами вводу/виводу та середовище розробки Processing/Wiring на мові програмування, що є спрощеною підмножиною C/C++. Arduino може використовуватися як для створення автономних інтерактивних об'єктів, так і підключатися до програмного забезпечення, яке виконується на комп'ютері. Інформація про плату знаходяться у відкритому доступі і можуть бути використані тими, хто воліє створювати плати власноруч.

Одна з основних характеристик – це розрядність мікроконтролера. Як і в більшості сучасних обчислювальних систем пам'ять, з якою працює мікроконтролер, логічно і схематично відділена від нього (хоч і може перебувати з ним на одному чіпі), і тому мікроконтролер не може безпосередньо

виконувати операції з даними, які в ній записані (наприклад, складати). Для цього йому потрібно скопіювати ці дані в спеціальні осередки пам'яті, які як раз є безпосередньою частиною мікросхеми і називаються регістрами. Після того, як операція буде виконана, дані з регістрів, як правило, копіюються назад в пам'ять. Так зроблено тому, що реєстрова пам'ять, яка працює на тих самих швидкостях, що і мікроконтролер, є дуже дорогою, тому її не може бути багато. З іншого боку, основна пам'ять, винесена в свою власну мікросхему, робиться за дещо іншою технологією, що дозволяє її істотно здешевити і, відповідно, зробити її більше, правда за все треба платити, тому подібна пам'ять, хоч її і багато, працює значно повільніше, ніж реєстрова – в десятки, а іноді і в сотні разів! Тому основний цикл роботи сучасного мікропроцесора виглядає так: скопіювати потрібні дані з основної пам'яті в регістри, виконати операцію над регістрами, скопіювати результат назад у зовнішню пам'ять або продовжити обчислення над значеннями.

Мікроконтролери для Arduino відрізняються наявністю заздалегідь прошитого в них завантажувача (англ. bootloader). За допомогою цього завантажувача користувач завантажує свою програму до мікроконтролеру без використання традиційних окремих апаратних програматорів. Завантажувач з'єднується з комп'ютером через інтерфейс USB (якщо він є на платі) або окремим перехідником UART-USB. Підтримка завантажувача вбудована Arduino IDE і виконується в один натиск миші.

Плата Arduino в основному складається з мікроконтролерів сімейства Atmel AVR: ATmega328, ATmega168, ATmega2560, ATmega32U4, ATTiny85, а також елементів для програмування та інтеграції з іншими пристроями.

Нижче прикладена таблиця 1, які мікроконтролери можна зустріти на вказаних розповсюджених платах. Поступово в лінійці плат з'явилися процесори ARM. Спочатку це був AT91SAM3X8E на платі класичного конструктиву (Due). Пізніше з'явилася лінійка плат Arduino MKR в конструктиві DIP, оснащена контролером SAMD21.

Таблиця 1 – Мікроконтролери

Назва мікроконтролеру	Тактова частота	Об'єм флеш-пам'яті	Об'єм оперативної пам'яті	Загальна кількість портів	Кількість ШИМ портів	Кількість АЦП портів	Перелік плат
ATmega 2560	16 МГц	256 КБ	8 КБ	54	15	16	Mega
ATmega 32U4	16 МГц	32 КБ	2,5 КБ	20	7	12	Leonardo, Micro, Yun
ATmega 328	16 МГц	32 КБ	2 КБ	14	6	8	Uno R3, Mini, Nano R2, Pro, Pro mini, Uno та Nano
ATtiny85	20 МГц	8 КБ	512 Б	6	4	4	Digispark
ATmega 168	16 МГц	16 КБ	1 КБ	14	6	8	Uno R1, Uno R2, Pro mini, Nano

На випадок затирання завантажувача або придбання мікроконтролера без завантажувача, розробники надають можливість прошити завантажувач у мікроконтролер самостійно. Для цього в Arduino IDE вбудована підтримка кількох популярних дешевих програматорів, а більшість плат Arduino має штирьовий роз'єм для внутрішньосхемного програмування (ICSP для AVR, JTAG або SWD для ARM). На багатьох платах наявний лінійний стабілізатор напруги +5В або +3,3В. Тактування здійснюється на частоті 16 або 8 МГц кварцовим резонатором. У мікроконтролер записаний завантажувач (bootloader), тому зовнішній програматор не потрібен. В рамках співпраці зі сторонніми виробниками в Arduino IDE була включена підтримка деяких апаратних засобів Intel x86. Intel Galileo та Intel Edison – Arduino-сумісні плати на Intel x86 архітектурі. Плати механічно і електрично сумісні з периферійними

платами Arduino. Плати функціонують під власною ОС Linux, поверх якої працює додаток, що дозволяє завантажувати і виконувати скетчі Arduino. Плати програмуються через USB, що можливо завдяки мікросхемі конвертера USB-to-Serial FT232R. [1]

Ардуіно та Ардуіно-сумісні плати спроектовані таким чином, щоб їх можна було за необхідності розширювати, додаючи до пристрою нові компоненти («shields»). Ці плати розширень підключаються до Ардуіно за допомогою встановлених на них штирових роз'ємів.

Також сторонніми виробниками випускаються набори електромеханічних елементів, орієнтованих на роботу спільно з платами Ардуіно (як правило, через спеціальні плати-«драйвери») – двигуни, електромагніти тощо.

У концепцію Ардуіно не входять корпусні чи монтажні деталі (окрім попередньо підібраних «комплектів» для розробки одного чи кількох пристроїв). Розробник обирає метод установки й механічного захисту процесорних плат та компонентів розширення самостійно.

Плати Arduino дозволяють використовувати значну кількість виводів мікроконтролера як вхідні/вихідні контакти у зовнішніх схемах. Наприклад, на платі UNO R3 CH340G розміщено 14 цифрових входів/виходів (позначені арабськими цифрами від 0 до 13) і 6 аналогових входів (позначені від А0 до А5). Ці сигнали доступні на платі через контактні майданчики або штирові роз'єми. Також існує багато різних зовнішніх плат розширення, які називаються «shields» («щити»), які приєднуються до плати Arduino через штирові роз'єми.

Також існує ряд уніфікованих плат, що допускає конструктивно жорстке з'єднання процесорної плати та плат розширення в стопку через штирові лінійки. Крім того, випускаються плати зі зменшеним форм-фактором, такі як Nano та Lilypad.

Arduino Nano – це повнофункціональний мініатюрний пристрій на базі мікроконтролера ATmega328 (Arduino Nano 3.0) або ATmega168 (Arduino Nano 2.x), адаптований для використання з макетною платою. За функціональністю пристрій схожий на Arduino Duemilanove, і відрізняється від нього розмірами, відсутністю роз'єму живлення, а також іншим типом (Mini-B) USB-кабелю (рис.1.1). Arduino Nano розроблено і випускається фірмою Gravitech

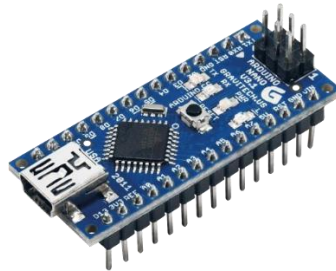


Рисунок 1.1 – Arduino nano 3.0

Arduino Uno – це пристрій на основі мікроконтролера ATmega328. До його складу входить все необхідне для зручної роботи з мікроконтролером: 14 цифрових входів/виходів (з них 6 можуть використовуватися в якості ШІМ-виходів), 6 аналогових входів, кварцовий резонатор на 16 МГц, роз'єм USB, роз'єм живлення, роз'єм для внутрисхемного програмування (ICSP) і кнопка скидання (рис. 1.2).



Рисунок 1.2 – Arduino UNO R3

Arduino Leonardo – це пристрій на базі мікроконтролера ATmega32U4 (рис.1.3). До його складу входить все необхідне для роботи з даним мікроконтролером: 20 цифрових входів / виходів (7 з яких можуть працювати в якості ШІМ-виходів, 12 – в якості аналогових входів), кварцовий резонатор на 16 МГц, роз’єм мікро-USB, роз’єм живлення, роз’єм для внутрисхемного програмування ICSP і кнопка скидання.

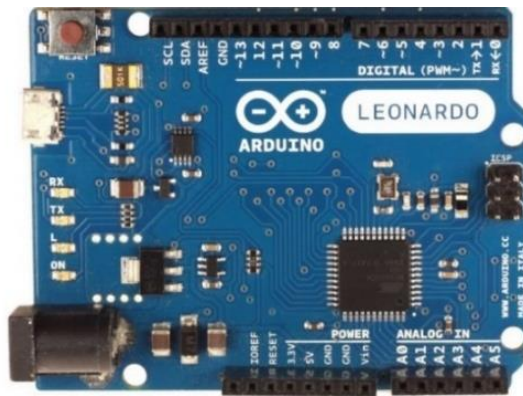


Рисунок 1.3 – Arduino Leonardo

Arduino Mega 2560 – це пристрій на основі мікроконтролера ATmega2560. До його складу входить все необхідне для зручної роботи з мікроконтролером: 54 цифрових входів/виходів (з яких 15 можуть використовуватися в якості ШІМ-виходів), 16 аналогових входів, 4 UART (апаратних прийомопередавача для реалізації послідовних інтерфейсів), кварцовий резонатор на 16 МГц, роз’єм USB, роз’єм живлення, роз’єм ICSP для внутрисхемного програмування і кнопка скидання. Arduino Mega сумісний з більшістю плат розширення, розроблених для Arduino Duemilanove і Diecimila.



Рисунок 1.4 – Arduino Mega 2560

Arduino Due – це пристрій на основі мікропроцесора Atmel SAM3X8E ARM Cortex-M3. Це перша плата Arduino на базі 32-розрядного мікроконтролера ARM. До її складу входять 54 цифрових виводів (з яких 12 можуть працювати в якості ШІМ-виходів), 12 аналогових входів, 4 UART (апаратних прийомопередавача, що здійснюють послідовну передачу даних), генератор тактової частоти на 84 МГц, USB з підтримкою технології OTG, 2 ЦАП (цифро-аналогових перетворювача), 2 TWI, роз'єм живлення, роз'єм SPI, роз'єм JTAG, кнопка скидання і кнопка очищення пам'яті. Розглянуті різновиди платформи Arduino в різній мірі можуть бути використані в якості апаратної основи при розробці різних проектів, використовуючи датчики та комплектуючі.



Рисунок 1.5 – Arduino Due

Велика кількість плат на базі платформи Arduino, що означає великий вибір серед різних рішень різних цінових категорій, зручна середа для розробки програмного забезпечення та можливість зручної роботи з різними зовнішніми виконавчими пристроями і є, на наш погляд, найбільшою перевагою цієї платформи. Виходячи з цих факторів, нами була обрана саме плата на платформі Arduino. [2]

1.2 Програмування для платформи Arduino

Як вже було сказано, основною мовою програмування для створення програмного коду проектів на базі платформи Arduino є C++.

C++ – компілюєма, статично типізована мова програмування загального призначення. Вона підтримує такі парадигми програмування, як процедурне, об'єктно-орієнтоване, узагальнене програмування.

Мова має багату стандартну бібліотеку, яка включає поширені контейнери і алгоритми, введення-виведення, регулярні висловлювання, підтримку багатопоточності та інші можливості. C++ поєднує властивості як високорівневих, і низькорівневих мов.

У порівнянні з його попередником – мовою C – найбільшу увагу приділено підтримці об'єктно-орієнтованого та узагальненого програмування. C++ широко використовується для розробки програмного забезпечення, будучи однією з найпопулярніших мов програмування.

Область її застосування включає створення операційних систем, різноманітних прикладних програм, драйверів пристроїв, додатків для систем, високопродуктивних серверів, а також ігор. Існує безліч реалізацій мови C++ як безкоштовних, так і комерційних і для різних платформ. Наприклад, на платформі x86 це GCC, Visual C++, Intel C++ Compiler, Embarcadero (Borland) C++ Builder та інші. C++ вплинув інші мови програмування, насамперед Java і C#.

Синтаксис C++ успадкований від мови C. Спочатку одним із принципів розробки було збереження сумісності з C. Проте C++ не є в строгому сенсі надмножиною C; безліч програм, які можуть однаково успішно транслюватися як компіляторами C, і компіляторами C++, досить великий, але не включає всі можливі програми на C. Мова виникла на початку 1980-х років, коли співробітник фірми Bell Labs Бйорн Страуструп придумав ряд удосконалень для мови C під власні потреби. Коли наприкінці 1970-х років Страуструп почав працювати в Bell Labs над завданнями теорії черг, він виявив, що спроби застосування існуючих на той час мов моделювання виявляються неефективними, а застосування високоефективних машинних мов занадто складно через їхню обмежену виразність. Так, мова Симула має такі можливості, які були б дуже корисні для розробки великого програмного забезпечення, але працює занадто повільно, а мова BCPL (її наступником став C) досить швидка, але дуже близька до мов низького рівня і не підходить для розробки великого програмного забезпечення.

Мова C, будучи базовою мовою системи UNIX, на якій працювали комп'ютери Bell, є швидкою, багатофункціональною та переносимою. Страуструп додав до неї можливість роботи з класами та об'єктами. Ранні версії мови, що спочатку називалися C with classes (Cі з класами), стали доступні з 1980 року.

Мова програмування Arduino називається Arduino C і є мовою C++ з фреймворком Wiring, вона має деякі відмінності в частині написання коду, з особливостями, що полегшують написання працюючої програми — є набір бібліотек, що включає функції та об'єкти. При компіляції IDE створює тимчасовий файл з розширенням *.cpp.

Програми, написані програмістом Arduino, називаються начерками або скетчами і зберігаються у файлах з розширенням *.ino. Ці файли перед компіляцією обробляються препроцесором Ардуіно. Також існує можливість створювати та підключати до проекту стандартні файли C++. Програміст повинен написати дві обов'язкові для Arduino функції `setup()` та `loop()`. Перша

викликається одноразово при старті, друга виконується у нескінченному циклі. У текст своєї програми (скетчу) програміст не повинен вставляти заголовні файли використовуваних стандартних бібліотек. Ці заголовні файли надасть препроцесор Arduino відповідно до конфігурації проекту. Однак користувачські бібліотеки потрібно вказувати. Так виглядає повний текст найпростішої скетчу миготіння світлодіодом, підключеного до 13-го піну Arduino, з періодом 2 секунди (півперіоду, тобто 1 секунду світлодіод горить, півперіоду не горить):

```
void setup ()
{
  pinMode (13, OUTPUT); // Призначення порту 13 як вихідного
}
void loop ()
{
  digitalWrite (13, HIGH); // Встановлення порта 13 в стан "1", світлодіод горить
  delay (1000); // Затримка виконання на 1000 мілісекунд
  digitalWrite (13, LOW); // Встановлення порта 13 в стан "0", світлодіод не горить
  delay (1000); // Затримка виконання на 1000 мілісекунд
}
```

Основною та найбільш популярною середою програмування для плат на платформі Arduino є фірмове програмне забезпечення Arduino IDE.

Популярність, відкритість та простота платформи Arduino викликала великий вал сторонніх програмних рішень. В основному це рішення навколо інтеграції компілятора та бутлоадера (завантажувача) Arduino в оболонки для програмістів (IDE). Великий список цих інструментів є тут. Серед них можна виділити як професійні інструменти, такі як Processing, Eclipse, Microsoft Visual Studio, Atmel Studio, так і інструменти для дітей, наприклад, Scratch for Arduino. Але так як Arduino IDE являється найбільш поширеною серед розробників цієї платформи, ми детально розглянемо її. На наш суб'єктивний

погляд, вона є найбільш зручною та інтуїтивно зрозумілою з точки зору організації.

Інтегроване середовище розробки Arduino IDE (рис. 1.6) — це багатоплатформовий додаток на Java, що включає в себе редактор коду, компілятор і модуль передачі прошивки в плату. Середовище розробки засноване на мові програмування Processing та спроектоване для програмування новачками, незнайомими близько з розробкою програмного забезпечення. Мова програмування аналогічна мові Wiring. Загалом, це C++, доповнений деякими бібліотеками. Програми обробляються за допомогою препроцесора, а потім компілюються за допомогою AVR-GCC.

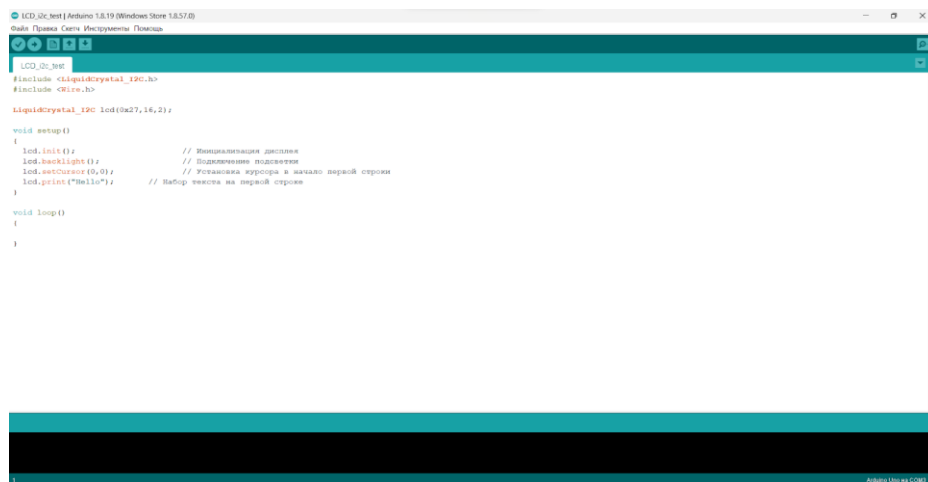


Рисунок 1.6 – Головне вікно програми Arduino IDE

Arduino IDE має можливість створення своїх програмно-апаратних платформ. Цією можливістю користуються сторонні компанії, які додають Arduino IDE свої набори плат і компіляторів-завантажувачів до них.

Окрім того, що для написання коду для робочої програми в Arduino IDE розробнику треба написати здебільшого тільки дві функції – setup() та loop() – Arduino IDE має ще багато зручностей для розробки. Наприклад це зручний менеджер бібліотек, що дає змогу швидко знайти та встановити потрібну для

роботи з визначеним обладнанням бібліотеку. Для цього в основному окні IDE потрібно лише натиснути на вкладку “Інструменти”, що знаходиться над полем для редагування коду, і в випаovшому списку вибрати “Менеджер бібліотек”. Після чого відкривається ще одна вікно, де буде показаний список доступних бібліотек (рис. 1.7)

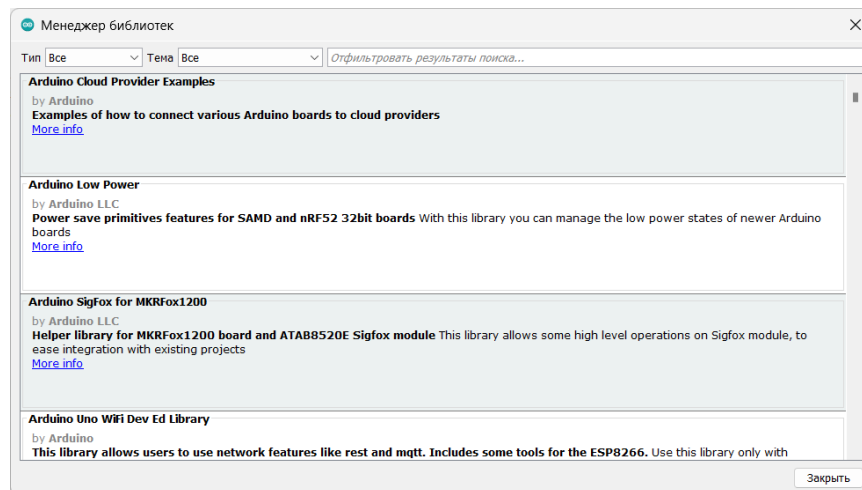


Рисунок 1.7 – Вікно менеджера бібліотек Arduino IDE

У випадковому списку поряд з написом “Тип” можна відфільтрувати лише потрібні бібліотеки, наприклад ті, які вже встановлені, ті, які можна оновити, бібліотеки розроблені розробниками платформи Arduino тощо.

Поряд з написом “Тема” можна побачити ще один випадковий список з фільтрами. У ньому доступні фільтри зв’язані з призначенням шуканої бібліотеки, а саме “Зв’язок”, “Обробка даних”, “Зберігання даних”, “Пристрої управління”, “Екран”, “Сенсори” тощо.

Також присутня строка пошуку, в яку можна ввести назву або частину опису конкретної бібліотеки. Нижче можна побачити сам список бібліотек, де написана назва бібліотеки, її автор та частина опису бібліотеки англійською мовою. Якщо натиснути на посилання з текстом “More info”, відкривається браузер з новою вкладкою, де буде відкритий відповідна сторінка з

інформацією. Частіш за все вона веде до відповідного розділу на GitHub або на відповідну сторінку веб-сайту Arduino. На рис 1.8 - рис 1.9 можна побачити приклад опису бібліотеки, різні попередження та додатки, а також список функцій бібліотеки.

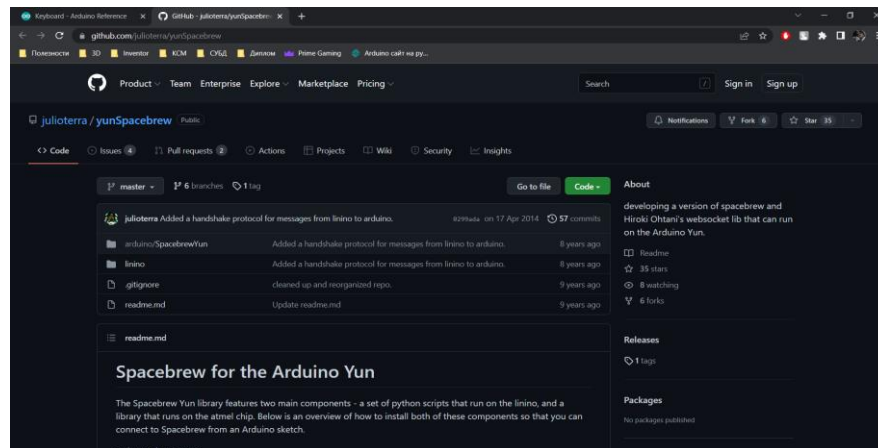


Рисунок 1.8 – Вигляд інформаційних сторінок GitHub

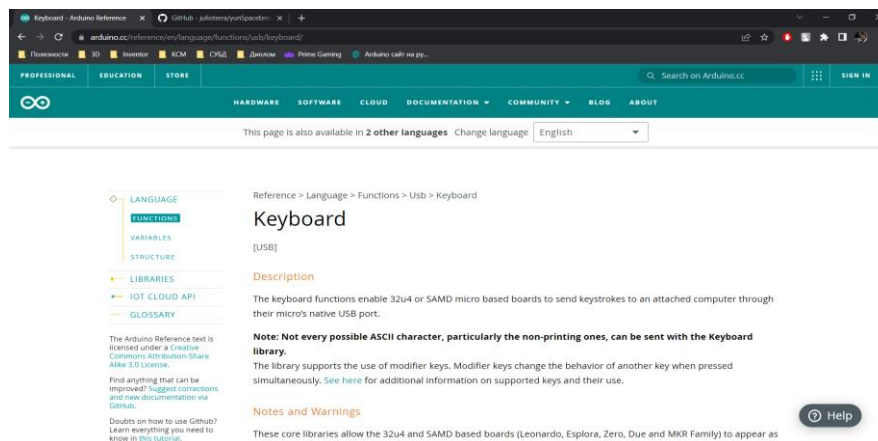


Рисунок 1.9 – Вигляд інформаційних сторінок з описом бібліотек

Після ознайомлення зі змістом бібліотеки потрібно лише навести курсор на потрібну бібліотеку в менеджері бібліотек, вибрати потрібну версію та натиснути кнопку “Встановити” (Рис. 1.10)

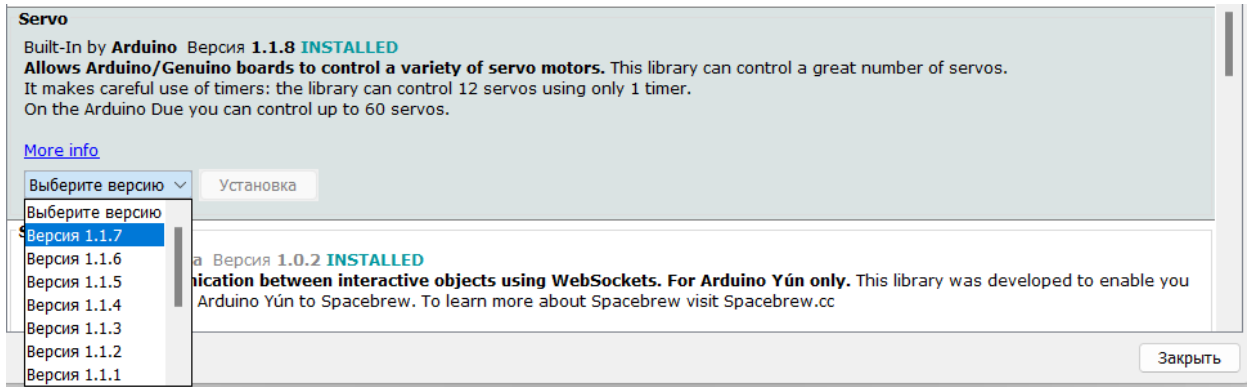


Рисунок 1.10 – Встановлення бібліотеки на прикладі “Servo”

Після цього у самому скетчі достатньо визвати цю бібліотеку за допомогою директиви “#include”. Наприклад, щоб визвати бібліотеку “Servo” після її встановлення достатньо додати рядок “#include <Servo.h>”.

1.3 Аргументи на користь роботи з платформою Arduino

Головною перевагою роботи Arduino з зовнішніми пристроями з точки зору розробки програмного забезпечення є наявність великої кількості готових бібліотек.

Бібліотека (від англ. library) – збірка об'єктів чи підпрограм для вирішення близьких за тематикою задач. У залежності від мови програмування бібліотеки містять об'єктні модулі чи сирцевий код та дані, допоміжні для задіяння та інтеграції нових можливостей в програмні рішення. Бібліотека може означати те саме, що модуль, або декілька модулів. З точки зору комп'ютерних наук бібліотеки діляться на статичні та динамічні.

Статичні бібліотеки можуть існувати у вигляді початкового тексту, що підключається програмістом до своєї програми на етапі написання (наприклад, для мови Fortran існує величезна кількість бібліотек для вирішення різних завдань саме в початкових текстах). Також статичні бібліотеки можуть існувати

у вигляді об'єктних файлів, що приєднуються (лінкуються) до виконуваної програми на етапі компіляції (у Microsoft Windows такі файли мають розширення “.lib”, у UNIX-подібних ОС — зазвичай “.a”). В результаті програма включає всі необхідні функції, що робить її автономною, але збільшує розмір.

Динамічні бібліотеки також називаються розподілюваними бібліотеками. Це окремі файли, що надають програмі набір використовуваних функцій для завантаження на етапі виконання при зверненні програми до ОС із заявкою на виконання функції з бібліотеки. Якщо необхідна бібліотека вже завантажена в оперативну пам'ять, програма використовуватиме завантажену копію бібліотеки. Такий підхід дозволяє зекономити час і пам'ять, оскільки декілька програм використовують одну копію бібліотеки, вже завантажену в пам'ять. Динамічні бібліотеки зберігаються зазвичай у визначеному місці й мають стандартне розширення. При написанні програми програмістові досить вказати транслятору мови програмування (компілятору або інтерпретатору), яку бібліотеку слід підключити і яку функцію використовувати зі вказаної бібліотеки. Ні початковий текст, ні виконуваний код функції не входить до складу програми.

Підводячи підсумок з вказаної інформації, основними перевагами перевагами роботи з платформою Arduino є:

- велика кількість плат на базі платформи Arduino;
- фінансова доступність компонентів;
- зручна середа для розробки програмного забезпечення;
- підтримка програмування мовою C++ з фреймворком Wiring;
- можливість зручної роботи з зовнішніми пристроями.

Виходячи з цих факторів, була обрана саме плата на платформі Arduino та в якості середи розробки була обрана Arduino IDE.

1.4 Постановка завдання.

Як вже було описано у попередніх підрозділах, платформа Arduino має великий потенціал з погляду створення різноманітних технічних проектів, завдяки своєму простому та інтуїтивно зрозумілому середовищу розробки. Через те, що розробляємий пристрій, у своїй суті являється GPS-трекером, було вирішено розробити перелік вимог до апаратної та програмної частини. Цей перелік надасть змогу ретельно продумати загальний алгоритм роботи пристрою та підібрати компоненти апаратної частини так, щоб з ними було зручніше працювати у процесі збірки та тестів. Алгоритм буде побудовано методом частинних цілей. Цей метод полягає у тому, що глобальна велика задача ділиться на окремі задачі. Якщо велику задачу, можливо, і не можна досягнути, зрозуміти, як її розв'язувати, то для кожної з поділених задач може існувати давно відомий алгоритм розв'язку, або пошук такого алгоритму є значно легшою задачею.

Перш за все, треба зрозуміти що таке GPS-трекер, та як взагалі він працює. GPS трекер – це пристрій, який використовує глобальну систему позионування (Global Positioning System) для визначення точного місцезнаходження об'єкта в будь-який момент часу. Зазвичай, пристрій складається з декількох ключових компонентів:

1. GPS модуль: Основний компонент, який отримує сигнали від супутників GPS і обчислює географічні координати (широту, довготу) і, у деяких випадках, висоту.
2. Мікроконтролер: Використовується для обробки даних, отриманих від GPS модуля, і для управління іншими компонентами трекера.
3. Комунікаційний модуль: Відповідає за передачу даних на сервер. Це може бути GSM/GPRS модуль, Wi-Fi модуль, або модуль LoRa, залежно від способу передачі даних, який ви виберете.
4. Антена: Забезпечує прийом сигналів від супутників GPS.

5. Живлення: Зазвичай складається з акумулятора або батарейки, яка забезпечує роботу всіх компонентів.

Трекер працює за простим алгоритмом. Спочатку, GPS модуль трекера приймає сигнали від принаймні чотирьох GPS супутників. Супутники передають дані про своє місцезнаходження і точний час. Модуль використовує отримані сигнали для обчислення часу, який знадобився сигналам для досягнення трекера. Знаючи точний час і швидкість сигналу, GPS модуль обчислює відстані до супутників і, використовуючи метод триангуляції, визначає координати трекера. Сутність методу триангуляції полягає в побудові на місцевості систем трикутників, у яких вимірюються всі кути та довжини деяких базисних сторін. Довжини інших сторін трикутників розраховуються за відомими формулами тригонометрії. Після перетворення модулем супутникових даних у координати, він передає їх мікроконтролеру. Отримавши їх, контролер може обробляти їх за потребою. Наприклад, формувати пакет даних для подальшої передачі, через комунікаційний модуль. Цей модуль передає координати на віддалений сервер. Це може бути здійснено через стільникову мережу (GSM/GPRS), Wi-Fi, LoRa, або інші засоби зв'язку. Сервер приймає дані і зберігає їх у базі даних. Веб-додаток або мобільний додаток може отримувати ці дані з сервера і відображати їх на мапі. Ось приклад простого коду, для зчитування даних з GPS модуля на Arduino. Цей код зчитує дані з GPS модуля і виводить їх у серійний монітор:

```
#include <TinyGPS++.h>
#include <SoftwareSerial.h>

// Встановлення GPS модуля
TinyGPSPlus gps;
SoftwareSerial ss(4, 3); // RX, TX
void setup() {
  Serial.begin(9600);
  ss.begin(9600);
}
```

```
void Loop() {  
  while (ss.available() > 0) {  
    gps.encode(ss.read());  
    if (gps.Location.isUpdated()) {  
      Serial.print("Latitude: ");  
      Serial.println(gps.Location.Lat(), 6);  
      Serial.print("Longitude: ");  
      Serial.println(gps.Location.Lng(), 6);  
      Serial.print("Altitude: ");  
      Serial.println(gps.altitude.meters());  
      Serial.print("Satellites: ");  
      Serial.println(gps.satellites.value());  
    }  
  }  
}
```

2. ПОШУК СПОСОБІВ РЕАЛІЗАЦІЇ ПРОЕКТУ

2.1 Вибір апаратного забезпечення

У попередньому підрозділі, було обговорено сам концепт пристрою, а також поверхнево сказано про основні деталі та компоненти конструкції. У цьому підрозділі, буде конкретно обумовлено про кожен елемент конструкції, а також його роль у апаратній реалізації проекту. При виборі компонентів для апаратної частини, ми звертали особу увагу на наступні критерії компонентів:

- дешевизна компонента;
- компактність компонента;
- необхідна напруга для коректної роботи;
- простота підключення та використання.

Для початку, розберемося із мікроконтролером проекту. В цілому, завдання мікроконтролера зводиться до того, щоб контролювати порти вводу-виводу, які розробники представили для зручності у вигляді пінів. До цих пінів зручно підключати зовнішні пристрої, що робить процедуру підключення Arduino в схему дуже простою справою.

Дані піни мають дві дуже важливі особливості, вони можуть як подавати напругу так і перевіряти наявність напруги, що дозволяє як подати необхідну напругу на пін так і перевірити наявність напруги на потрібних пінах, для цього при програмуванні Arduino необхідно вказати в якому режимі буде працювати той чи інший пін на вхід, тобто “INPUT”, перевіряючи наявність напруги або на вихід, тобто “OUTPUT”, подаючи напругу на пін. Як правило визначення режиму роботи пінів здійснюється у функції “setup()”, яка, наприклад, може виглядати наступним чином:

```
void setup()
{
    pinMode(2, OUTPUT);
    pinMode(3, INPUT);}

```

На цьому прикладі функції “`setup()`” наглядно показано, що пін під номером 2 працює у режиму виходу, а пін під номером 3 працює у режимі входу.

Також Arduino може надсилати текстові повідомлення на персональний комп'ютер. Здебільшого це зручно для тестування роботи проекту. Для того щоб скористатися виводом інформації на комп'ютер, необхідно у тілі функції “`setup()`” форматувати послідовний порт пристрою за допомогою функції “`.begin()`”.

Безпосередньо для виведення короткого повідомлення використовується функція “`print()`”. Обидві можна побачити на простому прикладі:

```
void setup()
{
    Serial.begin(9600);
    Serial.print("Hello!");
}
```

Arduino – це відкрита платформа. По суті, будь-який бажає може скачати схему з офіційного сайту або одного з популярних форумів, а потім зібрати плату на основі контролера Atmega. Необхідні електронні компоненти можна вельми не дорого купити в безлічі інтернетів-магазинів.

Таблиця 2 – Характеристика Arduino Uno

Мікроконтроллер	ATmega328
Робоча напруга	5В
Напруга живлення (рекомендоване)	7-12В
Напруга живлення (максимальне)	6-20В
Цифрові входи/виходи	14 (6 из которых ШІМ)
Аналогові входи	8
Максимальний струм одного виводу	40 мА
Flash-пам'ять	32 КБ (ATmega328)
SRAM	2 КБ (ATmega328)
EEPROM	1 КБ (ATmega328)
Тактова частота	16 МГц

Контроллер Nano є самим підходящим варіантом для початку роботи над проектом: вона має невеликий розмір 1.85 см x 4.3 см, досить доступна із-за масового випуску всіляких клонів, під неї написана величезна кількість безкоштовних уроків і скетчів.

Піни Ардуїно (рисунок 2.1) використовуються для підключення зовнішніх пристроїв і можуть працювати як в режимі входу (INPUT), так і в режимі виходу (OUTPUT). До кожного входу може бути підключений вбудований резистор 20-50 кОм за допомогою виконання команди `pinmode ()` у режимі `Input_pullup`. Допустимий струм на кожному з виходів – 20 мА, не більше 40 мА у піку.

Для зручності роботи деякі піни поєднують в собі декілька функцій:

- піни 0 і 1 – контакти UART (RX і TX відповідно);
- піни з 10 по 13 – контакти SPI (SS, MOSI, MISO і SCK відповідно);
- піни A4 і A5 – контакти I2c (SDA і SCL відповідно).

Таблиця 3 – Аналогові піни Arduino Uno

Пін	Адресація в скетчі
Аналоговий пін A0	A0
Аналоговий пін A1	A1
Аналоговий пін A2	A2
Аналоговий пін A3	A3
Аналоговий пін A4	A4
Аналоговий пін A5	A5
Аналоговий пін A6	A6
Аналоговий пін A7	A7

ARDUINO NANO PINOUT

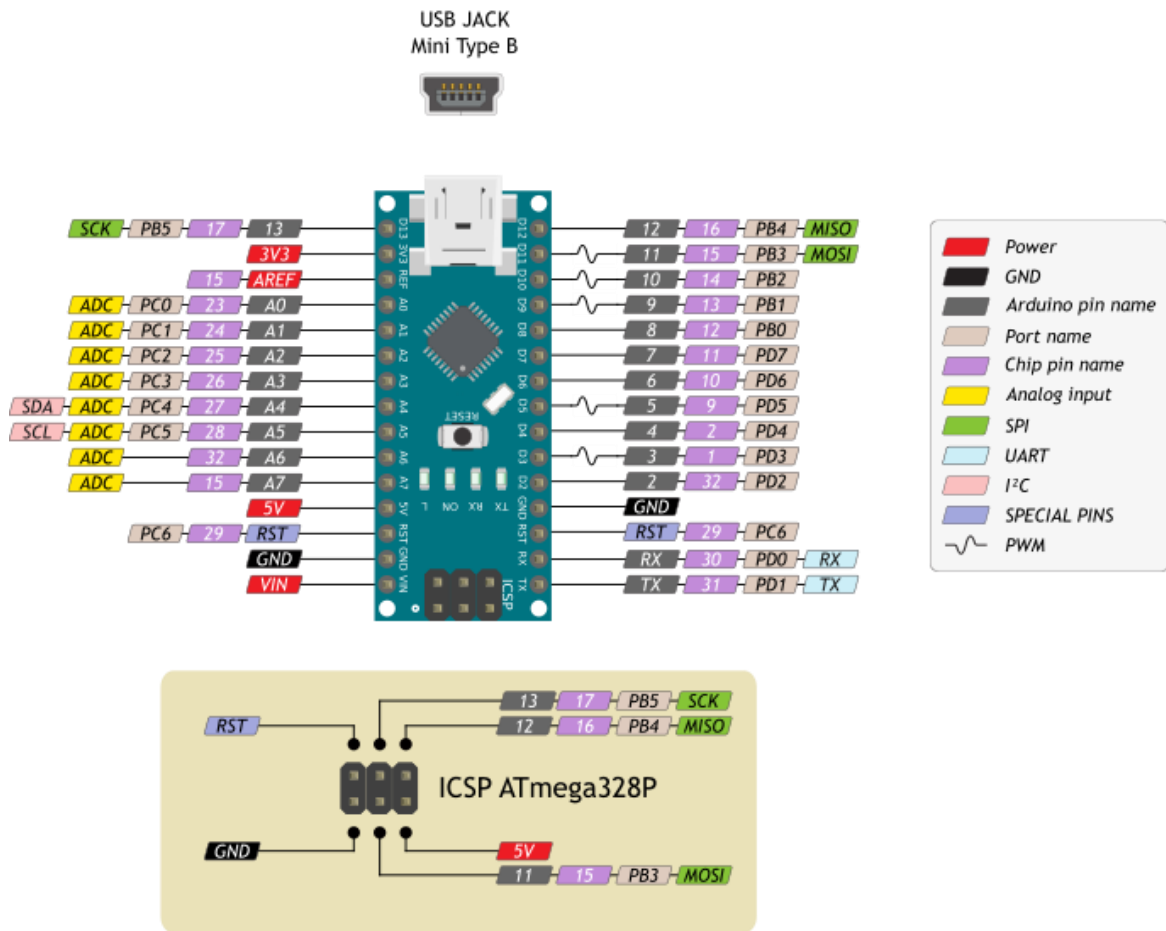


Рисунок 2.1 – Розпинування плати на базі ATMEGA 328

Піни з номерами від 0 до 13 є цифровими. Це означає, що ви можете прочитувати і подавати на них лише два види сигналів: HIGH і LOW. За допомогою ШІМ також можна використовувати цифрові порти для управління потужністю підключених пристроїв. Аналогові піни Arduino Nano призначені для підключення аналогових пристроїв і є входами для вбудованого АЦП, який в Arduino десятирозрядний.

Підключення будь-яких пристроїв до плати здійснюється шляхом приєднання до контактів, розташованих на платі контролера: одному з цифрових або аналогових пінов або пінам живлення. Простий світлодіод можна приєднати, використовуючи два контакти: землю (GND) і сигнальний (або

контакт живлення). Найпростіший датчик вимагатиме задіювати мінімум три контакти: два для живлення, один для сигналу.

Робоча напруга плати Arduino Uno – 5V. На платі встановлений стабілізатор напруги, тому на вхід можна подавати живлення з різних джерел. Окрім цього, плату можна живити з USB-пристроїв. Джерело живлення вибирається автоматично. Користувач має у своєму розпорядженні наступні варіанти живлення:

1. Живлення від зовнішнього адаптера, напруга, що рекомендується, від 7 до 12V. Максимальне напруга 20V, але значення вище 12V з високою ймовірністю швидко виведе плату з ладу. Напруга менше 7V може привести до нестабільної роботи, оскільки на вхідному каскаді може запросто втрачатися 1-2V. Для підключення живлення може використовуватися вбудований роз'єм DC 2.1 мм або безпосередньо вхід VIN для підключення джерела за допомогою дротів;
2. Живлення від USB-порту комп'ютера;
3. Подача 5 В безпосередньо на пін 5V. В цьому випадку обходиться стороною вхідний стабілізатор і навіть щонайменше перевищення напруги може привести до поломки пристрою.

Деякі модифікації стандартної плати Uno можуть підтримувати пам'ять з великими значеннями, чим в стандартному варіанті. Але слід розуміти, що для роботи з ними буде потрібно і додаткові бібліотеки. Плата Uno за умовчанням підтримує три типи пам'яті:

1. Flash – пам'ять об'ємом 32кБ. Це основне сховище для команд. Коли ви прошиваєте контроллер своїм скетчем, він записується саме сюди. 2кБ з даного пулу пам'яті відводиться на bootloader – програма, яка займається ініціалізацією системи, завантаження через USB і запуск скетчу;
2. оперативна пам'ять SRAM об'ємом 2 кБ. Тут по-умовчання зберігаються змінні і об'єкти, що створюються в ході роботи програми. Пам'ять

ця енерго-залежна, при виключенні живлення всі дані, зрозуміло, зітру-
ться;

3. незалежна пам'ять (EEPROM) об'ємом 1кБ. Тут можна зберігати дані, які не зітруться при виключенні контролера. Але процедура запису і про-
читування EEPROM вимагає використання додаткової бібліотеки, яка
доступна в Arduino IDE по-умовчання. Також необхідно пам'ятати про
обмеження циклів перезапису, властивих технології EEPROM

Сьогодні на ринку можна зустріти безліч варіантів плат Arduino. Найпо-
пулярнішими конкурентами Nano є плати Uno і Mega. Перша плата є самим
підходящим варіантом для початку роботи з платформою. Друга – для про-
єктів, де в схема досить складна і потрібна безліч виходів.

Сучасні плати Arduino Uno і Arduino Nano версії R3 мають, як правило,
у своєму розпорядженні загальний мікроконтролер: Atmega328. Ключовою
відмінністю є розмір плати і тип контактних майданчиків.

Габарити Arduino Uno: 6,8 см x 5,3 см.

Габарити Arduino Nano: 4,2 см x 1,85 см.

У Arduino Uno використовуються коннектори типа «мама», в Nano –
«гребінь» з ніжок, причому в деяких моделях контактні майданчики взагалі не
припаяні. Звичайно, більший розмір Uno в порівнянні з Nano в деяких випад-
ках є перевагою, а в деяких – недоліком. З платою великого розміру набагато
зручніше виробляти монтаж, але вона незручна в реальних проектах, оскільки
сильно збільшує габарити кінцевого пристрою.

На платах Arduino Uno традиційно використовується роз'єм TYPE-B
(широко застосовується також для підключення принтерів і багатофункціо-
нальних пристроїв). В деяких випадках можна зустріти варіант з роз'ємом
Micro USB.

У платах Arduino Nano стандартом є Mini або Micro USB. Також, відмін-
ності є і в роз'ємі живлення. У платі Uno є вбудований роз'єм DC, в Nano йому
просто не знайшлося місця. Окрім апаратних, існують ще невеликі відмінності

в процесі завантаження скетчу в плату. Перед завантаженням слід перекона-тися, що ви вибрали вірну плату в меню «Інструменти-плата».

Плата Mega в повній відповідності зі своєю назвою є на сьогоднішній день найбільшою за розміром і кількістю пінів контролерів Arduino. В порівнянні з нею в Uno значно менше пінів і пам'яті. Ось список основних відмінностей:

1. плата Mega використовує інший мікроконтроллер: Atmega 2560. Але тактова частота його рівна 16МГц, так само як і в Uno;
2. у платі Mega більша кількість цифрових пінів – 54 замість 14 в платі Uno. І аналогових – 16 / 6;
3. в платі Mega більше контактів, що підтримують апаратні переривання: 6 проти 2. Більше Serial портів – 4 проти 1;
4. за об'ємом пам'яті Uno теж істотно поступається Mega. Flash -пам'ять 32/256, SRAM – 2/8, EEPROM – 4/1.

Виходячи зі всього сказаного можна зробити вивід, що для великих складних проектів з програмами великого розміру і активним використанням різних комунікаційних портів краще вибирати Mega. Але ці плати дорожче Uno і займають більше місця, тому для невеликих проектів, що не використовують всі додаткові можливості Mega, сповна зійде Uno. Але для проектів, де важливим є розмір, та компактність, необхідно вибирати Nano [3].

Основним компонентом проекту є плата Arduino Nano. Тепер поговоримо про додаткові елементи конструкції. У попередньому підрозділі був опис їх функціоналу в апаратній реалізації. У цьому ж підрозділі буде більш конкретний опис кожного елементу пристрою. Почнемо з більшого до меншого.

GPS модуль є ключовим компонентом трекера, який забезпечує отримання точних координат для моніторингу місцезнаходження домашнього улюбленця. Його просте підключення та використання з Arduino Nano дозволяє швидко інтегрувати його у вашу систему для передачі даних через GSM модуль на віддалений сервер. NEO-6M – це популярний GPS модуль від

компанії u-blox, який використовується для отримання географічних координат на основі сигналів від супутників GPS (рис. 2.2).

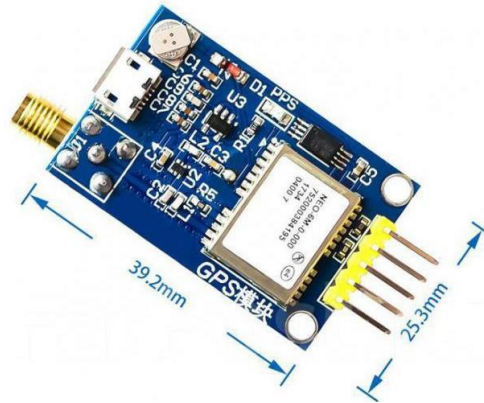


Рисунок 2.2 – NEO-6M SMA

Після живлення, модуль NEO-6M автоматично починає пошук супутників. Щоб визначити місцезнаходження, модулю потрібно отримати сигнали як мінімум від чотирьох супутників і цей процес може зайняти декілька хвилин в залежності від умов прийому. Чіп обробляє отримані сигнали та обчислює координати (широту, довготу, висоту) і час.

Таблиця 4 – Характеристики NEO-6M

Напруга живлення	3.3V - 5V
Споживання струму	45 мА
Точність позиціонування	2.5 метра (при хороших умовах прийому сигналу)

Через UART інтерфейс модуль передає дані про місцезнаходження до Arduino Nano. Мікроконтролер обробляє ці дані, зберігає їх або передає через GSM модуль на віддалений сервер або мобільний телефон.

GSM модуль SIM800 – це компактний модуль (рис. 2.3), який забезпечує передачу даних через мережу мобільного зв'язку, що дозволяє надсилати координати, отримані від GPS модуля NEO-6M, на віддалений сервер або мобільний телефон. Він підтримує функції голосових дзвінків, SMS, GPRS та інтернет-з'єднання. Його просте підключення та використання з Arduino Nano дозволяє швидко інтегрувати його у систему для ефективного моніторингу місцезнаходження домашнього улюбленця.

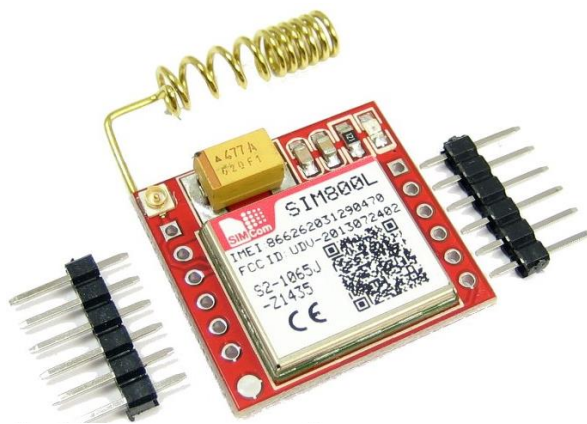


Рисунок 2.3 – SIM-800L

Після підключення живлення модуль автоматично починає ініціалізацію та реєстрацію в мережі мобільного оператора. Після успішної реєстрації модуль може відправляти і приймати SMS, здійснювати голосові дзвінки та встановлювати інтернет-з'єднання через команди AT.

Таблиця 5 – SIM800L

Напруга живлення	3.4V - 4.4V
Споживання струму	1mA – 2A
Частоти	Підтримка діапазонів GSM (850/900/1800/1900 MHz)
Підтримка протоколів	TCP/IP, HTTP, FTP, PPP

Arduino Nano надсилає команди AT до модуля для перевірки його стану та реєстрації в мережі. SIM800 приймає дані про координати від мікроконтролера та відправляє їх через SMS або інтернет на віддалений сервер.

Живлення системи буде відбуватися завдяки двом літійовим акумуляторам формату 18650 (рис. 2.4). Це надійне та поширене рішення для портативних електронних пристроїв, яке забезпечує високу ємність та стабільну напругу.



Рисунок 2.4 – Акумулятори формату 18650

Вибір пав саме на цей формат акумулятора, через його переваги, що є актуальними для розробки пристрою. По-перше акумулятори 18650 забезпечують більшу кількість енергії порівняно з іншими типами акумуляторів аналогічного розміру. Вони підтримують стабільну напругу протягом більшої частини циклу розряду та широко доступні, мають великий вибір моделей з різними характеристиками.

Зарядка пристрою буде відбуватися через роз'єм живлення на 9V. Задля забезпечення безпечної експлуатації літій-іонних акумуляторів, та контролю процесу зарядки і розрядки, буде використано контролер BMS NX-2S-A10 (рис. 2.5). Він захищає від перезаряду, перегріву, короткого замикання та інших несправностей.

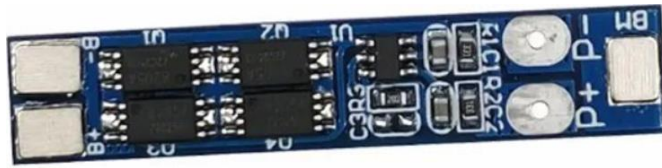


Рисунок 2.5 – НХ-2S-A10

Іншим важливим елементом у системі живлення є перетворювач напруги. MP1482 (рис.2.6) є високоефективним понижувальним перетворювачем і компактним рішенням для пониження напруги в системах з живленням від літій-іонних акумуляторів. Він забезпечує стабільне живлення для системи, зберігаючи високу ефективність і надійність.

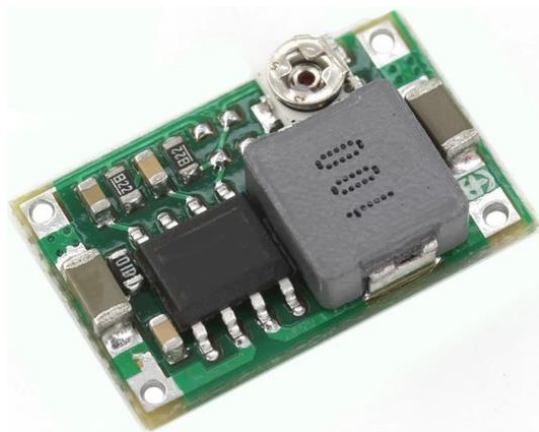


Рисунок 2.6 – MP1482

Перетворювач використовує перемикаючий регулятор, який забезпечує високу ефективність за рахунок перетворення енергії без значного виділення тепла. Вихідна напруга регулюється за допомогою зовнішніх резисторів, які встановлюють потрібну напругу на виході.

Контролер зарядки BMS 2S 8A 7.4V НХ-2S-A10 та понижувальний перетворювач MP1584 3A є критично важливими компонентами. Вони забезпечують безпеку, ефективність та надійність живлення для GPS трекера.

2.2 Вибір програмного забезпечення.

У цьому підрозділі буде розкрито тему створення веб-сайту для прийому, зберігання та відображення даних з GPS трекера на карті. В якості платформи для розробки буде використовуватися локальний веб-сервер Open Server (рис. 2.7).

Локальний веб-сервер – це програмне забезпечення, яке дозволяє створювати та тестувати веб-сайти на локальному комп'ютері перед тим, як викласти їх мережу Інтернет. Він надає такі ж можливості, як і хостинг-провайдер, але працює лише на локальній машині. Це зручно для розробки та тестування, оскільки дозволяє працювати без необхідності постійного інтернет-з'єднання. Сервер працює за наступним принципом. Програма обробляє HTTP-запити, приймає їх і відправляє відповідь. Коли ви вводите URL-адресу в браузері, веб-сервер обробляє цей запит і відправляє назад HTML-сторінку. Вона дозволяє виконувати серверний код, який генерує HTML-код або взаємодіє з базою даних, та зберігає дані сайту, такі як користувацькі облікові записи, пости, коментарі тощо.



Рисунок 2.7 – логотип Open Server

Open Server – це програмний комплекс для розробки та тестування веб-додатків, який дозволяє створювати повноцінне серверне середовище на локальному комп'ютері.

Він включає в себе всі необхідні компоненти, такі як веб-сервер Apache, інтерпретатор PHP, сервер баз даних MySQL та інші корисні інструменти для веб-розробки.

Таблиця 6 – Системні вимоги Open Server

Операційна система	Windows 7 SP1 x64 / Windows Server 2008 R2 SP1
Необхідні апаратні ресурси	от 500 МБ RAM и от 10 ГБ на диску
Системне ПО	MSVC++ 2005-2022 Redistributable Packages

Open Server працює як локальний сервер, що дозволяє розробникам тестувати та відлагоджувати веб-додатки на своєму комп'ютері без необхідності підключення до зовнішнього хостингу. Він створює серверне середовище, яке емулює роботу реального веб-сервера, забезпечуючи повну функціональність для розробки та тестування. Це дозволить працювати з сайтом локально, швидко вносити зміни та тестувати їх перед тим, як розгортати на реальному сервері. Обираючи саме цей програмний комплекс, необхідно звернути увагу на наступні особливості, що гратимуть важливу роль під час розробки:

1. комплексність: інструмент включає всі необхідні інструменти для розробки веб-додатків, зокрема Apache, PHP, MySQL, а також інші серверні компоненти. Це дозволяє зосередитися на розробці без потреби додаткового встановлення та налаштування компонентів;
2. простота використання: Завдяки інтуїтивно зрозумілому інтерфейсу, Open Server дуже простий у використанні, що робить його доступним навіть для початківців. Усі необхідні інструменти легко налаштовуються через єдину панель керування;
3. гнучкість: пакет програм підтримує різні конфігурації та версії компонентів, що дозволяє легко налаштувати середовище під конкретні потреби проекту. Це робить його ідеальним вибором для проектів, що вимагають різних налаштувань серверних служб;
4. портативність: Open Server є портативним, що дозволяє його легко переносити між різними комп'ютерами. Це особливо корисно для розробників, які працюють на кількох машинах або хочуть мати свої інструменти завжди під рукою.

Після встановлення Open Server, користувач має доступ до панелі керування, де можна запускати і зупиняти серверні служби, а також налаштовувати їх. Панель керування дозволяє:

- вибирати необхідні версії PHP, MySQL та інших компонентів;
- налаштовувати конфігураційні файли для кожного з компонентів;
- переглядати лог-файли для налагодження помилок;
- створювати та керувати віртуальними хостами для проектів.

Віртуальні хости дозволяють розробникам працювати з кількома веб-сайтами на одному сервері. В Open Server налаштування віртуальних хостів здійснюється через конфігураційні файли Apache або Nginx. Це дозволяє кожному проекту мати свій власний домен або піддомен, що спрощує тестування та розробку.

В проекті Open Server буде використовуватися для розгортання веб-сайту, який отримує дані з GPS трекера, зберігає їх у базі даних та відображає на карті за допомогою Open Street Map. Це дозволить працювати з сайтом локально, швидко вносити зміни та тестувати їх перед тим, як розгорнути на реальному сервері. Налаштування та використання сервера буде відбуватися за наступними пунктами:

1. запуск серверних служб: Після встановлення та налаштування Open Server, необхідно запустити всі необхідні серверні служби, такі як Apache, MySQL та PHP;
2. налаштування віртуальних хостів: Створити віртуальні хости для проекту, що дозволить працювати з декількома веб-додатками одночасно;
3. розгортання PHP скриптів: Розмістити PHP скрипти для обробки запитів від GPS трекера та взаємодії з базою даних;
4. налаштування бази даних: Створити необхідні таблиці у базі даних MySQL для зберігання даних, отриманих з GPS трекера.

Open Server є потужним і зручним інструментом для локальної розробки веб-додатків. Він забезпечує комплексне середовище, що включає всі

необхідні компоненти для створення, тестування та налагодження веб-сайтів. У поєднанні з Apache, PHP та MySQL, Open Server дозволяє швидко розгорнути веб-додатки та ефективно працювати над їх вдосконаленням. Це робить Open Server ідеальним вибором для реалізації проектів, що вимагають обробки та відображення даних у реальному часі.

Тепер більш детально розглянемо структуру сайту, та використання Open Street Map для відображення даних на карті та інтеграцію з сервером. Веб-додаток буде структуровано з кількох ключових компонентів. Насамперед це фронтенд – інтерфейс користувача, де дані з трекера, будуть відображатися на карті. Наступним компонентом виступає серверна частина, бекенд, яка обробляє запити, отримує дані від GSM модуля, та взаємодіє з БД. База даних виступає у ролі сховища, де зберігаються оброблені, супутникові дані, отримані від пристрою. Для створення додатку, буде використано наступні технології:

- HTML, CSS, JavaScript для фронтенду;
- PHP для бекенду;
- MySQL для бази даних;
- Open Street Map для відображення даних на карті.

OSM – це відкритий інтернет-проект (рис. 2.8) на кшталт Вікіпедії. Власне, Вікіпедія і стала джерелом натхнення для творців цих карт. Сенс полягає в тому, що будь-хто може завести собі акаунт на цьому інтернет-ресурсі і власноруч редагувати карту. Завдяки зусиллям ентузіастів карта постійно оновлюється і в багатьох регіонах має доволі детальний і деталізований вигляд. Дані збираються з різних джерел, включаючи GPS-пристрої, а потім інтегруються в єдину карту, яку можна використовувати у власних проектах.

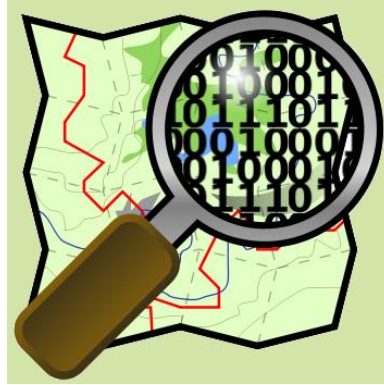


Рисунок 2.8 – Логотип проекту OSM

Створення веб-сайту для прийому, зберігання та відображення даних з GPS трекера за допомогою Open Server та Open Street Map включає налаштування локального сервера, створення бази даних, розробку бекенду на PHP та фронтенду на HTML, CSS і JavaScript з використанням бібліотеки Leaflet.js для інтеграції з OSM. Цей підхід дозволяє ефективно відображати дані трекера в реальному часі, забезпечуючи повну функціональність для проекту.

2.3 Вибір системи управління базами даних.

MySQL – це СУБД, яка використовує мову SQL (рис. 2.9) для доступу та маніпуляції даними. Вона є однією з найпопулярніших баз даних завдяки своїй надійності, продуктивності та простоті у використанні. Система володіє особливими характеристиками, які є перевагою у виборі саме цієї СУБД:

1. відкритий код: MySQL є програмою з відкритим кодом, що означає, що вона безкоштовна для використання та може бути змінена відповідно до потреб користувача;
2. портативність: MySQL працює на багатьох платформах, включаючи Windows, Linux, і macOS;
3. продуктивність: MySQL забезпечує високу продуктивність та масштабованість, що дозволяє працювати з великими обсягами даних;

4. безпека: MySQL підтримує різні методи автентифікації та шифрування, що забезпечує захист даних.



Рисунок 2.9 – Логотип MySQL

MySQL є реляційною базою даних, що означає, що дані організовані у вигляді таблиць. Таблиці складаються з рядків і стовпців, де кожен рядок представляє запис, а кожен стовпець - атрибут цього запису. Реляційна модель дозволяє зв'язувати дані між різними таблицями за допомогою ключів, що забезпечує ефективне зберігання та доступ до інформації. СУБД складається з наступних основних компонентів:

1. сервер MySQL: Основний компонент, що обробляє всі запити до бази даних.
2. клієнт MySQL: Інструмент командного рядка, який дозволяє взаємодіяти з сервером MySQL.
3. конектор MySQL: Набір бібліотек, що дозволяє додаткам взаємодіяти з MySQL за допомогою різних мов програмування.

Модель клієнт-сервер (рис. 2.10) є основною архітектурною моделлю, яка використовується для організації роботи з базами даних MySQL. У цій моделі сервер надає сервіси та ресурси, а клієнт їх використовує, посылаючи серверу запити та отримуючи відповіді. Модель складається з двох основних понять: сервер та клієнт. В першому понятті йдеться про центральний компонент, який керує доступом до бази даних. Сервер MySQL обробляє запити клієнтів, виконує операції з базою даних і повертає результати. У другому – про програму або пристрій, які взаємодіють із сервером, відправляючи запити та

отримуючи відповіді. Клієнти можуть бути різними: від командного рядка MySQL до веб-додатків, що працюють на PHP.

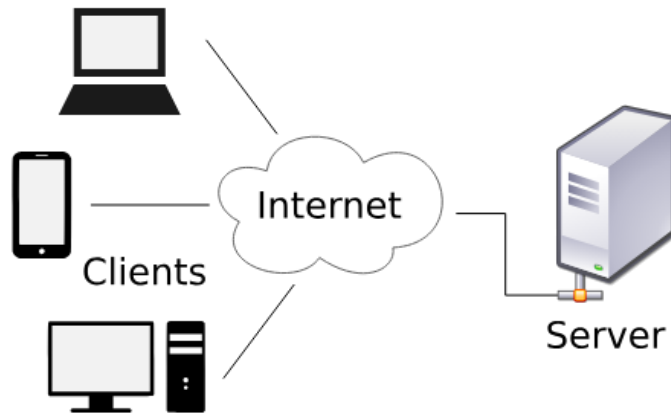


Рисунок 2.10 – Архітектура моделі клієнт-сервер

PHP – це мова сценаріїв загального призначення, особливо підходяща для веб-розробки і може бути вбудована у HTML. Вона широко використовується для створення динамічних веб-сторінок і веб-додатків. PHP має безліч функцій для взаємодії з різними системами управління базами даних. PHP (рис. 2.11) має вбудовану підтримку для роботи з СУБД, надаючи кілька способів для підключення, виконання запитів і обробки результатів. Основні способи взаємодії з MySQL у PHP включають використання бібліотек MySQLi та PDO.



Рисунок 2.11 – Логотип PHP

MySQLi надає об'єктно-орієнтований інтерфейс і функціональний інтерфейс для роботи з MySQL. Ось приклад використання MySQLi:

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "gps_database";

if ($_SERVER["REQUEST_METHOD"] == "GET") {
    $latitude = $_GET['latitude'];
    $longitude = $_GET['longitude'];
    $timestamp = $_GET['timestamp'];
    $conn = new mysqli($servername, $username, $password, $dbname);
    if ($conn->connect_error) {
        die("Підключення не вдалося: " . $conn->connect_error); }
    $stmt = $conn->prepare("INSERT INTO gps_data (latitude, longitude, timestamp)
VALUES (?, ?, ?)");
    $stmt->bind_param("sss", $latitude, $longitude, $timestamp);
    if ($stmt->execute()) {
        echo "Дані успішно збережені";
    } else {
        echo "Помилка: " . $stmt->error; }
    $stmt->close();
    $conn->close();
} else {
    echo "Неправильний метод запиту";
}
?>
```

Взаємодія з MySQL дозволяє PHP-скриптам ефективно обробляти та зберігати дані, що є критично важливим для вашого проекту GPS-трекера. PHP забезпечує обробку запитів від трекера, управління даними в базі даних MySQL і створення веб-інтерфейсу для візуалізації даних.

Після встановлення MySQL, або як у випадку з локальним сервером Open Server, він вже був у комплекті, необхідно запуснути сервер СУБД. Є можливість отримати доступ до MySQL через командний рядок або за допомогою інструменту phpMyAdmin, який також входить в комплект Open Server.

phpMyAdmin – це веб-додаток, написаний на PHP, та складається з двох основних компонентів: клієнтської, що уявляє собою інтерфейс, доступний через браузер та серверної, набором скриптів PHP, що взаємодіють з сервером MySQL для виконання запитів. Додаток дозволяє виконувати різні адміністративні завдання в MySQL. Він надає користувачу можливість створювати, змінювати, видаляти складові бази даних, такі як таблиці, поля, індекси, тощо. Завдяки phpMyAdmin можна керувати користувачами БД та їхніми правами доступу, а також виконувати SQL-запити.

При використанні Open Server, phpMyAdmin вже включений в комплект і налаштований для роботи. Після запуску Open Server, є можливість отримати доступ до phpMyAdmin через веб-інтерфейс.

Веб-додаток має наступні характеристики:

1. Веб-інтерфейс: Дозволяє взаємодіяти з MySQL через браузер.
2. Підтримка операцій з базами даних: Створення, копіювання, вилучення, перейменування і налаштування баз даних.
3. Підтримка таблиць: Створення, копіювання, вилучення, перейменування, зміна структури і індексів таблиць.
4. Виконання SQL-запитів: Можливість виконання будь-яких SQL-запитів через інтерфейс.
5. Керування користувачами: Додавання, видалення та зміна прав доступу користувачів MySQL.
6. Імпорт і експорт даних: Підтримка різних форматів даних, таких як SQL, CSV, Excel, XML, PDF та інші.

Додаток є потужним інструментом для управління базами даних MySQL, який значно спрощує адміністрування бази даних завдяки зручному веб-інтерфейсу. Він дозволяє виконувати всі необхідні операції з базами даних, таблицями та користувачами, забезпечуючи швидкий і ефективний доступ до даних. Використання phpMyAdmin проекті забезпечить легке керування базою даних та інтеграцію даних з GPS трекера на веб-сервер.

Задля коректної роботи сервера, передавання та обробки даних по протоколу HTTP, буде використано програмне забезпечення Apache HTTP Server (рис. 2.12), ке забезпечує обслуговування веб-сторінок. Він обробляє запити клієнтів (зазвичай через браузерери) та відповідає, відправляючи потрібні файли (HTML, CSS, JavaScript, зображення тощо). Apache може також виконувати серверні скрипти, такі як PHP, для динамічного генерування контенту.



Рисунок 2.12 – Логотип Apache

Apache є проектом з відкритим кодом, що означає, що його можна безкоштовно використовувати та змінювати. Він підтримує різні модулі, які можна підключати для розширення функціональності (наприклад, `mod_php` для підтримки PHP, `mod_ssl` для SSL/TLS тощо). Програмне забезпечення обробляє велику кількість запитів та не дивлячись на це, здатне забезпечувати стабільну роботу сайтів. Apache працює на більшості операційних систем, включаючи Linux, Windows та macOS. Конфігураційні файли дозволяють детально налаштувати сервер для задоволення специфічних потреб.

Apache є критично важливим компонентом у створенні веб-додатку, забезпечуючи обслуговування веб-сторінки та взаємодію з серверними скриптами. У поєднанні з MySQL та PHP, програмне забезпечення дозволяє створювати динамічні додатки, такі як система відстеження місцеположення на основі GPS. Коли трекер відправляє дані на сервер, Apache приймає ці дані у вигляді HTTP-запитів і передає їх до відповідних PHP-скриптів, які обробляють отриману інформацію і зберігають її в базі даних.

3. РОБОТА З КОМПОНЕНТАМИ

3.1 Побудова схеми та розробка прошивки для Arduino

У цьому підрозділі буде розглянуто конструкцію GPS трекера на базі Arduino Nano, включаючи підключення комплектуючих, питання енергоспоживання, а також процес розробки прошивки для мікроконтролера. Для початку було вирішено питання схеми пристрою. При розробці враховувалися габарити компонентів та основним критерієм була компактність приладу.

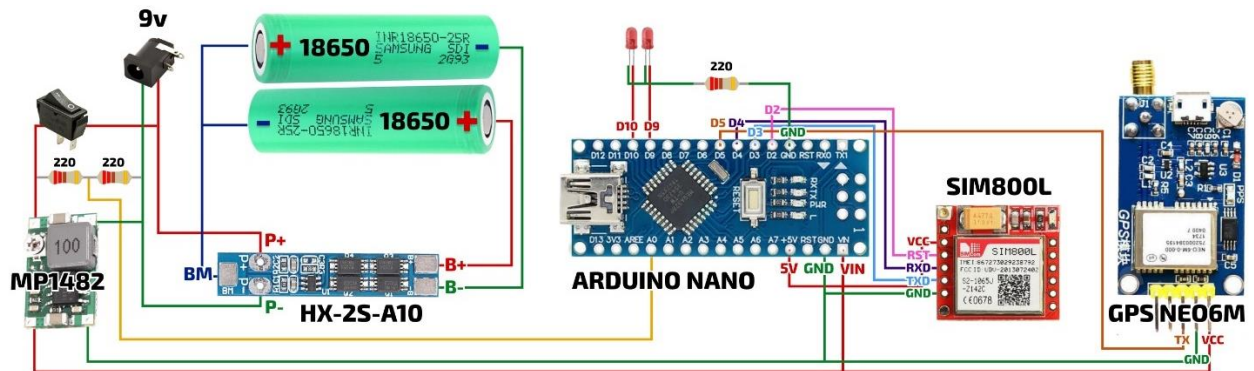


Рисунок 3.1 – Схема GPS трекера

На даній схемі показано підключення усіх підібраних апаратних компонентів: батарей, зарядного модуля, стабілізатора напруги, контролера Arduino Nano, GSM модуля SIM800L і GPS модуля NEO-6M. Також було підібрано декілька резисторів, для стабілізації напруги та захисту компонентів від перенавантаження. Особливо це стосується світлодіодів, необхідних для індикації стану роботи трекера. Підібране рішення є ефективним та надійним для створення компактного та функціонального GPS трекера.

Живлення системи відбувається завдяки двом літійовим акумуляторам 18650. Вони з'єднані послідовно, що дає сумарну напругу близько 7,4V. Але для роботи пристрою така напруга зavelика, тому для безпечної та стабільної

роботи компонентів підключено стабілізатор напруги MP1482. Використання стабілізатора дозволяє отримати стабільну напругу 5V з джерела живлення 7,4V, що важливо для правильного функціонування всіх електронних компонентів. Вхідний контакт (IN) підключений до виходу P+ від зарядного модуля NH-2S-A10 через резистори 220 Ом. Вихідний контакт (OUT) підключений до Arduino Nano для забезпечення стабільної напруги 5V. Зарядний модуль NH-2S-A10 забезпечує балансування і захист літієвих батарей під час заряджання, що запобігає їх пошкодженню, у випадку короткого замикання, або перезарядки. Контролер BMS підключено наступним чином: контакти B+ і B- підключені до батарей 18650, BM підключений до середньої точки між двома батареями, а P+ і P- підключені до стабілізатора MP1482, та роз'єму на 9V. Таке підключення компонентів живлення, дозволяє налагодити систему та стабілізувати напругу у роботі GPS трекера.

Також на схемі можна побачити три резистора, з номіналом 220 Ом, які підключені послідовно зі світлодіодами (LED) та у силовому ланцюзі (перед MP1482). Використання резисторів з такими значеннями забезпечує надійність і безпеку роботи всієї системи, запобігаючи пошкодженню компонентів через надмірний струм або напругу. Резистор перед світлодіодами обмежує струм, щоб запобігти їх пошкодженню. Зазвичай світлодіод має падіння напруги близько 2-3 Вольт і максимальний струм близько 20 mA. З огляду на те, що Arduino працює на 5 В, резистори потрібні для обмеження струму до безпечного рівня. Використання резисторів на 220 Ом гарантує, що через світлодіоди протікатиме струм приблизно 15-20 mA, що є достатньо для безпечної роботи. У силовому ланцюзі, резистори обмежують початковий струм заряджання і захищають стабілізатор від надмірного споживання струму.

Наступні компоненти, такі як GSM модуль SIM800L та GPS модуль NEO-6M, підключені згідно з даташитом модулів. З'єднання відбувається через піни D5-D2 на Arduino, де D5 підключений до GPS через TX. Живлення відбувається через VIN порт. Інші піни використовуються платою SIM800L.

D4 для RX, D3 для TX та D2 для RST. Живлення відбувається через пін 5V. Правильне підключення піни RX/TX забезпечує коректний обмін даними між Arduino Nano і модулями SIM800L та NEO-6M, дозволяючи отримувати GPS координати і передавати їх через GSM.

Після підключення усіх необхідних складових пристрою, наступним кроком відбувається прошивка мікроконтролера. Arduino IDE – сновне середовище для розробки та завантаження коду на Arduino. Під час розробки коду, було використано декілька бібліотек, що є обов’язковими для правильної роботи трекера. TinyGPS - це бібліотека з відкритим кодом для роботи з GPS-модулями, написана на мові C++. Вона використовується для зчитування даних з GPS-модулів:

- широта та довгота;
- час GPS;
- швидкість та напрямок руху;
- кількість супутників;
- точність визначення місцезнаходження;
- стан GPS (зафіксовано, шукається, не фіксується);
- інші дані, такі як висота, швидкість підйому/спускання.

TinyGPS підходить для широкого спектру GPS-модулів та може використовуватися з різними платформами мікроконтролерів. Бібліотека володіє широким спектром функцій, що роблять її цінним інструментом для роботи з GPS-модулями. Окрім зчитування, фреймворк також обробляє отримані супутникові дані. Відстань, напрямок, швидкість та час у дорозі, тощо. TinyGPS підтримує формат NMEA, який є стандартом для обміну даними GPS. Вона має простий інтерфейс API, що робить її легкою у використанні для початківців, та є безкоштовна бібліотека з відкритим кодом. Ось приклад зчитування основних даних GPS:

```

#include <TinyGPS.h>
SoftwareSerial ss(9, 10); // Підключення RX до 9, TX до 10
void setup() {
    Serial.begin(9600);
    ss.begin(9600);
}
void loop() {
    while (ss.available()) {
        char c = ss.read();
        if (gps.encode(c)) {
            // Нове речення NMEA отримано
            float lat = gps.f_get_latitude();
            float lon = gps.f_get_longitude();
            int course = gps.f_get_course();
            int satellites = gps.get_satellites_used();
            Serial.print("Широта: ");
            Serial.print(lat, 6);
            Serial.print(" Довгота: ");
            Serial.print(lon, 6);
        }
    }
}

```

Цей код ініціалізує бібліотеку TinyGPS та SoftwareSerial. В циклі loop() код зчитує дані з GPS-модулю через SoftwareSerial. Функція gps.encode() використовується для декодування даних NMEA. Якщо отримано нове речення NMEA, код витягує широту та довготу з об'єкта gps.

Для підключення GPS-модуля NEO-6M потрібен послідовний порт, але всі апаратні послідовні порти Arduino вже використовуються (Serial для налагодження, Serial1 для GSM-модему). Для вирішення цієї проблеми, була використана наступна бібліотека SoftwareSerial. Це бібліотека з відкритим кодом для Arduino, яка дозволяє емулювати послідовний порт на будь-яких вільних цифрових виводах. Наприклад, вона надає можливість вибрати будь-які два вільні цифрові виводи для RX (приймання) та TX (передачі) даних. У випадку із Arduino Nano, де кількість портів дуже обмежена, цей інструмент є незамінним у розробці. SoftwareSerial сумісна з більшістю послідовних пристроїв,

включаючи GPS-модеми, GSM-модеми, модулі Bluetooth, мікроконтролери та також є безкоштовною.

```
#include <SoftwareSerial.h>
SoftwareSerial ss(10, 11); // RX до 10, TX до 11
void setup() {
  Serial.begin(9600); // Ініціалізувати послідовний порт для налагодження
  ss.begin(9600); // Ініціалізувати SoftwareSerial
}
```

Цей рядок створює об'єкт SoftwareSerial, який використовує цифрові виводи 10 та 11 для RX та TX відповідно. Відбувається ініціалізація послідовного порта, після чого ss.begin(9600) ініціалізує SoftwareSerial з тією ж швидкістю передачі даних, що й апаратний послідовний порт.

Для налагодження роботи GSM модуля SIM800L, було використано AT-команди. AT-команди (від англ. Attention commands) - це набір команд, які використовуються для керування GSM-модемами. Ці команди дозволяють дозволяють надсилати SMS-повідомлення на інші номери телефонів, здійснювати дзвінки, підключатися до Інтернету через GPRS або UMTS, налаштовувати різні параметри, такі як PIN-код, номер IMEI, режим роботи та надають широкий спектр функцій для керування модемами. Формат AT-команд мають стандартизований формат, що робить їх легкими у використанні з різними GSM-модемами. Роздивимось декілька прикладів поширених AT-команд:

- AT: Ця команда використовується для перевірки того, чи підключений модем і чи готовий він до прийому команд;
- AT+CMGS: Ця команда використовується для надсилання SMS-повідомлення;
- AT+CALL: Ця команда використовується для здійснення голосового дзвінка;
- AT+CFUN: Ця команда використовується для вмикання або вимкнення GSM-модуля.

Ось приклад того, як використовується АТ-команда для надсилання SMS:

```
sendATCommand("AT+CMGS="+380671234567+"\r\n", false); // Вибір номера телефону
delay(100);
sendATCommand("Hello from Arduino!\r\n", false); // Введення тексту SMS
delay(100);
sendATCommand("\x1A", true); // Відправка SMS
```

Для розробки прошивки мікроконтролера, було вирішено створити перелік вимог до програмної частини. Цей перелік надасть змогу ретельно продумати загальний алгоритм роботи пристрою та підібрати компоненти апаратної частини так, щоб з ними було зручніше працювати з погляду програмної реалізації. Алгоритм буде побудовано методом частинних цілей. Цей метод полягає у тому, що глобальна велика задача ділиться (якщо це можливо) на окремі задачі. Якщо велику задачу не можна досягнути, зрозуміти, як її розв'язувати, то для кожної з поділених задач може існувати давно відомий алгоритм розв'язку, або пошук такого алгоритму є значно легшою задачею.

Перш за все, було вирішено розбити програмний код на окремі розділи, де кожний розділ це функція, яка відповідає за невеликий алгоритм роботи пристрою. Роздивимось кожну з цих функцій окремо, для кращого розуміння загальної роботи, програмної частини проекту.

Функція Setup ініціалізує послідовний зв'язок для Arduino Nano, GPS-модуля та GSM-модуля. Встановлюється стандартний інтервал для надсилання GPS-даних на веб-сервер. Налаштовується модуль SIM800L для GPRS-з'єднання та надсилає АТ-команди для встановлення з'єднання з стільниковою мережею. Відбувається ініціалізація світлодіодів LED для візуального індикації стану. Кожен світлодіод відповідає за окремий модуль.

Світлодіод SIM800L:

- зелений: SIM-карта вставлена та активна;
- миготливий зелений: Відбувається з'єднання з GPRS-мережею.

Світлодіод NEO-6M:

- зелений: GPS-модуль отримує сигнал від супутників і готовий до роботи;
- миготливий зелений: GPS-модуль шукає сигнал від супутників.

Функція `sendATCommand` надсилає AT-команду модулю SIM800L і чекає на відповідь. Якщо очікування активовано, він розбирає відповідь і друкує її на монітор послідовного порту, після чого повертає рядок відповіді.

Функція `waitResponse` чекає на відповідь від модуля SIM800L протягом 30-секундного тайм-ауту. Якщо отримано відповідь, він зберігає її у змінній і перевіряє наявність ключових слів, що свідчать про стан модема, силу сигналу та дані про місцезнаходження. Якщо протягом тайм-ауту не отримано відповіді, друкує повідомлення про тайм-аут і запускає скидання пристрою, після чого повертає рядок відповіді.

Функція `gps_info` перемикає послідовний зв'язок з GSM-модуля на GPS-модуль. Вона читає дані з GPS-модуля та перевіряє наявність нової інформації. Якщо наявні нові дані, вона витягує з даних GPS широту та довготу та друкує ці дані на монітор послідовного порту. Оновлює змінну `GpsStatus`, щоб вказати доступність даних GPS, після чого перемикає послідовний зв'язок назад на GSM-модуль.

Функція `opros_dev` надсилає AT-команди модулю SIM800L для опитування стану модема, сили сигналу та даних про місцезнаходження. Вона розбирає відповіді та оновлює змінні `simvseti`, `SvyazProcent`, `gsmlat` та `gsmlon` відповідно.

Функція `loop` періодично блимає світлодіодами, щоб вказати стан SIM800L та NEO-6M. Кожні 2 секунди вона викликає функцію `gps_info` для оновлення даних GPS. Кожні 20 секунд викликає функцію `opros_dev` для опитування стану SIM, сили сигналу та даних про місцезнаходження з GSM-мережі. За деякий інтервал, вона викликає функцію `send_site` для надсилання останніх даних GPS на веб-сервер.

Функція `send_site` визначає поточне місцезнаходження на основі змінної `GpsStatus`. Якщо дані доступні, вона надсилає запит на сервер.

3.2 Розробка веб-додатку

Процес створення веб-додатку, як і у випадку зі створення програмного коду для контролера Arduino, було поділено на декілька кроків. По-перше був налаштований локальний веб-сервер, завдяки програмному засобу Open Server. Після встановлення необхідного програмного обладнання, веб-сервер було запущено, де у якості хоста виступав персональний комп'ютер.

У середовищі розробки phpMyAdmin, була створена база даних `tracker_db`. Вона містить у собі таблицю для зберігання даних, створена виконанням простого SQL-запит для створення таблиці:

```
CREATE TABLE tracking_data (
    id INT AUTO_INCREMENT PRIMARY KEY,
    latitude VARCHAR(15),
    longitude VARCHAR(15),
    voltage FLOAT,
    signal_strength INT,
    location_source VARCHAR(10),
    timestamp TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

Таблиця містить у собі п'ять змінних: широту та довготу, напругу акумулятора, відсоток сили сигналу та місцезнаходження GPS трекера. Значення для змінних надходять від пристрою, у вигляді попередньо перелічених параметрів. Для прийому даних було розроблено окремий PHP скрипт для прийому даних від трекера та зберігання їх у базі даних:

```
if ($_SERVER["REQUEST_METHOD"] == "GET") {
    $latitude = $_GET['lat'];
    $longitude = $_GET['lon'];
    $voltage = $_GET['volt'];
    $signal = $_GET['signal'];
    $source = $_GET['source'];
```

```

$sql = "INSERT INTO tracking_data (latitude, longitude, voltage,
signal_strength, location_source)
VALUES ('$latitude', '$longitude', '$voltage', '$signal', '$source')";
if ($conn->query($sql) === TRUE) {
    echo "New record created successfully";
}

```

Наступним кроком було створено сам веб-додаток, який зможе коректно відображати дані з БД на мапі. Для виконання цієї задачі, була написана проста HTML сторінка. Основним інструментом для відстеження GPS трекера є API Open Street Map.

Tracker Map

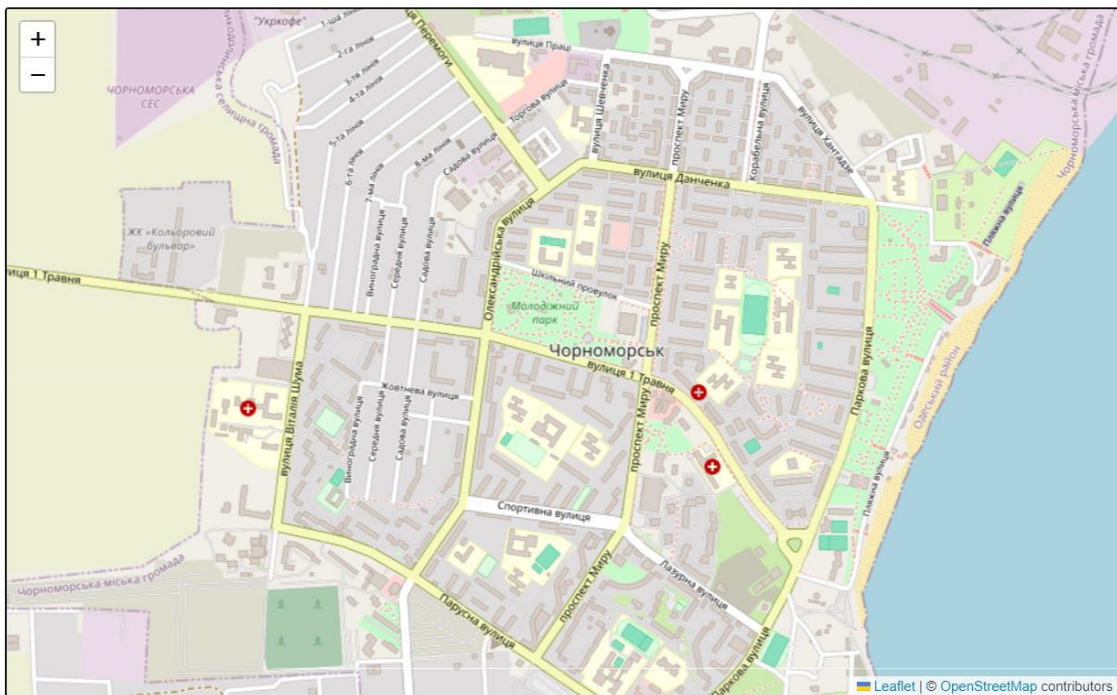


Рисунок 3.2 – Веб-додаток

ВИСНОВКИ

Метою проекту була розробка системи віддаленого моніторингу за домашнім улюбленцем на базі Arduino, та вирішення проблем подібних аналогів: велика ціна, або недостатньо якісна робота пристрою. Розроблений GPS трекер дозволяє відстежувати пересування тварини із бажаною точністю та швидкістю. Проект бюджетний, та задовольняє потреби віддаленого моніторингу, завдяки можливості переглядати дані о місцезнаходженні через спеціально розроблений веб-додаток. Така схема може бути використана в різних проєктах, від трекерів для транспортних засобів до систем моніторингу в реальному часі, що підвищує її практичну цінність і універсальність.

Недоліками конструкції є відсутність корпусу для пристрою та недостатньо надійний і зручний веб-додаток. Всі дані недоліки конструкції мають можливість майбутнього доопрацювання. В результаті вийде зручний та компактний GPS-трекер, що матиме велику кількість переваг перед іншими аналогами. Всі цілі, поставлені в момент становлення завдання, були виконані в повному обсязі.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Огляд платформ на базі Arduino. [Електронний ресурс]. – Режим доступу: <http://dspace.pnpu.edu.ua/bitstream/123456789/5112> – дата звернення 15.04.2024.
2. Документація Arduino Nano. [Електронний ресурс]. – Режим доступу: <https://docs.arduino.cc/hardware/nano> – дата звернення 21.04.2024.
3. Бібліотека SoftwareSerial. [Електронний ресурс]. – Режим доступу: <https://wiki-content.arduino.cc/en/Tutorial/LibraryExamples/> – дата звернення 23.04.2024.
4. Бібліотека HttpClient. [Електронний ресурс]. – Режим доступу: <https://www.arduino.cc/reference/en/libraries/httpclient/> – дата звернення 03.05.2024.
5. Створення веб-додатку HTML, CSS, JavaScript. [Електронний ресурс]. – Режим доступу: <https://developer.mozilla.org> – дата звернення 11.05.2024.
6. Бази даних. [Електронний ресурс]. – Режим доступу: <https://www.mongodb.com/docs/> – дата звернення 15.05.2024.
7. О.В. Глухов, О.О. Кравчук, Є.В. Левченко – Вивчення властивостей мікроконтролерів і електронних систем на базі платформи Ардуіно. – Навчальний посібник для студентів ВНЗ. –117с.
8. М. Ю. Лосєв, В. В. Федько. – Бази даних. – Навчально-практичний посібник для самостійної роботи студентів. –233 с.