

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ВОДНОГО ГОСПОДАРСТВА ТА
ПРИРОДОКОРИСТУВАННЯ

**Навчально-науковий інститут кібернетики, інформаційних
технологій та інженерії**

"До захисту допущена"

Зав. кафедри комп'ютерних наук та
прикладної математики

« ____ » _____ 2023 р.

КВАЛІФІКАЦІЙНА РОБОТА

Розробка транспортного локатору ARDTRACKER GPS

Виконала Касперська Анастасія Олександрівна

(прізвище, ім'я, по батькові)

(підпис)

група ІІЗ – 41

Керівник: к. ек. н., доцент кафедри комп'ютерних наук та прикладної
математики, Бачишина Л.Д.

(науковий ступінь, вчене звання, посада, прізвище, ініціали)

(підпис)

Рівне – 2024

ЗМІСТ

ЗМІСТ	2
РЕФЕРАТ.....	4
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	5
ВСТУП.....	6
РОЗДІЛ 1. РОЗУМІННЯ ПОНЯТТЯ ІОТ ТА ВИБІР МІКРОКОНТРОЛЛЕРА ...	8
1.1 Складові пристроїв Інтернету речей	8
1.2 Що впливає на вибір мікронтроллера	9
1.3 Особливості платформи розробки Arduino та огляд Arduino UNO	11
РОЗДІЛ 2. ЗАСТОСУВАННЯ СИСТЕМИ ГЕОЛОКАЦІЇ	15
2.1 Поняття глобальної системи позиціонування	15
2.2 Принцип роботи супутникових систем навігації.....	18
2.3 Варіанти GPS-модулей.....	19
2.4 Загальний огляд GPS-модулю NEO-6M.....	20
РОЗДІЛ 3. ЗАСТОСУВАННЯ СИСТЕМИ РАДІОЗВ'ЯЗКУ	27
3.1 Глобальна система мобільного зв'язку	27
3.2 Загальний сервіс пакетної радіопередачі	30
3.3 «Спілкування» за допомогою AT-команд.....	35
3.4 Огляд GSM/GPRS модуля SIM900A	38
РОЗДІЛ 4. РОЗРОБКА ПРИСТРОЮ	44
4.1 Основні етапи проекту	44
4.2 Допоміжні компоненти пристрою	45
4.3 Моделювання схеми	47
4.4 Підключення пристрою	48

4.5 Функціонал пристрою та пояснення коду	49
ВИСНОВКИ	59
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	60
ДОДАТКИ	62

РЕФЕРАТ

Кваліфікаційна робота: 68 сторінок, 30 рисунків, 8 таблиць, 13 джерел.

Метою кваліфікаційної роботи є розробка пристрою геолокації на базі GPS трекінгу та технології GSM.

Об'єктом дослідження є процес відправки і коректної форматизації повідомлень між контролером та модулями.

Предмет дослідження - мікроконтролер, який оснащено модулями для отримання та дистанційної передачі його розташування за допомогою обміну даними GPS бездротовими системами зв'язку стандарту GSM.

Методи дослідження – платформа Arduino, технологія GPS, технологія GSM/GPRS, середовище проектування Fritzing.

Новизна та практична цінність отриманих результатів. Визначення точних координат розташування об'єкту протягом багатьох років мало величезний попит. Це питання набуває ще більшої актуальності у воєнний час і виникає необхідність у бюджетних девайсах з відкритим кодом.

Ключові слова: геолокація, супутниковий зв'язок, мікроконтролер, Arduino, GPS-трекер, модуль зв'язку GSM/GPRS .

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

- IOT** – Internet of Things , концепція мережі, що складається із взаємозв'язаних фізичних пристроїв, які мають вбудовані давачі, програмне забезпечення, що дозволяє здійснювати передачу і обмін даними між фізичним світом і комп'ютерними системами в автоматичному режимі.
- GPS** – Global Positioning System, сукупність радіоелектронних засобів, що дозволяє визначати положення та швидкість руху об'єкта на поверхні Землі або в атмосфері.
- GSM** – Global System for Mobile Communications, міжнародний стандарт для мобільного цифрового стільникового зв'язку з високим рівнем безпеки за рахунок шифрування з відкритим ключем.
- GPRS** – General Packet Radio Service, надбудова над технологією мобільного зв'язку GSM , що здійснює пакетну передачу даних . Дозволяє користувачеві мережі стільникового зв'язку здійснювати обмін даними з іншими пристроями в мережі GSM та зовнішніми мережами, у тому числі Інтернет .
- NMEA 0183** – комбінований стандарт, який визначає як електричні так і функціональні характеристики комунікаційного протоколу для обміну інформацією між пристроями морського електронного обладнання.
- TinyGPS++** – компактна бібліотека Arduino IDE для аналізу та форматування речень NMEA (GPS).

ВСТУП

Завдання визначити місцезнаходження живих та неживих об'єктів у всі часи було неабияк актуальним. Супутникові навігаційні системи, незважаючи на виключно військове минуле, назавжди закріпились в розумі сучасного суспільства, як абсолютно повсякденна утилітарна функція. Практично всі люди щодня користуються послугами супутникових систем у своїх мобільних пристроях та автомобілях. У наш час, аббревіатура GPS є чи не синонімом для навігаційної супутникової системи і носить найбільш масовий характер по всій земній кулі.

Актуальність використання систем позиціонування зумовлена тим, що в наші дні персональні навігатори практично незамінні в житті сучасної людини. Під час поїздки новою місцевістю без навігатора обійтися справді неможливо - і не важливо, чи буде це мегаполіс із безліччю вулиць, чи сільська місцевість із стежинами. Використання паперових карт або варіант "запитати у кого-небудь" давно вважаються вже чимось застарілим: у кілька разів простіше задати маршрут у своєму навігаторі та швидко дістатися до пункту призначення.

Отже метою нашого проекту є розробка пристрою геолокації на базі GPS трекінгу та технології GSM.

Для досягнення цієї мети нам необхідно виконати наступні завдання:

- детально ознайомитись з документацією стосовно роботи GPS та GSM систем;
- визначити основні недоліки та переваги досліджуваних систем;
- відповідно до поставленої задачі обрати платформу розробки та контролер;
- провести аналіз доступних варіантів GPS, GSM, GPRS модулів, який буде найоптимальнішим для розробки;
- здійснити реалізацію програмного коду для коректного налаштування пристрою.

Представлений у даній роботі пристрій ARDTRACKER GPS може бути використаний для навігації під час подорожей морем, повітрям і землею. Також безліч людей використовують GPS трекери у туризмі для знаходження шляху у незнайомій місцевості. Вони користуються неабияким попитом серед працівників служб таксі, а також у щоденному використанні для моніторингу будь-яких видів транспорту. GPS є корисним інструментом у науці для надання даних, які раніше ніколи не були доступні в такій кількості та з подібним ступенем точності. Вчені використовують GPS для вимірювання руху тектонічних плит Землі або арктичних крижаних покривів, вулканічної активності .

Трекер допоможе точно визначати розташування об'єкту з точними координатами, швидкістю переміщення, часом та датою.

РОЗДІЛ 1. РОЗУМІННЯ ПОНЯТТЯ ІОТ ТА ВИБІР МІКРОКОНТРОЛЛЕРА

1.1 Складові пристроїв Інтернету речей

Девайс, що проектується в ході цієї кваліфікаційної роботи є, насамперед, пристроєм інтернету речей. Розумні пристрої IoT (Internet of Things) стали невід'ємною частиною нашого повсякденного життя.

Перш ніж поринути у тонкощі IoT-пристроїв, давайте почнемо з короткого огляду самого поняття "інтернету речей". Взагалі, це мережа взаємопов'язаних компонентів, які можуть взаємодіяти один з одним та обмінюватися даними через Інтернет. Вони оснащені датчиками, виконавчими механізмами та іншими передовими технологіями, які дозволяють їм збирати, аналізувати та передавати дані для найрізноманітніших цілей.

Інтернет-концепція речей (IoT) охоплює широкий спектр додатків, серед яких можна виділити розумні будинки, моніторинг охорони здоров'я, зондування навколишнього середовища, промислову автоматизацію. Використання можливостей таких пристроїв підприємствами та приватними особами може підвищити ефективність, покращити процес прийняття рішень та створити більш стійкі та взаємопов'язані системи.

Датчики є ключовими елементами IoT-пристроїв. Як наші органи чуття допомагають нам сприймати навколишній світ, так і датчики виконують роль "органів чуття" для різних пристроїв. Ці електронні компоненти IoT здатні виявляти та вимірювати такі фізичні величини, як температура, вологість, освітленість, рух та багато інших.

Датчики мають різні спеціалізації, кожна з яких призначена для вимірювання конкретного параметра. Наприклад, датчики температури визначають температуру навколишнього середовища, а датчики руху виявляють рух у межах своєї дії. Ці датчики генерують електричні сигнали, що відповідають вимірюваному параметру, і ці сигнали можуть бути оброблені та використані іншими компонентами IoT-пристроїв.

Пристрої IoT використовують модулі бездротового зв'язку для підключення та обміну даними з іншими пристроями або центральними системами. Ці модулі забезпечують надійний зв'язок, використовуючи бездротові протоколи, такі як Wi-Fi, Bluetooth, Zigbee або стільникові мережі 3G, 4G і 5G [1].

Модулі бездротового зв'язку дозволяють IoT-пристроєм передавати та отримувати дані через Інтернет або локальні мережі. Це забезпечує можливість моніторингу, контролю та координації в режимі реального часу між численною кількістю девайсів. Завдяки бездротовому зв'язку, пристрої Інтернету речей досягають високого рівня взаємодії та сумісності.

Мікроконтролери є "головним мозком" розумних пристроїв Інтернету речей. Вони поєднують в собі функції центрального процесора (CPU), пам'яті та інтерфейсів в одному чіпі. Мікроконтролери широко використовуються у вбудованих системах, таких як побутова техніка, автомобільні системи, медичні пристрої та промислові системи керування. Вони також використовуються в продуктах побутової електроніки, таких як ігрові системи, цифрові камери та аудіоплеєри. функціональністю IoT пристроїв. Вибір мікроконтролера залежить від конкретних вимог застосування.

1.2 Що впливає на вибір мікронтроллера

Типовий мікроконтролер складається з процесорного ядра, енергонезалежної та енергозалежної пам'яті, периферійних пристроїв вводу-виводу та різноманітних комунікаційних інтерфейсів. Ядро процесора відповідає за виконання запрограмованих команд і керує іншими компонентами мікроконтролера. Функціонал пам'яті мікроконтроллера полягає у зберіганні даних і програмного коду, а периферійні пристрої вводу-виводу забезпечують взаємодію із зовнішнім середовищем.

Вибір кращого контролера для проекту включає розгляд декількох факторів, заснованих на конкретних вимогах. Ось кілька ключових моментів, які слід враховувати під час вибору мікроконтролера:

1. Обчислювальна потужність

Слід врахувати потреби пристрою IoT у обчислювальній потужності. Вибирати мікроконтролер слід із потрібною обчислювальною потужністю залежно від складності завдань, які має виконувати пристрій.

2. Можливості підключення

Необхідно звернути увагу на потрібні протоколи зв'язку. Постає питання чи потрібний у девайсі стільниковий зв'язок, Wi-Fi, Bluetooth або інші типи підключення. Повинен використовуватися мікроконтролер, який підтримує потрібні стандарти підключення.

3. Споживана потужність

Обирати мікроконтролер необхідно врахувавши потреби вашого IoT-пристрою в електроживленні. Варто зупинитися на мікроконтролері з низьким енергоспоживанням та енергозберігаючими можливостями, якщо пристрій використовує батареї або він спроектований з урахуванням енергоефективності.

4. Середовище розробки

Також важливим аспектом є доступність програмних засобів мікроконтролера та простота розробки. Переконайтеся, що відоме інтегроване середовище розробки (IDE) та велика частина спільноти розробників підтримують мікроконтролер. Це гарантує наявність бібліотек, документації та допомоги для розробки та подальшого усунення несправностей.

5. Вартість та доступність

Оцінити вартість мікроконтролера та доступність комплектуючих у нашому регіоні також важливо. Мікроконтролер має знаходитися в рамках бюджету і бути легко доступним для покупки.

6. Безпека та Масштабованість

Потрібно звернути увагу на функції безпеки, що надаються мікроконтролером, особливо якщо ваша програма IoT має справу з конфіденційними даними або працює в потенційно вразливому середовищі. Серед важливих функцій, які слід підкреслити, є підтримка шифрування, безпечне завантаження та безпечні протоколи зв'язку.

Оцінка потенціалу масштабованості мікроконтролера є важливою. Він має задовольнити вимоги щодо майбутнього розширення або додаткових функціональних можливостей вашої програми IoT.

7. Надійність та якість

Репутація та надійність виробника мікроконтролера гарантує довгострокову службу пристрою. Огляди, дані про надійність та позитивні відгуки клієнтів допоможуть переконатися, що ви вибрали надійний та високоякісний мікроконтролер.

8. Сумісність та взаємодія

Якщо програма IoT вимагає інтеграції з іншими пристроями або платформами, потрібно розглянути аспекти сумісності та взаємодії. Мікроконтролер має безперешкодно взаємодіяти з іншими пристроями, хмарними платформами або протоколами, які стосуються проекту.

1.3 Особливості платформи розробки Arduino та огляд Arduino UNO

Arduino - це ефективний засіб розробки електронних пристроїв з відкритим вихідним кодом для зручної, безпосередньої взаємодії з навколишнім світом. Це апаратна платформа для взаємодії з різноманітними фізичними об'єктами [2]. Це невеличка проста плата з мікроконтролером та спеціальне середовище розробки для написання програмного коду для нього. Простота, доступність та необмежені можливості зробили Arduino універсальним пристроєм.

З допомогою платформи можна створити все, що не спаде на думку. Доречно використати для розробки різноманітних інтерактивних систем, які можуть керуватися датчиками та перемикачами. У свою чергу, ці системи будуть керувати роботою цілого ряду пристроїв, таких як двигуни, індикатори та інші додаткові елементи.

Завдяки універсальності, проекти Arduino можуть бути автономними, або працювати у взаємодії з програмним забезпеченням, встановленому на персональному комп'ютері (MaxMSP, Flash або Processing). Середовище розробки для її програмування має відкритий вихідний код та можна

використовувати абсолютно безкоштовно. Зручність, простота мови програмування та безліч доступних посібників для Arduino роблять роботу з нею невимушеною, а саму платформу шалено популярною [4]

Що стосується програмного рівня, то платформа має безкоштовне середовище розробки Arduino IDE. Мікроконтролери Arduino необхідно програмувати за допомогою мови, подібної C++. Але треба враховувати деякі відмінності: доповнити зрозумілими та простими функціями для зручного налаштування і керування платою. А вже компіляцію програмного коду з прошивкою мікроконтролера середовище розробки зробить саме. Виробником пропонується широкий вибір плат, які надають можливість реалізувати найрізноманітніші проекти. Треба лише підібрати необхідні характеристики мікроконтролера, визначитися з функціональним призначенням та кількістю на ній портів.

Однією з найпопулярніших та самою бюджетною платою Arduino вважається UNO. Попри доступну ціну, її функціоналу цілком вистачить задля реалізації переважної більшості цікавих проектів. Всі стандартні набори-конструктори Arduino укомплектовані платою UNO, яка стала своєрідним обличчям цієї компанії. Саме її і буде використано для реалізації даного проекту.

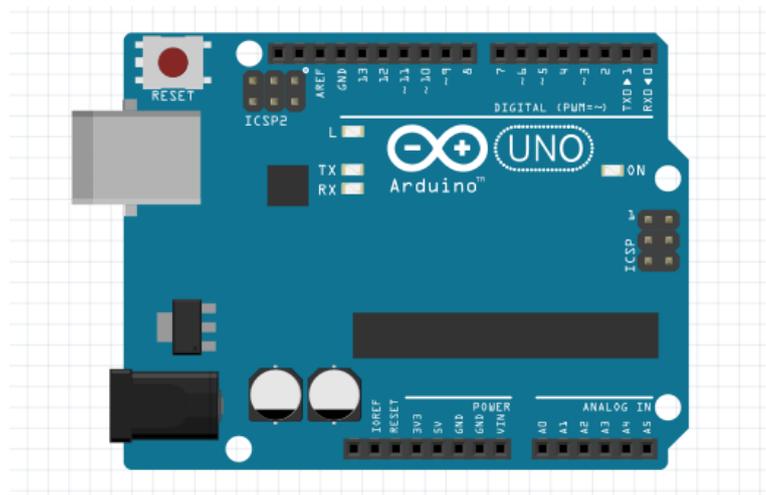


Рис.1.1. Схема плати Arduino UNO

До плати Arduino входять: мікроконтролер від Atmel AVR та необхідні елементи обв'язки для подальшого програмування та, за потреби, інтеграції з

іншими шилдами. На більшості плат встановлені лінійні стабілізатори напруги. Вони можуть бути на +3,3 В або +5 В.

Тактування здійснюється кварцовим резонатором на частоті 8 або 16 МГц. Деякі версії мають керамічний резонатор. Для збірки необхідного пристрою не потрібно мати спеціального обладнання. Використовується зручна спеціальна макетна дошка, на ній передбачені додаткові перемички та з'єднувальні дроти, які не треба припаювати.

Згідно рис.1.1 можна виділити кілька основних елементів:

- штекер для підключення, яким забезпечується зовнішнє джерело живлення. Він використовується нечасто, але у випадку підключення разом з USB-портом, живлення залишається лише зовнішнє, автоматично вимикаючи USB порт;
- USB-порт для приєднання робочої плати до персонального комп'ютера. Через цей порт можна завантажувати попередньо спроектовані програми. Таке під'єднання забезпечує подачу вхідного живлення через USB-порт;
- світлодіод-індикатор вбудований в схему Arduino для подачі сигналу про початок її роботи;
- всі входи і виходи пронумеровані, згідно нумерації, під'єднують різні зовнішні керовані пристрої. (вбудований індикатор-світлодіод підключений під №13).
- RESET-кнопка забезпечує повернення до налаштувань, передбачених виробником;

Для збірки необхідного пристрою не знадобиться спеціальне обладнання. Для цього використовується спеціальна макетна дошка з додатковими перемичками та дротами, що навіть не потребують пайки.

Специфіка виводів Arduino

На платформі розташовано входи-виводи, але не всі вони одного типу – серед контактів є цифрові та аналогові. На аналогових входах діапазон значень від 1 до 0 має додаткові ділення.

У цифрових виводах є тільки два значення, одне з яких — логічний нуль, друге – одиниця. При застосуванні це буде виглядати, як максимальна яскравість світлодіода при підключенні до цифрового контакту з логічною одиницею. А ось при подачі нуля – він світитися не буде, або згасне.

Специфіка в тому, що з цифровим під'єднанням не буде ніяких середніх значень. У випадку підключення світлодіода через аналоговий вхід/вихід з'явиться можливість плавного регулювання його яскравості. У більшості випадків аналогові типи виводів/входів використовують для підключення аналогових датчиків.

Можливості використання

Існує необмежена кількість варіантів використання платформи. Її універсальність дозволяє проектувати системи різного ступеню складності. За допомогою платформи Arduino легко вирішити проблеми:

- організації процесів промислового виробництва;
- військового характеру;
- при проведенні розвідувальної діяльності та розшукових заходів;
- діагностичного та медичного характеру;
- БПЛА та авіації (розробка, управління та геолокація);
- охоронної системи та держбезпеки;

розробки та виробництва робототехніки;

- у побуті, облаштуванні, благоустрої та техобслуговуванні будівель, тощо.

РОЗДІЛ 2. ЗАСТОСУВАННЯ СИСТЕМИ ГЕОЛОКАЦІЇ

2.1 Поняття глобальної системи позиціонування

Що стосується самого поняття GPS, слід зазначити, що ця абревіатура розшифровується як The Global Positioning System [3] або ж українською Глобальна система позиціонування. Вона належить космічній радіосистемі місцезнаходження NAVSTAR, що використовує навігаційні супутники і належить уряду США, вона розроблена для їх військових потреб (USAF).

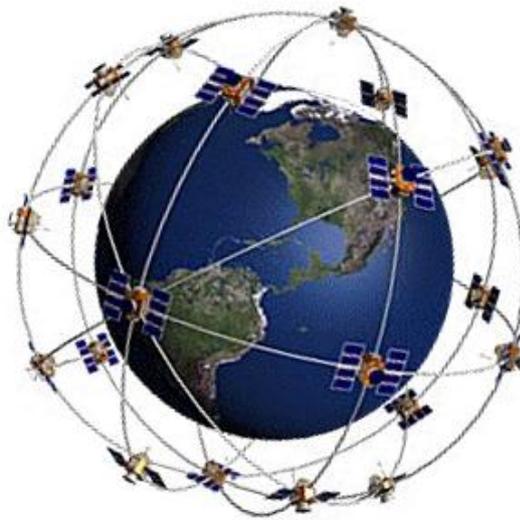


Рис. 2.1. Орбіти супутників NAVSTAR

Ця система дозволяє точно визначити тривимірне положення з точністю до метра та час з точністю до 10 наносекунд у всьому світі та цілодобово для спеціального військового обладнання. Двадцять один супутник GPS і три запасні супутники знаходяться на орбіті на висоті 10 600 миль над Землею. Супутники розташовані на шести орбітах під кутом 60 градусів. Кожна орбіта містить по чотири супутники, які забезпечують доступність до 4-12 супутників з будь-якої точки земної поверхні. Американська система NAVSTAR почалася з пуску першого супутника у лютому 1978 року.

На кожній орбіті розташовано по чотири супутники з висотою орбіт, приблизно 20200 км, і періодом обертв навколо Землі близько 12 годин.

Система GPS (Глобальна Система Позиціонування) вбудована в технологію радіонавігації, яка надає постійні дані про розташування, рух та час для всіх цивільних користувачів у всьому світі, дозволяючи вільний доступ до цієї інформації.

Кожен супутник містить комп'ютер, атомний годинник і радіо. Розуміючи свою орбіту та годинник, він постійно транслює зміну свого положення та час його фіксації. Раз на день кожен супутник перевіряє своє «відчуття» часу та положення за допомогою наземної станції та вносить незначні виправлення за потреби.

На землі будь-який GPS-приймач містить комп'ютер, який «тріангулює» його положення, отримуючи сигнали від трьох супутників. Результатом є географічне положення – довгота та широта – для більшості приймачів у межах 100 метрів.

Завдяки приймачу GPS, будь-хто може визначати своє місцезнаходження та поточний час.

GPS використовує тривимірну всесвітню геодезичну систему координат WGS 84 (Світова Геодезична Система 1984), за допомогою якої здійснює визначення місцезнаходження об'єкту на поверхні Землі.

Контроль за WGS84 здійснюється US National Geospatial-Intelligence Agency (Національне Геопросторове Агентство США). Ця система спочатку була розроблена для аеронавігації, але стала стандартом у всесвітній геодезії після того, як Міжнародна організація цивільної авіації (ICAO) ухвалила її у березні 1989 року. Після прийняття ІМО (Міжнародна морська організація) цей стандарт також став застосовуватися в морській транспортній галузі.

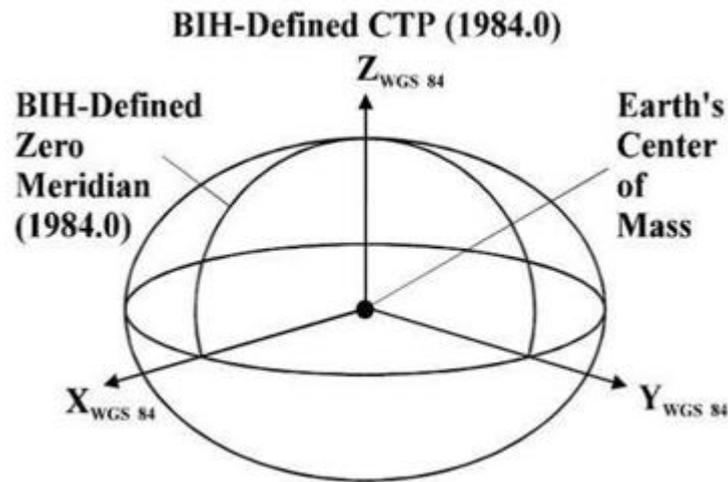


Рис. 2.2. Еліпсоїд WGS84

Еліпсоїд WGS84, що є базою для цієї системи, має центр мас землі і найкраще узгоджується з геоїдом. Такий тип еліпсоїда, який найкраще підходить для конкретної області геоїда, називається референц-еліпсоїдом (від лат. referens – допоміжний). Його характеристики, такі як велика та мала піввісь, а також стиснення, визначаються у таблиці нижче.

Стиснення – це відносна різниця довжин двох осей, яка виражена у формулі $f = (a-b)/a$. Стиснення є дуже малою величиною, тому зазвичай замість нього використовується величина $1/f$.

Таблиця 2.1.

Основні параметри референц-еліпсоїду WGS84

Параметр	Позначення	WGS84
Велика напівось	a	6378137 m
Геоцентрична гравітаційна стала	GM	398600.5 km ³ s ⁻²
Кутова швидкість	ω	7.292115 x 10 ⁻⁵ rad s ⁻¹
Сплюсненість	f	1/298.257223563
Норм. коефіцієнт другої зональної гармоніки гравітаційного потенціалу	\check{C}_{20}	-484,16685 x 10 ⁻⁶

2.2 Принцип роботи супутникових систем навігації

Завданням супутникової системи є вимірювання відстані від антени, розташованої на об'єкті, до супутників, чиє географічне розташування заздалегідь відоме. Таблиця, що містить інформацію про місцезнаходження супутників, називається альманахом, і вона вбудована в будь-який супутниковий приймач ще до початку вимірювань.

Зазвичай альманах зберігається у пам'яті приймача з моменту останнього вимкнення і використовується негайно, якщо дані актуальні. Кожен супутник передає весь альманах у своєму сигналі, що дозволяє приймачу обчислити положення об'єкта в просторі за допомогою елементарних геометричних розрахунків. У методі виміру відстані від супутника до антени приймача лежить визначення швидкості поширення радіохвиль. Кожен супутник навігаційної системи випромінює сигнали точного часу для можливості вимірювання часу поширення радіосигналу. [3,4]

Годинник супутникового приймача під час своєї роботи синхронізується із системним часом, аби потім обчислити затримку між зафіксованим часом випромінювання та часом прийому сигналу. Навігаційний приймач може обчислити координати антени, або визначити швидкість за допомогою ефекту Доплера, обробляючи інформацію, накопичену протягом певного періоду часу. Однак реальна робота системи є набагато складнішою і потребує вирішення різних проблем. Наприклад, неоднорідність гравітаційного поля Землі впливає на супутникові орбіти. Для усунення відсутності атомних годинників у навігаційного приймача потрібно отримати інформацію не менше, ніж від 3-4 супутників. Неоднорідність атмосфери може вплинути на швидкість та напрямок поширення радіохвиль. Також аби уможливити отримання сигналів від сторонніх наземних об'єктів, особливо в міських умовах, слід вжити додаткових заходів.

2.3 Варіанти GPS-модулей

Для розробки проектів Arduino існує достатньо розширена лінійка GPS-модулів від виробника. Кожен з них має певні переваги та специфічні моливості, тому, обираючи свій варіант треба врахувати всі необхідні умови та параметри.

Модуль GPS надає, в першу чергу, можливість визначення точного місця розташування з конкретними географічними даними, які включають координати, висоту над рівнем моря, тощо. Крім того, можливо одразу отримати:

- показники швидкості цілі;
- напрямок руху досліджуємого об'єкту;
- траєкторію переміщення;
- точний час розташування у точці;
- відповідну дату.

На практиці, зручно та швидко, а головне, з мінімальними затратами, вдасться отримати дані поточного позиціонування БПЛА, роботизованої техніки, метеорологічної апаратури та датчиків.

Приклади можливих варіантів GPS-модулів:

а) Майже найдешевший представник модулів – SKM53 GPS. Має низьке споживання струму з часом холодного запуску близько 36 секунд, гарячого – 1 сек. Для позиціонування використовує 66 каналів, стеження – 22 канали. Вбудована GPS-антена. Висока продуктивність навігації навіть за недостатніх умов видимості.

б) Модуль EM-411 працює на базі високопродуктивного чіпу з низьким рівнем споживання енергії SiRF StarIII. Великий обсяг пам'яті дозволяє зберігати дані альманаху (підтримує стандартний протокол NMEA0183). Час холодного старту – 45 секунд.

в) VK2828U7G5LF на базі чіпа Ublox UBX-G7020-KT. Модуль дозволяє отримувати координати GPS. У приймач вбудована пам'ять, тому можна зберігати потрібні налаштування. Він оснащений вбудованою керамічною

антенною, використовує комунікаційний протокол NMEA 0183. Напряга живлення 3,3-5В.

г) В даному проекті ARDTRACKER-GPS було використано зручний та компактний GPS приймач моделі Neo-6М. Він вироблений надійним швейцарським брендом – u-blox. Тепер розглянемо детальніше цей елемент нашого проекту.

2.4 Загальний огляд GPS-модулю NEO-6М

Цей продукт виробляється відомою компанією із Швейцарії u-blox. У модулі використані новітні технології, які допомагають визначити та доправити за запитом точну інформацію про місцезнаходження та положення у просторі об'єкта. Використовується для живлення модулю напруга від 3 до 5В.



Рис. 2.3. Модуль NEO-6М зовні

В лінійці пристроїв представлені 6 типів: М, G, Q та Р, V і Т, кожен з яких має свої унікальні та специфічні характеристики. Закладений час холодного старту коливається в районі 27-ми секунд, а швидкість його – 9600 біт/с, за замовчуванням.

Що стосується нашого модуля NEO-6М, то його відрізняє від інших типів моделей те, що він здатен виконувати до 5-ти оновлень поточного розташування об'єкту за одну секунду. Заявлена виробником точність складає $\pm 2,5$ м у горизонтальній системі координат [5]. Також, близько секунди часу, знадобиться U-blox6-механізму позиціонування для його першої TTFF-фіксації.

У чіпа NEO-6M є ще один плюс, який вигідно відрізняє його серед інших — практично найкращий PSM режим енергозбереження. Як це працює: значна економія електроспоживання отримується за рахунок почергового ввімкнення потрібних чи вибіркового вимкнення незадіяних частин працюючого приладу.

Отримана таким чином економія енергоспоживання модуля (в межах до 11 мА), робить його незамінним при застосуванні у приладах, з підвищеною чутливістю до необхідних витрат енергії. Це стосується, наприклад, переносних або наручних GPS-приладів.

Основою модуля є компактний за розміром GPS-чіп NEO-6M з кроком контактів 1/10 міліметра. З його основними технічними характеристиками можна ознайомитися у табл. 1.2. У модулі закладена здатність відстежування одразу 22 супутників, використовуючи 50 канали. При цьому, забезпечується достатньо високий рівень чутливості – до 161 дБ. А от споживання струму складає тільки 45 мА. Необхідно зауважити, через те, що робочою напругою NEO-6M є діапазон від 2.7В до 3.6В, на чіпі NEO-6M передбачений стабілізатор MIC 5205 з вихідною напругою 3.3В.

Таблиця 2.2.

Перелік основних технічних характеристик модуля NEO

Тип приймача	GPS L1 1575,42 МГц 50 каналів
Частота оновлення навігації	1 Гц (5 Гц максимум)
Межа точності горизонтального положення об'єкта	2,5 м
Час захоплення цілі	Холодний старт: 27 с Гарячий старт: 1 с
Чутливість навігації	-161 дБм

Прод. табл 2.2.

Протокол зв'язку	NMEA, двійковий UBX, RTCM
Послідовна швидкість передачі даних	від 4800 до 230400 (за замовчуванням 9600)
Робоча температура	-40°C ~ 85°C
Робоча напруга	2,7 В ~ 3,6 В
Робочий струм	45 мА
Опір TXD/RXD	510 Ом

Особливості відображення фіксації положення об'єкту світлодіодним індикатором:

Вбудований світлодіод чіпа GPS NEO-6M відображає дані про стан визначення поточної позиції. Розташування світлового індикатора на платі видно з рис.1.5.

Частота та тип мерехтіння підказує стан, в якому перебуває прилад або ціль дослідження на даному проміжку часу:

- абсолютно відсутнє мерехтіння вказує на пошук доступних супутників;
- періодичне мерехтіння з проміжком у 1 сек показує стабільно-фіксоване положення. Означає, що захоплено достатньо супутників для аналізу та роботи.
- повільне мерехтіння надає сигнал що дані вже отримані та передаються до порту.



Рис. 2.4. Вигляд світлодіодного індикатора чіпа NEO-6M

Особливості використання LDO-стабілізатора 3.3 В в NEO- 6М:

Для роботи мікрочіпу NEO6M необхідна напруга в межах від 2,7В до 3,6В, для підтримання цих умов, на модулі встановлений стабілізатор напруги MICREL MIC5205 (Ultra Low Dropout -3V3) з вихідним показником напруги 3.3В (див. рис. 2.5.).

Логічні контакти регулятора напруги стійкі до 5В, тому не виникає проблеми під'єднання до плати Arduino модуля NEO6M або будь-якого іншого логічного мікроконтролера < 5В. Треба зауважити, що перетворювач логічного рівня не знадобиться.



Рис. 2.5. Регулятор(стабілізатор) напруги модуля NEO- 6М

Мікросхема EEPROM та батарея (акумуляторна):

На модуль NEO-6М встановлена пам'ять EEPROM (НК24С32) в об'ємі 4 КБ, яка під'єднується до чіпу, використовуючи інтерфейс I2C (для швидкого та

надійного зв'язку в межах пристрою). Мікросхема необхідна для збереження необхідних налаштувань після вимкнення живлення.

Акумуляторна батарея, яка підзаряджається, використовується для підтримки резервного збереження інформації. Вона працює по принципу суперконденсатора, як видно з рис. 2.7.

Пам'ять EEPROM (HK24C32) двопровідна та послідовна підключена через порт D4 до мікросхеми самого модуля.

Спільно EEPROM і акумуляторна батарея допомагають BBR бат. RAM із збереженням:

- даних годинника;
- конфігурації самого модуля;
- останніх відміток положення (даних щодо орбіт GNSS).

Заряджається акумулятор в автоматичному режимі, який вмикається одразу з подачею живлення на модуль. Максимальний час для збереження даних без підключення та живлення обмежений до двох тижнів.

Завдяки тому, що при наступному включенні збережені дані годинника та останньої зафіксованої позиції, час до першого TTFF-визначення суттєво скорочується і складає близько 1 с. Це надає значну перевагу в часі при фіксуванні фактичного положення на даний момент.

Відсутність акумуляторної батареї кожен раз дає запуск GPS тільки з холодного стану з значно більшим періодом пуску. До того, початкове блокування GPS також буде займати більше часу.



Рис. 2.6. Пам'ять EEPROM та батарея чіпу NEO-6M

Активна антена (зовнішня) для NEO-6M:

З модулем в комплекті йде окрема зовнішня антена 25 мм х 25 мм. Її чутливість в межах

161 dBm, що достатньо для прийому радіосигналу від знайдених супутників. Кріплення здійснюється через роз'єм U.FL, який розташований на чіпі NEO-6M.

Схема під'єднання зовнішньої антени до модуля надана нижче на малюнку:

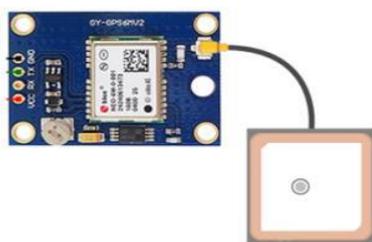


Рис. 2.7. Схема приєднання зовнішньої антени

Таблиця 2.3.

Приєднання NEO до UNO

Контакт NEO-6M	Контакт Arduino UNO
VCC (Power Supply Pin або контакт живлення)	5V
GND (Ground Pin або контакт заземлення)	GND
TX (Transmitter Pin або контакт передавача)	RX, будемо використовувати контакт D10
RX (Receiver Pin або контакт приймача)	TX, будемо використовувати контакт D9

На модулі NEO-6М є чотири контакти: RX, GND, VCC, та TX, призначення та розшифрування яких наведені у табл.1.3. Модуль NEO-6М під'єднано послідовно через контакти TX та RX до основної плати ArduinoUNO. Підключення вкрай просте, а його схема відображена на малюнку:

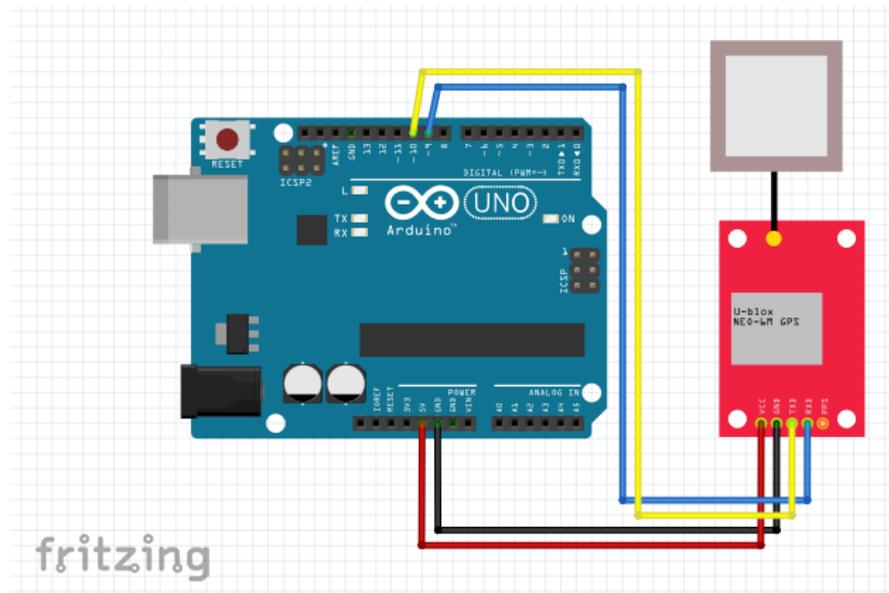


Рис. 2.8. Під'єднання NEO-6М до ArduinoUNO.

РОЗДІЛ 3. ЗАСТОСУВАННЯ СИСТЕМИ РАДІОЗВ'ЯЗКУ

3.1 Глобальна система мобільного зв'язку

Глобальна система мобільного зв'язку (GSM) є цифровою мобільною мережею, що широко використовується у Європі та інших регіонах світу. Вона використовує технологію множинного доступу з тимчасовим поділом (TDMA) і є найпоширенішою серед трьох цифрових бездротових телефонних технологій: TDMA, GSM і множинного доступу з кодовим поділом (CDMA) [6].

GSM цифрує та стискає дані, передаючи їх через канал із двома іншими потоками даних користувачів, кожен у власному часовому інтервалі. Мережа працює на частотах 900 МГц або 1800 МГц. Разом із іншими технологіями, GSM є частиною еволюції бездротових мобільних телекомунікацій, що включає в себе високошвидкісну передачу даних з комутацією каналів (HSCSD), розширене середовище GSM для передачі даних (EDGE), загальну пакетну радіопередачу (GPRS) і універсальну мобільну телекомунікаційну службу (UMTS).

Попередники GSM, зокрема Advanced Mobile Phone Service (AMPS) у США та Total Access Communication System (TACS) у Великобританії, були побудовані на базі аналогічної технології. Однак ці системи не змогли масштабуватися для залучення більшого числа користувачів. Недоліки цих систем підкреслили необхідність більш ефективної технології для мобільного зв'язку, яка б в майбутньому могла бути використана на міжнародному рівні [6,7].

Аби досягти цієї мети у 1983 році European Conference of Postal and Telecommunications Administrations (Європейська конференція поштових і телекомунікаційних адміністрацій) створила комітет для розробки європейського стандарту для цифрових телекомунікацій.

Також СЕРТ визначила критерії, яким повинна відповідати такого роду система:

- підтримка міжнародного роумінгу;

- підтримка портативних пристроїв;
- висока якість мовлення;
- підтримка нових послуг і можливості використання ISDN;
- низька вартість обслуговування.

Представники 13 європейських країн У 1987 році підписали договір про впровадження телекомунікаційного стандарту. Пізніше ЄС ухвалив закони, які вимагали впровадження GSM як стандарту на території Європи. Згодом відповідальність за увесь проект GSM було передано від CEPT до Європейського інституту стандартів телекомунікацій (ETSI).

Послуги мобільного зв'язку на базі стандарту GSM вперше були введені в експлуатацію в Фінляндії у 1991 році. У тому ж році стандартна смуга частот GSM була розширена з 900 МГц до 1800 МГц. У 2010 році технологія GSM охоплювала 80% світового ринку мобільного зв'язку. Проте декілька операторів зв'язку припинили експлуатацію своїх GSM-мереж, включаючи Telstra в Австралії.

Мережа GSM складається з чотирьох окремих частин, що працюють задля забезпечення її цілісності: сам мобільний пристрій, BSS (підсистема базової станції), NSS (підсистема комутації мережі) і OSS (підсистема експлуатації та підтримки).

Мобільний пристрій з'єднується з мережею за допомогою апаратного забезпечення. SIM (модуль ідентифікації абонента) надає мережі інформацію для ідентифікації мобільного користувача.

Організація мережі GSM

Підсистема базової станції (BSS) відповідає за обробку трафіку між мобільним телефоном і підсистемою комутації мережі (NSS). Вона складається з двох основних компонентів: базової приймально-передавальної станції (BTS) і контролера базової станції (BSC). BTS містить обладнання, яке взаємодіє з мобільними телефонами, включаючи приймально-передавальні пристрої та

антени, тоді як BSC є центральним управлінським пунктом [7]. BSC зв'язується з групою базових приймально-передавальних станцій і керує їх роботою.

Одна з частин архітектури мережі GSM, часто відома як базова мережа, відповідає за відстеження місцезнаходження абонентів, щоб забезпечити надання послуг стільникового зв'язку. Власники мобільних операторів контролюють цю частину, відому як NSS. NSS складається з різних компонентів, включаючи центр комутації мобільного зв'язку (MSC) і реєстратор місцезнаходження будинку (HLR). Ці компоненти виконують різноманітні завдання, включаючи маршрутизацію дзвінків і послугу коротких повідомлень (SMS), а також проведення автентифікації та зберігання інформації про обліковий запис абонента за допомогою SIM-карт

Деталі безпеки

Незважаючи на те, що технологія була розроблена як безпечна бездротова система, вона все ще може зазнавати атак. Тому GSM використовує певні заходи автентифікації, наприклад автентифікація «виклик-відповідь», яка пропонує користувачеві надати відповідь на запитання, а також попередньо заданий ключ у формі паролю або фрази.

Є кілька криптографічних алгоритмів безпеки, які використовує GSM, такі як потокові шифри, які шифрують цифри відкритого тексту. A5/1, A5/2 і A5/3 — це три потокові шифри, які забезпечують конфіденційність розмови користувача. Однак алгоритми як для A5/1, так і для A5/2 були зламані та опубліковані, тому вони чутливі до атак.

GSM використовує GPRS, послугу пакетного зв'язку, для передачі даних, наприклад під час перегляду веб-сторінок. Однак шифри, які використовує GPRS, GEA1 і GEA2, були розсекречені та опубліковані у 2011 році. Дослідники опублікували програмне забезпечення з відкритим кодом для перехоплення пакетів у мережі GPRS.

Обмеження GSM

Хоча GSM технологія є найкращим вибором для сучасних телекомунікаційних систем, вона не позбавлена недоліків. Нижче розглянуто деякі недоліки GSM.

1. Електронні перешкоди. Оскільки GSM використовує технологію передачі імпульсів, відомо, що він створює перешкоди для електроніки, наприклад слухових апаратів. Через ці електромагнітні перешкоди в деяких місцях, як-от аеропорти, заправні станції та лікарні, потрібно вимкнути мобільні телефони.

2. Затримка пропускної здатності . Під час використання технологій GSM декілька користувачів отримують доступ до однієї смуги пропускання, що іноді призводить до значної затримки , коли до мережі приєднується більше користувачів.

3. Обмежена швидкість передачі даних . GSM пропонує дещо обмежену швидкість передачі даних . Щоб досягти вищої швидкості передачі даних, користувач повинен перейти на пристрій із більш просунутими формами GSM.

4. Повторювачі . Технології GSM вимагають від операторів встановлення ретрансляторів для збільшення покриття.

3.2 Загальний сервіс пакетної радіопередачі

Загальна пакетна радіопередача (GPRS) — по суті, це технологія комутації пакетів, яка дозволяє передавати інформацію за допомогою мобільних мереж. GPRS підтримується мобільними телефонами, а також ноутбуками та кишеньковими пристроями, оснащеними модемами GPRS[8].

Цю технологію можна використовувати для полегшення з'єднань, пов'язаних з Інтернет-протоколами, які, для прикладу, можуть забезпечити функціонал комерційних та корпоративних програм. Перед передачею інформація розбивається на окремі пакети та направляється через базову мережу та радіо. На стороні одержувача дані повторно об'єднуються.

Глобальна система мобільного зв'язку (GSM) є основним стандартом для стільникової мережі другого покоління (2G), тоді як GPRS є вдосконаленою його версією. GPRS не схожий на службу коротких повідомлень GSM (GSM-SMS), яка має обмеження на довжину повідомлення у 160 байтів. GPRS з теоретичної точки зору має максимальну швидкість 115 Кбіт/с, хоча більшість мереж працюють на швидкості приблизно 35 Кбіт/с. GPRS іноді неофіційно називали 2,5G.

GPRS може працювати як із симетричною, так і з асиметричною конфігурацією, тоді як частота для будь-якого напрямку визначається тим, який із 12 класів провайдерів із кількома слотами вибрано. Визначення кількості часових інтервалів для кожного шляху здійснюється багатослотним класом обслуговування. Для кожного часового інтервалу забезпечується теоретична швидкість з'єднання 21,4 Кбіт/с. Одним із найпростіших є клас перший обслуговування, який передбачає одноразовий слот для кожного шляху. Клас обслуговування під номером 12, безумовно, найефективніший, із чотирма часовими слотами в кожному напрямку[8,9].

GSM-IP означає Інтернет-протокол глобальної системи мобільного зв'язку, а також альтернативну назву для GPRS. Це гарантує, що клієнти підключені, здійснюють аудіодзвінки та переглядають веб-сторінки. Цей метод забезпечує пакетний радіодоступ клієнтам множинного доступу з рівним часовим поділом (TDMA).

Раніше GPRS був найшвидшим варіантом, доступним у мережі, але сьогодні він поступається з точки зору швидкості та надійності, оскільки мережі 3G і 4G перевершують його. Однак він все ще використовується в деяких областях, зокрема в сільських регіонах і країнах, які не наважилися на більш сучасні технології. Коли мережа GPRS доступна, більшість смартфонів можуть використовувати її, хоча ті, хто звик до кращого з'єднання, помітять значно меншу пропускну здатність і тривале очікування під'єднання.

Як працює GPRS

GPRS додає кілька нових важливих функцій, щоб допомогти наскрізній передачі пакетів даних на основі IP. Стандарти GSM створили GPRS, що призвело до створення структури з визначеними атрибутами, інтерфейсами та міжмережевими операціями для допомоги в роумінгу.

GPRS удосконалює архітектуру стандарту GSM, щоб реалізувати пакетні послуги зв'язку зі швидкістю до 114 кбіт/с. Передача даних використовує мережу щоразу, коли це необхідно за допомогою режиму пакетної передачі. Завдяки стандарту GPRS користувачі могли отримувати оплату на основі обсягу фактичного обміну, а не тривалості з'єднання, що дозволяло їм залишатися на зв'язку без додаткових витрат[9].

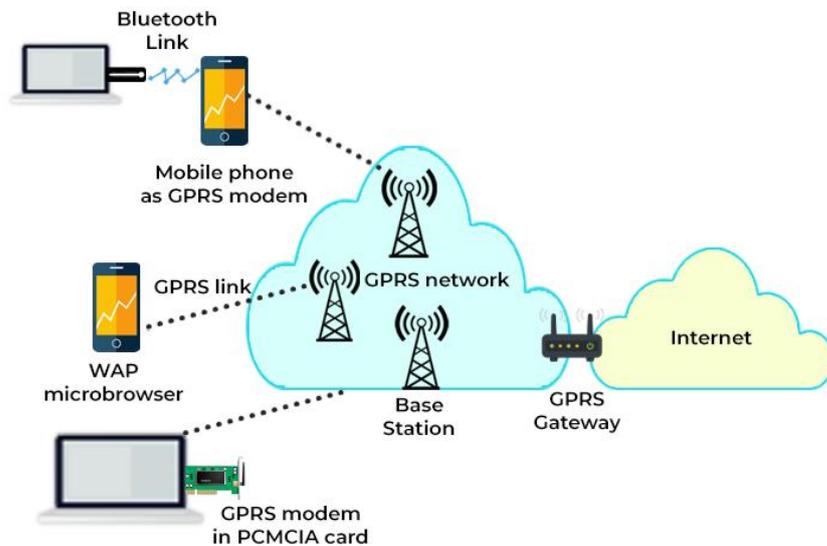


Рис. 3.1. Схема роботи системи GPRS

GPRS дозволяє операторам мереж мати більш ефективний метод розповсюдження даних своїм клієнтам, усуваючи вимогу використовувати мовні канали для передачі даних. Клієнти отримують у чотири рази більшу ставку для найбільш типових комутованих з'єднань при використанні GPRS-з'єднання.

Загальна середня швидкість служб передачі даних становить 9,6 Кбіт/с, тоді як швидкість GPRS коливається від 40 до 172,2 Кбіт/с. Користувачі також можуть отримати доступ до мультимедійних програм та інформації за

допомогою з'єднання GPRS. Щоб переглядати або використовувати дані немає необхідності у їх завантаженні.

Крім того, через обмеження швидкості передачі даних з комутацією каналів і обмеження служби коротких повідомлень (SMS) у 160 символів багато програм раніше були недоступні в мережах GSM. Система GPRS дозволяє мобільним службам використовувати програми, які раніше були у доступі лише на настільних ноутбуках і комп'ютерах. Це розкриває низку нових технологічних і творчих можливостей.

З'єднання GPRS можна розглядати як підмережу Інтернету, оскільки воно використовує подібні протоколи, а телефони GPRS можна розглядати як мобільні хости, оскільки обидва використовують однакові методи. GPRS працює на стільникових телефонах, ноутбуках та інших портативних пристроях із GPRS-модемом або SIM-карткою GPRS Subscriber Identity Module. Модем GPRS ідеально підходить для надсилання та отримання повідомлень з комп'ютера.

Ідея якості обслуговування (QoS) також включена в GPRS. Це стосується здатності адаптувати послугу до вимог програми. Застосовуються наступні умови QoS – пріоритет, надійність, затримка та пропускна здатність. Для інтеграції GPRS в архітектуру GSM необхідне включення нових вузлів мережі, які називаються GSN (вузли підтримки GPRS), розташовані в магістральній мережі:

SGSN (Serving GPRS Support Node) — це маршрутизатор, який контролює місце розташування найближчих станцій і пропонує інтерфейс передачі пакетів до порталу GGSN. Він відповідає за функції аутентифікації мобільних пристроїв GPRS, мережевої реєстрації, управління мобільності та збору даних про тарифікацію використання радіоінтерфейсу.

Шлюз, який з'єднується з іншими мережами передачі даних, — це GGSN (вузол підтримки шлюзу GPRS). GGSN відповідає за доставку IP-адреси до мобільних терміналів протягом усього часу з'єднання. Він підключається до зовнішніх мереж як в ролі маршрутизатору, так і в ролі інтерфейсу. Він містить інформацію про стільникову маршрутизацію GPRS, яка використовується для

маршрутизації пакетів через внутрішню магістраль на основі IP до відповідного вузла підтримки GPRS. GGSN також може працювати фільтром пакетів для вхідного трафіку та записувати дані про ціни, пов'язані із зовнішніми інтернет-сервісами.

Бездротові пакетні дані GPRS мають три основні характеристики:

1. Функція постійного підключення, що задовільняє потребу в телефонному з'єднанні, надаючи доступ до програм одним кліком миші.
2. Оновлені звичайні системи означають, що перевізникам не потрібно оновлювати своє обладнання; натомість GPRS ставиться поперх поточного обладнання.
3. Важлива частина майбутніх мереж, тобто базова мережа пакетної передачі даних для систем EDGE і WCDMA.

Переваги та проблеми GPRS:

Порівняно з оригінальною системою GSM, технологія GPRS надає різноманітні переваги користувачам і операторам мережі. Компанії зазвичай використовували це для створення точних можливостей даних за допомогою технології мобільного зв'язку. Давайте розглянемо деякі переваги GPRS.

Переваги GPRS:

1) Спрощення виставлення рахунків

У порівнянні з послугами з комутацією каналів передача пакетів GPRS пропонує кращу вартість споживачам. Коли йдеться про послуги з комутацією каналів, величина рахунку визначається довжиною з'єднання. Одним із істотних недоліків комутації каналів для IoT є економічна ефективність. Довжина з'єднання визначає експлуатаційні витрати абонента в комутаторі.

2) Підвищена швидкість

Значною перевагою системи GPRS є те, що вона забезпечує більш високу швидкість передачі даних, ніж GSM.

3) «Завжди ввімкнене» підключення

Ще одна перевага GPRS полягає в тому, що він завжди доступний. З'єднання GPRS забезпечує постійне підключення до Інтернету, швидкий обмін SMS. Це також була найшвидша мережа, яку коли-небудь бачили, коли її вперше випустили.

Попри переваги технології GPRS, деяких клієнтів можуть збентежити деякі недоліки:

- **Обмеженість**

GPRS технічно підходить для більшості випадків використання IoT, але не для всіх. 2G або 3G з GPRS зазвичай є теоретично прийнятним, якщо все, що вам потрібно, це повільна періодична передача даних. Однак для додатків, що інтенсивно передають дані, GPRS впливає на продуктивність системи.

- **Ризик зниження швидкості**

Щоб досягти максимальної швидкості передачі даних GPRS 172,2 кбіт/с, усі вісім часових інтервалів повинні використовуватися окремим користувачем без помилок. Підтримувані швидкості передачі даних (171,2/384 Кбіт/с) нижчі за поточні бездротові протоколи, такі як HSPA, LTE, LTE-advanced тощо [9].

Оператор мережі навряд чи дозволить одному клієнту GPRS зайняти всі часові інтервали. Крім того, передбачається, що перші GPRS-термінали будуть дуже обмеженими, з підтримкою лише одного, двох або трьох часових інтервалів.

3.3 «Спілкування» за допомогою AT-команд

Будь-який розробник, який зіштовхнеться із будь-яким стільниковим модемом, колись повинен буде використовувати так звані «AT-команди». Команди AT, спочатку розроблені виробником модемів Hayes як засоби для роботи з продуктами стаціонарного телефонного зв'язку, тепер використовуються всіма модемами всіх типів. В нашому випадку, AT-команди це основний метод спілкування із модулем [10].

АТ-команди в основному використовуються для налаштування модему та встановлення його мережевого підключення. Їх можна використовувати для отримання інформації про статус модуля та з'єднання, і це може бути дуже корисним для налагодження програм і підтвердження того, що модем працює належним чином: він підключився до потрібної мережі, використовує правильну стільникову технологію, увімкнено роумінг тощо.

Існує чотири типи АТ команд, таких як:

1. Тестові команди – використовуються для перевірки, чи підтримується команда МОДЕМОМ.

Синтаксис: АТ<command name>=?

Приклад: АТD=?

2. Команда читання – використовується для отримання налаштувань мобільного телефону або МОДЕМУ для операції.

Синтаксис: АТ<назва команди>?

Приклад: АТ+СВС?

3. Встановити команди – використовується для зміни налаштувань мобільного телефону або МОДЕМУ для операції.

Синтаксис: АТ<назва команди>=value1, value2, ..., valueN

Приклад: АТ+СSCA="380990000000", 120

4. Команди виконання – використовуються для виконання операції.

Синтаксис: АТ<назва команди>=parameter1, parameter2, ..., parameterN

Команди читання недоступні для отримання значення останнього параметра, призначеного в командах виконання, оскільки параметри команд виконання не зберігаються.

Приклад: АТ+СMSS=1,"380990000000", 120

Пояснення типових АТ-команд:

- 1) АТ – ця команда використовується для перевірки зв'язку між модулем і комп'ютером.

Приклад: - АТ - ОК

Команда повертає код результату ОК, якщо комп'ютер (послідовний порт) і модуль підключено належним чином. Якщо будь-який із модулів чи SIM-карт не працює, він поверне ERROR.

2) +CMGF – ця команда використовується для встановлення режиму SMS.

Текстовий режим або режим PDU (протокольний блок даних) можна вибрати, призначивши 1 або 0 у команді.

Синтаксис: AT+CMGF=<режим>

- 0: для режиму PDU;
- 1: для текстового режиму;

Текстовий режим SMS легше використовувати, але він надає обмежені функції SMS. PDU (протокольний блок даних) надає більший доступ до послуг SMS, але оператор вимагає знання TPDU на рівні бітів. Доступ до заголовків і тексту SMS доступний у шістнадцятковому форматі в режимі PDU, що дозволяє використовувати більше функцій.

Приклад: -AT+CMGF=1 -OK

3) +CMGW – Ця команда використовується для збереження повідомлення на SIM-карті.

Синтаксис: AT+CMGW="Номер телефону" > Повідомлення для збереження - Ctrl+z

Після введення AT+CMGW і номера телефону в наступному рядку з'являється знак «>», де можна ввести повідомлення. У цьому випадку можна вводити багаторядкові повідомлення. Ось чому повідомлення завершується комбінацією «Ctrl+z». Після натискання Ctrl+z на екрані відображається наступна інформація.

+CMGW: номер, на якому було збережено повідомлення

4) +CMGS – ця команда використовується для надсилання SMS-повідомлення на номер телефону.

Синтаксис: AT+CMGS= серійний номер повідомлення, яке потрібно надіслати.

Після введення команди AT+CMGS і серійного номера повідомлення на певну SIM-картку надсилається SMS.

Приклад: -AT+CMGS=1 -OK

5) ATD – Ця команда використовується для набору або виклику номера.

Синтаксис: ATD<номер телефону>;(Enter)

Приклад: -ATD123456789;

6) ATA – ця команда використовується для відповіді на виклик. Вхідний дзвінок позначається «RING», яке повторюється під час кожного гудка дзвінка. Після завершення виклику на екрані відображається «NO CARRIER».

Синтаксис: ATA(Enter)

Після натискання ATA, а потім клавіші Enter, на вхідний дзвінок буде прийнято відповідь.

Приклад: -RING -RING -ATA

7) ATH – ця команда використовується для відключення віддаленого зв'язку користувача з модулем GSM.

Синтаксис: ATH (Enter)

3.4 Огляд GSM/GPRS модуля SIM900A

Модуль GSM/GPRS SIM900A являє собою дводіпазонний GSM/GPRS-модем, який працює на частотах EGSM 900 МГц і DCS 1800 МГц, його можна застосовувати у різноманітних проєктах Інтернету речей. Фізичним інтерфейсом до мобільного чіпу є 68-контактна SMT-панель, яка забезпечує всі апаратні інтерфейси між модулем і платами клієнтів [11].

- Клавіатура та інтерфейс SPI-дисплея надають гнучкість при розробці індивідуальних проєктів.
- Serial порт і Debug порт допоможуть легко розробляти свої програми.
- Один аудіоканал включає мікрофонний вхід і вихід на динамік.

- Програмовані універсальні входи та виходи.

SIM900A розроблений з використанням технології енергозбереження, так що споживання струму складає всього 1,5 мА в режимі енергозбереження.

У основі модуля знаходиться стільниковий чіп 900А GSM від компанії Simcom. Робоча напруга цього чіпа варіюється від 3,4 В до 4,4 В, що робить його ідеальним для живлення безпосередньо від LiPo акумулятора. Завдяки цьому він є чудовим вибором для вбудовування в більш компактні проекти.

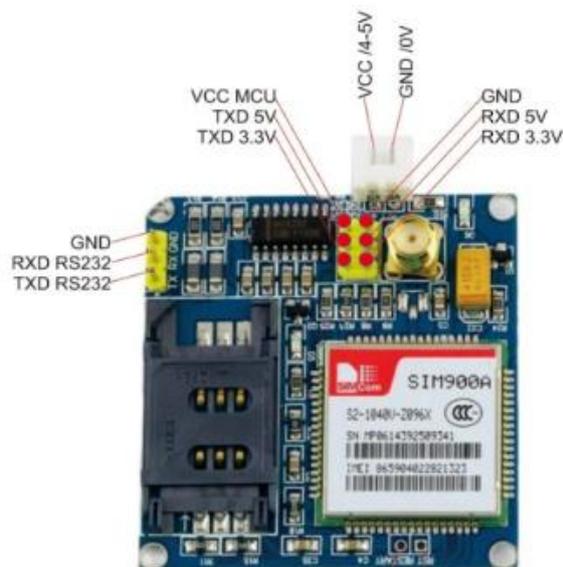


Рис. 3.1. Зовнішній вигляд та контакти SIM900A

Усі необхідні контакти SIM900A GSM розбиті на 13 роз'ємів, до яких також входять контакти, необхідні для зв'язку з мікроконтролером через UART. Розглянемо деталі стосовно розпіновки модуля, яку можна побачити на Рис.2.1.

- VCC - це контакт джерела живлення. Слід мати на увазі, що плата SIM900A має діапазон робочої напруги від 3,4 В до 4,4 В, тому підключення цього модуля до виходу 5 В Arduino, ймовірно, призведе до його пошкодження.
- RST (reset) – пін скидання. Слід на 100 мілісекунд подати сигнал LOW на цей пін аби виконати хард-скид.
- RxD (Receiver) – пін використовується для надсилання команд до модуля. Він має автоматичну передачу даних, тому швидкість передачі

даних, з якою ви надсилаєте AT- команду після скидання, є використовуваною швидкістю передачі даних.

- TxD (Transmitter) – пін передає дані від модуля до мікроконтролера (в нашому випадку Arduino UNO).
- GND - є контактом заземлення.

Модуль підтримує швидкість передачі в діапазоні від 1200 біт/с до 115 200 біт/с і має автоматичне визначення швидкості передачі даних.

Зв'язок з модулем:

Спілкування з цим модулем здійснюється через інтерфейс UART або RS232. Дані надсилаються до модуля або приймаються від модуля через інтерфейс UART.

Для живлення модуля зазвичай використовується стандартне джерело живлення +4,0 В. Для роботи модуля потрібна регульована напруга +4,5 В, будь-яка вища напруга може завдати шкоди модулю. Піковий струм від джерела живлення повинен становити 2А. Для зв'язку з контролером слід просто приєднати RXD модуля до TXD, а TXD - до RXD ARDUINO. Для отримання опорної напруги необхідно з'єднати заземлення контролера і модуля. Мікрофон підключається до AUDIO IN, а динамік або гарнітура - до AUDIO OUT. Звісно для налагодження стільникового зв'язку ми повинні вставити у модуль GSM функціональну SIM-карту. Після увімкнення модуля світлодіод NET LIGHT почне часто блимати, що вказує на успішне з'єднання.

Особливості роботи світлодіодних індикаторів:

Індикатор вказує на різні стани GSM-модуля, наприклад, увімкнення живлення, реєстрацію в мережі та GPRS-з'єднання. Коли модем увімкнено, світлодіод NETWORK буде блимати щосекунди. Після того, як модем зареєструється в мережі (це займає від 10 до 60 секунд), цей світлодіод буде блимати з кроком в 3 секунди. На цьому етапі вже можна почати використовувати функціонал модуля .



Рис. 3.2. Світлодіодні індикатори на SIM900A

Споживання енергії:

Однією з найважливіших частин стабільної роботи модуля SIM900A є забезпечення його достатнім живленням.

Модуль SIM900A, залежно від свого стану, може бути відносно енергоємним пристроєм. Максимальний струм, який споживає модуль, становить близько 2 А (особливо під час піку передачі). Зазвичай він не витрачає так багато, але може знадобитися близько 216 мА під час телефонних дзвінків або 80 мА під час передачі по мережі. Таблиця 3.1 узагальнює те, на що слід розраховувати:

Таблиця 3.1.

Режими споживання енергії SIM900A

Режим	Частота	Споживання
Вимкнення		60 μ A
Режим сну		1 мА
Режим очікування		18 мА
GPRS		453 мА

Прод.табл. 3.1

Пік передачі		2 А
Під час дзвінка	GSM850	199 мА
	EGSM900	216 мА
	DCS1800	146 мА
	PCS1900	131 мА

Детальніше про те, як відбувається приєднання модуля GSM/GPRS зв'язку SIM900А до Arduino Uno можна дізнатися з Таблиці 3.2.

Таблиця 3.2.

Приєднання SIM до UNO

Контакт SIM900А	Контакт Arduino UNO
VCC (Power Supply Pin або контакт живлення)	Живимо модуль не з UNO, а з акумулятора за допомогою DC-DC конвертера
GND (Ground Pin або контакт заземлення)	GND
TX (Transmitter Pin або контакт передавача)	RX, будемо використовувати контакт D8
RX (Receiver Pin або контакт приймача)	TX, будемо використовувати контакт D7

Схему того, як відбувається підключення SIM900A до живлення та плати контролера Arduino можна побачити на Рис. 3.3 нижче.

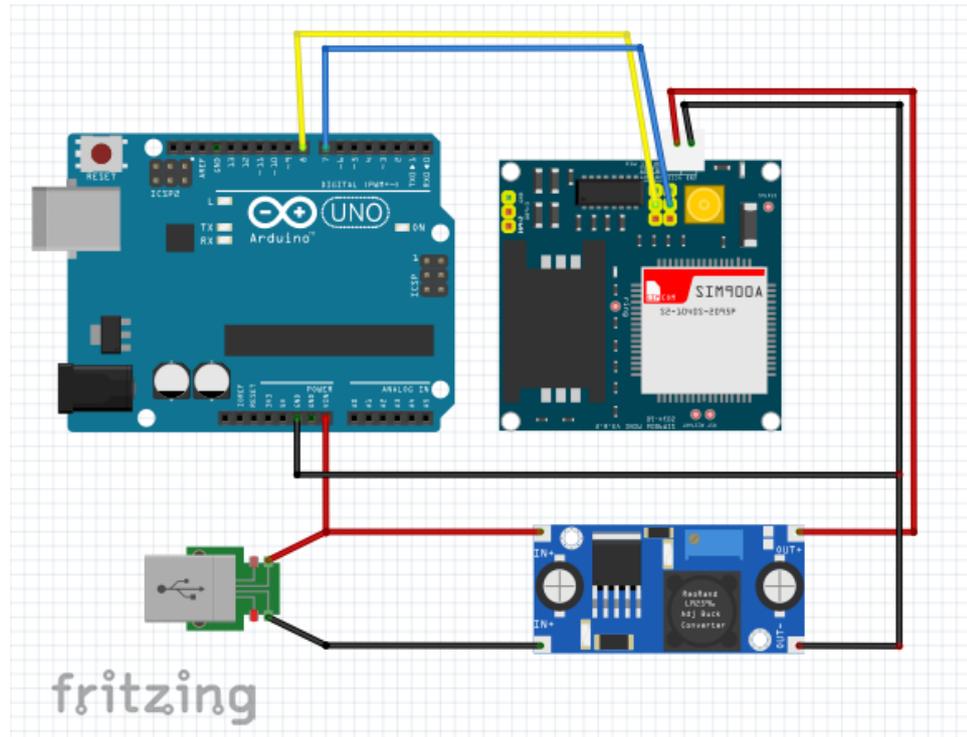


Рис.3.3. Схема підключення SIM900A та UNO

РОЗДІЛ 4. РОЗРОБКА ПРИСТРОЮ

4.1 Основні етапи проекту

Робота над проектом на базі Arduino, як і робота над будь-яким іншим проектом включає стандартні етапи.

Основні дії можна описати так:

Таблиця 4.1

Основні етапи роботи над проектом

Етап	Опис
Ідея	На цьому етапі нам треба зрозуміти які завдання вирішуватиме наш пристрій і що від нього потрібно.
Умови реалізації	Скільки ресурсів ми готові витратити і яка ітераційна витрата самого пристрою нам потрібна.
Елементи	Одне зі складних завдань, врахувавши велику різноманітність елементної бази, – вибір елементної бази. Полегшується за рахунок наступного етапу.
Моделювання схеми	Створення віртуальної схеми/моделі, підбір елементів та знаходження помилок на етапі моделювання.
Скетч	Створення програми, яка виконуватиметься мікроконтролером. Одна з двох основних складових роботи схеми має також підпункт про використання бібліотек та алгоритм роботи.
Складання	Остаточне складання схеми, її налаштування та перевірка виконання завдань.

4.2 Допоміжні компоненти пристрою

Конвертер DC-DC LM2596

Як вже згадувалося раніше, для коректної роботи нашого gsm/gprs модулю SIM слід підібрати окреме від контролера живлення, адже Arduino не може забезпечити цей модуль необхідними 2А струму і вихідна напруга плати у 5В чи 3.3В нам також не підійде.

Мій проект забезпечений живленням від зовнішнього акумулятора на основі Li-Pol батареї який на виході видає напругу 5-20 В та струм від 1,5А до 3А. В зв'язку з цим, для стабілізації живлення я використовую понижуючий перетворювач постійної напруги LM2596 Step Down DC-DC (див. рис. 4.3). Незалежно від вхідної напруги він стабілізує напругу на виході до десятих вольт і має на виході необхідні для проекту 2А струму .

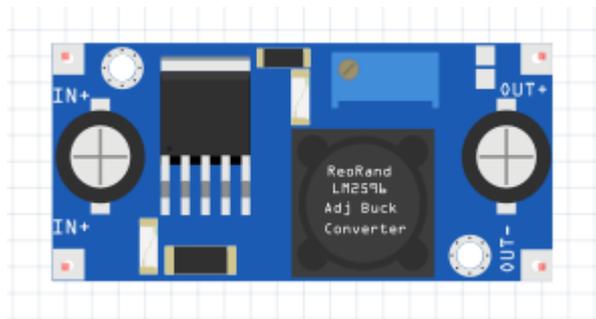


Рис. 4.1 Конвертер DC-DC LM2596 на схемі

Технічні характеристики:

- Вхідна напруга: 4 В-35 В;
- Вихідна напруга: 1.25-30 В;
- Вихідний струм: до 2 А (без встановлення радіатора);
- до 3А (макс., при організації хорошого охолодження котушки та мікросхеми);
- Ефективність перетворення: 92% (найвищий);
- Пульсація вихідного сигналу: 30мВ (можна зменшити встановивши конденсатор більшої ємності);

На платі є індикатор роботи пристрою (синій, червоний або зелений) – світлодіод, яскравість якого залежить від вихідної напруги.

Сенсорний ємнісний датчик-модуль на базі TTP223B

Важлива особливість функціоналу трекера – моніторинг аварій. Під час аварій люди часто не мають час, щоб вдатися до своїх мобільних телефонів, щоб повідомити членів сім'ї про інцидент. Тому мною було додано сенсор для відправки сигналу. Щойно буде подано сигнал, повідомлення про аварію буде доставлено членам родини, чії номери будуть зберігатися в прошивці пристрою, у вигляді GPS-координат місця аварії у Google Maps.

Для реалізації функціоналу мною було обрано датчик дотику на базі TTP223B (див. рис. 4.4). Він може використовуватись для комутації електричних кіл (вмикач/вимикач), також є відмінною заміною традиційним механічним кнопкам. Відрізняється підвищеною надійністю через відсутність рухомих частин та низьким енергоспоживанням.

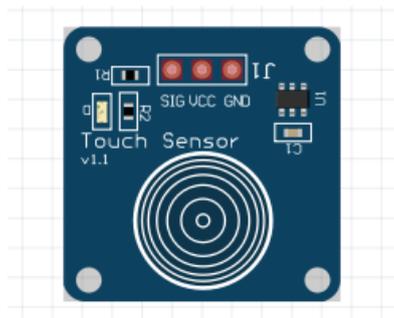


Рис. 4.2. TOUCH SENSOR TTP223B на схемі

Для використання сенсорної кнопки TTP223B її можна напряму підключити до нашого контролера Arduino. Напруга живлення модуля 2 – 5,5 В. На платі знаходиться світлодіод, позначений "D", який спалахує, коли на модуль подається живлення. На платі модуля передбачено чотири отвори для закріплення на плоскій поверхні. Сенсорна поверхня працює за ємнісною технологією. Спрацювання модуля на комутацію походить від торкання пальцем до сенсорного датчика. У стані спокою – на виході модуля низький рівень напруги, при торканні сенсора – з'являється високий рівень напруги. Після 12 секунд бездіяльності модуль перетворюється на режим зниженого енергоспоживання.

Датчик дотику TTP223В має один 3-х контактний роз'єм. Позначення контактів – SIG (вихідний цифровий сигнал); - VCC (напряга живлення); - GND (контакт заземлення).

Технічні характеристики

- Робоча напруга – 2-5,5 В;
- Чутливість – 0-50 пФ;
- Час відгуку (режим зниженого енергоспоживання) – 220 мс ;
- Час відгуку (активний режим) – 60 мс;
- Габарити - 29x16x2 мм;
- Вага – 3 г.

4.3 Моделювання схеми

Основна проблема у створенні проекту - його опис та моделювання на ранніх етапах, так ось з Arduino все набагато легше! Будь-яку схему можна перевірити, скориставшись зручним редактором схем, який також може моделювати саму програму.

Під час дослідження створення різноманітних проектів на Ардуїно, я наштовхнулась на таке програмне забезпечення як — Fritzing [].

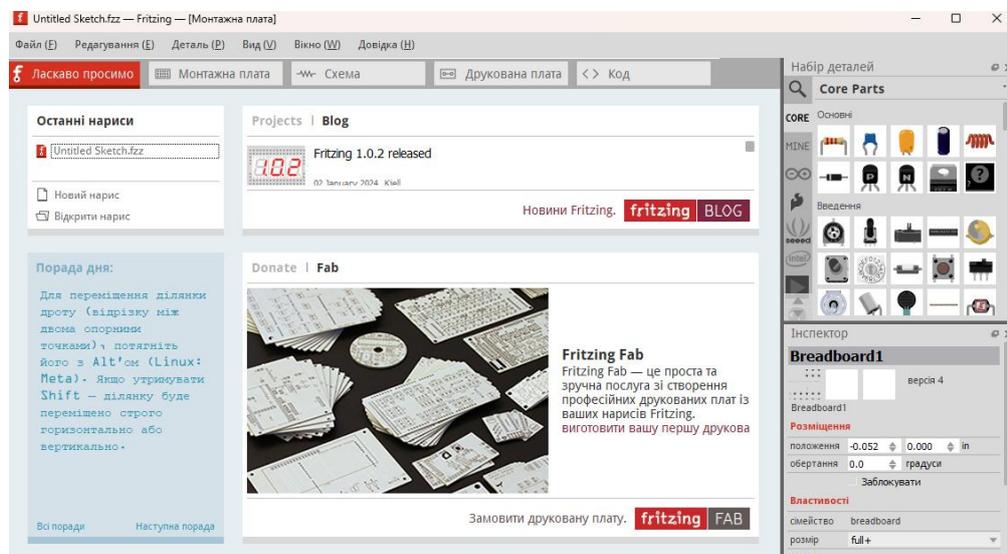


Рис.4.3. Головне меню Fritzing

Зручний інтерфейс безкоштовної, по суті альтернативної IDE для Arduino, дозволяє змоделювати схему будь-якої складності і з будь-якими елементами, які ми хотіли б. Більшість стандартних елементів вже є у бібліотеці програми, багато створено спільнотою чи є можливість створити самим, програма дозволяє нам це зробити. У пошуках відповідних комплексних програмних рішень зупинилась саме на цьому через великий функціонал програми.

Як підсумок, успішно змодельований проект нашого ArdtrackerGPS, схему підключення можна побачити нижче на рис. 4.2.

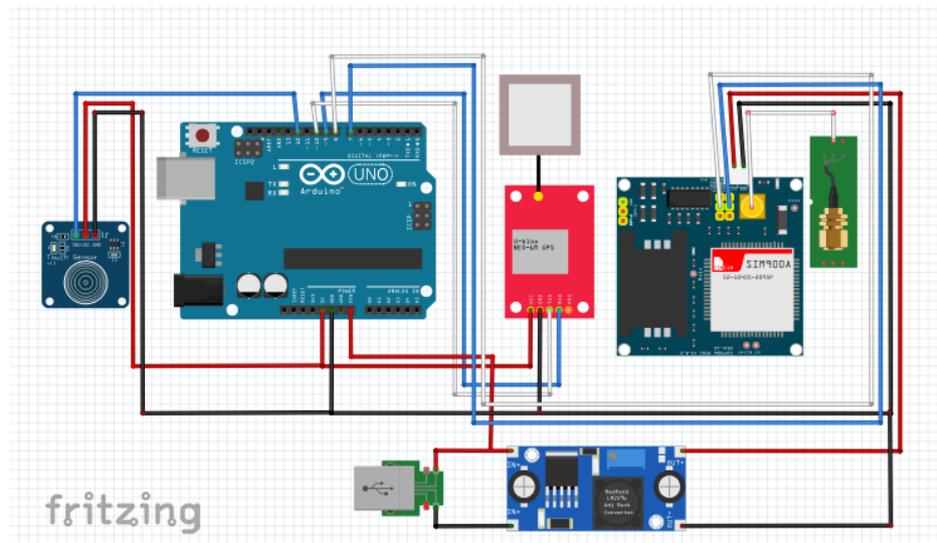


Рис. 4.4. Схема підключення пристрою у Fritzing

4.4 Підключення пристрою

Як видно зі схеми на рис. 4.4, живлення пристрою умовно розбито на дві гілки – живлення плати Arduino відбувається від зовнішнього акумулятора за допомогою USB кабеля та надходить на пін 5V. Друга гілка – живлення GSM/GPRS модуля за допомогою того ж USB кабеля, проте проходить стабілізацію через DC-DC ковнертер, який на виході до модуля налаштований на напругу 4В та струм 2А. Для коректної передачі даних напругу проведено спільне заземлення та контакти RX, TX на пінах 7 і 8 відповідно.

Що стосується підключення NeobM то тут все легше, оскільки живлення відбувається напругу за допомогою Arduino разом з контактами зв'язку на 9 та

10 пінах Arduino та заземлення як зазвичай. Сенсорний датчик під'єднано на 12 пін Arduino, живлення так само – напряму до пінів 5V та GND.

Результат фінальної спайки пристрою вже без макетної плати можна побачити на рис 4.4. В процесі пайки було здійснено налаштування DC-DC конвертера LM2596, в підборі необхідної напруги для модулю зв'язку допоміг цифровий мультиметр.

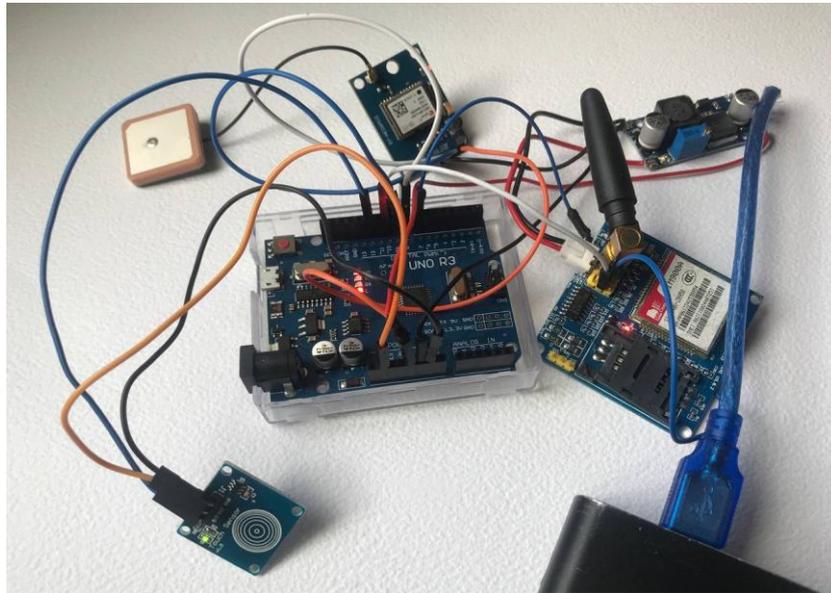


Рис. 4.4 Вигляд спаяного пристрою

4.5 Функціонал пристрою та пояснення коду

Розпочнемо з імпорту бібліотек та ініціалізації основних змінних, таких як контакти модулів GPS і GSM, сенсору, поле для зберігання номеру, на будемо отримувати СМС.

```

1  #include <SoftwareSerial.h>
2  #include <TinyGPS++.h>
3
4  const String PHONE = "+380997841***"; // телефон, на який будуть надсилатись смс
5  // RX/TX піни для модулів gps та gsm
6  #define gsmRX 7
7  #define gsmTX 8
8  #define gpsRX 9
9  #define gpsTX 10
10
11 // пін сенсору для відправки повідомлення про інцидент на дорозі
12 #define digital_sensor 12
13
14 TinyGPSPlus GPSplus;
15 SoftwareSerial NE06M(gpsRX, gpsTX);
16 SoftwareSerial SIM900A(gsmRX, gsmTX);
17

```

Рис. 4.5. Лістинг оголошення бібліотек та основних змінних

Ардуїно здійснює апаратну підтримку інтерфейсу послідовної передачі даних за допомогою пінів 0 і 1 (які також використовуються для зв'язку з комп'ютером за допомогою USB). Апаратна робота з послідовним інтерфейсом відбувається за допомогою вбудованого в мікроконтролер приймача UART . Він дозволяє мікроконтролеру ATmega328 обробляти дані навіть під час паралельної роботи над іншими завданнями.

Хоча контролер обмежений єдиним апаратним послідовним з'єднанням, ми маємо можливість програмно реалізувати додаткові підключення. Оскільки нам потрібно працювати одночасно з даними модулів GPS та GSM, то для програмної реалізації цих послідовних з'єднань на допомогу приходить бібліотека SoftwareSerial.h.

```

51 void setup()
52 {
53     SIM900A.begin(19200);
54     Serial.begin(GPSBaud);
55     NEO6M.begin(GPSBaud);
56
57     delay(5000); // час на підключення до мережі
58     Serial.println(" logging time completed!");
59

```

Рис. 4.6. Лістинг ініціалізації послідовних портів

SoftwareSerial — це бібліотека, яка забезпечує послідовний зв'язок на цифрових контактах, відмінних від послідовного порту. Використовуючи бібліотеку SoftwareSerial, ми можемо програмно створити кілька програмних послідовних портів зі швидкістю до 115200 біт/с.

Одна з основних проблем роботи з GPS приймачами, це розуміння вихідних даних та їх форматування для зручного сприйняття користувачем. Майже для всіх подібних геодатчиків, в тому числі і для обраного мною NEO6M, форматом вихідного потоку є речення NMEA (National Marine Electronics Association). Кожне поле речень складається з символів ASCII, що розділено комами. За замовчуванням модуль GPS NEO-6M здійснює відправку даних шиною UART раз на секунду (частота 1 Гц).

- GN – інформація отримана супутниками різних навігаційних систем.

Ідентифікатор рядка представлений трьома символами, що слідує за джерелом даних.

- GGA - Дані про останнє зафіксоване розташування.
- GLL – географічні координати.
- GSA – Інформація, що стосується активних супутників (що беруть участь у позиціонуванні).
- GSV - Інформація про всі супутників, що спостерігаються.
- RMC – Рекомендований мінімум навігаційних даних.
- VTG - Швидкість та курс щодо землі.
- ZDA - Дата та час.
- DHV - Інформація про швидкість руху приймача GNSS.
- GST – Статистика помилок позиціонування.
- TXT – Текстове повідомлення.

Дані складаються з параметрів, розділених комами. Тип та склад параметрів залежить від ідентифікатора рядка. Розглянемо приклад поля ідентифікатора GGA, оскільки саме повідомлення цього ідентифікатора ми форматуємо в кодї:

```
$GNGGA,101030.000,5646.96900,N,04740.69200,E,1,08,2.7,142.0
M,0.0,M,*FF/r/n
```

Призначення параметрів рядка GGA:

1. Час UTC у форматі "ГГХХСС.ССС". У прикладі "101030.000" = 10 год., 10 хв., 30.000 сек.
2. Широта у форматі "ГГММ.МММММ". У прикладі "5546.96900" = 55 ° 46.959 ' = 55.78265 °.
3. Напрямок широти: 'N'-північ / 'S'-південь. У прикладі 'N' – північ.
4. Довгота у форматі "ГГММ.МММММ". У прикладі "04740.69200" = 47 ° 40.692 ' = 47.6782 °.
5. Напрямок довготи: 'E'-схід/'W'-захід. У прикладі 'E' – схід.

6. Спосіб обчислення координат. У прикладі '1' – автономно.
7. Кількість активних супутників, від "00" до "12". У прикладі "08".
8. Горизонтальний геометричний фактор погіршення точності (HDOP). У прикладі "2.7".
9. – 10. Висота над рівнем моря (geoid), одиниці виміру висоти. У прикладі "142.0, М".
11. – 12. Різниця між еліпсоїдом землі та рівнем моря (geoid), одиниці виміру.

Тепер ми можемо розглянути функцію `displayInfo()`. Метою функції є зчитування часу, дати, широти, довготи та швидкості в кілометрах на годину з GPS приймача та вивід цих даних у монітор послідовного порту. Форматування потоку даних з NMEA на зрозумілий звичайному користувачеві формат здійснюється за допомогою бібліотеки `TinyGPS`.

```

211 // *****
212 // Функція для виводу даних в порт
213
214 void displayInfo()
215 {
216     Serial.print(F("Розміщення: "));
217
218     if (GPSplus.location.isValid())
219     {
220         Serial.print(GPSplus.location.lat(), 6);
221         Serial.print(F(", "));
222         Serial.print(GPSplus.location.lng(), 6);
223         Serial.print(" ");
224         Serial.print(F("Speed:"));
225         Serial.print(GPSplus.speed.kmph());
226     }
227     else
228     {
229         Serial.print(F("Дані локації не валідні"));
230     }

```

Рис. 4.8. Лістинг фрагменту функції для виводу локації пристрою

Функція `sms_send()` – визначена функція типу `void`, що приймає лише одну змінну як аргумент типу `string`. Цей параметр слугує контейнером для збереження тексту повідомлення. У цій функції ми передаємо номер, на який ми хочемо надіслати СМС та текст повідомлення за допомогою АТ команд.

```

162 // *****
163 // Функція відправки СМС
164
165 void sms_send(String message)
166 {
167     sms_text = message ;
168     sms_text = sms_text + "Час фіксації: " + myhrs + ":" + mymin + ":" + mysec + "\n";
169     sms_text = sms_text + "http://www.google.com/maps/place/" + mylati + "," + mylong ;
170     SIM900A.print("AT+CMGF=1\r"); // Команда на відправку смс
171     delay(100);
172     SIM900A.print("AT+CMGS=\"" + PHONE + "\"\r");
173     delay(100);
174     SIM900A.println(sms_text); //передаємо текст смс
175     delay(100);
176     SIM900A.println((char)26); // Кінець AT команди за допомогою ^Z (ASCII code 26)
177     delay(100);

```

Рис. 4.9. Лістинг функції для відправки SMS

Усі string змінні, що зберігають тексти різних варіантів повідомлень для майбутніх функцій заздалегідь ініціалізовано в невеликому блоці:

```

42
43 const String default_message = "Положення об'єкта зафіксовано :\n";
44 const String geoforce_message1 = "WARNING! Об'єкт трекінгу покинув геозону :\n ";
45 const String geoforce_message2 = "Welcome home! Об'єкт трекінгу увійшов до геозони :\n ";
46 const String incident_message = "WARNING! Відбувся інцидент за адресою :\n ";
47 const String speeding_message = "Перевищення швидкості 50 км/год :\n ";
48 const String connection_error = "GPS даних не виявлено. Перевірте обладнання \n";

```

Рис. 4.10. Лістинг string змінних для текстів SMS-повідомлень.

Варто звернути увагу на функцію `getDistance()`, на якій засновано нашу систему геозонування. Взагалі геозона – це свого роду уявна межа або ж віртуальна огорожа, що побудована навколо точки. Реалізація такої зони допоможе контролювати об'єкт трекінгу у разі небажаного перетину заданої території.

У сучасному Інтернеті речей геозони є незамінним інструментом не тільки для управління потоком інформації, але й для підвищення інтерактивності. Геозони можна встановлювати на програмних платформах і конкретні повідомлення доставляти мобільним користувачам, датчикам або пристроям у будь-якій геозоні. Інформація про певне місце розташування про об'єкти, оновлення, можливості тощо може бути доставлена на мобільні пристрої або мережі фіксованих «речей» [12].

```

184 // .....
185 // Обрахуємо дистанцію між "домом" та розташуванням трекера для геозонування
186
187 double getDistance(double lat1, double lon1, double lat2, double lon2) {
188     // змінні для обрахунків
189     double calc=0;
190     double calc2=0;
191     double diflat=0;
192     double diflon=0;
193
194     diflat = radians(lat2-lat1);
195     lat1 = radians(lat1);
196     lat2 = radians(lat2);
197     diflon = radians((lon2)-(lon1));
198
199     calc = (sin(diflat/2.0)*sin(diflat/2.0));
200     calc2 = cos(lat1);
201     calc2*=cos(lat2);
202     calc2*=sin(diflon/2.0);
203     calc2*=sin(diflon/2.0);
204     calc +=calc2;
205
206     calc=(2*atan2(sqrt(calc),sqrt(1.0-calc)));
207     calc*=6371000.0; //конвертація у метри
208
209     return calc;
210 }

```

Рис. 4.11. Лістинг функції для обрахунку дистанції

Суть функції полягає в тому аби обрахувати відстань між двома координатними GPS точками на Земній поверхні. Ця відстань, розрахована за допомогою формули гаверсинуса, слугуватиме радіусом нашої геозони.

Формула Гаверсинуса — це дуже точний спосіб обчислення відстані між двома точками на поверхні сфери за допомогою широти та довготи цих точок. Ця формула є повторним формулюванням сферичного закону косинусів. У комп'ютерних системах з низькою точністю з плаваючою комою формула сферичного закону косинусів може мати великі помилки округлення, якщо відстань невелика (якщо дві точки знаходяться на відстані кілометра одна від одної на поверхні Землі, косинус центрального кута близько 0,99999999).

Для сучасних 64-бітних чисел із плаваючою комою формула сферичного закону косинусів не має серйозних помилок округлення для відстаней, що перевищують кілька метрів на поверхні Землі. Формула Гаверсинусу (лат. *habersinus*, скорочення від *half the versed sine*) чисельно краще підходить для малих відстаней [13].

$$\text{haversine}(\theta) = \sin^2(\theta/2)$$

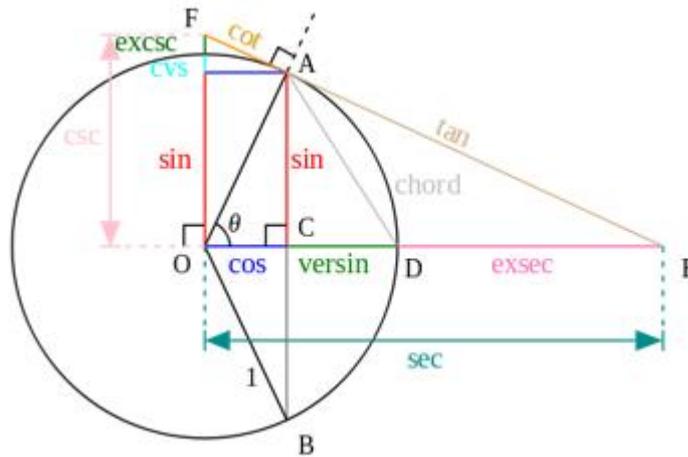


Рис. 4.12. Визначення тригонометричних функцій через коло

Наступне рівняння, це те, як ми можемо розписати наведену вище формулу, щоб мати можливість використати координати широти та довготи аби знайти відстань між точками А і В. Для зручності обчислень розбиваємо формулу на дві частини (4.1) та (4.2), також слід зауважити, що для переходу до тригонометричних функцій кути мають бути в радіанах:

$$a = \sin^2(\varphi_B - \varphi_A/2) + \cos \varphi_A \cdot \cos \varphi_B \cdot \sin^2(\lambda_B - \lambda_A /2) \quad (4.1)$$

$$c = 2 \cdot \operatorname{atan2}(\sqrt{a}, \sqrt{1-a}) \quad (4.2)$$

$$d = R \cdot c \quad (4.3)$$

де: φ – широта, λ – довгота, R – радіус Землі (середній радіус = 6371 км). Отриману відстань домножуємо на радіус Землі, аби конвертувати відстань у метри (4.3) і вже у `loop()` вона порівнюється з попередньо визначеним нами значенням відстані. Отже, якщо розрахована відстань більша за попередньо визначений максимум, це означає, що транспортний засіб або будь-який інший об'єкт перемістився за межі геозони. В такому разі користувачу надходить СМС-

сповіщення, яке повідомляє про це і надсилає URL-посилання з георозміщенням на Google Maps.

```

101 // Геозонування
102 if (hasRun == false){
103     if(distance > maxDist) {
104         sms_send(geoforce_message1);
105         hasRun = true;
106     }
107 }
108 if (hasRun == true){
109     if(distance < maxDist) {
110         sms_send(geoforce_message2);
111         hasRun = false;
112     }
113 }
114 // *****

```

Рис. 4.13. Лістинг реалізації геозони у loop()

Перейдемо до головних функцій будь-якого скетчу – loop() та setup(). Setup() в даному проекті слугує для ініціалізації роботи з послідовним портом та визначення режиму роботи піну сенсора. Крім цього всередині цієї функції відбувається згадана раніше перевірка SIM-модуля за допомогою AT-команд.

```

64 SIM900A.println("AT"); // Перевірка gsm модуля
65 delay(1000);
66 SIM900A.println ( " AT+CSQ " ); // Перевірка якості сигналу, діапазон значень 0-31, 31 є найкращим
67 delay(1000);
68 SIM900A.println("ATE1"); //Echo ON
69 delay(1000);
70 SIM900A.println("AT+CPIN?"); // Перевіряємо чи готова SIM
71 delay(1000);
72 SIM900A.println("AT+CMGF=1"); // Переходимо до першого режиму (текстовий)
73 delay(1000);
74 SIM900A.println("AT+CNMI=1,1,0,0,0"); // Вирішує, як слід обробляти щойно отримані SMS
75 delay(1000);
76
77

```

Рис. 4.14. Лістинг перевірки GSM модуля AT командами

Усередині функції loop() ми постійно перевіряємо, чи доступні на даний момент дані для виводу з нашого модуля GPS, потім ми зчитуємо та кодуємо дані, а потім негайно викликаємо функцію displayInfo() та активуємо GSM модуль, передаючи дані користувачу.

За активації сенсорного датчику, відповідній особі надсилається повідомлення про аварію. Повідомлення складається з URL-посилання карти Google і значень широти та довготи.

```
// ****  
// для сенсору  
  
if(digitalRead(digital_sensor) == HIGH)  
{  
  displayInfo();  
  sms_send(incident_message);  
  delay(2000);  
}
```

Рис. 4.15. Сповіщення за допомогою сенсору

Також слід згадати про реалізацію моніторингу перевищення швидкості водієм транспортного засобу. Перевірка швидкості відбувається за класикою всередині функції loop().

```
122 // ****  
123 // контроль швидкості  
124 if (GPSplus.speed.kmph() > maxSpeed )  
125 {  
126   sms_send(speeding_message);  
127   delay(2000);  
128   displayInfo();  
129 }  
130
```

Рис. 4.16. Попередження про перевищення швидкості

ВИСНОВКИ

Невід’ємною складовою сучасної навігації безумовно є система GPS, при чому не лише в цивільних, а й у військових сферах діяльності. Завданням цього проекту була розробка доступного та універсального трекера, який буде вигравати у ціні серед конкурентів на ринку, проте не поступатиметься у функціоналі. В ході виконання кваліфікаційної роботи було спроектовано та реалізовано такий пристрій – ArdtrackerGPS. Подібний пристрій може бути використаний для вирішення необмеженої кількості завдань – від відстеження цивільних об’єктів до оптимізації та автоматизації руху літальних апаратів. В процесі було розроблено наступний функціонал:

- алгоритм відстеження транспортного засобу в режимі реального часу шляхом відправки SMS-повідомлень за запитом;
- геозону, що являє собою свого роду протиугонну систему. Система сповіщає власника повідомленням у разі перетину межі;
- алгоритм, що попереджає про перевищення встановленої у системі допустимої швидкості;
- алгоритм моніторингу інцидентів на дорозі за допомогою сенсорного датчика.

Під час виконання даної роботи було систематизовано знання з програмування мікроконтролерів, що дозволило розробити систему з використанням модулів мобільного зв’язку та геолокації. Пристрій на основі цих технологій може стати в нагоді широкому колу користувачів. ArdtrackerGPS допоможе точно визначати розташування об’єкту та здійснювати його моніторинг в заданій зоні, що неодмінно стане у нагоді користувачу як в побуті, так і може стати основою для масштабного налагодження та оптимізації системи транспортного руху в різних сферах діяльності.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. A Complete Guide to Microcontrollers for IoT [Electronic resource] // Nabto. - Made of access: “<https://www.nabto.com/iot-microcontroller-guide/>”, free (viewed on 15.04.2024). - Title from the screen.
2. Arduino. [Електронний ресурс] // Вікіпедія. - Режим доступу: “<https://uk.wikipedia.org/wiki/Arduino>”, вільний (дата звернення: 15.04.2024). - Назва з екрану.
3. GPS [Електронний ресурс] // Вікіпедія. - Режим доступу: “<https://uk.wikipedia.org/wiki/GPS>”, вільний (дата звернення: 21.04.2024). - Назва з екрану.
4. GPS: What is it and how does it work? [Electronic resource] // Space. - Made of access: “<https://www.space.com/gps-what-is-it>”, free (viewed on 21.04.2024). - Title from the screen.
5. Neo-6m Datasheet [Electronic resource] // waveshare - Made of access: “<https://www.waveshare.com/w/upload/2/2c/NEO-6-DataSheet.pdf>”, free (viewed on 21.04.2024). - Title from the screen.
6. GSM [Електронний ресурс] // Вікіпедія. - Режим доступу: “<https://uk.wikipedia.org/wiki/GSM>”, вільний (дата звернення: 25.04.2024). - Назва з екрану.
7. GSM (Global System for Mobile communication) [Electronic resource] // TechTarget - Made of access: “<https://www.techtarget.com/searchmobilecomputing/definition/GSM>”, free (viewed on 26.04.2024). - Title from the screen.
8. GPRS [Електронний ресурс] // Вікіпедія. - Режим доступу: “<https://uk.wikipedia.org/wiki/GPRS>”, вільний (дата звернення: 10.05.2022). - Назва з екрану.
9. What Is GPRS (General Packet Radio Service)? Meaning, Working, Advantages, and Applications) [Electronic resource] // Spiceworks - Made of

- access: “<https://www.spiceworks.com/tech/networking/articles/what-is-gprs/>”, free (viewed on 10.05.2024). - Title from the screen.
10. An Introduction to AT Commands [Electronic resource] // Twilio - Made of access: “<https://www.twilio.com/docs/iot/supersim/introduction-to-modem-at-commands>”, free (viewed on 10.05.2024). - Title from the screen.
11. SIM900A Datasheet [Electronic resource] // DatasheetsPDF - Made of access: “<https://datasheetspdf.com/datasheet/SIM900A.html>”, free (viewed on 14.05.2024). - Title from the screen.
12. What is a geozone? [Electronic resource] // Wyld - Made of access: “<https://wyldnetworks.com/blog/what-is-a-geozone>”, free (viewed on 30.05.2024). - Title from the screen.
13. Distance on a sphere: The Haversine Formula [Electronic resource] // esri - Made of access: “<https://community.esri.com/t5/coordinate-reference-systems-blog/distance-on-a-sphere-the-haversine-formula/ba-p/902128>”, free (viewed on 30.05.2024). - Title from the screen.

ДОДАТКИ

Додаток А

Лістинг коду

```

#include <SoftwareSerial.h>
#include <TinyGPSPlus.h>

const String PHONE = "+380997841478"; // телефон, на який будуть надсилатись
смс
// RX/TX піни для модулів gps та gsm
#define gsmRX 7
#define gsmTX 8
#define gpsRX 9
#define gpsTX 10

// пін сенсору для відправки повідомлення про інцидент на дорозі
#define digital_sensor 12

TinyGPSPlus GPSplus;
SoftwareSerial NEO6M(gpsRX, gpsTX);
SoftwareSerial SIM900A(gsmRX, gsmTX);

static const uint32_t GPSBaud = 9600;
boolean hasRun = false;

char inchar;
String sms_text;
const double maxDist = 100;
const double maxSpeed = 20;
float initLong = 0;
float initLat = 0;

float longitude;
float latitude;
char buff[10];
double distance;

//тимчасове зберігання даних для виводу
String mylong = "";
String mylati = "";
String myhrs = "";
String mymin = "";
String mysec = "";

```

```

int gps_flag = 0;
int count = 0;
int msgstatus;

const String default_message = "Положення об'єкта зафіксовано :\n";
const String geoforce_message1 = "WARNING! Об'єкт трекінгу покинув геозону :\n ";
const String geoforce_message2 = "Welcome home! Об'єкт трекінгу увійшов до геозони :\n ";
const String incident_message = "WARNING! Відбувся інцидент за адресою :\n ";
const String speeding_message = "Перевищення швидкості 20 км/год :\n ";
const String connection_error = "GPS даних не виявлено. Перевірте обладнання \n";
const String message_received = "повідомлення надіслано: ";

void setup()
{
  SIM900A.begin(115200);
  Serial.begin(GPSBaud);
  NEO6M.begin(GPSBaud);

  delay(5000); // час на підключення до мережі
  Serial.println(" logging time completed!");

  randomSeed(analogRead(0));
  pinMode(digital_sensor, INPUT_PULLUP);

  SIM900A.println("AT"); // Перевірка gsm модуля
  delay(1000);
  SIM900A.println ( " AT+CSQ " ); // Перевірка якості сигналу, діапазон значень
0-31, 31 є найкращим
  delay(1000);
  SIM900A.println("ATE1"); //Echo ON
  delay(1000);
  SIM900A.println("AT+CPIN?"); // Перевіряємо чи готова SIM
  delay(1000);
  SIM900A.println("AT+CMGF=1"); // Переходимо до першого режиму
(текстовий)
  delay(1000);
  SIM900A.println("AT+CNMI=1,1,0,0,0"); // Вирішує, як слід обробляти щойно
отримані SMS
  delay(1000);
}

```

```

void loop()
{
  while (NEO6M.available() > 0)
  if (((GPSplus.encode(NEO6M.read())) && (gps_flag == 0))
  {
    displayInfo(); //транслюємо інформацію в порт
    gps_flag = 1;
    latitude = GPSplus.location.lat(), 6 ;
    longitude = GPSplus.location.lng(), 6 ;

    distance = getDistance(latitude, longitude, initlLat, initLong);
  }

  if (millis() > 5000 && GPSplus.charsProcessed() < 10)
  { //у разі, коли ми не отримуємо сигнал
    Serial.println(F("GPS даних не виявлено. Перевірте обладнання \n"));
    while(true);
  }
  //
  *****/
  // Геозонування

  if (hasRun == false){
    if(distance > maxDist) {
      sms_send(geoforce_message1);
      hasRun = true;
    }
  }
  if (hasRun == true){
    if(distance < maxDist) {
      sms_send(geoforce_message2);
      hasRun = false;
    }
  }
  //
  *****/
  // для сенсору

  if(digitalRead(digital_sensor) == HIGH)
  {
    displayInfo();
  }
}

```

```

    sms_send(incident_message);
    delay(2000);
    displayInfo();
}
//
*****
*****/
// КОНТРОЛЬ ШВИДКОСТІ
if (GPSplus.speed.kmph() > maxSpeed )
{
    sms_send(speeding_message);
    delay(2000);
    displayInfo();
}

count = count + 1;
if ( (count > 50) && ( count < 100))
{
    gps_flag = 0;
    NEO6M.begin(GPSBaud);
}
if ( count > 100 )
count = 0;
if(gps_flag == 1)
{
    SIM900A.begin(19200); // встановлюємо послідовний зв'язок знову
    if(SIM900A.available() > 0)
    {
        Serial.println(message_received);
        inchar=SIM900A.read();
        Serial.println(inchar);
        displayInfo();

        sms_send(default_message);
        inchar = 'z'; // "garbage value"
        delay(2000);
        gps_flag = 0;
        NEO6M.begin(GPSBaud); // активуємо gps назад замість gsm
        delay(100);
    }
}
}
}

```

```

//
*****
*****/
// Функція відправки СМС

void sms_send(String message)
{
  latitude = GPSplus.location.lat(), 6 ;
  longitude = GPSplus.location.lng(), 6 ;
  mylati = dtostrf(latitude, 3, 6, buff);
  mylong = dtostrf(longitude, 3, 6, buff);
  if (GPSplus.time.hour() < 10) myhrs = myhrs + "0";
  myhrs = myhrs + dtostrf(GPSplus.time.hour(), 2, 0, buff);
  if (GPSplus.time.minute() < 10) mymin = mymin + "0";
  mymin = mymin + dtostrf(GPSplus.time.minute(), 2, 0, buff);
  if (GPSplus.time.second() < 10) mysec = mysec + "0";
  mysec = mysec + dtostrf(GPSplus.time.second(), 2, 0, buff);

  sms_text = message ;
  sms_text = sms_text + "Час фіксації: " + myhrs + ":" + mymin + ":" + mysec + "\n";
  sms_text = sms_text + "http://www.google.com/maps/place/" + mylati + "," + mylong
;
  SIM900A.print("AT+CMGF=1\r"); // Команда на відправку смс
  delay(100);
  SIM900A.print("AT+CMGS=\"" + PHONE + "\"\r");
  delay(100);
  SIM900A.println(sms_text); //передаємо текст смс
  delay(100);
  SIM900A.println((char)26); // Кінець AT команди за допомогою ^Z (ASCII code
26)
  delay(100);
  SIM900A.println();
  delay(5000); // На відправку смс потрібен певний час
  Serial.println(sms_text);
  Serial.println("message sent.");
  sms_text = "";
}
//
*****
*****/
// Обраховуємо дистанцію між "домом" та розташуванням трекера для
геозонування

float getDistance(float lat1, float lon1, float lat2, float lon2) {

```

```

// змінні для обрахунків
float calc=0;
float calc2=0;
float diflat=0;
float diflon=0;

diflat = radians(lat2-lat1);
lat1 = radians(lat1);
lat2 = radians(lat2);
diflon = radians((lon2)-(lon1));

calc = (sin(diflat/2.0)*sin(diflat/2.0));
calc2 = cos(lat1);
calc2*=cos(lat2);
calc2*=sin(diflon/2.0);
calc2*=sin(diflon/2.0);
calc +=calc2;

calc=(2*atan2(sqrt(calc),sqrt(1.0-calc)));
calc*=6371000.0; //конвертація у метри

return calc;
}
void formatData() {
latitude = GPSplus.location.lat(), 6 ;
mylati = dtostrf(latitude, 3, 6, buff);
longitude = GPSplus.location.lng(), 6 ;
mylong = dtostrf(longitude, 3, 6, buff);

}
*****
*****/
// Функція для виводу даних в порт

void displayInfo()
{
Serial.print(F("Розміщення: "));

if (GPSplus.location.isValid())
{
Serial.print(GPSplus.location.lat(), 6);
Serial.print(F(", "));
Serial.print(GPSplus.location.lng(), 6);
Serial.print(" ");
}
}

```

```
    Serial.print(F("Speed:"));
    Serial.print(GPSplus.speed.kmph());
}
else
{
    Serial.print(F(" Дані локації не валідні"));
}
Serial.print(F(" Дата: "));

if (GPSplus.date.isValid())
{
    Serial.print(GPSplus.date.month());
    Serial.print(F("/"));
    Serial.print(GPSplus.date.day());
    Serial.print(F("/"));
    Serial.print(GPSplus.date.year());
}
else
{
    Serial.print(F(" Дані дати не валідні"));
}
Serial.print(F(" Час: "));
if (GPSplus.time.isValid())
{
    if (GPSplus.time.hour() < 10) Serial.print(F("0"));
    Serial.print(GPSplus.time.hour());
    Serial.print(F(":"));
    if (GPSplus.time.minute() < 10) Serial.print(F("0"));
    Serial.print(GPSplus.time.minute());
    Serial.print(F(":"));
    if (GPSplus.time.second() < 10) Serial.print(F("0"));
    Serial.print(GPSplus.time.second());
}
else
{
    Serial.print(F(" Дані часу не валідні"));
}
Serial.println();
}
```