

СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ВОЛОДИМИРА ДАЛЯ  
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ЕЛЕКТРОНІКИ  
КАФЕДРА ПРОГРАМУВАННЯ ТА МАТЕМАТИКИ

ПОЯСНЮВАЛЬНА ЗАПИСКА  
до кваліфікаційної випускної роботи

освітній ступінь	бакалавр <small>(бакалавр, магістр)</small>
спеціальність	123 «Комп'ютерна інженерія» <small>(шифра і назва спеціальності)</small>
спеціалізація	 <small>(назва спеціалізації)</small>

на тему «Комп'ютерна система ідентифікації геопозиціонування об'єктів у просторі»

Виконав: студент групи СКС-16	 <small>(підпис)</small>	Козуб Е.В. <small>(ініціали і прізвище)</small>
Керівник	 <small>(підпис)</small>	к.т.н., доц. Захожай О.І. <small>(ініціали і прізвище)</small>
Завідувач кафедри	 <small>(підпис)</small>	д.т.н., доц. Лифар В. О. <small>(ініціали і прізвище)</small>
Рецензент	 <small>(підпис)</small>	 <small>(ініціали і прізвище)</small>

СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ВОЛОДИМИРА ДАЛЯ  
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ЕЛЕКТРОНІКИ  
КАФЕДРА ПРОГРАМУВАННЯ ТА МАТЕМАТИКИ

освітній ступінь бакалавр  
(бакалавр, магістр)

спеціальність 123 «Комп'ютерна інженерія»  
(шифр і назва спеціальності)

спеціалізація \_\_\_\_\_  
(назва спеціалізації)

**ЗАТВЕРДЖУЮ**  
**Завідувач кафедри КНІ,**  
Д.Т.Н., доцент  
\_\_\_\_\_ Лифар В. О.  
“ \_\_\_\_\_ ” \_\_\_\_\_ 20\_\_ року

**З А В Д А Н Н Я**  
**НА КВАЛІФІКАЦІЙНУ ВИПУСКНУ РОБОТУ СТУДЕНТУ**  
**КОЗУБ ЕДУАРД ВІКТОРОВИЧ**

- Тема роботи «Комп'ютерна система ідентифікації геопозиціонування об'єктів у просторі»  
Керівник роботи: к.т.н., доцент Захожай О.І.,  
затверджений наказом університету від “ \_\_\_\_\_ ” \_\_\_\_\_ 20\_\_ року № \_\_\_\_\_
- Строк подання студентом роботи \_\_\_\_\_
- Вихідні дані до роботи: способи побудови систем ідентифікації геопозиціонування об'єктів у просторі. Розробка апаратного та програмного забезпечення.
- Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити): аналітичний огляд, розробка апаратного забезпечення, розробка програмного забезпечення, висновки.
- Перелік графічного матеріалу (з точним зазначенням обов'язкових креслеників)

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання \_\_\_\_\_

**КАЛЕНДАРНИЙ ПЛАН**

№ з/п	Назва етапів виконання кваліфікаційної випускної роботи	Строк виконання етапів	Примітка
1	Складання плану роботи	01.04.20	
2	Аналіз літератури	05.04.20	
3	Вивчення і підбирання матеріалу	10.04.20	
4	Написання розділів	20.04.20	
5	Оформлення пояснювальної записки	10.05.20	
6	Оформлення графічного матеріалу	25.05.20	
7	Підготовка доповіді і слайдів для презентації	07.06.20	

Студент

Керівник роботи

\_\_\_\_\_

(підпис)

\_\_\_\_\_

(підпис)

Козуб Е.В

\_\_\_\_\_

(ініціали і прізвище)

к.т.н., доц. Захожай О.І.

\_\_\_\_\_

(ініціали і прізвище)

## РЕФЕРАТ

Робота містить: 54 сторінок основного тексту, 0 сторінок додатків, 17 рисунків, 14 таблиць, 14 використаних джерел.

У даній роботі досліджено і розглянуто способи побудови систем ідентифікації геопозиціонування об'єктів у просторі, наведені основні рекомендації щодо створення схеми і друкованої плати, створено програмне забезпечення.

Створена комп'ютерна система ідентифікації геопозиціонування об'єктів у просторі з можливістю відслідковувати об'єкт за допомогою користувальницького інтерфейсу.

## ЗМІСТ

<b>ВСТУП</b>	5
<b>РОЗДІЛ 1. АНАЛІТИЧНИЙ ОГЛЯД</b>	7
1.1 Призначення і класифікація систем моніторингу рухливих об'єктів.	7
1.2 Трекер.	8
1.3 Програмне забезпечення.	11
<b>РОЗДІЛ 2. РОЗРОБКА АПАРАТНОГО ЗАБЕЗПЕЧЕННЯ</b>	13
2.1 Розробка структурної схеми.	13
2.2 Розробка схеми електричної принципової.	14
2.3 Вибір елементів в схемі.	16
2.4 Розробка конструкції друкованої плати.	22
2.5 Вибір умов охолодження і розрахунок теплового режиму.	28
2.6 Розрахунок надійності пристрою.	34
<b>РОЗДІЛ 3. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ</b>	36
3.1 Розробка програмного забезпечення мікроконтролера.	36
3.2 Розробка бази даних.	43
3.3 Розробка http-сервера.	45
3.4 Розробка користувальницького інтерфейсу.	47
<b>ВИСНОВКИ</b>	49
<b>СПИСОК ЛІТЕРАТУРИ</b>	50

## ВСТУП

Актуальність досліджень: кожен процес в будь-якій компанії, нехай навіть невеликий, потребує оптимізації. Засоби для досягнення ефективності будь-якого процесу розвиваються з кожним роком і невід'ємно входять в наше життя. Моніторинг місцезнаходження рухомих об'єктів дає додаткові можливості по аналізу і обліку робочого часу персоналу і транспортних засобів, дозволяє оптимізувати витрати ресурсів і оцінити ефективність їх використання.

Система моніторингу рухомих об'єктів - система побудована на основі систем супутникової навігації, обладнання та технологій стільникового і / або радіозв'язку, обчислювальної техніки і цифрових карт. Принцип роботи полягає у відстеженні та аналізі просторових і часових координат транспортного засобу.

Об'єкт досліджень: системи моніторингу рухомих об'єктів.

Предмет досліджень: способи побудови систем моніторингу рухомих об'єктів.

Завдання дослідження: розробка простої та дешевої системи моніторингу рухомих об'єктів з можливістю відслідковувати об'єкт за допомогою користувальницького інтерфейсу.

Методи дослідження: системний аналіз, низькорівневе програмування, високорівневе програмування.

Практичне значення отриманих результатів: отримані результати можуть бути використані в різних сферах застосування:

- транспортні компанії;
- страхові компанії;
- служби перевезення пасажирів;
- сільськогосподарські підприємства;
- автопарки;
- інкасаторські служби;
- торгові компанії;
- служби порятунку;

- охоронні служби;
- кур'єрська і поштові служби;
- будівельні компанії;
- таксопарки і диспетчерські служби таксі;
- служби екстреної допомоги;
- комунальні служби.

## РОЗДІЛ 1. АНАЛІТИЧНИЙ ОГЛЯД

### 1.1 Призначення і класифікація систем моніторингу рухливих об'єктів.

Система GPS-моніторингу складається з двох основних частин - пристрої для відстеження транспортних засобів і програмного забезпечення для управління автопарком. Коли парк обладнаний пристроями стеження, а компанія використовує передове програмне забезпечення для управління парком в реальному часі, ресурси, необхідні для щоденної експлуатації парку, можуть бути значно скорочені, а їх продуктивність збільшена.

Програмне забезпечення GPS-моніторингу призначене для збору даних про місцезнаходження транспортного засобу і візуалізації їх на карті. Воно відслідковує пристрої дуже просто і зручно, всі записи доступні в режимі реального часу, дані зберігаються в базі даних і можуть бути доступні в будь-який час в майбутньому. Трекери збирають різну інформацію, таку як координати, швидкість, висота та інше.

Системи моніторингу рухливих об'єктів класифікують за такими технічними та функціональними ознаками:

- За методом передавання інформації про місцезнаходження користувачу:
  - online – інформація про координати передається дистанційно;
  - offline – дані зчитуються після прибуття на диспетчерський пункт.

На об'єкті встановлюється модуль, що має наступні частини: модуль визначення місця розташування, модуль зберігання та модуль, що передає координатні дані. Програмне забезпечення мобільного модуля визначає координати, записує їх в модуль зберігання і по можливості відправляє за допомогою модуля передачі.

Модуль передачі може передавати дані, використовуючи бездротові мережі операторів мобільного зв'язку. Отримані дані обробляються і видаються диспетчеру у виді тексту або із використанням картографічної інформації.



У offline варіанті немає необхідності дистанційної передачі даних. Завдяки цьому можна використовувати більш дешеві мобільні модулі і дозволяє відмовитися від послуг операторів мобільного зв'язку.

За об'єктом моніторингу:

- людина – контроль переміщення, факту і часу відвідування заданих геозон (торгових точок, офісів клієнтів тощо);
- транспорт – контроль роботи і оптимізація діяльності транспортного парку;
- вантажі – контроль місцезнаходження і стану вантажу.

За способом живлення:

- автономні – працює від вбудованого акумулятора або батарейок;
- неавтономні – пристрій живиться від бортової мережі транспортного засобу.

Система моніторингу рухливих об'єктів включає наступні компоненти [1]:

- об'єкт, обладнаний контролером або трекером, котрий отримує дані від супутників і відправляє їх на серверний центр моніторингу;
- серверний центр, що має програмне забезпечення для прийому, зберігання, обробки і аналізу даних;
- комп'ютер диспетчера, який веде моніторинг.

Основним елементом в системі моніторингу рухливих об'єктів є – трекер, докладніше про нього розглянемо у пункті 1.2.

## **1.2 Трекер.**

Трекер – пристрій який визначає свої координати і певним чином обробляє ці дані. Трекери можуть бути активні, які дозволяють відслідковувати в режимі реального часу, і пасивні. Пасивні трекери не дозволяють користувачам стежити за кожним рухом відстежуваної людини або об'єкта. Інформація повинна бути отримана і потім завантажена на комп'ютер. Ці типи трекерів, як правило, більш доступні, так як вони не вимагають щомісячної плати. Активні трекери, з іншого боку, обробляють і доставляють дані в режимі реального часу, використовуючи

SIM-карту і приймач GSM. Диспетчери отримують інформацію в режимі реального часу через стільникову мережу GSM, щомісячна плата стягується за оплату послуг стільникового зв'язку. На рис. 1.1 зображені спрощені структурні схеми пристрою.

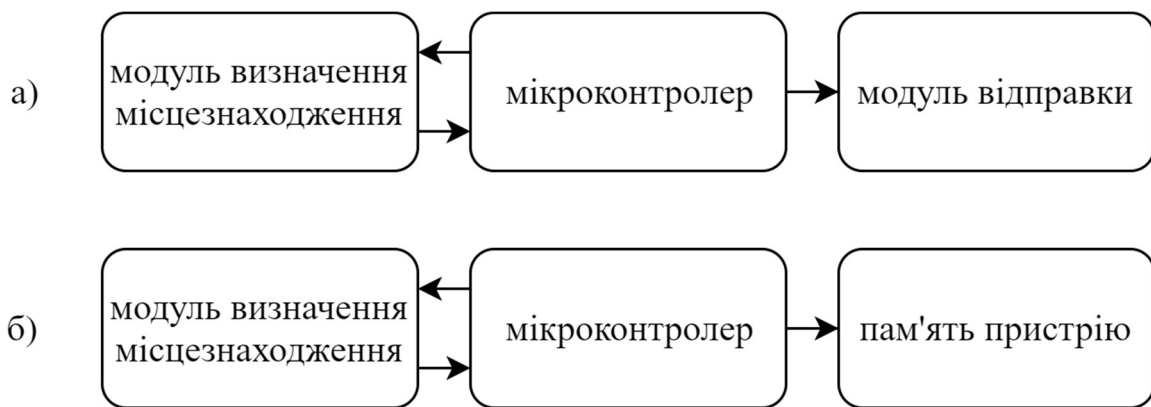


Рисунок 1.1 — Структурна схема а) online рішення б) offline рішення

Існує різноманіття способів визначення місцеположення, такі як супутникова навігація GNSS (Global Navigation Satellite System), визначення місця розташування по бездротових мережах WiFi (Wireless Fidelity) і по мережах мобільного зв'язку.

Найбільшу точність визначення місцеположення забезпечує GNSS. На даний момент супутникова навігація використовується у більшості систем відстежування місцеположення. Але, не во всіх прикладних застосуваннях вона може вирішити поставлені завдання. По-перше, пошук супутників іноді займає кілька хвилин, хоча бувають ситуації, в яких швидкість визначення важливіша за точність. По-друге, зазвичай, зв'язок зі супутниками втрачається у приміщеннях або під землею. У таких випадках, можна використовувати альтернативу – визначати місце розташування за допомогою найближчих GSM-вишок. Враховуючи наскрізне покриття мережі мобільних операторів, такий спосіб забезпечує стабільність визначення місцеположення.

Спосіб визначення розташування об'єктів полягає в тому, що мобільний телефон, або модуль мобільного зв'язку у будь-який момент часу прив'язаний до

однієї з найближчих базових станцій. Усі базові станції мають статичне місце розташування, тому, якщо в системі буде знаходитися інформація про координати їхнього розміщення – можна приблизно визначити місце розташування будь-якого GSM приймача. Всі модулі мобільного зв'язку підтримують команди, що дозволяють дізнатися яка базова станція обслуговує модуль у поточний момент і рівень сигналу від базової станції. Деякі модулі дозволяють отримувати інформацію не тільки про базову станцію прив'язки, але також і про сусідні, що дозволить підвищити точність локації.

Для побудови системи відстежування розташування рухомих об'єктів існує основна задача, пов'язана з визначенням місця розташування базових станцій. Для цього може бути використаний ряд доступних комерційних і загальнодоступних баз даних і послуг Cell ID.

Найбільша в світі відкрита база даних базових станцій – OpenCellID має в своїй базі приблизно 40 млн вишок. А в одній з найбільших комерційних баз – Unwiredlabs близько 150 млн. На рис. 1.2 представлена щільність покриття базовими станціями різних операторів поверхні Землі.

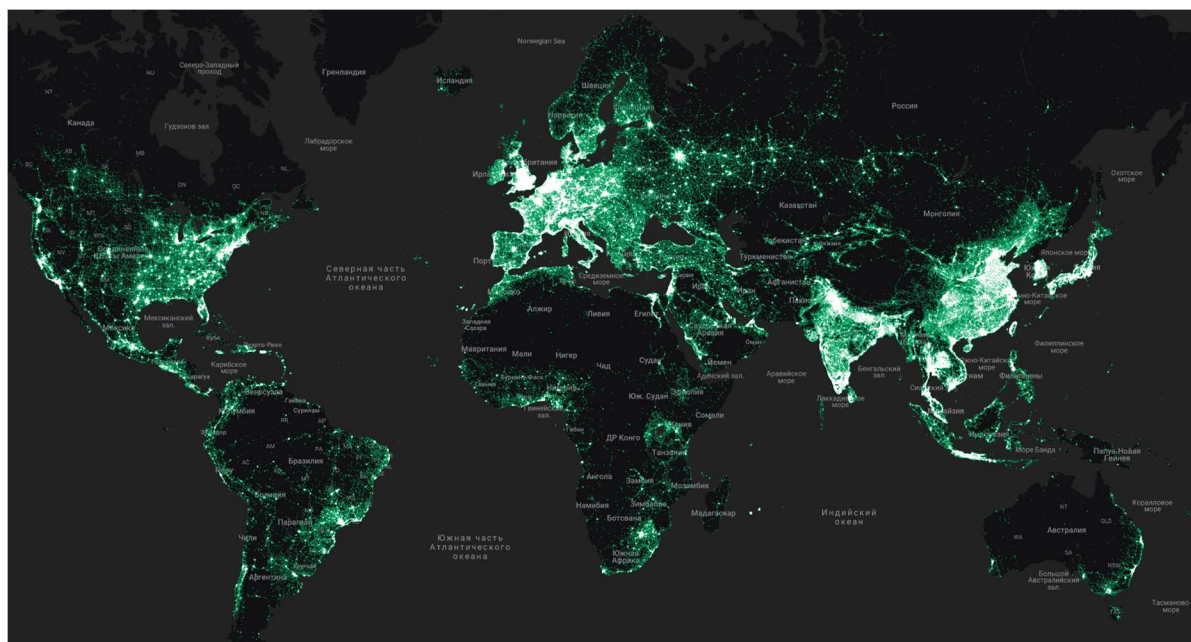


Рисунок 1.2 – Щільність розташування базових станцій мобільного зв'язку в базі Unwiredlabs

З рис. 1.2 можна зробити висновок, що на основі даних існуючих сервісів визначення розташування базових станцій можна успішно здійснювати

відстеження розташування рухомих об'єктів у найбільш розвинутих районах Землі. Тому, задача визначення місцеположення об'єктів по мережам мобільного зв'язку може бути успішно реалізована. Також слід зазначити, що відмова від використання GNSS значно зменшить вживану енергію для модулів локації, що дозволить збільшити їхню автономність та покращити маса-габаритні показники.

Інші елементи системи моніторингу рухливих об'єктів розглянемо у розділі 1.3.

### **1.3 Програмне забезпечення.**

Істотною відмінністю значної кількості систем моніторингу місцезнаходження, є функції серверного та клієнтського програмного забезпечення, так як є велике різноманіття способів обробки даних, створення звітів та ін.

Сервером може виступати як звичайний персональний комп'ютер з інстальованим програмним забезпеченням для нескладних систем моніторингу, так і складна система серверів, що складається з декількох серверів, які мають різні завдання, може виконувати одночасний моніторинг великої кількості автомобілів і гарантувати одночасне підключення до системи сотень диспетчерів (користувачів).

Диспетчерське програмне забезпечення для моніторингу рухомих об'єктів можна умовно розділити на декілька типів [1]:

- ПО, що використовує web-інтерфейс, що дозволяє уникнути установки будь-яких спеціальних компонентів і вести моніторинг з будь-якого комп'ютера, підключеного до інтернету;
- ПО, що має клієнтську частину, яке інстальюється на комп'ютери диспетчерів;
- ПО, що включає в себе всі компоненти, такі як карти і бази даних руху об'єктів на одному комп'ютері.

Різновидом варіанту, що використовує трірівневу архітектуру, де функції і компоненти центру обробки даних розподілені між декількома серверами: баз

даних, картографічної підсистеми, телекомунікаційним сервером і сервером додатка, що забезпечує роботу web-інтерфейсу користувача.

Програмне забезпечення для моніторингу як правило має ряд інтерфейсів. Вхід користувачів в систему моніторингу найчастіше захищений паролем для запобігання несанкціонованого доступу до інформації. У системах існує певна ієрархічна структура, при якій адміністратор системи моніторингу управляє правами доступу різних користувачів до різних об'єктів моніторингу і різних функцій програми.

Найпоширеніші функції, які присутні в більшості систем супутникового моніторингу:

- підключення та налаштування трекерів в системі;
- підключення та налаштування датчиків в системі;
- моніторинг поточного стану транспорту на мапі;
- моніторинг стану приладів і датчиків транспортного засобу;
- перегляд маршруту переміщення і пробігу автомобіля за вибраний інтервал часу;
- створення точок інтересу і геозон на карті;
- контроль переміщення з / в геозон;
- настройка повідомлень, що висилаються системою, коли відбуваються певні події (перевищення швидкості, слив палива і ін.);
- настройка шаблонів звітів, виконання звітів;
- побудова графіків на підставі даних системи;
- управління об'єктами моніторингу через SMS команди або CSD з'єднання;
- створення маршрутів і шляхових точок, контроль дотримання маршруту.

## РОЗДІЛ 2. РОЗРОБКА АПАРАТНОГО ЗАБЕЗПЕЧЕННЯ

### 2.1 Розробка структурної схеми.

Джерело живлення підключається до мікроконтролера, а для SIM-модуля, якому потрібна інша напруга для роботи, живлення підключається через регулятор напруги. Основою обробки інформації в схемі є - мікроконтролер. Мікроконтролер формує запити про поточне місцезнаходження для SIM-модуля. SIM-модуль їх виконує, відправляє результат мікроконтроллеру, мікроконтролер обробляє ці дані, формує повідомлення для сервера і дає команду відправити це повідомлення на сервер, після чого робить відлік до наступного циклу. Через те що мікроконтролер та SIM-модуль мають різні рівні напруги, двонаправлений транслятор напруги використовується для того, щоб мікроконтролер та SIM-модуль змогли спілкуватися через універсальний асинхронний приймач/передавач (UART). Електрична структурна схема показана на рис. 2.1.

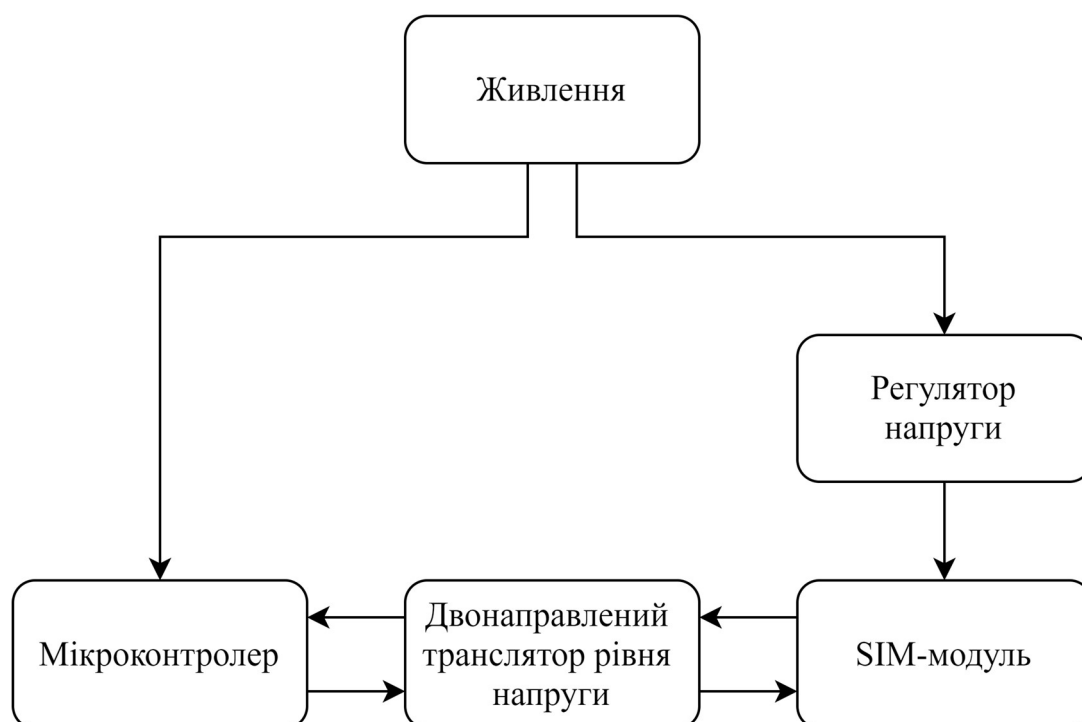


Рисунок 2.1 — Структурна схема

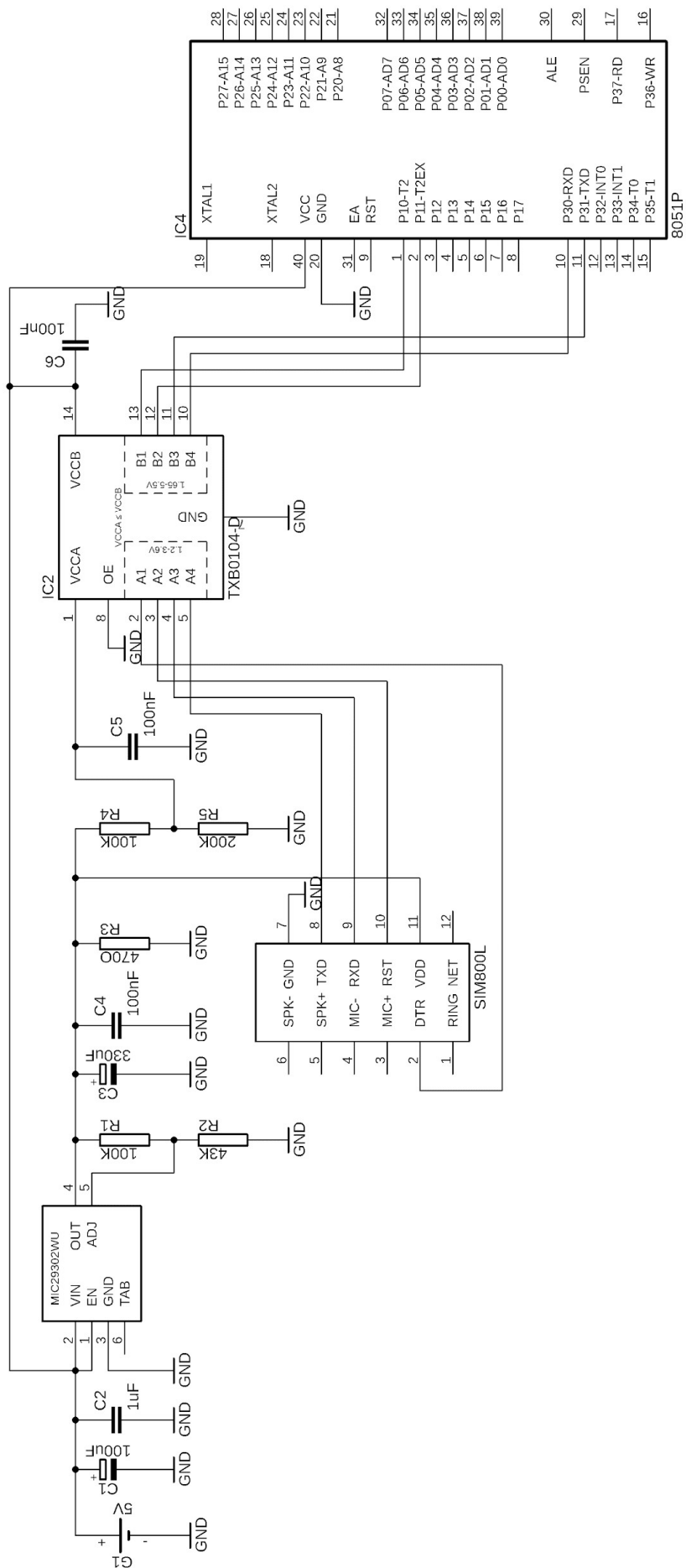
## 2.2 Розробка схеми електричної принципової.

Для роботи трекеру потрібно подавати напругу 5 В на клемний блок (X1), рекомендується використовувати батарею з напругою 5 В та струмом 2А. Живлення підключається до мікроконтролера до піну VCC, до регулятора напруги (пін VIN) та до двонаправленого транслятора напруги (пін VCCB). Регулятор напруги, зменшує напругу до рекомендованих для SIM-модуля 4.2 В, з'єднаний піном OUT з піном VDD SIM-модуля.

Конденсатори C1-C6 виконують роль блокувальних, тобто встановлені таким чином, що вони шунтують живлення мікросхеми і діють як місцеве джерело живлення. Конденсатори C1 та C3 беруть на себе фільтрацію низькочастотних складових, а C2 і C4-C6 - високочастотні.

За допомогою дільника напруги R1, R2 регулюють вихідну напругу лінійного регулятора напруги, подаючи на пін ADJ потрібну напругу. Дільник напруги R3, R4 регулює напругу на вході VCCA двонаправленого транслятора напруги, зменшуючи її з 4.2 В до 2.8 В.

До портів P10, P11, P30.RXD, P31.TXD мікроконтролера через двонаправлений транслятор рівня напруги підключені відповідно порти RST, DTR, TXD, RXD SIM-модуля. Двонаправлений транслятор використовується для узгодження напруг на портах SIM-модуля (2.8 В) та мікроконтролера (5 В).



Ри  
сун  
ок



### 2.3 Вибір елементів в схемі.

**Мікроконтролер 80C51.** На рис. 2.3 зображені розміри мікроконтролера 80C51. Основні параметри мікросхеми перераховані в табл. 2.1

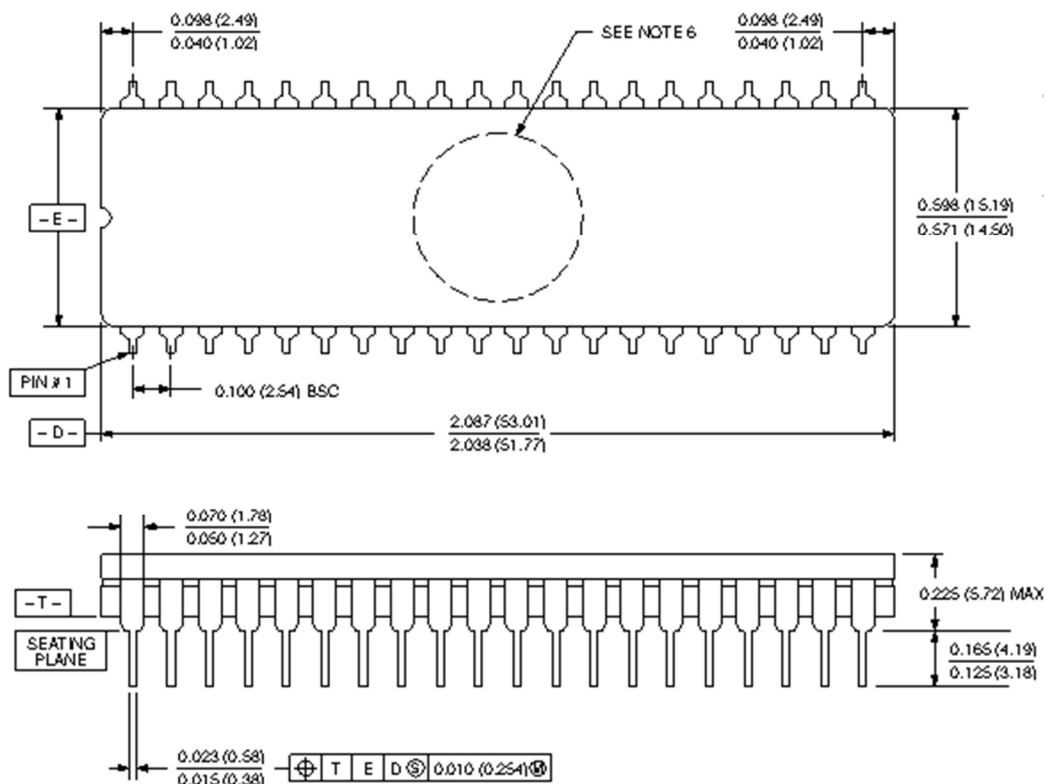


Рисунок 2.3 — Мікроконтролер 80C51

Таблиця 2.1 – основні параметри мікросхеми

Ширина шини даних	8 біт
Максимальна тактова частота	24 МГц
Розмір програмної пам'яті	8 кБайт
Розмір ОЗУ даних	256 Байт
Напруга живлення	5 В
Робоча температура	від -40 до 85 °С
Потужність	1.5 Вт

**Регулятор напруги MIC29302WU.** Високоточний, потужний, з низьким падінням напруги регульований стабілізатор напруги компанії "Micrel Semiconductor". Даний регулятор має типовим значенням падіння напруги (350-

425 mV при повному навантаженні) і дуже малим струмом замикання на "землю". Повністю захищений від над-струмових збоїв, зміни полярності, в тому числі і полярності вхідного сигналу, високотемпературних операцій, позитивних і негативних перехідних сплесків напруги. На рис. 2.4 зображено MIC29302WU. Розміри мікросхеми перераховані в табл. 2.2, а основні параметри – в табл. 2.3.

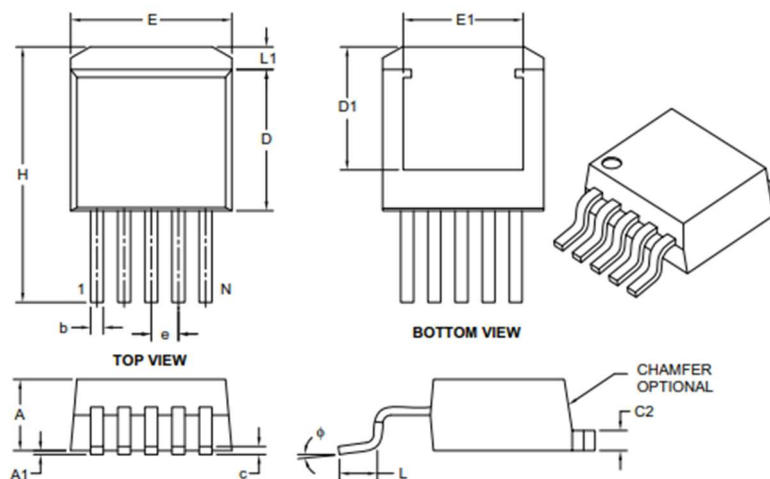


Рисунок 2.4 — Регулятор напруги MIC29302WU

Таблиця 2.2 — розміри регулятора напруги

	MIN	NOM	MAX		MIN	NOM	MAX
N	5			D1	6,86	-	-
e	1,7			c	0,35	-	0,74
A	4,05	-	4,83	C2	1,14	-	1,65
A1	0	-	0,25	b	0,50	-	0,99
E	9,65	-	10,67	L	1,73	-	2,79
E1	6,22	-	-	L1	-	-	1,70
D	8,38	-	9,65	φ	0°	-	8°
H	13,94	-	15,87			-	

Таблиця 2.3 — основні параметри регулятора напруги

Напруга - вхід (макс.)	26 В
Напруга - вихід (мінімум / фіксований)	1.25 В

Напруга - вихід (макс.)	25 В
Робоча температура	-40°C ~ 125°C
Поточний - вихідний	3А
Падіння потужності	0.37 В

**Двонаправлений транслятор рівня напруги TXB0104.** TXB0104 від компанії Texas Instruments є 4-бітний двонаправлений неінвертуючий перетворювач логічних рівнів з автоматичним визначенням напрямку передачі даних.

Використовує дві конфігуруються лінії живлення. Порт А мікросхеми пов'язаний з виведенням VCCA, на який можна подавати будь-яку напругу в діапазоні від 1.2В до 3.6В. Порт В пов'язаний з виведенням VCCB, на який можна подавати будь-яку напругу в діапазоні від 1.65В до 5.5В. Завдяки цьому, мікросхему можна використовувати в якості універсального низьковольтного двонаправленого перетворювача між вузлами з будь-яким напругою - 1.2В, 1.5В, 1.8В, 2.5В, 3.3В і 5В. Напруга на виводі VCCA не повинно перевищувати напругу на VCCB. TXB0104 зображено на рис. 2.5, основні параметри знаходяться в табл. 2.4.

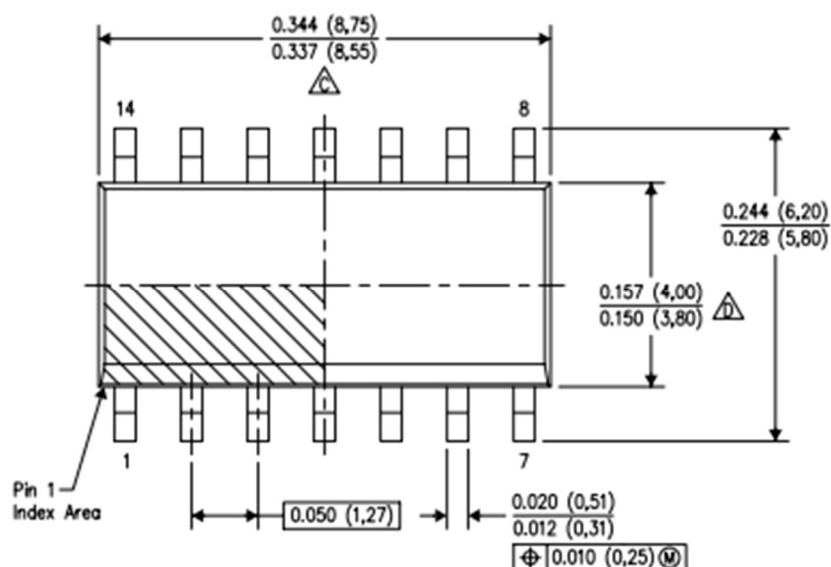


Рисунок 2.5 — Двонаправлений транслятор рівня напруги

Таблиця 2.4 — основні параметри транслятора рівня напруги

Тип каналу	Двонаправлений
Напруга - VCCA	1.2В ~ 3.6В
Напруга - VCCB	1.65В ~ 5.5В
Швидкість передачі даних	100Mbps
Робоча температура	-40°C ~ 85°C
Споживання електроенергії	5μA

**SIM800L - GSM/GPRS модуль.** Модуль SIM800L підтримує чотирисмугову мережу GSM / GPRS, доступну для віддаленої передачі даних GPRS та SMS-повідомлень. SIM800L зв'язується з мікроконтролером через порт UART, підтримує команду, що включає команди 3GPP TS 27.007, 27.005 та посилені ATC команди ATC. Він також має вбудований переклад рівня, тому він може працювати з мікроконтролером вищої напруги, ніж за замовчуванням 2,8 В. Крім того, плата також підтримує техніку A-GPS, яку називають мобільним позиціонуванням і отримує позицію за допомогою мобільної мережі. SIM800L зображено на рис. 2.6, основні параметри перераховані в табл. 2.5.

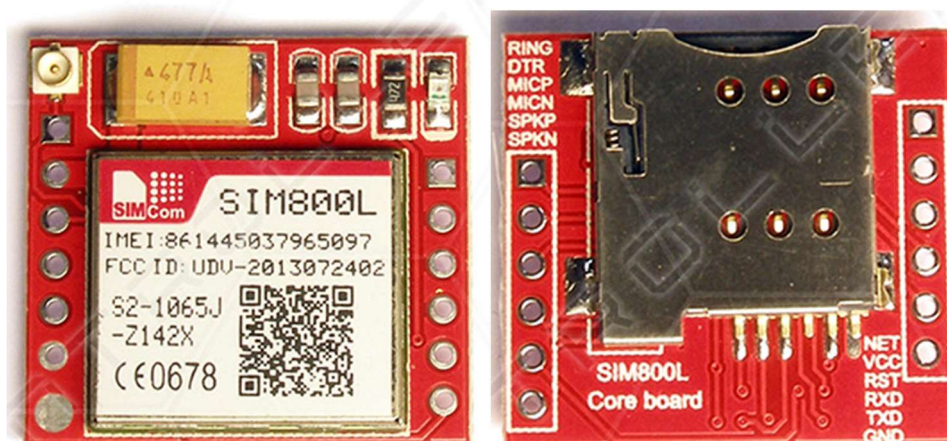


Рисунок 2.6 — SIM800L - GSM/GPRS модуль

Таблиця 2.5 — основні параметри SIM800L

Напруга живлення	3.7 В – 4.2 В
Ток режиму очікування	0.7 мА
Піковий струм	2А

Падіння потужності	0.35 В
Середній струм	0.45 А
Розміри	25 x 24 x 4 мм

### Конденсатори електролітичні С1 (EEE-FC1A101P), С3 (EEE-1AA331P).

На рис. 2.7 зображено конденсатор EEE-FC1A101P та його розміри. На рис. 2.8 зображено конденсатор EEE-1AA331P. та його розміри. У табл. 2.6 перераховані основні параметри елементів.

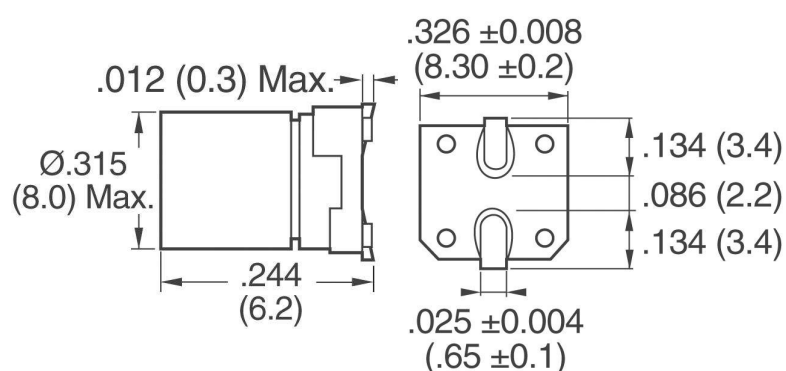


Рисунок 2.7 — Конденсатор С1 EEE-FC1A101P

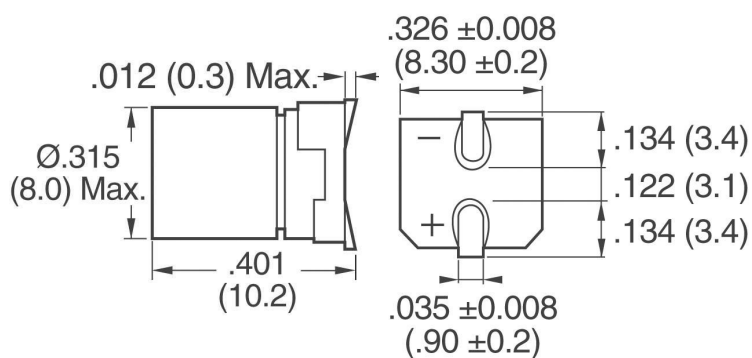


Рисунок 2.8 — Конденсатор С3 EEE-1AA331P

Таблиця 2.6 — основні параметри елементів С1, С3

	C1	C3
Ємність	100 $\mu$ F	330 $\mu$ F
Відхилення ємності	$\pm 20\%$	$\pm 20\%$
Температура навколишнього середовища	-40°C ~ 85°C	-40°C ~ 85°C

Мінімальне напрацювання	1000 годин	2000 годин
Номінальна напруга	10В	10В

**Конденсатори C2 (GRM155C81A105KA12D), C4, C5, C6 (GRM155R71A104JA01D).** На рис. 2.9 зображено конденсатори GRM155C81A105KA12D/GRM155R71A104JA01D, їх розміри знаходяться в табл. 2.7, в табл. 2.8 перераховані основні параметри елементів.

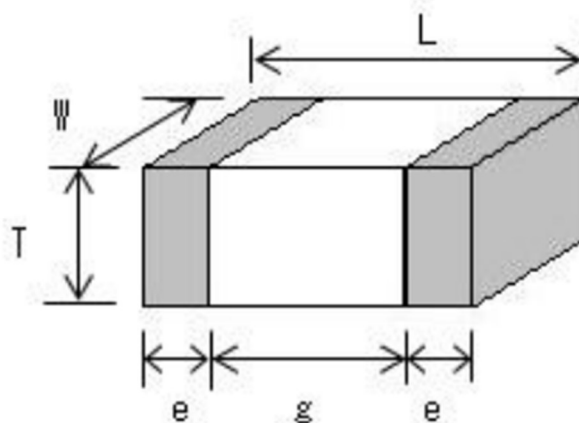


Рисунок 2.9 — Конденсатори C2, C4, C5, C6

Таблиця 2.7 — розміри конденсаторів C2, C4, C5, C6

L	W	T	e	g
1.0±0.05	0.5±0.05	0.5±0.05	від 0.15 до 0.35	0.3 мін.

Таблиця 2.8 — основні параметри елементів

	C2	C4, C5, C6
Ємність	1μF	0.1μF
Відхилення ємності	±10%	±10%
Температура навколишнього середовища	-55°C ~ 105°C	-55°C ~ 125°C
Номінальна напруга	10В	10В

**Резистори R1/R4, R2, R3, R5 (RC0402FR-07100KL, ERA-2AEB433X, ERA-2ARB471X, RC0402FR-07200KL).** На рис. 2.10 зображено резистор, в табл. 2.9 перераховані розміри елементів, в табл. 2.10 – основні параметри елементів.

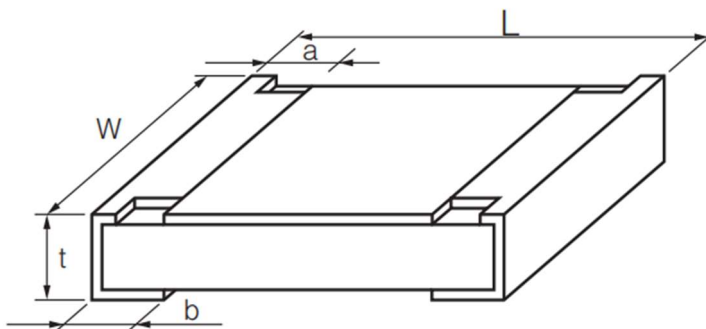


Рисунок 2.10 — Резистор RC0402FR-07100KL / ERA-2AEB433X / ERA-2ARB471X / RC0402FR-07200KL

Таблиця 2.9 — розміри резисторів

L	W	t	a	b
1.00±0.10	0.50±0.10	0.35±0.05	0.15±0.10	0.25±0.10

Таблиця 2.10 — основні параметри елементів

Опір RC0402FR-07100KL	100 кОм
Опір ERA-2AEB433X	43 кОм
Опір ERA-2ARB471X	470 Ом
Опір RC0402FR-07200KL	200 кОм
Відхилення ємності	±1%
Температура навколишнього середовища	-55°C ~ 155°C
Потужність	0.063W

#### 2.4 Розробка конструкції друкованої плати.

Розробка конструкції друкованої плати має такі основні стадії:

- вибір і обґрунтування типу друкованої плати;
- вибір і обґрунтування класу точності друкованої плати;
- вибір матеріалу, габаритних розмірів і конфігурації друкованої плати;
- попереднє розміщення навісних елементів;

- трасування провідників і розміщення елементів;
- розробка конструкторської документації друкованої плати.

У даному пристрої використовується одностороння друкована плата.

Відповідно до ГОСТ 23751-86 [9] друковані плати за точністю виконання елементів діляться на п'ять класів точності.

В даному проекті обраний другий клас точності, так як плати другого класу точності більш надійні і мають меншу вартість, ніж класи вищих рівнів. Плати другого класу точності мають наступні конструктивні параметри:

- мінімальне значення ширини провідника  $t = 0.45$  мм;
- мінімальне значення відстані між провідниками  $S = 0.45$  мм;
- гарантований поясок  $b = 0.2$  мм;
- граничне відхилення діаметра отвору  $\Delta d = \pm 0.15$  мм;
- граничне відхилення ширини друкованого провідника  $\Delta t = \pm 0.1$  мм;
- допуск на розташування осей отворів  $T_d = 0.15$  мм;
- допуск на розташування центрів контактних площадок  $T_D = 0.25$  мм.

В якості матеріалу для виготовлення друкованої плати використовується склотекстоліт, облицьований мідної оксидованою фольгою (СФ-1-35) [10]. Товщина фольги - 50 мкм, ширина доріжки - 0.508 мм. Товщина плати - 1.5 мм. Робочий діапазон температур: від  $-60$  ° С до  $+85$  ° С. Габаритні розміри друкованої плати - 50.8 мм на 73.66 мм.

Діаметр отвору під висновок вибирають з умови отримання зазору між висновком і стінкою отвори, що забезпечує капілярний проникнення припою в процесі пайки [11].

Мінімальний діаметр контактної площадки  $D$  навколо монтажного отвору з відомим діаметром  $d$  визначається за формулою

$$D = d + \Delta d_{\text{в.о.}} + 2b_{\text{н}} + \Delta t_{\text{в.про}} + 2\Delta d_{\text{тр}} + (T_d^2 + T_D^2 + \Delta t_{\text{н.о.}}^2)^{0.5}$$

де  $\Delta d_{\text{в.о.}}$  - верхнє граничне відхилення діаметра отвору - 0.15 мм;

$b_{\text{н}}$  - гарантійний поясок на зовнішньому шарі - 0.2 мм;

$t_{\text{в.про}}$  та  $t_{\text{н.о}}$  - верхнє і нижнє граничні відхилення ширини провідника - 0.1

мм;



$\Delta d_{\text{тр}}$ - значення підщавлювання діелектрика (для двосторонніх плат = 0.03);

$T_d$  - допуск на розташування осей отворів для відповідного класу точності і розмірів плати - 0.15 мм;

$T_D$  - допуск на розташування контактних площадок для відповідного класу точності і розмірів плати - 0.25 мм.

Діаметри висновків мікроконтролера = 0.58 мм

$$d_{\text{отв}} \approx 0.8 \text{ мм}$$

$$D_1 = 0.8 + 0.15 + 0.4 + 0.1 + 0.06 + (\sqrt{0.15^2 + 0.25^2 + 0.1^2}) = 1.8 \text{ мм}$$

Діаметри висновків SIM-модуля = 0.64 мм

$$d_{\text{отв}} \approx 0.8 \text{ мм}$$

$$D_1 = 0.8 + 0.15 + 0.4 + 0.1 + 0.06 + (\sqrt{0.15^2 + 0.25^2 + 0.1^2}) = 1.8 \text{ мм}$$

Діаметри отворів живлення

$$d_{\text{отв}} \approx 1.11 \text{ мм}$$




$$D_1 = 1.11 + 0.15 + 0.4 + 0.1 + 0.06 + (\sqrt{0.15^2 + 0.25^2 + 0.1^2}) = 2.13 \text{ мм}$$

Правила виконання креслень друкованої плати як деталі встановлені ГОСТ 2.417-91 [12]. Розміри на кресленні друкованої плати вказані за допомогою координатної сітки в прямокутній системі координат. Крок сітки 1,27 мм. Координатна сітка нанесена на частину поверхні друкованої плати. За початок відліку прийнятий лівий нижній кут друкованої плати [13].

Діаметр отвору, його умовний знак, діаметр контактної площадки, наявність металізації, кількість отворів об'єднані в табл. 2.11. Розміри та кількість контактних площадок перераховані в табл. 2.12.

Таблиця 2.11 — параметри монтажних отворів.

Умовне позначення контактних площадок	Діаметр отворів, мм	Наявність металізації в отворі	Кількість отворів	Мінімальний діаметр контактної площадки, мм
---------------------------------------	---------------------	--------------------------------	-------------------	---

	0.8	без металізації	52	1.8
	1.11	без металізації	2	3.15
	2.54	без металізації	4	-

Таблиця 2.12 — параметри контактних площадок

Умовне позначення	Кількість	Розмір, мм
	18	0.55*0.5
	5	1.04*4.04
	14	1.5*0.55

Складальне креслення пристрою показано на рис. 2.11, друкована плата - рис. 2.12.

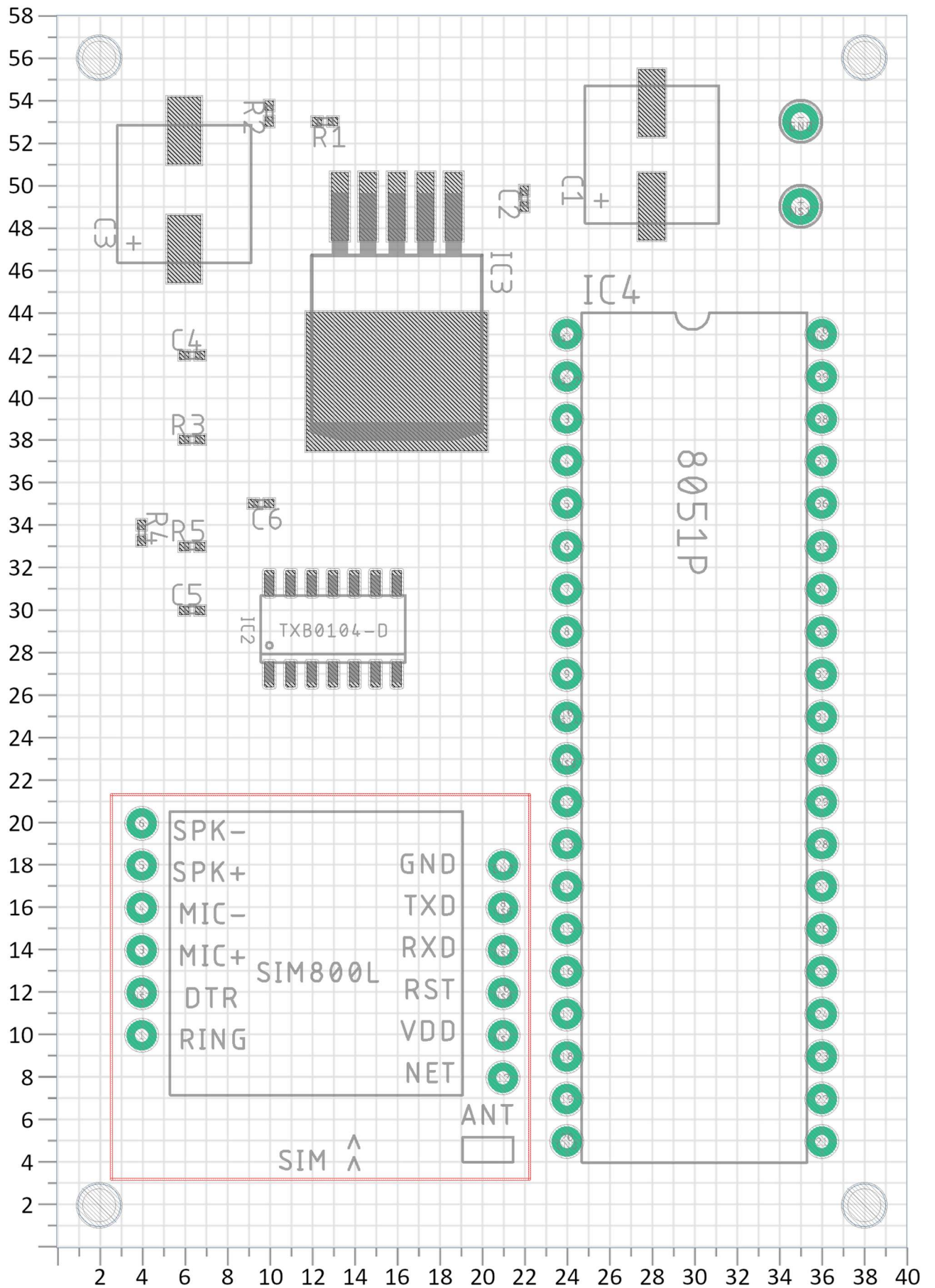


Рисунок 2.11 — Складальне креслення пристрою

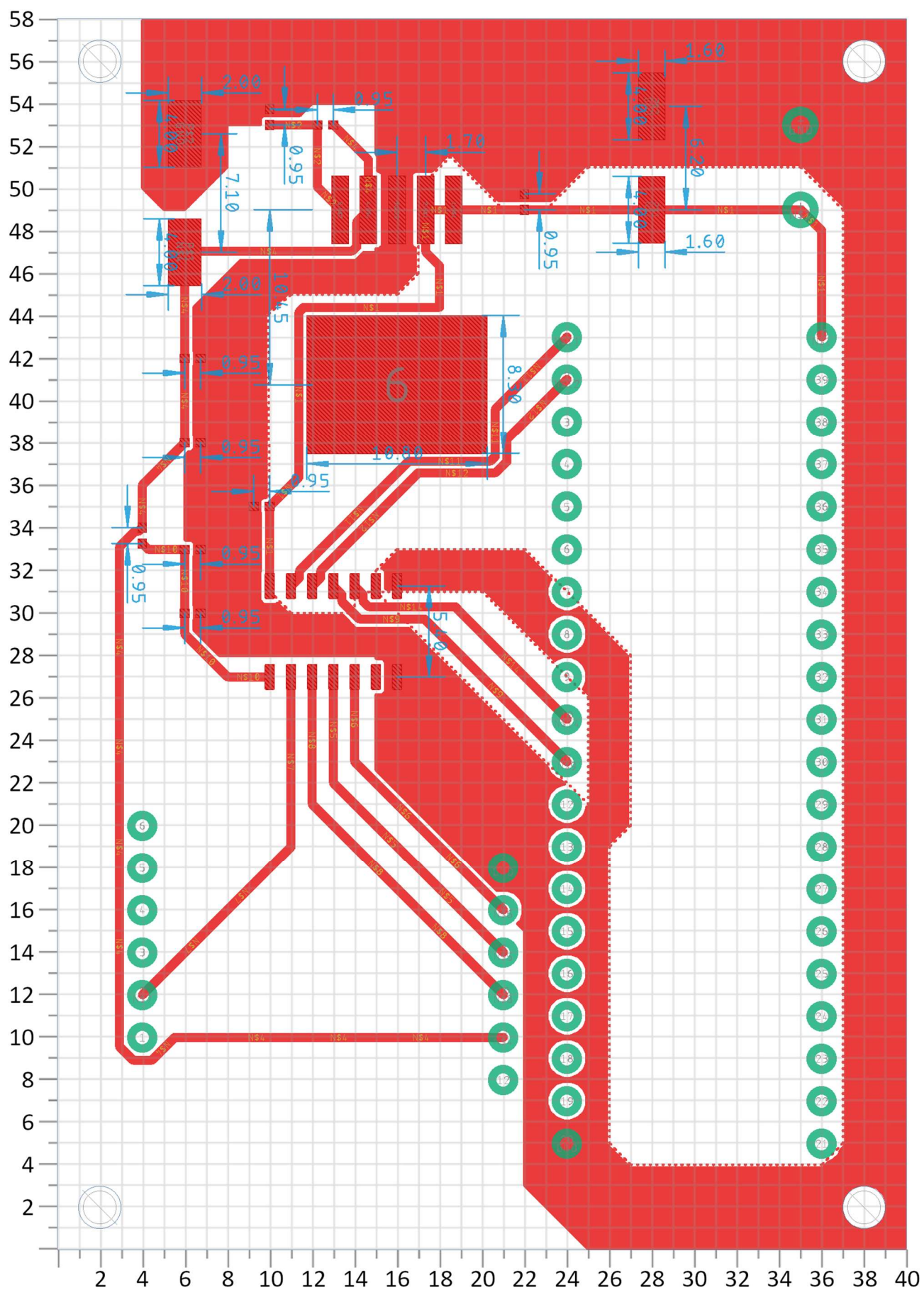


Рисунок 2.12 — Друкована плата

## 2.5 Вибір умов охолодження і розрахунок теплового режиму.

У процесі роботи електронного пристрою температура нагрівання його елементів не повинна перевищувати припустимих технічними умовами значень.

При конструюванні необхідні температурні умови досягаються застосуванням відповідних умов охолодження і раціональним компонованням.

### Попередній вибір системи охолодження

Попередній вибір системи охолодження проводиться за допомогою графіків, що характеризують область доцільного застосування різних способів охолодження.

Оцінка виробляється на підставі попередніх даних за величиною теплового потоку, що приходить на одиницю площі теплообміну :

$$p = \frac{P * k_p}{S_n}$$

де  $P = \sum p_i$  – сумарна розсіювальна потужність електронного пристрою, де  $p_i$  - потужність кожного елемента окремо.

$k_p$  – коефіцієнт, що враховує тиск повітря ( $k_p = 1$ );

$S_n = 2[l_1 l_2 + (l_1 + l_2) l_3 k_3]$  – поверхня теплообміну, обумовлена геометричними розмірами корпусу електронного пристрою, тобто довжиною  $l_1$ , шириною  $l_2$  основи пристрою і його висотою  $l_3$ , а також коефіцієнтом заповнення об'єму  $k_3 = \frac{V_{ап}}{V_{об}}$ , де  $V_{ап}$  - об'єм апаратури,  $V_{об}$  - об'єм об'єкта.

Корпус проектованого пристрою прямокутної форми з розмірами: 50.8\*73.66\*10.2 мм

$$P = 1.5 + 0.37 * 0.45 + 0.35 * 0.45 + 5 * 5 * 10^{-6} + 0.063 * 5 = 2.139 \text{ Вт}$$

$$\begin{aligned} V_{ап} &= 5 * 0.5 * 1 * 0.35 + 4 * 0.5 * 1 * 0.5 + 8.3 * 8.3 * 10.2 + 8.3 * 8.3 * 6.2 + \\ &+ 25 * 24 * 4 + 6.2 * 8.75 * 1.75 + 15 * 10 * 4.5 + 15.2 * 53 * 5.7 \\ &= 8893.53 \text{ (мм}^2\text{)} \end{aligned}$$

$$V_{об} = 50.8 * 73.66 * 10.2 = 38167.66 \text{ (мм}^2\text{)}$$

$$k_3 = \frac{8893.53}{38167.66} = 0.233$$

$$S_n = 2 * [50.8 * 73.66 + (50.8 + 73.66) * 10.2 * 0.233] = 8075.47 \text{ мм}^2 \\ = 0.008 \text{ м}^2$$

$$p = \frac{2.139 * 1}{0.008} = 264.88$$

$$\lg p = 2.42$$

Другим вхідним параметром є величина мінімально припустимого перегріву елементів пристрою  $\Delta T_{i \min} = T_{i \min} - T_c$ ,

де  $T_{i \min}$  – припустима температура корпусу найменш теплостійкого елемента;

$T_c$  – температура навколишнього середовища.

$$\Delta T_{i \min} = 85 - 45 = 40$$

на рис. 2.13 наведено області доцільного застосування різних способів охолодження в системі координат  $\Delta T_c, \lg p$  з цих графіків випливає, що для охолодження пристрою достатньо природного повітряного охолодження.

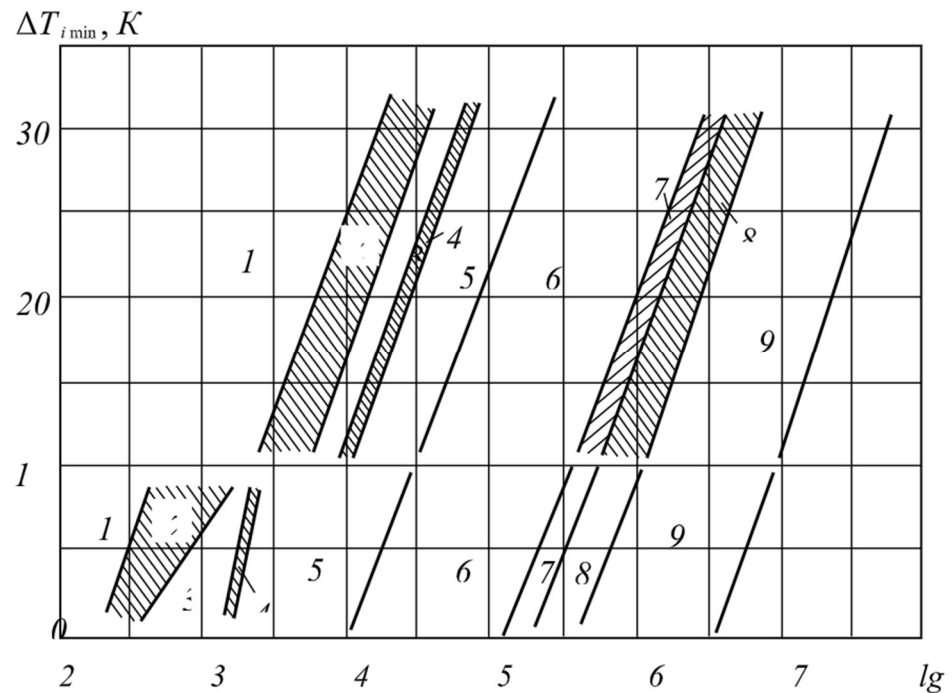


Рисунок 2.13 — Области доцільного застосування різних способів охолодження: 1 – природне повітряне; 2 – природне і примусове повітряне; 3 –

примусове повітряне; 4 – примусове повітряне і рідинне; 5 – примусове рідинне; 6 - примусове рідинне і природне випарне; 7 - примусове рідинне, примусове і природне випарне; 8 - природне і примусове випарне; 9 - примусове випарне.

### Розрахунок теплового режиму блоку в корпусі

Визначальними параметрами для розрахунку є питомі потужності розсіювання блоку пристрою в цілому  $p_k$  і нагрітої зони  $p_3$  відповідно:

$$p_k = P_0/S_k,$$

$$p_3 = P_0/S_3,$$

де  $P_0$  – потужність, що розсіюється пристроєм,;  $S_k$  - площа поверхні корпусу пристрою;  $S_3$  - умовна поверхня нагрітої зони.

У свою чергу площа поверхні корпусу пристрою й умовна поверхня нагрітої зони визначаються за формулами:

$$S_k = 2[l_1 l_2 + (l_1 + l_2)l_3],$$

$$S_3 = 2[l_1 l_2 + (l_1 + l_2)l_3 k_3],$$

$$S_k = 2 * [50.8 * 73.66 + (50.8 + 73.66) * 10.2] = 10022.84 \text{ мм}^2 = 0.01 \text{ м}^2$$

$$S_3 = 0.008 \text{ м}^2$$

$$p_k = \frac{2.139}{0.01} = 213.42 \text{ Вт/м}^2$$

$$p_3 = \frac{2.139}{0.008} = 264.88 \text{ Вт/м}^2$$

У загальному випадку перегрів корпусу герметичного електронного пристрою, що працює в нормальних кліматичних умовах, щодо навколишнього середовища визначається залежністю

$$\theta_1 = 0.1472 * p_k - 0.2962 * 10^{-3} * p_k^2 + 0.3127 * 10^{-6} * p_k^3$$

$$\begin{aligned} \theta_1 &= 0.1472 * 213.42 - 0.2962 * 10^{-3} * 213.42^2 + 0.3127 * 10^{-6} * 213.42^3 = \\ &= 20.96 \text{ }^\circ\text{C} \end{aligned}$$

Перегрів нагрітої зони визначається аналогічною залежністю:

$$\theta_2 = 0.1390 * p_3 - 0.1223 * 10^{-3} * p_3^2 + 0.0698 * 10^{-6} * p_3^3$$

$$\begin{aligned} \theta_2 &= 0.1390 * 264.88 - 0.1223 * 10^{-3} * 264.88^2 + 0.0698 * 10^{-6} * 264.88^3 = \\ &= 29.53 \text{ }^\circ\text{C} \end{aligned}$$

За отриманими даними визначається перегрів повітря в блоці (пристрої)

$$\theta_{\text{пов}} = 0.5 * (\theta_1 + \theta_2) = 25.25 \text{ } ^\circ\text{C}$$

Температурний режим окремих теплонавантажених елементів залежить від питомої потужності елемента і питомої потужності нагрітої зони.

### 80c51

Потужність елемента  $P_{\text{ел}} = 1.5 \text{ Вт}$

Площа поверхні елемента  $S_{\text{ел}} = 0.0048 \text{ м}^2$

Питома розсіювальна потужність  $p_{\text{ел}} = \frac{P_{\text{ел}}}{S_{\text{ел}}} = 312.5 \frac{\text{Вт}}{\text{м}^2}$ .

Перегрів поверхні елемента

$$\theta_{\text{ел}} = \theta_2 * (0.75 + 0.25 * \frac{p_{\text{ел}}}{p_3}) = 30.86 \text{ } ^\circ\text{C}.$$

Перегрів навколишнього середовища

$$\theta_{\text{ср}} = \theta_{\text{пов}} * (0.75 + 0.25 * \frac{p_{\text{ел}}}{p_3}) = 26.38 \text{ } ^\circ\text{C}.$$

Температура поверхні елемента

$$T_{\text{ел}} = \theta_{\text{ел}} + T_c = 26.84 + 45 = 75.86 \text{ } ^\circ\text{C}.$$

Температура навколишнього середовища

$$T_{\text{ср}} = \theta_{\text{ср}} + T_c = 23.34 + 45 = 71.38 \text{ } ^\circ\text{C}.$$

Так як температура поверхні елемента менше максимальної робочої температури, то елемент не перегрівається і працює нормально.

### MIC29302WU

Потужність елемента  $P_{\text{ел}} = 0.1665 \text{ Вт}$

Площа поверхні елемента  $S_{\text{ел}} = 0.000567 \text{ м}^2$

Питома розсіювальна потужність  $p_{\text{ел}} = \frac{P_{\text{ел}}}{S_{\text{ел}}} = 293.65 \frac{\text{Вт}}{\text{м}^2}$ .

Перегрів поверхні елемента

$$\theta_{\text{ел}} = \theta_2 * (0.75 + 0.25 * \frac{p_{\text{ел}}}{p_3}) = 30.34 \text{ } ^\circ\text{C}.$$

Перегрів навколишнього середовища

$$\theta_{\text{ср}} = \theta_{\text{пов}} * (0.75 + 0.25 * \frac{p_{\text{ел}}}{p_3}) = 25.93 \text{ } ^\circ\text{C}.$$

Температура поверхні елемента



$$T_{\text{ел}} = \theta_{\text{ел}} + T_c = 26.29 + 45 = 75.34 \text{ }^\circ\text{C}.$$

Температура навколишнього середовища

$$T_{\text{ср}} = \theta_{\text{ср}} + T_c = 22.86 + 45 = 70.93 \text{ }^\circ\text{C}.$$

Так як температура поверхні елемента менше максимальної робочої температури, то елемент не перегрівается і працює нормально.

### **ТХВ0104**

Потужність елемента  $P_{\text{ел}} = 0.000125 \text{ Вт}$

Площа поверхні елемента  $S_{\text{ел}} = 94 * 10^{-6} \text{ м}^2$

Питома розсіювальна потужність  $p_{\text{ел}} = \frac{P_{\text{ел}}}{S_{\text{ел}}} = 1.33 \frac{\text{Вт}}{\text{м}^2}$ .

Перегрів поверхні елемента

$$\theta_{\text{ел}} = \theta_2 * (0.75 + 0.25 * \frac{p_{\text{ел}}}{p_3}) = 22.19 \text{ }^\circ\text{C}.$$

Перегрів навколишнього середовища

$$\theta_{\text{ср}} = \theta_{\text{пов}} * (0.75 + 0.25 * \frac{p_{\text{ел}}}{p_3}) = 18.97 \text{ }^\circ\text{C}.$$

Температура поверхні елемента

$$T_{\text{ел}} = \theta_{\text{ел}} + T_c = 67.19 \text{ }^\circ\text{C}.$$

Температура навколишнього середовища

$$T_{\text{ср}} = \theta_{\text{ср}} + T_c = 63.97 \text{ }^\circ\text{C}.$$

Так як температура поверхні елемента менше максимальної робочої температури, то елемент не перегрівается і працює нормально.

### **SIM800L**

Потужність елемента  $P_{\text{ел}} = 0.1575 \text{ Вт}$

Площа поверхні елемента  $S_{\text{ел}} = 0.0027 \text{ м}^2$

Питома розсіювальна потужність  $p_{\text{ел}} = \frac{P_{\text{ел}}}{S_{\text{ел}}} = 58.33 \frac{\text{Вт}}{\text{м}^2}$ .

Перегрів поверхні елемента

$$\theta_{\text{ел}} = \theta_2 * (0.75 + 0.25 * \frac{p_{\text{ел}}}{p_3}) = 23.78 \text{ }^\circ\text{C}.$$

Перегрів навколишнього середовища

$$\theta_{\text{ср}} = \theta_{\text{пов}} * (0.75 + 0.25 * \frac{p_{\text{ел}}}{p_3}) = 20.33 \text{ }^\circ\text{C}.$$

Температура поверхні елемента

$$T_{\text{ел}} = \theta_{\text{ел}} + T_c = 68.78 \text{ } ^\circ\text{C}.$$

Температура навколишнього середовища

$$T_{\text{ср}} = \theta_{\text{ср}} + T_c = 65.33 \text{ } ^\circ\text{C}.$$

Так як температура поверхні елемента менше максимальної робочої температури, то елемент не перегрівается і працює нормально.

### **Резистори R1, R2, R3, R4, R5**

Потужність елемента  $P_{\text{ел}} = 2.94 * 10^{-5}$  Вт

Площа поверхні елемента  $S_{\text{ел}} = 1.75 * 10^{-7}$  м<sup>2</sup>

Питома розсіювальна потужність  $p_{\text{ел}} = \frac{P_{\text{ел}}}{S_{\text{ел}}} = 231.49 \frac{\text{Вт}}{\text{м}^2}$ .

Перегрів поверхні елемента

$$\theta_{\text{ел}} = \theta_2 * (0.75 + 0.25 * \frac{p_{\text{ел}}}{p_3}) = 28.6 \text{ } ^\circ\text{C}.$$

Перегрів навколишнього середовища

$$\theta_{\text{ср}} = \theta_{\text{пов}} * (0.75 + 0.25 * \frac{p_{\text{ел}}}{p_3}) = 24.45 \text{ } ^\circ\text{C}.$$

Температура поверхні елемента

$$T_{\text{ел}} = \theta_{\text{ел}} + T_c = 73.6 \text{ } ^\circ\text{C}.$$

Температура навколишнього середовища

$$T_{\text{ср}} = \theta_{\text{ср}} + T_c = 69.45 \text{ } ^\circ\text{C}.$$

Так як температура поверхні елемента менше максимальної робочої температури, то елемент не перегрівается і працює нормально.

## 2.6 Розрахунок надійності пристрою.

Визначення надійності пристрою здійснюється за методикою розрахунку за раптовими експлуатаційними відмовленнями за відомими показниками надійності елементів пристрою з урахуванням таких допущень: відмовлення елементів статистично незалежне і відмовлення будь-якого елемента призводить до відмовлення всього пристрою.

Інтенсивність відмовлень пристрою визначається за формулою:

$$\lambda_c = k_\lambda \cdot \sum_{i=1}^n a_i \cdot \lambda_{0i},$$

де  $\lambda_{0i}$  – інтенсивність відмовлення  $i$ -того елемента.

$k_\lambda = k_{\lambda 1} * k_{\lambda 2} * k_{\lambda 3}$  – поправочний коефіцієнт, що враховує умови експлуатації пристрою:  $k_{\lambda 1}$ - вплив механічних факторів;  $k_{\lambda 2}$  - вплив кліматичних факторів;  $k_{\lambda 3}$  - умови роботи при зниженому атмосферному тиску.

$a_i$  – поправочний коефіцієнт, що враховує вплив температури навколишнього середовища й електричне навантаження приладу через коефіцієнт навантаження  $k_H$ ;

$k_H$  – коефіцієнт навантаження, що представляє собою відношення робочого навантаження встановленого за визначальним параметром.

Середній наробіток на відмовлення всього пристрою визначається за формулою:

$$T_c = \frac{1}{\lambda_c}$$

### Резистори постійні металоплівкові

Інтенсивність відмов  $\lambda_0 = 0.4 * 10^{-6} \frac{1}{\text{год}}$

Коефіцієнт навантаження  $k_H = 0.5$

Поправочний коефіцієнт  $a = 1.07$

Кількість елементів – 5.

### Конденсатори оксидні

Інтенсивність відмов  $\lambda_0 = 0.8 * 10^{-6} \frac{1}{\text{год}}$

Коефіцієнт навантаження  $k_H = 0.7$

Поправочний коефіцієнт  $a = 7$

Кількість елементів – 2.

### **Конденсатори керамічні**

Інтенсивність відмов  $\lambda_0 = 0.7 * 10^{-6} \frac{1}{\text{год}}$

Коефіцієнт навантаження  $k_n = 0.7$

Поправочний коефіцієнт  $a = 2,8$

Кількість елементів – 4.

### **Мікросхеми інтегральні**

Інтенсивність відмов  $\lambda_0 = 1.5 * 10^{-6} \frac{1}{\text{год}}$

Кількість елементів – 4.

### **Плата друкована**

Інтенсивність відмов  $\lambda_0 = 0.1 * 10^{-6} \frac{1}{\text{год}}$ .

Кількість елементів – 1.

### **Пайка сполучна**

Інтенсивність відмов  $\lambda_0 = 0.04 * 10^{-6} \frac{1}{\text{год}}$ .

Кількість елементів – 94.

Інтенсивність відмови пристрою дорівнює:

$$\lambda_c = 1.46 * 1 * 1 * 10^{-6} * (0.4 * 1.07 * 5 + 0.8 * 7 * 2 + 0.7 * 2.8 * 4 + 1.5 * 4 + 0.1 + 0.04 * 94) = 4.5318 * 10^{-5} \frac{1}{\text{год}}$$

$$T_c = \frac{1}{\lambda_c} = \frac{1}{4.4069 * 10^{-5}} = 22066.09 \text{ ч} \approx 2.52 \text{ роки.}$$

## РОЗДІЛ 3. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Програмна частина системи складається з чотирьох елементів:

- програма мікроконтролера;
- http сервер;
- база даних;
- користувальницький інтерфейс.

Структурна схема програмного забезпечення системи зображена на рис.

3.1.

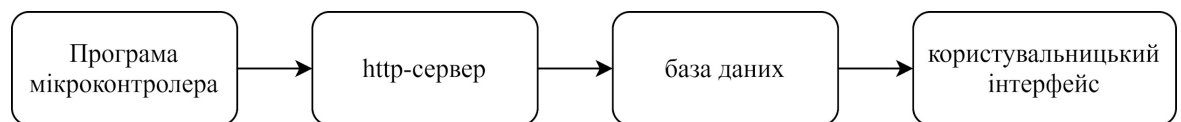


Рисунок 2.1 — Структурна схема програмного забезпечення

### 3.1 Розробка програмного забезпечення мікроконтролера.

Комп'ютерна програма - це список інструкцій, які дозволяють комп'ютеру виконувати певні завдання. Комп'ютерні програми можуть бути написані на мовах високого і низького рівня, в залежності від завдання і використовуваного обладнання.

Мови високого рівня написані у формі, близькій до нашого людської мови, що дозволяє програмісту просто зосередитися на розв'язуванні проблеми. Ніяких особливих знань про обладнання не потрібно, оскільки мови високого рівня створюють програми, які переносяться і не прив'язані до конкретного комп'ютера або мікрочіпу. Ці дружні програмісту мови називаються «високорівневими», оскільки вони далекі від інструкцій машинного коду, зрозумілих комп'ютеру. Приклади включають в себе: C ++, Java, Pascal, Python, Visual Basic.

Переваги:

- легше змінити, так як він використовує зрозумілі людині оператори;

- легше/швидше писати код, так як він використовує зрозумілі людині оператори;
- легше налагоджувати під час розробки;
- код призначений для роботи не тільки на одному типі машини.

Мови низького рівня використовуються для написання програм, що відносяться до конкретної архітектури і апаратного забезпечення комп'ютера певного типу. Вони ближче до рідної мови комп'ютера (бінарного), що ускладнює їх розуміння програмістами. Приклади мови низького рівня: мова асемблера, машинний код.

Мало хто з програмістів пишуть програми на асемблері низького рівня, але він все ще використовується для розробки коду для спеціального обладнання, такого як драйвери пристроїв. Його легко відрізнити від мови високого рівня, так як він містить мало відомих людських слів, але багато мнемонічного коду.

Переваги:

- може використовувати спеціальне обладнання або спеціальні машинно-залежні інструкції (наприклад, на конкретному чіпі);
- перекладена програма вимагає менше пам'яті;
- напишіть код, який може бути виконаний швидше;
- повний контроль над кодом;
- може працювати безпосередньо на місцях пам'яті.

В нашому випадку мікроконтролер має достатньо пам'яті для використання мови “високого” рівня. Розробка буде виконуватися на мові C у середовищі розробки Keil uVision. Keil uVision дозволяє працювати з проектами будь-якого ступеня складності, починаючи з введення і редагування вихідних текстів і закінчуючи внутрішньосхемною налагодженням коду і програмуванням ПЗУ мікроконтролера. Від розробника прихована велика частина другорядних функцій, що сильно розвантажує інтерфейс і робить управління інтуїтивно зрозумілим. Однак при зростанні складності реалізованих завдань, завжди можна задіяти весь потенціал модулів, що функціонують під управлінням єдиної оболонки. Серед основних програмних засобів Keil uVision можна відзначити.

1. Базу даних мікроконтролерів, що містить докладну інформацію про всі підтримувані пристрої. Тут зберігаються їх конфігураційні дані і посилання на джерела інформації з додатковими технічними описами. При додаванні нового пристрою в проект всі його унікальні опції встановлюються автоматично.

2. Менеджер проектів, котрий служить для об'єднання окремих текстів програмних модулів і файлів в групи, оброблювані за єдиними правилами. Подібне угруповання дозволяє набагато краще орієнтуватися серед безлічі файлів.

3. Вбудований редактор, який полегшує роботу з вихідним текстом за рахунок використання багатовіконного інтерфейсу, виділення синтаксичних елементів шрифтом і кольором. Існує опція настройки відповідно до смаків розробника. Редагування залишається доступним і під час налагодження програми, що дозволяє відразу виправляти помилки або відзначати проблемні ділянки коду.

4. Засоби автоматичної компіляції, асемблювання і компонування проекту, які призначені для створення виконуваного (завантажувального) модуля програми. При цьому між файлами автоматично генеруються нові асемблерні і компіляторний зв'язку, які в подальшому дозволяють обробляти тільки ті файли, в яких відбулися зміни або файли, що знаходяться в залежності від змінених. Функція глобальної оптимізації проекту дозволяє досягти найкращого використання регістрів мікроконтролера шляхом неодноразової компіляції вихідного коду. Компілятори uVision працюють з текстами, написаними на C або асемблері для контролерів сімейств ARM, MSC51, C166 і багатьох інших. Крім того можливе використання компіляторів інших виробників.

5. Відладчик-симулятор, відладжує роботу компільованої програми на віртуальній моделі мікропроцесора. Досить вірогідно моделюється робота ядра контролера і його периферійного обладнання: портів введення-виведення, таймерів, контролерів переривань. Для полегшення комплексного налагодження розробленого програмного забезпечення можливе підключення програмних моделей нестандартного обладнання.

Алгоритм роботи показаний на рис. 3.2.

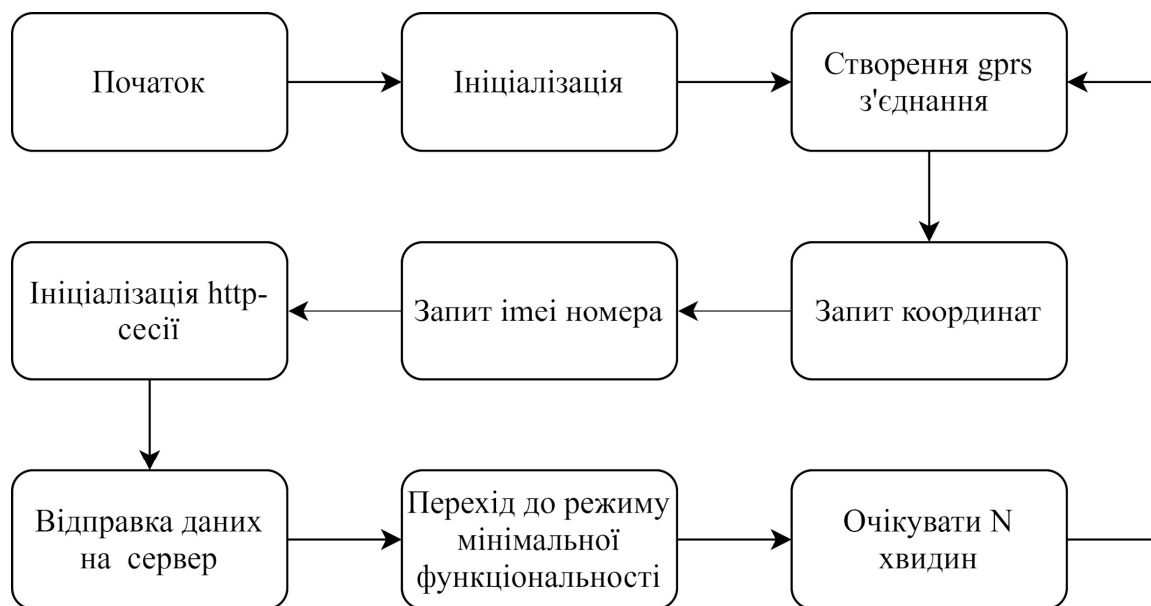


Рисунок 3.2 — Алгоритм роботи пристрою

Нижче наведено повний текст програми з коментарями:

```

#include <REG52.H>
// включає в себе оголошення спеціальних регістрів
#include <stdio.h>
// включає в себе оголошення прототипів для функцій вводу / виводу
*unsigned char * getstr(void) {
// функція для прийому відповіді з серійного порту
unsigned char buf[64];
// створення тимчасового сховища для відповіді
unsigned char *pbuf = buf;
// покажчик на buf
while (1) {
// безкінечний цикл
while (!RI);
// чекати поки RI дорівнює 0 (встановлюється апаратно при
// прийомі байта)

```



```

    RI = 0;
        // скидання прапора переривання приймача
    *pbuf = SBUF;
        // зчитування прийнятого символу з буферного регістра
        // приймача-передавача
    if (*pbuf == 0x0D) {
        // чи є прийнятий символ, символом переведення рядка
        *pbuf = '\0';
            // заміна символу переведення рядка на символ
            // позначення кінця ланцюжка символів
        break;
            // переривання циклу прийому
    }
    pbuf++;
        // збільшення покажчика на 1
    }
    return buf;
        // повернення з функції прийнятий рядок
}

void main (void) {
    // основна функція C. Тут починається виконання програми
    unsigned char *gsmloc;
    int c;
    unsigned char *imeiNumber;
    SCON = 0x50; // 0x50 = 01010000b
        // встановлення режиму роботи приймача-передавача на 8 бітний із
        // змінною швидкістю передачі, та дозволом на прийом
    TMOD = 0x21; // 0x21 = 00100001b

```

*/\* таймер 1: 8-бітний таймер, що автоматично перезавантажується.  
 "TH1" зберігає значення, яке має бути перезавантажено в "TL1" кожен раз  
 по переповненні \*/*

*/\* таймер 0: 16-бітний таймер/лічильник. "TH0" і "TL0" включені  
 послідовно \*/*

*TH1 = 0xFD;*

*// 0xFD відповідає 9600 бод при тактовій частоті мікроконтролера*

*// 11.059MHz*

*TR1 = 1;*

*// запуск таймера 1*

*while (1) {*

*printf ("AT+CFUN=0");*

*// перехід до режиму мінімальної функціональності*

*// SIM-модуля*

*printf ("at+sapbr=3,1,\"contype\", \"gprs\"\\r\\n");*

*// створюємо з'єднання gprs*

*printf ("at+sapbr=3,1,\"apn\", \"internet\"\\r\\n");*

*// установка настроек підключення*

*printf ("at+sapbr=1,1\\r\\n");*

*// встановлюємо GPRS з'єднання*

*printf ("at+cipgsmloc=1,1\\r\\n");*

*// команда запиту координат*

*gsmloc = getstr();*

*// функція для отримання відповіді від SIM-модуля*

*printf ("AT+SAPBR=0,1\\r\\n");*

*// закриваємо GPRS з'єднання*

*printf ("AT+GSN\\r\\n");*

*// команда запиту imei номера*

*imeiNumber = getstr();*

*// функція для отримання відповіді від SIM-модуля*

```

printf ("AT+HTTPINIT\r\n");
    // ініціалізація http сервісу
printf ("AT+HTTPPARA=\"CID\",1\r\n");
    // встановлення параметра CID для сервісу http
printf ("AT+HTTPPARA=\"URL\",\"http://mysite.ua/?all=");
printf (imeiNumber);
printf (gsmloc);
printf ("\r\n");
    // встановлення значення параметра URL
printf ("AT+HTTPACTION=0");
    // здійснити GET-запит до сервера
printf ("AT+CFUN=0");
    // перехід до режиму мінімальної функціональності
    // SIM-модуля
c = 16902; // 20 minutes
    /* таймер інкрементується 11059000/12 = 921583 раз в
секунду при цьому інтервал між перегрузками таймера становить
65536/921583 = 0.071 с. Отже після 20 хвилин таймер повинен
перезавантажитись 1200/0,071 = 16901.4 разів */
while (c > 0) {
    TH0 = 0x00;
    TL0 = 0x00;
        // обнулення таймера
    TR0 = 1;
        // запуск таймера 0
    while (TF0 == 0) {}
        // чекати на перегрузку таймера 0
    TF0 = 0;
        // скидання прапора переривання таймера
    c = c - 1;

```

```

        // зменшення лічильника перезавантажень таймера
    }
    printf ("AT+CFUN=1");
        // зміна режиму роботи SIM-модуля на режим
        // мінімальної функціональності
    }
}

```

### 3.2 Розробка бази даних.

Перш за все потрібно вибрати яку СУБД ми будемо використовувати у нашому проєкті. Незважаючи на те, що всі системи управління базами даних виконують одну і ту ж основну задачу (тобто дають можливість користувачам створювати, редагувати і отримувати доступ до інформації, що зберігається в базах даних), сам процес виконання цього завдання варіюється в широких межах. Крім того, функції і можливості кожної СУБД можуть істотно відрізнятися. Різні СУБД документовані по-різному: більш-менш ретельно. По-різному надається і технічна підтримка. З огляду на те, що у нашій системі немає особливих вимог для бази даних, вибір падає на MySQL.

MySQL - одна з найпопулярніших баз даних для веб-додатків. Фактично, є стандартом de facto для веб-серверів, які працюють під управлінням операційної системи Linux. MySQL - це безкоштовний пакет програм, однак нові версії виходять постійно, розширюючи функціонал і покращуючи безпеку.

Переваги:

- розповсюджується безкоштовно;
- прекрасно документована;
- пропонує багато функцій, навіть у безкоштовній версії;
- пакет MySQL включений в стандартні репозиторії найбільш поширених дистрибутивів операційної системи Linux, що дозволяє встановлювати її елементарно;
- підтримує набір призначених для користувача інтерфейсів;

- може працювати з іншими базами даних, включаючи DB2 і Oracle.

Недоліки:

- середня продуктивність;
- зміна структури даних може бути досить трудомістким процесом, особливо при великому кількості зв'язків між даними в різних таблицях і навіть при простому додаванні полів;
- чутливість до нестабільності сервера.

### Структура БД.

База даних складається з двох таблиць.

Таблиця “orders” – призначена для того, щоб ми могли провести зв'язок між imei-номером, котрий служить для нас унікальним ідентифікатором пристрою, та номером замовлення, місцеположення якого ми відстежуємо. Структура таблиці “orders” відображена у табл. 3.1.

Таблиця “locs” в свою чергу служить для зберігання координат місцезнаходження кожного замовлення. Структура таблиці “locs” відображена у табл. 3.2.

Таблиця 3.1 – структура таблиці “orders”

назва стовпця	тип даних	призначення
id	int(11)	унікальний ідентифікатор запису
imei	varchar(45)	унікальний ідентифікатор пристрою
orderid	varchar(45)	унікальний ідентифікатор замовлення

Таблиця 3.2 – структура таблиці “locs”

назва стовпця	тип даних	призначення
id	int(11)	унікальний ідентифікатор запису
orderid	varchar(45)	унікальний ідентифікатор замовлення

latitude	varchar(15)	широта
longitude	varchar(15)	довгота
date	DATETIME	дата запису

Нижче приведені скрипти для створення бази даних.

```
CREATE SCHEMA `dip`;
```

```
USE dip;
```

```
CREATE TABLE `dip`.`locs` (
```

```
    `id` INT NOT NULL AUTO_INCREMENT,
```

```
    `order` VARCHAR(45) NOT NULL,
```

```
    `latitude` VARCHAR(15) NOT NULL,
```

```
    `longitude` VARCHAR(15) NOT NULL,
```

```
    `date` DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
```

```
    PRIMARY KEY (`id`),
```

```
    UNIQUE INDEX `id_UNIQUE` (`id` ASC));
```

```
CREATE TABLE `dip`.`new_table` (
```

```
    `id` INT NOT NULL AUTO_INCREMENT,
```

```
    `imei` VARCHAR(45) NOT NULL,
```

```
    `order` VARCHAR(45) NOT NULL,
```

```
    PRIMARY KEY (`id`),
```

```
    UNIQUE INDEX `id_UNIQUE` (`id` ASC));
```

### 3.3 Розробка http-сервера.

У якості web-сервера була використана зв'язка “Nginx” та “Puma”. Nginx – це безкоштовний веб і поштовий проксі-сервер з асинхронною архітектурою і відкритим кодом. Це потужний інструмент для розгортання веб-сервера, який

при правильному налаштуванні перевершує Apache. Области застосування Nginx досить великі - від кешування HTTP до створення інвертованого проксі-сервера.

Puma – це простий, швидкий, багатопоточний сервер HTTP 1.1 для додатків Ruby / Rack. Puma призначена для використання як в середовищі розробки, так і на виробництві. Він відмінно підходить для паралельних реалізацій Ruby, таких як Rubinius і JRuby, а також для забезпечення підтримки робочих процесів для хорошої підтримки CRuby.

В нашій системі завдання сервера прийняти GET-запит від трекера, який має вигляд “http://mysite.ua/?all=990000862471854+CIPGSMLOC:0,138.896461,36.482664,2019/02/14,12:49:57”, обробити значення параметра “all” та зберегти потрібну інформацію у базі даних. Нижче наведений повний текст програми на мові Ruby з коментарями.

```
require 'sinatra'

# 'sinatra' це предметно-орієнтований каркас (DSL) для швидкого
# створення функціональних веб-додатків на Ruby
require 'mysql2'

# бібліотека для роботи з Mysql
CLIENT = Mysql2::Client.new(:host => "localhost", :username => "root",
:password => "Mk3rBwSkMu3dfY8k68rq", :database => "dip")

# підключення до бази даних
get '/' do
  # обробка get-запиту
  data = params['all'].split(" CIPGSMLOC: ")
  loc = data[1].split(",")
  if loc[0] == "0"
    CLIENT.query("SELECT * FROM orders WHERE
imei=#{data[0]}").each do |row|
      # запит до бд, щоб знайти усі замовлення за даним imei
```

```
CLIENT.query("INSERT INTO `dip`.`locs` (`longitude`, `latitude`,
`orderid`) VALUES ('#{loc[1]}', '#{loc[2]}', '#{row["orderid"]}');
```

```
# запис координат для кожного замовлення
```

```
end
```

```
end
```

```
end
```

### 3.4 Розробка користувальницького інтерфейсу.

Завданням користувальницького інтерфейсу є - відобразити останнє записане у БД місце розташування по ідентифікаційному номеру замовлення. Є багато варіантів реалізації користувальницького інтерфейсу, наприклад, веб-сторінка з формою, десктопна програма та ін. У нашому проекті буде використано телеграм бот. Боти - спеціальні акаунти в Telegram, створені для того, щоб автоматично обробляти і відправляти повідомлення. Користувачі можуть взаємодіяти з ботами за допомогою повідомлень, що відправляються через звичайні або групові чати. Логіка бота контролюється за допомогою HTTPS запитів до API. Боти – особливі акаунти, які не вимагають номера телефону при створенні. По суті, ці акаунти грають роль інтерфейсу до вашого сервісу, який працює на віддаленому сервері. Найцікавіше в ботах це те, що для їх створення не потрібно вивчати низькорівневі методи роботи з MTPROTO і шифруванням – спілкування з роботом організовано за допомогою звичайного HTTPS інтерфейсу з спрощеними методами Telegram API [8]. Нижче наведений повний текст програми на мові Ruby з коментарями.

```
require 'telegram/bot'
```

```
# Ruby обгортка для Bot API Telegram
```

```
require 'mysql2'
```

```
# бібліотека для роботи з Mysql
```

```
CLIENT = Mysql2::Client.new(:host => "localhost", :username => "root",
:password => "Mk3rBwSkMu3dfY8k68rq", :database => "dip")
```



```

# підключення до бази даних
token = '*YOUR_TELEGRAM_BOT_API_TOKEN*'
# токен бота
Telegram::Bot::Client.run(token) do |bot|
# створення бота
bot.listen do |message|
# цикл прослуховування оновлень бота
case message
when Telegram::Bot::Types::Message
open("logs/#{DateTime.now.strftime("%d.%m.%Y")}.txt", 'a+') do
|f|
f.puts "#{Time.at(message.date)}; chat_id:
#{message.chat.id}; username: #{message.from.username}; id: #{message.from.id};
text: #{message.text}"
# запис логів в файл
end
puts "#{Time.at(message.date)}; username:
#{message.from.username}; text: #{message.text}" # вивід в консоль
case message.text
when '/start'
bot.api.send_message(chat_id: message.chat.id, text:
"Hello, #{message.from.first_name}. Type order:XXXXXXXXXX to get location.")
end
if message.text.start_with?("order:")
orderid = message.text.strip.delete_prefix("order:")
CLIENT.query("SELECT * FROM locs WHERE
orderid='#{orderid}' ORDER BY id DESC LIMIT 1").each do |row|
text = "order: #{orderid} now
here:\nwww.google.com/maps/search/?api=1&query=#{row["latitude"]},#{row["lon
gitude"]}\n"

```

```
        bot.api.send_message(chat_id: message.chat.id,  
                             text: text)  
    end  
end  
end  
end  
end
```

## ВИСНОВКИ

1. В даному проекті розроблена комп'ютерна система моніторингу рухливих об'єктів з можливістю відслідковувати об'єкт за допомогою користувальницького інтерфейсу.
2. Були розглянуті завдання, які повинна виконувати система моніторингу рухливих об'єктів. Була проведена класифікація систем за технічними та функціональними ознаками, за об'єктом моніторингу. Також були розглянуті компоненти з яких складається система.
3. Був розглянутий трекер, як основний елемент в системі. Визначено його компоненти, функція та класифікація. З'ясовано, що визначення місцеположення об'єктів по мережам мобільного зв'язку може бути успішно реалізована.
4. Розроблено апаратне забезпечення: схема електрична принципова, підібрані комплектуючі, друкована плата, вибрані умови охолодження і розрахований тепловий режим, розрахунок надійності пристрою.
5. Розроблено програмне забезпечення: програма мікроконтролера, http сервер, база даних, користувальницький інтерфейс.
6. Отримана система може бути використана в сфері кур'єрських і поштових служб.

## СПИСОК ЛІТЕРАТУРИ

1. Спутниковый мониторинг транспорта. Режим доступа URL: [ru.wikipedia.org/wiki/Спутниковый\\_мониторинг\\_транспорта](http://ru.wikipedia.org/wiki/Спутниковый_мониторинг_транспорта)
2. HowStuffWorks: How GPS Receivers Work. Режим доступа URL: [electronics.howstuffworks.com/gadgets/travel/gps1.htm](http://electronics.howstuffworks.com/gadgets/travel/gps1.htm)
3. The power of Wifi geolocation. Режим доступа URL: [medium.com/@nlesconnec/the-power-of-wifi-geolocation-70d2494b066d](http://medium.com/@nlesconnec/the-power-of-wifi-geolocation-70d2494b066d)
4. Gsm Cell ID. Режим доступа URL: [https://en.wikipedia.org/wiki/GSM\\_Cell\\_ID](https://en.wikipedia.org/wiki/GSM_Cell_ID)
5. OpenCellID – The largest Open Database of Cell Towers. Режим доступа URL: <https://opencellid.org>
6. UnwiredLabs. Режим доступа URL: <https://unwiredlabs.com>
7. За пределами GSM: спутниковая связь в Wialon. Режим доступа URL: <https://gurtam.com/ru/blog/za-predelami-gsm-sputnikovaa-svaz-v-wialon>
8. Документация telegram: Боты. Режим доступа URL: <https://tlgrm.ru/docs/bots>
9. ГОСТ 23751-86. Плати друковані. Основні параметри конструкції. - Введ. 01.07.87.
10. ГОСТ 10316-78. Гетінакс і склотекстолит фольговані. Технічні умови. - Введ. 01.01.79 .
11. ГОСТ 2.755-87. Позначення умовні в графічних схемах. Пристрої комутаційні і контактні з'єднання. - Введ. 01.01.88.
12. ГОСТ 2.417-91. ЕСКД. Плати друковані. Правила виконання креслень. - Введ. 01.07.92.
13. ГОСТ 2.701-84. Схеми. Види і типи. Загальні вимоги до виконання. - Введ. 01.07.85.