

Центральноукраїнський національний технічний університет
Механіко-технологічний факультет
Кафедра кібербезпеки та програмного забезпечення

”Допущено до захисту”
Завідувач кафедри кібербезпеки
та програмного забезпечення
д.т.н., професор
_____ Олексій СМІРНОВ
« ____ » _____ 2023 р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за другим (магістерським) рівнем вищої освіти
на тему
“Дослідження та програмна реалізація системи GPS пристрою
на основі процесора ARM Cortex-A15 MPCore”

Виконав здобувач вищої освіти
II курсу, групи КІ-22М-2
ОПП «Комп’ютерна інженерія»
спеціальності 123 «Комп’ютерна інженерія»
_____ Лопотуха О.І.
« ____ » _____ 2023 р.

Керівник проекту
кандидат технічних наук, доцент
_____ Дреєв О.М.
« ____ » _____ 2023 р.
Рецензент _____

Центральноукраїнський національний технічний університет
Факультет *Механіко-технологічний*
Кафедра *Кібербезпеки та програмного забезпечення*
Рівень вищої освіти *магістр*
Галузь знань *12* "Інформаційні технології"
Спеціальність *123* "Комп'ютерна інженерія"
Освітньо-професійна (освітньо-наукова) програма "Комп'ютерна інженерія"

ЗАТВЕРДЖУЮ

Завідувач кафедри

д.т.н., проф.

Олексій СМІРНОВ

« 6 » вересня 2023 року

ЗАВДАННЯ НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ ЗА ДРУГИМ (МАГІСТЕРСЬКИМ) РІВНЕМ ВИЩОЇ ОСВІТИ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ

Лопотухі Олександр Ігоровичу

(прізвище, ім'я, по батькові)

- Тема роботи *Дослідження та програмна реалізація системи GPS пристрою на основі процесора ARM Cortex-A15 MPCore*
- Керівник роботи *Дреєв Олександр Миколайович, канд. техн. наук, доцент*
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом вищого навчального закладу № 35-13 від 04.08.2023 року
- Строк подання студентом роботи до захисту *10.12.2023 р.*
- Мета та завдання випускної кваліфікаційної роботи: *Метою розробки є дослідження та програмна реалізація системи GPS пристрою на основі процесора ARM Cortex-A15 MPCore*
- Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
 - Призначення та область використання.*
 - Наукова новизна.*
 - Перегляд аналогічних існуючих систем.*
 - Економічна ефективність розробленої програми.*
 - Опис і обґрунтування проектних рішень.*
 - Заходи з охорони праці та техніки безпеки.*
 - Етапи програмування системи.*
 - Висновки.*
 - Впровадження системи в промислову експлуатацію*

6. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

- | | |
|--|-----------------|
| <i>Наукова новизна</i> | <i>1 аркуш</i> |
| <i>Структурна схема системи</i> | <i>1 аркуш</i> |
| <i>Функціональна схема системи</i> | <i>1 аркуш</i> |
| <i>Діаграма процесів</i> | <i>1 аркуш</i> |
| <i>Блок-схема алгоритму роботи додатку</i> | <i>2 аркуша</i> |
| <i>Показники економічної ефективності</i> | <i>1 аркуш</i> |

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Економічний	Савеленко Г.В.	05.10.2023	14.11.2023
Охорона праці	Оришака О.В.	06.10.2023	16.11.2023

7. Дата видачі завдання « 6 » вересня 2023 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	Строк виконання етапів випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	Примітка
1.	Аналіз існуючих систем	10.10.2023 р.	
2.	Постановка задачі, оформлення ТЗ	15.10.2023 р.	
3.	Розробка моделі компонента	20.10.2023 р.	
4.	Розробка структур даних	25.10.2023 р.	
5.	Розробка алгоритмів зв'язку та відображення	30.10.2023 р.	
6.	Програмування алгоритмів	10.11.2023 р.	
7.	Розрахунок економічної ефективності	13.11.2023 р.	
8.	Розрахунки з охорони праці та техніки безпеки	15.11.2023 р.	
9.	Оформлення ПЗ	17.11.2023 р.	
10.	Попередній захист роботи	10.12.2023 р.	

Дата видачі завдання
« 6 » вересня 2023 р.

Підпис керівника

(прізвище та ініціали)Завдання прийнято до виконання
« 6 » вересня 2023 р.

Підпис здобувача

(прізвище та ініціали)

АНОТАЦІЯ

Лопотуха О.І. Дослідження та програмна реалізація системи GPS пристрою на основі процесора ARM Cortex-A15 MPCore. 123 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2023.

В даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи GPS пристрою на основі процесора ARM Cortex-A15 MPCore.

Метою розробки є дослідження та програмна реалізація системи GPS пристрою на основі процесора ARM Cortex-A15 MPCore.

Об'єктом дослідження є процес GPS пристрою на основі процесора ARM Cortex-A15 MPCore.

Предметом дослідження є методи GPS пристрою на основі процесора ARM Cortex-A15 MPCore.

Методи дослідження базуються на методах теорії геопозиціонування, методах математичної статистики, методах розробки програмного забезпечення.

Результат роботи – програмна реалізація системи GPS пристрою на основі процесора ARM Cortex-A15 MPCore.

В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.

Розроблено зручний інтерфейс користувача. Наведені інструкції по роботі з програмними засобами.

Програма може використовуватися на мобільному пристрої з ОС Android.

Програму розроблено в середовищі Delphi 10.

Ключові слова: комп'ютерна інженерія, GPS

ABSTRACT

Lopotukha O.I. Research and software implementation of the GPS device system based on the ARM Cortex-A15 MPCore processor. 123 Computer engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2023.

In this graduation thesis for the second (master's) level of higher education, software is developed, which is intended for the GPS system of the device based on the ARM Cortex-A15 MPCore processor.

The purpose of the development is the research and software implementation of the GPS device system based on the ARM Cortex-A15 MPCore processor.

The object of research is the GPS process of the device based on the ARM Cortex-A15 MPCore processor.

The subject of research is GPS methods of a device based on an ARM Cortex-A15 MPCore processor.

Research methods are based on geo-positioning theory methods, mathematical statistics methods, and software development methods.

The result of the work is a software implementation of the device's GPS system based on the ARM Cortex-A15 MPCore processor.

In the process of working on the software model, an analysis of existing hardware and software was performed. All components of the developed software are fully described.

A convenient user interface has been developed. Instructions for working with software tools are provided.

The program can be used on a mobile device with Android OS.

The program was developed in the Delphi 10 environment.

Keywords: computer engineering, GPS

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	3
ВСТУП.....	4
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ	7
1.1 Призначення системи.....	7
1.2 Область застосування.....	7
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	9
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	9
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	14
2.3 Розгорнута постановка завдання	20
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	22
3.1 Опис функціонування системи	22
3.2 Розробка структурної схеми.....	34
3.3 Розробка функціональної схеми	40
3.4 Розробка діаграми процесів.....	47
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	49
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	49
4.2 Захист розробленого програмного забезпечення.....	57
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	59
6 НАУКОВА НОВИЗНА	62

						ВКРМ-123.23.0040.00.00.ПЗ		
Вим	Арк.	№ докум.	Підп.	Дата				
Розроб.	Топотуха О.І.				Дослідження та програмна реалізація системи GPS пристрою на основі процесора ARM Cortex-A15 MPCore	Літ.	Аркуш	Аркушів
Перев.	Дресв О.М.					М	1	102
Н.контр.	Коваленко А.С.					ЦНТУ КІ-22М-2		
Затв.	Смірнов О.А.							

7 ЕКОНОМІЧНА ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ.....	63
7.1 Техніко економічне обґрунтування теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	63
7.2 Розрахунок трудомісткості розробки програмної продукції.....	65
7.3 Визначення чисельності виконавців і планового фонду зарплати.....	67
7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника.....	71
7.5 Визначення собівартості розробки та ціни програмної продукції.....	75
7.6 Визначення об'єму капітальних вкладень та експлуатаційних витрат у споживача програмної продукції.....	79
7.7 Визначення експлуатаційних витрат.....	79
7.8 Визначення економічної ефективності програмної продукції.....	81
7.9 Висновок.....	83
8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ	84
8.1 Вступ.....	84
8.2 Аналіз умов праці на робочому місці ІТ-фахівця.....	85
8.3 Пропозиції щодо підвищення працездатності ІТ-фахівців.....	88
8.4 Розрахункова частина	89
8.5 Висновки до розділу.....	92
9 ОСНОВНІ ВИСНОВКИ.....	94
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	96

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

- КПК – кишеньковий персональний комп'ютер
- РМД – робоче місце диспетчера
- СЕР – Circular Error of Probability – окружність можливої помилки
- GPS – глобальна система місцевизначення.
- GSM – система цифрового стільникового телефонного зв'язку
- IOD – issue of data – часовий ідентифікатор
- LAAS – навігаційний комплекс
- LAT – широта
- LON – довгота
- MFC – Microsoft Foundation Class library – бібліотека класів
- OLE – технологія зв'язування й вбудовування об'єктів
- POI – point of interest – вбудована база «цікавих» точок
- SA – режим обмеженого доступу
- WAAS – система супутників і наземних станцій, що робить коректування сигналів GPS

					ВКРМ-123.23.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

ВСТУП

Актуальність теми. Тримати завжди під рукою, а саме у своєму пристрої, установлені програми GPS-навігації просто необхідно, особливо для тих, хто часто подорожує по незнайомих містах і країнам або не представляє життя без походів на природу.

Ще однією причиною використання навігаторів є "вічні" пробки на дорогах. Відповідно до статистики, людина з особистим автомобілем за все життя простоює в пробках у середньому біля двох років.

Програми для навігації на вашому пристрої зможуть виручити вас у саму важку хвилину.

Навігатор дозволяє будувати маршрути для поїздок на суспільному транспорті або на особистому автомобілі з можливістю об'їзду пробок, знаходити адреси й організації з урахуванням вашого місцеположення.

Навігатор вчасно перебудовує маршрут, залежно від пробок і інших подій на дорогах, при цьому супроводжує кожний маневр маршруту голосовими підказками.

Цей додаток може попереджати на дорогах про камери ДПС, аварії, ремонтні роботи і інші події. Також навігаційний додаток здатний показувати додаткову інформацію про будинки, організації й інші об'єкти на карті.

Можна вказати проміжні точки маршруту, які вам необхідно відвідати по шляху в пункт призначення. Є присутньою можливість збереження потрібних адрес у Вибраному з метою наступного швидкого доступу до них.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи GPS пристрою на основі процесора ARM Cortex-A15 MPCore.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

					ВКРМ-123.23.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

– Огляд існуючих систем GPS пристрою на основі процесора ARM Cortex-A15 MPCore.

– Дослідження системи GPS пристрою на основі процесора ARM Cortex-A15 MPCore.

– Програмна реалізація системи GPS пристрою на основі процесора ARM Cortex-A15 MPCore.

Об'єктом дослідження є процес GPS пристрою на основі процесора ARM Cortex-A15 MPCore.

Предметом дослідження є методи GPS пристрою на основі процесора ARM Cortex-A15 MPCore.

Методи дослідження базуються на методах теорії геопозиціонування, методах математичної статистики, методах розробки програмного забезпечення.

Наукова новизна отриманих результатів. У процесі рішення завдань, обумовлених цілями дослідження, отримані наступні результати:

– Удосконалено метод GPS пристрою на основі процесора ARM Cortex-A15 MPCore.

– Розроблено вітчизняний продукт GPS пристрою на основі процесора ARM Cortex-A15 MPCore, який має більш широкі можливості, на відміну від існуючих аналогів.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі GPS пристрою на основі процесора ARM Cortex-A15 MPCore.

Достовірність наукових результатів підтверджена теоретичними викладеннями, даними комп'ютерного моделювання, коректними дослідженнями параметрів на функціонуючій обчислювальній мережі, а також відповідністю отриманих результатів окремим результатам, наведеним у науковій літературі.

Робота апробована на LVII Науково-технічній конференції здобувачів вищої освіти «Наука – виробництву», 2023, основні положення випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти надруковані у статті збірника праць молодих науковців ЦНТУ, випуск №14.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи GPS пристрою на основі процесора ARM Cortex-A15 MPCore, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

КБГПЗ - 2023

					VKPM-123.23.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Призначенням системи є реалізація програмного забезпечення GPS пристрою на основі процесора ARM Cortex-A15 MPCore. Програмне забезпечення, яке розробляється у даному проекті, пропонує:

- навігацію з голосовим супроводом для автомобілістів, велосипедистів і пішоходи, а також схеми маршрутів суспільного транспорту;
- актуальну інформацію про дороги (завантаженість, пробки);
- пошук і інформацію про організації, а також пошук по адресах;
- карту двох варіантів: класична карта й карта по фотографіях із супутника;
- динамічні POI – надають актуальну інформацію (ціни на бензин, розклад сеансів у кінотеатрах і т.д.) про різні типи об'єктів на карті;
- 3D-розв'язки;
- завантаження нових і відновлення старих карт прямо із пристрою;
- голосовий пошук;
- внутрішні схеми великих будинків (аеропорти, магазини, готелі й т.д.);
- панорамні фотографії вулиць.

1.2 Область застосування

Областю застосування є мобільні пристрої, у яких існує GPS-чип. Розглянемо основні сфери застосування GPS:

- Навігація – авіаційна, автомобільна й морська. Маючи на борті GPS пристрій обчислить оптимальний маршрут руху.

					ВКРМ-123.23.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

– Стільниковий зв'язок – у зоні покриття дозволяє визначити поточне місце розташування абонента/ів і забезпечити безперервність мобільного зв'язку.

– У геодезії система GPS може не тільки вимірювати простір і місцевість, визначити гравітаційне поле Землі, але й з точністю до міліметра розрахувати спорудження в будівництві.

– У картографії, GPS дозволяє визначити й відобразити на карті образно-знакові моделі простору, такі як – Земля, зоряні тіла, зоряне небо й Всесвіт.

– За допомогою GPS фахівці в області тектоніки можуть одержувати точні дані про структуру, тверду оболонку й рух Землі.

– GPS системи спостереження й моніторинг транспорту, укаже на швидкість і місце розташування транспортного засобу.

– Геотегінг – прив'язка графічних зображень до інформаційних ресурсів.

– Геокешінг – застосовується в туризмі, як спосіб активного відпочинку.

По даним GPS приладу можна виявити задані геокоординати, у яких заховані, приміром, схованки.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи GPS пристрою на основі процесора ARM Cortex-A15 MPCore, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					ВКРМ-123.23.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

2.1 Огляд існуючих систем, технологій, архітектур, програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти

BackCountry Навігатор

Ціна: безкоштовно / 9,99 доларів США / до 29,99 доларів США.

BackCountry Navigator – чудовий навігаційний додаток для туристів. Це досить дорого, але ви отримуєте кілька офлайн-топографічних карт. Він ідеально підходить для тих місць, де немає служб передачі даних. Також є кілька цікавих функцій. Ви можете додавати улюблені місця та маркери на карту, переглядати різні типи маршрутів тощо. Тут також є кілька варіантів карти. Купуючи додаток, ви отримуєте необмежений доступ до нього без реклами. Він потужний, але обов'язково спробуйте його, перш ніж купувати. Це досить дорого.

HERE WeGo Maps

Ціна: безкоштовно.

HERE WeGo є одним із небагатьох серйозних конкурентів Google Maps у сфері навігаційних програм. Він має простий, елегантний інтерфейс із опціями картографування по всьому світу. Ви можете завантажити карти для використання в режимі офлайн у вашому регіоні. Це має допомогти в регіонах з нерівним з'єднанням даних. ТУТ також показує вам інформацію про дорожній рух (якщо доступна), карти громадського транспорту, і ви можете налаштувати, зберігаючи місця для швидкого маршруту пізніше. Усе це безкоштовно. Існує також програма для створення карт, за допомогою якої ви також можете змінювати карти. Єдиним мінусом є слабка підтримка Android Auto.

					ВКРМ-123.23.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

MapFactor – один із простих популярних навігаційних додатків. Він містить основні функції навігації та GPS. Він використовує OpenStreetMap. Це означає, що ви можете отримати безкоштовні офлайн-карти, які оновлюються щомісяця, якщо хочете. Також є додаткові параметри карти, але вони можуть коштувати грошей. Деякі інші функції включають голосові вказівки, транскордонне маршрутизація, режими 2D і 3D, денні та нічні теми тощо. Ця програма GPS також підтримує десятки країн світу. Це не наполовину погано для того, що ви отримуєте.

MapQuest

Ціна: безкоштовно (з рекламою).

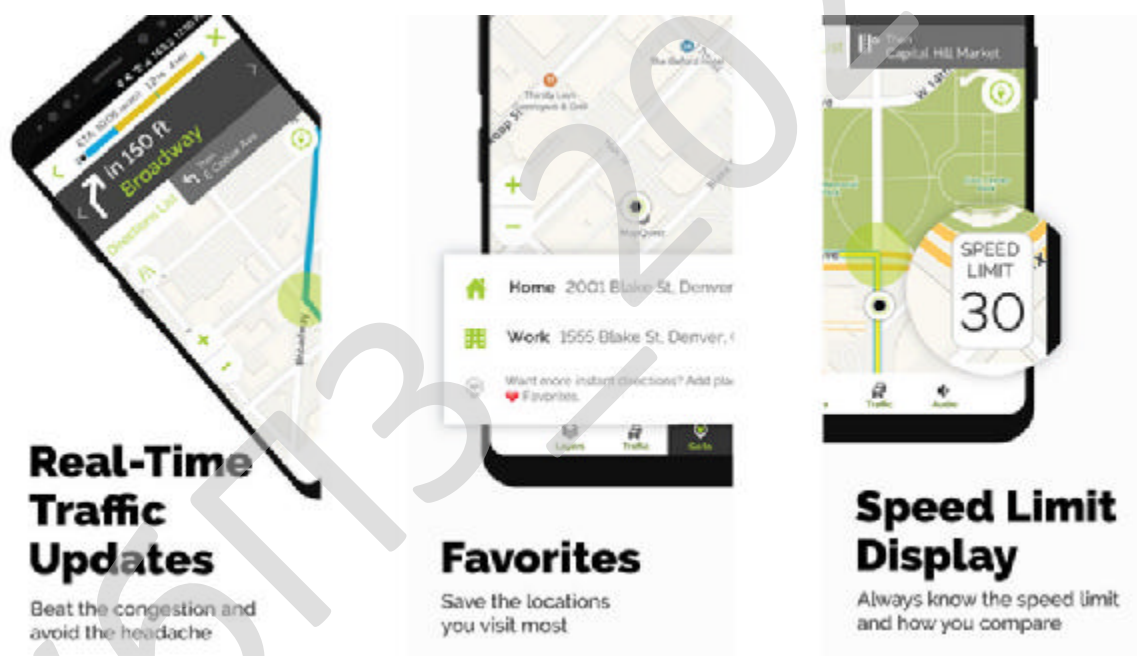


Рисунок 2.3 – MapQuest

За ці роки MapQuest зазнав багатьох змін. Однак це той самий надійний сервіс, про який ми знаємо ще зі старих добрих часів. Він використовує стандартні покрокові маршрути. Існують також інші функції, такі як поточні оновлення заторів (де доступно), зміна маршруту на основі заторів і послуга пошуку дешевшого бензину. Ви навіть можете використовувати його, щоб

					ВКРМ-123.23.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

викликати евакуатор у разі поломки. Він не такий хороший, як Google Maps, і не такий ефективний у пошуку дешевого бензину, як GasBuddy. Однак це чудовий спосіб вбити двох зайців одним пострілом.

Карти та Waze

Ціна: безкоштовно.

Карти Google і Waze – чудові програми GPS. Обидва вони належать Google. Карти Google є свого роду мірилом для навігаційних програм. У ньому є безліч місць, оглядів, маршрутів і фотографій більшості місць на рівні вулиць. Крім того, він має функції доступності, і ви можете завантажувати карти для використання в автономному режимі. Waze трохи простіше. Він чудово підходить для вказівок, особливо під час подорожей або щоденних поїздки.

Maps.Me

Ціна: безкоштовно.

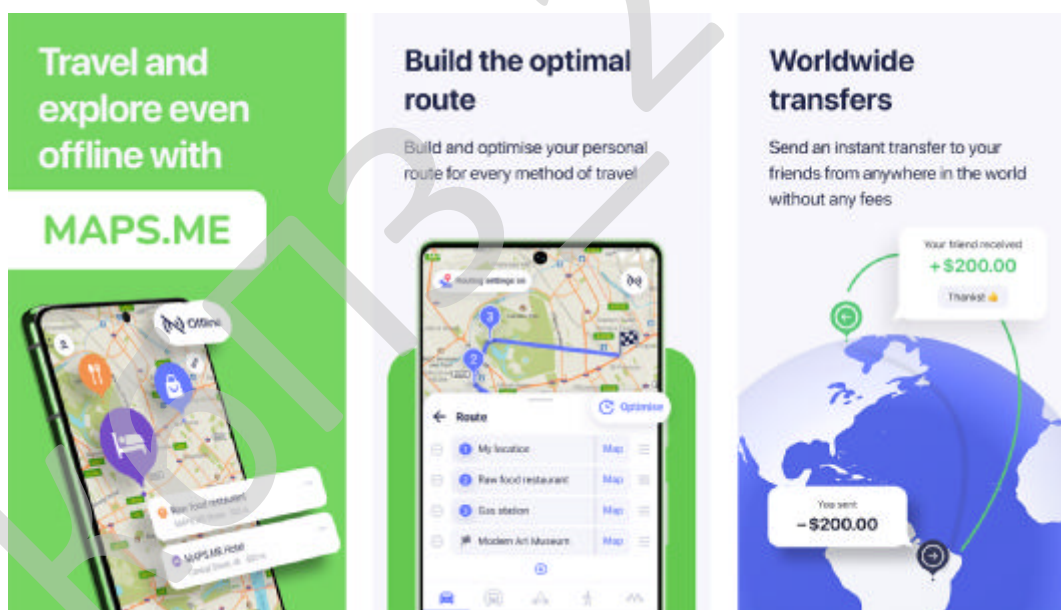


Рисунок 2.4 – Maps.me

Maps.me – ще один безкоштовний додаток для навігації та GPS. Він містить офлайн-карти, покрокові вказівки та оновлені карти через OpenStreetMap. Є також закладки місць, офлайн-пошук, дані про дорожній рух (де доступно) і

					ВКРМ-123.23.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

підтримка по всьому світу. Це майже якнайближче до конкурента Google Maps. Це повинно працювати для більшості людей більшу частину часу. Деякі повідомляли про деякі проблеми тут і там, але нічого надто тривожного. Програма абсолютно безкоштовна, без покупок у програмі. Є кілька оголошень, але вони не є великою проблемою.

OsmAnd

Ціна: безкоштовно / до \$8,99

OsmAnd – це інший тип навігаційної програми. Він призначений безпосередньо для людей, яким потрібні офлайн-карти. Ви можете завантажити купу з них, і більшість із них досить хороші. Безкоштовна версія програми має обмеження на завантаження. Ви можете придбати додаткові завантаження через покупки в програмі. Деякі інші функції включають покрокову навігацію, офлайн-пошук, карти для інших речей, окрім водіння, тощо. Можливо, це не так добре, як щось на кшталт HERE або Google Maps для використання в Інтернеті, але це одна з найкращих офлайн-додатків GPS, доступних наразі.

GPS-навігація Polaris

Ціна: безкоштовно / 0,99 дол

Polaris Navigation намагається бути універсальною навігаційною програмою, і в більшості випадків це вдається. Його найбільшою особливістю є те, що він має доступ до карт Google Maps, OpenStreetMap, карт MapQuest і карт велосипедних маршрутів. Отже, будь-яке джерело, яке ви хочете, ви отримуєте. Він також має кілька форматів координат, запис маршруту, унікальну систему керування маршрутними точками та ваші стандартні речі, як-от покрокові вказівки. Є також кілька функцій для таких заходів, як походи, кемпінг, риболовля, полювання та інші види активного відпочинку. Він виглядає старим, але працює добре.

Sygie: GPS, навігація та офлайн-карти

Ціна: безкоштовно / \$13,99 / \$17,99 / \$29,99

Sygie – надзвичайно популярна навігаційна програма, яку на сьогодні

					ВКРМ-123.23.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

завантажили понад 50 мільйонів. Як і інші в цьому списку, він надає завантажені карти для використання в автономному режимі, за винятком того, що тут використовуються карти TomTom. Він має такі звичайні функції, як покрокові вказівки, голосові вказівки (що включають голосові назви вулиць, щоб ви могли зосередитися на водінні), альтернативні маршрути та навіть дисплей обмеження швидкості.

Ви можете розблокувати додаткові функції, придбавши одну з трьох преміум-версій. Варіант за 13,99 доларів США надає вам базові функції, варіант за 17,99 доларів США додає оновлення трафіку в реальному часі, а версія з кінцевою ціною надає вам усе. Це дорого, але принаймні це не підписка.

GPS-навігатор Scout

Ціна: безкоштовно / \$24,99 на рік

Scout GPS – це фактично старіша програма GPS зі свіжим шаром фарби. Раніше він називався TeleNav, і він змінив назву, піктограму та зовнішній вигляд, щоб надати йому свіжого вигляду. Це все ще надійна та стабільна навігаційна програма з усіма основними функціями, включаючи покрокову навігацію. Однак Scout GPS зосереджується більше на людях, а не на навігації, що надає йому унікальності.

Він може автоматично надсилати людям повідомлення про ваш прибуток і сповіщати вас, коли ви потрібні іншим. Елемент соціальних мереж, що базується на місцезнаходженнях, допомагає відокремити його від зграї. Повна версія коштує \$24,99 на рік. Це також система, яку використовують деякі виробники автомобілів. Це не дуже важливо, але нам подобаються маленькі цікаві факти.

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Embarcadero Delphi, раніше Borland Delphi і Codegear Delphi, – інтегроване середовище розробки ПЗ для Microsoft Windows, Mac OS, iOS і Android мовою

					ВКРМ-123.23.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		14

Delphi (що раніше носила назву Object Pascal), створена спочатку фірмою Borland і на даний момент приналежна й розроблювальна Embarcadero Technologies. Embarcadero Delphi є частиною пакета Embarcadero RAD Studio і поставляється в чотирьох редакціях: Community (поширюється безкоштовно й має обмежену ліцензію на використання в комерційних цілях), Professional, Enterprise і Architect.

Delphi 10.4 Sydney

Випущено 26 травня 2020 року. RAD Studio Delphi 10.4 забезпечує значно поліпшену високопродуктивну нативну підтримку Windows, кращу продуктивність розробки, миттєві підказки code completion, прискорення виконання коду із синтаксисом керованих записів, поліпшення виконання паралельних завдань на сучасних багатоядерних CPU, а також містить більш 1000 виправлень багів, поліпшення продуктивності середовища й бібліотек і багато чого крім того.

Основні можливості Delphi 10.4.1:

– Істотні розширення для Windows: поліпшення для застосунків на моніторах 4K High DPI, інтеграція з новим WebView2 на базі Chromium, використання розширених title bars, таких же, як в Office, Explorer, Google Chrome.

– Керування пам'яттю в Delphi тепер стандартизоване на всіх підтримуваних платформах – мобільних, настільних і серверних – використовувачи класичну реалізацію керування пам'яттю об'єктів.

– Істотне поліпшення Delphi Code Insight (без можливого блокування IDE – в окремому процесі), що допоможе при роботі з великими проектами.

– Тип даних Delphi «record» тепер підтримуватимуть довільні ініціалізацію, фіналізацію й операції копіювання.

– Розширена підтримка бібліотек C++: ZeroMQ, SDL2, SOCI, libSIMDpp і Nematode.

– Відладник Win 64 (на LLDB) і збирач для C++.

					ВКРМ-123.23.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

– Поліпшення для C++: Включена велика кількість поліпшень STL з Dinkumware.

– Підтримка Metal Driver GPU для macOS і iOS.

– Вбудований Fmxlinux.

– Компонент Twebbrowser для iOS тепер реалізований на Wkwebview API.

Реалізація компонента Media Player для macOS тепер використовує Avfoundation.

Реалізований заново стилізуємий FMX компонент TМемо на платформі Windows значно поліпшений і тепер має відмінну підтримку ІМЕ.

– Численні поліпшення швидкості й стабільності роботи нашої бібліотеки The Parallel Programming Library (PPL).

– Додані оновлені драйвери для FireBird, PostgreSQL і SQLite.

– Клієнтські бібліотеки HTTP і REST Client розширені застосунковими можливостями роботи з HTTPS. Також були розширені можливості підтримки Amazon AWS services

– У технологію Visual LiveBindings внесена безліч поліпшень, у тому числі швидкодії, що стосуються, застосунків на VCL і FireMonkey

RAD Studio 10.4 Короткий огляд:

– Істотні розширення для Windows. Створення застосунків, що чудово виглядають, із чіткими елементами інтерфейсу на 4к моніторах High DPI за допомогою нової гнучкої підтримки стилів елементів керування на екрані. Інтеграція із сучасними, безпечними web-технологіями від Microsoft – новим WebView2 на базі Chromium. Використання сучасних розширених title bars, таких же, як в Office, Explorer, Google Chrome, у своїх проєктах. Істотні поліпшення надійності налагодження в новому відладнику для C++ Windows 64-bit.

– Зросла продуктивність розробки. Ріст продуктивності за рахунок миттєвої реакції підказок code completion у середовищі IDE. Краща сумісність із уже наявною кодовою базою, і спрощення програмування за рахунок уніфікованої архітектури керування пам'яттю. Швидке зв'язування даних і візуальних елементів за допомогою розширеної технології Visual LiveBindings з

					ВКРМ-123.23.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

підвищеною швидкодією. Просте використання розповсюджених бібліотек C++, наприклад, ZeroMQ, SDL2, SOCI, libSIMDpp і Nematode. Оновлена підтримка Amazon AWS cloud.

– Поліпшення швидкодії і якості. Більш 1000 поліпшень швидкодії і якості. Краща ефективність коду за допомогою нового синтаксису `custom managed records`. Більш швидке виконання паралельних завдань на сучасних багатоядерних CPU. Переконаєтеся в прискоренні відображення на екрані з підтримкою Metal API на macOS і iOS. Краща сумісність із уже наявною кодовою базою й спрощення програмування за рахунок уніфікованої архітектури керування пам'яттю.

Істотне поліпшення Delphi Code Insight

Як найбільше й головне поліпшення інструментів програмування Delphi за багато років, в 10.4 Delphi Code Insight реалізований через Language Server Protocol (LSP). LSP – це технологія генерації результатів для code completion, навігації й інших сервісів в окремому процесі. Це значить, що code completion і Code Insight одержать більш точні результати без блокування IDE. 10.4 забезпечує набагато більш високу продуктивність розроблювачів, які працюють із більшими проектами, що містять мільйони рядків коду.

Delphi Custom Managed Records

Ключове розширення мови Delphi: тип даних Delphi «record» тепер підтримуть довільні ініціалізацію, фіналізацію й операції копіювання. Управляйте тем, як ці структури створюються, копіюються й звільняються з допомогу вашого коду, який буде виконуватися у відповідний момент.

Це розширює потужність конструкцій records в Delphi, які використовуються щоб одержати більшу ефективність у порівнянні із класами.

Єдине керування пам'яттю

Керування пам'яттю в Delphi тепер стандартизоване на всіх підтримуваних платформах – мобільних, настільних і серверних – використовувачи класичну реалізацію керування пам'яттю об'єктів.

					ВКРМ-123.23.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

- Користувачам редакцій Enterprise або Architect доступна повна інтеграція Fmxlinux з IDE для створення клієнтських застосунків Linux з GUI.
- Компонент Twebbrowser для iOS тепер реалізований на Wkwebview API.
- Реалізація компонента Media Player для macOS тепер використовує Avfoundation.

Оновлений менеджер пакетів Getit

Менеджер пакетів Getit в IDE був значно вдосконалений.

Дати випуску релізів пакетів тепер видні, і можливе сортування списку по цих датах; відбір тільки встановлених пакетів, контенту, доступного тільки при наявності підписки, багато чого іншого.

Універсальний інсталятор для установки Online і Offline

В 10.4 включений новий універсальний інсталятор, який використовує технологію на базі Getit. Цей інсталятор підтримує як online, так і offline (з ISO) варіанти установки.

Тепер обоє варіанта установки дозволяють вам указати початковий набір можливостей RAD Studio для установки, наприклад, свою комбінацію мов програмування й цільових платформ, мов інтерфейсу, і додавати до нього або видаляти непотрібне в будь-який момент.

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випускню кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи GPS пристрою на основі процесора ARM Cortex-A15 MPCore.

В процесі розробки випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти необхідно виконати наступний обсяг роботи:

- а) провести аналіз існуючих систем-аналогів для виявлення їх позитивних і негативних якостей. Результати аналізу врахувати в подальших розробках;

					ВКРМ-123.23.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

б) вибрати та обґрунтувати методику побудови системи контролю роботи технологічного обладнання на виробництві в автоматизованому режимі. Розробити функціональну та структурну схеми системи;

в) розробити програмне забезпечення системи, що дозволить реалізувати поставлену технічним завданням задачу. Побудувати блок-схеми алгоритмів програми та підпрограми;

г) організувати інтерфейс користувача з метою формування та виводу на екран ЕОМ повідомлень про некоректні дії користувача та нестандартні ситуації в роботі технологічного обладнання;

д) розробити рекомендації по організаційних та методичних заходах, які забезпечать впровадження системи в промислову експлуатацію та її подальшу успішну експлуатацію;

е) провести розрахунки по визначенню економічної ефективності розробленої системи;

ж) розробити заходи по охороні праці при впровадженні та експлуатації системи, а також розробити заходи з цивільного захисту;

з) сформулювати висновки про виконаний обсяг робіт та одержані результати.

					ВКРМ-123.23.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Навігаційна система, заснована на супутниках, як Глобальна система позиціонування (GPS), складається з 24 супутникової мережі, розташованих на орбіті через Міністерство оборони США. Ця система в основному призначена для військових застосувань; однак у 1980 році уряд зробив систему доступною для цивільного використання. Ця система працює в будь-якому середовищі по всьому світу протягом 365 днів у будь-який час. GPS містить 24 супутники, які обертаються навколо сфери кожні 12 годин, щоб надати інформацію про час, положення та швидкість у всьому світі. Основною функцією GPS є точне визначення місць на земній кулі шляхом визначення відстані від супутників. Ця система дозволяє іншим чином записувати точні місця на земній кулі та допомагати вам орієнтуватися з цих місць. В основному ця система була розроблена в основному для військових застосувань, але в 1980 році вона стала доступною для цивільного використання. У цьому розділі розглядається огляд системи GPS, її роботи та використання.

Що таке система GPS?

Визначення. Повний термін GPS – це «система глобального позиціонування», яка є супутниковою навігаційною системою, яка надає користувачеві інформацію про місцезнаходження та час за будь-яких кліматичних умов. GPS також використовується для навігації в літаках, кораблях, автомобілях і вантажівках. Система надає важливі можливості військовим і цивільним користувачам по всьому світу. GPS забезпечує безперервне тривимірне позиціонування, навігацію та час у всьому світі в реальному часі.

					ВКРМ-123.23.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

Як працює система GPS?

GPS складається з трьох сегментів:

- Космічний сегмент: супутники GPS.
- Система контролю, якою керують військові США,.
- Сегмент користувачів, який включає як військових, так і цивільних користувачів та їх GPS-обладнання.

Космічний сегмент

Космічний сегмент – це кількість супутників у сузір'ї. Він складається з 29 супутників, які обертаються навколо Землі кожні 12 годин на висоті 12 000 миль. Функція космічного сегмента використовується для сигналів маршруту/навігації, а також для зберігання та повторної передачі повідомлення маршруту/навігації, надісланого сегментом керування. Ці передачі контролюються дуже стабільними атомними годинниками на супутниках. Космічний сегмент GPS утворений сузір'ям супутників із достатньою кількістю супутників, щоб гарантувати, що користувачі матимуть у полі зору принаймні 4 супутники одночасно з будь-якої точки на поверхні Землі в будь-який час.

Контрольний сегмент

Сегмент керування складається з головної станції керування та п'яти станцій моніторингу, обладнаних атомними годинниками, які розміщені по всьому світу. П'ять станцій моніторингу відстежують сигнали супутників GPS, а потім надсилають кваліфіковану інформацію на головну станцію керування, де відхилення переглядаються та надсилаються назад на супутники GPS через наземні антени. Сегмент керування також називається станцією моніторингу.

Сегмент користувача

Сегмент користувача складається з приймача GPS, який приймає сигнали від супутників GPS і визначає, на якій відстані він знаходиться від кожного супутника. В основному цей сегмент використовується для армії США, систем наведення ракет, цивільних програм для GPS майже в усіх сферах. Більшість цивільних використовують це від обстеження до транспортування до природних

					ВКРМ-123.23.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

ресурсів, а звідти – для сільськогосподарських цілей і картографування.

Наскільки точний GPS?

Зараз приймачі GPS є дуже точними, і їхня точність в основному залежить від численних змінних, зокрема іоносфери, доступних супутників, міського середовища тощо. Існують деякі фактори, які перешкоджають точності GPS, як-от наведені нижче.

Фізичні перешкоди

Вимірювання часу прибуття може спотворюватись через великі масиви, такі як будівлі, гори, дерева тощо.

Атмосферні ефекти

На пристрої GPS переважно впливають сонячні бурі, сильний штормовий покрив, іоносферні затримки тощо.

Ефемериди

У супутника орбітальна модель може бути неточною, інакше застарілою, хоча це стає дедалі рідкісним явищем.

Чисельні прорахунки

Це може бути функція, якщо апаратне забезпечення пристрою не планується до умов.

Штучне втручання

Штучні перешкоди в основному включають підробку або пристрої GPS-перешкод. На відкритих місцях точність пристрою висока, без сусідніх великих будівель, які можуть перешкоджати сигналу. Отже, цей ефект називається міським каньйоном. Після того, як пристрій потрапить у великі будівлі, супутниковий сигнал спочатку може бути заблокований, потім відбитий від високої будівлі, де б він не зчитувався через пристрій, що призвело до помилок супутникової відстані.

На щастя, проблеми, з якими стикається технологія GPS, були визнані та наближаються до вирішення. Тут точність, що забезпечується високоякісними приймачами, є кращою, ніж точність рівня 2,2 метра, у 95% випадків і вища, ніж

					ВКРМ-123.23.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

точність рівня 3 метри, у 99% випадків.

Як GPS визначає положення

Робота системи глобального позиціонування базується на математичному принципі «трилатерації». Позиція визначається на основі вимірювань відстані до супутників. На малюнку видно, що чотири супутники використовуються для визначення положення приймача на землі. Місцезнаходження цілі підтверджено 4-м супутником. І три супутники використовуються для відстеження місця розташування.

Четвертий супутник використовується для підтвердження цільового розташування кожного з цих космічних апаратів. Глобальна система позиціонування складається із супутника, станції керування, станції спостереження та приймача. GPS-приймач отримує інформацію із супутника та використовує метод тріангуляції для визначення точного місцезнаходження користувача.

GPS використовується для деяких інцидентів кількома способами, наприклад:

- Визначити розташування позицій; наприклад, вам потрібно передати пілоту гелікоптера по радіо координати вашого місця розташування, щоб пілот міг вас підібрати.
- Для навігації з одного місця в інше; наприклад, вам потрібно проїхати від оглядового пункту до периметру пожежі.
- Створення оцифрованих карт; наприклад, вам призначено накреслити периметр пожежі та гарячі точки.
- Щоб визначити відстань між двома різними точками.

Інші системи GPS

У всьому світі доступні різні системи GPS, як-от GNSS (Глобальна навігаційна супутникова система. Система GNSS класифікується за чотирма типами, як-от GPS у США, ГЛОНАСС у Росії, Galileo у ЄС, BeiDou у Китаї. Крім того, є два регіональні такі системи, як QZSS Японії та IRNSS/NavIC Індії.

						ВКРМ-123.23.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата			25

Системні трекери GPS

Загалом існує чотири типи GPS-трекерів, де одні трекери використовуються для відстеження транспортних засобів, а інші – для спостереження за людьми.

Персональні трекери

Ці трекери в основному використовуються для спостереження за людьми/домашніми тваринами. Як правило, ці трекери використовують для роботи персональний пристрій, а саме кишеньковий чіп або браслет. Після цього пристрої будуть увімкнені. Після ввімкнення оператори можуть дистанційно розміщувати та відстежувати пристрій.

GPS-пристрої відстеження, які використовуються для собак, називаються нашійниками з GPS. Ці пристрої відіграють ключову роль у відстеженні домашніх тварин, наприклад собак. Таким чином, ці нашійники забезпечують спокій, оскільки власники можуть постійно відстежувати та розміщувати своїх собак.

Відстеження активів

Трекери активів, такі як крихітні радіочіпи чи великі супутникові мітки, використовуються для неавтомобільних предметів, наприклад особистих трекерів. Цей вид стеження використовується в супермаркетах, щоб зупинити крадіжку зі зломом. Тож новим рішенням для цього є відстеження активів.

Після використання цих трекерів можна зменшити крадіжки візків у супермаркетах. Крім того, деякі трекери збільшаться, розпізнаючи товари в кошику, зіставляючи їх із карткою лояльності покупця та надсилаючи це через рекламну команду!

Стільникове відстеження транспортних засобів

Цей вид відстеження можна здійснювати через супутникову/стільникову мережу, яка майже напевно є одним із найпоширеніших видів відстеження GPS. Цей вид відстеження часто використовується порівняно з супутниковим відстеженням.

					ВКРМ-123.23.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

Ця система використовує пристрій для збору даних з транспортного засобу, а потім повідомляє дані за допомогою веж стільникового зв'язку. Порівняно з супутниковим відстеженням цей вид відстеження транспортних засобів є менш дорогим і повідомляє швидше. Як правило, компанії-доставники використовують GPS-відстеження, щоб спростити робочий процес обслуговування клієнтів, наприклад, зателефонувати в транспортний засіб і запитати місце, щоб повідомити своїх клієнтів.

Супутникове відстеження транспортних засобів

Порівняно з супутниковим відстеженням транспортних засобів, системи відстеження на основі осередків не працюватимуть належним чином, оскільки вежі осередків зайняті. Супутникове відстеження є найкращим рішенням цієї проблеми, оскільки ці мережі можуть отримувати оновлення з найвіддаленіших районів.

Автотранспортна корпорація Аляски в основному охоплює віддалені райони. Головне завдання GPS-відстеження полягає в тому, щоб отримувати оновлення, навіть коли вежі стільникового зв'язку недоступні. Ці трекери забезпечують стабільні оновлення, які використовувалися водіями транспортних засобів, щоб викликати допомогу, коли їхні мобільні телефони не працювали.

Джерела помилок сигналу GPS

Сигнал GPS і його точність можуть залежати від таких факторів

Затримки в іоносфері та тропосфері

Супутникові сигнали повільні, тому що вони проходять через атмосферу. Таким чином, ця система використовує фіксовану модель для часткового виправлення помилки.

Багатопроменевість сигналу

Сигнал GPS може відбиватися від таких об'єктів, як великі скелі, високі будівлі, перш ніж досягти приймача, щоб збільшити час проходження сигналу та спричинити помилки.

					ВКРМ-123.23.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

Помилки CLK приймача

Вбудований GPS-годинник у приймач може містити невеликі помилки синхронізації, оскільки він є низькоточним порівняно з атомними годинниками на супутниках GPS.

Орбітальні помилки

Повідомлене місце супутника може бути неправильним.

Помітна кількість супутників

Точність в основному залежить від того, коли приймач GPS помічає кількість супутників. Якщо сигнал заблоковано, ви можете отримати помилки визначення місцезнаходження. Зазвичай пристрої GPS не працюють під землею, однак нові приймачі з високою чутливістю здатні відстежувати деякі сигнали в будівлях, які знаходяться під деревами.

Супутникова геометрія або затінення

Супутникові сигнали дуже ефективні, якщо супутники розташовані під широкими кутами, а не в тісній групі чи лінії.

Вибіркова доступність

Щойно Міністерство оборони США застосує SA (вибіркову доступність) до супутників, сигнали стануть менш точними, щоб підтримувати «ворогів» за допомогою сигналів GPS, які є надзвичайно точними. Щоб підвищити точність цивільних GPS-приймачів, у 2000 році уряд вимкнув Selective Availability, що підвищило точність цивільних GPS-приймачів.

Коди та послуги GPS

Кожен супутник GPS використовується для передачі двох сигналів з різними частотами, наприклад L1 і L2. Простий метод, як-от трилатерація, використовується для пошуку місцезнаходження, як-от довгота, широта та висота приймача GPS. Цей метод також використовується для вимірювання розташування неідентифікованої точки за допомогою трьох ідентифікованих точок

					ВКРМ-123.23.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		28

GPS коди

GPS-коди доступні двох типів, як-от нижче.

- C/A код або Coarse Acquisition Code.
- P-код або точний код.

Код C/A можна визначити як сигнал із частотою «L1» змінюється за допомогою псевдовипадкової серії бітів 1,023 Мбіт/с і використовується громадськістю. Подібним чином сигнал із частотою «L2» можна змінити за допомогою псевдовипадкової серії бітів 10,23 Мбіт/с, тому це відомо як точний код. Цей код в основному використовується у військових системах позиціонування. Зазвичай цей тип коду передається в зашифрованому форматі, який називається Y-кодом.

P-код забезпечує кращі вимірювання порівняно з грубим кодом збору даних, оскільки швидкість передачі даних цього коду вища, ніж швидкість передачі даних коду грубого збору даних.

Послуги GPS

Система GPS надає два типи послуг, як-от нижче.

- PPS або служба точного позиціонування.
- SPS або служба стандартного позиціонування.

Приймачі служби точного позиціонування завжди відстежують два коди, такі як код C/A та P-код, на обох сигналах із двома частотами, наприклад L1 та L2. У приймачі Y-код розшифровується для отримання P-коду, тоді як приймачі SPS відстежують просто грубий код збору сигналу з L1.

Використання GPS-приймача

У системі GPS існує просто односторонній зв'язок від супутника до споживачів. Отже, кожному користувачеві не потрібен передавач, а просто приймач GPS. Здебільшого він використовується для виявлення точного місцезнаходження об'єкта. Він виконує це завдання за допомогою сигналів, отриманих із супутників. Блок-схема приймача GPS показана нижче, де кожна функція блоку присутня в приймачі, яка зазначена нижче.

					ВКРМ-123.23.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

Приймальна антена

Ця антена приймає супутникові сигнали, і в основному це антена з круговою поляризацією.

МШУ (підсилювач з низьким рівнем шуму)

Цей вид підсилювача посилює слабкий отриманий сигнал

Понижуючий конвертер

Цей вид перетворювача змінює частоту сигналу, який приймається, на сигнал ПЧ (проміжної частоти).

Підсилювач ПЧ

Цей вид підсилювача використовується для зміни сигналу ПЧ (проміжної частоти).

АЦП

Аналого-цифровий перетворювач використовується для перетворення сигналу з аналогового в цифровий. Проаналізуйте два блоки, а саме дискретизацію, а також квантування, які присутні в аналого-цифровому перетворювачі.

DSP

Цифровий сигнальний процесор створює грубий код збору даних.

Мікропроцесор

Мікропроцесор виконує обчислення позиції та надає синхронізацію сигналу для керування процесом додавання цифрових блоків. Він передає корисні дані на дисплей, щоб відобразити їх на дисплеї.

Існує кілька різних моделей і типів приймачів GPS. Під час роботи з GPS-приймачем важливо мати:

- Компас і карта.
- Завантажений кабель GPS.
- Деякі додаткові батареї.
- Знання про обсяг пам'яті приймача GPS для запобігання втраті даних, зменшення неточності даних або інших проблем.

					ВКРМ-123.23.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

- По можливості використовуйте зовнішню антену, особливо під кронами дерев, у каньйонах або під час руху.
- Налаштувати GPS-приймач відповідно до інциденту або нормативних актів агентства; система координат.
- Примітки, які описують, що ви зберігаєте в приймачі.

Помилка GPS

Існує багато джерел можливих помилок, які погіршують точність позицій, обчислених приймачем GPS. Час подорожі супутникових сигналів GPS може змінюватися атмосферними впливами; Коли сигнал GPS проходить через іоносферу та тропосферу, він заломлюється, внаслідок чого швидкість сигналу відрізняється від швидкості сигналу GPS у космосі. Іншим джерелом помилок є шум або спотворення сигналу, що спричиняє електричні перешкоди або помилки, властиві самому приймачу GPS.

Інформація про супутникові орбіти також спричинить помилки у визначенні позицій, оскільки супутники насправді не там, де «думав» приймач GPS на основі інформації, яку він отримав під час визначення позицій. Невеликі відхилення в атомних годинниках на борту супутників можуть призвести до великих помилок позиціонування; помилка годинника в 1 наносекунду означає помилку користувача на землі в 1 фут або 3 метри.

Ефект багатопроменевого поширення виникає, коли сигнали, що передаються від супутників, відбиваються від поверхні, що відбиває, перш ніж потрапити на антену приймача. Під час цього процесу приймач отримує сигнал у прямолінійному тракті, а також у затриманому тракті (кілька шляхів). Ефект схожий на привид або подвійне зображення на телевізорі.

Геометричне зменшення точності (GDOP)

Геометрія супутника також може впливати на точність позиціонування GPS. Цей ефект називається зменшенням геометричної точності (GDOP). Це стосується розташування супутників один відносно одного та є показником якості конфігурації супутника. Він може змінювати інші помилки GPS. Більшість

					ВКРМ-123.23.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		31

сигналами, тому вони не працюють у приміщенні, під водою, під деревами тощо.

– Найвища точність вимагає прямої видимості від приймача до супутника, тому GPS не дуже добре працює в міських умовах.

Використання системи GPS

Системи GPS дуже гнучкі, і ми можемо знайти цю систему в будь-якому секторі промисловості. Зараз GPS відіграє ключову роль у картографуванні лісів; допомагати фермерам збирати врожай на їхніх полях, а пілоти використовують літаки для навігації на землі, інакше в атмосфері. Ці системи є важливими частинами військового застосування та для екстрених екіпажів, щоб знайти людей, які потребують допомоги. Ці технології часто працюють у кількох регіонах, які ми зазвичай не розглядаємо.

Загалом системи GPS діляться на п'ять основних категорій, які включають наступне.

- Розташування використовується для визначення місця розташування.
- Навігація використовується, щоб дістатися з одного місця в інше.
- Відстеження використовується для спостереження за рухом об'єкта в іншому випадку.
- Картографування використовується для створення карт земної кулі.
- Хронометраж – це принести точний час на земну кулю.

У кожному разі використання GPS в основному залежить від вимірювання відстані до людей від багатьох супутників. GPS – це просто одна з різних груп супутників, які використовуються для визначення місця розташування. Чотири основні групи супутників, які використовуються у всьому світі, це GPS, ГЛОНАСС, Galileo та BeiDou. Ці технології в основному використовують сигнали за допомогою цих супутників для визначення відстані людей від усіх цих супутників. Отже, ці вимірювання можуть визначити, де б людина не перебувала у світі, і як знайти шлях до іншої позиції.

Таким чином, це все про огляд системи GPS, роботу, компоненти, переваги, недоліки та її застосування. Система GPS в основному

					ВКРМ-123.23.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		33

використовується для визначення надзвичайно точного часу за допомогою рубідієвих годинників над кожним супутником. Ці годинники дозволяють супутникам розпізнавати своє точне місцезнаходження в дуже точний час. Інформацію про час можна використовувати в різноманітних додатках, таких як точне землеробство, морська автономна гідрографія та гідрографія транспортних засобів.

3.2 Розробка структурної схеми

На рисунку 3.1 представлена структурна схема роботи системи. Основних сторінок (Main Page Basics) у навігаторі усього шість:

- Супутники.
- Карта.
- Навігація.
- Висотомір.
- Шляховий комп'ютер.
- Головне меню.

Звичайно є кнопка Page для перемикання між основними сторінками. Вибравши одну з Основних сторінок, можна викликати сторінки нижніх рівнів, призначені для відображення інформації в іншому виді або для налаштування основної сторінки. Завдяки такій простій і логічній організації подання даних, розібратися в роботі будь-якої моделі навігатора можна легко й швидко.

Сторінка «СУПУТНИКИ» («SATELLITE»)

Дуже важлива сторінка. Дозволяє в умовах затінення (якщо ви перебуваєте в місті, у горах або в густому лісі) правильно зорієнтуватися й вибрати потрібну позицію для поліпшення прийому. Показує розташування супутників на небі в цей момент, силу сигналу від кожного, що течуть координати й точність їхнього визначення.

					ВКРМ-123.23.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		34

Ще раз нагадаємо, що для визначення координат приймач повинен одержати стійкий сигнал мінімум від трьох супутників.

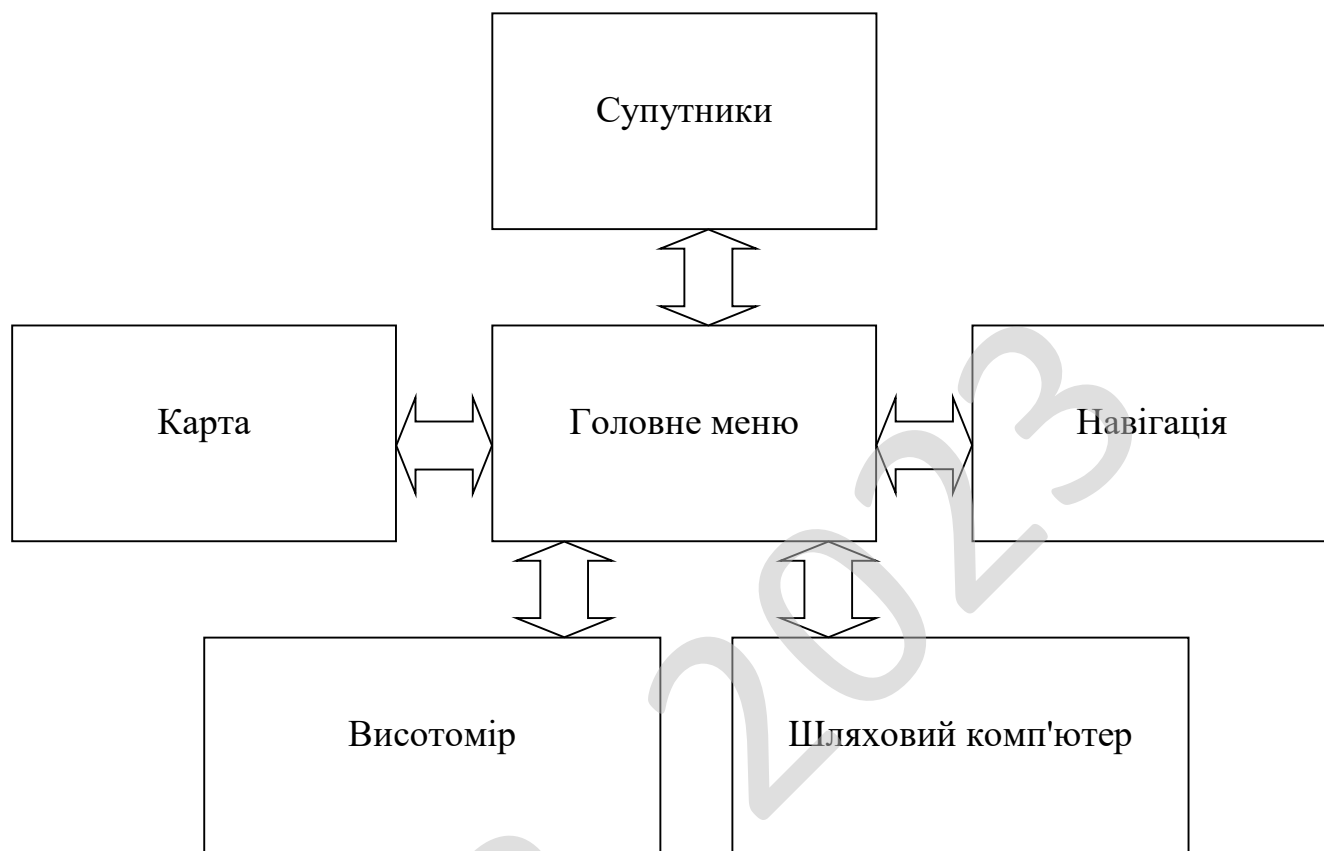


Рисунок 3.1 – Структурна схема системи

Сторінка «КАРТА» («MAP»)

У тому або іншому виді присутня у будь-якому навігаторі. На ній відображаються ваші Шляхові точки, Шляхи й Маршрути. Масштаб карти може мінятися від сотень км/см до одиниць м/см. Для зміни масштабу існують спеціальні кнопки. У навігаторах, що мають можливість завантаження електронних карт, на сторінці Карта відображаються електронні карти місцевості. На жаль, на електронних картах не всі ділянки місцевості однаково пророблені у всіх масштабах. Дуже часто зустрічаються «провали». Доцільно ретельно тестувати електронні карти у місцях, що вас цікавлять.

Сторінка «НАВІГАЦІЯ» («NAVIGATION»)

Основна сторінка при використанні приладу. Навігатори бувають дорогі і дешеві, прості і складні, але такий екран є в кожному. Як уже говорилося, головний і найбільше часто використовуваний режим навігації – це рух до наміченої точки. Даний режим наочніше всього реалізується на цій сторінці.

Сторінка «ВИСОТОМІР» («ALTIMETER»)

Висоту, щоправда, дуже приблизно, визначає будь-який навігатор, що «бачить» не менш 4-х супутників. Висота обчислюється відповідно до тієї моделі Землі, що обрана в початкових установах навігатора. Для точного виміру висоти служить Альтиметр (Барометричний висотомір).

Ця функція є не у всіх навігаторів, тільки в дорогих моделях. Принцип роботи – у визначенні атмосферного тиску й перерахування його у висоту над рівнем моря або над рівнем поверхні землі. Зрозуміло, що Висотомір має потребу в калібруванні перед кожним його використанням. Якщо атмосферний тиск міняється по яких-небудь погодних причинах, відповідно до зміни тиску будуть змінюватися й показання Висотоміра. Так як Висотомір по своїй суті є барометром, то з його допомогою корисно стежити за зміною атмосферного тиску для прогнозування зміни погоди.

Сторінка «ШЛЯХОВИЙ КОМП'ЮТЕР» («TRIP COMPUTER»)

Дуже цікава й корисна навігаційна функція. У найпростіших приладах як окрема сторінка відсутня, але основні елементи є на сторінці Навігації.

Шляховий комп'ютер визначає безліч параметрів вашого руху й видає їх на екран у зручному для вас виді. Цих параметрів може бути дуже багато, і кожний може для себе вибрати ті, які йому потрібні в цей момент. Перелічу деякі з них:

- Пройдена відстань.
- Час у русі.
- Час стоянок.
- Передбачуваний час прибуття в намічену точку.

- Поточна швидкість.
- Середня швидкість.
- Середня швидкість переміщення по всьому маршруті з урахуванням стоянок.
- Максимальна швидкість.
- Відстань до поточної точки.
- Відстань до кінця маршруту.
- Курс.
- Відхилення від курсу.
- Інші.

Сторінка «ГОЛОВНЕ МЕНЮ» («MAIN MENU»)

Як і сторінка Навігація, є в кожному GPS-приймачі. Із цієї сторінки можна викликати будь-яку функцію навігатора, там же зосереджені основні налаштування.

Підміну розділів Головного меню позначені піктограмами, які у всіх навігаторах виглядають приблизно однаково.

Відзначити (Mark) – дозволяє відзначити ваше місце розташування у вигляді шляхової точки. Кожній шляховій точці автоматично привласнюється тризначний номер. Ви можете відредагувати назву, замінивши номер ім'ям і призначивши точці який-небудь значок у вигляді піктограм (будинку, табору, готелю й т.д.).

Пошук (Find). Як уже було сказано, у пам'яті навігатора можна зберігати кілька сотень шляхових точок і цей режим допомагає швидко орієнтуватися у всьому їхньому великому списку.

Маршрути (Routes). У цьому режимі ви можете створювати й редагувати потрібні вам маршрути з наявних у пам'яті вашого приймача шляхових точок.

Шляхи (Tracks). Тут ви зберігаєте шляхи ваших переміщень, активізуєте режим руху уздовж обраного шляху, здійснюєте необхідні налаштування.

Налаштування (Setup). Всі загальні установки приладу перебувають у цьому підменю. Ви задаєте одиниці виміру навігаційних параметрів (відстань, швидкість, кути й т.д.), формат подання географічних координат, установлюється формат часу, налаштовуєте екран дисплея, робите всі системні установки.

Додатка (Accessories). Тут можуть бути зібрані додаткові функції приладу, безпосередньо не пов'язані з навігацією. Наприклад, у навігатора Garmin Etrex Vista – це дані про розташування й рух Сонця й Місяця (при активізації нагадує мініатюрний планетарій – дуже цікаво). До додаткових функцій також відносяться щоденник-календар, калькулятор, інформатор про найбільш сприятливий час для полювання й риболовлі в місці вашого розташування (дуже сумнівна, по-моєму, функція). Іноді в Додатки включають які-небудь іграшки й ще щось, що не має істотного значення.

Принцип роботи та склад обладнання системи керування позиціонуванням GPS. Виробник gps-трекера порту obd-II

Система керування GPS-позиціонуванням автомобіля в основному визначається за допомогою GPS автомобіля, і транспортний засіб призначений для керування та відстеження транспортного засобу в поєднанні з системою бездротового зв'язку. Було розроблено систему сигналізації глобального позиціонування транспортних засобів, поліцейські GPS-командні системи тощо. Він використовується для керування розкладом руху міських автобусів, сигналізації та планування руху транспортних засобів і суден у ландшафтній туристичній зоні, а також для планування та моніторингу транспортних засобів і суден від митниці, громадської безпеки, берегової оборони та інших відділів. Основними функціями центру моніторингу є:

1. Функція відстеження даних. Показати положення мобільного автомобіля в реальному часі в списку цнотливості. Наприклад, номер автомобіля, довгота, швидкість, напрямок, час, дата тощо.

Функція стеження за картинкою. Відображення інформації про місцезнаходження мобільного транспортного засобу на фоні відповідної

					ВКРМ-123.23.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		38

електронної наземної (морської) карти. Електронну наземну (морську) схему можна збільшувати, зменшувати, відновлювати, перемикати. Є два способи отримання: звичайне отримання та випадкові імена. Ви також можете надати функцію вибору траєкторії руху автомобіля.

3. Функція відображення моделювання. Увійдіть у комп'ютер і відобразіть відомості про місцезнаходження цілі. Функція команди рішення. Команди прийняття рішення зв'язуються з мобільними транспортними засобами за допомогою методу зв'язку. Метод зв'язку можна запланувати за допомогою тексту, коду, голосу тощо для досягнення команди планування.

4. Основні функції автомобільної частини включають:

(1) Функція надсилання інформації про позиціонування. GPS-приймач визначає місцезнаходження в режимі реального часу та надсилає інформацію про місцезнаходження в центр моніторингу через радіо.

(2) Функція відображення даних. Покажіть на дисплеї поточне положення вашого транспортного засобу. Такі як довгота, широта, швидкість і напрямок.

(3) Функція отримання команди планування. Отримайте центр моніторингу, щоб надіслати команду для відображення або видачі голосу на дисплеї.

(4) Функція сигналізації. У разі виникнення надзвичайної ситуації водій запускає сигналізацію, і центр моніторингу негайно показує інформацію про транспортний засіб, місце аварії, персонал автомобіля та іншу інформацію.

5. GPS-позиціонування автомобіля – це динамічне навігаційне позиціонування в одній точці. Його точність позиціонування становить близько 100 м зоряної величини. Щоб підвищити точність позиціонування, можна використовувати технологію диференціального GPS.

3.3 Розробка функціональної схеми

На рисунку 3.2 показана функціональна схема роботи системи. У ній можна чітко простежити як працює приймач GPS і розроблена програма.

Через розроблений інтерфейс програми, при правильному налаштуванні й запуску можна побачити наступні геопараметричні дані:

- геопараметрична широта;
- геопараметрична довгота;
- кут підйому;
- кут схилу;
- довжина маршруту;
- поточний час із виправленням;
- максимальна висота;
- швидкість розгону 0-100 km/h;
- швидкість розгону 80-120 km/h;
- загальний час руху;
- висота над рівнем моря;
- сторони світа, куди спрямований мобільний пристрій;
- поточний напрямок (сторона світу);
- година сходу сонця;
- час заходу сонця;
- прискорення при русі;
- горизонтальна швидкість при русі;
- вертикальна швидкість при русі.

Розглянемо схему зверху долілиць. Штучні супутники Землі подають сигнал на геопараметричну поверхню (землю) сигнал, мобільний пристрій з GPS приймачем через апаратну частину й операційну систему приймають цей сигнал у розроблену програму по запиту користувача.

					ВКРМ-123.23.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		40

У розробленої програми існують наступні можливості:

– Підтримка протоколів обміну навігаційною інформацією NMEA, SiRF binary і Garmin.

– Підтримка роботи із треками.

– Підтримка змінних скінів (візуальних оболонок програми).

– Фільтр, що налаштовується, відображення POI (точок інтересу – об'єктів різних типів, наприклад, заправлень, кафе, пересувними й стаціонарними камерами спостереження й т.д.).

– Можливість завантажувати сторонні бази Speedcam (радары, пересувні й стаціонарні камери спостереження за швидкістю, обмеження швидкості, попередження про погане дорожнє покриття, інше).

– Пошук по адресах і об'єктах на карті.

– Можливість побудови маршруту з необмеженою кількістю проміжних точок.

– Інтерфейс, що налаштовується, (можна відключати або включати кнопки керування в програмі).

– Підтримка сторонніх пакетів голосових підказок, можливість створювати власні.

– Підтримка тривимірних розв'язок

– Можливість підключення сервісу інформації про пробки.

Час «старту» GPS навігатора

Час «старту» необхідний навігаційному приймачу на визначення позиції після включення, залежить від наявної в пам'яті початкової інформації.

Виділяються наступні режими:

– «Холодний» старт («автопошук») – час, позиція, альманах і ефемериди невідомі.

– «Теплий» старт – позиція й ефемериди невідомі, час і альманах відомі.

– «Гарячий» старт («перезахват») – альманах, ефемериди відомі, час і позиція відомі з деякою помилкою.

					ВКРМ-123.23.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		42

Навігаційні повідомлення передані із супутників містять два типи даних – ефемериди й альманах супутників. В альманасі передаються параметри орбіти, за допомогою яких можна обчислити зразкове місце розташування супутників з достатньо великим ступенем погрішності. Альманах, що зберігається в пам'яті приймача, постійно оновлюється, тому що кожний супутник передає дані альманаху для всіх супутників угруповання. Час «життя» альманаху становить 2-3 місяці. Далі, величина накопиченої помилки в розрахунках буде неприпустимою.

Дані ефемерид містять параметри, що дозволяють більш точно обчислити поточне місце розташування супутників. На відміну від альманаху, кожний із супутників передає, тільки свої власні ефемериди. Час «життя» ефемерид не перевищує 4-6 годин.

Інформація даних ефемерид і альманаху, передана із супутників, постійно коректується. Це відбувається один раз у добу. Мережа наземних станцій, одержує інформацію із супутників, за аналогією зі звичайними користувачами, аналізує виміри, порівнює їх з опорними, розраховує коригувальні виправлення й передає їх на головну станцію, з якої здійснюється передача даних на супутники.

«Холодний» старт приймача може бути зв'язаний не тільки з його тривалою бездіяльністю, але переміщенням на велику відстань у виключеному стані.

Якщо перший випадок пов'язаний із застарілим альманахом і помилкою у визначенні поточного точного часу, то в другому випадку приймач, не знаючи про своє переміщення, буде намагатися знайти супутники, яких повинні бути видимі на «старому» місці. Користувач може «допомогти» приймачу й зменшити час «холодного» старту, указавши на базовій карті, зразкове «нове» місце розташування. Під час «холодного» старту приймач сканує весь діапазон можливих значень частот і тимчасових затримок навігаційних сигналів. При цьому, у багатоканальних приймачах, кілька каналів можуть використовуватися для пошуку одного супутника, щоб прискорити час його захвата. Після того, як

сигнал хоча б від одного супутника буде отриманий і розібраний, приймач буде мати повну інформацію про альманах всього угруповання й, по суті, перейде до «теплого» старту

При «теплому» старті, приймач, включений після 6-ї годин бездіяльності, почне «пошук» сигналів супутників, використовуючи значення поточного часу й дані, що зберігаються в пам'яті, альманаху. Буде здійснюватися пошук тільки тих супутників, які, по теоретичних розрахунках перебувають у видимій півкулі й повинні бути доступні приймачу. Відповідно, відомий досить вузький діапазон частот і часових затримок, що потрібно просканувати у процесі пошуку сигналів. Ця інформація істотно прискорює час захвата супутників, у порівнянні з «холодним» стартом, коли пошук ведеться на широкому діапазоні всіх можливих значень затримок і частот

Варто відзначити, що в момент включення, багатоканального приймача починає пошук сигналів з декількох супутників одночасно. Інформація передана із супутників прив'язана до єдиної шкали часу, містить однакову структуру й досягає антени приймача, приблизно в один й той же час. Тому дані ефемерид, одночасно захоплених супутників, надійдуть у приймач майже що одночасно. Якщо кількість таких супутників більше або рівняється трьом, то це дозволяє приймачу відразу ж розрахувати позицію. У випадку, коли сигнали блокуються перешкодами, то може знадобитися досить тривалий час на визначення позиції.

Наявність повністю отриманих ефемерид, не гарантує використання цього супутника у підрахунку позиції. Інформація передана в ефемеридах може бути неправильною, помилковою, або пов'язаною з несправністю в роботі супутника. Це може бути зв'язано не тільки з несправністю супутника, але й діагностичними роботами проведеними на його борту, процесом уведення його в експлуатацію або тестуванням нових режимів.

«Гарячий» старт пов'язаний з короткочасним вимиканням приймача (до 6-ї годин) не вимагає тривалого часу на визначення позиції. Це пояснюється тим, що отримані раніше ефемериди містять «свіжі» дані, використовувані для

					ВКРМ-123.23.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		44

визначення точних координат супутників і можуть використовувати в обчисленні позиції. У випадку включення приладу після граничного часу, ефемериди розглядаються застарілими й починає діяти принцип «теплого» старту. Якщо на момент включення приймача видимими залишилися менш 3-х супутників з «свіжими» ефемеридами, то для визначення позиції буде потрібно якийсь час на збір даних ефемерид нового супутника.

Дані ефемерид передаються в складі трьох пакетів. Кожний з пакетів містить однаковий часовий ідентифікатор (IOD – issue of data) по якому можна об'єднати загальну інформацію. Інформація ефемерид передана із супутників кожні 30 секунд, змінюється раз в 2 години, і містить однаковий на цей час IOD. Якщо один з пакетів був пропущений, або отриманий з помилками, то можна виділити аналогічний пакет з наступного повідомлення, перевірити його ідентифікатор і не чекаючи наступних пакетів, використовувати його з раніше отриманими. Це дозволяє приймачу прискорити час «старту».

Існує мінімальний можливий час, необхідний приймачу на «старт», і це визначається структурою переданого сигналу із супутників. Виробники навігаційної апаратури, використовуючи стандартні методи навігації, можуть наблизитися до цього часу, але зменшити його не зможуть. Одним з методів, призначених для рішення цієї проблеми, є Assisted-GPS (A-GPS). Його принцип полягає в обчислення точного місця розташування супутників без інформації ефемерид, на одержанні яких потрібен час. Обчислення здійснюється на використанні точних моделей орбіт супутників, доступних через спеціальні Інтернет – сервіси.

З іншого боку, максимальний час «старту» може значно перевищувати заявлене в технічній специфікації на навігатор час. Це пояснюється навколишніми умовами, у яких відбувається «захват» супутників і «старт» приймача. Якщо приймач перебуває в умовах сильних фізичних перешкод, то навігаційний сигнал піддається зовнішньому впливу, містить помилки й неправильно декодується. Більше, того геометричний фактор цих супутників, що

					ВКРМ-123.23.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		45

є одним із критеріїв точно визначення позиції, сильно погіршується. Всі ці умови можуть значно збільшити час «старту» приймача.

Обчислення радіуса окружності помилки для оцінки точності GPS-вимірів

Показник CEP – окружність можливої помилки (Circular Error of Probability) один з можливих шляхів оцінити точність вироблених GPS вимірів у даній точці тепер. Завдяки великій кількості факторів зовнішнього середовища впливаючих на виміри – в одній точці показання приладу будуть різними в різні моменти часу. До таких факторів відносяться вплив іоносфери, вплив нижніх шарів атмосфери, багатопроменевість, наявність перешкод на шляху сигналу. Показник CEP використовує опорну точку, або задаваємо користувачем, або що обчислюється як середнє геометричне між всіма вимірами, для того, щоб побудувати серію окружностей що показують відповідно 50, 90, 95, 99% можливої помилки.

Для того, щоб визначити CEP повинна бути взята серія вимірів зроблена в одній точці. Наприклад, включений і нерухливий GPS з інтервалом в 2-5 сік реєструє точки треку, які потім завантажуються, конвертуються в share-файл і аналізуються.

Очевидна регулярність розташування точок пов'язана з розрішенням цифрових значень видаваних GPS. Наприклад точність із якої GPS Garmin 12 видає координати – 0.000005 десяткових градусів по довготі, і 0.000005 по широті. Для обчислення CEP дані повинні бути спроектовані. Для обчислення виміряються відстані між середньою точкою й кожним виміром, а потім вираховується на якій відстані перебуває потрібний відсоток точок. Результати обчислення CEP 4 різних окружностей: Average = 6.999e+006 5.82936e+006, SD = 7.00012e+006 5.83025e+006, Circular Error Probabilities (CEP), 50% = 3.42281, 90% = 7.36774, 95% = 9.52791, 98% = 14.2946.

					ВКРМ-123.23.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		46

Приклад показує, що 50% точок перебувають на відстані 3.4 метра від середнього значення, 98% точок на відстані 14.2 метра від середнього. З діаграми також видний розкид помилки.

Розроблена програма є універсальною системою, що може використовуватися у всіх мобільних пристроях, підтримуючих систему навігації GPS і маючих віконний інтерфейс на основі операційної системи Windows mobile. Коли користувач налаштовує й запускає розроблену програму, відбувається пошук штучних супутників Землі для визначення координат. Сучасні приймачі GPS вбудовані в мобільні пристрої одночасно можуть приймати дані з 12 супутників, що забезпечує високу точність позиціонування (50 – 100 метрів). На схемі представлена геопараметрична сітка з поточним місцем розташування об'єкта й система супутників, що забезпечує визначення місця розташування. При поганому зв'язку із супутниками (знаходження в приміщеннях) необхідно, принаймні 3 супутники, що знаходяться не на одній лінії для визначення поточного положення.

Розглянувши усі блоки функціональної схеми перейдемо до розгляду діаграми взаємодії процесів, які відбуваються у системі.

3.4 Розробка діаграми процесів

Діаграма взаємодії процесів системи, розробленої у результаті виконання дипломного проектування, наведена на рисунку 3.3.

Після початку роботи розробленого ПЗ ми потрапляємо до головного блоку системи звідки через ланку дій відбувається наступне:

- Основний блок ПЗ.
- Модуль захисту ПЗ.
- Інтерфейс розробленого ПЗ.
- Бібліотеки налаштування та взаємодії з ARM Cortex-A15 MPCore.
- Моніторинг GPS підсистеми.

					ВКРМ-123.23.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		47

- Авторська форма ПЗ.
- Взаємодія з Google Maps.
- Налаштування ПЗ.
- Отримання та обробка первинних даних.
- Обробка та виведення розрахункових даних.

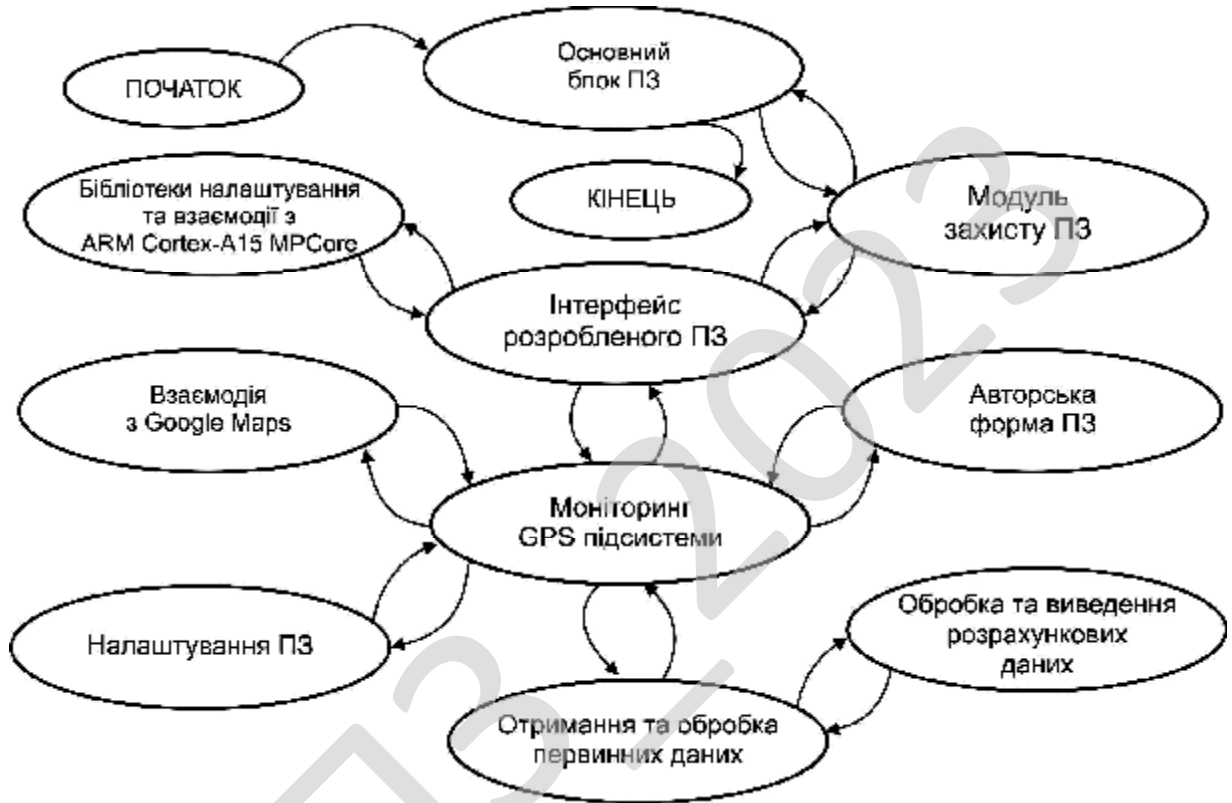


Рисунок 3.3 – Діаграма взаємодії процесів

Таким чином, розглянувши опис системи, структурну, функціональну схеми системи, та діаграму взаємодії процесів перейдемо до опису блок-схем основної програми, та підпрограм, які використовуються, для реалізації системи.

4 РЕАЛІЗАЦІЯ ПРОЕКТУ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ПРАВИЛЬНІСТЬ ПРОЕКТНИХ РІШЕНЬ

4.1 Блок-схеми та опис алгоритмів функціонування системи

Первинною стадією без якої не відбувається розробка програмного забезпечення це звичайно розробка блок-схем. На рисунку 4.1 зображена основна блок-схема програми, на рисунку 4.2 зображено роботу підпрограми.

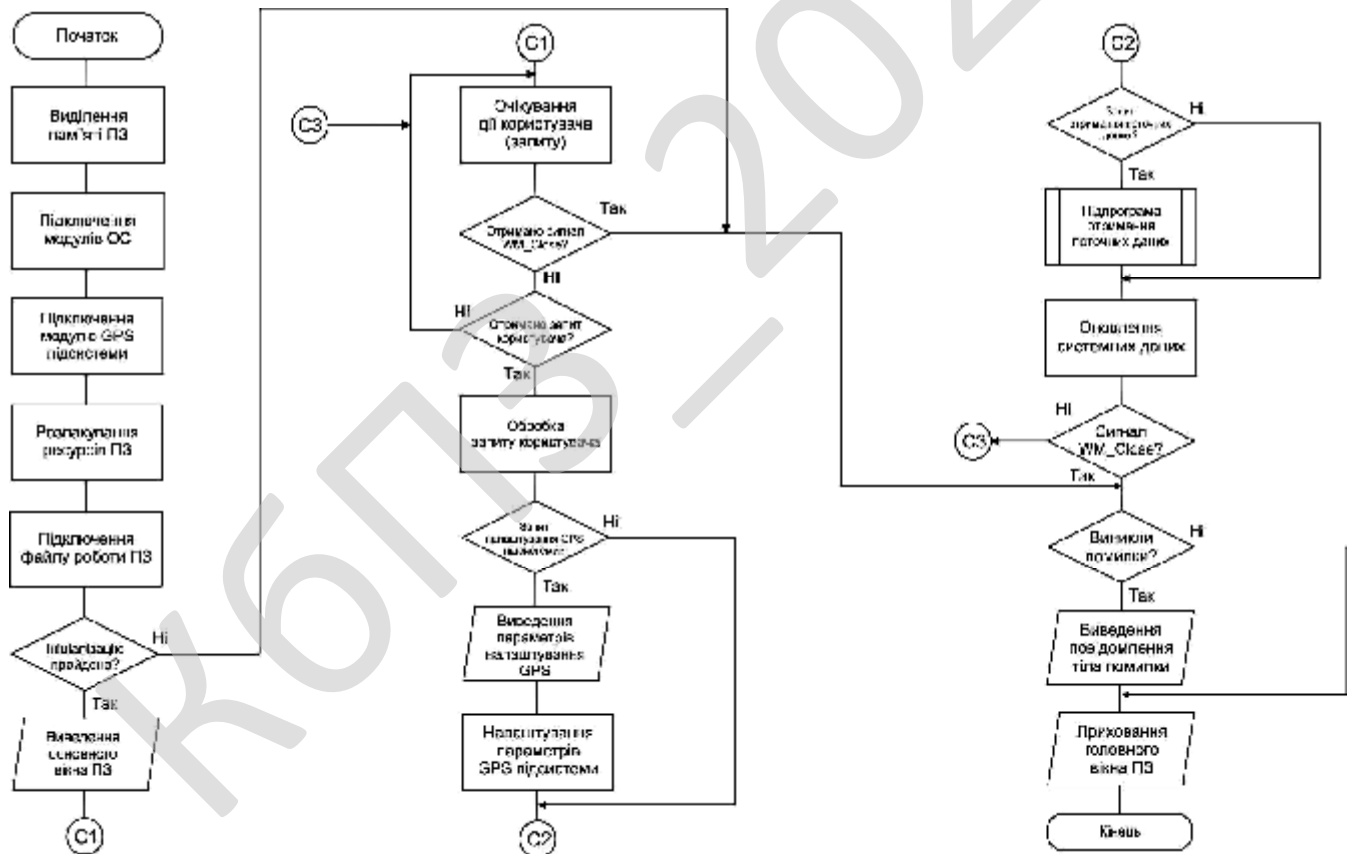


Рисунок 4.1 – Блок схема основної програми

З якої видно що робота основної програми складається з початкових етапів ініціалізації ПЗ, перевірки наявності ресурсів системи, блоку початку основного

циклу з чеканням запиту від користувача в якому відбувається виклик підпрограми та останньої стадії – перевірка поточного стану з завершенням роботи розробленого ПЗ.

При роботі підпрограми виконується основний функціонал системи з циклічними послідовностями, перевіркою поточного стану та поверненням в основну програму прапорів стану виконання.

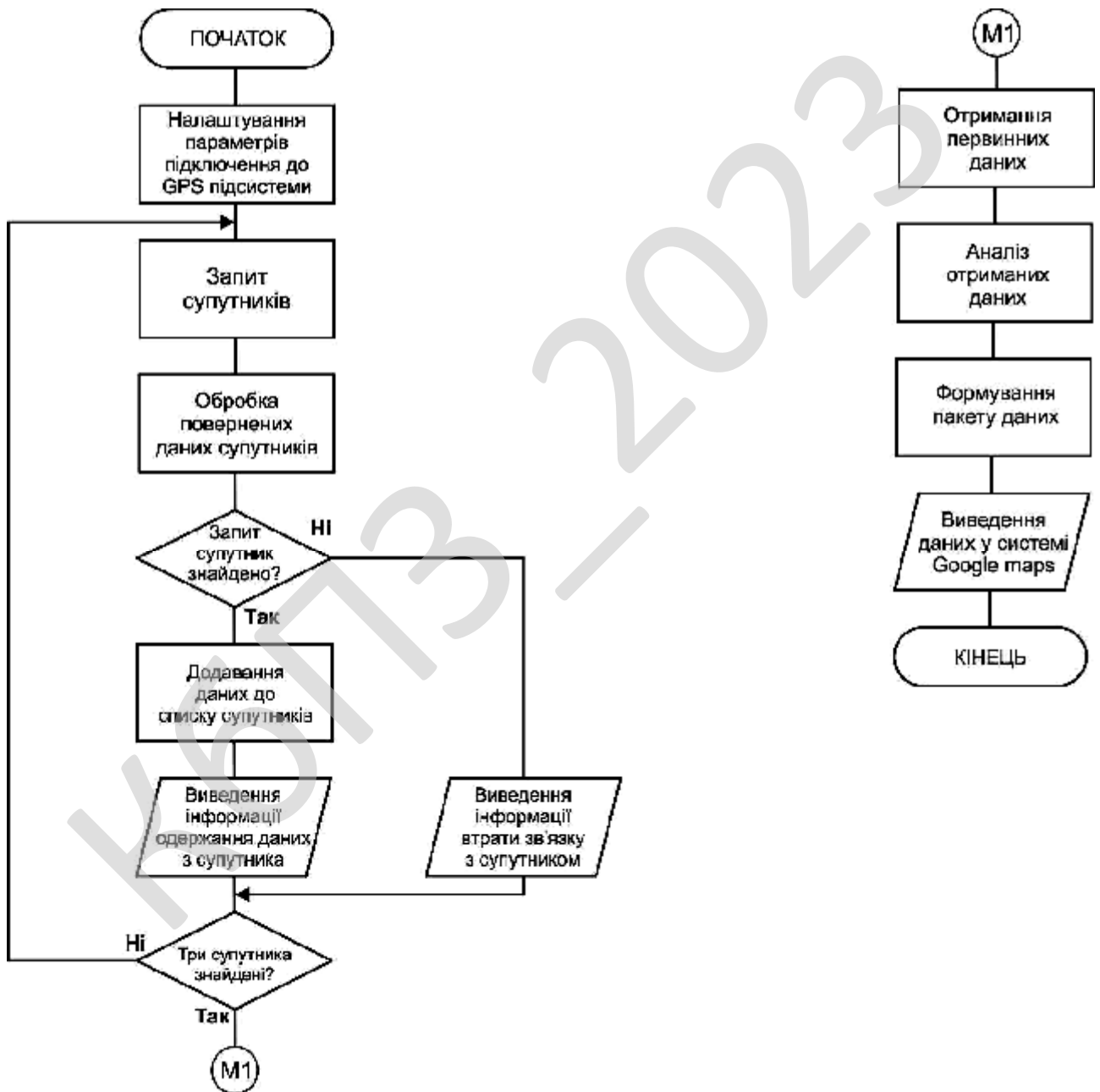


Рисунок 4.2 – Блок схема підпрограм

Блок-схеми є першоджерелами стратегії розвитку ПЗ. Тому від точності і детальної блок-схеми залежить результат всієї програми. При виборі початкової точки відліку при побудові схем я враховував, що виходячи з вибору мови програмування і інших технічних засобів, програма буде об'єктно-орієнтована що вимагає оптимізації програми високого рівня, також те, що при розробці програми слід надати особливу увагу модулю GPS захвату.

Крім цього було використано підходи з використанням UML, це уніфікована мова моделювання, використовується у парадигмі об'єктно-орієнтованого програмування. Є невід'ємною частиною уніфікованого процесу розробки програмного забезпечення. UML є мовою широкого профілю, це відкритий стандарт, що використовує графічні позначення для створення абстрактної моделі системи, називаної UML-моделлю.

UML був створений для визначення, візуалізації, проектування й документування в основному програмних систем. UML не є мовою програмування, але в засобах виконання UML-моделей як інтерпретованого коду можлива кодогенерація.

UML може бути застосовано на всіх етапах життєвого циклу аналізу бізнес-систем і розробки прикладних програм. Різні види діаграм які підтримуються UML, і найбагатший набір можливостей представлення певних аспектів системи робить UML універсальним засобом опису як програмних, так і ділових систем.

Діаграми дають можливість представити систему (як ділову, так і програмну) у такому вигляді, щоб її можна було легко перевести в програмний код. Основною причиною використання мови UML є спілкування розробників між собою. Крім того, UML спеціально створювалася для оптимізації процесу розробки програмних систем, що дозволяє збільшити ефективність їх реалізації у кілька разів і помітно поліпшити якість кінцевого продукту.

UML прекрасно зарекомендувала себе в багатьох успішних програмних проектах. Засоби автоматичної генерації кодів дозволяють перетворювати моделі

					ВКРМ-123.23.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		51

мовою UML у вихідний код об'єктно-орієнтованих мов програмування, що ще більш прискорює процес розробки. Практично усі CASE-засоби (програми автоматизації процесу аналізу і проектування) мають підтримку UML. Моделі розроблені в UML, дозволяють значно спростити процес кодування і направити зусилля програмістів безпосередньо на реалізацію системи.

Діаграми підвищують супроводжуваність проекту і полегшують розробку документації.

UML необхідний:

– Керівникам проектів, які керують розподілом завдань і контролем за проектом.

– Проектувальникам інформаційних систем які розробляють технічні завдання для програмістів.

– Бізнес-аналітикам, які досліджують реальну систему і здійснюють інжиніринг і реінжиніринг бізнесу компанії.

– Програмістам які реалізують модулі інформаційної системи.

При модифікації системи об'єктний підхід дозволяє легко включати в систему нові об'єкти і виключати застарілі без істотної зміни її життєздатності. Використання побудованої моделі при модифікаціях системи дає можливість усунути небажані наслідки змін, оскільки вони не ламають структури системи, а тільки змінюють поведінку об'єктів.

При складанні блок-схем програмного забезпечення і напрацювання алгоритмів я зіткнувся з масою проблем, які вимагали напрацювання процедур і функцій над основною проблематикою. Для чого були створені додаткові класи, типи даних і константи, що забезпечило вирішення проблем.

При розробці програми я штовхнувся із проблемою – зберігання даних. Необхідно дуже швидко передавати та обробляти дані та їх зберігати, але існуючі апаратні ресурси GPS навігатора не дозволяють це робити.

Для усунення цих недоліків було реалізовано бібліотеку обробки та стиску даних. Розглянемо реалізацію бібліотеки.

					ВКРМ-123.23.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		52

Для того, що б архівувати папку, необхідно спочатку склеювати всі файли, що перебувають у деякій папці. Для цього був розроблений наступний формат:

- 1-3 байт сигнатура, що позначає, що це файл необхідного формату;
- 4-7 байт кількість файлів у склеєному файлі;
- 8-N байт масив структур, описуючий всі файли в склеєному файлі, з наступною структурою: 2 байти довжина ім'я файлу, 4 байти – розмір файлу.

У даній структурі максимальний розмір одного файлу може бути максимум 2^{32} байт.

Розглянемо функцію, розпакування файлів.

```
Function DFolder(FolderPath,ArchivePath:string;
                DeActionFunction:TActionFuntion):boolean;
var
begin
    // перевірка формату
    DATA:=CreateFile(pchar(ArchivePath),GENERIC_READ+GENERIC_WRITE,FILE_SHARE_READ,0,OPEN_EXISTING,0,0);
    ReadFile(DATA,value,4,_readed,0);
    {кількість файлів}
    ReadFile(DATA,_Count,4,_readed,0);
    for i:=1 to _Count do
        begin
            path_in_archive_file := Read_LS(DATA);
            ReadFile(DATA,CurrFileSize,4,_readed,0);
            for j:=1 to CurrFileSize div MainBufferSize do
                begin
                    ReadFile(DATA,pBuff^,MainBufferSize,_readed,0);
                    WriteFile(CurrFile,pBuff^,_readed,_writed,0);
                end;
            ReadFile(DATA,pBuff^,CurrFileSize mod MainBufferSize,_readed,0);
            WriteFile(CurrFile,pBuff^,_readed,_writed,0);
            if FolderPath[Length(FolderPath)]'\ ' then
                Delete(FolderPath,Length(FolderPath),1);
        end;
    if path_in_archive_file[1]'\ ' then
        path_in_archive_file:= '\'+path_in_archive_file;
        path_in_archive_file:=FolderPath+path_in_archive_file;
        makedir(ExtractFileDir(path_in_archive_file));
```

Розглянемо розроблену функцію, що буде передавати параметри функції склеювання файлів (функція зворотного виклику).

```
type
TActionFuntion=function (SourceFileName, DestFileName:string):boolean;

function DoFolderAction(FolderPath, ArchivePath:string;
    ActionFunction :TActionFuntion):boolean;
var
    DATA, CurrFile :THandle;
    Value, i, CurrFileSize, _readed, _writed:DWORD;
    _Files:TStringList;
    path_in_archive_file :string;
    pBuff:Pointer;
begin
    Result:=false;
    if FolderPath[Length(FolderPath)]='\ ' then FolderPath:=FolderPath+'\ ';
    _Files:=GetAllFiles('*.*', FolderPath);
DATA:=CreateFile(pchar(ArchivePath), GENERIC_READ+GENERIC_WRITE, FILE_SHARE_READ, 0,
    CREATE_ALWAYS, 0, 0);
    Value:=ArchiveSignature;
    WriteFile(DATA, value, 4, _writed, 0);
    Value:=_Files.Count;
    WriteFile(DATA, value, 4, _writed, 0);
    { кількість файлів }
    for i:=0 to value-1 do
        begin
            Виклик функції зворотного виклику, який передаємо файл джерело й файл
результат, файл-результат це всього лише проміжний файл, потім ми його
видалимо. Змінна bFile оголошена як константа.
            if not ActionFunction(_Files.Strings[i], bFile) then
                begin
                    // якщо функція зворотного
                    // виклику завершилася невдало,
                    // виходимо з функції
                    CloseHandle(DATA);
                    exit;
                end;
            CurrFile :=
            CreateFile(pchar(bFile), GENERIC_READ, FILE_SHARE_READ, 0, OPEN_EXISTING, 0, 0);
            CurrFileSize := GetFileSize(CurrFile, nil);
```

					ВКРМ-123.23.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		54

```

{одержуємо "шлях файлу в папці"}
  path_in_archive_file :=copy(Files.Strings[i],
                             length(FolderPath),1000);
  if path_in_archive_file[1] = '\' then
  Delete(path_in_archive_file,1,1);
  Write_DD(DATA,path_in_archive_file);
  { ім'я файлу}
  {Функція Write_DD пише у файл стоку у форматі LS}
  WriteFile(DATA,currFileSize,4,_writed,0);
{
  розмір файлу,
  місце під буфер
}
pBuff:=VirtualAlloc(0,MainBufferSize,MEM_COMMIT+MEM_RESERVE,PAGE_READWRITE);
  repeat
    ReadFile(CurrFile,pBuff^,MainBufferSize,_readed,0);
    WriteFile(DATA,pBuff^,_readed,_writed,0);
  until _writed < MainBufferSize;
{
  ВИЗВОЛЕННЯ буфера
}
  VirtualFree(pBuff,MEM_RELEASE,0);
  CloseHandle(CurrFile);
  DeleteFile(bFile);
end;
CloseHandle(DATA);
Result:=True;
end;

```

Використовуваний основний запуск склейки файлів:

```

function DODO(fl1,fl2:string):boolean;
begin
  Result:=CopyFile(pchar(fl1),pchar(fl2),false);
end;

```

Розглянемо функцію одержання списку всіх файлів у папці, що передає шлях до папки у форматі TStringList.

```

function Get_All_Files(Filter, Folder: string):TFilesList;
var
  sr: TSearchRec;
  sDirList,_FilesList,_LST: TStringList;
  i,j: Integer;
begin

```

						ВКРМ-123.23.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата			55

```

j:=0;
_FilesList:= TStringList.create;
_FilesList.Clear;

```

На початку створюється об'єкт результат, потім відбувається пошук файлів, які перебувають у папці, шлях до якої був переданий як другий параметр.

Перші два знайдених файли ігноруються тому, що це папки й при переході на які відбувається перехід на один рівень нагору.

```

if FindFirst (Folder + Filter,faAnyFile, sr) = 0 then
  repeat
    j:=j+1;
    if j 0;
    FindClose(sr);

```

Процедура одержання списку розширеної інформації підпапок. Аркуш із результатом вертається як результат.

```

procedure Get_Sub_Dirs (Folder: string; sList: TStringList);
var
  sr: TSearchRec;
begin
  if FindFirst (Folder+'*.*',faDirectory, sr) = 0 then
    try
      repeat
        if (sr.Attr and faDirectory) = faDirectory then
          sList.Add (sr.Name);
        until FindNext(sr) 0;
      finally
        FindClose(sr);
      end;
    end;
end;

```

Створюємо об'єкт, у якому зберігатися список підпапок, і викликаємо для кожної підпапки саму себе. Після одержання списку файлів у кожної підпапки відбувається копіювання отриманого списку в результат.

```

SS:= TStringList.Create;
  try
    GetSubDirs(Folder, SS);
    for i:=0 to SS.Count -1 do
      if (SS[i] '.') and (SS[i] '..') then
        begin
          _LST:=GetAllFiles(Filter, IncludeTrailingPathDelimiter(Folder+SS[i]));

```

						ВКРМ-123.23.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата			56

```

for j:=0 to _LST.Count-1 do
  _FilesList.Add(_LST.Strings[j]);
  _LST.free;
  end;
finally
  SS.Free;
end;
Result:=_FilesList;
end;

```

4.2 Захист розробленого програмного забезпечення

Захист розробленого програмного забезпечення буде відбуватися за допомогою Sinople – симетричний блоковий криптоалгоритм, побудований на основі незбалансованої «мережі Фейстеля». Алгоритм розроблено у 2003 році.

Основні вимоги до алгоритму при його розробці:

- Можливість програмної і апаратної реалізації.
- Висока швидкість.
- Простота.
- Низькі вимоги до пам'яті.
- Високий рівень безпеки.

Алгоритм заснований на 32-розрядних операціях і має 64 раунду, серед яких два типи – С і D. D раунди спроектовані для досягнення максимальної дифузії, С раунди – для досягнення перемішування. F-функція D раунду використовує один з елементів блоку даних (D[3]) та поточного з'єднання (K[r]) для трансформації 3-х елементів блоку даних. F-функція С раунду, навпаки, використовує перші три елемента блоку даних і поточний з'єднання (K[r]) для трансформації останнього елемента блоку даних (D[3]). Раунди D-типу виконуються до раундів С-типу. Додавання ключів з даними проводиться тільки через таблиці замін. Операції XOR (додавання по модулю 2) обов'язково поєднуються з операціями ADD (додавання по модулю 2^{32}).

					ВКРМ-123.23.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		57

Таблиці замін спочатку запозичені з алгоритму MARS і містять 512 32-розрядних елементів, проте були жорстко проаналізовані на предмет посилення.

Ключове розклад було спроектовано з урахуванням вимог:

- Простота
- Використовується та ж процедура, що і при шифруванні та розшифрування
- Установка ключа займає менше часу, ніж зашифрування
- Виключення еквівалентних ключів
- Виключення слабких ключів

Алгоритм, згідно із заявою авторів, стійкий до лінійного і диференціального аналізу.

КБГПЗ-2023

					ВКРМ-123.23.0040.00.00.ПЗ	<i>Арк.</i>
<i>Вим.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		58

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Розглянемо розроблене ПЗ яке зображено на рисунку 5.1. З рисунку можна побачити що інтерфейс головного вікна розподілено на наступні розділи:

- Графічне представлення місцеположення супутника;
- Функціональних даних;
- Меню.

Розроблена програма має дуже простий і інтуїтивно зрозумілий інтерфейс з користувачем. Кожен, хто в достатньому обсязі володіє операційним середовищем без особливих складностей освоїть і цю програму, оскільки її інтерфейс інтуїтивно зрозумілий.

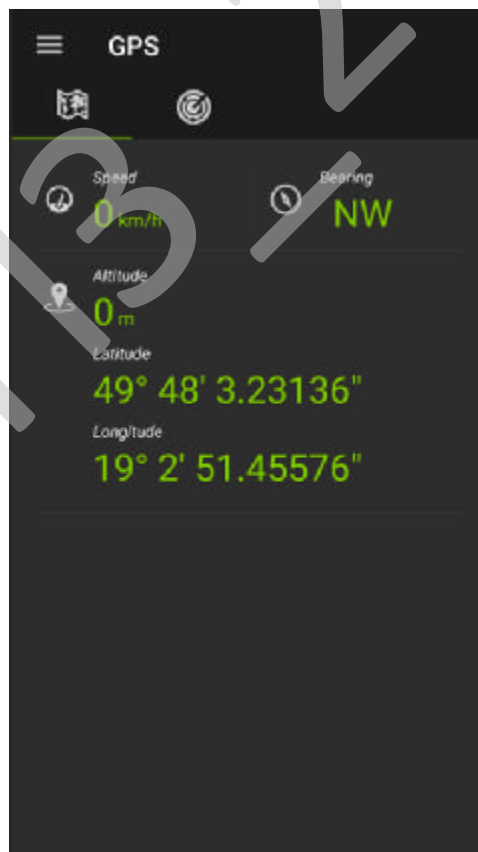


Рисунок 5.1 – Вікно розробленого ПЗ

					ВКРМ-123.23.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		59

Відображує кількість знайдених супутників у текстовому та графічному режимі. Показує якість сигналу зі супутників та геопараметричні характеристики – широта, довгота, висота над рівнем моря. Вказується поточний час та розраховується поточний рух.

Виведення даних та їх розрахунок починається зі знайдення мінімальної кількості супутників – 3. Максимальна кількість супутників – 12. На рисунку 5.3 зображена форма авторського права з виказанням даних розробника та дати розробки.



Рисунок 5.2 – Авторське право

Обрано умови розповсюдження – умовно-безплатне (Shareware). Під умовно-безплатним програмним забезпеченням можна розуміти спосіб або метод розповсюдження комерційного ПЗ на ринку (тобто на шляху до кінцевого

					ВКРМ-123.23.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		60

користувача), при якому випробувачеві пропонується обмежена за можливостями (не повнофункціональна або демонстраційна версія), терміном дії (тріал версія) або версія з вбудованим набридливим нагадуванням про необхідність оплати використання програми.

В угоді про використання (ліцензії для кінцевого користувача, EULA) також може бути обумовлена заборона на комерційне або професійне (не тестове) її використання.

Протягом певного терміну, що становить зазвичай тридцять днів, він може користуватися програмою, тестувати її, освоювати її можливості. Якщо після закінчення цього терміну користувач вирішить продовжити використання ПЗ, він зобов'язаний купити його (зареєструватися), заплативши авторові певну суму.

КБПЗ-2023

					VKPM-123.23.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		61

6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи GPS пристрою на основі процесора ARM Cortex-A15 MPCore.

Метою розробки є дослідження та програмна реалізація системи GPS пристрою на основі процесора ARM Cortex-A15 MPCore.

Об'єктом дослідження є процес GPS пристрою на основі процесора ARM Cortex-A15 MPCore.

Предметом дослідження є методи GPS пристрою на основі процесора ARM Cortex-A15 MPCore.

Методи дослідження базуються на методах теорії геопозиціонування, методах математичної статистики, методах розробки програмного забезпечення.

Наукова новизна отриманих результатів. У процесі рішення завдань, обумовлених цілями дослідження, отримані наступні результати:

– Удосконалено метод GPS пристрою на основі процесора ARM Cortex-A15 MPCore.

– Розроблено вітчизняний продукт GPS пристрою на основі процесора ARM Cortex-A15 MPCore, який має більш широкі можливості, на відміну від існуючих аналогів.

					VKPM-123.23.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		62

Продовження таблиці 7.1

1	2	3
8. Кількість форм вихідної інформації.	–	4
9. Мова програмування (1-6)	–	2
10. Попередній досвід (1-6)	–	3
11. Гнучкість проекту ПП (1-6)	–	3
12. Детальність проекту ПП (1-6)	–	2
13. Рівень спрацьованості колективу (1-6)	–	2
14. Ступінь вимірності процесів (1-6)	–	3
15. Необхідна надійність програмного забезпечення (1-6)	–	2
16. Розмір бази даних (порівняно з розміром програми) (1-6)	–	2
17. Складність кінцевого програмного продукту (1-6)	–	2
18. Необхідний рівень забезпечення повторного використання (1-6)	–	2
19. Документованість відповідно до планованого життєвого циклу (1-6)	–	2
20. Вимоги до швидкодії ПП (1-6)	–	2
21. Обмеження на розміри основного сховища даних (1-6)	–	2
22. Різноманітність використовуваних обчислювальних платформ (1-6)	–	2
23. Професійний рівень аналітиків (1-6)	–	2
24. Професійний рівень програмістів (1-6)	–	2
25. Постійність складу команди розробників (1-6)	–	2
26. Досвід розробки додатків (1-6)	–	2
27. Досвід роботи з обчислювальною платформою (1-6)	–	2

Продовження таблиці 7.1

1	2	3
28. Досвід роботи з мовою і інструментами середовища розробки (1-6)	–	2
29. Досвід роботи з програмними інструментами розробки (1-6)	–	3
30. Розробка ПО для декількох серверів одночасно (1-6)	–	2
31. Вимоги до дотримання встановленого графіка робіт (1-6)	–	2
32. Вартість ПЗ у розробника (НМА), грн	–	60000
33. Норматив додаткової зарплати, % :	Н _д	10
34. Норматив відрахувань у соціальні фонди, %	Н _с	22
35. Норматив загальногосподарських витрат, %	Н _г	15
36. Норматив витрат на освоєння нових мов програмування, %	Н _п	15
37. Рівень рентабельності програмної продукції, %	Р _е	50
38. Ставка податку на додану вартість, %	Н _{дв}	20

7.2 Розрахунок трудомісткості розробки програмної продукції

Значення трудомісткості розробки програмного забезпечення для стадій ТЗ, ЕК, ТП та ВП визначаємо по типовим нормам часу приведеним в додатках МВ. Стадія РП є найбільш тривалою і трудомісткою, що робить значний вплив на інші стадії проекту.

Визначимо трудомісткість розробки ПЗ для стадії РП.

Обчислюємо номінальні трудовитрати, люд-міс.:

$$T_{ном} = A \text{ Size}^B \quad (7.1)$$

де А – коефіцієнт Боєма, А=2,45;

Size – загальний об'єм відлагодженого програмного коду, тис. рядків;

B – показник ступеня, що визначається співвідношенням

$$B = 1,01 + 0,001 \sum W_i \quad (7.2)$$

де W_i – сумарне значення п'яти показників (МВ, додаток 2), що відображають особливості розробки проекту програмного продукту (ПП) і колективу розробників.

$$B = 1,01 + 0,001(2,43 + 3,64 + 3,38 + 3,95 + 2,73) = 1,026$$

$$T_{ном} = 2,45 \cdot 2,7^{1,026} = 6,78 \text{ люд-міс.}$$

Визначаємо уточнені (з урахуванням приведених в МВ додатку 3 сімнадцяти додаткових коефіцієнтів) трудовитрати, люд-міс.:

$$T_{уточн} = T_{ном} \cdot \Pi V_j, \quad (7.3)$$

де ΠV_j – добуток сімнадцяти додаткових коефіцієнтів, приведених в МВ додатку 3.

$$T_{уточн} = 6,78 \cdot (0,88 \cdot 0,93 \cdot 0,88 \cdot 0,91 \cdot 0,95 \cdot 1 \cdot 1 \cdot 0,87 \cdot 1,22 \cdot 1,16 \cdot 1,1 \cdot 1,1 \cdot 1,12 \cdot 1,1 \cdot 1,1 \cdot 1,1) = 9,37 \text{ люд-міс.}$$

Ці коефіцієнти дозволяють диференційовано оцінювати результати роботи програмістів, беручи до уваги швидкість програми, використання різноманітних обчислювальних платформ і інструментів розробки, взаємодію декількох серверів, вимоги до об'ємів баз даних і ін.

Визначаємо підсумкові трудовитрати по стадії робочий проект, люд-дні:

$$T_{РП} = 0,3 C T_{уточн}^{0,33+0,2(B-1,01)} S, \quad (7.4)$$

де C – визначений емпірично коефіцієнт, запропонований авторами методики, (МВ, додаток 4); S – коефіцієнт стиснення (або подовження) графіка робіт %, що дозволяє коректувати терміни розробки ПО згідно встановленим вимогам. Вибираємо в межах (25...350)%

$$T_{РП} = 0,3 \cdot 2,66 \cdot 9,37^{0,33+0,2(1,026-1,01)} \cdot 182 = 306 \text{ люд/день}$$

Для зручності визначення загальної трудомісткості на розробку програмного забезпечення результати розрахунків по стадіям зводимо до таблиці 7.2.

					ВКРМ-123.23.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		66

Таблиця 7.2 – Визначення трудомісткості розробки програмного забезпечення

Стадії розробки	Трудомісткість за типовими нормами та розрахунками	
	Величина, люд/дні	Підстава
Технічне завдання	9	Д5
Ескізний проект	10	Д6
Технічний проект	9	Д7
Робочий проект	306	Ф 7.1-7.4
Впровадження	13	Д13
Всього	347	–

7.3 Визначення чисельності виконавців і планового фонду зарплати

Чисельність ставок інженерів-програмістів для розробки програмного забезпечення визначається за формулою

$$Ч = \frac{T_{нз} N}{F_{pq} - H_{ев}}, \quad (7.5)$$

де F_{pq} – плановий фонд робочого часу одного спеціаліста, днів,

$T_{нз}$ – трудомісткість розробки програмного забезпечення люд-дні,

$$Ч = \frac{347 \cdot 1}{48 \cdot 5} = 8 \text{ ставки}$$

Чисельність інженерів-електронщиків для проведення технічного обслуговування та ремонту комп'ютерних мереж визначається в залежності від наявності технічних засобів і норм витрат часу на виконання профілактичних робіт на протязі року.

Визначаємо затрати часу на виконання профілактичних робіт по обслуговуванню обладнання за період розробки. Результати розрахунку зводимо до таблиці 7.3

Таблиця 7.3 – Затрати часу на виконання профілактичних робіт по обслуговуванню обладнання за розрахунковий період

Найменування обладнання	Профілактичне обслуговування			
	Кількість хв. на один. обл.	Кількість обладнання	Затрати часу в хв.	Затрати часу в год.
Системний блок ПК	90	11	990	16,5
Монітор	60	11	660	11
Клавіатура	30	11	330	5,5
Маніпулятор «мишка»	30	11	330	5,5
Принтер матричний	60	0	0	0,0
Принтер лазерний	120	3	360	6
Принтер струминний	60	1	60	1
Сканер	20	1	20	0,33
Концентратор-маршрутизатор	30	3	90	1,5
Кабельні господарства ЛОМ на 1 м.п.	2,5	450	1125	18,75
Копіювальний апарат	140	2	280	4,67
Усього за рік:			3 _ч	70,75

Час на профілактику обладнання в загальному балансі робочого часу інженерів-електронщиків не повинен складати більше 10%

Виходячи з цього фонд робочого часу інженерів-електронщиків складає:

$$\Phi_{op}^c = \frac{3_{ч} \cdot n_{mic}}{1,2} \quad (7.6)$$

$$\Phi_{op}^c = \frac{70,75 \cdot 2}{1,2} = 118 \text{ год}$$

Визначаємо необхідну кількість ставок штатного персоналу сектора ТО:

$$Ч_{ел} = \frac{\Phi_{op}^c}{F_{op} \cdot T_{зм}} \quad (7.7)$$

Продовження таблиці 7.4

Посада	Вид роботи	Час	Кількість штатних одиниць
Продакт-менеджер	Презентації нової продукції, пошук каналів збуту	2	0,5
	Підтримка постійних клієнтів	1	
	Оформлення договорів, ведення тендерів	0,5	
	Контроль взаєморозрахунків з постачальниками	0,5	
Всього		4	
Дизайнер WEB	Розробка концепції оформлення та інтерфейсу сайту, оптимізація дизайну існуючих, проектує їх структуру та навігацію	0,5	0,2
	Створення графічних і стилістичних елементів сайту	0,5	
	Оформлення банерів і промо-сторінок	0,3	
	Розміщення графіки і контенту на Інтернет сторінках	0,3	
Всього		1,6	
Інженер верстальник	Розробка та верстка макетів рекламної продукції та технічної документації	1	0,2
	Верстка друкованих видань	0,2	
	Додрукова підготовка макетів	0,2	
	Розміщення графіки і контенту на Інтернет сторінках	0,2	
Всього		1,6	

Складемо штатний розклад виконавців у таблицю 7.5.

Таблиця 7.5 – Штатний розклад виконавців

Посада	Кількість ставок	Середньо-місячний оклад, грн.	Всього за період розробки, грн.
Керівник (ІТ-менеджер)	1	15544	31088
Продакт-менеджер	0,5	13000	13000
Інженер-програміст	8	13500	216000
Інженер-електронщик	0,3	9000	5400
Інженер-системотехнік	0,2	9000	3600
Адміністратор мережі	0,2	11000	4400
Системний програміст	0,2	9000	3600
Дизайнер WEB	0,2	9000	3600
Інженер-верстальник	0,2	9000	3600
Бухгалтер-економіст	0,2	10000	4000
Всього за період розробки	$R_{cn}=11$	-	$\Phi_{роб}=288288$

Розрахуємо середньоденну зарплату одного виконавця:

$$z_{cd} = \frac{\Phi_{роб}}{R_{cn} F_{pq}}, \quad (7.8)$$

де $\Phi_{роб}$ – загальна сума зарплати за плановий період, грн.

$$z_{cd} = \frac{288288}{11 \cdot 48} = 546 \text{ грн.}$$

7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника

Балансова вартість будівель визначається з урахуванням кількості робочих місць виконавців, питомої площі на одне робоче місце, та вартості одного квадратного метра виробничої площі

					ВКРМ-123.23.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		71

$$B_{y\partial} = R_{cn}^1 S_y \Pi_{nl}, \quad (7.9)$$

де R_{cn}^1 – кількість робочих місць виконавців, шт. Приймаємо 8 робочих місць.

S_y – питома площа на одне робоче місце, m^2 ,

Π_{nl} – вартість одного квадратного метра площі, грн.

Згідно даних ТОВ науково-дослідницького консалтингового підприємства «Пектораль» ціна одного квадратного метра площі новобудови, вік якої не перевищує 25 років, по місту складає 400...1600 у.о./ m^2 . Враховуючи, що курс складає 1 у.о. = 37 грн. приймаємо для розрахунку вартість одного метра квадратного рівною 20000 грн./ m^2 . На кожне робоче місце у середньому потрібно 8 m^2 . З урахуванням цього:

$$B_{y\partial} = 8 \cdot 8 \cdot 20000 = 1280000 \text{ грн.}$$

Вартість передавальних пристроїв складає 10% від вартості будівель, і у даному випадку вона складе: 128000 грн.

Балансова вартість інвентарю розраховується за нормою 12500 грн на одне робоче місце. Тобто

$$I_{nb} = R_{cn}^1 \cdot \Pi_m, \quad (7.10)$$

де Π_m – ціна меблів для одного робочого місця, грн.

$$I_{nb} = 8 \cdot 12500 = 100000 \text{ грн}$$

Балансова вартість обчислювальної техніки визначається по оптовим цінам постачальника з врахуванням витрат на транспортування.

Специфікація на обчислювальну техніку наведена в таблиці 7.7. Дані по оптовій ціні на обладнання та комплектуючі вибирались за комерційною пропозицією фірми Brain за 25.10.23 – джерело <http://brain.com.ua/>

					ВКРМ-123.23.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		72

Таблиця 7.6 – Специфікація

Найменування комплектуючої або обладнання	Тип	Оптова ціна
Персональний комп'ютер		11771
Системний блок		7771
Процесор	INTEL Pentium G6405 (BX80701G6405) 1200, 2 ядра, 4 потоки, 4.1 GHz, TDP – 5 Вт, 14nm, BOX	-
Системна плата	ASUS PRIME H510M-K DDR4, LAN – 1 Гбит/с, D-Sub (VGA), HDMI, 1 x M.2 2280, 4 x SATA 6.0 Gb/s, Micro-ATX	-
Жорсткий диск	SSD M.2 2280 240GB Apacer 240 GB, 3150 MB/s, TLC, M.2, PCI Express 3.0 x4	-
Оперативна пам'ять	DDR4 16GB 3200 MHz Fury Beast Black Kingston Fury (ex.HyperX) (KF432C16BB/8) DDR4, 8 ГБ, В наборі – 2	-
Відеоадаптер	Intel UHD Graphics 610	-
Корпус	Vinga CS210B, Miditower, ATX	-
Кардридер внутрішній	Transcend TS-RDF8K USB 3.0	-
інше	Клавіатура, мишка	-
Монітор	Монітор BenQ GL2450HM Black	2600
Принтер лазерний	Canon i-SENSYS LBP6030W	2700
Принтер струминний	Epson Stylus Photo P50 (C11CA45341) + USB cable	5500
Сканер	Epson Perfection V37	2800
Копіювальний апарат	Canon i-SENSYS MF217W with Wi-Fi	5965
Пристрій безперебійного живлення	Powercom BNT-600AP USB	1400

					ВКРМ-123.23.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		73

Витрати на транспорт, монтаж та випробування можуть бути прийняті в межах до 10% від оптової ціни.

Для визначення необхідної кількості капітальних вкладень складемо таблицю 7.8.

Таблиця 7.7 – Балансова вартість обчислювальної техніки

Найменування обчислювальної техніки	Кількість, шт.	Ціна за одиницю, грн.	Витрати на транспортування, монтаж та випробування.	Загальна вартість, грн.
Персональні комп'ютери	8	11771	9416,8	103584,8
Принтер лаз.	2	2700	540	5940
Принтер струм.	1	5500	550	6050
Сканери	1	2800	280	3080
Копіюв. апарат	1	5965	596,5	6561,5
Всього	–	–	–	125216,3

Таблиця 7.8 – Вартість основних фондів та амортизаційні відрахування розробника

Групи та види основних фондів	Балансова вартість, грн.	Амортизація	
		Норма, %	Відрахування, грн.
1	2	3	4
Група 3			
1. Будівлі	1280000	-	-
2. Передавальні пристрої	128000	-	-
Всього по групі	1408000	5	70400

Продовження таблиці 7.8

1	2	3	4
Група 4			
3. Обчислювальна техніка	125216	-	-
Всього по групі	125216	50	62608
Група 5,6			
4. Вимірювальні пристрої	5190	-	-
5. Транспортні засоби	143000	-	-
6. Господарський інвентар	100000	-	-
Всього по групі	248190	20	49638
7. Нематеріальні активи	60000	10	6000
Разом	$K_p = 1841406$		$A_p = 188646$

Примітка: вартість автомобіля взята по даним з автосалону «Кіровоград-Авто», джерело <http://kirovograd-avto.ukravto.ua/catalog/tm-9/model-80/description>, складає 143000 грн.

7.5 Визначення собівартості розробки та ціни програмної продукції

Визначимо основну зарплату виконавців:

$$Z_o = \frac{Z_{cd} \cdot T_{nz}}{N_e}, \quad (7.11)$$

де N_e – Кількість екземплярів програм, шт.

$$Z_o = 546 \cdot 347 / 60 = 3158 \text{ грн}$$

Визначимо додаткову зарплату (оплата відпусток, виконання державних та суспільних обов'язків) на рівні 10%

					ВКРМ-123.23.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		75

$$Z_{\delta} = Z_o \cdot H_q \cdot 0,01, \quad (7.12)$$

де H_q – норматив додаткової зарплати, %

$$Z_{\delta} = 3158 \cdot 10 \cdot 0,01 = 316 \text{ грн}$$

Відрахування на соціальні потреби за нормативом $H_c = 22\%$ від суми основної та додаткової зарплати

$$C_{oc} = 0,01 \cdot H_c (Z_o + Z_{\delta}), \quad (7.13)$$

де H_c – відрахування на соціальні потреби, %

$$C_{oc} = 0,01 \cdot 22(3158 + 316) = 764 \text{ грн}$$

Визначимо загальногосподарські витрати (електроенергію, ремонт і утримання приміщень і т.д) за нормативом $H_z = 15\%$ від основної зарплати

$$G_{ocn} = Z_o \cdot H_z \cdot 0,01, \quad (7.14)$$

де H_z – загальногосподарські витрати, %

$$G_{ocn} = 3158 \cdot 15 \cdot 0,01 = 474 \text{ грн}$$

Визначимо витрати на матеріали для розробки програмної продукції за нормами споживання та діючими цінами за одиницю виміру:

$$Z_M = (Z_{M1} + Z_{M2} + Z_{M3}) / N_e, \quad (7.15)$$

де Z_{M1} – вартість паперу, грн., Z_{M2} – вартість запам'ятовуючих пристроїв, грн., Z_{M3} – вартість фарби, картриджів, тонеру, грн., N_e – кількість екземплярів програм, шт.

Згідно виданих норм приймаємо 0,5 пачки паперу на період розробки. Тоді, враховуючи, що вартість пачки паперу складає $C_n = 160$ грн., визначаємо вартість паперу за період розробки $N_M = 2$ міс:

$$Z_{M1} = C_n \cdot N. \quad (7.16)$$

$$Z_{M1} = 160 \cdot 0,5 = 80 \text{ грн.}$$

Згідно виданих норм до вартості запам'ятовуючих пристроїв входить вартість CD/DVD дисків в кількості 10 примірників:

$$Z_{M2} = \sum C_{\delta}, \quad (7.17)$$

					ВКРМ-123.23.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		76

де C_d – вартість дисків CD/DVD: CDR TDK 700Mb, 80Min, 52x Cake box – 49 грн/шт., DVD-R LG 4,7Gb, 16x speed Cake box – 49 грн/шт.

$$Z_{M2} = 49 \cdot 10 = 490 \text{ грн.}$$

Згідно виданих викладачем норм одноразовій заправці підлягають усі друкуючі пристрої і становить:

$$Z_{M3} = \sum C_z, \quad (7.18)$$

де: C_z – вартість розхідних матеріалів друкуючих пристроїв: відновлення та заправка картриджу для Canon i-SENSYS LBP6030W – 574 грн.; картридж для Epson Stylus Photo P50 – 558 грн.; відновлення картриджу для MF217W – 570 грн.

$$Z_{M3} = 574 + 558 + 570 = 1702 \text{ грн.}$$

$$Z_M = (80 + 490 + 1702) / 60 = 38 \text{ грн.}$$

Визначимо витрати на освоєння нових мов програмування або операційних систем за нормативом ($H_n = 15\%$) від основної зарплати виконавців

$$O_n = Z_o \cdot H_n \cdot 0,01, \quad (7.19)$$

де H_n – норматив витрат на освоєння нових мов програмування, %

$$O_n = 3158 \cdot 15 \cdot 0,01 = 474 \text{ грн}$$

Визначимо витрати на амортизацію основних фондів з урахуванням загальної річної суми амортизаційних відрахувань та кількості екземплярів програм ($N_e = 60$ прим.)

$$A_m = \frac{A_p \cdot N_{mic}}{N_e \cdot 12}, \quad (7.20)$$

де A_p – загальна річна сума амортизаційних відрахувань, грн.

$$A_m = 174246 \cdot 2 / (60 \cdot 12) = 484 \text{ грн}$$

Повна собівартість ПЗ визначається як сума витрат за попередніми статтями калькуляції

$$C_n = Z_o + Z_d + C_{oc} + \Gamma_{ocn} + Z_M + O_n + A_m. \quad (7.21)$$

$$C_n = 3158 + 316 + 764 + 474 + 38 + 474 + 484 = 5708 \text{ грн.}$$

Визначимо плановий прибуток за рівнем рентабельності (R_p) програмної продукції, яка залежить від складності програми та ступеня новизни задачі.

					ВКРМ-123.23.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		77

Для даного програмного забезпечення рівень рентабельності складає 50%

$$P_p = 0,01 \cdot P_n \cdot C_n, \quad (7.22)$$

де P_c – рівень рентабельності, %

$$P_p = 0,01 \cdot 50 \cdot 5708 = 2854 \text{ грн.}$$

Величини ціна підприємства, податок на додану вартість, відпускна ціна програмної продукції визначаються за формулами, приведеними в таблиці 7.9

Таблиця 7.9 – Нормативна калькуляція собівартості розробки програмного забезпечення задачі

Найменування статей витрат	Позначення	Величина, грн.
1	2	3
1. Основна зарплата виконавців	Z_o	3158
2. Додаткова зарплата виконавців	Z_d	316
3. Відрахування на соціальні потреби	C_{oc}	764
4. Загальногосподарські витрати	G_{ocn}	474
5. Витрати на матеріали	Z_M	38
6. Освоєння нових операційних систем, мов програмування	O_n	474
7. Амортизація основних фондів	A_m	484
8. Повна собівартість програмного забезпечення	C_n	5708
9. Плановий прибуток	P_p	2854
10. Ціна підприємства $C_n = C_n + P_p$	C_n	8562
11. Податок на додану вартість $ПДВ = 0,01 \cdot H_{ов} \cdot C_n$	$ПДВ$	1712,4
12. Відпускна ціна програмної продукції $Ц = Ц_n + ПДВ$	$Ц$	10274,4

7.6 Визначення об'єму капітальних вкладень у споживача програмної продукції

Об'єм капітальних вкладень у споживача програмної продукції визначаємо на основі балансової вартості основних фондів, яка враховує ціну, транспортно-заготівельні витрати, вартість будівель, монтажних та пусконаладжувальних робіт, а також витрати на випробування у виробничих умовах. Результати розрахунків зводимо у таблицю 7.10.

Таблиця 7.10 – Розрахунок об'єму капітальних вкладень у споживача програмної продукції

Найменування капітальних вкладень	Сума за варіантами, грн	
	Базовий	Новий
Вартість програмної продукції	–	10274
Всього капітальних витрат	–	10274

7.7 Визначення експлуатаційних витрат

Експлуатаційні витрати у споживача програмної продукції визначаємо при умові роботи підсистеми на протязі року. Результати зводимо до таблиці 7.11.

Таблиця 7.11 – Розрахунок експлуатаційних витрат у споживача програмної продукції

Найменування статей витрат	Позначення	Сума витрат за варіантами, грн.	
		Базовий	Новий
1. Витрати на технічне обслуговування	Z_p	33550	14762
2. Витрати на електроенергію	Z_{el}	5472	2736
3. Витрати на амортизацію	$Z_{ам}$	0	5137
Всього витрат за рік	I	39022	22635

Витрати на технічне обслуговування:

$$Z_p = T_p \cdot Z_z \cdot (1 + 0,01 \cdot H_q) \cdot (1 + 0,01 \cdot H_c), \quad (7.23)$$

де T_p – кількість годин обслуговування системи за рік, год.,

Z_z – заробітна плата обслуговуючого персоналу, грн/год

Після купівлі нового програмного забезпечення кількість профілактичних годин робіт зменшилася з 250 годин на рік до 110 годин на рік, тому витрати на технічне обслуговування зменшилися з

$$Z_{p \text{ баз}} = 250 \cdot 100 \cdot 1,1 \cdot 1,22 = 33550 \text{ грн.}$$

до

$$Z_{p \text{ нов}} = 110 \cdot 100 \cdot 1,1 \cdot 1,22 = 14762 \text{ грн.}$$

Витрати на електроенергію визначаються з урахуванням спожитої потужності ($P_{ел}$) в кіловатах, часу експлуатації технічних засобів (T_p) в годинах та ціни однієї кіловат-години ($C_{ел}$).

$$Z_{ел} = P_{ел} \cdot T_p \cdot C_{ел}. \quad (7.24)$$

$$Z_{ел \text{ баз}} = 0,2 \cdot 7200 \cdot 3,8 = 5472 \text{ грн}$$

$$Z_{ел \text{ нов}} = 0,2 \cdot 3600 \cdot 3,8 = 2736 \text{ грн}$$

Витрати по амортизації визначаються на основі норм амортизаційних відрахувань, вартості програмної продукції і основних фондів. Для розрахунку складаємо таблицю 7.12.

Таблиця 7.12 – Розрахунок амортизаційних відрахувань

Групи основних фондів	Норма амортизації і %	Балансова вартість, грн., за варіантами		Сума відрахувань, грн за варіантами	
		Базовий	Новий	Базовий	Новий
Програмна продукція	50	–	10274	–	5137
Всього відрахувань	-	–	10274	–	5137

$$T_{cn} = \frac{10274}{39022 - 22635} = 0,63 \text{ року}$$

Показники економічної ефективності програмної продукції зводимо до таблиці 7.13.

Таблиця 7.13 – Показники економічної ефективності програмної продукції

Найменування показників	Одиниця виміру	Величина
1. Кількість екземплярів програми	Прим.	60
2. Повна собівартість розробленої програми	Грн.	5708
3. Ціна розробленої програми	Грн.	8562
4. Плановий прибуток від реалізації розробленої програми	Грн.	2854
5. Рентабельність програмної продукції	%	50
6. Об'єм додаткових капітальних вкладень у виробника програмної продукції	Грн.	1841406
7. Загальний прибуток від реалізації програмної продукції	Грн.	171240
8. Величина економічного ефекту при виготовлені програмної продукції	Грн.	139799
9. Період окупності додаткових капітальних вкладень у виробника програмної продукції	Років	1,8
10. Об'єм додаткових капітальних вкладень у споживача програмної продукції	Грн.	10274
11. Величина економічного ефекту у користувача програмної продукції	Грн.	11250
12. Період окупності додаткових капітальних вкладень у користувача програмної продукції	Років	0,63

7.9 Висновки

Розроблена програма економічно вигідна. За рахунок впровадження програмного забезпечення досягається скорочення часу обробки інформації, підвищується культура праці, підвищення якості приймаючих управлінських рішень.

КБГПЗ-2023

					VKPM-123.23.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		83

8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

8.1 Вступ

Електронно-обчислювальна машина (ЕОМ) відіграє важливу роль у житті сучасної людини. Кожного дня мільйони людей використовують ЕОМ для пошуку необхідної інформації, спілкуванні у соціальних мережах, перегляду новин, роботи тощо. Багато людей користуються ЕОМ у професійних цілях, оскільки завдяки ЕОМ з'явилося багато нових професій.

Аналізуючи умови працівників ІТ-сфери, на перший погляд, може здатися, що працівники сфери інформаційних технологій не схильні до ризиків на виробництві, та якщо більш глибоко розглянути умови і специфіку праці фахівців сфері ІТ-індустрії, можна виявити ряд факторів які будуть мати негативний вплив на стан охорони праці, та на самого ІТ-фахівця зокрема. Сюди можна віднести як невідповідність освітлення, так і високий рівень шуму, що негативно позначатимуться як на емоційному так і на фізичному стані фахівця, призводитимуть до зниження ефективності праці та виробничих травм. Також, важливим моментом охорони праці ІТ-фахівця є врахування його психологічних можливостей (швидкість реакції, особливості пам'яті та уваги, емоційний стан, тощо). Для того, щоб забезпечити ефективну роботу ІТ-фахівця, потрібно враховувати та максимально компенсувати такі негативні фактори як: надмірне нервово-емоційне навантаження, довготривалі статичні перевантаження, обмежена рухова активність. Всі ці чинники призводить до різноманітних відхилень у стані здоров'я, зокрема до перевтоми, зниження фізичної та розумової працездатності, неврозів, захворювань серцево-судинної системи тощо. Метою даного розділу є огляд конкретних умов праці спеціаліста у сфері ІТ-індустрії. Завданнями для даного розділу є: аналіз умов праці на робочому місці фахівця ІТ-індустрії, розробка конкретних рекомендацій щодо покращення умов праці фахівців ІТ-

					ВКРМ-123.23.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		84

індустрії, огляд пожежної безпеки на ІТ-підприємстві та розрахунок системи загального штучного освітлення виробничого приміщення де працюють ІТ-фахівці.

8.2 Аналіз умов праці на робочому місці ІТ-фахівця

На робочому місці ІТ-фахівця (або програміста) виникають небезпечні та шкідливі для безпечної життєдіяльності фактори:

- підвищений рівень шуму;
- несприятливі мікрокліматичні умови;
- недостатній рівень освітленості;
- шкідливі речовини;
- підвищений рівень електромагнітних випромінювань радіочастот;
- висока напруга електричної мережі;
- статична електрика та інші.

Робота програміста супроводжується також підвищеним ступенем напруженості трудового процесу. При систематичному впливі виробничих факторів, які не відповідають нормативним показникам, зростає рівень професійно зумовленої захворюваності працюючих та можуть виникнути професійні захворювання органів зору, руху, нервової системи. Таким чином, вивчення умов праці на робочому місці програміста є необхідною умовою запобігання негативних наслідків впливу небезпечних та шкідливих факторів. Робоче місце, добре пристосоване до трудової діяльності інженера, правильно і доцільно організоване, щодо простору, форми, розміру забезпечує йому зручне положення при роботі і високу продуктивність праці при найменшому фізичному і психічному напруженні.

Нормування параметрів проводиться в залежності від періоду року та категорії важкості виконуваних робіт. Для постійних робочих місць, якими є робочі місця ІТ-фахівців, встановлені оптимальні параметри мікроклімату, а за

					ВКРМ-123.23.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		85

неможливості їх дотримання використовують допустимі параметри. Робота ІТ-фахівця за важкістю відноситься до Іа (роботи, що виконуються сидячи і не потребують фізичного напруження) та Іб (роботи, що виконуються сидячи, стоячи або пов'язані з ходінням та супроводжуються деяким фізичним напруженням) категорій. В таблиці 8.1. наведені оптимальні параметри мікроклімату в приміщеннях.

Таблиця 8.1 – Параметри мікроклімату для приміщень з ПК

Період року	Параметр мікроклімату	Величина
Холодний	Температура повітря в приміщенні; вологість; швидкість руху повітря	22...24°C; 40... 60%; до 0,1 м/с
Теплий	Температура повітря в приміщенні; вологість; швидкість руху повітря	23...25 °C 40...60% 0,1...0,2 м/с

Виміряні за допомогою приладів температура та вологість у приміщеннях праці ІТ-фахівців повинні відповідати зазначеним у таблиці для теплового періоду року. Слід зазначити, що для нормалізації параметрів мікроклімату слід використовувати у приміщеннях кондиціонування повітря, або забезпечити подачу свіжого повітря системами вентиляції. Норми подачі свіжого повітря наведені у таблиці 8.2.

Створення сприятливих умов праці і правильне естетичне оформлення робочих місць на виробництві має велике значення як для полегшення праці, так і для підвищення його привабливості, позитивно впливає на продуктивність праці. Забарвлення приміщень і меблів повинні сприяти створенню сприятливих умов для зорового сприйняття, гарного настрою. У службових приміщеннях, у яких виконується одноманітна розумова робота, що вимагає значної нервової напруги і

великого зосередження, забарвлення повинно бути спокійних тонів – малонасичені відтінки холодного зеленого або блакитного кольорів.

Таблиця 8.2 – Норми подачі свіжого повітря в приміщення

Характеристика приміщення	Об'ємна витрата свіжого повітря, що подається в приміщення, м ³ на одну людину в годину
Об'єм до 20 м ³ на людину	Не менше 30
20... 40 м ³ на людину	Не менше 20
Більше 40 м ³ на людину	Може біти використана природна вентиляція

При розробці оптимальних умов праці програміста необхідно враховувати освітленість. Рациональне освітлення робочого місця є одним з найважливіших факторів, що впливають на ефективність трудової діяльності людини, що попереджають травматизм і професійні захворювання. Правильно організоване освітлення створює сприятливі умови праці, підвищує працездатність і продуктивність праці. Освітлення на робочому місці програміста повинно бути таким, щоб працівник міг без напруги зору виконувати свою роботу. Стомлюваність органів зору залежить від ряду причин: недостатність освітленості; надмірна освітленість; неправильний напрям світла. Недостатність освітлення приводить до напруги зору, ослабляє увагу, приводить до настання передчасної стомленості. Надмірно яскраве освітлення викликає засліплення, роздратування і різь в очах. Неправильний напрямок світла на робочому місці може створювати різкі тіні, відблиски, дезорієнтувати працюючого. Всі ці причини можуть призвести до нещасного випадку або профзахворювань. [2]

8.3 Пропозиції щодо підвищення працездатності ІТ-фахівців

Поява та впровадження нових інформаційно-комунікаційних технологій зумовлює необхідність подальшого вдосконалення охорони праці фахівців ІТ-індустрії. Все це потребує розробки нових нормативно-правових актів з регламентації праці та відпочинку фахівців ІТ-індустрії і стандартів підприємств, центрів комп'ютерної техніки, центрів інформаційних технологій, сучасних комп'ютерних класів. Для підвищення розумової працездатності то зорової роботи повинна здійснюватися ергономічна оптимізація в рамках системи «оператор-термінал», яка сприятиме результативній фізичній та інтелектуальній працездатності і відновленню психосоматичного здоров'я фахівців ІТ-індустрії. Всі наведені заходи щодо вдосконалення охорони праці фахівців ІТ-індустрії повинні контролюватися службою охорони праці та комісією з охорони праці підприємства. Особливе значення у соціальному захисті цієї категорії працівників належить прийняття комплексного договору, який може забезпечити фахівців додатковими пільгами та компенсаціями.

Для більшого розуміння, пропозиції щодо підвищення працездатності ІТ-фахівців, розіб'ємо на декілька категорій:

Середовище і розпорядок праці. Для мінімізації негативних ефектів, що пов'язані з перевтомленням ІТ-фахівців, потрібно чітко прописати і реалізувати графік періодів праці-відпочинку, щоб фахівець міг можливість переключити увагу, дати можливість відпочити очам, мозку, елементарно, встати розім'яти ноги. Також потрібно зробити максимально комфортними умови мікроклімату у офісному приміщенні, де працюють ІТ-фахівці. Мається на увазі встановлення і експлуатація, коли виникає необхідність, кондиціонерів, опалення, та системи вентиляції, задля попередження перегрівання, переохолодження ІТ-фахівців, і подальшої неможливості ними виконувати свої функції. Також, за можливості, нами пропонується введення практики віддаленої праці ІТ-фахівцями, якщо

					ВКРМ-123.23.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		88

роботодавець не може забезпечити оптимальні і безпечні умови в офісному приміщенні, або якщо фахівця вони не влаштовують із певних причин.

Фізичні і психоемоційні чинники. Першим і найважливішим чинником, що впливає на працездатність ІТ-фахівців є робоче місце, і саме тому, роботодавець має забезпечити максимальний його комфорт і безпеку. Гарантією цих факторів може слугувати сертифікація меблів, що використовуються на підприємстві ІТ-галузі. Тому нами пропонується закупівля тільки меблів, які пошли сертифікацію на відповідність. Під психоемоційними чинниками ми розуміємо гарне самопочуття фахівців, позитивний настрій, гарний психологічний клімат у колективі, тощо. Задля того, щоб психоемоційні чинники мали максимально позитивний ефект, керівництву слід поводити заходи, які сприятимуть укріпленню і покращенню міжособистісних стосунків у колективі, таких як психологічні тренінги, тимбілдінг, спортивні змагання і естафети. Також, сюди можна віднести розробку і впровадження системи мотивації працівників, як фінансової, так моральної і адміністративної.

8.4 Розрахункова частина

Для захисного штучного заземлення застосовуються вертикальні електроди: металевий куток $80 \cdot 50 \cdot 6$ мм., (згідно з ДСТУ 8769:2018 «Кутики сталеві гарячекатані нерівнополичні. Сортамент») довжиною $L=1,7$ м., та горизонтальний електрод – металева полоса з перетином $60 \cdot 5$ мм. Напряга – $220/380$ В. Розрахункова схема розташування заземлюючих електродів – у ряд.

Розрахунок проведемо за допустимим опором розтіканню струму заземлювача.

Початкові дані для розрахунку захисного заземлення: тип верхнього шару ґрунта – чорнозем, нижнього шару ґрунта – глина (питомий опір $\rho_2 = 40$ Ом·м). Умовна товщина верхнього шару ґрунта: $H=0,5$ м. Відстань між вертикальними заземлювачами (електродами) $A=1,7$ м. Глибина закладення горизонтального

					ВКРМ-123.23.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		89

контура заземлення $t=0,6$ м. Опір заземлювача, який нормується: $R_{3H} = 4$ Ом.
Необхідно визначити необхідну кількість вертикальних заземлювачів та довжину
полоси (горизонтального заземлювача).

Розрахунок

Відстань від центра вертикального заземлювача до поверхні землі:

$$T=t+L/2=0,6+1,7/2=1,45 \text{ м.}$$

Розрахунковий питомий опір ґрунта (з врахуванням того, що фактично вся
конструкція заземлювача розташовується у нижньому шарі ґрунта):

$$\rho = \psi \rho = 1,36 \cdot 40 = 54,5 \text{ Ом}\cdot\text{м.}$$

де $\psi = 1,36$ – табличне значення коефіцієнта сезонності для відповідної
климатичної зони у багатошаровому ґрунті [10];

$\rho_2 = 40$ Ом·м. – табличне значення питомого опору нижнього шару ґрунта
(глина) [10].

Еквівалентний діаметр вертикального електрода (кутка) [10]:

$$D_e = 0,95 \cdot K = 0,95 \cdot 65 = 59,85 \text{ мм.} = 0,62 \text{ м.}$$

де $K = (80+50)/2 = 65$ мм. – середній розмір полиці металевих кутків.

Відношення $A/L = 1,7/1,7 = 1$

Опір розтіканню електричного струму одного електрода вертикального
заземлювача з урахуванням заглиблення заземлювача [10]:

$$\begin{aligned} R_0 &= 0,366(\rho/L)[\lg(2L/D_e) + (1/2)\lg((4T+L)/(4T-L))] = \\ &= 0,366(54,5/1,7)[\lg(2 \cdot 1,7/0,62) + (1/2)\lg((4 \cdot 1,45+1,7)/(4 \cdot 1,45-1,7))] = \\ &= 21,9 \text{ Ом.} \end{aligned}$$

Визначаємо коефіцієнт екранування вертикальних електродів $K_{ев} = 0,8$ при
орієнтовній кількості вертикальних електродів, яке дорівнює 5 [10].

Визначаємо необхідну кількість вертикальних електродів заземлювача (без
врахування горизонтального заземлювача), при $R_{3H} = 4$ Ом:

$$N = R_0 / (K_{ев} R_{3H}) = 21,9 / (0,8 \cdot 4) = 6,85 \approx 7 \text{ шт.}$$

Визначаємо довжину з'єднуючої полоси:

$$L_{\Pi} = 1,05 \cdot A \cdot N = 1,05 \cdot 1,7 \cdot 7 = 12,22 \approx 12 \text{ м.}$$

					ВКРМ-123.23.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		90

Опір розтіканню електричного струму з'єднуючої полоси з урахуванням кліматичного коефіцієнта питомого опору ґрунта K_{II} [10]:

$$R_{II} = 0,366(\rho \cdot K_{II} / L_{II}) \lg(2 \cdot L_{II}^2 / (B \cdot t)) = \\ = 0,366(40 \cdot 5 / 12) \cdot \lg((2 \cdot 12^2) / (0,06 \cdot 0,6)) = 23,5 \text{ Ом.}$$

де $K_{II} = 5$ – табличне значення кліматичного коефіцієнта питомого опору ґрунта для відповідної кліматичної зони для з'єднуючої полоси [10]:

$B = 60 \text{ мм.} = 0,06 \text{ м.}$ – ширина з'єднуючої полоси (задана).

Загальний опір розтіканню електричного струму заземлювача [10]:

$$R = (R_0 \cdot R_{II}) / (R_0 \cdot \eta_{II} + N \cdot R_{II} \cdot K_{ев}) = \\ = (21,9 \cdot 23,5) / (21,9 \cdot 0,75 + 9 \cdot 23,5 \cdot 0,8) = 3,48 \text{ Ом.}$$

де $\eta_{II} = 0,75$ – табличне значення коефіцієнта екранування з'єднуючої полоси [10].

Умова $R \leq R_{3H}$ виконується ($3,48 \leq 4$).

Так як при 7 вертикальних електродів R суттєво менше R_{3H} , зменшимо кількість вертикальних електродів N до 6 і виконаємо перерахунок. У результаті остаточно отримали: $R = 3,98 \text{ Ом.}$ при кількості вертикальних електродів $N = 6$.

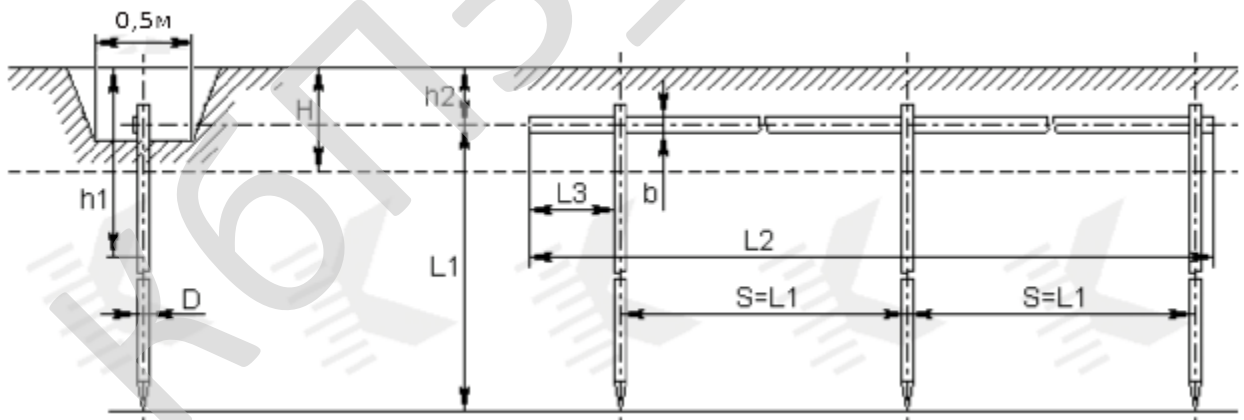


Рисунок 8.1 – Схема штучного заземлення.

8.5 Висновки до розділу

Дотримання всіх необхідних умов праці не лише сприяє збереженню здоров'я працівників, а також підвищує ефективність виробництва в цілому.

З цих міркувань було здійснено аналіз умов праці на робочому місці програміста, проведено розгляд небезпечних та шкідливих факторів, що негативно впливають на програмістів під час роботи, розроблені заходи з умов поліпшення охорони прац.

Виконано розрахунок захисного штучного заземлення, як одного з ключових факторів безпеки програміста.

Список використаних джерел інформації

1. Державні будівельні норми України: ДБН В.2.5-28:2018. – Режим доступу до ресурсу: <https://goo.su/9AkQ> (дата звернення 19.10.23).

2. Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин: ДСанПІН 3.3.2-007-98. – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/rada/show/v0007282-98> (дата звернення 19.10.23).

3. Закон України «Про охорону праці» від 14.10.1992 р. № 2694-ХІІ. – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/laws/show/2694-12> (дата звернення 19.10.23).

4. Зеркалов Д. В. Охорона праці в Галузі: Загальні вимоги: навч. посіб. Київ: Основа. 2011. 551 с.

5. Наказ Міністерства соціальної політики України 14.02.2018 № 207 «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями». – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/laws/show/z0508> (дата звернення 19.10.23).

6. Постанова № 42 від 01.12.1999 Головного державного санітарного лікаря України «Санітарні норми мікроклімату виробничих приміщень ДСН 3.3.6.042-

					ВКРМ-123.23.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		92

99. – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/rada/show/va042282-99> (дата звернення 19.09.23).

7. Оришака, О. В. Основи охорони праці: навч. посіб. / О. В. Оришака, Г. П. Горбачова, К. М. Марченко; М-во освіти і науки України, Центральноукраїн. нац. техн. ун-т. – Кропивницький : ЦНТУ, 2022. – 175 с. – Режим доступу до ресурсу: <http://dspace.kntu.kr.ua/jspui/handle/123456789/12161> (дата звернення 19.09.23).

8. Оришака О.В. Охорона праці в галузі та цивільний захист / О.В Оришака, Г.П. Горбачова, О.М. Мезенцева, К.М. Марченко, К.О. Буравченко; М-во освіти і науки України, Центральноукраїн. нац. техн. ун-т. – Кропивницький: Видавець Лисенко В.Ф., 2019. – 226 с. – Режим доступу до ресурсу: <http://dspace.kntu.kr.ua/jspui/handle/123456789/9258> (дата звернення 19.09.23).

9. Методичні рекомендації до виконання розділу "Заходи з охорони праці та техніки безпеки" випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти для здобувачів вищої освіти спеціальностей 123 "Комп'ютерна інженерія" та 122 "Комп'ютерні науки" / М-во освіти і науки України, Центральноукраїн. нац. техн. ун-т, каф. кібербезпеки та програм. забезпечення; [укл. О.В. Оришака, К.М. Марченко]. – Кропивницький: ЦНТУ, 2022. – 19 с. [Електронний ресурс]. – Режим доступу: <http://dspace.kntu.kr.ua/jspui/handle/123456789/12240> (дата звернення 19.09.23).

10. Охорона праці. Ч. 1. Захисне заземлення : метод. вказ. до викон. розрахунків з викор. персон. ЕОМ IBM сумісного типу / Кіровоград. ін-т с.-г. машинобуд. ; [укл. О. В. Оришака, Є. К. Солових, В. О. Оришака]. – Кіровоград : КІСМ, 1997. – 20 с. – Режим доступу: <http://dspace.kntu.kr.ua/jspui/handle/123456789/4358> (дата звернення 19.09.23).

					ВКРМ-123.23.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		93

9 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи GPS пристрою на основі процесора ARM Cortex-A15 MPCore.

В межах України в недостатній мірі представлені вітчизняні розробки в цій області.

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти наведені теоретичне узагальнення й рішення наукового завдання дослідження методів GPS пристрою на основі процесора ARM Cortex-A15 MPCore.

Рішення даного завдання полягало у вирішенні наступних задач:

- Був проведений огляд існуючих систем GPS пристрою на основі процесора ARM Cortex-A15 MPCore.
- Досліджена система GPS пристрою на основі процесора ARM Cortex-A15 MPCore.
- На основі отриманих результатів досліджень створена програмна реалізація системи GPS пристрою на основі процесора ARM Cortex-A15 MPCore.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання GPS пристрою на основі процесора ARM Cortex-A15 MPCore.

Проведено аналіз предметної галузі в ході якого були виявлені об'єкти, взаємодія яких носить істотний характер для функціональної діяльності предметної галузі, і їхні основні характеристики; побудована алгоритм і вибраний середовище розробки.

					ВКРМ-123.23.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		94

Розроблене програмне забезпечення має простий, дружній та зручний інтерфейс користувача, що забезпечує легкість у освоєнні роботи програмного продукту, зручність у використанні, і не потребує особливих спеціальних знань.

При створенні програмного забезпечення було використано об'єктно-орієнтований підхід, що відповідає сучасним тенденціям у галузі розробки комерційних програмних систем.

Програма реалізована на мові високого рівня Delphi 10. Дана мова програмування дозволяє найбільш ефективно обробляти дані. Це дозволило мінімізувати строк розробки програмного забезпечення, і, як слід, зменшити витрати на його розробку. Запропоноване програмне забезпечення ділиться на загальне програмне забезпечення, що поставляється із засобами обчислювальної техніки й спеціальне програмне забезпечення, що спеціально розроблене для даної конкретної системи й включає програми, що реалізують її функції.

Програма призначена для виконання під управлінням багатозадачної операційної системи Android.

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм Sinople.

В цілому створене програмне забезпечення підтверджує правильність використаних проектних рішень та повністю відповідає вимогам технічного завдання. Створене програмне забезпечення має потенційну можливість для подальшого вдосконалення і застосування у різних галузях.

Розроблена програма має реальний економічний ефект від її впровадження у виробництво у сумі 11250 грн. З урахуванням вартості розробки програми та обладнання, строк окуплення становить 0,63 роки.

					ВКРМ-123.23.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		95

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Лопотуха О.І. Дослідження та програмна реалізація системи GPS пристрою на основі процесора ARM Cortex-A15 MPCore // Збірник праць молодих науковців ЦНТУ. – Вип. 14. – Кропивницький: ЦНТУ, 2023.
2. І.В.Чихіра, А.Г. Микитишин Конспект лекцій з дисципліни «Програмування систем реального часу» / Укладачі : Чихіра І.В., Микитишин А.Г., – Тернопіль : Тернопільський національний технічний університет імені Івана Пулюя, 2016. – 76 с.
3. Jack Ganssle and Michael Barr. 2003. Embedded Systems Dictionary. CMP Books.
4. Технології інтернету речей. Навчальний посібник [Електронний ресурс]: / Б. Ю. Жураковський, І.О. Зенів; КПІ ім. Ігоря Сікорського. – Електронні текстові дані (1 файл: 12,5 Мбайт). – Київ: КПІ ім. Ігоря Сікорського, 2021. – 271 с.
5. Greg Dunko, Joydeep Misra, Josh Robertson, Tom Snyder “A reference guide to the Internet of Things” / 2017 Bridgera LLC, RIoT.
6. Donald Norris “Programming with STM32. Getting started with the Nucleo Board and C/C++” 416 p. 2018.
7. Neil Kolban “Kolban’s book on ESP32”. Texas, USA. 951 p.
8. Smirnov, O., Odarchenko, R., Smirnova, T., Bondar, S., Volosheniuk, D. «Optimal Structure Construction of Private 5G Network for the Needs of Enterprises». *Lecture Notes on Data Engineering and Communications Technologies*, 2023, 178, pp. 208–223.
9. Smirnova, T., Gnatyuk, S., Yudin, O., Sydorenko, V., Polozhentsev, A., «The Model for Calculating the Quantitative Criteria for Assessing the Security Level of Information and Telecommunication Systems». *CEUR Workshop Proceedings Volume 3156*, 2022, Pages 390-399.

					ВКРМ-123.23.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		96

10. Smirnova T., Gnatyuk S., Berdibayev R., Avkurova Zh., Iavich M. «Cloud-Based Cyber Incidents Response System and Software Tools». *Communications in Computer and Information Science*, 2021, vol 1486. Springer, Cham. pp 169-184.

11. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova T. «Non-binary constant weight coding technique». *CEUR Workshop Proceedings*. Volume 2740, 2020, Pages 102-114.

12. Smirnov O., Alimseitova Zh., Adranova A., Akhmetov B., Lakhno V., Zhilkishbayeva G. «Models and algorithms for ensuring functional stability and cybersecurity of virtual cloud resources». *Journal of theoretical and applied information technology* Vol.98. No 21, 2020, P. 3334-3346.

13. Smirnov O., Kuznetsov A., Kiian A., Cherep A., Kanabekova M., Chepurko I. «Testing of code-based pseudorandom number generators for post-quantum application». *2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT)*, Ukraine, Kyiv, May 14-18. 2020. P. 172-177.

14. Smirnov O., Kuznetsov A., Pushkar'ov A., Serhiienko R., Babenko V., Kuznetsova T., «Representation of Cascade Codes in the Frequency Domain». In: Radivilova T., Ageyev D., Kryvinska N. (eds) *Data-Centric Business and Applications. Lecture Notes on Data Engineering and Communications Technologies*, vol 48. Springer, Cham. 2021. pp 557-587.

15. Smirnov, O., Markovets, O. Vovk, N., Turchyn, Y., «Model of informational support for social network administrators' content creation». *CEUR Workshop Proceedings* Volume 2616, 2020, Pages 125-136.

16. Smirnov, O., Drieieva, H., Drieiev, O., Polishchuk, Y., Brzhanov, R., Aleksander, M. «Method of fractal traffic generation by a model of generator on the graph». *CEUR Workshop Proceedings* Volume 2616, 2020, Pages 366-379.

17. Smirnov, O., Drieieva, H., Drieiev, O., Simakhin, V., Bondar, S., Odarchenko, R. «Managing multifractal properties of the binary sequence generated

with the Markov chains», *CEUR Workshop Proceedings* Volume 2608, 2020, Pages 633-645.

18. Smirnov O. Kuznetsov A., Zaichenko Yu., Pastukhov M., Oleshko O., Kuznetsova K., «Formation of Discrete Signals with Special Correlation Properties». *International Conference on Information and Telecommunication Technologies and Radio Electronics, UkrMiCo 2019*; Odessa; Ukraine; 9-13 September 2019. P.22-28.

19. Smirnov, O., Kuznetsov, A., Kolovanova, I., Kuznetsova, T., «Noise immunity of the algebraic geometric codes». *International Journal of Computing*; 2019, Volume 18, Issue 4 – Research Institute for Intelligent Computer Systems – 2019. – P. 393-407.

20. Smirnov, O., Kuznetsov, A., Reshetniak, O., Ivko, N., Katkova, T., Kuznetsova, T., «Generators of Pseudorandom Sequence with Multilevel Function of Correlation». *2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T)*, Kyiv, Ukraine, 8 – 11 October 2019. P.517-522.

21. Smirnov, O., Odarchenko, R., Abakumova, A., Usik, P., Kundyz, M., «QoE optimization technique for media delivery in 5G networks». *2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T)*, Kyiv, Ukraine, 8 – 11 October 2019. P.597-601.

22. Smirnov, O., Krasnobayev, V., Yanko, A., Kuznetsova, T. «Methods of nulling numbers in the system of residual classes». *CEUR Workshop Proceedings*, Vol 2588, P. 90-106, 2019.

23. Smirnov, O., Kuznetsov, A., Kovalchuk, D., Averchev, A., Pastukhov, M., Kuznetsova, K., «Formation of Pseudorandom Sequences with Special Correlation Properties», *2019 3rd International Conference on Advanced Information and Communications Technologies, AICT -2019/ Lviv, Ukraine, 2-6 July, 2019*, P. 395-399.

24. Smirnov, O., Kuznetsov, A., Kiian, A., Zamula, A., Rudenko, S., Hryhorenko, V., «Variance Analysis of Networks Traffic for Intrusion Detection in

					БКРМ-123.23.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		98

Smart Grids», *2019 IEEE 6th International Conference On Energy Smart Systems (2019 IEEE ESS)*, Kyiv, Ukraine April 17-19, 2019 P. 353-358.

25. Smirnov, O., Kuznetsov, A., Kavun, S., Babenko, B., Nakisko, O., Kuznetsova, K., «Malware Correlation Monitoring in Computer Networks of Promising Smart Grids», *2019 IEEE 6th International Conference On Energy Smart Systems (2019 IEEE ESS)*, Kyiv, Ukraine April 17-19, 2019 P. 347-352.

26. Smirnov, O., Kuznetsov, A., Kovalchuk, D., Pastukhov, M., Kuznetsova, K., Prokopovych-Tkachenko, D., «Discrete Signals with Special Correlation Properties», *CEUR Workshop Proceedings Volume 2353, CEUR Workshop Proceedings 2019, Pages 618-629.*

27. Smirnov A.A., Kuznetsov A.A., Danilenko D.A., Berezovsky A., «The statistical analysis of a network traffic for the intrusion detection and prevention systems», *Telecommunications and Radio Engineering.* – Volume 74, Issue 1. – Begel House Inc. – 2015. – P. 61-78.

28. Вінтенко Б.Ю., Смірнов О.А., Коваленко А.С., Смірнов С.А., Буравченко К.О. «Дослідження вимог міжнародних стандартів IEC60880 та IEC62138 з розробки програмного забезпечення інформаційно-керуючих систем АЕС, важливих для безпеки». *Системи управління, навігації та зв'язку*, 2023, вип. 3(73), С. 155-166.

29. Аль-Мудхафар Акіл Абдулхуссейн М., Смірнова Т.В., Буравченко К.О., Смірнов О.А. «Метод оцінки та підвищення користувальницького досвіду абонентів в програмно-конфігурованих мережах на основі використання машинного навчання». *Сучасні інформаційні системи*, 2023, том 7, № 2, С. 49-56.

30. Смірнова Т.В., Гнатюк С.О., Сидоренко В.М., Юдін О.Ю., Сидоренко С.Ю., «Модель визначення критичності галузевих інформаційно-телекомунікаційних систем». *Проблеми інформатизації та управління*, № 2(70). 2022. С. 28-37.

31. Смірнов О.А., Смірнова Т.В., Якименко Н.М., Смірнов С.А., Поліщук Л.І., «Дослідження стійкості до диференціального криптоаналізу запропонованої

					ВКРМ-123.23.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		99

функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» *Системи управління, навігації та зв'язку*, 2022, № 3(69). С. 93-98.

32. Смірнов О.А., Смірнова Т.В., Якименко Н.М., Поліщук Л.І., Смірнов С.А. «Дослідження статистичної стійкості та швидкісних характеристик запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» *Вісник Хмельницького національного університету. Серія: «Технічні науки»*, № 2 (307). С. 46-52. 2022.

33. Смірнов О.А., Смірнова Т.В., Константинова Л.В., Смірнов С.А., Якименко Н.М., «Дослідження стійкості до лінійного криптоаналізу запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» *Системи управління, навігації та зв'язку*, 2022, № 1(67). С. 84-89.

34. Смірнов О.А., Смірнова Т.В., Буравченко К.О., Кравченко С.С., Горбов В.О., «Хмарна система підтримки прийняття рішень технологічного процесу відновлення поверхонь конструкцій і деталей машин». *Сучасні інформаційні системи*. 2021. Т. 5, № 4. С. 79-95

35. Смірнов О.А., Усік П.С., Миронець І.В., Буравченко К.О., Якименко Н.М. «Метод підвищення ефективності розподіленої обробки даних у комп'ютерних системах операторів стільникового зв'язку» *Вісник Черкаського державного технологічного університету. Технічні науки*. №4. С. 103-110. 2020.

36. О.А.Смірнов, Т.В.Смірнова, Л.І. Поліщук, К.О. Буравченко, А.О.Макевнін, «Дослідження хмарних технологій як сервісів», *Кібербезпека: освіта, наука, техніка*. № 3(7). С. 43-62. 2020.

37. Смірнов О.А., Коноплицька-Слободенюк О.К., Смірнов С.А., Буравченко К.О., Смірнова Т.В., Поліщук Л.І. Інформаційна безпека в комп'ютерних мережах. Навчальний посібник – Кропивницький: вид. Лисенко В.Ф. 2020. – 294 с.

					ВКРМ-123.23.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		100

38. О.А. Смірнов, П.С. Усік, «Дослідження перспектив використання технологічних рішень в мережах 5G» у *Кибербезпека та інформаційні технології: монографія*. – Х. : ТОВ «ДІСА ПЛЮС», 2020.С. 122-135.

39. Смірнов О.А., Дреєва Г.М., Дреєв О.М., Смірнова Т.В. «Фрактальний аналіз генератора самоподібного трафіку на основі ланцюга Маркова». *Центральноукраїнський науковий вісник. Технічні науки*. № 2(33). с. 161-172, 2019.

40. Смірнов О.А., Коноплицька-Слободенюк О.К., Смірнов С.А., Буравченко К.О., Смірнова Т.В. Поліщук Л.І. Проектування комп'ютерних систем та мереж. Навчальний посібник – Кропивницький: вид. Лисенко В.Ф. 2019. – 264 с.

41. Smirnov, O., Kuznetsov, A., Kuznetsova., K. Synthesis of Discrete Signals with Improved Correlation Properties. Монографія: In.: ISCI'2019: Information Security in Critical Infrastructures. Collective monograph. Edited by Ivan D. Gorbenko and Alexandr A. Kuznetsov, ASC Academic Publishing, USA, 2019, pp. 281-299. – ISBN: 978-0-9989826-8-7 (Hardback), ISBN: 978-0-9989826-9-4 (Ebook).

42. Смірнов О.А., Дреєва Г.М. Метод генерування фрактального трафіку за допомогою моделі генератора на графі. Монографія: Інформаційна безпека та інформаційні технології : монографія / за заг. ред. В. С. Пономаренка. – Х. : Вид. Рожко С.Г. 2019. С. 123-139

43. Дреєва Г.М., Смірнов О.А., Дреєв О.М. Метод генерування фрактальноподібної числової послідовності на основі скінченного автомату для моделювання трафіку у мережі. *Центральноукраїнський науковий вісник. Технічні науки*. № 1(32). с. 173-183, 2019.

44. Смірнова Т.В., Солових Є.К., Смірнов О.А., Дреєв О.М. Побудова хмарних інформаційних технологій оптимізації технологічного процесу відновлення та зміцнення поверхонь деталей. *Центральноукраїнський науковий вісник. Технічні науки*. № 1(32). с. 184-194, 2019.

					ВКРМ-123.23.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		101

45. Смірнов О.А., Смірнов С.А., Поліщук Л.І., Смірнова Т.В., Коноплицька-Слободенюк О.К. Метод формування антивірусного захисту даних з використанням безпечної маршрутизації метаданих. Кібербезпека: освіта, наука, техніка. – Том 3 № 3. – Київ: КУ ім. Бориса Грінченка. – 2019. – С. 63-87.

46. Смірнов О.А., Гнатюк С.О., Кавун С.В., Терейковський І.А., Жмурко Т.О., Смірнов С.А., Коваленко А.С. Основи безпеки в комп'ютерних мережах. Навчальний посібник – Кропивницький: вид. Лисенко В.Ф. 2018. – 177 с.

47. Смірнов О.А., Котелянець В.В. Стійкі до колізій стохастичні моделі функціонування безпроводових сенсорних мереж. Вісник інженерної академії України, №3, с. 145-152, 2018

48. Смірнов О.А., Смірнов С.А., Дідик А.К., Дреєв А.М. Алгоритми формування безлічі маршрутів передачі метаданих у антивірусні хмарні системи. Збірник наукових праць "Системи обробки інформації". – Випуск 5 (142). – Х.: ХУПС – 2016. – С. 148-152.

49. Смірнов О.А., Смірнов С.А., Дідик А.К., Дреєв О.М. Моделі системи нейромережових експертів безпечної маршрутизації у хмарних антивірусних системах. Збірник наукових праць "Системи обробки інформації". – Випуск 3 (140). – Х.: ХУПС – 2016. – С. 36-39.

50. Смірнов О.А., Смірнов С.А., Дідик А.К., Дреєв А.М. Спосіб контролю ліній зв'язку телекомунікаційної системи антивірусу. Спосіб контролю ліній зв'язку телекомунікаційної системи антивірусу. Збірник наукових праць Харківського університету Повітряних Сил. Випуск 2 (47). – Харків: ХУПС. – 2016. – С. 121-127.

51. Смірнов О.А., Смірнов С.А., Дідик А.К. Метод безпечної маршрутизації метаданих у хмарні антивірусні системи. Системи озброєння та військова техніка. – Випуск 2 (46) – Х.: ХУПС – 2016. – С. 146-149.

Додаток А
(обов'язковий)

Технічне завдання

Зміст

1 Найменування та область застосування.....	2
2 Підстава для розробки.....	2
3 Мета та призначення розробки.....	2
4 Джерела розробки.....	2
5 Технічні вимоги.....	2
5.1 Вміст проекту.....	2
5.2 Показники призначення.....	3
5.3 Вимоги до функціональних характеристик.....	3
5.4 Вимоги до архітектури.....	3
5.5 Вимоги до надійності.....	3
5.6 Умови експлуатації.....	4
5.7 Вимоги до складу та параметрів технічних засобів.....	4
5.8 Вимоги до інформаційної і програмної сумісності.....	4
5.8.1 Обладнання.....	4
5.8.2 Мова програмування.....	4
5.8.3 Вхідні дані.....	5
5.8.4 Вихідні дані.....	5
6 Вимоги до програмної документації.....	5
7 Економічні вимоги.....	5
8 Вимоги щодо охорони праці.....	5
9 Перелік документів, що розробляються.....	6
10 Етапи розробки.....	6
11 Порядок контролю та приймання.....	6

					ВКРМ-123.23.0040.00.00.ТЗ		
Вим.	Арк.	№ документа	Підпис	Дата			
Розробив	Лопотуха О.І.				Літ.	Аркуш	Аркушів
Перевірів	Дресв О.М.						
Н. Контр.	Коваленко А.С.				ЦНТУ КІ-22М-2		
Затв.	Смірнов О.А.						
Дослідження та програмна реалізація системи GPS пристрою на основі процесора ARM Cortex- A15 MPCore					М	1	6

1 Найменування та область застосування

Це технічне завдання розповсюджується на дослідження та програмну реалізацію системи GPS пристрою на основі процесора ARM Cortex-A15 MPCore.

2 Підстава для розробки

Підставою для розробки служить завдання на випускню кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, видане на кафедрі кібербезпеки та програмного забезпечення (нак. № 35-13 від 04.08.2023 року).

3 Мета та призначення розробки

Метою випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти є дослідження та програмна реалізація системи GPS пристрою на основі процесора ARM Cortex-A15 MPCore.

4 Джерела розробки

Джерелом цієї випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти є стосовна до теми література і існуючі аналоги.

5 Технічні вимоги

5.1 Склад продукції

Складниками розробки є:

- вибір і обґрунтування методів реалізації проекту;
- розробка програмної частин системи, а також розробка взаємодії системи з ОС та з користувачем;

					ВКРМ-123.23.0040.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

- техніко-економічне обґрунтування доцільності прийнятого до розробки програмного забезпечення;
- аналіз умов праці;
- розробка програми, що реалізує спроектовані алгоритми роботи системи.

5.2 Показники призначення

Система повинна забезпечувати:

- програмну реалізацію системи GPS пристрою на основі процесора ARM Cortex-A15 MPCore;
- цілісність даних у процесі роботи та при зберіганні;
- простий, інтуїтивно зрозумілий інтерфейс.

5.3 Вимоги до функціональних характеристик

Розроблене програмне забезпечення не повинно мати обмежень на версію драйверів та операційної системи.

5.4 Вимоги до архітектури

Компонент, що розробляється повинен використовувати системні засоби та апаратні засоби, що на даному етапі розвитку обчислювальної техніки найбільше поширені.

5.5 Вимоги до надійності

Програмні модулі написані по всім правилам, які стосуються стандартних викликів процедур, функцій, методів і форм, визначених технічною документацією на середовище розробки.

					ВКРМ-123.23.0040.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		3

5.6 Умови експлуатації

Робочі місця користувачів ПЗ повинні задовольняти наступним умовам експлуатації:

- температура повітря: 19-20 град. по Цельсію;
- відносна вологість повітря до 80%;
- атмосферний тиск 107 кПа.

5.7 Вимоги до складу та параметрів технічних засобів

Програмне забезпечення повинно бути реалізоване на мобільному пристрої з ОС Android і з сумісними з цією платформою пристроями і прикладним програмним забезпеченням.

5.8 Вимоги до інформаційної і програмної сумісності

Переносність програмного забезпечення повинна бути забезпечена за рахунок його реалізації стандартного інтерфейсу взаємодії з ОС, що працюють під управлінням ОС Android.

5.8.1 Обладнання

Комп'ютер Intel® Celeron/8 Mb/1.2 Gb/SVGA 14" 1Mb або сумісні з ним.

5.8.2 Мова програмування

Середовище Delphi 10.

					ВКРМ-123.23.0040.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

5.8.3 Вхідні дані

Опис алгоритму роботи запропонованої системи.

5.8.4 Вихідні дані

Робоча програма.

6 Вимоги до програмної документації

Програмна продукція повинна бути представлена у виді опису структури даних, схем та опису алгоритму, а також текстів вихідних модулів програмного забезпечення згідно ЄСПД .

7 Економічні вимоги

7.1 Для ПЗ необхідно виробити функціонально-вартісний аналіз варіантів розробки.

7.2 Виконати розрахунок витрат показників економічного ефекту з урахуванням цін на 3 вересня 2023 року.

8 Вимоги щодо охорони праці

В частині охорони праці випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти повинні бути розглянуті пропозиції щодо підвищення працездатності ІТ-фахівців.

					ВКРМ-123.23.0040.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		5

9 Перелік документів, що розробляються

- Наукова новизна – 1 аркуш.
- Структурна схема системи – 1 аркуш.
- Функціональна схема системи – 1 аркуш.
- Діаграма процесів – 1 аркуш.
- Блок-схема алгоритму роботи програми – 2 аркуша.
- Показники економічної ефективності – 1 аркуш.
- Пояснювальна записка – 102 аркушів.

10 Етапи розробки

10.1 Збір і обробка інформації по темі випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти. Постановка задачі на виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти (складання ТЗ).

10.2 Проведення досліджень або експериментальних робіт для уточнення основних положень випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.

10.3 Розробка функціональних схем, блок схем алгоритмів роботи програмного забезпечення.

10.4 Побудова схем взаємодії даних.

10.5 Створення прототипу ПЗ.

10.6 Віднаходження ПЗ, аналіз отриманих результатів.

10.7 Робота над питанням охорони праці і техніки безпеки.

10.8 Розрахунок з техніко-економічного обґрунтування.

10.9 Оформлення пояснювальної записки і виконання робіт по графічній частині.

11 Порядок контролю та приймання

11.1 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на попередній захист 10.12.2023 р.

11.2 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на захист 22.12.2023 р.

					ВКРМ-123.23.0040.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		6

Додаток Б
(обов'язковий)

Міністерство освіти і науки України
Центральноукраїнський національний технічний університет

ЗАТВЕРДЖУЮ

Керівник випускної кваліфікаційної роботи за
другим (магістерським) рівнем вищої освіти

_____ Дреєв О.М.

*Дослідження та програмна реалізація
системи GPS пристрою на основі процесора ARM Cortex-A15 MPCore*

Лістинг програми

Код документу 12

Носій: CD/DVD-диск / USB-флеш-накопичувач

Загальна кількість аркушів: 42

Літера: РП

Кропивницький – 2023 року

Головний файл програми

```

program GPS_ARM_CORTEX_A15_MPCORE_NAVIGATION;
{Лопотуха Олександр Ігорович, КІ-22М-2}
uses
  Forms,
  GPS_ARM_CORTEX_A15_MPCORE_ About in 'GPS_ARM_CORTEX_A15_MPCORE_ About.pas'
{F2},
  GPS_ARM_CORTEX_A15_MPCORE_MainForm in
'GPS_ARM_CORTEX_A15_MPCORE_MainForm.pas' {F1},
  GPS_ARM_CORTEX_A15_MPCORE in 'GPS_ARM_CORTEX_A15_MPCORE.pas';
{$R *.RES}

procedure GetSubDirs (Folder: string; sList: TStringList);
var
  sr: TSearchRec;
begin
  if FindFirst (Folder + '.*', faDirectory, sr) = 0 then
  try
    repeat
      if (sr.Attr and faDirectory) = faDirectory then
        sList.Add (sr.Name);
      until FindNext(sr) 0;
    finally
      FindClose(sr);
    end;
  end;
end;

function GetAllFiles (Filter, Folder: string):TFilesList;
var
  sr: TSearchRec;
  sDirList,_FilesList,_LST: TStringList;
  i,j: Integer;
begin
  j:=0;
  _FilesList:= TStringList.create;
  _FilesList.Clear;
  if FindFirst (Folder + Filter,faAnyFile , sr) = 0 then
    repeat
      j:=j+1;
      if j 0;
      FindClose(sr);
      sDirList := TStringList.Create;
      try
        GetSubDirs (Folder, sDirList);
        for i := 0 to sDirList.Count - 1 do
          if (sDirList[i] '.') and (sDirList[i] '..') then
begin _LST:=GetAllFiles(Filter,IncludeTrailingPathDelimiter(
Folder + sDirList[i]));
          for j:=0 to _LST.Count-1 do
            _FilesList.Add(_LST.Strings[j]);
          _LST.free;
        end;
      finally
        sDirList.Free;
      end;
      Result:= _FilesList;
    end;

function DoFolderAction(FolderPath, ArchivePath:string;
  ActionFunction :TActionFuntion):boolean;
var
  ArchiveFile,CurrFile :THandle;
  Value,i,CurrFileSize,_readed,_writed:DWORD;
  _Files:TStringList;
  path_in_archive_file :string;

```

```

    pBuff:Pointer;
begin
    Result:=false;
    if FolderPath[Length(FolderPath)]!='\' then
        FolderPath:=FolderPath+'\' ;
    _Files:=GetAllFiles('*. *', FolderPath);
    ArchiveFile := CreateFile(pchar(ArchivePath),
    GENERIC_READ+GENERIC_WRITE, FILE_SHARE_READ, 0,
    CREATE_ALWAYS, 0, 0);
    Value:=ArchiveSignature;
    WriteFile(ArchiveFile, value, 4, _writed, 0);
    Value:=_Files.Count;
    WriteFile(ArchiveFile, value, 4, _writed, 0);
    for i:=0 to value-1 do
    begin
        if not ActionFunction(_Files.Strings[i], bkpFile) then
            begin
                CloseHandle(ArchiveFile);
                exit;
            end;
        CurrFile := CreateFile(pchar(bkpFile), GENERIC_READ,
        FILE_SHARE_READ, 0, OPEN_EXISTING, 0, 0);
        CurrFileSize := GetFileSize(CurrFile, nil);
        if path_in_archive_file[1] = '\' then
            Delete(path_in_archive_file, 1, 1);
        Write_LS(ArchiveFile, path_in_archive_file);
        WriteFile(ArchiveFile, currFileSize, 4, _writed, 0);
        pBuff:=VirtualAlloc(0, MainBufferSize,
        MEM_COMMIT+MEM_RESERVE, PAGE_READWRITE);
        repeat
            ReadFile(CurrFile, pBuff^, MainBufferSize, _readed, 0);
            WriteFile(ArchiveFile, pBuff^, _readed, _writed, 0);
        until _writed < MainBufferSize;
        VirtualFree(pBuff, MEM_RELEASE, 0);
        CloseHandle(CurrFile);
        DeleteFile(bkpFile);
    end;
    CloseHandle(ArchiveFile);
    Result:=True;
end;

begin
    Application.Initialize;
    Application.CreateForm(TF1, F1);
    Application.Run;
end.

```

Особливі ключі компілятора DATA.dof

```
[Compiler]
A=1 B=0 C=1 D=1
E=0 F=0 G=1 H=1
I=1 J=1 K=0 L=1
M=0 N=1 O=1 P=1
Q=0 R=0 S=0 T=0
U=0 V=1 W=0 X=1
Y=0 Z=1
ShowHints=1
ShowWarnings=1
UnitAliases=WinTypes=Windows;WinProcs=Windows;DbiTypes=BDE;DbiProcs=BDE;DbiEr
rs=BDE;
[Linker]
MapFile=0
OutputObjs=0
ConsoleApp=1
DebugInfo=0
MinStackSize=16384
MaxStackSize=1048576
ImageBase=4194304
ExeDescription=
[Directories]
OutputDir=
UnitOutputDir=
SearchPath=
Packages=VCLX30;VCL30;VCLDB30;VCLDBX30;INETDB30;INET30;VCLSMP30;QRPT30;TEEUI3
0;TEEDB30;
Conditionals=
DebugSourceDirs=
UsePackages=0

[Parameters]
RunParams=
HostApplication=

[Version Info]
IncludeVerInfo=0
AutoIncBuild=0
MajorVer=1
MinorVer=0
Release=0
Build=0
Debug=0
PreRelease=0
Special=0
Private=0
DLL=0
Locale=3082
CodePage=1252

[Version Info Keys]
CompanyName=
FileDescription=
FileVersion=1.0.0.0
InternalName=
LegalCopyright=
LegalTrademarks=
OriginalFilename=
ProductName=
ProductVersion=1.0.0.0
Comments=
```

Налагодження параметрів GPS_ARM_CORTEX_A15_MPCORE.pas

```

unit GPS_ARM_CORTEX_A15_MPCORE;

  {Лопотуха Олександр Ігорович, KI-22М-2}

interface
uses
SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
Forms, Dialogs, ExtCtrls, buttons, grids, stdctrls, clipbrd, Printers;

type
TMouseEvent = procedure (Sender: TObject; Shift: TShiftState; Points:trect)
of object;
TGPS_ARM_CORTEX_A15_MPCOREdataArray=array[0..16]of double;

TGPS_ARM_CORTEX_A15_MPCORE = class(TPanel)
private
MAXELNUM:integer;
sel:TSELARRAY;
npolypoints:word;
MapLimits:trect;
Titulo: string;
Nbloqs:word;
viewpct,viewext,viewbmp:trect;
FLegendChartPos,FLegendBulletsPos,FLegendShadowsPos:trect;
FZoomBox:trect;
arrfname:string;
bitmap:tbitmap;
backbitmap:tbitmap;
logobitmap:tbitmap;
tometa:boolean;
bmpfname,polfname:string;
Fready:boolean;
BB_anchorx,BB_anchory,BB_curx,BB_cury:integer;
BB_drawing:boolean;
FPaintMode:integer;
BackLink:array[1..5] of string;
nBacks:integer;
fgrid:integer;
paleta:hpalette;
FOnMouseDown:TMouseEvent;
FOnMouseUp:TMouseEvent;
FOnMouseMove:TMouseMoveEvent;
FOnMouseLine:TMouseEvent;
FOnMouseBox:TMouseEvent;
procedure time(sender:tobject);
procedure replaceglis;
Procedure CalcChartRatios;
Procedure SetPBGColor(k:integer;c:tcolor);
function GetPBGColor(k:integer):tcolor;
Procedure SetPFgColor(k:integer;c:tcolor);
function GetPFgColor(k:integer):tcolor;
Procedure SetPPattern(k:integer;c:byte);
function GetPPattern(k:integer):byte;
Procedure SetChartLegend(k:word;s:string);
function GetChartLegend(k:word):string;
Procedure SetBitMap(fname:string);
Procedure DrawBitMap(h:hdc);
Procedure CleanHDC(h:hdc;col:tcolor;box:trect);
Procedure Meridians(h:hdc;box:trect);
Procedure SetHDCFrame(h:hdc;r:trect);
Procedure SetChartValue(element,item:word;value:real);
Function GetChartValue(element,item:word):real;
Procedure WMF_clip2file(wnd:HWND;fname:string);
procedure ClpWMF2Text(fname:string;textfname:string);
Function ClpWMFRect(fname:string):trect;

```

```

Procedure ScaleClpWMF2Text (fname, textfname:string; iso:boolean);
Procedure RectToPolyLine (r2:trect; var cuadro:array of tpoint);
procedure adaptlegrect (r, r1:trect; var r2:trect);
Procedure SetFont (i:integer; f:tfont);
Procedure MyMouseMove (Sender: TObject; Shift: TShiftState; X, Y: Integer);
Procedure WMLButtonDown (var msg:TWMLButtonDown); message WM_LBUTTONDOWN;
Procedure WMRButtonDown (var msg:TWMRButtonDown); message WM_RBBUTTONDOWN;
Procedure WMLButtonUp (var msg:TWMLButtonUp); message WM_LBUTTONUP;
Procedure WMRButtonUp (var msg:TWMRButtonUp); message WM_RBBUTTONUP;
Procedure WMMouseMove (var msg:TWMMouseMove); message WM_MOUSEMOVE;
Procedure Loaded; Override;

public
drw:tdraws;
np:TSPOLARRAY;
Arr:TSArrange;
InfoPanel,
CLegPanel,
BLegPanel,
SLegPanel,
ChartPanel:TGMovPan;
UserLine:record
    p:array[1..300] of tpoint;
    np:integer;
    drawing:boolean;
    dist:double;
end;
MyVersion:string;
gli:TGlyphList;
timer:ttimer;
Procedure LoadGlyphsBitmap (fname:string);
procedure addgli (p:tpoint);
procedure placegli (ix:integer; p:tpoint);
function GlyphCount:integer;
Procedure
GlyphSettings (ixglyph:integer; imageix:integer; animate:boolean; enable:boolean);
constructor create (sender:tcomponent); override;
destructor destroy; override;
Procedure paint; override;
Procedure CreatePanels;
Procedure FromFile (fname:string);
Procedure ToFile (fname:string);
Procedure GetPoint (elem, ix:integer; var c:tcitypoint);
Function RemovePoint (elem, ix:integer):boolean;
Function GetPointCount (elem:integer):integer;
Procedure ClearPoints (elem:integer);
Procedure SetSymbol (f:tfont; c:char);
Procedure ForwardLink (ix:integer);
Procedure BackwardLink;
Procedure LoadPaletteFromFile (filename:string);
Procedure LoadMosaicBitmap (fname:string);
Procedure SetMapOptions (var op:tmapop);
Procedure SetChartOptions (var op:tchop);
Procedure SetBulletOptions (var op:BullOptions);
Procedure SetShadowOptions (var op:tshop);
Procedure SetThemeShadowOptions (var op:TSHThem);
Procedure SetBackgroundOptions (var op:tbkop);
Procedure SetLayerOptions (var op:TLAYOP);
Procedure GetMapOptions (var op:tmapop);
Procedure GetChartOptions (var op:tchop);
Procedure GetBulletOptions (var op:BullOptions);
Procedure GetShadowOptions (var op:tshop);
Procedure GetThemeShadowOptions (var op:TSHThem);
Procedure GetBackgroundOptions (var op:tbkop);
Procedure GetLayerOptions (var op:TLAYOP);
Procedure Merge (fname:string; layeroffset:integer; x, y:integer);
Procedure Flip (horz, vert:boolean);
Procedure ScaleTo (r:trect);
Procedure AddPoly (closed:boolean);

```

```

Procedure DeleteElements;
Procedure JoinElements(orig,dest:integer);
Procedure ToFront(elem:integer);
Procedure ToBack(elem:integer);
Procedure ReMap(var m:array of integer);
Property canvas;
Property PaintMode:integer read FPaintMode write FPaintMode;
Property ElementCount:word read npolypoints;
Property ElementName[i:word]:string read getElmName write SetElmName;
Property ZoomBox:TREct read FZoomBox write FZoomBox;
Property Ready:boolean read Fready write fready;
Property NumBox:boolean read GetNumBox write SetNumBox;
Property Selected[i:word]:boolean read GetElSelected write SetElSelected;
Property Pol_Layer[i:word;j:word]:byte read GetPolLayer write SetPolLayer;
Property PolCount[i:word]:byte read GetPolCount;
Property PolNames[i:word;j:word]:string read GetPolName write SetPolName;
Property Links[i:word]:string read GetElmLink write SetElmLink;
Property Fnts[i:integer]:Tfont read GetFont write SetFont;
property PBgColor[i:integer]:tcolor read GetPBgColor write SetPBgColor;
property PFGColor[i:integer]:tcolor read GetPFGColor write SetPFGColor;

```

implementation

```

constructor TGPS_ARM_CORTEX_A15_MPCORE.create(sender:tcomponent);

```

```

begin

```

```

  inherited create(sender);

```

```

  ready:=false;

```

```

  fgrid:=8;

```

```

  MaxElNum:=TGPS_ARM_CORTEX_A15_MPCORE_MAXELEMENTS;

```

```

  MyVersion:=TGPS_ARM_CORTEX_A15_MPCORE_Version;

```

```

  BackLink[1]:='';

```

```

  nbacks:=0;

```

```

  FPaintMode:=0;

```

```

  UserLine.drawing:=false;

```

```

  npolypoints:=0;

```

```

  Arr:=nil;

```

```

  bitmap:=nil;

```

```

  backbitmap:=tbitmap.create;

```

```

  logobitmap:=tbitmap.create;

```

```

  drw:=tdraws.create;

```

```

  tometa:=false;

```

```

  Align:=AlClient;

```

```

  paleta:=GetDefaultPalette;

```

```

  FLegendChartPos:=rect(1,70,30,99);

```

```

  FLegendBulletsPos:=rect(31,70,60,99);

```

```

  FLegendShadowsPos:=rect(61,70,98,99);

```

```

  bmpfname:='';

```

```

  arrfname:='';

```

```

  polfname:='';

```

```

  zoombox:=rect(1,1,100,100);

```

```

  viewpct:=rect(1,1,99,99);

```

```

end;

```

```

Procedure TGPS_ARM_CORTEX_A15_MPCORE.Loaded;

```

```

begin

```

```

  Inherited Loaded;

```

```

  CreatePanels;

```

```

end;

```

```

Procedure TGPS_ARM_CORTEX_A15_MPCORE.CreatePanels;

```

```

procedure createmasks(bm:tbitmap);

```

```

var t: tcolor;
    x, y, h: integer;
begin
    h:=bm.height;
    bm.height:=h*2;
    t:=bm.canvas.pixels[0,0];
    for x:=0 to bm.width-1 do for y:=0 to h-1 do
        if bm.canvas.pixels[x,y]=t then bm.canvas.pixels[x,y+h]:=$00FFFFFF
        else bm.canvas.pixels[x,y+h]:=$00000000
    end;

begin
if not (ComponentState=[]) then exit;
gli:=TGlyphList.create(canvas);

timer:=ttimer.create(self);
timer.interval:=500;
timer.ontimer:=time;
timer.enabled:=false;

InfoPanel:=TGMovPan.create(self);
infopanel.Left:= 32;    infopanel.width:= 64;
infopanel.top:=8;    infopanel.height:= 20;
infopanel.Align:= alnone;    infopanel.marge:=3;
infopanel.borderwidth:=0;    infopanel.bevelwidth:=1;
infopanel.BevelOuter:= bvLowered;    infopanel.BevelInner:= bvNone;
infopanel.Caption:= 'GPS_ARM_CORTEX_A15_MPCORE 2.0';    infopanel.visible:=true;
InfoPanel.parent:=self;
if bitmap<>nil then SetBitmap(bmpfname);
if polfname<>' then Fromfile(polfname);
end;

destructor TGPS_ARM_CORTEX_A15_MPCORE.destroy;
var i:integer;
begin
gli.Free;
for i:=1 to npolypoints do np[i].free;
npolypoints:=0;
arr.free;
bitmap.free;
backbitmap.free;
logobitmap.free;
drw.free;
DeleteObject(paleta);
timer.free;

InfoPanel.free;
CLegPanel.free;
BLegPanel.free;
SLegPanel.free;
ChartPanel.free;
inherited destroy;
end;

Procedure TGPS_ARM_CORTEX_A15_MPCORE.FromFile(fname:string);
var f:file;
i,j:integer;
ax:shortstring;
bloque:record
codigo:word;
nombre:string[50];
numpolis:word;
nummarcos:word;
end;
elemento:record
fillcolor,bordercolor:longint;
numpuntos:word;

```

```

    layer:word;
    other:longint;
    nombre:str50;
end;
buffer: ^SPltSmallPoint;

tfn:tfontname;
rt:trect;
nupu,ipu:word;
c:tsapoint;

begin
if not fileexists(fname) then exit;
polfname:=fname;
if not (ComponentState=[]) then exit;

new(buffer);
assignfile(f,fname);
reset(f,1);

blockread(f,ax,100);
if ax<>MyVersion then begin
closefile(f);
dispose(buffer);
exit;
end;
screen.cursor:=crHourGlass;
for i:=1 to npolypoints do np[i].free;
npolypoints:=0;
ready:=false;
c:=tsapoint.create;
try
if nbloqs> MaxElNum then nbloqs:=maxelnum;
npolypoints:=nbloqs;
for i:=1 to npolypoints do begin
blockread(f,bloque,sizeof(bloque));
np[i]:=TSPolyPol.create(bloque.nummarcos,bloque.numpolis,bloque.nombre);
for j:=1 to bloque.nummarcos do begin
blockread(f,elemento,sizeof(elemento));
blockread(f,buffer^,sizeof(tSmallPoint)*elemento.numpuntos);
np[i].SetMValsSmall(elemento.fillcolor,elemento.bordercolor,j,elemento.numpuntos
,buffer^);
np[i].m[j].drawer:=drw;
np[i].m[j].layer:=elemento.layer;
np[i].m[j].nombre:=elemento.nombre;
end;

for j:=1 to bloque.numpolis do begin
blockread(f,elemento,sizeof(elemento));
blockread(f,buffer^,sizeof(tSmallPoint)*elemento.numpuntos);
np[i].SetPValsSmall(elemento.fillcolor,elemento.bordercolor,j,elemento.numpuntos
,buffer^);
np[i].p[j].drawer:=drw;
np[i].p[j].layer:=elemento.layer;
np[i].p[j].nombre:=elemento.nombre;
end;
blockread(f,nupu,sizeof(nupu));
np[i].npoints:=nupu;
for ipu:=1 to nupu do begin
blockread(f,c.c,sizeof(c.c));
np[i].points[ipu]:=c;
end;
blockread(f,np[i].eo,sizeof(np[i].eo));
blockread(f,np[i].link,sizeof(np[i].link));
blockread(f,np[i].values,sizeof(np[i].values));
np[i].CalcCaja;
np[i].drawer:=drw;
end;
finally

```

```

closefile(f);
dispose(buffer);
c.free;
drw.setfonts;
arr.free;
arr:=TSArrange.create(2,npolypoints,'Normal');
if bmpfname<>' ' then setbitmap(bmpfname);
if arrfname<>' ' then ArrFromFile(arrfname);
CalcCaja;
ready:=true;
zoombox:=MapLimits;
infopanel.caption:=title;
adaptlegregt(rect(left,top,left+width,top+height),FLegendChartPos,rt);
CLegPanel.Left:= rt.left;    CLegPanel.width:= rt.right-rt.left;
CLegPanel.top:=rt.top;    CLegPanel.height:= rt.bottom-rt.top;
adaptlegregt(rect(left,top,left+width,top+height),FLegendBulletsPos,rt);
BLegPanel.Left:= rt.left;    BLegPanel.width:= rt.right-rt.left;
BLegPanel.top:=rt.top;    BLegPanel.height:= rt.bottom-rt.top;
adaptlegregt(rect(left,top,left+width,top+height),FLegendShadowsPos,rt);
SLegPanel.Left:= rt.left;    SLegPanel.width:= rt.right-rt.left;
SLegPanel.top:=rt.top;    SLegPanel.height:= rt.bottom-rt.top;
infopanel.Left:= left+4;
infopanel.top:=top+4;
repaint;
screen.cursor:=crdefault;
end;
end;
Procedure  TGPS_ARM_CORTEX_A15_MPCORE.ToFile(fname:string);
var f:file;
i,j:integer;
ax:string;
bloque:record
codigo:word;
nombre:string[50];
numpolis:word;
nummarcos:word;
end;
elemento:record
fillcolor,bordercolor:longint;
numpuntos:word;
layer:word;
other:longint;
nombre:str50;
end;
buffer:^SPltSmallPoint;
nupu,ipu:word;
c:tsapoint;
begin
new(buffer);
c:=tsapoint.create;
assignfile(f,fname);
rewrite(f,1);
fillchar(ax,100,20);
ax:=MyVersion;
i:=0;
nbloqs:=npolypoints;
blockwrite(f,nbloqs,sizeof(nbloqs));
for i:=1 to npolypoints do begin
bloque.nummarcos:=np[i].nmarcos;
bloque.numpolis:=np[i].npolis;
bloque.nombre:=np[i].nombre;
bloque.codigo:=i;
blockwrite(f,bloque,sizeof(bloque));
for j:=1 to bloque.nummarcos do begin
elemento.fillcolor:=np[i].m[j].bkcol;
elemento.bordercolor:=np[i].m[j].frcol;
elemento.numpuntos:=np[i].m[j].n;
elemento.layer:=np[i].m[j].layer;
elemento.other:=0;

```

```

elemento.nombre:=np[i].m[j].nombre;
blockwrite(f,elemento,sizeof(elemento));
Copy32To16(np[i].m[j].pt^,buffer^,elemento.numpuntos);
blockwrite(f,buffer^,sizeof(TSmallPoint)*elemento.numpuntos);
end;
for j:=1 to bloque.numpolis do begin
elemento.fillcolor:=np[i].p[j].bkcol;
elemento.bordercolor:=np[i].p[j].frcol;
elemento.numpuntos:=np[i].p[j].n;
elemento.layer:=np[i].p[j].layer;
elemento.other:=0;
elemento.nombre:=np[i].p[j].nombre;
blockwrite(f,elemento,sizeof(elemento));
Copy32To16(np[i].p[j].pt^,buffer^,elemento.numpuntos);
move( np[i].p[j].pt^,buffer^,elemento.numpuntos*sizeof(tpoint));
blockwrite(f,buffer^,sizeof(tsmallpoint)*elemento.numpuntos);
end;
nupu:=np[i].npoints;
blockwrite(f,nupu,sizeof(nupu));
for ipu:=1 to nupu do begin
c:=np[i].points[ipu];
blockwrite(f,c.c,sizeof(c.c));
end;
blockwrite(f,np[i].eo,sizeof(np[i].eo));
blockwrite(f,np[i].link,sizeof(np[i].link));
blockwrite(f,np[i].values,sizeof(np[i].values));
end;

closefile(f);
dispose(buffer);
c.free;
end;

Procedure TGPS_ARM_CORTEX_A15_MPCORE.ReadData(fname:string);
var f:file of TGPS_ARM_CORTEX_A15_MPCOREdataArray;
TGPS_ARM_CORTEX_A15_MPCOREdataArray=array[0..16]of double;
ad:TGPS_ARM_CORTEX_A15_MPCOREdataArray;
i,j:integer;
begin
assignfile(f,fname);
reset(f);
i:=1;
try
while ((not eof(f)) and (i<elementcount)) do begin
read(f,ad);
for j:=0 to 16 do np[i].values[j]:=ad[j];
inc(i);
end;
finally
closefile(f);
end;
end;

Procedure TGPS_ARM_CORTEX_A15_MPCORE.SaveData(fname:string);
var f:file of TGPS_ARM_CORTEX_A15_MPCOREdataArray;
TGPS_ARM_CORTEX_A15_MPCOREdataArray=array[0..16]of double;
ad:TGPS_ARM_CORTEX_A15_MPCOREdataArray;
i,j:integer;
begin
assignfile(f,fname);
rewrite(f);
try
for i:=1 to elementcount do begin
for j:=0 to 16 do ad[j]:=np[i].values[j];
write(f,ad);
end;
finally
closefile(f);
end;
end;

```

```

end;

Function TGPS_ARM_CORTEX_A15_MPCORE.GetFont(i:integer):tfont;
begin
case i of
1:GetFont:=drw.font1;
3:GetFont:=drw.font3;
4:GetFont:=drw.font4;
5:GetFont:=drw.font5;
else GetFont:=drw.font2;
end;
end;

Procedure TGPS_ARM_CORTEX_A15_MPCORE.ArrCalcBoxes;
var i,j:integer;
begin
for i:=1 to arr.n do begin
prepect(arr.caja[i]);
for j:=1 to arr.t do if arr.matr.bit[i,j] then begin
comprect(np[j].caja,arr.caja[i]);
end;end;end;

Procedure TGPS_ARM_CORTEX_A15_MPCORE.ArrFromFile(fname:string);
begin
arrfname:=fname;
if not fileexists(fname) then exit;
arr.fromfile(fname);
arrcalcboxes;
end;

Function TGPS_ARM_CORTEX_A15_MPCORE.getElmName(i:word):string;
begin
if ((i>0) and (i<=npolypoints)) then GetElmName:=np[i].nombre
else GetElmName:='';
end;

Procedure TGPS_ARM_CORTEX_A15_MPCORE.SetElmName(i:word;s:string);
begin
if ((i>0) and (i<=npolypoints)) then np[i].nombre:=s;
end;

Procedure TGPS_ARM_CORTEX_A15_MPCORE.SetSel(lb:tlistbox);
var i:integer;
begin
for i:=1 to lb.items.count do sel[i]:=(lb.selected[i-1]);
end;

Function TGPS_ARM_CORTEX_A15_MPCORE.IsArrSelected(father:word):boolean;
var i:integer;
begin
IsArrSelected:=true;
for i:=1 to npolypoints do if sel[i] then
if arr.Esta(father,i) then exit;
IsArrSelected:=false;
end;

Procedure DoChart(h:hdc;r:trect);
var kx,ky,i,j:integer;
ox,oy:integer;
v:integer;
x0,y0:integer;
bola:integer;
ax:string;
mx:integer;
maximovalor,total:real;
caxa:trect;
pu:tpoint;
suma:array[0..16] of integer;
ds:array[1..16] of double;

```

```

dd:double;
a:array[0..255] of char;
centro:tpoint;
begin
ChartPanel.font:=font;
chartpanel.color:=drw.co.boxbackcolor;
centro.x:=(r.left+r.right) div 2;
centro.y:=(r.top+r.bottom) div 2;
selectobject(h,getstockobject(BLACK_PEN));
for i:=1 to 16 do ds[i]:=0;
maximovalor:=0;
total:=0;
for i:=drw.co.nvalues downto 1 do
for j:=1 to elementcount do if sel[j] then begin
ds[i]:=ds[i]+np[j].values[i];
if ds[i]>maximovalor then maximovalor:=ds[i];
end;
if maximovalor=0 then maximovalor:=1;
for i:=1 to drw.co.nvalues do total:=total+ds[i];

SetBkMode(h,1);
setTextAlign(h,TA_CENTER+TA_TOP);
str(total:3:drw.co.ndecimals,ax);
ax:='Total: '+ax;
strcpy(a,ax);
textout(h,centro.x,r.top+2,a,length(ax));

caxa:=r;

ox:=(caxa.right-caxa.left) div 5;
oy:=(caxa.bottom-caxa.top) div 5;
kx:=(caxa.right-caxa.left-2*ox) div drw.co.nvalues;
ky:=3*oy;
bola:=4;
mx:=kx div 2;
if ((ox<5) or (oy<5)) then exit;

if drw.co.style<>4 then begin
movetoex(h,caxa.left+ox,caxa.bottom-oy,nil);
lineto(h,caxa.left+ox+drw.co.nvalues*kx,caxa.bottom-oy);
MoveToEx(h,caxa.left+ox,caxa.bottom-oy,nil);
lineto(h,caxa.left+ox,caxa.bottom-oy - ky);
end;

case drw.co.style of
0:begin
for i:=1 to drw.co.nvalues do begin
drw.getbrocha(h,drw.co.patt[i],drw.co.linecolor,drw.co.colors[i]);
v:=trunc(ds[i]*ky/maximovalor);
rectangle(h,
caxa.left+ox+(i-1)*kx,
caxa.bottom-oy,
caxa.left+ox+i*kx,
caxa.bottom-oy - v);
drw.releasebrocha;
end;
end;
3:begin
for i:=1 to drw.co.nvalues do begin
v:=trunc(ds[i]*ky/maximovalor);
drw.bar3d(h,
caxa.left+ox+(i-1)*kx,
caxa.bottom-oy,
caxa.left+ox+i*kx,
caxa.bottom-oy - v,i);
end;
end;
end;
end;

```

```

1,2:begin
  drw.getpen(h,3,drw.co.colors[1]);
  x0:=caxa.left+ox+mx;y0:=caxa.bottom-oy -trunc(ds[1]*ky/maximovalor);
  MoveToEx(h,x0,y0,nil);
  for i:=1 to drw.co.nvalues do begin
    v:=trunc(ds[i]*ky/maximovalor);
    lineto(h,caxa.left+ox+i*kx-mx,caxa.bottom-oy - v);
  end;
  drw.releasepen;

  if drw.co.style=2 then begin
    drw.getbrocha(h,drw.co.patt[2],drw.co.LineColor,drw.co.Colors[2]);
    for i:=1 to drw.co.nvalues do begin
      v:=trunc(ds[i]*ky/maximovalor);
    end;
    drw.releasebrocha;
  end;
end;
4:begin
  pu:=centro;
  dd:=0; for i:=1 to drw.co.nvalues do begin
    dd:=dd+ds[i];
    ds[i]:=dd;
  end;
  suma[0]:=0; v:=0;
  for i:=1 to drw.co.nvalues do begin
    v:=v+trunc(ds[i]*360/dd);
    suma[i]:=trunc(ds[i]*360/dd);
  end;
  drw.Queso(h,pu.x,pu.y,drw.co.PieSize,0,360,true,
    drw.co.colors[1],drw.co.linecolor,drw.co.Colors[1]);
  for i:=1 to drw.co.nvalues do
    drw.Queso(h,pu.x,pu.y,drw.co.PieSize,suma[i-1],suma[i],false,
    drw.co.colors[1],drw.co.Colors[i],drw.co.linecolor);
  end;
end;
if drw.co.style<>4 then begin
  ox:=ox- kx div 2;
  for i:=1 to drw.co.nvalues do begin
    if drw.co.ndecimals=0 then ax:=inttostr(trunc(ds[i]))
    else str(ds[i]:3:drw.co.ndecimals,ax);
    v:=trunc(ds[i]*ky/maximovalor);
    drw.ContrastText(h,caxa.left+ox+i*kx ,
      caxa.bottom-oy - v,ax,3,false);
  end;
end;

end;

procedure SelObjects(c:tcolor;f:tfont);
begin
  oldf:=selectobject(h,f.handle);
  settextcolor(h,f.color);
  drw.getbrocha(h,0,f.color,c);
  rectangle(h,rto.left,rto.top,rto.right,rto.bottom);
  drw.releasebrocha;
end;

var mm,ww:double;
begin
  chartpanel.visible:=true;
  if not tometa then begin
    h:=ChartPanel.canvas.handle;
    rto:=rect(1,1,ChartPanel.width,ChartPanel.height);
    SelObjects(ChartPanel.color,drw.font4);
    Dochart(h,rto);
  end
  else begin
    mm:=(r.right-r.left)/width;

```

```

ww:=(r.bottom-r.top)/height;
rto.left:=trunc(r.left+ChartPanel.left*mm);
rto.top:=trunc(r.top+ChartPanel.top*ww);
rto.right:=trunc(r.left+(ChartPanel.left+ChartPanel.width)*mm);
rto.bottom:=trunc(r.top+(ChartPanel.top+ChartPanel.height)*ww);
DoChart(h,rto);
end;
end;

Procedure ChartLegend(h:hdc;r:trect);
var i:integer;
wy,wb,wm:integer;
a:array[0..255] of char;
begin
wy:=(r.bottom-r.top) div (drw.co.nvalues+1);
wb:=trunc(0.60*wy);
wm:=trunc(0.20*wy);
if abs(font.size) >wy then font.size:=wy;
SetBkMode(h,1);
setTextAlign(h,TA_CENTER+TA_TOP);
strcpy(a,drw.co.title);
textout(h,(r.left+r.right) div 2,r.top+2,a,length(drw.co.title));
setTextAlign(h,TA_LEFT+TA_BOTTOM);
for i:=1 to drw.co.nvalues do begin
Bar(h,r.left+wb,r.top+wy*i+wm,
r.left+wb+wb,r.top+wy*i+wm+wb,i);
strcpy(a,drw.co.LegTexts[i]);
textout(h,r.left+3*wb,r.top+wy*i+wm+wb,a,length(drw.co.LegTexts[i]));
end;
end;

Procedure BulletLegend(h:hdc;rl:trect);
var i,j,k,k0,k2:integer;
wy,wm,wb:integer;
a:array[0..255] of char;
q:real;
ax:string;
pc:pchar;
oldf:hfont;
h1,h2,h3:integer;
f:tfont;
fh1:integer;
begin
case drw.bo.applyAs of
0,2:begin
k0:=trunc(drw.bo.size);
wb:=k0 div drw.bo.legendpoints;
q:=drw.maxval0 / drw.bo.legendpoints;
wy:=(rl.bottom-rl.top) div (drw.bo.legendpoints+1);
wm:=trunc(0.20*wy);
if abs(font.size) >wy then font.size:=wy;
SetBkMode(h,1);
setTextAlign(h,TA_CENTER+TA_TOP);
strcpy(a,drw.co.LegTexts[drw.bo.BulletSource]);
textout(h,(rl.left+rl.right) div
2,rl.top+2,a,length(drw.co.LegTexts[drw.bo.BulletSource]));
setTextAlign(h,TA_LEFT+TA_BOTTOM);
for i:=1 to drw.bo.legendpoints do begin
j:=drw.bo.legendpoints-i+1;
drw.fontsymbol.size:=k*2;
k2:=k ;
pc:=@drw.bo.thesymbol;
settextcolor(h,drw.bo.linecolor);
setbkmode(h,1);
oldf:=selectobject(h,drw.fontsymbol.handle);
fh1:=abs(drw.fontsymbol.height) div 2 ;
setTextAlign(h,TA_LEFT+TA_TOP);
textout(h,rl.left+wm,rl.top+wy*i+wm-fh1,pc,1);
selectobject(h,oldf);

```

```

    str(q*j:3:drw.mo.decimals,ax);
    strcpy(a,ax);
    textout(h,rl.left+2*k0+wm,rl.top+wy*i+wm,a,length(ax));
    if drw.bo.applyAs=0 then begin
        movetoex(h,rl.left+wm+k0 ,rl.top + wy * i ,nil);
        lineto(h,rl.left+2*k0+wm,rl.top + wy * i);
    end;
end;
end;
end;
1:begin
    wy:=(rl.bottom-rl.top) div (drw.bo.legendpoints+1);
    wb:=trunc(0.80*wy);
    wm:=trunc(0.10*wy);

    k:=100 div drw.bo.legendpoints;
    q:=drw.maxval0 / drw.bo.legendpoints;

    if abs(font.size) >wy then font.size:=wy;
    SetBkMode(h,1);
    setTextAlign(h,TA_CENTER+TA_TOP);
    ax:=drw.co.LegTexts[drw.bo.BulletSource];
    strcpy(a,ax);
    textout(h,(rl.left+rl.right) div 2,rl.top+2,a,length(ax));
    setTextAlign(h,TA_LEFT+TA_BOTTOM);

    for i:=1 to drw.bo.legendpoints do begin
        drw.getbrocha(h,0,drw.bo.shadows[k*i],drw.bo.shadows[k*i]);
        ellipse(h,rl.left+wb,rl.top+wy*i+wm,rl.left+wb+wb,rl.top+wy*i+wm+wb);
        drw.releasebrocha;
        str(q*i:3:drw.mo.decimals,ax); strcpy(a,ax);
        textout(h,rl.left+3*wb,rl.top+wy*i+wm+wb,a,length(ax));
    end;

end;
end;
5:begin
    rectangle(h,rl.left,rl.top,rl.right,rl.bottom);
    h1:=rl.top+font.size*3;
    h2:=rl.top+rl.bottom -h1;
    h2:=h2 div drw.bo.custom.nranges;

    setTextAlign(h,TA_CENTER+TA_TOP);
    strcpy(a,drw.co.LegTexts[drw.bo.BulletSource]);
    textout(h,(rl.left+rl.right) div
2,rl.top+2,a,length(drw.co.LegTexts[drw.bo.BulletSource]));
    setTextAlign(h,TA_LEFT+TA_TOP);

    for i:=1 to drw.bo.custom.nranges do begin
        strcpy(a,drw.bo.custom.ranges[i].text);
        textout(h,44,h1+(i-1)*h2 + (drw.bo.custom.ranges[i].size div
2),a,length(drw.bo.custom.ranges[i].text));
    end;
    settextrcolor(h,drw.bo.linecolor);
    f:=tfont.create;
    f.name:=drw.bo.custom.fontname;
    for i:=1 to drw.bo.custom.nranges do begin
        f.size:=drw.bo.custom.ranges[i].size;
        oldf:=selectobject(h,f.handle);
        pc:=@drw.bo.custom.ranges[i].symbol;
        textout(h,16,h1+(i-1)*h2 ,pc,1);
        selectobject(h,oldf);
    end;
    f.free;
end;
end;
end;
end;

```

```

Procedure TGPS_ARM_CORTEX_A15_MPCORE.CalcChartRatios;
var i,j:integer;
    mx,mx0,mx1:real;

```

```

begin
mx:=0;mx0:=0;mx1:=0;
for i:=1 to npolypoints do if sel[i] then begin
if np[i].values[drw.bo.bulletsource]>mx0 then
mx0:=np[i].values[drw.bo.bulletsource];
if np[i].values[drw.so.source]>mx1 then mx1:=np[i].values[drw.so.source];
for j:=1 to drw.co.nvalues do
if np[i].values[j]>mx then mx:=np[i].values[j];
end;
drw.maxval0:=mx0;
drw.maxval1:=mx1;
drw.maxvals:=mx;
end;

Procedure TGPS_ARM_CORTEX_A15_MPCORE.SetHDCFrame(h:hdc;r:trect);
begin
viewext:=MapLimits;
drw._view:=fzoombox;
drw._back:=r;
adaptlegrect(r,viewpct,drw._frame);
if drw.mo.aspect=1 then begin
if drw._view.right-drw._view.left<drw._view.bottom-drw._view.top
then begin drw._view.right:=drw._view.left+drw._view.bottom-drw._view.top;end
else begin drw._view.bottom:=drw._view.top+drw._view.right-drw._view.left;end;
end;
if (drw._view.right-drw._view.left) >0 then
drw.Rx:=(drw._frame.right-drw._frame.left)/(drw._view.right-drw._view.left)
else drw.Rx:=1;
if (drw._view.bottom-drw._view.top)>0 then
drw.Ry:=(drw._frame.bottom-drw._frame.top)/(drw._view.bottom-drw._view.top)
else drw.Ry:=1;
if drw.rx=0 then drw.rx:=1;
if drw.ry=0 then drw.ry:=1;
end;

Procedure TGPS_ARM_CORTEX_A15_MPCORE.FillNames(lb:tlistbox);
var i,j,k:integer;
begin
lb.clear;
for i:=1 to npolypoints do lb.items.add(inttostr(i)+'/'+np[i].nombre);
end;

Procedure TGPS_ARM_CORTEX_A15_MPCORE.FillArranges(lb:tlistbox);
begin
if not ready then exit;
arr.alltolistbox(lb);
end;

procedure TGPS_ARM_CORTEX_A15_MPCORE.ArrSet(father,son:word;ok:boolean);
begin
if not ready then exit;
arr.Poner(father,son,ok);
end;

Procedure TGPS_ARM_CORTEX_A15_MPCORE.ArrFromListBox(lb:tlistbox;father:word);
begin
if not ready then exit;
arr.FromListBox(lb,father);
ArrCalcBoxes;
end;

Procedure TGPS_ARM_CORTEX_A15_MPCORE.ArrToListBox(lb:tlistbox;father:word);
begin
if not ready then exit;
arr.ToListBox(lb,father);
end;

Procedure TGPS_ARM_CORTEX_A15_MPCORE.ArrSetName(name:string;father:word);

```

```

begin
if not ready then exit;
if father>arr.n then exit;
arr.titulos[father]^:=name;
end;

Procedure TGPS_ARM_CORTEX_A15_MPCORE.ArrSetNumber(n:word);
begin
arr.setnumber(n);
end;

Function TGPS_ARM_CORTEX_A15_MPCORE.ArrGetNumber:word;
begin
ArrGetNumber:=arr.n;
end;

Procedure TGPS_ARM_CORTEX_A15_MPCORE.SelArrGroup(father:integer;ok:boolean);
var i:integer;
begin
if father>0 then
if father<=arr.n then
for i:=1 to arr.t do if arr.Matr.Bit[father,i] then sel[i]:=ok;
end;

Function TGPS_ARM_CORTEX_A15_MPCORE.GetElSelected(i:word):boolean;
begin
GetElSelected:=false;
if i>0 then if i<=npolypoints then GetElSelected:=Sel[i];
end;

Procedure TGPS_ARM_CORTEX_A15_MPCORE.SetElSelected(i:word;ok:boolean);
begin
if i>0 then if i <=npolypoints then Sel[i]:=ok;
end;

Procedure TGPS_ARM_CORTEX_A15_MPCORE.SetAllSelections(ok:boolean);
var i:integer;
begin
for i:=1 to elementcount do selected[i]:= ok;
end;

Procedure TGPS_ARM_CORTEX_A15_MPCORE.SetBitMap(fname:string);
begin
bmpfname:=fname;
bitmap.free;
bitmap:=nil;
if fileexists(fname) then begin
bitmap:=tbitmap.create;
bitmap.loadfromfile(fname);
drw.mo.mapimage:=2;
end;
end;

Procedure TGPS_ARM_CORTEX_A15_MPCORE.DrawBitMap(h:hdc);
var ew,ow:longint;
w:trect;
vex,vey,vox,voy:integer;
rwb:record x,y:real;end;
wb:trect;
bm:tbitmap;
i,j:integer;
wx,wy:double;
wf:trect;

begin
if bitmap=nil then exit;
if drw.mo.mapimage=0 then exit;
if drw.mo.mapimage=2 then begin

```

```

wx:=Bitmap.width/(viewext.right-viewext.left);
wy:=Bitmap.height/(viewext.bottom-viewext.top);

w:=drw._view;
if w.left<viewext.left then w.left:=viewext.left;
if w.top<viewext.top then w.top:=viewext.top;
if w.right>viewext.right then w.right:=viewext.right;
if w.bottom>viewext.bottom then w.bottom:=viewext.bottom;

wb.left:=Trunc((w.left-viewext.left)*wx);
wb.right:=Trunc((w.right-viewext.left)*wx);
wb.top:=Trunc((w.top-viewext.top)*wy);
wb.bottom:=Trunc((w.bottom-viewext.top)*wy);

drw.ConvertaRect(w,wf);

if tometa then begin
  bm:=tbitmap.create;
  bm.assign(bitmap);
  StretchBlt(bm.canvas.handle,0,0,wb.right-wb.left,wb.bottom-wb.top,
  bitmap.canvas.handle,wb.left,wb.top,wb.right-wb.left,wb.bottom-
wb.top,SRCCOPY);

  bm.width:=wb.right-wb.left;bm.height:=wb.bottom-wb.top;
  StretchBlt(h,wf.left,wf.top,wf.right-wf.left,wf.bottom-wf.top,
  bm.canvas.handle,0,0,bm.width,bm.height,SRCCOPY);
  bm.destroy;
end
else
with wf do
  StretchBlt(h,left,top,right-left,bottom-top,
  bitmap.canvas.handle,wb.left,wb.top,wb.right-wb.left,wb.bottom-
wb.top,SRCCOPY);
end
else begin
with drw._frame do
  StretchBlt(h,left,top,right-left,bottom-top,
  bitmap.canvas.handle,0,0,bitmap.width,bitmap.height,SRCCOPY);
end;
end;

Procedure TGPS_ARM_CORTEX_A15_MPCORE.CleanHDC(h:hdc;col:tcolor;box:trect);
var b,oldb:hbrush;
p,oldp:hpen;
i,j,k,wy:integer;
gr:TGrArray;
w,hh,x,y:integer;
bm:tbitmap;
begin
CalcColGradients(drw.bko.mincolor,drw.bko.maxcolor,drw.bko.GradItems,gr);
if not tometa then box:=drw._back;
w:=(box.right-box.left) div drw.bko.GradItems;
hh:=(box.bottom-box.top) div drw.bko.GradItems;
b:=drw.GetSolidBrush(gr[drw.bko.GradItems]);
oldb:=selectobject(h,b);

fillrect(h,rect(box.left,box.top,box.right,box.bottom),b);
selectobject(h,oldb);deleteobject(b);

case drw.bko.effect of
1:for i:=1 to drw.bko.GradItems do begin
b:=drw.GetSolidBrush(gr[i]);
selectobject(h,b);
fillrect(h,rect(box.left,box.top+(i-1)*hh,box.right,box.top+i*hh),b);
selectobject(h,oldb);deleteobject(b);
end;
2:for i:=1 to drw.bko.GradItems do begin

```

```

b:=drw.GetSolidBrush(gr[i]);
selectobject(h,b);
fillrect(h,rect(box.left+(i-1)*w,box.top,box.left+i*w,box.bottom),b);
selectobject(h,oldb);deleteobject(b);
end;
3:begin
x:=(box.right-box.left) div 2;
y:=(box.bottom-box.top) div 2;
w:=x div 20;hh:=y div 20;
for i:=drw.bko.GradItems downto 1 do begin
b:=drw.GetSolidBrush(gr[i]);
selectobject(h,b);
p:=createpen(0,0,gr[i]);
oldp:=selectobject(h,p);
ellipse(h,box.left+x-w*i,box.top+y-hh*i,box.left+x+w*i,box.top+y+hh*i);
selectobject(h,oldp);deleteobject(p);
selectobject(h,oldb);deleteobject(b);
end;
end;
4:begin
x:=(box.right-box.left) div 2;
y:=(box.bottom-box.top) div 2;
w:=x div drw.bko.GradItems;hh:=y div drw.bko.GradItems;
for i:=drw.bko.GradItems downto 1 do begin
b:=drw.GetSolidBrush(gr[i]);
selectobject(h,b);
fillrect(h,rect(box.left+x-w*i,box.top+y-hh*i,box.left+x+w*i,box.top+y+hh*i),b);
selectobject(h,oldb);deleteobject(b);
end;
end;
5:begin
if not BackBitmap.empty then begin
x:=(box.right-box.left) div BackBitmap.width;
y:=(box.bottom-box.top) div BackBitmap.height;
for i:=0 to x do for j:=0 to y do
BitBlt(h,box.left+i*BackBitmap.width,box.top+j*BackBitmap.height,BackBitmap.widht
h,BackBitmap.height,BackBitmap.canvas.handle,0,0 ,SRCCOPY);
end;
end;
end;
end;

Procedure  TGPS_ARM_CORTEX_A15_MPCORE.Meridians(h:hdc;box:trect);
var i,j,w,hh,x,y:integer;
p,oldp:hpen;
pix:tcolor;
bm:tbitmap;
begin
if drw.bko.lines then begin
if not tometa then box:=drw._back;
w:=drw.bko.pctw;w:=trunc( w* (box.right-box.left)/ 100.0);
hh:=drw.bko.pctv;hh:=trunc( hh * (box.bottom-box.top)/ 100.0);
x:=(box.right-box.left) div w; y:= (box.bottom -box.top) div hh;
p:=createpen(0,0,drw.bko.linecolor);
oldp:=selectobject(h,p);
for i:=1 to x do begin
movetoex(h,box.left+i*w,box.top,nil);
lineto(h,box.left+i*w,box.bottom);
end;
for i:=1 to y do begin
movetoex(h,box.left,box.top+i*hh,nil);
lineto(h,box.right,box.top+i*hh);
end;
selectobject(h,oldp);
deleteobject(p);
end;
if drw.bko.logoon then begin
i:=LogoBitmap.width div 2;
j:=LogoBitmap.height div 2;

```

```

box.left:=box.left+1;box.right:=box.right-
1;box.top:=box.top+1;box.bottom:=box.bottom-1;
case drw.bko.Logopos of
  1:begin x:=box.left;y:=box.top;end;
  2:begin x:=(box.left+box.right) div 2 - i;y:=box.top;end;
  3:begin x:=box.right-LogoBitmap.width;y:=box.top;end;
  4:begin x:=box.left;y:=(box.top+box.bottom) div 2 -j;end;
  5:begin x:=(box.left+box.right) div 2 - i;y:=(box.top+box.bottom) div 2 -
j;end;
  6:begin x:=box.right-LogoBitmap.width;y:=(box.top+box.bottom) div 2 -j;end;
  7:begin x:=box.left;y:=box.bottom-LogoBitmap.height;end;
  8:begin x:=(box.left+box.right) div 2 - i;y:=box.bottom-LogoBitmap.height;end;
  9:begin x:=box.right-LogoBitmap.width;y:=box.bottom-LogoBitmap.height;end;
  else begin x:=box.left;y:=box.top;end;
end;
end;
end;

Procedure TGPS_ARM_CORTEX_A15_MPCORE.SetColors(t:byte;f,b:tcolor;ix:word);
begin
np[ix].setcolors(t,f,b);
end;

Procedure TGPS_ARM_CORTEX_A15_MPCORE.SetPBGColor(k:integer;c:tcolor);
var i:integer;
begin
for i:=1 to np[k].npolis do np[k].p[i].bkcol:=c;
for i:=1 to np[k].nmarcos do np[k].m[i].bkcol:=c;
end;

function TGPS_ARM_CORTEX_A15_MPCORE.GetPBGColor(k:integer):tcolor;
begin
if np[k].npolis>0 then
GetPBGColor:=np[k].p[1].bkcol
else if np[k].nmarcos>0 then
GetPBGColor:=np[k].m[1].bkcol
else GetPBGColor:=clwhite;
end;

Procedure TGPS_ARM_CORTEX_A15_MPCORE.SetPFgColor(k:integer;c:tcolor);
var i:integer;
begin
for i:=1 to np[k].npolis do np[k].p[i].frcol:=c;
for i:=1 to np[k].nmarcos do np[k].m[i].frcol:=c;
end;

function TGPS_ARM_CORTEX_A15_MPCORE.GetPFgColor(k:integer):tcolor;
begin
if np[k].npolis>0 then
GetPFgColor:=np[k].p[1].FRcol
else if np[k].nmarcos>0 then
GetPFgColor:=np[k].m[1].FRcol
else GetPFgColor:=clBLACK;
end;

Procedure TGPS_ARM_CORTEX_A15_MPCORE.SetPPattern(k:integer;c:byte);
var i:integer;
begin
for i:=1 to np[k].npolis do np[k].p[i].trama:=c;
for i:=1 to np[k].nmarcos do np[k].m[i].trama:=c;
end;

function TGPS_ARM_CORTEX_A15_MPCORE.GetPPattern(k:integer):byte;
begin
if np[k].npolis>0 then
GetPPattern:=np[k].p[1].trama
else if np[k].nmarcos>0 then
GetPPattern:=np[k].m[1].trama
else GetPPattern:=0;

```

```

end;

Procedure TGPS_ARM_CORTEX_A15_MPCORE.SetChartLegend(k:word;s:string);
begin
if k>16 then exit;
drw.co.LegTexts[k]:=s;
end;

function  TGPS_ARM_CORTEX_A15_MPCORE.GetChartLegend(k:word):string;
begin
if k>16 then exit;
GetChartLegend:=drw.co.LegTexts[k];
end;

Procedure TGPS_ARM_CORTEX_A15_MPCORE.SetShadows(a,b:tcolor);
begin
drw.calcshadows(a,b);
end;

Procedure TGPS_ARM_CORTEX_A15_MPCORE.Applyshadows;
var i,j:integer;
aux:double;
c:integer;
pcts:array[0..250] of record
    f,s:double;
    v:double;
end;

Procedure CalcFreqs;
var i,j,k:integer;
d,m:double;
s:double;
nump:integer;
begin
d:=drw.maxvall / 250;
fillchar(pcts,sizeof(pcts),0);
nump:=0;
for i:=1 to 250 do pcts[i].v:=i*drw.maxvall/250;
for i:=1 to npolypoints do if sel[i] then begin
k:=trunc(np[i].values[drw.so.source] *250/drw.maxvall);
if drw.sot.calc=1 then
pcts[k].f:=pcts[k].f+1
else pcts[k].f:=pcts[k].f+np[i].values[drw.so.source];
inc(nump);
end;
s:=0;
for i:=1 to 250 do begin
s:=s+pcts[i].f;
pcts[i].s:=s;
end;
if drw.sot.calc=1 then m:=nump / drw.so.legendpoints
else m:=s / drw.so.legendpoints;
for i:=1 to drw.so.legendpoints do
for j:=1 to 250 do
if pcts[j].s>= m*i then begin
drw.brkshadow[i]:=pcts[j].v;
break;
end;
drw.brkshadow[drw.so.legendpoints]:=drw.maxvall;
for i:=1 to npolypoints do if sel[i] then begin
m:=np[i].values[drw.so.source];
for j:=1 to drw.so.legendpoints do
if m<=drw.brkshadow[j] then begin
np[i].SetColors(0,0,drw.sot.colsf[j]);
break;
end;
begin
calcchartrratios;
drw.brkshadow[drw.so.legendpoints]:=drw.maxvall;
case drw.sot.kind of

```

```

0:begin
drw.calcsadows(drw.so.mincolor,drw.so.maxcolor);
for i:=1 to drw.so.legendpoints do
drw.brkshadow[i]:=(drw.maxvall/drw.so.legendpoints)*i;
drw.brkshadow[drw.so.legendpoints]:=drw.maxvall;
for i:=1 to npolypoints do if sel[i] then begin
c:=trunc(np[i].values[drw.so.source]*100/drw.maxvall);
if c<1 then c:=1;if c>100 then c:=100;
np[i].SetColor(0,0,drw.shadow[c]);
end;
end;
1:begin
if drw.sot.calc>0 then calcfrecs
else begin
for i:=1 to drw.so.legendpoints do
drw.brkshadow[i]:=(drw.maxvall/drw.so.legendpoints)*i;
drw.brkshadow[drw.so.legendpoints]:=drw.maxvall;
for i:=1 to npolypoints do if sel[i] then begin
c:=trunc(np[i].values[drw.so.source]*drw.so.legendpoints/drw.maxvall);
if c<1 then c:=1;if c>drw.so.legendpoints then c:=drw.so.legendpoints;
np[i].SetColor(0,0,drw.sot.colsf[c]);
end;
end;
end;
2:begin
for i:=1 to npolypoints do if sel[i] then begin
aux:=np[i].values[drw.so.source];
for j:=1 to drw.so.legendpoints do
if ((aux>=drw.sot.vals[j].min) and (aux<=drw.sot.vals[j].max)) then begin
np[i].SetColor(0,0,drw.sot.cols[j]);
break;
end;
end;

Procedure TGPS_ARM_CORTEX_A15_MPCORE.RectToPolyLine(r2:trect;var cuadro:array of
tpoint);
begin
cuadro[0].x:=r2.left;cuadro[0].y:=r2.top;
cuadro[1].x:=r2.right;cuadro[1].y:=r2.top;
cuadro[2].x:=r2.right;cuadro[2].y:=r2.bottom;
cuadro[3].x:=r2.left;cuadro[3].y:=r2.bottom;
cuadro[4].x:=r2.left;cuadro[4].y:=r2.top;
end;

procedure TGPS_ARM_CORTEX_A15_MPCORE.adaptlegrect(r,r1:trect;var r2:trect);
begin
r2.left:=trunc((r1.left/100)*(r.right-r.left)+r.left);
r2.top:=trunc((r1.top/100)*(r.bottom-r.top)+r.top);
r2.right:=trunc((r1.right/100)*(r.right-r.left)+r.left);
r2.bottom:=trunc((r1.bottom/100)*(r.bottom-r.top)+r.top);
end;

Procedure TGPS_ARM_CORTEX_A15_MPCORE.ClipWMF;
var
hdcmeta:hdc;
hmeta:hmetafile;
r:trect;
twmf:tmetafile;
begin
tometa:=true;
hdcmeta:=createmetafile(nil);
r:=rect(0,0,width,height);
SetWindowExtEx(hdcmeta,width,height,nil);
SetWindowOrgEx(hdcmeta,0,0,nil);
SetViewPortOrgEx(hdcmeta,0,0,nil);
SetViewPortExtEx(hdcmeta,width,height,nil);
Draw(hdcmeta,r);
hmeta:=closemetafile(hdcmeta);
twmf:=tmetafile.create;

```

```

twmf.width:=width;
twmf.height:=height;
with TMetafileCanvas.Create(twmf, 0) do
try
PlayMetaFile(handle,hmeta);
finally
free;
end;
twmf.enhanced:=false;
clipboard.assign(twmf);
twmf.free;
tometa:=false;
end;
Procedure TGPS_ARM_CORTEX_A15_MPCORE.WMF_clip2file(wnd:HWND;fname:string);
var
hGMem:longint;
lpmfp:^tMETAFILEPICT ;
f:file;
a:array[0..250] of char;
begin
OpenClipboard(Wnd);
hGMem:= GetClipboardData(CF_METAFILEPICT) ;
if (hGMem<>0) then begin
lpmfp:= GlobalLock(hGMem);
strcpy(a,fname);
copymetafile(lpmfp^.hmf,a);
GlobalUnlock(hGMem);
end;
CloseClipboard;
end;

procedure TGPS_ARM_CORTEX_A15_MPCORE.ClpWMF2Text(fname:string;textfname:string);
var tm:tmetaheader;
f:file;
ft:textfile;
rs:longint;
cf:word;
i,r,j:word;
s:longint;
tlf:tlogfont;
n,np:word;
ltp:^SPltSmallPoint;
begin
new(ltp);
assignfile(f,fname);
reset(f,1);
blockread(f,tm,sizeof(tm));
n:=0;
while not eof(f) do begin
blockread(f,rs,4);
blockread(f,cf,2);
if cf=$0324 then inc(n);
for j:=1 to rs-3 do blockread(f,cf,2);
end;
closefile(f);
if n=0 then exit;

assignfile(f,fname);
reset(f,1);

assignfile(ft,textfname);
rewrite(ft);

blockread(f,tm,sizeof(tm));
I:=1;

while not eof(f) do begin

```

```

blockread(f,rs,4);
blockread(f,cf,2);
if ((cf=$0324) and (i<maxelnum)) then begin
blockread(f,np,2);
blockread(f,ltp^,np*sizeof(TSmallPoint));
writeln(ft,':Element '+inttostr(i));
writeln(ft,'+P1');
for j:=1 to np do
writeln(ft,inttostr(ltp^[j].x)+' , '+inttostr(ltp^[j].y));
writeln(ft);
inc(i);
end
else begin
for j:=1 to rs-3 do if not eof(f) then blockread(f,cf,2);
end;
end;
closefile(f);
closefile(ft);
dispose(ltp);
end;

```

```

Function TGPS_ARM_CORTEX_A15_MPCORE.ClpWMFRect(fname:string):trect;
var tm:tmetaheader;
f:file;
rs:longint;
cf:word;
i,r,j:word;
s:longint;
tlf:tlogfont;
n,np:word;
ltp:^SPltSmallPoint;
re:trect;
begin
new(ltp);
assignfile(f,fname);
reset(f,1);
blockread(f,tm,sizeof(tm));
n:=0;
re.left:=32000;re.top:=32000;re.bottom:=-32000;re.right:=-32000;

while not eof(f) do begin
blockread(f,rs,4);
blockread(f,cf,2);
if cf=$0324 then inc(n);
for j:=1 to rs-3 do if not eof(f) then blockread(f,cf,2);
end;
closefile(f);
if n=0 then exit;

assignfile(f,fname);
reset(f,1);
blockread(f,tm,sizeof(tm));
I:=1;

while not eof(f) do begin
blockread(f,rs,4);
blockread(f,cf,2);
if ((cf=$0324) and (i<maxelnum)) then begin
blockread(f,np,2);
blockread(f,ltp^,np*sizeof(TSmallPoint));
for j:=1 to np do begin
if ltp^[j].x<re.left then re.left:=ltp^[j].x;
if ltp^[j].y<re.top then re.top:=ltp^[j].y;
if ltp^[j].x>re.right then re.right:=ltp^[j].x;
if ltp^[j].y>re.bottom then re.bottom:=ltp^[j].y;
end;
inc(i);
end

```

```

else begin
    for j:=1 to rs-3 do if not eof(f) then blockread(f,cf,2);
end;
end;
closefile(f);
dispose(ltp);
ClpWMFRect:=re;
end;

Procedure
TGPS_ARM_CORTEX_A15_MPCORE.ScaleClpWMF2Text(fname,textfname:string;iso:boolean);
var tm:tmetaheader;
f:file;
ft:textfile;
rs:longint;
cf:word;
i,r,j:word;
s:longint;
tlf:tlogfont;
n,np:word;
ltp:^SPltSmallPoint;
e,re:trect;
sx,sy:real;
lt:integer;
puntos:integer;
a:array[0..255] of char;
ax:string;
begin
e:=rect(0,0,10000,10000);
re:=ClpWMFRect(fname);

if (re.right-re.left)>(re.bottom-re.top) then begin
sx:=(e.right-e.left)/(re.right-re.left);
sy:=(e.bottom-e.top)/(re.right-re.left);
end
else begin
sx:=(e.right-e.left)/(re.bottom-re.top);
sy:=(e.bottom-e.top)/(re.bottom-re.top);
end;

new(ltp);
assignfile(f,fname);
reset(f,1);
blockread(f,tm,sizeof(tm));
n:=0;
while not eof(f) do begin
blockread(f,rs,4);
blockread(f,cf,2);
if cf=$0324 then inc(n);
if cf=$0325 then inc(n);
for j:=1 to rs-3 do if not eof(f) then blockread(f,cf,2);
end;
closefile(f);
if n=0 then exit;

assignfile(f,fname);
reset(f,1);

assignfile(ft,textfname);
rewrite(ft);

blockread(f,tm,sizeof(tm));
I:=1;
puntos:=0;
while not eof(f) do begin
blockread(f,rs,4);
blockread(f,cf,2);
if ((cf=$0324) and (i<maxelnum)) then begin

```

```

blockread(f,np,2);
blockread(f,ltp^,np*sizeof(TSmallPoint));
writeln(ft,':Element '+inttostr(i));
writeln(ft,'+P'+inttostr(i)+' #2');
for j:=1 to np do begin
  ltp^[j].x:=trunc((ltp^[j].x - re.left)*sx + e.left);
  ltp^[j].y:=trunc((ltp^[j].y - re.top)*sy + e.top);
  writeln(ft,inttostr(ltp^[j].x)+ ' , '+inttostr(ltp^[j].y));
end;
writeln(ft);
inc(i);
puntos:=1;
end
else if ((cf=$0325) and (i<maxelnum)) then begin
blockread(f,np,2);
blockread(f,ltp^,np*sizeof(TSmallPoint));
writeln(ft,':Element '+inttostr(i));
writeln(ft,'_L'+inttostr(i)+' #2');
for j:=1 to np do begin
  ltp^[j].x:=trunc((ltp^[j].x - re.left)*sx + e.left);
  ltp^[j].y:=trunc((ltp^[j].y - re.top)*sy + e.top);
  writeln(ft,inttostr(ltp^[j].x)+ ' , '+inttostr(ltp^[j].y));
end;
writeln(ft);
inc(i);
end

else if ((cf=$0521) and (i<100)) then begin
blockread(f,lt,2);
blockread(f,a,lt); a[lt+1]:=#0; ax:=strupas(a);
writeln(ft,'/T'+inttostr(puntos)+' #3');
blockread(f,lt,2);
write(ft,inttostr(lt)+' , ');
blockread(f,lt,2);
writeln(ft,inttostr(lt)+' , '+ ax);
inc(puntos);
end
else begin
  for j:=1 to rs-3 do if not eof(f) then blockread(f,cf,2);
end;
end;
closefile(f);
closefile(ft);
dispose(ltp);
end;

Procedure TGPS_ARM_CORTEX_A15_MPCORE.FromWMF(fname:string);
var tmf:tmetafile;
a:array[0..255] of char;
ax,bx:string;
begin
GetWindowsDirectory(a,200);
ax:=strupas(a)+'\temp.clp';
bx:=strupas(a)+'\temp.txt';

tmf:=tmetafile.create;
tmf.loadfromfile(fname);
clipboard.assign(tmf);
WMF_clip2file(parent.handle,ax);
tmf.destroy;
ScaleClpWMF2Text(ax,bx,true);
FromTextFile(bx);
end;

Procedure TGPS_ARM_CORTEX_A15_MPCORE.FromClipWMF;
var a:array[0..255] of char;
ax,bx:string;
begin

```

```

if not clipboard.hasformat(CF_metafilepict) then exit;
GetWindowsDirectory(a,200);
ax:=strpas(a)+'\temp.clp';
bx:=strpas(a)+'\temp.txt';
WMF_clip2file(parent.handle,ax);
ScaleClpWMF2Text(ax,bx,true);
FromTextFile(bx);
end;

```

```

Procedure TGPS_ARM_CORTEX_A15_MPCORE.SaveWMF(fname:string);
var tmf:tmetafile;
begin
ClipWMF;
tmf:=tmetafile.create;
tmf.assign(clipboard);
tmf.savetofile(fname);
tmf.destroy;
end;

```

```

Procedure TGPS_ARM_CORTEX_A15_MPCORE.ClipBmp;
var
r:trect;
bm01:tbitmap;
begin
if not ready then exit;
bm01:=tbitmap.create;
try
r:=rect(0,0,width,height);
bm01.width:=(r.right-r.left)+1;
bm01.height:=(r.bottom-r.top)+1;
bm01.canvas.copyrect(r,canvas,r);
clipboard.assign(bm01);
finally
bm01.free;
end;
end;

```

```

Procedure TGPS_ARM_CORTEX_A15_MPCORE.SaveBmp(fname:string);
var bm:tbitmap;
r:trect;
begin
if not ready then exit;
bm:=tbitmap.create;
r:=rect(0,0,width,height);
bm.width:=(r.right-r.left)+1;
bm.height:=(r.bottom-r.top)+1;
bm.canvas.copyrect(r,canvas,r);
bm.SaveToFile(fname);
bm.free;
end;

```

```

Procedure
TGPS_ARM_CORTEX_A15_MPCORE.SnapShot(destinationcanvas:tcanvas;destrect:trect;map
rect:trect);
var r:trect;
begin
if not ready then exit;
drw.convertarect(maprect,r);
StretchBlt(destinationcanvas.handle,
destrect.left,
destrect.top,
destrect.right-destrect.left+1,
destrect.bottom-destrect.top+1,
canvas.handle,
r.left,
r.top,

```

```

    r.right-r.left+1,
    r.bottom-r.top+1,
    SRCCOPY);

end;

Procedure
TGPS_ARM_CORTEX_A15_MPCORE.SetChartValue (element,item:word;value:real);
begin
if element>npolypoints then exit;
if element<1 then exit;
if item>drw.co.nvalues then exit;
np[element].values[item]:=value;
end;

Function  TGPS_ARM_CORTEX_A15_MPCORE.GetChartValue (element,item:word):real;
begin
GetChartValue:=0;
if element>npolypoints then exit;
if element<1 then exit;
if item>drw.co.nvalues then exit;
GetChartValue:=np[element].values[item];
end;

Procedure  TGPS_ARM_CORTEX_A15_MPCORE.ValuesToGrid(grid:tstringgrid);
var i,j:integer;
ax:string;
begin
with grid do begin
colcount:=drw.co.nvalues+2;
rowcount:=npolypoints+1;
fixedcols:=0;
fixedrows:=0;
cells[0,0]:='';
cells[1,0]:='Bullet';
for i:=1 to drw.co.nvalues do cells[1+i,0]:=drw.co.LegTexts[i];
for i:=1 to npolypoints do cells[0,i]:=np[i].nombre;
for i:=1 to npolypoints do
cells[1,i]:=fcadd(np[i].values[0],3,drw.co.ndecimals);
for i:=1 to drw.co.nvalues do
for j:=1 to npolypoints do
cells[i+1,j]:=fcadd(np[j].values[i],3,drw.co.ndecimals);
end;
end;

Procedure  TGPS_ARM_CORTEX_A15_MPCORE.ValuesFromGrid(grid:tstringgrid);
var i,j:integer;
ax:string;
begin
with grid do begin
drw.co.nvalues:=colcount-2;
for i:=0 to drw.co.nvalues do drw.co.LegTexts[i]:=cells[1+i,0];
for i:=1 to rowcount-1 do np[i].nombre:=cells[0,i];
for i:=1 to rowcount-1 do np[i].values[0]:=fnum(cells[1,i]);
for i:=1 to drw.co.nvalues do
for j:=1 to rowcount-1 do
np[j].values[i]:=fnum(cells[i+1,j]);
end;
end;
CalcChartRatios;
end;

Procedure  TGPS_ARM_CORTEX_A15_MPCORE.ImportDataFromGrid(grid:tstringgrid);
var i,j,k,code:integer;
r:real;
ax:string;

```

```

begin
if grid.colcount>18 then grid.colcount:=18;
drw.co.nvalues:=grid.colcount-2;
for i:=1 to grid.rowcount-1 do begin
ax:=uppercase(grid.cells[0,i]);
for j:=1 to ElementCount do
if uppercase(ElementName[j])=ax then
for k:=1 to grid.colcount-1 do begin
val(grid.cells[k,i],r,code);
chartvalue[j,k-1]:=r;
end;
end;
for i:=1 to grid.colcount-1 do chartlegend[i-1]:=grid.cells[i,0];
end;

Function  TGPS_ARM_CORTEX_A15_MPCORE.GetPatternBits(i:word):string;
begin
if i>8 then exit;
if i=0 then GetPatternBits:='000000'
else GetPatternBits:=trama[i];
end;

Procedure TGPS_ARM_CORTEX_A15_MPCORE.SetPatternBits(i:word;s:string);
begin
if ((i<1) or (i>8)) then exit;
trama[i]:=copy(s,1,64);
end;

Procedure TGPS_ARM_CORTEX_A15_MPCORE.Bar(h:hdc;x1,y1,x2,y2:integer;pat:word);
begin
if pat>16 then exit;if pat<1 then exit;
drw.GetBrocha(h,drw.co.patt[pat],drw.co.linecolor,drw.co.colors[pat]);
rectangle(h,x1,y1,x2,y2);
drw.ReleaseBrocha;
end;

Function TGPS_ARM_CORTEX_A15_MPCORE.GetPolLayer(element,pol:word):byte;
begin
GetPolLayer:=1;
if element=0 then exit; if pol=0 then exit;
if (element>npolypoints) then exit;
if (pol>np[element].npolis) then exit;
GetPolLayer:=np[element].p[pol].layer;
end;

Procedure TGPS_ARM_CORTEX_A15_MPCORE.SetPolLayer(element,pol:word;l:byte);
begin
if element=0 then exit; if pol=0 then exit;
if (element>npolypoints) then exit;
if (pol>np[element].npolis) then exit;
np[element].p[pol].layer:=l;
end;

Function TGPS_ARM_CORTEX_A15_MPCORE.GetPolCount(element:word):byte;
begin
GetPolCount:=0;
if element=0 then getpolcount:=npolypoints;
if (element>npolypoints) then exit;
GetPolCount:=np[element].npolis;
end;

Function TGPS_ARM_CORTEX_A15_MPCORE.GetPolName(element,pol:word):string;
begin
GetPolName:='';
if element=0 then exit; if pol=0 then exit;
if (element>npolypoints) then exit;
if (pol>np[element].npolis) then exit;
GetPolName:=np[element].p[pol].nombre;
end;

```

```

Procedure TGPS_ARM_CORTEX_A15_MPCORE.SetPolName(element,pol:word;s:string);
begin
if element=0 then exit; if pol=0 then exit;
if (element>npolypoints) then exit;
if (pol>np[element].npolis) then exit;
np[element].p[pol].nombre:=s;
end;

```

```

Procedure TGPS_ARM_CORTEX_A15_MPCORE.ShowPolNames(b:boolean);
begin
drw.showpols:=b;
end;

```

```

Function TGPS_ARM_CORTEX_A15_MPCORE.GetDistance(p1,p2:tpoint):double;
begin
GetDistance:=distancia(p1.x,p1.y,p2.x,p2.y)*drw.mo.scale;
end;

```

```

Procedure TGPS_ARM_CORTEX_A15_MPCORE.Setnumbox(b:boolean);
begin
drw.numbox:=b;
end;
Function TGPS_ARM_CORTEX_A15_MPCORE.GetNumBox:boolean;
begin
GetNumBox:=drw.numbox;
end;

```

```

Function TGPS_ARM_CORTEX_A15_MPCORE.ElementbyName(s:string):integer;
var i,k:integer;
begin
k:=1;
for i:=1 to elementcount do
if pos(uppercase(s),uppercase(elementname[i]))>0 then begin
k:=i;
break;
end;
ElementbyName:=k;
end;

```

```

Function TGPS_ARM_CORTEX_A15_MPCORE.Nearest(p:tpoint):integer;
var i,j,k:integer;
mybox:trect;
begin
k:=0;
Nearest:=0;
for i:=npolypoints downto 1 do
if np[i].npolis>0 then begin
if ((p.x>np[i].caja.left) and (p.x<np[i].caja.right)
and (p.y>np[i].caja.top) and (p.y<np[i].caja.bottom))
then begin Nearest:=i;exit;end;
end;
Nearest:=k;
end;

```

```

Function TGPS_ARM_CORTEX_A15_MPCORE.NearestElement(p:tpoint):integer;
var i,j,k:integer;
mybox:trect;
begin
k:=0;
NearestElement:=0;
for i:=npolypoints downto 1 do if sel[i] then
if np[i].npolis>0 then begin
if ((p.x>np[i].caja.left) and (p.x<np[i].caja.right)
and (p.y>np[i].caja.top) and (p.y<np[i].caja.bottom))
then begin NearestElement:=i;exit;end;
end;
end;

```

```

NearestElement:=k;
end;

procedure muerte;
begin
showmessage('Fatal error reading '+fname);
halt;
end;
var i,j,k,im,ip,ipu:integer;
ax,bx,nom:string;
fin:boolean;
p:tpoint;
f:textfile;
nsp:^tnsp;
n:word;
buffer:^SPltpoint;
maxcaja:trect;
c:char;
thelayer:integer;
lm:longint;
begin
reset(f);
ax:='';
for i:=1 to npolypoints do begin
while ((ax[1]<>':') and (not eof(f))) do begin
readln(f,ax);
ax:=ltrim(ax);
end;
if eof(f) then muerte;
infopanel.caption:=inttostr(i)+' of '+ inttostr(npolypoints);
infopanel.repaint;
ax:=rltrim(derc(ax,':'));
np[i]:=TSPolyPol.create(nsp^[i].m,nsp^[i].p,ax);
np[i].npoints:=nsp^[i].pu;
im:=0;ip:=0; ipu:=0;
for j:=1 to nsp^[i].m + nsp^[i].p + nsp^[i].pu do begin
repeat
readln(f,ax);
ax:=ltrim(ax);
until ((ax[1] in ['_','+','/']) or eof(f));
if eof(f) then muerte;
c:=ax[1];
nom:=rltrim(izqc(derc(ax,c),'#'));
ax:=rltrim(derc(ax,'#'));
if ax='' then ax:='1';
thelayer:=num(ax);
if c='_' then begin
inc(im);
np[i].SetMVals(clwhite,clblack,im,k,buffer^);
np[i].m[im].layer:=thelayer;
np[i].m[im].drawer:=drw;
np[i].m[im].nombre:=nom;
end
else
if c='+' then begin
inc(ip);
np[i].SetPVals(clwhite,clblack,ip,k,buffer^);
np[i].p[ip].layer:=thelayer;
np[i].p[ip].drawer:=drw;
np[i].p[ip].nombre:=nom;
end;
end;
np[i].CalcCaja;
comprect(np[i].caja,maxcaja);
np[i].drawer:=drw;
end;
closefile(f);
dispose(nsp);
dispose(buffer);

```

```

for i:=1 to npolypoints do sel[i]:=true;
if arr<>nil then arr.destroy;
arr:=TSArrange.create(2,npolypoints,'Normal');
ready:=true;
Titulo:=fname;
Nbloqs:=npolypoints;
MapLimits:=MaxCaja;
end;

Function
TGPS_ARM_CORTEX_A15_MPCORE.AddPoint(elem:integer;c:tcitypoint):integer;
var a:tsapoint;
begin
AddPoint:=0;
if elem>npolypoints then exit; if elem<1 then exit;
if np[elem].npoints>15 then exit;
a:=tsapoint.create;
a.c:=c;
np[elem].npoints:=np[elem].npoints+1;
np[elem].points[np[elem].npoints]:=a;
AddPoint:=np[elem].npoints;
a.free;
end;

Procedure TGPS_ARM_CORTEX_A15_MPCORE.GetPoint(elem,ix:integer;var c:tcitypoint);
begin
if elem>npolypoints then exit; if elem<1 then exit;
if ix<1 then exit;if ix> np[elem].npoints then exit;
c:=np[elem].points[ix].c;
end;

Function TGPS_ARM_CORTEX_A15_MPCORE.RemovePoint(elem,ix:integer):boolean;
var i:integer;
begin
removepoint:=false;
if elem>npolypoints then exit; if elem<1 then exit;
if np[elem].npoints<1 then exit;
if ix<1 then exit;if ix> np[elem].npoints then exit;
for i:=ix to np[elem].npoints-1 do np[elem].points[i]:=np[elem].points[i+1];
np[elem].npoints:=np[elem].npoints-1;
Removepoint:=true;
end;

Function TGPS_ARM_CORTEX_A15_MPCORE.GetPointCount(elem:integer):integer;
begin
GetPointCount:=0;
if elem>npolypoints then exit; if elem<1 then exit;
if np[elem].npoints<1 then exit;
GetPointCount:=np[elem].npoints;
end;

Procedure TGPS_ARM_CORTEX_A15_MPCORE.ClearPoints(elem:integer);
begin
if elem>npolypoints then exit; if elem<1 then exit;
np[elem].npoints:=0;
end;

Procedure TGPS_ARM_CORTEX_A15_MPCORE.SetSymbol(f:tfont;c:char);
begin
drw.fontsymbol.assign(f);
drw.asymbol:=c;
end;

procedure TGPS_ARM_CORTEX_A15_MPCORE.time(sender:tobject);
begin
replaceglis;
gli.paint(canvas);
end;

```

```

Procedure TGPS_ARM_CORTEX_A15_MPCORE.MyMouseUp(Sender: TObject; Button:
TMouseButton; Shift: TShiftState; X, Y: Integer);
var p:tpoint;
i,k:integer;
pu:tpoint;
pr:trect;
begin
case drw.mo.mousemode of
8:begin
    p.x:=x;p.y:=y;
    p:=PointInMap(p);
    if button=mbright then begin

p.x:=(GPS_ARM_Cortex_A15_MPCorebox.right+GPS_ARM_Cortex_A15_MPCorebox.left) div
2;

p.y:=(GPS_ARM_Cortex_A15_MPCorebox.top+GPS_ARM_Cortex_A15_MPCorebox.bottom) div
2;

        end;
        k:=(zoombox.right-zoombox.left) div 2;
        i:=(zoombox.bottom-zoombox.top) div 2;
        fzoombox.left:=p.x-k; fzoombox.right:=p.x+k;
        fzoombox.top:=p.y-i; fzoombox.bottom:=p.y+i;
        repaint;
end;

10:if bb_drawing then begin
    bb_drawing:=false;
    DrawFocusRect(canvas.handle,rect(bb_anchorx,bb_anchory,bb_curx,bb_cury));
    p:=PointInMap(point(bb_anchorx,bb_anchory));
    if button=mbright then SelFromBox(GPS_ARM_Cortex_A15_MPCorebox,true)
    else
        if ((bb_curx-bb_anchorx>10) and (bb_cury-bb_anchory>10)) then
            begin
                pr.left:=p.x;pr.top:=p.y;
                p:=PointInMap(point(bb_curx,bb_cury));
                pr.right:=p.x;pr.bottom:=p.y;
                if ssCtrl in Shift then SelFromBox(pr,false)
                else SelFromBox(pr,true);
            end
            else begin
                k:=Nearest(p);
                if k>0 then begin
                    if ssCtrl in Shift then selected[k]:=true
                    else selected[k]:=not selected[k];
                end;
            end;
            invalidate;
        end;

1,9,12,15:if bb_drawing then begin
    bb_drawing:=false;
    DrawFocusRect(canvas.handle,rect(bb_anchorx,bb_anchory,bb_curx,bb_cury));

    if drw.mo.mousemode =9 then begin
        pr.left:=trunc((bb_anchorx-left)/(drw._back.right-drw._back.left)*100);
        pr.right:=trunc((bb_curx-left)/(drw._back.right-drw._back.left)*100);
        pr.top:=trunc((bb_anchory-top)/(drw._back.bottom-drw._back.top)*100);
        pr.bottom:=trunc((bb_cury-top)/(drw._back.bottom-drw._back.top)*100);
    end;

    case drw.mo.mousemode of
    9:
        if ((bb_curx-bb_anchorx>10) and (bb_cury-bb_anchory>10)) then
            begin
                viewpct:=pr;
                infopanel.caption:=inttostr(viewpct.left)+' '+
                inttostr(viewpct.top)+' - '+'

```

```

    inttostr(viewpct.right)+' '+
    inttostr(viewpct.bottom);

end
else viewpct:=rect(1,1,99,99);

1:
if ((bb_curx-bb_anchorx>10) and (bb_cury-bb_anchory>10)) then
begin
    p:=PointInMap(point(bb_anchorx,bb_anchory));
    fzoombox.left:=p.x;fzoombox.top:=p.y;
    p:=PointInMap(point(bb_curx,bb_cury));
    fzoombox.right:=p.x;fzoombox.bottom:=p.y;
end
else

begin
p.x:=x;p.y:=y;
p:=PointInMap(p);

k:=(zoombox.right-zoombox.left) div 2;
i:=(zoombox.bottom-zoombox.top) div 2;
fzoombox.left:=p.x-k; fzoombox.right:=p.x+k;
fzoombox.top:=p.y-i; fzoombox.bottom:=p.y+i;

end;

12:if Assigned(FOnMouseBox) then begin
    if ((bb_curx-bb_anchorx>10) and (bb_cury-bb_anchory>10)) then
    begin
        p:=PointInMap(point(bb_anchorx,bb_anchory));
        pr.left:=p.x;pr.top:=p.y;
        p:=PointInMap(point(bb_curx,bb_cury));
        pr.right:=p.x;pr.bottom:=p.y;
    end
    else pr:=GPS_ARM_Cortex_A15_MPCorebox;
    OnMouseBox(self,shift,pr);
end;
15:begin
    if ((bb_curx-bb_anchorx>20) and (bb_cury-bb_anchory>20)) then
    begin
        p:=PointInMap(point(bb_anchorx,bb_anchory));
        pr.left:=p.x;pr.top:=p.y;
        p:=PointInMap(point(bb_curx,bb_cury));
        pr.right:=p.x;pr.bottom:=p.y;
    end
    else pr:=GPS_ARM_Cortex_A15_MPCorebox;
    scaleto(pr);
end;
end;
invalidate;
end;
3:begin
    bb_drawing:=false;
    invalidate;
end;
4,11:if bb_drawing then begin
    bb_drawing:=false;
    SetRop2(canvas.handle,r2_not);
    movetoex(canvas.handle,drw.cvx(bb_anchorx),drw.cvy(bb_anchory),nil);
    lineto(canvas.handle,drw.cvx(bb_curx),drw.cvy(bb_cury));
    if drw.mo.mousemode=11 then if assigned(FOnMouseLine)
    then OnMouseLine(self,shift,rect(bb_anchorx,bb_anchory,bb_curx,bb_cury));
end;
5:if bb_drawing then begin
    bb_drawing:=false;
    SetRop2(canvas.handle,r2_not);
    p.x:=abs(bb_curx-bb_anchorx);

```

```

    p.y:=p.x;

    k:=abs(drw.cvx(bb_curx)-drw.cvx(bb_anchorx));
    pu.x:=drw.cvx(bb_anchorx);
    pu.y:=drw.cvy(bb_anchory);
    Arc(canvas.handle,pu.x-k,pu.y-k,
        pu.x+k,pu.y+k,
        pu.x,pu.y,pu.x,pu.y);
end;
16:   if not userline.drawing then begin
        if button=mbright then invalidate;
    end;
end;
end;

Procedure TGPS_ARM_CORTEX_A15_MPCORE.WMLButtonDown(var msg:TWMLButtonDown);
begin
if not ready then exit;
MyMouseDown(self,mbleft,KeysToShiftState(msg.keys),msg.xpos,msg.ypos);
If Assigned(FOnMouseDown) then
OnMouseDown(self,mbleft,KeysToShiftState(msg.keys),msg.xpos,msg.ypos);
end;

Procedure TGPS_ARM_CORTEX_A15_MPCORE.WMRButtonDown(var msg:TWMRButtonDown);
begin
if ready then begin
MyMouseDown(self,mbright,KeysToShiftState(msg.keys),msg.xpos,msg.ypos);
If Assigned(FOnMouseDown) then
OnMouseDown(self,mbright,KeysToShiftState(msg.keys),msg.xpos,msg.ypos)
else if drw.mo.mousemode=0 then parent.dispatch(msg);
end
else inherited;
end;

Procedure TGPS_ARM_CORTEX_A15_MPCORE.WMLButtonUp(var msg:TWMLButtonUp);
begin
if not ready then exit;
MyMouseUp(self,mbleft,KeysToShiftState(msg.keys),msg.xpos,msg.ypos);
if ready then If Assigned(fOnMouseUp) then
OnMouseUp(self,mbleft,KeysToShiftState(msg.keys),msg.xpos,msg.ypos);
inherited;
end;

Procedure TGPS_ARM_CORTEX_A15_MPCORE.WMRButtonUp(var msg:TWMRButtonUp);
begin
if ready then begin
MyMouseUp(self,mbright,KeysToShiftState(msg.keys),msg.xpos,msg.ypos);
if ready then If Assigned(fOnMouseUp) then
OnMouseUp(self,mbright,KeysToShiftState(msg.keys),msg.xpos,msg.ypos);
end;
end;

Procedure TGPS_ARM_CORTEX_A15_MPCORE.WMMouseMove(var msg:TWMMouseMove);
begin
if not ready then exit;
MyMouseMove(self,KeysToShiftState(msg.keys),msg.xpos,msg.ypos);
if ready then if Assigned(fOnMouseMove) then
OnMouseMove(self,KeysToShiftState(msg.keys),msg.xpos,msg.ypos);
end;

Procedure TGPS_ARM_CORTEX_A15_MPCORE.ToFront(elem:integer);
var auxp:TSPolyPol;
i:integer;
auxs:boolean;
begin
if ((elem<1) or (elem>npolypoints)) then exit;
auxp:=np[elem];
auxs:=selected[elem];
for i:=elem to npolypoints-1 do begin

```

```

np[i]:=np[i+1];
selected[i]:=selected[i+1];
end;
np[npolypoints]:=auxp;
selected[npolypoints]:=auxs;
end;

```

```

Procedure TGPS_ARM_CORTEX_A15_MPCORE.ToBack(elem:integer);
var auxp:TSPolyPol;
i:integer;
auxs:boolean;
begin
if ((elem<1) or (elem>npolypoints)) then exit;
auxp:=np[elem];
auxs:=selected[elem];
for i:=elem downto 2 do begin
np[i]:=np[i-1];
selected[i]:=selected[i-1];
end;
np[1]:=auxp;
selected[1]:=auxs;
end;

```

```

Procedure TGPS_ARM_CORTEX_A15_MPCORE.ReMap(var m:array of integer);
var sp:^TSPOLARRAY;
i:integer;
begin
new(sp);
for i:=1 to npolypoints do sp^[i]:=np[m[i]];
for i:=1 to npolypoints do np[i]:=sp^[i];
dispose(sp);
end;

```

```

Function TGPS_ARM_CORTEX_A15_MPCORE.IsInBox(ix:integer;r:trect):boolean;
var rr:trect;
begin
IntersectRect(rr,r,np[ix].caja);
IsInBox:=((rr.left<>0) or (rr.right<>0) or (rr.bottom<>0) or (rr.top<>0));
end;

```

```

Procedure TGPS_ARM_CORTEX_A15_MPCORE.SelFromBox(r:trect;cleanup:boolean);
var i:integer;
begin
if cleanup then begin
for i:=1 to elementcount do selected[i]:= IsInBox(i,r);
end
else
for i:=1 to elementcount do if IsInBox(i,r) then selected[i]:= true;
end;

```

```

Procedure TGPS_ARM_CORTEX_A15_MPCORE.ForwardLink(ix:integer);
var ax:string;
begin
if ix>0 then
if np[ix].link<>' ' then
if fileexists(np[ix].link) then begin
if nbacks<5 then inc(nbacks);
BackLink[nbacks]:=polfname;
ax:=np[ix].link;
FileName:=ax;
drw.mo.mousemode:=14;
end;
end;

```

```

Procedure TGPS_ARM_CORTEX_A15_MPCORE.BackwardLink;
var ax:string;
begin
if nbacks>0 then begin
if fileexists(BackLink[nbacks]) then begin

```

```

ax:=BackLink[nbacks];
FileName:=ax;
drw.mo.mousemode:=14;
end;
dec(nbacks);
end;
end;

Procedure TGPS_ARM_CORTEX_A15_MPCORE.SetBackgroundOptions(var op:tbkop);
begin
drw.bko:=op;
if drw.bko.tilefname<>' ' then if fileexists(drw.bko.tilefname)
then backbitmap.loadfromfile(drw.bko.tilefname);
if drw.bko.logofname<>' ' then if fileexists(drw.bko.logofname)
then logobitmap.loadfromfile(drw.bko.logofname);
end;

Procedure TGPS_ARM_CORTEX_A15_MPCORE.CalcCaja;
var i:integer;
begin;
prepect(MapLimits);
for i:=1 to npolypoints do compect(np[i].caja,MapLimits);
end;

Procedure TGPS_ARM_CORTEX_A15_MPCORE.Merge(fname:string;layeroffset:integer;
x,y:integer);
var aGPS_ARM_Cortex_A15_MPCore:TGPS_ARM_CORTEX_A15_MPCORE;
i,j,k:integer;
begin
if not fileexists(fname) then exit;
ready:=false;
aGPS_ARM_Cortex_A15_MPCore:=TGPS_ARM_CORTEX_A15_MPCORE.create(self);
aGPS_ARM_Cortex_A15_MPCore.filename:=fname;
for i:=1 to aGPS_ARM_Cortex_A15_MPCore.elementcount do
np[elementcount+i]:=aGPS_ARM_Cortex_A15_MPCore.np[i];
npolypoints:=npolypoints+aGPS_ARM_Cortex_A15_MPCore.elementcount;
aGPS_ARM_Cortex_A15_MPCore.npolypoints:=0;
aGPS_ARM_Cortex_A15_MPCore.free;
CalcCaja;
ready:=true;
zoombox:=MapLimits;
end;

Procedure TGPS_ARM_CORTEX_A15_MPCORE.ScaleTo(r:trect);
var sb,aux:trect;
kx,ky:double;
x0,y0,x1,y1:integer;
i,j:integer;
function newx(x:integer):integer;
begin
newx:=trunc(x1+(x-x0)*kx);
end;
function newy(y:integer):integer;
begin
newy:=trunc(y1+(y-y0)*ky);
end;

function newrect(t:trect):trect;
var tt:trect;
begin
tt.left:=newx(t.left);
tt.right:=newx(t.right);
tt.top:=newy(t.top);
tt.bottom:=newy(t.bottom);
newrect:=tt;
end;
begin
for i:=1 to elementcount do if selected[i] then
np[i].scaleto(newrect(np[i].caja));

```

```

end;

Procedure TGPS_ARM_CORTEX_A15_MPCORE.AddPoly(closed:boolean);
var i,j:integer;
var slt:^SPltpoint;
begin
if userline.np<2 then exit;
npolypoints:=npolypoints+1;
np[npolypoints]:=TSPolyPol.create(1,0,'New');
slt:=addr(userline.p);
with np[npolypoints] do begin
SetMVals(clred,clblue,1,userline.np,slt^);
m[1].drawer:=drw;
m[1].layer:=2;
m[1].nombre:='Line 1';
CalcCaja;
drawer:=drw;
end;
end;
calccaja;
selected[npolypoints]:=true;
end;

Procedure TGPS_ARM_CORTEX_A15_MPCORE.DeleteElements;
var i,j,k:integer;
ts:tspolypol;
begin
k:=npolypoints;
for j:=k downto 1 do if selected[j] then begin
if j<npolypoints then begin
ts:=np[j];
np[j]:=np[npolypoints];
selected[j]:=false;
ts.free;
end
else begin
np[npolypoints].free;
end;
npolypoints:=npolypoints -1;
end;
end;

Procedure TGPS_ARM_CORTEX_A15_MPCORE.JoinElements(orig,dest:integer);
var i,j,k:integer;
ts:tspolypol;
begin
if orig<npolypoints then begin
ts:=np[orig];
np[orig]:=np[npolypoints];
ts.free;
end
else begin
np[npolypoints].free;
end;
npolypoints:=npolypoints -1;

end;

procedure TGPS_ARM_CORTEX_A15_MPCORE.posterxN(Nx,Ny:integer);
VAR
P: TPosterPrint;
begin
if not ready then exit;
if (( nx<1) or (ny<1) or (nx>10) or (ny>10)) then exit;
screen.cursor:=crhourglass;
p:=TPosterPrint.create(self);
try
with p do begin

```

```

    HorzPages:=nx;
    VertPages:=ny;
    HorzMarge:=100;
    VertMarge:=100;
    Image:='temp.wmf';
    LabelPages:=true;
end;
SaveWMF('temp.wmf');
p.doprint;
finally
p.free;
screen.cursor:=crdefault;
end;
end;

Function TGPS_ARM_CORTEX_A15_MPCORE.GetArea(elem:integer):longint;
begin
GetArea:=trunc(np[elem].GetArea(drw.mo.scale,drw.mo.scale));
end;

Procedure TGPS_ARM_CORTEX_A15_MPCORE.LoadPaletteFromFile(filename:string);
begin
deleteobject(paleta);
paleta:=CreatePaletteFromFile(filename);
end;

Procedure TGPS_ARM_CORTEX_A15_MPCORE.LoadMosaicBitmap(fname:string);
begin
drw.mosaicbmp.loadfromfile(fname);
end;

procedure TGPS_ARM_CORTEX_A15_MPCORE.placegli(ix:integer;p:tpoint);
var pp:tpoint;
begin
if ((ix>0) and (ix<=gli.count)) then begin
gli.glyph[ix].mappos:=p;
drw.convertapoint(p,pp);
gli.glyph[ix].pos:=pp;
gli.glyph[ix].paint(canvas);
end;
end;

procedure TGPS_ARM_CORTEX_A15_MPCORE.addgli(p:tpoint);
begin
gli.add(0);
placegli(gli.count,p);
end;

procedure TGPS_ARM_CORTEX_A15_MPCORE.replaceglis;
var i:integer;
p,pp:tpoint;
begin
for i:=gli.count downto 1 do begin
drw.convertapoint(gli.glyph[i].mappos,p);
gli.glyph[i].pos:=p;
end;
end;

function TGPS_ARM_CORTEX_A15_MPCORE.GlyphCount:integer;
begin
GlyphCount:=gli.count;
end;

Procedure
TGPS_ARM_CORTEX_A15_MPCORE.GlyphSettings(ixglyph:integer;imageix:integer;animate
:boolean;enable:boolean);
begin
if gli.glyph[ixglyph]<>nil then with gli.glyph[ixglyph] do begin

```

```
if imageix < gli.gn then index := imageix;  
animated := animate;  
enabled := enable;  
end;  
end;
```

```
Procedure TGPS_ARM_CORTEX_A15_MPCORE.LoadGlyphsBitmap(fname:string);  
begin  
gli.LoadFromFile(fname);  
end;  
  
end.
```

K6П3-2023

Файл авторського права About.pas

```
unit About;

{Лопотуха Олександр Ігорович, КІ-22М-2}
interface

uses WinTypes, WinProcs, Classes, Graphics, Forms, Controls, StdCtrls,
    Buttons, ExtCtrls;

type
    TAboutBox = class(TForm)
        Panel1: TPanel;
        OKButton: TBitBtn;
        ProgramIcon: TImage;
        ProductName: TLabel;
        Version: TLabel;
        Copyright: TLabel;
        Label1: TLabel;
        procedure OKButtonClick(Sender: TObject);
    private
        { Private declarations }
    public
        { Public declarations }
    end;

var
    AboutBox: TAboutBox;

implementation

{$R *.DFM}

procedure TAboutBox.OKButtonClick(Sender: TObject);
begin
    close;
end;

end.
```