

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Радіотехнічний факультет

Кафедра прикладної радіоелектроніки

До захисту допущено:

В.о. зав.кафедрою

_____ Михайло СТЕПАНОВ

«__» _____ 20__ р.

Дипломний проєкт

на здобуття ступеня бакалавра

за освітньою програмою «Інтелектуальні технології мікросистемної
радіоелектронної техніки»

за спеціальністю 172 «Телекомунікації та радіотехніка»

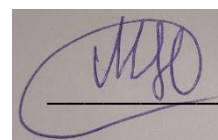
на тему: Пристрій автоматичного керування автомобілем

Виконав (-ла):

студент (-ка) IV курсу, групи РІ-зп71

Наумов Максим Юрійович

Прізвище, ім'я, по батькові



Керівник: Доцент Тарабаров Сергій Борисович



Посада, науковий ступінь, вчене звання,

Прізвище, ім'я, по батькові

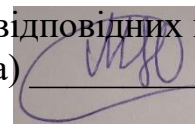
Рецензент: Мирончук Олександр Юрійович

Посада, науковий ступінь, вчене звання,

Прізвище, ім'я, по батькові

Засвідчую, що у цьому дипломному
проєкті немає запозичень з праць інших
авторів без відповідних посилань.

Студент (-ка)



Київ – 2021 року

**Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»**

Радіотехнічний факультет

Кафедра прикладної радіоелектроніки

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 172 «Телекомунікації та радіотехніка»

Освітньо-професійна програма «Інтелектуальні технології мікросистемної радіоелектронної техніки»

ЗАТВЕРДЖУЮ

В.о. зав. кафедрою

_____ Михайло СТЕПАНОВ

«__» _____ 20__ р.

ЗАВДАННЯ

на дипломний проєкт студенту

Наумову Максиму Юрійовичу

1. Тема проєкту «Пристрій автоматичного керування автомобілем», керівник проєкту Доцент Тарабаров Сергій Борисович, затверджені наказом по університету від « 08 » грудня 2021 р. №3935-с
2. Термін подання студентом проєкту 10 грудня 2021 року
3. Вихідні дані до проєкту: напруга живлення +5В , перевірка функціонування автопілоту, робочий діапазон відносної вологості 0% – 95%, діапазон робочих температур -10⁰С – +40⁰С.
4. Зміст пояснювальної записки: вступ, аналіз технічного завдання, огляд існуючих рішень, розробка структурної та електричної схеми, проектування електронного модуля, розроблення алгоритму самостійного функціонування, охорона праці.

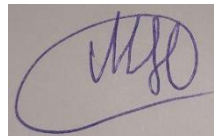
5. Перелік графічного матеріалу (із зазначенням обов'язкових креслеників, плакатів, презентацій тощо) Схема електрична структурна, схема електрична принципова, кресленик друкованої плати, складальний кресленик електронного модуля та приладу.

6. Дата видачі завдання 11 квітня 2021р.

Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1	Синтез структурної схеми	12.11 – 28.10 2021	
2	Синтез принципової схеми	03.11 – 14.11.2021	
3	Вибір елементної бази	03.11 – 14.11.2021	
4	Проробка плати та електронного модуля	15.11 - 28.11.2021	
5	Проробка конструкції приладу	24.11 - 28.11.2021	
6	Виконання розрахунків на працездатність	31.11 – 05.12.2021	
7	Оформлення конструкторської документації	05.12 – 09.12.2021	
8	Оформлення дипломного проєкту	05.12 – 09.12.2021	

Студент



Максим НАУМОВ

Керівник



Сергій ТАРАБАРОВ

ВІДОМІСТЬ ДИПЛОМНОГО ПРОЄКТУ

№ з/п	Формат	Позначення	Найменування	Кількість листів	Примітка
1	A4	PI81.464419.001.T3	Завдання на дипломний проєкт	2	
2	A4	PI81.464419.001 ПЗ	Пояснювальна записка	52	
3	A2	PI81.466396.001	Плата	1	
4	A2	PI81.466396.001СК	Друкований вузол	1	
5	A4	PI81.466396.001.E3	Схема електрична принципова	1	

				PI81.466396.001		
	ПБ	Підп.	Дата			
Розробн.				Пристрій автоматичного керування автомобілем	Лист	Листів
Керівн.			1		1	
Н/контр.					КПІ ім. Ігоря Сікорського Каф. ПРЕ,Гр. PI-п81	
Зав.каф.						

Анотація

Дипломний проєкт складається з 52 сторінок, 9 ілюстрацій та 7 посилань.

Метою данного дипломного проєкту є розробка пристрою автоматичного керування автомобілем. Основним призначенням данного пристрою є частково чи повністю зменшити навантаження від керування автомобілем шляхом передачі його на даний пристрій. У проєкті був проведений огляд аналогів, розроблено структурні та електричні принципові схеми, та проведений огляд сетчу.

Пристрій призначений як для монтажу його в автомобіль, так і створенням автомобіля на його базі. Також пристрій має вивід параметрів екран.

Ключові слова: Arduino, автоматичне керування, автопілот.

Summary

The diploma project consists of 52 pages, 9 illustrations and 7 references.

The purpose of this diploma project is to develop an automatic vehicle control device. The main purpose of this device is to partially or completely reduce the load from driving a car by transferring it to this device. The project reviewed analogues, developed structural and electrical schematics, and reviewed the network.

The device is intended both for its installation in the car, and creation of the car on its basis. The device also has an output screen.

Keywords: Arduino, automatic control, autopilot.

Пояснювальна записка
до дипломного проєкту
на тему: Пристрій автоматичного
керування автомобіля

Київ – 2021 року

Зміст

Вступ		3
1. Аналіз технічного завдання		4
2. Огляд існуючих рішень		4
3. Розробка структурної та електричної схеми		7
3.1 Розробка структурної схеми		7
3.2 Вибір модулів		8
3.3 Розробка схеми електричної принципової		11
4. Проектування електронного модуля		13
4.1 Обґрунтування методу виготовлення друкованої плати		13
4.2 Обґрунтування вибору матеріалу плати		14
4.3 Обґрунтування вибору припою		14
4.4 Вибір класу точності плати		14
5. Розроблення алгоритму самостійного функціонування		15
5.1 Структурна схема алгоритму керування автомобілем		15
5.2 Розробка програми для Arduino		16
6. Охорона праці		19
6.1 Повітря робочої зони		19
6.2 Безпека в надзвичайних ситуаціях		20
6.3 Електробезпека		21
6.4 Електромагнітні випромінювання радіочастотного діапазону		22

						PI-81.466396.001		
Змн.	Лист	№ докум	Підпис	Дата				
Розроб.	Наумов М.				Пристрій автоматичного керування автомобілем	Лім	Арк	Аркушів
Перев.	Тарабаров С						1	
Реценз.								
Н. Контр.								
Затвердив	Тарабаров С.							

6.5 Оптичне випромінювання монітору	23
6.6 Електростатичне поле ВДТ ПЕМ	23
Висновки	26
Перелік посилань	27
Додаток А. Технічне завдання	28
Додаток Б. Лістинг програми керування	29

Вступ

Ні для кого не секрет, що сьогоденні автомобілі це транспортні засоби значно складнішого характеру ніж 20 років тому. З плином часу та розвитком технологій з'явилися електронні системи, що дозволяють відслідковувати найдрібніші похибки в параметрах роботи автомобіля, що значно спрощує його догляд і роботу водієві.

Однією із таких електронних систем є пристрій автоматичного керування автомобілем, що являє собою багатофункціональний прилад якій дає змогу спростити життя водієві, а саме бере на себе функції керування транспортним засобом і автоматично відслідковує все, що відбувається як в самому транспортному засобі так і поза його межами. Маючи доволі обширну кількість функцій та можливостей. Він спрощує експлуатацію транспортного засобу до меж можливого.

Така система окрім спрощення життя водієві забезпечила кращу ситуацію з безпекою на дорозі, оскільки швидкість реагування її на навколишні фактори швидша за швидкість реагування людини. Ця швидкість забезпечена завдяки закладеним алгоритмам в “мізки” данного пристрою.

Закладені алгоритми диктують данній системі як реагувати на ті чи інші обставини, перепони чи загрози на дорозі.

Метою моєї дипломної роботи було дослідити, відтворити та вдосконалити одну із таких систем на базі портативного мікроконтролера Arduino Uno. Представлена робота демонструє можливість створення таких систем в бюджетному сегменті для середньостатистичного споживача, а також можливість для здешевлення виробництва автомобілів з такими системами.

1. Аналіз технічного завдання

З огляду до вимог технічного завдання було вирішено, що пристрій буде інтегруватись в системи автомобіля, виносні давачі будуть підключатись до плати за допомогою роз'ємів. Магнітометр, який виконує функцію компасу має знаходитись в віддаленні від основної електроніки, краще на даху, чи над ним. Також слід передбачити відповідні кріплення для давачів, для надійного кріплення на автомобіль, та в разі потреби швидкої заміни.

2. Огляд існуючих рішень

Пристрої для автоматичного керування автомобілем починають широко використовуватись для керування легковими та вантажними автомобілями в промисловості та в побуті.

Зараз автопілот почав використовуватись в таких галузях як:

— Таксі;

- Китай – Alibaba Group;
- Росія – ЯндексТаксі.

— Автобуси;

- Фінляндія – FABULOS;
 - Китай – Alpha;
 - Швеція - Scania Citywide LF;
- Та інші.

— Вантажоперевезення

- США – TuSimple;
- Китай - China Mobile і компанією Jiuquan Iron and Steel Group.

Наразі автопілоти мають такі марки автомобілей:

— Tesla Motors - Tesla Model;

					PI-81.466396.001	Арк.
						3
Зм.	Арк.	№ докум	Підпис	Дата		

- Waymo;
- Honda;
- Volkswagen;
- Alibaba Group;
- Google (нині Alphabet);
- Toyota.

Наразі весь транспорт, в якому наявний автопілот знаходиться в тестовому режимі та пересувається тільки з пілотом. Пілот в свою чергу має спостерігати за ситуацією на дорозі, технічним станом та уникати аварійних ситуацій. Зазвичай для створення автопілоту для автомобіля використовуються деталі (радіокрмпоненти та вузли), які виробляють відомі концерни які заповнили ринок. Але вони мають декілька недоліків:

- 1) висока ціна;
- 2) неможливість встановити на більшість існуючих моделей авто;
- 3) складність виготовлення, отримання та монтажу модулів;
- 4) більшість промислових авто наразі мають низьку швидкість, що затрудняє рух в місті;
- 5) при повній автоматизації транспорту велика кількість людей втратить роботу.

Також автопілот має велику кількість переваг над звичайним транспортом:

- 1) перевезення вантажів в небезпечних умовах, таких як в місцях катаклізм чи в місцях проведення бойових дій;
- 2) зниження ціни за перевіз вантажів;
- 3) більш економне використання пального;
- 4) економія часу так як під час дороги буде можливість зайнятись своїми справами;
- 5) зниження кількості ДТП та людських жертв;

					PI-81.466396.001	Арк.
						5
Зм.	Арк.	№ докум	Підпис	Дата		

б) в майбутньому підвищення пропускної здатності доріг за рахунок звуження ширини смуг.

Наразі до автопілота мають велику кількість вимог таких як:

- 1) надійність програмного забезпечення яке встановлюється на безпілотні автомобілі;
- 2) автономний автомобіль має до сантиметрів знати де та в якому положенні він знаходиться;
- 3) карти мають відображати положення автомобіля, та давати йому розуміння що знаходиться за поворотом (ДТП, затори і т.д.);
- 4) безпілотне авто має розуміти трактування всіх знаків та розміток які присутні на дорогах загального користування ;
- 5) вони мають слідкувати за всім трафіком, що відбувається навколо них та уникати можливих ДТП (Автомобілі «Tesla» вже мають можливість передбачати можливі дорожні ситуації такі як ДТП які можуть відбутись перед автомобілем чи можливе підрізання авто стороннім водієм та уникати їх);
- б) оминати перепони які можуть траплятись на шляху автомобіля (припарковані автомобілі на узбіччі, раптова поява людей та звірів на проїжджій частині, блокування дороги обвалами, поваленими деревами і т.д.);

Під час руху враховувати погодні умови (мокра, слизька, засніжена дорога чи туман);

					PI-81.466396.001	Арк.
						6
Зм.	Арк.	№ докум	Підпис	Дата		

3. Розробка структурної та електричної схеми

Проаналізувавши принцип роботи, та визначившись з функціональними особливостями пристрою було вирішено створити функціональну схему.

3.1 Розробка структурної схеми

Структурна схема пристрою наведена на рис. 3.1.



Рисунок 3.1 – Структурна схема пристрою

За рахунок датчиків та зчитування координат з GPS модуля, мікроконтроллер має можливість знати поточне місце знаходження, напрямок руху, наявність перепон та оминати їх.

Зм.	Арк.	№ докум	Підпис	Дата

PI-81.466396.001

Арк.

7

Мікроконтролер. Пристрій побудований на основі мікроконтролера *Atmega 328* (рис.3.2).

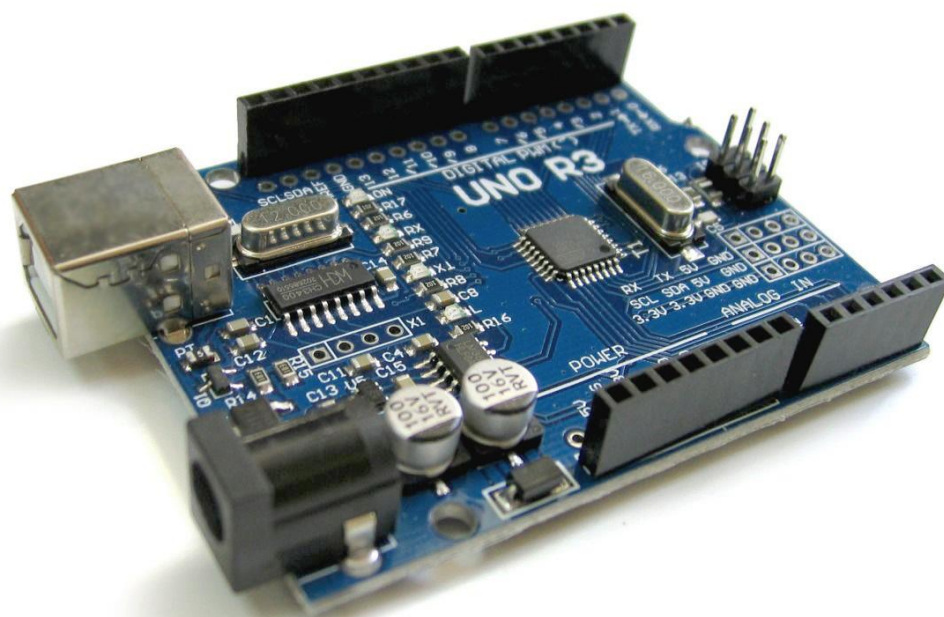


Рисунок 3.2 – Модуль Arduino Uno [1]

Це 8-бітний мікроконтролер з тактовою частотою 20 МГц. Мікроконтролер має на борту 16-бітний таймер, шість каналів АЦП та шість каналів широтно- імпульсної модуляції (ШІМ), лічильник реального часу з окремим генератором.

На базі даного мікроконтролера існує велика кількість плат з потрібним нам функціоналом, тому слід використати готовий модуль Arduino Uno.

ЖК-дисплей. SPI/I2C 1602 LCD – Це ЖК-дисплей з підсвічуванням і встановленим SPI/I2C конвертором, на рахунок від більшості дисплеїв, керується контролером Arduino за допомогою 2-х проводового зв'язку, що заощаджує цифрові піни контролера для додаткової периферії. (рис.3.3).

Зм.	Арк.	№ докум	Підпис	Дата

PI-81.466396.001

Арк.

8



Рисунок 3.3 – Модуль SPI/I2C 1602 LCD [2]

Драйвер. SunFounder L293D – це двоканальний драйвер, що дозволяє працювати з великими напругами та струмами, за допомогою мікросхеми L293D є можливість використовувати двигуни постійного струму і джерела живлення до 36v (рис.3.4).

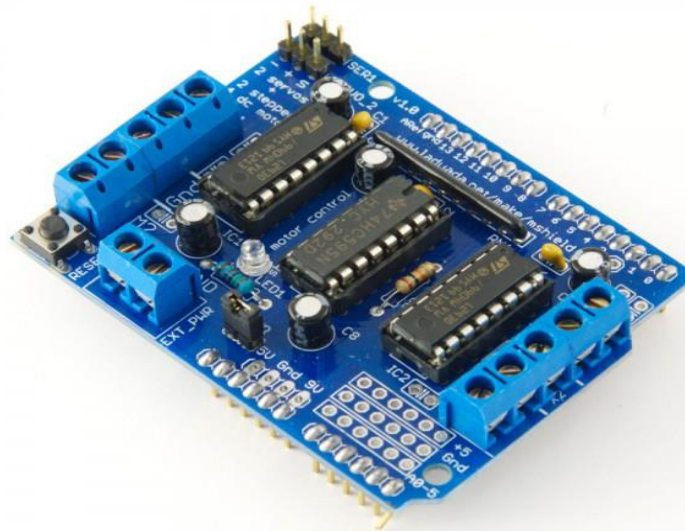


Рисунок 3.4 – Модуль SunFounder L293D [3]

Зм.	Арк.	№ докум	Підпис	Дата

PI-81.466396.001

Арк.

9

GPS модуль. Adafruit Ultimate GPS – це GPS модуль, який прийшов на заміну своїм аналогам, має менші розміри ніж були та побудований на базі чіпсета MTK 3339, високоякісного GPS модуля який може відстежувати до 22-х супутників на 66 каналах, має високочутливий приймач та вбудовану антену.

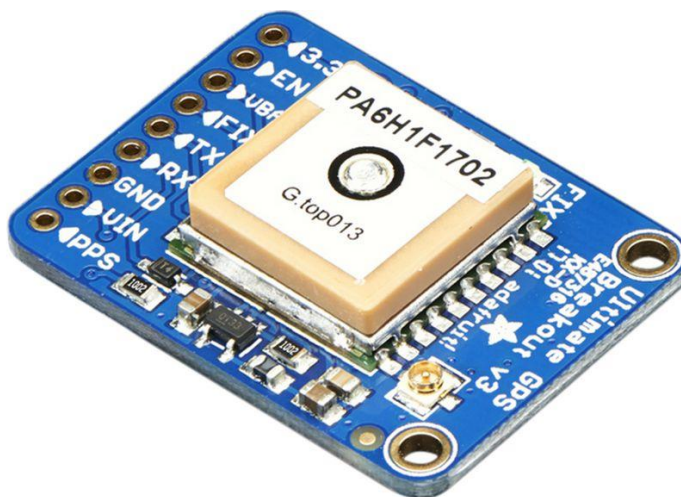


Рисунок 3.5 – Модуль Adafruit Ultimate GPS [4]

Магнітометр. Магнітометр HMC5883L – використовується як компас, має бути встановлений горизонтально, та на невеликій відстані від зовнішніх електромагнітних перепон, працює по протоколу I2C.

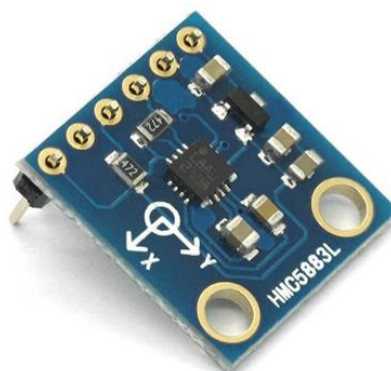


Рисунок 3.6 – Модуль Магнітометр HMC5883L [5]

Зм.	Арк.	№ докум	Підпис	Дата

PI-81.466396.001

Арк.

10

Далекомір. Ультразвуковий далекомір HC-SR04 [6] – має просту будову, легко підключає, має високу надійність, в даному проекті використовується як давач перепони, має вимірюваний діапазон до 500 см, точність до 0.3см, кут відбиття до 15°.



Рисунок – 3.7 Модуль Ультразвуковий далекомір HC-SR04 [6]

Електричні зв'язки складових структурної схеми відображені схемою принциповою.

3.2 Розробка схеми електричної принципової

Схема електрична принципова наведена на рис. 3.8.

Зм.	Арк.	№ докум	Підпис	Дата

PI-81.466396.001

Арк.

11

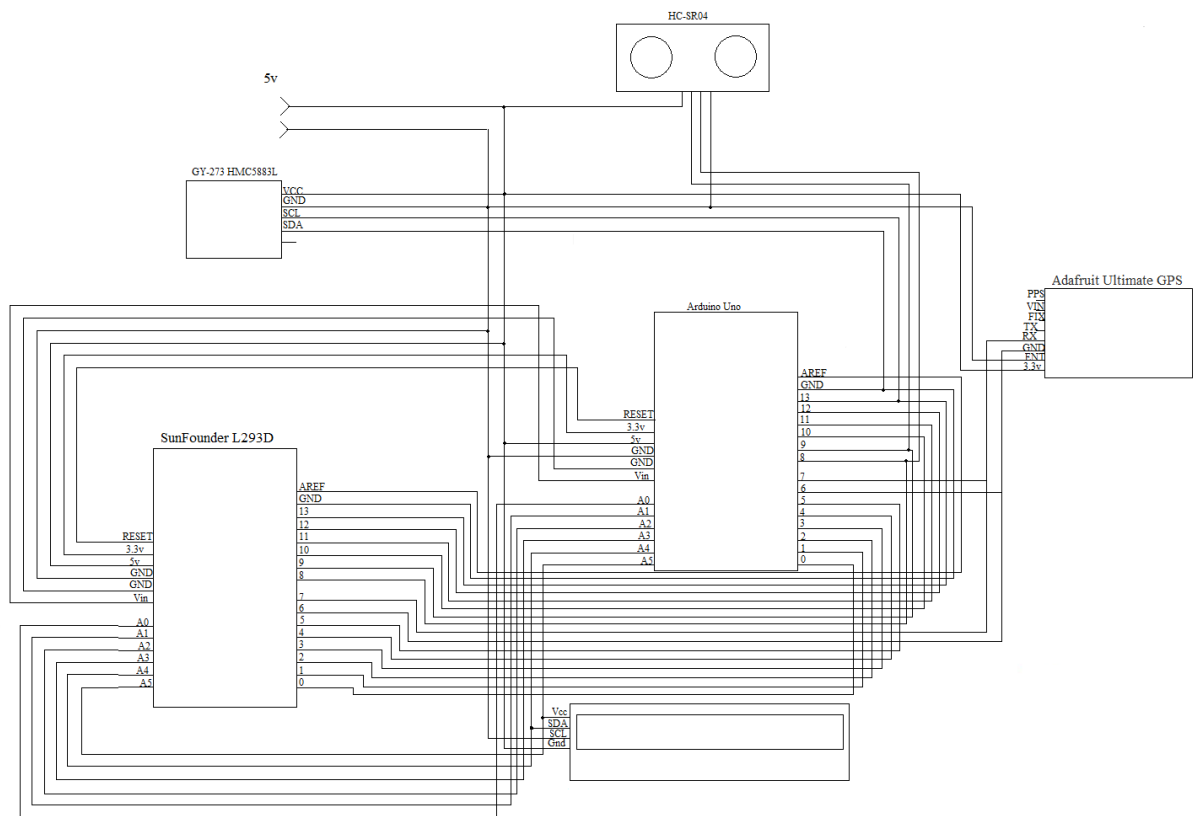


Рисунок 3.8 – Схема електрична принципова

Модуль Arduino Uno являється керуючим модулем данного пристрою, SunFounder L293D – модуль, який відповідає за керування двигунами, HC-SR04 виявляє перепони на шляху, магнітометр HMC5883L в парі з Adafruit Ultimate GPS виявляє в яку сторону повернутий пристрій та його координати.

Зм.	Арк.	№ докум	Підпис	Дата

PI-81.466396.001.E3

Арк.

12

4 Проєктування електронного модуля

В розділі наведено обґрунтування методу виготовлення друкованої плати, матеріалу плати, використаного припою та матеріалу плати.

4.1 Обґрунтування методу виготовлення друкованої плати

Від матеріалу який використовується при виготовленні друкованої плати (ДП) залежать конструктивно-технологічні, експлуатаційні та техніко-економічні характеристики приладу. На сьогоднішній день стало поширеним використання слоїстих та литих діелектриків, фольгованих електролітичною мідною фольгою, товщина якої може становити 5, 18, 35, 50 мкм.

До матеріалу ДП висуваються наступні вимоги:

- чистота фольги – 99,5 %;
- шорсткість не гірше 0,4 мкм;
- висока теплостійкість (витримувати до 260 С° впродовж 5-20 с);
- питомий опір діелектрика – не менше 5×10^{11} Ом/см.

Від матеріалу залежать електроізоляційні властивості, механічна міцність, стабільність параметрів при впливі агресивних середовищ і кліматичних умов, що змінюються. Зазвичай для даних цілей використовуються гетинакс і склотекстоліт.

Склотекстоліт має високі характеристики в широкому діапазоні температур (від -60 до +150) С°, має низьке водопоглинання 0,2–0,8% та високий об'ємний поверхневий опір.

					PI-81.466396.001	Арк.
						13
Зм.	Арк.	№ докум	Підпис	Дата		

4.2 Обґрунтування вибору матеріалу плати

Для виготовлення ДП обрано склотекстоліт *FR-4*. Це пресований матеріал основою якого є склотканина, просочена епоксидним сполучником. Цей матеріал відрізняється високою теплостійкістю (до 180 С°), а також високими електроізоляційними і механічними властивостями; добре обробляється. З двох сторін нанесена мідна електролітична фольга. Товщину фольги обрано 35 мкм, товщину основи – 1,6 мм.

4.3 Обґрунтування вибору припою

Враховуючи доступність, міцність та ціну припою, було обрано без свинцевий припій SAC 305, даний припій забезпечує міцні паяльні з'єднання, та має підвищену втомну міцність, сумісний з будь якими флюсами.

4.4 Вибір класу точності плати

Випуск друкованих плат 2-го і 1-го класів здійснюється на рядовому обладнанні, а іноді навіть на обладнанні, не призначеному для виготовлення друкованих плат. Такі ПП з невисокими (і навіть з низькими) конструктивними параметрами призначені для недорогих пристроїв з малою щільністю монтажу. До цього класу належать друковані плати любительського і макетного рівня, часто одиничного або дрібносерійного виробництва. Так як в нас не дуже висока щільність монтажу ми використовуємо плату другого класу точності.

					PI-81.466396.001	Арк.
Зм.	Арк.	№ докум	Підпис	Дата		14

5 Розроблення алгоритму самостійного функціонування

Мови програмування для платформи Arduino основані на C/C++ та скомпоновані з бібліотекою AVR Libc, що дозволяє використовувати будь-які її функції. На даний час програмування на платформі Arduino — це найзручніший спосіб програмування приладів на мікроконтролерах.

Сучасні мікроконтролери мають досить велику швидкодію, наразі розроблено мови програмування високого рівня, що полегшують розробку програмного забезпечення, та дозволяють писати код для Arduino людям, які не знають досконало мови програмування.

В даній дипломній роботі використовувалась мова програмування C++, бо данна система не є складною та вимогливою та не потребує високого рівня в знанні програмування, сама платформа Arduino має достатній об'єм та швидкодію, для данного програмного забезпечення та софт досить легко компілюється, та завантажується в саму плату, за допомогою Arduino IDE.

Arduino IDE – інтегроване середовище розробки, створене для Windows, MacOS і Linux, розроблене на мовах C та C++, призначена для створення і завантаження програм, на Arduino-сумісні плати.

5.1 Структурна схема роботи програми для Arduino

Структурна схема являється досить легкою та заключається в тому, що даний пристрій аналізує своє положення в просторі, передає інформацію на плату Arduino, та в свою чергу звіряє інформацію де наразі вона знаходиться і в якому напрямку їй потрібно пересуватись, та в випадку якщо вона бачить перепону, вона коректує маршрут.

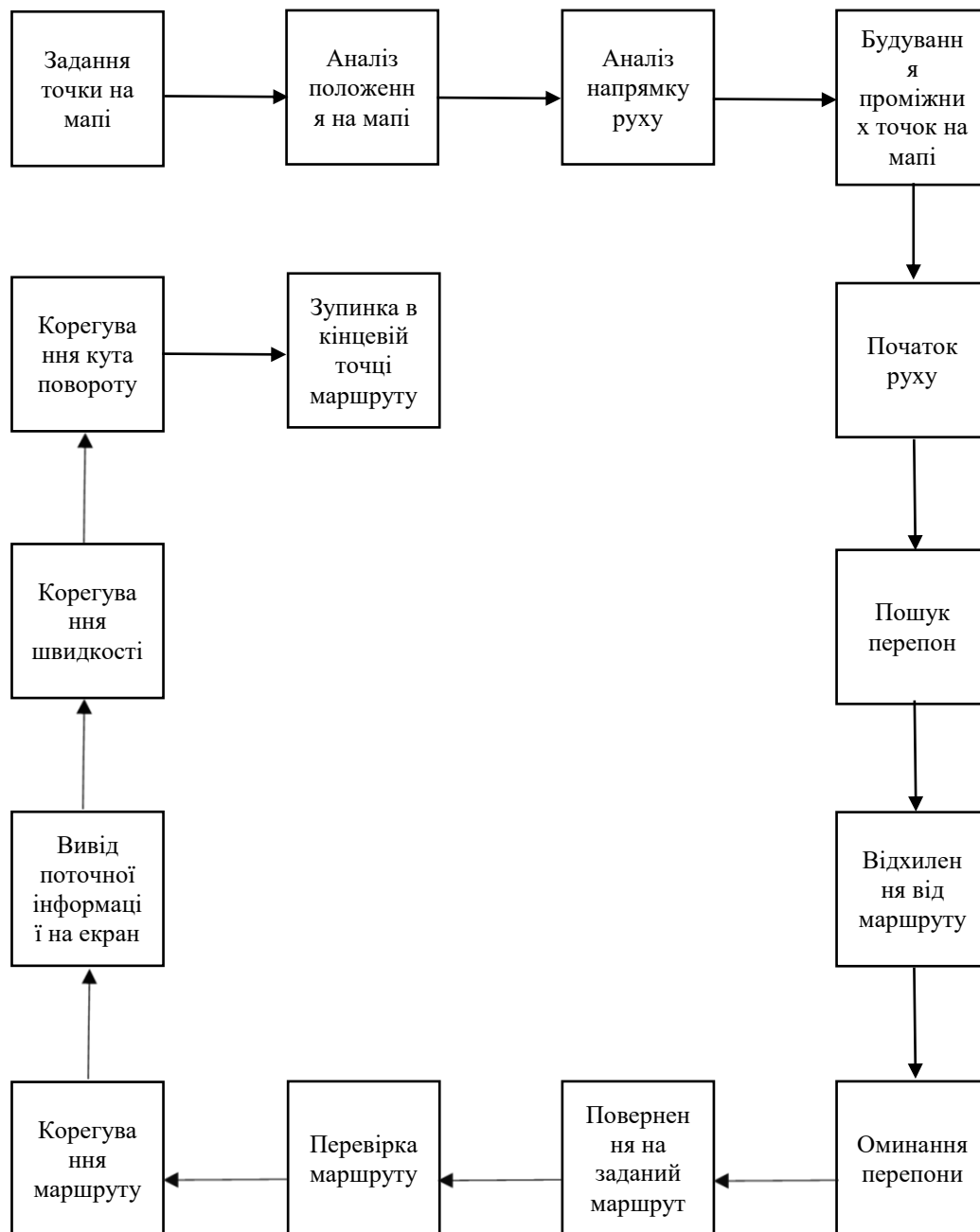


Рисунок 5.1 - Структурна схема роботи програми для Arduino

За рахунок давачів та зчитування координат з GPS модуля, мікроконтроллер має можливість знати поточне місце знаходження, напрямок руху, наявність перепон та оминати їх.

Зм.	Арк.	№ докум	Підпис	Дата

PI-81.466396.001

Арк.

16

5.2 Розробка програми для Arduino

Розберемо код по блокам

1. Наразі проводимо налаштування драйверів
2. Проводимо налаштування магніметра (компаса); використовує I2C
3. Проводимо налаштування контролерів двигуна як для приводу, так і для рульового управління (повороту).
4. Підключення LCD Display
5. Проводимо налаштування ультразвукового дальнометра
6. Проводимо навігацію компасу(макгітомеру)
7. Підключення GPS навігатору
8. Перевіряємо точки в маршруті
9. Прописуємо повороти
10. Обираємо в уникнення об'єкту
11. Виставляємо швидкість (range: 0 - 255)
12. Прописуємо оновлення GPS
13. Прописуємо порівняння координат
14. Маємо можливість вимкнути порівняння
15. Прописуємо підключення та запуск дисплею
16. Прописуємо команду запуску компасу
17. Запускаємо GPS і встановлюємо конфігурацію
18. Очікування сигналу GPS
19. Очікування натиску клавіші для початку руху
20. За відсутності кнопки, маємо можливість поставити зворотній відлік
21. Прописуємо початкову точку старту
22. Прописуємо оброблення інформації про GPS
23. Додаємо навігацію і прорахування оптимального маршруту
24. Додаємо можливість оминати перешкоди
25. Маємо можливість обрати час оновлення показання монітору
26. Після оновлення даних GPS оновлює положення та курс
27. Оновлення курсу та відстані до точки виходячи з нашого нового положення
28. Обчислення кута поворота, для направлення до місця призначення
29. Налаштування обертання компаса
30. Обчислення, в який бік повернути щоб стати на правильну дорогу

					PI-81.466396.001	Арк.
Зм.	Арк.	№ докум	Підпис	Дата		17

Якщо похибка мінімальна, не звертати з оптимального маршруту

31. Додаємо можливість сповільнення перед поворотом

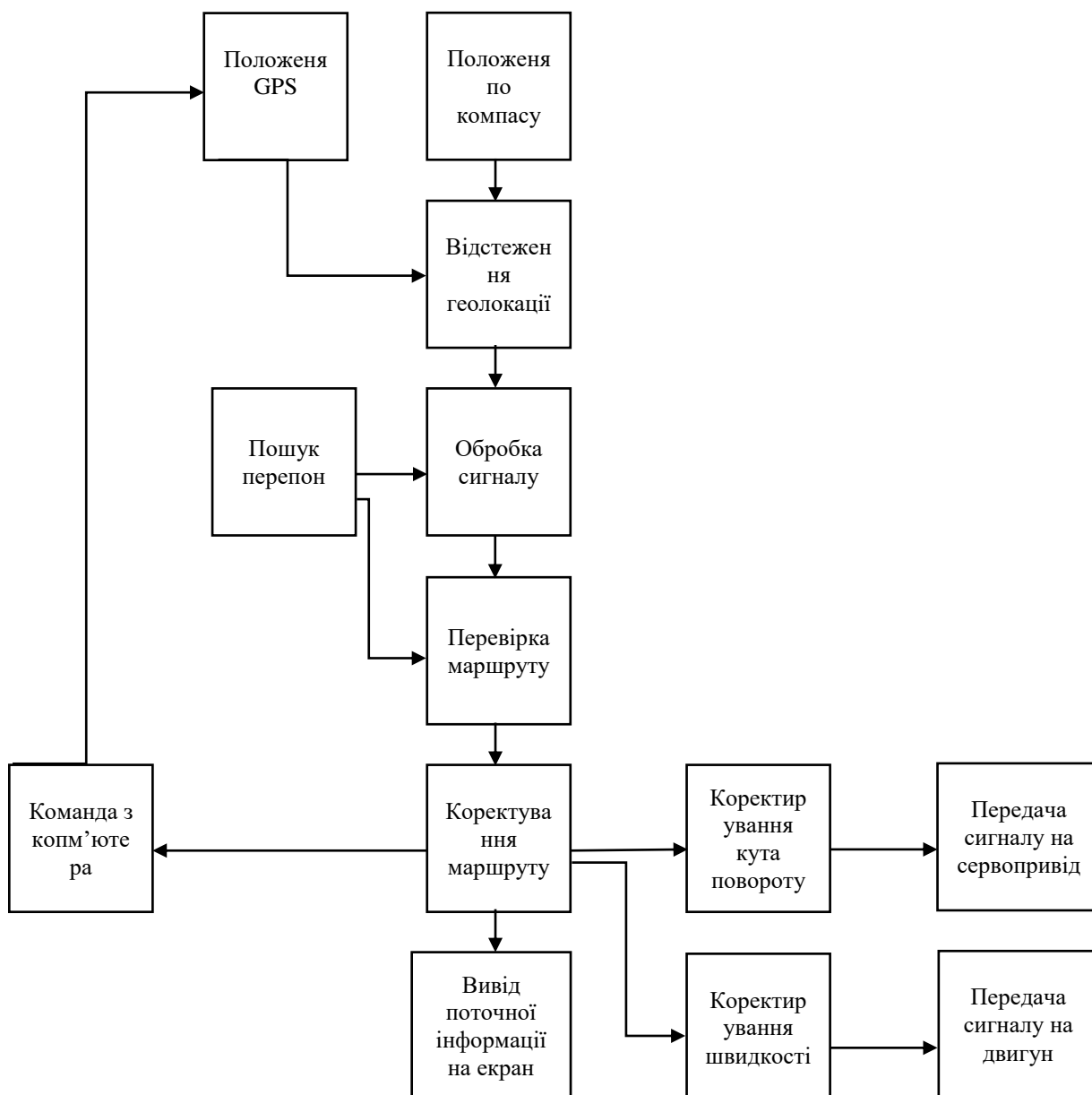
32. Зупинка із-за сильного зближення з об'єктом

33. В випадку, якщо неможливо об'їхати, то додаємо можливість від'їхати назад.

Код програми наведено у додатку Б.

5.3 Принцип роботи пристрою

В структурній схемі наведеній на рисунку 5.2 відображено принцип роботи готового пристрою, а саме передача сигналів і дії пристрою. Від задання маршруту, початку руху до об'їзду перепон та регулювання швидкості і кута повороту.



Зм.	Арк.	№ докум	Підпис	Дата

PI-81.466396.001

Арк.

19

Рисунок 5.2 - Структурна схема роботи програми для Arduino

6. Охорона праці

6.1 Повітря робочої зони

Під час роботи ВДТ при наявності радіовипромінювань високих частот, сильних електричних полів, а також НРВ в повітрі закритих помешкань створюється підвищене утримання позитивних і негативних легких іонів.

Експериментально встановлено, що аероіони є найбільш чутливим фізичним індикатором забруднення повітря, а головне - роблять безпосередній вплив на здоров'я людини. Негативні іони діють цілощодо організму, підвищуючи його опір, у той час як позитивні іони гальмують життєдіяльність організму і сприяють розвитку психічних захворювань.

Кількість легких аероіонів повинно відповідати вимогам «Санітарно-гігієнічних норм припустимих рівнів іонізації повітря виробничих і суспільних помешкань» №2152-80.

6.2 Безпека в надзвичайних ситуаціях

Пожежна безпека

Відповідно до НАПБ Б.03.002-2007 та ОНТП 24-86 робоче приміщення лабораторії відноситься до категорії В по вибухопожежній небезпеці.

Відповідно до ПУЕ-87 та ДНАОП 0.00-1.32-01 клас робочих зон приміщення лабораторії по пожежонебезпеці - П-Па. Можливими причинами пожежі в приміщенні є несправність електроустаткування, коротке замикання проводки, і порушення протипожежного режиму (використання побутових нагрівальних приладів, паління).

У зв'язку з цим, відповідно до вимог ПБЕ та ПУЕ, необхідно передбачити наступні заходи.

- Ретельну ізоляцію всіх струмоведучих провідників до робочих

					РІ-81.466396.001	Арк.
						20
Зм.	Арк.	№ докум	Підпис	Дата		

місце, періодичний огляд та перевірку ізоляції.

- Строге дотримання норм протипожежної безпеки на робочих місцях.
- Відповідні організаційні заходи (заборона паління, інструктаж). Для гасіння пожежі в робочому приміщенні лабораторії (клас пожежі - наявність електрообладнання під напругою) використовуються вогнегасники ОП-1 — “Момент” (2 шт.). Додатково в коридорі розташовані вогнегасники ОХП-10. Також на сходовій клітці розташований пожежний кран. Така кількість первинних засобів пожежогасіння відповідає вимогам ISO 3941-77, якими передбачена обов’язкова наявність двох вогнегасників до 100 м² площі підлоги для приміщення типу конструкторське бюро.

Згідно вимог ДБН В.2.5-56-2014 робоче приміщення лабораторії необхідно оснастити системою автоматичної пожежної сигналізації.

Будинок має два евакуаційних виходів: через головнийхід і додатковий евакуаційний вихід. Шляхи евакуації відповідають установленим нормам. Двері відкриваються назовні. Коридор веде до двох сходових кліток, одна з яких виходить безпосередньо на вулицю, а друга має вихід на вулицю через вестибуль і головний вхід. Сходова клітка виконана з непальних матеріалів. Сходи мають природне бічне освітлення і штучне евакуаційне освітлення. Сходові площадки ширше коридорів. Усі співробітники ознайомлені з планом евакуації.

6.3 Електробезпека

Все наявне в лабораторії електроустаткування можна віднести до І (системні блоки) та ІІ (ВДТ) класів щодо електрозахисту (ГОСТ 12.2.007.0-75).

Помешкання лабораторії по рівню небезпеки поразки людей електричним струмом згідно ОНТП 24-86 та ПБЕ можна віднести, до помешкань без підвищеної небезпеки, тому що:

- відносна вологість повітря не перевищує 75%;
- матеріал підлоги (паркет) є діелектриком;

					PI-81.466396.001	Арк.
Зм.	Арк.	№ докум	Підпис	Дата		

- температура повітря не досягає значень, більших 35 °С;
- відсутня можливість одночасного доторку людини до з'єднаних із землею частин металоконструкцій будівлі, технологічних апаратів, механізмів і т.п., з одного боку, і до металевих частин електроустаткування - з іншого боку;
- відсутні хімічно агресивні середовища.

Електромережа трьохфазна з глухозаземленою нейтраллю, із зануленнями повторним заземленням нульового проводу відповідно до вимог ПБЕ та ПУЕ. Мережа обладнана автоматом струмового захисту, розрахованого на струм 10А. Час спрацювання автомату 0,1с.

Для того, щоб не допустити ураження людини електричним струмом при виникненні аварійних ситуацій необхідно заземлити все обладнання, що працює від мережі 220В, 50Гц. Опір нульового проводу повинен бути таким, щоб при замиканні на корпус або нульовий провід виникав струм короткого замикання, сила якого повинна перевищувати в 1,4 рази номінальний струм спрацювання автомата струмового захисту (при струмі короткого замикання менше 100А).

Знайдемо напругу на корпусах електрообладнання при його аварійному режимі роботи. Опір заземлюючих пристроїв не перевищує значень встановлених ГОСТ12.1.038-88, яке складає при часі дії 0,1с – 500 В. Дане значення не перевищує допустимих значень згідно вимог.

Виконано всі необхідні заходи щодо електробезпеки відповідно до ГОСТ12.3.019-80 та ПУЕ. Додаткових заходів по електробезпеці впроваджуватине потрібно.

6.4 Електромагнітні випромінювання радіочастотного діапазону

ВДТ на основі ЕПТ є джерелом випромінювань і полів різноманітних частот. Основними джерелами є блоки кадрової і рядкової разгортки, відрізок високовольтного проводу й анод. Ця напруга від блока разгортки до анода трубки передається за допомогою неекранованого відрізка високовольтного

					PI-81.466396.001	Арк.
Зм.	Арк.	№ докум	Підпис	Дата		21

проводу, розташованого на зворотній стороні кінескопа.

З однієї сторони він через обмотку автотрансформатора заземлений на корпус, а з іншої сторони живить анод ЕПТ. Тому його можна уявити в якості коротко заземленого штиря без ємності на кінці, тобто як антену, що випромінює. Випромінювання від анода ЕПТ, діаграма спрямованості якого має головний максимум, перпендикулярний до площини екрана кінескопа, безпосередньо спрямоване на людину, що працює на ВДТ. Припустимі норми для напруженості електричного поля на відстані 1 м від екрана зазначені в ГОСТ 12.1.006-84.

Відповідно до паспортних даних використовуваних в робочому приміщенні ВДТ рівні їх ЕМВ відповідають вимогам "Тимчасовим санітарним нормам для В.Ц." №4559-88 і ГОСТ 12.1.006-84 і не мають загрози для користувача.

6.5 Оптичне випромінювання монітору

Як вказувалося раніше, енергія електронного пучка за допомогою люмінофора перетворюється в тому числі й у випромінювання оптичного діапазону, що містить іонізуюче ультрафіолетове (УФВ), видимого діапазону й інфрачервоне (ІЧВ) випромінювання. УФВ залежить від використовуваного складу люмінофору й в основному пов'язано з зелено-блакитними видами люмінофору, а не з жовто-жовтогарячими. Інтенсивність випромінювання видимого діапазону 400-700 нм залежить від відстані. Яскравість випромінювання від екрана залежить від типу ЕПТ і анодні напруги.

Відповідно до НРБ-76/87 визначається нормування іонізуючого випромінювання. Для використовуваних ВДТ фактичне значення щільності потоку УФВ і ближнього ІЧВ по паспортним даним істотно нижче чинних норм відповідно до (Санітарних норм мікроклімату і виробничих помешкань №4559-88) і не мають загрози для користувача.

					PI-81.466396.001	Арк.
Зм.	Арк.	№ докум	Підпис	Дата		22

6.6 Електростатичне поле ВДТ ПЕМ

- Джерелом електростатичного поля є напруга, підведена до аноду ЕПТ, що для різних типів кінескопів лежить у межах 6-30 кВ. На ЕПТ накопичується електростатичний заряд. Розмір цих зарядів залежить від таких чинників: потенціалу розгону для прискорення руху електронів у напрямку до кінескопу;
 - накопичення заряджених часток на поверхні кінескопу (який буде зменшувати результуюче поле);
 - вологості повітря.

На відстані 0.1-0.5 м від екрана напруженість електричного поля слабко залежить від відстані і її можна вважати постійною, далі вона зменшується обернено пропорційно відстані, а на великих відстанях - обернено пропорційно квадрату відстані. Максимальна напруженість поля знаходиться у самої поверхні екрана.

Для учнів і студентів, відповідно до –Тимчасової санітарної норми і правила устрою устаткування, утримання і режиму роботи на ЕОМ і ВДТ у кабінетах обчислювальної техніки і дисплейних класів усіх типів середніх навчальних закладів № 5146-89, напруженість електростатичного поля при роботі на ВДТ повинна бути не більш 15 кВ/м, що й забезпечується у використаному ВДТ.

6.7 Заходи щодо нормалізації умов праці

Для усунення шкідливого впливу НРВ на організм інженерів-програмістів можна рекомендувати скоротити час перебування за екраном дисплея до 4 годин у зміну, причому після 2 годин безупинної роботи радиться 30-хвилинна перерва (бажано на відкритому повітрі).

З метою автоматичної підтримки параметрів мікроклімату в необхідних межах протягом всіх сезонів року,

					PI-81.466396.001	Арк.
						23
Зм.	Арк.	№ докум	Підпис	Дата		

очищення повітря від пилюки і шкідливих речовин, зниження рівня іонізації в помешканні лабораторії провадиться вентиляція за допомогою двох витяжних вентиляторів типу АИСИ-4 і щоденневологе прибирання.

Світильники розташовані в два ряди і під'єднані до різних фаз електромережі для усунення мерехтіння світлового потоку (коефіцієнт пульсацій менше 10%).

Для ослаблення шкідливого впливу електростатичних полів у лабораторії застосовуються захисні скляні фільтри (екрани) з електропровідним покриттям, що має відвід для заземлення, що прикріплюються на екран монітора.

Для зниження рівня шуму в лабораторії використовується шумопоглинаюче облицювання з перфорованим покриттям: гіпсові плити товщиною 7-9 мм із заповненням із склотканини.

					PI-81.466396.001	Арк.
						25
Зм.	Арк.	№ докум	Підпис	Дата		

Висновок

У дипломному проекті наведені результати розробки пристрою автоматичного керування автомобілем. В ході виконання роботи над дипломним проектом підібрана елементна база яка опирається на якісні та не дорогі елементи. Розроблена друкована, загальний вигляд конструкції приладу та матеріали плати.

Також за допомогою програмного коду вдалося максимально автоматизувати всі процеси. Автопілот може пересуватись по вулицям за допомогою GPS модуля який направляє пристрій по заданим точкам на мапі. За допомогою давачів має можливість оминати більшість перепон, а плата керування регулює швидкість та відслідковує скільки саме ще займе маршрут.

В середовищі Altium Designer було спроектовано плату та готову 3D модель. Також створена конструкторська документація. Розроблений прилад задовольняє вимоги технічного завдання.

Приблизна ціна компонентів автопілоту – 873 UAN, або 33 USD.

					PI-81.466396.001	Арк.
						26
Зм.	Арк.	№ докум	Підпис	Дата		

Перелік посилань

1. Модуль Arduino Uno - <http://arduino.ru/Hardware/ArduinoBoardUno>
2. Модуль SPI/I2C 1602 LCD - <https://www.chipdip.ru/product/iic-i2c-twi-1602-lcd-blue>
3. Модуль SunFounder L293D - <https://radioprogram.ru/shop/merch/42>
4. Модуль Adafruit Ultimate GPS - <https://www.adafruit.com/product/746>
5. Модуль Магнітометр HMC5883L - https://cdn-shop.adafruit.com/datasheets/HMC5883L_3-Axis_Digital_Compass_IC.pdf
6. Модуль Ультразвуковий далекомір HC-SR04 - <https://www.chipdip.ru/product/hc-sr04>
7. Методичні вказівки до розробки розділу «Охорона праці» в дипломних проектах/роботах для студентів РТФ освітнього рівня – бакалавр. Укладачі: С.Ф. Каштанов, А.Н. Гусєв – Київ НТУУ –КПІ

					PI-81.466396.001	Арк.
Зм.	Арк.	№ докум	Підпис	Дата		27

Додаток А Технічне завдання (копія)

Вимоги життєздатності та стійкості до зовнішніх впливів і чинників

Кліматичне виконання виробу – УХЛ 1 (згідно ГОСТ 15150-69);

Вимоги надійності

Пристрій повинен бути відновлюваний і ремонтнопридатний згідно ДСТУ 2470-94.

Час безвідомої роботи 20000 годин.

Вимоги до експлуатації, зручності технічного обслуговування та ремонту

Умови експлуатації:

- 1) Діапазон робочих температур від -10 °С до 40 °С;
- 2) Діапазон відносної вологості повітря від 0% до 95%;

Передбачити легкий доступ для ремонту.

Вимоги до взаємозамінності

Має бути можливість заміни елементної бази приладу.

Вимоги до вартості

Менше 50 USD.

Вимоги до функціоналу

- 1) Можливість самостійного пересування;
- 2) Можливість самостійного оминання перепон;
- 3) Можливість самостійного регулювання швидкості;
- 4) Можливість самостійної зупинки.

Вимоги до виводу необхідних параметрів

- курс до поточної маршрутної точки;
- відстань (в метрах) до поточної маршрутної точки;
- відстань до будь-яких об'єктів;
- швидкість транспортного засобу;

										Арк.
										28
Зм.	Арк.	№ докум	Підпис	Дата						

PI-81.466396.001

- поточне місце знаходження робота між маршрутними точками.

Пристрій буде інтегруватись в системи автомобіля, виносні давачі будуть підключатись до плати за рахунок роз'ємів. Магнітометр, який виконує функцію компасу має знаходитись в віддаленні від основної електроніки, краще на даху, чи над ним. Також слід передбачити відповідні кріплення для датчиків, для надійного кріплення на автомобіль, в разі потребі швидкої заміни.

					PI-81.466396.001	Арк.
						28
Зм.	Арк.	№ докум	Підпис	Дата		

Додаток Б Лістинг програми керування

```
#include // used by: motor driver
#include // motor driver
#include // motor driver
#include // Ping sonar
#include // LCD library
#include // part of mag sensor
#include // mag sensor
#include // custom class to manage GPS waypoints
#include // GPS
#include // used by: GPS
#include // used by: GPS
#include // simple moving average class; for Sonar functionality
#include // include for stepper motor turning

// Проводимо налаштування магніметра (компаса); використовує I2C
Adafruit_HMC5883_Unified compass =
Adafruit_HMC5883_Unified(12345);
sensors_event_t compass_event;

Adafruit_MotorShield AFMS = Adafruit_MotorShield();

// Проводимо налаштування контролерів двигуна як для приводу, так і для
рульового управління (повороту).
```

									Арк.
									29
Зм.	Арк.	№ докум	Підпис	Дата					

PI-81.466396.001

```
Adafruit_StepperMotor *turnMotor = AFMS.getStepper(200, 1); // I used a NEMA 17 stepper from my 3D printer, which is 200 steps/revolution
```

```
Adafruit_DCMotor *driveMotor = AFMS.getMotor(3);
```

```
#define TURN_LEFT -10
```

```
#define TURN_RIGHT 10
```

```
#define TURN_STRAIGHT 0
```

```
// Обгортка для серводвигунів
```

```
void forwardstep1() {
```

```
turnMotor->onestep(FORWARD, DOUBLE);
```

```
}
```

```
void backwardstep1() {
```

```
turnMotor->onestep(BACKWARD, DOUBLE);
```

```
}
```

```
AccelStepper stepper1(forwardstep1, backwardstep1);
```

```
// LCD Display
```

```
LiquidCrystal_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE); //
```

```
Встановлення адреси та розміру LCD
```

```
#define LEFT_ARROW 0x7F
```

```
#define RIGHT_ARROW 0x7E
```

```
#define DEGREE_SYMBOL 0xDF
```

```
//Проводимо алаштування ультразвукового дальнометра
```

```
#define TRIGGER_PIN 11
```

					PI-81.466396.001	Арк.
						30
Зм.	Арк.	№ докум	Підпис	Дата		

```
#define ECHO_PIN 11

#define MAX_DISTANCE_CM 250 //Максимальна дистанція -500см.
#define MAX_DISTANCE_IN (MAX_DISTANCE_CM / 2.5) // В дюймах
int sonarDistance;

NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE_CM);
```

```
MovingAverage sonarAverage(MAX_DISTANCE_IN);
```

```
// Проводимо навігацію компасу(макгітомеру)
```

```
int targetHeading; // Куди потрібно повернути, “щоб дійти” до поточного маршруту
```

```
int currentHeading; // Де ми фактично зараз знаходимось
```

```
int headingError;
```

```
#define HEADING_TOLERANCE 5
```

```
// GPS Navigation
```

```
#define GPSECHO false // не забути встановити TRUE для коректировки, якщо потрібно!
```

```
SoftwareSerial mySerial(8, 7);
```

```
Adafruit_GPS GPS(&mySerial);
```

```
boolean usingInterrupt = false;
```

```
float currentLat,
```

```
currentLong,
```

```
targetLat,
```

```
targetLong;
```

```
int distanceToTarget, // Поточна відстань до цілі
```

```
originalDistanceToTarget; // Відстань до маршруту, коли ми починаємо до
```

НЬОГО ВИХОДИТЬ

									Арк.
									31
Зм.	Арк.	№ докум	Підпис	Дата					

PI-81.466396.001

```

// Перевіряємо точки в маршруті
    #define WAYPOINT_DIST_TOLERANCE 5 // Відстань в метрах до шляху,
після підходу до маршруту продовження руху
    #define NUMBER_WAYPOINTS 5 // Введення кількості точок шляху
    int waypointNumber = -1; // З якої точки початок, 0 це перша точка, -1 це
поточна точка
    waypointClass waypointList[NUMBER_WAYPOINTS] =
{waypointClass(50.353697, 30.950085), waypointClass(50.353910, 30.949654),
waypointClass(50.354120, 30.949081), waypointClass(50.354363, 30.948192),
waypointClass(50.354535, 30.947675) }; //

// Повороти
enum directions {left = TURN_LEFT, right = TURN_RIGHT, straight =
TURN_STRAIGHT} ;
directions turnDirection = straight;

// Відстань уникнення об'єкта
    #define SAFE_DISTANCE 70
    #define TURN_DISTANCE 40
    #define STOP_DISTANCE 12

// швидкість (range: 0 - 255)
    #define FAST_SPEED 150
    #define NORMAL_SPEED 125
    #define TURN_SPEED 100
    #define SLOW_SPEED 75

```

					PI-81.466396.001	Арк.
Зм.	Арк.	№ докум	Підпис	Дата		32

```
int speed = NORMAL_SPEED;
```

```
// IR приймач
```

```
#ifdef USE_IR
```

```
#include "IRremote.h"
```

```
#define IR_PIN 5
```

```
IRrecv IR_receiver(IR_PIN);
```

```
decode_results IR_results;
```

```
#endif
```

```
#define IR_CODE_FORWARD 0x511DBB
```

```
#define IR_CODE_LEFT 0x52A3D41F
```

```
#define IR_CODE_OK 0xD7E84B1B
```

```
#define IR_CODE_RIGHT 0x20FE4DBB
```

```
#define IR_CODE_REVERSE 0xA3C8EDDB
```

```
#define IR_CODE_1 0xC101E57B
```

```
#define IR_CODE_2 0x97483BFB
```

```
#define IR_CODE_3 0xF0C41643
```

```
#define IR_CODE_4 0x9716BE3F
```

```
#define IR_CODE_5 0x3D9AE3F7
```

```
#define IR_CODE_6 0x6182021B
```

```
#define IR_CODE_7 0x8C22657B
```

```
#define IR_CODE_8 0x488F3CBB
```

```
#define IR_CODE_9 0x449E79F
```

					Арк.
					33
Зм.	Арк.	№ докум	Підпис	Дата	

PI-81.466396.001


```
#define IR_CODE_STAR 0x32C6FDF7
#define IR_CODE_0 0x1BC0157B
#define IR_CODE_HASHTAG 0x3EC3FC1B
```

```
// Оновлення GPS
```

```
SIGNAL(TIMER0_COMPA_vect)
{
GPS.read();
}
```

```
void useInterrupt(boolean v)
{
if (v) {
```

```
// Порівня координи
```

```
OCR0A = 0xAF;
TIMSK0 |= _BV(OCIE0A);
usingInterrupt = true;
} else {
```

```
// Вимкнути порівня
```

```
TIMSK0 &= ~_BV(OCIE0A);
usingInterrupt = false;
}
}
```

```
void setup()
{
```

											Арк.
											34
Зм.	Арк.	№ докум	Підпис	Дата	PI-81.466396.001						

```
Serial.begin(115200);
```

```
// Запуск дисплею
```

```
lcd.begin(20, 4);
```

```
#ifdef USE_LCD_BACKLIGHT
```

```
lcd.backlight();
```

```
#else
```

```
lcd.noBacklight();
```

```
#endif
```

```
lcd.clear();
```

```
#ifdef USE_GRAPHING
```

```
createLCDChars();
```

```
#endif
```

```
// старт двигунів
```

```
AFMS.begin();
```

```
driveMotor->setSpeed(NORMAL_SPEED);
```

```
// turnMotor->setSpeed(255); // максимально швидкість, можливі порушення  
маршруту
```

```
stepper1.setMaxSpeed(100.0);
```

```
stepper1.setAcceleration(100.0);
```

```
stepper1.moveTo(0);
```

Зм.	Арк.	№ докум	Підпис	Дата

PI-81.466396.001

Арк.

35

```

// запуск компасу
    if(!compass.begin())
    {
        #ifdef DEBUG
        Serial.println(F("COMPASS ERROR"));
        #endif
        lcd.print(F("COMPASS ERROR"));
        loopForever();
    }

//запустити GPS і встановити конфігурацію
    GPS.begin(9600);
    GPS.sendCommand(PMTK_SET_NMEA_OUTPUT_RMCGGA);
    GPS.sendCommand(PMTK_SET_NMEA_UPDATE_1HZ);
    GPS.sendCommand(PGCMD_NOANTENNA);
    useInterrupt(true);
    delay(1000);

// Очікування сигналу
    #ifndef NO_GPS_WAIT
    lcd.setCursor(0, 0);
    lcd.print(F("Autonomous R/C car"));
    lcd.setCursor(0, 1);
    lcd.print(F("Built by Darrell Lee"));
    unsigned long startTime = millis();
    while (!GPS.fix) // Очікування поновлення сигналу
    {

```

										Арк.
										36
Зм.	Арк.	№ докум	Підпис	Дата	PI-81.466396.001					

```
lcd.setCursor(0, 2);  
lcd.print(F("Waiting for GPS: "));  
lcd.print((int) (millis() - startTime) / 1000); // Час без сигналу  
if (GPS.newNMEAreceived())  
GPS.parse(GPS.lastNMEA());  
}
```

```
#endif
```

```
#ifdef USE_IR
```

```
IR_receiver.enableIRIn();
```

```
//Очікування натиску клавіші для початку руху
```

```
lcd.setCursor(0, 3);
```

```
lcd.print(F("Press key to start"));
```

```
while(!IR_receiver.decode(&IR_results)) ; // Очікування
```

```
IR_receiver.resume();
```

```
#else
```

```
//за відсутності кнопки, поставити зворотній відлік
```

```
lcd.clear();
```

```
lcd.print(F("GPS Acquired"));
```

```
lcd.setCursor(0, 1);
```

```
lcd.print(F("Starting in..."));
```

```
lcd.setCursor(0, 2);
```

```
for (int i = 10; i > 0; i--)
```

```
{
```

									Арк.
									37
Зм.	Арк.	№ докум	Підпис	Дата					

PI-81.466396.001

```
lcd.print(i);  
lcd.print(F(" "));  
if (GPS.newNMEAreceived())  
GPS.parse(GPS.lastNMEA());  
delay(500);  
}  
#endif
```

```
// Отримання посаткової точки старту
```

```
nextWaypoint();  
  
}
```

```
void loop()  
{
```

```
#ifdef USE_IR  
checkKillSwitch();  
#endif
```

```
// Оброблення GPS
```

```
if (GPS.newNMEAreceived()) // Оновлення інформації про GPS  
{  
if(GPS.parse(GPS.lastNMEA()) )  
processGPS();  
}
```

					PI-81.466396.001	Арк.
						38
Зм.	Арк.	№ докум	Підпис	Дата		

```

// Навігація

currentHeading = readCompass();
calcDesiredTurn(); // Оптимальний маршрут без перешкод

// Рух між перешкод
checkSonar();
moveAndAvoid();
stepper1.run();

// Оновлення показання монітору
updateDisplay();

}

//

// Після оновлення даних GPS оновлює положення та курс
void processGPS(void)
{
currentLat = convertDegMinToDecDeg(GPS.latitude);
currentLong = convertDegMinToDecDeg(GPS.longitude);

if (GPS.lat == 'S') // make them signed
currentLat = -currentLat;
if (GPS.lon == 'W')
currentLong = -currentLong;

// Оновити курс та відстань до точки виходячи з нашого нового положення
distanceToWaypoint();
courseToWaypoint();

```

					PI-81.466396.001	Арк.
Зм.	Арк.	№ докум	Підпис	Дата		39

```

    }

    void checkSonar(void)
    {

int dist;

        dist = sonar.ping_in(); // Відстань від датчика
        if (dist == 0) //Якщо занадто далеко для вимірювання, повернути
максимальну відстань
        dist = MAX_DISTANCE_IN;
        sonarDistance = sonarAverage.add(dist);
    }

    int readCompass(void)
    {
        compass.getEvent(&compass_event);
        float heading = atan2(compass_event.magnetic.y,
compass_event.magnetic.x);

//!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

        #define DEC_ANGLE 0.2382
        heading += DEC_ANGLE;

        if(heading < 0)
            heading += 2*PI;

```

					PI-81.466396.001	Арк.
Зм.	Арк.	№ докум	Підпис	Дата		40

```

if(heading > 2*PI)
heading -= 2*PI;

float headingDegrees = heading * 180/M_PI;

return ((int)headingDegrees);
}

void calcDesiredTurn(void)
{
// Обчислення кута поворота, для направлення до місця призначення
headingError = targetHeading - currentHeading;

// Налаштування обертання компаса
if (headingError < -180)
headingError += 360;
if (headingError > 180)
headingError -= 360;

// Обчислення, в який бік повернути щоб стати на правильну дорогу
if (abs(headingError) <= HEADING_TOLERANCE) // Якщо похибка
мінімальна, не звертати
turnDirection = straight;
else if (headingError < 0)
turnDirection = left;
else if (headingError > 0)
turnDirection = right;
else
turnDirection = straight;

```

Зм.	Арк.	№ докум	Підпис	Дата

PI-81.466396.001

Арк.

41


```

}

void moveAndAvoid(void)
{

if (sonarDistance >= SAFE_DISTANCE)
{
if (turnDirection == straight)
speed = FAST_SPEED;
else
speed = TURN_SPEED;
driveMotor->setSpeed(speed);
driveMotor->run(FORWARD);

stepper1.moveTo(turnDirection);
return;
}

if (sonarDistance > TURN_DISTANCE && sonarDistance <
SAFE_DISTANCE) // Сповільнення перед поворотом
{

if (turnDirection == straight)
speed = NORMAL_SPEED;
else
{
speed = TURN_SPEED;
stepper1.moveTo(turnDirection);

```

Зм.	Арк.	№ докум	Підпис	Дата

PI-81.466396.001

Арк.

42

```

}
driveMotor->setSpeed(speed);
driveMotor->run(FORWARD);
return;
}

```

```

if (sonarDistance < TURN_DISTANCE && sonarDistance > STOP_DISTANCE)

```

```

// Час повертати, щоб уникнути перепони

```

```

{
speed = SLOW_SPEED;
driveMotor->setSpeed(speed); // slow down
driveMotor->run(FORWARD);
switch (turnDirection)
{
case straight:
{
if (headingError <= 0)
turnDirection = left;
else
turnDirection = right;
stepper1.moveTo(turnDirection);
break;
}
case left:
{
stepper1.moveTo(TURN_RIGHT);

break;
}
}
}

```

```

    case right:
    {

stepper1.moveTo(TURN_LEFT);

    break;
    }
}

return;
}

    if (sonarDistance < STOP_DISTANCE) // Зупинка ізза сильного зближення
з об'єктом
    {
    driveMotor->run(RELEASE);
    stepper1.moveTo(TURN_STRAIGHT);

    turnDirection = straight;
    driveMotor->setSpeed(NORMAL_SPEED); // Під'їхати назад
    driveMotor->run(BACKWARD);
    while (sonarDistance < TURN_DISTANCE)
    {
    if(GPS.parse(GPS.lastNMEA() )
    processGPS();
    currentHeading = readCompass();
    calcDesiredTurn();
    checkSonar();
    updateDisplay();

```

										Арк.
										44
Зм.	Арк.	№ докум	Підпис	Дата	PI-81.466396.001					

```

delay(100);
}
driveMotor->run(RELEASE);
return;
}
}

```

```

void nextWaypoint(void)

```

```

{
waypointNumber++;
targetLat = waypointList[waypointNumber].getLat();
targetLong = waypointList[waypointNumber].getLong();

if ((targetLat == 0 && targetLong == 0) || waypointNumber >=
NUMBER_WAYPOINTS) // last waypoint reached?
{
driveMotor->run(RELEASE); // make sure we stop
stepper1.moveTo(TURN_STRAIGHT);

lcd.clear();
lcd.println(F("* LAST WAYPOINT *"));
loopForever();
}

processGPS();
distanceToTarget = originalDistanceToTarget = distanceToWaypoint();

```

					Арк.
					45
Зм.	Арк.	№ докум	Підпис	Дата	

PI-81.466396.001

```

courseToWaypoint();

}

int distanceToWaypoint()
{

float delta = radians(currentLong - targetLong);
float sdlong = sin(delta);
float cdlong = cos(delta);
float lat1 = radians(currentLat);
float lat2 = radians(targetLat);
float slat1 = sin(lat1);
float clat1 = cos(lat1);
float slat2 = sin(lat2);
float clat2 = cos(lat2);

delta = (clat1 * slat2) - (slat1 * clat2 * cdlong);
delta = sq(delta);
delta += sq(clat2 * sdlong);
delta = sqrt(delta);
float denom = (slat1 * slat2) + (clat1 * clat2 * cdlong);
delta = atan2(delta, denom);
distanceToTarget = delta * 6372795;

//Перевірка, чи досягнуто поточної точки
if (distanceToTarget <= WAYPOINT_DIST_TOLERANE)
nextWaypoint();

return distanceToTarget;

```

Зм.	Арк.	№ докум	Підпис	Дата

PI-81.466396.001

Арк.

46

```
}
```

```
int courseToWaypoint()  
{  
float dlon = radians(targetLong-currentLong);  
float cLat = radians(currentLat);  
float tLat = radians(targetLat);  
float a1 = sin(dlon) * cos(tLat);  
float a2 = sin(cLat) * cos(tLat) * cos(dlon);  
a2 = cos(cLat) * sin(tLat) - a2;  
a2 = atan2(a1, a2);  
if (a2 < 0.0)  
{  
a2 += TWO_PI;  
}  
targetHeading = degrees(a2);  
return targetHeading;  
}
```

```
double convertDegMinToDecDeg (float degMin)  
{  
double min = 0.0;
```

```
double decDeg = 0.0;
```

```
min = fmod((double)degMin, 100.0);
```

									Арк.
									47
Зм.	Арк.	№ докум	Підпис	Дата				PI-81.466396.001	

```
degMin = (int) ( degMin / 100 );
```

```
decDeg = degMin + ( min / 60 );
```

```
return decDeg;
```

```
}
```

```
void updateDisplay(void)
```

```
{
```

```
static unsigned long lastUpdate = millis();
```

```
unsigned long currentTime;
```

```
currentTime = millis();
```

```
if (lastUpdate > currentTime)
```

```
lastUpdate = currentTime;
```

```
if (currentTime >= lastUpdate + 500 )
```

```
{
```

```
lastUpdate = currentTime;
```

```
lcd.clear();
```

```
lcd.print(F("tH= "));
```

```
lcd.print(targetHeading, DEC);
```

```
lcd.write(DEGREE_SYMBOL);
```

```
lcd.print(F(" cH= "));
```

```

lcd.print(currentHeading, DEC);
lcd.write(DEGREE_SYMBOL);
lcd.print(turnDirection);
lcd.setCursor(0, 1);
lcd.print(F("Err "));
if (headingError < 0)
lcd.write(LEFT_ARROW);
lcd.print(abs(headingError), DEC);
if (headingError > 0)
lcd.write(RIGHT_ARROW);
lcd.print(F(" Dist "));

lcd.print(distanceToTarget, DEC);
lcd.print(F("m "));
#ifdef USE_GRAPHING
lcd.write(map(distanceToTarget, 0, originalDistanceToTarget, 0, 7));
#endif
lcd.setCursor(0, 2);
lcd.print(F("Snr "));
lcd.print(sonarDistance, DEC);
#ifdef USE_GRAPHING
lcd.write(map(sonarDistance, 0, MAX_DISTANCE_IN, 0, 7));
#endif
lcd.print(F(" Spd "));
lcd.print(speed, DEC);
#ifdef USE_GRAPHING
lcd.write(map(speed, 0, 255, 0, 7));

```

					PI-81.466396.001	Арк.
Зм.	Арк.	№ докум	Підпис	Дата		49


```
#endif  
lcd.setCursor(0, 3);  
lcd.print(F("Mem "));  
lcd.print(freeRam(), DEC);  
lcd.print(F(" WPT "));  
lcd.print(waypointNumber + 1, DEC);  
lcd.print(F(" OF "));  
lcd.print(NUMBER_WAYPOINTS - 1, DEC);
```

```
#ifdef DEBUG  
Serial.print(F("LAT = "));  
Serial.print(currentLat);  
Serial.print(F(" LON = "));  
Serial.println(currentLong);  
Serial.print(F(" Dist "));  
Serial.print(distanceToWaypoint());  
Serial.print(F("Original Dist "));  
Serial.println(originalDistanceToTarget);
```

```
Serial.print(F("Compass Heading "));  
Serial.println(currentHeading);  
Serial.print(F("GPS Heading "));  
Serial.println(GPS.angle);
```

```
#endif
```

```
}
```

```
}
```

									Арк.
									50
Зм.	Арк.	№ докум	Підпис	Дата					

PI-81.466396.001

```

int freeRam ()
{
extern int __heap_start, *__brkval;
int v;
return (int) &v - (__brkval == 0 ? (int) &__heap_start : (int) __brkval);
}
void loopForever(void)
{
while (1)
;
}

```

```

#ifdef USE_GRAPHING
void createLCDChars(void)
{
int lvl = 0;
byte arry[8];
for (int a = 7; a >= 0; a--)
{
for (int b = 0; b <= 7; b++)
{
if (b >= lvl)
arry[b] = B11111;
else

```

```

arry[b] = B10001;
}

```

```

lcd.createChar(a, arry);
lv1++;
}}

#endif

#ifdef USE_IR

void checkKillSwitch(void)
{
if(IR_receiver.decode(&IR_results))
{
stepper1.moveTo(TURN_STRAIGHT);

driveMotor->run(RELEASE);

lcd.clear();
lcd.print(F("Press to resume"));
delay(1000);

IR_receiver.resume();
while(!IR_receiver.decode(&IR_results)) ;
IR_receiver.resume();
}
}

#endif

```

Зм.	Арк.	№ докум	Підпис	Дата

PI-81.466396.001

Арк.

52