

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Тернопільський національний економічний університет
Факультет комп'ютерних інформаційних технологій
Кафедра комп'ютерної інженерії

Сомонюк Віталій Юрійович

**Система управління робототехнічною платформою
на основі радіомодуля та акселерометра / Robotic
platform management system based on radio module
and accelerometer**

Спеціальність: 123 – Комп'ютерна інженерія
Освітньо-професійна програма – Комп'ютерна інженерія

Випускна кваліфікаційна робота

Виконав: студент групи КСМ-43/2
Сомонюк Віталій Юрійович

Науковий керівник
Ігнатєв І.В.

ТЕРНОПІЛЬ -2019

РЕЗЮМЕ

Дипломний проект містить 61 сторінок пояснюючої записки, 17 рисунків, 8 таблиць, 1 додаток. Обсяг графічного матеріалу 2 аркуші формату А3.

Метою дипломного проекту є розроблення системи управління робототехнічною платформою.

В ході випускної кваліфікаційної роботи був проведений аналіз основних аспектів і проблем дослідження алгоритмів керування систем пересування робототехнічної платформи. Було розроблено робототехнічну платформу на основі мікроконтролера Arduino UNO R3 з використанням акселерометра MPU 6050 та з виростанням безпроводних передавачів, що дозволяє керувати робототехнічною платформою на відстані до 200 м.

Розробки в цій роботі можуть бути застосовані, як в практичних цілях, так і для отримання теоретичних матеріалів для досліджень. Робота має велику практичну цінність. Згідно з таблицею 4.6 бачимо, що розроблюваний програмно – апаратний засіб є більш економічно ефективний за наявні аналоги. Дане порівняння показує, що вказаний програмно – апаратний продукт має переваги в порівнянні з аналогами, зокрема: надійність, простота використання, ергономічність, зручність. Згідно проведеного економічного обґрунтування зазначений програмно – апаратний засіб є конкурентоздатним і має право існування на ринку. Крім того, отримано економічний ефект у розмірі 40545,77 грн. і тому розробка і впровадження цього програмно – апаратного засобу є економічно доцільною.

Ключові слова: РОБОТОТЕХНІЧНА ПЛАТФОРМА, АКСЕЛЕРОМЕТР, ARDUINO

RESUME

The diploma project contains 61 pages of explanatory note, 17 figures, 8 tables, 1 appendix. Volume of graphic material 2 sheets of A3 format.

The purpose of the diploma project is to develop a control system for a robotic platform.

During the final qualifying work the analysis of the basic aspects and problems of research of algorithms of management of systems of movement of a robotic platform was carried out. A robotic platform based on the Arduino UNO R3 microcontroller was developed using the MPU 6050 accelerometer and with the growth of wireless transmitters, which allows to control the robotic platform at a distance of up to 200 m.

Developments in this work can be used both for practical purposes and to obtain theoretical materials for research. The work has great practical value. According to Table 4.6, we see that the developed software and hardware is more cost-effective than existing counterparts. This comparison shows that this software and hardware product has advantages over analogues, in particular: reliability, ease of use, ergonomics, convenience. According to the conducted economic substantiation the specified software - hardware is competitive and has the right of existence in the market. In addition, an economic effect of UAH 40,545.77 was obtained. and therefore the development and implementation of this software and hardware is economically feasible.

Keywords: ROBOTIC PLATFORM, ACCELEROMETER, ADRUINO

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

ПАЗ – програмно-апаратне забезпечення.

ОЗП – оперативний запам'ятовуючий пристрій.

ПЗП – постійна запам'ятовуюча пам'ять.

ФНЧ – фільтр низьких частот.

SPI – послідовний периферійний інтерфейс.

					БР.КСМ.07180/17.00.00.000.ПЗ	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

На сьогоднішній день роботи все частіше беруть участь в нашому повсякденному житті. В результаті прискороного розвитку технологій, в останні кілька років, використання робототехнічних комплексів зростає експоненціально.

Роботи використовуються в самих різних областях, в автомобільній промисловості, в медичній промисловості роботи, де роботи можуть виконувати хірургічні операції, для уникнення появи людського фактору, у військовій промисловості, для видалення наземних мін, знешкодження бомб, ліквідація небезпечних техногенних зон, в космічній промисловості, в побуті, для прибирання приміщень.

Існують роботи, що вимагають особливої уваги - це автономні мобільні роботи. Даний тип роботів створений для вирішення завдань навігації, планування, локалізації.

Багато дослідні інститути, університети і промисловість вкладують багато ресурсів і часу на досліджень в цій області. Розробка мобільних роботів є складним завданням, пов'язаної з необхідними ресурсами і навичками в багатьох областях, такі як розробка програмного забезпечення, штучний інтелект, механічне проектування, електротехніка, сигнал обробки сенсорної техніки або теорії управління. Сучасні мобільні робототехнічні системи часто є монолітними.

В даний час велика увага приділяється розробці робототехнічних комплексів з високою пристосованістю до руху по складних траєкторіях і складних поверхнях.

У даній роботі найбільшу увагу було приділено робототехнічним комплексам, що використовують колесний спосіб пересування.

					БР.КСМ.07180/17.00.00.000.ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

1 ПОНЯТТЯ ПРОГРАМНО-АПАРАТНОГО ЗАСОБУ

1.1 Апаратне забезпечення

Комп'ютерна система складається з двох основних елементів: апаратного і програмного забезпечення. Апаратне забезпечення - це частини, які можна фізично торкнутися. Програмне забезпечення являє собою набір інструкцій для комп'ютера для виконання конкретних операцій. Нам потрібні як апаратне, так і програмне забезпечення для роботи ПАЗ. По суті справи, персональний комп'ютер являє собою такий собі електронний конструктор, який користувач може збирати самостійно. І від того, які елементи він буде використовувати - в кінцевому рахунку буде залежати його потужність і функціональні можливості. Для того щоб користувач міг відразу ж приступити до роботи на ПК він повинен мати наступний, мінімальний набір компонентів:

1. Системний блок - не є якоюсь однією, окремою частиною персонального комп'ютера, що виконує одну-єдину функцію, як, наприклад, клавіатура або миша (введення даних і управління). Системний блок (див. рисунок 1.1) містить цілий набір пристроїв, кожне з яких виконує ряд певних дій.



Рисунок 1.1 – Вигляд ситемного блоку

					БР.КСМ.07180/17.00.00.000.ПЗ	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дата		

2. Материнська плата (див. рисунок 1.2) – електронний пристрій, який є центром всієї системи. До материнської плати за допомогою спеціальних роз'ємів підключаються інші компоненти системного блоку і зовнішньої периферії.

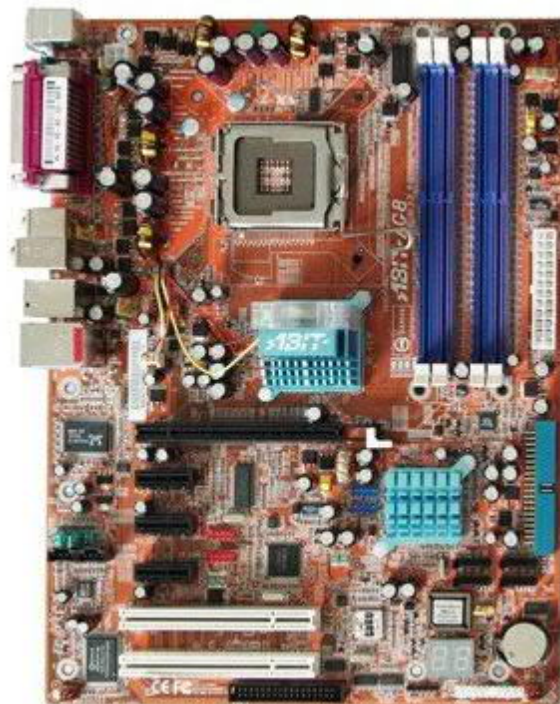


Рисунок 1.2 – Вигляд материнської плати

3. Центральний процесор – є мозком цифрової системи. Саме він, виконуючи команди програмного забезпечення (програм, завантажених в комп'ютер), змушує працювати материнську плату і виконувати різні дії - зчитувати дані з клавіатури, виводити зображення на монітор, друкувати на принтері і т.д. Фізично мікропроцесор (див. рисунок 1.3) являє собою інтегральну схему - тонку пластинку кристалічного кремнію прямокутної форми площею всього кілька квадратних міліметрів, на якій розміщені схеми, що реалізують всі функції процесора. Кристал-пластинка зазвичай міститься в пластмасовому або керамічний плоский корпус і з'єднується золотими провідками з металевими штирями, щоб його можна було приєднати до материнської плати комп'ютера.

Для персональних комп'ютерів найбільш широко використовуються процесори Intel. Додаткову інформацію про них можна прочитати на сайтах виробників процесорів:

- Intel – <http://www.intel.com/>
- AMD – <http://www.amd.com/>

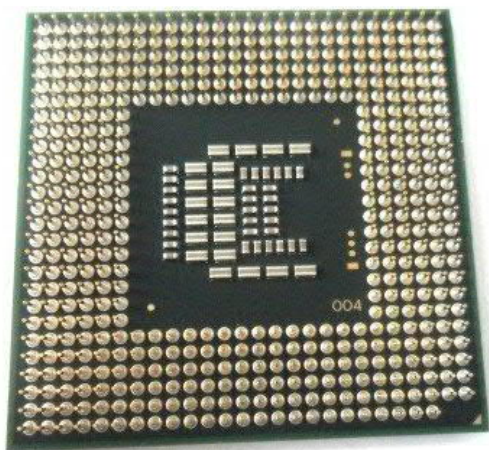


Рисунок 1.3 – Вигляд центрального процесора

4. Оперативна пам'ять (див. рисунок 1.4) – електронний компонент, що дозволяє процесору зберігати необхідні дані під час своєї роботи. Пам'ять містить дані, поки до неї підводиться живлення. Коли комп'ютер вимкнений - весь вміст у пам'яті стирається.



Рисунок 1.4 – Вигляд оперативної пам'яті

У разі якщо є необхідність зберігання даних при вимкненому живленні, існує інший вид пам'яті званий ПЗП. Така пам'ять, якщо в неї записати дані,

					БР.КСМ.07180/17.00.00.000.ПЗ	Арк.
						12
Змн.	Арк.	№ докум.	Підпис	Дата		

буде зберігати їх незалежно від того подається на неї живлення чи ні.

5. Накопичувач на жорсткому диску (див. рисунок 1.5) – пристрій призначений для запису і читання різних призначених для користувача даних - програм, фотографій, відео, текстів і т.д., Випускається різної ємності - від декількох гігабайт і до декількох терабайт. Використовується в системному блоці в якості основного сховища всіх даних, так як має високу швидкість читання-запису (набагато вище, ніж у пристроїв для роботи з оптичними дисками і флеш - накопичувачами). Крім внутрішніх жорстких дисків є і зовнішні (переносні) жорсткі диски , які можна переставляти з машини на машину, використовуючи спеціальні бокси або перехідники. Зовнішні жорсткі диски мають власний корпус і джерело живлення, що економить простір всередині корпусу комп'ютера і зменшує навантаження на блок живлення комп'ютера.



Рисунок 1.5 – Зображення накопичувача

6. Відеокарта (див. рисунок 1.6) – плата, яка встановлюється в системний блок і дозволяє здійснювати вивід зображень на монітор або сучасний LED або LCD телевізор. Від того, який обчислювальної потужності відеокарта встановлена в системний блок залежать його графічні властивості - здатність зображення, кількість відтворюваних кольорів, глибина кольору, швидкість виведення картини на екран, швидкість обробки відеозображень і ще ряд

					БР.КСМ.07180/17.00.00.000.ПЗ	Арк.
						13
Змн.	Арк.	№ докум.	Підпис	Дата		

додаткових параметрів. Сучасні відеоадаптери мають пам'ять, яка дозволяє не використовувати ОЗП комп'ютера для формування зображення. Пам'ять відеоадаптерів називається відеопам'яттю. Крім того, більшість відеоадаптерів оснащені власними графічними сопроцесорами, необхідними для обробки зображення. Такі адаптери часто називаються графічними прискорювачами.



Рисунок 1.6 – Зображення відеокарти

7. Блок живлення – пристрій забезпечує живлення всіх компонентів, що входять до складу системного блоку. Потужності цього блоку має вистачати для нормальної роботи всіх складових, в іншому випадку він швидко вийде з ладу через перевантаження.

8. Клавіатура – являє собою пристрій для введення інформації в комп'ютер. Містить набір букв і цифр відповідає стандарту друкарської машинки, а також додаткові клавіші для управління курсором і виконують різні функції в залежності від програмного забезпечення (функціональні клавіші). Розташування клавіш на клавіатурі має певну стандартизовану розкладку (порядок руху), яка називається QWERTY. Останнім часом все більше поширення набувають клавіатури, мають додаткові клавіші для керування мультимедійними функціями комп'ютера - звуком, відео, запуском програвачів, а також для швидкої навігації в інтернеті.

9. Маніпулятор миша – являє собою пристрій для управління графічним покажчиком (стрілкою) на екрані в додатках. За допомогою цього маніпулятора

					БР.КСМ.07180/17.00.00.000.ПЗ	Арк.
						14
Змн.	Арк.	№ докум.	Підпис	Дата		

можна переміщати графічні об'єкти, натискати віртуальні кнопки, виконувати певні дії. Основною перевагою «мишки» є можливість миттєво потрапити покажчиком в будь-яку точку екрану.

10. Монітор – стара назва цього пристрою - дисплей . Одна з важливих частин персонального комп'ютера. Від технічних можливостей монітора буде залежати не тільки якість виведеного на екрані зображення, але і комфортність роботи за ним.

1.2 Програмне забезпечення

Програмне забезпечення почало ставати загальним терміном у 1960-х роках, щоб відрізнити комп'ютерне обладнання від програм, що входять до нього. Алан Тюрінг (1912-1954), піонер англійського комп'ютерного вченого і математика, вперше запропонував цей термін. Комп'ютерні фахівці сьогодні називають Тьюрінга батьком теоретичної інформатики та штучного інтелекту. Джон Тьюкі (1915-2000), американський математик, вперше застосував цей термін у друку в 1958 році. Програмне забезпечення відноситься до програм, які працюють на комп'ютерах і виконують певні функції. Це весь набір програм, процедур і пов'язаної з ним документації, що мають комп'ютерні системи. Без нього комп'ютер, смартфон, ноутбук або планшет не працюватимуть. Насправді, без нього, вони були б настільки ж корисні для вас, як ручна друкарська машинка. Комп'ютерні програми в ній, частини з кодуванням та інструкціями, з іншого боку, є програмним забезпеченням.

Програмне забезпечення є змінною частиною комп'ютера, а апаратне забезпечення - незмінною частиною, тобто тією частиною, яка залишається незмінною. Зазвичай розбивають програмне забезпечення на прикладне та системне програмне забезпечення. Прикладне програмне забезпечення

					БР.КСМ.07180/17.00.00.000.ПЗ	Арк.
						15
Змн.	Арк.	№ докум.	Підпис	Дата		

відноситься до програм, які виконують типи робіт, які безпосередньо цікавлять користувача. Системне програмне забезпечення включає операційні системи та будь-яку програму, що підтримує прикладну програму. Проміжне програмне забезпечення - це програмування, що опосередковує між додатками та системами. Проміжне програмне забезпечення також включає програми, які дозволяють використовувати файли з однієї системи в іншій операційній системі. Програмне забезпечення відноситься до всіх інструкцій, що технічний пристрій - комп'ютер, смартфон і т.д. - слід. Перш ніж зберігати його на дисках, флеш-накопичувачах USB або завантажувати через Інтернет, він прийшов на "м'яких носіях". Магнітні диски, магнітні стрічки та паперові перфокарти - приклади м'яких носіїв.

Прошивка - це стиль комп'ютерної програми для конкретного обладнання. Ми, як правило, зберігаємо прошивку на флеш-пам'яті або мікросхемі ROM в електронному пристрої. З постійним розвитком нових програм і додатків, концепція категорій стала важкою темою. Немає набору категорій, про які ми всі погоджуємося. Ми класифікуємо комп'ютерне програмне забезпечення відповідно до загальних функцій, типів або сфери використання. Три широкі класифікації:

– Прикладне програмне забезпечення: комп'ютерні програми для виконання таких завдань, як обробка тексту, малювання, відтворення чи прослуховування музики. Програми для обліку та планування також є прикладами прикладного програмного забезпечення.

– Системне програмне забезпечення: це загальний термін для комп'ютерних програм, які ми використовуємо для запуску та запуску комп'ютерних систем.

– Засоби програмування: програми, які розробники використовують для створення, налагодження, підтримки або іншої підтримки інших програм і програм.

Цей термін в цілому відноситься до простих програм, які ми можемо

					БР.КСМ.07180/17.00.00.000.ПЗ	Арк.
						16
Змн.	Арк.	№ докум.	Підпис	Дата		

об'єднати для виконання завдання. Ці інструменти програмування використовуються, оскільки механік може використовувати декілька ручних інструментів для виправлення двигуна автомобіля.

1.3 Аналіз робототехнічних платформ

Робот - це автономна машина, здатна відчувати своє середовище, виконувати обчислення для прийняття рішень і виконувати дії в реальному світі.

Роботи - різноманітна група. Деякі ходять по двох, чотирьох, шести або більше ногах, а інші можуть вийти на небо. Деякі роботи допомагають лікарям робити операцію всередині вашого тіла; інші мучаться в брудних фабриках. Є роботи розміром з монети і роботів більше, ніж автомобіль.

Фактично, термін робот означає різні речі для різних людей. Навіть самі робототехники мають різні уявлення про те, що є роботом. І для більшості з нас, наукова фантастика сильно вплинула на те, як ми очікуємо, що робот виглядатиме і зможе це зробити. Нелегко визначити, що таке роботи, і їх нелегко класифікувати. Кожен робот має свої унікальні особливості, і в цілому роботи сильно відрізняються за розміром, формою і можливостями. Тим не менш, багато роботів поділяють різні функції. Розглянемо основні категорії роботів.

Дрони (див. рисунок 1.7) – також називаються безпілотними літаками, безпілотні літаки бувають різних розмірів і мають різний рівень автономії.

					БР.КСМ.07180/17.00.00.000.ПЗ	Арк.
						17
Змн.	Арк.	№ докум.	Підпис	Дата		



Рисунок 1.7 – Зображення робота дрона

Приклади включають популярну серію фантомів DJI і анафії Parrot's, а також військові системи, такі як Global Hawk, що використовуються для тривалого спостереження.

Гуманоїди (див. рисунок 1.8) – це, мабуть, тип робота, про який думають більшість людей, коли вони думають про робота. Приклади гуманоїдних роботів включають Honda Asimo, яка має механічний вигляд, а також андроїди, подібні серії Geminoid, які призначені для вигляду людей.



Рисунок 1.8 – Зображення робота гуманоїда

Роботи маніпулятори (див. рисунок 1.9) – виглядає як величезна автоматизована рука під контролем системи електрокерування. У конструкції пристроїв відсутній пневматика або гідравліка, все побудовано на електромеханіки. Принцип дії пристрою залежить від його програмного

					БР.КСМ.07180/17.00.00.000.ПЗ	Арк.
						18
Змн.	Арк.	№ докум.	Підпис	Дата		

забезпечення і оснащення, і якщо на початку свого розвитку основна мета була звільнення працівників від важкого і небезпечного виду робіт, то сьогодні спектр виконуваних завдань значно зріс.

Використання роботизованих помічників дозволяє справлятися одночасно з декількома завданнями:

- скорочення робочих площ і вивільнення фахівців (їх досвід і знання можуть бути використані на іншій ділянці);
- збільшення обсягів виробництва;
- підвищення якості продукції;
- завдяки безперервності процесу скорочується цикл виготовлення.

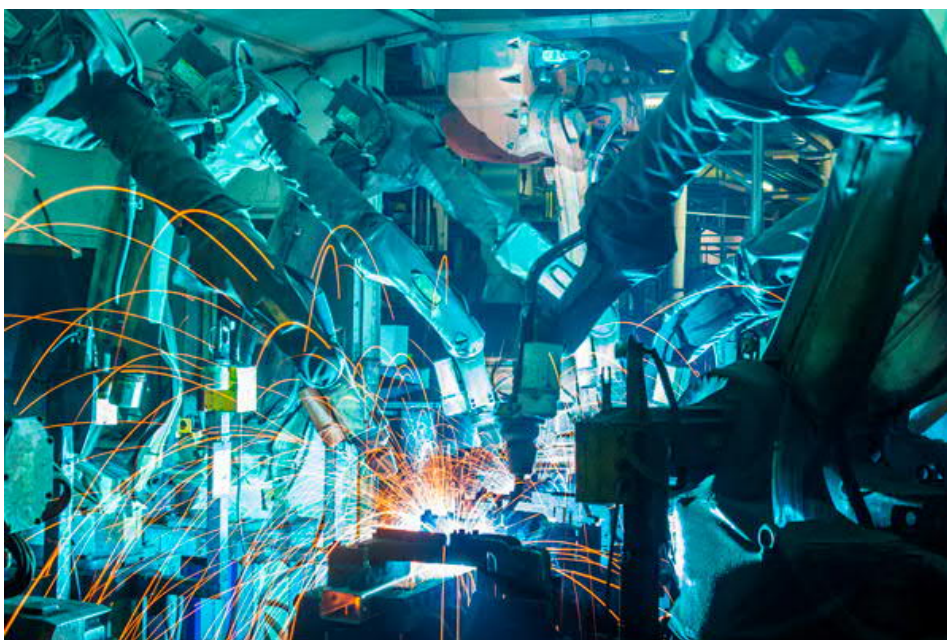


Рисунок 1.9 – Зображення робота маніпулятора

Мобільні роботи (див. рисунок 1.10) – призначені для виконання тільки транспортних операцій з перевезення вантажів (робокари), часто не мають маніпуляторів, а забезпечуються спрощеними одне – і двоступеневих вантажно-розвантажувальними пристроями, або останні операції виконуються стаціонарними маніпуляторами, що знаходяться в місцях зупинки транспортних роботів.

					БР.КСМ.07180/17.00.00.000.ПЗ	Арк.
						19
Змн.	Арк.	№ докум.	Підпис	Дата		

Мобільні роботи, що включають в себе чутливі елементи, виконавчі механізми, комп'ютери і володіють елементами штучного інтелекту, представляють досить зручний об'єкт для постановки, вивчення і знаходження рішень сучасних проблем мехатроніки. Залучення студентів і аспірантів до цих досліджень вже зараз дозволяє домогтися якісного поліпшення підготовки висококваліфікованих фахівців.

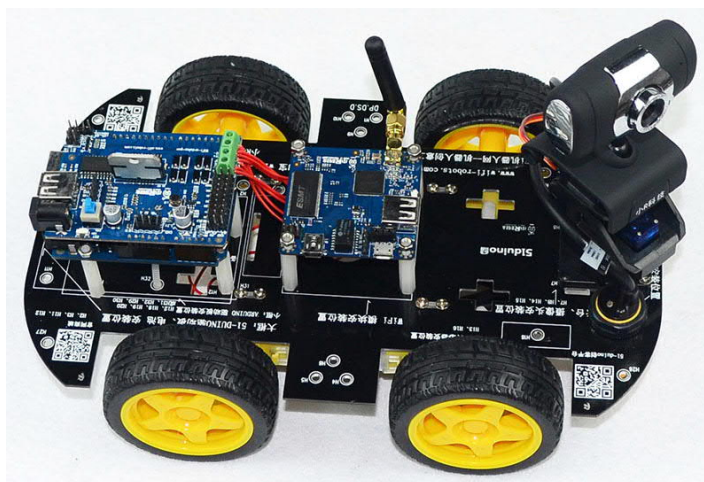


Рисунок 1.10 – Зображення мобільного робота

					БР.КСМ.07180/17.00.00.000.ПЗ	Арк.
						20
Змн.	Арк.	№ докум.	Підпис	Дата		

2 ПРОЕКТУВАННЯ СИСТЕМИ УПРАВЛІННЯ РОБОТОТЕХНІЧНОЮ ПЛАТФОРМОЮ НА ОСНОВІ АКСЕЛЕРОМЕТРА ТА МІКРОКОНТРОЛЕРА

2.1 Устаткування програмно-апаратного засобу

До устаткування програмно-апаратного засобу відносять:

- резистор;
- транзистор;
- конденсатор;
- Arduino Uno;
- Arduino Nano;
- двигун постійного струму;
- драйвер двигуна;
- сервопривод;
- акселерометр;
- безпроводний передавач.

Резистор (див. рисунок 2.1) – це електричний компонент, який обмежує або регулює потік електричного струму в електронній схемі. Резистори можуть також використовуватися для забезпечення конкретної напруги для активного пристрою, такого як транзистор . Всі інші фактори, які дорівнюють, в ланцюзі постійного струму, струм через резистор обернено пропорційний його опору , і прямо пропорційний напрузі на ньому. Це добре відомий закон Ома . У ланцюгах змінного струму це правило також застосовується до тих пір, поки резистор не містить індуктивності або ємності. Резистори можуть бути виготовлені різними способами. Найбільш поширеним типом в електронних пристроях і системах є вуглецевий резистор . Тонко дисперсний гранульований вуглець (графіт) змішується з глиною і зміцнюється. Стійкість залежить від частки вуглецю до глини; чим вище це співвідношення, тим нижче опір. Інший тип резистора виконаний з намотування ніхромового або подібного дроту на

					БР.КСМ.07180/17.00.00.000.ПЗ	Арк.
						21
Змн.	Арк.	№ докум.	Підпис	Дата		

ізолюючому вигляді. Цей компонент, званий дротяним резистором, здатний обробляти вищі струми, ніж резистор вуглецевої композиції того ж фізичного розміру. Однак, оскільки дріт намотується в котушку, компонент діє як індуктор s , а також проявляє опір. Це не впливає на продуктивність в ланцюгах постійного струму, але може мати несприятливий вплив на ланцюги змінного струму, оскільки індуктивність робить пристрій чутливим до змін частоти.



Рисунок 2.1 – Резистор з дротяними аксіальними виводами

Транзистор (див. рисунок 2.2) – це пристрій, який регулює струм або напругу і діє як перемикач або затвор для електронних сигналів. Транзистори складаються з трьох шарів напівпровідникового матеріалу, кожен з яких здатний проводити струм. Транзистор був винайдений трьома вченими в Bell Laboratories в 1947 році, і він швидко замінив вакуумну трубку як електронний регулятор сигналу. Транзистор регулює потік струму або напруги і діє як перемикач або затвор для електронних сигналів. Транзистор складається з трьох шарів напівпровідникового матеріалу, кожен з яких здатний переносити струм. Напівпровідник є матеріалом, таким як германій і кремній, який проводить електрику "напів-ентузіазмом". Він знаходиться десь між справжнім провідником, таким як мідь і ізолятор (подібно пластиковій обмотці навколо дротів).

					БР.КСМ.07180/17.00.00.000.ПЗ	Арк.
						22
Змн.	Арк.	№ докум.	Підпис	Дата		

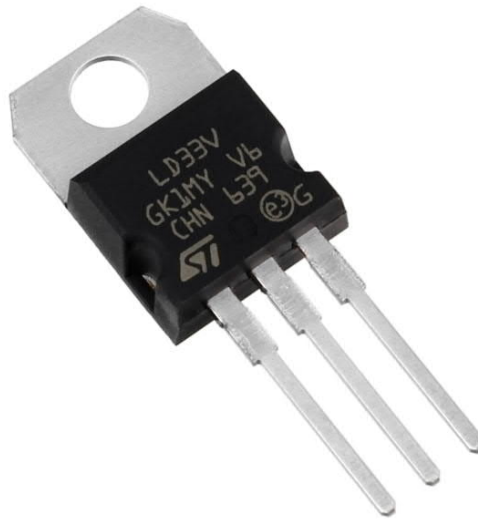


Рисунок 2.2 – Зображення транзистора LD33V

Конденсатор (див. рисунок 2.3) є пасивним електронним компонентом, який зберігає енергію у вигляді електростатичного поля. У найпростішій формі конденсатор складається з двох провідних пластин, розділених ізоляційним матеріалом, званим діелектриком. Ємність прямо пропорційна площі поверхні пластин і обернено пропорційна відстані між пластинами. Ємність також залежить від діелектричної проникності речовини, що розділяє пластини.

Стандартною одиницею ємності є фарад, скорочено. Це велика одиниця; більш загальними одиницями є мікрофарад, скорочено μF ($1 \text{ мкФ} = 10^{-6} \text{ F}$) і пікофарад, скорочено pF ($1 \text{ pF} = 10^{-12} \text{ F}$).

Конденсатори можуть бути виготовлені на інтегральних схемах. Вони зазвичай використовуються в поєднанні з транзисторами в динамічній оперативній пам'яті. Конденсатори допомагають підтримувати вміст пам'яті. Завдяки своїм мінімальним фізичним розмірам ці компоненти мають низьку ємність. Вони повинні перезаряджатися тисячі разів на секунду або втратить свої дані. Великі конденсатори використовуються в джерелах живлення електронного обладнання типу падіння, включаючи комп'ютери та їхні периферійні пристрої. У цих системах конденсатори згладжують випрямлені утиліти змінного струму.

					БР.КСМ.07180/17.00.00.000.ПЗ	Арк.
						23
Змн.	Арк.	№ докум.	Підпис	Дата		



Рисунок 2.3 – Зображення керамічного та електролітичного конденсаторів

Arduino (див. рисунок 2.4) - це платформа з відкритим кодом, яка використовується для проектів електроніки. Arduino складається як з фізичної програмованої плати (часто називається мікроконтролер), так і з програмним забезпеченням, або IDE (Integrated Development Environment), що працює на вашому комп'ютері, використовується для запису та завантаження комп'ютерного коду на фізичну плату.

Платформа Arduino стала досить популярною серед людей, які тільки починають працювати з електронікою, і не дарма. На відміну від більшості попередніх програмованих плат, Arduino не потребує окремого обладнання щоб завантажити новий код на плату - можна просто використовувати USB-кабель. Крім того, Arduino IDE використовує спрощену версію C++, що полегшує навчання програмі. Arduino надає стандартний форм-фактор, який розбиває функції мікроконтролера на більш доступний пакет.

					БР.КСМ.07180/17.00.00.000.ПЗ	Арк.
						24
Змн.	Арк.	№ докум.	Підпис	Дата		

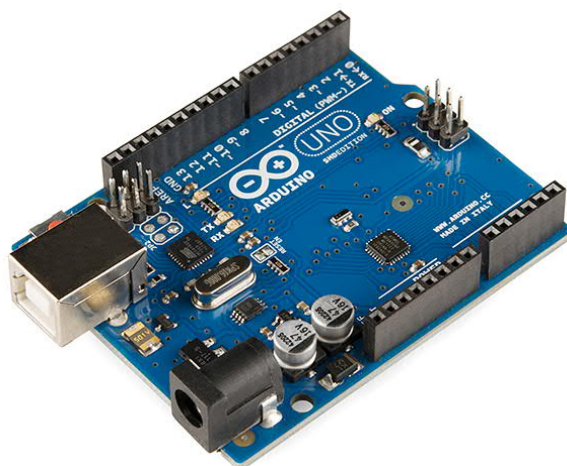


Рисунок 2.4 – Зображення Arduino Uno R3

Arduino Nano (див. рисунок 2.5) - це невелика, сумісна, гнучка плата мікроконтролера, розроблена компанією Arduino.cc в Італії, на базі ATmega328p (Arduino Nano V3.x) / Atmega168 (Arduino Nano V3.x). Він поставляється з точно такою ж функціональністю, що і в Arduino UNO, але зовсім у невеликих розмірах. Він поставляється з робочою напругою 5В, проте вхідна напруга може змінюватися від 7 до 12В. Arduino Nano Pinout містить 14 цифрових та 8 аналогових контактів.

Кожен з цих цифрових та аналогових контактів призначений для декількох функцій, але їхня головна функція повинна бути налаштована як вхідний або вихідний.

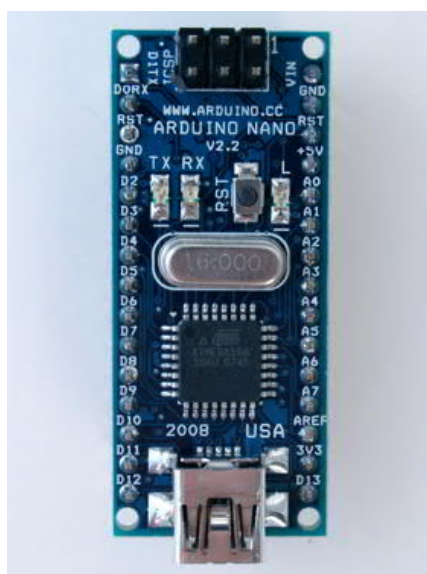


Рисунок 2.5 – Зображення Arduino Nano 2.2

					БР.КСМ.07180/17.00.00.000.ПЗ	Арк.
						25
Змн.	Арк.	№ докум.	Підпис	Дата		

Двигуни постійного струму (див. рисунок 2.6) є компонентами руху, які приймають електричну потужність у вигляді постійного струму (або деяку маніпульовану форму постійного струму) і перетворюють його в механічне обертання. Двигуни роблять це завдяки використанню магнітних полів, що виникають від електричних струмів для стимулювання обертання ротора, закріпленого на вихідному валу.

Вихідний крутний момент і швидкість обертання залежить від електричного входу і конструкції двигуна. Згідно з найпоширенішими сьогоднішніми угодами про найменування промисловості, існують три підтипи двигунів постійного струму: двигуни щіткового двигуна, двигуни постійного магнітного струму і універсальні двигуни постійного струму.

Двигун постійного струму - це електричний механізм, який перетворює електричну енергію в механічну енергію.



Рисунок 2.6 – Зображення металічного редукторного двигуна

Драйвер двигуна – пристрій, який виступає посередником між мікроконтролером, акумуляторами та двигуном. Контролер двигуна необхідний, тому що мікроконтролер може, як правило, забезпечувати лише приблизно 0,1 А, тоді як більшість приводів (двигуни постійного струму, серводвигуни тощо) потребують декількох ампер. В розробленому програмно-апаратному засобі використовуються драйвер двигуна постійного струму

					БР.КСМ.07180/17.00.00.000.ПЗ	Арк.
						26
Змн.	Арк.	№ докум.	Підпис	Дата		

L298N (див. рисунок 2.7). Даний модуль можна використовувати для двигунів, напруга живлення яких знаходиться в діапазоні від 5 до 35 вольт.

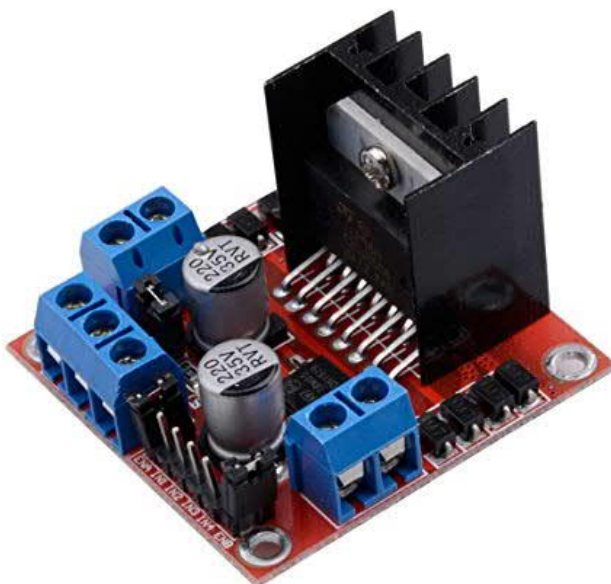


Рисунок 2.7 – Зображення драйвера двигуна L298N

Сервопривод є поворотним приводом або двигуном, що дозволяє здійснювати точний контроль з точки зору кутового положення, прискорення і швидкості, можливостей, яких не має звичайний двигун. Він використовує звичайний двигун і з'єднує його з датчиком для зворотного зв'язку з позицією. Контролер є найскладнішою частиною серводвигуна, оскільки він спеціально розроблений для цієї мети.

Сервопривод насправді не є певним класом двигунів, але є комбінацією конкретних частин, які включають в себе двигун постійного або змінного струму, і придатні для використання в замкнутій системі управління. Вони використовуються в робототехніці, автоматизованому виробництві та обробці обчислювальних машин.

Сервопривод (див. рисунок 2.8) являє собою механізм із замкненим циклом, який використовує зворотний зв'язок позиції для контролю швидкості обертання і положення. Керуючим сигналом є вхідний, або аналоговий, або цифровий, який представляє собою кінцеву команду позиції для валу. Тип датчика служить сенсором, що забезпечує зворотний зв'язок швидкості та

					БР.КСМ.07180/17.00.00.000.ПЗ	Арк.
						27
Змн.	Арк.	№ докум.	Підпис	Дата		

положення. У більшості випадків повідомляється лише про позицію. Остаточне положення повідомляється контролеру, і це порівнюється з входом вихідної позиції, а потім, якщо є невідповідність, двигун переміщується, щоб дістатися до правильного положення.



Рисунок 2.8 – Зображення сервоприводу MG995

Акселерометр - пристрій, що вимірює зміни прискорення гравітації в пристрої, в якому він може бути встановлений. Акселерометри використовуються для вимірювання прискорення, нахилу і вібрації в численних пристроях.

У стані спокою акселерометр вимірює 1 г: гравітаційне тяжіння Землі, яке реєструє 9,81 м/с або 32,185 футів в секунду. Акселерометри, які використовують п'єзоелектричний ефект, вимірюють невелику зміну напруги. Інші вимірюють потужність між двома компонентами.

Для визначення руху в декількох напрямках акселерометр повинен бути сконструйований з багато осьовими датчиками або декількома датчиками лінійної осі. Три лінійних акселерометра є достатніми для вимірювання руху в трьох вимірах. В розроблюваному програмно-апаратному засобі використовуються мікросхема MPU6050 (див. рисунок 3.9), яка містить акселерометр.

					БР.КСМ.07180/17.00.00.000.ПЗ	Арк.
						28
Змн.	Арк.	№ докум.	Підпис	Дата		

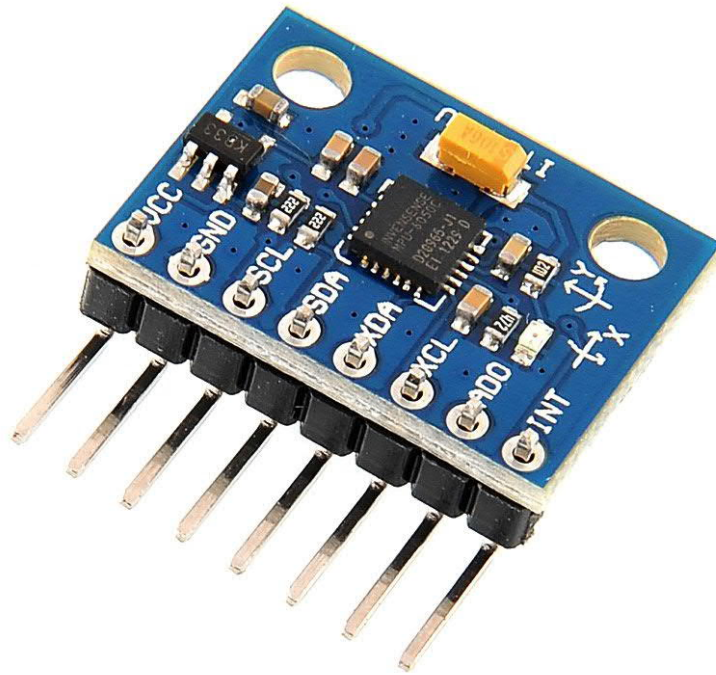


Рисунок 2.9 – Зображення мікросхеми MPU 6050

Бездротовий приймально передавальний модуль nrf24l01 (див. рисунок 2.10). nrf24l01 + є однокристальним радіоприйомним приймачем. Радіопередавачі та приймачі включають в себе генератор частоти, поліпшений регулятор режиму ShockBurst, підсилювач потужності, модулятор і демодулятор кристала. Можна вибрати канал і протокол виходу, встановивши через порт SPI. Споживання струму для nRF24L01 + вкрай низьке - в режимі передавача, коли потужність передачі 0dBm, споживання струму становить лише 11.3mA; в режимі прийому - 13,5 mA; в режимі вимкнення та вимкнення живлення споживання ще нижчі. Що стосується програми, то вона широко використовується в багатьох пристроях, таких як бездротова миша і клавіатура, ігровий пульт, набір пульта дистанційного керування, сенсор промисловості, іграшки тощо.

					БР.КСМ.07180/17.00.00.000.ПЗ	Арк.
						29
Змн.	Арк.	№ докум.	Підпис	Дата		

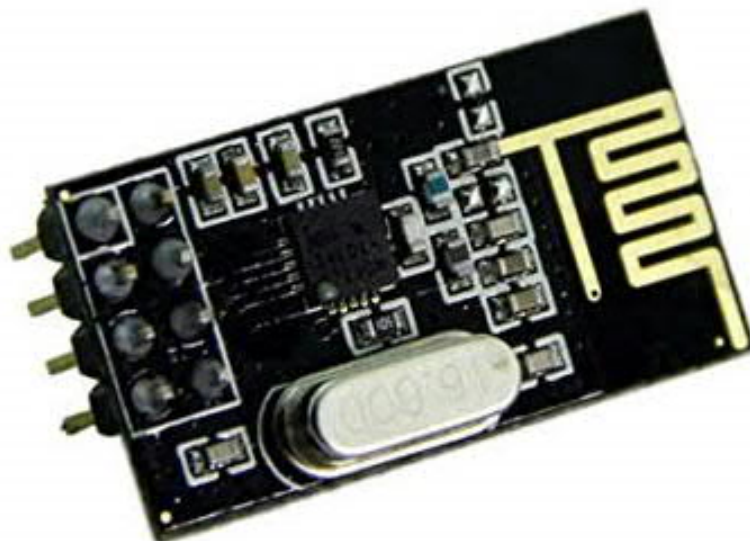


Рисунок 2.10 – Зображення Wi-Fi модуля nrf24101

За допомогою модуля можна зв'язати кілька пристроїв для передачі даних по радіоканалу. Можна об'єднати до семи приладів в одну загальну радіомережу на частоті 2,4 ГГц, один з модулів буде виступати в ролі передавача, інші – приймача.

2.2 Проектування програмно – апаратного засобу

Проектування апаратної частини системи полягає у визначенні набору модулів, що складають її, поділ їх на апаратні і програмні.

Важливе місце в процесі розробки автоматизованої системи займає виділення процесу її проектування. При проектуванні свого майбутнього програмного продукту я обрав об'єктний підхід. На рисунку 2.11 зображена схема розроблювальної апаратної частини програмно-апаратного засобу.

					БР.КСМ.07180/17.00.00.000.ПЗ	Арк.
						30
Змн.	Арк.	№ докум.	Підпис	Дата		

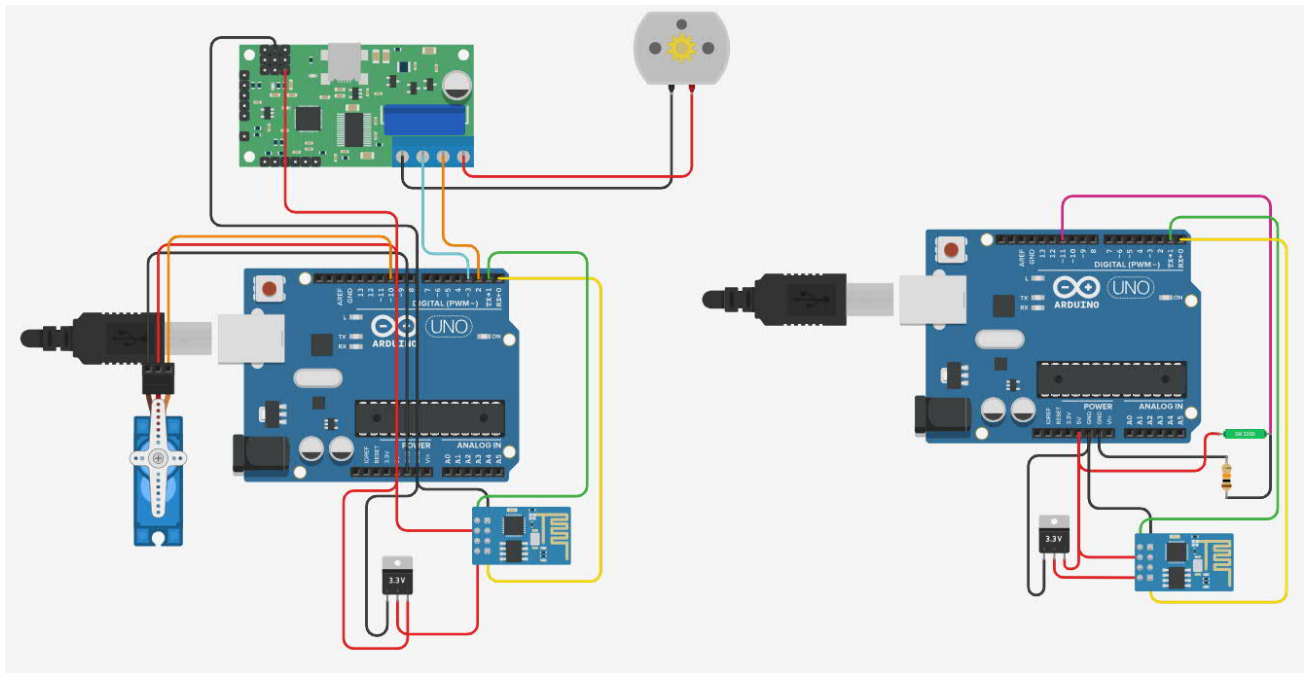


Рисунок 2.11 – Схема апаратної частини

Результатом проектування програмної частини є детальна модель розробки програмного продукту. Я обрав каскадну модель, оскільки вона найбільше підходить до вибраного мною об'єктного підходу. Каскадна модель проста і зрозуміла, хоч і не так корисна, як раніше. Проблема в тому, що вона має на увазі тільки одну роль для користувачів - специфікація вимог. Також вона має на увазі, що всі вимоги можуть бути встановлені заздалегідь. На жаль, вони ростуть і змінюються на всіх етапах, ініціюючи обробку поряд з консультаціями в кожній ітерації. Це сприяло народженню нових моделей розробки.

Переваги каскадної моделі:

- Кожна фаза має точні контрольні дані;
- Кожна фаза виконується в своєму порядку;
- Відмінно працює при розробці невеликих проектів з прозорими вимогами;
- Ілюструє афоризми ідея передувє дизайну і дизайн передувє коду.

Недоліки каскадної моделі:

- Хиткі межі проекту під час його життєвого циклу;

									Арк.
									31
Змн.	Арк.	№ докум.	Підпис	Дата	БР.КСМ.07180/17.00.00.000.ПЗ				

- Працюючий прототип відходить на задній план;
- Невизначеність з високим ступенем ризику;
- Погано підходить для складних і об'єктно-орієнтованих проектів;
- Погано підходить для проектів, розрахованих на великий період часу;
- Закладені вимоги можуть змінюватися з високим ступенем ймовірності.

Каскадна (або послідовна) модель, передбачає строго послідовне в часі і однократне виконання всіх фаз проекту з детальним попереднім плануванням і визначеними вимогами.

У процесі деталізації способів розв'язку поставленого завдання було розроблено інформаційну схему рішення задачі, де вказується джерела приймання та передачі інформації, потоки інформації, блоки перетворення даних периферійними пристроями, блоки перетворення інформації за допомогою Wi-Fi.

2.3. Розробка архітектури програмної частини

Архітектурне проектування – це перший етап процесу проектування, на якому визначаються підсистеми, а також структура управління і взаємодії підсистем. Ефективність функціонування інформаційної системи багато в чому залежить від її архітектури. Метою архітектурного проектування є опис архітектури програмного забезпечення.

Проектування архітектури програмного забезпечення – це процес розробки, що виконується після етапу аналізу і формулювання вимог. Задача такого проектування – перетворення вимог до системи у вимоги до програмного забезпечення і побудова на їхній основі архітектури системи. Побудова архітектури системи здійснюється шляхом визначення цілей системи,

					БР.КСМ.07180/17.00.00.000.ПЗ	Арк.
						32
Змн.	Арк.	№ докум.	Підпис	Дата		

її вхідних і вихідних даних, декомпозиції системи на підсистеми, компоненти або модулі та розроблення її загальної структури.

Архітектура системи, як правило, базується на одному або кількох відомих архітектурних стилях, таких як багаторівнева архітектура, розподілена архітектура, модульна архітектура, конвеєрна архітектура та інші.

Архітектура програмного забезпечення – це структура програми або обчислювальної системи, яка містить програмні компоненти, видимі зовні властивості цих компонентів, а також відносини між ними. Цей термін також відноситься до документування архітектури програмного забезпечення.

Для відображення архітектури програмного продукту потрібно розробити: функціональну схему програмного продукту, структурну схему програмного продукту, здійснити проектування баз даних, розробити та описати алгоритм функціонування програмного продукту та його підпрограм.

На основі вищеописаного матеріалу постає питання побудови структурної схеми програмного продукту. На рисунку 2.2 зображена схема, яка демонструє структуру програмних компонентів і показує взаємодію між окремими компонентами програми і фрагменти сценарію роботи.

Структурна схема показує основні проектні рішення функціональної, технічної чи організаційної структури проєктованого програмного забезпечення. На структурній схемі є відображення виробничих розділів об'єкту, що підлягають автоматизації, органів керування процесами, технічних засобів, що забезпечують реалізацію основних процесів автоматизації, що забезпечують роботу системи.

					БР.КСМ.07180/17.00.00.000.ПЗ	Арк.
						33
Змн.	Арк.	№ докум.	Підпис	Дата		

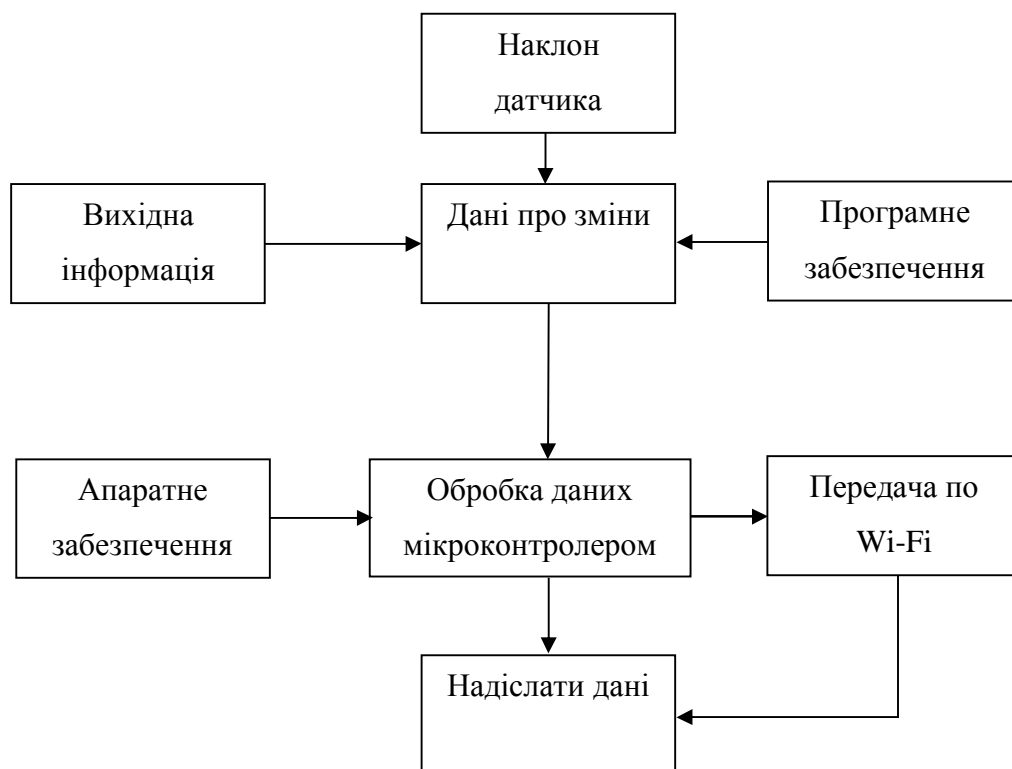


Рисунок 2.12 – Структурна схема програмного продукту

Алгоритм - це докладний ряд інструкцій для виконання операції або вирішення проблеми. У нетехнічному підході ми використовуємо алгоритми у повсякденних завданнях, такі як рецепт для випікання торта або підручник з самостійної роботи.

Технічно, комп'ютери використовують алгоритми для перерахування детальних інструкцій для виконання операції. Наприклад, для обчислення зарплати працівника комп'ютер використовує алгоритм. Для виконання цього завдання в систему необхідно ввести відповідні дані. З точки зору ефективності, різні алгоритми здатні легко і швидко виконувати операції або вирішувати проблеми. Як згадувалося раніше, алгоритми можуть бути використані в багатьох областях, і вони часто представлені у вигляді блок-схеми для візуального розуміння. Іншими словами, блок-схема - це схема, яка представляє алгоритм, що показує кроки в різних коробках і відображає процес, з'єднавши коробки разом. Ось кілька прикладів застосування алгоритмів у формах блок-схеми. Алгоритми вже є основою автоматичних технологій навчання, або машинного навчання. В даній дипломній роботі написані

алгоритми передавача та приймача. Нижче описаний алгоритм роботи передавача (Додаток А).

При вимкненому акселерометрі неможна здійснювати будь яких маніпуляцій, якщо ж він увімкнений то:

1. Ініціалізуємо акселерометр(калібрування, увімкнення датчиків);
2. Ініціалізуємо змінні val_1 та val_2(На початку роботи ці змінні створюються та запам'ятовують значення осі X та Y);
3. Визначаєм положення акселерометра відносно осі X і записуємо в val_1, аналогічно з Y - записуємо значення в val_2.
4. Коли крок 2-3 завершено відбувається відправлення сигналу на мобільну платформу, після чого крок 2-3 повторюється.

Та алгоритм роботи приймача сигналу (Додаток Б):

1. Ініціалізуємо входи виходи мікроконтролерної системи (відбувається увімкнення всіх компонентів);
2. На приймачі у нас з'явилась нова змінна msg1 – вона приймає значення val_1 та val_2, які надходять з передавача;
3. Коли сигнал msg1 дорівнює 1 (ми ж знаєм що таке двійкова система числення), тоді msg1 розуміє конкретно яка з двох змін передає 1 і задіює певні компоненти, які відповідають за рух платформи:
 - Коли кут нахилу ≥ 0 або ≤ 50 – рухаємось назад;
 - Коли кут нахилу ≥ 76 або ≤ 126 – рухаємось вперед;
 - Коли кут нахилу ≥ 127 або ≤ 175 – рухаємось праворуч;
 - Коли кут нахилу ≥ 215 або ≤ 255 – рухаємось ліворуч.
4. Процес у пункті 3 повторюється до тих пір коли поступає сигнал.

					БР.КСМ.07180/17.00.00.000.ПЗ	Арк.
						35
Змн.	Арк.	№ докум.	Підпис	Дата		

2.4 Аналіз інструментальних засобів розробки

Для розробки програмного продукту я буду використовувати середовище програмування Arduino IDE. Arduino IDE складається з двох речей - Arduino і IDE. Arduino - це серія мікроконтролерів - Uno, Mega, Nano, Mini тощо. Тепер, будь-який мікроконтролер (тут це Atmega 328 IC на Arduino Uno або Atmega 1280 на arduino Mega).

IDE (Integrated Development Environment) - це в основному програмне забезпечення, яке дозволяє краще редагувати, компілювати та налагоджувати код. Arduino IDE працює на платформі Java. Можна співвіднести це з Eclipse, який є іншою середовищем розробки для Java. Таким чином, мова java має різні IDE, які полегшують використання мови для певної мети. Однак, Eclipse не підтримує функції та команди, які працюють на платі Arduino. Отже, ця Arduino IDE в основному має вбудовані функції і команди, які хоч і працюють на платформі Java, налаштовані для роботи на пристрої arduino. борту. Таким чином, Arduino IDE служить для редагування коду, його компіляції, налагодження, а потім запису коду в arduino dev.

Arduino інтегроване середовище розробки (IDE) є крос-платформним додатком (для ОС Windows, MacOS, Linux), який написаний на мові програмування Java. Він використовується для написання і завантаження програм на плату Arduino.

Вихідний код для IDE випускається під загальнодоступною ліцензією GNU версії 2. Arduino IDE підтримує мови C і C ++ з використанням спеціальних правил структурування коду. Arduino IDE надає бібліотеку програмного забезпечення з проекту Wiring, яка надає безліч загальних процедур введення і виведення. Для написаного користувачем коду потрібні тільки дві основні функції - для запуску ескізу і основного циклу програми, які компілюються і зв'язуються з заглушкою програми main () в виконувану циклічну виконавчу програму за допомогою ланцюжка інструментів GNU,

					БР.КСМ.07180/17.00.00.000.ПЗ	Арк.
						36
Змн.	Арк.	№ докум.	Підпис	Дата		

також включеною в дистрибутив IDE. Arduino IDE (див. рисунок 2.13) використовує програму avrdude для перетворення коду, в текстовий файл в шістнадцятковому кодуванні, який завантажується в плату Arduino за допомогою програми-завантажувача в прошивці плати.

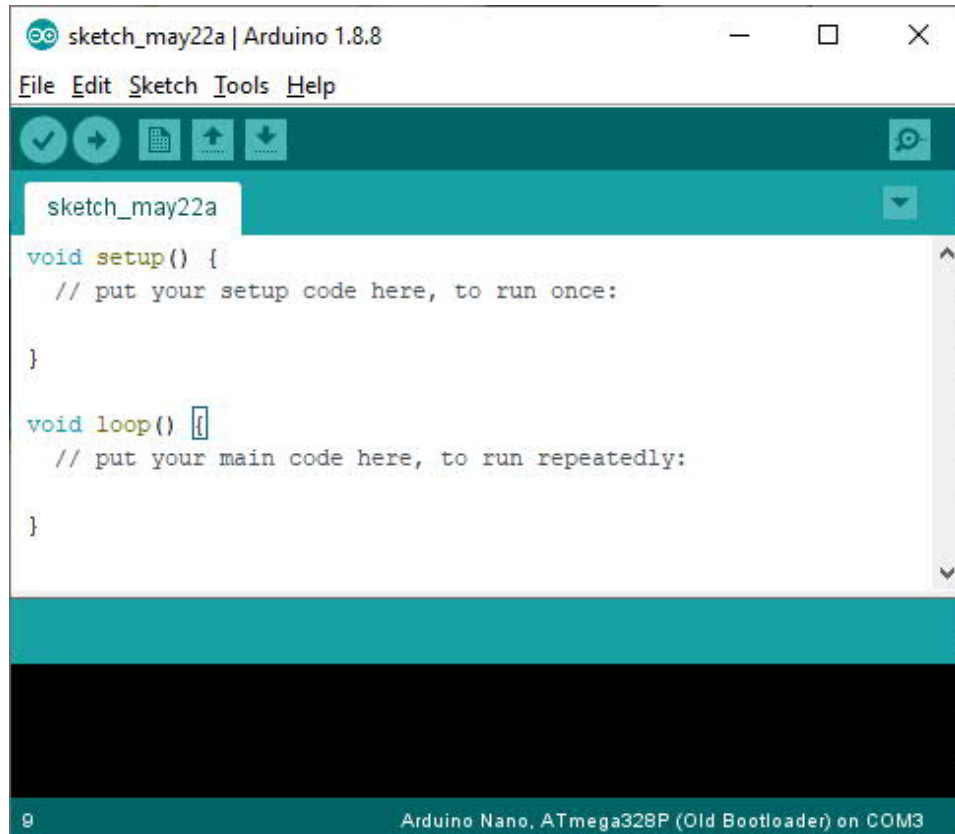


Рисунок 2.13 – Зовнішній вигляд Arduino IDE 1.8.8

Середовище розробки в основному складається з трьох розділів:

1. Панель інструментів.
2. Текстовий редактор.
3. Панель виводу інформації.

Панель, що з'являється у верхній частині, називається панель інструментів, яка складається з п'яти різних варіантів.

- файл – в цьому розділі можна відкрити нове вікно для написання коду або відкриття існуючого;
- редагувати – використовується для копіювання та вставлення коду з подальшою модифікацією шрифту;

- скетч – для компіляції та програмування;
- інструменти – в основному використовується для тестування проектів.

Розділ програміст на цій панелі використовується для запису завантажувача до нового мікроконтролера;

- допомога – у випадку, якщо ви відчуваєте сумніви щодо програмного забезпечення, доступна повна допомога від початку роботи з усунення несправностей.

Ці шість кнопок (див. рисунок 2.14), що з'являються на вкладці меню пов'язані із запущеною програмою.

1) галочка, що з'являється у круговій кнопці, використовується для перевірки коду;

2) клавіша зі стрілкою буде завантажувати та передавати необхідний код на плату Arduino;

3) пунктирний папір використовується для створення нового файлу;

4) стрілка вгору призначена для відкриття існуючого проекту Arduino;

5) стрілка вниз використовується для збереження поточного запущеного коду;

б) кнопка, що з'являється у верхньому правому куті - це серійний монітор - окреме спливаюче вікно, яке діє як незалежний термінал і відіграє важливу роль для надсилання та отримання серійних даних. Ви також можете перейти до панелі інструментів і вибрати послідовний монітор, або натиснути клавіші Ctrl + Shift + M відразу відкриє його. Послідовний монітор дійсно допоможе налагодити написані ескізи, де ви зможете отримати інформацію про те, як працює ваша програма. Модуль Arduino повинен бути підключений до комп'ютера за допомогою кабелю USB, щоб активувати серійний монітор.



Рисунок 2.14 – Кнопки на панелі інструментів Arduino IDE 1.8.8

Головний екран під меню кнопок відомий як простий текстовий редактор (див. рисунок 2.15), який використовується для написання необхідного коду.

					БР.КСМ.07180/17.00.00.000.ПЗ	Арк.
						38
Змн.	Арк.	№ докум.	Підпис	Дата		

```
sketch_may22a $
#include <LiquidCrystal.h>
LiquidCrystal lcd(7, 6, 5, 4, 3, 2);

int switchPin1 = 8;
int switchPin2 = 9;

void setup(){
  pinMode(switchPin1, INPUT);
  pinMode(switchPin2, INPUT);
  lcd.begin(16, 2);
}

void loop(){

  if(digitalRead(switchPin1) == LOW){
    lcd.setCursor(0, 0);
    lcd.print("          ");
  }
  else{
    lcd.setCursor(0, 0);
    lcd.print("Somonyuk Vitaliy");
  }
}
```

Рисунок 2.15 – Зображення текстового редактора

У нижній частині головного екрана описується панель виводу (див. рисунок 2.16), в якій головним чином висвітлюється стан компіляції запущеного коду пам'ять, що використовується кодом, і помилки, що сталися в програмі. Потрібно виправити ці помилки перед тим, як завантажувати файл у модуль Arduino.



Рисунок 2.16 – Зображення помилки на панелі виводу

Бібліотеки дуже корисні для додавання додаткових функцій до модуля Arduino. Існує список бібліотек, які можна додати, натиснувши кнопку Sketch на панелі меню, а також вибрати Include Library. Більшість бібліотек

попередньо інсталювані та постачаються з програмним забезпеченням Arduino. Проте їх також можна завантажити з зовнішніх джерел.

Щоб завантажити Sketch, потрібно вибрати відповідну плату (див. рисунок 2.17), що використовується, і порти для цієї операційної системи. Коли натиснем Tools у меню, воно відкриється, далі перейдемо до розділу Board і виберем плату, на якій хочемо працювати. Аналогічно, COM1, COM2, COM4, COM5, COM7 або вище зарезервовані та USB-плати.

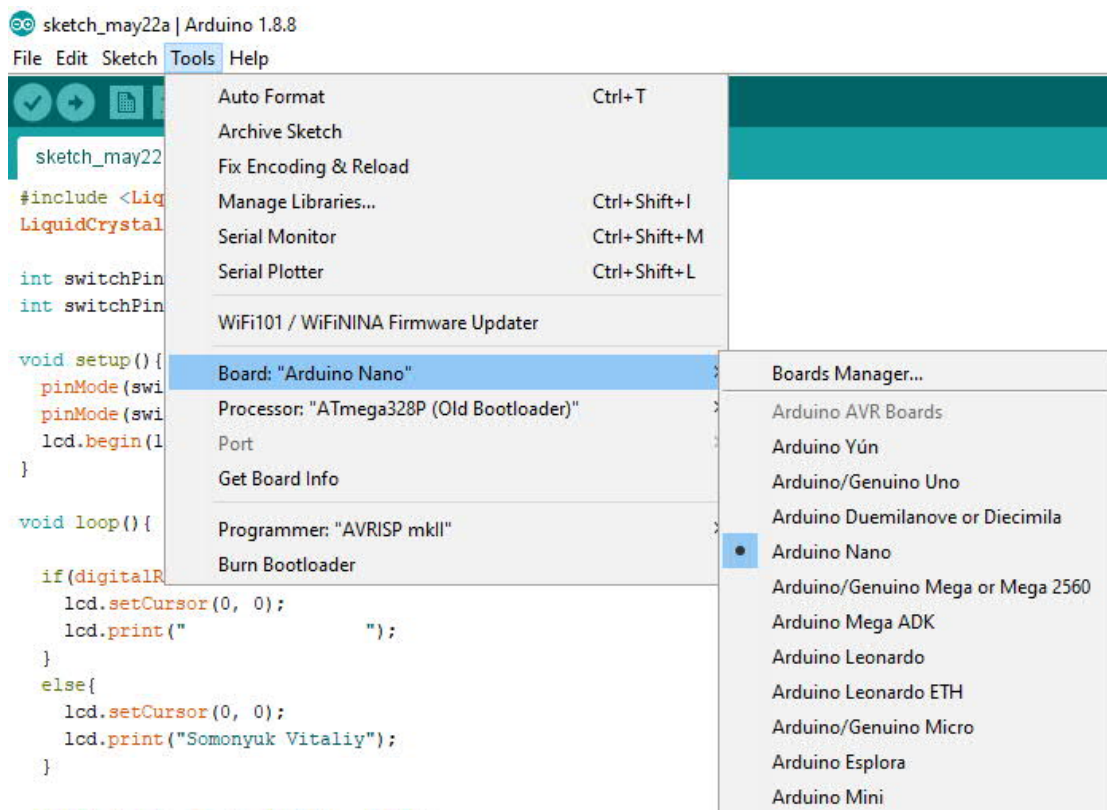


Рисунок 2.17 – Зображення вибору плати

3 ПРОГРАМНО-АПАРАТНА РЕАЛІЗАЦІЯ СИСТЕМИ УПРАЛІННЯ РОБОТОТЕХНІЧНОЮ ПЛАТФОРМАЮ НА ОСНОВІ АКСЕЛЕРОМЕТРА ТА МІКРОКОНТРОЛЕРА

3.1 Програмна реалізація системи управління робототехнічною платформаю на основі акселерометра та мікроконтролера

В якості керування на відстані використовується акселерометр MPU 6050 (рисунок 2.9) та два Wi-Fi модуля nRf24l01 (рисунок 2.10). Написаний код написаний на мові програмування C. C - це мова програмування високого рівня та загального призначення, яка ідеально підходить для розробки програмного забезпечення або портативних програм. Спочатку C був призначений для написання системного програмного забезпечення, C був розроблений в Bell Labs Деннісом Рітчі для операційної системи Unix на початку 1970-х років. Зайняв одне з найбільш поширених мов, C має компілятор для більшості комп'ютерних систем і вплинув на багато популярних мов, зокрема C ++.

C належить до структурованих, процедурних парадигм мов. Він перевірений, гнучкий і потужний і може бути використаний для різних застосувань. Хоча високий рівень, C і мова асемблера мають багато однакових атрибутів. Робота передавача дуже проста, якщо живлення увімкнене то передаємо сигнал на приймач. Пакет та час передачі сигналу зображено на рисунку 3.1.

Розглянемо частину коду, яка відповідає за передачу сигналу:

```
#include<Wire.h>
#include <SPI.h>
#include "RF24.h"
RF24 radio(9,10);
const uint64_t pipe = 0xE8E8F0F0E1LL;
int msg[1];
int val_1;
int val_2;
```

					БР.КСМ.07180/17.00.00.000.ПЗ	Арк.
						41
Змн.	Арк.	№ докум.	Підпис	Дата		

```

const int MPU_addr=0x68;
int16_t AcX,AcY;

void setup()
{
  Wire.begin();
  Wire.beginTransmission(MPU_addr);
  Wire.write(0x6B);
  Wire.write(0);
  Wire.endTransmission(true);
  Serial.begin(9600);

  radio.begin();
  radio.openWritingPipe(pipe);
}

void loop()
{
  Wire.beginTransmission(MPU_addr);
  Wire.write(0x3B);
  Wire.endTransmission(false);
  Wire.requestFrom(MPU_addr,14,true);

  AcX=Wire.read()<<8|Wire.read();
  AcY=Wire.read()<<8|Wire.read();

  val_1 = map(AcX, -17000, 17000, 0, 127),msg[0] = val_1,radio.write(msg, 1);
  val_2 = map(AcY, -17000, 17000, 128, 256),msg[0] = val_2,radio.write(msg, 1);}

```

					БР.КСМ.07180/17.00.00.000.ПЗ	Арк.
						42
Змн.	Арк.	№ докум.	Підпис	Дата		



Рисунок 3.1 – Зображення часу передачі сигналу

Приймач працює за подібним алгоритмом, якщо сигнал передає передатчик то потрібно робити певні дії. Розглянемо код приймача сигналу:

```
#include <SPI.h>
#include <Servo.h>
#include "RF24.h"
Servo myservo;
RF24 radio(9,10);
const uint64_t pipe = 0xE8E8F0F0E1LL;
int msg[1];
int data;
int pos;
int fspeed;
int bspeed;
const int in3 = 2;
const int in4 = 4;
const int enb = 3;
void setup()
{
  pinMode(in3, OUTPUT);
  pinMode(in4, OUTPUT);
  pinMode(enb, OUTPUT);
}
```

					БР.КСМ.07180/17.00.00.000.ПЗ	Арк.
						43
Змн.	Арк.	№ докум.	Підпис	Дата		


```

myservo.attach(8);
delay(50);
radio.begin();
radio.openReadingPipe(1,pipe);
radio.startListening();}
void stop(){
  analogWrite(enb, 0);
  digitalWrite(in3, LOW);
  digitalWrite(in4, LOW);
  pos = map(data, 0, 127, 0, 0);
}
void forward(int fspeed)
{
  digitalWrite(in3, LOW);
  digitalWrite(in4, HIGH);
  analogWrite(enb, bspeed);
}
void backward(int bspeed)
{
  digitalWrite(in3, HIGH);
  digitalWrite(in4, LOW);
  analogWrite(enb, fspeed);
}
void loop()
{
  if (radio.available())radio.read(msg, 1);
  if (msg[0] >=0 && msg[0] <=50)data = msg[0], bspeed = map(data, 189, 128, 200, -255),
backward(bspeed);
  if (msg[0] >=76 && msg[0] <=126)data = msg[0], fspeed = map(data, 255, 201, 200, -
255), forward(fspeed);

  if (msg[0] >51 && msg[0] <75)data = msg[0], stop();
  if (msg[0] >176 && msg[0] <214)data = msg[0], stop();

  if (msg[0] >=127 && msg[0] <=175)data = msg[0],myservo.write(-80);

```

					БР.КСМ.07180/17.00.00.000.ПЗ	Арк.
						44
Змн.	Арк.	№ докум.	Підпис	Дата		

```

if (msg[0] >=215 && msg[0] <=255)data = msg[0],myservo.write(160)
}

```

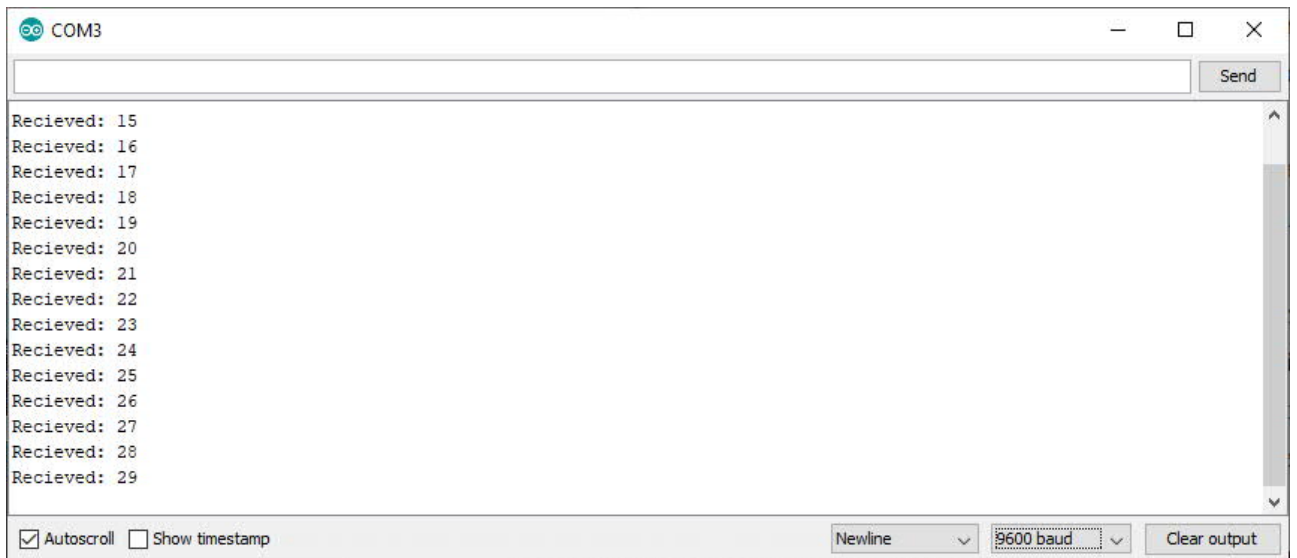


Рисунок 3.2 – Зображення отриманих пакетів на приймачі

На рисунку 3.2 зображено тільки порядок отриманих пакетів, які надходять з передавача.

3.2 Налаштування акселерометра MPU 6050

Обраний для реалізації системи датчик MPU 6050 має велику кількість регістрів, розібравшись з роботою яких можна налаштувати всі важливі параметри датчика, такі як: гіроскоп та акселерометр. Акселерометр і гіроскоп фільтруються в відповідно до значення поля DLPF_CFG, як показано в таблиці 3.1 нижче.

					БР.КСМ.07180/17.00.00.000.ПЗ	Арк.
						45
Змн.	Арк.	№ докум.	Підпис	Дата		

Таблиця 3.1 Характеристики вбудованого цифрового ФНЧ

DLPF_CFG	Акселерометр (частота 1кГц)		Гіроскоп		
	Пропускна здатність	Затримка (мс)	Пропускна здатність	Затримка (мс)	Тактова частота (кГц)
0	260	0.0	256	0.98	8
1	184	2.0	188	1.9	1
2	94	3.0	98	2.8	1
3	44	4.9	42	4.8	1
4	21	8.5	20	8.3	1
5	10	13.8	10	13.4	1
6	5	19.0	5	18.6	1
7	RESERVED		RESERVED		8

Далі потрібно налаштувати діапазони гіроскопа та акселерометра. Поле FS_SEL вибирає діапазон гіроскопа в наведені нижче таблиці 3.2.

Таблиця 3.2 Діапазони гіроскопа

FS_SEL	Діапазон
0	$\pm 250^\circ/\text{с}$
1	$\pm 500^\circ/\text{с}$
2	$\pm 1000^\circ/\text{с}$
3	$\pm 2000^\circ/\text{с}$

Поле ASF_SEL вибирає діапазон акселерометра в наведені нижче таблиці 3.3.

Таблиця 3.3 Діапазони акселерометра

AFS_SEL	Діапазон
0	$\pm 2g$
1	$\pm 4g$
2	$\pm 8g$
3	$\pm 16g$

Залежно від обраного діапазону акселерометр і гіроскоп мають власні коефіцієнти перетворення, які представлені в таблицях 3.2 і 3.3. Цей коефіцієнт перетворення впливає на похибку датчика, тому його слід враховувати.

Таблиця 3.4 Залежність коефіцієнту перетворення від діапазону акселерометра

AFS_SEL	Діапазон	Коефіцієнт перетворення
0	$\pm 2g$	16384 квант/g
1	$\pm 4g$	8192 квант/g
2	$\pm 8g$	4096 квант/g
3	$\pm 16g$	2048 квант/g

Таблиця 3.5 Залежність коефіцієнту перетворення від діапазону гіроскопа

FS_SEL	Діапазон	Коефіцієнт перетворення
0	$\pm 250^\circ/c$	131 квант/ $^\circ/c$
1	$\pm 500^\circ/c$	65.5 квант/ $^\circ/c$
2	$\pm 1000^\circ/c$	32.8 квант/ $^\circ/c$
3	$\pm 2000^\circ/c$	16.4 квант/ $^\circ/c$

З таблиць видно, що найменші діапазони акселерометра і гіроскопа мають найбільший коефіцієнт перетворення. Однак в розроблені системі датчик

					БР.КСМ.07180/17.00.00.000.ПЗ	Арк.
						47
Змн.	Арк.	№ докум.	Підпис	Дата		

кріпиться на верхню кінцівку, а це значить, що діапазон гіроскопа $\pm 250^\circ / \text{с}$ і акселерометра $\pm 2g$ відповідають вимогам до системи.

Після реалізації програмного забезпечення були зроблені експерименти в статиці і динаміці для перевірки коректності роботи всієї системи. Нижче наведені результати експерименту в статиці з почерговою орієнтацією датчика вертикально по кожній з осей.

Показання акселерометра і гіроскопа при вертикальному положенні по осі X показані на рисунках 3.3 та 3.4.

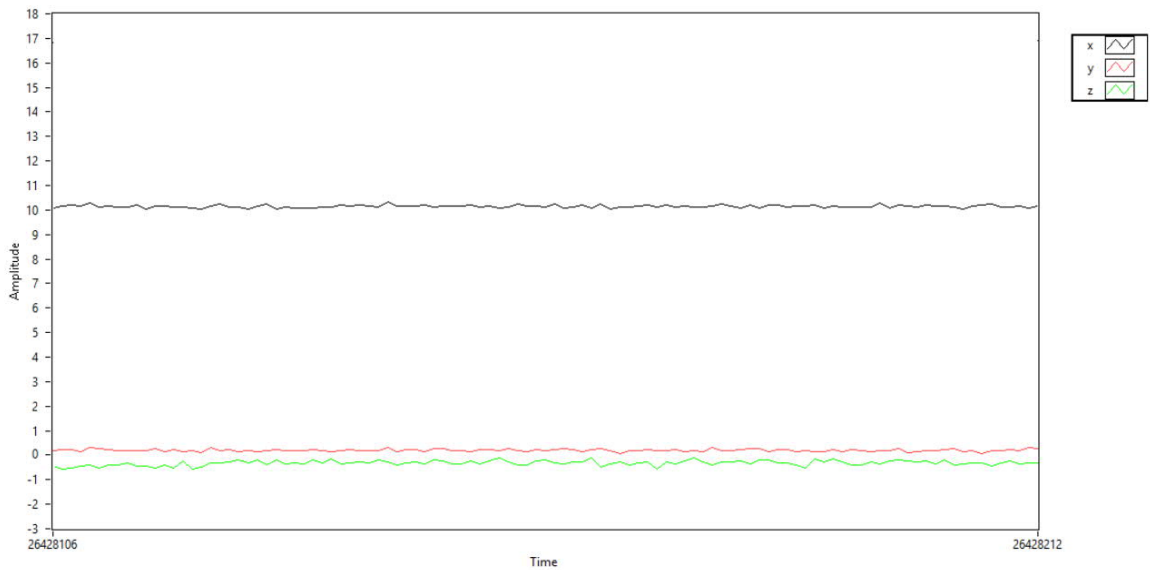


Рисунок 3.3 Показання акселерометра з вертикальному положенні осі X

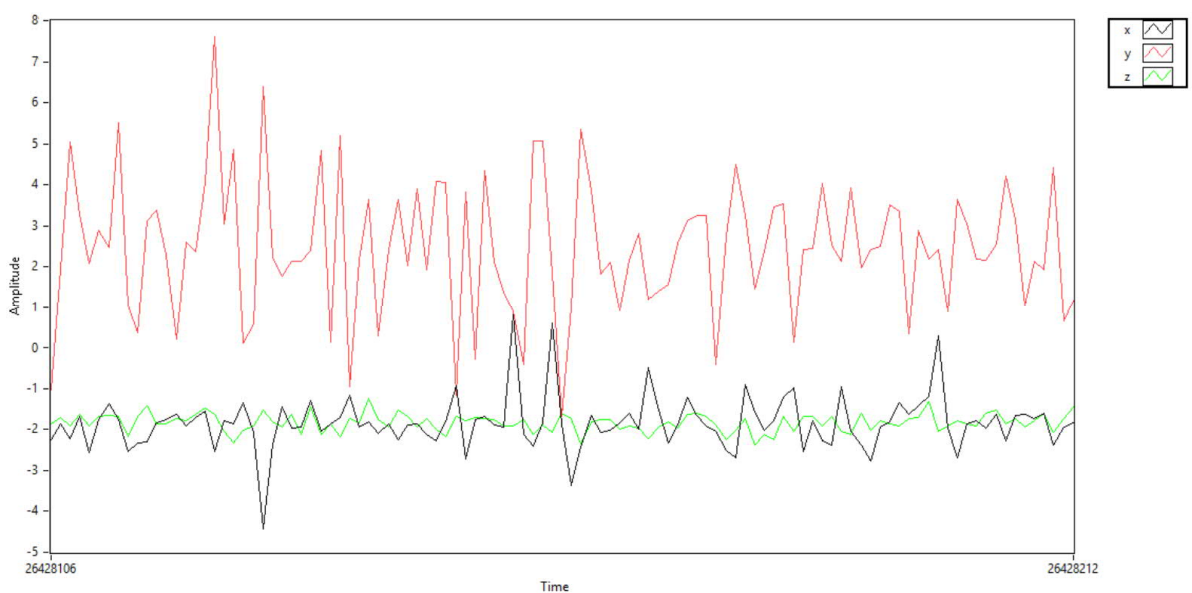


Рисунок 3.4 Показання гіроскопа з вертикальному положенні осі X

					БР.КСМ.07180/17.00.00.000.ПЗ	Арк.
						48
Змн.	Арк.	№ докум.	Підпис	Дата		

Показання акселерометра і гіроскопа при вертикальному положенні по осі Y показані на малюнках 3.5 і 3.6.

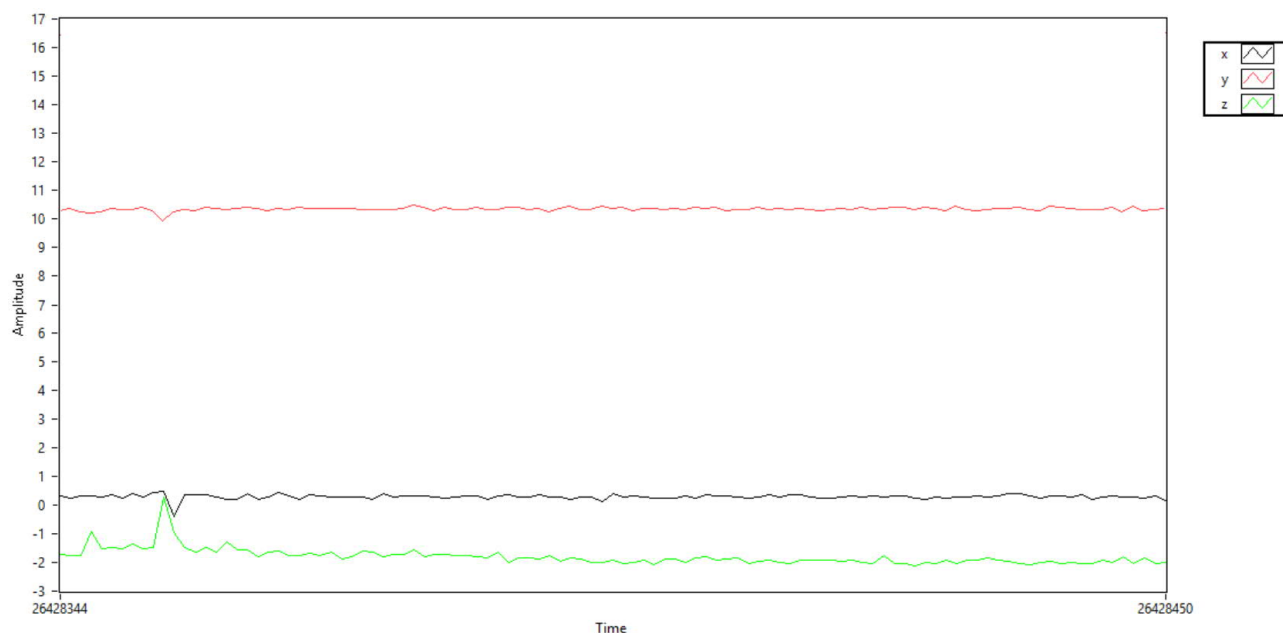


Рисунок 3.5 Показання акселерометра з вертикальному положенні осі Y

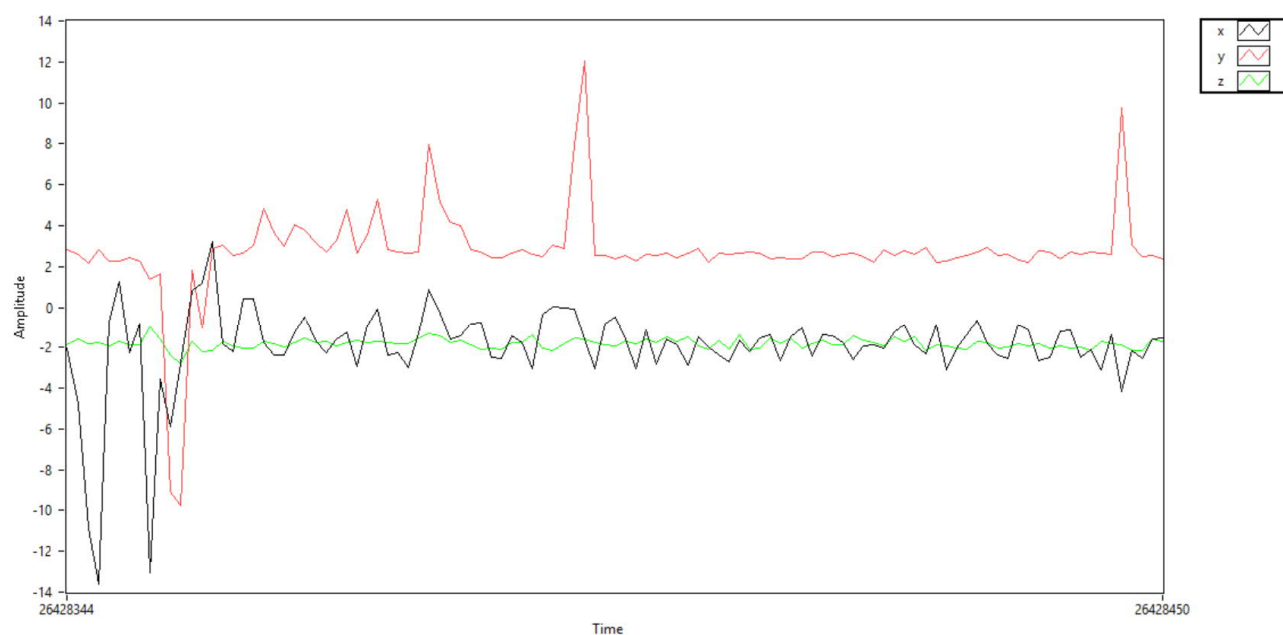


Рисунок 3.6 Показання гіроскопа з вертикальному положенні осі Y

Результати, отримані при налаштуванні датчика в статичному положенні, цілком відповідають очікуванням. Значення лінійного прискорення при вертикальній орієнтації на кожну з осей приблизно дорівнює g (9.81 м / с^2) по

					БР.КСМ.07180/17.00.00.000.ПЗ	Арк.
						49
Змн.	Арк.	№ докум.	Підпис	Дата		

цій осі, значення інших осей коливаються близько нуля. Це свідчить про те, що дані, отримані з датчика коректні. Невелика похибка може бути обумовлена зовнішніми впливами на датчик під час налаштування, а також інструментальною похибкою самого датчика.

3.3 Опис логічної структури програмної частини робототехнічної платформи

Логічна структура програмно-апаратного засобу містить набір функціонально-логічних модулів. В процесі розробки робототехнічної платформи виконуються роботи, що включають процедури та об'єкти, які представляють собою стандартні бібліотеки. Багато мікроконтролерів мають вбудовані периферійні пристрої SPI, які обробляють всі деталі надсилання та отримання даних, і можуть робити це з дуже високою швидкістю. Протокол SPI також досить простий, що можна писати свої власні процедури, щоб маніпулювати лініями введення-виведення в належній послідовності для передачі даних.

Якщо використовувати Arduino, можна взаємодіяти з пристроями SPI двома способами:

1. Можна використовувати команди `shiftIn ()` і `shiftOut ()`. Це програмні команди, які будуть працювати на будь-якій групі контактів, але будуть дещо повільними.

2. Або можна скористатися бібліотекою SPI, яка використовує вбудований у мікроконтролер апаратне забезпечення SPI. Це значно швидше, ніж вищезазначені команди, але він працюватиме лише на певних контактах.

Під час налаштування інтерфейсу потрібно вибрати деякі параметри. Ці параметри повинні відповідати параметрам пристрою.

					БР.КСМ.07180/17.00.00.000.ПЗ	Арк.
						50
Змн.	Арк.	№ докум.	Підпис	Дата		

SPI може працювати на дуже високих швидкостях (мільйони байт в секунду), що може бути занадто швидким для деяких пристроїв. Для розміщення таких пристроїв можна регулювати швидкість передачі даних. У бібліотеці Arduino SPI швидкість встановлюється функцією `setClockDivider ()`, яка розділяє головні годинники (16 МГц на більшості Arduinos) до частоти між 8 МГц (/ 2) і 125 кГц (/ 128).

Якщо використовувати бібліотеку SPI, потрібно використовувати надані штирі SCK, MOSI і MISO, оскільки апаратні засоби підключені до цих контактів.

Arduino бібліотека Servo являє собою набір функцій для управління сервоприводами. Дана бібліотека дає можливість управляти відразу дванадцятьма сервоприводами за допомогою більшості мікроконтролерів Ардуіно. Деякі плати Ардуіно дозволяють підключати менше сервоприводів (такі плати як Arduino LilyPad) так як у їх менше цифрових вводів / висновків. Інші плати дають можливість управляти відразу 48-ю сервоприводами (Arduino Mega).

Використання бібліотеки Servo накладає деякі обмеження. На всіх платах крім Arduino Mega, при роботі з даною бібліотекою, пропадає можливість використовувати цифрові Піни 9 і 10 в режимі ШІМ. На платі Ардуіно Мега режим ШІМ стає недоступний на пінах 11 і 12, тільки при підключенні більше 12 сервоприводів.

					БР.КСМ.07180/17.00.00.000.ПЗ	Арк.
						51
Змн.	Арк.	№ докум.	Підпис	Дата		

4 ТЕХНІКО-ЕКОНОМІЧНЕ ОБГРУНТУВАННЯ РОЗРОБКИ КОМП'ЮТЕРНИХ СИСТЕМ

4.1 Розрахунок витрат на розробку програмно-апаратного засобу

Витрати на виконання програмно-апаратного засобу розраховуються шляхом складання калькуляції кошторисної вартості за наступними статтями:

- витрати на оплату праці;
- відрахування на соціальні заходи;
- матеріальні витрати;
- витрати на використання комп'ютерної техніки;
- витрати на використання спецобладнання для наукових (експериментальних) робіт;
- накладні витрати;
- інші витрати.

4.1.1 Розрахунок витрат на оплату праці

Витрати на оплату праці включають заробітну плату (ЗП) всіх категорій працівників, безпосередньо зайнятих на всіх етапах проектування. Розмір ЗП обчислюється на основі трудомісткості відповідних робіт у людино-днях та середньої ЗП відповідних категорій працівників.

У розробці програмно – апаратного засобу задіяні наступні спеціалісти - розробники, а саме: керівник проекту; студент-дипломник; консультант техніко-економічного розділу (таблиця 4.1).

Таблиця 4.1 - Вихідні дані для розрахунку витрат на оплату праці

№ п/п	Посада виконавців	Місячний оклад, грн.
1	Керівник ДП, старший викладач	4916

					БР.КСМ.07180/17.00.00.000.ПЗ	Арк.
						52
Змн.	Арк.	№ докум.	Підпис	Дата		

Продовження таблиці 4.1

2	Консультант техніко-економічного розділу, доцент	6086
3	Студент	1330

Витрати на оплату праці розробників проекту визначаються за формулою:

$$B_{оп} = \sum_{i=1}^N \sum_{j=1}^M n_{ij} \cdot t_{ij} \cdot C_{ij}, \quad (4.1)$$

де n_{ij} – чисельність розробників i -ої спеціальності j -го тарифного розряду, осіб;
 t_{ij} – затрачений час на розробку проекту співробітником i -ої спеціальності j -го тарифного розряду, год;
 C_{ij} – годинна ставка працівника i -ої спеціальності j -го тарифного розряду, грн.,

Середньо годинна ставка працівника може бути розрахована за формулою:

$$C_{ij} = \frac{C_{ij}^0(1+h)}{PЧ_i}, \quad (4.2)$$

де C_{ij} – основна місячна заробітна плата розробника i -ої спеціальності j -го тарифного розряду, грн.;
 h – коефіцієнт, що визначає розмір додаткової заробітної плати (при умові наявності доплат);
 $PЧ_i$ – місячний фонд робочого часу працівника i -ої спеціальності j -го тарифного розряду, год. (приймаємо 168 год.).

Середня годинна ставка керівника ДП дорівнює:

$$C_{ij} = \frac{4916(1+0,10)}{168} = 32,19 \text{ грн/год.}$$

Середня годинна ставка консультанта техніко-економічного розділу ДП дорівнює:

					БР.КСМ.07180/17.00.00.000.ПЗ	Арк.
						53
Змн.	Арк.	№ докум.	Підпис	Дата		

$$C_{ij} = \frac{6086(1+1,10)}{168} = 76,07 \text{ грн/год.}$$

Середня годинна оплата студента дорівнює:

$$C_{ij} = \frac{1330}{168} = 7,92 \text{ грн/год.}$$

Результати розрахунку вносимо до таблиці 4.2.

Таблиця 4.2 - Розрахунок витрат на оплату праці

№ п/п	Посада виконавців	Час розробки, год	Погодинна заробітна плата, грн/год.	Витрати на розробку, грн
1	Керівник ДП, доцент	16	32,19	515,04
2	Консультант техніко-економічного розділу, доцент	2	76,07	152,14
3	Студент	90	7,92	712,8
Разом				1379,98

4.1.2 Відрахування на соціальні заходи

Величну відрахувань у спеціальні державні фонди визначають у відсотковому співвідношенні від суми основної та додаткової заробітних плат. Згідно діючого нормативного законодавства сума відрахувань у спеціальні державні фонди складає 20,5 % від суми заробітної плати:

$$B_{\phi} = \frac{20,5}{100} * 1379,98 = 282,89 \text{ грн.}$$

					БР.КСМ.07180/17.00.00.000.ПЗ	Арк.
						54
Змн.	Арк.	№ докум.	Підпис	Дата		

4.1.3 Розрахунок витрат на матеріали та комплектуючі

У таблиці 4.3 наведений перелік купованих виробів і розраховані витрати на них.

Таблиця 4.3- Розрахунок витрат на матеріали та комплектуючі

№ п/п	Найменування купованих виробів	Одиниця виміру	Ціна, грн	Кількість купованих виробів	Сума, грн	Транспортні витрати (10% від суми)	Загальна сума, грн
1	Комплект розумного автомобіля (Elecrow 4WD RC)	шт	1231,10	1	1231,10	123,11	1354,21
2	Arduino UNO R3	шт	185	1	185	18,5	203,5
3	ARDUINO NANO V3 AVR ATMEGA328	шт	128	1	128	12,8	140,8
4	Драйвер двигуна L298N	шт	68	1	68	6,8	74,8
5	Радіомодуль NRF24L01	шт	31	2	62	6,2	68,2
Разом							1841,51

4.1.4 Накладні витрати

Накладні витрати проектних організацій включають три групи видатків: витрати на управління, загальногосподарські витрати, невиробничі витрати. Вони розраховуються за встановленими відсотками до витрат на оплату праці. Середньостатистичний відсоток накладних витрат приймемо 70% від заробітної плати:

$$H=0,7*1379,98=965,98 \text{ грн.}$$

					БР.КСМ.07180/17.00.00.000.ПЗ	Арк.
						55
Змн.	Арк.	№ докум.	Підпис	Дата		

4.1.5. Інші витрати

Інші витрати є витратами, які не враховані в попередніх статтях. Вони становлять 10% від заробітної плати:

$$I=0,1*1379,98=137,99 \text{ грн.}$$

Витрати на розробку програмно – апаратного засобу складають:

$$K_1 = B_{OP} + B_{\Phi} + B_{PB} + H + I \quad (4.3)$$

$$K_1 = 1379,98 + 282,89 + 1841,51 + 965,98 + 137,99 = 4608,35 \text{ грн.}$$

Витрати на відлагодження і дослідну експлуатацію програмно – апаратного засобу визначаємо за формулою:

$$K_2 = S_{m.g.} \cdot t_{\text{від}} \quad (4.4)$$

де $S_{m.g.}$ - вартість однієї машино-години роботи ПК, *грн./год.*

$t_{\text{від}}$ - комп'ютерний час, витрачений на відлагодження і дослідну експлуатацію створеного програмно – апаратного засобу, *год.*

Загальна кількість днів роботи на комп'ютері дорівнює 30 днів. Середній щоденний час роботи на комп'ютері – 2 години. Вартість години роботи комп'ютера дорівнює 5,2 грн. Тому

$$K_2 = 5,2 \cdot 60 = 312 \text{ грн.}$$

На основі отриманих даних складаємо кошторис витрат на розробку програмно – апаратного засобу(таблиця 4.4).

					БР.КСМ.07180/17.00.00.000.ПЗ	Арк.
						56
Змн.	Арк.	№ докум.	Підпис	Дата		

Таблиця 4.4 - Кошторис витрат на розробку програмно – апаратного засобу

№ п/п	Найменування витрат	Сума витрат, грн.
1	Витрати на оплату праці	1379,98
2	Відрахування у спеціальні державні фонди	282,89
3	Витрати на куповані вироби	1841,51
4	Накладні витрати	915,28
5	Інші витрати	137,99
6	Витрати на відлагодження і дослідну експлуатацію програмно – апаратного засобу	312
Разом		4499,12

4.2 Визначення експлуатаційних витрат

Для оцінки економічної ефективності розроблюваного програмно - апаратного засобу слід порівняти його з аналогом, тобто існуючим програмно - апаратним засобом ідентичного функціонального призначення.

Експлуатаційні одноразові витрати по програмно – апаратному засобі і аналогу включають вартість підготовки даних і вартість роботи комп'ютера (за час дії програми):

$$E_{\text{п}} = E_{\text{1п}} + E_{\text{2п}} \quad (4.5)$$

де $E_{\text{п}}$ - одноразові експлуатаційні витрати на програмно – апаратний засіб (аналог), грн.;

$E_{\text{1п}}$ - вартість підготовки даних для експлуатації програмно – апаратного засобу (аналогу), грн.;

$E_{\text{2п}}$ - вартість роботи комп'ютера для розробки програмно – апаратного засобу (аналогу), грн.

					БР.КСМ.07180/17.00.00.000.ПЗ	Арк.
						57
Змн.	Арк.	№ докум.	Підпис	Дата		

Річні експлуатаційні витрати $B_{\text{еп}}$ визначаються за формулою:

$$B_{\text{еп}} = E_{\text{п}} * N_{\text{п}} \quad (4.6)$$

де $N_{\text{п}}$ - періодичність експлуатації програмно – апаратного засобу (аналогу), раз/рік(кількість робочих днів у році).

Вартість підготовки даних для роботи на комп'ютері визначається за формулою:

$$E_{\text{п}} = \sum_{i=1}^n n_i t_i c_i \quad (4.7)$$

де i - категорії працівників, які приймають участь у підготовці даних ($i=1,2,\dots,n$);

n_i - кількість працівників i -ої категорії, осіб.;

t_i - трудомісткість роботи співробітників i -ої категорії по підготовці даних, год.;

c_i - середнього годинна ставка працівника i -ої категорії з врахуванням додаткової заробітної плати, що знаходиться із співвідношення:

$$c_i = \frac{c_i^0 (1+b)}{m} \quad (4.8)$$

де c_i^0 - основна місячна заробітна плата працівника i -ої категорії, грн.;

b - коефіцієнт, який враховує додаткову заробітну плату (прийmemo 0,57;

m - кількість робочих годин у місяці, год.

Для роботи з даними як для поточного програмно – апаратного засобу так і для аналогу потрібен один працівник, основна місячна заробітна плата якого складає: $c^0 = 3200$ грн. Тоді:

					БР.КСМ.07180/17.00.00.000.ПЗ	Арк.
						58
Змн.	Арк.	№ докум.	Підпис	Дата		

$$c_i = \frac{3200(1 + 0,57)}{22 * 8} = 28,55 \text{ грн/год}$$

Трудомісткість обробки даних для програмно – апаратного засобу (таблиця 4.5) складає 1 год.

Таблиця 4.5 - Розрахунок витрат на обробку даних та реалізацію програмно – апаратного засобу на комп'ютері.

№	Час роботи співробітників, год.	Середньогодинна заробітна плата, грн./год.	Витрати, грн.
Програмно – апаратний засіб			
1	1	7,92	7,92
Аналог			
1	1	28,55	28,55

Витрати на експлуатацію комп'ютера визначається за формулою:

$$E_{2П} = t * S_{МГ} \quad (4.9)$$

де t - витрати машинного часу для реалізації програмно – апаратного засобу (аналогу), год.;

$S_{МГ}$ - вартість однієї години роботи комп'ютера, грн./год(приймаємо 5,2 грн/год).

$$E_{2П} = 1 * 5,2 = 5,2 \text{ грн.}; E_{2а} = 1,5 * 5,2 = 7,8 \text{ грн.}$$

$$E_{П} = 7,92 + 5,2 = 13,12 \text{ грн.}; E_{а} = 28,55 + 7,8 = 36,35 \text{ грн.}$$

$$B_{еп} = 13,12 * 250 = 3280 \text{ грн.}; B_{еа} = 36,35 * 250 = 9087,5 \text{ грн.}$$

					БР.КСМ.07180/17.00.00.000.ПЗ	Арк.
						59
Змн.	Арк.	№ докум.	Підпис	Дата		

4.3 Розрахунок ціни споживання програмно – апаратного засобу

Ціна споживання - це витрати на придбання і експлуатацію програмно – апаратного засобу за весь строк його служби:

$$Ц_{C(П)} = Ц_{П} + B_{(E)NPV} \quad (4.10)$$

де $Ц_{П}$ - ціна придбання програмно – апаратного засобу, грн.:

$$Ц_{П} = K(1 + \frac{П_p}{100}) + K_0 + K_k \quad (4.11)$$

де K - кошторисна вартість;

$П_p$ - рентабельність;

K_0 - витрати на прив'язку та освоєння програмо – апаратного засобу на конкретному об'єкті, грн.;

K_k - витрати на доукомплектування технічних засобів на об'єкті, грн.;

Договірна ціна ($Ц_D$) для проектних рішень розраховується за формулою:

$$Ц_D = B_{КС} \cdot \left(1 + \frac{P}{100}\right) \quad (4.12)$$

де $B_{КС}$ – кошторисна вартість, грн.;

P - середній рівень рентабельності, % (приймаємо 25% за погодженням з керівником).

$$Ц_D = 4499,12 \cdot (1 + 0,25) = 5623,9 \text{ грн.}$$

					БР.КСМ.07180/17.00.00.000.ПЗ	Арк.
						60
Змн.	Арк.	№ докум.	Підпис	Дата		

Вартість витрат на експлуатацію програмно – апаратного засобу (за весь час його експлуатації), грн.:

$$B_{\text{епрв}} = \sum_{t=0}^T \frac{B_{\text{еп}}}{(1+R)^t} \quad (4.13)$$

де $B_{\text{еп}}$ - річні експлуатаційні витрати, грн.;

T - термін служби програмно – апаратного засобу, років;

R - річна ставка проценту банку.

$$B_{\text{епрв}} = \sum_{t=1}^5 \frac{3280}{(1+0,12)^t} = 9318,19 \text{ грн.}$$

$$B_{\text{епрв}} = \sum_{t=1}^5 \frac{9087,5}{(1+0,12)^t} = 25816,76 \text{ грн.}$$

Тоді ціна споживання програмно – апаратного засобу дорівнюватиме:

$$Ц_{\text{сп}} = 5623,9 + 9318,19 = 14942,09 \text{ грн.}$$

Аналогічно визначається ціна споживання для аналогу:

$$Ц_{\text{са}} = 1500,0 + 25816,76 = 27316,76 \text{ грн}$$

4.4 Визначення показників економічної ефективності

Економічний ефект в сфері розробки програмно – апаратного засобу:

					БР.КСМ.07180/17.00.00.000.ПЗ	Арк.
						61
Змн.	Арк.	№ докум.	Підпис	Дата		

$$E_{\text{ПР}} = C_{\text{П}} - C_{\text{А}} \quad (4.14)$$

$$E_{\text{ПР}} = 5623,9 - 1500,0 = 4123,9 \text{ грн.}$$

Річний економічний ефект в сфері експлуатації:

$$E_{\text{КС}} = B_{\text{ЕА}} - B_{\text{ЕП}} \quad (4.15)$$

$$E_{\text{КС}} = 9087,5 - 3280 = 5827,5 \text{ грн.}$$

Додатковий економічний ефект у сфері експлуатації:

$$\Delta E_{\text{екс}} = \sum_{t=1}^T E_{\text{екс}} (1 + R)^{T-t} \quad (4.16)$$

$$\Delta E_{\text{екс}} = \sum_{t=1}^5 5827,5 (1 + 0,12)^{5-t} = 36421,87 \text{ грн.}$$

Сумарний ефект складає:

$$E = E_{\text{пр}} + \Delta E_{\text{екс}} = 4123,9 + 36421,87 = 40545,77 \text{ грн.}$$

Таблиця 4.6 - Показники економічної ефективності програмно – апаратного засобу.

№	Найменування	Одиниці вимірювання	Значення показників	
			Базовий варіант	Новий варіант
1	Капітальні вкладення	грн.	-	1900
2	Ціна придбання	грн.	1500,0	5623,9

					БР.КСМ.07180/17.00.00.000.ПЗ	Арк.
						62
Змн.	Арк.	№ докум.	Підпис	Дата		

Продовження Таблиці 4.6

3	Річні експлуатаційні витрати	<i>грн.</i>	9087,5	3280
4	Ціна споживання	<i>грн.</i>	27316,76	14942,09
5	Економічний ефект в сфері проектування	<i>грн.</i>	-	4123,9
6	Економічний ефект в сфері експлуатації	<i>грн.</i>	-	5827,5
7	Додатковий ефект в сфері експлуатації	<i>грн.</i>	-	36421,87
8	Сумарний ефект	<i>грн.</i>	40545,77	

					БР.КСМ.07180/17.00.00.000.ПЗ	Арк.
						63
Змн.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

В ході випускної кваліфікаційної роботи був проведений аналіз основних аспектів і проблем дослідження алгоритмів керування систем пересування робототехнічної платформи. Було розроблено робототехнічну платформу на основі мікроконтролера Arduino UNO R3 з використанням акселерометра MPU 6050 та з виростанням безпроводних передавачів, що дозволяє керувати робототехнічною платформою на відстані до 200 м.

Розробки в цій роботі можуть бути застосовані, як в практичних цілях, так і для отримання теоретичних матеріалів для досліджень. Робота має велику практичну цінність. Згідно з таблицею 4.6 бачимо, що розроблюваний програмно – апаратний засіб є більш економічно ефективний за наявні аналоги. Дане порівняння показує, що вказаний програмно – апаратний продукт має переваги в порівнянні з аналогами, зокрема: надійність, простота використання, ергономічність, зручність. Згідно проведеного економічного обґрунтування зазначений програмно – апаратний засіб є конкурентоздатним і має право існування на ринку. Крім того, отримано економічний ефект у розмірі 40545,77 грн. і тому розробка і впровадження цього програмно – апаратного засобу є економічно доцільною.

					БР.КСМ.07180/17.00.00.000.ПЗ	Арк.
						64
Змн.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Лавріщева К. М. Програмна інженерія. Загальний курс: підручник / К. М. Лавріщева. – К.:2007. – 445 с.
2. Буч Г. Об'єктно-орієнтовне проектування з прикладами застосування / Г. Буч. – К. : Академія, 2002. – 488с.
3. Шеховцов В. А. Операційні системи/ В. А. Шеховцов. – К. : ВНУ, 2005. – 576 с.
4. Пасічник В. В., Шаховська Н.Б. Сховища даних: навчальний посібник. Львів: «Магнолія 2006», 2008. – 492 с.
5. Грицюк Ю.І., Рак Т.Є. Об'єктно-орієнтоване програмування мовою С++ : навчальний посібник. – Львів : Вид-во Львівського ДУ БЖД, 2011. – 404 с.
6. Вирт Н. Алгоритмы и структуры данных / Н. Вирт ; пер. с англ. – М. : Мир, 1989. – 360 с.
7. Липаев В. В. Програмная инженерия. Методологические основы / В. В. Липаев. – М. : ТЕИС, 2006. – 608 с.
8. Глинський Я. М., Анохін В.Є., Рязька В.А. С++ і С++ Builder: Навчальний посібник 5-те вид. – Львів: СПД Глинський, 2011. – 192 с.
9. Глинський Я. М., Анохін В.Є., Рязька В.А. Паскаль. Turbo Pascal і Delphi: Навчальний посібник 10-те вид., без змін. – Львів: СПД Глинський, 2009. – 192 с.
10. Голощук Р. О., Шаховська Н.Б. Алгоритми і структури даних. – Львів: «Магнолія 2006», 2014. – 216 с.
11. Електронний ресурс. Режим доступу: <https://uk.wikipedia.org/wiki/>
12. Електронний ресурс. Режим доступу: <http://www.programmersforum.ru>
13. Електронний ресурс. Режим доступу: <http://www.cyberforum.ru>
14. Електронний ресурс. Режим доступу: <http://forum.ru-board.com>
15. Електронний ресурс. Режим доступу: <http://arduino.ru>

					БР.КСМ.07180/17.00.00.000.ПЗ	Арк.
						65
Змн.	Арк.	№ докум.	Підпис	Дата		