Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Факультет комп’ютерної інженерії та управління
(повна назва)

Кафедра електронних обчислювальних машин
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

Рівень вищої освіти другий (магістерський)

Система на платформі Arduino програмного управління переміщеннями колісного роботу на підставі даних від датчику одометру
(тема)

Виконав:

студент II курсу, групи СПм-20-1

Мар’янов С. О.
(прізвище, ініціали)

Спеціальність

123 «Комп’ютерна інженерія»
(код і повна назва спеціальності)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма

Системне програмування
(повна назва освітньої програми)

Керівник: проф. Каргін А.О.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри ЕОМ _____ Коваленко А.А.
(підпис) (прізвище, ініціали)

2021 р.

Харківський національний університет радіоелектроніки

Факультет комп’ютерної інженерії та управління

Кафедра електронних обчислювальних машин

Рівень вищої освіти другий (магістерський)

Спеціальність 123 «Комп’ютерна інженерія»
(код і повна назва)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма Системне програмування
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
“____” _____ 20__ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студенту Мар’янову Сергію Олександровичу
(прізвище, ім’я, по батькові)

1. Тема роботи Система на платформі Arduino програмного управління переміщеннями колісного роботу на підставі даних від датчику одометру

затверджена наказом по університету від “ 5 ” листопада 2021 р. № 1657 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 13 грудня 2021 р.

3. Вхідні дані до роботи
завдання на кваліфікаційну роботу

4. Перелік питань, що потрібно опрацювати у роботі
розробка системи програмного управління переміщеннями колісного роботу

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) рисунки, таблиця, слайд-презентація – 11 слайдів

6. Консультанти розділів роботи (заповнюється за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Аналіз проблеми та огляд існуючих рішень	09.11.21-12.11.21	
2.	Вибір технологій розробки	13.11.21-18.11.21	
3.	Розробка алгоритмічного забезпечення	19.11.21-22.11.21	
4.	Розробка програмних модулів	23.11.21-29.11.21	
5.	Відлагодження програмних модулів	30.11.21-03.12.21	
6.	Оформлення матеріалів атестаційної роботи	04.12.21-07.12.21	
7.	Подання кваліфікаційної роботи керівникові та	08.12.21-09.12.21	
	її попередній захист		
8.	Подання атестаційної роботи	10.12.21-11.12.21	

Дата видачі завдання 8 листопада 2021 р.

Студент _____
(підпис)

Керівник роботи _____ проф. Каргін А.О.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 68 с., 10 рис., 2 дод., 15 джерел.

РОБОТ, СИСТЕМА, МАРШРУТ, ОДОМЕТР, ARDUINO, ДАТЧИК, IOT.

Метою кваліфікаційної роботи є розробка системи програмного управління переміщеннями колісного роботу за допомогою одометрчиних даних.

У ході виконання кваліфікаційної роботи було проаналізовано методи програмного управління переміщенням робота за керування за допомогою заздалегідь занесеного маршруту до пам’яті робота як один із актуальних методів. Отриманий пристрій повинен реалізовувати такі функції як: рух по одному з безлічі можливих маршрутів використовуючи в управлінні інформацію від датчика одометру.

ABSTRACT

Master’s thesis: 68 pages, 8 figures, 2 appendices, 15 sources.

ROBOT, SYSTEM, ROUTE, ODOMETER, ARDUINO, SENSOR, IOT.

The major goal of this thesis ıs to develop a software control system for wheel movements using odometric data.

In the course of the qualification work, the methods of software control of robot movement control were analyzed with the help of a pre-recorded route in the robot’s memory as one of the actual methods. The resulting device must implement such functions as: movement on one of many possible routes using control information from the odometer sensor.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

CPU(Central processing unit) - центральний процесор

GPIO(General Purpose Input/Output).- універсальний інтерфейс для введення/виведення

GPU (Graphic processing unit) - графічний процесор

IDE(Integrated Development Environment) - Інтегроване середовище розробки

IoT(Internet of Things) - інтернет речей

RX — Recieve (получение)

TX - Transmitter (передатчик)

USB(Universal Serial Bus) - універсальна послідовна шина

ВСТУП

Одним з ключових завдань мобільної робототехніки є пошук маршруту для руху та його оптимізація. Переміщаючись у робочій місцевості, робот повинен постійно оцінювати навколишню обстановку, визначати своє положення та положення навколишніх об’єктів. Існує безліч різних способів, за допомогою яких робот може визначати власне положення та будувати маршрут між точками призначення. При переміщеннях на вулиці використовується технологія супутникової навігації, а навколишні об’єкти виявляються за допомогою камер або далекомірів. У разі переміщення всередині приміщень за допомогою камер та далекомірів будується віртуальна модель простору, за якою робот орієнтується надалі. Вищеописані методи мають загальний характері і застосовні у довільних ситуаціях, але через це вони дуже складні у реалізації ще не застосовуються широко у повсякденному житті.

Зазвичай автономні робототехнічні системи проектуються під конкретні завдання. Такий підхід дозволяє формалізувати вимоги до системи та розробити всі можливі алгоритми реакції на зміну стану навколишнього оточення. Популярна сфера застосування роботів – сучасна побутова техніка, де набувають популярності автоматичні роботи - пилососи, газонокосарки та інші. Такі побутові пристрої автоматизують роботу при мінімальному втручанні людини, при цьому виконують миючі та чистячі функції, які використовуються в побуті не гірше, а то й краще людини. Ці пристрої визначають та обходять перешкоди, які зустрічають на своєму шляху. Також, одним із досить жорстко формалізованих завдань може бути переміщення об’єктів усередині виробничого приміщення. Як правило, при перевезенні вантажів на складах або у виробничих цехах роботи долають той самий маршрут постійно. Відповідно, цей маршрут заздалегідь відомий і для нього можна розробити систему управління рухом робота.

Мобільні роботи, що пересуваються в цехах за маршрутом, оснащуються різними сенсорними пристроями для сприйняття навколишнього оточення: ІЧ-датчиками, камерами, датчиками безпеки тощо. Але на відміну від розроблених в лабораторних умовах роботів, реальні роботи працюють аж ніяк не в лабораторних умовах – найчастіше напрямна лінія може бути пошкоджена або прихована за будь-яким об’єктом, деякі маршрути можуть перетинатися або частково перериватися. На реальний маршрут можуть накладатися різні обмеження, наприклад: деякі ділянки маршруту можуть бути заборонені для руху, деякі можуть бути досягнуті тільки після проїзду через інші. Для того розпізнавання перетинів напрямних ліній може застосовуватися безліч різних способів, наприклад, в подібних цілях часто застосовуються камери. Але обробка зображень вимагає великої продуктивності бортового комп’ютера робота, тому такі рішення не завжди можуть бути застосовані. Також, необхідно зауважити наскільки робот вимогливий до попереднього упорядкування середовища: обов’язкові початкові умови (строго певний початковий стан робота) і жорсткі вимоги до швидкості переміщення і миттєві режими розгону / гальмування робота. У цих випадках класичний метод програмного керування не можливо використовувати. У методі програмного управління з зворотними зв’язками для ідентифікації події виходу робота на певні позиції використовують інформацію від датчиків, а не орієнтуються на інтервали часу. Як зазначалося в постановці завдання, в управлінні використовується інформація від одометру датчика. Означаючи ці дані є у вигляді відстані що пройдена роботом.

1 АНАЛІЗ АКТУАЛЬНОСТІ ПРОГРАМНО-АПАРАТНОЇ СИСТЕМИ

1.1 Робот як об’єкт дослідження

Робототехніка - це перетин науки, техніки та технологій, у результаті якого створюються машини, звані роботами, які замінюють (або відтворюють) дії людини. Роботи набувають інтелектуальних і механічних здібностей, які не виключають можливості створення машини, подібної до людини, в майбутньому.

1.1.1 Робототехніка і її напрямки

Під роботизованими системами можна зрозуміти будь яку мехатронну систему яка об’єднує в собі керуючі і виконавчі компоненти. Роботи стали не просто дуже поширеними в світі, а вже проникли в усі сфери життя, що спостерігається на прикладі промислових або мобільних роботів. Будь-який робот є складну систему, тому його розробка та проектування можуть базуватися на мехатронному підході до створення технічних систем. Такий підхід передбачає поділ складної системи на більш прості мехатронні модулі, що реалізує тісну інтеграцію механічних, електромеханічних та електронних компонентів, виконавчої та керуючої частин. Одну з ключових ролей роботизованих систем відіграють приводи, які є модулями руху. Конструкція та розрахунок таких модулів має свої особливості.

Надійність роботи залежить від надійності всіх вузлів роботизованої системи. Слід зазначити, що структура системи визначається призначенням робота, його вантажопідйомністю та іншими технічними характеристиками. Це особливо важливо для важких роботів, які використовуються при роботі в шахтах або на підприємствах важкого машинобудування, де є підвищені навантаження. Необхідно зауважити, що створення таких систем зумовлює

необхідність розробки нових методів дослідження та проектування. Природно, що розробка нових методів проектування та дослідження має базуватися на комп’ютерній технології, що забезпечує безпосередню реалізацію всіх етапів роботизованої системи від побудови математичної моделі до управління функціональним рухом у реальному часі. Тому розробка нових методів побудови роботів є актуальною.

Розробка робототехнічних пристроїв може включати всі етапи, починаючи від моделювання моделі об’єкта управління і закінчуючи використанням цих пристроїв на цільові платформи. Описувані нижче етапи застосовні як із створенні мобільних, і промислових роботів і комплексних робототехнічних систем. Прикладами об’єктів промислової робототехніки можуть бути маніпулятори або просунуті верстати тощо, а до мобільних робот відносяться БПЛА, колісні та крокуючі роботи тощо, а також групи з цих роботів. Типовий процес розробки складається з наступних етапів:

- створення системної моделі;
- синтез системи керування;
- автоматична генерація коду та його тестування.

Перед побудовою моделі слід розуміти, по-перше, що потрібно від моделі, а по-друге, ніж у моделі можна знехтувати. Побудова моделі у випадку починається з умовного поділу об’єкта на частини, після чого кожна з цих частин моделюється. У модель може входити практично вся інформація про нього: опис структури, алгоритм управління, тестові сценарії та інше, що необхідно для розробки. Для будь-якого типу моделі серед MATLAB&Simulink можна знайти відповідний інструмент. Наприклад, є можливість створювати моделі фізичних систем, що містять компоненти різних інженерних сфер діяльності: механічних, електричних, гідравлічних та інших. Для уточнення параметрів моделі можна використовувати експериментальні дані з реального обладнання незалежно від того, чи воно поруч чи далеко/недоступно в даний момент. Для деяких завдань необхідно моделювати графічне 3D-оточення об’єкта та взаємодіяти з ним у процесі

симуляції.

Споживчі та економічні тенденції у всьому світі стимулюють розвиток багатофункціонального малосерійного контрактного виробництва, яке має бути універсальним, продуктивним та ефективним. Сучасні інтелектуальні системи та роботи стають все більш корисними, продуктивними, розумними та здатними вирішувати широке коло складних завдань. Тим не менш, основне завдання індустрії - переоснастити виробництво промислової робототехнікою - не буде реалізовано доти, доки в рамках виробничих цехів не буде впроваджено нові економічно ефективні досягнення в галузі робототехнічних датчиків, засобів зв’язку, компонентів живлення, безпеки роботів, а також їх мініатюризації.

1.1.3 Процеси роботизації

Автоматизація — широкое поняття, яке описує всі процеси, які виконуються автоматично, за допомогою ПО або роботів. То є, це ті завдання, які виконуються без участі людини. Роботизация же охватывает лише ту частину цієї практики, коли на заміну людям приходять фізичні механізми. Уже зараз людина окружає велику кількість технологій і розробок, які автоматизують процеси: додатки, боти, асистенти і так далі. Цей підхід також активно починає використовувати бізнес. В останній час все більше компаній переходять до автоматизації, а також починають залучати роботи для виконання деяких завдань. Завдяки цьому ефективність бізнесу зростає, а витрати на персонал знижуються.

Один з прикладів автоматизації — термінали для самообслуговування в магазинах. Це спеціальні пристрої, які дозволяють самостійно пробити товар на касі та розплатитися за нього. Це тренд, до якого будуть приєднатися все більше ритейлерів у майбутньому. Згідно з дослідженням компанії Zendesk, 67% клієнтів бажають самостійно обслуговувати себе, чим звертатися до представника компанії. Практика самообслуговування продовжить

розвиватися, і вже зараз можна спрогнозувати, як вона буде виглядати в майбутньому. У 2016 році Amazon представила магазин без кас і, відповідно, кассирів. Навіть не треба рахувати покупки — вони автоматично додаються у віртуальну корзину, коли клієнт бере товар у руки. Коли відвідувач взяв усі продукти, які йому потрібні, він може просто вийти з магазину, а гроші спишуться з карт.

Якщо брати до уваги роботизацію, цей підхід активно поширюється з виробництва, де роботи замінюють людей до виконання рутинної роботи. Наприклад, компанія Bosch створила роботизованого асистента APAS, який допомагає людині під час виконання дій, що повторюються. Людина може легко взаємодіяти з APAS без побоювань за своє життя - помічник зупиняється, коли знаходиться дуже близько до людей. Важливо також, що розробку Bosch можна навчити виконанню великої кількості завдань. І для цього не знадобляться навіть знання в програмуванні. Ще один приклад роботизації - автоматизація складів за допомогою спеціальних пристроїв. Компанія Symbotic виробляє самоврядних роботів, які пересуваються по складу та перевозять ящики із загального контейнера на сортувальні стрічки для подальшої доставки.

Якщо не брати до уваги скорочення витрат на персонал та підвищення ефективності виробництва, головне, що автоматизація дозволить компаніям – це швидко створювати індивідуальні продукти під кожного споживача. На сьогоднішній день переважає конвеєрний принцип виробництва. У цій моделі кожен співробітник відповідає за свою ділянку роботи та виконує лише його. Він рідко переходить на інші процеси та переучується. У такій концепції масового виробництва створюється багато варіацій товару, але вони призначені для абстрактної усередненої людини. Не саме для вас, а для людини схожої на вас. У майбутньому завдяки роботизації компанії зможуть створювати кастомні товари, оскільки роботи можуть бути перепрограмовані і перевчені за секунди. Перші зародки цієї тенденції видно вже зараз. Компанія Adidas активно будує нові фабрики Speedfactory. Це заводи

кросівок, де використовується мінімум персоналу, а деякі процеси, які традиційно виконували люди — зшивання, склеювання — зараз виконуються роботами. Завдяки цим фабрикам Adidas не лише радикально зменшить час для виробництва – з двох-трьох місяців до одного дня – а й зможе створювати унікальні кросівки для клієнтів.

Автоматизація та роботизація вплине не тільки на бізнес та професії людей, а й на світову економіку та політику. Важливо розуміти, що зміни, пов’язані з цими підходами, відбуватимуться не в усьому світі. Найбільш активне поширення автоматизація отримає лише в розвинених - США, Канаді, країнах ЄС - і деяких країнах, що розвиваються - наприклад, Китаї та Індії. Країнам важливо не прогаяти момент переходу до автоматизованого виробництва. Саме завдяки такому підходу в майбутньому їх виробничі потужності залишаться актуальними та конкурентоспроможними, що дасть змогу державам продовжити свій розвиток, а не назавжди залишитися країнами третього світу. У тому числі це стосується й України. У майбутньому розрив між країнами, які перейшли до автоматизації, та рештою стрімко збільшуватиметься.

1.1.4 Internet of Things

Нині новою технологією є Інтернет речей (IoT). У проекті IoT усі фізичні об’єкти (речі) з’єднуються разом за допомогою інфраструктури Інтернету. Плата Arduino є одним з найважливіших об’єктів у цій екосистемі. В першу чергу слід зауважити, що IoT не є технологією в прямому сенсі цього слова. IoT – це концепція, ідея обчислювальної мережі фізичних об’єктів, мають технології для взаємодії один з одним і / або з зовнішнім середовищем. На відміну від звичайного Інтернету, який лише спосіб об’єднання комп’ютерів, це – спроба вийти за його межі. Таким чином, IoT можна назвати наступним кроком розвитку «Фізичного» інтернету (Мається на увазі сам Інтернет, а не Web), мета якого - об’єднати всі системи заради

забезпечення додаткової вигоди для кінцевого споживача Найпоширенішим прикладом Інтернету речей є технологія «розумний будинок». «Розумними» називають будинки, в яких за безпекою, енергозбереженням і комфортом стежить програмне забезпечення, що об'єднує побутові прилади в єдину систему за допомогою технології передачі даних. Будинок самостійно управляє освітленням і вимикає світло в кімнаті, якщо в ній не знаходиться людина, стежить за протічкою водостічних труб і станом якості повітря, попереджає про можливості загоряння, автоматично поливає домашні рослини і годує домашніх вихованців, якщо Ви затрималися на роботі або виїхали у відрядження.

З появою Arduino ситуація дещо змінилася. Тепер таку систему може реалізувати кожен з мінімальними фінансовими вкладеннями. Arduino - це електронний конструктор і зручна платформа для швидкої розробки електронних пристроїв для новачків і професіоналів. Платформа користується величезною популярністю в усьому світі завдяки зручності і простоті мови програмування, а також відкритої архітектури і програмного коду. Пристрій програмується через USB без використання программаторов. Arduino дозволяє комп'ютеру вийти за рамки віртуального світу в фізичний і взаємодіяти з ним. Пристрої на базі Arduino можуть отримувати інформацію про навколишнє середовище за допомогою різних датчиків, а також можуть управляти різними виконавчими пристроями. Таким чином, можливість реалізувати найпростішу систему «розумного будинку» стає доступна кожному.

Система має клієнт-серверну архітектуру і побудована за принципом «товстий - тонкий клієнт». Таким чином, основну обчислювальну роботу, таку як збір і аналіз статистики, виконує віддалений web-сервер, а мобільний додаток і Arduino тільки відправляють і обробляють інформацію. Arduino (в ролі тонкого клієнта) підключається до сервера і оповіщає його про будь-які зміни (зміна рівня освітлення, температури, вологості грунту, стан повітря і ін). Сервер ж, в свою чергу, записує цю інформацію в базу даних і відправляє

її мобільному клієнту, де вона відображається зручному вигляді. Через мобільний додаток користувач може відправити команду на сервер, де вона буде оброблена і перенаправлено керуючим механізмам Arduino.

Дана розробка є комплексним вирішенням для управління «розумним будинком» на платформі Arduino. Розробляється система легко розгортається і масштабується в домашніх умовах, не вимагає специфічних навичок і досвіду програмування під мобільні платформи. Використовуючи дану розробку, користувач може в режимі реального часу отримувати актуальні показники датчиків, здійснювати віддалене управління з будь-якої точки земної кулі. Є можливість відстежувати зібрану статистику і, навіть вирушаючи у відрядження, бути впевненим в тому, що не забув вимкнути праску і закрити двері, а несподівана подія, таке як протікання водостічних труб, може бути миттєво усунуто.

1.2 Аналіз ринку існуючих пристроїв

В даний час існує кілька популярних моделей створення колісного робота на базі апаратної платформи Arduino UNO:

- двоколісний самокат, що самобалансується;
- робот, що рухається по чорній лінії;
- робот, що рухається на світ;
- робот, що уникає перешкоди, виявляє та відстежує
- об’єкт.

Більшість запропонованих рішень реалізують примітивні функції лінійного логічного умовного управління, не використовуючи методів регулювання. У зв’язку з цим рух таких роботів часто ламаний, з різкими змінами швидкості та напрямки руху, неоптимальний та неадаптивний..

1.2.1 Огляд основних функцій

У робототехніці два класи найбільш використовуваних роботів – маніпуляційні та мобільні роботи. Такі роботи успішно використовуються практично у всіх сферах життя людини: промисловості, будівництві, господарстві, військовій справі. Основними завданнями будь-якого робота є полегшення праці та зниження ризику життя людини.

Маніпуляційний робот містить дві органічно пов’язані частини: пристрій керування та маніпулятор. Пристрій управління включає чутливі (сенсорні) пристрої, пристрої обробки та зберігання інформації (обчислювальний пристрій, накопичувачі інформації), пристрій управління приводами. Маніпулятор з погляду механіки та теорії механізмів – це складний просторовий керований механізм з декількома ступенями свободи, що містить жорсткі та пружні ланки, передачі та приводи. Маніпуляційний робот є єдиною динамічною системою. Зважаючи на її складність при дослідженні доводиться виділяти і розглядати окремо механізми, приводи та систему управління.

Мобільний робот – це автоматична машина, оснащена деякими системами пересування та підсистемами (датчиками), що дозволяють роботу орієнтуватися у просторі, визначати сили, що впливають на нього у навколишньому середовищі.

Колісні роботи являють собою деякі платформи, що переміщаються за допомогою коліс. Їх число може змінюватись від одного колеса до стільки, скільки знадобиться для переміщення в деяких умовах. Найбільш поширеними є роботи на чотирьох колесах. Ця конструкція забезпечує оптимальні показники швидкості та стійкості. Двоколісні роботи у своїй роботі використовують кут нахилу корпуса робота. Для того, щоб дізнатися цей кут, застосовують різного виду гіроскопічні пристрої. Необхідність використання гіроскопів для утримання рівноваги пов’язана з динамікою зворотного маятника. Одноколісні роботи являють собою розвиток двоколісних роботів. Для переміщення по нерівних поверхонь таких як трава каміння і просто нерівності краще використовувати шестиколісних роботів.

Вони мають більше зчеплення порівняно з чотириколісним. Можна виявити такі плюси використання колісних роботів:

- високі швидкісні показники;
- легка зміна траєкторії руху;
- плавність та рівномірність руху.

Але, як і все існують такі недоліки:

- складність подолання перешкод із різким перепадом висоти;
- рухи лише у двох напрямках, що знаходяться у площині коліс;
- складність утримання горизонтальної орієнтації корпусу поверхні з різними рівнями висот.

1.2.2 Способи керування роботом

Однією з фундаментальних проблем мобільної робототехніки є автономне переміщення. Для виконання цього завдання робот повинен знати своє положення щодо становища своєї мети. Для ефективного функціонування інтелектуальні роботи забезпечені системою сприйняття довкілля, засобами аналізу ситуацій та здійснюють планування руху. При цьому виключно важливою є теоретико-механічна складова задачі, що включає дослідження загальних глобальних властивостей керованої системи.

Різноманітність систем управління роботами можна групувати за різними ознаками, наприклад: за складом і типом вхідних пристроїв, показниками якості управління, видом траєкторії руху тощо. Проте є досить загальні ознаки, які характеризують процес управління роботом. Насамперед це спосіб управління, який визначається залежно від ступеня участі оператора в управлінні роботом. За цією ознакою системи управління поділяються на два великі класи (Рисунок 1.1.):

- людино-машинні, до яких належать системи дистанційного та інтерактивного керування, що мають оператора безпосередньо в контурі керування;

- автоматичні, коли оператор залишається поза контуром управління та взаємодіє з роботом тільки на етапі навчання.

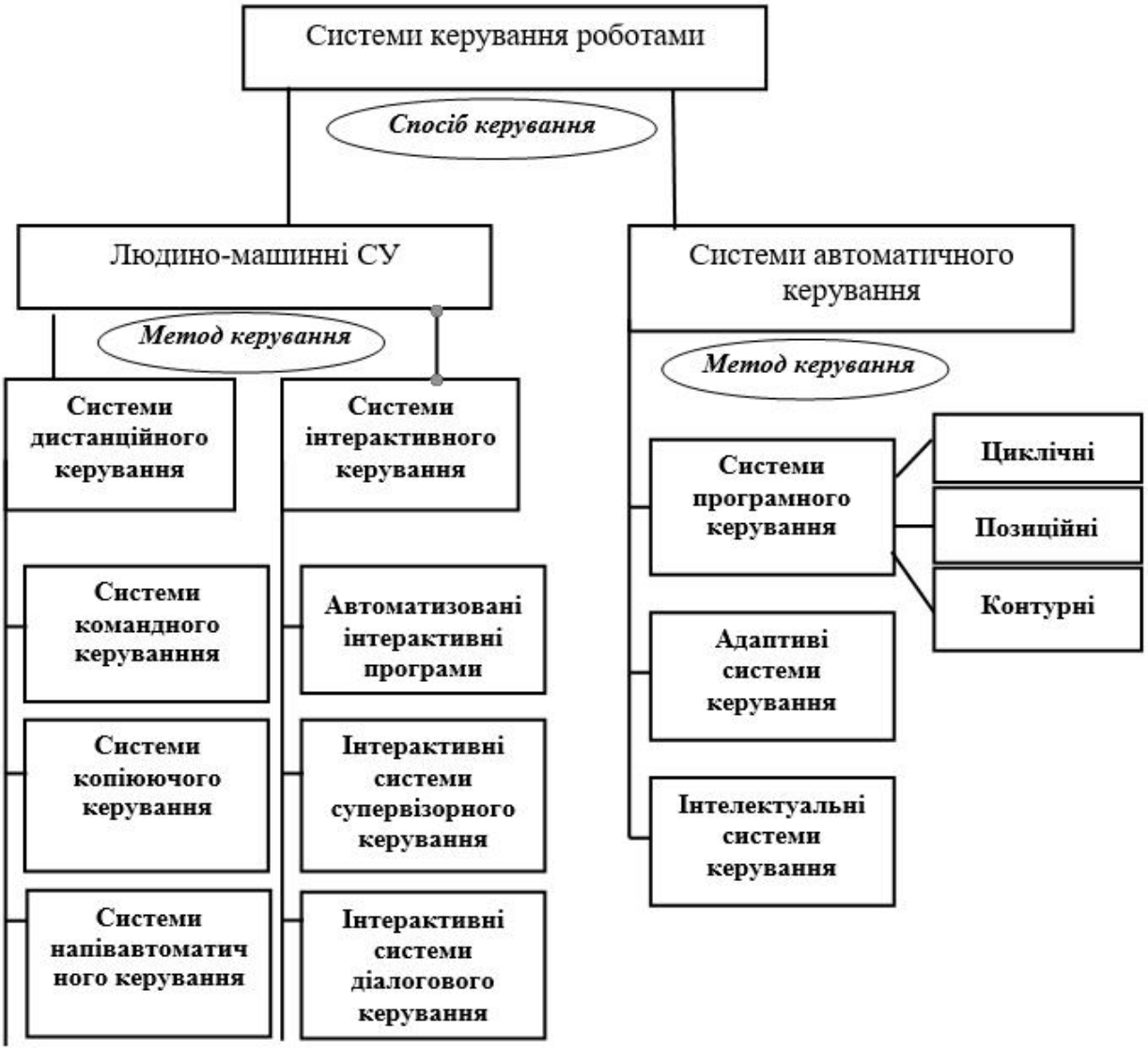

Рисунок 1.1 - Системи керування роботами

Також важливою ознакою є метод управління, який можна застосувати для подальшої класифікації виділених класів. Людино-машинні системи дистанційного та інтерактивного управління роботами відповідно до методів управління діляться на шість основних груп:

- Системи командного управління, в яких оператор включає окремо приводи кожної ланки робота дистанційно шляхом натискання на відповідні

кнопки (тумблери) пульта керування;

- системи копіюючого управління, в яких оператор дистанційно керує роботом за допомогою пристрою, що задає, кінематично подібного виконавчому пристрою робота (рух кожної ланки задає пристрою передається на відповідну ланку виконавчого пристрою за принципом системи, що стежить);

- системи напівавтоматичного керування, в яких оператор, натискаючи на багатоступеневу керуючу рукоятку, задає бажаний рух захватного пристрою, а спеціалізований обчислювач (ЕОМ) за електричними сигналами від датчиків рукоятки обчислює та формує відповідні сигнали керування для приводів усіх ступенів рухливості;

- автоматизовані інтерактивні системи управління, у яких лише частина операцій виконується автоматично, а інші надані оператору;

- інтерактивні системи супервізорного управління, в яких оператор, що спостерігає по екрану (дисплею) обстановку в місці дії робота, подає окремі команди - вказівки, за сигналами від них включаються ті чи інші програми автоматичної дії робота;

- інтерактивні системи діалогового управління, що відрізняються від інтерактивних систем супервізорного управління тим, що робот не тільки виконує команди оператора, але й активно допомагає йому в розпізнаванні обстановки та прийнятті рішень.

Головна особливість систем автоматичного управління роботами – відсутність безпосередньої участі людини у процесі управління. Функція оператора полягає лише у навчанні, запуску та подальшому періодичному спостереженні за роботою робота. Системи автоматичного управління відповідно до методів управління, що використовуються, діляться на системи:

- програмного управління, основою яких є синтез руху робота за заздалегідь розрахованою переважно жорсткою програмою. Програма зберігається в пам’яті обчислювального пристрою і може бути змінена

перепрограмування в новому циклі навчання робота. У системах програмного управління не передбачається відпрацювання інформації, що усуває невизначеність характеристик зовнішнього середовища, хоча інформація про внутрішній фазовий стан робота використовується в законі управління. У свою чергу, системи програмного управління поділяються на циклові, позиційні та контурні;

- адаптивного управління, рух робота в яких організується за гнучко змінюваними або коригованими програмами. У цьому перебудова програм відбувається у відповідь зміни умов довкілля. Для отримання зовнішньої інформації адаптивні системи управління забезпечуються різноманітними засобами відчуття;

- інтелектуального управління, у яких програма руху робота взагалі не задається, а синтезується системою управління на основі опису зовнішнього середовища, сукупності правил можливої поведінки у середовищі та наявної цільової установки задачі.

Основна відмінність інтелектуальних систем управління від попередніх - здатність вилучати з даних не лише інформацію, а й знання. З цією метою системи відчуття доповнюються системами розуміння (уявлення знань).

Найбільш «жорстким» управлінням мають роботи з програмним управлінням, а найбільш гнучким – роботи з інтелектуальним управлінням. Більшість експлуатованих роботів належить до першого покоління програмно-керованих автоматів. Головна перевага роботів з програмним управлінням полягає в тому, що вони мають широке застосування за досить простого конструктивного виконання. Найбільша ефективність їх проявляється в умовах монотонно-циклічних операцій при порівняно рідкісних переналагодження на новий вид робіт. Обсяг подібних операцій буде значним і в майбутніх виробництвах, тому з розвитком наступних поколінь промислових роботів потреба у простих програмних роботах не зменшиться. Вони успішно вдосконалюватимуться і надалі.

Головна особливість систем автоматичного управління роботами - їхня

автономна робота без безпосередньої участі людини. Функції оператора складаються лише в навчанні, запуску та подальшому періодичному спостереженні за роботою робота.

Закон управління у системі автоматичного управління реалізується як програми управління. При цьому можливі різні режими відпрацювання заданої програми: управління за жорсткою програмою без її перебудови в процесі роботи, адаптивна зміна програми в залежності від умов роботи, формування управління за відсутності заданої у явному вигляді програми.

Жорстка відпрацювання програми й у роботів першого покоління. Це найпростіший вид управління, особливо в реалізації його з урахуванням розімкнутого циклу регулювання. У складніших випадках застосовують замкнутий принцип управління. Наявність системи відчуття, що дозволяє роботу пристосовуватися до повністю визначеної обстановці під час виконання програми, відрізняє автоматичні роботи другого покоління, звані адаптивними.

Здатність сприйняття, розпізнавання довкілля, побудови її моделі, прийняття рішення про зміну поведінки під час виконання завдання уражає роботів третього покоління. Такі роботи прийнято називати інтелектуальними, оскільки вони технічно відтворюють окремі складні інтелектуальні функції, властиві людині.

Проте слід відрізняти штучний, чи машинний, інтелект роботів від природного інтелекту живих організмів. Маючи лише зовнішнім подібністю проявів, інтелектні функції робота реалізуються інших фізичних принципах і мають обмежені можливості. Разом про те навіть настільки досконалий технічний інтелект значно спрощує і автоматизує вирішення завдання синтезу управляючих програм робота, особливо у випадках, коли роботу доводиться діяти самостійно, не вдаючись до допомоги людини, як у роботі, і у процесі навчання.

Таким чином, складність розв’язуваних системою автоматичного управління задач є основою для поділу роботів на покоління. Роботи

першого покоління добре освоєні промисловістю. Знайдено економічно доцільні способи їхнього серійного виготовлення. Роботи другого і особливо третього покоління поки що є унікальними і, як правило, дорогими. Їх використовують під час роботи в екстремальних умовах, де праця людини неможлива. Водночас накопичено перший досвід промислових застосувань таких роботів, успішність якого дозволить їм швидко поширитися, насамперед, у точному машинобудуванні, електронній промисловості, приладобудуванні та низці інших галузей.

Порівняння різних систем автоматичного керування можна провести за такими характеристиками:

- тип траєкторії руху робота;
- цикл управління;
- джерела інформації для синтезу закону управління;
- алгоритм управління;
- спосіб програмування системи управління.

Циклові системи управління забезпечують лише дискретну траєкторію руху. Кожна її дискретна точка відповідає одній комбінації з наявної загальної кількості поєднань крайніх положень ланок маніпулятора. Позиціонування у крайньому становищі реалізується зазвичай механічним упором. Головний недолік дискретності траєкторії – неповна досяжність точок робочої зони робота.

Позиційні системи управління забезпечують квазінеперервну траєкторію. Це означає, що траєкторія, по суті, безперервна, але позиціонування можливе лише дискретних значеннях безперервної функції. Число дискретних значень залежить від можливостей керуючого пристрою і становить від кількох сотень до кількох тисяч точок.

Контурні, адаптивні та інтелектні системи управління забезпечують безперервні траєкторії руху, тобто можливість позиціонування у будь-якій точці робочої зони робота. Більше того, параметри руху (швидкість, прискорення, гальмування) можна змінювати у процесі переміщення

траєкторією.

Циклові системи управління мають розімкнений цикл управління, тобто сигнали управління надходять із програматора безпосередньо на привід робота. У позиційних системах цикл замкнений за точками позиціонування і розімкнуто для руху між точками позиціонування, яке відбувається за заданим законом. У контурних системах цикл управління замкнутий за внутрішньою інформацією про динамічний стан ланок робота, а адаптивні системи мають додаткові контури Для організації замкнутого циклу зовнішньої інформації про динамічний стан середовища. Інтелектні системи, крім того, володіють замкнутими циклами за внутрішніми та зовнішніми знаннями.

У циклових системах зазвичай використовуються інформаційні датчики. Виняток становлять кінцеві вимикачі, що фіксують знаходження ланок у крайніх положеннях. У позиційно-контурних системах кожна ланка забезпечується обов’язково датчиком положення, сигнал якого надходить у пристрій порівняння для формування сигналу неузгодженості між встановленими і поточними положеннями ланок. У адаптивних системах до датчиків внутрішньої інформації додаються датчики зовнішнього відчуття, а інтелектуальних системах засоби відчуття об’єднуються з базами знань.

У циклових системах алгоритм управління формується як логічної послідовності спрацьовувань ланок маніпулятора. У позиційних системах поруч із логічною послідовністю руху ланок існує ще закон інтерполяції, визначальний рух між точками позиціонування. У контурних, адаптивних та інтелектуальних системах алгоритм управління синтезується в результаті розв’язання диференціального рівняння, що описує динаміку робота.

Циклові системи програмуються шляхом фізичного налаштування механічних упорів та ручного набору циклограми на програматорі. Позиційно-контурні системи програмуються як безпосереднього навчання з використанням мови машинних кодів чи із застосуванням мов програмування високого рівня. Використовуються також спеціальні мови

програмування, що враховують специфіку робота. Для адаптивних та інтелектуальних систем характерні автоматизовані та автоматичні способи програмування.

1.3 Постановка задачі

Метою роботи є розробка системи програмного управління переміщеннями колісного роботу який в управлінні використовує інформацію від одометру датчика, тобто дані є у вигляді відстані що пройдена роботом.

Отримана системи повинен реалізовувати такі функції як. отримувати одометричні дані та виконувати точний рух за попередньо заданим маршрутом.

Роботу над пристроєм можна розділити на дві частини:

- робота над програмною реалізацією;
- безпосередньо робота з системою.

Обидва пункти важливі, так як вони грають важливу роль при виборі пристрою користувачем серед доступних на ринку. Ключовим фактором при роботі над другою частиною є швидкодія, точність і коректність роботи програми.

Надалі, при роботі над розвитком програми, можливо варто вдатися до використання більш точних геометричних розрахунків, але на даному етапі цього достатньо.

Процес роботи повинен бути швидким, точним і ефективним при низькій собівартості пристрою. В цьому і полягає актуальність даної роботи.

2 АНАЛІЗ ТА ВИБІР АПАРАТНОЇ ПЛАТФОРМИ

2.1 Апаратні системи типу Single-board computer

Одноплатний комп’ютер - самодостатній комп’ютер, зібраний на одній друкованій платі, на якій встановлені мікропроцесор, оперативна пам’ять, системи введення-виведення і інші модулі, необхідні для функціонування комп’ютера. На одній платі зібрано все мінімально необхідне для роботи. Як правило, це CPU, GPU, їх обв’язка і, можливо, USB і мережеві інтерфейси - як провідні, так і бездротові. Відеовиходи можуть бути різні: від застарілого VGA або композитного відео до цілком сучасного HDMI - аби вбудоване відеоядро підтримувало. Живлення відбувається від роз’єму USB і вимагає ток в діапазоні 500-1500 мА. Ідея зробити маленький комп’ютер лежить в умах інженерів вже багато років. Одним з найпопулярніших типів мікрокомп’ютерів є смартфон - симбіоз комп’ютера з його операційною системою та функціоналом і стільникового телефону. Поширення смартфонів на базі процесорів з архітектурою ARM, як виробників процесорів робити мініатюрні, але потужні обчислювальні системи, так і конструкторів електроніки розвиватися в напрямку одноплатних систем. Sistem-on-a-Chip (SoC), або ж Single-board computer (SBC) - англомовна назва цього типу комп’ютерів. Має два головних напрямки - одноплатні комп’ютери на Windows, з процесорами типу Intel Atom. Найчастіше це планшети. Одноплатні комп’ютери на процесорах з ARM архітектурою. Яскравим представником подібних пристроїв є приставки до телевізора «Android TV Box», планшети з ОС Android і подібними, одноплатні ПК подібні Raspberry Pi.

Кількість завдань, з якими впорається одноплатний комп’ютер, досить велика. Список починається домашнім комп’ютером і закінчується роутерами і модемами. Наприклад, на такий пристрій можна без проблем встановити операційну систему Linux, перетворивши його в пристрій для роботи з

документами, веб-серфінгу, прослуховування музики та інших нескладних завдань. Деякі моделі здатні відтворювати відеоролики в якості до 1080p зі звичним графічним інтерфейсом. Єдине місце, де можуть виникнути проблеми, - це інша архітектура. Як правильно, багато програм мають версію для ARM-архітектури, а якщо ні - то легко знайти аналог, але специфічний софт на цій платформі все ще трапляється рідко.

Інша сторона медалі - автоматизація і вузька спеціалізація. Так, для того ж Raspberry Pi існують прошивки, що дозволяють перетворити його в Wi-Fi роутер, мережеве сховище даних або бездротової мережевий плеєр. Дуже часто такі рішення є максимально завершеними і вимагають мінімальної метушні, вимагаючи від встановлює лише записати потрібний образ на носій даних (найчастіше - карту пам'яті) і вказати необхідні налаштування. Можливості воістину безмежні - ви можете спорудити цілий розумний будинок, використовуючи лише пару-трійку таких плат. Звичайно, заради цього доведеться попітніти з налаштуванням всіх таких пристроїв, але навіть тут існують готові рішення, які вимагають лише мінімальних зусиль.

Не варто також забувати, що набирає популярність явище під назвою «інтернет речей» (IoT). Наприклад, Microsoft збирається випустити спеціальну версію Windows 10 IoT Edition. Пристрій, що використовує спеціальні протоколи, зможе взаємодіяти з телефонами. З точки зору розвитку ідеї одноплатних комп'ютерів це дійсно крок вперед: до цього управління можна було здійснювати лише через маршрутизатор. Зараз ця ланка зникає, дозволяючи пристроїв спілкуватися безпосередньо, та ще й використовуючи звичні для цього методи - наприклад, для взаємодії з Arduino використовується відомий і популярний у всьому світі C.

2.2 Можливості Arduino у робототехніці

В даний час існує багато мікроконтролерів і платформ для управління фізичними процесами по відношенню до мікропроцесорних систем.

Більшість із цих пристроїв об’єднують різнорідну програмну інформацію в просту у використанні збірку. Компанія Arduino (Італія), в свою чергу, також спрощує процес роботи з мікроконтролерами, але надає ряд переваг перед іншими пристроями через простого і зрозумілого середовища програмування, низької ціни і різноманітності плат розширення. Для викладачів, студентів та аматорів платформа Arduino може стати ключовим елементом для досліджень і вирішення проблем у галузях мехатроніки та робототехніки.

2.2.1 Історія розвитку Arduino

На початку 2000-х створення прототипів електроніки було або дорогим, і складним, або й тим і іншим. Це створило проблему для Массімо Банзі та його студентів з Ivrea Interaction Design Institute, які не могли дозволити собі проекти мікроконтролерів BASIC Stamp за 100 доларів США, які тоді використовувалися. Отже, спираючись на роботу одного зі своїх учнів, Ернандо Барраги, Банзі та його колеги вирішили створити недорогу платформу, яка б полегшила створення електроніки.

Команда створила платформу Wiring Barragá, яка складалася з друкованої плати, мікроконтролера ATmega168 та IDE на основі Processing. Команда Банзі розширила проект і додала підтримку менш дорогого мікроконтролера ATmega8, щоб допомогти досягти цільової ціни на плату: набагато доступнішу 30 доларів. Ця ціна допомогла запустити Arduino – як команда назвала свою плату, після того, як паб Банзі та його команда часто відвідували – у потужний центр виробництва електроніки DIY, яким він є сьогодні. Хоча в новій моделі Arduino Uno Rev3 (Рисунок 1.1) використовується потужніший мікроконтролер ATmega328p, він зберіг ціну нижче 30 доларів, що дозволяє любителям, студентам і підприємцям вчитися, створювати та кодувати власні електронні пристрої без грошей. З тих пір, як була створена перша дошка, для спільноти DIY були доступні різноманітні зміни, доповнення та розширення платформи. Сюди входять Arduino Nano,

напіврозмірний варіант Uno, призначений для меншої електроніки, Uno Wi-Fi з вбудованим бездротовим підключенням і набір «щитів», щоб додати додаткову функціональність, як-от більше пам’яті та сховища, з’єднання Ethernet або елементи керування двигуном.

Для тих, хто не знайомий, асортимент плат може бути складним, але тут ми розглянемо основи Arduino, його варіації, як додати нові функції за допомогою щитів, а також набори, які постачаються з усім необхідним для початку.

Рисунок 2.1 – Arduino Uno

Arduino — це інструмент для проектування електронних пристроїв, які більш тісно взаємодіють із фізичним середовищем, ніж звичайні персональні комп’ютери, які фактично не виходять за рамки віртуальності. Це платформа, призначена для управління фізичними процесами за допомогою ЕСМ з відкритим кодом, побудована на простій друкованій платі з сучасним середовищем для написання програмного забезпечення. Плати Arduino побудовані на основі мікроконтролерів Atmel, а також зв’язкових елементів для програмування та інтеграції з іншими схемами. Лінійний стабілізатор напруги +5В або +3,3В розташований на платі. Синхронізація здійснюється

на частотах 8, 16 або 87 МГц за допомогою кварцевого генератора. Мікроконтролер попередньо прошивається завантажувачем, тому зовнішній програматор не потрібен.

На концептуальному плата програмується через A5-232. Інтегроване середовище розробки Arduino — це кросплатформний додаток Java, що включає редактор коду, компілятор і модуль передачі мікропрограми на плату. Середовище розробки засноване на мові програмування Processing і призначене для програмування початківцями, які не знайомі з розробкою програмного забезпечення. Строго кажучи, це мова C++, доповнена деякими бібліотеками. Програми обробляються за допомогою попереднього процесора, а потім копіюються за допомогою AVR-GCC. Переваги плат сімейства Arduino:

- велика кількість доступних опцій в лінійці Arduino з можливістю вибору найбільш підходящого готового контролера з великого списку пристроїв, які мають широкий діапазон змінних параметрів;
- наявність плат розширення, призначених для збільшення функціональності та виконання конкретних технічних завдань без необхідності самостійного проектування додаткової периферії (плати керування двигуном, плати датчиків, бездротових інтерфейсів, дисплеїв, пристроїв введення) - кілька десятків типів, більше 300 версій;
- середовище програмування, яке повністю адаптоване для кінцевого користувача і підходить для всієї лінійки плат Arduino та їх клонів, включаючи програмне забезпечення для програмування контролерів для Android;
- безкоштовна ліцензія на пристрої та програмне забезпечення.

Arduino Uno - це контролер, побудований на ATmega328. Платформа має 14 цифрових входів/виходів (6 з яких можна використовувати як ШІМ-виходи), 6 аналогових входів, кварцевий генератор 16 МГц, роз’єм USB, роз’єм живлення, роз’єм ICSP і кнопку скидання. Для роботи необхідно

підключити платформу до комп’ютера через USB-кабель або живити від адаптера AC/DC або батарейок.

Завдяки технічній оснащеності платформа Arduino ідеально підходить для навчального процесу проектування різних мехатронних систем і роботів завдяки чіткому середовищу програмування та можливості спостерігати фізичні процеси в реальному часі. Більш потужні плати Arduino (Due) застосовні для вирішення складних технічних завдань, пов’язаних з розробкою великих проектів та їх складною автоматизацією.

При подачі живлення на контролер Arduino автоматично починається виконання завантаженої в нього програми, а якщо програма відсутня або написана неправильно, виникає збій, який або зупиняє виконання команд, або викликає зависання програми. Номер програми, що виконується, зберігається в спеціальній комірці пам’яті, яка називається лічильником команд. Мова програмування, що використовується в проектах Arduino, заснована на C++. Це одна з найбільш широко використовуваних мов програмування, яка підтримує як низькорівневі команди, так і конструювання складних об’єктів.

На платформі Arduino вивчаються електронні компоненти, їх робота, підключення та програмування. Програмування здійснюється в Arduino IDE на C++ (з деякими модифікаціями). Отримавши базові знання з управління електронними компонентами, слухачі реалізують творчі проекти, складність і функціональність яких обмежені лише фантазією автора.

Arduino дозволяє учням познайомитися зі світом електроніки, зрозуміти принципи роботи електронних компонентів, побачити безліч різноманітних незвичних для них датчиків і пристроїв. Проекти, створені на цій платформі, можна використовувати вдома за призначенням (наприклад, автоматизація побутової техніки), що є додатковою мотивацією до навчання. Заняття з робототехніки на базі Arduino сприяють розвитку політехнічних компетенцій, необхідних у сучасній професійній діяльності в галузі автоматизації та IT.

Використання цієї платформи для навчальних закладів дозволяє розвивати навички програмування на практиці, а також вивчати основи схемотехніки.

Систематичний огляд літератури з даної теми виявив сильні сторони використання Arduino як допоміжного інструменту при підготовці студентів інженерного профілю спеціальних дисциплін з проектування мікроробототів. Ефективність навчання основам робототехніки залежить від організації занять, що проводяться з використанням таких методів: Пояснювально-ілюстративного викладу інформації різними способами (пояснення, розповідь, бесіда, інструктаж, демонстрація, робота з технологічними картами тощо); Евристичний метод творчої діяльності (формування творчих моделей тощо) Проблемний метод — це постановка проблеми та самостійний пошук її вирішення учнями; Програмований метод — це сукупність операцій, які необхідно виконати в ході практичної роботи (комп’ютерна майстерня, проектна діяльність); Репродуктивний метод – відтворення знань і способів діяльності (збір моделей і конструкцій за моделлю, бесіда, вправи за аналогом), Частково-пошуковий метод – розв’язання проблемних завдань за допомогою вчителя; пошуковий метод – індивідуальне вирішення проблеми; метод постановки проблеми - постановка проблеми вчителю, розв’язання проблеми вчителем, участь учнів у процесі розв’язування.

Предметом робототехніки є створення та використання роботів та інших робототехнічних засобів різного призначення. Виникнувши на основі кібернетики та механіки, робототехніка, у свою чергу, породила нові напрямки розвитку цих наук. Для кібернетики це пов’язано насамперед з інтелектуальним керуванням, яке потрібне роботам, а для механіків – за рахунок багатоланцюгових механізмів і маніпуляторів. Робота можна визначити як універсальну машину для виконання механічних дій, наприклад тих, які виробляє людина, яка виконує фізичну роботу. Від створення перших роботів і до сьогодні взірцем для них є фізичні можливості людини. Саме

прагнення замінити людину у важкій праці породило першу ідею робота, потім перші спроби її втілити і, нарешті, призвело до появи та розвитку сучасної робототехніки та робототехнічної промисловості.

Універсальність роботів передбачає можливість виконання цілеспрямованих дій, які потребують певних інтелектуальних здібностей. Це відкриває широкі можливості для використання роботів як основного технологічного обладнання (у таких видах робіт, як монтажні, зварювальні роботи, фарбування тощо), а також допоміжного обладнання – для заміни робітників, які займаються обслуговуванням. такого обладнання.

Як уже зазначалося, об’єктивною причиною виникнення та розвитку робототехніки стала історична потреба в сучасному виробництві гнучкої автоматики з відстороненням людини від безпосередньої участі в машинному виробництві та відсутністю традиційних засобів автоматизації для цієї мети. Тому завданням робототехніки поряд із створенням власне засобів робототехніки є розробка на їх основі систем і комплексів різного призначення. Системи та комплекси, автоматизовані за допомогою роботів, називають робототехнічними.

2.2.2 Модулі Arduino

Під час роботи з Arduino користувачі часто хочуть використовувати зовнішні модулі для різних завдань. Це можуть бути датчики для термометра, рідкокристалічний дисплей або підключення до абсолютно іншого комп’ютера чи плати розробника. Зазвичай всі вони однаково підключаються до Arduino, і ця інструкція представить чотири поширені способи підключення зовнішнього обладнання до вашої плати розробника.

Простіше кажучи, «Модуль» у середовищі Arduino — це готова до використання плата з датчиком або пристроєм введення на ньому. Щоб використовувати датчик або пристрій (наприклад, РК-дисплей), вам потрібно кілька додаткових компонентів. Модуль Arduino об’єднує всі необхідні

компоненти та з’єднання, необхідні для роботи датчика (або пристрою), і представляє вам готовий до використання продукт.

Перш ніж почати, користувачі повинні з’ясувати, який стандарт зв’язку використовує ваш зовнішній модуль. Найкраще подивитися таблицю або інструкцію пристрою (або подивитися на контакти введення/виводу на самому модулі). Пристрої, які можуть взаємодіяти через послідовне з’єднання, зазвичай мають контакти TX і RX, компоненти, сумісні з I2C, часто відкривають з’єднання SDA і SCL, а пристрої SPI повинні мати контакти з позначками MOSI, MISO і SCK (іноді також CLK або SCLK). Використовується для зв’язку між плату Arduino та комп’ютер або інші пристрої. Усі плати Arduino мають принаймні один послідовний порт (також відомий як UART або USART), а деякі мають кілька.

Наступним кроком після розуміння того, як підключитися до певного пристрою, є знання того, куди підключаються ці контакти на Arduino. Зверніть увагу, що на Arduino Uno, Nano, Mini і Mega контакти 0 і 1 використовуються для зв’язку з комп’ютером. Підключення до цих контактів може призвести до невдалого завантаження. Необхідно звертати увагу на те, що MISO, MOSI і SCK доступні в однаковому положенні майже на всіх Arduinos, щоб дозволити дизайн щитів, які працюють на кожній підтримуваній платі. Як згадувалося, послідовний порт - це те, що використовує комп’ютер для зв’язку з Arduino. Він також може надсилати повідомлення на Arduino і виробляти вихідні дані для налагодження. Однак можна використовувати послідовний порт для зв’язку з іншими платами та модулями розробників. Як видно вище, дві плати Arduino підключаються через їхній послідовний порт.

У протоколі зв’язку I2C (Рисунок 1.2) кожен підключений пристрій повинен мати унікальну адресу, яку зазвичай можна знайти в його таблиці. Arduino IDE постачається зі стандартною бібліотекою, яка допомагає в цьому процесі. Зауважте, що в деяких випадках вам знадобляться підтягуючі резистори на лініях SDA і SCL, щоб підтягнути їх високо, коли вони не

використовуються. Однак у багатьох модулях вже є відповідні резистори.

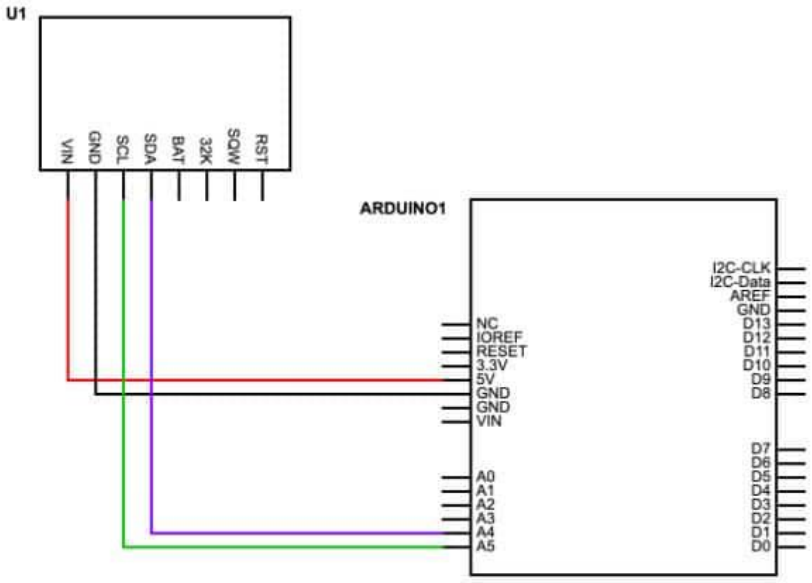

Рисунок 2.2 Підключення за протоколом I2C

У SPI (Рисунок 1.3) завжди є головний пристрій, який керує одним або кількома зовнішніми підлеглими. Як згадувалося вище, зазвичай для цього протоколу потрібні три рядки. Лінія MISO використовується підлеглими для відправки даних на головний пристрій. MOSI призначений для відправки даних до ведених, а сигнал CLK синхронізує всі компоненти. Однак для кожного периферійного пристрою є додаткова лінія SS, що означає підпорядкований вибір. Це дозволяє майстрові вибрати активний периферійний пристрій, щоб дозволити кільком пристроям використовувати одні й ті ж три лінії.

Користувачам також доведеться з’ясувати, яка максимальна швидкість передачі їх модуля, чи мають вони надсилати дані в зворотному порядку (спочатку LSB), і як повинен вести себе годинник. Потім вони повинні передати ці значення стандартній бібліотеці Arduino SPI, коли ви хочете спілкуватися з модулем.

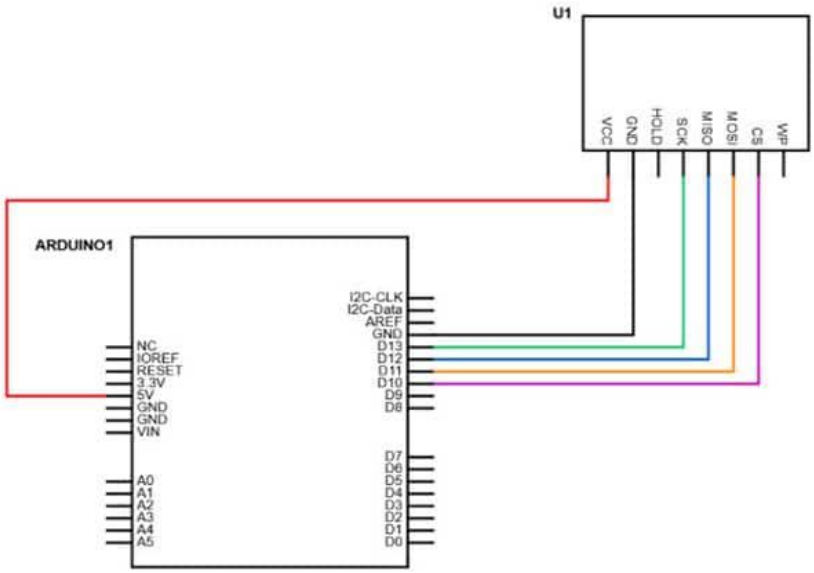

Рисунок 2.3 Підключення зовнішнього модулю до контактів SPI

Як згадувалося на початку, є також четвертий спосіб підключення периферійних пристроїв до вашого Arduino. Деякі пристрої не використовують жоден із методів, які обговорювалися досі. У цих випадках таблиця даних пристрою надасть конкретну інформацію про те, які з’єднання необхідно встановити і як реалізувати сам протокол зв’язку.

2.2.3 Аналіз існуючих моделей Arduino

Сьогодні існує безліч чудових багатофункціональних платформ налагодження, які можуть використовуватися для розробки цифрових додатків на мікроконтролерах, і цей сегмент ринку постійно розвивається. На даний момент офіційно представлено 22 різні моделі платформи Arduino. Є кілька версій платформ Arduino. Остання версія Leonardo базується на мікроконтролері ATmega32u4. Uno, як і попередня версія Duemilanove, побудовані на мікроконтролері Atmel ATmega328. Старі версії платформи Diecimila і перша робоча Duemilanoves були розроблені на основі Atmel ATmega168, раніше версії використовували ATmega8. Arduino Mega2560, своєю чергою, побудована на мікроконтролері ATmega2560.

Arduino Uno є стандартною платою Arduino і, можливо, найпоширенішою. Вона заснована на чіпі ATmega328, що має на борту 32 КБ флеш-пам’яті, 2 Кб SRAM та 1 Кбайт EEPROM пам’яті. На периферії має 14 дискретних (цифрових) каналів введення/виведення та 6 аналогових каналів введення/виведення, це дуже різнобічно-корисні девайси, що дозволяють перекривати більшість аматорських завдань у галузі мікроконтролерної техніки. Максимальна довжина та ширина друкованої плати Uno складає 6.9 см і 5.4 см відповідно, з урахуванням роз’єму USB та роз’єму живлення, що виступають за межі плати. Чотири отвори кріплення дозволяють прикріплювати плату до поверхні або корпусу. Дана плата контролера є однією з найдешевших і найчастіше використовуваних.

Arduino Leonardo (Рисунок 2.3) це та сама Arduino Uno, але з іншим мікроконтролером, який знаходиться в тому ж класі, але має деякі відмінності позитивного характеру. Більше аналогових входів (12 проти 6) для сенсорів, більше каналів ШІМ (7 проти 6), більше пінів з апаратним перериванням (5 проти 2), роздільні незалежні serial-інтерфейси для USB і UART. Arduino Leonardo може вдаватися клавіатурою або мишею (HID-пристроєм) для комп’ютера. Це дозволяє легко зробити власний пристрій введення. Через розпинування трохи відмінне від Arduino Uno можлива несумісність з деякими платами розширення.

Рисунок 2.3 Arduino Leonardo

Arduino Nano це функціональний аналог Arduino Uno, але розміщений на мініатюрній платі. Відмінність полягає у відсутності свого гнізда для зовнішнього живлення, використанням чіпа FTDI FT232RL для USB-Serial перетворення (або CH340G, потрібно встановити відповідні драйвера) та застосуванням mini-USB кабелю для взаємодії замість стандартного. В іншому, начинка та способи взаємодії збігаються з базовою моделлю. Платформа має штиркові контакти, що дозволяє легко встановлювати її на макетну плату. Використовуйте Arduino Nano там, де важлива компактність, а можливостей Arduino Mini або мало, або не хочеться займатися пайкою.

Arduino Mega - як Arduino Uno, але на базі потужнішого мікроконтролера тієї ж архітектури. Відмінний вибір на виріст або якщо Arduino Uno перестала справлятися. У рази більше пам’яті: 256 КБ проти 32 КБ постійної та 8 КБ проти 2 КБ оперативної. У рази більше портів: 60 з них 16 аналогових та 15 з ШІМ. Трохи довше за базову Arduino Uno: 101×53 мм проти 69×53 мм.

Arduino Due (Рисунок 2.4) одна з найпродуктивніших плат від Arduino на мікроконтролері Cortex-M3 за формами-фактором аналогічна Arduino Mega. Процесор на 84 МГц та 512 КБ пам’яті. 66 пінів введення-виведення, з яких 12 можуть бути аналоговими входами, 12 підтримують ШІМ і всі 66 можуть бути налаштовані, як апаратні переривання. Вбудований контролер шини CAN дозволяє створювати мережу Due або взаємодіяти з автомобільною електронікою. Два канали ЦАП дозволяють синтезувати стереозвук з роздільною здатністю 4,88 Гц. Рідною напругою для плати є 3.3 В, а не традиційні 5 В. Необхідно стежити, щоб периферія, що вибирається, підтримувала роботу з цим рівнем або ставити перетворювачі рівнів напруги.

Arduino Mini та сама Arduino Uno, але в іншому форм-факторі. Компактна: лише 30×18 мм. Через форм-фактор не можна без хитрощів встановлювати плати розширення Arduino. Передбачається з’єднання з додатковими модулями проводами та/або через макетну плату. На платі немає USB-порту, тому потрібно прошивати через окремий USB-Serial

адаптер.

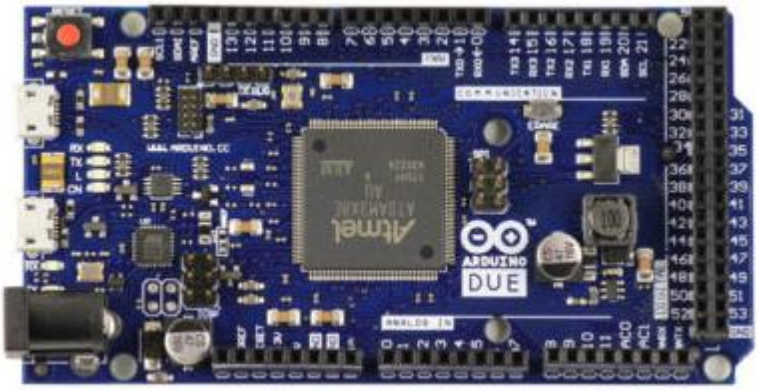

Рисунок 2.4 Arduino Due

Arduino Micro – це Arduino Leonardo, виконаний на компактній платі. Відмінність полягає у відсутності власного гнізда для зовнішнього живлення, але може бути підведено безпосередньо до контакту Vi. В іншому, начинка та способи взаємодії збігаються з Arduino Leonardo. Він також має один мікроконтролер ATmega32u4 як для прошивки через USB, так і для виконання програм; також може виступати в ролі клавіатури чи миші; надає ту ж кількість пам’яті, цифрових, аналогових та ШІМ-портів.

З платою Arduino M0 (Рисунок 2.5) виконувати складні математичні розрахунки, отримувати більш точні аналогові значення і слухати музику безпосередньо з мікроконтролера. Arduino M0 заснована на 32-бітному ARM-процесорі ATSAMD21G18 від Atmel з обчислювальним ядром Cortex M0. Мікроконтролер працює на частоті 48 МГц. А завдяки своїй 32-бітній архітектурі він виконує більшість операцій над цілими числами лише за один такт. На відміну від більшості плат Arduino, рідною напругою Arduino M0 Pro є 3.3, а не 5 В. Відповідно, виходи для логічної одиниці видають 3.3, а в режимі входу очікують приймати не більше 3.3 В. Arduino M0 дивиться в бік USB через віртуальний serial-порт, не через апаратний. Це означає, що 0-й та 1-й контакти апаратного порту залишаються вільними і ви можете

використовувати їх одночасно з комунікацією з комп’ютером. Віртуальний serial-порт доступний через об’єкт SerialUSB, а апаратний - через об’єкт Serial1.

Arduino LilyPad – досить цікавий пристрій. Він випадає зі звичних стереотипів про звичайне Arduino, тому що має прямокутну, а круглу форму. По-друге, воно не підтримує механічні з’єднання із шилдами. Воно призначене для невеликих автономних пристроїв. Кругла форма продиктувала те, що роз’єм рівномірно розподілені по колу, і його невеликий розмір (2 дюйми в діаметрі) робить його ідеальним для переносних пристроїв. Цей пристрій легко сховати, і кілька виробників розробили пристрої, спеціально для LilyPad: екрани, датчики світла, навіть коробки батарей живлення, які можуть бути зашиті в тканину. Для того, щоб зробити LilyPad якнайменше і якомога легше, наскільки можливо, були принесені деякі жертви. LilyPad не має регулятора напруги на борту, так що йому для живлення буде необхідно забезпечити принаймні 2.7 вольт, і не більше 5.5 вольт.

Рисунок 2.5 Arduino M0

2.2.3 Характеристики й методи використання одометру в робототехніці

Одометр буває:

- механічним;
- електромеханічним;
- електронним.

Електронні одометри мають деякі переваги перед іншими типами одометрів. Великим плюсом електронного одометру є те, що він не обнуляється як це відбувається у механічоному типі. При повному оберті коліщат механічного одометру дані обнуляються. Зазвичай вони обежуються показаннями в 999 тис. км.

Користувачі дуже часто сприймають спідометр і одометр за один прилад тому, що найчастіше показники одометру розташовані поряд або на шкалі спідометра. Одометр може відображати показники загального кілометражу та денний пробіг. При необхідності є можливість обнулити показання.

В роботі одометр представлений у вигляді інфрачервоного датчику H206 та диском з 20 отворами, що закріплений на осі двигуна. Принцип його роботи заключаєтьмя в тому, що датчик H206, з диском розміщеним між його зубцями, реагує на зміну сигналу при кожному проходженні отворів на диску між зубцями. Так, як диск має 20 отворів, і реагування датчика відбуваєтьмя на будь-яку зміну сигналу то з цього слідує, що колесо зробить повний оберт після 40 реагувань датчика.

Відстань, пройдена за певний проміжок часу (кілометраж) та швидкість разраховуються на підставі даних, отриманих від одометру.

В роботі платою Arduino буде виявлено 40 переривань за час повного обороту колеса. З цього слідує, що пройдена роботом відстань за один оберт колеса, є довжина кола цього колеса. Так, як радіус колеса відомий, пройдена дистанція може бути розрахована за формулою:

$$S = 2\pi r * (\frac{intr}{40}), \quad (2.1)$$

де r – радіус колеса у метрах, а intr – це кількість повних обертів колеса.

Плата Arduino має можливість за допомогою функції `millis()` визначати час. Починаючи з часу подачі живлення на плату Arduino значення функції збільшується кожну мілісекунду на 1. Це дозволяє нам зберегти значення функції `millis()` у змінній при виклику цієї функції на перше виникле переривання, і за допомогою віднімання значення змінної при другому перериванні ми маємо можливіть розрахувати час з поточного значення функції `millis()`.

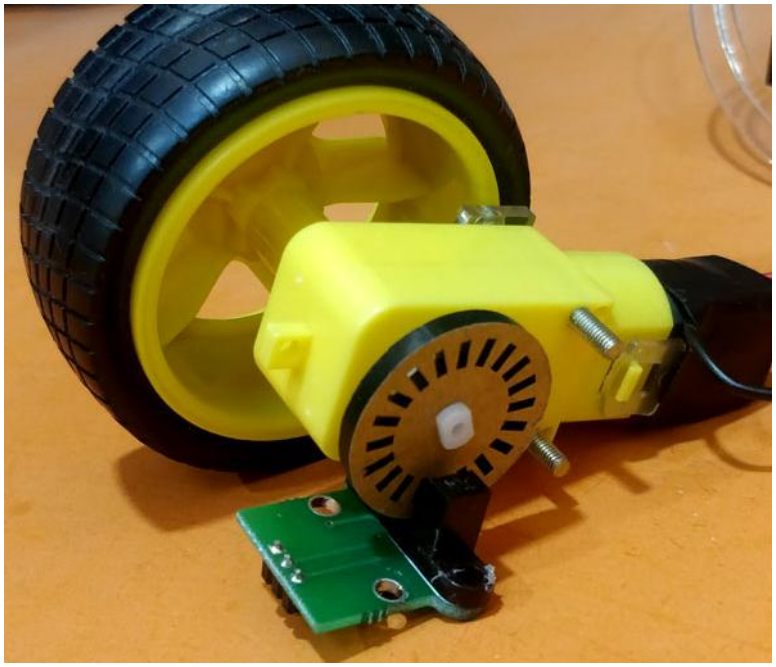

Рисунок 2.6 – Апаратна реалізація датчика одометра

Time taken = current time – previos time (2.2)

```
timetaken = millis() – pevtime
```

Якщо нам відомий час між двома перериваннями то ми маємо можливість знайти число обертів колеса в хвилину (rpm) по наступній приведеній формулі. Даною представленою формулою (1000/*timetaken*) ми можемо розрахувати кількість обертів колеса в секунду (RPS - Revolutions per second) і помножити це число на 60 для перетворення RPS в RPM (Revolutions per minute – число обертів в хвилину).

$$rpm = (1000/timetaken)*60 \quad (2.3)$$

Коли нам відоме число обертів колеса в хвилину (rpm) ми маємо можливість знайти швидкість робота завдяки відомому радіусу колеса по наступній формулі: *Velocity = 2πr * RPS * radius of wheel*

$$v = radius\ of\ wheel * rpm * 0.104 \quad (2.4)$$

Формула що представлена вище може розрахувати швидкість в м/с. Після спрощення вираження *V = 2π * RPS * radius of wheel* вийшло число 0.104.

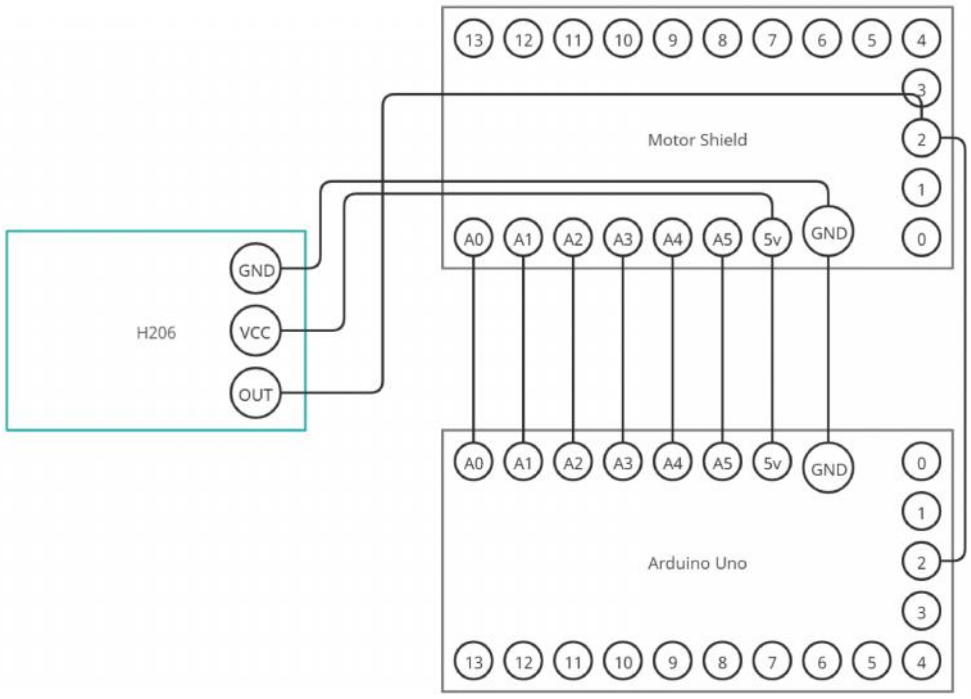

Рисунок 2.7 – Схема з’єднання компонентів на базі датчика H206

Електрична схема з’єднання компонентів керування роботом зображена на рисунку 2.7. Конструкція Motor Shield передбачає з’єднання типу «тато» під кожним піном тому, що це плата розширення. Завдяки цим

зв’язнам Motor Shield може бути приєднана до плати Arduino Uno, яка має зв’язки типу «мама». Завдяки цьому Motor Shield представляє доступ до усіх пінів плати Arduino Uno. Датчик H206 конструктивно має три виходи: GND, VCC, OUT. На схемі зображено приєднання цих виходів до відповідних пінів на Motor Shield.

3 ОПИС СТРУКТУРИ ТА АЛГОРИТМІВ РОБОТИ

Розглядаються нові методи управління та навігації роботами, здатними до автономного поводження в умовах недетермінованої робочої середовища. Діяльність людини-оператора зводиться до спостереження за функціонуванням робототехнічної системи і до постановки потокових завдань. При цьому додається зворотна зв’язок, яка може бути і мовною. Таким чином, завдання управління набуває характеру діалогу людини і робота, що супроводжується графічними та мовними повідомленнями. Істотну роль при цьому відіграє система навігації, оскільки робот повинен самостійно оцінювати навколишню обстановку і планувати свій шлях, у тому числі і за наявності інших рухомих об’єктів у робочій зоні. Автоматичне рішення цих завдань істотно полегшує завдання оператора, але потребує розробки «інтелектуальної» системи управління автономним роботом. До числа таких завдань належить і задача автоматичного повернення робота при втраті зв’язку з оператором, рішення якої підвищує надійність робототехнічної системи. Зміна характеру діяльності оператора, який тепер не керує безпосередньо рухами робота, призводить до зміни характеру системи управління, оскільки вона повинна враховувати можливості сприйняття оператора та характер прийнятих ним рішень. Одним із шляхів вирішення цієї задачі є застосування нечіткої логіки як на етапі сприйняття інформації, так і на етапах планування дій та прийняття оперативних рішень. Застосування «природних» просторово-часових відносин і лінгвістичних змінних ще більше наближає процес управління до діалогу оператора з роботом, що дозволяє визначити робототехнічну систему такого типу як систему кооперативного управління. На практиці більшість завдань, що вирішуються автономними мобільними роботами, таких як моніторинг місцевості, радіаційна та хімічна розвідка, боротьба з пожежами та стихійними лихами вимагають участі групи мобільних роботів. Деякі

результати, що стосуються групового управління роботами, також представлені в роботі.

3.1 Інтегроване середовище розробки

IDE (або інтегроване середовище розробки) - це програма, призначена для розробки програмного забезпечення. Як випливає з назви, IDE об’єднує кілька інструментів, спеціально призначених для розробки. Ці інструменти зазвичай включають редактор, призначений для роботи з кодом (наприклад, підсвічування синтаксису і автодоповнення); інструменти збірки, виконання та налагодження; і певну форму системи управління версіями. Більшість IDE підтримують безліч мов програмування і мають багато функцій, через що можуть бути великими, займати багато часу для завантаження і установки і вимагають глибоких знань для правильного використання. З іншого боку, є редактори коду, які представляють собою текстовий редактор з підсвічуванням синтаксису і можливостями форматування коду. Більшість хороших редакторів коду можуть виконувати код і використовувати відладчик, а кращі навіть можуть взаємодіяти з системами управління версіями. У порівнянні з IDE, хороший редактор коду, як правило, легковагі і швидше, але часто ціною меншою функціональності. Набір функцій різних середовищ може відрізнятися, але є набір базових речей, що спрощують програмування:

- збереження файлів. Якщо IDE або інший редактор не дають можливості зберегти роботу і пізніше все відкрити в тому ж стані, в якому воно було під час закриття, то не дасть змогу працювати;
- запуск коду з середовища. Те ж саме, якщо потрібно вийти з середовища для запуску коду, то це не більше, ніж простий текстовий редактор;
- підтримка налагодження. Можливість крок за кроком виконати код є базовою функцією всіх IDE і більшості хороших редакторів коду;

- підсвічування синтаксису. Можливість швидко знайти ключові слова, змінні та інше робить читання і розуміння коду на порядок простіше;

- автоматичне форматування коду. Будь- який редактор або IDE, який дійсно таким є, розпізнає двокрапку після while або or вираження і автоматично зробить відступ на наступному рядку.

Зрозуміло, є безліч інших функцій, від які були б тільки на користь, але наведені вище - основні функції, якими повинеа володіти комфортне середовище розробки.

Arduino - це інструмент для тестування електронних пристроїв, якы більш щільно працюють з навколишнього фізичного середовища, ніж стандартні комп'ютери. Це платформам, призначена для керування фізичними процесами з використанням ЕВМ з відкритим програмним кодом, побудована на простій печатній платі с сучасним середовищем для написання програмного забезпечення. Плати Arduino будуються на основі мікроконтролерів фірми Atmel, а також елементів обв'язки для програмування і інтеграції з іншими схемами. На платах присутній лінійний стабілізатор напруги +5 В або +3,3 В. Тактування здійснюється на частотах 8, 16 або 87 МГц кв. У мікроконтролер попередньо прошивається завантажувач, тому зовнішній програматор не потрібний. На концептуальному рівні всі плати програмуються через RS-232. Інтегроване середовище розробки Arduino - це кросплатформовий застосунок на Java, що включає в себе редактор кода, компілятор і модель передачі прошивки на плату. Середовище розробки засноване на мові програмування Processing і спроектована для програмування новачками, близько не знайомими з розробкою програм, це - мова C++, ще доповненена деякими бібліотеками.

Найпершою, а, найчастіше, і єдиною програмою для початківців працювати з контролером Ардуїно стає IDE Arduino (рисунок 3.1) – інтегроване середовище розробки від творців платформи, застосовується на Windows, MacOS та Linux. Це середовище розроблено на Сі та C++. Призначена вона для створення та завантаження програм на Arduino-сумісні

плати, а також на плати інших виробників, для розробки прошивок та завантаження їх у пам’ять мікроконтролера. Середовище розробки Arduino IDE складається із набору специфічних бібліотек та драйверів, необхідних для коректної роботи з контролером. За допомогою вбудованих бібліотек завжди можна підключити додатковий датчик або дисплей до контролера. Можна перевірити код на наявність помилок. Програма вкаже точне місце помилки в коді або де було введено неправильні дані. Якщо їх не виправити, програма просто не зможе скомпілювати код, і не вдасться залити скетч в контролер.

Перша версія середовища розробки Arduino офіційно з’явилася у серпні 2005 року. З того часу програма кілька разів зазнавала революційних змін. До 2011 року номери версій складалися з однієї цифри (останньою була 0023 – версія від 09.11.2011). Потім з’явився звичніший запис з «мінорними» та «мажорними» оновленнями. Так, Arduino 1.0.0 з’явилася 30.11.2011, а версія 1.0.1 – у травні 2012 року. Ключовими за кількістю оновлень та пов’язаних з ними проблем сумісності стали переходи з версії 1.0.7 та 1.5. Багато бібліотек, написаних під «старі версії» програми не могли компілюватися в нових через зміни у системних бібліотеках та інших внутрішніх механізмах складання проекту. Особливо це виявилося при переході на 1.6 версію. Якщо у вашому проекті є стародавні бібліотеки, які некоректно працюють із сучасними версіями IDE, вам доведеться змінювати їх самому, чекати, поки це зроблять автори або підкоряться обставинам і встановити ту версію, з якою буде працювати бібліотека.

Середовище розробки Arduino є текстовим редактором програмного коду, область повідомлень, вікно виведення тексту (консоль), панель інструментів і кілька меню. Для завантаження програм та зв’язку середовище розробки підключається до апаратної частини Arduino. Нову версію завжди можна знайти на офіційному сайті arduino.cc.

Меню "Файл". Опустивши очевидні пункти меню, хотілося б згадати такий пункт, як Папка зі скетчами. За замовчуванням Arduino IDE зберігає

кожен скетч окрему папку. Ім’я папки збігається з ім’ям, вказаним для скетчу під час збереження. Змінити робочу директорію папок зі скетчами можна в пункті меню “Налаштування”.

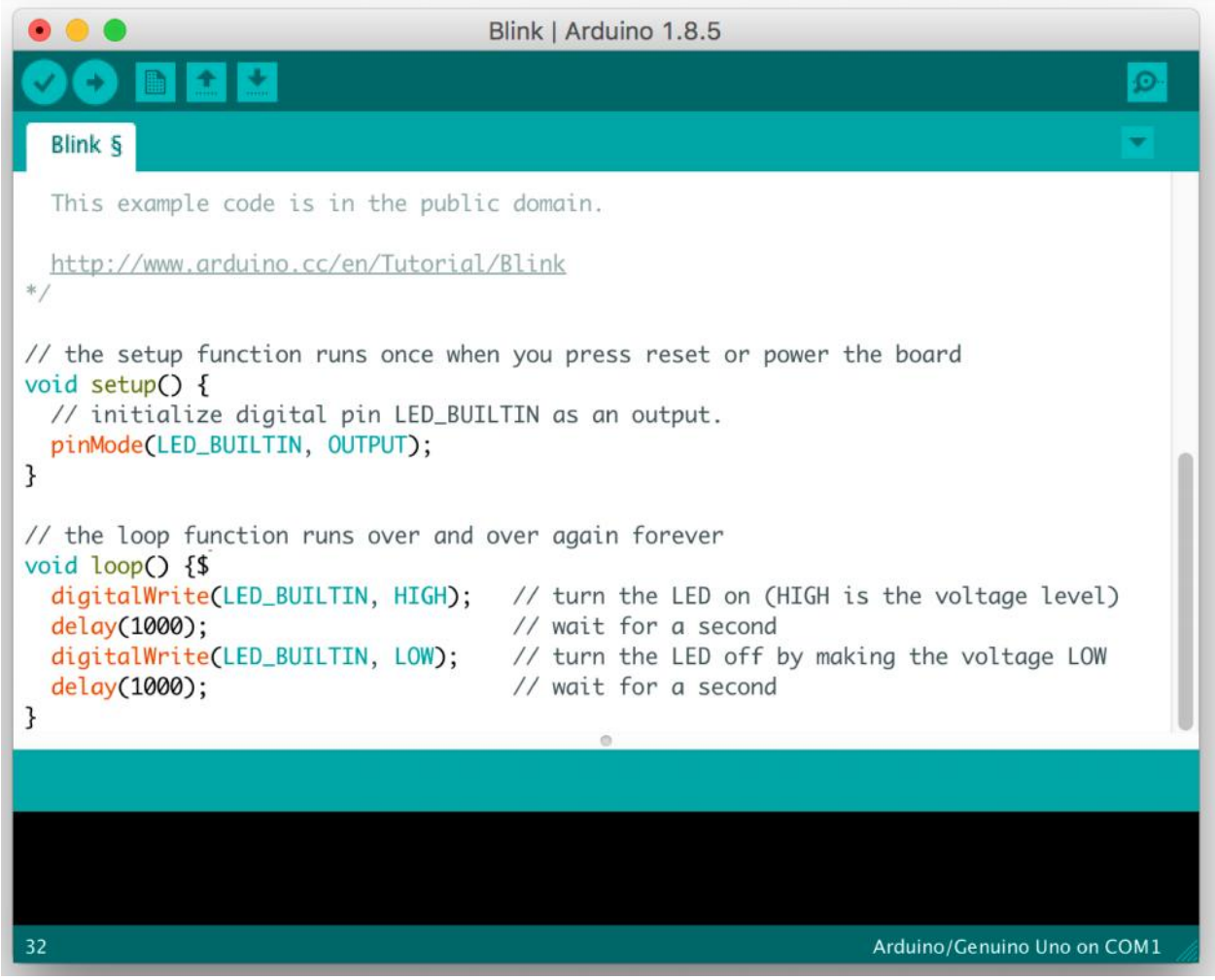

Рисунок 3.1 – Інтерфейс Arduino IDE

У меню "Редагування" розташовані команди для роботи з кодом вашої програми. Часті команди зручні наявністю комбінацій для швидкого доступу за допомогою клавіатури. Зручними функціями є можливість копіювання для форумів і в HTML форматі, що дозволяють ділиться вашими скетчами, зберігаючи наочність розмітки у вигляді BB кодів або HTML розмітки відповідно.

Далі меню "Скетч". В даному меню продубльована команда з панелі

керування "Перевірити/Компілювати". Виконання якої призведе до перевірки вашого коду на помилки, а у разі їх відсутності – до компіляції.

Пункт меню "Показати папку скетчів" відкриє робочу директорію Arduino IDE, вказану в налаштуваннях. "Додати файл..." дозволяє відкрити текстовий файл (або скетч) в окремій вкладці. Окремо хотілося б зупинитись на імпорті бібліотек. Arduino IDE містить безліч попередньо встановлених бібліотек. Їх список, а також довідкове керівництво англійською можна знайти на arduino.cc.

Бібліотеки додають додаткову функціональність скетчам, наприклад, при роботі з апаратною частиною або обробкою даних. Одна або кілька директив #include будуть розміщені на початку коду скетчу з подальшою компіляцією бібліотек та разом зі скетчем. Завантаження бібліотеки потребує додаткового місця в пам’яті Arduino. Для встановлення сторонніх бібліотек можна скористатись командою “Імпортувати бібліотеку…”/”Add Library…”.

В якості альтернативи можливе додавання бібліотек безпосередньо до робочої директорії до папки libraries. При такому підході необхідно перезапустити IDE. У меню “Сервіс” необхідно вказати модель вашої Arduino плати, а також COM-порт, до якого вона підключена.

Зручною функцією є автоформатування, яка дозволяє виправити огріхи в розмітці скетчу і привести його в легкозчитуваний вигляд. Особливо актуально під час копіювання сторонніх програм.

Середовище Arduino IDE дозволяє залити bootloader на atmega мікроконтролери. У меню “Программатор” вказується використовуваний пристрій. Команда “Записати завантажувач” здійснює заливку бутлоадера для зазначеної моделі плати. Пункт меню “Монітор порту” викликає вікно обмінюватись повідомленнями з arduino через COM порт.

Меню “Довідка”: Пункти меню ведуть на відповідні статті на офіційний сайт (англійська). Команди панелі управління, як бачимо, дублюють найбільш актуальні пункти меню. Програми пишуться мовою C++ і називається скетч. Скетч набирається в текстовому редакторі, який має всі

необхідні інструменти, такі як: скасування останніх дій, вирізка/копіювання/вставка, пошуку тексту з можливістю заміни, автоматичного форматування тексту в формат легко читати. Під час збереження, експорту, компіляції проекту, а також завантаження його в плату, в області повідомлень з’являються повідомлення про помилки, пояснення та інше. Монітор серійного порту показує повідомлення, отримані від Arduino, наприклад стан змінних, показання датчиків. Крім того, через нього можна передавати інформацію на плату для управління Arduino. Кнопки панелі інструментів дозволяють перевірити та записати програму, створити, відкрити та зберегти скетч, відкрити моніторинг серійного порту:

- перевірка скетчу на помилки та його компіляція;
- завантаження скетчу в плату та його виконання;
- створення нового скетчу;
- відкриває збережений скетч з папки на комп’ютері;
- зберігає раніше написаний скетч до папки комп’ютера.

Монітор послідовного порту відкриває нове вікно з інформацією, що отримується з Arduino. При натисканні кнопки скетч в арудино виконується спочатку. Через нього за допомогою функції Serial ви зможете передавати інформацію з плати на комп’ютер і назад у режимі поточного часу.

Додаткові можливості зібрані у п’ять груп меню: Файл, Правка, Ескіз, Інструменти, Допомога. Можливість скористатися командою меню визначається роботою, яка виконується в даний момент.

3.2 Розробка програмної частини

Аналіз керуючої програми для прямих підключень показує, наскільки вимогливий метод для попереднього замовлення середовища: обов’язкові початкові умови (строго певний початковий стан роботи) і суворі вимоги до швидкості і миттєвим режимам прискорення/уповільнення робота. Якщо перша вимога може бути виконана певною організацією, то друга майже

ніколи не виконується. У програмному методі керування зворотним зв’язком певні позиції використовують інформацію від датчиків для ідентифікації вихідної події, а не фокусуються на часовому інтервалі. Як наголошується в постановці завдання, управління використовує інформацію з датчика одометра. Позначимо ці дані у вигляді-відстані, пройденого роботом. Оскільки датчик встановлений на поворотному столі, доступна інформація про напрямок, в якому надходить інформація про перешкоду. Ці дані знаходяться зліва, справа і спереду від робота. У керуючій програмі зі зворотним зв’язком у форматі команди, наприклад, надалі буде введений додатковий параметр-умова виконання команди. Інтерпретація цієї команди полягає в тому, що в автономному режимі САУ нижнього рівня, використовуючи інформацію від датчика, видає команду при появі значення і інформує верхній рівень про завершення команди.

Під управлінням такої програми траєкторія роботи більше не буде чутлива до змін швидкості і прискорення її руху, але поява перешкод на ділянках траси між поворотами або під час поворотів і раніше призведе до порушень необхідної траєкторії.

Для початку необхідно отримати вхідну інформацію. Вона отримується із файлу `route.txt` який знаходиться на карті пам’яті, яка в свою чергу знаходиться в кард рідері типу microSD. Для цього підключаємо кард рідер для карт пам’яті типу microSD.

Лістинг 3.1 – Підключення та перевірка карти пам’яті типу microSD

```
Serial.begin(9600);
Serial.print("Initializing SD card... ");

if (!SD.begin(CS_PIN)) {
  Serial.println("Card initialization failed!");
  while (true);
}

Serial.println("initialization done.");
printFile = SD.open("route.txt ");
```

```
  if (!printFile) {
    Serial.print("The text file cannot be opened");
    while(1);
      }
```

В файлі route.txt міститься маршрут для робота у форматі таких команд напрямку: Forward, Backward, Left, Right. Поряд із командами напрямку міститься відстань у сантиметрах яку повинен подолати робот. Так як напрямок і відстань є невід'ємними атрибутами кожного кроку маршруту робота, тому вони повинні бути об'єднані в структуру даних. Цю структуру даних ми будемо сприймати як крок запланованого маршруту роботу. З цього слідує, що повний маршрут робота є масивом структур. Так, як ми не можемо знати наперед кількість кроків на які поділений маршрут в завантаженому файлі route.txt використовується динамічний масив структур. За допомогою циклу while відбувається завантаження маршруту з файлу route.txt в динамічний масив (лістинг 3.2).

Лістинг 3.2 – Завантаження маршруту в масив структур

```
struct Step{
  String direction;
  int distance;};

...

  route = new Step [steps_number];
  String direction_temp;
  String distance_temp;
  int index = 0;
  while (printFile.available()) {
    direction_temp = printFile.readStringUntil(',');
    distance_temp = printFile.readStringUntil('\n');
    route[index].direction = direction_temp;
    route[index].distance = distance_temp.toInt();
    index++;  }
```

Далі береться кожен крок із масиву, проходить вибір напрямку руху по вказаній команді в та виконується рух на задану відстань. Пройдена відстань

рахується завдяки датчику швидкості LM393. Розраховуємо довжину окружності колеса = P x D = 3,14 x 5 = 15,7 см; Вираховуємо скільки оборотів повинні зробити двигуни роботи, щоб він проїхав 2 метри = 200 : 15,7 = 12,7 оборотів. Включаємо в програму двигунів, а по енкодеру в програмі визначаємо скільки двигунів зробили обороти, як тільки вони стануть 12,7, то двигуни відключаються і для повороту в залежності від угла та напряму повороту підраховуємо кількість оборотів одного з колес.

Сприймаючий елемент модуля – оптична пара ITR9608. В одному виступі корпусу розташований інфрачервоний світлодіод, спрямований на фототранзистор, що знаходиться в іншому виступі. Якщо в щілину між виступами внести непрозору пластину, то ІЧ випромінювання світлодіода перекривається і фототранзистор закривається. Такий компонент іноді називають фотопереривником, але він лише фіксує переривання світлового потоку. З емітера транзистора сигнал надходить у схему приладу. Світло перегороджує пластина, закріплена на рухомій частині контрольованого механізму. За допомогою датчика визначають переміщення в крайні положення площини, що рухається, коли пов’язаний з нею елемент конструкції входить в датчик. Вимірюють параметри обертання різних механічних деталей. Так відбувається перетворення механічних параметрів електричні величини і далі в програмні значення.

Датчик швидкості обертання FC-03 використовується для вимірювання частоти обертання. Визначити з його допомогою напрямок обертання не можна. На валу двигуна чи шестерні редуктора встановлюється диск з отворами. Оптопара модуля має проріз шириною 5 мм. При обертанні диска, як зображено на анімації, постійно прорізи чергуються отвори і ділянки пластини. Датчик перетворює чергування елементів диска електричні імпульси. Електроніка модуля робить сигнал датчика придатним для сприйняття цифровими логічними мікросхемами або мікроконтролером.

ВИСНОВКИ

У першому розділі були досліджені роботизовані системи, що дало змогу оцінити можливість і актуальність впровадження роботизованих систем в різні сфери життя людини. Також були проаналізовані вже існуюсі пристрої з основною ідеєю керування користуючить одометричними даними. Також були проаналізовані вже існуючі пристрої. Розгялянуті пристрої сформували основні переваги і недоліки тих, чи інших варіантів побудови системи. В розділі наведене пояснення операційних систем, що можуть підходити для створення пристрою. Окрім популярних операційних систем були також розглянуті вузьконаправлені. В заключення розділу була сформована мета роботи, основні задачі, які повинна виконувати готова система та основні вимоги, яким вона повинен відповідати.

Другий розділ направлений на аналіз та вибір апаратної платформи. Були сформовані мінімальні вимоги щодо апаратної частини проекту. Дослідження апаратних систем типу Single-board computer дало розумніння можливостей цих міні комп’ютерів. Так як система не потребує великих матетматичних обчислень, можливості апаратних систем типу Single-board computer задовольняють мінімальні вимоги. Так як на риноку одноплатних комп’ютерів представлено достатньо багато продуктів був проведений аналіз цього ринку і більш детально зупинився на середовищі Arduino.

В третьому розділі наведений опис програмної частини системи, вибір середовища розробки та основні елементи. Середовище Arduino дозволяє реалізувати всі необхідні завдання які поставлено перед системою для розробника максимально просто і швидко.

Метод управління переміщенням робота з використанням одометричних даних вирішує основні проблеми методу програмного управління з зворотними зв’язками. Під управлінням такої програми траєкторія робота вже не буде чутлива до варіацій швидкості і прискорення

його руху, однак поява перешкод на ділянках маршруту між поворотами або під час розворотів і раніше, призведе до порушень необхідної траєкторії пересування. Данну систему можна використовувати в загальних ситуаціях, а також видозмінювати та дороблювати, в залежності від конфігурацій РП, а також задач, які ставляться перед системою.

В майбутньому є найпріорітетнішим інтеграція якомога більше датчиків які сигналізують про перешкоди та доповнити систему логікою долання чи реагування на перешкоди, що дощволить більш безпечно використовувати систему як в жилих приміщеннях так і на виробництві. Також необхідна можливість будувати мережу із подібних роботизованих систем, що дозволить у разі виходу із строю одного із екземплярів замінити його іншим на виконуваному маршруті.

У метода управління переміщенням робота з використанням одометричних даних є великий потенціал на майбутнє. Ця система зможе не просто виконувати свої функції по оптимізації витрат на великих виробництвах та зменшити кількість збоїв на конвеєрних лініях.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Ghlukhov, O. V., Kravchuk, O. O. & Levchenko, Je. V. (2020). Creation of a laboratory workshop based on the Arduino platform and its role in teaching students of technical universities of all forms of education in the specialty "Electronics". In Proceedings of the eleventh international scientific-practical conference «Prospects for the development of modern science and education», Lviv, 15-16 June, 2020: 13, Lviv

2. Введение в интелектуальные машины. Книга 1. Интелектуальные регуляторы / А.А. Каргин ¬ Донецк: Норд-Пресс, ДонНу, 2010.

3. Ускорение и торможение движения робота. [Електронний ресурс]. Режим доступу : https://uchiurok.ru/blog/arduino/kejsy-zadanij-dlyaizucheniya/uskorenie-i-tormozhenie-dvizheniya-robota.html

4. Умный Интернет вещей – кто он и с чем его едят? [Електронний ресурс]. – Режим доступу до ресурсу: https://habrahabr.ru/post/259243/.

5. Собираем роботов–самоходов на Arduino. [Електронний ресурс]. Режим доступу : https://xakep.ru/2014/10/30/robots-arduino/

6. Авотин Е. В., Епишин К. В. Подвижность автономного транспортного робота / Е. В. Авотин, К. В. Кпишин // Инноватика и экспертиза. – 2013. – Выпуск 2(11). – с. 67-72

7. Роботы. Большая энциклпедия / - «Эксмо», 2017. – 144с.

8. Измерение скорости, пройденного пути и угла поворота с помощью Arduino [Електронний ресурс]. – 2020. – Режим доступу до ресурсу: https://microkontroller.ru/arduino-projects/izmerenie-skorosti-projdennogo-puti-i-ugla-s-pomoshhyu-arduino-i-datchika-lm393/

9. Kargin A., Petrenko T. Internet of Things smart rules engine // 2018 International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC&T). – IEEE, 2018. – C. 639-644.

10. Омельченко Е. Я., Танич В. О., Маклаков А. С., Карякина Е. А.

Краткий обзор и перспективы применения микропроцессорной платформы Arduino // ЭС и К. 2013. №21.

11. Лукьянов Андрей Анатольевич Методы управления движением экспериментального мобильного сервисного робота // Вестник ИрГТУ. 2014. №3 (19).

12. Крупский А.В. Робот как объект изучения и инструмент познания в учебном процессе по информатике // Образовательный вестник «Сознание». 2018. №10.

13. Роботизация: обзор сферы и опыт внедрения в компаниях [Електронний ресурс]. – 2020. – Режим доступу до ресурсу: https://netology.ru/blog/03-2021-what-is-rpa

14. РОБОТОТЕХНИКА [Електронний ресурс]. – 2019. – Режим доступу до ресурсу: https://exponenta.ru/robotics.

15. Алгоритми локальної навігації і картографії для бортової системи управління автономного мобільного робота

2012 / Кучерської Роман Володимирович, Манько Сергій Вікторович