

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

**«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Факультет електроніки

Кафедра Акустичних та мультимедійних електронних систем

До захисту допущено:

Завідувач кафедри

 _ Сергій НАЙДА

« 10 » червня 2022 р.

Дипломна робота

на здобуття ступеня бакалавра

**за освітньо-професійною програмою «Електронні системи мультимедія та
засоби Інтернету речей»**

спеціальності 171 «Електроніка»

на тему: «Електронний тренажер для водія»

Виконав:

студент IV курсу, групи ДВ - 81

Павлюк Богдан Володимирович



Керівник:

Старший викладач

Гумен Тамара Федосіївна



Рецензент:

доцент кафедри ЕПС КПІ ім. Ігоря Сікорського, доцент, к.т.н.



Клен Катерина Сергіївна

Засвідчую, що у цій дипломній роботі
немає запозичень з праць інших авторів
без відповідних посилань.

Студент



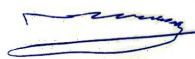
Київ – 2022 року

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

Кафедра Акустичних та мультимедійних електронних систем
Рівень вищої освіти – перший (бакалаврський)
Спеціальність – 171 «Електроніка»
Освітньо-професійна програма «Електронні системи мультимедія та засоби Інтернету речей»

ЗАТВЕРДЖУЮ

Завідувач кафедри



Сергій НАЙДА

«02» травня 2022 р.

ЗАВДАННЯ

на дипломну роботу студенту

Павлюку Богдану Володимировичу

1. Тема роботи «Електронний тренажер для водія», керівник роботи Гумен Тамара Федосіївна

затверджені наказом по університету від «06» червня 2022 р. № 911-с

2. Термін подання студентом роботи 10.06.2022р

3. Вихідні дані до роботи. Мікроконтролери AVR. Платформа Arduino, мікроконтролер Arduino Leonardo, програмне забезпечення Arduino IDE .

4. Зміст роботи. Розгляд виникнення, розвитку та використання мікроконтролерів. Фізична реалізація мікроконтролерів AVR на платах Arduino. Створення електронного тренажера для водія.

5. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо) презентація з наведеними результатами розроблення електронного тренажера.

6. Дата видачі завдання 02.05.2022р

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1	Написання першого розділу	12.05.2022	виконано
2	Написання другого розділу	20.05.2022	виконано
3	Написання третього розділу	25.05.2022	виконано
4	Підготовка матеріалів до друку та оформлення пояснювальної записки	01.06.2022	виконано
5	Підготовка та оформлення презентації для доповіді	10.06.2022	виконано

Студент



Богдан ПАВЛЮК

Керівник



Тамара ГУМЕН

Київ – 2022 року

УДК 658.562:778.5

РЕФЕРАТ

Павлюк Б.В. «Електронний тренажер для водія»: дипломна робота бакалавра : 171 Електроніка. - КПІ ім. Ігоря Сікорського, Київ, 2022. - 68 с.

Ключові слова: мікроконтролер, Arduino, AVR, Arduino Leonardo , Arduino IDE, електронні системи.

Об'єктом дослідження є використання мікроконтролерів в електронних системах.

Метою роботи є аналіз переваг та недоліків існуючих електронних систем, та створення власної системи.

Методом дослідження є теоретичний огляд особливостей використання мікроконтролерів в електронних системах.

У результаті виконання дипломної роботи проаналізовано принципи роботи мікроконтролерів, проведено аналіз окремих налагоджувальних плат. Створено власну електронну систему.

Галузь застосування: конструювання електронних систем.

УДК 658.562:778.5

ABSTRACT

Pavlyuk Bohdan "Electronic simulator for the driver": Bachelor Thesis: 171 Electronics. - KPI them. Igor Sikorsky, Kyiv, 2022. 68 p.

Keywords: microcontroller, Arduino, AVR, memory, electronic systems.

The object of research is the use of microcontrollers in electronic systems.

The aim of the work is to analyze the advantages and disadvantages of existing electronic systems and create your own system.

The research method is a theoretical study of the features of electronic systems.

As a result of the thesis, the principles of operation of microcontrollers were analyzed, the analysis of individual debug boards was performed. Created its own electronic system.

Scope: design of electronic systems.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ І СКОРОЧЕНЬ.....	7
ВСТУП.....	8
1 МІКРОКОНТРОЛЕРИ ЯК ОСНОВА ЕЛЕКТРОННИХ СИСТЕМ	10
1.1 Історія виникнення та розвитку мікроконтролерів	10
1.2 Використання мікроконтролерів у сьогоденні.....	15
1.3 Висновки до розділу	19
2 ФІЗИЧНА РЕАЛІЗАЦІЯ МІКРОКОНТРОЛЕРІВ AVR НА ПЛАТАХ	
ARDUINO.....	20
2.1 Архітектура мікроконтролерів AVR.....	20
2.2 Налагоджувальні плати Arduino	27
2.3 Програмне забезпечення для роботи з мікроконтролером	39
2.4 Висновки до розділу	41
3 СТВОРЕННЯ ТРЕНАЖЕРУ ДЛЯ ВОДІЯ.....	42
3.1 Вибір компонентів.....	42
3.2 Особливості конструкції тренажеру	45
3.3 Переваги використання	47
3.4 Конструкція тренажеру	48
3.5 Висновки до розділу	53
ВИСНОВКИ.....	55
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ.....	56
ДОДАТОК А. Лістинг коду	59

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ І СКОРОЧЕНЬ

АЦП	– Аналого-цифровий перетворювач
ПК	– Персональний комп'ютер
ЦАП	– Цифро-аналоговий перетворювач
ЦП	– Центральний процесор
ALU	– Arithmetic logic unit (арифметико-логічний пристрій)
CISC	– Complex Instruction Set Computers (комп'ютер зі складним набором команд)
EEPROM	– Electrically Erasable Programmable Read-Only Memory
EPROM	– Erasable Programmable Read Only Memory
I/O	– input/output (вхід/вихід)
RAM	– Random Access Memory (пам'ять з довільним доступом)
RISC	– Reduced Instruction Set Computer (обчислення зі скороченим набором команд)
VLSI	– Very Large Scale Integration (схеми надвеликого рівня інтеграції)

ВСТУП

З моменту створення, мікроконтролери зайняли важливе місце у нашому повсякденні. Їх покращення та пошук нових сфер застосування не припиняється й досі. Якщо раніше мікроконтролер був інструментом лише для професіоналів, то з моменту винайдення налагоджувальних плат, до розроблення своїх власних електронних систем та впровадження їх в життя, може долучитись будь-хто. Сфер застосувань мікроконтролерів є безліч: медицина, побутові товари, побутова техніка, автомобілебудування, розваги та багато інших.

Мікроконтролери широко використовуються в ігровій індустрії. Існує досить велика кількість симуляторів (ігор), які можуть використовувати майбутні водії чи водії-початківці. Проте, щоб розкрити потенціал симуляторів, потрібно використовувати відповідне обладнання. Саме тому, розробка електронного тренажеру для водіїв необхідна. Вона дозволить залучити велику кількість людей не тільки цікаво проводити свій вільний час, а й здобути чи покращити базові навички водіння автомобіля.

Створений тренажер цілком може задовольнити цю потребу, оскільки він містить всі необхідні органи управління, а його конструкція наближена до справжнього автомобіля.

Актуальність роботи визначається необхідністю наявності електронних тренажерів з розширеним функціоналом для підвищення якості підготовки слухачів в автошколах.

Метою роботи є аналіз переваг та недоліків існуючих електронних систем, та створення власної системи – електронного тренажеру для водія.

Методом дослідження є теоретичне дослідження особливостей електронних систем створених на базі AVR мікроконтролерів.

Об'єктом дослідження є використання мікроконтролерів в електронних системах.

Новизна роботи полягає у створенні електронного тренажеру для водія, що має розширений функціонал – важіль поворотів, важіль двірників, перемикач режиму фар, та більшу сферу застосування.

Практична цінність полягає в тому, що електронний тренажер, отриманий в ході виконання досліджень, може бути використаний для навчання майбутніх водіїв.

1 МІКРОКОНТРОЛЕРИ ЯК ОСНОВА ЕЛЕКТРОННИХ СИСТЕМ

1.1 Історія виникнення та розвитку мікроконтролерів

Мікроконтролер - це інтегральна схема (IC) VLSI (Very Large Scale Integration), яка містить електронно-обчислювальний блок і логічний блок (об'єднано відомий як ЦП), пам'ять (пам'ять програм і пам'ять даних), I/O порти (порти вводу-виводу) і кілька інших компонентів, інтегрованих на одному чіпі (рис. 1.1)[1].

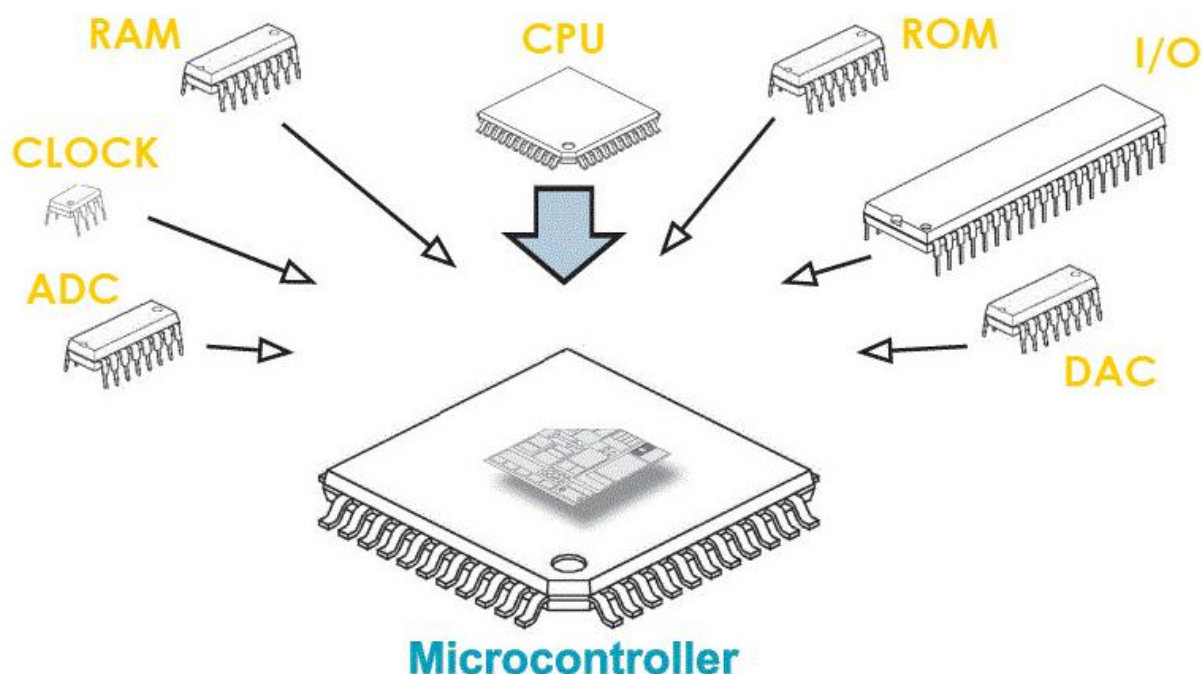


Рисунок 1.1 – Складові частини мікроконтролера [1]

Мікроконтролер був розроблений компанією Intel у 1980 році [2]. 8051 є одним з найпопулярніших 8-розрядних мікроконтролерів і поєднує в собі набір інструкцій, який дозволяє чітко кодувати невелику програму вводу-виводу з достатньою потужністю та достатньо великим простором для програм, що дає змогу використовувати його з мовою C.

За останнє десятиліття відбулася захоплююча еволюція можливостей мікропроцесорів. Розвиток 16 і 32-розрядних мікропроцесорів сприяв зростанню потужних персональних комп'ютерів, які використовуються в усіх сферах

життя. Завдяки своїй потужності та швидкості обробки, 16 та 32-розрядні мікропроцесори також використовуються при розробленні автономних продуктів, таких як електронні прилади, які вимагають складних можливостей керування. В еволюції можливостей мікропроцесора, замість того, щоб зосередитися на більшій ширині слів і адресних просторах, увага акцентується на надзвичайно швидкому контролі в реальному часі. Розроблення мікроконтролерів було зосереджено на інтеграції засобів, необхідних для підтримки швидкого керування, в єдиний чіп.

У 1976 році Intel представила стандартний 8-розрядний мікроконтролер 8048 (рис. 1.2).



Рисунок 1.2 - 8-розрядний мікроконтролер Intel 8048 [3]

Ця ж компанія продовжувала розвивати однокристалічні мікроконтролери. У 1980 році Intel представила мікроконтролер 8051 з більшою продуктивністю, ніж 8048 (рис 1.3). Завдяки перевагам 8051 програми мікроконтролерів досягли пікового рівня. 8-розрядний мікроконтролер, сімейства 8051, швидко зайняв позиції другого покоління мікроконтролерів світового стандарту.



Рисунок 1.3- 8-розрядний мікроконтролер Intel 8051 [4]

Завдяки передовій напівпровідниковій технології стало можливим інтегрувати більше 100000 транзисторів на одному кремнієвому чіпі. Корпорація Intel використала цю передову технологію і розробила нове покоління однокристальних 16-розрядних мікроконтролерів під назвою MCS-96 (сімейство 8096). Покоління 8096 пропонує найвищий рівень системної інтеграції, який коли-небудь був досягнутий на однокристальному мікроконтролері з 120000 транзисторів. Цей мікроконтролер має 16-розрядний ЦП, 8 Кбайт програмної пам'яті, 232 байти пам'яті даних і функції аналогового і цифрового типу I/O.

Сімейство мікроконтролерів Motorola було вперше представлено в 1978 році і побудовано за тим же зразком, що і мікропроцесор 6800 (рис. 1.4) [5].



Рисунок 1.4- Мікропроцесор Motorola 6800 [5]

Незважаючи на те, що сімейство мікроконтролерів 6801 було розроблено подібно до мікропроцесора 6800, його конструкція та набір інструкцій були

змінені, щоб відповідати керуванню додатків. Сімейство мікроконтролерів 6801 включає в себе порти введення/виводу на мікросхемі, пристрій асинхронного послідовного зв'язку та 16-розрядні модулі таймера. Мікроконтролери 6801, 6803, 6805, 6811 доступні в компанії Motorola. Сімейство мікроконтролерів 6811 має різні версії з ПЗУ, ОЗП, EPROM та EEPROM. Ці версії позначаються суфіксами та цифрами. Хоча не важливо, як мікроконтролери розвивалися до такого рівня, де вони є сьогодні, це цікава історія, яка може допомогти зрозуміти, де мікроконтролер AVR вписується в загальну ієрархію інформаційних технологій (IT) та електронних продуктів. Що ще важливіше, завдяки такому розумінню можна краще вибирати та приймати рішення про те, коли і де використовувати мікроконтролер, надавши перевагу іншим альтернативам.

До середини 1980-х більшість електронних продуктів все ще створювалися з використанням надзвичайно складних і розумних комбінаторних логічних схем, реалізованих за допомогою жахливої кількості мікросхем! Починаючи з початку 1980-х років, меншість виробників почали вбудовувати мікропроцесори в свою продукцію, щоб зменшити кількість мікросхем, що знизило витрати виробництва і, таким чином, знизило ціни для кінцевих користувачів. Найперші 8-розрядні мікропроцесори, такі як Intel 8080 або Zilog Z80, вперше з'явилися наприкінці 1970-х років і були значним прогресом у порівнянні з тим, що було раніше. Інженери та дизайнери невдовзі зрозуміли, що, вставивши мікропроцесор у пристрій, можна не тільки змусити його робити набагато більше, але й оновити його набагато дешевше, якщо виявляться дефекти або недоліки оригінального дизайну. Багато дефектів продукту тепер можна було б усунути, використовуючи напівкваліфіковану працю для підключення змінного ПЗУ (пам'ять лише для читання) (це було в дні до програмованої флеш-пам'яті), замість того, щоб використовувати кваліфіковану працю для дорогої переробки або заміни тисяч повноцінних друкованих плат. У 1980-х роках все більше і більше продуктів мали мікропроцесор як свою основу. Незважаючи на те, що мікропроцесори були значним покращенням того, що вони замінили, вони не повністю призводили до зниження витрат і складності дизайну продукту. Проблема полягала в тому, що

для того, щоб мікропроцесор міг робити щось корисне, він мав бути оточений великою кількістю додаткових мікросхем для введення-виводу, і зазвичай йому потрібні були й інші підтримуючі мікросхеми — наприклад, чіпи годинника реального часу та адресні дешифратори. До 1990-х років вдосконалене оброблення кремнію та технології виробництва мікросхем призвели до можливості помістити все більше схем на один чіп. Одним із способів використання цього було доповнення мікропроцесора додатковими функціями та функціями, які раніше були реалізовані окремими зовнішніми мікросхемами. Щоб відрізнити ці нові супермікрочіпи від їх простіших попередників, їх почали називати мікроконтролерами.

Деякі приклади функцій, які перейшли від зовнішніх мікросхем до складу мікроконтролера.

- Послідовні порти, щоб дозволити підсистемі спілкуватися з настільним комп'ютером або іншими пристроями, обладнаними портом RS232.
- Таймери, які дозволяють мікроконтролеру мати точний контроль часу на мікросхемі та виконувати події з точними попередньо встановленими інтервалами. Таймери також дозволили мікроконтролерам генерувати музику та звуки, оскільки можна було гарантувати точність інтервалу.
- Послідовні цифрові канали, щоб мікроконтролери могли спілкуватися один з одним лише через два з'єднувальні дроти.
- Аналого-цифрові перетворювачі, що дозволяють системі мікроконтролера сприймати аналогові сигнали та зберігати або обробляти їх як цифрові дані.
- Цифро-аналогові перетворювачі, які дозволяли мікроконтролерам взаємодіяти із зовнішніми пристроями, такими як двигуни, які потребують безперервної напруги.
- Вхідні порти для визначення стану увімкненого/вимкненого речей у зовнішньому світі.
- Вихідні порти для вмикання/вимкнення речей у зовнішньому світі.

1.2 Використання мікроконтролерів у сьогоденні

Наш світ наповнений електронними продуктами, які розважають нас, полегшують наше життя, дозволяють спілкуватися один з одним, допомагають запобігти захворюванням та багато іншого. Найчастіше в основі всіх цих електронних продуктів лежить мікроконтролер. Цей дивовижний пристрій — це мозок, який контролює всі дії та виконує обчислення в електронному пристрої. Завдяки безмежним можливостям мікроконтролера можна використовувати багато функцій і не обмежуватися однією окремою програмою.

Мікроконтролер керує апаратним забезпеченням, яке є ключем до роботи цих пристроїв [6]. Наприклад, він отримує вхідні дані від кнопок і клавіш і керує периферійними ланцюгами, такими як двигуни та дисплеї, на основі попередньо встановлених програм

Побутова електроніка.

Споживча електроніка — це термін, який визначає електронне обладнання, яке використовується щодня (переважно у власному домі). Це дуже широкий термін, який охоплює багато інших категорій, таких як розваги, спілкування, відпочинок, кухонна техніка тощо. Споживчу електроніку часто називають «чорним товаром», оскільки її електроніка розміщена в темному корпусі.

Нижче наведено кілька прикладів споживчої електроніки, яка має в собі мікроконтролер.

- Радіо
- Мікрохвильова піч
- Телебачення
- Обігрівач/Вентилятор
- Калькулятор
- MP3-плеєри
- 3D принтери

Медицина

Медицина отримала величезну користь від використання мікроконтролерів. Як можна собі уявити, медична практика до появи мікроконтролера була б досить примітивною. Крім того, фраза «краще попередити, ніж лікувати» у медичних програмах це означає, що якщо можна виявити проблему на ранніх стадіях, то є більше шансів її усунути. Хоча людство, має дивовижні здібності та інтелект, існує межа того, що можна виявити в людському тілі. Мікроконтролери разом з іншими технологіями дозволяють виявляти захворювання людського організму, поки не стало надто пізно.

Ці пристрої включають.

- Рентгенівський знімок.
- Магнітно-резонансна томографія (МРТ).
- Електрокардіограма (ЕКГ).
- Комп'ютерна томографія (КТ).
- Сонографія (УЗД).
- Монітори серцевого ритму.
- Монітори артеріального тиску.
- Крім виявлення, мікроконтролери використовуються в медичних пристроях, щоб допомогти виконувати функції органів, які з часом можуть зіпсуватися, працювати неправильно або були пошкоджені. Це такі пристрої.
- Діалізні апарати (виконує функцію печінки).
- Кардіостимулятори (використовуються для контролю аномального серцевого ритму).
- Слухові апарати.
- Роботизовані протези кінцівок.
- Системи моніторингу рівня глюкози (для цукрового діабету 1 типу).

Мікроконтролери також використовуються в хірургії. Роботи використовуються для допомоги хірургам у різних видах операцій.

Розваги

У епоху до електроніки та мікроконтролерів була обмежена кількість вправ, які допомагають скоротити час. Такі речі, як читання книги, прогулянка, гра на гітарі, гра в хованки тощо. Сьогодні маємо безліч пристроїв, щоб розважитися. Ще до мікроконтролера існували електронні пристрої, які могли допомогти розважитися та скоротити час. Але за допомогою мікроконтролера ці пристрої зросли не тільки кількісно, але й у складності, і в якості.

Старий транзисторний радіоприймач міг бути обмежений кількома станціями з низькою якістю звуку. Але тепер цифрове радіо має можливість отримувати доступ до всіх станцій у діапазоні FM та AM з високою якістю звуку, а також відтворювати компакт-диски.

Нижче наведено кілька прикладів електронних пристроїв, які допоможуть так чи інакше розважитися.

- Телебачення
- Комп'ютер/ноутбук
- Проектор
- Ipad
- Ipod
- Цифрове радіо
- Xbox
- Ігрова приставка
- Nintendo
- Kindle
- Смартфон

Прилади управління процесами

Прилади управління процесами — це галузь техніки, яка займається теорією керування. Вони використовуються для виробництва обладнання, яке допомагає проєктувати, контролювати та керувати різними процесами в різних галузях промисловості. Ці процеси гарантують, що загальна система є продуктивною, ефективною та повторюваною. Загальна система складається з

програмного та апаратного забезпечення, яке включає в себе датчики, виконавчі механізми (механічні та електричні) і, звичайно, мікроконтролер. Наприклад, на хімічному заводі наявність системи, яка контролює температуру та гарантує, що вона залишається у визначених межах, є надзвичайно важливою для уникнення потенційних катастроф, таких як вибухи.

Нижче наведено галузі, які використовують приладдя та контроль процесів.

- Нафта і газ.
- Нафтохімічний.
- Очищення стічних вод.
- Тканина.
- Їжа та напої.

Зв'язок

Спілкування є необхідністю в повсякденному житті. Комунікація необхідна для різних цілей; ділитися інформацією, робити заяви, ставити запитання, висловлювати бажання/потреби, підтримувати зв'язок з близькими, висловлювати емоції та багато іншого незалежно від часових і просторових координат. Саме використання мікроконтролерів в багатьох електронних комунікаційних пристроях, дозволяє миттєво спілкуватися з близькими, колегами, незнайомими людьми в реальному часі навіть якщо вони знаходяться в іншій країні.

Такі пристрої, як:

- Смартфони.
- Текстові повідомлення.
- Електронна пошта.
- Комп'ютери та ноутбуки.
- Відеодзвінки.
- Електронна пошта.
- Ведення блогів.
- Рації.
- Телефони.
- Факси.

Безпека

На жаль, світ сповнений багатьох небезпек. Однак наявність систем, які контролюють стан повідомляють про небезпеку, є найкращим способом уникнути травм або смерті. Такі технології, як.

- Пожежна сигналізація.
- Сигналізатори диму.
- Системи безпеки.
- PIR датчики.
- Системи моніторингу газу.

Головне завдання мікроконтролера — виконувати обчислення та керувати завданнями. Кращою аналогією та способом візуалізації цього є людське тіло. Організм людини представляє собою електронну систему з безлічу підсистем: вхідних (почуттів), виходів (м'язів) та інших підсистем (органів, таких як серце, печінка, нирки тощо). Мозок найкраще представляє функціональні можливості мікроконтролера. Він отримує сигнали від ваших органів чуття і посилає сигнали іншим частинам тіла, таким як органи та м'язи, контролюючи загальну роботу.

Вбудована система — це поєднання апаратних і програмних компонентів. Мікроконтролери використовуються точно так само у вбудованих системах у багатьох програмах. Вони гарантують, що всі частини системи працюють злагоджено, щоб досягти того, для чого ця конкретна система була створена.

1.3 Висновки до розділу

Розглянуто історію виникнення перших мікроконтролерів, які дали поштовх розвитку сучасних електронних систем, що використовуються в різних сферах життя.

Розглянуто сфери застосування мікроконтролерів в електронних системах на конкретних прикладах.

2 ФІЗИЧНА РЕАЛІЗАЦІЯ МІКРОКОНТРОЛЕРІВ AVR НА ПЛАТАХ ARDUINO

2.1 Архітектура мікроконтролерів AVR

Існує ряд популярних сімейств мікроконтролерів, які використовуються в різних програмах відповідно до їх можливостей і можливостей для виконання бажаного завдання. Найбільш поширеними з них є мікроконтролери 8051, **AVR** і PIC [7].

AVR був розроблений у 1996 році компанією Atmel Corporation. Архітектура AVR була розроблена Альф-Егілом Богеном і Вегардом Волланом. AT90S8515 був першим мікроконтролером, який заснований на архітектурі AVR, однак першим мікроконтролером, який вийшов на комерційний ринок, був AT90S1200 в 1997 році.

Мікроконтролери AVR доступні в трьох категоріях.

1. **TinyAVR** – Менше пам'яті, малий розмір, підходить лише для простих програм.
2. **MegaAVR** – це найпопулярніші з них, які мають хороший обсяг пам'яті (до 256 КБ), більшу кількість вбудованих периферійних пристроїв і підходять для середніх і складних програм.
3. **XmegaAVR** – використовується в комерційних цілях для складних програм, які потребують великої пам'яті програм і високої швидкості.

У таблиці 2.1 порівняно вищезгадані мікроконтролери серії AVR.

Таблиця 2.1 – Порівняння мікроконтролерів AVR

Назва серії	Шпильки	Флеш-пам'ять	Особлива функція
TinyAVR	6-32	0,5-8 КБ	Невеликий за розміром
MegaAVR	28-100	4-256 КБ	Розширені периферійні пристрої
XmegaAVR	44-100	16-384 КБ	DMA, система подій у комплекті

Особливістю AVR є їх швидкість. Мікроконтролер AVR виконує більшість інструкцій за один цикл виконання. AVR приблизно в 4 рази швидше, ніж PIC, вони споживають менше енергії і можуть працювати в різних режимах енергозбереження. Проведемо порівняння між трьома найбільш часто використовуваними сімействами мікроконтролерів (табл. 2.2).

Таблиця 2.2 – Порівняння сімейств мікроконтролерів

	8051	PIC	AVR
ШВИДКІСТЬ	Повільно	Помірний	Швидко
ПАМ'ЯТЬ	Маленький	Великий	Великий
АРХІТЕКТУРА	CISC	RISC	RISC
АЦП	Не присутній	Вбудований	Вбудований
Таймери	Вбудований	Вбудований	Вбудований
ШІМ канали	Не присутній	Вбудований	Вбудований

AVR - це 8-розрядний мікроконтролер, що належить до сімейства комп'ютерів зі скороченим набором команд (RISC). Це означає, що мікроконтролер здатний передавати та отримувати 8-розрядні дані. Доступні регістри введення/виводу мають 8 біт. Контролери сімейства AVR мають архітектуру на основі регістру, а це означає, що обидва операнди для операції зберігаються в регістрі, а результат операції також зберігається в регістрі. На рис. 2.1 показано простий приклад виконання операції АБО між двома вхідними регістрами та збереження значення у вихідному регістрі.

В архітектурі RISC набір команд комп'ютера не тільки менший, але й простіший і швидший в роботі. Іншим типом категоризації є CISC (комп'ютери з комплексним набором інструкцій).

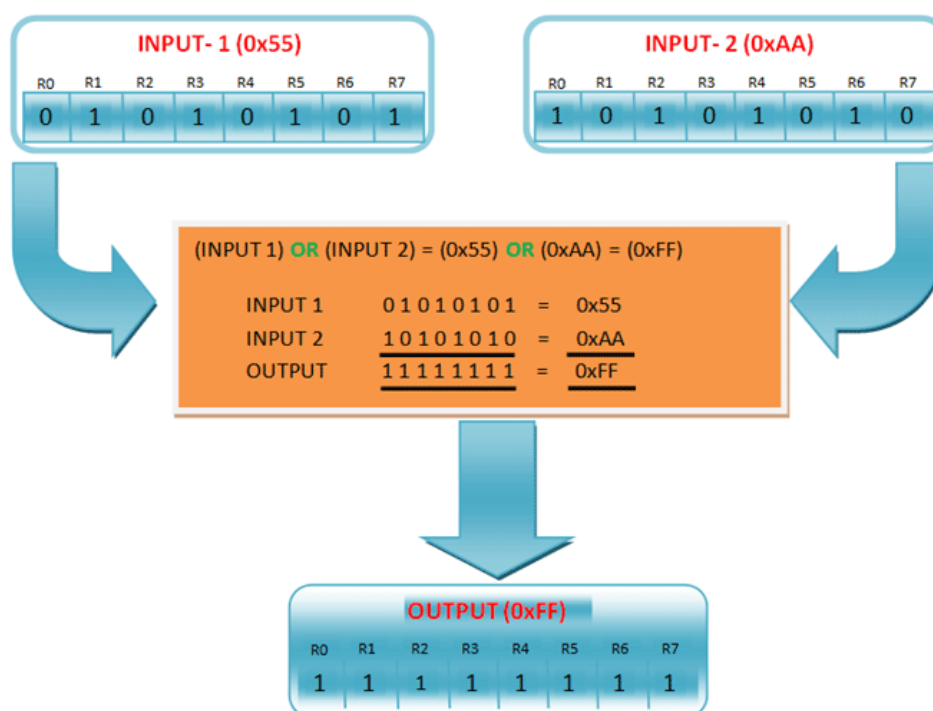


Рисунок 2.1- Блок-схема виконання операції АБО між двома вхідними регістрами та збереженням значень у вихідному регістрі [8]

Мікроконтролери AVR засновані на передовій архітектурі RISC і складаються з 32 х 8-розрядних робочих регістрів загального призначення. Протягом одного тактового циклу AVR може приймати вхідні дані з двох регістрів загального призначення і передавати їх в ALU для виконання запиту і передавати результат назад у довільний регістр. ALU може виконувати як арифметичні, так і логічні операції над входами з регістра або між регістром і константою. Операції з одним реєстром, такі як прийом доповнення, також можуть виконуватися в ALU. AVR не має такого регістра, як акумулятор, як у мікроконтролерів сімейства 8051; операції можуть виконуватися між будь-яким з регістрів і можуть зберігатися в будь-якому з них.

AVR відповідає формату Гарвардської архітектури, в якому процесор оснащений окремою пам'яттю та шинами для інформації про програми та дані. Поки виконується поточна інструкція, наступна попередньо витягується з пам'яті програми (рис.2.2)

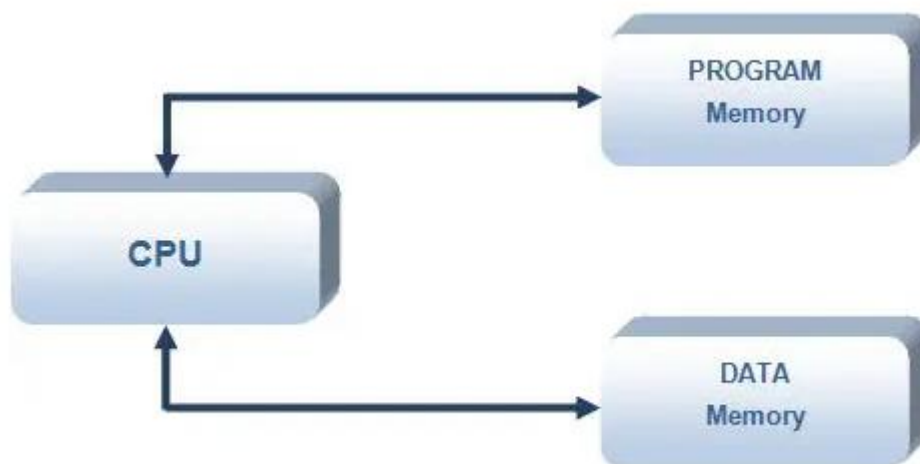


Рисунок 2.2 – Блок-схема архітектури пам'яті в AVR [8]

За один цикл AVR може виконувати 1 мільйон інструкцій в секунду, якщо частота циклу становить 1 МГц. Чим вище робоча частота контролера, тим більше буде швидкість його обробки. Для оптимізації енергоспоживання зі швидкістю обробки, необхідно відповідно вибрати робочу частоту.

Є два варіанти мікроконтролера Atmega16.

1. Atmega16 : Діапазон робочих частот становить 0 – 16 МГц.
2. Atmega16L : Діапазон робочих частот становить 0 – 8 МГц.

Якщо використовується кристал 8 МГц = 8×10^6 Герц = 8 мільйонів циклів, то AVR може виконати 8 мільйонів інструкцій.

АТ відноситься до виробника Atmel. Mega означає, що мікроконтролер відноситься до категорії MegaAVR. **16** означає пам'ять контролера, яка становить 16 КБ, рис. 2.3.



Рисунок 2.3 – Принцип іменування мікроконтролерів AVR [8]

Блоки архітектури Atmega16

Порти введення-виводу : Atmega16 має чотири (PORTA, PORTB, PORTC і PORTD) 8-розрядні порти введення-виводу.

Внутрішній калібрований генератор : Atmega16 оснащений внутрішнім генератором для керування його годинником. За замовчуванням Atmega16 налаштований на роботу з внутрішнім каліброваним генератором 1 МГц. Максимальна частота внутрішнього генератора становить 8 МГц. Крім того, ATmega16 може працювати за допомогою зовнішнього кварцевого генератора з максимальною частотою 16 МГц рис.2.4.

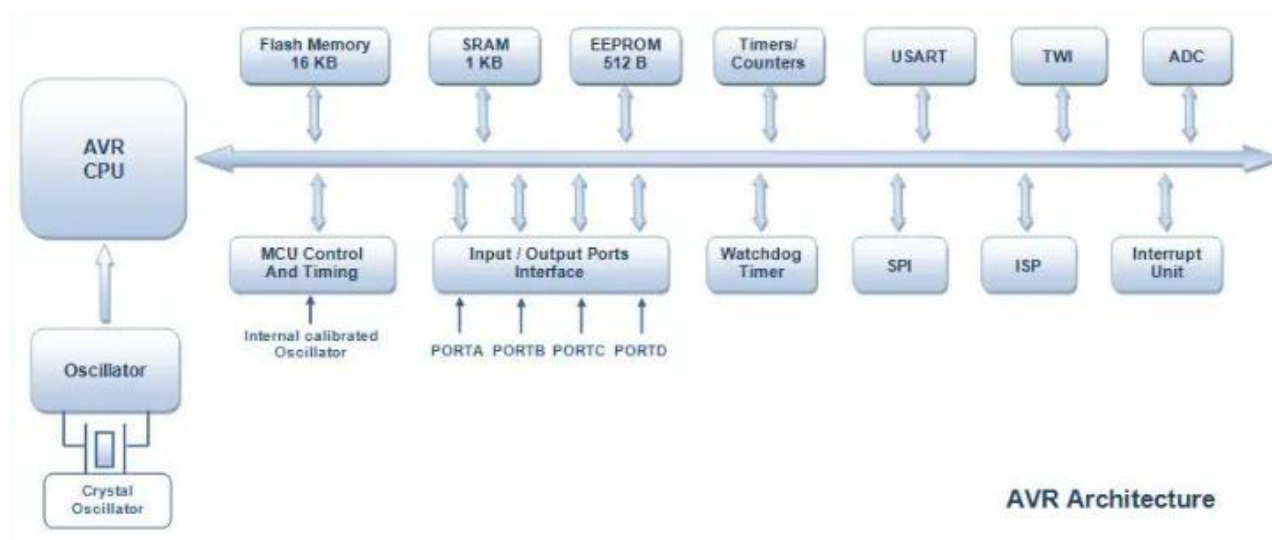


Рисунок 2.4 – Блок-схема архітектури Atmega16 [9]

Інтерфейс АЦП : Atmega16 оснащений 8-канальним АЦП (аналого-цифровий перетворювач) з роздільною здатністю 10 біт. АЦП зчитує аналоговий вхід, наприклад, вхід датчика, і перетворює його в цифрову інформацію, зрозумілу мікроконтролеру.

Таймери/лічильники : Atmega16 складається з двох 8-розрядних та одного 16-бітного таймера/лічильника. Таймери корисні для генерування точних дій, наприклад, для створення часових затримок між двома операціями.

Сторожовий таймер : Сторожовий таймер присутній із внутрішнім осцилятором. Сторожовий таймер безперервно відстежує та скидає контролер,

якщо код зупиняється під час виконання будь-якої дії протягом більш ніж визначеного інтервалу часу.

Переривання : Atmega16 складається з 21 джерела переривань, чотири з яких є зовнішніми. Решта - це внутрішні переривання, які підтримують периферійні пристрої, такі як USART, АЦП, таймери тощо.

USART : Універсальний синхронний та асинхронний інтерфейс приймача та передавача доступний для взаємодії із зовнішнім пристроєм, здатним здійснювати послідовний зв'язок (передача даних побітна).

Регістри загального призначення : Atmega16 оснащений 32 регістрами загального призначення, які з'єднані безпосередньо з арифметичним логічним блоком (ALU) ЦП.

Пам'ять : Atmega16 складається з трьох різних секцій пам'яті.

1. Flash EEPROM : Flash EEPROM або проста флеш-пам'ять використовується для збереження програми, скинутої або записаної користувачем на мікроконтролер. Його можна легко стерти електрично як єдине ціле. Флеш-пам'ять є енергонезалежною, тобто вона зберігає програму навіть при відключенні живлення. Atmega16 доступний з 16 Кбайт програмованого Flash EEPROM.

2. Байтова адресована EEPROM : це також енергонезалежна пам'ять, яка використовується для зберігання даних, наприклад значень певних змінних. Atmega16 має 512 байт EEPROM, ця пам'ять може бути корисною для зберігання коду замку, якщо ми розробляємо програму, як-от електронний дверний замок.

3. SRAM : статична пам'ять із довільним доступом, це енергонезалежна пам'ять мікроконтролера, тобто дані втрачаються, як тільки живлення вимикається. Atmega16 оснащений 1 КБ внутрішньої SRAM. Невелика частина SRAM відведена для регістрів загального призначення, які використовуються центральним процесором, а частина — для периферійних підсистем мікроконтролера.

ISP : сімейство контролерів AVR має програмовану флеш-пам'ять, яку можна програмувати, не видаляючи мікросхему зі схеми, ISP дозволяє перепрограмувати контролер, поки він знаходиться в схемі програми.

SPI : послідовний периферійний інтерфейс , порт SPI використовується для послідовного зв'язку між двома пристроями на спільному джерелі тактування. Швидкість передачі даних SPI більша, ніж у USART.

TWI : двопровідний інтерфейс (TWI) можна використовувати для створення мережі пристроїв, багато пристроїв можуть бути підключені через інтерфейс TWI, утворюючи мережу, пристрої можуть одночасно передавати і приймати і мати свою унікальну адресу.

ЦАП : Atmega16 також оснащений інтерфейсом цифро-аналогового перетворювача (ЦАП), який можна використовувати для зворотної дії, що виконується АЦП. ЦАП можна використовувати, коли є необхідність перетворення цифрового сигналу в аналоговий.

Сімейство MegaAVR

ATmega8 і Atmega32 є іншими членами контролерів серії MegaAVR. Вони дуже схожі на ATmega16 за архітектурою. На ринках також доступні контролери MegaAVR малої потужності. У таблиці 2.3 показано порівняння між різними членами сімейства MegaAVR:

Таблиця 2.3 – Характеристики мікроконтролерів сімейства MegaAVR

Назва частини	ROM	ОЗП	EEPROM	I/O Pins	Таймер	Переривання	Робоча напруга	Робоча частота	Упаковка
ATmega8	8 КБ	1 КБ	512B	23	3	19	4,5-5,5 В	0-16 МГц	28

Продовження таблиці 2.3.

ATmega 8L	8 КБ	1 КБ	512В	23	3	19	2,7-5,5 В	0-8 МГц	28
ATmega 16	16 КБ	1 КБ	512В	32	3	21	4,5-5,5 В	0-16 МГц	40
ATmega 16L	16 КБ	1 КБ	512В	32	3	21	2,7-5,5 В	0-8 МГц	40
ATmega 32	32 КБ	2 КБ	1 КБ	32	3	21	4,5-5,5 В	0-16 МГц	40
ATmega 32L	32 КБ	2 КБ	1 КБ	32	3	21	2,7-5,5 В	0-8 МГц	40

2.2 Налагоджувальні плати Arduino

Arduino — це платформа з відкритим кодом, яка використовується для створення електронних проєктів. Arduino складається як із фізичної програмованої плати (часто її називають мікроконтролером), так і з частини програмного забезпечення або IDE (Інтегроване середовище розробки), яке працює на вашому комп'ютері, використовується для запису та завантаження комп'ютерного коду на фізичну плату [10].

Апаратне та програмне забезпечення Arduino було розроблено для художників, дизайнерів, любителів, хакерів, новачків і всіх, хто цікавиться створенням інтерактивних об'єктів або середовищ. Arduino може взаємодіяти з кнопками, світлодіодами, двигунами, динаміками, GPS-пристроями, камерами, Інтернетом і навіть смартфоном або телевізором. Ця гнучкість у поєднанні з тим фактом, що програмне забезпечення Arduino є безкоштовним, апаратні плати досить дешеві, а програмне й апаратне забезпечення легко освоїти, привели до

великої спільноти користувачів, які надали код та випустили інструкції для величезної різноманітності проектів на базі Arduino.

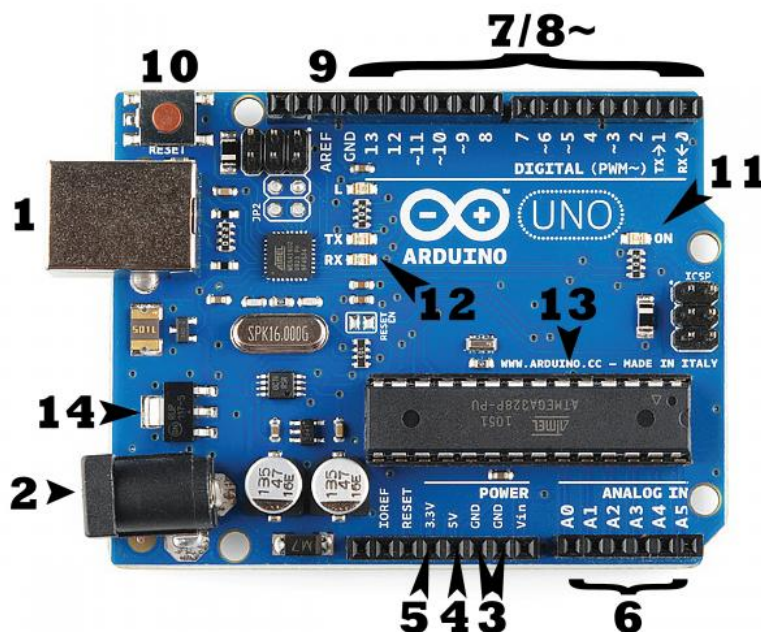


Рисунок 2.5 – Розміщення основних компонентів плати [11]

Існує багато різновидів плат Arduino, які можна використовувати для різних цілей. Деякі плати виглядають трохи інакше, ніж наведені нижче, але більшість Arduino мають більшість цих спільних компонентів рис. 2.5.

Живлення

Кожна плата Arduino потребує способу підключення до джерела живлення. Arduino UNO можна жити від USB-кабелю, що йде від вашого комп'ютера, або від настінного джерела живлення, який підключений до гнізда. На рис.2.5 з'єднання USB позначено **(1)**, а роз'єм для ствола – **(2)**. USB-з'єднання – це також спосіб завантаження коду на плату Arduino. Рекомендована напруга для більшості моделей Arduino становить від 6 до 12 Вольт.

Виводи

Шпильки на Arduino – це місця, де з'єднуються дроти для побудови схем. Зазвичай вони мають чорні пластикові «роз'єми», які дозволяють просто підключити дріт прямо в плату. Arduino має кілька різних типів контактів, кожен з яких позначений на платі та використовується для різних функцій.

- **GND (3)** : скорочення від «земля». На Arduino є кілька контактів GND, будь-який з яких можна використовувати для заземлення схеми.
- **5 В (4) і 3,3 В (5)** : Контакт 5 В подає 5 вольт живлення, а контакт 3,3 В – 3,3 вольта. Більшість простих компонентів, які використовуються з Arduino, працюють від 5 або 3,3 вольт.
- **Аналоговий (6)** : Область контактів під міткою «Аналоговий вхід» (A0–A5 на UNO) є контактами аналогового входу. Ці контакти можуть зчитувати сигнал з аналогового датчика (наприклад, датчик температури) і перетворювати його в цифрове значення, яке ми можемо прочитати.
- **Цифровий (7)** : Навпроти аналогових контактів розташовані цифрові контакти (від 0 до 13 на UNO). Ці контакти можна використовувати як для цифрового входу (наприклад, щоб дізнатися, чи натиснуто кнопку), так і для цифрового виходу (наприклад, для живлення світлодіода).
- **ШИМ (8)** : Знак тильда (~) біля деяких цифрових контактів (3, 5, 6, 9, 10 і 11 на UNO) діють як звичайні цифрові контакти, але також можуть використовуватися для широтно-імпульсної модуляції (ШИМ). Ці контакти здатні імітувати аналоговий вихід (наприклад, згасання світлодіода).
- **AREF (9)** : Означає аналогове опорне значення. У більшості випадків ви можете залишити цю шпильку в спокої. Іноді використовується для встановлення зовнішньої опорної напруги (від 0 до 5 вольт) як верхньої межі для контактів аналогового входу.

Кнопка скидання

Arduino має кнопку скидання **(10)**. Натискання на неї тимчасово з'єднає контакт скидання із землею та перезапустить будь-який код, завантажений на Arduino. Це може бути дуже корисно, якщо ваш код не повторюється, але ви хочете перевірити його кілька разів. .

Світлодіодний індикатор живлення

Світлодіод поруч зі словом «ON» **(11)** повинен світитися щоразу, коли ви підключаєте Arduino до джерела живлення.

Світлодіоди TX RX

TX скорочення для передачі, RX скорочення для прийому. Ці позначки досить часто зустрічаються в електроніці для позначення контактів, що відповідають за послідовний зв'язок. У нашому випадку на Arduino UNO є два місця, де відображаються TX і RX - один раз за допомогою цифрових контактів 0 і 1, а другий раз поруч зі світлодіодами індикаторів TX і RX **(12)**. Ці світлодіоди дадуть нам гарні візуальні індикації, коли Arduino отримує або передає дані (наприклад, коли ми завантажуюмо нову програму на плату).

Основна IC

Інтегральна схема **(13)**. Це мозок Arduino. Основна мікросхема на Arduino дещо відрізняється від плати до плати, але зазвичай належить до лінійки мікросхем ATmega від компанії ATMEL. Це може бути важливим, оскільки може знадобитися знати тип IC (разом із типом плати), перш ніж завантажувати нову програму з програмного забезпечення Arduino. Цю інформацію зазвичай можна знайти в письмовому вигляді на верхній стороні IC.

Регулятор напруги

Регулятор напруги **(14)**. Стабілізатор напруги контролює кількість напруги, яка надходить на плату Arduino. Звичайно, він має свої межі, тому не слід підключати Arduino до джерела живлення, що перевищує 20 вольт.

Зазвичай, щоб створити прототип або пристрій на платформі Arduino, потрібна прототипна плата, оснащена мікроконтролером того типу, який би відповідав нашим потребам, модуль розширення, програмне забезпечення Arduino IDE, кабель USB, блок живлення та ПК. В даний час екосистема включає в себе різні типи Arduino, з різними мікроконтролерами і величезною кількістю модулів розширення.

Найпоширеніші плати Arduino

Arduino Uno — це плата мікроконтролера на основі ATmega328P. Вона має 14 цифрових входів/виходів (з яких 6 можна використовувати як ШІМ-виходи), 6 аналогових входів, керамічний резонатор 16 МГц (CSTCE16M0V53-R0), USB-

з'єднання, роз'єм живлення, роз'єм ICSP і кнопку скидання. Вона містить все необхідне для підтримки мікроконтролера рис 2.6.

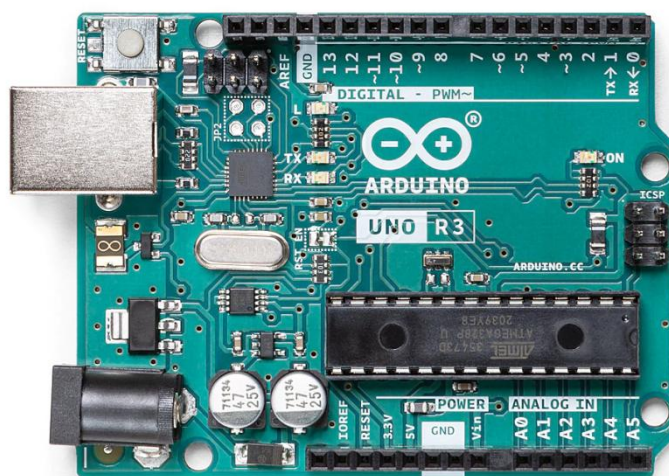


Рисунок 2.6 – Вигляд плати Arduino Uno [12]

«Uno» означає один на італійській мові. Вона була обрана для відзначення випуску програмного забезпечення Arduino (IDE) 1.0. Плата Uno і версія 1.0 програмного забезпечення Arduino (IDE) були еталонними версіями Arduino, які тепер розвинулися до нових версій. Плата Uno є першою в серії плат USB Arduino та еталонною моделлю для платформи Arduino. Характеристики плати Arduino Uno наведено в таблиці 2.4.

Таблиця 2.4 – Характеристики плати Arduino Uno [12]

МІКРОКОНТРОЛЕР	<u>ATmega328P</u>
РОБОЧА НАПРУГА	5В
ВХІДНА НАПРУГА (РЕКОМЕНДОВАНО)	7-12В
ВХІДНА НАПРУГА (ЛІМІТ)	6-20В

Продовження таблиці 2.4.

ЦИФРОВІ КОНТАКТИ ВВОДУ/ВИВОДУ	14 (з яких 6 забезпечують вихід ШІМ)
ШІМ ЦИФРОВІ КОНТАКТИ ВВОДУ/ВИВОДУ	6
ВИВОДИ АНАЛОГОВОГО ВХОДУ	6
ПОСТІЙНИЙ СТРУМ НА КОНТАКТ ВВЕДЕННЯ/ВИВОДУ	20 мА
ПОСТІЙНИЙ СТРУМ ДЛЯ КОНТАКТУ 3,3 В	50 мА
ФЛЕШ-ПАМ'ЯТЬ	32 КБ (АТmega328Р), з яких 0,5 КБ використовується завантажувачем
SRAM	2 КБ (АТmega328Р)
EEPROM	1 КБ (АТmega328Р)
ТАКТОВА ШВИДКІСТЬ	16 МГц
LED_BUILTIN	13
ДОВЖИНА	68,6 мм
ШИРИНА	53,4 мм
ВАГА	25 г

Arduino Mega 2560 — це плата мікроконтролера на базі АТmega2560 . Вона має 54 цифрових контакти входу/виходу (з яких 15 можна використовувати як виходи ШІМ), 16 аналогових входів, 4 UART (апаратні послідовні порти), кварцевий генератор 16 МГц, USB-з'єднання, роз'єм живлення, роз'єм ICSP, і

кнопка скидання рис 2.7 . Характеристики плати Arduino Mega наведено в таблиці 2.5.



Рисунок 2.7 – Вигляд плати Arduino Mega [13]

Таблиця 2.5 – Характеристики плати Arduino Mega2560 [13]

МІКРОКОНТРОЛЕР	<u>ATmega2560</u>
РОБОЧА НАПРУГА	5В
ВХІДНА НАПРУГА (РЕКОМЕНДОВАНО)	7-12В
ВХІДНА НАПРУГА (ЛІМІТ)	6-20В
ЦИФРОВІ КОНТАКТИ ВВОДУ/ВИВОДУ	54 (з яких 15 забезпечують вихід ШІМ)
ВИВОДИ АНАЛОГОВОГО ВХОДУ	16
ПОСТІЙНИЙ СТРУМ НА КОНТАКТ ВВЕДЕННЯ/ВИВОДУ	20 мА
ПОСТІЙНИЙ СТРУМ ДЛЯ КОНТАКТУ 3,3 В	50 мА

Продовження таблиці 2.5.

ФЛЕШ-ПАМ'ЯТЬ	256 КБ, з яких 8 КБ використовується завантажувачем
SRAM	8 КБ
EEPROM	4 КБ
ТАКТОВА ШВИДКІСТЬ	16 МГц
LED_BUILTIN	13
ДОВЖИНА	101,52 мм
ШИРИНА	53,3 мм
ВАГА	37 г

Arduino Nano — це невелика, укомплектована і зручна для роботи плата на основі ATmega328 рис 2.8.

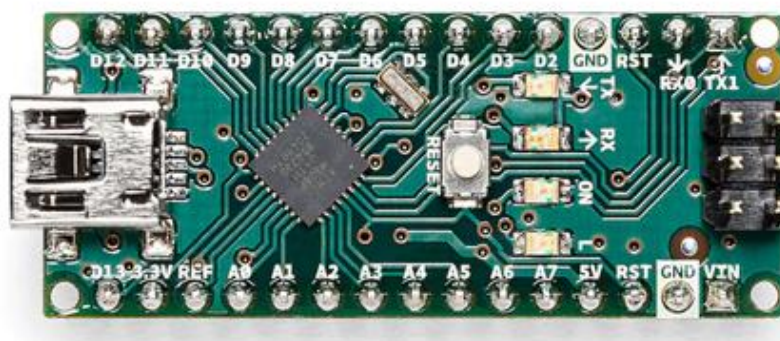


Рисунок 2.8 – Вигляд плати Arduino Nano [14]

У неї відсутній лише роз'єм для живлення постійного струму. Плата працює із USB-кабелем Mini-B замість стандартного. Характеристики плати Arduino Nano наведено в таблиці 2.6.

Таблиця 2.6 – Характеристики плати Arduino Nano [14]

МІКРОКОНТРОЛЕР	ATmega328
АРХІТЕКТУРА	AVR
РОБОЧА НАПРУГА	5 В
ФЛЕШ-ПАМ'ЯТЬ	32 КБ, з яких 2 КБ використовується завантажувачем
SRAM	2 КБ
ТАКТОВА ШВИДКІСТЬ	16 МГц
АНАЛОГОВІ КОНТАКТИ IN	8
EEPROM	1 КБ
ПОСТІЙНИЙ СТРУМ НА КОНТАКТИ ВВОДУ/ВИВОДУ	40 мА (контакти введення-виводу)
ВХІДНА НАПРУГА	7-12В
ЦИФРОВІ КОНТАКТИ ВВОДУ/ВИВОДУ	22 (6 з яких ШІМ)
ШІМ ВИХІД	6
СПОЖИВАННЯ ЕНЕРГІЇ	19 мА
РОЗМІР РСВ	18 x 45 мм
ВАГА	7 г

Arduino Micro — це плата мікроконтролера, заснована на ATmega32U4. Розроблена спільно з Adafruit . Вона має 20 цифрових входів/виходів (7 з яких можна використовувати як ШІМ-виходи, а 12 як аналогові входи), кварцевий

генератор 16 МГц, з'єднання micro USB, роз'єм ICSP і кнопку скидання. Вона містить все необхідне для підтримки мікроконтролера рис 2.9. Вона має форм-фактор, який дозволяє легко розмістити її на макетній дошці. Характеристики плати Arduino Micro наведено в таблиці 2.7.

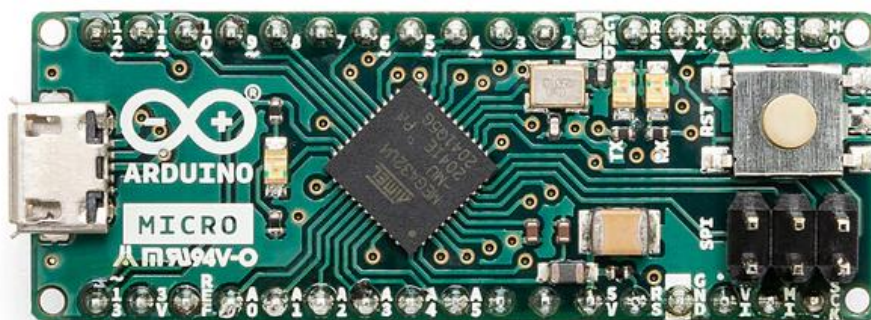


Рисунок 2.9 – Вигляд плати Arduino Micro [15]

Плата Micro схожа на Arduino Leonardo тим, що ATmega32U4 має вбудований USB-зв'язок, що усуває необхідність у вторинному процесорі. Це дозволяє Micro відображатися на підключеному комп'ютері як миша та клавіатура, на додаток до віртуального (CDC) послідовного / COM порту.

Таблиця 2.7 – Характеристики плати Arduino Micro [15]

МІКРОКОНТРОЛЕР	<u>ATmega32U4</u>
РОБОЧА НАПРУГА	5В
ВХІДНА НАПРУГА (РЕКОМЕНДОВАНО)	7-12В
ІІІМ КАНАЛИ	7

Продовження таблиці 2.7.

ЦИФРОВІ КОНТАКТИ ВВОДУ/ВИВОДУ	20
АНАЛОГОВІ ВХІДНІ КАНАЛИ	12
ПОСТІЙНИЙ СТРУМ НА КОНТАКТ ВВЕДЕННЯ/ВИВОДУ	20 мА
ПОСТІЙНИЙ СТРУМ ДЛЯ КОНТАКТУ 3,3 В	50 мА
ФЛЕШ-ПАМ'ЯТЬ	32 КБ (АТmega32U4), з яких 4 КБ використовується завантажувачем
SRAM	2,5 КБ (АТmega32U4)
EEPROM	1 КБ (АТmega32U4)
ТАКТОВА ШВИДКІСТЬ	16 МГц
LED_BUILTIN	13
ДОВЖИНА	48 мм
ШИРИНА	18 мм
ВАГА	13 г

Arduino Leonardo — це плата мікроконтролера на основі АТmega32u4 . Вона має 20 цифрових контактів входу/виходу (з яких 7 можна використовувати як ШІМ-виходи і 12 як аналогові входи), кварцевий генератор 16 МГц, з'єднання micro USB, роз'єм живлення, роз'єм ICSP і кнопку скидання рис 2.10. Характеристики плати Arduino Leonardo наведено в таблиці 2.8.

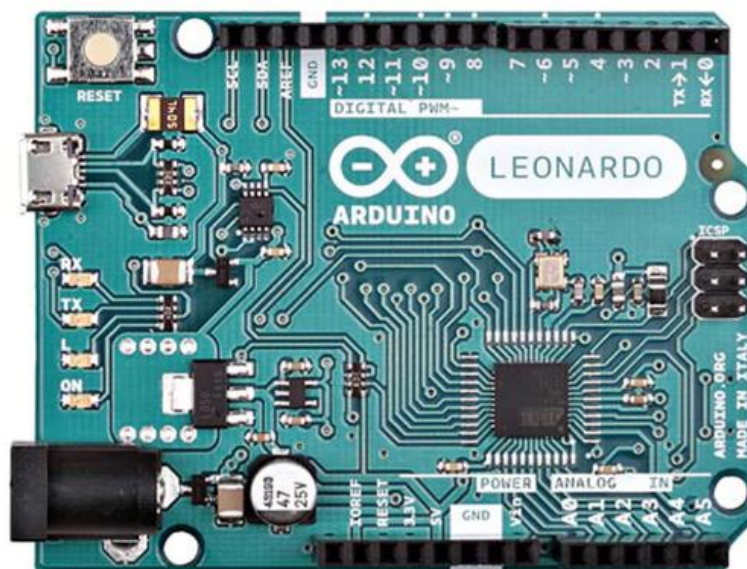


Рисунок 2.10 – Вигляд плати Arduino Leonardo [16]

Leonardo відрізняється від усіх попередніх плат тим, що ATmega32u4 має вбудований USB-зв'язок, що усуває необхідність у вторинному процесорі. Це дозволяє Леонардо відображатися на підключеному комп'ютері як миша і клавіатура, а також віртуальний (CDC) послідовний / COM-порт.

Таблиця 2.8 – Характеристики плати Arduino Leonardo [16]

МІКРОКОНТРОЛЕР	ATmega32u4
РОБОЧА НАПРУГА	5В
ВХІДНА НАПРУГА (РЕКОМЕНДОВАНО)	7-12В
ВХІДНА НАПРУГА (МЕЖІ)	6-20В
ЦИФРОВІ КОНТАКТИ ВВОДУ/ВИВОДУ	20

Продовження таблиці 2.8.

ІІМ КАНАЛИ	7
АНАЛОГОВІ ВХІДНІ КАНАЛИ	12
ПОСТІЙНИЙ СТРУМ НА КОНТАКТ ВВЕДЕННЯ/ВИВОДУ	40 мА
ПОСТІЙНИЙ СТРУМ ДЛЯ КОНТАКТУ 3,3 В	50 мА
ФЛЕШ-ПАМ'ЯТЬ	32 КБ (АТmega32u4), з яких 4 КБ використовується завантажувачем
SRAM	2,5 КБ (АТmega32u4)
EEPROM	1 КБ (АТmega32u4)
ТАКТОВА ШВИДКІСТЬ	16 МГц
ДОВЖИНА	68,6 мм
ШИРИНА	53,3 мм
ВАГА	20 г

2.3 Програмне забезпечення для роботи з мікроконтролером

Arduino IDE — це інструмент з відкритим вихідним кодом, який дозволяє користувачам писати, а також завантажувати код у робоче середовище в режимі реального часу. Оскільки написаний код буде переміщено в хмару, ним часто користуються ті, кому потрібен додатковий рівень резервування.

Arduino IDE пропонує повну сумісність з будь-якою програмною платою на базі Arduino. Програмне забезпечення можна легко розгорнути в будь-якій

операційній системі Linux, Mac або Windows. Більшість його частин написані на JavaScript для безперебійної компіляції та редагування. Хоча головна мета інструмента заснована на написанні коду, він пропонує кілька важливих функцій. Наприклад, Arduino IDE дозволяє користувачам ділитися важливою інформацією про проект із зацікавленими сторонами компанії. Користувачам надається свобода вносити внутрішні макети та змінювати схеми, коли це необхідно. Для тих, кому потрібна допомога в процесі встановлення, доступні вичерпні посібники. Підручники доступні для користувачів, які мають невеликий досвід роботи з фреймворком інструмента. Arduino IDE високо оцінили користувачі за простоту використання.

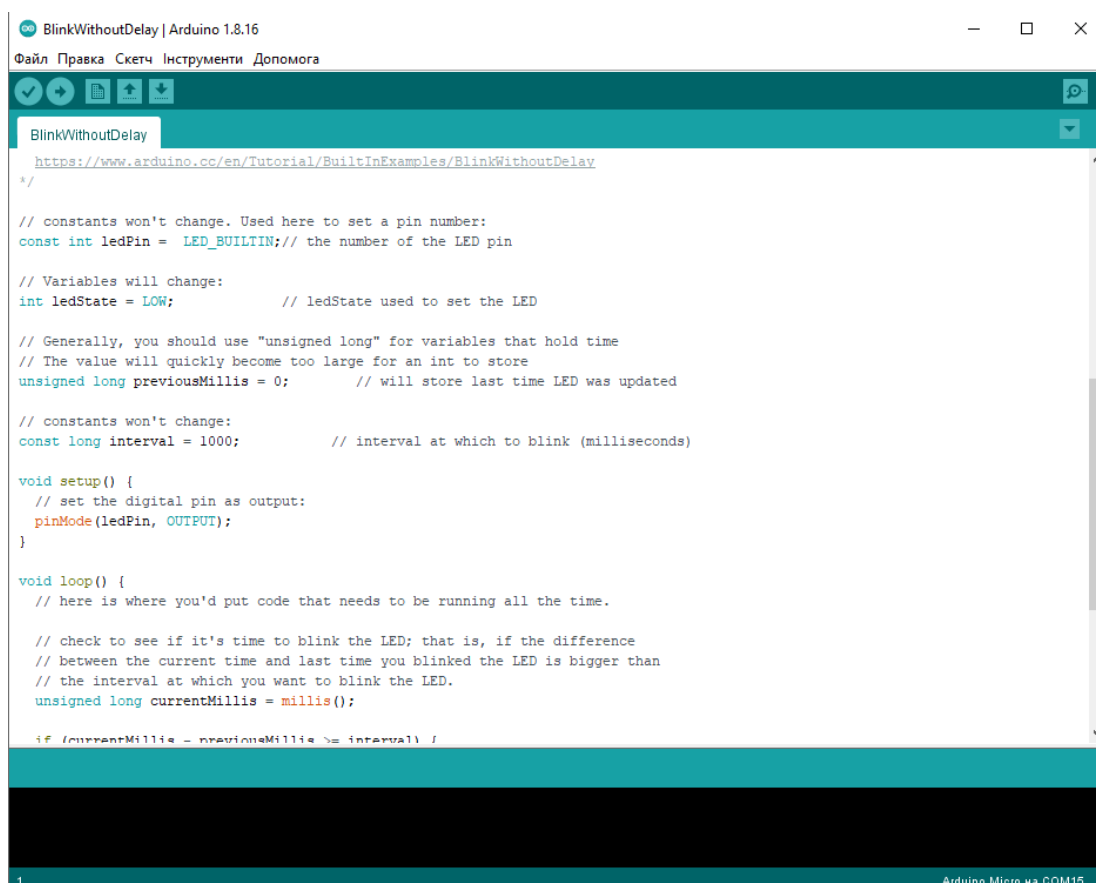


Рисунок 2.11 – Інтерфейс програмного середовища Arduino IDE

Він може виконувати складні процеси, зводячи ресурси комп'ютера до мінімуму. Інструмент полегшує користувачам доступ до своїх бібліотек. У той же час він пропонує оновлену підтримку останніх плат Arduino, які можуть допомогти користувачам створювати ескізи за допомогою останньої версії

IDE. Інструмент полегшує користувачам доступ до своїх бібліотек. У той же час він пропонує оновлену підтримку останніх плат Arduino, які можуть допомогти користувачам створювати ескізи за допомогою останньої версії IDE.

Бібліотеки Arduino — це зручний спосіб обміну кодом, таким як драйвери пристроїв або часто використовувані допоміжні функції. Arduino IDE постачається з набором стандартних бібліотек для широко використовуваних функцій. Ці бібліотеки підтримують усі приклади, що входять до IDE. Функціональні можливості стандартної бібліотеки включають основні функції зв'язку та підтримку деяких найпоширеніших типів обладнання, таких як серводвигуни та символічні РК-дисплеї.

2.4 Висновки до розділу

Досліджено архітектуру мікроконтролерів AVR. Розглянуто основні архітектурні блоки Atmega16. Проаналізовано основні відмінності різних категорій мікроконтролерів AVR, та сфери їх використання. Проведено порівняння сімейств мікроконтролерів AVR, PIC та 8051.

Розглянуто основні компоненти налагоджуваних плат Arduino, та їх роль. Проаналізовано технічні характеристики різних налагоджуваних плат Arduino. Наведено інформацію про їх технічні характеристики.

Розглянуто програмне середовище для роботи з налагоджувальними платами. Наведено основні можливості та функціонал.

3 СТВОРЕННЯ ТРЕНАЖЕРУ ДЛЯ ВОДІЯ

3.1 Вибір компонентів

Першочерговим завданням при створенні будь-якого пристрою є вибір компонентної бази. Для створення електронного тренажеру для водія необхідно визначитись з такими компонентами.

- Налагоджувана плата
- Датчик положення валу руля
- Датчик положення педаль
- Датчики положення важелів повороту та склоочисника

Вибір компонентів на пряму впливає на функціональні можливості тренажеру, особливості його експлуатації, налаштування та

Налагоджувальна плата

Налагоджувальна плата – це мозок приладу. Саме вона буде оброблювати всі команди користувача та комунікувати з персональним комп'ютером.

При виборі налагоджувальної плати потрібно звертати увагу на такі характеристики.

- Мікроконтролер
- Кількість цифрових портів
- Кількість аналогових портів
- Живлення мікроконтролера
- Обсяг Flash пам'яті

Першочергове значення має вибір мікроконтролера. Оскільки тренажер планується використовувати в якості ігрового руля в симуляторах для водіїв, то мікроконтролер повинен сприйматись персональним комп'ютером не тільки як віртуальний COM порт, а як HID пристрій. Для цього в мікроконтролері повинен бути інтегрований конвертер USB, який дозволить комп'ютеру сприймати мікроконтролер як клавіатуру, або мишку. В налагоджувальних платах Arduino, вбудований інтегрований конвертор USB використовують мікроконтролери ATmega32u4. Найпоширенішими налагоджувальними платами Arduino, що

використовують даний тип мікроконтролера являються Arduino Leonardo та Arduino Micro. Характеристики вказаних налагоджувальних плат наведені вище.

Перевагою використання Arduino Micro є її ціна та габарити. Різниця в ціні між Arduino Micro та Arduino Leonardo є незначною, але функціонально, вона поступається конкуренту. Хоча в технічних характеристиках цих плат вказана однакова кількість цифрових та аналогових входів-виходів, на практиці вона відрізняється. Фактично в платі Arduino Leonardo (рис. 3.1) є 12 цифрових пінів (7 з яких можна використовувати як ШІМ) та 6 аналогових (які можна використовувати як цифрові), які можна використовувати при розробці тренажеру, тоді як на платі Arduino Micro є 12 цифрових пінів та 4 аналогових.

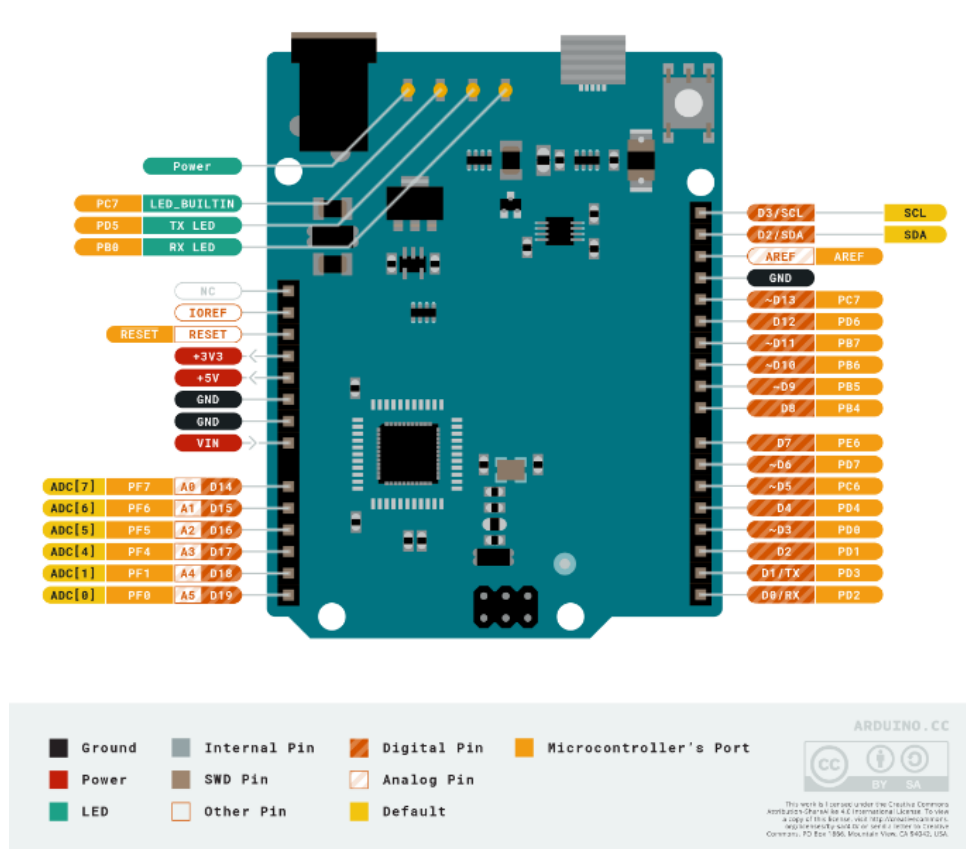


Рисунок 3.1 Розпінування плати Arduino Leonardo [17]

Оскільки плата Arduino Leonardo має більшу кількість аналогових входів-виходів (що грає ключову роль), то буде використовуватись саме вона.

Датчик положення валу руля

Наступним елементом з яким потрібно визначитись, є датчик положення валу руля. Цей датчик повинен зчитувати в якому положенні перебуває руль і відповідно до отриманих даних Є два елементи, які можна використовувати в якості датчика: потенціометр або енкодер. Перевагою потенціометра є його ціна, простота використання та кількість портів, які використовуються для його підключення. Слабкою стороною потенціометра є його точність. При створенні тренажеру може використовуватись енкодер, проте для досягнення високої точності вимірювань прийдеться використовувати оптичний енкодер, який коштує досить дорого та потребує додаткових механізмів в тренажері (калібрування положення руля в момент його підключення до комп'ютера). Також енкодер використовуватиме більше портів мікроконтролера. Виходячи з цих міркувань, в даному проекті доцільніше використовувати потенціометр на 20 кОм (рис 3.2).

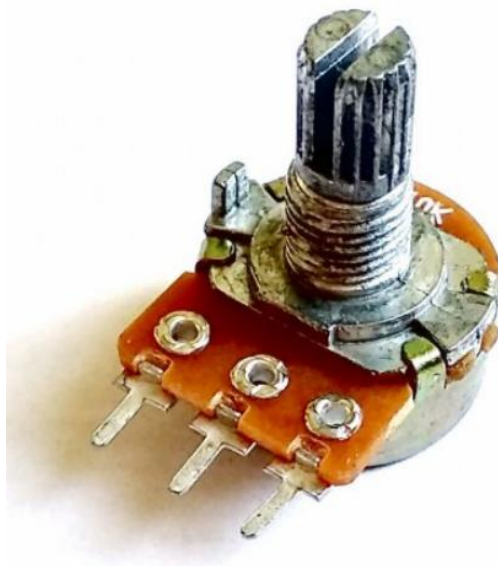


Рисунок 3.2 – Потенціометр на 20к Ом [18]

Датчик положення педаль

В тренажері планується використання трьох педаль, які матимуть однаковий принцип роботи, тому і датчики будуть однаковими. По аналогії з датчиком

положення валу руля, доцільно використовувати три потенціометри на 20 kОм, із-за зручності їх встановлення та використання.

Датчики положення важелів повороту та склоочисника

Важелі поворотів та склоочисників матимуть по три положення. Для забезпечення їх функціонування можна створити власний механізм перемикачів режимів, або використати готові прилади, які можуть забезпечити коректну роботу пристрою. В роботі буде використовуватись готовий варіант механізму перемикачів режимів, а саме: тумблер на три положення (рис. 3.3). Він забезпечить надійне та чітке перемикачів режимів, та спростить будову тренажеру.



Рисунок 3.3 – Тумблер на три положення [19]

3.2 Особливості конструкції тренажеру

Особливістю створеного тренажеру є максимальна наближеність по функціоналу до реального водійського керма. Тренажер має три механізми, яких немає жоден аналог, а саме.

- Важіль поворотів (рис. 3.4)
- Важіль склоочисника (аналогічний важелю поворотів)
- Перемикач режиму освітлення (рис. 3.4)



Рисунок 3.4 Вигляд важелю повороту та перемикача режиму освітлення

Існуючі ігрові маніпулятори позбавлені цих механізмів, оскільки для повсякденного використання (у відеоіграх) користувачі не потребують даного функціоналу.

Він вважається надлишковим. Але для повноцінного навчання водінню та формування правильної моторики рук під час водіння, саме ці функціональні вузли відіграють значну роль. При навчанні водінню людей, які ще ні разу не водили автомобіль, це дозволить сформувати розуміння, де розташовуються органи керування індикацією, освітленням та допоміжними засобами автомобіля.

В тому випадку, коли тренажером будуть користуватись водії-початківці, їм не прийдеться підлаштовуватись під маніпулятор, оскільки всі органи керування будуть знаходитись саме в тих місцях де й повинні бути.

3.3 Переваги використання

Електронний тренажер матиме низку вагомих переваг та недоліків над існуючими аналогами.

Основні переваги.

- **Новий функціонал** дозволить використати потенціал симуляторів для водія у повному обсязі .
- **Модульність.** Кожен елемент тренажера під'єднуватиметься до «бази» руля окремо, за допомогою конекторів RJ45 (рис.3.5).



Рисунок 3.5 – Конектор RJ45 [20]

Це дозволить під'єднувати коробку передач та педалі тільки тоді, коли цього буде потребувати користувач. Це додасть зручності для транспортування та зберігання тренажеру. Також, користувач зможе самостійно замінити пошкоджений кабель, чи підібрати кабель з оптимальною для нього довжиною.

- **Відсутність необхідності під'єднання додаткового живлення.** Тренажер функціонуватиме за рахунок під'єднання до комп'ютера через USB порт.
- Рульова база, педалі та коробка передач будуть відображатись, як **один ігровий пристрій**. Це дасть змогу користувачу використовувати тренажер в симуляторах, які не підтримують два ігрові пристрої.

- Тренажер може використовуватись **як для ігор, так і для навчання** водінню підлітків чи водіїв-початківців в автошколах.

Недолік тренажеру.

- **Відсутність зворотної віддачі руля.** Для впровадження цієї функції необхідно напряду взаємодіяти з ігровим середовищем.

3.4 Конструкція тренажеру

Тренажер виготовлено з використанням пластикових та металевих елементів. Більшість пластикових елементів була надрукована на 3D принтері по оригінальним 3D моделям.

Тренажер складається з трьох елементів.

- База
- Коробка передач
- Педалі

База

База вважається основою тренажеру, оскільки в собі вона містить його ключові елементи, такі як: плата управління, кермо, важелі поворотів та склоочисника, роз'єми для під'єднання педаль та коробки передач та роз'єм для підключення до комп'ютера.



Рисунок 3.6 – База електронного тренажеру

В центральній частині керма, під кришкою, знаходиться чотири кнопки (по одній кнопці з кожного краю кришки), що з'єднанні між собою послідовно. Завдяки цьому при натисканні на центральну частину в будь – якому місті, як мінімум одна кнопка спрацює. Таким чином, це дає змогу використовувати центральну частину керма для подання звукового сигналу в симуляторі.

Кермо з'єднане з базою за рахунок металевої осі, яка проходить через всю базу. На початку і в кінці бази розміщені підшипники, що робить обертання керма легким та плавним. В тренажері реалізована зворотна тяга керма, що повертає його в центральне положення при його відпусканні.

Важелі поворотів та склоочисників прикріплені до тумблерів, що встановлені з двох сторін бази. За рахунок використання тумблерів, досягнуто чітке перемикання положень важелів та характерного звуку.

Педалі

Вся конструкція педаль виконана з металу задля надання їй необхідної міцності (рис. 3.7). Особливістю даної конструкції є принцип приведення педалей в рух.



Рисунок 3.7 – Педалі електронного тренажера

В більшості сучасних ігрових маніпуляторах, взаємодія користувача з педалями відбувається за рахунок натиску педаль вниз (рис 3.8), тоді як в справжньому авто, для того щоб привести педаль в рух, потрібно натискати педаль вперед; Така конструкція є більш наближеною до тієї, що використовується в автомобілях, а отже, краще підходить для навчання водіїв.



Рисунок 3.8 – Вигляд ігрового маніпулятора Logitech G29 Driving Force [21]

Каркас педаль має два металеві обмежувачі, які не дають педалям підніматись та опускатись нижче передбачених для цього діапазонів. Реалізовано

механізм повернення педаль в початкове положення, при їх відпусканні. Натискання на педалі обробляється платою Arduino за допомогою зчитування показань з потенціометрів, встановлених на нижньому обмежувачі (рис. 3.9).

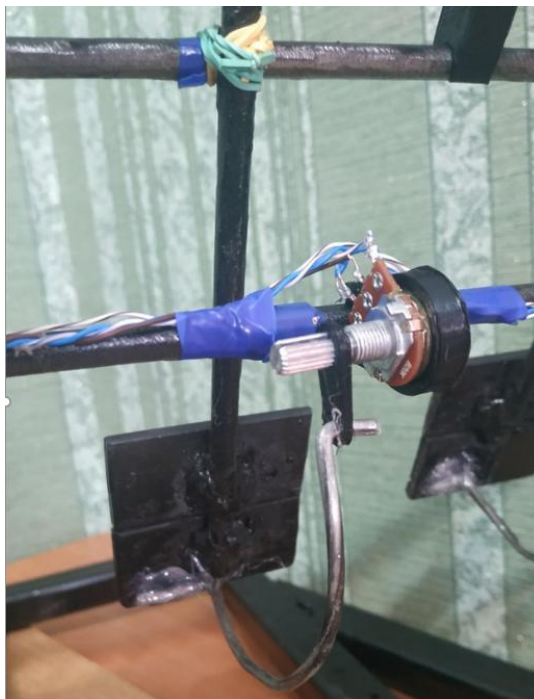


Рисунок 3.9 – Механізм взаємодії педалі з потенціометром

Коробка передач

Не менш важливим елементом керування є коробка передач (рис. 3.10). Корпус коробки та всі кріплення виконані з пластику. В якості важелю використано металевий стержень, оскільки він буде використовуватись в якості провідника. В середині корпусу, реалізовано механізм переведення важелю в центральне положення (нейтральна передача). Важіль прикріплений до дна коробки рухомим механізмом.

Коробка передач має 6 активних положень та одне пасивне (нейтральна передача). Закріплення важелю в активних положеннях відбувається за рахунок пластикових фіксаторів, кожен з яких має свій контакт. При фіксації важеля, контакт замикається з важельом і передає сигнал на плату, де оброблюється і передається комп'ютеру.



Рисунок 3.10 – Коробка передач

Схема під'єднання електронних елементів наведена на рисунку 3.11. Для покращення сприймання схеми, контактні доріжки, що розміщені на фіксаторах коробки передач та механізму перемикавання режиму фар, представлені у вигляді тактових кнопок.

При програмуванні тренажеру було враховано специфіку роботи з органами управління в авто-симуляторах. Лістинг коду наведено в Додатку А.

Демонстрація роботи тренажеру в симуляторі «3D Інструктор» наведена в презентації диплому.

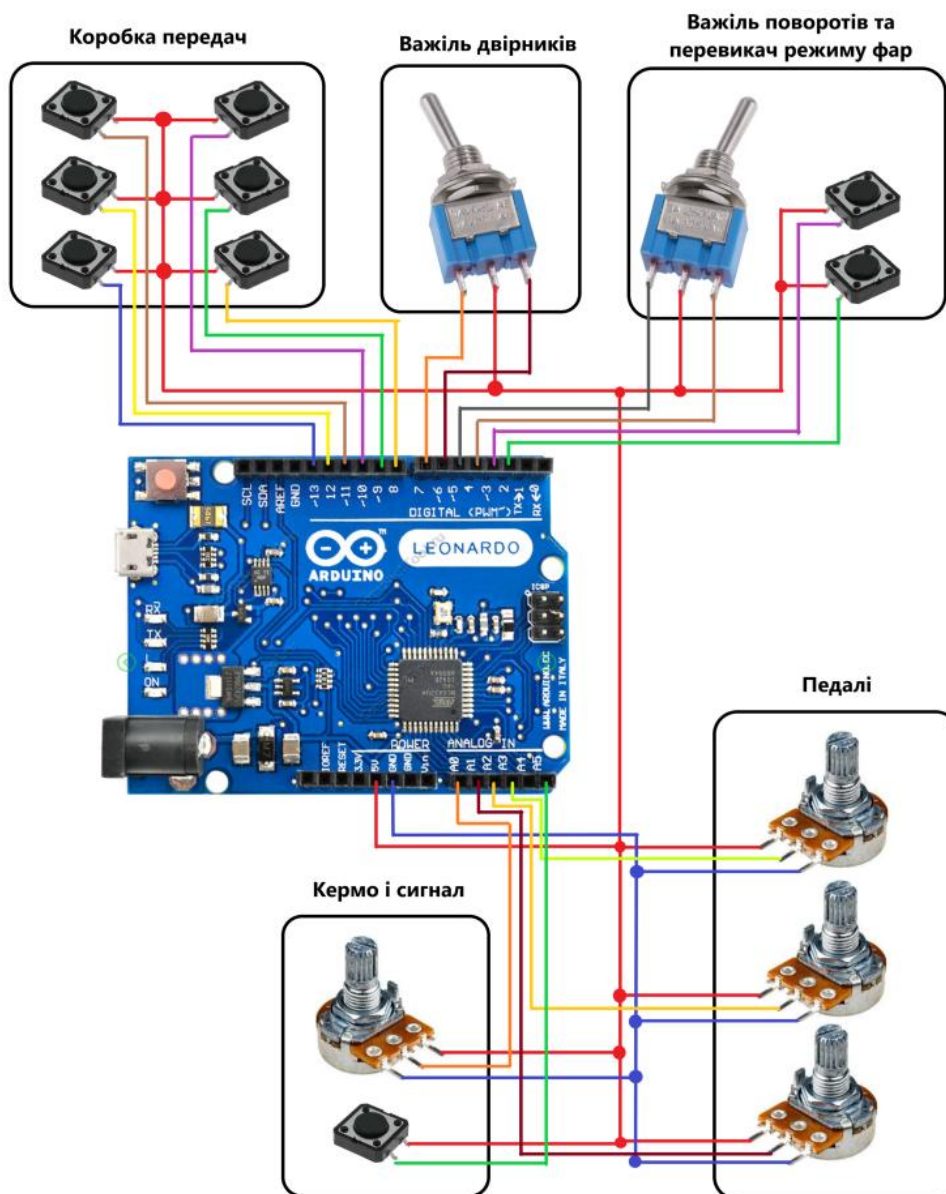


Рисунок 3.11 – Схема під'єднання елементів системи до плати

3.5 Висновки до розділу

Аргументовано критерії вибору компонентів для виготовлення електронного тренажера для водія.

Наведено основні компоненти, які використовуватимуться при його виробництві.

Описано додаткові функції тренажера, яких не мають конкуренти.

Проаналізовано комерційний ринок існуючих аналогів. Вказано на переваги та недоліки створеного електронного тренажеру для водія.

Розглянуто будову окремих елементів системи. Вказано реалізовані механізми та функції тренажеру. Наведено схему під'єднання електронних компонентів до плати.

ВИСНОВКИ

В дипломній роботі розглянуто історію виникнення перших мікроконтролерів, які дали поштовх розвитку сучасних електронних систем, що використовуються в різних сферах життя. Розглянуто сфери застосування мікроконтролерів в електронних системах на конкретних прикладах.

Досліджено архітектуру мікроконтролерів AVR. Розглянуто основні архітектурні блоки Atmega16. Проаналізовано основні відмінності різних категорій мікроконтролерів AVR, та сфери їх використання. Проведено порівняння сімейств мікроконтролерів AVR, PIC та 8051.

Розглянуто основні компоненти налагоджуваних плат Arduino, та їх роль. Проаналізовано технічні характеристики різних налагоджуваних плат Arduino. Наведено інформацію про їх технічні характеристики. Розглянуто програмне середовище для роботи з налагоджувальними платами. Наведено основні можливості та функціонал.

Аргументовано критерії вибору компонентів для виготовлення електронного тренажеру для водія. Наведено основні компоненти, які використовуються при його виробництві.

Описано додаткові функції тренажеру, яких не мають конкуренти. Проаналізовано комерційний ринок існуючих аналогів. Вказано на переваги та недоліки створеного електронного тренажеру для водія.

Розглянуто будову окремих елементів системи. Вказано реалізовані механізми та функції тренажеру. Наведено схему під'єднання електронних компонентів до плати.

Реалізовано електронний тренажер в симуляторі «3D Інструктор»

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Basics of Microcontrollers – History, Structure and Applications// Вебсайт «Electronics Hub» URL: <https://www.electronicshub.org/microcontrollers-basics-structure-applications/> (дата звернення 05.05.2022р.).
2. Microcontroller – Invention History and Story Behind the Scenes // Вебсайт «Circuits Today»
URL: <https://www.circuitstoday.com/microcontroller-invention-history> (дата звернення 05.05.2022р.).
3. Intel 8048// Вебсайт «Вікіпедія»
URL: https://uk.wikipedia.org/wiki/Intel_8048 (дата звернення 05.05.2022 р.).
4. Intel 8051// Вебсайт «Вікіпедія»URL: https://ru.wikipedia.org/wiki/Intel_8051 (дата звернення 05.05.2022 р.).
5. Історія розвитку процесорів // Вебсайт «PCnews»
URL: https://pcnews.ru/blogs/istoria_razvitia_processorov_iz_70_h_v_90_e-684761.html#gsc.tab=0 (дата звернення 05.05.2022 р.).
6. What Things Use a Microcontroller// Вебсайт «Electronics Hub»
URL: <https://www.techwalla.com/articles/what-things-use-a-microcontroller> (дата звернення 06.05.2022 р.).
7. Atmel ‘AVR’ arhitecture// Вебсайт «TechniCodes»
URL: <https://technicodes.weebly.com/avrarch.html> (дата звернення 10.05.2022 р.).
8. AVR Microcontroller // Вебсайт «ELECTRONICMAFFIA» URL: <http://electronicmaffia.weebly.com/avr-microcontroller.html> (дата звернення 10.05.2022 р.).
9. Вебсайт «ReserchGate» URL: https://www.researchgate.net/figure/Architecture-of-ATmega16_fig3_285925465 (дата звернення 10.05.2022 р.).
10. ARDUINO – MICROCONTROLLERS// Вебсайт «Electronic Components»
URL: <https://www.tme.eu/en/news/library-articles/page/42541/arduino-microcontrollers-for-everyone/> (дата звернення 11.05.2022 р.).

11. Документація для Arduino Uno// Вебсайт «arduino.ua»

URL: <https://doc.arduino.ua/ru/hardware/Uno> (дата звернення 11.05.2022 р.).

12. Офіційний вебсайт Arduino// Arduino Uno

URL: https://store.arduino.cc/products/arduino-uno-rev3?_gl=1*1jorcju*_ga*NTEaNzc3MjU0LjE2NTAxOTkxOTg.*_ga_NEXN8H46L5*MTY1MzcxMjQ5OC41LjEuMTY1MzcxMjUwNy41MQ.. (дата звернення 12.05.2022 р.).

13. Офіційний вебсайт Arduino// Arduino Mega

URL: https://store.arduino.cc/products/arduino-mega-2560-rev3?_gl=1*ibc8ku*_ga*NTEaNzc3MjU0LjE2NTAxOTkxOTg.*_ga_NEXN8H46L5*MTY1MzcxMjQ5OC41LjEuMTY1MzcxMjcwMi41OQ.. (дата звернення 19.05.2022 р.).

14. Офіційний вебсайт Arduino// Arduino Nano

URL: https://store.arduino.cc/products/arduino-nano?_gl=1*1sez2cc*_ga*NTEaNzc3MjU0LjE2NTAxOTkxOTg.*_ga_NEXN8H46L5*MTY1MzcxMjQ5OC41LjEuMTY1MzcxMjcyNy4zNA.. (дата звернення 19.05.2022 р.).

15. Офіційний вебсайт Arduino// Arduino Micro

URL: https://store.arduino.cc/products/arduino-micro?_gl=1*1uxcdqp*_ga*NTEaNzc3MjU0LjE2NTAxOTkxOTg.*_ga_NEXN8H46L5*MTY1MzcxMjQ5OC41LjEuMTY1MzcxMjc0OC4xMw.. (дата звернення 19.05.2022 р.).

16. Офіційний вебсайт Arduino// Arduino Leonardo

URL: https://store.arduino.cc/products/arduino-leonardo-with-headers?_gl=1*1jwe3rg*_ga*NTEaNzc3MjU0LjE2NTAxOTkxOTg.*_ga_NEXN8H46L5*MTY1MzcxMjQ5OC41LjEuMTY1MzcxMjc3OS40Ng.. (дата звернення 19.05.2022 р.).

17. Розпинівка плати Arduino Leonardo // Вебсайт «Arduino.cc»

URL: <https://docs.arduino.cc/hardware/leonardo> (дата звернення 19.05.2022 р.)

18. Потенціометр 20 кОм // Сторінка інтернет-магазину «TRACKER-PI.COM.UA» URL: <https://tracker-pi.com.ua/p725733429-peremennyj-rezistor-potentsiometr.html> (дата звернення 01.06.2022 р.).

19. Тумблер на три положення// Інтернет сторінка магазину «Епіцентр» URL: <https://epicentrk.ua/ua/shop/mplc-tumbler-svetopribor-serednii-3-i-polozhennya-15a-tip-1122-latun-1ec340f3-ca4f-6142-8296-df25a61a43df.html> (дата звернення 02.06.2022 р.).

20. Конектор RJ45 // Сторінка інтернет-магазину «Розетка» URL: <https://rozetka.com.ua/ua/318864658/p318864658/> (дата звернення 02.06.2022 р.).

21. Ігровий маніпулятор Logitech G29 Driving Force // Сторінка інтернет-магазину «Розетка» URL: <https://rozetka.com.ua/ua/272910891/p272910891/>

ДОДАТОК А.
ЛІСТИНГ КОДУ

```
#include "HID-Project.h"
```

```
void setup() {  
  pinMode(A4, INPUT_PULLUP);  
  pinMode(A5, INPUT_PULLUP);  
  pinMode(2, INPUT_PULLUP);  
  pinMode(3, INPUT_PULLUP);  
  pinMode(4, INPUT_PULLUP);  
  pinMode(5, INPUT_PULLUP);  
  pinMode(6, INPUT_PULLUP);  
  pinMode(7, INPUT_PULLUP);  
  pinMode(8, INPUT_PULLUP);  
  pinMode(9, INPUT_PULLUP);  
  pinMode(10, INPUT_PULLUP);  
  pinMode(11, INPUT_PULLUP);  
  pinMode(12, INPUT_PULLUP);  
  pinMode(13, INPUT_PULLUP);  
  Gamepad.begin();  
  Serial.begin(9600);  
}
```

```
void loop() {  
  axes ();  
  base_button ();  
  transmission_buttons();  
}  
  
int a0 = A0;  
int a1 = A1;  
int a2 = A2;  
int a3 = A3;
```

```

int valA0;
int valA1;
int valA2;
int valA3;

int x0 = 20;//20
int y0 = 800;//900
void axes () {
    valA0 = analogRead(a0);
    if (valA0 < x0)valA0 = x0;
    if (valA0 > y0)valA0 = y0;
    valA0 = -1 * map (valA0, x0, y0, -32767, 32767);
    Gamepad.xAxis(valA0);

    valA1 = analogRead(a1);
    if (valA1 < 490)valA1 = 490;
    valA1 = map (valA1, 490, 1023, -128, 127);
    Gamepad.zAxis(valA1);

    valA2 = analogRead(a2);
    if (valA2 < 668)valA2 = 668;//698
    valA2 = map (valA2, 668, 1023, -32767, 32767);
    Gamepad.rxAxis(valA2);

    valA3 = analogRead(a3);
    if (valA3 < 520)valA3 = 520;//550
    valA3 = map (valA3, 520, 1023, -32767, 32767);
    Gamepad.ryAxis(valA3);

}

```

```
int x = 100;
```

```
int t1 = 0;
```

```
int t2 = 0;
```

```
bool A5_p = true;
```

```
bool b2_p = false;
```

```
bool b2_b = false;
```

```
bool b2_end = true;
```

```
bool b3_p = false;
```

```
bool b3_b = false;
```

```
bool b3_end = true;
```

```
bool b4_p = false;
```

```
bool b5_p = false;
```

```
bool b6_p = false;
```

```
bool b7_p = false;
```

```
void base_button () {
```

```
    if (analogRead (A5) < 100 && A5_p == false && b2_p == true ) {
```

```
        A5_p = true;
```

```
        b2_p = false;
```

```
        Serial.println(" A5 on");
```

```
    }
```

```
    if (digitalRead (2) == LOW && (A5_p == true || b3_p == true) && b2_p == false ) {
```

```
        t1 = 2;
```

```
        if (b3_p == true)t1 = 1;
```

```
        A5_p = false;
```

```
        b3_p = false;
```

```

    b2_p = true;
    b2_end = false;
    b2_b = true;
    Serial.println(" D2 on");
}
if (b2_b == true && t1 == 2 ) {
    t1 = 0;
    Gamepad.press(2);
    Gamepad.write();
    Serial.println(" Button 2 on");
    delay(x);
    Gamepad.release (2);
    Gamepad.write();
    Serial.println(" Button 2 off");
    delay(x);
    b2_b = false;
    if (t2 == 1) t2 = 2;
}
if (b3_b == true && t2 == 2) {
    t2 = 0;
    Gamepad.press(3);
    Gamepad.write();
    Serial.println(" Button 3 on");
    delay(x);
    Gamepad.release (3);
    Gamepad.write();
    Serial.println(" Button 3 off");
    delay(x);
    b3_b = false;
    if (t1 == 1) t1 = 2;
}

```

```

}
if (digitalRead (2) == HIGH && b2_p == false && b2_end == false ) {
    b2_end = true;
    b2_b = true;
    t1 = 2;
    Serial.println(" D2 off");
}
if (digitalRead (3) == LOW && b2_p == true && b3_p == false) {
    t2 = 1;
    b3_p = true;
    b2_p = false;
    b3_end = false;
    b3_b = true;
    Serial.println(" D3 on");
}
if (digitalRead (3) == HIGH && b3_p == false && b3_end == false ) {
    b3_end = true;
    b3_b = true;
    t2 = 2;
    Serial.println(" D3 off");
}
////////////////////////////////////
if (digitalRead(4) == LOW && b4_p == false) {
    b4_p = true;
    Gamepad.press(4);
    Gamepad.write();
    Serial.println(" Button 4 on");
    delay(x);
    Gamepad.release (4);
    Gamepad.write();
}

```

```
    Serial.println(" Button 4 off");
}
if (digitalRead(4) == HIGH && b4_p == true) {
    b4_p = false;
    Gamepad.press(4);
    Gamepad.write();
    Serial.println(" Button 4 on");
    delay(x);
    Gamepad.release (4);
    Gamepad.write();
    Serial.println(" Button 4 off");
}
if (digitalRead(5) == LOW && b5_p == false) {
    b5_p = true;
    Gamepad.press(5);
    Gamepad.write();
    Serial.println(" Button 5 on");
    delay(x);
    Gamepad.release (5);
    Gamepad.write();
    Serial.println(" Button 5 off");
}
if (digitalRead(5) == HIGH && b5_p == true) {
    b5_p = false;
    Gamepad.press(5);
    Gamepad.write();
    Serial.println(" Button 5 on");
    delay(x);
    Gamepad.release (5);
    Gamepad.write();
```

```

    Serial.println(" Button 5 off");
}
if (digitalRead(6) == LOW && b6_p == false) {
    b6_p = true;
    Gamepad.press(6);
    Gamepad.write();
    Serial.println(" Button 6 on");
    delay(x);
    Gamepad.release (6);
    Gamepad.write();
    Serial.println(" Button 6 off");
}
if (digitalRead(6) == HIGH && b6_p == true) {
    b6_p = false;
    Gamepad.press(6);
    Gamepad.write();
    Serial.println(" Button 6 on");
    delay(x);
    Gamepad.release (6);
    Gamepad.write();
    Serial.println(" Button 6 off");
}

if (digitalRead(7) == LOW && b7_p == false) {
    b7_p = true;
    Gamepad.press(7);
    Gamepad.write();
    Serial.println(" Button 7 on");
    delay(x);
    Gamepad.release (7);

```

```

    Gamepad.write();
    Serial.println(" Button 7 off");
}
if (digitalRead(7) == HIGH && b7_p == true) {
    b7_p = false;
    Gamepad.press(7);
    Gamepad.write();
    Serial.println(" Button 7 on");
    delay(x);
    Gamepad.release (7);
    Gamepad.write();
    Serial.println(" Button 7 off");
}
}

```

```

void transmission_buttons() {
    if (!digitalRead(8)) {
        Gamepad.press(8); delay(x);
    }
    else {
        Gamepad.release(8);
    }
    if (!digitalRead(9)) {
        Gamepad.press(9); delay(x);
    }
    else {
        Gamepad.release(9);
    }
    if (!digitalRead(10)) {
        Gamepad.press(10); delay(x);
    }
}

```

```
}  
else {  
    Gamepad.release(10);  
}  
if (!digitalRead(11)) {  
    Gamepad.press(11); delay(x);  
}  
else {  
    Gamepad.release(11);  
}  
if (!digitalRead(12)) {  
    Gamepad.press(12); delay(x);  
}  
else {  
    Gamepad.release(12);  
}  
if (!digitalRead(13)) {  
    Gamepad.press(13); delay(x);  
}  
else {  
    Gamepad.release(13);  
}  
if (analogRead(A4) < 50) {  
    Gamepad.press(1); delay(x);  
}  
else {  
    Gamepad.release(1);  
}  
Gamepad.write();  
}
```