

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ
АВТОМАТИЗОВАНИХ СИСТЕМ

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «АВТОМАТИЗАЦІЯ ПРОМИСЛОВОГО ОБЛАДНАННЯ НА БАЗІ
МІКРОКОНТРОЛЛЕРА ARDUINO»

на здобуття освітнього ступеня бакалавра
зі спеціальності 126 Інформаційні системи та технології
(код, найменування спеціальності)
освітньо-професійної програми Інформаційні системи та технології
(назва)

*Кваліфікаційна робота містить результати власних досліджень. Використання
ідей, результатів і текстів інших авторів мають посилання на відповідне
джерело*

(підпис)

Іван БРАТАШОВ

Ім'я, ПРІЗВИЩЕ здобувача

Виконав: здобувач вищої освіти гр. ІСД-41

Іван БРАТАШОВ

Ім'я, ПРІЗВИЩЕ

Керівник:

Ольга ЖИДКА

науковий
ступінь,
вчене звання

Ім'я, ПРІЗВИЩЕ

Рецензент:

науковий
ступінь,
вчене звання

Ім'я, ПРІЗВИЩЕ

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ
ТЕХНОЛОГІЙ**

Навчально-науковий інститут інформаційних технологій

Кафедра Інженерії програмного забезпечення автоматизованих

систем Ступінь вищої освіти Бакалавр

Спеціальність 126 Інформаційні системи та технології

Освітньо-професійна програма Інформаційні системи та технології

ЗАТВЕРДЖУЮ

Завідувач кафедру ІПЗАС

_____ Каміла
СТОРЧАК

« ____ » _____ 20 ____ р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ СТУДЕНТУ**

Браташову Івану Руслановичу

(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи: «Автоматизація промислового обладнання на базі мікроконтролера Arduino»

керівник кваліфікаційної роботи Ольга ЖИДКА

(ім'я, ПРІЗВИЩЕ, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від « ____ » _____ 20 ____ р. № ____.

2. Строк подання кваліфікаційної роботи: « ____ » _____ 20 ____ р.

3. Вихідні дані до кваліфікаційної роботи: Промислове обладнання;

Платформа Arduino;

Програмне забезпечення;

Науково-технічна література.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібні розробити)

1. Дослідження доцільності автоматизації промислового обладнання на базі Arduino.

2. Аналіз технічних засобів для розробки системи.

3. Результати виконаної роботи та висновки.

5. Перелік ілюстративного матеріалу: *презентаці*

6. Дата видачі завдання: « ___ » _____ 20__ р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз наявної науково-технічної літератури		
2	Вивчення матеріалів для подальшої взаємодії з ними		
3	Розробка проекту для кваліфікаційної роботи		
4	Написання кваліфікаційної роботи		
5	Оформлення роботи: вступ, висновки, реферат		
6	Розробка демонстраційних матеріалів		

Здобувач вищої освіти

_____ (підпис)

Іван БРАТАШОВ

_____ (Ім'я, ПРІЗВИЩЕ)

Керівник роботи
кваліфікаційної роботи

_____ (підпис)

Ольга ЖИДКА

_____ (Ім'я, ПРІЗВИЩЕ)

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня бакалавра: 51 стор., 33 рис., 15 джерел.

Об'єкт дослідження: Розробка автоматизаційної системи на основі мікроконтролера Arduino для управління промисловим обладнанням.

Предмет дослідження: Розробка та інтеграція автоматизованої системи з використанням мікроконтролера Arduino.

Мета роботи: Головною метою цього проєкту є аналіз та документування процесу розробки автоматизованої системи, що базується на мікроконтролері Arduino. Цей проєкт слугує ілюстративним прикладом того, як система забезпечує управління окремими компонентами промислового обладнання.

У процесі дослідження було звернено увагу на мікроконтролер Arduino, мову програмування C Arduino та різноманітні додаткові модулі, призначені для спільної роботи з Arduino. Розглядалися їхні можливості, архітектурні рішення та способи взаємодії між ними.

В результаті цього дослідження було зроблено висновки щодо доцільності використання технологій Arduino.

КЛЮЧОВІ СЛОВА: ARDUINO, C ARDUINO, ПРОМИСЛОВЕ ОБЛАДНАННЯ, АВТОМАТИЗОВАНА СИСТЕМА

ABSTRACT

The text part of the qualification work for the bachelor's degree: 51 pages, 33 figures, 15 sources.

Object of Research: Development of an automation system based on an Arduino microcontroller for controlling industrial equipment.

Subject of Research: Development and integration of an automated system using an Arduino microcontroller.

Research Objective: The main goal of this project is to analyze and document the process of developing an automated system based on the Arduino microcontroller. This project serves as an illustrative example of how the system provides control of individual components of industrial equipment.

In the research process, attention was paid to the Arduino microcontroller, the Arduino C programming language, and various add-on modules designed to work together with the Arduino. Their capabilities, architectural solutions and methods of interaction between them were considered.

Results: As a result of this study, conclusions were made regarding the feasibility of using Arduino technologies.

KEY WORDS: ARDUINO, C ARDUINO, INDUSTRIAL EQUIPMENT, AUTOMATED SYSTEM

ЗМІСТ

ВСТУП	9
1 ОГЛЯД КЛЮЧОВИХ ІНСТРУМЕНТІВ ПРОЄКТУ	12
1.1 Різновид моделей Arduino	12
1.2 Промислове обладнання, підлягаюче автоматизації	14
1.3 Принцип роботи пакетоформуючої машини.....	15
1.4 Датчики, модулі та інші елементи, які є ключовою ланкою в процесі автоматизації.....	23
2 РОЗРОБКА ПРОЄКТУ АВТОМАТИЗОВАНОЇ ПАКЕТОФОРМУЮЧОЇ МАШИНИ.....	26
2.1 Постановка задачі проєкту	26
2.2 Розробка проєкту.....	26
3 ІНШІ ПРИКЛАДИ ЗАСТОСУВАННЯ ARDUINO	47
3.1 Покращення якості життя завдяки технологіям IoT	47
3.2 Автоматизована система регулювання положення краю полотна	48
ВИСНОВКИ.....	53
ПЕРЕЛІК ПОСИЛАНЬ.....	54
ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (презентація).....	56

ВСТУП

Актуальність теми. Створення автоматизованої системи на базі мікроконтролера Arduino для використання на промисловому обладнанні дозволяє автоматизувати промислові верстати. Це дозволяє зробити верстати автоматичними, щоб роль людини у їх використанні знизилася до мінімуму. Як під час індустріальної революції, коли людина винайшла перший промисловий верстат, це призвело до поліпшення якості кінцевого продукту, та збільшення кількості готового продукту, що один оператор верстату може виробляти за свій робочий час. Звичайно, це призведе до зникнення багатьох робочих місць, але точність та ще вища якість продукції відкриє можливість створення ще більш складних пристроїв, які будуть потребувати обслуговування. Це створить нові робочі місця на заміну зникшим старим. Також процес автоматизації промислового обладнання дозволить уникнути величезній кількості виробничих травм, які може отримати оператор верстату, коли він не притримується техніки безпеки.

Також в нашому суспільстві, зникло поняття «пролетаріату», прошарку суспільства, який був прив'язаний до конкретного місця роботи, бо людина з цього прошарку не могла швидко переміщатися до свого місця роботи з іншого куточка міста або навіть області. Це призводило до того, що людина була змушена селитися біля свого місця роботи, не просто так з'явилося поняття «містоутворююче підприємство». Коли з прогресом технологій у людини з'явилася можливість переміщатися дешевше та швидше, вона змогла вибрати місце роботи та звільнитися зі свого старого місця роботи та йти туди, де більше платять, так з'явилося поняття «прокаріат» - прошарок людей, з настабільним місцем працевлаштування, які постійно шукають більш прибуткове місце роботи. Це створило проблему для власників малих підприємств, коли вони брали на роботу необучену людину та вчили її з нуля працювати за конкретним верстатом, методом спроб та помилок, зробивши купу браку, а матеріал має свою вартість. Після чого цей вже навчений робітник, бачить оголошення від крупної

компанії, що шукається робітник на такому самому або подібному верстаті, та заробітна плата вище, чим та, яку він отримує на старому місці роботи, бо крупна компанія може собі дозволити платити більш високу заробітню плату.

Таким чином малі фірми є базами для навчання некваліфікованих робітників для більш великих фірм, які в процесі переманюють вже навчених робітників, та мала фірма змушена брати нового робітника і знову його навчати, робітника, який в процесі навчання зробить купу браку, який коштує грошей.

Автоматизація має запобігти цьому процесу, це дозволить настільки спростити працю за верстатом, що навчити нового працівника можна буде лише за кілька годин. Та коли крупна фірма переманить чергового робітника, цей робітник не буде знати всіх нюансів, як працює станок, адже на автоматизованому верстаті де він працював до цього було лише пара кнопок, які треба було натискати в певному порядку, а верстат робив всю роботу сам.

Таким чином, кваліфікація оператора стає зовсім не ключовим фактором, який потрібен для виробництва якісної продукції. Також це спрощує навчання працівника, та запобігає виробництву бракованої продукції.

Об'єкт дослідження - процес розробки автоматизованої системи на базі мікроконтроллера Arduino для автоматизації промислового обладнання поліграфії.

Предмет дослідження - автоматизована система на базі мікроконтроллера Arduino.

Мета - розробка автоматизованої системи на базі мікроконтроллера Arduino для промислового обладнання поліграфії.

Основні завдання дослідження включають:

Вивчення технічних особливостей мікроконтроллера Arduino, з основним наголосом на його можливостях та способах взаємодії.

Створення системи, дозволяючої спростити роботу оператора верстату.

Проведення оцінки функціональності, безпеки та продуктивності системи.

Практична значущість. Результати дослідження можуть бути використані для автоматизації не лише представленого в цій роботі промислового обладнання,

але ще й у багатьох інших сферах промисловості.

Наукова новизна. Що стосується наукової новизни, то напрям автоматизації можна назвати слідуючим кроком в розвитку промисловості, як індустріальна революція прийшла на заміну ручної праці, потім електрифікація прийшла на заміну механіці, зараз на заміну живому оператору приходять роботизація. Так, багато людей втратять свої місця роботи, але на заміну старим місцям роботи, з'являться нові. Місця роботи, які будуть стосуватися вже не виробництва продукції, а обслуговування роботизованих машин, що будуть займатися виробництвом. Все, що можна спостерігати сьогодні стосовно змін у виробництві, могли спостерігати люди, які жили у ХІХ столітті, коли на території Європи вирувала індустріальна революція.

1 ОГЛЯД КЛЮЧОВИХ ІНСТРУМЕНТІВ ПРОЄКТУ

1.1 Різновид моделей Arduino

Arduino (Ардуіно) – апаратна обчислювальна платформа для аматорського конструювання, основними компонентами якої є плата мікроконтролера з елементами вводу/виводу та середовище розробки Processing/Wiring на мові програмування, що є спрощеною підмножиною C/C++. Arduino може використовуватися як для створення автономних інтерактивних об'єктів, так і підключатися до програмного забезпечення, яке виконується на комп'ютері (наприклад: Processing, Adobe Flash, Max/MSP, Pure Data, SuperCollider). Інформація про плату (малюнок друкованої плати, специфікації елементів, програмне забезпечення) знаходяться у відкритому доступі і можуть бути використані тими, хто воліє створювати плати власноруч.

Назва Arduino походить від бару в Івреа, Італія, де зустрічалися деякі із засновників проєкту. Бар був названий на честь Ардуїн I, який був маркграфом Маршу Івреї та королем Італії з 1002 по 1014 роки [1].

Arduino UNO – найпоширеніша модель Arduino. Її використовують частіше всього, як універсальну плату для створення проєктів. (Рис. 1.1)

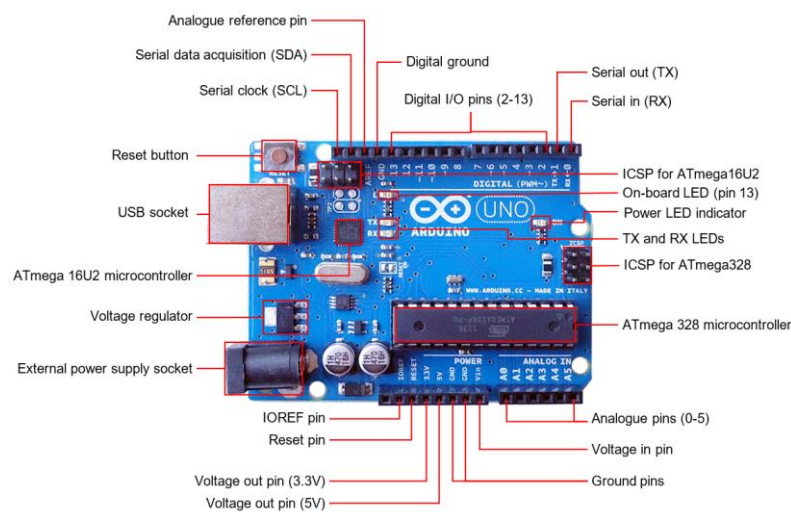


Рис. 1.1 Arduino UNO [2]

Arduino NANO – ще одна модель Arduino, яка не поступається за популярністю UNO, а то й деколи більш популярна за UNO. Вона дешевше коштує, має менший розмір, але пам'яті має менше ніж UNO. Хоча, цього достатньо, щоб робити невеликі проекти. (Рис. 1.2)

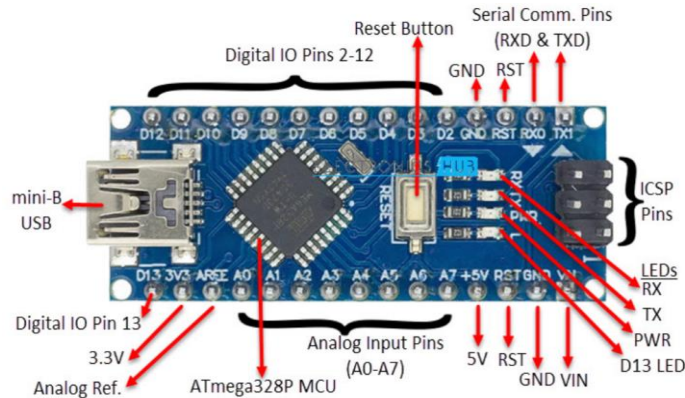


Рис. 1.2 Arduino NANO [3]

Arduino MEGA 2560 – це вже модель Arduino, яка спрямована саме на промисловість, оскільки має багато пам'яті та пінів. Вона може керувати багатьма аспектами промислового обладнання та її зазвичай ставлять на серйозні проекти. (Рис. 1.3)

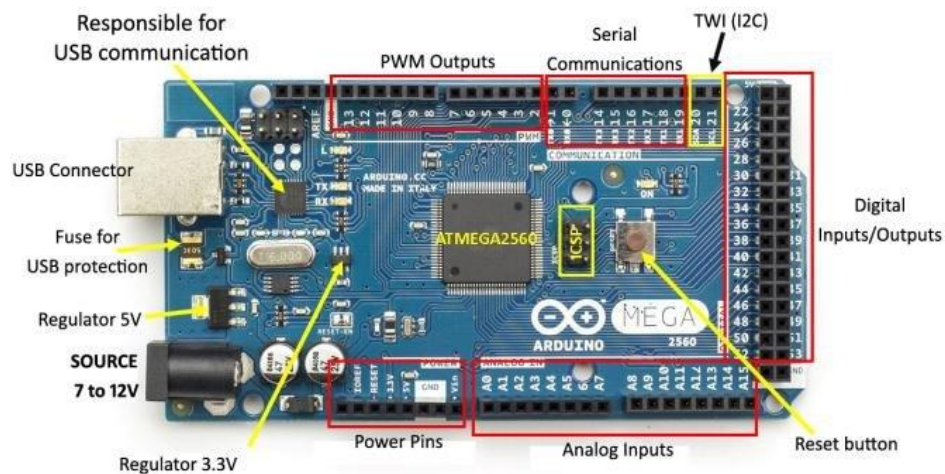


Рис. 1.3 Arduino MEGA 2560 [4]

Arduino Leonardo – модель Arduino, яка за зовнішнім виглядом дуже схожа на модель UNO. Але, вона відрізняється тим, що на відміну від UNO, яку не можна використовувати, як засіб вводу для комп'ютера, модель Leonardo можна

використовувати саме таким чином. На ній дуже добре робити саме інструменти вводу для комп'ютера. (Рис. 1.4)

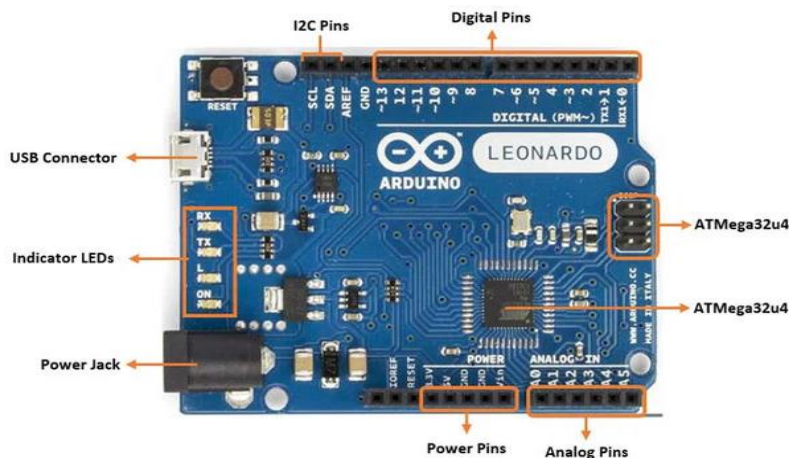


Рис. 1.4 Arduino Leonardo [5]

Arduino Pro Mini – ця модель Arduino більше схожа на модель NANO, але як і модель Leonardo, Arduino Pro Mini має можливість бути інструментом вводу для комп'ютера. Вона має менше пам'яті за Leonardo, але її можна також використовувати як такий економ-варіант. (Рис. 1.5)

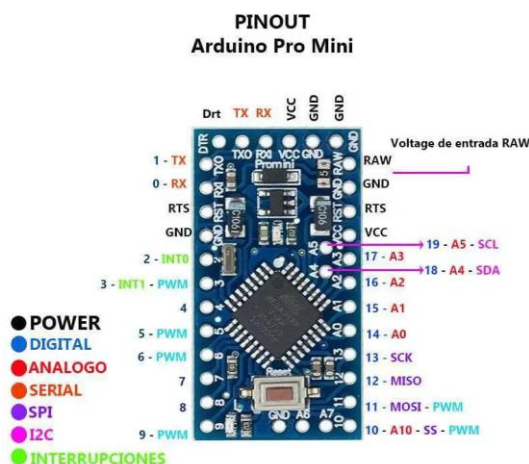


Рис. 1.5 Arduino Pro Mini [6]

1.2 Промислове обладнання підлягаюче автоматизації

Основною метою даної кваліфікаційної роботи є автоматизація промислового обладнання, тому було вибрано як об'єкт для автоматизації —

пакетоформуючу машину, яка використовується в поліграфії для формування та виробництва поліетиленової упаковки для будь-якої продукції. (Рис. 1.6)



Рис. 1.6 Пакетоформуюча машина

Оскільки фотографії машини використовувати доволі незручно, спеціально для цієї кваліфікаційної роботи була створена спрощена 3D-модель, що містить ключові вузли пакетоформуючої машини. 3D-модель була створена за допомогою програми Blender. (Рис. 1.7)

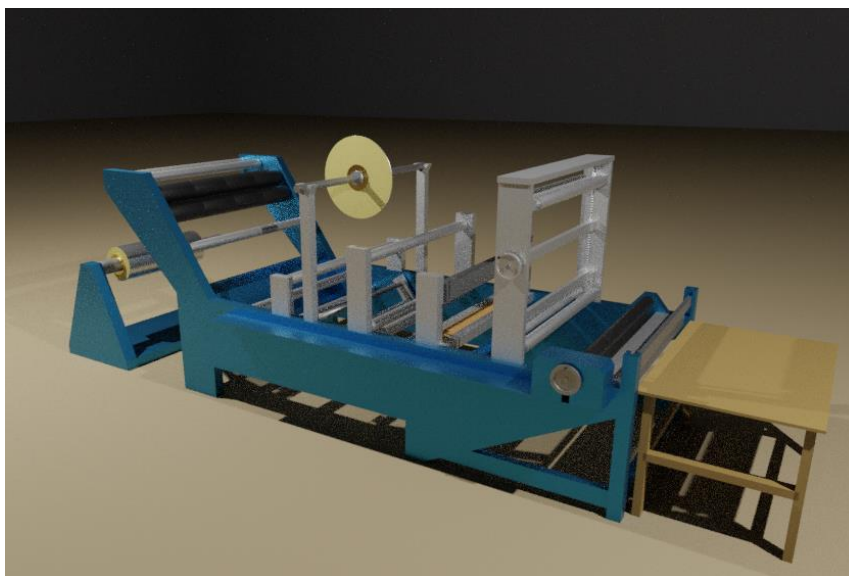


Рис. 1.7 Спрощена 3D-модель пакетоформуючої машини

1.3 Принцип роботи пакетоформуючої машини

Основною метою пакетоформуючої машини є виробництво поліетиленових пакетів, котрі в подальшому будуть використовуватись, як упаковка для продукції

інших підприємств. (Рис. 1.8)

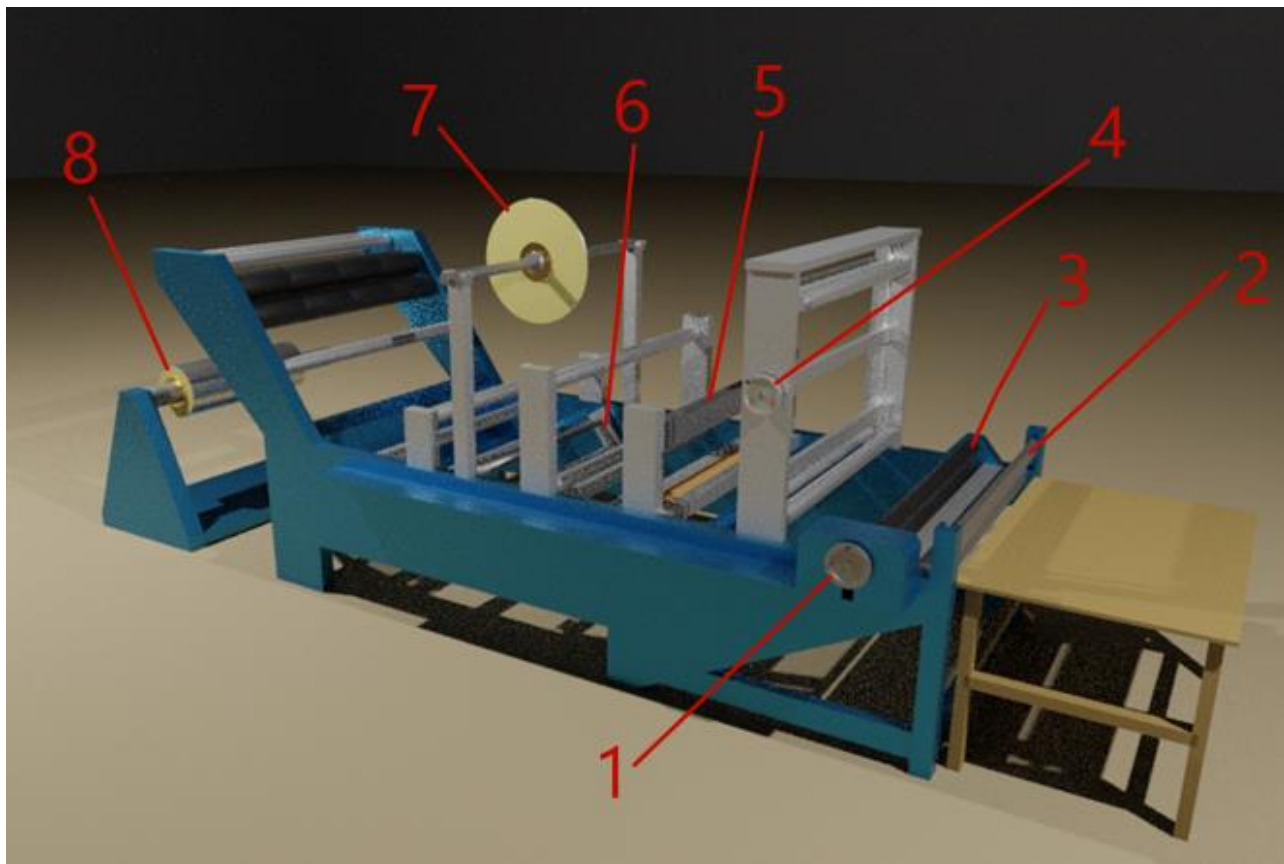


Рис. 1.8 Спрощена 3D-модель пакетоформуючої машини

- Протяжний вузол;
- гільйотина;
- протяжні вали;
- вузол налаштування лінії різку;
- вузол пайки;
- вузол подачі полотна;
- вузол клейкої стрічки;
- вузол розмотки.

Протяжний вузол – це вузол, який відповідає за протягування певної довжини полотна, який у подальшому підпадатиме під гільйотину і буде нарізатись на окремі пакети. (Рис. 1.9)

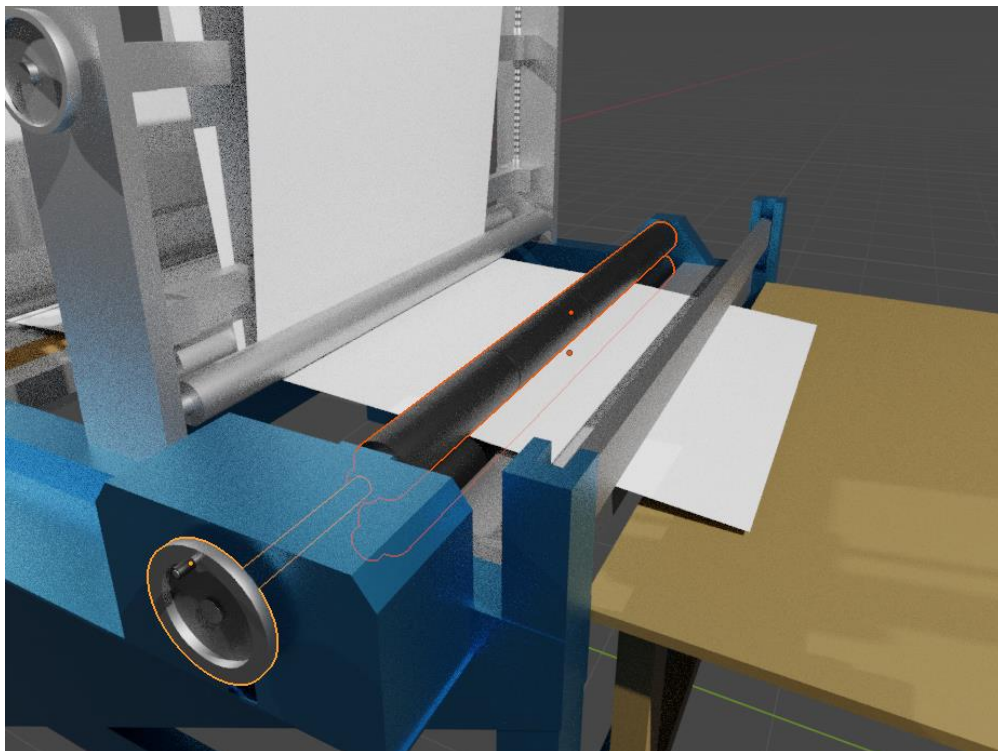


Рис. 1.9 Протяжний вузол (вигляд збоку)

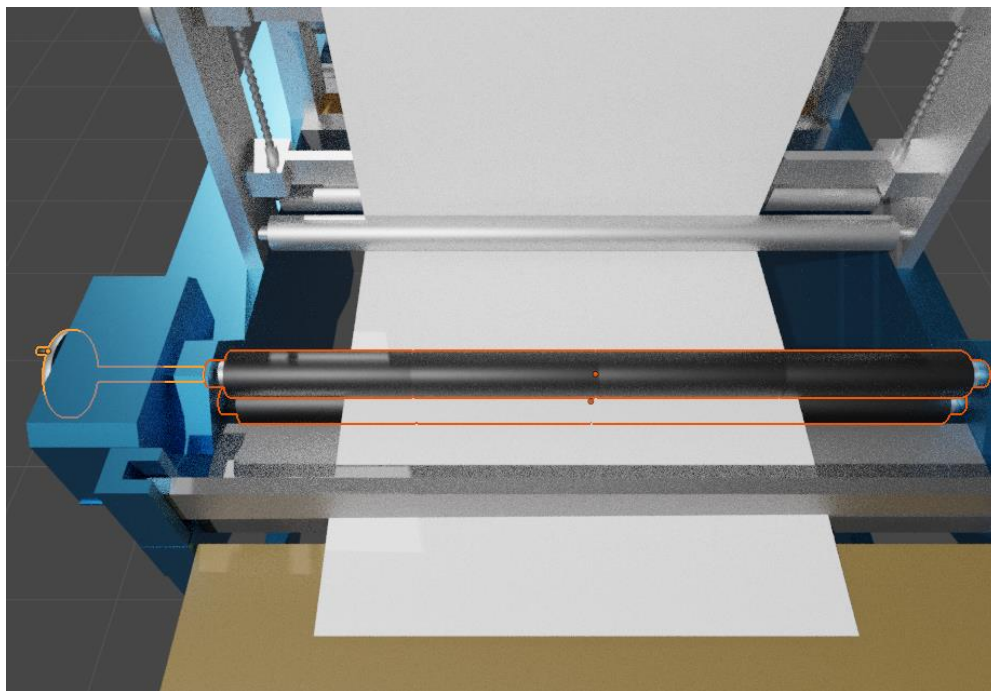


Рис. 1.10 Протяжний вузол (вигляд зпереду)

Гільйотина – це вузол пакетоформуючої машини, який відповідає за нарізку полотна на окремі пакети. Гільйотина зазвичай не є прямою, але як ножиці має вигин вниз з однієї сторони, бо інакше вона різати не буде. (Рис. 1.11)

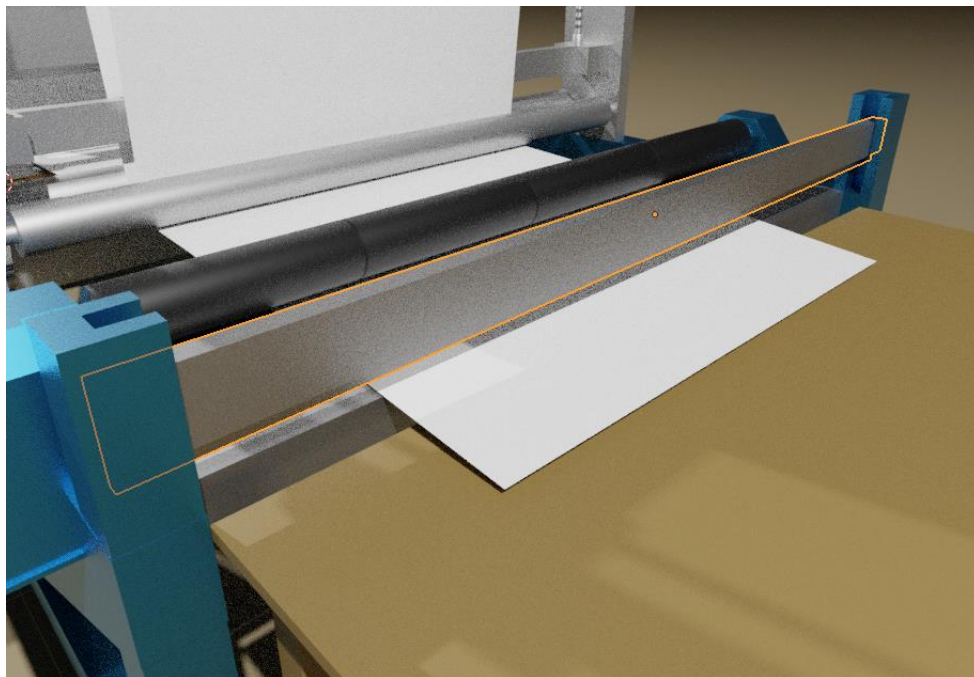


Рис. 1.11 Гільйотина

Протяжні вали – це частина протяжного вузла, саме вони здійснюють протяжку полотна. (Рис. 1.12)

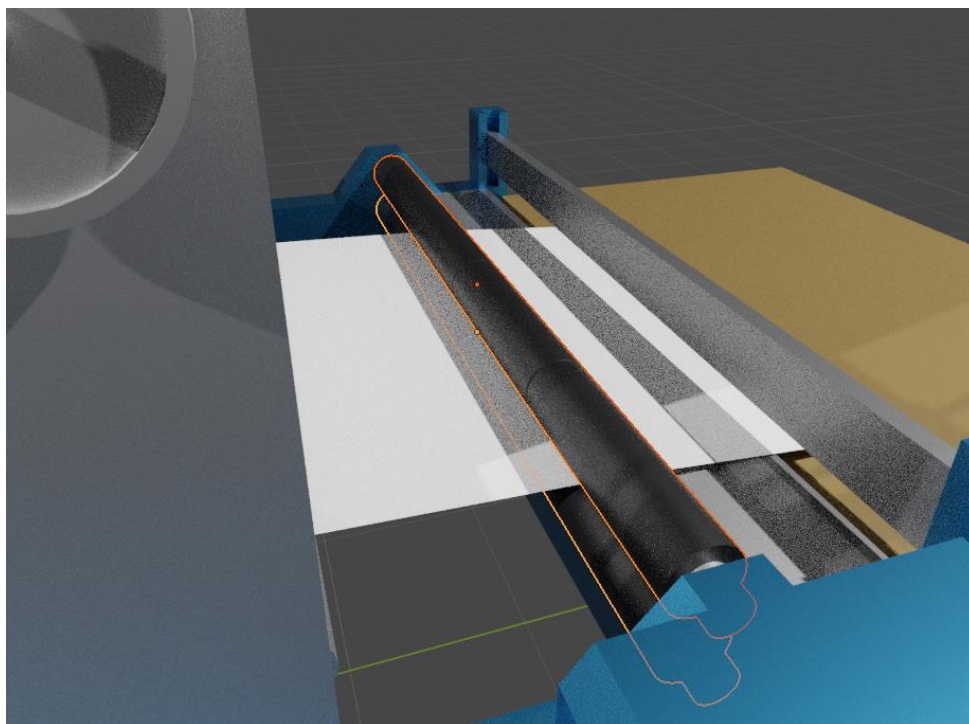


Рис. 1.12 Протяжні вали

Вузол налаштування лінії різку – це вузол, який дозволяє вручну регулювати протяжку полотна, а саме довжину полотна, що підпадає під гільйотину, це

дозволяє вручну налаштувати місце, де саме на полотні буде падати гільйотина, бо вона має попадати точно між двома швами, які робить паяльник, це називається – лінією різку. (Рис. 1.14)

Але іноді протягується невірна довжина полотна, це дає збої в роботі машини, та на лінії різку, де падає гільйотина, опиняється не стрічка з двома швами, які робить паяльник, а середина пакету, таким чином, полотно нарізається на пакети у неправильних місцях, це робить пакети не запаяними з усіх сторін, окрім верхньої частини, як повинно бути. Це є браком. Тому оператор має постійно дивитися за лінією різку, та у разі неполадок, крутити вентиля на вузлі налаштування лінії різку, щоб пакетоформувальна машина сама по собі не наробила купу браку. (Рис. 1.13)

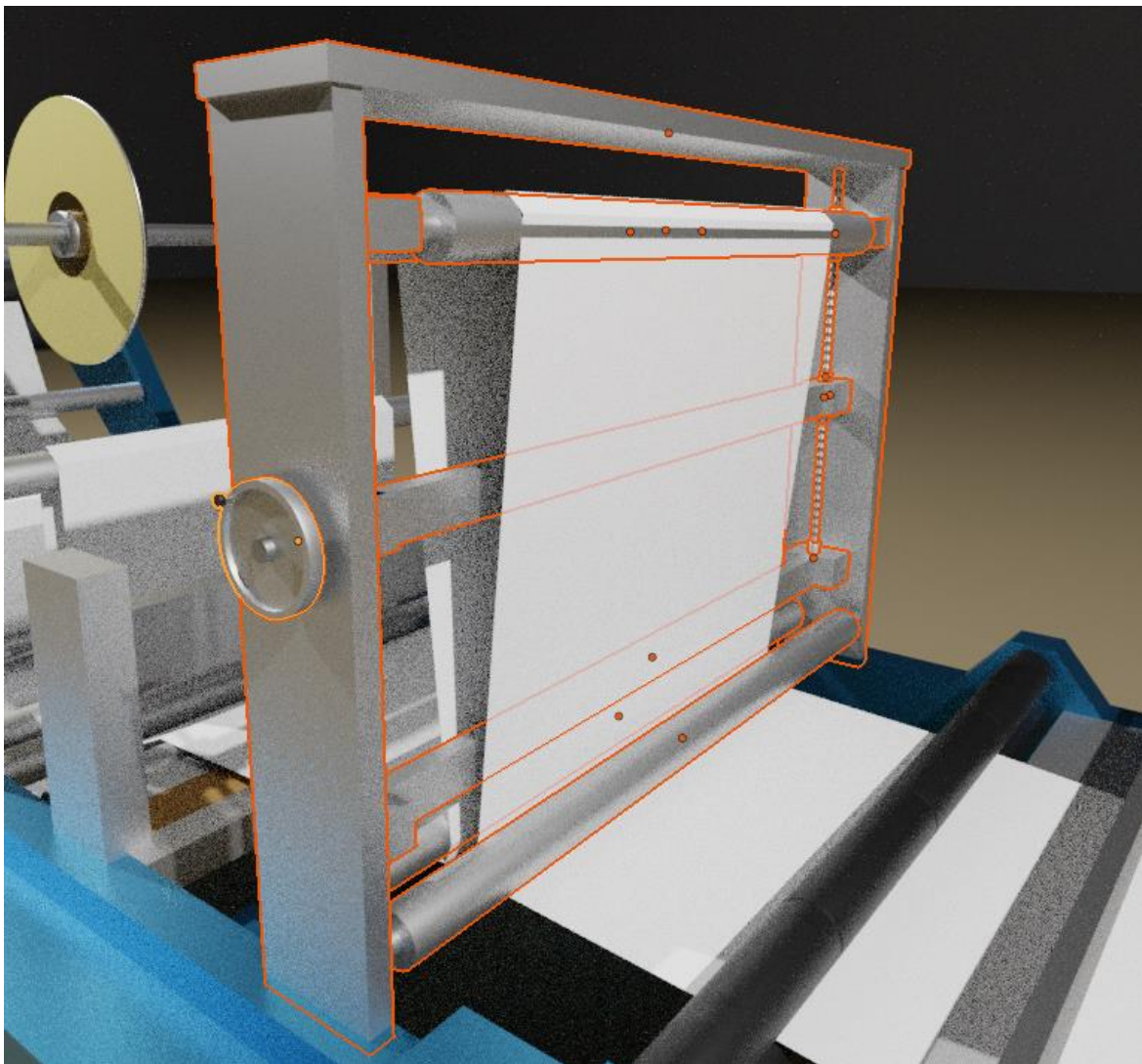


Рис. 1.13 Вузол налаштування лінії різку

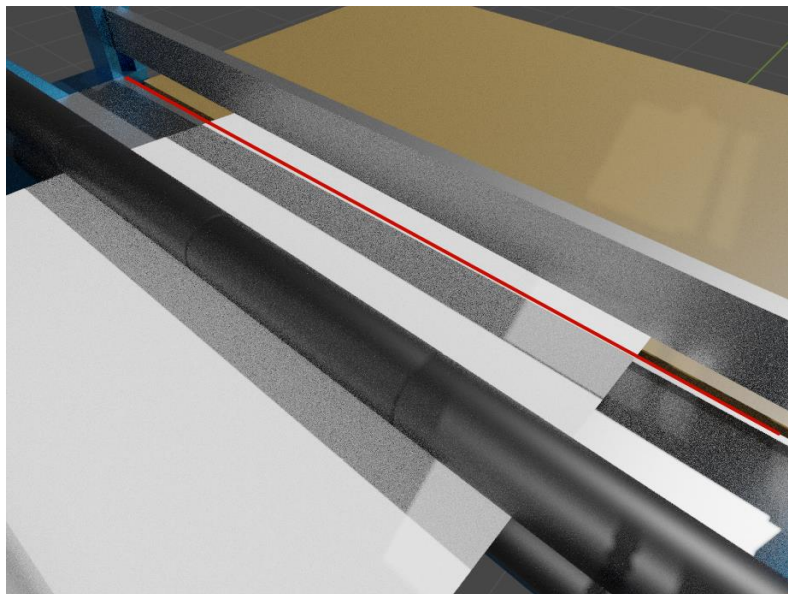


Рис. 1.14 Червоним позначена лінія різку

Вузол пайки – це вузол пакетоформувальної машини, який відповідає за спаювання між собою стінок звернутого завчасно полотна, який йде з вузла розмотки. Паяльник влаштований таким чином, що у нього є два елемента, які мають торкатися полотна, тим самим роблячи два поперечних шва на полотні. Саме ці шви мають знаходитися на лінії різку, і між цими двома швами має падати гільйотина, таким чином готові пакети мають по одному запаяному шву з лівої та правої сторони, а відстань між ними є шириною пакета. Зазвичай, температура паяльника коливається від 130°C до 150°C . (Рис. 1.15)

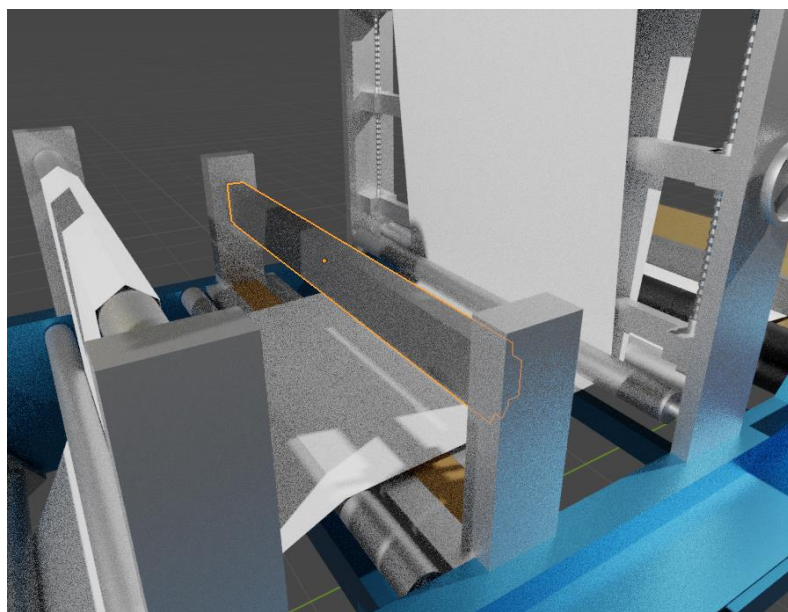


Рис. 1.15 Паяльник

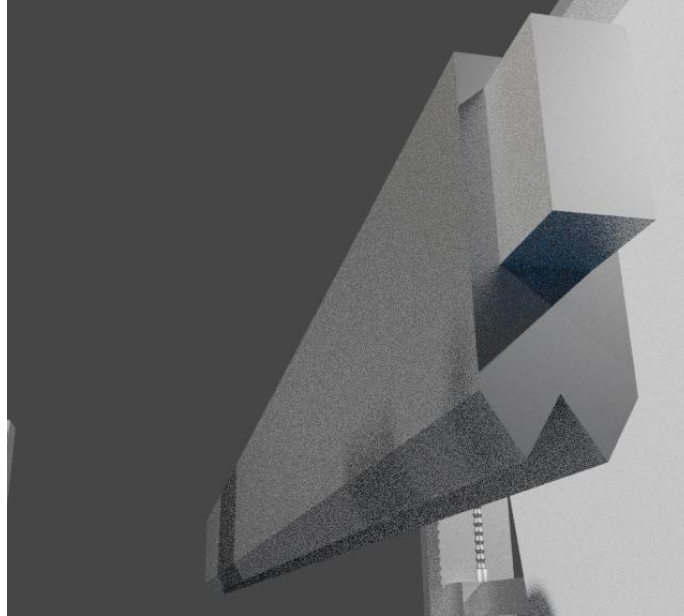


Рис. 1.16 Форма паяльника

Вузол подачі полотна – це вузол пакетоформувальної машини, що відповідає за відтягування певної довжини полотна, це дозволяє протяжним валам мати не дуже натягнуте полотно, щоб не збивати налаштування ширини пакету. Цей вузол рухається ввєрх та вниз, таким чином відтягує полотно вниз з вузла розмотки та при русі ввєрх віддає відтягнуту частину полотна далі. (Рис. 1.17)

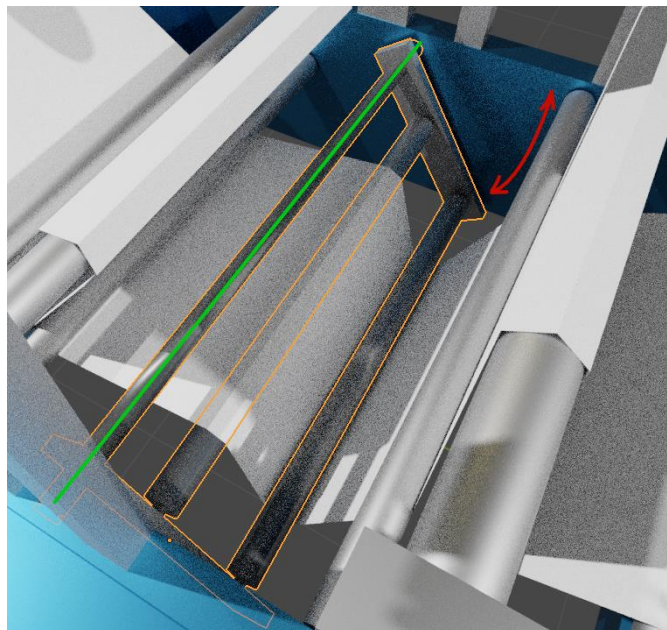


Рис. 1.17 Вузол подачі полотна (червоним позначено напрямок руху, зеленим позначено вісь, навколо якої обертається вузол подачі полотна)

Вузол клейкої стрічки – це вузол пакетоформувальної машини, що відповідає за нанесення на полотно клейкої стрічки. Це дозволяє виробляти вид пакетів, у яких з верхньої сторони є клейка стрічка, для зручності запаковування. Цей вузол здійснюється лише тоді, коли поступає замовлення на певну кількість пакетів саме з клейкою стрічкою. (Рис. 1.18)

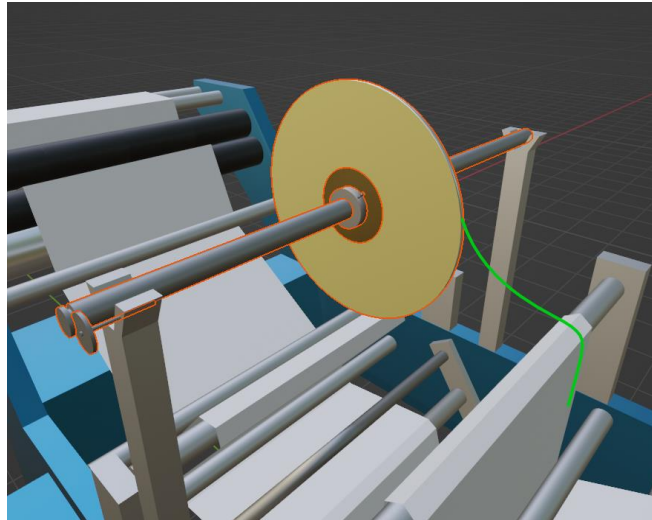


Рис. 1.18 Вузол клейкої стрічки (рулон по центру – сам рулон з клейкою стрічкою, зеленим позначено саму стрічку, яка наклеєна на полотно)

Вузол розмотки – це вузол пакетоформувальної машини, що відповідає за тримання рулона завчасно звернутого полотна, та його розмотування іншими вузлами пакетоформуючої машини. (Рис. 1.19)

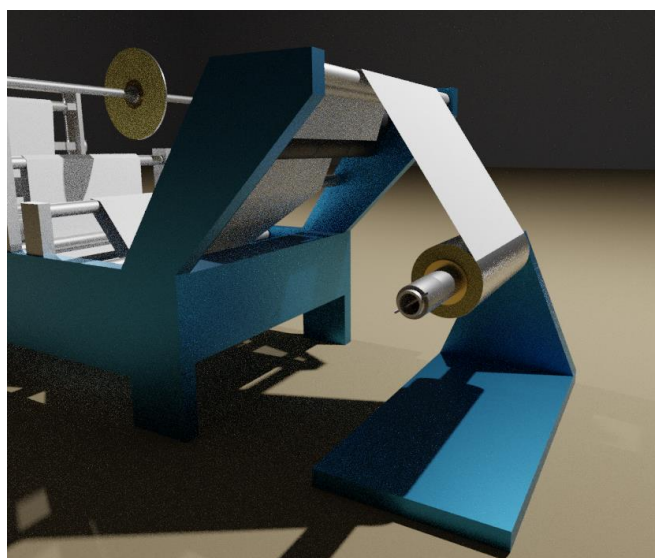


Рис. 1.19 Вузол розмотки (вигляд зправа)

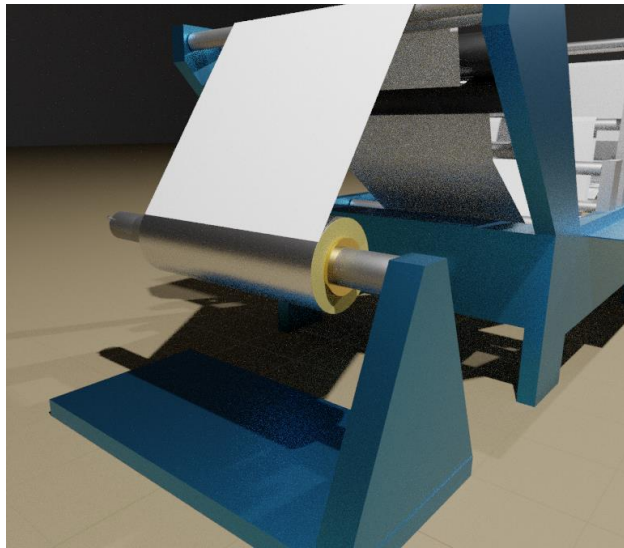


Рис. 1.20 Вузол розмотки (вигляд зліва)

1.4 Датчики, модулі та інші елементи, які є ключовою ланкою в процесі автоматизації

Для автоматизації пакетоформувальної машини, нам потрібні датчики, які будуть передавати сигнали на Arduino, та за допомогою яких мікроконтроллер Arduino буде розуміти, коли треба запускати свої алгоритми.

Інфрачервоний датчик – це датчик, який спрацьовує коли перед ним з’являється перешкода. (Рис. 1.21)



Рис. 1.21 Інфрачервоний датчик [7]

Крокові двигуни – це двигуни, які обертають свій ротор відповідно кількості заданих кроків, один крок – $1,8^\circ$. (Рис. 1.22)



Рис. 1.22 Кроковий двигун Nema23 [8]



Рис. 1.23 Кроковий двигун Nema43 [9]

Драйвер для крокових двигунів – дозволяє керувати кроковими двигунами. Без нього крокові двигуни працювати не будуть. (Рис. 1.24)



Рис. 1.24 Драйвер крокового двигуна TB6600 [10]



Рис. 1.25 Драйвер крокового двигуна DM542 [11]

2 РОЗРОБКА ПРОЄКТУ АВТОМАТИЗОВАНОЇ ПАКЕТОФОРМУЮЧОЇ МАШИНИ

2.1 Постановка задачі проекту

Головною задачею проекту є створення системи на базі Arduino, яка би могла спростити життя оператору пакетоформуючої машини, а також збільшити незалежність власників малих підприємств від навчання потенційних нових операторів машини, які у процесі навчання нароблять купу браку, що б'є по кишені власника підприємства. Таким чином ця система має спростити користування пакетоформуючою машиною, щоб вона не потребувала від потенційного оператора високої кваліфікації.

2.2 Розробка проєкту

Перелік конкретних задач:

- Аналіз слабких місць пакетоформуючої машини;
- розробка системи заміни ручного керування на автоматичне;
- розробка схеми проєкту;
- написання коду;
- інтеграція системи в пакетоформуючу машину.

Аналіз слабких місць пакетоформуючої машини. Об'єктивно, слабкими місцями пакетоформуючої машини є саме ті місця, де потрібна увага оператора, а саме слідкування за лінією різку, слідкування за клейкою стрічкою та крок пакету (ширина пакету в см.), бо варто оператору відволіктися, машина наробить купу браку.

Лінія різку любить їздити зі сторони в сторону, і гільйотина може не попадати по проміжку між двома паяними швами, тим самим відрізає края майбутнього пакету.

Клейка стрічка також любить їздити зі сторони в сторону та наклеюватись не на правильне місце, а де небудь. Хоча в цьому винен не вузол клейкої стрічки, а саме полотно, яке також їздить зі сторони в сторону.

Розробка системи заміни ручного керування на автоматичне. Для зміни ручного керування на автоматичне, нам і знадобиться Arduino. Для створення системи слідкування за клейкою стрічкою, нам знадобиться сім інфрачервоних датчиків, кроковий двигун Nema23 та його драйвер TB6600.

Для створення стабільної лінії різку, ми можемо спробувати замінити механічні частини пакетоформуальної машини, яка була спроектована та виготовлена на початку другої половини ХХ століття (тому в ній більша частина – механіка), на електричні, в нашому випадку – на Arduino. А саме, ми можемо замінити головний двигун на кроковий, який одночасно приводить у дію вузол протяжки полотна, гільйотину та паяльник (гільйотина та паяльник рухаються синхронно). Таким чином ми зможемо зробити панель, яка-би налаштовувала ширину пакету. Це дасть можливість уникнути довгої підготовки машини до роботи, коли доводиться прицілюватись з міліметрами, бо замовник замовив пакети з шириною, наприклад 9 см., а з попереднього замовлення була інша ширина, то доводиться дуже довго міняти налаштування ширини, бо система механічна та дуже приблизна. А коли буде електронна система, можна буде задати потрібну ширину, та одразу почати робити пакети з потрібною шириною.

Розробка схеми проєкту. Я розробив дві схеми, перша – схема автоматизації підлаштування клейкої стрічки, та друга – схема автоматизації головного двигуна пакетоформуальної машини. Почнемо з розгляду схеми автоматизації підлаштування клейкої стрічки.

Оскільки їздить зі сторони в сторону саме полотно, тоді і інфрачервоні датчики треба поставити для нагляду саме за полотном. Отже, ми маємо сім інфрачервоних датчиків, кроковий двигун Nema23 та драйвер до нього.

Ми можемо розташувати інфрачервоні датчики в рядок та поцепити з нижньої сторони відносно полотна, також нам необхідно, щоби датчики стояли

з краю полотна так, щоби полотно закривало собою лише половину датчиків. Це дасть можливість слідкувати за рухом полотна зі сторони в сторону за допомогою інфрачервоних датчиків. Датчики в свою чергу будуть подавати сигнал на Arduino, тоді як Arduino, в свою чергу буде подавати сигнал кроковому двигуну, який буде рухати рулон з клейкою стрічкою на певну відстань. Таким чином, ми зможемо автоматизувати слідкування за клейкою стрічкою, та оператору пакетоформуєчої машини залишиться лише міняти рулони клейкої стрічки, коли вони закінчатся. (Рис. 2.1)

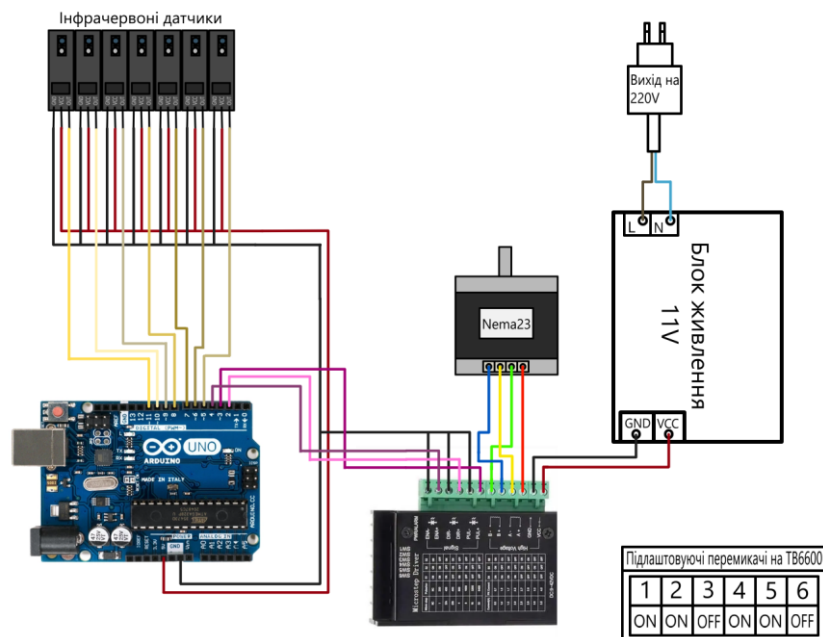


Рис. 2.1 Схема застосування описаних вище комплектуючих для реалізації проекту автоматизації системи підлаштування клейкої стрічки

Друга система – автоматизована система головного двигуна. Фактично, в данному випадку ми можемо замінити головний двигун змінного струму на кроковий двигун, проте, кроковий двигун Nema23, навряд-чи підійде для цієї системи, оскільки в нього може не вистачити потужності. Тому для цієї системи я обрав кроковий двигун Nema43. (Рис. 2.2)

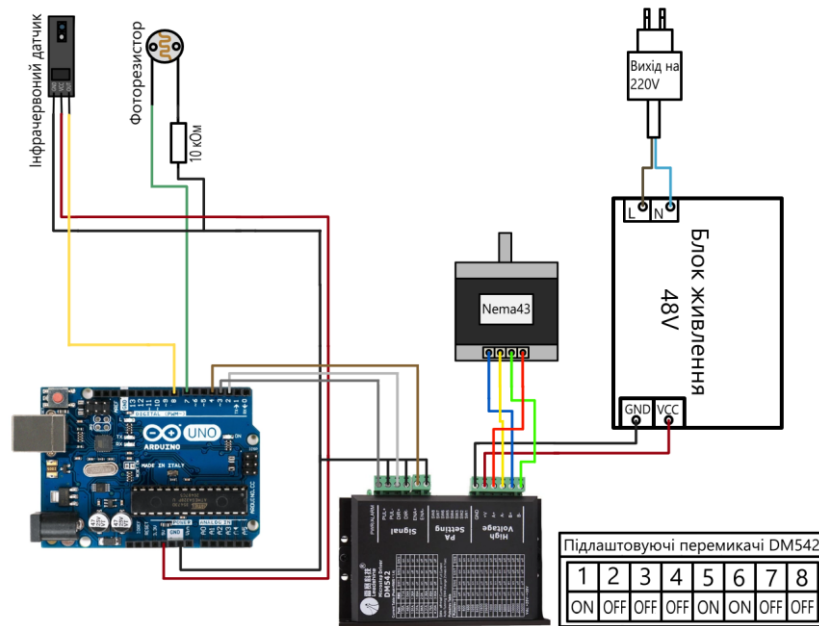


Рис. 2.2 Схема застосування описаних вище комплектуючих для реалізації проекту автоматизована система головного двигуна

Написання коду. Перша система, про яку йдеться у роботі – це система автоматизації підлаштування клейкої стрічки. В цьому коді мені довелося використовувати дві бібліотеки під назвою “GyverButton.h” та “GyverStepper.h”, щоб було зручніше працювати з кнопками та кроковим двигуном.

Код автоматизованої системи підлаштування клейкої стрічки наведений нижче.

```
#include <GyverStepper.h>
```

```
#include <GyverButton.h>
```

```
GStepper <STEPPER2WIRE> stepper(2048, 3, 2, 4); // 3-pul (фіалетовий), 2-  
dir (рожевий), 4-en (пурпурний)
```

```
GButton Кнопка1(5, LOW_PULL, NORM_OPEN);
```

```
GButton Кнопка2(6, LOW_PULL, NORM_OPEN);
```

```
GButton Кнопка3(7, LOW_PULL, NORM_OPEN);
```

```
GButton Кнопка4(8, LOW_PULL, NORM_OPEN);
```

```

GButton Кнопка5(9, LOW_PULL, NORM_OPEN);
GButton Кнопка6(10, LOW_PULL, NORM_OPEN);
GButton Кнопка7(11, LOW_PULL, NORM_OPEN);

void setup() {
  Serial.begin(9600);

  stepper.setAcceleration(1000); // <----- Встановлюємо кі-сть оборотів в
хвилину в дужках
  stepper.setMaxSpeed(1000); // <----- Встановлюємо швидкість (в кроках за
секунду). Сам рух запускається функцією run
  stepper.enable();

}
void loop() {

stepper.tick(); //Без цієї строки мотор буде клацати, но обертатись не буде
!!!

//-----Перший датчик-----
Кнопка1.tick();

if (Кнопка1.isPress()){
  Serial.println("В право");

  if (!stepper.tick()) {

    stepper.reset();
    stepper.setTarget(2050); //<----- Ставимо в дужках кі-сть кроків від старту
до зупинки двигуна

```

```
delay(100);
}
```

```
if (stepper.tick()) {
```

```
stepper.reset();
```

```
stepper.setTarget(2050); //<----- Ставимо в дужках кі-сть кроків від старту
до зупинки двигуна
```

```
delay(100);
}
```

```
}
```

```
if (Кнопка1.isRelease()){
```

```
Serial.println("В ліво");
```

```
if (!stepper.tick()) {
```

```
stepper.reset();
```

```
stepper.setTarget(-2050); //<----- Ставимо в дужках кі-сть кроків від
старту до зупинки двигуна
```

```
delay(100);
}
```

```
if (stepper.tick()) {
```

```
stepper.reset();
```

stepper.setTarget(-2050); //<----- Ставимо в дужках кі-сть кроків від
 старту до зупинки двигуна

```
delay(100);
```

```
}
```

```
}
```

```
//-----/Перший датчик-----
```

```
//-----Другий датчик-----
```

```
Кнопка2.tick();
```

```
if (Кнопка2.isPress()){
```

```
  Serial.println("В право");
```

```
if (!stepper.tick()) {
```

```
  stepper.reset();
```

stepper.setTarget(2050); //<----- Ставимо в дужках кі-сть кроків від старту
 до зупинки двигуна

```
delay(100);
```

```
}
```

```
if (stepper.tick()) {
```

```
  stepper.reset();
```

stepper.setTarget(2050); //<----- Ставимо в дужках кі-сть кроків від старту
 до зупинки двигуна


```

    delay(100);
  }

}

if (Кнопка2.isRelease()){
  Serial.println("В ліво");

  if (!stepper.tick()) {

    stepper.reset();
    stepper.setTarget(-2050); //<----- Ставимо в дужках кі-сть кроків від
    старту до зупинки двигуна

    delay(100);
  }

  if (stepper.tick()) {

    stepper.reset();
    stepper.setTarget(-2050); //<----- Ставимо в дужках кі-сть кроків від
    старту до зупинки двигуна

    delay(100);
  }

}

//-----/Другий датчик-----
//-----Третій датчик-----

```

```
Кнопка3.tick();
```

```
if (Кнопка3.isPress()){  
    Serial.println("В право");
```

```
if (!stepper.tick()) {
```

```
    stepper.reset();
```

```
    stepper.setTarget(2050); //<----- Ставимо в дужках кі-сть кроків від старту
```

до зупинки двигуна

```
    delay(100);
```

```
    }
```

```
if (stepper.tick()) {
```

```
    stepper.reset();
```

```
    stepper.setTarget(2050); //<----- Ставимо в дужках кі-сть кроків від старту
```

до зупинки двигуна

```
    delay(100);
```

```
    }
```

```
}
```

```
if (Кнопка3.isRelease()){
```

```
    Serial.println("В ліво");
```

```
if (!stepper.tick()) {
```

```

stepper.reset();
stepper.setTarget(-2050); //<----- Ставимо в дужках кі-сть кроків від
старту до зупинки двигуна

delay(100);
}

if (stepper.tick()) {

stepper.reset();
stepper.setTarget(-2050); //<----- Ставимо в дужках кі-сть кроків від
старту до зупинки двигуна

delay(100);
}

}

//-----/Третій датчик-----
//-----Четвертий датчик-----
Кнопка4.tick();

if (Кнопка4.isPress()){
Serial.println("В право");

if (!stepper.tick()) {

stepper.reset();
stepper.setTarget(2050); //<----- Ставимо в дужках кі-сть кроків від старту
до зупинки двигуна

```

```
delay(100);
}
```

```
if (stepper.tick()) {
```

```
  stepper.reset();
```

```
  stepper.setTarget(2050); //<----- Ставимо в дужках кі-сть кроків від старту
до зупинки двигуна
```

```
  delay(100);
}
```

```
}
```

```
if (Кнопка4.isRelease()){
```

```
  Serial.println("В ліво");
```

```
if (!stepper.tick()) {
```

```
  stepper.reset();
```

```
  stepper.setTarget(-2050); //<----- Ставимо в дужках кі-сть кроків від
старту до зупинки двигуна
```

```
  delay(100);
}
```

```
if (stepper.tick()) {
```

```
  stepper.reset();
```

stepper.setTarget(-2050); //<----- Ставимо в дужках кі-сть кроків від
 старту до зупинки двигуна

```
delay(100);
```

```
}
```

```
}
```

```
//-----/Четвертий датчик-----
```

```
//-----П'ятий датчик-----
```

```
Кнопка5.tick();
```

```
if (Кнопка5.isPress()){
```

```
  Serial.println("В право");
```

```
if (!stepper.tick()) {
```

```
  stepper.reset();
```

stepper.setTarget(2050); //<----- Ставимо в дужках кі-сть кроків від старту
 до зупинки двигуна

```
delay(100);
```

```
}
```

```
if (stepper.tick()) {
```

```
  stepper.reset();
```

stepper.setTarget(2050); //<----- Ставимо в дужках кі-сть кроків від старту
 до зупинки двигуна

```

    delay(100);
  }

}

if (Кнопка5.isRelease()){
  Serial.println("В ліво");

  if (!stepper.tick()) {

    stepper.reset();
    stepper.setTarget(-2050); //<----- Ставимо в дужках кі-сть кроків від
    старту до зупинки двигуна

    delay(100);
  }

  if (stepper.tick()) {

    stepper.reset();
    stepper.setTarget(-2050); //<----- Ставимо в дужках кі-сть кроків від
    старту до зупинки двигуна

    delay(100);
  }

}

//-----/П'ятий датчик-----
//-----Шостий датчик-----

```

```
Кнопка6.tick();

if (Кнопка6.isPress()){
  Serial.println("В право");

  if (!stepper.tick()) {

    stepper.reset();
    stepper.setTarget(2050); //<----- Ставимо в дужках кі-сть кроків від старту
до зупинки двигуна

    delay(100);
  }

  if (stepper.tick()) {

    stepper.reset();
    stepper.setTarget(2050); //<----- Ставимо в дужках кі-сть кроків від старту
до зупинки двигуна

    delay(100);
  }

}

if (Кнопка6.isRelease()){
  Serial.println("В ліво");

  if (!stepper.tick()) {
```

```

stepper.reset();
stepper.setTarget(-2050); //<----- Ставимо в дужках кі-сть кроків від
старту до зупинки двигуна

```

```

delay(100);
}

```

```

if (stepper.tick()) {

```

```

stepper.reset();
stepper.setTarget(-2050); //<----- Ставимо в дужках кі-сть кроків від
старту до зупинки двигуна

```

```

delay(100);
}

```

```

}

```

```

//-----/Шостий датчик-----

```

```

//-----Сьомий датчик-----

```

```

Кнопка7.tick();

```

```

if (Кнопка7.isPress()){
  Serial.println("В право");
}

```

```

if (!stepper.tick()) {

```

```

stepper.reset();
stepper.setTarget(2050); //<----- Ставимо в дужках кі-сть кроків від старту
до зупинки двигуна

```



```
delay(100);  
}
```

```
if (stepper.tick()) {
```

```
stepper.reset();
```

```
stepper.setTarget(2050); //<----- Ставимо в дужках кі-сть кроків від старту  
до зупинки двигуна
```

```
delay(100);  
}
```

```
}
```

```
if (Кнопка7.isRelease()){
```

```
Serial.println("В ліво");
```

```
if (!stepper.tick()) {
```

```
stepper.reset();
```

```
stepper.setTarget(-2050); //<----- Ставимо в дужках кі-сть кроків від  
старту до зупинки двигуна
```

```
delay(100);  
}
```

```
if (stepper.tick()) {
```

```
stepper.reset();
```

```
stepper.setTarget(-2050); //<----- Ставимо в дужках кі-сть кроків від
старту до зупинки двигуна
```

```
    delay(100);
  }

}

//-----/Сьомий датчик-----
}
```

Друга система, про яку йдеться у роботі – це система автоматизації головного двигуна. В цьому коді мені також довелося використовувати дві бібліотеки під назвою “GyverButton.h” та “GyverStepper.h”, щоб було зручніше працювати з кнопками та кроковим двигуном.

```
#include <GyverStepper.h>
#include <GyverButton.h>

int СКОРОСТЬ = 1000; // Початкове значення швидкості
int РассТоRНуЕ = 1000; // Початкова відстань (Ширина пакета від порізу
до порізу)
int DeJеу = 3000;

GButton IK_Datchik(8, LOW_PULL, NORM_OPEN);
GButton FotoDatchik(7, LOW_PULL, NORM_OPEN);

GStepper<STEPPER2WIRE> stepper(2048, 3, 2, 4); // 3-pul (сірий), 2-dir
(білий), 4-en (коричневий)
```

```

void setup() {
  Serial.begin(9600);

  stepper.setAcceleration(1000); // <----- Встановлюємо кі-сть оборотів в
хвилину в дужках
  stepper.setRunMode(KEEP_SPEED); // режим підтримання швидкості
  stepper.setSpeed(СКОРОСТЬ); // встановлення швидкості в кроках/сек
  stepper.enable();
}

void loop() {

  if (!stepper.tick()){
    stepper.setSpeed(СКОРОСТЬ);
    delay(100);
  }

  // Обробка інфрачервоного датчика
  IK_Datchik.tick();

  if (IK_Datchik.isPress()) {
    stepper.setSpeed(0);
    delay(100);
  }

  if (IK_Datchik.isRelease()) {
    stepper.setSpeed(СКОРОСТЬ);
    delay(100);
  }
}

```

```
// Обробка фотодатчика
FotoDatchik.tick();

if (FotoDatchik.isPress()) {
    stepper.setSpeed(0);
    delay(100);
}
}
```

Інтеграція системи в пакетоформуючу машину. Інтегрувати системи в пакетоформуючу машину можна наступним чином, що стосується вузлу клейкої стрічки, то інфрачервоні датчики можна поставити з нижньої сторони від полотна, таким чином, датчики будуть слідкувати за рухом полотна зі сторони в сторону, а кроковий двигун можна поставити на вентиль з різьбою, за допомогою якого оператор керує пересуванням рулону з клейкою стрічкою. Таким чином, коли інфрачервоні датчики спрацьовують – вони дають сигнал на мікроконтроллер Arduino, а Arduino в свою чергу дає сигнал кроковому двигуну. Таким чином рулон клейкої стрічки пересувається в потрібну сторону без участі оператора. (Рис. 2.3, 2.4)

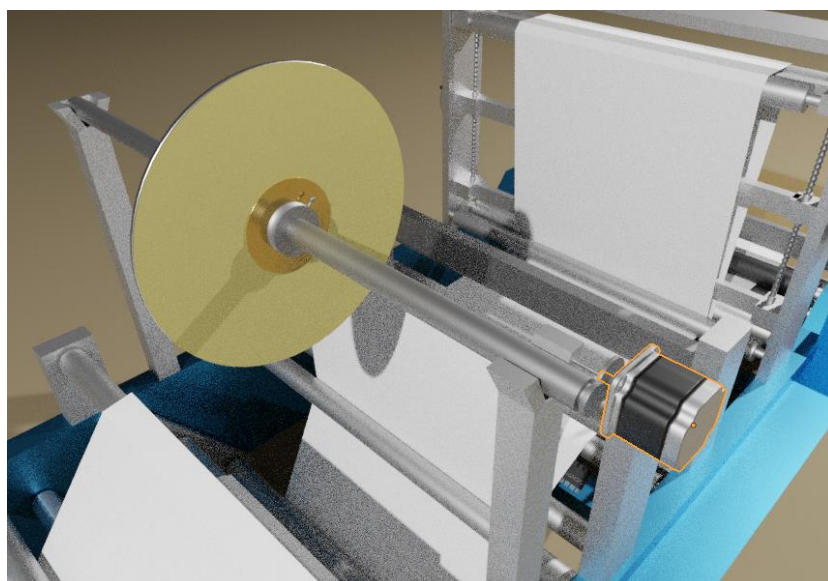


Рис. 2.3 Спосіб встановлення крокового двигуна

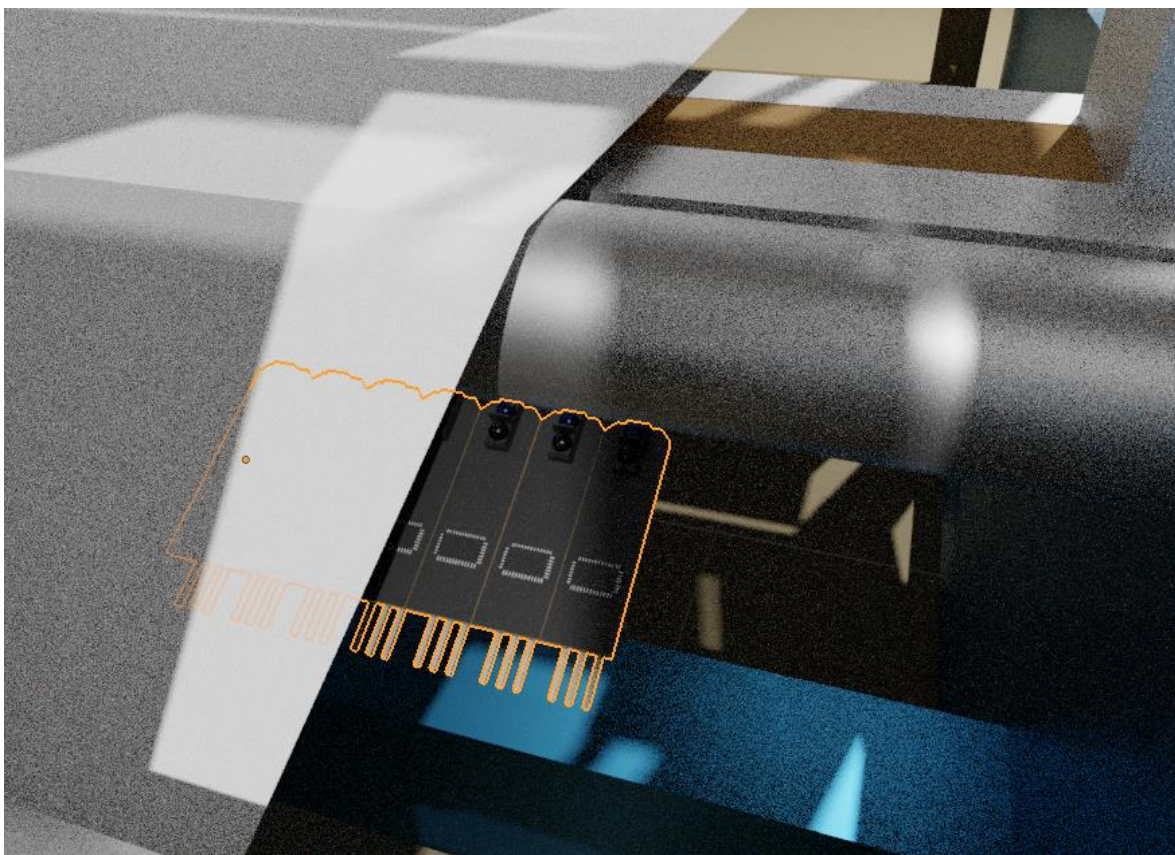


Рис. 2.4 Спосіб встановлення інфрачервоних датчиків

Стосовно автоматизованої системи головного двигуна, можна розмістити інфрачервоний датчик перед гільйотиною, та коли гільйотина опуститься – датчик активується та передасть сигнал на Arduino, а та в свою чергу зупинить кроковий двигун, що відповідатиме за протяжку полотна. Коли гільйотина підніметься, датчик деактивується та передасть сигнал на Arduino, щоб та відновила роботу крокового двигуна. Але ця система може працювати лише тоді, коли виробляються прозорі пакети. Але іноді пакети йдуть з друком, тоді можна використовувати фотодатчик, який буде відслідковувати фотомітку та зупинити головний двигун, після чого повинна опуститися гільйотина, активується інфрачервоний датчик, та коли гільйотина підніметься – інфрачервоний датчик деактивується та тим самим запусить головний двигун. Таким чином, оператор не буде змушений постійно наглядати та корегувати лінію різь. (Рис. 2.5, 2.6)

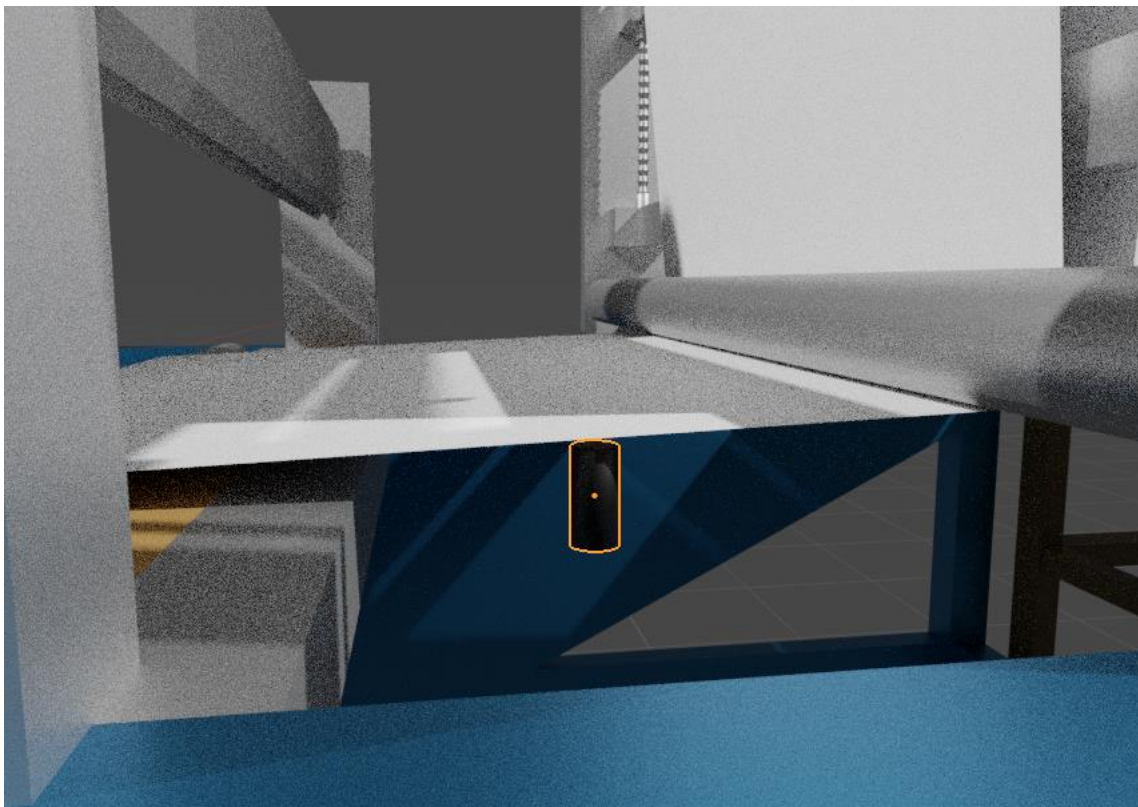


Рис. 2.5 Місце розташування фотодатчика

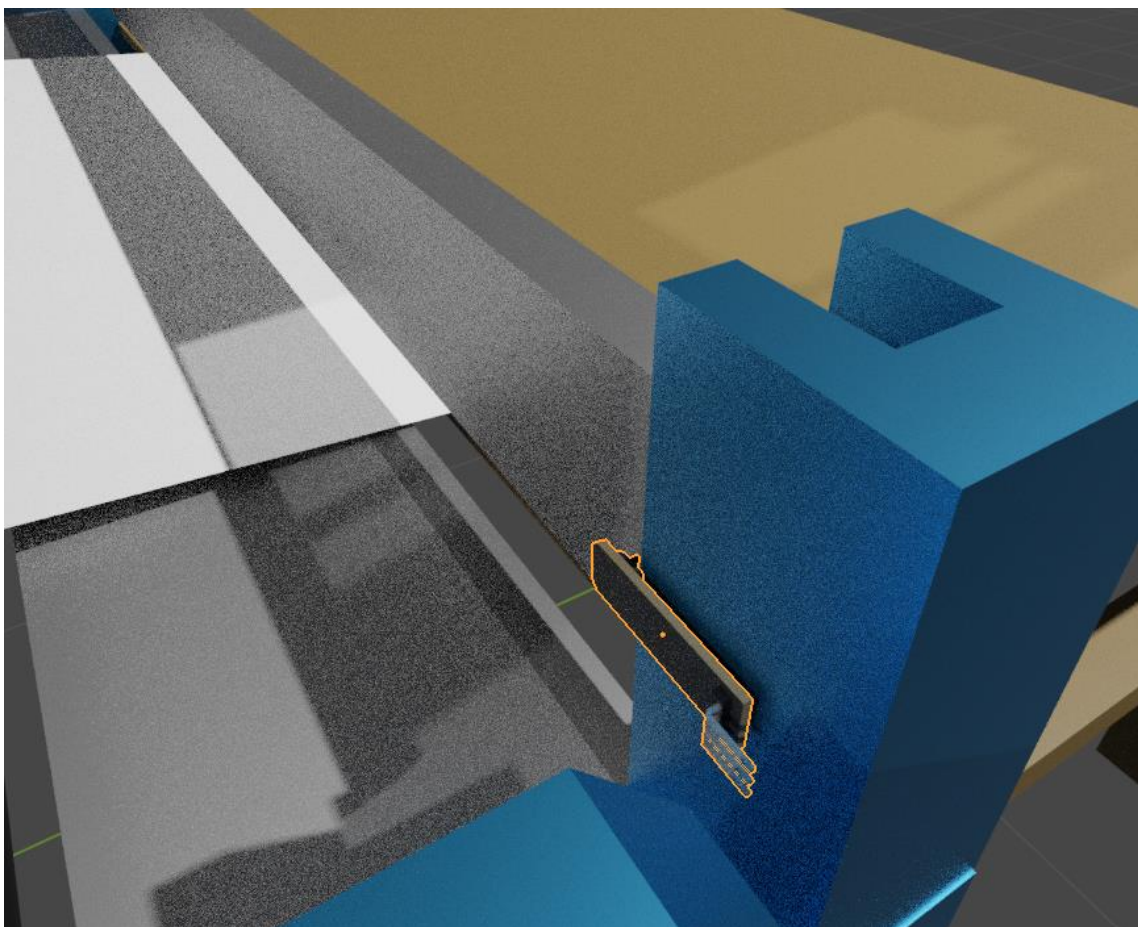


Рис. 2.6 Місце розташування інфрачервоного датчика

3 ІНШІ ПРИКЛАДИ ЗАСТОСУВАННЯ ARDUINO

3.1 Покращення якості життя, завдяки технологіям ІоТ

Протягом історії людства, різні технології займають і завжди займали дуже важливу роль у житті людей. У минулому столітті, дуже важливою складовою життя людини стала електроніка, яка дозволила спростити побутові справи, які раніше займали більш довгий час і велику витрату ресурсів. Наприклад, таке можна сказати про висвітлення житла людини. Якщо раніше доводилося закуповувати багато свічок, і весь час при потребі їх підпалювати та гасити, бо вони мали властивість швидко згоряти, то тепер можна обійтися електричною лампочкою, яка може перегоріти, але не тому, що минув термін експлуатації. До того ж електрична лампочка світитиме куди як яскравіше, ніж свічка. Нині гострішою стала тема автоматизації вже створених у минулому систем. Як би зробити так, щоб там, де потрібна була присутність людини, можна було б обійтися і без людини, щоб так би мовити - "пожинати плоди, при цьому посіявши зерно лише один раз". Тут на допомогу приходять технології інтернет-речей. Тепер людина, щоб увімкнути світло, не повинна йти до вимикача. Можна зробити так, щоб він міг двічі ляснути в долоні, і світло ввімкнеться. Те ж саме можна сказати і про інші прилади, від електричного чайника до відкриття та закриття дверей. Є чудовий американський фільм - "Назад у майбутнє", знятий 1985-го року. На самому початку фільму, була сцена, коли у відведений час, роботизований маніпулятор відкривав собачий корм і висипав його в собачу миску. Те, що в середині минулого століття було фантастикою – у наш час, на наших очах стає реальністю.

Мікроконтролер Arduino, і кілька датчиків підключених до нього, дають можливість керувати, наприклад включенням і вимкненням побутових приладів, чи то світло, чи електричний чайник. Наприклад, можна використовувати Wi-Fi модуль і керувати платою Arduino через смартфон, перебуваючи ще в дорозі додому. Це дозволить підігріти воду в чайнику, до моменту приходу додому. Це

можна реалізувати наступним чином: беремо мікроконтролер Arduino, до нього підключаємо Wi-Fi модуль, датчик температури і твердотільне реле, після чого - пишемо код і програмуємо наш мікроконтролер Arduino. За допомогою смартфона ми через Wi-Fi зможемо керувати діями Arduino, тут нам і стане в нагоді твердотільне реле, яке з'єднуватиме дрiт живлення для електричного чайника. У чайника, мабуть, краще переробити кнопку активації, щоб вона завжди була в увімкненому положенні. Твердотільне реле по команді користувача вмикатиме чайник, і це дозволить його включати дистанційно. У програмі можна вказати, що при досягненні ста градусів за Цельсієм, датчик температури спрацьовує та відправляє на мікроконтролер Arduino сигнал, що настав час вимкнути чайник, а мікроконтролер Arduino, у свою чергу відправляє сигнал на твердотільне реле, яке розмикає з'єднання між проводом живлення включеним у розетку та електричним чайником, що дозволяє його вимкнути.

Автоматизація – це наступний крок на шляху покращення якості життя людини, бо прилади вже є, залишилося зробити так, щоб вони робили свою роботу з мінімальним втручанням людини [14].

3.2 Автоматизована система регулювання положення краю полотна перемотально-різальної машини

Перемотально-різальна машина – ще одна машина, яка використовується в напрямку поліграфії для перемотки та нарізки рулонів поліетилену. Якщо оглянути мінуси цієї машини, то можна помітити, що ця машина є дуже застарілою на сьогоднішній день, в основному це стосується механічної частини машини та дуже застарілої електроніки, яка займає великий простір та дуже часто ламається. Саме систему протидії поломкам і буде розглянуто нижче. Якщо конкретніше, то буде розглянуто автоматизовану систему регулювання положення краю полотна перемотально-різальної машини. (Рис. 3.1)



Рис. 3.1 Перемотально-різальна машина [15]

Цьому верстату також можна провести автоматизацію ключових вузлів, але в нашому випадку ми розглянемо автоматизацію системи регулювання положення краю полотна перемотально-різальної машини.

Рулони поліетилену мають бути з рівними торцями, інакше це може вважатися браком, бо з рулоном, у якого торці не рівні буде дуже складно працювати в майбутньому.

На перемотально-різальній машині є вузол, який оператори машини називають «глазок». Фактично, це фотодатчик з влаштованим ліхтариком, який світить на край полотна таким чином, щоб край полотна був по центру світла ліхтаря, фотодатчик спрацьовує на відблиск світла. Оскільки вал, на який лягає полотно та на який світить ліхтарик має чорний колір, а полотно зазвичай або біле, або прозоре та глянцеве, то коли полотно з'їжджає, фотодатчик спрацьовує на контрасті полотна і валу, та дає команду на двигун, що рухає вліво чи вправо вал, на якому висить розмотуємий рулон. Також є ручне керування двигуном, що рухає в сторони вал розмотки.

Перелік конкретних задач:

- Аналіз слабких місць перемотально-різальної машини;
- розробка схеми проєкту;
- написання коду;
- інтеграція системи в перемотально-різальну машину.

Аналіз слабких місць перемотально-різальної машини. Як з'ясувалося, система дуже часто ламається та оператору машини доводиться вручну керувати підлаштуванням краю полотна вже тоді, коли він бачить, що полотно попливло на намотуємому ролоні. Ломаються зазвичай сенсори налаштування «глазку», без яких він не реагує на контраст полотна та вала.

Це можна виправити, зробивши систему на базі Arduino, також це допоможе в подальшому автоматизувати інші вузли машини, що зможуть взаємодіяти з системою автоматизованого «глазку».

Сам «глазок» – представляє з себе пристрій, який складається з двох фотодатчиків та ліхтарика. Фотодатчики направлені в одному напрямку, а ліхтарик має світити таким чином, щоб світло від ліхтарика відбивалося від перешкоди, куди направлен погляд «глазку», та попадало на фотодатчики, один з яких має працювати, в залежності від вибраного режиму. Перемикач режимів знаходиться на пристрої, та може бути не задіян в системі на Arduino.

Розробка схеми проєкту. Оскільки «глазок» має два режими, реагування на світлі кольори, та реагування на темні кольори, в схемі проєкту будуть використані два фотодатчики, модель Arduino UNO та реле. Логіка системи наступна: коли фотодатчик активується – Arduino перемикає реле, коли фотодатчик деактивується – Arduino перемикає реле в зворотнє положення. Реле в свою чергу буде керувати полярністю двигуна, що має рухати вал розмотки вліво чи вправо. Таке реле називається реле реверсу. (Рис. 3.2)

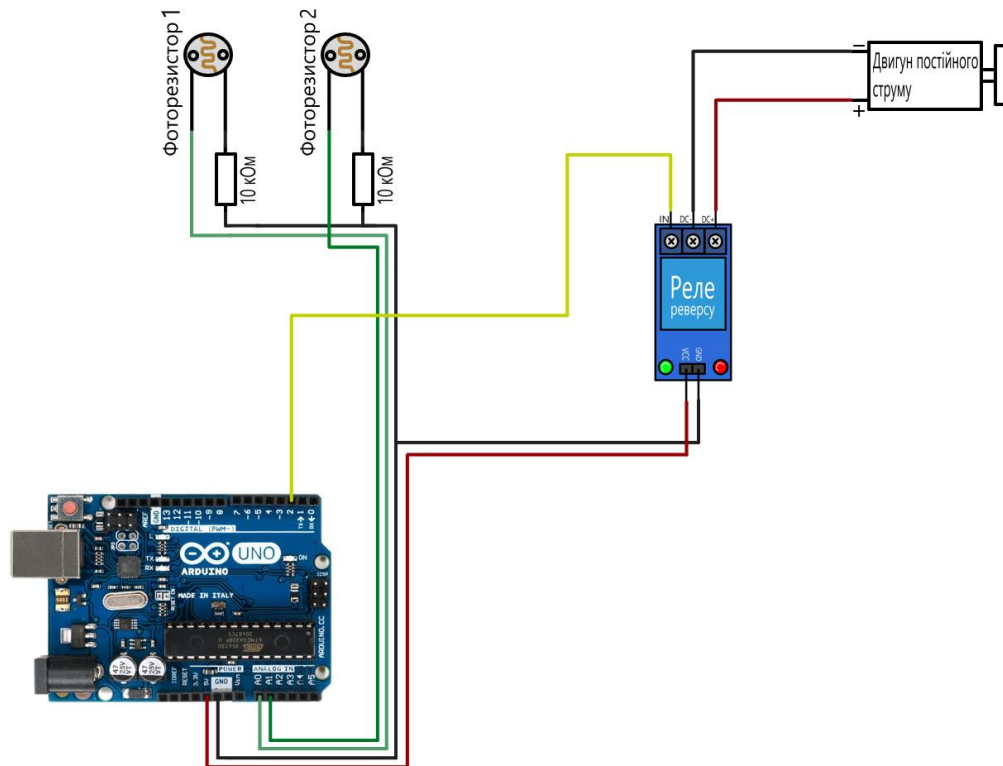


Рис. 3.2 Спрощена схема автоматизованої системи регулювання краю
ПОЛОТНА

Написання коду. Код автоматизованої системи регулювання краю полотна наведено нижче.

```
const int photoSensorPin1 = A0; // Пін першого фотодатчика
const int photoSensorPin2 = A1; // Пін другого фотодатчика
const int relayPin = 2; // Пін, к котрому підключено реле

void setup() {
    pinMode(photoSensorPin1, INPUT); // Встановлюємо пін першого
    фотодатчика як вхід
    pinMode(photoSensorPin2, INPUT); // Встановлюємо пін другого
    фотодатчика як вхід
    pinMode(relayPin, OUTPUT); // Встановлюємо пін реле як вихід
}
```

```

void loop() {
    int sensorValue1 = analogRead(photoSensorPin1); // Зчитуємо значення з
першого фотодатчика
    int sensorValue2 = analogRead(photoSensorPin2); // Зчитуємо значення з
другого фотодатчика

    // Якщо перший фотодатчик активований
    if (sensorValue1 > 500) {
        digitalWrite(relayPin, HIGH); // Вмикаємо реле
    }
    // Якщо другий фотодатчик активований
    else if (sensorValue2 > 500) {
        digitalWrite(relayPin, HIGH); // Вмикаємо реле
    }
    else {
        digitalWrite(relayPin, LOW); // Вимикаємо реле
    }

    delay(100); // Невелика затримка для стабілізації
}

```

Інтеграція системи в перемотально-різальну машину. Інтеграція автоматизованої системи регулювання краю полотна може бути здійснена наступним чином: двигун постійного струму розташований на торці розмотувального валу, таким чином він може викручувати та закручувати стрижень з різьбленням, котрий знаходиться вкрученим у розмотувальний вал з його торця. Таким чином двигун рухає розмотувальний вал зі сторони в сторону. Реле реверсу має перемикає полярність двигуна, таким чином змушуючи його обертатися в ту чи іншу сторони в залежності від сигналу фотодатчика.

ВИСНОВКИ

В цій кваліфікаційній роботі було наглядно розглянуто можливості використання мікроконтролера Arduino для автоматизації застарілого промислового обладнання на прикладі пакетоформуючої машини та перемотально-різальної машини, які використовуються у напрямку поліграфії. Таким чином можна автоматизувати обладнання та верстати в дуже широкому спектрі напрямків промисловості. Це дасть можливість дешево та доступно модернізувати застаріле обладнання під сьогоденні реалії, а не купувати нове обладнання, яке коштує величезних грошей. Також це дасть можливість зменшити поріг кваліфікованості операторів для роботи за цими верстатами та машинами. Таким чином, робота за машиною або верстатом стане значно простіше, продукція буде більш якісною, браку буде менше та потенційних операторів, які могли-би впоратися з керуванням цього обладнання буде все більше, бо досвід роботи за верстатом або машиною буде відігравати не таку важливу роль, яка є зараз.

ПЕРЕЛІК ПОСИЛАНЬ

1. Wikipedia. Arduino
URL: <https://uk.wikipedia.org/wiki/Arduino>
2. Internet of Things. Arduino Uno R3
URL: <https://bdavison.napier.ac.uk/iot/Tutorials/Arduino/setup/>
3. Nextpcb. The Ultimate Guide to Arduino Nano Pinout
URL: <https://www.nextpcb.com/blog/arduino-nano-pinout>
4. ResearchGate. Block Diagram Arduino MEGA 2560
URL: https://www.researchgate.net/figure/Block-Diagram-1-ArduinoMEGA2560-The-Arduino-Mega-2560-is-a-type-of-microcontroller_fig5_281538436
5. FabtoLab. Arduino Leonardo Board
URL: <https://www.fabtolab.com/arduino-leonardo-board>
6. MakersElectronics. Arduino Pro Mini 3.3V 8Mhz
URL: <https://makerselectronics.com/product/arduino-pro-mini-3-3v-8mhz>
7. MyProject. TCRT5000 інфрачервоний датчик стеження
URL: https://myproject.com.ua/tcrt5000-infrachervonij-datchik-stezhennja-ua.html?gclid=EAIaIQobChMIIsbjbhf6UhgMVO1qRBR2Ifwn2EAQYBiABEgIbWfD_BwE&utm_source=google&utm_medium=cpc&utm_campaign=New_Company
8. мікроАмпер. КРОКОВИЙ ДВИГУН NEMA23 57BYGH241-3004A-8 (57HS41)
URL: https://uamper.com/index.php?route=product/product&path=380&product_id=2265&gad_source=1&gclid=EAIaIQobChMIxrPD2oSVhgMVtF6RB R3bHADbEAQYAyABEgLL4fD_BwE
9. Prom. Кроковий двигун NEMA 43HS5842 5.0A 21N.m
URL: https://prom.ua/p365921995-shagovyj-dvigatel-nema.html?utm_source=google_product&utm_medium=cpc&utm_content=pla&utm_campaign=KT_cpc_1_5297199152&gad_source=1&gclid=EAIaIQobChMIo6r6y7qWhgMVUZqDBx3OTAsKEAQYBSABEgKj6PD_BwE

10. Prom. Драйвер крокового двигуна TB6600 4A
URL: https://prom.ua/p1563948721-drajver-shagovogo-dvigatelya.html?utm_source=google_product&utm_medium=cpc&utm_content=pla&utm_campaign=KT_cpc_1_5297199152&gad_source=1&gclid=EAIAIQobChMI5rGotoaVhgMVfxiiAx1BLAnWEAQYAyABEgKOHvD_BwE
11. Rozetka. Цифровий драйвер LeadShine крокового двигуна DM542 4A
URL: https://rozetka.com.ua/ua/400517259/p400517259/?utm_l=r&gad_source=1&gclid=EAIAIQobChMIqpyovbaWhgMVj5CDBx1M3wCKEAQYBiABEgLwNfD_BwE
12. Rozetka. Відладочна плата Arduino Rev3 для розробки проектів та прототипів з USB кабелем
URL: https://rozetka.com.ua/ua/282342243/p282342243/?gad_source=1&gclid=EAIAIQobChMI6qyNqYeWhgMVIqaDBx2ZBwTOEAQYBSABEgLhIPD_BwE
13. Інтернет речей (IoT) – революція в повсякденному житті за допомогою зв'язку.
URL: <https://mediacom.com.ua/internet-rechej-iot-revoljutsiya-v-povsyakdennomu-zhitti-za-dopomogoyu-zvyazku/>
14. Браташов І.Р. «Покращення якості життя, завдяки технологіям IoT». Тези доповіді V МІЖНАРОДНА НАУКОВО-ТЕХНІЧНА КОНФЕРЕНЦІЯ «Сучасний стан та перспективи розвитку IoT» – Київ, 18 квітня 2024 р.
15. Пекінг Трейд Групп. Виробництво гнучких пакувальних матеріалів
URL: <https://ecolegion.kiev.ua>
16. Giovanni Organtini. Physics Experiments with Arduino and Smartphones (2021)
17. Jesús López-Belmonte, José-Antonio Marín-Marín, Rebeca Soler Costa, Antonio José Moreno Guerrero. Arduino Advances in Web of Science. A Scientific Mapping of Literary Production
18. Massimo Banzi, Michael Shiloh. Getting Started with Arduino The Open Source Electronics Prototyping Platform 3rd Edition (2015)
19. Jeremy Blum. Exploring Arduino: Tool and Techniques for Engineering Wizardry 2nd Edition (2019)
20. John Boxall. Arduino Workshop, 2nd Edition: A Hands-on Introduction with 65 Projects (2021)

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (презентація)

Державний університет інформаційно-комунікаційних технологій

Кафедра Інженерії програмного забезпечення автоматизованих систем

КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Автоматизація промислового обладнання на базі мікроконтролера Arduino»

на здобуття освітнього ступеня бакалавра
зі спеціальності 126 Інформаційні системи та технології
освітньо-професійної програми Інформаційні системи та технології

Виконав: Браташов І.Р, ІСД-41

Науковий керівник роботи:

Жидка О.В.

Київ - 2024

- ▶ **Актуальність теми:** створення автоматизованої системи на базі мікроконтролера Arduino для використання на промисловому обладнанні дозволяє автоматизувати промислові верстати. Це дозволяє зробити верстати автоматичними, щоб роль людини у їх використанні знизилася до мінімуму.
- ▶ **Наукова новизна:** напрям автоматизації можна назвати наступним кроком в розвитку промисловості, як індустріальна революція прийшла на заміну ручної праці, потім електрифікація прийшла на заміну механіці, зараз на заміну живому оператору приходять роботизація.
- ▶ **Об'єкт дослідження:** процес розробки автоматизованої системи на базі мікроконтролера Arduino для автоматизації промислового обладнання, а саме пакетоформуючої машини та перемотально-різальної машини, які використовуються у напрямку поліграфії.
- ▶ **Предмет дослідження:** процес розробки автоматизованої системи на базі мікроконтролера Arduino, включаючи архітектурні рішення, структуру коду та процес роботи за готовим автоматизованим станком.
- ▶ **Мета дослідження:** Метою дослідження є детальне вивчення та аналіз процесу розробки автоматизованої системи на базі Arduino.
- ▶ **Завдання дослідження:**
 - ▶ 1. Розробка автоматизованої системи підлаштування клейкої стрічки пакетоформуючої машини
 - ▶ 2. Розробка автоматизованої системи головного двигуна пакетоформуючої машини
 - ▶ 3. Розробка автоматизованої системи регулювання краю полотна перемотально-різальної машини

Пакетоформуєча машына



Рисунок 1.1 Зовнішній вигляд пакетоформуєчої машыны

▶ 3

3D-модель пакетоформуєчої машыны

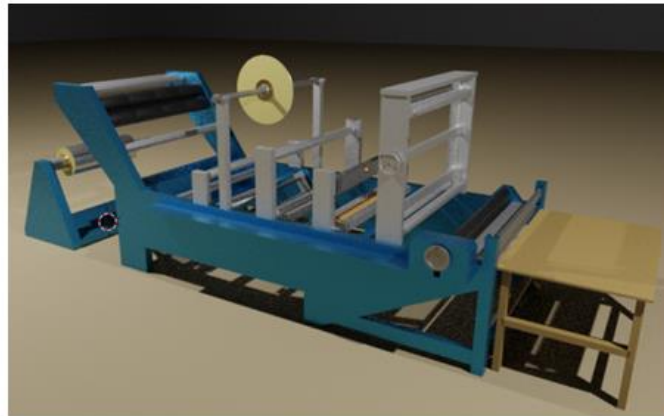


Рисунок 1.2 3D-модель пакетоформуєчої машыны, разроблена спецыяльна для
цього праекту

▶ 4

Схема автоматизованої системи підлаштування клейкої стрічки пакетоформуючої машини

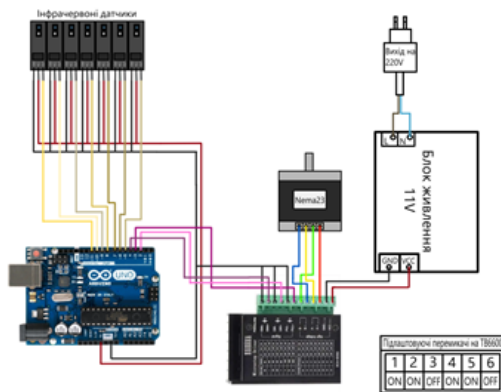


Рисунок 1.3 Схема автоматизованої системи підлаштування клейкої стрічки пакетоформуючої машини

Логіка системи: край полотна розташовується посередині блоку з інфрачервоних датчиків, таким чином, щоб половина датчиків реагувала на полотно, а половина ні. Та коли полотно буде з'їжджати в одну з сторін – один з датчиків буде спрацьовувати та подавати сигнал на Arduino, для того, щоб Arduino дала сигнал на кроковий двигун, який за допомогою різьбленого прута буде рухати вал, який тримає рулон з клейкою стрічкою, щоб вона переміщувалась на деякий інтервал вслід за полотном.

▶ 5

Вузол клейкої стрічки

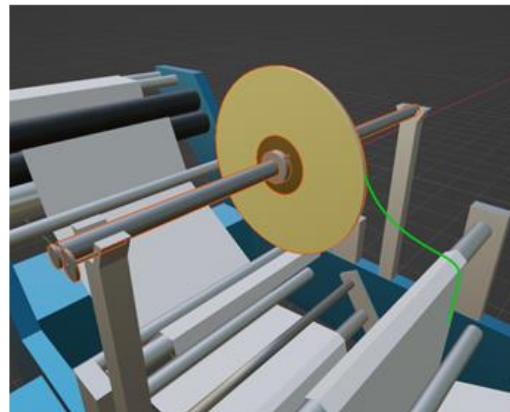


Рисунок 1.4 Зовнішній вигляд вузла клейкої стрічки (зеленим позначена клейка стрічка, яка переходить з рулону на полотно)

▶ 6

Місце розташування інфрачервоних датчиків та крокового двигуна

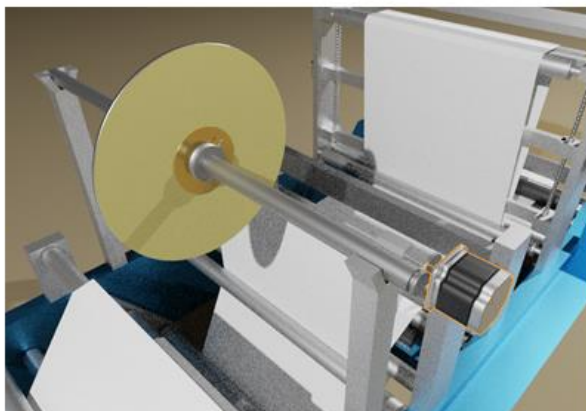


Рисунок 1.5 Розташування крокового двигуна

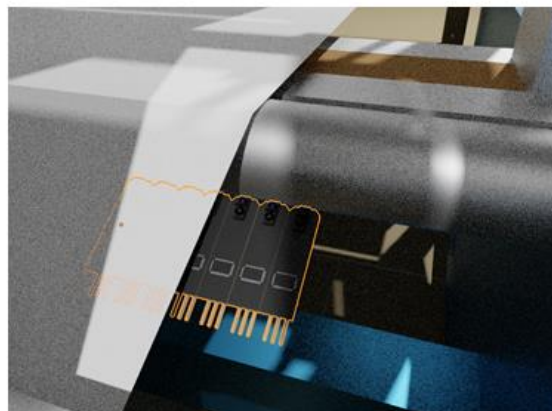


Рисунок 1.6 Розташування інфрачервоних датчиків

▶ 7

Гільйотина та паяльник

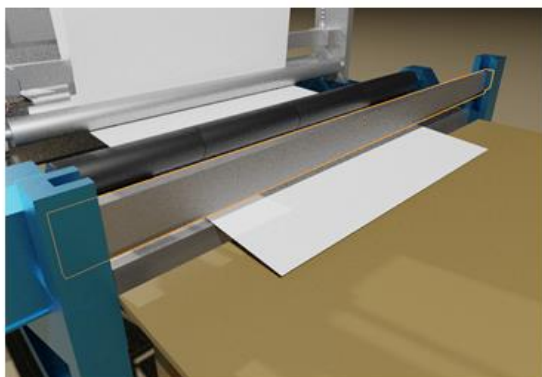


Рисунок 2.1 Гільйотина

Гільйотина відповідає за нарізання полотна на окремі пакети, суть в тому, що вона має попадати між двома швами на полотні, які робить паяльник. Ці шви – є краями пакетів.

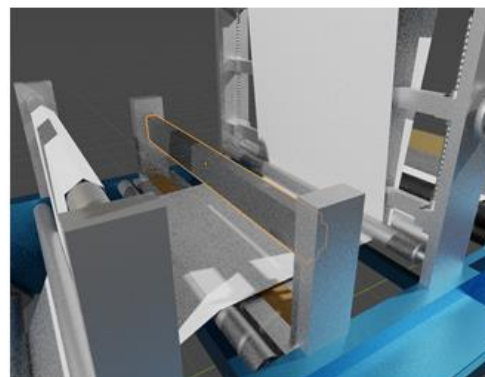


Рисунок 2.2 Паяльник

Паяльник в свою чергу влаштований таким чином, що у нього є дві точки дотику до полотна. Це дозволяє робити два шва одночасно.

▶ 8

Лінія різу

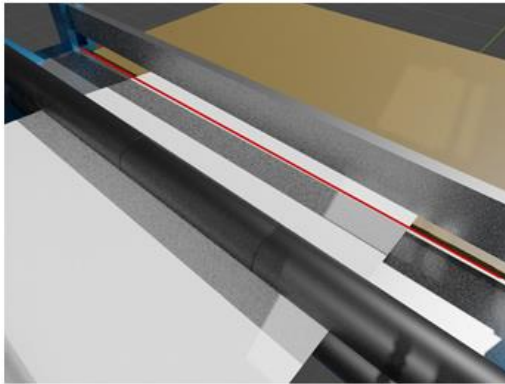


Рисунок 2.3 Червоним позначена лінія різу

Лінія різу – це місце, куди б'є гільотина, саме цей аспект роботи на пакетоформуючій машині потребує найбільшої уваги оператора машини. Оскільки дуже часто полотно з'їжджає та гільотина може відрізати полотно в неправильному місці, що вважається браком, бо виробляемі пакети можуть бути без одного запаяного краю, але з двома швами з іншої сторони. Щоб цього не траплялося оператор наглядає та корегує лінію різу.

▶ 9

Схема автоматизованої системи головного двигуна пакетоформуючої машини

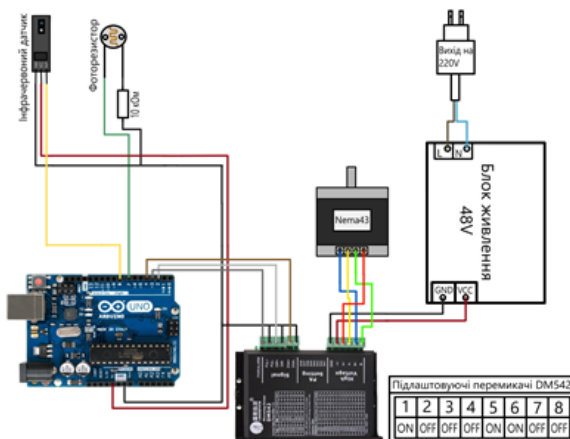


Рисунок 2.4 Схема автоматизованої системи головного двигуна пакетоформуючої машини

Логіка системи: можна розмістити інфрачервоний датчик перед гільотиною, та коли гільотина опуститься – датчик активується та передасть сигнал на Arduino, а та в свою чергу зупинить кроковий двигун, що відповідатиме за протяжку полотна. Коли гільотина підніметься, датчик деактивується та передасть сигнал на Arduino, щоб та відновила роботу крокового двигуна. Але ця система може працювати лише тоді, коли виробляються прозорі пакети. Але іноді пакети йдуть з друком, тоді можна використовувати фотодатчик, який буде відслідковувати фотомітку та зупинити головний двигун, після чого повинна опуститися гільотина, активується інфрачервоний датчик, та коли гільотина підніметься – інфрачервоний датчик деактивується та тим самим запусить головний двигун. Таким чином, оператор не буде змушений постійно наглядати та корегувати лінію різу.

▶ 10

Розташування інфрачервоного датчика та фотодатчика

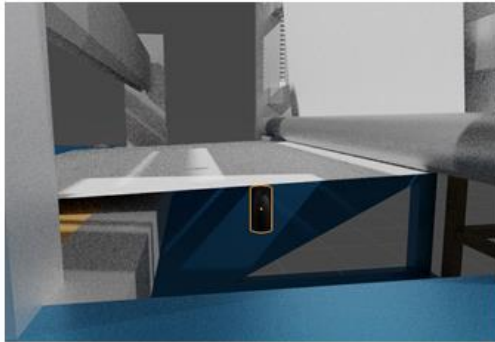


Рисунок 2.5 Розташування фотодатчика

Як було описано вище, фотодатчик відповідає лише за фотомітку, тому може не задіюватись при виробництві прозорих пакетів.

► 11

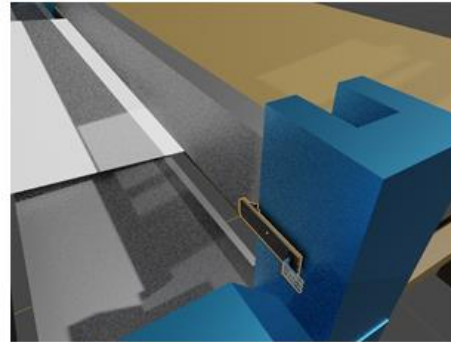


Рисунок 2.6 Розташування інфрачервоного датчика

Інфрачервоний датчик – головний датчик в цій системі, саме він буде давати команду на запуск головного двигуна, який має проїхати заданий інтервал, після чого зупинитись.

Висновки

- В результаті цієї кваліфікаційної роботи, я зробив наглядні приклади, як можна використовувати мікроконтролер Arduino для модернізації застарілого обладнання. Ця робота є прикладом, як автоматизація має допомогти науково-технічному прогресу швидше перейти у наступну стадію промислової революції.
- Апробація
 1. Браташов І.Р. «Поліпшення промисловості завдяки технологіям IoT». І Всеукраїнська науково-технічна конференція «Технологічні горизонти: дослідження та застосування інформаційних технологій для технологічного прогресу України і світу» – Київ, 28 листопада 2023 р.
 2. Браташов І.Р. «Покращення якості життя завдяки технологіям IoT». Тези доповіді V МІЖНАРОДНА НАУКОВО-ТЕХНІЧНА КОНФЕРЕНЦІЯ «Сучасний стан та перспективи розвитку IoT» – Київ, 18 квітня 2024 р.

► 15

Дякую за увагу!

