

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
УМАНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ САДІВНИЦТВА**

ФАКУЛЬТЕТ ЕКОНОМІКИ І ПІДПРИЄМНИЦТВА

Кафедра інформаційних технологій

Допущено до захисту

«_____» _____ 20__ р.

Зав. кафедри, к.т.н., доцент

_____ Ліщук Р.І.

Кваліфікаційна робота бакалавра

**РОЗРОБКА ІНФОРМАЦІЙНОЇ КОМП'ЮТЕРИЗОВАНОЇ ПІДСИСТЕМИ
ВАГОВОГО ДОЗАТОРА**

Виконав: студент 4 курсу, 41-кн групи

Освітньо-професійної програми

«Комп'ютерні науки»

спеціальності 122 «Комп'ютерні науки»

Григорина Євгеній Сергійович

Науковий керівник:

д.т.н., професор _____

Кучерук Володимир Юрійович

Умань – 2023

ВСТУП

Застосування вагових дозаторів є важливою складовою процесу виробництва в багатьох галузях, таких як харчова промисловість, фармацевтика, хімічна промисловість та інші. Вагові дозатори використовуються для точного вимірювання та дозування різних матеріалів, що дозволяє забезпечити високу якість продукції та оптимізувати виробничі процеси.

Проте, існують певні проблеми, пов'язані зі старіючими або недостатньо ефективними інформаційними системами, що керують ваговими дозаторами. Недостатня автоматизація, відсутність централізованого керування та моніторингу, а також недостатня інтеграція з іншими процесами виробництва можуть призводити до зниження продуктивності та збільшення витрат.

Метою даної кваліфікаційної роботи є розробка інформаційної комп'ютеризованої підсистеми для вагового дозатора, яка буде забезпечувати автоматизоване керування, моніторинг та інтеграцію з іншими процесами виробництва.

Для досягнення цієї мети передбачається вирішення таких завдань:

Систематизація та розширення теоретичних і практичних знань з обраного напрямку підготовки, що включає вивчення сучасних технологій та методів управління ваговими дозаторами.

Розвиток навичок самостійної роботи при вирішенні конкретних питань та завдань, що поставлені в роботі, зокрема аналізу існуючих систем керування ваговими дозаторами та розробки нових алгоритмів і методів.

Підготовка студента до самостійної практичної діяльності в умовах сучасного виробництва шляхом вивчення та застосування інформаційних технологій для автоматизації процесів управління ваговими дозаторами.

- Об'єкт дослідження

Об'єктом дослідження є інформаційні процеси в сфері використання вагових дозаторів та технології їх автоматизації шляхом застосування комп'ютерної та комунікаційної техніки. В рамках дослідження будуть

розглянуті основні процеси керування, моніторингу та інтеграції, а також застосовані інформаційні технології.

- Предмет дослідження

Предметом дослідження є інструментальні засоби, такі як математичні моделі, методи та інформаційні технології, що використовуються для дослідження та управління ваговими дозаторами. Вивчення цих засобів дозволить розробити ефективну інформаційну комп'ютеризовану підсистему для вагового дозатора.

- Методи дослідження

Для досягнення поставленої мети в роботі використовуватимуться такі методи дослідження: аналіз наукових джерел, вивчення існуючих систем керування ваговими дозаторами, проведення експериментальних вимірювань та тестувань, моделювання та аналіз даних, програмування та реалізація комп'ютерної підсистеми.

- Інформаційна база дослідження

При підготовці кваліфікаційної роботи використовувалися наукові статті, нормативно-правові акти, державні та міжнародні стандарти, що стосуються використання вагових дозаторів та інформаційних технологій у виробництві.

РОЗДІЛ 1. ВВЕДЕННЯ У ТЕОРИТИЧНИЙ ОГЛЯД ПРЕДМЕТНОЇ ГАЛУЗІ

1.1 Предмет дослідження

Предметом дослідження є комп'ютеризована підсистема вагового дозатора.

Ваговий дозатор - це пристрій, що використовується для точного вимірювання та дозування різних матеріалів у виробничих процесах. Підсистема вагового дозатора має на меті автоматизувати процес дозування матеріалів, забезпечуючи точне та контрольоване додавання речовини або матеріалу у певній кількості. Ця підсистема включає в себе ваговий датчик, комп'ютерну систему керування, програмне забезпечення та інші необхідні компоненти.

Ваговий датчик використовується для вимірювання маси матеріалу, який подається до дозатора. Він здатен точно визначити масу, що дозволяє дозатору забезпечувати точне дозування матеріалів. Неточне дозування може призвести до зниження цінності і якості продукту, перевитрати дефіцитних компонентів і збільшення вартості готової продукції.

Комп'ютерна система керування ваговим дозатором відповідає за контроль та регулювання процесу дозування. Вона отримує дані від вагового датчика, обробляє їх за допомогою програмного забезпечення та виконує необхідні дії для забезпечення точної дози матеріалу.

Для забезпечення точного дозування матеріалу з необхідною точністю використовується процес вимірювання заданої кількості. Точність дозування визначається технологічними вимогами і може бути обґрунтована економічними міркуваннями.

У сучасних умовах, для успішного вдосконалення процесу дозування, необхідно розглядати дозатор як складову частину комплексного устаткування, а не самостійний елемент. Ефективна та якісна робота такої системи залежить від надійності кожного окремого елемента. При проектуванні таких систем можна враховувати всі фактори, які впливають на їх роботу, і процес дозування може менше піддаватися зовнішнім впливам, що призводить до стабільної роботи дозаторів.

1.2 Методи використання вагових дозаторів

Завдяки технологічному прогресу, сьогодні ми маємо доступ до різноманітних технічних пристроїв, які значно полегшують нашу роботу. Одним з таких пристроїв є вагові дозатори, які є незамінними на виробничих та промислових підприємствах.

Вагові дозатори використовуються для автоматичного дозування матеріалів та виробленої продукції. Вони здатні точно виміряти та розподілити речовини, які потім використовуються у сумішах або упаковуються. Ці пристрої досить часто використовуються для зважування сипучих, в'язких і рідких продуктів.



Рис.1.1 Ваговий дозатор

Вагові дозатори представляють собою компактні машини з простою установкою і конструкцією. Зараз вони мають значно розширену функціональність, що дозволяє не тільки виміряти продукти, але й контролювати

їх порції та споживання. Загалом, вагові дозатори можна розділити на дві основні групи: механічні та автоматичні.

- Механічні дозатори вимагають використання спеціальних вантажів і мають більшу похибку у вимірюванні.
- Автоматичні дозатори мають значно меншу похибку, тому вони широко використовуються на заводах, підприємствах та фірмах.

Технологічний прогрес у сфері вагових дозаторів дозволив зробити їх більш точними, функціональними й надійними. Використання таких пристроїв сприяє автоматизації процесів та забезпечує високу якість та ефективність виробництва.

Існують два основних методи дозування для сипучих і рідких матеріалів: об'ємний та ваговий. Зазвичай, цінні матеріали дозуються за вагою, тоді як менш цінні матеріали - за обсягом.

- Об'ємний метод дозування відрізняється легкістю в установці та експлуатації дозуючого обладнання. Однак його низька точність обмежує його застосування. Невисока точність дозування може призвести до значних відхилень у складі сумішей і погіршення якості готової продукції, а також до зайвого використання початкових матеріалів. Це призводить до того, що об'ємний метод рідше використовується, особливо для великої кількості матеріалів.

- Ваговий метод дозування надає більш точні результати, ніж об'ємний метод. Для вагового дозування матеріалів можна використовувати різні типи ваг, такі як платформні, настільні або лабораторні в залежності від ваги потрібних доз та умов процесу дозування.

Наприклад, при використанні товарних платформних ваг з вказівним пристроєм для дозування сипких матеріалів, на платформу ваги розміщується бункер потрібного обсягу. Матеріал відважується з бункера, який має випускний отвір, що може бути закритий клапаном. Бункер з'єднується з вагою за допомогою гнучкої манжети, а спеціальна труба використовується для випуску

повітря з бункера. Таким чином, усі операції з піднесення та пересипання матеріалу відпадають.

Спеціальні ваги для дозування точно фіксують та відважують задану дозу матеріалу. Наприклад, на дозаторі, створеному на базі товарних ваг з максимальним навантаженням 500 кг, можна легко відважити дозу 200 кг з точністю до 200 г або з похибкою 0,1%.

Однак використання звичайних ваг як дозаторів зазвичай не є ефективним, оскільки цей метод вимагає багато часу. Замість цього, для вагового дозування в сталому технологічному процесі раціональніше використовувати спеціальні ваги, які адаптовані до конкретних умов процесу. На сьогоднішній день широко застосовуються спеціальні ручні, напівавтоматичні та автоматичні дозатори в різних галузях виробництва.

Механічні пристрої потребують спеціальних вантажів для своєї роботи, тому мають велику похибку. У автоматичних дозаторах похибка значно менша, тому вони майже повністю витіснили механічні дозатори з ринку.

Проте об'ємний метод залишається вибором для дозування рідин та порошкоподібних матеріалів зі сталою зернистістю. Для рідин можна досягти високої ідентичності і точності за допомогою об'ємних мірних пристроїв, завдяки їх здатності заповнювати ємності будь-якої форми та сталості об'ємної ваги. Тому в деяких технологічних процесах, коли сипучі матеріали дозуються ваговим методом, рідини додаються за об'ємом згідно з рецептурою.

Задовільні результати можна отримати також застосовуючи об'ємний метод дозування для порошкоподібних матеріалів зі сталою зернистістю, які не утворюють грудок або не злипаються.

1.3 Автоматичні дозатори дискретної дії

Автоматичні дозатори дискретної дії розроблені для вимірювання матеріалів порціями з точністю, де дозування відбувається протягом повторюваного циклу.

На деяких сучасних спеціалізованих підприємствах об'ємне дозування втратило свою точність і перевагу отримало дозування за масою. Точність

об'ємного дозування складає приблизно $\pm 3\%$ у кращому випадку, тоді як при дозуванні за масою можна досягти точності від $\pm 0,1\%$ до $\pm 1\%$, залежно від типу дозатора та матеріалу.

Дозування за масою виконується різними методами на різних типах ваг, з урахуванням потужності підприємств, особливостей технологічного процесу та асортименту виробленої продукції. Виробництво використовує дозатори маси на млинах, елеваторах, зерноховищах, стрижнях, складах, харчовій та комбікормовій промисловості.

Зазвичай в дозовані матеріали включають зерно, комбікорм, буряк, картоплю та інші продукти. Порційні дозатори перетворюють безперервний потік матеріалу на рівні масою порції. Конструкція таких дозаторів і їх точність залежать від фізико-механічних властивостей зважуваних матеріалів. Для дозаторів маси необхідно, щоб маса окремих частинок сипучих матеріалів була значно меншою за допустиму похибку ваги.

Гранулометричний склад матеріалу та характеристика його сипучості є основними факторами, які впливають на процес зважування. Висока продуктивність і точність зважування ускладнюють конструкцію дозаторів маси дискретної дії. Вимоги до високої точності вирішуються шляхом зміни режиму живильника. Коли механізм ваги наближається до рівноваги, живильник автоматично перемикається на режим малої продуктивності та додавання матеріалу. У процесі зважування не завжди можна досягти необхідної точності за потрібною продуктивністю, тому часто використовують комбіновані ваги. Спочатку проводиться зважування при великій швидкості, а потім матеріал передається на наступний пристрій для зважування, де він додаватиметься до маси порції. Це дозволяє значно підвищити продуктивність.

1.4 Електро-тензометричні дозатори

Останнім часом електро-тензометричні методи зважування стали широко поширеними в конструкції автоматичних дозаторів. Ці методи засновані на зміні електричного опору дротяних датчиків під дією деформації дроту, спричиненої вагою матеріалу або навантаженням. Результати вимірів перетворюються в

електричні величини, що дозволяє передавати їх на відстань та застосовувати стандартні виконавчі механізми та прилади.

Електро-тензометричні вагові пристрої складаються з трьох основних частин: вагового "тензоелемента" з датчиками, що сприймають навантаження, підсилювальної частини та вказівного або записуючого обладнання для фіксації результатів зважування.

Тензоелемент включає дротяні датчики, які складаються з решітки і основи. Решітка зазвичай виготовляється з дроту високої чутливості з малим температурним коефіцієнтом опору. Геометрія датчиків залежить від умов їх застосування.

Датчики наклеюються на сталевий стрижень спеціальним клеєм, а потім цей стрижень розміщується в герметичному металевому кожуху з висновком для проводів. З'єднання датчиків організовуються за схемою моста Уітсон.

При зважуванні, коли на тензоелемент діє навантаження, він деформується, і це призводить до зміни електричного опору датчиків. Зміни опору підсилюються електронними підсилювачами. Електричні сигнали передаються на вимірювальну апаратуру, де їх можна зчитати на циферблатному вказівнику або зафіксувати на іншому обладнанні для реєстрації.

Однією з переваг електро-тензометричних ваг є можливість дистанційної передачі показань. Прилади можуть бути встановлені на відстані до 100 метрів від точки зважування, що дозволяє зручно розташовувати реєструючі прилади в інших приміщеннях або на центральному пульті управління. Показання можуть передаватися одночасно в кілька точок або зосереджуватися з різних ваг на одному місці. Зважування можна також записувати за допомогою віддалених рекордерів.

Електро-тензометричні ваги є гнучкими та зручними в експлуатації, забезпечуючи точне зважування та зручне зчитування результатів навіть на відстані.

1.5 Недоліки і переваги підсистеми вагового дозатора

Підсистема вагового дозатора, яка використовується в різних промислових процесах, має свої переваги і недоліки.

Переваги:

- **Висока точність дозування:** Однією з основних переваг вагового дозатора є його здатність забезпечувати високу точність дозування матеріалів. Завдяки використанню вагового датчика, який міряє масу матеріалу, можна досягти високої точності у дозуванні.
- **Мінімальні втрати матеріалу:** Вагові дозатори дозволяють мінімізувати втрати матеріалу шляхом точного контролю подачі. Це особливо важливо для дорогоцінних або обмежених ресурсів, де навіть невеликі втрати можуть вплинути на економічну рентабельність процесу.
- **Контроль якості:** Вагові дозатори дозволяють забезпечувати високу якість продукції шляхом точного дозування матеріалів. Це особливо важливо в галузях, де точність дозування має прямий вплив на якість кінцевого продукту, наприклад, в харчовій або фармацевтичній промисловості.
- **Універсальність:** Вагові дозатори можуть бути налаштовані для різних типів матеріалів і процесів. Вони можуть працювати з різними ваговими діапазонами і регулювати швидкість подачі матеріалу відповідно до вимог процесу. Це робить їх універсальними і гнучкими для застосування в різних галузях промисловості.
- **Моніторинг:** Ваговий дозатор дозволяє здійснювати контроль і моніторинг процесу дозування. Завдяки використанню вагового датчика, можна в режимі реального часу контролювати кількість поданого матеріалу і вносити необхідні корективи для забезпечення стабільного процесу.

Недоліки:

- **Вплив зовнішніх факторів:** Підсистема вагового дозатора може бути вразливою до зовнішніх факторів, таких як вібрації, температурні зміни або поштовхи. Ці фактори можуть вплинути на точність вимірювання і спотворити результати дозування.

- **Чутливість до забруднень:** Ваговий дозатор може бути чутливим до забруднень, особливо коли мова йде про дрібні матеріали або матеріали з високою адгезією. Забруднення на датчику або в системі дозування можуть спотворити вимірювання і призвести до неточного дозування.

- **Складність налаштування:** Налаштування вагового дозатора може бути складним і вимагати додаткових знань і навичок. Встановлення правильних параметрів, таких як чутливість датчика, може бути складною задачею і вимагати досвіду для досягнення оптимальних результатів.

- **Обмежена масштабованість:** Вагові дозатори можуть бути обмежені у вагових діапазонах, які вони можуть ефективно обробляти. Великі масштаби вимірювання можуть вимагати використання додаткових або спеціалізованих систем дозування.

- **Потреба у додатковому обладнанні:** Вагові дозатори часто потребують додаткового обладнання, такого як системи керування, інтерфейси з комп'ютером або програмне забезпечення для керування і моніторингу процесу. Це може збільшувати вартість встановлення та підтримки системи.

- **Вартість:** Вагові дозатори можуть бути досить вартісними в порівнянні з іншими методами дозування. Вони включають в себе додаткове обладнання, таке як вагові датчики і системи керування, що збільшує загальну вартість устаткування.

Враховуючи ці переваги і недоліки, вибір вагового дозатора повинен здійснюватися з урахуванням конкретних вимог і характеристик процесу. У деяких випадках використання вагового дозатора може бути невід'ємною складовою для досягнення високої точності при дозуванні. Однак, існують ситуації, коли інші методи дозування можуть бути більш ефективними.

1.6 Тензодатчики та датчики ваги

Багато користувачів віддають перевагу використанню електронних телекомунікаційних мереж у своїй роботі через їх зручність і точність. Для забезпечення стійкості і точності отриманих вимірювань, широко використовуються тензодатчики для ваг.

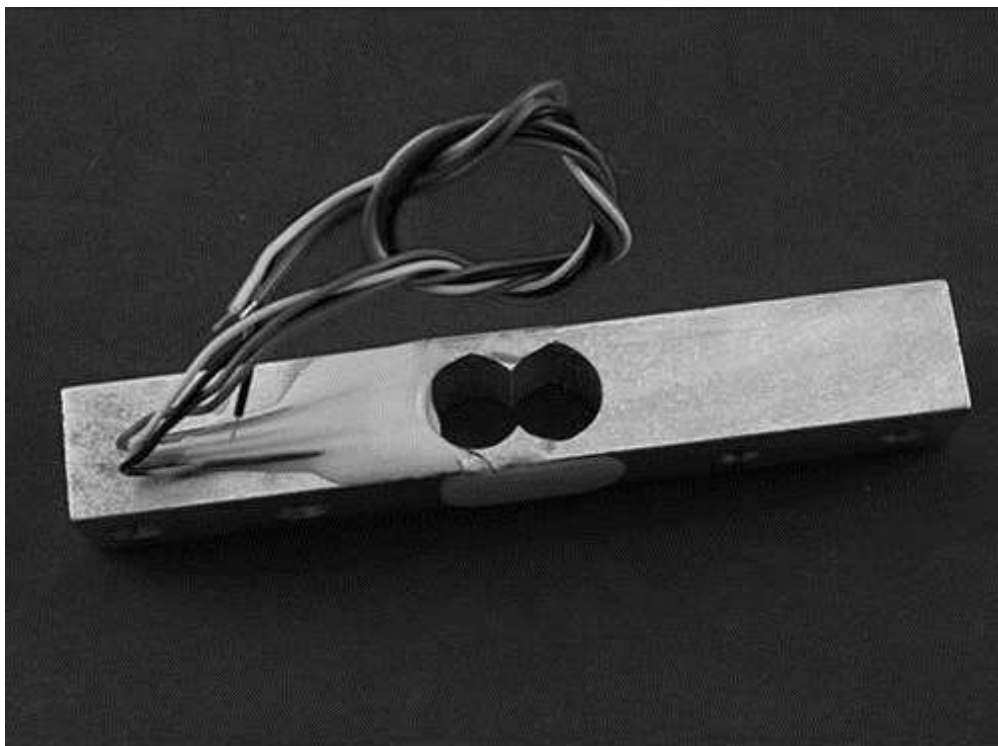


Рис. 1.2 Тензодатчик

Вимірювання деформації має велике значення в різних галузях промисловості. У таких випадках використовуються тензодатчики ваги, які перетворюють рівень деформації на електричні сигнали, що дозволяють визначити ступінь стиснення або розтягування об'єкта. Сучасні тензодатчики ваги є резистивними перетворювачами, які є важливою складовою обладнання для вимірювання ваги. Вони використовуються для вимірювання ваги шляхом впливу навантаження на датчик, який передає сигнал на ваговий індикатор через контрольні контакти. Індикатор отримує цю інформацію, яка представляє результат зважування.

Існує багато типів тензодатчиків та сфер їх застосування. Тензометричні датчики можуть бути класифіковані не тільки за їх форм-фактором, але й за конструкцією. Наприклад, балкові тензодатчики мають схожу форму з балками та працюють на основі принципу вигину. Вони використовуються у різних вагах, включаючи підлогові електронні ваги. Максимальне навантаження для балкових тензодатчиків зазвичай становить від кількох кілограм до 5 тонн.

Одноточкові тензодатчики також мають схожу форму з балками, але для вагів достатньо одного такого датчика. Вони можуть компенсувати згинальні

сили і дозволяють створити економічно ефективну конструкцію ваги. Максимальне навантаження для одноточкових тензодатчиків знаходиться в діапазоні від 2 до 3 кілограм.

Тензодатчики колонного та циліндричного типу використовуються для роботи з великими навантаженнями. Циліндричні тензодатчики часто застосовуються в бункерному зважуванні, зазвичай разом з ваговими модулями. Колонні та циліндричні тензодатчики також зустрічаються в автомобільних вагах. Максимальне навантаження для цих типів тензодатчиків може бути в діапазоні від 250 кг до кількох сотень тонн.

Тензодатчики S-подібної форми призначені для навантажень розтягування, що виникають, наприклад, в ковшових вагах. Їх також називають датчиками розтягування через основний напрям навантаження. Максимальне навантаження для S-подібних тензодатчиків може становити від 25 кг до 10 тонн. Вибір оптимального типу тензодатчика слід здійснювати залежно від вимог області застосування та передбачуваного навантаження.

Використання тензодатчиків у вагах для точного вимірювання різних навантажень є широко поширеним явищем, яке можна спостерігати в будь-якому супермаркеті. Більшість товарів, що продаються в упаковках, мають встановлену масу, що свідчить про їх фасування за допомогою машин з використанням тензодатчиків. У м'ясному відділі касових точок також зазвичай розташовані ваги, оснащені тензодатчиками. Готові торти часто виробляються на заводах, де компоненти рецепту змішуються у великому резервуарі з використанням ваг, що мають вбудовані тензодатчики. При доставці борошна на заводи, його зважують на автомобільних терезах, що можуть виміряти масу всієї вантажівки і мають в собі кілька тензодатчиків. Зважування, яке виконується за допомогою тензодатчиків, імовірно, є найточнішим методом визначення маси матеріалу. Цей метод широко використовується практично в усіх галузях. Подані приклади лише невелика частина ситуацій, які демонструють різноманітність тензодатчиків і сфер їх застосування.

Тензодатчики призначені для вимірювання сили або ваги в умовах, що можуть бути непридатними для багатьох інших датчиків. Вони використовуються як невід'ємна частина електронних систем зважування, але водночас є дуже вразливими до пошкоджень.

Можливі причини поломки тензометричних датчиків включають:

- **Перевантаження:** надмірне навантаження на тензодатчик може пошкодити його і спотворити вимірювання.
- **Вплив небезпечних факторів:** такі як блискавка, стрибки напруги, хімічні речовини або волога можуть спричинити його поломку.
- **Неправильна установка:** неправильне встановлення тензодатчика, наприклад, надмірне затягування або недостатнє закріплення, може призвести до його поломки.
- **Вібрації та удари:** сильні вібрації або ударні навантаження можуть пошкодити тензодатчик і зіпсувати його вимірювання.
- **Пошкодження кабелю:** пошкодження кабелю, наприклад, розриви, розрізи або замикання, можуть призвести до втрати передачі даних або неправильних показань.

Для виявлення проблем з тензодатчиками і тестування їх працездатності рекомендується виконати наступні кроки:

- **Перевірити систему зважування на наявність замикань або пошкоджень, зокрема на платформі.**
- **Оглянути тензодатчики на наявність корозії або інших ознак пошкоджень.**
- **Перевірити кабельні з'єднання між тензодатчиками, розподільною коробкою та ваговим терміналом.**
- **Використовувати високоточний цифровий вольтметр і омметр для вимірювання балансу нуля і цілісності мостової схеми тензометричного датчика.**
- **Використовувати мегаомметр для вимірювання опору ізоляції тензодатчика.**

- Порівнювати результати вимірювань з параметрами, зазначеними в сертифікаті калібрування тензодатчика.

В роботі тензодатчика можуть бути похибки, наприклад помилока нуля є однією з типових похибок вимірювання тензодатчиків. Вона виникає, коли тензодатчик не повертається до свого початкового значення після зняття навантаження. При правильному функціонуванні тензодатчика, коли навантаження відсутнє, вихідний сигнал датчика ваги повинен бути рівним нулю або дуже близьким до нуля.

Похибка нуля може мати декілька причин. По-перше, внутрішні напруги в тензодатчику можуть генерувати невеликі електричні сигнали, навіть коли навантаження відсутнє. Ці внутрішні напруги можуть появлятися через електричні компоненти, такі як підсилювачі, перемикачі або опір.

Другою причиною похибки нуля є дрейф компонентів. З часом електричні компоненти тензодатчика можуть зазнавати змін, що впливає на їх вихідний сигнал. Наприклад, опір може змінюватися під впливом температури або старіння компонентів. Це може спричиняти зсув нульового сигналу тензодатчика.

Третя причина похибки нуля - неправильне калібрування. Якщо тензодатчик не був належним чином калібрований або налаштований перед використанням, можуть виникати похибки нуля. Неправильне налаштування нульової точки або некоректне установлення тензодатчика можуть призводити до зсуву нульового сигналу.

Для усунення або зменшення похибки нуля рекомендується провести правильне калібрування тензодатчика. Це включає налаштування нульової позиції або корекцію нульового сигналу. Також слід дотримуватися рекомендацій виробника щодо правильного встановлення та використання тензодатчика. Регулярний моніторинг функціонування тензодатчика і перевірка його нульового сигналу допоможуть виявити похибку і прийняти відповідні заходи для корекції. Крім того, при виборі тензодатчика слід звернути увагу на

якість виробника і характеристики пристрою. Якісні тензодатчики зазвичай мають меншу похибку нуля та кращу стабільність.

Тензорезистори відрізняються від звичайних резисторів тим, що вони можуть змінювати своє значення опору залежно від механічних навантажень. Вони складаються з тонкої нитки чутливого сплаву, розташованої по ґратах, що нагадує лабіринт. Коли тензорезистор розтягується або стискається, змінюється його опір. Зазвичай такі зміни мають дуже малі значення, тому для точного оброблення цих даних потрібне високоякісне обладнання.

Німецькі тензодатчики відомі своєю високою якістю і високою ціною. Вони виробляються з використанням передових технологій, які дозволяють перевищити заявлені характеристики. Це призводить до того, що працездатність цих датчиків перевищує очікування покупців. Однак, вартість може бути значним фактором, що обмежує їх доступність.

У середньому ціновому сегменті тензодатчики можуть бути на 50% дешевшими, ніж їх дорогі аналоги. Ці датчики виробляються в країнах, таких як США, Корея, Китай. Вони поєднують в собі вигідну комбінацію ціни і якості. Більшість ваг та вимірювального обладнання використовують саме ці тензодатчики, оскільки вони гарантують надійну роботу і відповідають вимогам термінів експлуатації.

У низькому ціновому сегменті тензодатчики представлені продукцією з Кореї, Китаю, Туреччини. Вони також можуть бути на ринку і в готових електронних вагах. Однак, ці датчики зазвичай мають нижчу якість, що відображається в їх низькій ціні. Хоча іноді економія може бути привабливою, використання низької якісних тензодатчиків може призвести до витрат в майбутньому, оскільки вони можуть виявитися менш надійними та швидше зношуватися.

Отже, вибір тензодатчика залежить від бюджету, вимог до якості та довіри до виробника. Кожен ціновий сегмент має свої переваги та обмеження, і розумний підхід полягає в знаходженні балансу між ціною і якістю відповідно до конкретних потреб і можливостей.

1.7 Мета і завдання роботи

Метою даної роботи є розробка інформаційної комп'ютеризованої підсистеми для вагового дозатора, яка забезпечить підвищення точності, стабільності та ефективності процесу дозування. Головною метою є впровадження автоматизованого контролю та управління ваговим дозатором з використанням сучасних інформаційних технологій.

Для досягнення поставленої мети необхідно вирішити наступні завдання:

- Аналіз сучасного стану вагових дозаторів: Провести детальний аналіз та огляд існуючих вагових дозаторів, їхніх переваг та недоліків. Визначити основні технічні характеристики, принципи роботи, використовувані методи дозування та управління. Це дозволить з'ясувати сучасний стан цієї області та визначити потенційні напрями поліпшень.

- Вивчення інформаційних технологій в автоматизованому контролі вагових дозаторів: Ознайомитись з різними інформаційними технологіями, що використовуються в автоматизованому контролі вагових дозаторів. Розглянути можливості застосування сучасних програмних засобів, сенсорів, мережевих протоколів та інших компонентів для побудови ефективної інформаційної підсистеми.

- Розробка методів та алгоритмів розв'язання задач вагового дозування: Визначити та розробити методи та алгоритми, які дозволять досягти точності та стабільності в процесі вагового дозування. Розглянути різні підходи до калібрування вагових датчиків, використання компенсаційних алгоритмів, оптимізацію параметрів дозування та інші техніки для досягнення найкращих результатів.

- Розробка інформаційної комп'ютеризованої підсистеми вагового дозатора: Розробити архітектуру та програмне забезпечення інформаційної підсистеми, яке забезпечить збір, обробку та аналіз даних, керування процесом дозування, відображення результатів та взаємодію з оператором. Забезпечити інтеграцію підсистеми зі зовнішніми пристроями та системами для обміну даними та виконання вимог виробничого середовища.

- Тестування та експериментальне дослідження: Провести експериментальні дослідження і тестування розробленої інформаційної підсистеми на реальних вагових дозаторах. Оцінити її ефективність, точність, стабільність та надійність. Виявити можливі проблеми, провести аналіз результатів і внести необхідні виправлення для подальшого вдосконалення системи.

- Аналіз отриманих результатів і висновки: Проаналізувати результати експериментальних досліджень, порівняти їх зі стандартами та вимогами. Зробити висновки про ефективність розробленої інформаційної підсистеми та можливості подальшого вдосконалення. Визначити переваги та обмеження системи, а також внести рекомендації для подальшого використання вагового дозатора з інформаційною підсистемою.

Загальна мета роботи полягає у розробці інформаційної комп'ютеризованої підсистеми вагового дозатора, що дозволить покращити точність, стабільність та ефективність процесу дозування. Завдання роботи включають аналіз сучасного стану вагових дозаторів, вивчення інформаційних технологій в автоматизованому контролі, розробку методів та алгоритмів розв'язання задач дозування, реалізацію інформаційної підсистеми, проведення експериментальних досліджень та аналіз отриманих результатів.

РОЗДІЛ 2. ВИЗНАЧЕННЯ ТЕХНОЛОГІЇ ДЛЯ РОЗРОБКИ ПІДСИСТЕМИ ВАГОВОГО ДОЗАТОРА

2.1 Платформа Arduino

Незалежно від конкретної моделі, плати Arduino мають схожу структуру. На платі розміщений мікроконтролер, який є основною обчислювальною одиницею. Він забезпечує обробку і виконання програм, які розробляються.

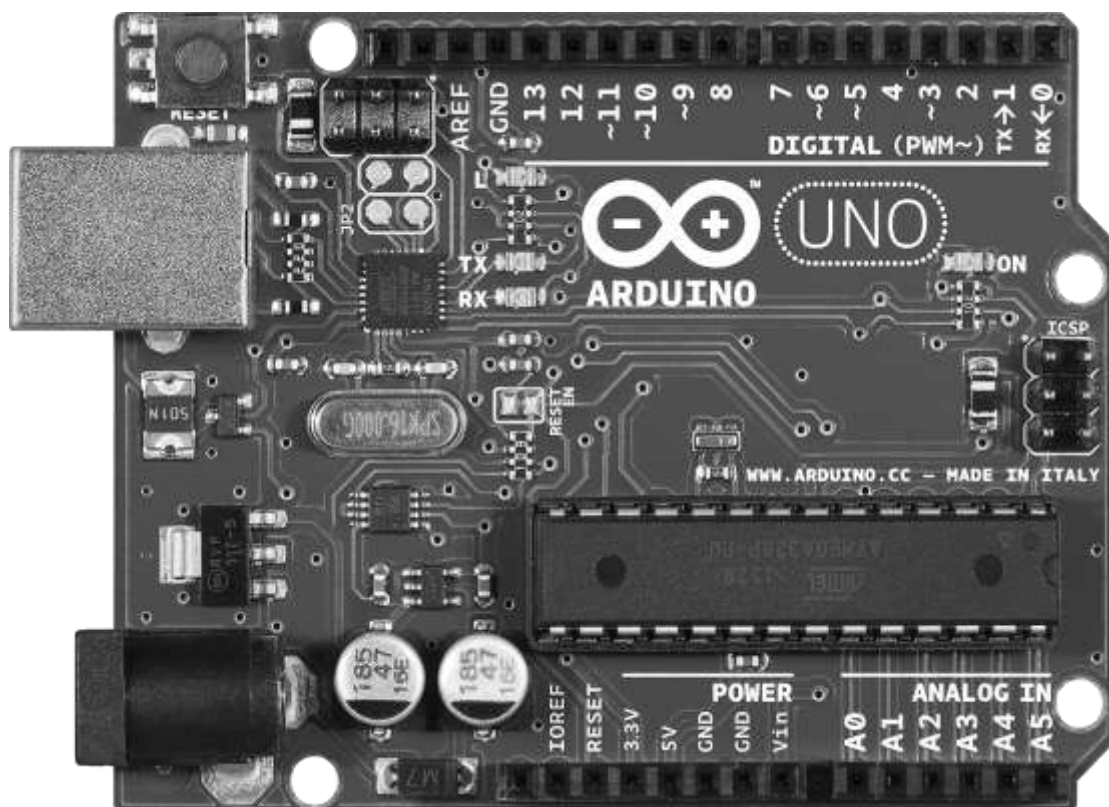


Рис. 2.1 – Плата Arduino Uno

Поруч з мікроконтролером розташовані інші мікросхеми, які відповідають за різні функції, такі як обмін даними через USB інтерфейс. Цей інтерфейс дозволяє підключати плату до комп'ютера для програмування та завантаження коду. Наявність USB порту на платі спрощує взаємодію з Arduino.

Контакти вводу-виводу (англ. input/output, I/O) є важливою частиною плати Arduino. Ці контакти забезпечують комунікацію з зовнішнім світом, дозволяючи нам підключати датчики, вивідні пристрої та інші компоненти до плати Arduino.

Окрім плати розробника, Arduino також надає програмне забезпечення, відоме як IDE (Integrated Development Environment). Це спеціальне середовище розробки, яке дозволяє писати, перевіряти та завантажувати програми на плату Arduino. IDE має зручний інтерфейс та набір інструментів для розробки програмного коду.

Отже, плата розробника Arduino, разом з програмним забезпеченням IDE, утворюють основу цієї платформи, яка надає зручність у розробці електронних проектів і взаємодії з різними пристроями.

Arduino IDE - це вбудоване середовище розробки, яке використовується для програмування плат Arduino. Ця програма має багато функцій, що спрощують процес розробки та взаємодії з платою Arduino.

Однією з головних функцій Arduino IDE є можливість написання коду для плат Arduino. В ній можна створювати програми, використовуючи мову програмування Arduino, яка базується на мові C++, проте Arduino IDE дещо відрізняється за своєю функціональністю, а саме за такими операторами як `setup()`: Вона викликається лише один раз при старті скетча і використовується для початкової конфігурації. В цій функції можна встановити параметри зв'язку, ініціалізувати змінні та виконати інші необхідні налаштування. Друга функція `loop()`: Після виконання функції `setup()`, яка встановлює початкові налаштування, функція `loop()` починає свою безперервну роботу. Вона виконується у циклі і дозволяє програмі виконувати потрібні обчислення та реагувати на події. Ще вона забезпечує безперервне виконання основної логіки програми, такої як зчитування датчиків, аналіз даних та взаємодія із зовнішніми пристроями. Вона дозволяє проекту функціонувати впродовж усього часу роботи мікроконтролера.

Також Arduino має набір портів, які можуть бути використані для отримання інформації зовнішніх джерел або встановлення значень для зв'язку зі зовнішніми пристроями. Ці порти можуть використовуватись для керування різними пристроями, наприклад, світлодіодами, реле або кнопками, а також для отримання аналогових даних від датчиків.

Arduino IDE має вбудований компілятор, який перетворює написаний код в машинний код, зрозумілий для мікроконтролера на платі Arduino. Це дозволяє перевірити правильність синтаксису та виявити помилки в програмі до її завантаження на плату.

Іншою важливою функцією Arduino IDE є можливість завантаження скомпільованого коду на плату Arduino. За допомогою USB-порту плати та спеціального програматора в IDE, можна переслати програму на плату для її виконання. Це дозволяє тестувати та запускати проекти на реальній апаратурі.

Крім того, Arduino IDE має вбудований налагоджувач, який допомагає виявляти та виправляти помилки в програмі. Ми можемо встановлювати точки зупинки в коді, стежити за значеннями змінних та аналізувати виконання програми, щоб знайти й усунути помилки.

Окрім основних функцій, Arduino IDE також підтримує розширення, такі як шилди Arduino. Шилди - це додаткові модулі, які можна підключити до плати Arduino для розширення її функціональності. Arduino IDE дозволяє взаємодіяти з цими модулями та використовувати їх у проектах.

Отже, Arduino IDE - це потужний інструмент, який забезпечує зручне та ефективне програмування плат Arduino. Він дозволяє розробляти проекти, тестувати їх на платі, та взаємодіяти з різноманітними модулями для розширення можливостей вашого проекту.

Arduino здобув популярність завдяки своїй простоті використання та гнучкості. Люди можуть створювати програмовані електронні системи з використанням Arduino, навіть не маючи глибоких знань з електроніки. У цьому їм значно допомагають шилди.

Один з основних принципів Arduino полягає в тому, що вся складність вбудована у шилди. Наприклад, для побудови системи з підключенням до мережі Ethernet, можна використовувати Ethernet-шилд. Цей шилд має вбудований контролер Ethernet, який виконує всю необхідну роботу з комунікацією через Ethernet.

Без шилда довелося б самотійно розробляти та підключати контролер Ethernet, що потребувало б значних зусиль та знань. Однак, завдяки шилду, достатньо просто підключити його до плати Arduino, і всі необхідні з'єднання та програмні бібліотеки вже вбудовані. Ми можемо просто використовувати функції бібліотек для взаємодії з контролером Ethernet без необхідності вдаватися в складні деталі його роботи. Це дозволяє значно спростити процес розробки, навіть для тих, хто не має глибоких знань з електроніки. Бібліотеки Arduino можна легко встановити з офіційного веб-сайту Arduino або з використанням Arduino. Існує велика спільнота користувачів Arduino, яка активно створює, розповсюджує та оновлює бібліотеки для різних проектів. Також можна створювати свої власні бібліотеки, якщо потрібно.

У бібліотеках Arduino зібрані набори функцій, які спрощують програмування мікроконтролерів. Ці бібліотеки забезпечують доступ до різноманітних функцій, таких як робота з пінами, комунікація з іншими пристроями через різні протоколи (наприклад, I2C, SPI, UART), робота з LCD-дисплеями, робота з сенсорами, керування моторами та багато іншого.

Arduino має відкрите співтовариство, що сприяє його популярності. Усі схеми та документація для плат Arduino є відкритими та доступними для всіх. Це означає, що можна використовувати цю інформацію для розробки своїх власних плат або модифікації існуючих. Велике співтовариство користувачів Arduino також активно допомагає одне одному, ділиться знаннями та досвідом, що робить Arduino ще більш доступним для широкого кола людей.

Таким чином, Arduino завоював популярність завдяки комбінації простоти використання, готових модулів розширення (шилдів) та активній спільноті. Це робить його ідеальним вибором для початківців та досвідчених розробників, які шукають швидко та просте рішення для своїх проектів.

2.2 Підключення Ардуїно до тензодатчика за допомогою модуля HX711

Для досягнення точного вимірювання ваги, необхідно розробити та зібрати конструкцію датчика, а також мати під рукою еталонні ваги для порівняння результатів з модулем. Для створення ваг на базі платформи Arduino, у датчика

є отвори з різьбленням, які дозволяють збирати необхідну конструкцію. Основним компонентом модуля є аналого-цифровий перетворювач з 24-бітною роздільною здатністю для ваг, який використовує мікросхему HX711. Цей перетворювач підключається до тензодатчика.

Таким чином, шляхом збирання конструкції датчика ваги та використання аналого-цифрового перетворювача HX711, можна забезпечити точне вимірювання ваги та порівнювати його з еталонними значеннями.

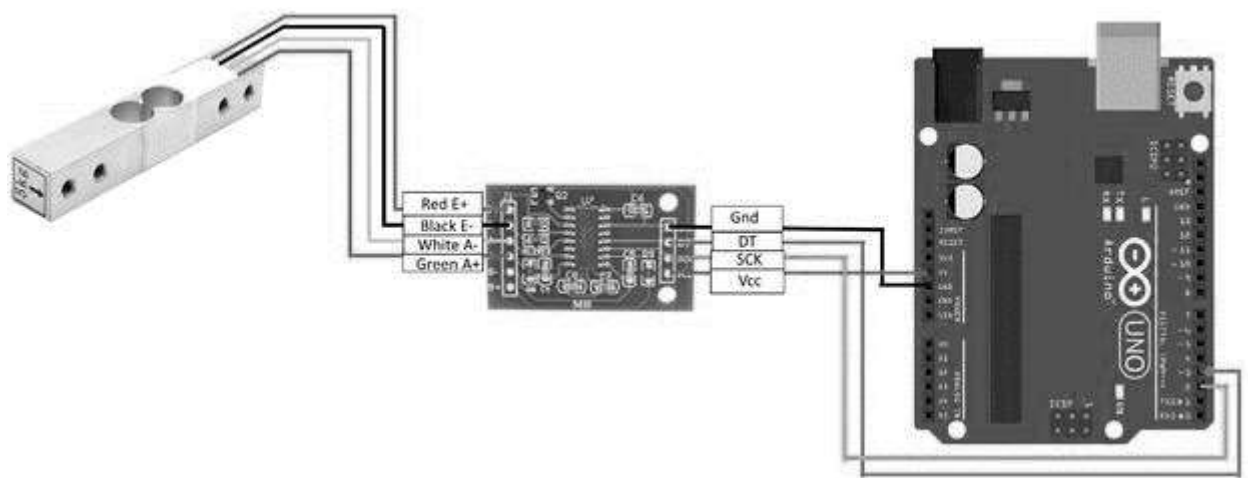


Рис.2.2 Схема підключення тензодатчика до Arduino Uno за допомогою АЦП HX711

Після складання ваги на платформі Arduino, необхідно провести калібрування тензорезистивного датчика. Можна використати еталонні ваги для вимірювання та порівняння результатів. За допомогою скетчу для калібрування, коригується значення тензодатчика, щоб вони відповідали еталонним значенням.

Калібрування тензорезистивного датчика ваги з використанням модуля HX711 включає два основних етапи: нульове калібрування і калібрування масштабу.

На початку проводиться нульове калібрування. Це означає, що датчик ваги повинен бути у спокійному стані і без будь-якого навантаження. Запускається спеціальний скетч на платі Ардуіно, який встановлює нульове значення ваги,

відоме також як "Offset" або "Zero". Це значення фіксується і використовується як початкове значення ваги.

Після нульового калібрування проводиться калібрування масштабу. Для цього на датчику розміщується еталонна вага, яка повинна бути точно відомою і має бути виміряна раніше. Значення еталонної ваги вводиться в скетч для калібрування. Після запуску скетчу, воно отримує значення ваги від датчика.

Отримані значення ваги під час нульового калібрування і калібрування масштабу використовуються для обчислення коефіцієнтів калібрування. Масштабний коефіцієнт (Scale Factor) розраховується як різниця між еталонною вагою і нульовим значенням, поділена на різницю між еталонним значенням і нульовим значенням. Зміщення (Offset) обчислюється, віднявши від еталонної ваги добуток масштабного коефіцієнта і нульового значення.

Отримані коефіцієнти калібрування (масштабний коефіцієнт і зміщення) застосовуються до програми або скетчу для Ардуіно. Ці коефіцієнти використовуються для коригування зчитуваних значень ваги з датчика. Шляхом множення значення з датчика на масштабний коефіцієнт і додавання зміщення отримується скориговане значення ваги.

Калібрування тензорезистивного датчика ваги з модулем HX711 дозволяє забезпечити більш точні вимірювання ваги, адаптуючи його до конкретних умов і вимог.

Підключається датчик до модуля HX711 згідно зі схемою, а сам модуль HX711 підключається до плати Arduino. З'єднуються порти DT і SCK з будь-якими доступними цифровими портами мікроконтролера (наприклад, можна використовувати порти 2 і 3). Після підключення складеної схеми завантажується скетч для калібрування датчика в плату Arduino.

2.3 LCD та OLED дисплеї

LCD (Liquid Crystal Display) дисплеї - це популярні викружні екрани, які використовуються для відображення тексту, чисел та графіки. Вони засновані на технології рідинних кристалів, що забезпечує їм низьке споживання енергії і

можливість відображення на них інформації навіть при прямому сонячному світлі.

Основні характеристики LCD дисплеїв:

- Типи LCD дисплеїв:

- Текстові LCD дисплеї: Вони здатні відображати обмежену кількість символів, які можуть бути організовані в рядки та стовпці. Типово використовуються для виводу текстової інформації.

- Графічні LCD дисплеї: Вони здатні відображати зображення, графіку та текст. Мають більшу гнучкість у відображенні різних елементів на екрані.

- Роздільна здатність (Resolution): Визначає кількість рядків та стовпців, які можуть бути відображені на екрані. Наприклад, 16x2 означає 16 символів в рядку та 2 рядки.

- Контролер: Керує роботою LCD дисплею, приймає команди та дані для відображення. Часто використовуються контролери типу HD44780.

- Зворотне підсвічування (Backlight): Багато LCD дисплеїв мають вбудовану підсвітку, яка дозволяє читати інформацію на екрані в темряві. Зазвичай використовується LED підсвічування, але існують і інші технології.

- Взаємодія з мікроконтролерами: Для виводу інформації на LCD дисплей необхідно використовувати спеціальну бібліотеку для контролю дисплею. Ці бібліотеки дозволяють керувати пінами дисплея, встановлювати текст, числа, графіку та контролювати підсвічування.

- Керування текстом і графікою: LCD дисплеї дозволяють виводити текст, числа та символи у визначеному розташуванні. Більш розширені графічні дисплеї дозволяють відображати точки, лінії та інші елементи графіки, дозволяючи створювати власні зображення.

- Мінімальна геометрія: Кожен символ або графічний елемент на LCD дисплеї зазвичай складається з пікселів або сегментів. Розмір та форма символів обмежені геометрією дисплея.

- Інтерфейси: LCD дисплеї можуть підключатись до мікроконтролерів через різні інтерфейси, такі як паралельний, I2C, SPI тощо. Вибір інтерфейсу залежить від доступних пінів на мікроконтролері та можливостей дисплея.

- Додаткові функції: Деякі LCD дисплеї мають додаткові функції, такі як кнопки управління, контрастність, регулювання підсвічування, звукові сигнали та інші.



Рис.2.2 LCD дисплей 1602

LCD дисплеї широко використовуються в різних пристроях, таких як цифрові годинники, термостати, вимірювальні прилади, дозатори, мікроконтролерні проекти, пристрої вбудованої системи та багато іншого. Вони є досить простими у використанні і доступними, що робить їх популярними серед електронних проектів та прототипів.

Для підключення і налаштування LCD дисплея до Arduino Uno потрібно зробити кроки:

- Підключення апаратної частини:
 - Підключити дисплей до Arduino Uno за допомогою проводів.
 - З'єднати пін RS (Register Select) на дисплеї з піном 12 на ArduinoUno.
 - З'єднати пін E (Enable) на дисплеї з піном 11 на Arduino Uno.

- З'єднати піни D4, D5, D6, D7 на дисплеї з пінами 5, 4, 3, 2 відповідно на Arduino Uno.

- Підключити пін VSS (Ground) на дисплеї до землі (GND) на ArduinoUno.

- Підключити пін VDD (Power) на дисплеї до позитивного живлення (5V) на Arduino Uno.

- Підключення програмного забезпечення:

- Встановлюється бібліотека LiquidCrystal для Arduino IDE. Ця бібліотека дозволяє легко керувати LCD дисплеями.

- У вкладці "Sketch" потрібно вибрати "Include Library" -> "LiquidCrystal", щоб додати бібліотеку до програми Arduino.

- Ініціалізація дисплея в коді:

- У блоку `setup()` додати наступні рядки коду:

```
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);  
lcd.begin(16, 2);
```

Перший рядок коду ініціалізує об'єкт `lcd` із пінами, до яких підключений LCD дисплей.

Другий рядок коду вказує розмір дисплея (16 стовпців, 2 рядки).

- Використання дисплея:

- Для виводу тексту на дисплей можна використати наступні функції:

- `lcd.setCursor(col, row)` - встановлює позицію курсора на дисплеї (стовпець, рядок).

- `lcd.print(text)` - виводить текст на поточну позицію курсора, а

- `lcd.clear()` - очищує вміст дисплея.

OLED Дисплеї

OLED-дисплеї (органічні світлодіодні дисплеї) є одними з типів дисплеїв, які використовують органічні світлодіоди для відтворення зображень. Вони володіють кращою якістю зображення, більшим контрастом і ширшими кутами огляду порівняно з іншими типами дисплеїв, такими як LCD.

OLED-дисплеї складаються з безлічі невеликих світлодіодів, які створюють світло під час проходження електричного струму через них. Оскільки олед-дисплеї не використовують підсвітку, вони можуть демонструвати чорний колір шляхом вимкнення відповідних світлодіодів, що дозволяє досягти високого контрасту і глибини чорного. Крім того, олед-дисплеї забезпечують більш насичені кольори та кращу репродукцію кольорової гами порівняно з LCD-дисплеями.

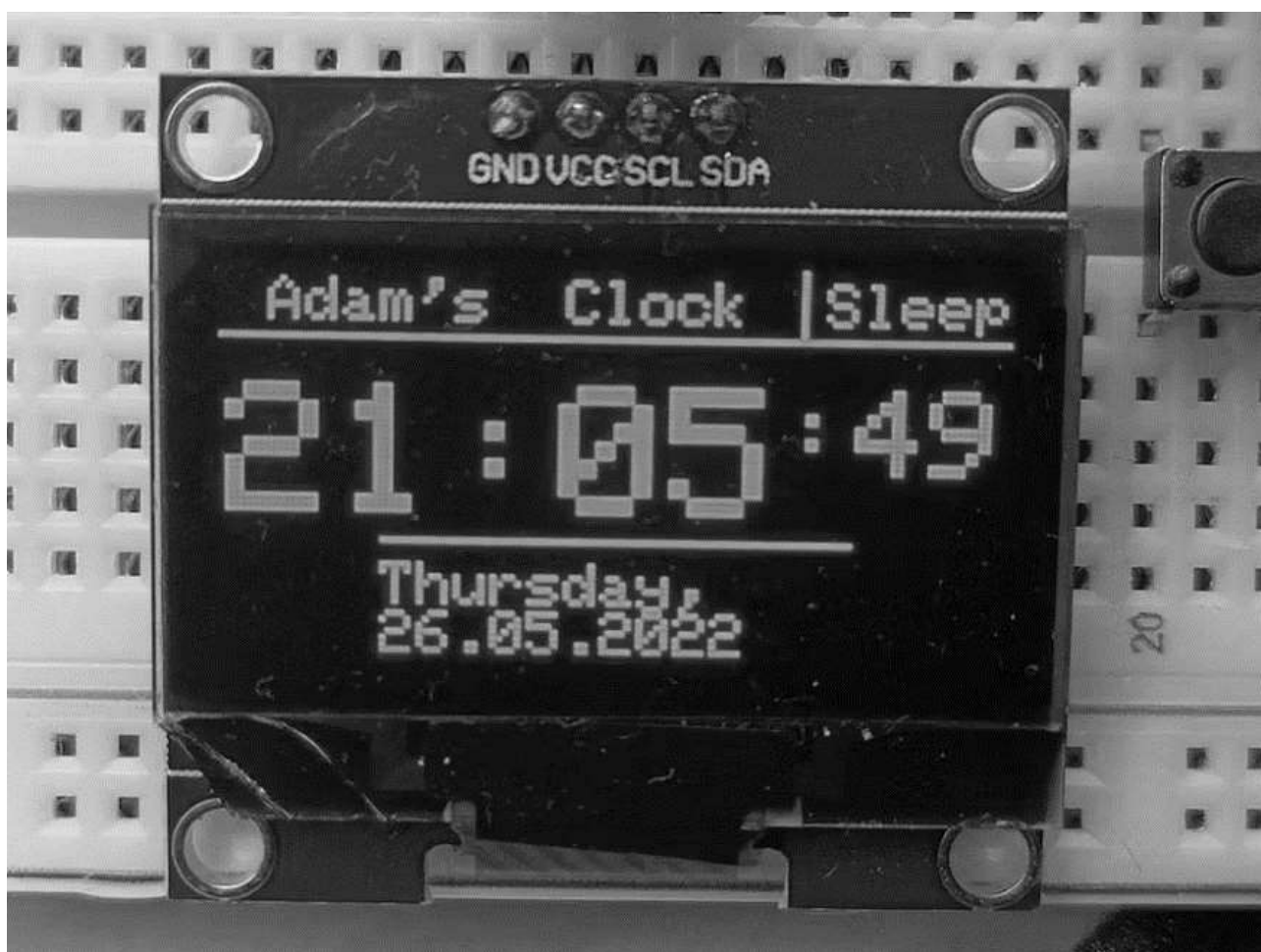


Рис.2.3 OLED дисплей SH1106

SH1106 є однією з багатьох мікросхем управління олед-дисплеями. Ця мікросхема широко використовується для керування олед-дисплеями з роздільною здатністю 128x64 пікселів, які використовуються у різних електронних пристроях, таких як мікроконтролери, ардуіно-плати та інші.

SH1106 має вбудований контролер, який керує відображенням зображень на олед-дисплеї. Вона підтримує різні режими роботи, включаючи текстовий,

графічний і комбінований режими, що дозволяє відображати текст, лінії, графіку та інші об'єкти на дисплеї.

SH1106 підтримує комунікацію з мікроконтролером або іншими пристроями через різні інтерфейси, включаючи I2C і SPI. Це робить її сумісною з багатьма платформами та мікроконтролерами, що робить її популярним вибором для багатьох проектів.

Олед-дисплеї на базі SH1106 мають досить хорошу якість зображення, відмінну чіткість і контрастність, а також широкий кут огляду. Вони також володіють низьким споживанням енергії, що робить їх ефективними для використання в пристроях з обмеженим джерелом живлення.

Загалом, олед-дисплеї SH1106 є популярними компонентами для вбудованих систем і проектів, які потребують невеликих, енергоефективних та високоякісних дисплеїв.

2.4 Емуляція Arduino в Proteus

Якщо у нас немає фізичних плат ми можемо завжди звернутися до такої програми як Proteus. Він є комплексним програмним забезпеченням, розробленим компанією Labcenter Electronics. Він надає розробникам електроніки інструменти для симуляції, моделювання та дизайну електронних схем та друкованих плат (PCB).

Основні компоненти Proteus включають ISIS (Interactive Simulation Integrated Software) і ARES (Advanced Routing and Editing Software). ISIS використовується для створення електронних схем та симуляції їх роботи, а ARES - для проектування PCB.

ARES надає різноманітні функції для розміщення компонентів на платі, створення шляхів з'єднання між ними, розміщення отворів для монтажу компонентів. За допомогою ARES можна створювати як одношарові, так і багатошарові плати зі складними маршрутами.

Основні можливості ARES включають розміщення компонентів, маршрутизацію шляхів, розміщення отворів, управління шарами, перевірку правил проектування та 3D-візуалізацію. Розміщення компонентів дозволяє нам

розташувати компоненти на платі, встановити їх розташування та орієнтацію. Маршрутизація шляхів автоматично створює шляхи з'єднання між компонентами згідно з електричними з'єднаннями зі схеми. Також можна вручну визначати шляхи маршрутизації.

Розміщення отворів дозволяє додавати точки для монтажу компонентів, таких як піни, резистори, конденсатори та інші. Управління шарами дозволяє використовувати багатошарові плати з різними шарами для маршрутизації сигналів, живлення та заземлення.

Завдяки 3D-візуалізації можна переглянути плату у тривимірному режимі, що дозволяє оцінити фізичний вигляд та розміщення компонентів. Це допомагає зрозуміти, як плата виглядатиме в реальному світі.

Після завершення проектування у ARES ми можемо експортувати готовий дизайн PCB для подальшого виготовлення фізичної плати. ARES є потужним інструментом для створення як простих, так і складних друкованих плат і допомагає розробникам забезпечити ефективну та надійну реалізацію своїх проектів.

У Proteus є вбудована бібліотека електронних компонентів, яка містить широкий спектр деталей, таких як мікроконтролери, сенсори, дисплеї, транзистори, резистори, конденсатори та багато іншого. З цієї бібліотеки можна вибирати необхідні компоненти та використовувати їх у проектах.

У Proteus можна створювати схеми, розміщувати компоненти на робочому полі, підключати їх разом за допомогою проводів та з'єднувачів, налаштовувати параметри компонентів та визначати спосіб взаємодії між ними.

Симуляція в Proteus дозволяє перевіряти роботу електронної схеми, вводити вхідні дані та спостерігати за результатами. За допомогою інструментів для аналізу сигналу та відлагодження можна перевірити правильність роботи схеми та виявити можливі проблеми.

Після успішної симуляції в Proteus можна перейти до проектування PCB у ARES. Ми можемо розміщувати компоненти на PCB, встановлювати їх

розташування та з'єднувати шляхами згідно з електричними з'єднаннями зі схеми.

Proteus надає можливість візуалізувати результати проекту, включаючи взаємодію з OLED дисплеєм та іншими компонентами. Ми можемо спостерігати виведені дані на OLED дисплей та переконатися, що вони відповідають очікуваним значенням ваги з тензодатчика.

Протеус також підтримує співпрацю з зовнішніми програмами для розробки електроніки. Наприклад, ми можемо імпортувати схеми із таких популярних програм, як Altium Designer, Eagle та KiCad, для подальшого моделювання та симуляції у Proteus. Це дає можливість співпрацювати з існуючими проектами та використовувати різноманітні інструменти, що надаються цими програмами. Proteus також підтримує використання мови програмування, такої як C або Assembler, для створення власних компонентів або розширення функціональності існуючих. Можна створювати власні моделі компонентів і визначати їх поведінку у симуляції, що дозволяє нам точно налаштовувати і адаптувати Proteus до нашого конкретного проекту.

Ще одна цікава можливість Proteus - імпорт та експорт даних у форматах, таких як Gerber, DXF, IDF тощо. Це дозволяє обмінюватися даними з іншими програмами для проектування друкованих плат або використовувати виведені дані для виробництва фізичних плат у спеціалізованих обладнаннях. Такий обмін даними дозволяє забезпечити безперервність робочого процесу від електронної схеми до фізичної реалізації.

Нарешті, Proteus має можливість роботи з багатьма користувачами через мережу. Завдяки цьому, декілька розробників можуть працювати над проектом одночасно, спільно редагувати схеми та РСВ, обмінюватися даними та співпрацювати над проектом в режимі реального часу. Це особливо корисно для команд, що працюють над великими проектами або розташовані на різних географічних місцях.

Використання Proteus дозволяє розробникам перевіряти та оптимізувати свої проекти до фізичної реалізації, що економить час та зусилля при розробці електроніки.

Також варто відзначити, що Proteus постійно оновлюється та покращується компанією Labcenter Electronics. В оновленнях можна очікувати наявності нових функцій, покращеної підтримки компонентів та інструментів, що дозволить використовувати передові технології у своїх проектах електроніки.

РОЗДІЛ 3. СТВОРЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Приклад використання модуля HX711:

Ваги, які використовують чип HX711, мають просту структуру, що складається з датчика, мікроконтролера (МК) та самого чипа. Логіка побудови коду та проекту в цьому випадку також нескладна - потрібно лише правильно послідовно підключити всі компоненти. Залежно від проекту, до перерахованих компонентів можна додати інтерфейс, який виводитиме отримані дані. Наприклад, модуль Bluetooth, який дозволяє під'єднати систему до смартфона і відображати всю інформацію безпосередньо на ньому. Якщо необхідно посилити сигнал, можна також підключити кілька чипів послідовно.

Датчики, які використовують цей чип, застосовуються на різних пристроях, від 3D-принтерів до спеціальних пристроїв для тестування продукції. Чіп видає дані про тиск, а з використанням відповідних формул у коді ці дані можна перетворити на кілограми ваги. Таким чином, якщо потрібно виміряти силу, можна застосувати відповідні алгоритми для перекладу одиниць, і в результаті ви отримаєте зовсім інший пристрій. Сфери застосування таких датчиків залежать від творчості інженера, але найпоширеніше їх використання спостерігається саме в 3D-принтерах.

Комбінування модуля HX711 з іншими компонентами та розробка відповідного коду дозволяють створювати різноманітні пристрої, які засновані на вимірюванні ваги і можуть бути використані у різних галузях.

3.2 Підключення модуля HX711 до Arduino

Тензодатчики мають мостову схему підключення, що передбачає використання чотирьох провідників з різним кольоровим маркуванням. Два провідники служать для подачі опорної напруги на два мостових плеча, а інші два провідники призначені для зняття вихідної напруги з інших двох плечей. Вихідна напруга потім підключається до вхідного підсилювача, такого як мікросхема HX711.

Для подальшої обробки і передачі інформації дані з HX711 підключаються до плати Arduino UNO. За допомогою проводів GND і VCC HX711 з'єднується з

відповідними точками GND і 5V на роз'ємі POWER плати Arduino UNO. Контакти DT і SCK підключаються до точок A1 і A0 на роз'ємі ANALOG IN. Таким чином, тензодатчик HX711 можна підключити до різних пристроїв, таких як рідкокристалічний дисплей LCD 1602 або комп'ютер, використовуючи USB-порт та стандартні бібліотеки Arduino.

Оскільки вихідна напруга вимірювального моста залежить від змінних значень, вона перетворюється на бінарний код за допомогою АЦП. Діапазон контрольованих напруг визначається коефіцієнтом посилення, який обирається. Наприклад, при коефіцієнті 128 діапазон вимірювання складає від -20 mV до +20 mV, при коефіцієнті 64 - від -40 mV до +40 mV, а при коефіцієнті 32 - від -80 mV до +80 mV. Важливо зазначити, що ці значення є точними лише при напрузі живлення +5V. У разі, якщо вхідна напруга виходить за межі діапазону, АЦП буде виходити за межі значень 800000h або 7FFFFFFh.

```
//код для калібрування
//код для калібрування
#include "HX711.h"

HX711 scale(A1, A0); // DT, CLK

float Calibration_Factor_Of_Load_cell = -3.7; // цей калібрувальний
коефіцієнт налаштовується відповідно до тензодатчика

float U;
float O;

void setup() {
    Serial.begin(9600);
    Serial.println("HX711 calibration sketch");
    Serial.println("Remove all weight from scale");
    Serial.println("After readings begin, place known weight on scale");
    Serial.println("Press + or a to increase calibration factor");
    Serial.println("Press - or z to decrease calibration factor");

    scale.set_scale();
    scale.tare(); //Сбросьте масштаб до 0
    long zero_factor = scale.read_average(); //Отримуєм базове читання
    Serial.print("Zero factor: "); //Це можна використовувати для усунення
    необхідності тарування шкали. Корисно у проектах постійного масштабу.
```

```

    Serial.println(zero_factor);
}

void loop() {

    scale.set_scale(Calibration_Factor_Of_Load_cell); //цей калібрувальний
коєфіцієнт редагується
    Serial.print("Reading: ");
    U = scale.get_units();
    if (U < 0)
    {
        U = 0.00;
    }
    O = U * 0.035274;
    Serial.print(O);
    Serial.print(" grams");
    Serial.print(" Calibration_Factor_Of_Load_cell: ");
    Serial.print(Calibration_Factor_Of_Load_cell);
    Serial.println();

    if (Serial.available())
    {
        char temp = Serial.read();
        if (temp == '+' || temp == 'a')
            Calibration_Factor_Of_Load_cell += 1;
        else if (temp == '-' || temp == 'z')
            Calibration_Factor_Of_Load_cell -= 1;
    }
}

```

Даний код є прикладом програми для підключення модуля HX711 до Arduino та його калібрування для вимірювання ваги. Основна функція цієї програми полягає у зчитуванні ваги зі зважувального сенсора, обробці цих даних і виведенні результату на монітор послідовного порту.

Основні кроки, які виконує програма:

- Підключення необхідної бібліотеки: `#include "HX711.h"`. Це бібліотека, яка надає функціональність для взаємодії з модулем HX711.

- Ініціалізація об'єкта `scale` класу HX711 з використанням пінів Arduino, підключених до модуля HX711: `HX711 scale(A1, A0);`. A1 відповідає за передачу даних (DT), а A0 - за синхронізацію (CLK).

- Встановлення коефіцієнта калібрування `Calibration_Factor_Of_Load_cell`. Цей коефіцієнт дозволяє коректно перетворювати значення, отримані від сенсора, на вагу. Значення цього коефіцієнта можна змінювати в процесі калібрування.

- У функції `setup()` налаштовуються початкові параметри. Виводяться деякі вказівки на монітор послідовного порту, ініціалізується з'єднання з монітором.

- У функції `loop()` виконується основний цикл програми. Спочатку встановлюється поточне значення калібрувального коефіцієнта: `scale.set_scale(Calibration_Factor_Of_Load_cell);`.

- Далі зчитується вага за допомогою `U = scale.get_units();`. Значення отриманої ваги перетворюється у грами за формулою `O = U * 0.035274;`. Ці значення виводяться на монітор послідовного порту.

- Програма також читає команди з монітора послідовного порту. Якщо отримано символ '+' або 'a', то калібрувальний коефіцієнт збільшується на 1. Якщо отримано символ '-' або 'z', то калібрувальний коефіцієнт зменшується на 1, це дозволяє користувачу вручну налаштовувати калібрування сенсора.

3.3 Підключення тензодатчика до Arduino

Для виконання цього завдання потрібно мати наступні компоненти:

- Мікроконтролер Arduino Uno / Arduino Nano / Arduino Mega.
- Модуль тензодатчика HX711.
- Проводи з "тато-тато" та "тато-мама" роз'ємами.
- Паяльник.

Для створення ваг з тензодатчиком необхідно придумати та скласти конструкцію, а також мати під рукою ваги для порівняння даних з модулем. Модуль HX711 має отвори з різьбленням для зручного монтажу.

Після зібрання ваги з Arduino необхідно виконати калібрування тензодатчика. Підключіть датчик до модуля HX711 за схемою, яка була наведена вище, а потім підключіть модуль до плати Arduino. Порти DT і SCK можна підключити до будь-яких цифрових портів мікроконтролера (у прикладі коду використовувалися порти 2 і 3). Після збирання схеми завантажується код для калібрування датчика на плату Arduino.

Скетч для калібровки тензодатчика HX711:

```
#include <HX711.h>

HX711 scale(2, 3); // порти DT, CLK

float calibration_factor = -3.7; // калібрування датчика
float units;
float ounces;

void setup() {
    Serial.begin(9600);
    Serial.println("HX711 calibration sketch");
    Serial.println("Remove all weight from scale");
    Serial.println("Press + or a to increase calibration factor");
    Serial.println("Press - or z to decrease calibration factor");

    scale.set_scale();
    scale.tare(); // скиднемо датчик на 0

    long zero_factor = scale.read_average();
    Serial.print("Zero factor: ");
    Serial.println(zero_factor);
}

void loop() {
    scale.set_scale(calibration_factor); // застосовуємо калібрування

    Serial.print("Reading: ");
    units = scale.get_units(), 10; // Виконуємо 10 раз заміри
    if (units < 0)
    {
        units = 0.00;
    }
    ounces = units * 0.035274; // переводим середне значення в грами
```

```

Serial.print(ounces);
Serial.print(" grams");
Serial.print(" calibration_factor: ");
Serial.print(calibration_factor);
Serial.println();

if (Serial.available())
{
    char temp = Serial.read();
    if (temp == '+' || temp == 'a')
        calibration_factor += 1;
    else if (temp == '-' || temp == 'z')
        calibration_factor -= 1;
}
}

```

Пояснення до коду:

Результат поправочного коефіцієнта, який буде виведений на монітор порта, слід використовувати для подальшої роботи з ваговим датчиком.

Якщо потрібно змінити коефіцієнт, ми можемо використовувати монітор порта для надсилання символу "+" або "-" до плати Arduino Uno.

Даний код є скетчем для підключення тензодатчика HX711 до Arduino та його калібрування для вимірювання ваги. Основна мета цього коду - зчитувати вагу з тензодатчика, калібрувати його та виводити результат на монітор послідовного порту.

Основні кроки, які виконує програма:

- Підключається необхідна бібліотека: `#include <HX711.h>`. Це бібліотека, яка забезпечує функціональність для взаємодії з модулем HX711.
- Ініціалізується об'єкт `scale` класу HX711 з використанням пінів Arduino, на які підключений модуль HX711: `HX711 scale(2, 3);`. Пін 2 (DT) використовується для передачі даних, а пін 3 (CLK) - для синхронізації.
- Встановлюється коефіцієнт калібрування `calibration_factor`. Цей коефіцієнт дозволяє коректно перетворювати значення, отримані від тензодатчика, на вагу. Його значення можна змінювати в процесі калібрування.

- У функції `setup()` виконується налаштування початкових параметрів. На монітор послідовного порту виводяться деякі інструкції, ініціалізується з'єднання з монітором.

- У функції `loop()` виконується основний цикл програми. Спочатку встановлюється поточне значення калібрувального коефіцієнта: `scale.set_scale(calibration_factor);`.

- Здійснюється зчитування ваги за допомогою `units = scale.get_units(), 10;`. Це виконується 10 разів для отримання середнього значення. Якщо отримане значення ваги менше 0, то йому присвоюється значення 0.

- Значення ваги перетворюється у грами за допомогою формули `ounces = units * 0.035274;`. Отримане значення виводиться на монітор послідовного порту.

- Програма також зчитує наявність доступних даних з монітора послідовного порту. Якщо отримано символ '+' або 'a', то калібрувальний коефіцієнт збільшується на 1. Якщо отримано символ '-' або 'z', то калібрувальний коефіцієнт зменшується на 1. Це дає можливість користувачу вручну налаштовувати калібрування тензодатчика.

Цей код дозволяє підключити та калібрувати тензодатчик HX711 з використанням Arduino, що дозволяє вимірювати точну вагу.

Скетч для тензодатчика (hx711) Arduino

```
#include <HX711.h>
#include <LiquidCrystal.h>

HX711 scale(2, 3); // порти DT, CLK
LiquidCrystal lcd(12, 11, 5, 4, 3, 2); // піни RS, E, D4, D5, D6, D7
(залежить від вашого підключення)

float calibration_factor = -3.7; // калібрування датчика
float units;
float ounces;

void setup() {
    Serial.begin(9600);
    lcd.begin(16, 2); // Ініціалізуємо LCD дисплей з розмірами 16x2
```

```

    scale.set_scale();
    scale.tare(); // скидуєм датчик на 0
    scale.set_scale(calibration_factor); // калібрування
}

void loop() {
    Serial.print("Reading: ");

    for(int i=0; i<10; i++) units += scale.get_units(), 10; // виконуєм
заміри 10 раз
    units / 10; // робим значення на 10
    ounces = units * 0.035274; //переводим значення в грами

    Serial.print(ounces);
    Serial.print(" grams");
    Serial.println();

    // Виводимо дані на LCD дисплей
    lcd.setCursor(0, 0);
    lcd.print("Weight: ");
    lcd.print(ounces);
    lcd.print(" g");
}

```

За допомогою поправочного коефіцієнта з першої програми, ваги будуть показувати точні значення в грамах.

Цей код використовує Arduino та дві бібліотеки - HX711 та LiquidCrystal - для вимірювання ваги за допомогою тензодатчика HX711 та виводу результатів на LCD дисплей.

Основна структура коду:

- Необхідні бібліотеки:

```

#include <HX711.h>
#include <LiquidCrystal.h>

```

Бібліотека 'HX711' використовується для взаємодії з тензодатчиком HX711, а 'LiquidCrystal' - для керування LCD дисплеєм.

- Ініціалізація об'єктів:

```

HX711 scale(2, 3); // порти DT, CLK
LiquidCrystal lcd(12, 11, 5, 4, 3, 2); // піни RS, E, D4, D5, D6, D7

```

``scale`` - об'єкт типу HX711, який використовується для взаємодії з тензодатчиком. Піни ``2`` та ``3`` вказують на піни мікроконтролера, до яких підключені піни ``DT`` та ``CLK`` тензодатчика.

``lcd`` - об'єкт типу LiquidCrystal для управління LCD дисплеєм. Піни ``12``, ``11``, ``5``, ``4``, ``3``, ``2`` вказують на піни мікроконтролера, до яких підключені піни ``RS``, ``E``, ``D4``, ``D5``, ``D6``, ``D7`` дисплея.

- Налаштування:

```
void setup() {
  Serial.begin(9600);
  lcd.begin(16, 2);

  scale.set_scale();
  scale.tare();
  scale.set_scale(calibration_factor);
}
```

У функції ``setup()`` налаштовуються потрібні параметри. В цьому випадку:

- Встановлюється швидкість передачі даних через Serial порт (``Serial.begin(9600)``).

- Ініціалізується LCD дисплей, вказуючи розміри (16x2) (``lcd.begin(16,2)``).

- Калібрування тензодатчика. ``set_scale()`` встановлює значення масштабу, ``tare()`` встановлює поточне значення як нульове, а ``set_scale(calibration_factor)`` використовується для калібрування шкали вимірювання.

- Основний цикл програми:

```
void loop() {
  Serial.print("Reading: ");

  for(int i=0; i<10; i++)
    units += scale.get_units(), 10;
  units / 10;
  ounces = units * 0.035274;

  Serial.print(ounces);
}
```

```

Serial.print(" grams");
Serial.println();

lcd.setCursor(0, 0);
lcd.print("Weight: ");
lcd.print(ounces);
lcd.print(" g");
}

```

В цьому циклі виконуються наступні дії:

- Здійснюється серія вимірювань ваги за допомогою `scale.get_units()`.
- Цикл `for` повторюється 10 разів, і кожне вимірювання додається до змінної `units` (`units += scale.get_units(), 10`).
- Значення `units` ділиться на 10 для отримання середнього значення.
 - Конвертація ваги в грами (`ounces = units * 0.035274`).
 - Результати виводяться через Serial порт (`Serial.print()`) і на LCD дисплей (`lcd.print()`).

Таким чином, код зчитує дані з тензодатчика HX711, обробляє їх та виводить на Serial порт та LCD дисплей.

Даний код виконується безперервно в функції `loop()`, що дозволяє постійно отримувати та обробляти дані з тензодатчика ваги.

У початку циклу виконується виведення тексту "Reading: " через Serial порт за допомогою `Serial.print()`.

Далі, виконується цикл `for`, який повторюється 10 разів. Кожен раз вимірюється вага за допомогою функції `scale.get_units()`, і отримане значення додається до змінної `units`. Значення 10 у рядку `units += scale.get_units(), 10` виглядає неправильно. Ймовірно, правильний код повинен мати `units += scale.get_units()`. Таким чином, цей цикл накопичує 10 вимірів в змінній `units`.

Після циклу `for`, значення змінної `units` ділиться на 10, щоб отримати середнє значення. Однак, у коді цей рядок `units / 10;` не має жодного ефекту, оскільки результат не присвоюється жодній змінній.

Далі, змінна `units` множиться на коефіцієнт 0.035274 для конвертації ваги в грами, і результат зберігається в змінній `ounces`.

Після цього результати виводяться на Serial порт за допомогою функцій ``Serial.print()`` і ``Serial.println()``. Таким чином, на Serial порті виводяться значення ваги у грамах.

Далі, на LCD дисплей виводиться текст "Weight: " та значення ваги у грамах за допомогою функцій ``lcd.setCursor()``, ``lcd.print()``. Таким чином, на LCD дисплеї виводиться текст "Weight: " та виміряна вага в грамах.

Цикл ``loop()`` повторюється безперервно, забезпечуючи постійне вимірювання ваги, оновлення виводу на Serial порті та LCD дисплеї.

3.4 Підключення OLED дисплея до плати Arduino

Щоб вивести показники ваги тензодатчика на OLED дисплей, нам потрібно буде зчитувати значення АЦП модуля HX711 та перетворити його на вагу. Нижче наведено деталі коду з використанням бібліотеки HX711 які зчитують дані з модуля та виводять їх на OLED дисплей.

Ми будемо використовувати тензодатчик з обмеженою вагою до 5 кілограмів, нам потрібно змінити значення калібрування (CALIBRATION_FACTOR) відповідно. Це значення використовується для перетворення вимірювань, отриманих з тензодатчика, на вагу у вибрані одиниці (у даному випадку, в кілограми).

Наведений код призначений для підключення OLED дисплея до плати Arduino Uno та вимірювання ваги за допомогою тензодатчика та модуля HX711. Давайте розглянемо його детально:

У коді використовуються наступні бібліотеки:

HCuOLED.h: Бібліотека для керування OLED дисплеєм, Ця бібліотека надає зручні функції для виводу тексту, графіки та керування дисплеєм

SPI.h: Бібліотека для взаємодії з пристроями по шині SPI.

SPI.HX711.h: Бібліотека для роботи з модулем HX711.

Потім Оголошуються піни:

```
#define CS_DI 10
#define DC_DI 9
#define RST_DI 8
```

У цьому рядку ми визначаємо піни Arduino, до яких підключені піни Chip Select (CS), Data/Command (DC) та Reset (RST) OLED дисплея.

Налаштування параметрів дисплея:

```
HCuOLED HCuOLED(SH1106, CS_DI, DC_DI, RST_DI);
```

За допомогою цього рядка коду створюється об'єкт HCuOLED для керування OLED дисплеєм. В конструктор передаються параметри, такі як тип дисплея (SH1106), піни CS, DC та RST.

Метод setup():

```
void setup()
```

```
{
    HCuOLED.Reset();
}
```

Як нам уже відомо цей метод викликається один раз під час початку роботи мікроконтролера Arduino. У цьому випадку викликається функція Reset() для ініціалізації OLED дисплея.

Метод loop():

```
void loop()
{
    float weight = scale.get_units();

    HCuOLED.ClearDisplay();
    HCuOLED.Cursor(0, 0);
    HCuOLED.SetFont(MedProp_11pt);
    HCuOLED.Print("Weight: ");
    HCuOLED.Print(weight / 1000.0, 2);
    HCuOLED.Print(" kg");
    HCuOLED.Refresh();

    delay(100);
}
```

Це головний нескінченний цикл програми, який виконується постійно після функції setup(). У цьому циклі вимірюється значення ваги з тензодатчика за допомогою функції get_units() та зберігається у змінній weight. Потім OLED дисплей очищується за допомогою функції ClearDisplay(). Наступні рядки встановлюють позицію курсора (Cursor()), вибирають шрифт (SetFont()) та

виводять текст на дисплей (Print()). Значення ваги перетворюється на кілограми, розділені з двома знаками після коми. Після виводу на дисплей, викликається функція Refresh(), щоб оновити вміст дисплея. Задана максимальна вага в цьому коді обмежена до 5 кілограмів. Якщо значення ваги перевищує це обмеження, воно автоматично встановлюється на максимальне значення (5000 грам). На останок, за допомогою delay(100) вставляється затримка 100 мілісекунд для стабілізації вимірювання перед наступною ітерацією.

Функція калібрування та налаштування HX711:

```
scale.begin(HX711_DOUT_PIN, HX711_CLK_PIN);
scale.set_scale(CALIBRATION_FACTOR);
scale.tare();
```

Ці рядки коду виконуються у функції setup(). Вони ініціалізують модуль HX711, встановлюють значення калібрування (CALIBRATION_FACTOR) та обнуляють початкове значення (tare). Загалом, цей код дозволяє підключити OLED дисплей до плати Arduino Uno, виміряти вагу за допомогою тензодатчика та вивести це значення на дисплей.

Повний код представлений у додатку A2.

Зберігання інформації на SD-карту

Ще ми можемо розглянути варіант підключення SD-карти до Arduino для збереження ваги на фізичному носії. SD-карта, яка використовується у проекті, підключається за допомогою SD-адаптера, що спрощує з'єднання з Arduino. Для забезпечення зчитування та запису даних на SD-карту була використана бібліотека SD, яка надає необхідні функції для роботи з SD-картою.

Давайте розберемо що використовувалося для написання коду:

- Підключення бібліотек:

- `SD.h`: Ця бібліотека дозволяє взаємодіяти з SD-картою. Вона надає функції для ініціалізації SD-карти, відкриття/створення файлів, запису та читання даних.

- `Wire.h`: Ця бібліотека потрібна для роботи з тензодатчиком HX711 через I2C, але в даному конкретному коді не використовується напряму.

- ``HX711.h``: Ця бібліотека використовується для отримання даних з тензодатчика HX711. Вона містить функції для зчитування та калібрування ваги.

- Для підключення SD-карти до Arduino використовуються наступні піни:

- ``LOADCELL_DOUT_PIN``: Це пін DOUT тензодатчика, до якого підключений вихід даних (DOUT).

- ``LOADCELL_SCK_PIN``: Це пін SCK тензодатчика, до якого підключений вхід тактових сигналів (SCK).

- ``SD_CHIP_SELECT_PIN``: Це пін Chip Select (CS) адаптера SD-карти, який використовується для взаємодії з SD-картою.

- Глобальні змінні:

- ``dataFile``: Об'єкт типу ``File``, який використовується для зберігання файлу на SD-карті.

- Метод ``setup()``:

- Ініціалізація Serial-порту для виведення повідомлень.

- Ініціалізація SD-карти за допомогою функції ``SD.begin()``. Якщо ініціалізація не вдалася, виводиться повідомлення з помилкою.

- Ініціалізація тензодатчика HX711 за допомогою функції ``scale.begin()``.

- Калібрування тензодатчика ваги за допомогою функції ``scale.set_scale()`` та ``scale.tare()``.

- Відкриття файлу "data.txt" на SD-карті за допомогою функції ``SD.open()``. Якщо відкриття файлу не вдалося, виводиться повідомлення з помилкою.

- Основний цикл ``loop()``:

- Зчитування ваги з тензодатчика за допомогою функції ``scale.get_units()``. Отримана вага зберігається у змінну ``weight``.

- Виведення ваги на Serial Monitor за допомогою функцій ``Serial.print()`` та ``Serial.println()``.

- Запис ваги у файл на SD-карті за допомогою функцій ``dataFile.print()`` та ``dataFile.println()``.

- Затримка 1 секунда перед наступним зчитуванням ваги.

Цей код ініціалізує SD-карту та тензодатчик HX711, калібрує тензодатчик, зчитує вагу, виводить її на Serial Monitor та зберігає в файлі на SD-карті. Ви можете змінити назву файлу або внести інші зміни, які вам потрібні, щоб відповідати вашим вимогам.

За допомогою вбудованої бібліотеки SD ми можемо здійснювати операції зчитування та запису даних на SD-карту. У нашому випадку, ми зберігаємо виміри ваги, отримані з тензодатчика, у файлі з назвою "data.txt". Важливо звернути увагу, що SD-карта повинна бути вірно відформатована у форматі FAT16 або FAT32 перед використанням.

Таким чином, підключення SD-карти до Arduino дозволяє зберігати виміри ваги на зовнішньому носії для зручного зберігання та подальшого аналізу даних. Повний код наведено у додатку A3.

ВИСНОВОК

В ході кваліфікаційної роботи було визначено, що застосування вагових дозаторів у виробництві є важливою складовою для забезпечення високої якості продукції та оптимізації виробничих процесів. Однак, існують проблеми пов'язані зі старіючими або недостатньо ефективними інформаційними системами керування ваговими дозаторами.

Метою даної роботи було розробити інформаційну комп'ютеризовану підсистему для вагового дозатора, яка забезпечує автоматизоване керування, моніторинг та інтеграцію з іншими процесами виробництва. Для досягнення цієї мети було вирішено завдання систематизації та розширення теоретичних і практичних знань з обраного напрямку підготовки, розвитку навичок самостійної роботи та підготовки до самостійної практичної діяльності.

Об'єктом дослідження були інформаційні процеси в сфері використання вагових дозаторів та їх автоматизації, а предметом дослідження - інструментальні засоби, такі як математичні моделі, методи та інформаційні технології. Для досягнення поставленої мети було використано методи аналізу наукових джерел, вивчення існуючих систем керування, експериментальних вимірювань та тестувань, моделювання, програмування та реалізації комп'ютерної підсистеми.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Бурмістенков О. П. Електромеханічні пристрої (теорія і практика) / О.П. Бурмістенков, І. В Петко., М. Й. Бондаренко, Б. М.Злотенко. - Київ: КНУТД, 2017. – 260 с.
2. Вагові дозатори. Види. Характеристики [Електронний ресурс] - Режим доступу до ресурсу: <https://keli.com.ua/vagovi-dozatori-vidi-harakteristiki/>.
3. Грицюк Ю.І., Рак Т.Є., Програмування мовою С++ навчальний посібник, Львів, 2011.
4. Колонтаєвський Ю. П., Сосков А. Г. Промислова електроніка та мікросхемотехніка: теорія і практикум: Навч. посіб. /За ред.. А. Г.Соскова. 2-ге вид. –К.: Каравела, 2004. – 432 с .
5. М. Матвієноко – Основи електротехніки, 2016. 226 с.
6. Новацький Н.О. - Мікропроцесорні та мікроконтролерні системи, Київ, 2018
7. Попович М. Г., Лозинський О. Ю., Мацько Б. М. та ін. Електромеханічні системи автоматичного керування та електроприводи: навч. посіб./ Під ред. М. Г. Поповича – К.: Либідь, 2013. – 680 с.
8. Програмування на Arduino [Електронний ресурс] - Режим доступу до ресурсу: <https://doc.arduino.ua/ru/prog/>
9. Розробка та дослідження електромеханічних систем та складових електроприводів. Навчальний посібник./ Попович М. Г., Кострицький В. В., Артеменко, Л. Ф., Бурмістенков О. П. та ін. – К.: КНУТД, 2011. – 492 с.
10. Технічна енциклопедія TechTrend [Електронний ресурс] - Режим доступу до ресурсу: <https://techtrend.com.ua/index.php?newsid=9421> .
11. Що таке тензодатчик і як він працює [Електронний ресурс]. – 2019. - Режим доступу до ресурсу: <https://obrii.com.ua/main/19502-shho-take-tenzodatchik-i-yak-vin-pratsyuje.html> .
12. Bjarne Stroustrup - Programming: Principles and Practice Using C++, 2014. – 1236 с.
13. Fundamentals of Device and Systems Packaging: Technologies and Applications, Second Edition 2nd Edition by Rao Tummala (Author)., 2019. — p.828.

14. Simon Monk - Programming Arduino: Getting Started with Sketches.
15. Simon Monk - Programming Arduino: Getting Started with Sketches, Second Edition.