МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
УМАНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ САДІВНИЦТВА
Факультет економіки і підприємництва
Кафедра інформаційних технологій

Допущено до захисту
"_____" _____ 2023 р.
Зав. кафедри, к.т.н., доцент
_____ **Ліщук Р.І.**

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

Розробка інформаційної підсистеми для групової доїльної установки

Виконав – студент 4 курсу, групи 41-кн
освітньої програми "Комп'ютерні науки"
спеціальності 122 Комп'ютерні науки
Бондаренко Олег Андрійович
Науковий керівник – д.т.н., професор
Кулаков Павло Ігорович

Умань – 2023

Зміст

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ..... 7

ВСТУП……..8

РОЗДІЛ 1. АНАЛІТИЧНИЙ ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ..... 10

1.1. Об’єкт та предмет дослідження 10

1.2. Особливості отримання інформації про технологічні параметри на груповій доїльній установці..... 14

1.3. Постановка задачі..... 21

РОЗДІЛ 2. ВИБІР ТЕХНОЛОГІЙ ТА АРХІТЕКТУРА ПІДСИСТЕМИ 22

2.1. Обґрунтування вибору мови програмування для створення програмного забезпечення для персонального комп’ютера підсистеми 22

2.2. Технологія розробки JumpStarter C for AVR 27

2.3. Розробка архітектури інформаційної підсистеми для групової доїльної установки 34

РОЗДІЛ 3. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ІНФОРМАЦІЙНОЇ ПІДСИСТЕМИ ……...37

3.1. Розробка програмного забезпечення інформаційної підсистеми для групової доїльної установки 37

3.2. Розробка програмного забезпечення для мікроконтролера блоку забезпечення обміну даними 42

ВИСНОВОК..... 44

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ 45

ДОДАТОК А..... 47

ДОДАТОК Б 54

Перелік умовних позначень, скорочень і термінів

ПК	Персональний комп’ютер
БЗПД	Блок забезпечення процесу доїння
БЗОД	Блок забезпечення обміну даними
ЗВКМ	Засіб вимірювання кількості молока
ЗТ	Зчитувач транспондерів
БКЗТ	Блок керування зчитувачем транспондерів
БКПТ	Блок контролю проходження тварини
БУДМ	Блок управління дозатором молока
ЛПМ	Лічильник порцій молока
ЗКПОО	Засіб контролю параметрів охолоджувального обладнання
ЗКПСМ	Засіб контролю параметрів сирого молока
ЗКПВО	Засіб контролю параметрів вакуумного обладнання
БД	База даних
ПЗ	Програмне забезпечення.
ПД	Програмний додаток.
ПЗП	Постійний запам’ятовуючий пристрій
ІС	Інформаційна система
ООП	Об’єктно-орієнтоване програмування

Вступ

Молочне тваринництво займає особливу роль в сільськогосподарській галузі, молочна продукція задовольняє значну частину потреб населення багатьох країн в продуктах харчування. Виробництво молока базується на великій кількості взаємопов'язаних технологічних процесів, які в цілому формують систему виробництва продукції тваринництва. Більшість українських тваринницьких ферм мають низький рівень інформатизації та автоматизації, що безпосередньо впливає на якість молочної сировини та збільшує її собівартість. Внаслідок цього факту, ступінь ефективності продуктивності праці, методів вирощування, годівлі, обліку та доїння далека від максимального значення. Широко використовується застаріле обладнання, в багатьох випадках наявний неналежний рівень організації і технічного забезпечення тваринницьких ферм.

Виробництво молока являє собою біотехнічну систему, до складу якої входять машини, технології, тварини та оператори [1-3]. Сучасні ферми, розраховані на велику кількість тварин та їх доїльні установки потребують впровадження рішень, які надають можливість оптимально управляти стадом. Системи автоматичного управління (САУ) молочною фермою забезпечують управління технологічним процесом виробництва молока, оптимізацію продуктивності, здоров'я та репродуктивності тварин, годівлі, ефективності роботи персоналу [4-6]. Важливою складовою таких САУ є мікроконтролерні засоби управління технологічними процесами, які є невід’ємною складовою процесу виробництва молока. Особливо важливу роль має інформаційні системи (ІС) та інформаційні підсистеми, які входять до їх складу [7].

Актуальність

Впровадження інформаційних технологій та відповідних систем, як слідує з світового досвіду, призводить до збільшення продуктивності праці, зменшення завантаження працівників, збільшує ефективність технологічного процесу виробництва молока. Таким чином, розвиток теорії та практичного

аспекту впровадження засобів інформатизації у технологічний процес виробництва молока є актуальним завданням.

Об'єкт дослідження – процес, концепція та архітектура побудови інформаційних підсистем для групової доїльної установки.

Предмет дослідження – методи інформатизації молочних ферм обладнаних груповою доїльною установкою.

Мета дослідження

Метою та завданням бакалаврської кваліфікаційної роботи є концептуальна розробка інформаційної підсистеми параметрів технологічного процесу виробництва молока на груповій доїльній установці, архітектури інформаційної підсистеми та її компонентів, розробка програмного забезпечення для інформаційної підсистеми та програмного забезпечення для мікроконтролерів, під управлінням яких працюють окремі компоненти підсистеми.

Протягом виконання кваліфікаційної бакалаврської роботи необхідно провести аналіз наукової та технічної літератури, технічної документації, міжнародних стандартів у галузі інформатизації та автоматизації тваринницьких ферм, розробити архітектуру інформаційної підсистеми для групової доїльної установки, структурні схеми її окремих компонентів та програмних модулів.

РОЗДІЛ 1. АНАЛІТИЧНИЙ ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Об’єкт та предмет дослідження

У технологічному процесі виробництва молока технічні ланки вступають у пряму взаємодію з біологічними об’єктами та утворюють біотехнічну систему «людина-тварина-машина» [8]. Схему взаємодії складових елементів технологічного процесу виробництва молока наведено на рис. 1.1.

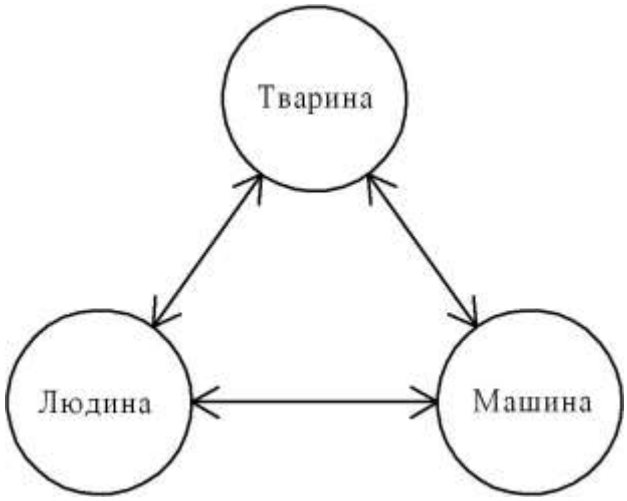


Рисунок 1.1 – Схема взаємодії складових елементів технологічного процесу виробництва молока

Основне завдання, яке вирішується при виробництві молока, є забезпечення якісного видоювання та високої продуктивності тварин протягом усього періоду експлуатації. До складових елементів технологічного процесу виробництва молока входять умови утримання тварин, їх індивідуальні особливості, конструкція і режими роботи обладнання, кваліфікація операторів машинного доїння, а також умови їх праці. Ефективність технологічного процесу виробництва молока залежить від відповідності його складових фізіологічним потребам тварин і якості взаємозв’язків між складовими елементами. Внаслідок взаємодії складових елементів, вихідні параметри технологічного процесу виробництва молока в значній мірі характеризують стан кожного з них.

Також вихідні параметри кожного складового компоненту характеризують стан інших компонентів, що входять до складу технологічного процесу. Досвід виробництва молока показує, що в однакових умовах, при використанні одних і тих самих установок і тварин, різні оператори доїння одержують різні показники продуктивності, якості молока і захворюваності тварин. Таким чином, ефективність систем доїння залежить від відношення дояра до своїх обов'язків. Особливо роль відіграє людина в застарілих системах доїння, де на оператора покладається виконання трудомістких технологічних операцій, пов'язаних з підготовкою тварин, підключенням апаратів, проведення додоювання, зніманням апаратів. Важливий вплив на характеристики роботи оператора доїння мають комплексні показники ергономічності установки [9].

В технологічному процесі виробництва молока на формування рефлексу молоковіддачі та захворюваність маститом впливають типи доїльних апаратів, режими їх роботи, тип установки, спосіб утримання, технологія і методика доїння. Відсутність у складі установки систем підготовки до доїння та засобів визначення параметрів молоковіддачі призводить до порушень технологічного процесу, погіршення секреторної функції молочної залози, високого рівня захворюваності, зниження якості молока і погіршення санітарного стану, збільшення бактеріальної заплідненості молока. Ефективність технологічного процесу виробництва молока, окрім технологічного та людського факторів, залежить від індивідуальних характеристик тварин, зокрема від їх придатності до машинного доїння. Наявність у стаді невеликої кількості непридатних до машинного доїння тварин призводить до порушень регламенту технологічного процесу доїння. Тому для забезпечення ефективності технологічного процесу виробництва молока необхідна селекційна робота з метою створення однорідних стад з великими адаптаційними можливостями і придатних до машинного доїння.

Практика виробництва молока свідчить, що існуючі системи не завжди забезпечують якісний технологічний процес і взаємозв'язок між його

складовими компонентами, є недостатньо ефективними і потребують удосконалення. Збільшення ефективності тваринницьких ферм потребує оптимізації взаємодії складових «людина-тварина-машина» у різних системах доїння, удосконалення технології та обладнання.

Оптимізація взаємодії машини, людини та тварини, створення доїльного апарата, який забезпечує ефективний рефлекс молоковіддачі та підвищення продуктивності тварин, є складним завданням комплексного характеру, яке в теперішній час повністю не вирішено. Для стимуляції рефлексу молоковіддачі, під час підготовки необхідний масаж, тому як нервові закінчення закладені глибоко в шкірі, якість цієї операції залежить виключно від оператора. В процесі доїння дійкова гума механічно діє на рецептори молочної залози, які знаходяться глибоко у тканинах. Внаслідок подразнення рецепторів виникає збудження, яке по волокнах нерва досягає спинного мозку. В спинному мозку нервовий сигнал розділяється на два сигнали. Один з сигналів по короткій дузі повертається до м'язових елементів залози. В результаті відбувається розслаблення цистерни і молоко виводиться з цистерни та вивідних протоків. Це перша фаза молоковіддачі, яка виникає через 2 - 6 с після впливу та продовжується 25 - 30 с. По другій дузі нервові імпульси передаються до гіпофізу, який забезпечує виділення гормону окситоцину в кров. Окситоцин, досягаючи залози, викликає скорочення альвеол. Таким чином, протягом другої фази виникає виведення молока з альвеол. Час від початку подразнення до припуску молока, який називається латентним періодом, складає 40 - 50 с. Необхідна для молоковіддачі концентрація окситоцину в крові підтримується від 2 до 5 хвилин. Руйнація окситоцину супроводжується закінченням молоковіддачі. Вищенаведене явище є фізіологічною основою правила - процес доїння повинен відбуватися так, щоб максимально використати дію рефлексу молоковіддачі. Високий рівень тиску у молочній залозі утримується протягом 1,5 - 2 хвилини, після чого починає зменшуватися, лактаційний рефлекс складає приблизно 4 - 5 хвилин. При нормальних умовах доїння, тиск

починає знижуватися після того, як відбудеться видоювання 50 % молока, через 4 - 5 хвилин корова повинна віддати не менше 80 - 90 % молока, після чого починається третя фаза доїння, під час якої здійснюється додоювання.

При безприв'язному утриманні тварин реалізується дві концепції: «людина вирішує, коли тварині доїтися, тварина вирішує, коли їй харчуватися» та «тварина вирішує, коли їй доїтися і коли харчуватися». При організації доїльно–молочних відділень тваринницьких ферм згідно з першою концепцією використовуються групові доїльні установки, конвеєрні доїльні установки, доїльні установки з прохідними станками. При організації доїльно–молочних відділень згідно з другої концепцією використовуються доїльні роботи.

У теперішній час найбільш розповсюдженими груповими доїльними установками є установки «Ялинка» та «Паралель». Алгоритм роботи цих установок повністю збігається, різниця полягає в різному розташуванні тварин. На установці «Паралель» тварини розташовані паралельно в доїльних станках та перпендикулярно робочій зоні дояра, на установці «Ялинка» – паралельно у доїльних станках та під певним кутом до робочої зони дояра.

Доїльні станки, які входять до складу групової доїльної установки, знаходяться по обидві сторони робочої зони доярів. Кожну сторону обслуговує один дояр, кількість станків, які обслуговуються одним доярем може дорівнювати від 4 до 12, з обох сторін технологічні операції виконуються одночасно і аналогічно. Протягом певного часу в доїльні станки однієї робочої зони входить кількість тварин, яка дорівнює кількості станків. Після цього дояр готує першу тварину до доїння протягом часу, далі починається процес її доїння, а дояр переходить до наступної тварини і виконує аналогічні операції. Після того, як закінчиться доїння усіх тварин у станках, вони виходять з установки. Далі виконується доїння наступної групи тварин. Цей процес продовжується до тих пір, поки не буде завершено доїння усіх тварин в стаді. На доїльних установках типу «Ялинка» та «Паралель» можливе використання

двох типів доїльних апаратів: з функцією керування процесом доїння та без функції керування процесом доїння.

1.2 Особливості отримання інформації про технологічні параметри на груповій доїльній установці

Переважна більшість інформації про параметри технологічного процесу виробництва молока отримується за допомогою відповідних інформаційних підсистем, які входять до складу інформаційної системи тваринницької ферми. Основним типом інформаційних підсистем, які використовуються у доїльно-молочних відділеннях тваринницьких ферм, є інформаційні підсистеми параметрів технологічного процесу виробництва молока. Вони призначені для отримання інформації про параметри технологічних процесів, необхідного перетворення та обробки цієї інформації, представлення її у необхідному вигляді, здійснення функцій контролю параметрів технологічного процесу. Розглянемо наведену на рис. 1.2 узагальнену структурну схему інформаційної підсистеми параметрів технологічного процесу виробництва молока, до складу якої входять практично усі структурні елементи, використання яких можливе у таких підсистемах [10]. До складу узагальненої інформаційної підсистеми входять блоки забезпечення процесу доїння (БЗПД), блок забезпечення обміну даними (БЗОД), зчитувачі транспондерів (ЗТ), блоки керування зчитувачем транспондерів (БКЗТ), блоки контролю проходження тварини (БКПТ), блоки керування селекційними воротами (БКСВ), блоки керування чесальною установкою (БКЧУ), блоки керування ваговими платформами (БКВП), ЗВ кількості молока (ЗВКМ), блоки управління дозатором молока (БУДМ), лічильники порцій молока (ЛПМ), блоки відображення параметрів (БВП), засоби контролю параметрів сирого молока (ЗКПСМ), засоби контролю параметрів охолоджувального обладнання (ЗКПОО), засоби контролю параметрів вакуумного обладнання (ЗКПВО), засоби контролю параметрів гноєзбирального обладнання (ЗКПГО), засоби контролю параметрів доїльного

обладнання (ЗКПДО), сервер інформаційної підсистеми, за допомогою якого забезпечується функціонування підсистеми.

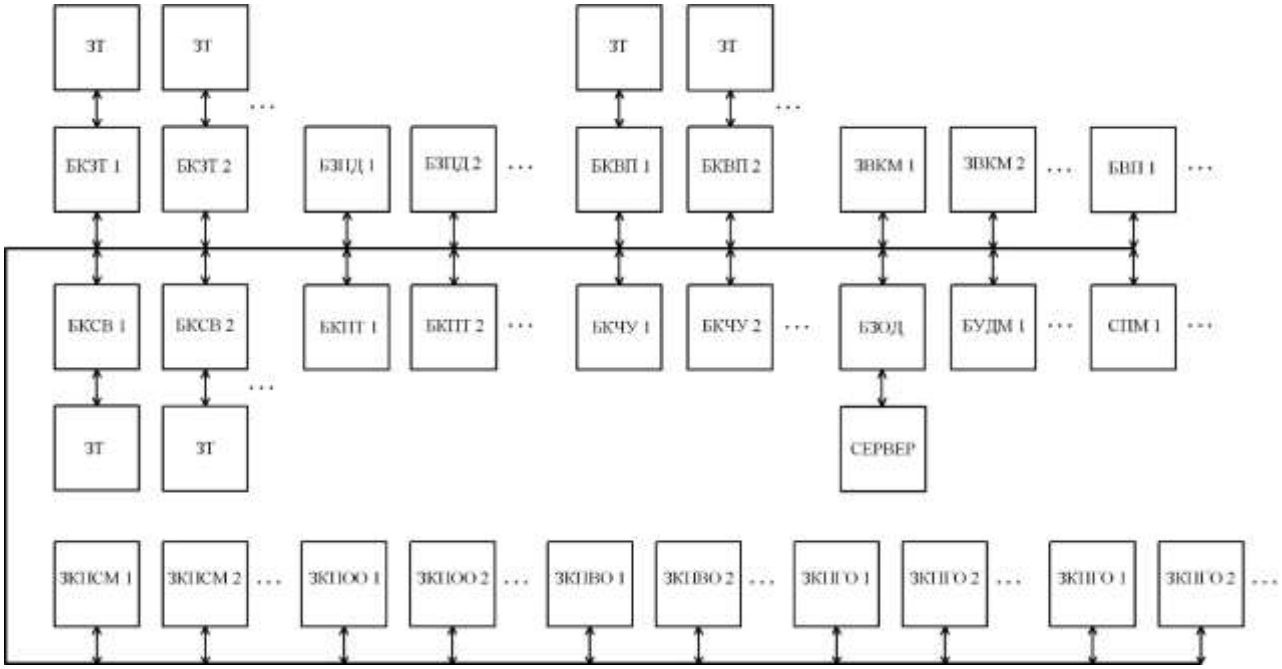


Рисунок 1.2 – Узагальнена структурна схема інформаційної підсистеми параметрів технологічного процесу виробництва молока

Також до складу інформаційної системи можуть входити специфічні структурні елементи, які використовуються в особливих випадках. До таких елементів, наприклад, відноситься засіб вимірювання кутової швидкості платформи конвеєрної доїльної установки, який використовується у системі автоматичного управління її кутовою швидкістю. Будь-яка доїльна установка має у своєму складі доїльні станки або стійлові місця, розташування яких визначається способом утримання тварин, типом доїльної установки та її структурою. Вони можуть бути обладнані доїльними апаратами з функцією керування процесом доїння та без функції керування процесом доїння, в обох випадках можливе використання інформаційних підсистем. Найважливішим складовим елементом більшості інформаційних підсистем є блок забезпечення процесу доїння. При використанні доїльних роботів, групових та конвеєрних

доїльних установок, доїльних установок з прохідними станками, БЗПД стаціонарно встановлюється у доїльному станку разом з доїльним апаратом. На стійловій доїльній установці БЗПД переміщується від стійла до стійла разом з доїльним апаратом та доярем. При використанні доїльних апаратів з функцією керування процесом доїння, які завжди обладнані електромагнітними пульсаторами, БЗПД забезпечує таке управління роботою пульсатора, яке спрямовано на оптимізацію процесу доїння. Оптимальним є такий процес доїння, під час якого імітується теля, тобто в залежності від виміряного значення інтенсивності молоковіддачі змінюється режим пульсацій вакууму у доїльних стаканах і доїльний апарат налаштовується на індивідуальні особливості конкретної тварини. Доїльний апарат без функції керування процесом доїння може бути обладнаний електромагнітним або механічним пульсатором. При використанні таких доїльних апаратів БЗПД не здійснює керування процесом доїння, режим доїння у цьому випадку незмінний, або змінюється в залежності від поточного часу доїння, а не в залежності від інтенсивності молоковіддачі. Також функціональним призначенням БЗПД може бути вимірювання, контроль, візуалізація певної кількості параметрів технологічного процесу виробництва молока, визначення моменту закінчення доїння, зняття доїльного апарата, отримання ідентифікаційної та вимірювальної інформації від транспондерів, які встановлені на тваринах, обмін інформацією з сервером інформаційної підсистеми та іншим обладнанням. Окрім цього, за допомогою БЗПД можливе ручне введення та передача до сервера інформаційної підсистеми інформації про температуру тіла тварин, їх вагу, травмованість, наявність та ступінь маститу, наявність стану охоти, необхідність розколу, ознаки, за якими тварину не можна доїти (наприклад, застосування антибіотиків), інформації про пошкодження або втрату транспондера, деяких інших параметрів. Після закінчення доїння тварини або групи тварин, отриману вимірювальну та службову інформацію БЗПД передає до сервера інформаційної підсистеми, де здійснюється її обробка за допомогою

спеціалізованого програмного забезпечення. Слід відзначити, що на доїльних установках найнижчої цінової категорії, БЗПД можуть бути відсутні, але інформаційні підсистеми можуть використовуватись. У цьому випадку вимірювання та контроль обмеженої кількості параметрів технологічного процесу виробництва молока забезпечується за допомогою іншого обладнання.

Основним елементом, який забезпечує обмін даними між сервером підсистеми та іншими її складовими, є БЗОД. Завданням БЗОД є управління потоками даних, які виникають в системі, буферизація та перетворення даних, перетворення інтерфейсів, забезпечення завадостійкості тощо. Уся вимірювальна та службова інформація елементами системи передається до БЗОД, який після необхідної її обробки, передає ці данні до сервера підсистеми. Передача даних від сервера до елементів системи також здійснюється за посередництва БЗОД. Слід відзначити, що потоки вимірювальної інформації у інформаційній підсистемі мають випадковий характер, характеристики інформаційних потоків в значній мірі визначаються типом доїльної установки, об'єм вимірювальної та допоміжної інформації у підсистемі відносно невеликий, тому не виникає необхідності у використанні швидкодіючих інтерфейсів для обміну даними. Найбільш важливими параметрами інтерфейсу, у цьому випадку, є його надійність, життєздатність та завадостійкість, тому що робота підсистеми здійснюється в умовах високого рівня промислових завад та великої імовірності механічних пошкоджень елементів системи.

При застосуванні у інформаційній підсистемі автоматичної або автоматизованої ідентифікації, на тваринах закріплюються транспондери, отримання інформації від них здійснюється за допомогою ЗТ. Транспондери бувають пасивними та активними, пасивні транспондери не мають власного джерела живлення і використовуються тільки для ідентифікації. До складу активних транспондерів, які мають власне джерело живлення, можуть входити засоби вимірювання активності та румінації.

БКЗТ призначений для забезпечення живлення ЗТ, керування режимами його роботи, дешифрації та перетворення даних, які надходять від ЗТ, перетворення інтерфейсів, виявлення помилок читання коду транспондера шляхом аналізу контрольних сум або інших методів, передачі даних отриманих від ЗТ до інших елементів інформаційної підсистеми. В залежності від способу та місця ідентифікації тварин та типу установки, БКЗТ може бути інтегрований в БЗПД, БКСВ або БКВП.

За допомогою БКПТ встановлюється факт проходження тварини до доїльного станка доїльної установки, вагової платформи, селекційних воріт, доїльного робота. Фактично, БКПТ здійснює операцію підрахунку тварин у потоці. Необхідність використання БКПТ зумовлена тим, що внаслідок втрати транспондера або помилки ідентифікації можлива втрата інформації, яка стосується як окремих тварин, так і їх груп. Застосування БКПТ дозволяє зменшити ці втрати.

За допомогою БКСВ забезпечується управління потоком тварин у доїльно-молочному відділенні тваринницької ферми. Після закінчення доїння тварини прямують до місця утримання, на їх шляху розташовуються селекційні ворота, за допомогою яких конкретна тварина спрямовується в місце утримання або в спеціальне приміщення, в якому проводяться ветеринарні та зоотехнічні процедури. Кількість БКСВ залежить від типу та структури доїльної установки, управління БКСВ здійснюється за допомогою інформації, яка надходить від сервера підсистеми. Кожний БКСВ обладнаний БКПТ та ЗТ, відповідно, до його складу входить інтегрований БКЗТ. Після ідентифікації у БКСВ аналізується інформація стосовно цієї тварини, яка надійшла від сервера, після чого приймається рішення про напрямок її руху. Рішення про розкол може бути прийняте на основі результатів контролю параметрів ТПВКМ, або на основі інформації, яку дояр вручну вводить до сервера або БЗПД.

Сучасні тваринницькі ферми з безприв’язним утриманням обладнуються автоматичними установками для чесання тварин, які розташовуються у місцях

їх руху. Управління чесальною установкою здійснюється за допомогою БКЧУ, який забезпечує управління напрямком обертання електродвигуна, вимірювальний контроль струму споживання електродвигуна з метою запобігання його перевантаження, вимірювання та вимірювальний контроль інших необхідних параметрів.

За допомогою вагової платформи здійснюється вимірювання та контроль ваги тварин. Забезпечення функціонування вагової платформи здійснюється за допомогою БКВП. У сучасних системах цей процес автоматизований, тому БКВП обладнується ЗТ. При відсутності автоматичних вагових платформ результати вимірювання ваги вводяться вручну до сервера підсистеми або до БКВП.

Будь-яке доїльно-молочне відділення тваринницької ферми має у своєму складі одну або декілька ємностей для зберігання молока, де воно перемішується або охолоджується протягом часу зберігання. Усі розрахунки з покупцями молока здійснюються на основі результатів вимірювання його кількості у цих ємностях. Для вимірювання кількості цього молока використовується ЗВКМ, який є складовою частиною інформаційної підсистеми.

Виключно на стійлових доїльних установках, у складі інформаційної підсистеми застосовуються БУДМ та ЛПМ. БУДМ керує електромагнітними клапанами дозатора, здійснює підрахунок порцій молока для визначення удою у кожній лінії, за допомогою БЗОД здійснює обмін даними з сервером. ЛПМ виконує такі самі функції, що і БУДМ, окрім управління електромагнітними клапанами.

БВП іноді використовується на доїльних установках різних типів, за допомогою цього функціонального вузла у інформаційній підсистемі візуалізуються деякі параметри технологічного процесу виробництва молока та результати їх вимірювального контролю.

Виробниками і покупцями сирого молока використовуються ЗКПСМ для контролю його якості. За допомогою ЗКПСМ контролюються такі параметри: жирність, вміст соматичних клітин, густина, в'язкість, pH, вміст білку, вміст лактози, температура, вміст води, вміст солі, бактеріальна заплідненість, деякі інші параметри.

За допомогою ЗКПГО вимірюються та контролюються параметри, необхідні для забезпечення роботи гноєзбирального обладнання. З метою запобігання примерзання скребка у холодну пору року здійснюється контроль температури підлоги, при її зниженні нижче порогового рівня забезпечуються антипримерзальні рухи скребка. Для калібрування скреперної системи, компенсації зміни довжини троса, реалізації алгоритму усунення перешкод, здійснюється вимірювання та вимірювальний контроль струму споживання привідного електродвигуна та лінійного зміщення скребка відносно початкового положення.

При використанні ЗКПВО, на основі результатів вимірювального контролю струму споживання, фазних напруг, температури обмоток, кутової швидкості, здійснюється діагностика технічного стану основного та резервного електродвигунів вакуумної установки. Окрім того, на основі результатів вимірювання розрідження у різних місцях вакуумпроводу, здійснюється автоматична підтримка його оптимального значення.

ЗКПДО призначені для вимірювання та контролю комплексу параметрів доїльного обладнання, до яких відносяться шпаруватість, амплітуда, частота, тривалість, співвідношення тактів, тривалість перехідних процесів та фронтів пульсуючого вакууму у доїльних стаканах доїльних апаратів. Також ці засоби здійснюють автоматичний контроль технічного стану та діагностику доїльного обладнання, контроль наявності пульсацій у доїльних стаканах, контроль технічного стану пульсаторів, контроль положення маніпулятора з метою запобігання його падінню, вимірювання температури та рівня води і миючих

засобів у системі промивання, деяких інших, специфічних для кожної доїльної установки параметрів.

Невід'ємною складовою частиною технологічного процесу виробництва молока є його охолодження. Завдяки зниженню температури створюються умови, при яких стримується розвиток мікроорганізмів і збільшується час зберігання. За допомогою ЗКПОО здійснюється контроль рівня, температури та ваги сирого молока у охолоджувальних ємностях, забезпечується контроль функціонування, діагностика та управління роботою складових елементів охолоджувального обладнання.

1.3 Постановка задачі

Ціллю бакалаврської кваліфікаційної роботи є розробка інформаційної підсистеми для групової доїльної установки, створення програмного забезпечення для персонального комп’ютера, який є компонентом підсистеми, програмного забезпечення для мікроконтролера, під управлінням якого працює один з найважливіших компонентів підсистеми – блок забезпечення обміну даними.

Завданням розробки є створення на рівні архітектури та структурних схем інформаційної підсистеми, яка призначена для використання на молочних тваринницьких фермах обладнаних груповою доїльною установкою, та забезпечує отримання, зберігання, обробку та представлення у необхідному вигляді інформації про параметри технологічного процесу виробництва молока. Інформаційна підсистема для групової доїльної установки забезпечує передавання інформації до персонального комп’ютера та представлення отриманої інформації у числовому та графічному вигляді на екрані монітора.

РОЗДІЛ 2. ВИБІР ТЕХНОЛОГІЙ ТА АРХІТЕКТУРА ПІДСИСТЕМИ

2.1 Обґрунтування вибору мови програмування для створення програмного забезпечення для персонального комп’ютера підсистеми

Мови та технології зазвичай розраховані на конкретні завдання та цілі. Наприклад, деякі мови використовуються тільки для розробки веб-сайтів, інші –

тільки для написання програм для Windows, а деякі можуть поєднувати обидві функції. Для того, щоб зробити вибір якісної та об’єктивної мови програмування, необхідно порівняти найбільш актуальні мови. На сьогодні найбільш актуальними мовами для вирішення поставленої задачі є C# і Java. Спочатку розглянемо мову програмування C#. C# – об'єктно-орієнтована мова програмування з системою безпечного типу для платформи .NET. Мова була розроблена Андерсом Ґалесбергом, Скоттом Вілтамутом і Пітером Голдом за підтримки Microsoft. Синтаксис мови C# подібний до мов програмування C++ і Java. C# має сувору статичну типізацію, підтримує поліморфізм, перевантаження операторів, посилання на функції-члени класу, властивості, події, атрибути, винятки та коментарі XML. C#, який дуже відрізняється від своїх попередників – C++, Delphi, Modules і Smalltalk – виключає певні шаблони, які спричиняють проблеми в розробці програмного забезпечення: так, C# не підтримує успадкування кількох класів (C++ підтримує) або виведення типу (на відміну від Haskell). Коли мова йде про C#, слід виділити наступні переваги:

- використовує проміжний код, тому може працювати на будь-якому комп’ютері, незалежно від апаратного та програмного забезпечення системи;
- повністю об'єктно-орієнтований – можна писати тільки код для класів;
- є багато додаткових функцій, які допомагають програмувати;
- може автоматично звільняти непотрібну пам'ять;
- підтримує політику .NET.

- програма C# може складатися з одного або кількох файлів, що містять вихідний код мови програмування C#. Кожен із цих файлів має розширення .cs. Недоліки включають поганий графічний інтерфейс платформи X; C# є внутрішньою частиною платформи .NET, тому сервер, на якому виконується програма, має бути на базі Windows; C# не дуже гнучкий, оскільки він в основному покладається на платформу .Net.

Далі розглянемо Java, об’єктно-орієнтовану мову програмування, розроблену компанією Sun Microsystems у 1991 році та офіційно випущену 23 травня 1995 року. Мова запозичує багато свого синтаксису з C і C++. Зокрема, він заснований на об'єктній моделі C++, але з модифікаціями. Виключає можливість певних конфліктних ситуацій, які можуть виникнути через помилки програмування, полегшує процес розробки об’єктно-орієнтованих програм. Набір операцій, які програміст повинен виконати на C/C++, делегується віртуальній машині. Java в першу чергу розроблена як незалежна від платформи мова з меншою кількістю апаратних функцій низького рівня, які можуть уповільнювати роботу програм порівняно з такими програмами, як C++. Java дозволяє при необхідності викликати підпрограми, написані на інших мовах програмування. Існує багато переваг використання Java як мови програмування.

Java є простою мовою програмування, яку легко вивчити та зрозуміти. Завдяки своєму простому коду Java є однією з найпростіших для вивчення та впровадження мов програмування. Крім того, Java видаляє всі складні функції C

і C++, такі як покажчики, структури, агрегати, і спрощує реалізацію коду. Однією з головних переваг Java є те, що це об'єктно-орієнтована мова програмування. Java є безпечною мовою. Такі мови, як C і C++, використовують покажчики для доступу до сховища. Це загроза безпеці, оскільки покажчики можуть призвести до неавторизованого доступу до пам’яті. Java також використовує концепції ООП, такі як інкапсуляція, абстракція та

успадкування, щоб покращити безпеку та запобігти доступу неавторизованих користувачів. Java не залежить від платформи, підтримує функцію WORA (Write Once Run Anywhere). Програми Java, написані на цій системі, можуть працювати на будь-якій іншій системі Java. Сумісність Java не залежить від операційної системи чи апаратного забезпечення, що робить платформу Java незалежною та надзвичайно гнучкою. Java – мова програмування високого рівня. Java має інтерпретатор, який перекладає код на мову машинного рівня, щоб машина могла його зрозуміти. Java є портативною мовою. Це пояснюється тим, що Java не залежить від платформи і не вимагає спеціального обладнання для роботи. Фактично це робить Java сумісною майже з усіма можливими пристроями. У Java JVM автоматично керує пам'яттю. Тому щоразу, коли об’єкт не посилається на клас і йому потрібно назвати, JVM автоматично видаляє його з програми, тому нам не потрібно писати будь-який додатковий код. Тому Java підтримує автоматичне збирання сміття. Java підтримує багатопотоковість. Пропускна здатність — це найменша можлива одиниця процесу. Багатопоточність є ключем до максимального використання процесора. Java – це мова програмування, яка підтримує багатопотоковість. Java дозволяє нам запускати кілька потоків одночасно. Вони мають спільну пам’ять для підвищення ефективності та продуктивності програми. Java регулярно отримує оновлення для виправлення помилок. Це робить Java однією з найстабільніших мов програмування. Оновлення негайно виправляють майже всі помилки. Тому важливо регулярно оновлювати Java. Java забезпечує ефективну стратегію розподілу пам'яті. За потреби JVM виділяє пам’ять з однієї з двох частин. Це дозволяє ефективно керувати сховищем.

У Java також є свої недоліки. Java споживає більше пам'яті, ніж мови програмування, такі як C і C++. Java також повільніше в порівнянні з ними, що пов’язано з додатковою роботою інтерпретатора для перетворення коду на машинну мову. JVM виконує різні функції бекенда, які сповільнюють швидкість роботи програми. Оскільки Java підтримує автоматичне збирання

сміття, вона завжди працює на стороні сервера, що впливає на продуктивність. Java значно відстає, коли справа доходить до графічного інтерфейсу. Конструктор графічного інтерфейсу Java не може створити складний інтерфейс користувача. У Java є багато фреймворків графічного інтерфейсу, таких як Swing, SWT, JavaFX, JSF тощо. Java абсолютно не має можливості зберігати дані користувача. В основному він зосереджений на зберіганні даних, але не захищений інструментами резервного копіювання. Java займає більше пам'яті, ніж інші мови програмування, такі як C і C++ [11]. Java використовує збирач сміття, але це негативно впливає на продуктивність. Java має багато багатослівних і складних

синтаксисів. Іноді ці складні синтаксиси важко запам’ятати. З цих причин багато

програмістів віддають перевагу Python або C++ перед Java, оскільки вони мають

відносно прості речення. Отже, Java має як переваги, так і недоліки. Мінуси Java відносно менші в порівнянні з плюсами, тому її доцільність для використання при розробці інформаційної технології досить висока. На основі аналізу двох мов програмування обрано Java для розробки даного програмного забезпечення.

З метою розробки серверних програм Java зазвичай використовують три середовища розробки, а саме IntelliJ IDEA, Eclipse і NetBeans. IntelliJ IDEA – це середовище розробки для створення програмних застосунків за допомогою Java. IntelliJ IDEA розроблялася компанією JetBrains. Він доступний у вигляді ліцензованої версії спільноти Apache 2 і власної комерційної версії. Обидва середовища можна використовувати для розробки інформаційної технології. Однак саме IntelliJ IDEA було вибрано для реалізації інформаційної технології в даній розробці. IntelliJ IDEA надає рекомендації щодо завершення коду, аналізу коду та перевірених інструментів рефакторингу. Середовище має такі інструменти, як контроль версій, зв’язок з багатьма мовами і фреймворками, це

середовище може стежити за контекстом розробника і самостійно запускати потрібні інструменти.

Особливості IntelliJ IDEA:

- розумний висновок: у ньому перераховані найбільш релевантні символи, які стосуються поточного контексту. Він безперервно переміщує новітні класи, методи тощо, збільшує список пропозицій. Відповідно, заповнення коду проводиться швидше.

- аналіз потоку даних: дане середовище може аналізувати потік даних та передбачати можливий символ при виконанні.

- мовне введення: можна легко інтегрувати фрагменти іншої мови у свій код Java, наприклад SQL.

- IntelliJ рекомендує глибокий та ефективний рефакторинг, так як він має необхідні знання про використання символів.

- середовище пропонується з широким набором вбудованих інструментів, як наприклад GIT, контроль версій, декомпілятор, покриття, база даних SQL тощо.

- наявний потужний компілятор, що може виявляти дублікати коду тощо.

- має надійну інтеграцію з серверами додатків.

Отже, завдяки наведеним перевагам, було вибрано мову програмування Java та середовище розробки IntelliJ IDEA.

Графічний інтерфейс користувача будемо розробляти за допомогою зручного інструменту – бібліотеки Swing, яка була розроблена для забезпечення більш функціонального набору програмних компонентів, ніж її попередники. Swing – це графічна бібліотека, яка пропонує велику офіційну документацію, гнучкість компонентів і відносно низький бар’єр для входу. Багато сучасних IDE включають графічні редактори для форм Swing. Компоненти Swing написані на Java і повністю незалежні від платформи. Переваги також включають такі функції:

- вікна програми малюються на графічному процесорі, це дозволяє анімацію в програмі;

- компоненти Swing реалізовані за принципом Model Look Controller, це дозволяє швидко додати потрібну функціональність до існуючого компонента;

- безпека коду завдяки реалізації на Java;

- широкий спектр середовищ розробки: Eclipse, Intellij IDEA, NetBeans;

- широкий вибір бібліотек з додатковими компонентами;

- велика спільнота прихильників, ви можете знайти відповідь на будь-яке питання великого значення в розвитку, обмежене в часі.

2.2 Технологія розробки JumpStarter C for AVR

Компілятори ImageCraft є похідними від компілятора LCC. Таким чином, портативний інтерфейс компіляторів LCC виконує такі оптимізації:

- Алгебраїчні спрощення та постійне згортання. Компілятор може замінити алгебраїчні вирази більш простими виразами. Компілятор також обчислює постійні вирази та «згортає» їх (наприклад, 1+1 стає 2). Компілятор також виконує ці оптимізації для констант з плаваючою комою, і результати можуть дещо відрізнятися, якщо константи з плаваючою комою не «складаються». Це пояснюється тим, що точність і діапазон операцій з плаваючою комою головного процесора (наприклад, Intel процесори) відрізняються від цільового ЦП. Усунення загального підвиразу основного блоку. Вирази, які повторно використовуються в базовому блоці (тобто послідовність прямолінійного коду без переходів), можуть кешуватися у створеному компілятором тимчасовій кеш-пам’яті та не обчислюватися повторно.

- Перемикач оптимізацій. Компілятор аналізує значення перемикачів і генерує код, використовуючи комбінацію бінарного пошуку та таблиць переходів. Таблиці переходів ефективні для щільно упакованих значень перемикачів, а двійковий пошук швидко знаходить правильну таблицю

переходів. У випадку, коли значення широко поширені або їх небагато, виконується простий пошук. Генератор коду компілятора використовує техніку, яка називається переписування дерева знизу вгору за допомогою динамічного програмування, для генерації коду складання, тобто згенерований код локально є оптимальним. Крім того, серверна частина може виконувати такі оптимізації.

- Оптимізація Peephole. Хоча локально оптимальний, згенерований код може містити надлишкові фрагменти, що є результатом різних операторів C. Оптимізація Peephole усуває деякі з цих надмірностей.

- Розподіл регистрів. Для цілей із декількома машинними регістрами (наприклад, AVR, MSP430 і ARM) для кожної функції компілятор виконує розподіл регістрів і намагається упакувати якомога більше локальних змінних у машинні регістри і таким чином підвищити продуктивність згенерованого коду. Вкористовується складний алгоритм, який аналізує використання змінної (наприклад, програмний діапазон, де вона використовується) і може навіть помістити кілька змінних в один реєстр, якщо їх використання не збігається. Це може викликати плутанину під час налагодження, оскільки може здатися, що змінні змінюють значення, коли їх зміни не очікуються. Однак це правильно, якщо регістр, раніше виділений для змінної, тепер призначений для іншої змінної.

Code Compressor доступний у версіях ADVANCED і PROFESSIONAL компіляторів. Він працює після зв’язування всієї програми та замінює часто використовувані фрагменти коду викликами функцій. Він може зменшити розмір програми на 30% (типове значення становить від 5% до 15%), не спричиняючи надто сильного уповільнення швидкості виконання.

Машинно-незалежний оптимізатор — це сучасний оптимізатор функціонального рівня, доступний у професійній версії компіляторів. Він виконує такі оптимізації на функціональному рівні, враховуючи вплив структур потоку керування. Присвоєння констант локальній змінній відстежується, а використання змінної замінюється константою, якщо це можливо. У поєднанні

з постійним складанням це може бути дуже ефективною оптимізацією. Розподільник регистрів доповнюється процесом побудови «веб» (відмінного від Інтернет-веб), який ефективно обробляє різні варіанти використання однієї змінної як кількох змінних, що дозволяє виконувати ще кращий розподіл регистрів.

Навіть із розширеним розподільником регістрів іноді цільовий процесор просто не має достатньо машинних регістрів, щоб зберігати всі кандидатські локальні змінні в регістрах. Ці нерозподілені змінні повинні бути поміщені в стек. Однак велике зміщення стека, як правило, менш ефективно для більшості цільових процесорів. Використовуючи той самий швидкий алгоритм, бекенд оптимально упаковує нерозподілені змінні, ділячи простір один з одним, якщо це можливо, зменшуючи загальний розмір стека функцій.

ImageCraft доклав значних зусиль для створення найсучаснішої інфраструктури оптимізатора. Наразі оптимізація MIO сприяє головним чином швидкості виконання та деякому невеликому покращенню розміру коду. Оптимізацію стиснення коду можна ввімкнути на додаток до оптимізації MIO. Ця комбінація дає найменший код, одночасно зменшуючи деяке зниження швидкості, введене компресором коду.

За деякими винятками, базова програма керування мікроконтролера складається з наступних частин:

- Деякі функції низького рівня, які взаємодіють з апаратним забезпеченням, наприклад, читання та запис регістрів вводу-виводу.

- Регістр та інший код ініціалізації системи.

- Набір обробників переривань для обробки даних реального часу, наприклад, введення сигнала датчика, спрацьовування таймера тощо.

- Набір функцій обробки «високого рівня», наприклад, робота із зібраними даними.

- Функція контролю. Це може бути основна процедура, яка проходить через усі функції високого рівня, або може бути перемикач завдань або RTOS, яка викликає функції за потреби.

Написання коду для мікроконтролера (MCU) зазвичай вимагає ініціалізації периферійних пристроїв мікроконтролера шляхом запису різних значень до їхніх регістрів вводу-виводу, а потім читання та запису в інші регістри вводу-виводу для виконання деяких функцій, таких як перетворення аналогового сигналу в цифрове значення за допомогою АЦП. С виділяється тим, що дозволяє писати такий код, не вдаючись до написання коду асемблера, оскільки регістри вводу-виводу зазвичай відображаються таким чином, що можна посилатися на них за назвою, наприклад:

unsigned char c = PINA;

У деяких продуктах включений конструктор програм, який генерує код ініціалізації периферійних пристроїв за допомогою інтерфейсу «point-and-click». Хоча таблиця даних все одно потрібна, вона може значно заощадити час на початку проектів.

Поширеним завданням програмування мікроконтролера є ввімкнення або вимкнення деяких бітів у регістрах вводу-виводу. Стандартний С добре підходить для перетворення бітів без використання інструкцій зі складання чи інших нестандартних конструкцій С. С визначає деякі порозрядні оператори, які є особливо корисними. Хоча компілятори генерують оптимальні інструкції для бітових операцій, вони можуть бути застосовані навіть для 8-бітних змінних.

a | b - порозрядний або. Вираз, позначений a, порозрядно поєднується з виразом, позначеним b. Це використовується для ввімкнення певних бітів, особливо коли використовується у формі призначення |=. Наприклад:

PORTA |= 0x80;

a & b - порозрядні і. Цей оператор корисний для перевірки, чи встановлено певні біти. Наприклад:

if ((PINA & 0x81) == 0)

Навколо виразів оператора & потрібні круглі дужки, оскільки він має нижчий пріоритет, ніж оператор ==. Це джерело багатьох програмних помилок у програмах на C.

a ^ b - порозрядне виняткове або. Цей оператор корисний для доповнення. Наприклад, у наступному випадку біт 7 перевертається:

PORTA ^= 0x80;

~a - порозрядне доповнення. Цей оператор виконує доповнення до виразу. Це особливо корисно в поєднанні з побітовим оператором і оператором для вимкнення певних бітів:

PORTA &= ~0x80;

Компілятор генерує оптимальні машинні інструкції для цих операцій. Наприклад, інструкцію sbic можна використовувати для порозрядного розгалуження, а оператор для умовного розгалуження на основі стану біта.

Оскільки компілятор реалізує мову ANSI C, поширеним методом розробки програми є використання компілятора для ПК, наприклад Borland C або Visual C, і спочатку налагодження логіки програми шляхом компіляції програми як програми для ПК. Очевидно, що апаратні частини повинні бути ізольовані та замінені або видалені за допомогою фіктивних процедур.

Як правило, 95% або більше коду програми можна налагодити за допомогою цього методу. Якщо у програмі, здавалося б, випадковий збій із змінними, які мають дивні значення або лічильник програми у дивних місцях, то, можливо, у програмі є перезапис пам’яті. Необхідно переконатися, що змінні покажчика вказують на дійсні місця пам’яті та що стеки не перезаписують пам’ять даних.

Code::Blocks IDE (C::B) — це кросплатформна C/C++ IDE із відкритим кодом, заснована на концепції функціональних розширень за допомогою плагінів. Це дозволяє розробникам надавати плагіни, які покращують IDE без жорсткого кодування цих покращень у код ядра IDE. C::B має підтримку

робочого простору та проекту, а також перегляд символів (наприклад, перехід до оголошення чи реалізації функції), а редактор підтримує всі сучасні функції, такі як підсвічування синтаксису та згортання коду. Базовий C::B є дуже гнучким і може підтримувати різноманітні хост-компілятори та крос-компілятори.

Менеджер проектів IDE дозволяє згрупувати список файлів у проект. Це дозволяє розбити програму на невеликі модулі. Коли ви виконуєте функцію збірки проекту, перекомпільовуються лише вихідні файли, які були змінені. Залежності файлу заголовка генеруються автоматично. Тобто, якщо вихідний файл містить файл заголовка, тоді вихідний файл буде автоматично перекомпільовано, якщо файл заголовка зміниться. На відміну від IDE у попередніх версіях продуктів, C::B не використовує стандартний make-файл, а використовує внутрішню схему на основі XML.

Налагоджувач JumpStart (JDB або JDB-AVR) — це налагоджувач вихідного коду C, який повністю інтегровано з IDE CodeBlocks. Для його використання не потрібно завантажувати додаткове програмне забезпечення. Особливості JDB:

- Крапки зупинки
- Змінне вікно спостереження
- Стеки викликів
- Степінг на рівні джерела та степінг на рівні складання
- Перегляд вмісту IO регистру
- Вступити, переступити і повернутися кроком
- «Poor Man’s Trace» - записи інструкцій і значень регістрів процесора
- Скидання цілі

JDB повністю інтегрований із IDE CodeBlocks, на 100% сумісний із JumpStarter C для AVR, а IDE+JDB завантажується та працює набагато швидше, ніж інші IDE/дебагери, пропонуючи користувачам можливості редагування, збирання та налагодження. Після встановлення драйвера модуля для

конкретного проекту ви можете встановити параметр модуля налагодження, викликавши діалогове вікно. Щоб використовувати JDB, потрібен модуль налагодження/програмування обладнання. Протягом багатьох років Atmel виготовив наступні модулі з різними характеристиками та цінами:

- AVR Dragon
- JTAGICE MkII
- JTAGICE3
- Atmel-ICE

Існують інші модулі, призначені лише для програмування (наприклад, STK500/STK600), але їх не можна використовувати для налагодження. JDB-AVR сумісний з усіма цими модулями.

2.3 Розробка архітектури інформаційної підсистеми для групової доїльної установки

Розроблена в процесі виконання кваліфікаційної бакалаврської роботи архітектура програмного забезпечення інформаційної підсистеми для групової доїльної установки наведена на рис. 2.1.

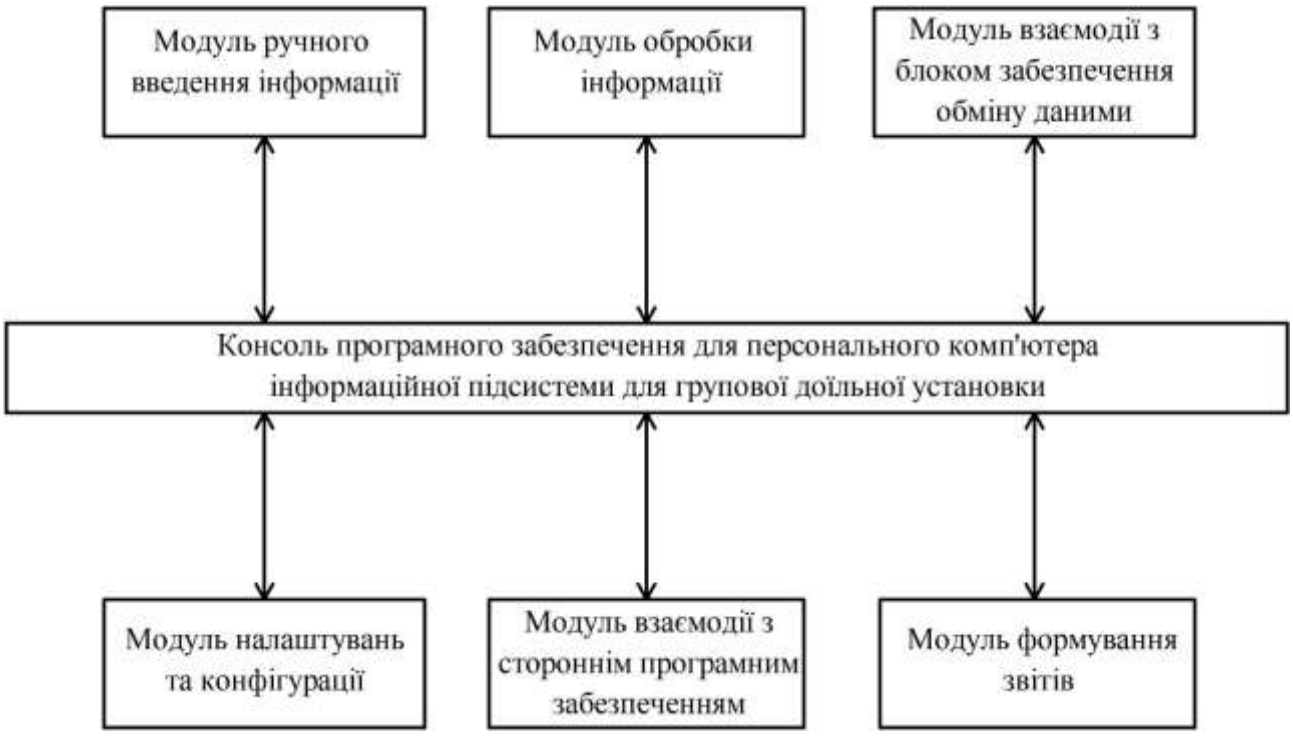


Рисунок 2.1 – Архітектура програмного забезпечення
інформаційної підсистеми для групової доїльної установки

Інформаційна підсистема для групової доїльної установки, що розробляється, призначена для використання на тваринницьких фермах, які спеціалізуються на виробництві молока і призначена для отримання, обробки, представлення та обліку інформації про параметри технологічного процесу виробництва молока з використанням групових доїльних установок з метою ефективного управління фермою. Інформаційна підсистема може працювати виключно на молочних фермах з безприв'язним утриманням тварин, дозволяє підвищити якість технологічного процесу виробництва молока та забезпечує ефективність роботи персоналу.

Архітектура програмного забезпечення інформаційної підсистеми складається з шести основних програмних модулів: модуля ручного введення інформації, модуля взаємодії з блоком забезпечення обміну даними, модуля налаштувань та конфігурації, модуля обробки інформації, модуля формування звітів, модуля взаємодії з стороннім програмним забезпеченням. Розроблене для використання на тваринницьких молочних фермах програмне забезпечення для інформаційної підсистеми для групової доїльної установки призначене для управління стадом, забезпечення контролю параметрів технологічного процесу виробництва молока на групових доїльних установках, функціонування різних варіантів інформаційних систем. До складу інформаційної підсистеми для групової доїльної установки входить блок забезпечення обміну даними, структурна схема якого наведена на рис. 2.2.

Розроблений на рівні структурної схеми та програмного забезпечення блок забезпечення обміну даними призначений для роботи у складі інформаційних підсистем на групових доїльних установках типу «Ялинка» і «Паралель», а також у складі доїльних установок «Карусель», «Тандем», доїльній установці з паралельно-прохідними станками та на стійлових доїльних

установках. Блок забезпечення обміну даними забезпечує високонадійний двонаправлений обмін даними за шістьма каналами між сервером системи та усім допоміжним обладнанням доїльної установки за допомогою двонаправленої гальванічно ізольованої «струмової петлі» при використанні відповідного програмного забезпечення.

До складу блока забезпечення обміну даними входить хаб, який дозволяє забезпечити обмін даними між USB - портом сервера пдсистеми, та чотирма USB - пристроями у форматі USB 3.0. Перші три USB - порта хаба використовуються для забезпечення обміну даними за шістьма каналами з складовими елементами нформаційної підсистеми для групової доїльної установки.

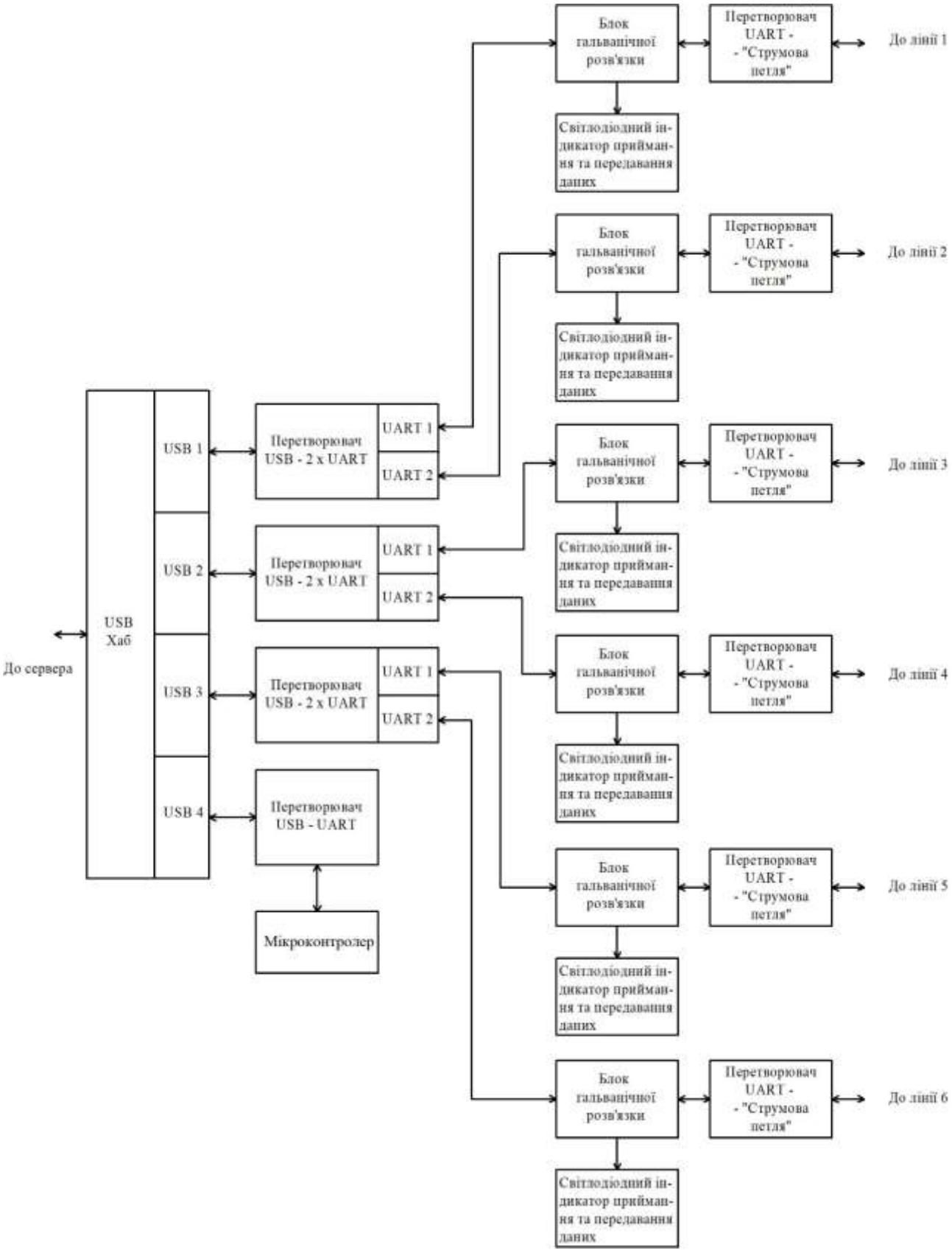


Рисунок 2.2 – Структурна схема блока забезпечення обміну даними

За допомогою перетворювачів USB – 2×UART, кожний USB - порт хаба забезпечує обмін даними у форматі інтерфейсу USB 3.0 за двома лініями послідовного двонаправленого обміну даними у форматі UART. Для забезпечення високої завадостійкості при обміні даними забезпечується оптична гальванічна розв'язка сигнальних кіл UART. Безпосередній обмін даними між блоком забезпечення обміну даними та складовими елементами інформаційної підсистеми здійснюється за допомогою двонаправленої «струмової петлі» у дуплексному режимі, з передаванням контрольних сум для підвищення надійності обміну даними. Перетворення сигналів UART в сигнали «струмової петлі», та зворотне перетворення, здійснюється за допомогою шести перетворювачів «UART - струмова петля» [12]. Процес прийому та передавання даних за кожним з каналів візуалізується за допомогою світлодіодних індикаторів. Максимальна відстань обміну даними, яку забезпечує БЗОД, складає 500 м, максимальна кількість пристроїв, які можуть здійснювати обмін даними при використанні одного каналу, дорівнює шістнадцяти. Блок забезпечення обміну данимии має інтегровану енергонезалежну пам'ять для резервного зберігання даних сервера інформаційної підсистеми. Четвертий USB - порт хаба використовується для обміну даними з мікроконтролером, який входить до складу блоку забезпечення обміну даними, та призначений для захисту програмного забезпечення персонального комп'ютера від несанкціонованого копіювання. У додатку Б наведено фрагмент лістингу програмного забезпечення для мікроконтролера блока забезпечення обміну даними на мові програмування JumpStarter C for AVR.

РОЗДІЛ 3. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ІНФОРМАЦІЙНОЇ ПІДСИСТЕМИ

3.1 Розробка програмного забезпечення інформаційної підсистеми для групової доїльної установки

Програмне забезпечення створено на мові програмування Java в середовищі IntelliJ IDEA та може функціонувати на платформах Windows 7, Windows 8 та Windows 10. Лістинг розробленого програмного забезпечення додатку А. Програмне забезпечення для інформаційної підсистеми для групової доїльної установки встановлюється на сервері інформаційної підсистеми, робота з програмним забезпеченням може здійснюватися з локальних або віддалених робочих станцій за допомогою веб-браузерів Mozilla Firefox та Google Chrome, або безпосередньо з сервера. Робочі станції з'єднуються з сервером загальної інформаційної системи молочної ферми через комп'ютерну мережу, віддалені робочі станції з'єднуються з сервером через Internet, у цьому випадку сервер інформаційної підсистеми повинен мати статичну IP-адресу.

Модуль ручного введення інформації призначений для внесення до бази даних інформації про тварину, а саме про номер у стаді номер, номер стада, номер тварини у державній базі даних, результати проведення контрольних доїнь, результати визначення температури та ваги, інформацію про батьків тварини, номер групи тварин, номер транспондера, породу, параметри екстер'єру та вади, кличку, дату народження, дати отелів, дати ветеринарних оглядів та їх результати, дати та параметри запліднень та абортів, інформацію про лікування та неможливість доїння, інформацію про розкол, коментарі та іншу інформацію. Окрім того, вручну вводиться інформація, яка необхідна для управління стадом або групами тварин, саме про день першого запліднення, день першого отелювання, кількість днів між заплідненням та запуском, кількість днів між заплідненням та пізнім запуском, кількість днів між заплідненням та отелюванням, кількість днів між отелюванням та

запліднення, кількість днів між абортом та заплідненням, кількість днів між ректальним дослідженням, кількість днів між заплідненням та початком запуску. Також забезпечується запис у відповідну базу даних інформації про область, населений пункт, поштову адресу, назву господарства, дані про директора, зоотехніків, осіб, які відповідають за роботу в піддсистемі, ідентифікатори операторів, що мають дозвіл на роботу у підсистемі, інші деякі параметри.

За допомогою модуля взаємодії з блоком забезпечення обміну даними, інформація про параметри технологічного процесу, яка отримана за допомогою складових елементів інформаційної підсистеми для групової доїльної установки, автоматично записується в базу даних. Також, за допомогою цього модуля здйснюється передавання службових даних від сервера підсистеми до її складових елементів з метою проведення налаштувань, встановлення режимів роботи та управління, відображення інформації. Інформація, обмін якою проводиться за допомогою цього модуля, залежить від конфігурації групової доїльної установки, методу утримання тварин, технічних можливостей та характеристик обладнання, яке використовується на молочній фермі. У типовій інформаційній підсистемі, яка використовується при прив’язному утриманні, цей модуль, у взаємодії з блоком забезпечення обміну даними, надає можливість отримання інформації від встановлених на доїльній установці блоків забезпечення процесу доїння про кількість повторних під'єднань доїльного апарату, наявність холостих доїнь, тривалість холостого доїння, кількість відпадань апарата, тривалість доїння, інтервал припуску, удій протягом одного доїння, удій у перші три тридцятисекундні інтервали з початку доїння, код транспондера, наявність маститу та ступінь у чвертях вимені, травмованість тварини, наявність стану «охоти», номер, електропровідність молока. За допомогою цього програмного модуля до блоку забезпечення процесу доїння передається інформація про номер тварини, заборону доїння та інформації про аномально низький удій. При взаємодії з

блоком управління селекційними воротами, модуль отримує код транспондера та забезпечує передавання даних про розколу. При взаємодії з зчитувачем транспондерів за допомогою цього модуля отримується код транспондера та їх кількість. При застосуванні блока управління чесальною установкою отримується код транспондера та інформація про тривалість та кількість разів користування установкою.

Модуль налаштувань та конфігурації забезпечує конфігурування програмного забезпечення інформаційної підсистеми для функціонування з різними структурами групових доїльних установок та різними варіантами інформаційних систем. За допомогою цього модуля програмне забезпечення налаштовується у відповідності до типу та структури доїльної установки, кількості траншей групової доїльної установки, кількості каналів обміну даними, типу блока забезпечення процесу доїння, кількості блоків забезпечення процесу доїння, розподілу мережевих номерів обладнання, кількості траншей, що обслуговуються одним доярем, кількості доярів, кількості селекційних воріт, кількості доїнь. Також за допомогою цього модуля встановлюється тип та номер порта для обміну даними з сервером, часовий інтервал між опитуваннями складових елементів інформаційної підсистеми, швидкість обміну даними.

Модуль обробки інформації забезпечує проведення інтерполяції та апроксимації даних про параметри технологічного процесу виробництва молока, розрахунок різноманітних залежностей, в тому числі лактаційних кривих, проведення контролю параметрів технологічного процесу виробництва молока, аналіз отриманих результатів з метою оцінки стану складових елементів технологічного процесу та можливості їх відхилення від норми, визначення тварин, у яких виявлено підозру на захворювання, прогнозування стану. Нижче наведено фрагмент лістингу програмного забезпечення модуля обробки інформації, який відповідає за її виведення.

```
import java.awt.BasicStroke;
```

```
import java.awt.Color;
import java.awt.Font;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.Rectangle;
import java.awt.geom.RoundRectangle2D.Double;
import java.util.Random;
import javax.swing.JPanel;
import org.w3c.dom.css.Rect;
public class GraphicsPanel extends JPanel {
private Line line;
public GraphicsPanel() {
line = new Line(25, 100, 30, 20);
}
@Override
public void paintComponent(Graphics g) {
Graphics2D g2 = (Graphics2D) g;
// g2.draw(line);
Color newColor = new Color(139, 0, 0);
g2.setStroke(new BasicStroke(6.0f));
g2.setColor(newColor);
g2.drawLine(9, 530, 777, 530);
Font font1 = new Font("TimesRoman", Font.BOLD, 14);
Font font = new Font("TimesRoman", Font.BOLD, 10);
Font font2 = new Font("TimesRoman", Font.BOLD, 18);
g2.setStroke(new BasicStroke(3.0f));
Color newColor1 = new Color(0, 58, 250, 100);
g2.setColor(newColor1);
for (int i = 0; i < 255; i++) {
```

```
float h = 500f / (float) Main.max * (float) Main.col[i];
int hh = (int) h;
g2.drawLine(9 + i * 3, 527, 9 + i * 3, 527 - hh);
Main.col[i]=0;
}
// g2.setStroke(new BasicStroke(6.0f));
/*
* g2.setColor(newColor1); g2.drawLine(30, 505, 750, 505);
* g2.drawLine(30, 480, 750, 480); g2.drawLine(30, 455, 750, 455);
* g2.drawLine(30, 430, 750, 430); g2.drawLine(30, 405, 750, 405);
* g2.drawLine(30, 380, 750, 380); g2.drawLine(30, 355, 750, 355);
* g2.drawLine(30, 330, 750, 330); g2.drawLine(30, 305, 750, 305);
* g2.drawLine(30, 280, 750, 280); g2.drawLine(30, 255, 750, 255);
* g2.drawLine(30, 230, 750, 230); g2.drawLine(30, 205, 750, 205);
* g2.drawLine(30, 180, 750, 180); g2.drawLine(30, 155, 750, 155);
* g2.drawLine(30, 130, 750, 130); g2.drawLine(30, 105, 750, 105);
* g2.drawLine(30, 80, 750, 80); g2.drawLine(30, 55, 750, 55);
* g2.setColor(Color.BLUE);
*/
// g2.drawRect(30, 50, 250, 250);
// g2.drawLine(750, 530, 750, 30);
}
}
```

Модуль формування звітів створює набір звітів, структуру яких визначає оператор системи. Формуються звіти за тваринами, за тваринами з негативною динамікою розвитку, за тваринами з підозрами на різні захворювання, звіти з результатами контролю параметрів технологічного процесу.

Забезпечується формування звітів з будь-якою конфігурацією - за стадом, за групою тварин, за фермами, за траншеями, за часовими проміжками.

Модуль взаємодії з іншим програмним забезпеченням організовує експорт даних, отриманих в результаті роботи програмного забезпечення інформаційної підсистеми, до електронних таблиць та інших популярних програм.

3.2 Розробка програмного забезпечення для мікроконтролера блока забезпечення обміну даними

Алгоритм роботи блока забезпечення обиіну даними, який входить до складу інформаційної підсистеми для групової доїльної установки, полягає в наступному. Після початку роботи доїльної установки, блок забезпечення обміну даними у безперервному режимі здійснює опитування стану усіх складових елементів інформаційної підсистеми. Опитування здійснюється послідовно, у відповідності до попередньо запрограмованих унікальних номерів складових елементів інформаційної підсистеми для групової доїльної установки, цей процес відбувається циклічно, до закінчення роботи установки. Обмін інформацією між блоком забезпечення обміну даними та елементами підсистеми забезпечується за допомогою фізичного середовища «струмова петля» та дводротової лінії обміну даними з гальванічною оптичною розв’язкою. Для контролю правильності переданих та прийнятих даних використовуються контрольні суми. Після передавання адреси та коду запиту на приймання даних від складового елементу інформаційної підсистеми протягом певного часового інтервалу очікується відповідь. Якщо дані у відповідь не надійшли, це свідчить про їх неготовність, тому блок забезпечення обміну даними інкрементує адресу і звертається з запитом на передавання даних до наступного за адресою компонента підсистеми. Наступний раз звернення з запитом на передавання даних до складового елементу відбудеться тільки у наступному циклі опитування, після того, як будуть опитані усі інші елементи.

У додатку Б наведено лістинг програмного забезпечення для мікроконтролера на мові програмування JumpStarter C for AVR для мікроконтролера ATTINY2313A виробництва компанії Microchip, під управлінням якого працює блок забезпечення обміну даними, який входить до складу інформаційної підсистеми для групової доїльної установки. Цей фрагмент відповідає за встановлення інтервалу очікаваняя даних під час взаємодії з блоком забезпечення процесу доїння.

```
procedure cycle_right( sec : byte );
begin
   Case sec of
     0 :      portB := %00000001;
     2 :      portB:= %00000011;
     6 :      portB := %00000010;
     41 :     portB := %00000000;
     43 :     portB := %00001100;
     46 :     portB := %00011100;
     51 :     portB := %00010110;
     53 :     portB := %00000110;
     59 :     portB := %00000000;
     62 :     portD := portD or %00100000;
     91 :     portD := portD and %11011111;
     96 :     portD := portD or %01000000;
     98 :     portD := portD and %10111111;
       OnCycle2 := false;
   endcase;
   inc(cnt2);
end;
```

Висновок

У бакалаврській кваліфікаційній роботі розроблено інформаційну підсистему для групової доїльної установки. У якості об’єкта дослідження обрано процес, концепція та архітектура побудови інформаційних підсистем для групової доїльної установки. Предметом дослідження є методи інформатизації молочних ферм обладнаних груповою доїльною установкою.

Розроблена підсистема призначена для впровадження на підприємствах які виробляють коров’яче молоко. Підсистема забезпечує отримання, обробку, представлення у зручному для оператора вигляді інформації про параметри технологічного процесу виробництва коров’ячого молока з метою підвищення ефективності управління молочною фермою. Інформаційна підсистема забезпечує передачу інформації про параметри технологічного процесу до сервера на основі персонального комп’ютера та представлення отриманої інформації у зручному для керівника ферми вигляді.

У першому розділі роботи проведено аналіз проблеми та завдань дослідження, здійснено огляд існуючих методів отримання інформації про параметри технологічного процесу виробництва молока, обрано об’єкт дослідження та сформульовані задачі дослідження.

У другому розділі проведено обгрунтування вибору архітектури інформаційної підсистеми для групової доїльної установки, обрано мову програмування для програмного забезпечення сервера підсистеми та розглянуто технологію розробки програмного забезпечення для мікроконтролерів JumpStarter C for AVR, розроблено архітектуру підсистеми.

У третьому розділі розроблено програмне забезпечення для сервера інформаційної підсистеми для групової доїльної установки та програмне забезпечення для мікроконтролера блоку забезпечення обміну даними.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Кучерук, В. Ю. Класифікація інформаційно-вимірювальних систем для доїльно-молочних відділень тваринницьких ферм / В. Ю. Кучерук, Є. А. Паламарчук, П. І. Кулаков, Т. В. Гнесь // Вимірювальна та обчислювальна техніка в технологічних процесах. – 2015. – №2. – с. 89–93.

2. Войтюк, В. В. Вимірювання витрати молока в молочній лінії доїльного апарата : автореф. дис. // канд. техн. наук : / В. В. Войтюк ; Національний університет «Львівська політехніка». – Л., 2011. – 21 с.

3. Кулаков, П. І. Оцінювання стану тварин на основі результатів вимірювального контролю їх зоотехнічних параметрів / П. І. Кулаков // Вісник інженерної академії України. – 2014. – № 3 – 4. – с. 154–158.

4. Кулаков, П. І. Класифікація систем ідентифікації тварин для доїльно-молочних відділень тваринницьких ферм / П. І. Кулаков // Вимірювальна та обчислювальна техніка в технологічних процесах. – Одеса, 2015, с. 134.

5. Дмитрів, В. Т. Основи теорії машиновикористання у тваринництві / В. Т. Дмитрів. – Львів, 2008. – 260 с.

6. Rotz, C. A. Automatic Milking Systems, Farm Size, and Milk Production / C. A. Rotz, C. U. Coiner, K. J. Soder // Journal of Dairy Science. – 2003. – V. 86. – № 12. – P. 1605–1614.

7. Brandt, M. Invited review: Technical solutions for analysis of milk constituents and abnormal milk / M. Brandt, A. Haeussermann, E. Hartung // Journal of Dairy Science. – 2010. – V. 93. – № 2. – P. 427–436.

8. Chagunda, M. G. A model for detection of individual cow mastitis based on an indicator measured in milk / M. G. Chagunda, N. C. Friggens, M. D. Rasmussen, T. Larsen. // Journal of Dairy Science. – 2006. – V. 89. – P. 2980–2998.

9. Meyer, Bertrand (1997). Object-Oriented Software Construction. Prentice Hall. ISBN 0-13-629155-4.

10. Метатрон 21. Нове покоління. Високотехнологічний стрибок в доїльній

техніці. Рекламний проспект. / GEA Group. – 2013. – 7 с.

11. Java Introduction – [Електронний ресурс] – Режим доступу: https://www.w3schools.com/java/java_intro.asp.

12. Шрюфер Е. Обробка сигналів: цифрова обробка дискретизованих сигналів / За ред В.П.Бабака. – К.: Либідь, 1992. – 296 с.