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 зоотехнічних параметрів тварин

Виконав – студент 4 курсу, групи 41-кн
освітньої програми "Комп'ютерні науки"
спеціальності 122 Комп'ютерні науки
Пірогов Богдан Вікторович
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 зоотехнічні параметри тварин 14

1.3. Постановка задачі 19

РОЗДІЛ 2. ВИБІР ТЕХНОЛОГІЙ ТА АРХІТЕКТУРА ПІДСИСТЕМИ 20

2.1. Технологія розробки AVR Studio C..... 20

2.2. Технологія взаємодії PHP та MySQL 25

2.3. Розробка архітектури інформаційної підсистеми зоотехнічних параметрів тварин 31

РОЗДІЛ 3. РОЗРОБКА ІНФОРМАЦІЙНОЇ ПІДСИСТЕМИ ЗООТЕХНІЧНИХ ПАРАМЕТРІВ ТВАРИН 34

3.1. Розробка програмного забезпечення інформаційної підсистеми зоотехнічних параметрів тварин..... 34

3.2. Розробка програмного забезпечення для мікроконтролера блока забезпечення процесу доїння 41

ВИСНОВОК..... 43

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ 44

ДОДАТОК А..... 46

ДОДАТОК Б 52

Перелік умовних позначень, скорочень і термінів

ІС	Інформаційна система
ІПС	Інформаційна підсистема
САУ	Система автоматичного управління.
БД	База даних.
ПЗ	Програмне забезпечення.
ПД	Програмний додаток.
ПЗП	Постійний запам’ятовуючий пристрій.
ІПЗПТ	Інформаційна підсистема зоотехнічних параметрів тварин
СУБД	Система управління базами даних
ЗВКМ	Засіб вимірювання кількості молока
ВП	Вимірювальний перетворювач

Вступ

У загальному вигляді, виробництво коров’ячого молока являє собою біотехнічну систему, до складу якої входить набір машин, технологій, груп тварин та колективів операторів. До складу таких систем можуть включатися не лише технічні та біологічні об’єкти, а і деякі абстрактні елементи, такі як організація і управління технологічними процесами [1-3]. Сучасні тваринницькі ферми з великою кількістю тварин та їх доїльно-молочні відділення потребують впровадження технічних рішень, котрі дають можливість відслідковувати розвиток кожної тварини і оптимально керувати процесом її утримання. Системи автоматичного управління (САУ) тваринницькою фермою в автоматичному режимі забезпечують комплексне управління технологічним процесом виробництва молока, оптимізацію продуктивності тварин, здоров'я стада, репродуктивної здатності стада, годівлі, ефективністю праці обслуговуючого персоналу. Також при їх використанні здійснюється облік готового продукту, аналіз та контроль зоотехнічних параметрів тварин, аналіз показників роботи технологічного обладнання. Використання САУ забезпечує ефективне управління стадом та тваринницькою фермою в цілому [4-6]. Невід’ємною складовою сучасних САУ є мікропроцесорні засоби отримання інформації про зоотехнічні параметри тварин, які є складовою частиною процесу виробництва молока. Серед таких засобів особливо важливу роль відіграють інформаційні системи (ІС) та інформаційні підсистеми, що входять до їх складу [7].

Актуальність

Наразі, при сучасному стані технологій виробництва молока в Україні, в умовах незадовільно розвинутої автоматизації, підвищення продуктивності праці на тваринницьких фермах є важливим та актуальним завданням. Впровадження інформаційних систем та технологій, як випливає з світового досвіду, збільшує продуктивність праці, зменшує завантаження персоналу, підвищує якість технологічного процесу виробництва молока. Виходячи з

цього, подальший розвиток теорії і практики розробки та впровадження засобів автоматизації та інформатизації зоотехнічних технологій є актуальним завданням.

Об’єкт дослідження – процес та концепція побудови інформаційної підсистеми зоотехнічних параметрів тварин.

Предмет дослідження – методи інформатизації підприємств з виробництва коров’ячого молока.

Мета дослідження

Метою та завданням бакалаврської кваліфікаційної роботи є розробка інформаційної підсистеми зоотехнічних параметрів тварин, структурної схеми підсистеми та її складових компонентів, розробка програмного забезпечення для персонального комп’ютера інформаційної підсистеми та програмного забезпечення для мікроконтролерів, що входять до складу компонентів підсистеми.

Під час виконання кваліфікаційної роботи бакалавра необхідно провести аналіз наукової та технічної літератури, різноманітної технічної документації, стандартів у галузі інформатизації та автоматизації технологічних процесів виробництва молока, здійснено розробку архітектури інформаційної підсистеми зоотехнічних параметрів тварин, розробку структурної схеми її компонентів та програмних модулів.

РОЗДІЛ 1. АНАЛІТИЧНИЙ ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Об’єкт та предмет дослідження

Для оцінки якості виконання доярами своїх обов'язків, за допомогою сучасних інформаційних підсистем зоотехнічних параметрів тварин (ІПЗПТ) здійснюється вимірювання та контроль цілого ряду параметрів, більшість з яких регламентуються стандартом. Якість роботи доярів, в певній мірі, характеризує тривалість роботи доїльної установки T_U, яка повинна знаходитись в певних межах, які залежать від статистичних характеристик стада або групи тварин, кількості доїльних апаратів, кількості тварин, алгоритму роботи доїльного апарату, типу доїльної установки, якості роботи дояра, інших факторів [8-10]. Також робота доярів характеризується кількістю повторних під'єднань доїльних стаканів K_{PDS}, кількістю випадків холостих доїнь K_{HD} (доїльний апарат працює, а припуску молока немає), тривалістю холостих доїнь T_{HD}, кількістю відпадань доїльного апарату K_{DA}.

Тривалість доїння t_{TD} певним чином характеризує стан тварини. Якщо тривалість доїння конкретної тварини значно збільшилася у порівнянні з середнім значенням, це може свідчити про погану підготовку тварини до доїння доярем, або наявність стресового стану. Середнє значення тривалості доїння конкретної тварини змінюється на протязі періоду лактації та визначається шляхом усереднення певної кількості результатів вимірювань.

Середня інтенсивність молоковіддачі I_{MS} також певним чином характеризує стан тварини, її визначають як відношення разового удою до тривалості доїння. Якщо середня інтенсивність молоковіддачі конкретної тварини значно зменшилася у порівнянні з середнім значенням, це може свідчити про наявність маститу, або погану підготовку тварини до доїння, або наявність стресового стану. Середнє значення інтенсивності молоковіддачі конкретної тварини може визначатися шляхом усереднення результатів багаторазових вимірювань. Миттєве значення інтенсивності молоковіддачі I_{MV} контролюється протягом усього часу доїння з метою визначення моменту

переходу до режиму додоювання та моменту закінчення доїння і автоматичного або ручного зняття доїльних стаканів.

Важливими зоотехнічними параметрами, які характеризують якість роботи дояра, є інтенсивність молоковіддачі в перші тридцять секунд після початку доїння I_{30}, інтенсивність молоковіддачі на часовому проміжку від тридцяти до шістдесяти секунд після початку доїння I_{60}, інтенсивність молоковіддачі на часовому проміжку від шістдесяти до дев'яноста секунд після початку доїння I_{90}. Параметр I_{30} повинен мати значення не менше ніж 1,5 кг/хв, а параметри I_{60} та I_{90} знаходитись в межах від 4 кг/хв до 11 кг/хв. Відхилення цих параметрів від норми свідчить про неякісну підготовку тварини до доїння, наявність у тварини стресового стану, можливу наявність маститу [11].

Також важливим показником стану тварини та якості роботи дояра є латентний період. Для вимірювання латентного періоду необхідно визначити момент початку подразнення дійок, що на практиці зробити дуже важко. Тому вимірювання латентного періоду замінюють вимірюванням часу припуску молока, який визначається як час після початку доїння, за який удій тварини склав 100 г молока. В середньому, якщо тварина не тугодійна, не хвора, не знаходиться в стані стресу та якісно підготована доярем до доїння, то на протязі 20 с після початку доїння її мінімальний удій повинен складати 100 г молока. Якщо час, за який було отримано 100 г молока, більше норми, це свідчить про стресовий стан тварини, який можливо виник внаслідок неякісної підготовки її до доїння.

Також одним з найважливіших зоотехнічних параметрів є удій тварини. Розрізняють разовий удій V_R протягом одного доїння, добовий удій V_D, удій за період лактації V_L. Суттєве зниження разового або добового удою свідчить про можливе захворювання тварини, або її стресовий стан, або про неналежне виконання своїх обов'язків доярем під час підготовки тварини до доїння. На сучасних молочних фермах доїння може здійснюватись два рази на добу – вранці та ввечері, або три рази на добу – вранці, вдень та ввечері. Якщо

ранковий, денний, або вечірній удій менше на двадцять і більше відсотків аналогічного удою за попередній день, то це є ознакою неякісної підготовки тварини до доїння доярем [12].

Температура тіла тварини є важливим показником її стану. З метою раннього розпізнавання хвороб необхідно контролювати ректальну температуру тіла тварини. Контроль температури здійснюється під час планових ветеринарних оглядів та під час знаходження тварини на доїльній установці. Результати контролю температури за допомогою відповідних технічних засобів передаються до серверу ІПЗПТ. Дуже часто підвищення температури, або ж навпаки, її зниження – це перша ознака початку захворювання, яке можна розпізнати. Найчастіше, причиною підвищення температури в період після отелення, є післяотельна інтоксикація. Друга причина, яка найчастіше є причиною підвищення температури, це мастити. Причини маститу – незадовільна гігієна корівника та доїння, а також незадовільна організація годівлі. При порушенні загального стану і підвищенні температури потрібне швидке і ефективне лікування для того, щоб уникнути подальшого спаду продуктивності або ж втрати тварини.

У теперішній час мастит є однією з найбільших проблем молочного скотарства у багатьох країнах світу. Внаслідок захворювання на мастит знижується продуктивність тварин, погіршується якість молока та його технологічні властивості, підвищуються витрати на діагностику і лікування тварин, виникає соціальна загроза попадання маститного молока у продаж. При виникненні маститу змінюється цілий комплекс параметрів сирого молока, в тому числі його електропровідність. Найбільший вплив на збільшення значення електропровідності молока мають іони хлоридів та натрію, концентрація яких при маститах збільшується. Виходячи з цього, шляхом вимірювального контролю значення однієї тільки електропровідності, можна з доволі високим ступенем вірогідності діагностувати мастит, в тому числі на початкових стадіях.

Оцінка стану тварини на основі результатів визначення тільки одного зоотехнічного параметру має відносно низьку достовірність внаслідок наявності великої кількості факторів, які впливають на його значення [13]. Відхилення стану тварини від норми супроводжується одночасною зміною певної кількості параметрів. На рисунку 1.1, для прикладу, наведені типові експериментальні залежності добового удою та електропровідності молока, від дня лактації, при виникненні маститу.

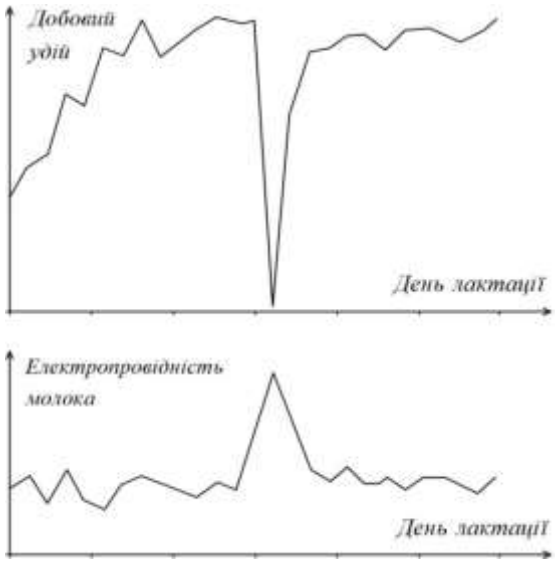


Рисунок 1.1 – Типові експериментальні залежності добового удою та електропровідності молока від дня лактації при виникненні маститу

Як слідує з рисунку 1.1, майже одночасно із збільшенням електропровідності значно зменшується добовий удій. Експериментальне виявлення цих двох факторів свідчить про наявність у тварини початкових стадій маститу з достатньо високим ступенем імовірності. Але остаточний висновок можна зробити тільки після проведення відповідних лабораторних досліджень проб молока та секретів вимені.

У більшості ІПЗПТ здійснюється визначення зоотехнічних параметрів, які стосуються певних груп тварин. До таких параметрів, відносяться загальна кількість отриманого молока V_S, загальний удій певної групи тварин V_G, удій,

отриманий кожним доярем V_D, загальний удій отриманий у доїльному станку V_{ST}, тривалість роботи установки T_U, середній удій тварини у групі або стаді за добу V_{SD}, середній удій тварини у групі або стаді за період лактації V_{SL}, середній час доїння тварини у групі або стаді t_{DS}, параметри сирого молока, які отримані від групи тварин.

В переважній більшості випадків ІПЗПТ є одним з найважливіших елементів інформаційних систем для доїльно-молочних відділень тваринницьких ферм та САУ ферм. За допомогою ІПЗПТ здійснюється вимірювання та контроль комплексу різноманітних та різнорідних зоотехнічних параметрів. На основі результатів вимірювального контролю та вимірювання параметрів можуть бути виявлені та встановлені підозри на захворювання тварин, виявлений їх стресовий стан, проведена оцінка якості підготовки доярем тварин до доїння, проведена комплексна оцінка якості роботи персоналу ферми, проведено аналіз якості селекційної підібраності стада або груп тварин. Іноді параметри тварин, які вимірюються або контролюються за допомогою ІПЗПТ, не є індивідуальними параметрами. У деяких випадках вони відносяться до певної групи тварин, склад якої залежить від типу доїльної установки.

1.2 Особливості отримання інформації про зоотехнічні параметри тварин

Найчастіше, для вимірювання удою та інших зоотехнічних параметрів тварин, використовують ковшовий сенсор або спеціалізовані оптичні та кондуктометричні сенсори [14]. За допомогою відповідних засобів на їх основі забезпечується вимірювання удою, тривалості доїння, інтенсивності потоку молока та інших параметрів, які необхідні для САУ та для реалізації алгоритму оптимального доїння. Досягнута у теперішній час точність вимірювання в багатьох випадках недостатня для відмови від трудомісткої процедури періодичних контрольних доїнь, оцінювання та прогнозування стану тварин,

визначення моменту закінчення доїння для зняття доїльного апарату, забезпечення оптимального процесу доїння. Використання традиційних вимірювачів кількості рідини, запозичених з інших галузей, виявилося малоефективним, оскільки процес молоковіддачі є унікальним, а точність показань молокомірів залежить від фізико-хімічних властивостей молока, динаміки молоковіддачі, ряду інших чинників. При використанні оптичного сенсора інтенсивності потоку молока для доїльного апарату неможливо забезпечити високу точність вимірювання удою та інших похідних параметрів внаслідок значного впливу на результат вимірювання піни та пузирів повітря, які обов'язково виникають на виході колектора. Окрім того, додаткову похибку вносить нерівномірність потоку молока та неминуча зміна геометричних розмірів та форми пузирів повітря. Внаслідок наявності та випадкового характеру появи пузирів та піни у молочній лінії доїльного апарату може значно збільшуватись похибка вимірювального перетворення традиційних кондуктометричних витратомірів. У оптичного кореляційного витратоміру молока похибка вимірювання значно збільшується при великих значеннях молочного потоку та при наявності у ньому піни та пузирів повітря [15]. Це пов'язано з тим, що вихідні сигнали його розташованих на певній відстані фотоприймачів, які корелюються, у цих умовах втрачають подібність. При використанні ротаційних витратомірів для вимірювання інтенсивності молочного потоку та удою, виникає швидке їх забруднення молочним каменем, що призводить до збільшення похибки вимірювання та відмов, а за допомогою існуючих технічних та хімічних засобів неможливо забезпечити їх якісне післядоїльне промивання. Це приводить до збільшення бактеріальної заплідненості молока і, відповідно, до зниження його якості і закупівельної ціни. Використання ультразвукових витратомірів для вимірювання удою та інтенсивності молочного потоку виявилося малоефективним внаслідок значного збільшення похибки вимірювання внаслідок наявності пузирів та піни, а також великого значення похибки вимірювання при малих потоках молока.

Широко розповсюджені ковшові сенсори відрізняються надійністю та мають високі метрологічні характеристики при вимірюванні удою. При використанні ковшових сенсорів удій визначається з певною дискретністю (як правило сто грам), що унеможливлює високоточне вимірювання миттєвої інтенсивності молоковіддачі, що є дуже важливим для визначення моменту зняття доїльного апарату.

Перспективним є непряме вимірювання удою та інших вторинних зоотехнічних параметрів на основі прямого вимірювання рівню молока у молокоприймальній камері доїльного апарату, що зумовлює необхідність розробки спеціалізованого сенсора рівню. Для використання у доїльному обладнанні бажано використовувати безконтактні сенсори, наприклад фотоелектричні, у яких відсутні рухомі частини. Дана обставина зумовлена тим, що наявність рухомих частин значно погіршує якість промивки, внаслідок чого збільшується бактеріальна заплідненість молока.

Розглянемо конструкцію та принцип дії молокоприймальної камери доїльного апарату, схематичне креслення якої наведено на рисунку 1.2.

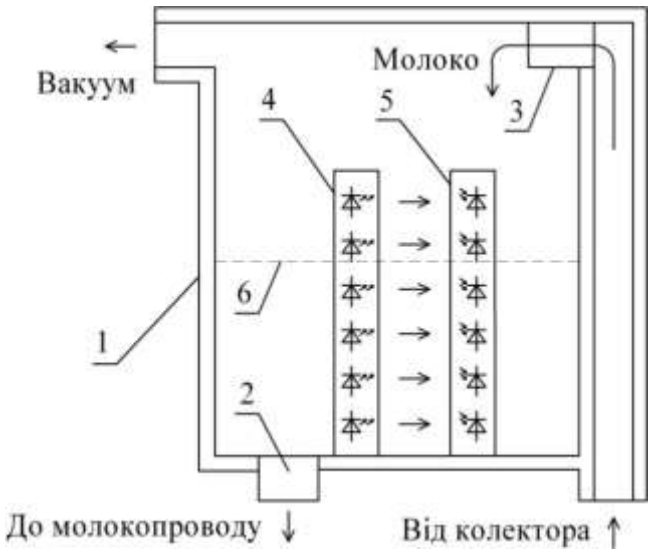


Рисунок 1.2 – Молокоприймальна камера доїльного апарату з фотоелектричним сенсором рівню молока: 1 - корпус камери, 2 - зливний

електроклапан, 3 - піновідділювач, 4 - інфрачервоні випромінювачі, 5 - фотоприймачі, 6 - рівень молока

Під дією вакууму, молоко з колектора всмоктується через піновіддюлювач в молокоприймальну камеру. У цій камері розташований розглянутий вище ВП рівню рідини та зливний електроклапан. В процесі доїння рівень молока в молокоприймальній камері збільшується. Після досягнення певного рівню відкривається зливний клапан і накопичене в камері молоко скидається у загальний молокопровід. В залежності від інтенсивності молоковіддачі встановлюється пороговий рівень, при якому здійснюється скидання молока.

Позначимо через S_M площу дна молокоприймальної камери. Об'єм *i*-тої порції молока визначається виразом:

$$V_{Mi} = S_M H_{Mi} , \tag{1.1}$$

де H_{Mi} - рівень молока в камері перед скиданням *i*-тої порції.

Тоді разовий удій тварини визначається виразом:

$$V_R = S_M \sum_{i=1}^{N_M} H_{Mi}, \tag{1.2}$$

де N_M – кількість порцій молока, скинутих в молокопровід на протязі тривалості доїння однієї тварини.

Протягом доїння вимірюється поточний час та тривалість доїння кожної тварини t_{TD}. Інтенсивність молоковіддачі I_{MS} визначають як відношення разового удою до тривалості доїння:

$$I_{MS} = \frac{V_R}{t_D} \tag{1.3}$$

Інтенсивність молоковіддачі на протязі перших тридцяти секунд після початку доїння визначається виразом:

$$I_{30} = \frac{V_{M30}}{T_{30}}, \tag{1.4}$$

де V_{M30} – удій тварини на тридцятій секунді після початку доїння; T_{30} –часовий інтервал тривалістю тридцять секунд.

Інтенсивність молоковіддачі на протязі часового інтервалу від тридцяти до шістдесяти секунд після початку доїння визначається виразом:

$$I_{60} = \frac{V_{M60} - V_{M30}}{T_{30}} \tag{1.5}$$

де V_{M60} – удій тварини на шістдесятій секунді після початку доїння.

Інтенсивність молоковіддачі на протязі часового інтервалу від шістдесяти до дев'яноста секунд після початку доїння визначається виразом:

$$I_{90} = \frac{V_{M90} - V_{M60}}{T_{30}} \tag{1.6}$$

де V_{M90} – удій тварини на дев'яностій секунді після початку доїння.

Миттєва інтенсивність молоковіддачі визначається як перша похідна удою за часом, часовий інтервал цифрового диференціювання у цьому випадку значно менший, ніж час, за який у молокоприймальній камері формується порція молока. Позначимо через H_{M1} рівень молока у камері в момент часу t_1, а через H_{M2} – рівень молока у камері в момент часу t_2. Тоді вираз, за яким визначається миттєва інтенсивність молоковіддачі:

$$I_{MV} = \frac{S_M \left(H_{M2} - H_{M1}\right)}{t_2 - t_1}. \tag{1.7}$$

Вимірювання латентного періоду замінюють вимірюванням часу припуску молока T_P, який визначається як час після початку доїння, за який удій тварини склав сто грам молока. Об'єм молокоприймальної камери доїльного апарату перевищує об'єм, який відповідає ста грамам молока, тобто $N_M = 1$. Виходячи з цього, об'єм молока, який відповідає ста грамам його ваги, визначається виразом:

$$V_{M100} = S_M H_{M100}, \tag{1.8}$$

де H_{M100} – рівень молока у молокоприймальній камері, який відповідає ста грамам його ваги.

Алгоритм контролю часу припуску молока полягає в наступному. Через двадцять секунд після початку доїння здійснюється вимірювання об'єму молока у молокоприймальній камері. Якщо виміряне у цей момент часу значення

об'єму молока більше або дорівнює V_{M100}, вважається, що час припуску відповідає нормі.

1.3 Постановка задачі

Ціллю бакалаврської кваліфікаційної роботи є розробка інформаційної підсистеми зоотехнічних параметрів тварин, розробка програмного забезпечення для персонального комп’ютера, що входить до складу підсистеми, програмного забезпечення для мікроконтролера, під управлінням якого працюють складові елементи інформаційної підсистеми.

Завданням розробки є створення на концептуальному рівні підсистеми, яка призначена для використання на підприємствах з виробництва коров’ячого молока, та дозволяє отримувати, зберігати, обробляти та представляти у необхідному вигляді інформацію про зоотехнічні параметри дійних тварин. Інформаційна підсистема забезпечує передавання отриманої 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 Технологія розробки AVR Studio C

Мова програмування С програмного пакету AVR Studio використовується для написання програмного забезпечення для мікроконтролерів AVR-архітектури компанії Microchip, які входять до складу компонентів інформаційної підсистеми. AVR Studio - це інструментарій для розробки програмного забезпечення, який забезпечує розробників інтегрованим середовищем розробки, що включає компілятор С, асемблер, компонувальник і відладчик.

Модулі мікроконтролера розташовані у виділених безперервних блоках у просторі пам’яті, і їх можна побачити як інкапсульовані одиниці. Модулі інкапсульовані в структурах С, в яких містяться всі регістри модулів. Розглянемо іменування та методи доступу до реєстру та сумісні із заголовком AVR файли, які забезпечують читабельність і переносимість кодів, написаних у С-коді.

Встановлення та очищення бітів регістру є основними операціями, які використовуються у вбудованому програмуванні та представляють техніку, на якій базується будь-яка програма. Операції Read-Modify-Write є класом атомарних операцій. Вони зчитують місце пам’яті та одночасно записують у нього нове значення, або повністю нове значення, або частину попереднього значення. Оскільки він має широке застосування, читання значення біта часто використовується в умовних виразах (наприклад, оператор if) і як умова у виразах циклу (наприклад, оператор while). Загальним випадком використання цієї техніки є опитування прапора переривання, що означає зчитування значення біта та виконання набору інструкцій, якщо біт встановлено/очищено. Біти регістру можна встановлювати, очищати та читати за допомогою бітових масок, наданих файлом заголовка.

Щоб встановити певний біт у регістрі, рекомендованим стилем кодування є використання екземпляра структури, оголошеного у файлі заголовка, і визначення макросу бітових масок. Використання двійкового оператора АБО з бітовою маскою гарантує, що інші параметри бітів, зроблені в цьому регістрі, залишаться незмінними, не впливаючи на операцію.

```
ADC0.CTRLA |= ADC_ENABLE_bm; /* Enable the ADC peripheral */
```

Щоб очистити біт із регістра, двійковий оператор I буде застосовано між регістром і об’єднаним значенням бітової маски. Ця операція також зберігає інші налаштування бітів незмінними.

```
ADC0.CTRLA &= ~ADC_ENABLE_bm; /* Disable the ADC peripheral */
```

Бітову маску ADC_ENABLE_bm визначено у файлі заголовка, як показано нижче.

```
#define ADC_ENABLE_bm 0x01 /* ADC Enable bit mask */
```

Читання бітового значення необхідно для різних програм. Наприклад, прапор ADC RESRDY встановлюється апаратним забезпеченням, коли результат ADC готовий. Наведений нижче список кодів показує, як перевірити, чи встановлено цей біт, і виконати деякі інструкції в цьому випадку.

```
if(ADC0.INTFLAGS & ADC_RESRDY_bm) /* Check if the ADC result is ready */
{
 /* Insert some instructions here */
}
```

Щоб перевірити, чи біт очищений, і виконати інструкції або зачекати, поки він залишається нульовим, можна використати наступний код. У цьому випадку інструкції всередині цього циклу виконуватимуться, доки не буде готовий результат аналого-цифрового перетворення.

```
while(!(ADC0.INTFLAGS & ADC_RESRDY_bm))
{
/* Insert some instructions here */
```

```
}
```

Крім того, до регістрів можна отримати доступ також за допомогою визначень макросу з файлу заголовка. Наведений нижче приклад показує, як встановити біт за допомогою цих визначень.

```
ADC0_CTRLA |= ADC_ENABLE_bm; /* Enable the ADC peripheral */
```

Щоб очистити біт за допомогою визначень макросу, можна використати наступний приклад коду.

```
ADC0_CTRLA &= ~ADC_ENABLE_bm; /* Disable the ADC peripheral */
```

Наведений нижче лістинг показує, як перевірити, чи встановлено цей біт, і виконати деякі інструкції в цьому випадку.

```
if(ADC0_INTFLAGS & ADC_RESRDY_bm) /* Check if the ADC result is ready */
{
 /* Insert some instructions here */
}
```

Щоб перевірити, чи біт нульовий, і виконати інструкції, поки він залишається нульовим, можна використати наступний код. У цьому випадку інструкції всередині цього циклу виконуватимуться, доки не буде готовий результат АЦП.

```
while(!(ADC0_INTFLAGS & ADC_RESRDY_bm))
{
 /* Insert some instructions here */
}
```

Альтернативним способом встановлення, очищення та читання бітів регістра є використання позицій бітів, які також надаються файлом заголовка пристрою. Щоб встановити біт за допомогою макросу позиції біта та отримати доступ до регістрів за допомогою оголошення структури, можна використати наступний приклад коду.

```
ADC0.CTRLA |= (1 << ADC_ENABLE_bp); /* Enable the ADC peripheral */
```

Щоб очистити біт за допомогою макросу позиції біта, подано наступний приклад.

ADC0.CTRLA &= ~(1 << ADC_ENABLE_bp); /* Disable the ADC peripheral */

Щоб перевірити, чи встановлено біт, можна використати наступний код.

```
if(ADC0.INTFLAGS & (1 << ADC_RESRDY_bp)) /* Check if the ADC result is ready */
{
 /* Insert some instructions here */
}
```

Щоб перевірити, чи біт нульовий, і виконати інструкції, поки він залишається нульовим, можна використати наступний код. У цьому випадку інструкції всередині цього циклу виконуватимуться, доки не буде готовий результат АЦП.

```
while(!(ADC0.INTFLAGS & (1 << ADC_RESRDY_bp)))
{
 /* Insert some instructions here */
}
```

Щоб встановити біт за допомогою макросу позиції біта та отримати доступ до регістрів за допомогою визначень макросу, можна використати наступний приклад коду.

ADC0_CTRLA |= (1 << ADC_ENABLE_bp); /* Enable the ADC peripheral */

Щоб очистити біт за допомогою макросу позиції біта, подано наступний приклад.

ADC0_CTRLA &= ~(1 << ADC_ENABLE_bp); /* Disable the ADC peripheral */

Щоб перевірити, чи встановлено біт, можна використати наступний код.

if(ADC0_INTFLAGS & (1 << ADC_RESRDY_bp)) /* Check if the ADC result is ready */

```
{
 /* Insert some instructions here */
}
```

Щоб перевірити, чи біт нульовий, і виконати інструкції, поки він залишається нульовим, можна використати наступний код. У цьому випадку інструкції всередині цього циклу виконуватимуться, доки не буде готовий результат АЦП.

```
while(!(ADC0_INTFLAGS & (1 << ADC_RESRDY_bp)))
{
 /* Insert some instructions here */
}
```

Під час написання драйверів для типів модулів, які мають кілька екземплярів, той факт, що всі екземпляри мають однакову карту регістрової пам’яті, можна використати, щоб зробити драйвер придатним для повторного використання для всіх екземплярів типу модуля. Якщо драйвер приймає аргумент покажчика, що вказує на відповідний екземпляр модуля, драйвер можна використовувати для всіх модулів цього типу. Це є великою перевагою з точки зору портативності. Крім того, написаний код може бути переносним між пристроями одного сімейства.

У пристрої з декількома таймерами/лічильниками функції ініціалізації та доступу до цих модулів можуть спільно використовуватися всіма екземплярами модулів замість копіювання тих самих рядків коду для кожного екземпляра. Навіть незважаючи на те, що передача вказівника модуля до функцій буде невеликою, загальний розмір коду буде зменшено, оскільки код повторно використовується для всіх екземплярів кожного типу модуля. Крім того, за допомогою цього підходу можна значно скоротити час розробки, вартість обслуговування та мобільність.

У пристрої з декількома таймерами/лічильниками функції ініціалізації та доступу до цих модулів можуть спільно використовуватися всіма екземплярами

модулів замість копіювання тих самих рядків коду для кожного екземпляра. Навіть незважаючи на те, що передача вказівника модуля до функцій буде невеликою, загальний розмір коду буде зменшено, оскільки код повторно використовується для всіх екземплярів кожного типу модуля. Крім того, за допомогою цього підходу можна значно скоротити час розробки, вартість обслуговування та мобільність.

2.2. Технологія взаємодії PHP та MySQL

Інтерфейс програмування додатків, або API, визначає набір класів, методів, функцій та змінних, які можна викликати з програми для виконання поставлених завдань. Стосовно PHP додатків, які повинні взаємодіяти з базами даних, необхідні для цього API, як правило, представлені PHP-модулями. API можуть бути процедурними чи об'єктно-орієнтованими. При використанні процедурних API викликаються функції для виконання будь-яких операцій, а у разі об'єктно-орієнтованих використовуються класи і потім викликаються методи створених об'єктів. Другий підхід, як правило, кращий, тому що він більш сучасний і сприяє написанню більш організованого коду.

API MySQL надає кілька способів підключення до бази даних із PHP-додатку. У документації MySQL термін конектор (connector) відноситься до частини програмного забезпечення, яка відповідає за підключення до сервера MySQL. MySQL надає безліч конекторів для різних мов програмування, зокрема PHP.

Для забезпечення взаємодії PHP програми з сервером баз даних необхідно написати PHP-код, який виконує підключення до сервера, виконання запитів до бази даних тощо. Від програмного забезпечення сервера потрібно надати API, яке PHP-додаток зможе використовувати, а також функціонал, відповідальний за взаємодію програми з сервером. Програмне забезпечення, що реалізує такий функціонал, зазвичай називають конектором, оскільки воно дозволяє додатку

підключитися до сервера баз даних. У ряді випадків конектор для потреб потребує додаткові бібліотеки.

Драйвером називається частина програмного забезпечення, що відповідає за взаємодію програми з конкретним типом серверів баз даних. Драйвер також може звертатися до зовнішніх бібліотек, таких як клієнтська бібліотека MySQL або драйвер MySQL. Ці бібліотеки реалізують низькорівневий протокол взаємодії із сервером MySQL. Як приклад можна навести рівень абстракції для роботи з базами даних PHP (PDO), який може використовувати один із кількох драйверів, специфічних для конкретних баз даних. Може використовуватися драйвер PDO MYSQL, який дозволяє PDO взаємодіяти з MySQL-сервером. У документації MySQL термін драйвер означає ділянку програмного коду, що входить до складу конектора і відповідає за зв'язок із конкретною системою управління базами даних.

У документації до PHP неодноразово використовується термін «модуль». PHP-код як такий складається з ядра і приєднаних до нього необов'язкових модулів, які збільшують коло завдань, які може виконувати ядро. Модулі, що відносяться до MySQL, такі як mysqli і драйвер PDO MySQL, взаємодіють з ядром за допомогою фреймворку PHP модулів. Зазвичай модулі надають API-інтерфейс PHP-програмісту, щоб той міг програмно використовувати можливості модуля. Однак деякі модулі, які використовують фреймворк PHP-модулі, не надають програмістам жодних інтерфейсів. Драйвер PDO MySQL, наприклад, не надає API. Він надає інтерфейс лише абстрактного шару PDO, що лежить вище. Терміни API і модуль не можна сприймати як синоніми, оскільки модуль може не надавати API програмісту.

API надає на вибір два набори інструментів для підключення до сервера баз даних MySQL:

- модуль PHP mysqli;
- об'єкти даних PHP (PDO).

Модуль mysqli, або як його ще називають покращений (improved) модуль MySQL, був розроблений, щоб дати можливість програмістам повною мірою скористатися функціоналом MySQL сервера версій 4.1.3 і вище. Модуль mysqli включається до постачання PHP версій 5 і вище, mysqli має ряд переваг та удосконалень у порівнянні з mysql, які полягають у наступному:

- об'єктно-орієнтований інтерфейс;
- підтримка запитів, що готуються;
- підтримка мультизапитів;
- підтримка транзакцій;
- поліпшені можливості налагодження.

Поряд із об'єктно-орієнтованим інтерфейсом модуль надає і процедурний інтерфейс. Модуль mysqli збирається за допомогою фреймворку модулів PHP, його вихідний код розташований у директорії ext/mysqli.

Об'єкти даних PHP, або PDO, являють собою абстракцію конектора баз даних для PHP додатків. PDO надає API інтерфейс взаємодії з базою даних, який залежить від конкретної СУБД. Теоретично, при використанні PDO можна поміняти сервер баз даних, наприклад Firebird на MySQL, і це призведе лише до незначних змін в PHP-коді. Як інші подібні абстракції можна привести JDBC для Java-додатків і DBI для Perl. Поряд з перевагами PDO, такими як простота та переносимість API, є його головний недолік: PDO підтримує не всі можливості сервера баз даних, доступні в останніх версіях MySQL. Наприклад, засобами PDO не можна створювати множинні запити, хоча MySQL їх підтримує. PDO збирається за допомогою фреймворку модулів PHP, його вихідний код розташований у директорії ext/pdo.

Драйвер PDO MySQL не є API як таким, принаймні з точки зору програміста. Драйвер PDO MYSQL розташовується між самим PDO та сервером MySQL. Програміст викликає функції інтерфейсу API PDO, а PDO у свою чергу використовує драйвер PDO MySQL для обміну даними та командами з сервером MySQL. Драйвер PDO MYSQL лише один із багатьох

PDO-драйверів. Для більшості СУБД є свої PDO драйвери, наприклад драйвери для Firebird або PostgreSQL серверів. Драйвер PDO MYSQL збирається за допомогою фреймворку модулів PHP, його вихідний код розташований у директорії ext/pdo_mysql. Він не надає API-інтерфейс програмісту PHP.

За замовчуванням запити використовують режим буферизації. Це означає, що результати запиту негайно надходять від сервера MySQL до PHP та зберігаються у пам'яті PHP-процесу. Це дозволяє робити додаткові операції, такі як підрахунок кількості рядків та переміщення (пошук) поточного покажчика результату. Це також дозволяє виконувати нові запити в тому ж з'єднанні під час роботи над результатом останнього запиту. Недоліком режиму буферизації є те, що великі набори результатів можуть вимагати досить багато оперативної пам'яті. Пам'ять залишається зайнятою до тих пір, поки всі покажчики на результуючий набір не будуть видалені або результуючий набір не буде звільнений, що автоматично відбувається під час закінчення останнього запиту. Термін "результат збереження" ("store result") також використовується для режиму буферизації, оскільки весь результуючий набір зберігається відразу.

Небуферизовані MySQL запити виконуються та повертають resource, поки дані очікують на отримання на сервері MySQL. Це використовує менше пам'яті PHP, але може збільшити навантаження на сервер. Поки повний результуючий набір не буде переданий із сервера, ніякі подальші запити не можуть бути надіслані через це з'єднання. Небуферизовані запити можуть бути відкладені як "use result".

Наступні характеристики буферизованих запитів повинні бути використані у випадках, коли очікується лише обмежений результуючий набір або необхідно знати кількість рядків, що повертаються до прочитання усіх рядків. Небуферизований режим має бути використаний якщо очікується на великі розміри результуючих наборів. Оскільки буферизовані запити застосовуються за замовчуванням, такі приклади демонструють, як виконати

небуферизовані запити в кожному з API. Нижче наведено приклад лістингу небуферизованого запиту:

```
<?php
$mysqli  = new mysqli("localhost", "my_user", "my_password", "world");
$uresult = $mysqli->query("SELECT Name FROM City",
MYSQLI_USE_RESULT);
if ($uresult) {
  while ($row = $uresult->fetch_assoc()) {
     echo $row['Name'] . PHP_EOL;
  }
}
?>
```

Кодування символів має бути чітко визначено, оскільки впливає на кожну дію, у тому числі на дії з наслідками безпеки. Наприклад, механізми екранування (такі як mysqli_real_escape_string() для mysqli та PDO::quote() для PDO_MySQL) залежать від цих налаштувань. Важливо розуміти, що ці функції не використовують кодування символів, визначене в запиті, наприклад, такі запити не впливатимуть на поведінку цих функцій.

Модуль mysqli надає подвійний інтерфейс програмісту. Підтримуються як процедурна, і об'єктно-орієнтована парадигми програмування. Користувачі, що переходять зі старого модуля mysql, можливо, нададуть перевагу процедурному інтерфейсу. Він дуже схожий на інтерфейс старого модуля, і в багатьох випадках функції відрізняються тільки префіксом в імені. Деякі mysqli-функції приймають дескриптор з'єднання першим аргументом, на відміну від відповідних їм функцій старого модуля, які приймають його як останній необов'язковий аргумент. На додаток до процедурного, користувачі можуть використовувати об'єктно-орієнтований інтерфейс. Об'єктно-орієнтований інтерфейс пропонує функції згруповані за метою їх застосування,

що полегшує їх пошук і освоєння. Тим не менш, у практичних прикладах до функцій наводиться код для обох парадигм.

Якихось принципових відмінностей у продуктивності між інтерфейсами немає. Користувачі вільні у виборі інтерфейсу, ґрунтуючись на особистих уподобаннях. Після виконання запиту результати можна отримати відразу, або прочитати рядок за рядком з сервера. Буферизація набору результатів за клієнта дозволяє серверу якомога раніше звільнити ресурси, пов'язані з результатами запиту. Тому рекомендується використовувати буферизовані набори результатів. mysqli::query() поєднує виконання запиту та буферизацію набору результатів. PHP-програми можуть вільно оперувати даними всередині буферизованих результуючих наборів. Швидка навігація рядками наборів обумовлена тим, що набори повністю розташовуються в пам'яті клієнта. Слід пам'ятати, що найчастіше обробка результатів клієнта простіше, ніж засобами сервера.

Якщо клієнтські ресурси обмежені, і в той же час не потрібно підтримувати низьке навантаження на сервер, можна використовувати буферизовані результуючі набори. Навігація за такими таблицями неможлива, тому що так чи інакше мають бути оброблені всі рядки набору. Функції mysqli::query(), mysqli::real_query() і mysqli::multi_query() призначені для виконання запитів, що не готуються. На рівні протоколу клієнт-серверної взаємодії MySQL за виконання запитів відповідають команда COM_QUERY та текстовий протокол. Коли використовується текстовий протокол, сервер MySQL перед відправкою клієнту перетворює всі дані в результуючому наборі текстових рядків. Це перетворення виконується незалежно від типу даних SQL-стовпця результуючої таблиці. Клієнтські бібліотеки mysql, своєю чергою, отримують усі дані, приймаючи їх за рядки. На клієнті не проводиться жодного зворотного перетворення до вихідних типів, всі дані, отримані програмою залишаються PHP рядками. Якщо використовується бібліотека mysqlnd, можна включити перетворення цілих чисел і чисел з плаваючою точкою зі стовпців

таблиці в PHP числа. Це робиться завданням налаштування підключення MYSQLI_OPT_INT_AND_FLOAT_NATIVE. У такому випадку mysqlnd перевірятиме метадані стовпців і перетворюватиме SQL-числа цих полів у PHP-числа, якщо ці значення не виходять за рамки допустимих діапазонів типів даних PHP. Тобто, наприклад, SQL INT число потрапить до PHP додатку у вигляді цілого.

2.3. Розробка архітектури інформаційної підсистеми зоотехнічних параметрів тварин

Розроблена в процесі виконання кваліфікаційної бакалаврської роботи архітектура інформаційної підсистеми зоотехнічних параметрів тварин наведена на рис. 2.1.

Інформаційна підсистема зоотехнічних параметрів тварин, що розробляється, призначена для використання на підприємствах з виробництва коров'ячого молока і забезпечує отримання, обробку, представлення у необхідному вигляді та безперервний облік інформації про комплекс зоотехнічних параметрів дійних тварин з метою ефективного управління стадом. Інформаційна підсистема може працювати на фермах з прив'язним та безприв'язним утриманням тварин, дозволяє значно підвищити якість управління стадом, збільшує ефективність роботи персоналу ферми.



Рисунок 2.1 – Архітектура інформаційної підсистеми
зоотехнічних параметрів тварин

Архітектура програмного забезпечення інформаційної підсистеми складається з шести основних програмних модулів: модуля ручного введення інформації, модуля взаємодії з обладнанням, модуля налаштувань та конфігурації, модуля обробки інформації, модуля формування звітів, модуля взаємодії з стороннім програмним забезпеченням.

Розроблене для використання на тваринницьких підприємствах програмне забезпечення для інформаційної підсистеми зоотехнічних параметрів тварин призначене для комплексного управління молочним стадом, забезпечення контролю параметрів технологічного процесу виробництва молока, функціонування різних варіантів інформаційних систем.

До складу інформаційної підсистеми зоотехнічних параметрів тварин входить блок забезпечення процесу доїння, структурна схема якого наведена на рис. 2.2.



Рисунок 2.2 – Структурна схема блока забезпечення процесу доїння

Блок забезпечення процесу доїння призначений для використання на доїльних установках типу «Ялинка», «Паралель», «Тандем», та на доїльній установці з паралельно-прохідними станками та може використовуватися у складі інформаційної підсистеми або в автономному режимі.

Блок забезпечення процесу доїння призначений для роботи з доїльними апаратами без функції керування процесом доїння, за його допомогою забезпечується вимірювання індивідуального удою тварин без їх ідентифікації, вимірювання інтенсивності молоковіддачі, визначення моменту закінчення доїння на основі результатів вимірювального контролю миттєвої інтенсивності молоковіддачі, формування сигналу необхідності ручного зняття доїльного апарата, передавання інформації про удій тварини до засобу вимірювання кількості молока (ЗВКМ).

Електричне живлення блока забезпечення процесу доїння забезпечується від загальної лінії живлення електронного обладнання доїльної установки

постійною напругою 24 В, для живлення активних електронних компонентів застосовується постійна напруга 5 В, яка отримується за допомогою відповідного перетворювача. При використанні блока забезпечення процесу доїння, для вимірювання параметрів молоковіддачі може бути застосований ковшовий сенсор, або сенсор рівня молока у молокоприймальній камері доїльного апарата.

У випадку застосування ковшового сенсора, блок забезпечення процесу доїння забезпечує вимірювання індивідуального удою тварин в діапазоні від 0,1 кг до 99,9 кг з дискретністю 0,1 кг, зведена похибка вимірювання не перевищує 3 %. При використанні фотоелектричного сенсора рівня молока у молокоприймальній камері доїльного апарата зведена похибка вимірювання індивідуального удою не перевищує 1,5 %. Відносна похибка вимірювання та формування необхідних для функціонування блока забезпечення процесу доїння часових інтервалів – 0,1 %.

РОЗДІЛ 3. РОЗРОБКА ІНФОРМАЦІЙНОЇ ПІДСИСТЕМИ ЗООТЕХНІЧНИХ ПАРАМЕТРІВ ТВАРИН

3.1. Розробка програмного забезпечення інформаційної підсистеми зоотехнічних параметрів тварин

Програмне забезпечення створено на мові програмування PHP четвертої версії та може функціонувати на платформах Windows XP, Windows 7, Windows 8, Windows 10, Linux Mandriva 2010.1, Ubuntu 11.04, Ubuntu 12.04. Лістинг програмного забезпечення додатку А. Програмне забезпечення для інформаційної підсистеми зоотехнічних параметрів тварин інсталюється на автономному сервері інформаційної підсистеми, робота з програмним забезпеченням здійснюється виключно з локальних або віддалених робочих станцій за допомогою веб-браузерів Mozilla Firefox, Google Chrome, або Internet Explorer. Локальні робочі станції з’єднуються з сервером інформаційної

системи тваринницької ферми через комп’ютерну мережу, віддалені робочі станції з’єднуються з сервером через мережу Internet, у цьому випадку сервер повинен мати статичну IP-адресу.

За допомогою програмного модуля ручного введення інформації оператор здійснює занесення до бази даних такої інформації про тварину: стадний номер, номер стада, номер тварини у національній базі даних, результати контрольних доїнь, результати вимірювання температури тіла та ваги, інформацію про батьків тварини, номер групи, у якій знаходиться тварина, номер транспондера, породу, параметри екстер’єру та його вади, кличку, дату народження, дати отелів, дати ветеринарних досліджень та їх результати, дати та параметри штучних запліднень та абортів, інформацію про лікування та неможливість доїння, інформацію про необхідність розколу, коментарі та іншу індивідуальну інформацію. Окрім цього, вручну вводиться певна інформація, яка необхідна для управління стадом або групою тварин. До такої інформації відноситься 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 заплідненням, кількість днів між абортом та заплідненням, кількість днів між ректальним дослідженням, що має негативні результати, та заплідненням, кількість днів між заплідненням та початком підготовки до запуску. Також забезпечується запис у відповідну базу даних інформації про державу, область, район, населений пункт, поштову адресу, назву ферми, особисті та контактні дані директора, головного зоотехніка, осіб, які відповідають за експлуатацію системи, ідентифікатори операторів, що мають дозвіл на роботу у системі, деякі інші параметри. Нижче наведено фрагмент лістингу програмного забезпечення модуля ручного введення інформації.

```
$cw_c="Картка племінної тварини";
$title="Інтернет-Ферма: ".$cw_c;
```

```
$cowc_prep=$_GET[cowc_prep];
$cowc_buf=$_GET[cowc_buf];
$cowc_cancel=$_GET[cowc_cancel];
//echo "buttons: $cowc_prep $cowc_buf $cowc_cancel<br><br>";
//Dbase_connect(); Dbase_select();
if ( strlen( $local_id )<8 ) {
        $local_id=md5( uniqid( rand(), true ));
        CookieSet( "_ccwid", "$local_id" );
} else {
        $local_id1=Var_FromDb( "$local_id", "0", $userCoo );
}
Var_ToDb( "varchar", "tmp_".$local_id, $local_id, $userCoo );

$buf_cows="tmp_".$local_id."c";
$buf_oxes="tmp_".$local_id."o";

$modif_date=date( "Y-m-d" ); $modif_time=date( "H:i:s" );
$cow_id=$_GET[cow_id];
$changed_input_name=$_GET[changed_input_name];
$random_key=$_GET[random_key]*1;
$ret0=$_GET[ret0];//link to return to previous page
if ( $ret0=="00" ) $ret_url="../".$hFrm['0500'];
if ( @$cowc_cancel ) {
        mysql_query( "DROP TABLE IF EXISTS $buf_cows" );
        mysql_query( "DROP TABLE IF EXISTS $buf_oxes" );
        Res_Draw( 3, $ret_url, "",
$cw_c.": ".$_13_card_no_changes_done_cap, $php_mm_tip[0] );
} else if ( @$cowc_buf ) {
        Data__GET();
```

```
        Data_UPDA( $buf_cows, $buf_oxes, 1 );
        if ( $changed_input_name=="m_id" ) $res=EquIds( $buf_cows, "mth_id",
$cow_id, $m_id );
        if ( $changed_input_name=="f_id" ) $res=EquIds( $buf_cows, "fth_id",
$cow_id, $f_id );
        if ( $changed_input_name=="mm_id" ) $res=EquIds( $buf_cows, "mth_id",
$m_id, $mm_id );
        if ( $changed_input_name=="fm_id" ) $res=EquIds( $buf_cows, "fth_id",
$m_id, $fm_id );
        if ( $changed_input_name=="mf_id" ) $res=EquIds( $buf_oxes, "mth_id",
$f_id, $mf_id );
        if ( $changed_input_name=="ff_id" ) $res=EquIds( $buf_oxes, "fth_id", $f_id,
$ff_id );
        if ( $changed_input_name=="mmm_id" ) $res=EquIds( $buf_cows, "mth_id",
$mm_id, $mmm_id );
        if ( $changed_input_name=="fmm_id" ) $res=EquIds( $buf_cows, "fth_id",
$mm_id, $fmm_id );
        if ( $changed_input_name=="mfm_id" ) $res=EquIds( $buf_oxes, "mth_id",
$fm_id, $mfm_id );
        if ( $changed_input_name=="ffm_id" ) $res=EquIds( $buf_oxes, "fth_id",
$fm_id, $ffm_id );
        if ( $changed_input_name=="mmf_id" ) $res=EquIds( $buf_cows, "mth_id",
$mf_id, $mmf_id );
        if ( $changed_input_name=="fmf_id" ) $res=EquIds( $buf_cows, "fth_id",
$mf_id, $fmf_id );
        if ( $changed_input_name=="mff_id" ) $res=EquIds( $buf_oxes, "mth_id",
$ff_id, $mff_id );
        if ( $changed_input_name=="fff_id" ) $res=EquIds( $buf_oxes, "fth_id",
$ff_id, $fff_id );
```

```
        if ( $userdata_invalid==0 ) {
                $update_query="UPDATE $cows SET
                 modif_uid='$userCoo', modif_date='$modif_date'
                 WHERE id='$cow_id'";
                $update_err=Sql_query( $update_query );
                Data_UPDA( $buf_cows, $buf_oxes, 0 );
                Data_UPDA( $cows, $oxes, 0 );
                $xa='m'; $xb=$m_id; $res11=Adata_UPDA( $cows, $xb ); $res11_r=8;
$res11_c=8;
                if ( $res11==0 ) $aq="UPDATE $c_f2ml SET id='$xb'";
                else {
                        $aq="INSERT INTO $c_f2ml (`id`";
                        for ( $iii=1; $iii<=$res11_r; $iii++ ) for ( $jjj=1; $jjj<=$res11_c;
$jjj++ ) {
                                $idxx=Int2StrZ( $iii, 2 ).Int2StrZ( $jjj, 2 ); $el=$xa.$idxx;
                                if ( $jjj!=6 & $jjj!=8 ) {
                                        $aq=$aq.", `a1_".$idxx."`";
                                }
                        }
                        $aq=$aq.") VALUES ('".$xb."' ";
                }
                for ( $iii=1; $iii<=$res11_r; $iii++ ) for ( $jjj=1; $jjj<=$res11_c; $jjj++ )
{
                        $idxx=Int2StrZ( $iii, 2 ).Int2StrZ( $jjj, 2 ); $el=$xa.$idxx;
                        $a1[$iii][$jjj]=$_GET[$el];
                        if ( trim( $a1[$iii][$jjj])=="" ) $a1[$iii][$jjj]=-1;
                        $a1[$iii][$jjj]=Str2Int( $a1[$iii][$jjj] );
                        if ( $jjj!=6 & $jjj!=8 ) {
```

```
                    if ( $res11==0 ) $aq=$aq.", a1_".$idxx."='".$a1[$iii][$jjj]."'
";
                    else $aq=$aq.", '".$a1[$iii][$jjj]."' ";
                }
            }
```

За допомогою модуля взаємодії з обладнанням, інформація про зоотехнічні параметри, яка отримана за допомогою відповідних складових елементів інформаційної підсистеми, записується в базу даних в автоматичному режимі. Окрім того, за допомогою цього модуля забезпечується передача службових даних від сервера підсистеми до її складових елементів з метою забезпечення певних налаштувань, встановлення необхідних режимів роботи та управління, відображення необхідної інформації. Інформація, обмін якою здійснюється при використанні цього модуля, залежить від структури доїльної установки, способу утримання тварин, технічних можливостей та характеристик обладнання, яке використовується на тваринницькій фермі. У типовій інформаційній системі, яка використовується при безприв’язному утриманні тварин, цей модуль, у програмній взаємодії з інтерфейсним блоком, забезпечує можливість отримання інформації від встановлених на доїльній установці блоків забезпечення процесу доїння про кількість повторних під'єднань доїльних стаканів, наявність випадку холостих доїнь, тривалість холостого доїння, кількість відпадань доїльного апарата, тривалість доїння, час припуску, разовий удій протягом одного доїння, удій у перші три тридцятисекундні інтервали з початку доїння, код транспондера, наявність маститу або його ступінь у чвертях вимені, наявність травмованості тварини, наявність у тварини стану «охоти», мережевий номер, електропровідність молока. До блоку забезпечення процесу доїння, за допомогою цього програмного модуля, забезпечується передача інформації про стадний номер тварини, заборону доїння тварини та інформації про низький удій.

У інформаційних системах для ферм, які використовуються при прив’язному утриманні тварин, вищевказаний модуль, у програмній взаємодії з інтерфейсним блоком, забезпечує отримання інформації про номер лінії стійлової доїльної установки, загальний удій групи тварин у лінії, наявність води в молоці у кожній лінії, середню електропровідність молока у лінії. При взаємодії з блоком управління селекційними воротами, цей модуль забезпечує отримання коду транспондера тварини та передачу даних про необхідність розколу. При взаємодії з блоком управління зчитувачем транспондерів за допомогою цього модуля отримується код транспондера тварини та їх кількість у потоці. При застосуванні блока управління чесальною установкою отримується код транспондера тварини та інформація про тривалість та кількість разів користування чесальною установкою.

Модуль налаштувань та конфігурації забезпечує налаштування програмного забезпечення інформаційної підсистеми для функціонування з різними типами доїльних установок та різними варіантами структур інформаційних систем. За допомогою цього модуля програмне забезпечення налаштовується відповідно до типу доїльної установки та способу утримання, кількості робочих траншей групової установки, загальної кількості каналів обміну даними, кількості каналів обміну даними для роботи з обладнанням у кожній траншеї, типу блока забезпечення процесу доїння, загальної кількості блоків забезпечення процесу доїння, кількості блоків забезпечення процесів доїння в одній траншеї, розподілу мережевих номерів блоків забезпечення процесів доїння та іншого обладнання, кількості робочих траншей, що обслуговуються одним доярем, кількості доярів, кількості селекційних воріт та чесальних установок, часу початку доїнь та їх кількості. Також за допомогою цього модуля встановлюється тип та номер порта для обміну даними з сервером, часовий інтервал між опитуваннями функціональних складових елементів інформаційної системи, швидкість обміну даними.

Модуль обробки інформації забезпечує проведення інтерполяції та апроксимації результатів вимірювання параметрів технологічного процесу виробництва молока, розрахунок залежностей параметрів технологічного процесу виробництва молока від дня лактації, в тому числі лактаційних кривих, здійснення вимірювального контролю параметрів технологічного процесу виробництва молока, аналіз отриманих результатів вимірювання та вимірювального контролю з метою оцінки стану складових елементів технологічного процесу виробництва молока та небезпеки відхилення їх стану від норми, встановлення тварин, у яких виявлено підозру на певні захворювання, прогнозування стану тварин.

За допомогою модуля формування звітів здійснюється автоматичне створення комплексу звітів, структура яких визначається оператором системи. Можуть формуватися звіти за окремими тваринами, за тваринами з негативною динамікою стану, за тваринами з підозрами на захворювання, звіти з результатами вимірювального контролю параметрів технологічного процесу виробництва молока. Також можливе формування звітів з практично будь-якою конфігурацією даних за стадом, групою тварин, фермами, робочим траншеями, часовими проміжками. Модуль взаємодії з стороннім програмним забезпеченням здійснює експорт отриманих в результаті роботи програмного забезпечення даних до електронних таблиць, та популярного програмного комплексу «Uniform Agri».

3.2. Розробка програмного забезпечення для мікроконтролера блока забезпечення процесу доїння

Алгоритм роботи блока забезпечення процесу доїння, який входить до складу інформаційної підсистеми зоотехнічних параметрів тварин, полягає в наступному. Після підготовки тварини та вдягання доїльних стаканів, дояр натискає кнопку «Пуск». За сигналом, який надходить на дискретний вхід мікроконтролера після натискання цієї кнопки, починається вимірювання

параметрів молоковіддачі. Результат вимірювання поточного удою відображається на цифровому світлодіодному індикаторі. При використанні ковшового сенсора автоматично виявляється наявність «нормально замкненого» та «нормально розімкненого» геркона, програмно реалізовано захист від «дрижання контактів». При використанні сенсора рівня молока у молокоприймальній камері доїльного апарата, двонаправлений обмін даними у послідовному форматі між сенсором та мікроконтролером забезпечується за допомогою інтегрованого у мікроконтролер модуля UART. Через одну хвилину після початку доїння блок забезпечення процесу доїння починає здійснювати вимірювальний контроль миттєвої інтенсивності молоковіддачі, у момент її зниження нижче ніж 0,2 л/хв, за допомогою світлодіодного індикатора необхідності зняття доїльного апарата формується відповідний сигнал дояру. Передавання інформації про удій до засобу вимірювання кількості молока забезпечується за допомогою «струмової петлі» та дводротової лінії передачі даних з гальванічною розв'язкою. У додатку Б наведено лістинг програмного забезпечення для мікроконтролера на мові програмування С яке створено у пакеті програм AVR Studio. Нижче наведено фрагмент лістингу програмного забезпечення для мікроконтролера ATTINY2313 виробництва компагії Microchip, під управлінням якого працює блок забезпечення процесу доїння, який входить до складу інформаційної підсистеми зоотехнічних параметрів тварин. Цей фрагмент відповідає за взаємодію з зчитувачем системи радіочастотної ідентифікації тварин.

```
void AutoReaderCheck( void )
{
  IE2 &=~ URXIE1;             //Disable READER USART interrupt
  P5OUT |= BIT3; // reset reader
  while (CounterA < 10); // pause
  IE2 |= URXIE1;              //Enable READER USART interrupt
  P5OUT &=~ BIT3; // start reader
```

```
  while ((CounterA < 58) & (!RFID_Ready));
  IE2 &=~ URXIE1;             //Disable READER USART interrupt
  P5OUT &=~ BIT3; // reset reader again
  if (!RFID_Ready) MilkCnt = MilkCnt + 10; // test error;
}
void AutoCheckComputerLine( void )
{ unsigned char k, badblock;
  badblock = 0;
  for (k = 0; k <= 50; k++)
  {
    RxBufPtr0 = 0;
    TXBUF0 = k;
    while ((UTCTL0&0x01)==0);
    while ((RxBufPtr0 == 0) & (RespRepeater == 0));
    if (UART0RxBuff[0] != k) badblock++; // test error;
  }
  if (badblock > 10) MilkCnt = MilkCnt + 100; // test error;
}
```

Висновок

У бакалаврській кваліфікаційній роботі розроблено інформаційну підсистему зоотехнічних параметрів тварин. У якості об’єкта дослідження обрано процес та концепція побудови інформаційної підсистеми зоотехнічних параметрів тварин. Предметом дослідження є методи інформатизації підприємств з виробництва коров’ячого молока.

Розроблена підсистема може бути використана на підприємствах з виробництва коров’ячого молока і забезпечує отримання, обробку, представлення у необхідному вигляді та безперервний облік інформації про комплекс зоотехнічних параметрів дійних тварин з метою ефективного управління стадом. Інформаційна підсистема забезпечує передачу інформації

про зоотехнічні параметри тварин до персонального комп'ютера та представлення отриманої інформації у необхідному для зоотехніка вигляді.

У першому розділі проведено аналіз стану проблеми та завдань дослідження, здійснено огляд існуючих засобів отримання інформації про зоотехнічні параметри тварин, обрано об'єкт дослідження та сформульовані задачі дослідження.

У другому розділі проведено огляд та обгрунтування вибору архітектури інформаційної підсистеми зоотехнічних параметрів тварин, розглянуто технологію розробки програмного забезпечення для мікроконтролерів AVR Studio та взаємодію PHP та MySQL, архітектуру підсистеми.

У третьому розділі розроблено програмне забезпечення для персонального комп'ютера інформаційної підсистеми зоотехнічних параметрів тварин та програмне забезпечення для мікроконтролера блоку забезпечення процесу доїння на мові програмування С.

Розроблена інформаційна підсистема має широкі межі застосування у промисловому виробництві молока та наукових, в тому числі зоотехнічних, дослідженнях.

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. Звєрєва, Г. В. Рекомендації з профілактики неплідності худоби / Г. В. Звєрєва, В. А. Яблонський, М. В. Косенко [та ін.]. – Львів, 2001. – 18 с.

6. Пешук, Л. В. Проблема маститу в стадах великої рогатої худоби молочного напряму / Л. В. Пешук // Вісник аграрної науки. – 2001. – № 9. – с. 32–35.

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

8. 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.

9. Nielen, M. Electrical Conductivity of Milk: Measurement, Modifiers, and Meta Analysis of Mastitis Detection Performance / M. Nielen, H. Deluyker, Y. H. Schukken, A. Brand // Journal of Dairy Science. – 1992. – V. 75. – № 2. – P. 606–614.

10. 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.

11. 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.

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

13. Метатрон 21. Нове покоління. Високотехнологічний стрибок в доїльній техніці. Рекламний проспект. / GEA Group. – 2013. – 7 с.

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

15. Kucheruk, V. Yu. Development of the mathematical model of light flux intensity sensor based on the photodiode and operational amplifier / V. Kucheruk, P. Kulakov, Yu. Moskvichova, N. Gadzhula, P. Kissabekova, A. Kulakova, D. Karabekova, A. Khassenov // Bulletin of the Karaganda University. «Physics» series., ISSN 2518-7198, № 1(97) / 2020, p. 50 – 57, DOI: 10.31489/2020Ph1/50-57;