

**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 ідентифікації**  
**тварин для групової доїльної установки**

Виконав – студент 4 курсу, групи 41-кн  
освітньої програми "Комп'ютерні науки"  
спеціальності 122 Комп'ютерні науки

**Півень Ігор Валентинович**

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 засобів ідентифікації тварин<br>на групових доїльних установках.....                                        | 16 |
| 1.3. Постановка задачі .....                                                                                                 | 21 |
| РОЗДІЛ 2. ВИБІР ТЕХНОЛОГІЙ ТА АРХІТЕКТУРА ПІДСИСТЕМИ .....                                                                   | 22 |
| 2.1. Обґрунтування вибору мови програмування для створення<br>програмного забезпечення для сервера підсистеми.....           | 22 |
| 2.2. Технологія розробки Code Composer Studio .....                                                                          | 27 |
| 2.3. Розробка архітектури інформаційної підсистеми<br>ідентифікації тварин для групової доїльної установки.....              | 32 |
| РОЗДІЛ 3. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ<br>ІНФОРМАЦІЙНОЇ ПІДСИСТЕМИ .....                                                | 36 |
| 3.1. Розробка програмного забезпечення інформаційної підсистеми<br>ідентифікації тварин для групової доїльної установки..... | 36 |
| 3.2. Розробка програмного забезпечення для мікроконтролера<br>блока керування зчитувачем транспондерів .....                 | 39 |
| ВИСНОВОК.....                                                                                                                | 41 |
| СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ .....                                                                                         | 42 |
| ДОДАТОК А.....                                                                                                               | 44 |
| ДОДАТОК Б .....                                                                                                              | 51 |

## **Перелік умовних позначень, скорочень і термінів**

|      |                                         |
|------|-----------------------------------------|
| ПК   | Персональний комп'ютер                  |
| БЗОД | Блок забезпечення обміну даними         |
| ЗТ   | Зчитувач транспондерів                  |
| БКЗТ | Блок керування зчитувачем транспондерів |
| БКПТ | Блок контролю проходження тварини       |
| БД   | База даних                              |
| ЗВ   | Засіб вимірювання                       |
| ПЗ   | Програмне забезпечення.                 |
| ПД   | Програмний додаток.                     |
| ПЗП  | Постійний запам'ятовуючий пристрій      |
| ІС   | Інформаційна система                    |
| ООП  | Об'єктно-орієнтоване програмування      |

## Вступ

Системи ідентифікації тварин є важливим елементом технологічного процесу виробництва коров'ячого молока, функціонування засобів вимірювального контролю параметрів технологічного процесу без систем ідентифікації відбувається із значним обмеженням функціональності. Завданням систем ідентифікації тварин є встановлення відповідності між номером тварини у стаді, та результатами вимірювання і контролю зоотехнічних параметрів, які пов'язані з цією твариною [1, 2]. Внаслідок виникнення помилок ідентифікації може бути втрачена інформація про одну або декілька тварин, в результаті чого знижується ефективність функціонування тваринницької ферми. Виходячи з цього, до систем ідентифікації тварин висуваються високі вимоги стосовно надійності їх роботи [3]. Системи ідентифікації, які використовуються у технологічному процесі виробництва коров'ячого молока, мають специфічні особливості, які визначаються способом утримання тварин та конструктивними особливостями доїльної установки, параметри цих систем регулюються міжнародними стандартами [4 - 6]. Системи ідентифікації входять до складу доїльних комплексів, які пропонують практично усі світові виробники [7 - 9].

### Актуальність

Використання інформаційних технологій та впровадження їх у промисловий сектор, як слідує з світової практики, стимулює підвищення продуктивності праці, сприяє зменшенню завантаження працівників, збільшує ефективність технологічних процесів. Інформаційна підсистема ідентифікації тварин призначена для впровадження на молочних тваринницьких фермах, що використовують групові доїльні установки та засоби ідентифікації тварин. Виходячи з цього, розвиток теорії та практики впровадження засобів інформатизації у технологічний процес виробництва молока є актуальним завданням, тим більше, що в Україні відсутні у серійному виробництві сучасні

засоби ідентифікації тварин під час руху та інформаційні підсистеми для забезпечення їх роботи.

**Об’єкт дослідження** – процес, концепція та архітектура побудови засобів ідентифікації тварин та інформаційних підсистем ідентифікації для групової доїльної установки.

**Предмет дослідження** – методи інформатизації молочних ферм обладнаних засобами ідентифікації тварин під час доїння.

### **Мета дослідження**

Метою та завданням бакалаврської кваліфікаційної роботи є концептуальна розробка інформаційної підсистеми ідентифікації тварин для групової доїльної установки, архітектури інформаційної підсистеми та її компонентів, розробка програмного забезпечення для інформаційної підсистеми та програмного забезпечення для мікроконтролера блока управління зчитувачем транспондерів.

Протягом виконання кваліфікаційної бакалаврської роботи необхідно провести аналіз наукової та технічної літератури, технічної документації, міжнародних стандартів у галузі ідентифікації тварин, розробити архітектуру інформаційної підсистеми, структурні схеми її окремих компонентів та програмних модулів.

## РОЗДІЛ 1. АНАЛІТИЧНИЙ ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ

### 1.1 Об'єкт та предмет дослідження

На рис. 1.1 наведено класифікацію систем ідентифікації тварин для тваринницьких ферм [10].

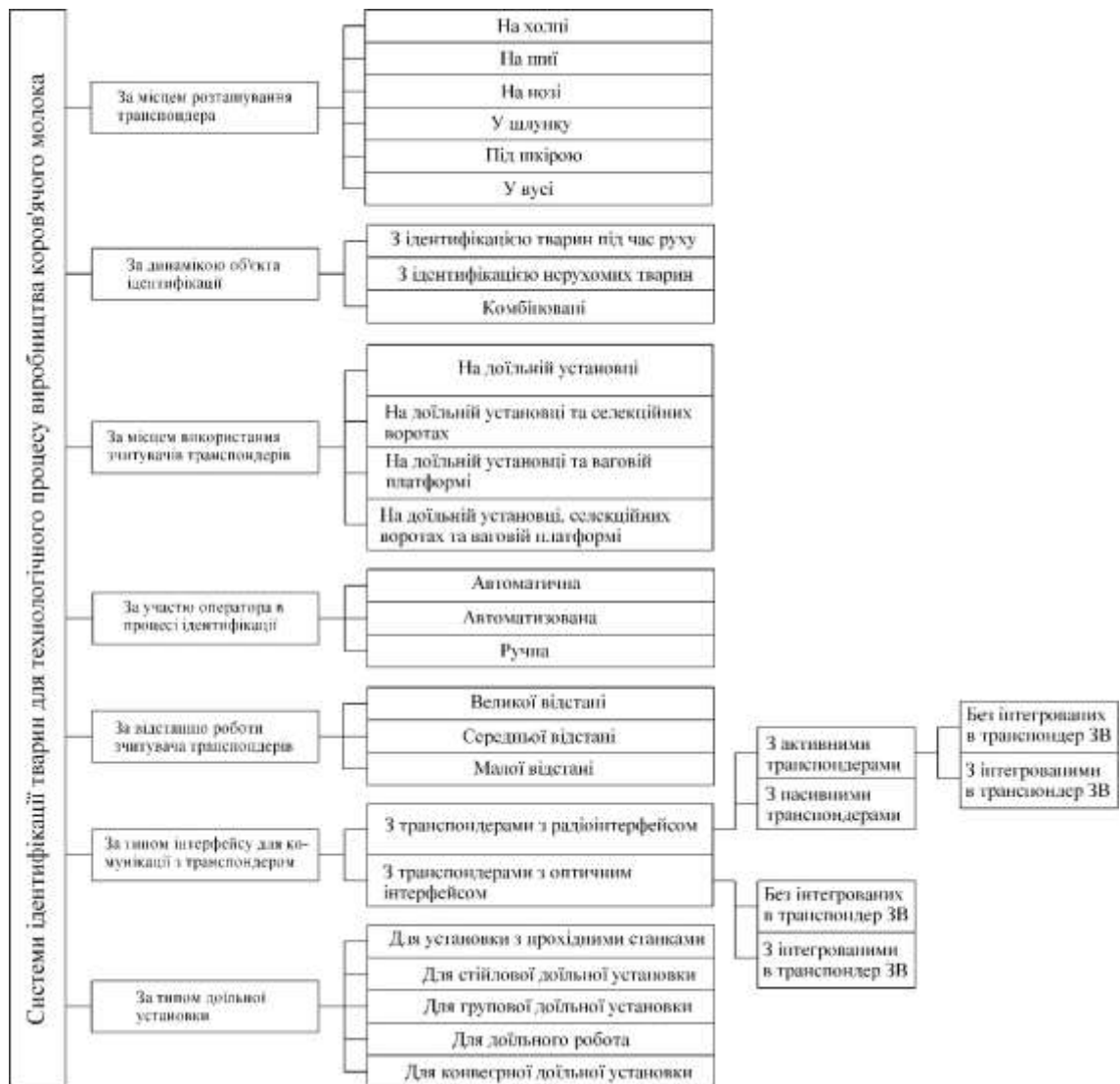


Рисунок 1.1 – Класифікація систем ідентифікації тварин для тваринницьких ферм

Системи ідентифікації тварин для тваринницьких ферм за місцем розташування транспондера поділяються на такі, у яких транспондер закріплений на шії, холсті, вусі, нозі, шлунку, або під шкірою тварини.

За динамікою об'єкта ідентифікації, системи ідентифікації тварин для тваринницьких ферм поділяються на системи з ідентифікацією тварин під час руху, на системи з ідентифікацією нерухомих тварин, та на комбіновані системи. При використанні селекційних воріт тварини ідентифікуються під час руху. На стійлових доїльних установках, доїльних установках з прохідними станками, при використанні доїльних робіт здійснюється ідентифікація нерухомих тварин. На групових доїльних установках, конвеєрних доїльних установках та на вагових платформах може здійснюватись ідентифікація рухомих та нерухомих тварин. Якщо на доїльній установці та ваговій платформі здійснюється ідентифікація нерухомих тварин, а селекційні ворота на фермі не використовуються, то така система ідентифікації відноситься до систем з ідентифікацією нерухомих тварин. У випадку, якщо на доїльній установці, ваговій платформі та селекційних воротах здійснюється ідентифікація рухомих тварин, то така система відноситься до систем з ідентифікацією тварин під час руху. Якщо на фермі здійснюється ідентифікація тварин як під час руху, так і в нерухомому положенні, то така система відноситься до комбінованих систем ідентифікації.

За місцем використання зчитувача транспондерів (ЗТ) у системах ідентифікації тварин, вони поділяються на такі, у яких здійснюється ідентифікація на доїльній установці, на селекційних воротах та доїльній установці, на доїльній установці та ваговій платформі, на доїльній установці та селекційних воротах і ваговій платформі [11].

За участю оператора в процесі ідентифікації, системи ідентифікації тварин поділяються на автоматичні, автоматизовані, ручні. При ручній ідентифікації номер тварини вводиться вручну до блока забезпечення процесу доїння. При автоматизованій ідентифікації кожний блок забезпечення процесу доїння обладнаний ЗТ, який перед початком підготовки тварини до доїння дояр вручну наближає до транспондера. При використанні автоматичних систем, ідентифікація тварин відбувається без участі оператора.

За відстанню роботи ЗТ системи ідентифікації тварин поділяються на малої відстані, середньої відстані, великої відстані. ЗТ малої відстані використовуються при автоматизованій ідентифікації у доїльних станках. ЗТ середньої відстані використовуються при автоматичній ідентифікації нерухомих тварин, та іноді, для ідентифікації рухомих тварин, ЗТ великої відстані – при автоматичній ідентифікації рухомих та нерухомих тварин.

За типом інтерфейсу для комунікації з транспондером системи ідентифікації тварин поділяються на системи з транспондерами з оптичним інтерфейсом та системи з транспондерами з радіоінтерфейсом. Конструкція кріплення транспондерів з оптичним інтерфейсом така, що трансивер завжди знаходиться зверху тварини на її холці, а ЗТ знаходиться у доїльному станку над твариною. При входженні тварини до доїльного станка транспондер активізується за сигналом від ЗТ, після чого здійснюється обмін даними. Усі типи транспондерів з оптичним інтерфейсом активні, тобто потребують автономних елементів живлення, які треба періодично замінювати. Найбільш розповсюдженими у теперішній час є системи ідентифікації тварин з використанням транспондерів з радіоінтерфейсом, так звані системи радіочастотної ідентифікації [11]. У таких системах обмін даними між ЗТ та транспондером здійснюється за допомогою радіозв'язку. Використовуються як активні, так і пасивні транспондери для систем радіочастотної ідентифікації. Завдяки наявності автономного джерела живлення, при використанні активних транспондерів досягається велика відстань зчитування і можливість інтеграції ЗВ активності та румінації. Але найбільш розповсюдженими є засоби радіочастотної ідентифікації тварин з пасивними транспондерами. До складу такого транспондера входить індуктивна антена, трансивер та мікропроцесор. Коли транспондер знаходиться у робочій зоні ЗТ, його коливальна система накопичує енергію, якої достатньо для приведення у дію мікропроцесора і трансивера. Мікропроцесор генерує у відповідь на запит ЗТ пакет даних з

кодом транспондера, ЗТ дешифрує код транспондера, після чого він передається до сервера підсистеми. Сервер, на основі отриманого коду транспондера, визначає номер тварини у стаді, та повертає його до блока забезпечення процесу доїння для візуалізації.

Системи ідентифікації тварин з транспондерами з оптичним інтерфейсом та системи радіочастотної ідентифікації з активними транспондерами поділяються на системи з транспондерами з інтегрованими засобами вимірювання (ЗВ) та без інтегрованих ЗВ. Системи з транспондерами без інтегрованих ЗВ призначені виключно для визначення стадного номера тварини. У системах з транспондерами з інтегрованими ЗВ здійснюється вимірювання активності тварини та румінації.

За типом доїльної установки системи ідентифікації тварин можуть бути призначені для конвеєрної, групової, стійлової доїльної установки, для доїльної установки з прохідними станками, для доїльного робота.

На доїльних установках з прохідними станками, на стійлових доїльних установках та у складі доїльних роботів здійснюється ідентифікація тільки нерухомих тварин. Використання систем ідентифікації тварин нерухомих тварин обмежується економічними факторами, тому що собівартість таких систем значно більша, ніж систем ідентифікації тварин під час руху. Також такі системи можуть бути застосовані на конвеєрних і групових доїльних установках, на вагових платформах. Імовірність помилки ідентифікації нерухомих тварин значно менша, ніж помилки ідентифікації під час руху.

На рис. 1.2 наведено варіант структурної схеми автоматичної системи ідентифікації нерухомих тварин у доїльних станках на основі ЗТ середньої або великої відстані. У цьому варіанті системи ідентифікації кожний доїльний станок обладнаний ЗТ великої або середньої відстані. Після входження тварини у станок здійснюється її автоматична ідентифікація, після чого починається процес доїння. Код транспондера дешифрується за допомогою відповідного

блока керування зчитувачем транспондерів (БКЗТ), який в свою чергу передає його до блока забезпечення обміну даними (БЗОД).

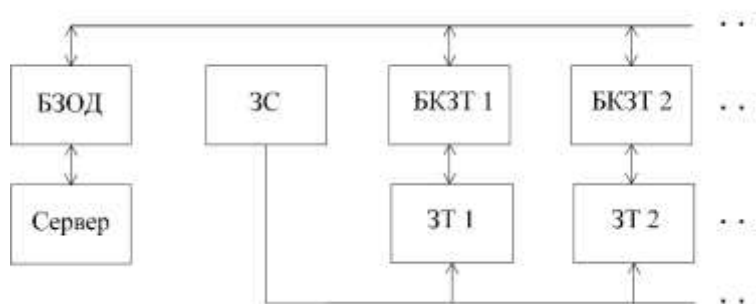


Рисунок 1.2 – Структурна схема автоматичної системи ідентифікації нерухомих тварин у доїльних станках на основі ЗТ середньої або великої відстані

БЗОД передає отриманий код до сервера системи, який визначає стадний номер тварини та передає його до відповідних блоків забезпечення процесу доїння. Отриманий стадний номер візуалізується на індикаторі блока забезпечення процесу доїння, дояр може перевірити його відповідність реальному номеру тварини, та у випадку виникнення помилки ідентифікації вручну скорегувати. Якщо автоматична ідентифікація тварини неможлива внаслідок втрати або виходу з ладу транспондера, дояр має можливість ввести номер тварини вручну. Якщо ідентифікація тварини не відбулася, то результати контролю відповідних параметрів будуть віднесені до умовної тварини. У ЗТ великої та середньої відстані потужність радіовипромінювання може досягати десятків Вт. У наведеній системі вони розташовуються на невеликій відстані один від одного і не можуть працювати одночасно внаслідок взаємного впливу. Для усунення взаємного впливу ЗТ у системі використовується засіб синхронізації (ЗС), який в будь-який момент часу забезпечує роботу тільки одного ЗТ, тобто за його допомогою забезпечується послідовна періодична активація усіх ЗТ системи. Якщо використовуються транспондери та ЗТ з оптичним інтерфейсом, то ЗС не потрібний.

Структурна схема іншого варіанта автоматичної системи ідентифікації нерухомих тварин у доїльних станках, при використанні радіочастотної

ідентифікації на основі ЗТ великої або середньої відстані та локальних антен, наведена на рис. 1.3.

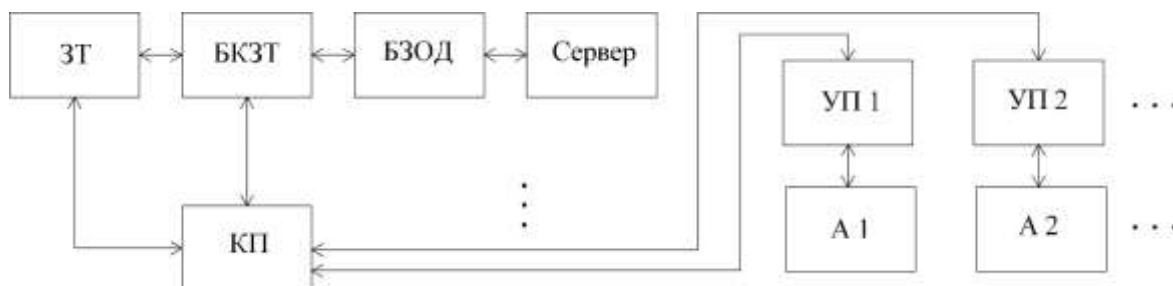


Рисунок 1.3 – Структурна схема автоматичної системи радіочастотної ідентифікації нерухомих тварин при використанні ЗТ великої або середньої відстані та локальних антен

У цьому варіанті системи кожний доїльний станок обладнаний локальною антеною (А). Кожна антена з'єднана з узгоджувальними пристроями (УП), які в свою чергу, за допомогою коаксіальних кабелів з'єднані з комутаційним пристроєм (КП). УП необхідні для узгодження кіл ЗТ з параметрами антенного тракту, які значно залежать від довжини коаксіальних кабелів. За командами, які надходять від БКЗТ, КП здійснює послідовне під'єднання УП до вихідних та вхідних кіл ЗТ для зчитування кодів транспондерів в усіх доїльних станках. З виходу КП сигнал надходить на вхід ЗТ, який визначає код транспондера та передає його до БКЗТ. БКЗТ за посередництва БЗОД передає код разом з номером доїльного станка до сервера, який визначає стадний номер тварини та повертає його до відповідного БЗПД.

На рис. 1.4 наведено структурну схему автоматизованої системи радіочастотної ідентифікації нерухомих тварин на основі ЗТ малої відстані. У цій системі кожний блок забезпечення процесу доїння обладнаний ЗТ, який має відстань зчитування не більшу десяти - п'ятнадцяти сантиметрів та невелику потужність радіовипромінювання. Невелика потужність зумовлює можливість одночасної роботи ЗТ внаслідок відсутності взаємного впливу. Ідентифікація тварин здійснюється автоматизовано, дояр перед початком підготовки тварини

підносить ЗТ до транспондера, який розташований на задній нозі. Код транспондера за допомогою БКЗТ передається до сервера системи за посередництва БЗОД.

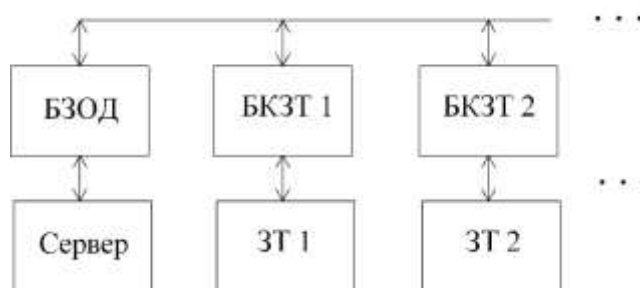


Рисунок 1.4 – Структурна схема автоматизованої системи радіочастотної ідентифікації нерухомих тварин на основі ЗТ малої відстані

Сервер визначає стадний номер тварини та повертає його до відповідного БЗПД. Отриманий від сервера стадний номер відображається на індикаторі БЗПД, дояр може перевірити його відповідність реальному номеру, та при виявленні помилки ідентифікації вручну скорегувати. При неможливості ідентифікації дояр вводить номер тварини вручну, за допомогою клавіатури БЗПД. Якщо ідентифікація взагалі неможлива, то результати контролю відповідних параметрів будуть віднесені до умовної тварини. При використанні такої системи збільшується трудомісткість роботи дояра та збільшується тривалість роботи доїльної установки.

## 1.2 Особливості засобів ідентифікації тварин на групових доїльних установках

Ідентифікація тварин під час руху відбувається виключно в автоматичному режимі з використанням ЗТ середньої та великої відстані. Цей спосіб ідентифікації може бути використаний на вході на конвеєрну або групову доїльну установку, на селекційних воротах та автоматичній ваговій платформі. ЗТ середньої відстані можуть використовуватись тільки якщо транспондери закріплені на нозі, шиї або у вусі тварини, у цьому випадку, вони

повинні бути закріплені з тієї сторони тварини, з якої знаходиться ЗТ. На рис. 1.5 наведено варіант структурної схеми системи ідентифікації тварин під час руху для групових доїльних установок.

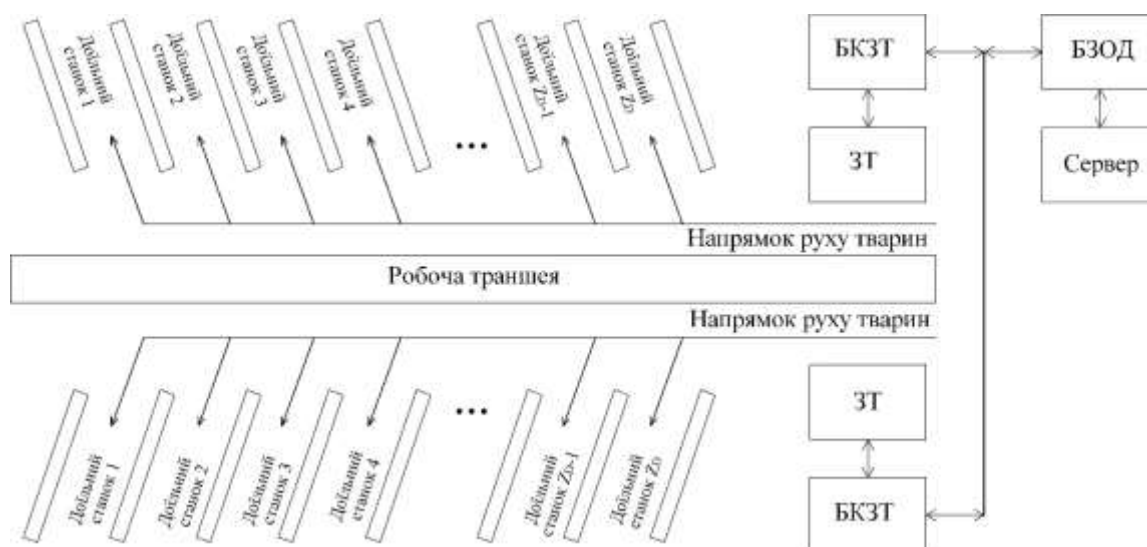


Рисунок 1.5 – Структурна схема системи ідентифікації тварин під час руху для групових доїльних установок

До групових доїльних установок відносяться установки типу «Паралель» та «Ялинка», різниця між ними полягає у різному розташуванні тварин у доїльних станках. Групова доїльна установка складається з певної кількості робочих траншей, сторони кожної траншеї обладнані стаціонарними доїльними станками. Вхід та вихід тварин на сторону траншеї здійснюється групами. Під час входу тварин, вони на шляху до доїльних станків проходять через робочу зону ЗТ. Конструктивно, прохід виконаний так, що тварина яка увійшла першою, буде знаходитись в першому доїльному станку, друга – в другому, і т. д. При проходженні тварин через робочу зону ЗТ коди транспондерів дешифруються за допомогою БКЗТ та передаються до БЗОД. БЗОД передає їх до сервера системи, де визначаються номери тварин у стаді, які відповідають кодам транспондерів, та порядок їх розташування у доїльних станках.

При певних умовах, на групових доїльних установках, при використанні систем ідентифікації тварин під час руху, можливе виникнення помилок ідентифікації, які зумовлені такими причинами. Ідентифікація може не

відбутися, або відбутися з помилкою, внаслідок біологічних особливостей руху тварин (наприклад, коли тварина увійшла на прохід до доїльних станків, ідентифікувалася, а потім знову вийшла з проходу), порушення регулярності черги, відсутності або втрати транспондера на окремих тваринах. Окрім того, транспондер може бути механічно пошкоджений, не потрапити в робочу зону ЗТ, при використанні радіочастотної ідентифікації знаходитись в положенні, коли площини індуктивних антен ЗТ та транспондера взаємно перпендикулярні. Наслідком виникнення помилки ідентифікації, може бути втрата ідентифікаційної інформації про всіх тварин групи. У наведеному варіанті системи ця проблема або взагалі не вирішується, або вирішується шляхом ручного корегування результатів ідентифікації доярками під час підготовки тварини до доїння.

Для зменшення впливу помилок ідентифікації, у інформаційних підсистемах використовується система ідентифікації тварин під час руху з ідентифікацією проходження тварини, структурну схему якої наведено на рис. 1.6.

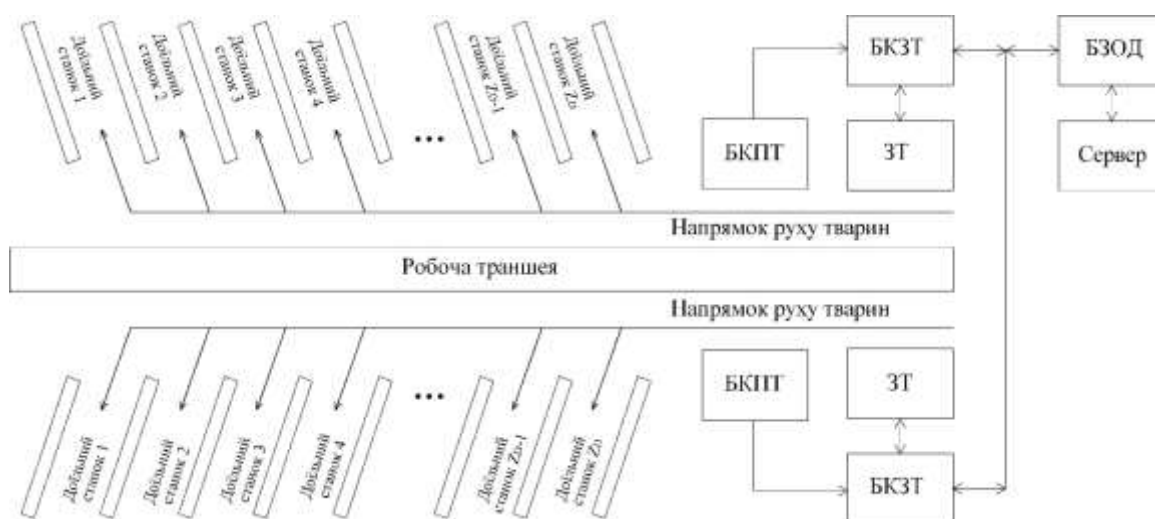


Рисунок 1.6 – Структурна схема системи ідентифікації тварин під час руху для групових доїльних установок з ідентифікацією проходження тварини

До складу наведеної структурної схеми системи ідентифікації входить блок контролю проходження тварини (БКПТ), який реалізується на основі

давачів наявності об'єкта. Принцип його дії заснований на проведенні вимірювального контролю часу знаходження тварини в робочій зоні давача. На шляху до доїльного станка тварина проходить через робочу зону ЗТ, після чого – через робочу зону БКПТ. При входженні кожної тварини групи на доїльну сторону, до БКЗТ надходить код транспондера від ЗТ, і через певний проміжок часу – сигнал від БКПТ. Наявність одного з цих сигналів є ознакою проходження тварини до станка. Таким чином, кожному транспондеру відповідає сигнал БКПТ, порядковий номер якого дорівнює номеру доїльного станка, у якому знаходиться відповідна тварина. Якщо виникла помилка ідентифікації, сигнал від БКПТ все одно надходить до БКЗТ. Ознакою того, що одна з тварин не ідентифікована, є два послідовних сигнали від БКПТ, що надійшли до БКЗТ. Порядковий номер другого сигналу БКПТ є номером доїльного станка, у якому знаходиться не ідентифікована тварина. При виникненні помилок ідентифікації однієї або декількох тварин групи, буде втрачена інформація, яка стосується тільки цих тварин, інформація яка стосується ідентифікованих тварин не буде втрачена.

На рис. 1.7 наведено структурну схему системи ідентифікації тварин під час руху для групових доїльних установок з контролем присутності тварини у доїльному станку. У цій системі кожний доїльний станок обладнаний блоком контролю наявності тварини (БКНТ), для його реалізації використовуються сенсори наявності об'єкта. При входженні кожної тварини групи на доїльну сторону, до БКЗТ спочатку надходить код транспондера від ЗТ, і через певний проміжок часу – сигнал від БКНТ, який містить інформацію про номер доїльного станка. Якщо виникла помилка ідентифікації, сигнал від БКНТ все одно надходить до БКЗТ. Таким чином, ознакою того, що одна з тварин не ідентифікована, є два послідовних сигнали від БКНТ, що надійшли до БКЗТ. Навіть якщо не ідентифікована тварина швидше зайшла до доїльного станка, ніж ідентифікована попередня, алгоритмічно визначається номер станка, в якому знаходиться не ідентифікована тварина. У цій системі також при

виникненні помилок ідентифікації однієї або декількох тварин групи, буде втрачена інформація, яка стосується тільки цих тварин.

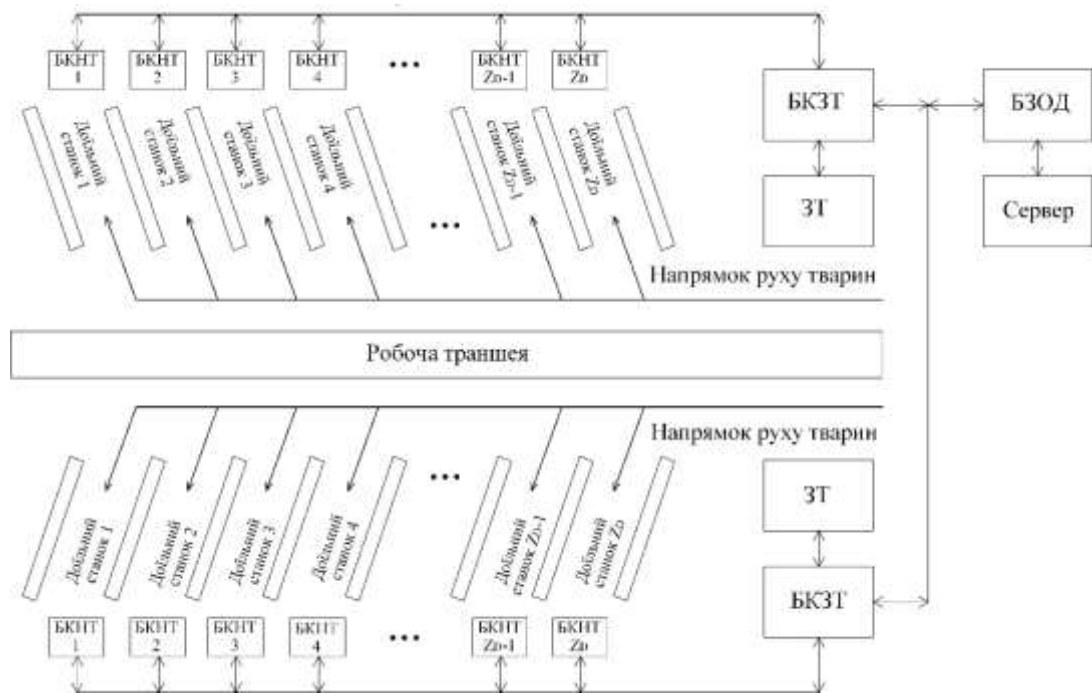


Рисунок 1.7 – Структурна схема системи ідентифікації тварин під час руху для групових доїльних установок з контролем присутності тварини у доїльному станку

На рис. 1.8 наведено структурну схему системи ідентифікації тварин під час руху для селекційних воріт.

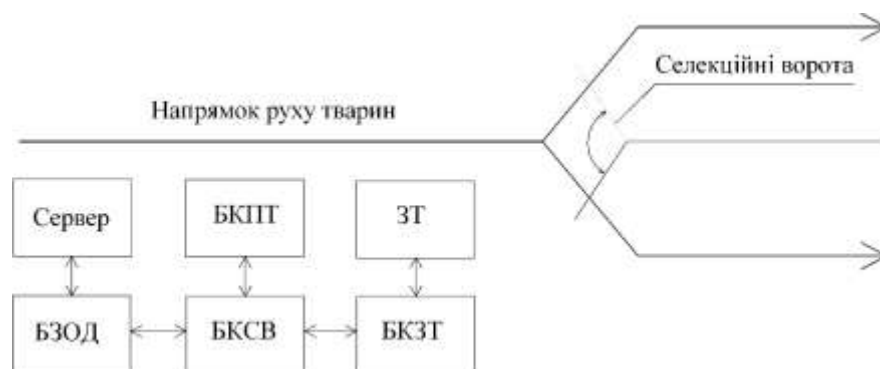


Рисунок 1.8 – Структурна схема системи ідентифікації тварин під час руху для селекційних воріт

Коли тварина входить в прохід селекційних воріт, транспондер попадає в робочу зону ЗТ, його код зчитується і передається до БКЗТ, який, в свою чергу, передає його до блоку керування селекційними воротами (БКСВ). БКСВ за допомогою БЗОД передає отриману інформацію до сервера системи. Сервер на основі отриманого коду транспондера встановлює номер тварини в стаді і необхідний напрямок її руху. Ця інформація сервером передається до БЗОД, який передає її до БКСВ. На основі отриманої інформації БКСВ здійснює управління приводом селекційних воріт для направлення тварини в потрібному напрямку. За допомогою БКПТ здійснюється виявлення тварини при виникненні помилки ідентифікації, у цьому випадку, за сигналом від БКПТ, БКСВ встановлює селекційні ворота у таке положення, щоб не ідентифікована тварина була відокремлена.

### **1.3 Постановка задачі**

Ціллю бакалаврської кваліфікаційної роботи є розробка інформаційної підсистеми ідентифікації тварин для групової доїльної установки, розробка програмного забезпечення для сервера підсистеми, який є її компонентом, програмного забезпечення для мікроконтролера, який є основним елементом одного з найважливіших складових компонентів підсистеми – блока керування зчитувачем транспондерів.

Завданням розробки є концептуальна розробка інформаційної підсистеми ідентифікації тварин для групової доїльної установки, архітектури інформаційної підсистеми та її компонентів, розробка програмного забезпечення для інформаційної підсистеми та програмного забезпечення для мікроконтролера блока управління зчитувачем транспондерів.

## РОЗДІЛ 2. ВИБІР ТЕХНОЛОГІЙ ТА АРХІТЕКТУРА ПІДСИСТЕМИ

### 2.1 Обґрунтування вибору мови програмування для створення програмного забезпечення для сервера підсистеми

Вирішувати задачі створення програмного забезпечення для інформаційних підсистем та баз даних можна на досить широкому діапазоні мов програмування. Серед них, наприклад, C#, C++, Java. Простота вивчення та кодування, а також наявність великої кількості сторонніх бібліотек, призначених для завдань розпізнавання образів, призвели до вибору мови Python і використання інших бібліотек як платформи розробки [12].

Python був створений наприкінці 1980-х років і вперше випущений у 1991 році Гвідо ван Россумом як наступник мови програмування ABC. Випущений у 2000 році Python 2.0 представив нові функції, такі як розуміння списків і система збору сміття з підрахунком посилань, і був припинений у версії 2.7 у 2020 році. Випущений у 2008 році Python 3.0 був основною редакцією мови, але він не був повністю сумісним із попередніми версіями, і більшість коду Python 2 не змінилася, коли він працював на Python 3.

Python – це інтерпретована об'єктно-орієнтована мова програмування високого рівня з динамічною семантикою. Вбудовані високорівневі структури даних у поєднанні з динамічною типізацією та динамічним зв'язуванням роблять його ідеальним для швидкої розробки додатків, а також як мову сценаріїв або з'єднувальної мови для з'єднання існуючих компонентів. Простий у вивченні синтаксис Python підкреслює читабельність, таким чином зменшуючи витрати на обслуговування програми.

Python підтримує модулі та пакунки, які заохочують модульність програми та повторне використання коду. Інтерпретатор Python і обширна стандартна бібліотека вільно доступні у вихідній або двійковій формі на всіх основних платформах, включаючи Windows під час розробки основної програми, і вільно розповсюджуються. На даний момент вона зрівняна з Java як

друга за популярністю мова програмування у світі.

Python є багатопарадигмальною мовою програмування. Об'єктно-орієнтоване програмування та структуроване програмування повністю підтримуються, і багато його функцій підтримують функціональне та аспектно-орієнтоване програмування (включаючи метапрограмування та метаоб'єкти (магічні методи)). Розширення підтримують багато інших парадигм, включаючи розробку контрактів і логічне програмування.

Python використовує динамічну типізацію та комбінацію підрахунку посилок і визначеного циклом збирача сміття для керування пам'яттю. Він також має динамічне визначення імен (пізні зв'язування), яке зв'язує імена методів і змінних під час виконання.

Python був розроблений, щоб забезпечити певну підтримку функціонального програмування в традиції Lisp. Має можливості фільтрації, відображення та скорочення; розуміння списків, словники, набори та генератор виразів. Стандартна бібліотека має два модулі (itertools і functools), які реалізують функціональні інструменти, запозичені з Haskell і Standard ML.

Основна філософія мови коротко викладена в документації Python (PEP 20), яка містить такий девіз:

- красиве краще, ніж потворне;
- явне краще, ніж неявне;
- просте краще складного;
- читабельність важлива.

Python не має всіх своїх функціональних можливостей, вбудованих у його ядро, але розроблений таким чином, щоб бути дуже розширюваним. Ця компактна модульність робить його особливо популярним як спосіб додавання програмованих інтерфейсів до існуючих програм. Бачення Ван Россума щодо невеликої базової мови з великою стандартною бібліотекою та перекладачем, що легко розширюється, виникло через його розчарування ABC, які віддавали перевагу протилежному підходу.

Python був розроблений, щоб бути легкою для читання мовою. Його форматування, як правило, абсолютно безладне, і він використовує англійські ключові слова, тоді як інші мови використовують знаки пунктуації. На відміну від багатьох інших мов, у ній не використовуються фігурні дужки для розділення блоків, а крапка з комою необов'язкова для операторів після. Замість цього, Python використовує пробільний відступ для розділення блоків. Відступ відбувається після певних операторів; зменшення відступу означає кінець поточного блоку. Тому візуальна структура програми точно відображає змістову структуру програми. Цю особливість іноді називають «зовнішніми» правилами, і в деяких інших мовах вона є, але в більшості мов відступ не має семантичного значення.

Функціональне програмування – це парадигма програмування, у якій програми будуються за допомогою застосування та композиції функцій. Це декларативна парадигма програмування, у якій визначення функцій є деревами виразів, кожне з яких повертає значення, а не послідовність імперативних операторів, які змінюють стан програми.

У функціональному програмуванні функції розглядаються як громадяни першого класу, що означає, що їх можна прив'язувати до імен (включаючи локальні ідентифікатори), передавати як аргументи та повертати з інших функцій, як і будь-який інший тип даних. Це дозволяє писати програми в декларативному та складному стилі, де невеликі функції поєднуються в модульний спосіб.

Функціональне програмування іноді трактується як синонім функціонального програмування, підмножини функціонального програмування, яке розглядає всі функції як детерміновані математичні функції або чисті функції. Коли чиста функція викликається з деякими заданими аргументами, вона завжди повертає той самий результат і на неї не впливає будь-який стан змінної чи інші побічні ефекти. Це відрізняється від нечистих підпрограм, поширених у імперативному програмуванні, які можуть мати

побічні ефекти (такі як зміна стану програми або отримання введення користувача). Функціональне програмування виникло в академічних колах і розвинулося з лямбда-числення, яке є формальною обчислювальною системою, заснованою лише на функціях. Функціональне програмування історично було менш популярним, ніж імперативне програмування, але сьогодні багато функціональних мов використовуються в промисловості та освіті.

Найпомітніші особливості функціонального програмування такі:

- Мови функціонального програмування розроблені навколо концепції математичних функцій, які використовують умовні вирази та рекурсію для виконання обчислень.

- Функціональне програмування підтримує функції вищого порядку та функції відкладеного оцінювання.

- Мови функціонального програмування не підтримують керування потоком, наприклад оператори циклу та умовні оператори, такі як оператори If-Else та Switch. Вони безпосередньо використовують функції та виклики функцій.

- Як ООП, функціональні мови програмування підтримують такі популярні концепції, як абстракція, інкапсуляція, успадкування та поліморфізм.

Функціональне програмування має такі переваги:

- Функціональне програмування не має стану, тому немає побічних ефектів і можна писати код без помилок.

- Функціональні мови програмування не мають змінних стану, тому змінити стан не проблема. «Функції» можуть бути запрограмовані паралельно як «інструкції». Такий код підтримує легке повторне використання та перевірку.

- Функціональна програма складається з незалежних блоків, які можуть працювати одночасно. Тому такі програми більш ефективні.

- Функціональне програмування підтримує вкладені функції.

- Функціональне програмування підтримує лінійні функціональні

конструкції, такі як лінійні списки, лінійні карти тощо.

Як недолік, функціональне програмування вимагає багато пам'яті. Оскільки немає поняття стану, нові об'єкти повинні створюватися кожного разу, коли виконується дія. Функціональне програмування використовується в ситуаціях, коли над одним набором даних необхідно виконати багато різних операцій.

Велику стандартну бібліотеку Python часто вважають однією з найбільших переваг Python, оскільки вона надає інструменти, придатні для багатьох завдань. Однак для нашого конкретного завдання нам потрібно використовувати кілька зовнішніх бібліотек і компонентів.

TensorFlow є найбільш широко використовуваною бібліотекою під час розробки додатків. Вона використовується як для досліджень, так і для розробки власних продуктів Google. Основний API для використання бібліотеки реалізований для Python (використовується, як описано вище). Також є реалізації в R, C#, C++, Haskell, Java, Go та Swift.

Це продовження закритого проекту DistBelief. TensorFlow спочатку був розроблений командою Google Brain для внутрішнього використання в Google. У 2015 році система отримала відкритий доступ за ліцензією Apache 2.0 Open.

Важливою частиною бібліотеки є Keras API. Він використовується для реалізації нашої основної моделі прогнозування. Keras – це API високого рівня TensorFlow 2.0: сумісний високопродуктивний інтерфейс для вирішення проблем машинного навчання з акцентом на сучасне глибинне навчання. Він надає фундаментальні абстракції та будівельні блоки для розробки та надання рішень машинного навчання з високою частотою ітерацій. Keras також є дуже гнучкою основою для сучасних дослідницьких ідей. Keras дотримується принципу інкрементальної складності: це полегшує початок роботи, але все ще дозволяє обробляти будь-які складні випадки використання, вимагаючи лише інкрементального навчання на кожному кроці. Бібліотека містить багато

вбудованих наборів плиток, а також підтримує спеціальні набори плиток за допомогою API-ключів.

## **2.2 Технологія розробки Code Composer Studio**

Мікропроцесорні прилади дозволяють вирішувати програмним методом частину завдань, які в звичайних приладах вирішуються апаратними засобами. Використання мікропроцесорів має і недоліки, в першу чергу – складність апаратури і досить висока її вартість. В перспективі, враховуючи швидке зниження цін на елементи мікропроцесорних систем, можна чекати значного здешевлення мікропроцесорних приладів. В деяких випадках швидкодія АЦП і мікропроцесора виявляються недостатніми для проведення необхідних операцій або розрахунків в реальному масштабі часу. При цьому іноді виявляється доцільним застосувати масштабно-часове перетворення досліджуваного сигналу, зробивши його повільнішим. Тому постійно ведуться роботи з підвищення швидкодії і розрядності мікропроцесорів, що випускаються промисловістю, розширюючи тим самим можливості мікропроцесорних засобів. При розробці мікропроцесорних засобів вимірювання найбільш трудомісткою частиною виявляється розробка програмного забезпечення, вартість якого може значно перевищувати вартість апаратних засобів. Проте розробці апаратної частини і, зокрема, вибору мікропроцесора також приділяється багато уваги. А зі зростанням розрядності та обчислювальної потужності мікропроцесорів багато уваги приділяється методам зниження споживаної потужності мікропроцесорними засобами.

Зростаючий попит на продукцію з автономним (батарейним) живленням викликав необхідність створення мікроконтролерів з низьким і наднизьким енергоспоживанням. Компанія Texas Instruments розробила і серійно випускає сім'ю мікроконтролерів MSP430, яка задовольнить вимоги найвимогливішого розробника пристроїв на мікроконтролерах. Застосовуючи мікроконтролери MSP430, розробники отримують потужний інструмент для роботи з

аналоговими і цифровими сигналами при наднизькому споживанні енергії батареї.

RISC (Reduced Instruction Set Computer) – архітектура процесора зі скороченим набором команд. Найбільш важливі відмітні особливості RISC–архітектури: архітектура реєстр-реєстр, прості способи адресації, прості команди і великий реєстровий файл. Мікроконтролер MSP430 має 27 основних інструкцій і 24 додаткових інструкції, що значно спрощує процес генерації команд. Відсутні спеціальні команди звернення до акумулятора, пам'яті або до периферійних пристроїв, що істотно підвищує ефективність роботи процесора. Ядро процесора – 16-бітове RISC ALU і шістнадцять 16-бітових реєстрів. Чотири реєстри виконують функції програмного лічильника (PC), реєстра статусу (SR), покажчика стека (SP) і реєстра констант (CG). Інші дванадцять 16-бітових реєстри – повністю у розпорядженні користувача. Реєстри загального призначення використовуються для зберігання змінних, покажчиків і для операцій з даними. Процесор звертається до цих реєстрів безпосередньо, що сприяє високій ефективності роботи мікроконтролера MSP430. Час виконання команд 1–4 машинні цикли (1–4 мкс).

Приємне доповнення до програмного забезпечення пакет FPP (Floating Point Package). Математика з плаваючою крапкою потрібна, якщо розмір вживаних чисел дуже великий. Пакет з плаваючою крапкою оперує з 24-бітовою і 40-бітовою мантиєю і розроблений для мікроконтролерів сім'ї MSP430. У пакет входять конвертери, що переводять числа з двійкового та двійково-десятькового формату в числа формату «плаваюча крапка», також передбачено і зворотне перетворення. Пакет з плаваючою крапкою використовує особливості RISC архітектури сім'ї мікроконтролерів MSP430. Під час роботи підпрограми, що використовує пакет FPP, аргументи копіюються в реєстри R4–R15 і всі обчислення відбуваються в реєстровій пам'яті. Після завершення обчислень результат зберігається у вершині стека.

Code Composer Studio (CCS) – Integrated Development Environment (IDE), інтегроване середовище розробки програмного забезпечення та створення коду для цифрових сигнальних процесорів (Digital Signal Processor – DSP) і/або ARM-процесорів сім'ї TMS320 та інших процесорів Texas Instruments (TI), таких як MSP430. Інтегральне середовище Code Composer Studio містить операційну систему реального часу DSP/BIOS. Також до складу продукту входять емулятори та підтримка JTAG-орієнтованого відлагодження. Середовище CCS містить такі інструменти, як редактор вихідного тексту, компілятор, компанувальник (link editor, linker), налагоджувач (debugger), симулятори для всіх типів процесорів TI, засоби візуалізації та багато інших допоміжних інструментів. Це єдиний графічний інтерфейс користувача, який дозволяє проводити покрокову розробку та відлагодження програмного коду. Інтуїтивно зрозумілі інструменти дозволяють розробнику швидко почати свою власну розробку, поступово освоюючи нові функціональні можливості та покращуючи продуктивність власних рішень. Засіб розробки CCS базується на програмному забезпеченні з відкритим вихідним кодом Eclipse. Використання саме цього продукту обумовлене тим, що Eclipse забезпечує ефективну структуру програмного забезпечення, яку підтримує велика кількість розробників вбудованих рішень. IDE Code Composer Studio була створена в рамках стандарту eXpressDSP, який розробляється фірмою Texas Instrument і представляє з себе універсальну технологію розробки програмного забезпечення для процесорів TI. Такій IDE притаманний ряд властивостей, найбільш суттєвими з яких є: - інтеграція усіх засобів розробки (редактор, відлагоджувач, менеджер проектів тощо) в одно застосування, що має зручний інтерфейс; - потужний промисловий Cі-компілятор, асемблерний оптимізатор, компанувальник; - масштабоване ядро (DSP/BIOS II) реального часу. Містить набір стандартних інструментів взаємодії DSP-системи з периферією; - програмний симулятор, що дозволяє вести відлагодження на програмній моделі вибраної сім'ї ЦСП;

- аналізатор реального часу, призначений для моніторингу стану системи без зупинки процесора;
- компілятор, що оптимізує код програми під певну платформу (розмір коду, оптимальність за рахунок використання конвеєризації і паралельних блоків);
- візуальний компанувальник, що дозволяє за допомогою графічного інтерфейсу розташовувати код програми і дані в пам'яті;
- можливість перегляду виду сигналів в різних графічних форматах;
- відкрита вбудовувана архітектура, що дозволяє інтегрувати в систему засоби сторонніх розробників.

Основні функціональні можливості IDE CCS наведені на рис. 2.1.



Рисунок 2.1 – Функціональні можливості Code Composer Studio 4.1

Після запуску CCS відкривається основне вікно, яке поділено на дві ділянки – робочу та ділянку навігації проекту. В ділянці навігації проекту відображається вся інформація про поточний проект – вихідні файли, командний файл компанувальника, підключені бібліотеки і т. ін. Основною програмною одиницею в CCS є проект (файл з розширенням \*.pjt). Проект містить в собі вихідні файли (тексти програм мовою Cі (\*.c) або асемблера

(\*asm), бібліотеки (\*.lib), командний файл компанувальника (\*.cmd)) та файл налаштувань (установки компілятора та компонувальника). На рис. 2.2 зображено частину основного вікна проекту в CCS та наведений перелік основних складових проекту.

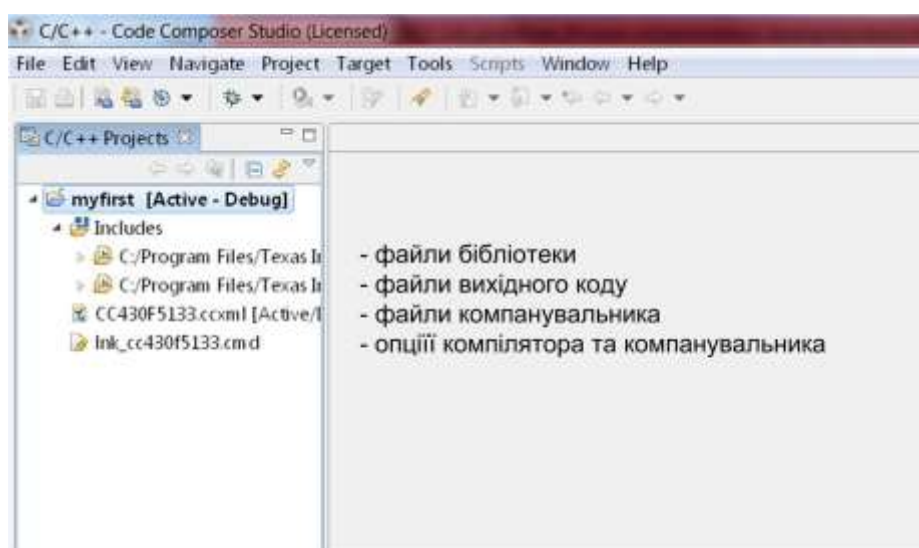


Рисунок 2.2 – Складові проекту

Основні етапи створення нового проекту в інтегрованому середовищі розробки Code Composer Studio наведені на рис. 2.3.

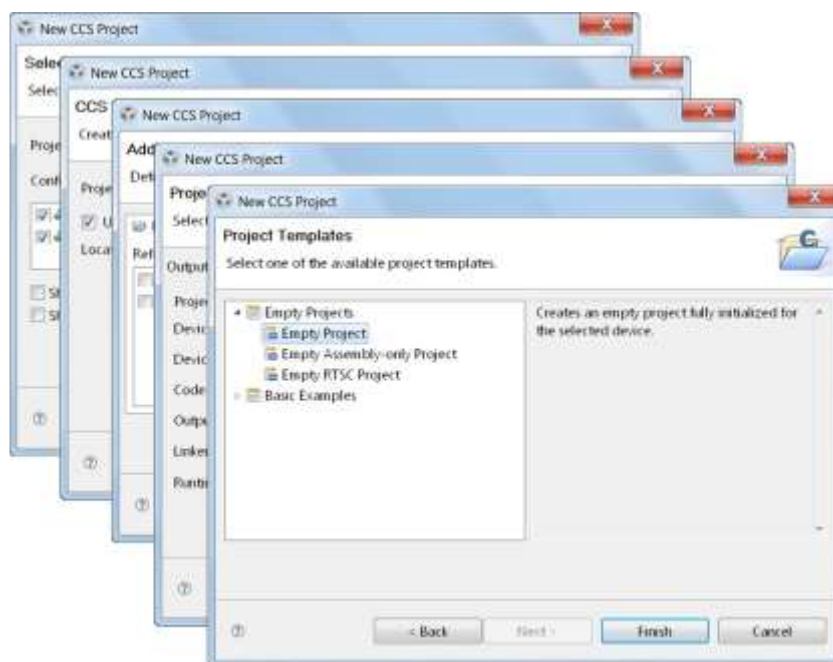


Рисунок 2.3 – Основні етапи створення нового проекту в CCS

### 2.3. Розробка архітектури інформаційної підсистеми ідентифікації тварин для групової доїльної установки

Розроблена в процесі виконання кваліфікаційної бакалаврської роботи архітектура програмного забезпечення інформаційної підсистеми ідентифікації тварин для групової доїльної установки наведена на рис. 2.4.

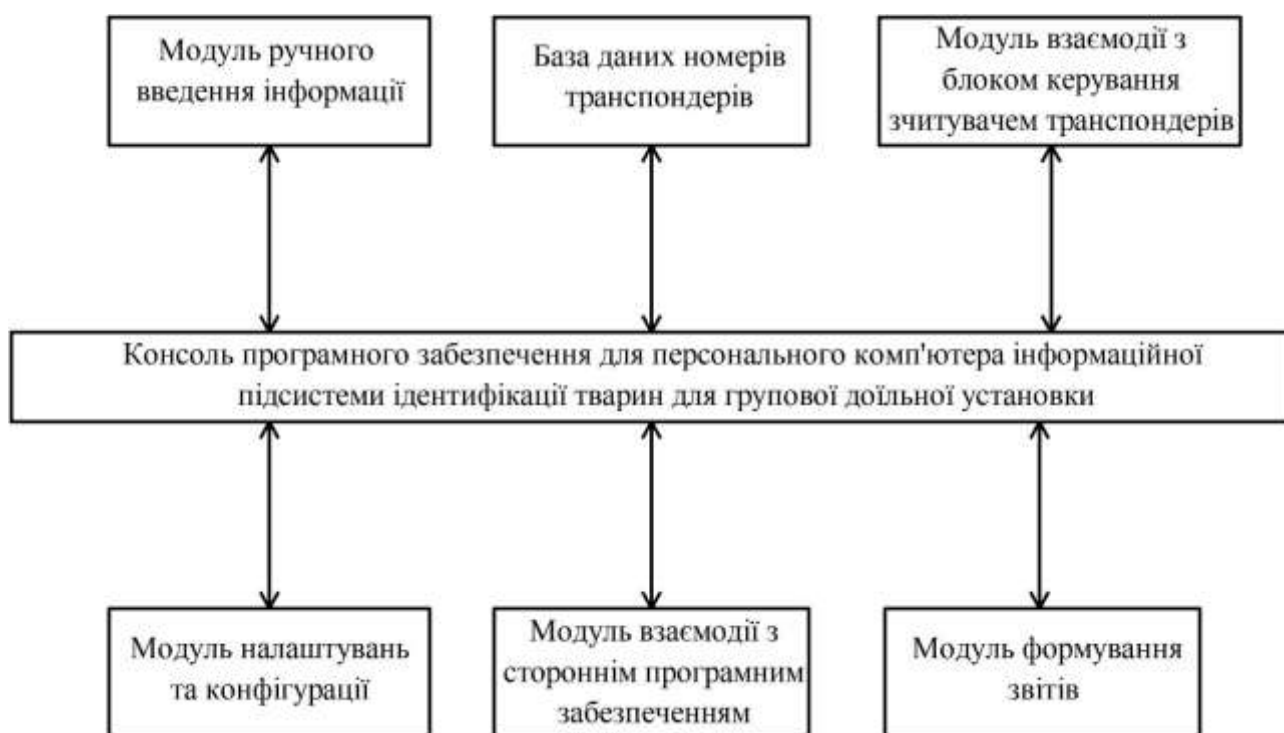


Рисунок 2.4 – Архітектура програмного забезпечення інформаційної підсистеми ідентифікації тварин для групової доїльної установки

Інформаційна підсистема ідентифікації тварин для групової доїльної установки, яка розробляється, буде використовуватися на тваринницьких фермах, які займаються виробництвом молока і призначена для отримання, обробки та представлення інформації про результати радіочастотної ідентифікації тварин на групових доїльних установках, встановлення відповідності номера транспондера тварини стадному номеру, ручного введення даних про відповідність номерів транспондерів стадним номерам тварин, корекції даних про результати ідентифікації. Інформаційна підсистема може працювати тільки на фермах обладнаних груповою доїльною установкою та системою радіочастотної ідентифікації тварин під час руху. Підсистема дозволяє покращити технологічний процес виробництва молока та забезпечує ефективність роботи персоналу.

Архітектура програмного забезпечення інформаційної підсистеми ідентифікації тварин складається з шести програмних модулів: модуля ручного

введення інформації, бази даних номерів транспондерів, модуля взаємодії з блоком керування зчитувачем транспондерів, модуля налаштувань та конфігурації, модуля формування звітів, модуля взаємодії з стороннім програмним забезпеченням.

Розроблене програмне забезпечення для інформаційної підсистеми ідентифікації тварин для групової доїльної установки призначене для забезпечення функціонування обліку тварин та їх зоотехнічних параметрів і результатів удою на групових доїльних установках. До складу інформаційної підсистеми ідентифікації тварин входить блок керування зчитувачем транспондерів, структурна схема якого наведена на рис. 2.5. Цей пристрій встановлюється на вході до кожної сторони робочої траншеї установки. Постійна напруга живлення 24 В надходить до блока від загальної лінії живлення електронного обладнання доїльної установки. Для живлення мікроконтролера блока використовується постійна напруга 3,3 В, для живлення інших активних електронних компонентів блока використовується постійна напруга 5 В, вищевказані напруги отримуються за допомогою відповідних перетворювачів. Блок керування зчитувачем транспондерів розрахований на сумісне функціонування з зчитувачами типу PNL-2530, PNL-4060-3, або PNL-60120 виробництва Allflex USA Inc., та транспондерами формату HDX з робочою частотою 134 кГц, що закріплюються у вусі або на шиї тварини. Радіочастотна ідентифікація здійснюється під час руху тварин при входженні їх на сторону доїльної установки. При коректному читанні коду транспондера, ця подія візуалізується за допомогою відповідного світлодіодного індикатора.

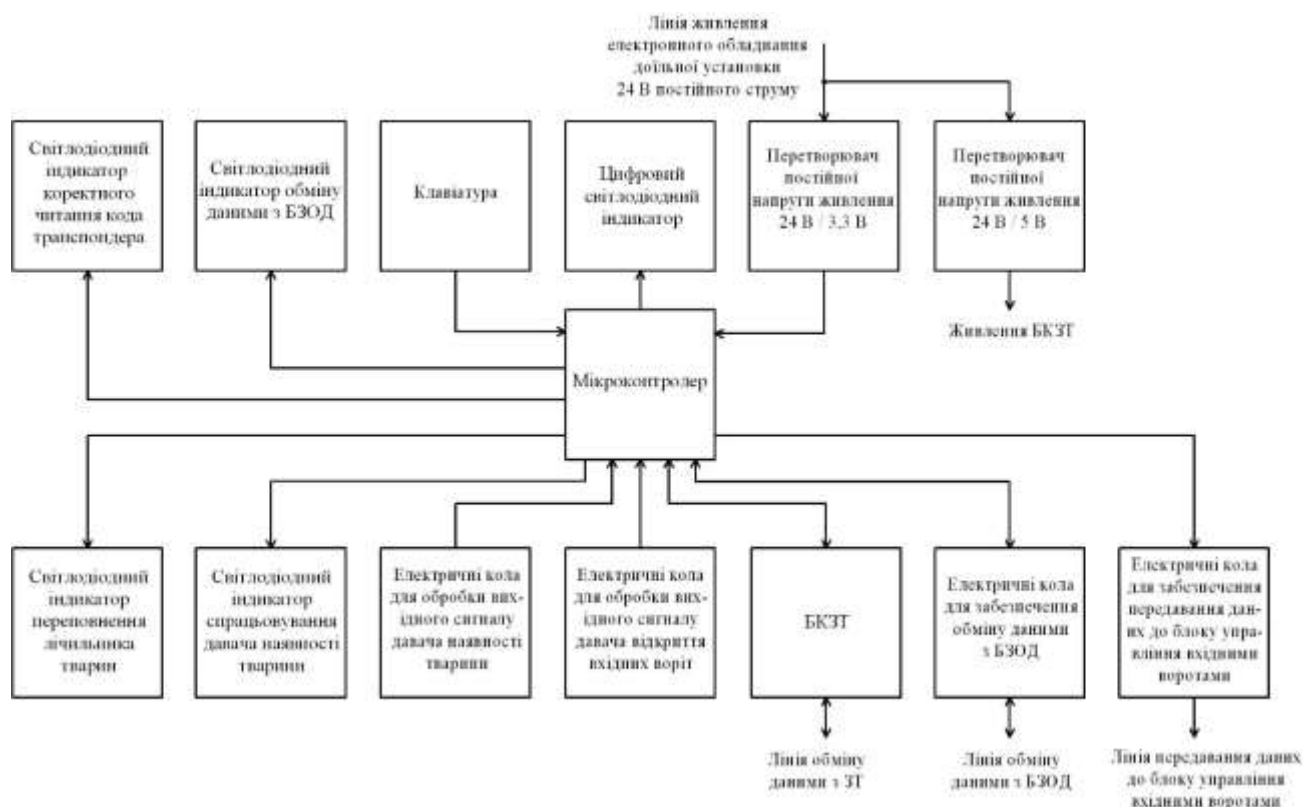


Рисунок 2.5 – Структурна схема блока керування зчитувачем транспондерів

У складі блока, за допомогою відповідних апаратних та програмних засобів, реалізований блок контролю проходження тварини. У якості датчика наявності тварини використовується оптичний датчик присутності об'єкта WE-T3AD виробництва «Highly Electric», або аналогічний. Вихідний сигнал датчика наявності тварини, після необхідної обробки, надходить на вхід мікроконтролера. Використовуючи оптимальну лінійну фільтрацію, яка реалізована програмно, здійснюється ідентифікація проходження тварин та їх підрахунок у потоці. Факт спрацювання датчика наявності тварини візуалізується за допомогою відповідного світлодіодного індикатора. При переповненні програмного лічильника тварин, параметри якого налаштовуються у відповідності з кількістю доїльних станків на стороні установки, здійснюється візуалізація цієї події за допомогою світлодіодного індикатора переповнення лічильника тварин. За допомогою блока забезпечується передача до сервера системи коду транспондера або інформації

про помилку ідентифікації тварини, номерів доїльних станків, у яких знаходиться ідентифіковані та неідентифіковані тварини. Наявність процесу обміну даними з БЗОД візуалізується за допомогою відповідного світлодіодного індикатора. Для забезпечення обміну даними з БЗОД використовується двонаправлена гальванічно ізольована «струмова петля», обмін даними між блоком та БЗОД здійснюється у дуплексному режимі, з передаванням контрольних сум для підвищення надійності обміну даними. Блок забезпечує управління вхідними воротами сторони доїльної установки шляхом передавання відповідних даних до блоку управління воротами. За допомогою чотирирозрядного цифрового світлодіодного індикатора візуалізуються три останніх цифри коду транспондера, поточний номер тварин групи у черзі, положення вхідних воріт, часовий інтервал перед закриттям вхідних воріт, час проходження тварини, мережевий номер першого БЗПД на стороні установки, мережевий номер останнього БЗПД на стороні установки, надходження коду транспондера, наявність тварини без транспондера.

## **РОЗДІЛ 3. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ІНФОРМАЦІЙНОЇ ПІДСИСТЕМИ**

### **3.1 Розробка програмного забезпечення інформаційної підсистеми ідентифікації тварин для групової доїльної установки**

Програмне забезпечення створено на мові програмування Python в та може функціонувати на платформах Windows 7, Windows 8 та Windows 10. Лістинг розробленого програмного забезпечення додатку А. Програмне забезпечення для інформаційної підсистеми ідентифікації тварин для групової доїльної установки встановлюється на сервері інформаційної підсистеми, робота з програмним забезпеченням здійснюється локально, безпосередньо з сервера підсистеми, або з віддалених робочих станцій за допомогою веб-браузерів Mozilla Firefox та Google Chrome. Робочі станції з'єднуються з сервером загальної інформаційної системи тваринницької ферми через комп'ютерну мережу.

Модуль ручного введення призначений для введення до бази даних ідентифікаційної інформації про тварину, а саме про її номер у стаді номер, номер транспондера, номер стада, ідентифікатор у державній базі даних, коментарі та іншу інформацію.

База даних номерів транспондера призначена для управління таблицями відповідності номерів транспондерів стадним номерам тварин, зберігання даних про історію зміни транспондерів, про додавання нових транспондерів, обліку транспондерів, відповідності номерів транспондерів групам тварин та стадам.

За допомогою модуля взаємодії з блоком керування зчитувачем транспондерів здійснюється налаштування його режимів роботи, формуються команди управління, перевіряються контрольні суми, отримується інформація про номер транспондера та надсилається інформація про відповідний стадний номер тварини. Інформація, яка передається та приймається за допомогою цього модуля, залежить від структури доїльної установки, способу утримання тварин, характеристик обладнання ферми. Також цей модуль приймає участь у

взаємодії з блоком управління селекційними воротами, модуль приймає код транспондера та здійснює передавання даних про розкол.

Модуль налаштувань та конфігурації забезпечує управління програмним забезпеченням інформаційної підсистеми для роботи з різними типами доїльних установок та різними варіантами зчитувачів транспондерів. За допомогою цього модуля програмне забезпечення налаштовується у відповідності до типу та структури доїльної установки, моделі зчитувача транспондерів, кількості каналів зв'язку, типу блока забезпечення процесу доїння, їх кількості, встановлення мережних номерів, кількості селекційних воріт. Окрім того, за допомогою цього модуля встановлюється номер порта для обміну даними з сервером та швидкість обміну даними. Нижче наведено фрагмент лістингу програмного забезпечення модуля налаштувань та конфігурації.

```
FEATURES_DICT = dict(zip(FEATURES, COLUMNS))
# Sizes of the training and evaluation datasets TRAIN_SIZE = 16000
EVAL_SIZE = 8000
# Specify model training parameters BATCH_SIZE = 16
EPOCHS = 10
BUFFER_SIZE = 2000 OPTIMIZER = 'SGD'
LOSS = 'MeanSquaredError'
METRICS = ['RootMeanSquaredError']
l8sr = ee.ImageCollection('LASAT/LC08/C01/T1_SR')
# Cloud masking function def maskL8sr(image):
cloudShadowBitMask = ee.Number(2).pow(3).int() cloudsBitMask =
ee.Number(2).pow(5).int()
qa = image.select('pixel_qa')
mask1 = qa.bitwiseAnd(cloudShadowBitMask).eq(0).And(
qa.bitwiseAnd(cloudsBitMask).eq(0))
```

```

mask2 = image.mask().reduce('min')
mask3 = image.select(opticalBands).gt(0).And(
image.select(opticalBands).lt(10000)).reduce('min') mask =
mask1.And(mask2).And(mask3)
return image.select(opticalBands).divide(10000).addBands(
image.select(thermalBands).divide(10).clamp(273.15, 373.15)
.subtract(273.15).divide(100)).updateMask(mask)
# The image input data is a cloud-masked median composite image =
l8sr.filterDate('2023-01-01', '2023-12-31').map(maskL8sr).median()
# Use folium to visualize the imagery
tiles=mapid['tile_fetcher'].url_format, attr='Data &copy; <a
name='median composite',
tiles=mapid['tile_fetcher'].url_format, attr='Data &copy; <a
name='thermal',
).add_to(map) map.add_child(folium.LayerControl()) map
# Prepare the response (what we want to predict)
nlcd = ee.Image('USGS/NLCD/NLCD2016').select('impervious') nlcd =
nlcd.divide(100).float()
mapid = nlcd.getMapId({'min': 0, 'max': 1})
map = folium.Map(location=[38., -122.5]) folium.TileLayer(
name='nlcd impervious',
).add_to(map) map.add_child(folium.LayerControl()) map
featureStack = ee.Image.cat([ image.select(BANDS), nlcd.select(RESPONSE)
]).float()
list = ee.List.repeat(1, KERNEL_SIZE) lists = ee.List.repeat(list, KERNEL_SIZE)
kernel = ee.Kernel.fixed(KERNEL_SIZE, KERNEL_SIZE, lists) arrays =
featureStack.neighborhoodToArray(kernel)
polyImage = ee.Image(0).byte().paint(trainingPolys, 1).paint(evalPolys, 2)
polyImage = polyImage.updateMask(polyImage)

```

```
mapid = polyImage.getMapId({'min': 1, 'max': 2, 'palette': ['red', 'blue']})
map = folium.Map(location=[38., -100.], zoom_start=5) folium.TileLayer
tiles=mapid['tile_fetcher'].url_format, attr='Map &copy;
```

Модуль взаємодії з іншим програмним забезпеченням забезпечує експорт та імпорт даних, отриманих в процесі функціонування програмного забезпечення інформаційної підсистеми ідентифікації тварин до електронних таблиць та інших популярних програм.

Модуль формування звітів призначений для створення набору тих звітів, які визначить оператор системи. Формуються звіти за групами тварин, стадами, активними транспондерами та стадними номерами тварин.

### **3.2 Розробка програмного забезпечення для мікроконтролера блока керування зчитувачем транспондерів**

Алгоритм роботи блока керування зчитувачем транспондерів, який входить до складу інформаційної підсистеми ідентифікації тварин для групової доїльної установки, полягає в наступному. Після початку роботи групової установки, блок керування зчитувачем транспондерів у безперервному режимі здійснює опитування стану зчитувача. Після того, як транспондер попадає в робочу зону зчитувача, відбувається дешифрація його номера. Після цього блок керування зчитувачем транспондерів очікує сигналу від сенсора проходження тварини, який свідчить про те, що тварина увійшла на доїльну установку, після цього розраховується номер доїльного станка. Далі отримані дані через блок забезпечення обміну даними надсилаються до сервера підсистеми, який в свою чергу за номером транспондера визначає стадний номер тварини та надсилає його до блока забезпечення процесу доїння, який знаходиться у попередньо визначеному доїльному станку. Обмін інформацією забезпечується за допомогою фізичного середовища «струмова петля» та дводрової лінії обміну

даними з гальванічною оптичною розв'язкою. Для контролю правильності переданих та прийнятих даних використовуються контрольні суми.

У додатку Б наведено лістинг програмного забезпечення для мікроконтролера на мові програмування C++ для мікроконтролера MSP430F149IPM виробництва компанії Texas Instruments, під управлінням якого працює блок керування зчитувачем транспондерів, який входить до складу інформаційної підсистеми ідентифікації тварин для групової доїльної установки. Цей фрагмент відповідає за дешифрацію кода транспондера.

```
switch( Key )
{
    case 129: // "go" TAG Number 1
        for (k = 0; k <= 2; k++)
        {
            if (!CowStackOverflow)
            {
                TAG1();
                IsCRC_check();
            }
        }
        break;
    case 65: // "again" immitating TAG Number 2
        for (k = 0; k <= 2; k++)
        {
            if (!CowStackOverflow)
            {
                TAG2();
                IsCRC_check();
            }
        }
    }
```

```
}
break;
```

## **Висновок**

У бакалаврській кваліфікаційній роботі розроблено інформаційну підсистему ідентифікації тварин для групової доїльної установки. Об'єктом дослідження є процес, концепція та архітектура побудови засобів ідентифікації тварин та інформаційних підсистем ідентифікації. Предметом дослідження є методи інформатизації молочних ферм обладнаних засобами ідентифікації тварин під час доїння.

Розроблена підсистема призначена для впровадження на тваринницьких фермах, які обладнані груповими доїльними установками та засобами ідентифікації тварин. За допомогою підсистеми забезпечується отримання, обробка та представлення у необхідному форматі інформації про результати радіочастотної ідентифікації тварин на групових доїльних установках, що дозволяє збільшити ефективність управління молочною фермою. Інформаційна підсистема забезпечує передачу інформації про номер транспондера тварини до сервера на основі персонального комп'ютера та прийом інформації про відповідний стадний номер тварини, який знаходиться у базі даних.

У першому розділі роботи здійснено аналіз поточних невирішених завдань в галузі ідентифікації тварин та сформульовано завдання дослідження, здійснено огляд існуючих методів та способів ідентифікації тварин, проведено аналіз засобів ідентифікації та діючих стандартів у цій галузі.

Другий розділ присвячений обґрунтуванню структури та архітектури інформаційної підсистеми ідентифікації тварин для групової доїльної установки, обґрунтовано вибір мови програмування для програмного забезпечення сервера підсистеми та розглянуто технологію розробки програмного забезпечення для мікроконтролерів, розроблено архітектуру підсистеми.

У третьому розділі проведено розробку програмного забезпечення для сервера інформаційної підсистеми та програмне забезпечення для мікроконтролера блоку керування зчитувачем транспондерів.

### СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. ISO 11784/85. Radio frequency identification of animals / International Standard Organization. – Режим доступу : <http://www.iso.org>.
2. Кучерук, В. Ю. Класифікація систем ідентифікації тварин 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. – № 1 (50). – с. 252–256.
3. 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.
4. Allflex / Allflex USA Inc. – Режим доступу : <http://www.allflexusa.com>.
5. RFID Journal / RFID journal LLC. – Режим доступу : <http://www.rfidjournal.com>.
6. Domdouzis, K. Radio-frequency identification (RFID) applications: A brief introduction / K. Domdouzis, B. Kumar, C. Anumba // Adv. Engineering Informatics. – 2007. – P. 350–355.
7. Radio Frequency Identification RFID – a basic primer / AIM International, Inc. white paper, 1998. – 56 p.
8. Breiner, S. J. Practices and perceptions of cow-calf producers regarding the National Animal Identification System / S. J. Breiner, D. A. Blasi, K. M. Boone, T. C. Schroeder, S. A. Grau // Journal of Animal Science. – 2007. – № 85. – P. 611–617
9. Ghirardi, J. J. Long term comparative trial of ear tags and ceramic boluses for the electronic identification of beef cattle under European rangeland conditions / J. J. Ghirardi, G. Caja, C. Conill, M. Hernández-Jover, D. Garín // Journal of Animal Science. – 2004. – № 82. – P. 351–359

10. Кучерук, В. Ю. Засоби радіочастотної ідентифікації для доїльно-молочних відділень тваринницьких ферм / В. Ю. Кучерук, Є. А. Паламарчук, П. І. Кулаков // Вимірювальна та обчислювальна техніка в технологічних процесах. – 2014. – № 3(48). – с. 145–150.

11. Basarab, J. Read rate on two multi-panel RFID reader systems for use in beef cattle / J. Basarab, L. Erickson, J. Kopp [et al] // Alberta Food and Rural Development New Initiative Fund, Alberta, Canada, Project Number : 2005007, 2007. – 254 p.

12. What is Python? Executive Summary [Online]. Available: <https://www.python.org/doc/essays/blurb/>