

# Animal Care Automated System - Software Development and Implementation

S. O. Ogunlere  
Dept. of Comp. Science  
Babcock University, Nig

Seun Ebiesuwa  
Dept. of Comp. Science  
Babcock University, Nig

G. I. Amugo  
Dept. of Comp. Science  
Babcock University, Nig

## ABSTRACT

In the field of animal care, farmers have made several breakthroughs, making sure animals are well fed at the right time and not being neglected. However, there is yet much problem in the homes and other areas where animals are being used. Radio Frequency Identifier (RFID) are being used mostly in farms to identify the animals in case of theft, and in order to recognize and differentiate the animals. However, that does not help in making sure they are fed at the right time and adequately taken care off when their owners are not available to do so. This study focuses on solving the problem of adequately caring for pets and animals either by feeding, checking their wellbeing, and other challenges associated with their welfare through mobile device system using intelligent RFID technology that incorporates both hardware and software developmental tools to can handle these responsibilities without human intervention. This study is a follow up to an earlier study on animal care automated system published in the November edition of IJCA journal on the hardware architectural aspect using RFID tags to different animals, in other to observe behavior and other key wellbeing measures that offer a comprehensive depiction to animal welfare.

## Keywords

Radio Frequency Identifier, Animal wellbeing, Event manager Information, Validation of images, Software process Development.

## 1. INTRODUCTION

It was concluded in the article [1] that further work on the software development and implementation is being awaited on the above subject matter. The work presented in this write-up is a clear indication of the fulfilment to the awaited software development that is the driving force behind the Animal Care Automated System Hardware Architectural design. Below are recaps of [1] in continuation with the software programming in C Language.

## 2. RFID SOFTWARE ARCHITECTURE DEVELOPMENT AND VALIDATION

There are four major components that the RFID software consists of and they are listed in the following order:

1. RFID Event Manager
2. RFID Information Server
3. RFID Management Console
4. RFID Configuration Manager

The RFID event manager converses with RFID sensor devices to bring about information from the animal's smart cage. This evidence can be kept in the RFID information server for current activities or future study. The RFID event manager is

a disseminated platform involving one control station and single or multiple execution agents [2], [3]. The execution agent and control station are fitted on the same pet owner PC or mobile device for processing information, interactive with the hardware devices, and posting the information to the pet owner. The control station constantly observes the status of the execution agents. To offer high processing accessibility if an execution agent flops, the configuration object is provisioned to another execution agent.

The RFID configuration manager is a Graphical Unit Interface (GUI) application used to state the set of devices coupled to the RFID event manager. The RFID configuration manager statically explain the manner to execute the information within the RFID event manager and where to refer the information after dispensation.

The RFID Management console looks after and manage the prestige of the devices that are linked to the RFID event manager. The RFID management console permits pet owners to look after the prestige of the animal well-being and welfare statistics and alter the runtime parameters for each.

## 2.1 Processing RFID Event Manager Information

For intense information produced by the devices linked to the RFID event manager, four main mechanisms are involved:

- a) **RFID Configuration Manager:** - to statically express a single or multiple connectors that post information to the automated application.
- b) **Application Level Events (ALE):** - postulates the kind of actions the automated application consumes. When ALE is being used, the application produces XML messages that describe the events of concern. The automated application requires to be programmed to handle the XML messages that are acknowledged encompassing the required information. Any of the Java or non-Java application that obeys with the universal ALE specification can converse with the RFID event manager.
- c) **JAVA API'S:** - the Java Reader Client APIs controls devices, are used to program RFID tags, the read and write user memory and tag identification on RFID tags by making use of the Java library bundled with the RFID software. The automated application converses straight with the RFID event manager by using Java Remote Method Invocation (Java RMI) without the necessity to alter amongst protocols and data representation.
- d) **WEB Service for Device Access:** -The web applications provide a 3 Dimension video that can reproduce the pet or animal activities in the cage.

The user can cope by using the enabled smart phone or PC to control, the entire process. In order to improve the 3-Dimension component, the Web-GL technology can be used because it can produce and cope with 3-dimensional graphics unswervingly, permitting the communication with the 3 Dimension environment.

## 2.2 Development and Validation of Image Monitoring Process (Software Process)

C language Data was used for the software programming which is received by GPS from the satellites. This application supports pet or animal owners in the automated animal welfare system, because it is able to bring out raw data, kept in a relational database, in a way to deliver to the end-users (pet and animal owners) with an effective breakdown that recapitulates main animal welfare parameters [4], [5], [6]. The application primarily processes the raw data with the help of the three stages of the RCP (RSSI Chebyshev Ping-pong). When an RFID tag is read by one of the reader's antennas, the well-being status of the animal is uniquely individualized [7], [8]. The user or pet owner included in the software programming in order to obtain notification on any form of intrusion on the animals and generates an array of images.

The steps involved in image generation processes are:

- i. First Step: Image was read from the image database.
- ii. Second Step: In order to increase the quality of the image, it was Pre-processed. Preprocessing encompasses identification and labeling of the objects that is contained in the images with the help of an image query processing algorithm. The main output of the pre-processing step will be a set of records, one for each image, encompassing the object identifiers for the objects involved in the image.
- iii. Third Step: The Transformation of images was performed into database similar to a table. Each row in the table represents a pixel. Thus the number of rows (cardinality) matches to the entire number of pixels in an image. The columns correspond to the features related with a pixel. These features can encompass means, contrast, local variation, etc.
- iv. Fourth Step: Once the database table was attained, completed feature extraction follows. Features characteristics of the objects of interest, if carefully selected, are demonstrative of the supreme relevant information that the image has to offer for a comprehensive characterization of the lesion.
- v. Fifth Step: Once features have been removed, achieve mining using suitable data mining methods to recognize appropriate patterns.
- vi. Sixth Step: The resulting patterns were assessed and construed to attain the ultimate knowledge, which was applied to applications. The algorithm used is Content Based Image Retrieval (CBIR). The main objective of this algorithm is to put together the difference between the two images characterized in the form of two matrixes to produce the resultant matrix. Between the two input images, one image is the grey scale of the original image and the second one is the image gotten from the computation of the standard deviation of the input image. Segmentation and abstraction were then performed on the resultant image by the procedure of thresholding. It is a two-step procedure where image features are

removed in step one to a distinguishable extent. When the intrusion is detected the calculated result is given to the alert system (pet owner mobile phone or PC) via serial communication port.

### 2.2.1 Algorithm

- |         |  |
|---------|--|
| Step 1: | Start  |
| Step 2: | Image acquiring of monitoring area.  |
| Step 3: | Apply the image mining algorithm to identify change in settled reference background  |
| Step 4: | If there is no change in the newly acquired image and settled reference go to step 5 |
| Step 5: | Apply the CBIR algorithm to identify the intrusion.                                  |
| Step 6: | Activate the alert system.   |
| Step 7: | Stop   |

## 3. SOFTWARE PROGRAMMING IN C LANGUAGE

```
// Add as many wanted Tag ID here
char tagOk1[] ="370015E15291";
char tagOk2[] ="3700169B48F2";
char tagOk3[] ="18002C2B7768";
char tagOk4[] ="FFFFFFFFFFFF";
char data_temp, RFID_data[12];
char inputTag[12];

#include<LiquidCrystal.h>
LiquidCrystal lcd(2, 3, 4, 5, 6, 7);

#include <VarSpeedServo.h>
VarSpeedServo myservo;
const int servoPin = 9; // MOTOR TO SERVE FOOD

// SET GATE MOTOR PARAMETER
int IN1 = 13;
int IN2 = 10;
int ENA = 11;

// SET OPERATION DELAY PARAMETER
int hours = 0; // start hours
int minutes = 0; //start min
int seconds = 10; //start seconds

// SET EXTERNAL DELAY PARAMETER
const int ext_delay = 14; // external delay input pin
int delayState = 0; // variable for reading the delay status

int water_pump = 8; // PUMP FOR WATER
//int gate_control = 10; // CONTROL PENT DOOR
int trigger_Buzzer = 18; // SOUND TO DRIVE PETS INTO
PENT
int Rx_TX_Relay = 12; // CONNECT BOARD TO RFID
MODULE
int status_led = 17; // STATUS LED
int off_board = 15; // BOARDS POWER PIN

void setup() {
  pinMode(ext_delay, INPUT);
  pinMode(water_pump, OUTPUT);
  //pinMode(gate_control, OUTPUT);
  pinMode(status_led, OUTPUT);
  pinMode(trigger_Buzzer, OUTPUT);
  pinMode(Rx_TX_Relay, OUTPUT); // CONNECTS
  RECEIVER OF BOARD TO TRANSMITTER OF RFID
  READER
```

```
pinMode(off_board, OUTPUT);

pinMode(IN1,OUTPUT);
pinMode(IN2,OUTPUT);

myservo.attach(servoPin);
myservo.write(0,225,true);
lcd.begin(16, 2);
Serial.begin(9600);

lcd.clear();
lcd.setCursor(2, 0);
lcd.print("ANIMAL CARE");
lcd.setCursor(1, 1);
lcd.print("ROBOTIC SYSTEM");
delay(5000);
//delay(1000);

// SET TIME TO INITIALIZE DEVICE
//for (int j=1; j<=12; j=j+1)
//  for (int j=1; j<=5; j=j+1)
//  {
//    lcd.clear();
//    lcd.setCursor(4, 0);
//    lcd.print("WELCOME!");
//    delay(1000);
//    lcd.clear();
//    delay(200);
//  }
//  digitalWrite(status_led, HIGH);

// TIME TO SERVE FOOD
  lcd.clear();
  lcd.setCursor(3 , 0);
  lcd.print("Serve Food");
  delay(1000);
  myservo.write(180,10,true); // RELEASE FOOD
  delay(5000);
  lcd.clear();
  lcd.setCursor(4 , 0);
  lcd.print("Success!");
  lcd.setCursor(3 , 1);
  lcd.print("Lock Basin");
  delay(2000);
  myservo.write(0,10,true); // STOP FOOD RELEASE

// TIME TO SERVE WATER
  lcd.clear();
  lcd.setCursor(2 , 0);
  lcd.print("Serve Water");
  delay(1000);
  digitalWrite(water_pump, HIGH); // START WATER
PUMP
  delay(8000);
  lcd.clear();
  lcd.setCursor(4 , 0);
  lcd.print("Success!");
  lcd.setCursor(3 , 1);
  lcd.print("Lock Pump");
  delay(5000);
  digitalWrite(water_pump, LOW); // STOP WATER PUMP
/*
  lcd.clear();
  lcd.setCursor(2 , 0);
  lcd.print("Food & Water");
  lcd.setCursor(0 , 1);
  lcd.print("Serve Successful");
  delay(2000);

  */
  lcd.clear();
  }
void loop() {
  /*
  // CHECK IF THE EXTERNAL DELAY IS ACTIVE
  delayState = digitalRead(ext_delay);
  if (delayState == HIGH) {
    digitalWrite(Rx_TX_Relay, LOW); // MODULE
    DISCONNECTED WITH BOARD
  }
  */
  digitalWrite(status_led, LOW );
  // WAIT FOR BOARD TO CONNECT READER
  lcd.setCursor(1, 0);
  lcd.print( "Access Enables" );
  lcd.setCursor(1, 1);
  lcd.print( "In Few Seconds" );
  delay(3000);
  lcd.clear();
  lcd.print( "Waiting..." );
  delay(150);
  while (hours > 0 || minutes > 0 || seconds >= 0) {
  lcd.setCursor(4, 2);
  /*
  (hours < 10) ? lcd.print( "0" ) : NULL;
  lcd.print(hours);
  lcd.print( ":" );
  (minutes < 10) ? lcd.print( "0" ) : NULL;
  lcd.print(minutes);
  lcd.print( ":" );
  (seconds < 10) ? lcd.print( "0" ) : NULL;
  lcd.print(seconds);
  lcd.display();
  */
  stepDown();
  delay(1000);
  }
  void stepDown() {
  if (seconds > 0) {
  seconds -= 1;
  } else {
  if (minutes > 0) {
  seconds = 59;
  minutes -= 1;
  } else {
  if (hours > 0) {
  seconds = 59;
  minutes = 59;
  hours -= 1;
  }
  }
  operate_led();

  if ((hours == 0) && (minutes == 0) && (seconds == 0)) {
  //digitalWrite(Rx_TX_Relay, HIGH); // MODULE
  CONNECTED TO BOARD
  lcd.clear();
  lcd.setCursor(5, 0);
  lcd.print( "Ready!" );
  delay(150);
  }
  // CHECK IF THE EXTERNAL DELAY IS ACTIVE
  delayState = digitalRead(ext_delay);
  if (delayState == HIGH) {
    digitalWrite(Rx_TX_Relay, LOW); // MODULE
    DISCONNECTED WITH BOARD
  }
  }
}
```

```
digitalWrite(trigger_Buzzer, HIGH);

lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Mission Complete");
delay(2000);
lcd.clear();
for (int j=1; j<=5; j=j+1)
{
  lcd.clear();
  delay(200);
  lcd.setCursor(4, 0);
  lcd.print("GOODBYE!");
  delay(1000);
}
//digitalWrite(gate_control, HIGH); // GATE OPEN FOR
PET TO ENTER
gate_open();
delay(8000); // DELAY TO MAKE SURE ALL PETS
ENTER
//digitalWrite(gate_control, LOW);
gate_close();
delay(3000); // DELAY FOR GATE TO SHUT B/4 BOARD
SWITCH-OFF
digitalWrite(trigger_Buzzer, LOW);
delay(1000);
digitalWrite(off_board, HIGH); // BOARD SWITCH-OFF

} else { digitalWrite(Rx_TX_Relay, HIGH);
}
if (Serial.available()){

int count = 0;
while(Serial.available() && count < 12) {
  inputTag[count] = Serial.read();
  count++;
  delay(10);
}
Serial.println(" Reading Card... Please Wait ");
lcd.clear();
lcd.setCursor( 0 , 0);
lcd.print("Trying Access...");
int compare = 1;
compare = (strncmp(tagOk1, inputTag,12)) ; // if both tags
are equal strncmp returns a 0
if (compare == 0){
  delay(3000);
  //digitalWrite(gate_control, HIGH);
  gate_open();

  lcd.clear();
  lcd.setCursor( 0 , 0);
  lcd.print("Welcome 'SCOOPY'");
  //delay(500);

  delay(2000); // delay of 2 secs for motor to respond
  //digitalWrite(gate_control, LOW);
  gate_close();
}
compare = (strncmp(tagOk2, inputTag,12)) ; // if both tags
are equal strncmp returns a 0
if (compare == 0){
  delay(3000);
  //digitalWrite(gate_control, HIGH);
  gate_open();

  lcd.clear();
  lcd.setCursor( 0 , 0);
  lcd.print("You Are Welcome");
  //delay(500);
  lcd.setCursor( 0 , 1);
  lcd.print("BANGOOG");
  delay(2000); // delay of 2 secs for motor to respond
  //digitalWrite(gate_control, LOW);
  gate_close();
}
compare = (strncmp(tagOk3, inputTag,12)) ; // if both tags
are equal strncmp returns a 0
if (compare == 0){
  delay(3000);
  //digitalWrite(gate_control, HIGH);
  gate_open();

  lcd.clear();
  lcd.setCursor( 0 , 0);
  lcd.print("You Are Welcome");
  //delay(500);
  lcd.setCursor( 2 , 1);
  lcd.print("KINKY");
  delay(2000); // delay of 2 secs for motor to respond
  //digitalWrite(gate_control, LOW);
  gate_close();
}
compare = (strncmp(tagOk4, inputTag,12)) ; // if both
tags are equal strncmp returns a 0
if (compare == 0){
  lcd.clear();
  lcd.setCursor( 3 , 0);
  lcd.print("BAD CARD!");
  delay(500);
  // myServo.write(0);
  //digitalWrite(gate_control, LOW);
}
else {
  delay(3000);
  lcd.clear();
  lcd.setCursor(2 , 0);
  delay(300);
  lcd.print("Gate Locked!");
}
}
}

void operate_led() {
  digitalWrite(status_led, HIGH);
  delay(300);
  digitalWrite(status_led, LOW);
  delay(10);
}

void gate_open() {
  analogWrite(ENA, 200); // motor speed
  digitalWrite(IN1,LOW); // rotate forward
  digitalWrite(IN2,HIGH);
  delay(2000);
}

void gate_close() {
  analogWrite(ENA, 200); // motor speed
  digitalWrite(IN1,HIGH); // rotate reverse
  digitalWrite(IN2,LOW);
  delay(2000);
}
}
```

#### **4. CONCLUSION**

This study examined the use of RFID technology in the modern approach to the animal welfare system, it has been observed that the RFID technology has the capacity to automatically, remotely and unremarkably perceive animal performance patterns in a zoo, circus, themes, parks, or homes and ensures that all the welfare parameters for determining their well-being is adequately observed, and transmitted, to their owner. The design and implementation of the RFID technology in the animal welfare system which incorporate a high level language programming as a driven force (interface), has confirmed to be a prosperous means for proficiently gathering high-standard, quality, and frequent data automated remotely. The study adopted an approach of assigning RFID tags to different animals, so as to observe behavior and other key wellbeing measures to offer a comprehensive depiction of animal welfare. This work signifies the secondary stage of a multi-layered project that enhances operational functionality capabilities to advance animal care.

Future focus will also be extended to integration of artificial intelligence (in form of robotics) into the process of feeding the animals.

#### **5. REFERENCES**

- [1] Ogunlere S. O., Fagbohun S. A., Iwala O. K., (2017). Animal Care Automated System, *IJCA*, Vol. 177, No. 2, pp. 23 – 31
- [2] Nagle et al. (2003). Monitoring and classifying the behavior of a herd of sheep using ad hoc wireless sensor networks and artificial intelligence, *7th World Congress on Computers in Agriculture and Natural Resources Conference Proceedings*
- [3] Taylor, K. M., (2004). *TinyDB by remote*. Sydney, Australia: Australian Mote Users' Workshop, Automated Assessment of Animal Health and Wellbeing
- [4] Hamadani H., Khan A. A., (2015). Automation in livestock farming – A technological revolution, *International Journal of Advanced Research*, Vol. 3, Issue 5, pp. 1335 – 1344
- [5] Haapala, H. (2003). Operation of electronic identification of cattle in Finland. *The Proceedings of the 4th European Conference in Precision Agriculture*.
- [6] Grothmann, A. N. (2012). *Automatic feeding systems for dairy cattle – potential for optimization in dairy farming*. Valencia, Spain
- [7] Voulodimos C. Z., (2010). A complete farm management system based on animal identification using RFID. *Computers and Electronics in Agriculture*, Vol. 70, Issue 2, pp. 380-388.
- [8] Akram A., Omar H., & Al-Zaabi A. Y., (2011). RFID design, simulation, and implementation – Proceedings of ASEE, Northeast Section Annual Conference, University of Hartford.