Bachelor’s Degree Report (BSc)

AUTOMATED FISH FEEDER
Using Internet of Things (IoT)

A Project Report

By

Okechukwu OHAMADIKE
(A00014573)

Supervisors: Dr. Babatunde Ogunleye, Engr. Emmanuel Nicholas

School of Information, Technology & Computing
American University of Nigeria

December 2017
AUTOMATED FISH FEEDER

Using Internet of Things (IoT)

Okechukwu OHAMADIKE
(A00014573)

BSc Report

Project Report submitted in
Full fulfillment of the requirements for the
Degree of Bachelor in Telecommunications and Wireless Technology

Supervisors: Dr. Babatunde Ogunleye, Engr. Emmanuel Nicholas

School of Information, Technology & Computing
American University of Nigeria

December 2017
DECLARATION

I declare that this project work is carried out by myself and has not been previously submitted for the degree. And that the report is written unaided in my own words, apart from any quoted material which are identified clearly in the correct manner and fully acknowledged work by others. The work and the report were carried out under the guidance of Babatunde Ogunleye and Emmanuel Nicholas

Student's Name:..........................................................Signature:..............................................

Supervisor's Name:..........................................................Signature:..............................................

Supervisor's Name:..........................................................Signature:..............................................

Program Chair's Name:......................................................Signature:..............................................

Dean's Name:..........................................................Signature:..............................................

SDP Coordinator’s Name:............................................Signature:..............................................
ABSTRACT

Automated Fish Feeder is a system, which focuses on aiding small scale fish farmers to feed their farms efficiently. By using Arduino and Internet of Things, this is made possible. This provides fish farmers with time and adequate information as the system is set to feed at several time intervals and send information through the internet should the feed container be low or empty.

In addition, the system has so many potentials. The system can be embed into other systems such as a water changing system. By doing this, it could be more convenient for farmer as it reduces the cost of lab and no matter where they are, their farm can still be taken care of. The Aim of the project is to provide an efficient way of feeding a small scale fish farm.
ACKNOWLEDGEMENT

We would like to thank the following people for their contribution to the success of this project:

I am eternally grateful to God Almighty my Dad, my best friend; for giving me the grace to see this project to now and further advancement of the project.

I am eternally grateful to Chief & Mrs Uche Ohamadike my parents; for financing the whole project and uplifting me every step of the way.

I am eternally grateful to Dr. Babatunde Ogunleye and Engr. Emmanuel Nicholas the supervisors of this project; for their expertise in some key software elements used in this project.

I am eternally grateful to Mr. Joseph Dalughut, a friend and roommate; for his expertise in Arduino and tutoring me.

I am eternally grateful to Late Mrs. Euphemia Ekwuomadu my aunt who passed away on October, 2017 and was laid to rest on the 8th of December, 2017; for the many lessons and love she showed to me and my family, thank you.

I am eternally grateful to everyone in the office of Dean of student affairs, my family in the Non-Denominational Christian Fellowship for their endless prayers and support and everyone that has influenced me, good and bad.

[BO3]
# Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>DECLARATION</td>
<td>iii</td>
</tr>
<tr>
<td>ABSTRACT</td>
<td>iv</td>
</tr>
<tr>
<td>ACKNOWLEDGEMENT</td>
<td>v</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>vii</td>
</tr>
<tr>
<td>ACRONYMS</td>
<td>viii</td>
</tr>
<tr>
<td><strong>CHAPTER ONE</strong></td>
<td></td>
</tr>
<tr>
<td>1.0 Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1.1 BACKGROUND AND MOTIVATION</td>
<td>1</td>
</tr>
<tr>
<td>1.2 PROBLEM STATEMENT</td>
<td>2</td>
</tr>
<tr>
<td>1.3 PROJECT OBJECTIVE</td>
<td>2</td>
</tr>
<tr>
<td>1.4 OUTLINE OF THE REPORT</td>
<td>2</td>
</tr>
<tr>
<td><strong>CHAPTER TWO</strong></td>
<td>4</td>
</tr>
<tr>
<td>2 REQUIREMENTS</td>
<td></td>
</tr>
<tr>
<td>2.1.1 Arduino</td>
<td>4</td>
</tr>
<tr>
<td>2.1.2 Server</td>
<td>4</td>
</tr>
<tr>
<td>2.1.3 IoT technology</td>
<td>4</td>
</tr>
<tr>
<td>2.2 FUNCTIONAL REQUIREMENTS</td>
<td>5</td>
</tr>
<tr>
<td>2.2.1 Arduino</td>
<td>5</td>
</tr>
<tr>
<td>2.2.2 Server</td>
<td>5</td>
</tr>
<tr>
<td>2.2.3 IoT Technology</td>
<td>6</td>
</tr>
<tr>
<td>2.3 HARDWARE REQUIREMENTS</td>
<td>6</td>
</tr>
<tr>
<td><strong>CHAPTER THREE</strong></td>
<td>14</td>
</tr>
<tr>
<td>3 DESIGN</td>
<td></td>
</tr>
<tr>
<td>3.1 HARDWARE DESIGN</td>
<td>14</td>
</tr>
<tr>
<td>3.1.1 Connections on the Arduino Board</td>
<td>14</td>
</tr>
<tr>
<td>3.2 SOFTWARE DESIGN</td>
<td>15</td>
</tr>
<tr>
<td>3.2.1 Node- Red</td>
<td>15</td>
</tr>
<tr>
<td><strong>CHAPTER FOUR</strong></td>
<td>17</td>
</tr>
<tr>
<td>4 Conclusion and future work</td>
<td></td>
</tr>
<tr>
<td>4.1 CONCLUSION</td>
<td>17</td>
</tr>
<tr>
<td>4.2 FUTURE IMPLEMENTATION</td>
<td>17</td>
</tr>
<tr>
<td>References</td>
<td>19</td>
</tr>
<tr>
<td>APPENDIX</td>
<td>20</td>
</tr>
</tbody>
</table>
LIST OF FIGURES

Figure 2-1  Arduino Uno R3.......................................................................................................................... 7
Figure 2-2  Ultrasonic sensor.......................................................................................................................... 8
Figure 2-3 ESP8266 Wi-Fi Module ............................................................................................................... 9
Figure 2-4 5v Servo Motor .................................................................................................................................. 10
Figure 2-5  Jumper Cables .................................................................................................................................... 11
Figure 2-6 5v to 3v Voltage regulator ........................................................................................................ 12
Figure 2-7 LED lights on Arduino Board ..................................................................................................... 13
Figure 3-1 Circuit Diagram of Sensor, Servo Motor and ESP8266 connected to Arduino Uno.......................................................... 14
Figure 3-2 Node-red Command prompt Initiation ............................................................................................. 15
Figure 3-3 Node-red interface on the Web Browser ...................................................................................... 16
<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IoT</td>
<td>Internet of Things</td>
</tr>
<tr>
<td>GPIO</td>
<td>General Purpose Input/Output</td>
</tr>
<tr>
<td>I/O</td>
<td>Input / Output</td>
</tr>
<tr>
<td>HTTP</td>
<td>Hypertext Transfer Protocol</td>
</tr>
<tr>
<td>SOC</td>
<td>System on Chip</td>
</tr>
<tr>
<td>TCP/IP</td>
<td>Transfer Control Protocol/ Internet Protocol</td>
</tr>
<tr>
<td>LED</td>
<td>Light Emitting Diode</td>
</tr>
</tbody>
</table>
CHAPTER ONE

1.0 Introduction

1.1 BACKGROUND AND MOTIVATION

Fish Farming is the principal form of aquaculture, it involves raising fish commercially in ponds or tanks mostly for food. In Nigeria, Out of 100 percent of fish harvested, 80 percent serves as the food consumed and the remaining 20 percent goes into the manufacturing of products such as fish oil, fertilizers, and animal food. [1]

Fisheries and aquaculture are integral parts of agriculture which were found to have the capacity to increase the country’s GDP (Gross Domestic Product) and it is a very lucrative business in Nigeria.

However, it is one thing to have a fish farm and it another thing to maintain a fish farm. Having worked in my father’s fish farm and my supervisors also having had some experience in the fish farm business, we observed that there were several challenges that accompanies fish farming business such as;

- Lack of time for the farmers
- Loss on investment due to fishes consuming each other
- High Cost of Labour
1.2 PROBLEM STATEMENT

Running a fish farm business requires a lot of time, it is capital intensive and all risk must be calculated to ensure that there is maximum return of and on investment. When it comes to feeding a fish farm, it requires a lot of money, time to constant supervise and feed the fishes on time, but farmers have other things on their schedule than sitting around all attending to their farmers. Sometimes, even external help that is paid for can be very risky as some if not all can be nonchalant.

This fishes tend to feed on the weak or injured among them on two scenarios which are;

- Improper feeding due to minimization of fish pallets
- Improper feeding due to late timing

Farmers lose a lot of money each time a single fish is consumed, as 1 kg weight of fish is worth three thousand naira (N 3,000).

1.3 PROJECT OBJECTIVE

The Aim of the project is to provide an efficient way of feeding a small scale fish farm.

1.4 OUTLINE OF THE REPORT

Chapter 1 introduces the system to be built, the motivation for it and the projects aims and objectives.

Chapter 2 outlines the requirements needed to develop the system, to ensure it meets the scope and functions efficiently. These requirements includes software and hardware requirements for building the system. A full description of each component used to make to execute the system
Chapter 3 explains the hardware and software design of the system. Details of how the hardware are coupled together and details of how the software is used to send information through the internet.

Chapter 4 concludes the report describing some of the challenges that was faced, describing the future goals, in terms of the improvement of the system.
CHAPTER TWO

2 REQUIREMENTS

The requirements describe the system to be built. It outlines the expected functionality of the system, as well as how it should behave upon implementation and use. The functional requirements constitute the specific functions of the system and what it should do. The hardware requirements specify the hardware components utilized in the system. The requirements are categorized under the three parts that make up the system: The Arduino, Server, and IoT technology.

2.1.1 Arduino

The arduino is a microprocessor board. The Integrated Development Environment (IDE) is use to write all instructions needed to be carried out by the system, from timing to reading the feeder container, all are done on the arduino. All wiring are done on the arduino board.

2.1.2 Server

The server is the bridge between the system and the using. The project uses Node-red as a local server to receive information from the sensor connected to the Arduino and sends an email to user using HTTP.

2.1.3 IoT technology

The whole of concept of device to human communication is based off on a new technology called IoT. This is the process of connecting all physical objects using sensor to communicate with each other or human through internet infrastructure.\[87\]
2.2 FUNCTIONAL REQUIREMENTS

The functional requirements specify the functions of the system. It defines the functionality of a system or one of its subsystems. It also depends upon the type of software, expected users and the type of system where the software is used. These are statements of what each part of the system should do and should also be described clearly about the system services in detail.\[BO8\]

2.2.1 Arduino

The Arduino is considered as the brains behind the functionality of the system. A microprocessor and micro controller that has all instruction uploaded into it. Instructions that controls various hardware components of system such as the servo motor and the ultrasonic sensor are found in the Arduino. In this system the Arduino handles all wiring of the various components. Using Multi-threading the Arduino handles all timing of the various hardware components\[BO9\] of the system.

Note

- The Arduino only execute what it has been programmed to execute

2.2.2 Server

This is the software functional requirement of the system. It serves as a bridge between Arduino and Farmer. In this project it accepts all readings from the Arduino based on the codes written, this could be from the sensor. Establishes connections with farmer through the Wi-Fi module and on the node-red interface decides what kind of message to send and finally sends message to the farmer through email.

Note

- There must be intranet or internet connection to ensure the server functions efficiently,
2.2.3 IoT Technology

This is the process of connecting all physical objects using sensor to communicate with each other or human through internet infrastructure. In other words this technology ensures proper connection of all physical objects and establishes communication between sensors and the server through the Wi-Fi.

2.3 HARDWARE REQUIREMENTS

The system is built with a few hardware components, these include:

- Arduino Uno
- Ultra-sonic sensor
- Wi-Fi Module
- Servo Motor
- Jumper Cables
- 5v to 3v power regulator
- LED light bulbs
The Arduino Uno

The Arduino Uno is a microcontroller board based on the ATmega328P (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with AC-to-DC adapter or battery to get started.

**The Arduino Uno Specifications**

<table>
<thead>
<tr>
<th>Specification</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microcontroller</td>
<td>ATmega328P</td>
</tr>
<tr>
<td>Operating Voltage</td>
<td>5V</td>
</tr>
<tr>
<td>Input Voltage (recommended)</td>
<td>7-12V</td>
</tr>
<tr>
<td>Input Voltage (limit)</td>
<td>6-20V</td>
</tr>
<tr>
<td>Digital I/O Pins</td>
<td>14 (of which 6 provide PWM output)</td>
</tr>
</tbody>
</table>
The Ultrasonic Sensor

An Ultrasonic sensor is a device that can measure the distance to an object by using sound waves. It measures distance by sending out a sound wave at a specific frequency and listening for that sound wave to bounce back.
The ESP8266 Wi-Fi Module

The ESP8266 Wi-Fi Module is a self-contained SOC with integrated TCP/IP protocol stack that can give any microcontroller access to your Wi-Fi network. The ESP8266 is capable of either hosting an application or offloading all Wi-Fi networking functions from another application processor.

Note

- The Wi-Fi chip requires a lot of power so as to function effectively.
- A 9 volts battery or batteries will not solve the problem. Create a cable connection that can be plug to an electric outlet through to the Arduino.
The Servo Motor

![Figure 2-4 5v Servo Motor](image)

A servo motor is a rotary actuator that allows for precise control of angular position. It consists of a motor coupled to a sensor for position feedback. In this project, this controls the lid of the feeder container and by extension controls the amount of feed disposed.

Note

- The servo motor is programmed to function based on angles. Codes are seen below in the appendix.
The Jumper Cables

The Jumper Cable is an electrical wire or group of them in a cable with a connector or pin at each end or sometimes without them, which is normally used to interconnect the components of a breadboard or other prototype or test circuit, internally or with other equipment or components, without soldering.

Note

- These wires or cable are very important to the system. Connections to the Arduino must be checked to ensure that no cable or wire is loose or weak in terms of cutting.
The 5v to 3v Voltage Regulator

![Figure 2-6 5v to 3v Voltage regulator](image)

Specifications:

- **Input Voltage:** DC 4.7V - 7V (Input must 1V greater than output)
- **Input Current:** max. 800mA
- **Output Voltage:** 3.3V DC ±0.05V
- **Size:** 2.5cm x 1.1cm

**Note**

- Absence of the regulator means the Wi-Fi chip would consume most of the power flowing into the Arduino board.
The LED lights

![LED lights on Arduino Board](image)

LED stands for 'Light Emitting Diode', a semiconductor device that converts electricity into light. LED lights are super energy efficient, using approximately 85% less energy than halogen or incandescent lighting – meaning significant savings on your power bills. [BO12]
3 DESIGN

3.1 HARDWARE DESIGN

The system consists of several hardware components such as: the Arduino Uno, the 5v servo motor, the ESP8266 Wi-Fi module and the Ultrasonic sensor. These components has been described above.

3.1.1 Connections on the Arduino Board

Several pins connections enable the Arduino to control several components of the system to work. The ultrasonic sensor has two pins: the trigger pin for sending out an output and the echo pin to detect its receipt pin. The LEDs used to indicate several level of feed container and also the connection to a Wi-Fi network. The servo motor is also connected using jumper cables. The ESP8266 is also connected through the 5v to 3v voltage regulator, not directly to board as it consumed more power than all component. All of which was coupled together into a power outlet back box. The pin connection were all connected through a bread board before coupled together and are shown below.

![Figure 3-1 Circuit Diagram of Sensor, Servo Motor and ESP8266 connected to Arduino Uno](image-url)
3.2 SOFTWARE DESIGN

The system consists of several software components such as the Arduino Integrated Development environment where all the codes were written and sent to the Arduino on how various hardware components will function and the node–red server which is initiated using command prompt.

3.2.1 Node-Red

Node-RED is a programming tool for wiring together hardware devices, APIs and online services in new and interesting ways. It provides a browser-based editor that makes it easy to wire together flows using the wide range of nodes in the palette that can be deployed to its runtime in a single-click. To further study about node-red visit: www.nodered.org

![Node-red Command prompt Initiation](image)
Figure 3-3 Node-red interface on the Web Browser
CHAPTER FOUR

4 Conclusion and future work

4.1 CONCLUSION

This project is a proof of concept for a controlled automated system and IoT, which aids in feeding a fish farm, monitors and sends information over the internet. Although a lot of challenges were faced, such as the appropriate container to use, how to power the Wi-Fi module and also the use of multithreading on the code, we able to overcome most difficulties eventually.

4.2 FUTURE IMPLEMENTATION

The following improvements can be implemented to further improve the system:

• Improve the structure of the system
• Develop a mobile app for the system
• Embed the system to other systems like a water changing system
• Improve means of communication especially for illiterate farmers
• Improve functionality
References


APPENDIX

#include <Servo.h>
#include<SoftwareSerial.h>

//led pins
const int ledPinFull = 8;
const int ledPinEmpt = 9;
const int ledPinMid = 10;
const int ledPinWifiOn = 6;

const int rstPin = 11;

//ultrasonic sensor pins
const int trigPin = 13;
const int echoPin = 12;

//esp8266 wifi
SoftwareSerial esp(2, 3); // TX, RX
String WSSID = "Mr. E";
String WPASS = "cb5655f0f6af";
String hostname = "fish";
bool listening = false;
bool reading = false;

bool hasWifiConnection = false;
bool hasSentInitial = false;

//servo
const int servoPin = 4;
const int servoRunTrials = 3; //change this to change the number of times the servo runs after startup
int servoPos = 0;
int servoRunCount = 0;
Servo servo;

//#define servoCycle 300000L //5 minutes
#define servoCycle 2000L //2 seconds
#define sensorCycle 1000L //every 1 seconds
#define espCycle 2000L //every 2 seconds

//threading
unsigned long servoThreadLastMillis = 0;
unsigned long sensorThreadLastMillis = 0;
unsigned long espThreadLastMillis = 0;

const int emptyCm = 20;
const int fullCm = 5;

float lastLevel = -1;
int lastState = -1;

const int STATE_EMPTY = 0;
const int STATE_CRITICAL = 1;
const int STATE_FULL = 2;

int state = -1;

void setup() {
  pinMode(ledPinFull, OUTPUT);
  pinMode(ledPinEmpt, OUTPUT);
  pinMode(ledPinMid, OUTPUT);
  pinMode(ledPinWifiOn, OUTPUT);

  void runSensor(){
    Serial.print("Running sensor");
    Serial.println();
    long duration, inches, cm;

    // The sensor is triggered by a HIGH pulse of 10 or more microseconds.
    // Give a short LOW pulse beforehand to ensure a clean HIGH pulse:
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);

    // Read the signal from the sensor: a HIGH pulse whose
duration is the time (in microseconds) from the sending
// of the ping to the reception of its echo off of an object.
duration = pulseIn(echoPin, HIGH);

    // convert the time into a distance
    //inches = microsecondsToInches(duration);
cm = microsecondsToCentimeters(duration);
  bool stateChange = false;
    Serial.print("NOTE:: actual cm: ");
    Serial.print(cm);
    Serial.println(" ");
    Serial.println(" ");
    if (cm <= fullCm){
      stateChange = (state != STATE_FULL);
      state = STATE_FULL;
      cm = fullCm;
    }else if( cm <= emptyCm){
      stateChange = (state != STATE_CRITICAL);

}else if( cm <= emptyCm){
      stateChange = (state != STATE_CRITICAL);
}
state = STATE_CRITICAL;
} else {
    stateChange = (state != STATE_EMPTY);
    state = STATE_EMPTY;
    cm = emptyCm;
}
indicateFeedLevel(cm);
if (stateChange) {
    alert();
}
// delay(1000);
}

void runServo() {
    if (servoRunCount >= servoRunTrials) {
        return;
    }
    Serial.print("Running servor");
    Serial.println();
    servo.attach(servoPin);
    Serial.print("Servo initial angle: ");
    Serial.print(servo.read());
    Serial.println();
    for (servoPos = 0; servoPos < 180; servoPos += 1) // goes from 0 degrees to 180 degrees
    {
        // in steps of 1 degree
        servo.write(servoPos); // tell servo to go to position in variable 'pos'
        delay(15); // waits 15ms for the servo to reach the position
    }
    delay(100);
    for (servoPos = 180; servoPos >= 1; servoPos -= 1) // goes from 180 degrees to 0 degrees
    {
        servo.write(servoPos); // tell servo to go to position in variable 'pos'
        delay(15); // waits 15ms for the servo to reach the position
    }