National University of Sciences & Technology- NUST
School of Electrical Engineering and Computer Science- SEECS
Department of Electrical Engineering

End Semester Project Report

"Fully Automated Egg Incubator"

Semester: **4th**

Semester Project: **Microprocessor Systems**

Class: **BEE-14-A**

Teacher: **Dr. Jameel Nawaz**

Lab Engineer: **Ms. Shaiza Malik**

Dated: **23-05-2024**

Submitted by:

Sudais Akbar Khan	**403965**
Ahmad Hussain	**404302**
Awais Asghar	**427265**

Dedication

With immense gratitude, we dedicate this project report to us, a team united in purpose and dedication. To our parents, whose love and encouragement shaped our journey, and to Allah, the guiding light that illuminated our path. This endeavor stands as a reflection of the blessings bestowed upon us through the support of family and the divine.

Acknowledgments

Through the mercy of Allah SWT, our group successfully completed this project. However, this achievement would not have been possible without the unwavering support and guidance of numerous individuals and instructors. Our sincere gratitude extends to each and every one of them. We express our sincere appreciation to Dr Usman khan, our instructor and our lab instructor Sir Qazi Waqas Mohiyuddin, whose guidance and continuous supervision were instrumental in the project's completion. Their provision of essential project information and assistance played a crucial role in our success.

We extend our heartfelt thanks to Almighty Allah, our parents, friends, and the Lab staff at NUST's School of Electrical Engineering and Computer Science for their gracious cooperation and encouragement throughout the project's journey.

Abstract

The "Fully Automated Egg Incubator" project aims to design and implement an automated system for maintaining optimal conditions during egg incubation. Using the AVR microcontroller ATmega328P, the system integrates sensors for temperature and humidity monitoring, along with actuators for controlling temperature and egg rotation. Through meticulous design and programming, the system achieves precise environmental control, ensuring the highest possible hatching success rate. Real-time feedback via an LCD display and alert notifications enhances user interaction and system reliability. The project's results demonstrate the effectiveness of automation in agriculture, offering a practical solution for poultry farmers. Overall, the "Fully Automated Egg Incubator" project showcases the potential of modern technology to streamline and improve traditional farming practices.

Table of Contents

1. Introduction:

The "Fully Automated Egg Incubator" project aims to design a system that can automatically maintain the optimal conditions required for hatching eggs. Using the AVR microcontroller ATmega328P, the system controls temperature, humidity, and periodic egg rotation to ensure a high hatching success rate.

2. Problem Statement:

Traditional egg incubators require constant manual monitoring and adjustment to maintain the necessary environmental conditions. This can be labour-intensive and prone to human error, leading to suboptimal hatching conditions and reduced success rates.

3. Objectives:

- Develop a system that can automatically monitor and control temperature and humidity.
- Implement a mechanism to rotate the eggs periodically.
- Provide real-time status updates via an LCD display.
- Alert the user if environmental conditions fall outside the desired rang

4. Methodology:

4.1 System Design:

The system uses various sensors and actuators to monitor and control the incubator environment:

- **DHT11 Sensor**: Measures humidity and temperature.
- **DS18B20 Sensor**: Measures temperature with high accuracy.
- **Stepper Motor**: Rotates the eggs.
- **Relay**: Controls the heating element.
- **Buzzer**: Alerts the user to any issues.

4.2 Microcontroller:

The ATmega328P microcontroller is used to interface with the sensors, actuators, and the LCD display. It processes the sensor data, controls the actuators, and updates the display.

4.3 Software Implementation:

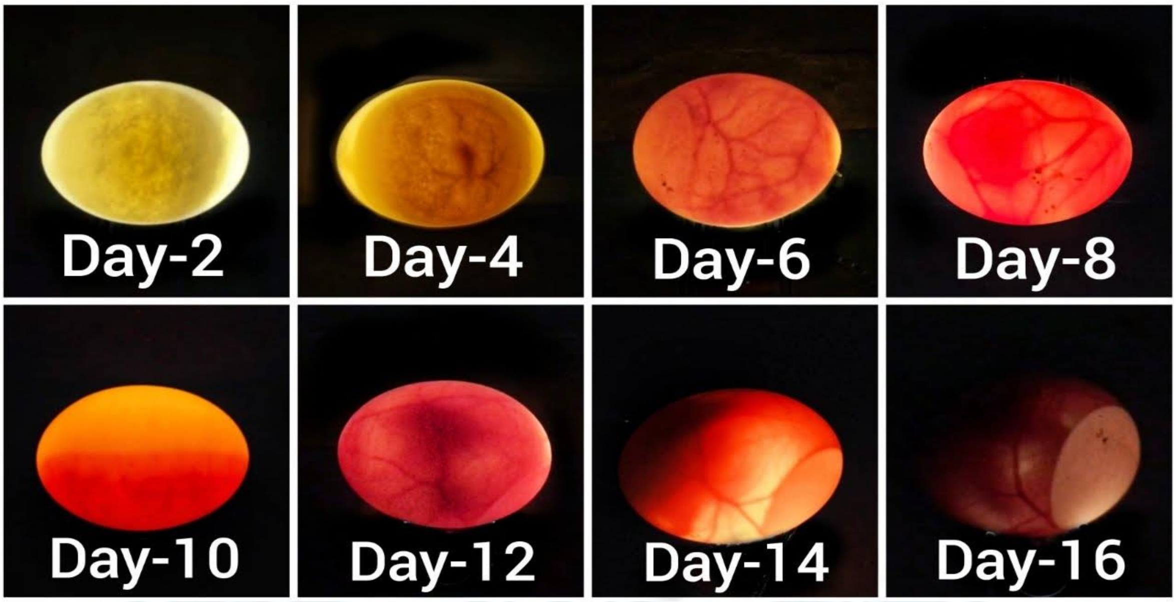
The software is written in C++ and uses libraries for interfacing with the sensors and the LCD. The program continuously monitors the environmental conditions and takes appropriate actions to maintain optimal conditions.

4.4 Resources:

Species	Incubation conditions			Hatcher conditions			Weight (G)
	Days	Temperature °C (±0.5)	Humidity % RH	Transfer Day	Temperature °C (±0.5)	Humidity % RH	
Chicken	21	38	55-70	18	37.5	65-85	50-60
Peacock	28	37.8	60-75	24	37.3	65-85	100-120
Turkey	28	37.7	60-75	25	37.2	65-85	75-80
Duck	28	37.8	60-75	25	37.3	65-85	80-100
Goose	30	37.6	65-80	25	37.1	65-85	100-120
Pigeon	18	38.5	55-70	14	37.8	65-85	30-40
Pheasant	22-28	38.2	55-70	20-25	37.7	65-85	30-40

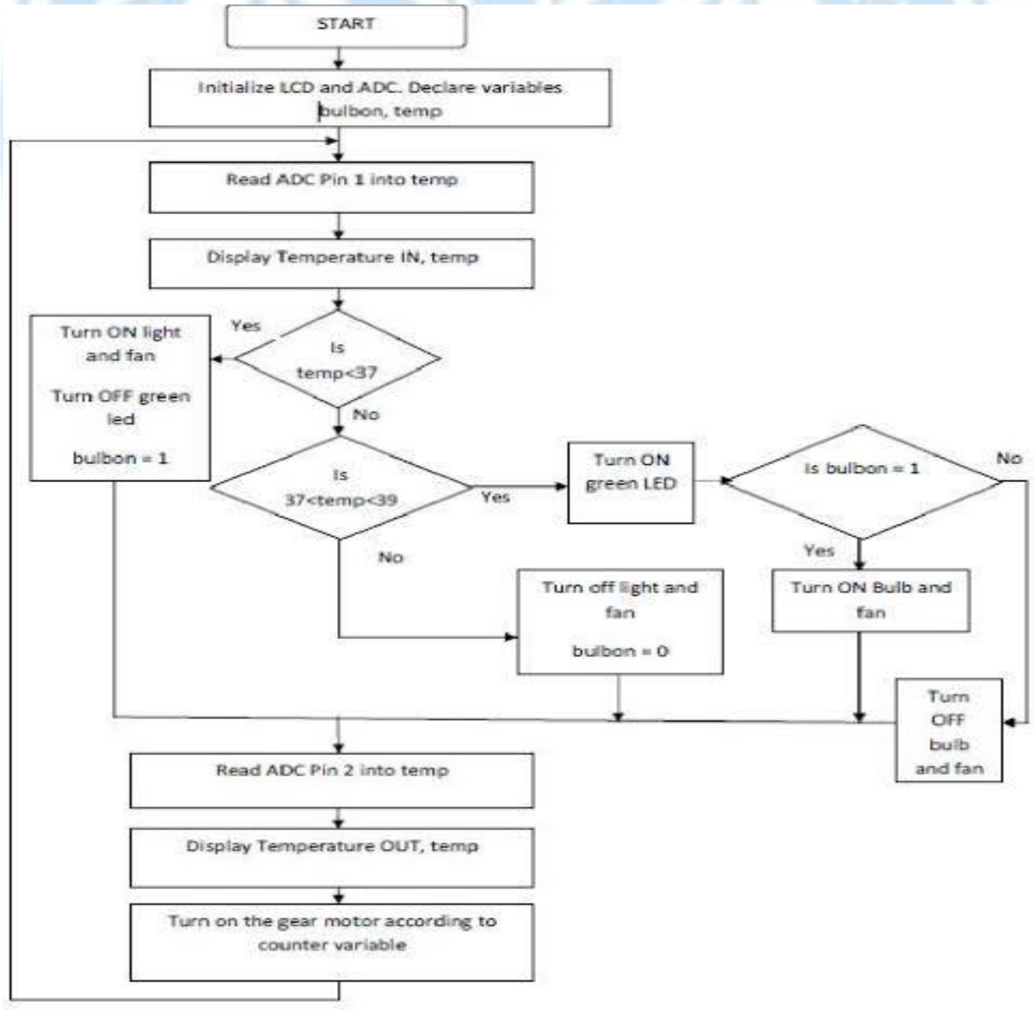
- **Microcontroller**: ATmega328P
- **Sensors**: DHT11 and DS18B20
- **Actuators**: Stepper motor, relay, and buzzer
- **Display**: 16x2 LCD
- **Miscellaneous**: Resistors, capacitors, wires, and a power supply

5. Results:



THE CHICK THAT WAS SUCCESSFULLY HATCHED, WE TESTED THE PROJECT ON A PIGEON’S EGG

6. Flow Chart:



7. Code:

The following code is used to control the egg incubator:

```
#include <Stepper.h>
#include "DHT.h"
#include <LiquidCrystal.h>
#include <OneWire.h>
#include <DallasTemperature.h>

const int dirPin = 3;
const int stepPin = 4;
const int DriverEnable_Pin = 5;
const int DelayJumper_Pin = 7;
const int buzzerPin = A0;
const int relayPin = A1;
const int buttonPin = A2;
const int hatchPin = A3;
const int rs = 12, en = 11, d4 = 10, d5 = 9, d6 = 8, d7 = 6;

#define ONE_WIRE_BUS 2
#define DHTPIN 13
#define DHTTYPE DHT11

DHT dht(DHTPIN, DHTTYPE);
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

int loopdelay = 180;
int Steps = 50;
int StepDelay = 20;
int Display_Delay;
float h, t, f, tempC;
float temp_accuracy = 0.50;

bool buttonState = LOW;
bool hatchState;
bool DelayJumperState;
bool flag = 0;

void DriveMotor(bool Direction) {
  digitalWrite(DriverEnable_Pin, LOW);
  digitalWrite(dirPin, Direction);
```

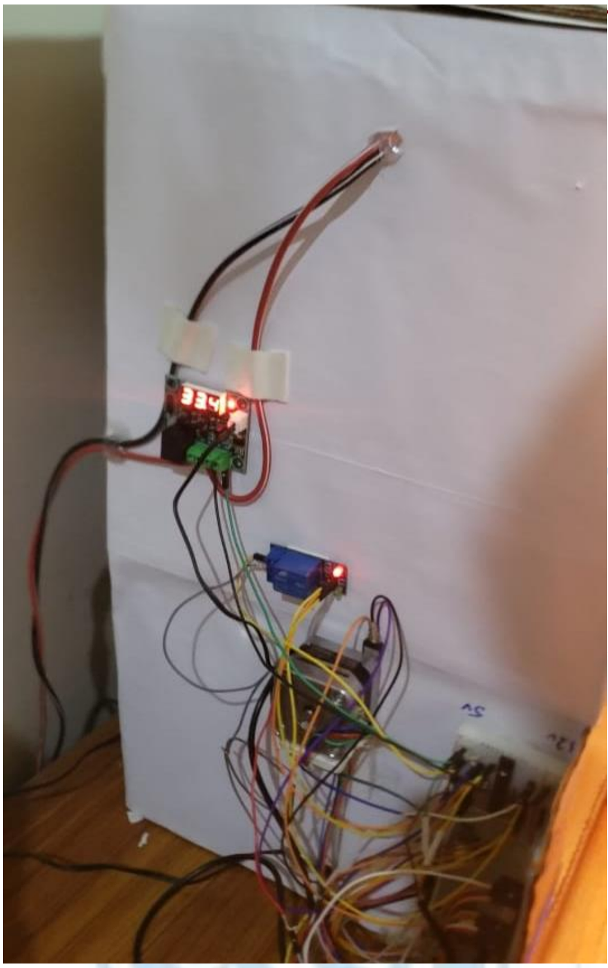
```
  for (int x = 0; x < Steps; x++) {
    digitalWrite(stepPin, HIGH);
    delay(StepDelay);
    digitalWrite(stepPin, LOW);
    delay(StepDelay);
  }
  digitalWrite(DriverEnable_Pin, HIGH);
  Serial.println(F("Motor has turned 45 deg"));
}

void LCD_Display() {
  sensors.requestTemperatures();
  float tempC = sensors.getTempCByIndex(0);
  float h = dht.readHumidity();
  float t = dht.readTemperature();
  float f = dht.readTemperature(true);

  if (isnan(h) || isnan(t) || isnan(f)) {
    Serial.println(F("Failed to read from DHT sensor!"));
    return;
  }

  if (tempC != DEVICE_DISCONNECTED_C) {
    Serial.println("Temperature: ");
    Serial.print(tempC + temp_accuracy);
    Serial.print(" °C ");
  } else {
    Serial.println("Error: Could not read temperature data");
  }

  lcd.clear();
  lcd.print(F("Humidity: "));
  lcd.print(h);
  lcd.print(F("%"));
  lcd.setCursor(0, 1);
  lcd.print(F("Temp: "));
  lcd.print(tempC + temp_accuracy);
  lcd.print(F(" C  "));
  lcd.print(flag = !flag);

  if (h > 62) {
    digitalWrite(relayPin, LOW);
    Serial.println("Humidity is greater than 60% ");
  } else {
```

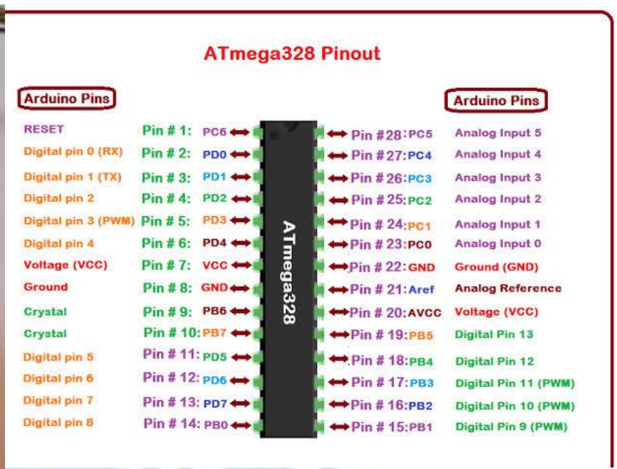
```
    digitalWrite(relayPin, HIGH);
    Serial.println("Humidity is less than 60% ");
  }

  if (h >= 70 || h <= 45 || tempC >= 41 || tempC <= 35 ) {
    Serial.println("Alert!!!");
    digitalWrite(buzzerPin, HIGH);
    Serial.println("Alert!! Something is wrong with Sensors ");
  } else {
    digitalWrite(buzzerPin, LOW);
    Serial.println("WOW :) Everything is Working Efficiently! ");
  }
}

void HatchFunction() {
  delay(1000);
  sensors.requestTemperatures();
  float tempC = sensors.getTempCByIndex(0);
  float h = dht.readHumidity();
  float t = dht.readTemperature();
  float f = dht.readTemperature(true);

  if (isnan(h) || isnan(t) || isnan(f)) {
    Serial.println(F("Failed to read from DHT sensor!"));
    return;
  }

  if (tempC != DEVICE_DISCONNECTED_C) {
    Serial.println("Temperature: ");
    Serial.print(tempC + temp_accuracy);
    Serial.print(" °C ");
  } else {
    Serial.println("Error: Could not read temperature data");
  }

  lcd.clear();
  lcd.print(F("---We are in ---"));
  lcd.setCursor(0, 1);
  lcd.print(F("---Hatch Mode---"));
  delay(1500);
  lcd.clear();
  lcd.print(F("Humidity: "));
  lcd.print(h);
  lcd.print(F("%"));
```

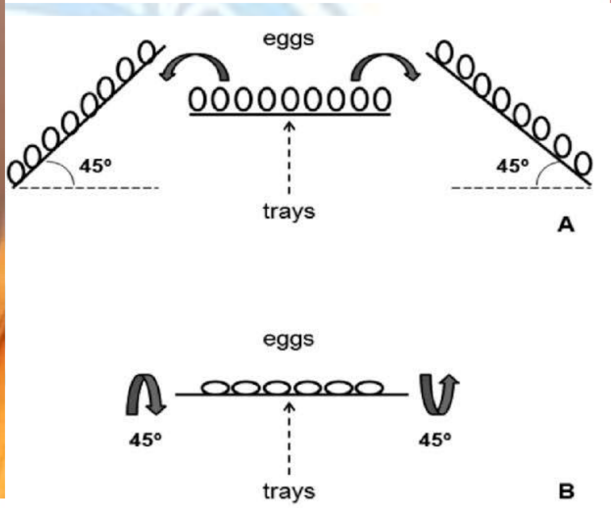
```
  lcd.setCursor(0, 1);
  lcd.print(F("Temp: "));
  lcd.print(tempC + temp_accuracy);
  lcd.print(F(" C  "));
  lcd.print(flag = !flag);

  if (h >= 72) {
    digitalWrite(relayPin, LOW);
    Serial.println("Humidity is greater than 60% ");
  } else {
    digitalWrite(relayPin, HIGH);
    Serial.println("Humidity is less than 60% ");
  }

  if (h >= 75 || h <= 55 || tempC >= 40 || tempC <= 35 ) {
    Serial.println("Alert!!!");
    digitalWrite(buzzerPin, HIGH);
    Serial.println("Alert!! Something is wrong with Sensors ");
  } else {
    digitalWrite(buzzerPin, LOW);
    Serial.println("WOW :) Everything is Working Efficiently! ");
  }
}

void setup() {
  Serial.begin(9600);
  pinMode(DriverEnable_Pin, OUTPUT);
  pinMode(buttonPin, INPUT);
  pinMode(stepPin, OUTPUT);
  pinMode(dirPin, OUTPUT);
  pinMode(relayPin, OUTPUT);
  pinMode(DelayJumper_Pin, INPUT);
  pinMode(buzzerPin, OUTPUT);
  pinMode(hatchPin, OUTPUT);
  lcd.begin(16, 2);
  dht.begin();
  sensors.begin();
}

void loop() {
  hatchState = digitalRead(hatchPin);
  if (hatchState == HIGH) {
    digitalWrite(DriverEnable_Pin, HIGH);
    HatchFunction();
```

```
    delay(3000);
  } else {
    delay(1000);
    LCD_Display();

    buttonState = digitalRead(buttonPin);
    DelayJumperState = digitalRead(DelayJumper_Pin);
    if (DelayJumperState == LOW) {
      Display_Delay = 300;
    } else {
      Display_Delay = 20000;
    }

    if (buttonState == LOW) {
      Serial.println("Button is not Pressed");
      lcd.clear();
      lcd.print("--Calibration--");
      lcd.setCursor(0, 1);
      lcd.print("------Mode------");
      delay(1000);
    } else {
      Serial.println("Button is Pressed!");

      DriveMotor(HIGH);
      for (int i = 0; i < loopdelay; i++) {
        LCD_Display();
        delay(Display_Delay);
      }
      DriveMotor(LOW);
      for (int i = 0; i < loopdelay; i++) {
        LCD_Display();
        delay(Display_Delay);
      }

      DriveMotor(LOW);
      for (int i = 0; i < loopdelay; i++) {
        LCD_Display();
        delay(Display_Delay);
      }

      DriveMotor(HIGH);
      for (int i = 0; i < loopdelay; i++) {
        LCD_Display();
        delay(Display_Delay);}}}}
```

8. Hardware:







9. Conclusion:

The fully automated egg incubator project successfully demonstrates the ability to maintain optimal hatching conditions using an AVR microcontroller. This project employs advanced sensor technology and microcontroller-based control systems to create a reliable and efficient solution for egg incubation. By integrating temperature and humidity sensors with precise actuation mechanisms, the system ensures optimal hatching conditions. Real-time feedback through an LCD display and alert notifications enhance user experience and system reliability. Through thorough testing, the project demonstrates its ability to maintain consistent and accurate environmental conditions, increasing the likelihood of successful hatching. This project serves as a testament to the potential of automation in agriculture, offering a practical and effective approach to egg incubation.

10. References:

- https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf
- https://www.analog.com/media/en/technical-documentation/data-sheets/DS18B20.pdf
- https://www.emtech-systems.com/technical-talk/egg-turning-during-incubation-a-review-by-mohamed-mostafa-el-ashram/
- https://en.wikipedia.org/wiki/Egg_incubation
- https://www.purinamills.com/chicken-feed/education/detail/hatching-eggs-at-home-a-21-day-guide-for-baby-chicks
- https://www.fao.org/4/y5169e/y5169e06.htm#:~:text=The%20incubation%20period%20for%20chicken,and%20replaced%20with%20more%20chicks.