### Incubator (egg)

## Submitted by,

## Alwin K G

## Alex Shaju

**CONTENT**

1. Abstract
2. Introduction
3. Block Diagram
4. Components
5. Program
6. Circuit Diagram
7. Working
8. Conclusion

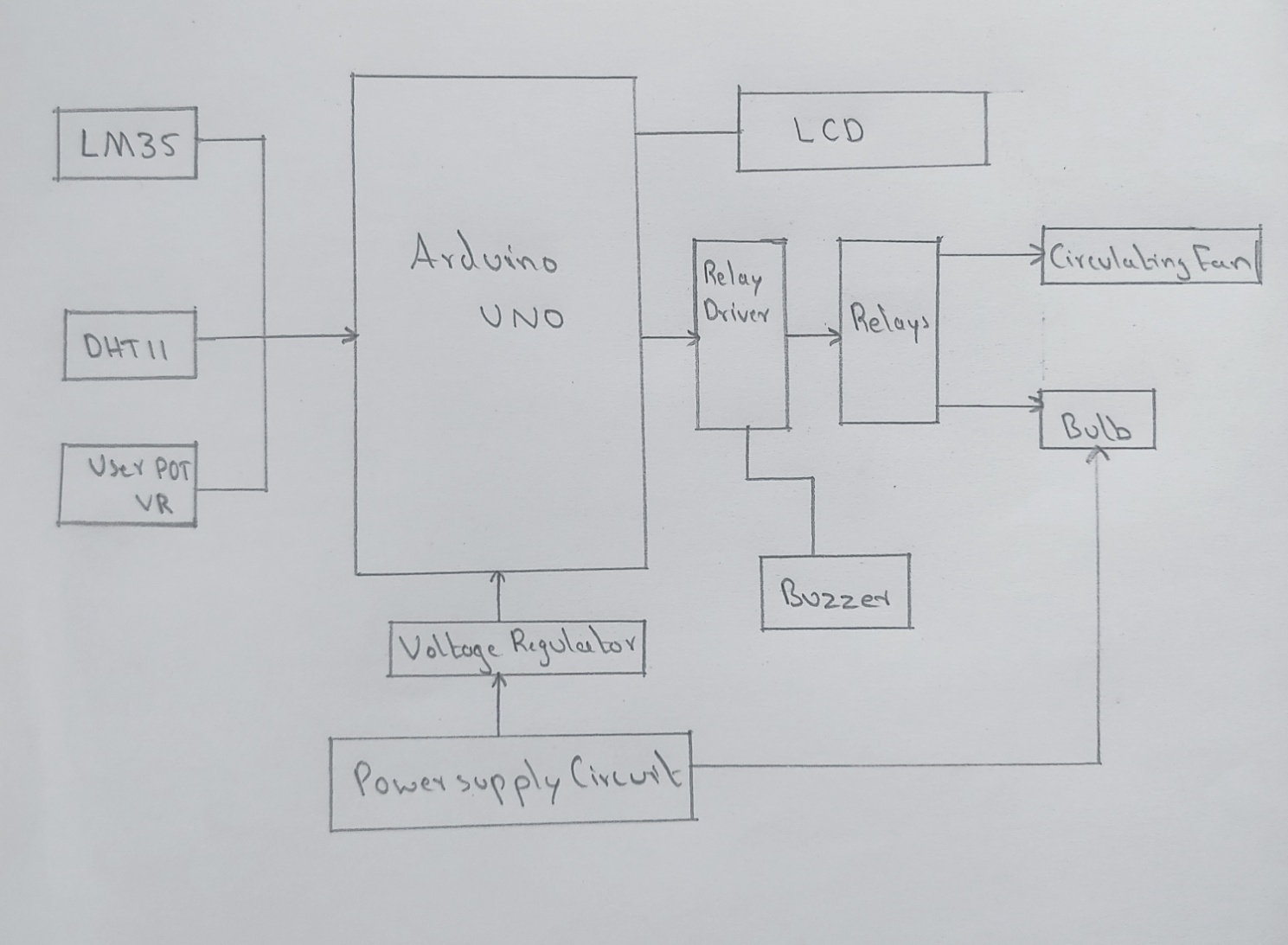
**ABSTRACT**

Due to its central position in the production chain, *in-ovo* development is influenced by pre-incubation factors that affect the quality of embryonated eggs and incubation conditions themselves, and both may influence egg hatchability and chick quality, as well as bird survival, growth performance, and phenotype in the field. The evolution of the incubation process over the years is characterized by significant scientific and technological development. Presently, the main current focuses of research are the manipulation of thermal incubation conditions, eggshell temperature, and the integrated effects of factors that influence incubation.

**INTRODUCTION**

Our project decribes about the way of working of an Incubator. An **incubator** is a device simulating [avian incubation](https://en.wikipedia.org/wiki/Avian_incubation) by keeping [eggs](https://en.wikipedia.org/wiki/Egg_(biology)) warm at a particular temperature range and in the correct [humidity](https://en.wikipedia.org/wiki/Humidity) with a turning mechanism to hatch them. The purpose of an incubator is that, It is used to regulate environmental conditions such as temperature, humidity and turning for successful hatching of the fertile eggs placed in an enclosure. It is often used for growing bacterial cultures, hatching eggs artificially, or providing suitable conditions for a chemical or biological reaction. Scientific knowledge on incubation acquired over the years shows that the physical factors to which the eggs are subjected before and during incubation determine the production efficiency of hatcheries and poultry farms. Nevertheless, little is known about the effective participation of the integrated effects of physical factors during ontogenetic development on the phenotype of poultry during the different stages of production.

**BLOCK DIAGRAM**



**COMPONENTS**

Arduino UNO : Here, we have used an Arduino UNO to control the operation of the system. Arduino UNO is a microcontroller board based on ATmega328P. It has 14 digital input/output pins, 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header and a reset button.

LM35 Temperature Sensor : LM35 is a temperature measuring device having an analog output voltage proportional to the temperature. It does not require any external calibration circuitary. The sensitivity of LM35 is 10mV/degree Celsius.

Humidity Sensor : To control the humidity of the system, we need to measure the humidity first. To measure humidity, here in our project we’ll use DHT11 as our humidity sensor.

LCD : (Liquid Crystal Display) 16x2 LCD means it can display 16 characters per line and there are 2 such lines. In this LCD each character is displayed in 5x7 pixel matrix. The 16x2 intelligent alphanumeric dot matrix display is capable of displaying 224 different characters and symbols. This LCD has two registers, namely , Command and Data.

Circulating Fan : We can use the cooling fan as our circulating fan. One fan will be good for our incubator. These circulating fans should be positioned inside in a circular way. This will maintain good air circulation as well as it will reduce concentrated heat build-up. A positioning like this will be enough to maintain proper air circulation.

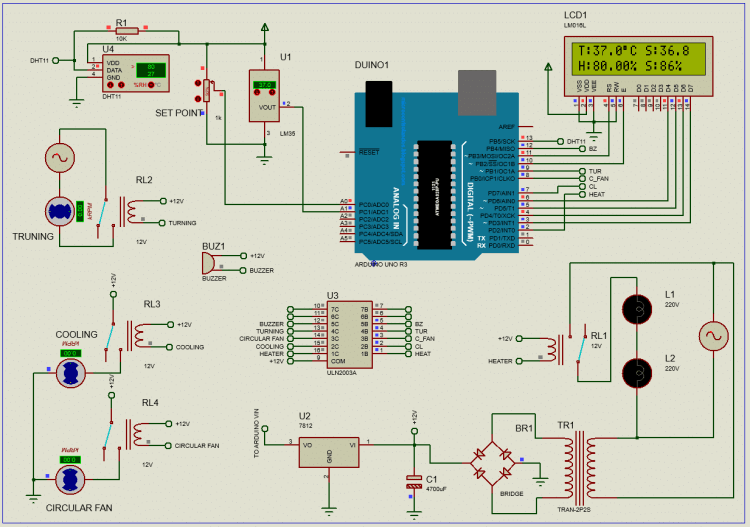
Voltage Supply : Used to give voltage supply to the system.

## Relay switch : Relays are good for AC bulb switching. Although there are multiple ways to do this job as a basic one we can use a relay for both bulb and fan control.

Bulb : Incubator was designed with the 80-watt light bulb. Bulb was used as heat source and was installed inside the box in such a way that it was connected with the Digital Temperature controller and 12v supply.

Apart from these we have used wires, female port, cardboard box, thermocol, plug, holder, and nylon tie, etc…

**CIRCUIT DIAGRAM**



**PROGRAM**

#include <LiquidCrystal.h>  
#include <Servo.h>  
#include <dht.h>  
#define DHT11 A0  
const int ok = A1;  
const int UP = A2;  
const int DOWN = A3;  
const int bulb = A4;  
const int vap = A5;  
const int rs = 12;  
const int en = 11;  
const int d4 = 5;  
const int d5 = 4;  
const int d6 = 3;  
const int d7 = 2;  
int ack = 0;  
int pos = 0;  
int sec = 0;  
int Min = 0;  
int hrs = 0;  
int T\_threshold = 25;  
int H\_threshold = 35;  
int SET = 0;  
int Direction = 0;  
boolean T\_condition = true;  
boolean H\_condition = true;  
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);  
Servo motor;  
dht DHT;  
void setup()  
{  
pinMode(ok, INPUT);  
pinMode(UP, INPUT);  
pinMode(DOWN, INPUT);  
pinMode(bulb, OUTPUT);  
pinMode(vap, OUTPUT);  
digitalWrite(bulb, LOW);  
digitalWrite(vap, LOW);  
digitalWrite(ok, HIGH);  
digitalWrite(UP, HIGH);  
digitalWrite(DOWN, HIGH);  
motor.attach(7);  
motor.write(pos);  
lcd.begin(16, 2);  
Serial.begin(9600);  
lcd.setCursor(5, 0);  
lcd.print("Digital");  
lcd.setCursor(4, 1);  
lcd.print("Incubator");  
delay(1500);  
}  
void loop()  
{  
if (SET == 0)  
{  
lcd.clear();  
lcd.setCursor(0, 0);  
lcd.print("Set Temperature:");  
lcd.setCursor(0, 1);  
lcd.print(T\_threshold);  
lcd.print(" \*C");  
while (T\_condition)  
{  
if (digitalRead(UP) == LOW)  
{  
T\_threshold = T\_threshold + 1;  
lcd.setCursor(0, 1);  
lcd.print(T\_threshold);  
lcd.print(" \*C");  
delay(200);  
}  
if (digitalRead(DOWN) == LOW)  
{  
T\_threshold = T\_threshold - 1;  
lcd.setCursor(0, 1);  
lcd.print(T\_threshold);  
lcd.print(" \*C");  
delay(200);  
}  
if (digitalRead(ok) == LOW)  
{  
delay(200);  
T\_condition = false;  
}  
}  
lcd.clear();  
lcd.setCursor(0, 0);  
lcd.print("Set Humidity:");  
lcd.setCursor(0, 1);  
lcd.print(H\_threshold);  
lcd.print("%");  
delay(100);  
while (H\_condition)  
{  
if (digitalRead(UP) == LOW)  
{  
H\_threshold = H\_threshold + 1;  
lcd.setCursor(0, 1);  
lcd.print(H\_threshold);  
lcd.print("%");  
delay(100);  
}  
if (digitalRead(DOWN) == LOW)  
{  
H\_threshold = H\_threshold - 1;  
lcd.setCursor(0, 1);  
lcd.print(H\_threshold);  
lcd.print("%");  
delay(200);  
}  
if (digitalRead(ok) == LOW)  
{  
delay(100);  
H\_condition = false;  
}  
}  
SET = 1;  
}  
ack = 0;  
int chk = DHT.read11(DHT11);  
switch (chk)  
{  
case DHTLIB\_ERROR\_CONNECT:  
ack = 1;  
break;  
}  
if (ack == 0)  
{  
lcd.clear();  
lcd.setCursor(0, 0);  
lcd.print("Temp:");  
lcd.print(DHT.temperature);  
lcd.setCursor(0, 1);  
lcd.print("Humidity:");  
lcd.print(DHT.humidity);  
if (DHT.temperature >= T\_threshold)  
{  
delay(3000);  
if (DHT.temperature >= T\_threshold)  
{  
digitalWrite(bulb, LOW);  
}  
}  
if (DHT.humidity >= H\_threshold)  
{  
delay(3000);  
if (DHT.humidity >= H\_threshold)  
{  
digitalWrite(vap, LOW);  
}  
}  
if (DHT.temperature < T\_threshold)  
{  
delay(3000);  
if (DHT.temperature < T\_threshold)  
{  
digitalWrite(bulb, HIGH);  
}  
}  
if (DHT.humidity < H\_threshold)  
{  
delay(3000);  
if (DHT.humidity < H\_threshold)  
{  
digitalWrite(vap, HIGH);  
}  
}  
sec = sec + 1;  
if (sec == 60)  
{  
sec = 0;  
Min = Min + 1;  
}  
if (Min == 60)  
{  
Min = 0;  
hrs = hrs + 1;  
}  
if (hrs == 8 && Min == 0 && sec == 0)  
{  
for (pos = 0; pos <= 180; pos += 1)  
{  
motor.write(pos);  
delay(25);  
}  
}  
if (hrs == 16 && Min == 0 && sec == 0)  
{  
hrs = 0;  
for (pos = 180; pos >= 0; pos -= 1)  
{  
motor.write(pos);  
delay(25);  
}  
}  
}  
if (ack == 1)  
{  
lcd.clear();  
lcd.setCursor(0, 0);  
lcd.print("No Sensor data.");  
lcd.setCursor(0, 1);  
lcd.print("System Halted.");  
digitalWrite(bulb, LOW);  
digitalWrite(vap, LOW);  
}  
delay(1000);  
}

**WORKING**

An incubator is based on the principle that microorganisms require a particular set of parameters for their growth and development. All incubators are based on the concept that when organisms are provided with the optimal condition of temperature, humidity, oxygen, and carbon dioxide levels, they grow and divide to form more organisms.

After sensing the input parameters, Arduino decides when to turn on the bulb. The temperature should be kept around 36.8’C. When the temperature is low, bulb will be turned on automatically. If the temperature is higher than 0.5’C from the setpoint, sensors will be turned off. At the same time, the buzzer will alarm at a temperature that is 2’C higher than the set point. The circulating fan is controlled according to the bulb. When the bulb is on, the circulating fan is on. When the bulb is off but the temperature is higher than the setpoint, the circulating fan runs periodically(a few moments on, a few moments off).

The turning motor will be turned on for 3-4 seconds after every 30 minutes.

**CONCLUSION**

Physical exchanges between eggs and environment are required for *in-ovo* development. Deficient exchanges negatively affect the incubation process, while excessive exchanges may improve incubation efficiency. Physical exchanges depend firstly on eggshell porosity and conductance and on temperature and relative humidity differences between the eggs and the environment. These factors are maternally influenced by egg quality, egg storage conditions and duration, and the incubation condition. The maternal effects on the physical exchanges show that the optimal storage and incubation conditions vary with breeder age and egg weight. However, optimal storage and incubation conditions as a function of egg weight or weight range still need to be established. Although this proposal seems to be unrealistic at first sight, it may be feasible as the control systems of incubation physical condition are further technologically developed, allowing their easier and quicker determination. In addition, incubation conditions, which are essential for maximizing the production efficiency of hatcheries, may be optimized.

This Project gave us a wonderful experience about the way of working of electronics. We have learned many things during this journey. We learned that to make a simple model of Incubator requires hard work , dedication and knowledge about things based on electronics.