SEMESTER 2 OF SESSION 23/24 (A232)
SKIH3113 SENSOR BASED SYSTEM

ASSIGNMENT 1: Egg Monitoring System

PREPARED BY:

NAME	MATRIC NO.
MOHAMAD TAHFIZZUDEEN BIN MOHD HANAPI	288334

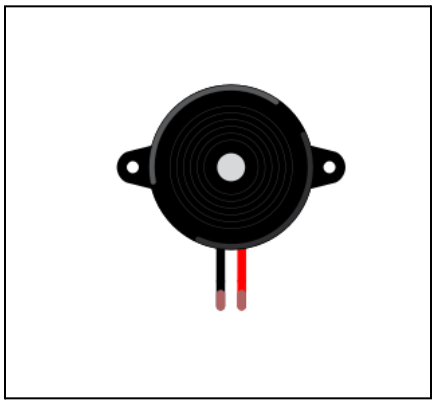
……..
Mohamad Tahfizzudeen Bin Mohd Hanapi

Table Of Content

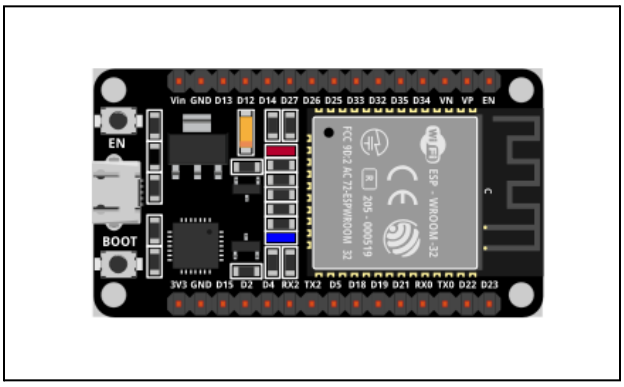
1.0 Introduction

Egg has The system will monitor the egg environment to produce the best egg product. The DHT22 will measure the temperature and humidity of the egg/egg container. At the same time, the ultrasonic will detect if any obstacles/animals/bugs can destroy the egg. For normal conditions, the green LED will light on and the other output components will remain turned off. If there are any obstacles detected nearly 10 cm from the ultrasonic, the system will trigger the triggerAlert functions where the green LED will turn off, the red LED will light on, and the servo will move 180 degrees to get rid of the obstacles/animals and the buzzer will turn on to give alert to the monitor/owner.

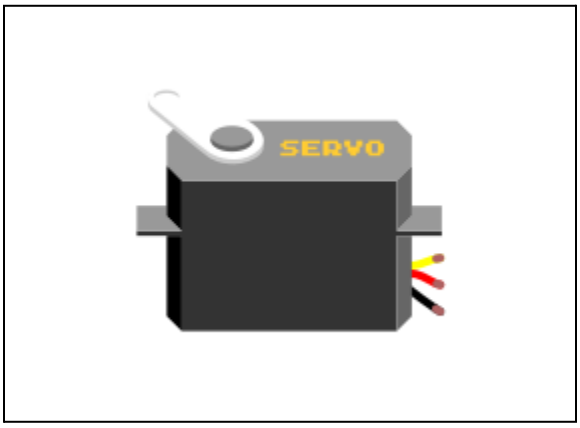
2.0 Components



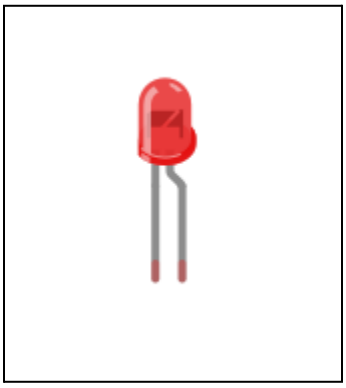
Buzzer



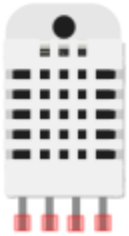
ESP32



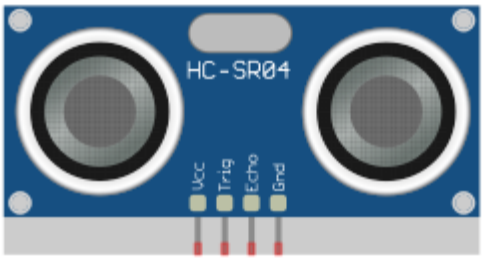
Servo Motor



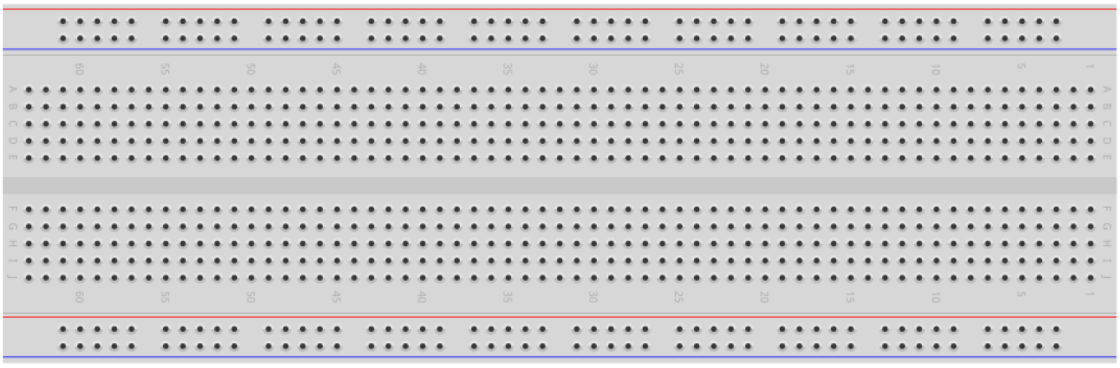
LED



DHT22 Sensor

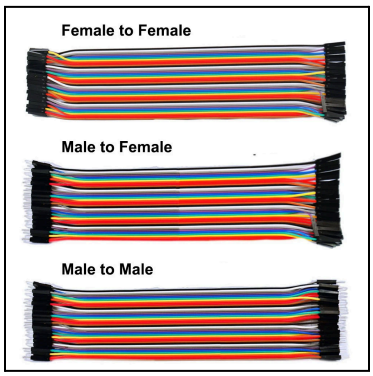


Ultrasonic Sensor



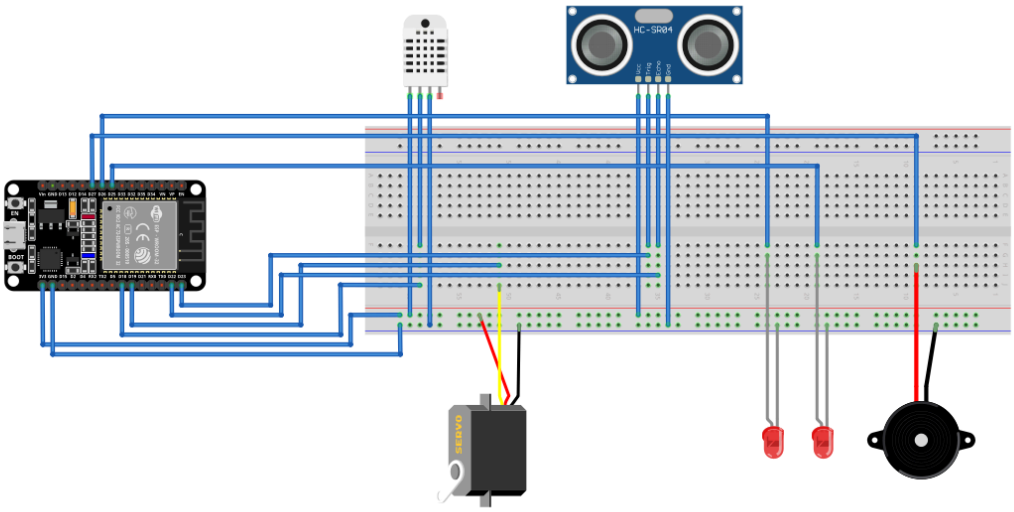
Breadboard

USB to Micro USB cable



Jumper wyre

3.0 System Design



4.0 Framework Code

```
#include <DHT.h>
#include <ESP32Servo.h>

#define DHTPIN 18       // DHT22 Sensor connected to GPIO18
#define SERVOPIN 19     // Servo signal pin connected to GPIO19
#define ECHO_PIN 22     // Ultrasonic sensor echo pin connected to GPIO22
#define TRIG_PIN 23     // Ultrasonic sensor trigger pin connected to GPIO23
#define RED_LED 26      // Red LED pin connected to GPIO26
#define GREEN_LED 25    // Green LED pin connected to GPIO25
#define BUZZER 27       // Buzzer pin connected to GPIO27

#define DHTTYPE DHT22   // DHT sensor type

#define MIN_TEMP 30       // Minimum temperature threshold (in Celsius)
#define MAX_DISTANCE 10   // Maximum distance threshold (in cm)

DHT dht(DHTPIN, DHTTYPE);   // Initialize DHT object
Servo servo;                // Initialize Servo object

void setup() {
  Serial.begin(115200);                       // Start serial communication
  pinMode(RED_LED, OUTPUT);                   // Set red LED pin as output
  pinMode(GREEN_LED, OUTPUT);                 // Set green LED pin as output
  pinMode(BUZZER, OUTPUT);                    // Set buzzer pin as output
  pinMode(TRIG_PIN, OUTPUT);                  // Set trigger pin as output
  pinMode(ECHO_PIN, INPUT);                   // Set echo pin as input

  dht.begin();                                // Initialize DHT sensor
  servo.attach(SERVOPIN);                     // Attach servo to SERVOPIN
}
```

```
31
32  void loop() {
33    float temperature = dht.readTemperature();    // Read temperature from DHT sensor
34    float humidity = dht.readHumidity();          // Read humidity from DHT sensor
35    float distance = getDistance();               // Read distance from ultrasonic sensor
36
37    Serial.print("Temperature: ");                // Print temperature value
38    Serial.print(temperature);
39    Serial.print(" °C\tHumidity: ");             // Print humidity value
40    Serial.print(humidity);
41    Serial.print(" %\tDistance: ");              // Print distance value
42    Serial.print(distance);
43    Serial.println(" cm");
44
45    digitalWrite(RED_LED, LOW);                  // Turn off red LED
46
47    if (distance > MAX_DISTANCE) {               // Check if distance is greater than maximum distance threshold
48      digitalWrite(GREEN_LED, HIGH);             // Turn on green LED
49      digitalWrite(BUZZER, LOW);                 // Turn off buzzer
50      if (servo.attached()) {                     // Check if servo is attached
51        servo.detach();                           // Detach servo
52      }
53    } else {
54      digitalWrite(GREEN_LED, LOW);              // Turn off green LED
55      digitalWrite(BUZZER, HIGH);                // Turn on buzzer
56      if (!servo.attached()) {                    // Check if servo is not attached
57        servo.attach(SERVOPIN);                   // Attach servo
58      }
59      triggerAlert();                             // Call triggerAlert function
60    }
61
62    delay(2000);                                  // Delay for 2 seconds
63  }
65  float getDistance() {
66    digitalWrite(TRIG_PIN, LOW);                 // Set trigger pin to LOW
67    delayMicroseconds(2);                         // Wait for 2 microseconds
68    digitalWrite(TRIG_PIN, HIGH);                // Set trigger pin to HIGH
69    delayMicroseconds(10);                        // Wait for 10 microseconds
70    digitalWrite(TRIG_PIN, LOW);                 // Set trigger pin to LOW
71    long duration = pulseIn(ECHO_PIN, HIGH);      // Measure duration of echo pin
72    return duration * 0.034 / 2;                  // Calculate distance in cm
73  }
74
75  void triggerAlert() {
76    digitalWrite(GREEN_LED, LOW);                // Turn off green LED
77    digitalWrite(RED_LED, HIGH);                 // Turn on red LED
78    digitalWrite(BUZZER, HIGH);                  // Turn on buzzer
79    servo.write(180);                             // Rotate servo to 180 degrees
80    delay(1000);                                  // Delay for 1 second
```

5.0 Additional Features

Adding the real time sensor to get the real time data by using the Real Time Censor (RTC) and the Firebase software may increase the effectiveness of the system. Temperature, humidity, and distance readings from sensors would all be continuously streamed to the Firebase real-time database by the system. A web or mobile application would communicate with the Firebase database by connecting and continuously listening for updates. The program would get instant updates when fresh sensor data is written to the database, allowing users to keep an eye on the environmental conditions surrounding the eggs in real-time. Users get real-time insights into the incubation environment because of Firebase's real-time synchronization, which makes sure that changes in sensor data are mirrored instantly across all linked clients. Furthermore, the monitoring interface might be secured by implementing Firebase Authentication, which would only permit authorized users to view the real-time data.

6.0 Links

The components, Fritzing design and Arduino code can be accessed here:
Github link:
https://github.com/TahfizHanapi/SKIK3113-Sensor-Based-SYstem.git

Video link:
https://youtu.be/vsWIxgK_4kU?si=jfGZ8vFvyXnCvaqC