**CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CH.NO** | **TITLE OF TABLE** | **PAGE NO** |

**CHAPTER 1: INRODUCTION 2**

**CHAPTER 2:** **WORKING PROCESS 4**

**CHAPTER 3: CONCLUSION 18**

**CHAPTER 4:REFERENCES 19**

**ABSTRACT:**

From last few years, the chicken production in the world has been increasing gradually because of standardized farming management and good manufacturing practices. According to world’s agricultural produce survey, chicken is the most favourite produce, since it is a nutrient rich food providing high protein, low fat and low cholesterol, and lower energy than other kinds of poultries. From last few days around the world, there has been an increased level of awareness regarding the safety of food products like chickens and there has been a high demand for good quality chicken food.This work aims to provide details on how to build an IoT Enabled Smart Poultry Farm using low cost commodity hardware and open source software. A comprehensive system was built usingNodeMCU, DHTsensor, Ultrasonicsensor, MQ2sensor (Gassensor), Servomotor.The system has been thoroughly investigated for various physical parameters associated with effective poultry management which includes temperature and to on or off the motor. It was found thatthe system not only monitors these parameters, but also regulates these parameters effectively. The framework wasobserved to be very useful for farmers as they could easily access and control the system remotely using their handheldmobile devices. The system reduces human intervention, savestime, optimizes resource utilization and increases poultryproduction.

**INTRODUCTION:**

INDIA is considered as an agricultural wealthycountry in terms of food and environmental resources. Nevertheless, such prosperity was gradually regressed directlycontributing to a low agricultural productivity and farmerincomes. The farmers additionally lacked insights in agricultural marketing strategies and high-quality productionplanning. According to world’s agricultural produce, chicken is themost favoured produce, since it is a nutrient-rich food providinghigh protein, low fat and cholesterol, and lower energy anotherkind of poultries. Also, it is quite easy to look after andpropagate its species. For 5 years, the chicken productionhas been increasing on an average of 4.63% yearly because ofstandardized farming management and good manufacturingpractices, leading to more chicken consumption and anincreased export number of both domestic and internationaldestinations. On the other hand, a lacking of labour inchicken production processes has affected fresh chickenexport, which is found to be the principal problem. Another significant obstacle can be wrong knowledge sharingand folk wisdom in chicken farming which effects efficiency.

This study aims to set up a new model by using a moderntechnology applied to chicken farming known as a "SmartFarm" or "Intelligent Farm", which is expected to clear upthe IOT Based smart poultry farming using commodity hardware and software farming. Smart Farm could perceiveany changed information derived from a semi-automaticmicroprocessor, alarming all notification to a connected personalcomputer. The farm monitoring could be conducted viaapplication programs on smart phones for convenience use,time saving, and reduced labour dependency, cost.

**PROBLEM STATEMENT:**

This project proposes the new model by using advanced modern technology to make traditional chicken farming smarter. Smart farm gives the environmental parameter statistics like temperature, humidity, smoke, weather condition etc. The health of chicken depends on the environment in the poultry farm. If the environmental condition is not suitable then there may be problem with growth of the chicken and their health issues. Poultry farm are designed in such way that, environmental conditions can be altered by providing facilities like ventilation, cooling and lightening on rough, wall and floor.

We can remotely monitor environmental parameters in a poultry farm. we can monitor the water level in the tank.

**WORKING PROCESS:**

The main aim of the project is to make the traditional poultry farming smarter by using IoT.Here we are designing poultry in such a way that environmental conditions can be altered.

Firstly,by using gas sensor we can detect the harmful gases in the farm.So we will use MQ2 sensor and we can display that message if any harmful gas is detected in the farm by using OLED and a message will be sent to the owner of the farm.For that he has to install the app in his phone.  If the analogsensor value is greater than 45 then the person will get a message called DANGER.

After that DHT 11 sensor is used to detect the temperature in the farm. If the temperature is greater than 35 in the farm then the owner will get a message that the Temperature is HIGH.

But if the temperature is less than 25 the person should get a message that temperature is LOW.

Later by using Ultrasonic Sensor the owner of the farm can be able to know the water level in the tank. If the distance falls below 5 then the person will get a message like NO water for Hens.

But if the distance raises above 10 then the person will get a message like water overflow.

Based on ultrasonic sensor servomotor will rotate 90 degrees and after 2000 ns delay it reaches again its original position.

**COMPONENTS USED:**

A comprehensive system was built using:

1.Hardware

2.Software

The hardware components are:

* NodeMCU
* DHT sensor
* Ultrasonic sensor
* MQ2 sensor (Gas sensor)
* Servomotor
* OLED (to display)

**1.NodeMCU:**

NodeMCU is an open source [IoT](https://en.wikipedia.org/wiki/Internet_of_Things) platform. The term "NodeMCU" by default refers to the firmware rather than the development kits. NodeMCU provides access to the [GPIO](https://en.wikipedia.org/wiki/General-purpose_input/output) (General Purpose Input/Output).It includes [firmware](https://en.wikipedia.org/wiki/Firmware) which runs on the [ESP8266](https://en.wikipedia.org/wiki/ESP8266) [Wi-Fi](https://en.wikipedia.org/wiki/Wi-Fi) [SoC](https://en.wikipedia.org/wiki/System_on_a_chip) from [Espressif Systems](https://en.wikipedia.org/w/index.php?title=Espressif_Systems&action=edit&redlink=1), and hardware which is based on the ESP-12 module.

An open source firmware and development kit that helps you to prototype your IoT product within a few Lua script lines.



**Features:**

Open-source, Interactive, Programmable, Low cost, Simple, Smart, WI-FI enabled.

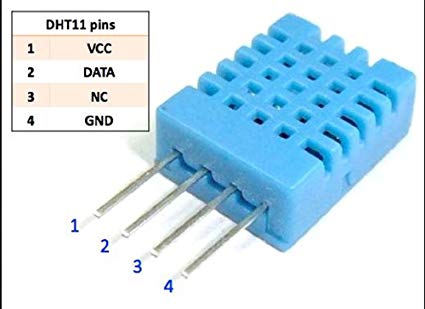
**2.DHT Sensor:**

This DHT11 Temperature and Humidity Sensor features a calibrated digital signal output with the temperature and humidity sensor capability. It is integrated with a high-performance 8-bit microcontroller. Its technology ensures the high reliability and excellent long-term stability.  This sensor includes a resistive element and a sensor for wet NTC temperature measuring devices. It has excellent quality, fast response, anti-interference ability and high performance.

Each DHT11 sensors features extremely accurate calibration of humidity calibration chamber. The calibration coefficients stored in the OTP program memory, internal sensors detect signals in the process, we should call these calibration coefficients. The single-wire serial interface system is integrated to become quick and easy. Small size, low power, signal transmission distance up to 20 meters, enabling a variety of applications and even the most demanding ones. The product is 4-pin single row pin package. Convenient connection, special packages can be provided according to users need.

**Specification:**

* Supply Voltage: +5 V
* Temperature range :0-50 °C error of ± 2 °C
* Humidity :20-90% RH ± 5% RH error
* Interface: Digital

****

**3.Ultrasonic Sensor:**

As the name indicates, Ultrasonic sensors measure distance by using ultrasonic waves. The sensor head emits an ultrasonic wave and receives the wave reflected back from the target. Ultrasonic Sensors measure the distance to the target by measuring the time between the emission and reception.

****

**Ultrasonic Sensor Pin Configuration:**

|  |  |  |
| --- | --- | --- |
| **Pin Number** | **Pin Name** | **Description** |
| 1 | Vcc | The Vcc pin powers the sensor, typically with +5V |
| 2 | Trigger | Trigger pin is an Input pin. This pin has to be kept high for 10us to initialize measurement by sending US wave. |
| 3 | Echo | Echo pin is an Output pin. This pin goes high for a period of time which will be equal to the time taken for the US wave to return back to the sensor. |
| 4 | Ground | This pin is connected to the Ground of the system. |

**4.MQ2 Sensor:**

MQ2 gas sensor can be used to detect the presence of LPG, Propane and Hydrogen, also could be used to detect Methane and other combustible steam, it is with low cost and suitable for different application. Sensor is sensitive to flammable gas and smoke. Smoke sensor is given 5 volts to power it.

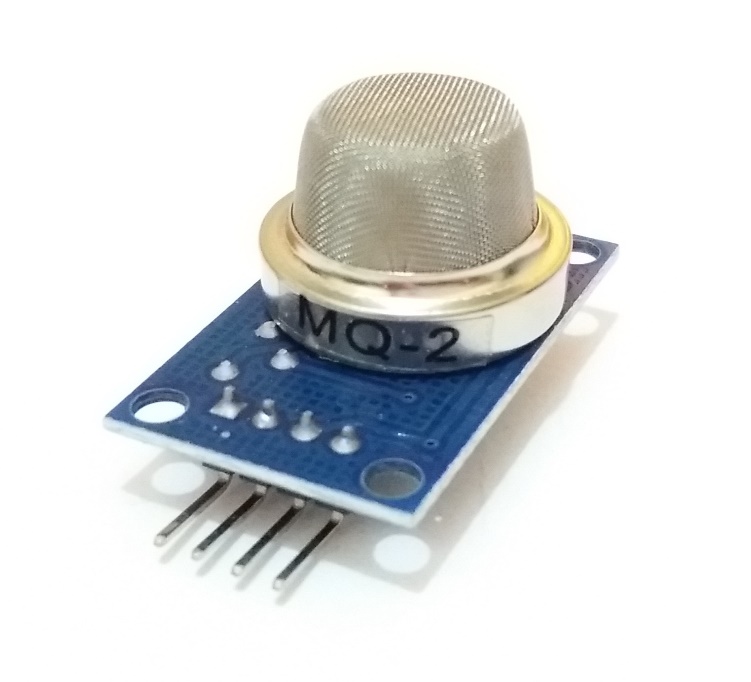
**Pin Configuration:**

1.Vcc - This pin powers the module, typically the operating voltage is +5V

2.Ground - Used to connect the module to system ground

3.Digital Output - You can also use this sensor to get digital output from this pin, by setting a threshold value using the potentiometer

4.Analog Output - This pin outputs 0-5V analog voltage based on the intensity of the gas



**5.Servomotor:**

A servomotor is a [rotary actuator](https://en.wikipedia.org/wiki/Rotary_actuator) or [linear actuator](https://en.wikipedia.org/wiki/Linear_actuator) that allows for precise control of angular or linear position, velocity and acceleration. It consists of a suitable motor coupled to a sensor for position feedback. It also requires a relatively sophisticated controller, often a dedicated module designed specifically for use with servomotors.Servomotors are not a specific class of motor although the term servomotoris often used to refer to a motor suitable for use in a [closed-loop control](https://en.wikipedia.org/wiki/Closed-loop_control) system.

Servomotors are used in applications such as [robotics](https://en.wikipedia.org/wiki/Robotics), [CNC machinery](https://en.wikipedia.org/wiki/CNC_machine) or manufacturing. A servomotor is a [closed-loop](https://en.wikipedia.org/wiki/Closed-loop_controller) [servomechanism](https://en.wikipedia.org/wiki/Servomechanism) that uses position feedback to control its motion and final position. The input to its control is a signal (either analogue or digital) representing the position commanded for the output shaft.The motor is paired with some type of [encoder](https://en.wikipedia.org/wiki/Encoder) to provide position and speed feedback. In the simplest case, only the position is measured. The measured position of the output is compared to the command position, the external input to the controller. If the output position differs from that required, an [error signal](https://en.wikipedia.org/wiki/Error_signal) is generated which then causes the motor to rotate in either direction, as needed to bring the output shaft to the appropriate position. As the positions approach, the error signal reduces to zero and the motor stops.



**6.OLED:**

The OLED display module is one of the most attractive display available for a microcontroller. It has a good view angle and pixel density which makes it reliable for displaying small level graphics. Interfacing this IC with MCU can either be done using IIC or using SPI hence helps to save some pins as well. OLED is organic light emitting diode that emits light in response to an electric current. OLED display works with no backlight so it can display deep black levels. It is small in size and light in weight than Liquid Crystal Displays128x64 OLED display is simple dot matrix graphic display. It has 128 columns and 64 rows which make it display of total 128x64 = 8192 pixels. By just turning on/off these pixel’s led we can display graphical image of any shape on it.

OLED display is used for displaying text, images and various patterns. It is also suitable for mobile phone sub-display, MP3 player, calculators etc. OLED display has 256 steps for brightness control. OLED display also available with different resolution like 128x32, 128x64.

**Pin Configuration:**

* Ground (Gnd) - Connected to the ground of the circuit
* Vcc (Vdd,5V) - Can be powered by either 3.3V or 5V
* SCL (D0, CLK) - It is a clock signal. This pin transmits clocks to slave, SCL. Data will be sent to other devices on clock tick event. Only master device has control over this SCL line
* SDA (D1, MOSI) - SDA is used to transmit data between master and slave. The data and acknowledgement are sent through SDA.



The software components are:

* Arduino IDE
* MIT App Inventor 2

**1.Arduino IDE:**

The [Arduino](https://en.wikipedia.org/wiki/Arduino) integrated development environment ([IDE](https://en.wikipedia.org/wiki/Integrated_development_environment)) is a [cross-platform](https://en.wikipedia.org/wiki/Cross-platform) application (for [Windows](https://en.wikipedia.org/wiki/Windows), [macOS](https://en.wikipedia.org/wiki/MacOS), [Linux](https://en.wikipedia.org/wiki/Linux)) that is written in the programming language [Java](https://en.wikipedia.org/wiki/Java_(programming_language)). It is used to write and upload programs to Arduino compatible boards, but also, with the help of 3rd party cores, other vendor development boards.

The source code for the IDE is released under the [GNU General Public License](https://en.wikipedia.org/wiki/GNU_General_Public_License), version 2. The Arduino IDE supports the languages [C](https://en.wikipedia.org/wiki/C_(programming_language)) and [C++](https://en.wikipedia.org/wiki/C%2B%2B) using special rules of code structuring. The Arduino IDE supplies a [software library](https://en.wikipedia.org/wiki/Software_library) from the [Wiring](https://en.wikipedia.org/wiki/Wiring_(development_platform)) project, which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub *main()* into an executable [cyclic executive](https://en.wikipedia.org/wiki/Cyclic_executive) program with the [GNU toolchain](https://en.wikipedia.org/wiki/GNU_toolchain), also included with the IDE distribution. The Arduino IDE employs the program *avrdude* to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware.



**2.MIT App Inventor 2:**

App Inventor for Android is an [open-source](https://en.wikipedia.org/wiki/Open-source_software) web application originally provided by [Google](https://en.wikipedia.org/wiki/Google), and now maintained by the [Massachusetts Institute of Technology](https://en.wikipedia.org/wiki/Massachusetts_Institute_of_Technology) (MIT), which allows newcomersto [computer programming](https://en.wikipedia.org/wiki/Computer_programming) to create [software applications](https://en.wikipedia.org/wiki/Application_software) for the [Android](https://en.wikipedia.org/wiki/Android_(operating_system)) operating system (OS).It uses a graphical interface very similar to [Scratch](https://en.wikipedia.org/wiki/Scratch_(programming_language)) and the [StarLogo TNG](https://en.wikipedia.org/wiki/StarLogo_TNG) [user interface](https://en.wikipedia.org/wiki/User_interface), which allows users to [drag-and-drop](https://en.wikipedia.org/wiki/Drag-and-drop) visual objects to create an application that can run on Android devices. In creating App Inventor, Google drew upon significant prior research in educational computing, as well as work done within Google on online development environments.

App Inventor and the projects on which it is based are informed by [constructionist learning](https://en.wikipedia.org/wiki/Constructionist_learning) theories, which emphasizes that programming can be a vehicle for engaging powerful ideas through active learning. As such, it is part of an ongoing movement in computers and education that began with the work of [Seymour Papert](https://en.wikipedia.org/wiki/Seymour_Papert) and the MIT Logo Group in the 1960s and has also manifested itself with [Mitchel Resnick](https://en.wikipedia.org/wiki/Mitchel_Resnick)'s work on [Lego Mindstorms](https://en.wikipedia.org/wiki/Lego_Mindstorms) and [StarLogo](https://en.wikipedia.org/wiki/StarLogo).App Inventor also supports the use of [cloud data](https://en.wikipedia.org/wiki/Cloud_database) via an experimental [FirebaseDB](https://en.wikipedia.org/wiki/Firebase#Realtime_Database) component.



**CODE:**

#include <Wire.h>

#include <Adafruit\_SSD1306.h>

#include <Adafruit\_GFX.h>

#define SSD1306\_LCDHEIGHT 64

#define OLED\_ADDR 0x3C

Adafruit\_SSD1306 display(-1);

#if (SSD1306\_LCDHEIGHT != 64)

#error("Height incorrect, please fix Adafruit\_SSD1306.h!");

#endif

#include "DHT.h"

#include <Servo.h>

Servo myservo;

#define DHTPIN D0

#define DHTTYPE DHT11

DHT dht(DHTPIN, DHTTYPE);

int temp\_led=D7;

#define echopin D3

#define trigpin D4

#include <ESP8266WiFi.h>

#include <PubSubClient.h>

const char\* ssid = "JioFi2\_0C3198";

const char\* password = "s98xqs67ik";

String command;

#define ORG "n64yty"

#define DEVICE\_TYPE "new"

#define DEVICE\_ID "newdevice"

#define TOKEN "123456789"

char server[] = ORG ".messaging.internetofthings.ibmcloud.com";

char topic[] = "iot-2/evt/Data/fmt/json";

char authMethod[] = "use-token-auth";

char token[] = TOKEN;

char clientId[] = "d:" ORG ":" DEVICE\_TYPE ":" DEVICE\_ID;

WiFiClient wifiClient;

PubSubClient client(server, 1883,wifiClient);

void setup()

{

Serial.begin(9600);

dht.begin();

Serial.print("Connecting to ");

Serial.print(ssid);

WiFi.begin(ssid, password);

while (WiFi.status() != WL\_CONNECTED) {

delay(500);

Serial.print(".");

Serial.println("");

}

Serial.print("WiFi connected, IP address: ");

Serial.println(WiFi.localIP());

pinMode(temp\_led,OUTPUT);

pinMode(D3,INPUT);

pinMode(D4,OUTPUT);

myservo.attach(D5);

display.begin(SSD1306\_SWITCHCAPVCC, OLED\_ADDR);

display.clearDisplay();

display.setTextSize(1);

display.setTextColor(WHITE);

display.setCursor(10,10);

display.print("poultry farm");

Serial.println("poultry farm");

display.display();

}

void loop()

{

//gas sensor start

int analogSensor = analogRead(A0);

Serial.print("Pin A0: ");

Serial.println(analogSensor);

delay(500);

if (analogSensor >45)

{

display.begin(SSD1306\_SWITCHCAPVCC, OLED\_ADDR);

display.clearDisplay();

display.setTextSize(1);

display.setTextColor(WHITE);

display.setCursor(10,10);

display.print("Please switch on the Exhaust fans");

display.display();

Serial.println("Please switch on the Exhaust fans");

}

else

{

}

delay(100);

//gas sensor end

//dht11 start

float t = dht.readTemperature();

Serial.println(t);

delay(500);

if(t>35)

{

display.begin(SSD1306\_SWITCHCAPVCC, OLED\_ADDR);

display.clearDisplay();

display.setTextSize(1);

display.setTextColor(WHITE);

display.setCursor(10,10);

display.print("Temperature is HIGH");

display.display();

Serial.println("Temperature is HIGH");

digitalWrite(temp\_led, HIGH); //Red

delay(200);

}

else if(t<25)

{

display.begin(SSD1306\_SWITCHCAPVCC, OLED\_ADDR);

display.clearDisplay();

display.setTextSize(1);

display.setTextColor(WHITE);

display.setCursor(10,10);

display.print("Temperature is LOW");

display.display();

Serial.println("Temperature is LOW");

}

else

{

digitalWrite(temp\_led, LOW);

}

//dht11 end

//ultrasoinc start

digitalWrite(D4,HIGH);

delay(1000);

digitalWrite(D4,LOW);

int duration = pulseIn(echopin,HIGH);

int distance = (duration/2)\*0.0343;

Serial.println(distance);

delay(500);

if(distance<5)

{

display.begin(SSD1306\_SWITCHCAPVCC, OLED\_ADDR);

display.clearDisplay();

display.setTextSize(1);

display.setTextColor(WHITE);

display.setCursor(10,10);

display.print("Please fill up the water tanks");

display.display();

Serial.println("Please fill up the water tanks");

delay(500);

}

else if(distance>10)

{

display.begin(SSD1306\_SWITCHCAPVCC, OLED\_ADDR);

display.clearDisplay();

display.setTextSize(1);

display.setTextColor(WHITE);

display.setCursor(10,10);

display.print("Water tank is full");

display.display();

Serial.println("Water tank is full");

delay(500);

}

else

{

}

//ultrasoinc end

//servo start

myservo.write(90);

delay(2000);

myservo.write(0);

delay(100);

//servo end

PublishData(t, distance, analogSensor);

}

void PublishData(float t, float distance, float analogSensor)

{

if (!!!client.connected()) {

Serial.print("Reconnecting client to ");

Serial.println(server);

while (!!!client.connect(clientId, authMethod, token)) {

Serial.print(".");

delay(500);

}

Serial.println();

}

String payload = "{\"d\":{\"t\":";

payload += t;

payload+="," "\"distance\":";

payload += distance;

payload+="," "\"analogSensor\":";

payload += analogSensor;

payload += "}}";

Serial.print("Sending payload: ");

Serial.println(payload);

if (client.publish(topic, (char\*) payload.c\_str()))

{

Serial.println("Publish ok");

}

else

{

Serial.println("Publish failed");

}

}

**ADVANTAGES:**

1. This system replaces traditional farm into an intelligent farm.

2. It Provide quicker and accurate information about different parameters to farmers.

3. System Required Low cost, asset saving and productive management in chicken farm.

4. The smart monitoring of different parameter like temperature, light, humidity, gas etc by using wireless sensor network.

**CONCLUSION:**

The traditional way of chicken farming is replacing with the smart and intelligent chicken farming using embedded system based innovative application. It helps the farmers real time controlled and monitoring environmental aware context parameters such as temperature, humidity, air quality, light intensity and controlling filter fan, ventilation window. This smart system can effectively control the farm from any location and reduces cost time and man power. This will improve productivity and quality of chickens in poultry farming. In the future advanced IOT based technologies should be use for monitoring and controlling health related parameters of chicken to improve quality and productivity of chicken farming, which will Result into profits for farmers and quality food for human being.

**REFERENCES:**

[www.ijarcs.info/index.php/Ijarcs/article/download/2665/2653](http://www.ijarcs.info/index.php/Ijarcs/article/download/2665/2653)

<http://www.iotgyan.com/iot-projects/iot-enabled-smart-poultry-farm>

<https://www.robot-r-us.com/vmchk/sensor-temp/humid/dht11-temperature-and-humidity-sensor.html>

<https://www.nodemcu.com/index_en.html>

<https://components101.com/ultrasonic-sensor-working-pinout-datasheet>

<https://components101.com/mq2-gas-sensor>

<https://en.wikipedia.org/wiki/Servomotor>

<https://en.wikipedia.org/wiki/OLED>