Project Title: An Automated Poultry Farm using IoT

Team Members

1. Samson Otieno Ooko 220014196
2. Oluwatobi Oyinlola 219012916
3. Marvin Muyonga Ogore 220014154
4. Kannole Veronic Eugene 220012874
5. Uwamariya Marriette 219014505

Project Supervisor

1. Prof Kayal

Project Aim/Objectives

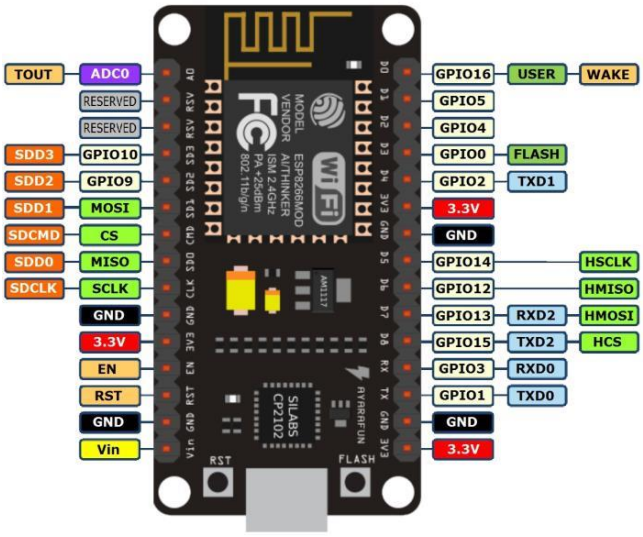
The aim of this project was to automate operations in a poultry farm using the latest IoT technologies. The specific objectives were:

1. To monitor the poultry house temperature and send alerts with time stamps when the temperature goes beyond or below allowed limits.
2. To automatically feed the chicken at required intervals of time.
3. To send automatic periodic reminders for recommended vaccinations.

Requirements

The following are needed in order to undertake the project:

1. ***Hardware***
    a. *NodeMCU Microcontroller with a USB cable*

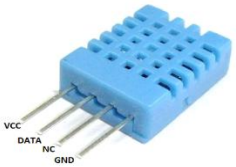


NodeMCU is an open source platform based on ESP8266 which can connect objects and let data transfer using the Wi-Fi protocol. In addition, by providing some of the most important features of microcontrollers such as GPIO, PWM, ADC, and etc, it can solve many of the project's needs alone.

b. *Real time clock (DS3231)*

The RTC module runs on a battery and can keep track of the time even if we reprogram the microcontroller or disconnect the main power. The DS3231 is a low-cost, highly accurate Real Time Clock which can maintain hours, minutes and seconds, as well as, day, month and year information.

c. *Temperature sensor (DHT11)*



The DHT11 is a basic, ultra-low-cost digital temperature and humidity sensor. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air, and spits out a digital signal on the data pin.

d. *Servo Motor*

A servo motor is an electrical device which can push or rotate an object with great precision. If you want to rotate and object at some specific angles or distance, then you use servo motor. It is just made up of simple motor which run through servo mechanism

e. *LED*

A light-emitting diode (LED) is a semiconductor device that emits light when an electric current is passed through it

f. *Jumper Wires*

Jumper wires are simply wires that have connector pins at each end, allowing them to be used to connect two points to each other without soldering.

g. *Breadboard*

A breadboard is a rectangular plastic board with a bunch of tiny holes in it. These holes let you easily insert electronic components to prototype

h. *Resistor 220 Ohms*

2. ***Software***
    a. *Arduino IDE*
    The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.
    b. *Firebase*
    Firebase a google based mobile and web application development platform. Firebase gives functionalities like analytics, databases, messaging and crash reporting so you can move quickly and focus on your users.

Hardware Connection

The above hardware components will be connected as follows:

a. *RTC Connection*
Make a circuit as per the given below diagram. In Real-Time Clock there are eight pins available. We will be using only four pins. Connect the pins as below given.
- VCC to 3.3V
- GND to Ground
- SDA to D2 (GPIO 4)
- SCL to D1 (GPIO 5)

b. *Connection to Servo*
The next job is to connect your servo motor. If your servo has White - Red - Black wires, then connect it as follows
- White wire connects to Digital pin D5
- Black wire connects to GND pin
- Red wire connects to 3V3 pin

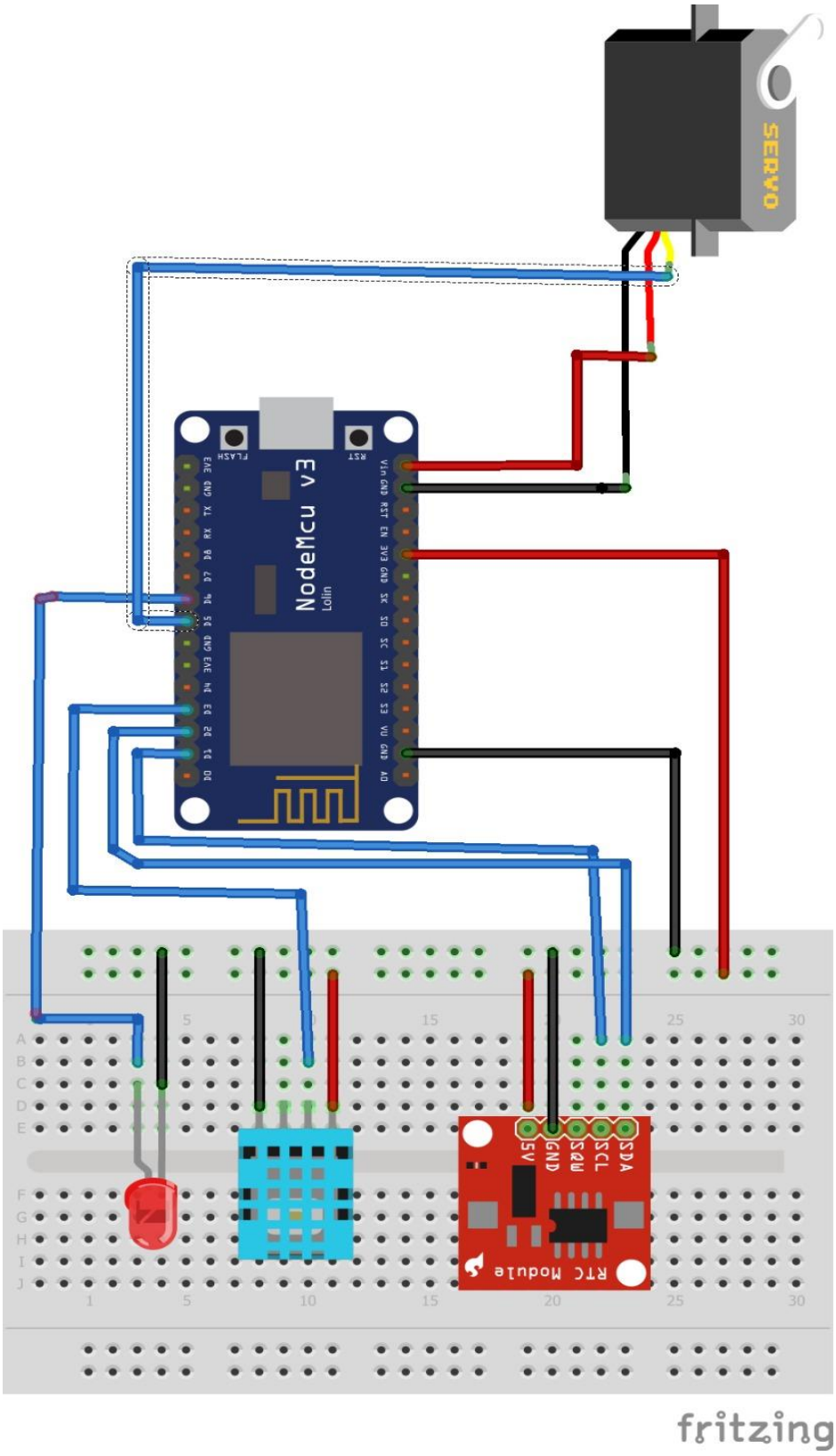
c. *Connections to DHT11*
The wiring connections for the temperature sensor are made as follows:
- Pin 1 of the DHT11 goes into +3v of the NodeMCU.
- Pin 2 of the DHT11 goes into Digital Pin D3 of the NodeMCU.
- Pin 3 of the DHT11 goes into Ground Pin (GND) of the NodeMCU

d. *Connection to LED*
A node of the LED is connected to the D6 pin of the NodeMCU, the cathode of the LED is connected with the one terminal of the resistor and another terminal of the resistor is connected to the ground pin.

e. *Connection to bread board if preferred*
In case a bread board is used, the 3.3V and ground pins of the NodeMCU will be connected to the board and all the other components will also connect to the 3.3v and ground lines of the board.

After all the components have been correctly connects the USB can then be connected to the computer for uploading of code

Circuit Diagram



Software Set Up

The following steps will be followed in setting up the software

1. *Arduino IDE installation*

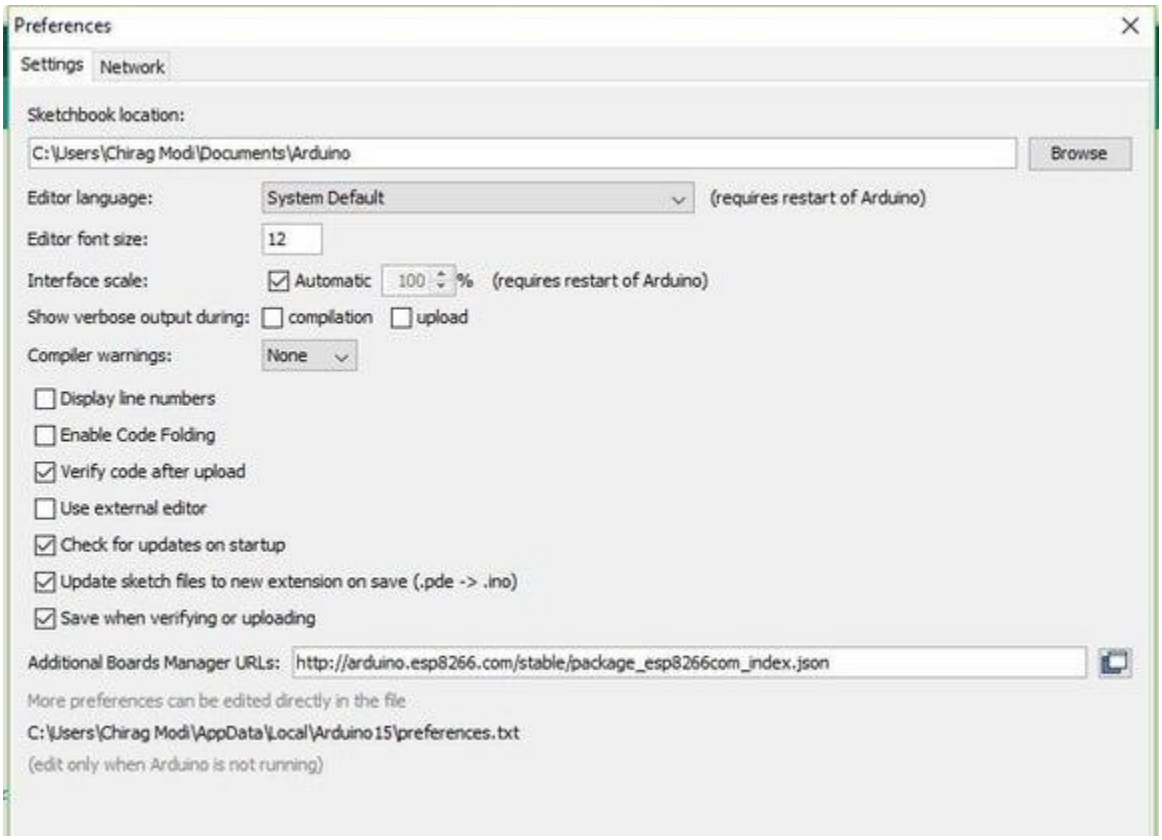
   To download the software go to the Arduino site: https://www.arduino.cc

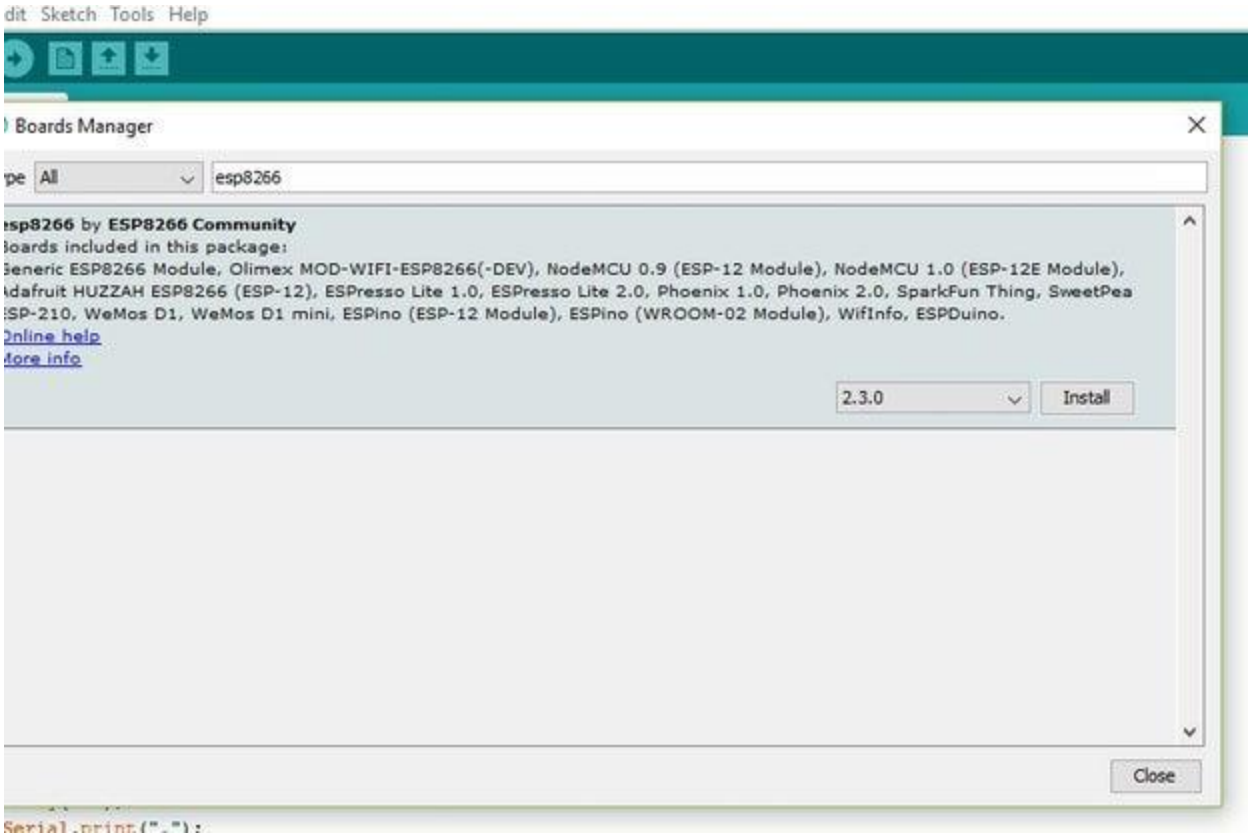
   a. Click on software click on either Windows, Mac or Linux based on your Operating System.
   b. You can donate if you want or just download.
   c. When this is done, you will simply need to continue the steps to download it to your computer.

2. *Arduino IDE setup*

   After downloading the Arduino IDE navigate to:

   a. File tab and then click on Preferences.
   b. In the additional Boards Manager URLs add the following link (http://arduino.esp8266.com/stable/package_esp8266com_index.json)
   c. Click OK and then navigate to Tools - Boards - Boards Manager
   d. In the search field type esp8266 > click the esp8266 by ESP8266 Community
   e. Click Install

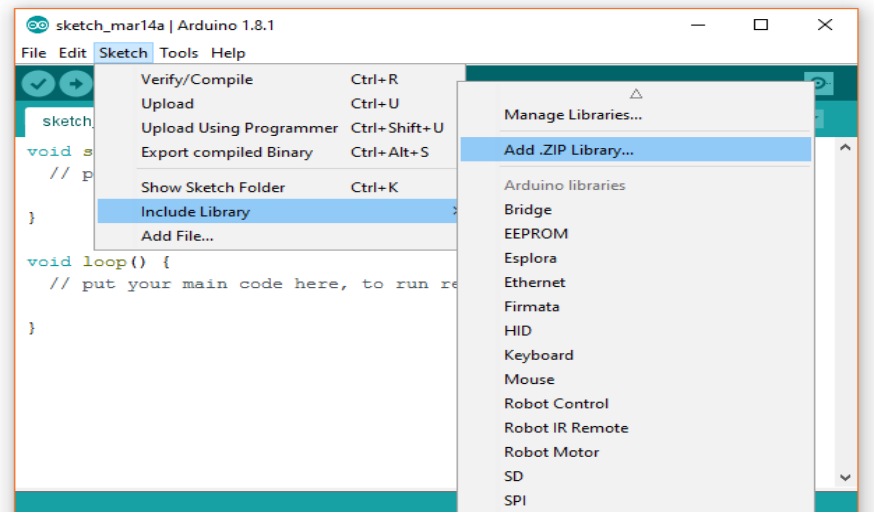




3. *Installation of Packages*

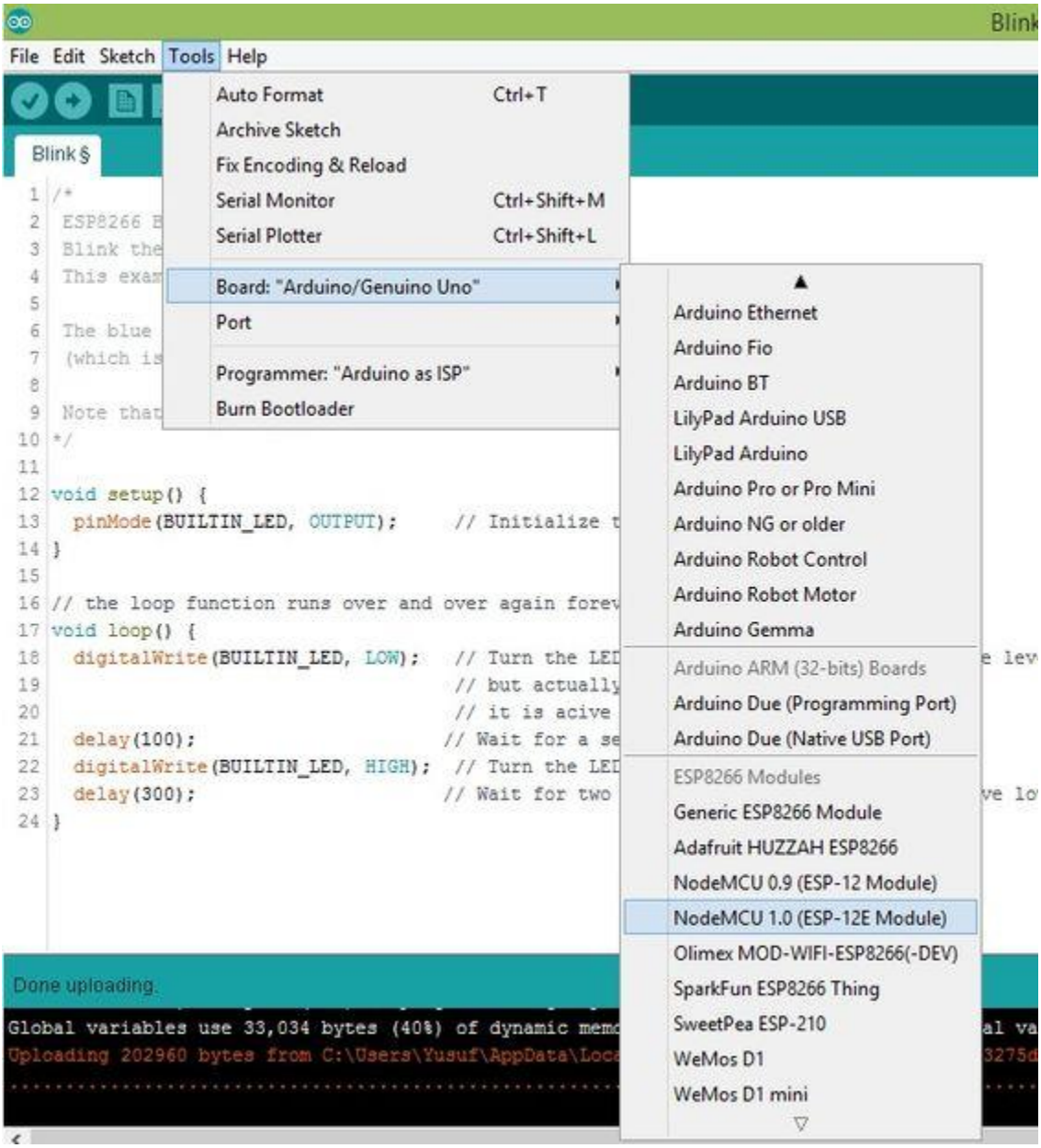
You will need to install additional libraries for the code to work

a. Go to Sketch > Include Library > Manage Libraries.
Search for DHT.h, ESP8266WiFi.h, FirebaseArduino.h, RTClib.h and Servo.h
b. You can also search for the zip file of the libraries online then install using add a .zip Library link in include library



4. Board and Port Selection

After all libraries have been installed you will need to select the ESP8266 board and COM port



5. Uploading and compilation of code

After all has been set up the next step is to compile and upload the code to the board

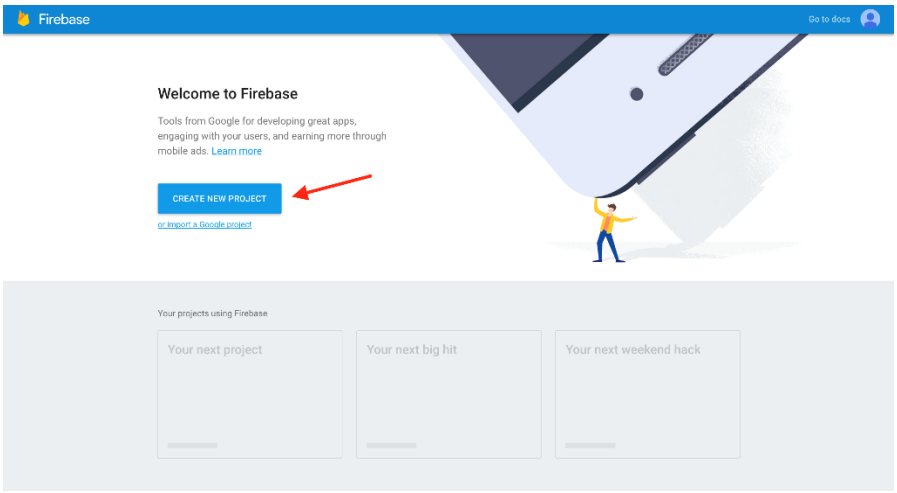
Firebase Set Up

a. Create an account/login

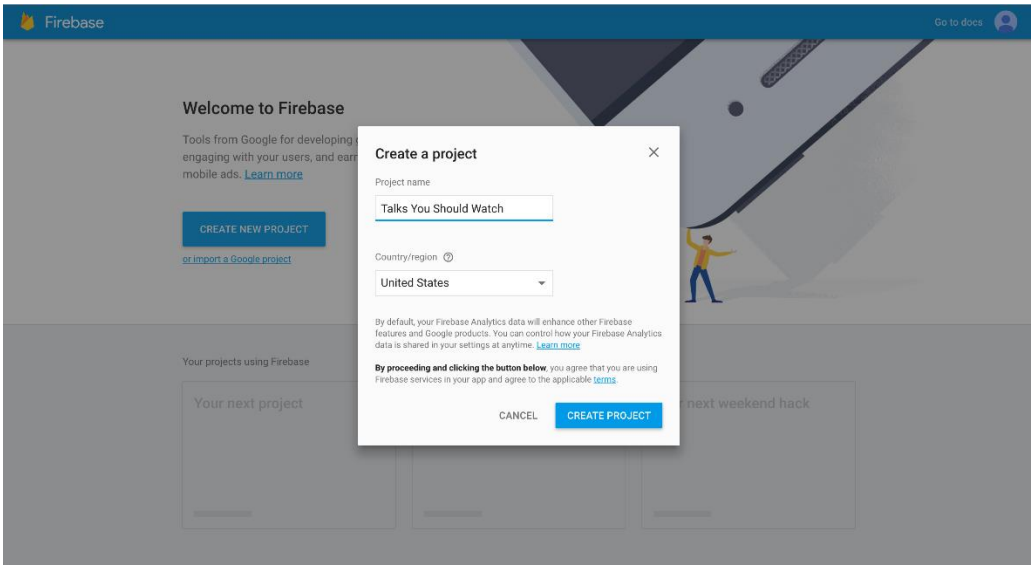
Go to the Firebase website and sign up for an account if you don't already have one. You can log in with a Google account for easy access.

b. Create a new project

When you log in, you should be directed to the Firebase console. You can manage all of your projects here. Go ahead and create a new one by clicking the blue **CREATE NEW PROJECT** button.



Give your new project a name.

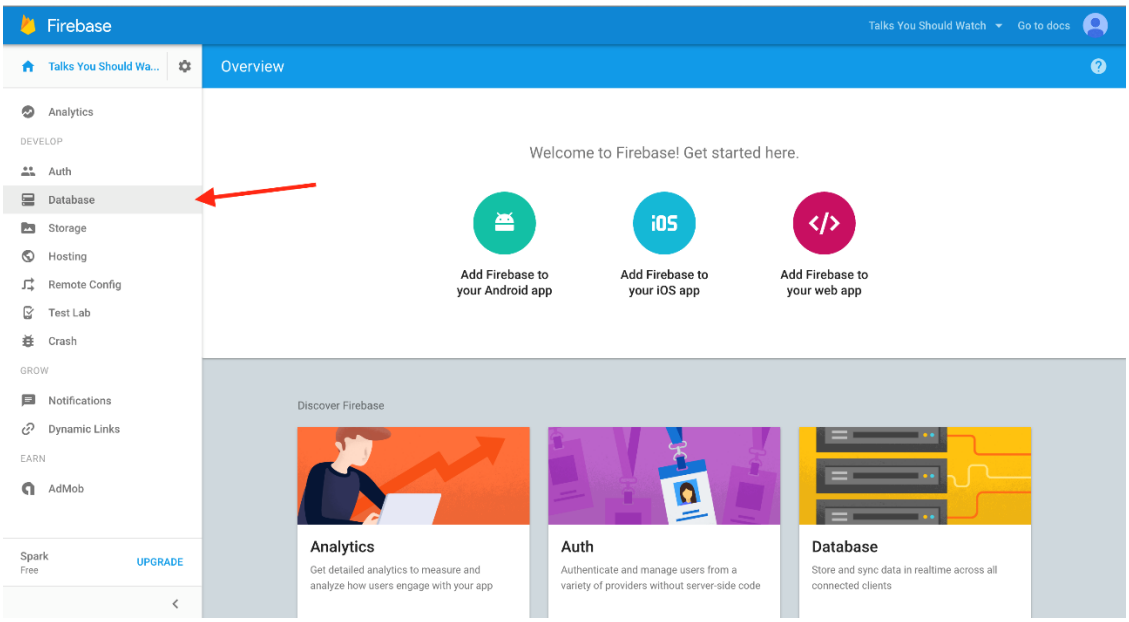


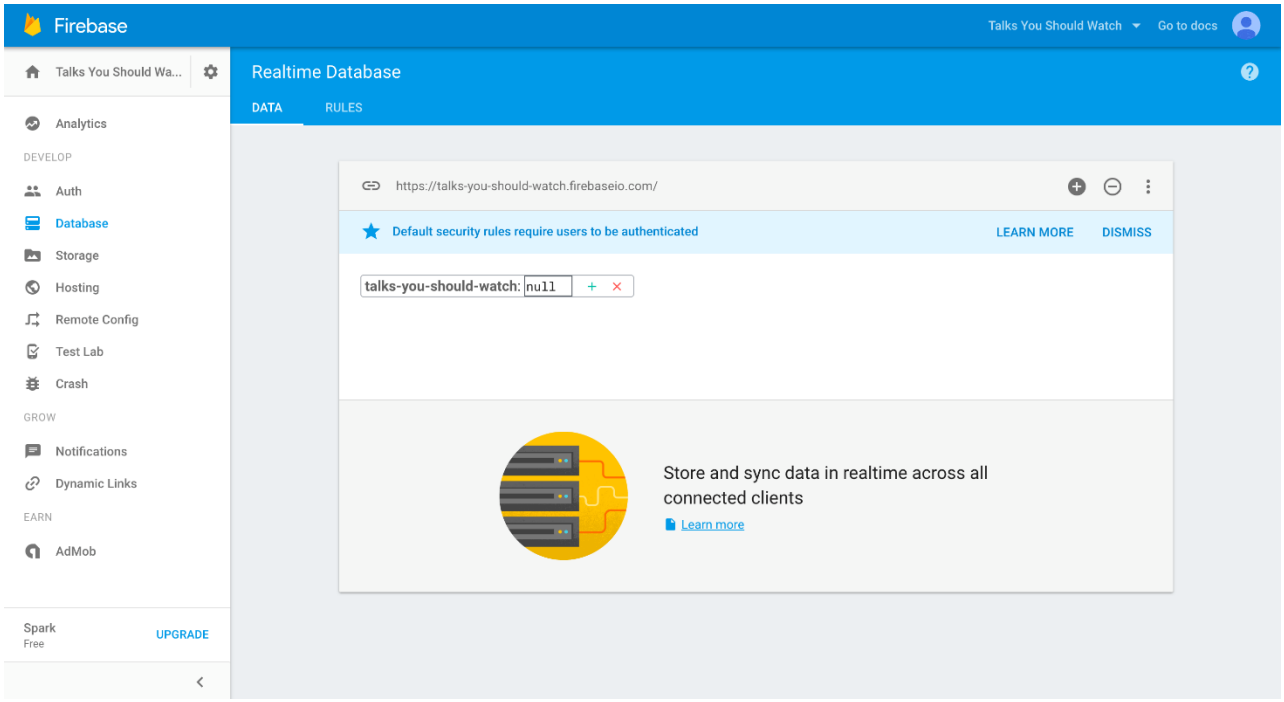
Once you create your project, you'll be redirected to your project's console overview. If you check the URL at the top of your browser, you'll see something like **https**

https://console.firebase.google.com/u/1/project/team2-temp/database/team2-temp/data, where the part of the URL after **/project/** matches your project's name.

c. Database Link

Your database is available in the database link on the project window



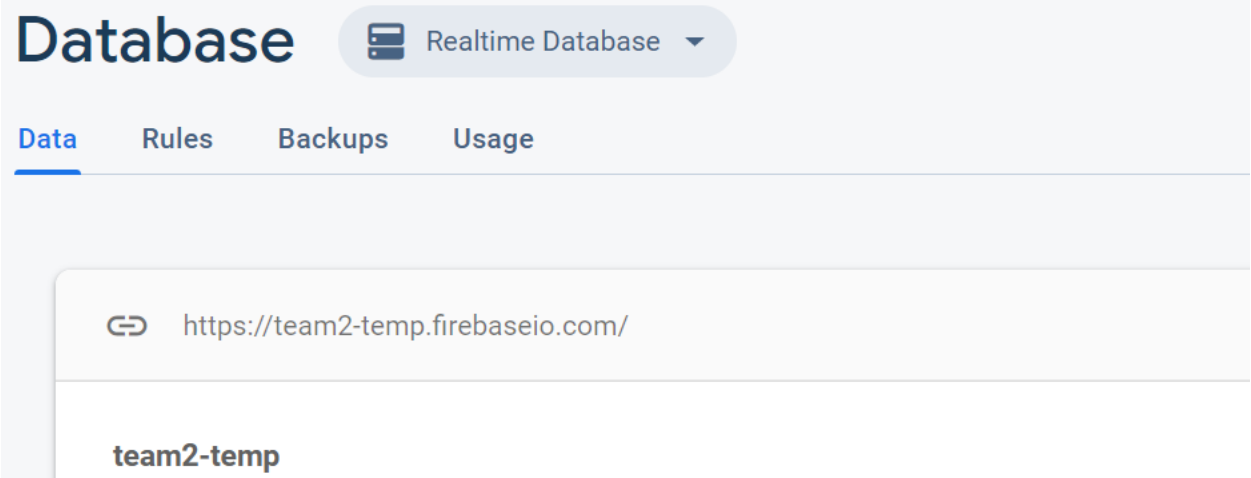


d. Database link

To run the project you will need to get the database link and the secret key

For database link on database and copy the link without https and replace in the following line of code

#define FIREBASE_HOST "team2-temp.firebaseio.com"



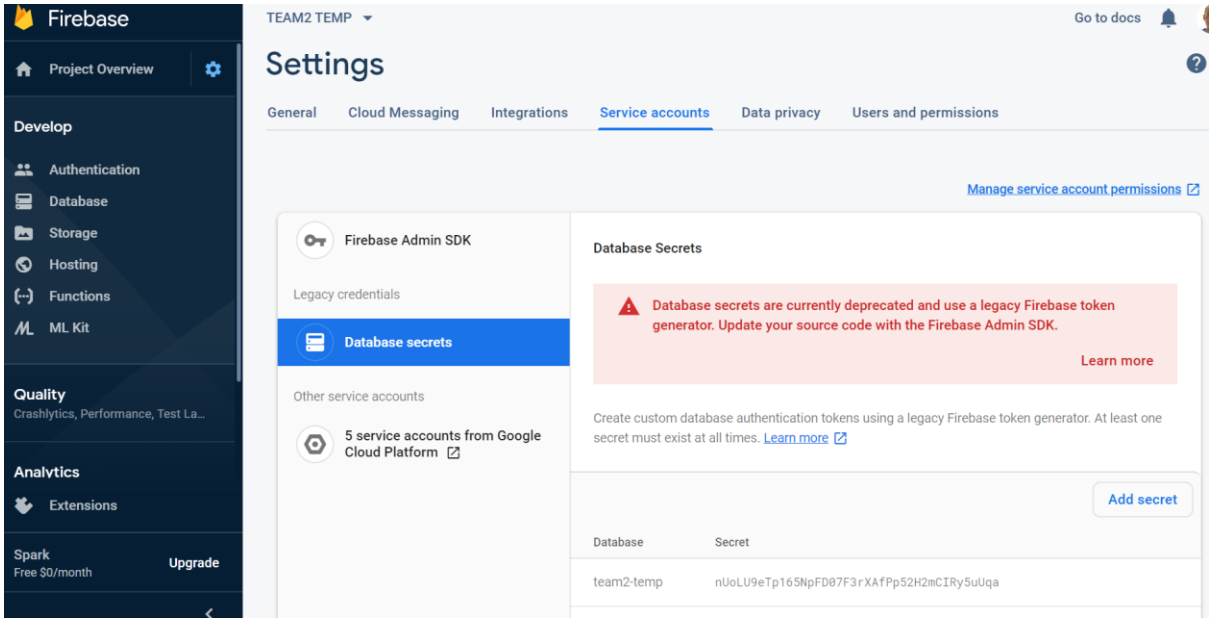
e. Database Secret

For the database secret click on Project overview/user and permissions

Then click on service accounts

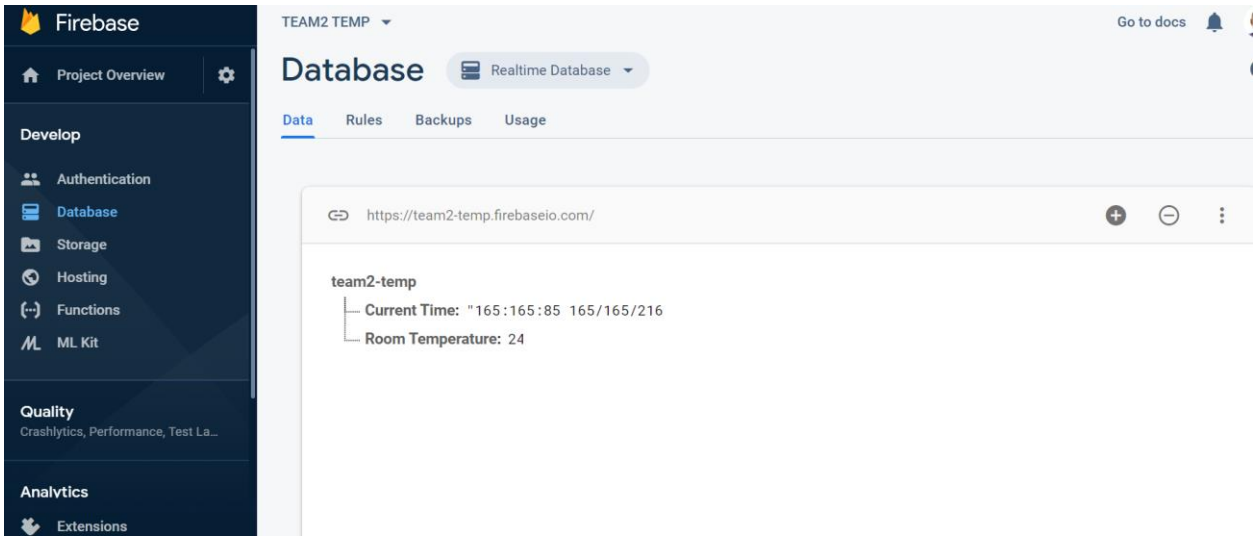
Click database secrete and copy to the following line of code

#define FIREBASE_AUTH "nUoLU9eTp165NpFD07F3rXAfPp52H2mCIRy5uUqa"



f. Database View

To view the database click on database and select real-time database

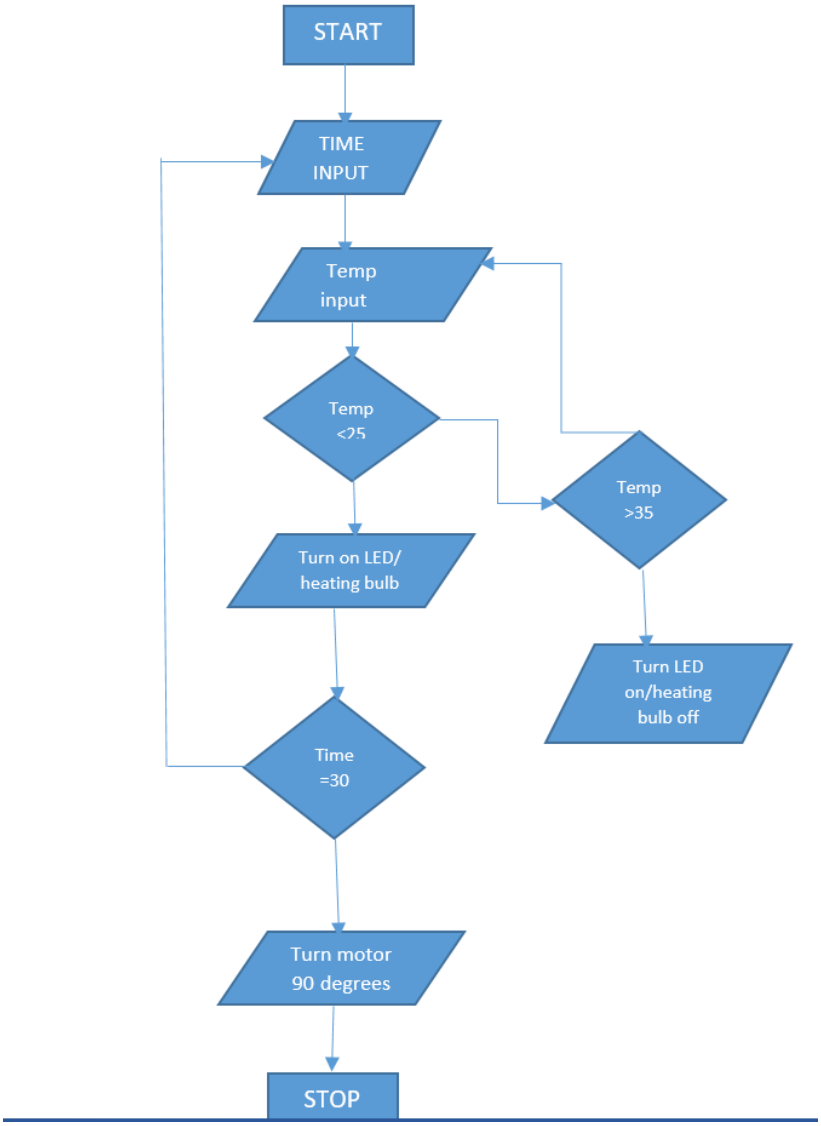


Working Principle

The system will work as follows:

a. The temperature sensor will be able to capture the temperature of the poultry house after every two seconds and also send to firebase. In case the temperature goes below defined levels an alert is printed on the screen and the LED is lit. This can further be improved so that the heating bulb in a chicken house is automatically switched on or off.
b. The real time clock is used to track time so that the servo motor turns by 90 degrees after every 30 minutes so that the chicken are fed. The clock is also used to set automatic alerts to the screen when vaccination times are reached.

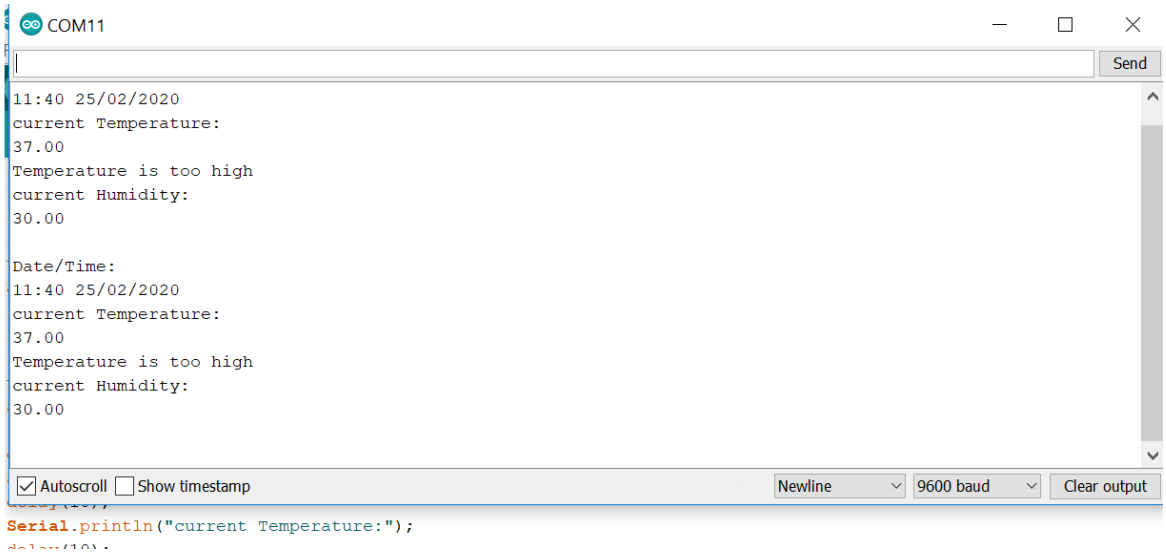
This can be summarized as per the flow chart below:



Implementation

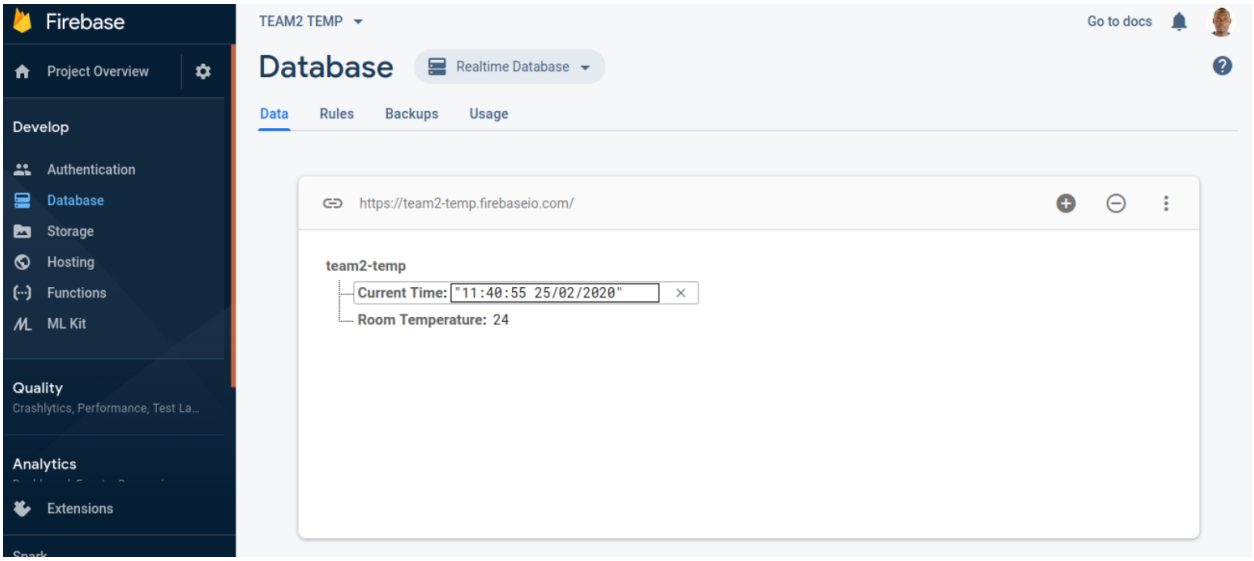
The system was implemented as per the instructions given above and the following are images from setup and output:

a. *Serial output screens*
Shows both outputs related to temperature and servo operation



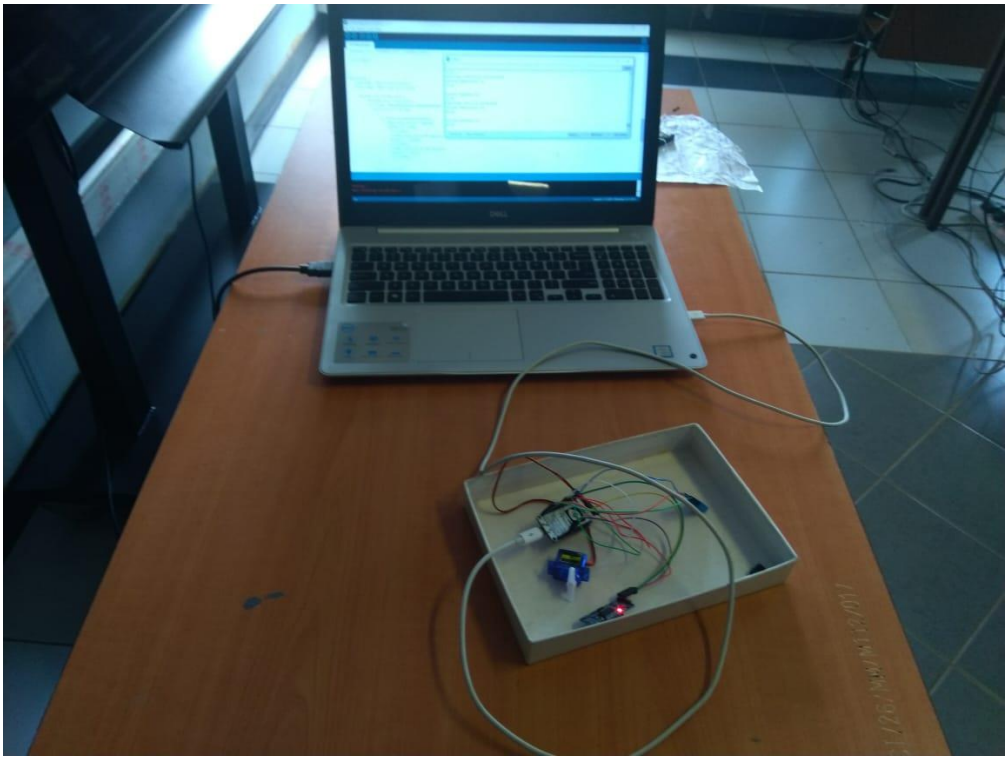
b. *Firebase screen shot*

The screen shot shows time and temperature reading as sent to firebase



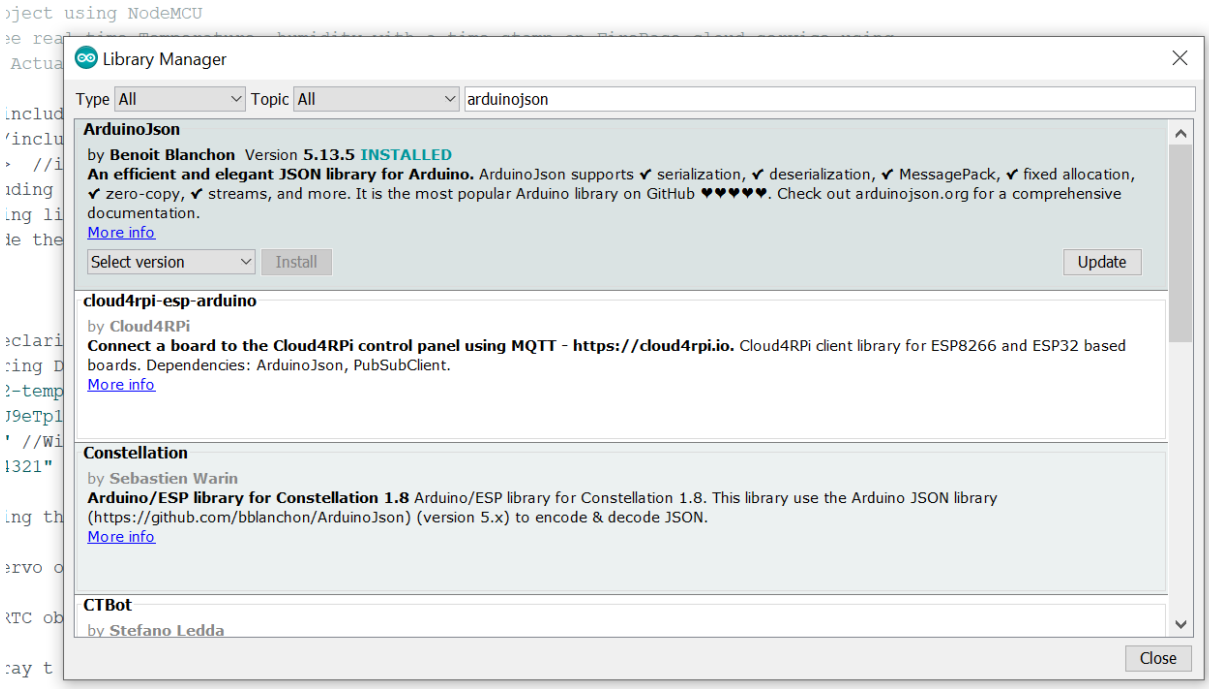
c. *Image of the Circuit*

This shows the image of the connected circuit



Implementation Challenges

The only challenge experienced during the implementation is that firebase is not compatible with the current version (Version 6) of ArduinoJson so you have to install a version 5.X of ArduinoJson under sketch library manager:



Possible Application Areas

This project can be applied in the following areas

- In a poultry farm to monitor temperature and feed the chicken
- In other animal farms to automatically feed animals
- For environmental monitoring
- Temperature sensitive applications
- Time sensitive applications

References

https://create.arduino.cc/projecthub/electropeak/getting-started-w-nodemcu-esp8266-on-arduino-ide-28184f

https://circuitdigest.com/article/servo-motor-basics

https://www.adafruit.com/product/386

https://www.ledsmagazine.com/leds-ssl-design/materials/article/16701292/what-is-an-led

http://blog.sparkfuneducation.com/what-is-jumper-wire

https://www.arduino.cc/en/main/software

https://firebase.google.com/

https://iot-guider.com/esp8266-nodemcu/learn-interfacing-ds3231-rtc-module-with-nodemcu/

https://www.instructables.com/id/Interfacing-Servo-Motor-With-NodeMCU/

https://www.instructables.com/id/Interface-DHT11-Humidity-Sensor-Using-NodeMCU/

https://nodemcu.readthedocs.io/en/master/modules/rtctime/