HERB CLOUD

Project report submitted in partial
fulfillment of the requirement for the course of

PROJECT DESIGN PRACTICUM
(1154ME201)

By

GUNDA SAI KRISHNA (17UECN0017)
KUNAL PRASAD (17UECN0027)
MOHAMMAD UMAR (17UECN0033)

Under the guidance of

Mrs. K. Prema
Assistant Professor
Department of Computer Science and Engineering

DEPARTMENT OF MECHANICAL ENGINEERING

SCHOOL OF MECHANICAL AND CONSTRUCTION

VEL TECH RANGARAJAN Dr. SAGUNTHALA R & D INSTITUTE OF SCIENCE AND TECHNOLOGY
(Deemed to be University Estd u/s 3 of UGC Act, 1956)

CHENNAI 600 062, TAMILNADU, INDIA

November, 2019

BONAFIDE CERTIFICATE

This is to certify that the project entitled "**HERB CLOUD**" submitted by **GUNDA SAI KRISHNA** (17UECN0017), **KUNAL PRASAD** (17UECN0027) and **MOHAMMAD UMAR** (17UECN0033) in partial fulfillment for the requirements for the course of **PROJECT DESIGN PRACTICUM** (1154ME201) is an authentic work carried out by them under my supervision and guidance.

To the best of my knowledge, the matter embodied in the project report has not been submitted to any other University/Institute for the award of any Degree, Course or Diploma.

Signature of Supervisor

Mrs.K.Prema M. Tech,
Assistant Professor,
Dept. of CSE,
Vel Tech Rangarajan Dr. Sagunthala
R&D Institute of Science and Technology
Avadi, Chennai – 600 062

Signature of Head & Dean

Dr. A.T. Ravichandran M.E, PhD,
Professor,
Dept. of Mechanical Engineering,
Vel Tech Rangarajan Dr. Sagunthala
R&D Institute of Science and Technology
Avadi, Chennai – 600 062

DECLARATION

We declare that this written submission represents our ideas in my own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Gunda Sai Krishna
(17UECN0017)
Dated: 24th November 2019

Kunal Prasad
(17UECN0027)
Dated: 24th November 2019

Mohammad Umar
(17UECN0033)
Dated: 24th November 2019

Vel Tech
Rangarajan Dr. Sagunthala
R&D Institute of Science and Technology
(Deemed to be University Estd. u/s 3 of UGC Act, 1956)

CERTIFICATE OF EVALUATION

Institute : Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology

Degree : Bachelor of Technology

Programme : Computer Science Engineering

Semester : ***ODD/2019***

Name of the Students	**1. GUNDA SAI KRISHNA** **2. KUNAL PRASAD** **3. MOHAMMAD UMAR**
Title of the Project	**HERB CLOUD**
Name of the Course handling Faculty	**Dr. S. Balaguru, M.Tech, Ph. D,** Professor, Department of Mechanical Engineering, Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology.

This report of the project work submitted by the above students in partial fulfilment for course was evaluated and confirmed to be the report of the work done by above student. Submitted for the Viva-Voce held on ……………………… at Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology.

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

This project, though done by us, would not have been possible without the support of various people by their co-operation have helped in bringing out this project successfully.

We would like to thank our **Founder President Col. Prof. Dr. Vel. R. RANGARAJAN., BE (Elect.), M.S.(Auto.)., D.Sc.** and **Vice Chairman and Chancellor Dr. SAGUNTHALA RANGARAJAN., MBBS.,** for providing us ambient learning experience at our institution.

We are greatly thankful to our **Chairperson Managing Trustee Dr. RANGARAJAN MAHALAKSHMI KISHORE., B.E. (I.E), MBA. (UK), Ph.D., and Vice President Mr. KISHORE KUMAR., B.E., MBA,** for their encouragement and valuable academic support in all aspects.

A special thanks to our **Vice Chancellor Prof.V.S.S. KUMAR., B.E. (CIVIL), M.Tech.(BSCM), Ph.D. (CM), POST.DOCT(CEM)** for providing all facilities to complete our project.

We thank our **Registrar**, **Mrs. N.S. PREMA** for encouraging us to complete our project and served as inspiration for us to perform well.

We are thankful to our esteemed **Director Academics Dr. ANNE KOTESWARA RAO, Ph.D.,** for providing a wonderful environment to complete our project successfully.

We record indebtedness to **Head of the Department / Dean Prof. Dr. A.T. RAVICHANDRAN., M.E., Ph.D.,** for immense care and encouragement towards us throughout the course of this project.

We thank our **Course Handling Faculty Prof. Dr. S. BALAGURU., M.Tech., Ph.D.,** for the valuable information shared on proceeding with our project.

We also take this opportunity to express a deep sense of gratitude to Our **Internal Supervisor Mrs. K. PREMA., M.Tech.,** for her cordial support and guidance, she helped us in completing this project through various stages.

We thank our department faculty, supporting staffs, parents, and friends for their help and guidance to complete this project.

GUNDA SAI KRISHNA (VTU9697) (17UECN0017)
KUNAL PRASAD (VTU10313) (17UECN0027)
MOHAMMAD UMAR (VTU9692) (17UECN0033)

TABLE OF CONTENTS

ABSTRACT

Modern day lifestyle has many benefits over the older ones. With increased health-care, better living conditions and advances in technology, it has nevertheless made out lives much easier and comfortable. On the other side of the representation of this modern-day lifestyle, is something we all have felt and occasionally find escape to. The busy work-life, schedules and living in a literal concrete jungle takes tolls from the best of us. We long for recreation and connection to the nature.

The project heals in this regard, as it offers chance to take care, nurture and grow a plant in your own home. We have brought together – integrated – technologies from various reals of computer science, electronics engineering and networking to bring an autonomous system, capable of learning on itself and responding to every request from the user – to help nurture, and grow the plant.

We have used a combination of sensing equipment, cloud-to-system networking, graphic user interfaces and other means of interaction with the system – which in turn means interaction with the plant – to take care of daily plant needs and also optimize per-user per-plant per-location basis to form a sort of bond between the plant and user that goes into realms of personalized sustenance.

With minimum efforts from the user, he/she can slowly but steady take advantage of the system to grow their wish of an indoor plant, see them grow, bond with it as the days pass and smile on the evening sun with a healthy plant on the vicinity.

CHAPTER 1

INTRODUCTION

1.1 Aim of the Project

The aim of the project is the design and fabrication of a feasible modular smart indoor plant system with applications of modern technologies such as the Internet of Thing and If This…Then That protocol. During the duration of the design and fabrication stages, the team members will also have an in-depth knowledge of the technologies being used, and see the practical application and implementation of the subjects and courses studied during their term as undergraduate engineering students.

1.2 Project Domain

Multiple programming languages of both embedded systems and UI applications have been used in the project. Inter-communication protocols such as TCP/IP and over secure channels have also been used.

The network architecture and connectivity between various sensing, calculation and transfer systems have also been implemented with sophisticated knowledge of IoT device routing, IFTTT protocols and GUI programming.

1.3 Problem Statement

Modern day lifestyle is often very tedious and time-constrained for most of us. This lifestyle cannot promise as much recreational times that we used to have few decades ago. In addition, most job-seeking youth come to the cities, where the greenery isn't as good as in the sub-urban areas. The main idea behind this project is help these people with a small - household passion of growing plants.

Research shows that indoor plants can improve mood, atmosphere and overall health of the home basis. With integration to safety protocols, this system provides and solves many such issues with ease.

CHAPTER 2

LITERATURE SURVEY

This chapter gives an overall summary and review of various literary surveys done in the topic concentration on the proposed system in this project. There have been numerous studies on the topics of Automation, Agriculture using application of robots, the Internet of Things and Overall health benefits to human whilst near plants.

The authors in the 2010 paper on “Wireless home automation networks: A survey of architectures and technologies” have studies in detail on how by using various sensors and actuators we can enable monitoring and control application for user comfort and efficient home management schemes.[1] for further details.

A group of four researchers from the field of computer science and engineering, on 2015, did a detailed study on the purpose of autonomous agricultural mobile robots capable of processing and monitoring field operations such as precision framing, fertilization, disease diagnosis, enabling a proof of concept that such a system would be feasible and also help boost production, reducing long-time cost over time as it matures.[2] for further details on the subject.

A much deeply researched and versatile field of study encompassing the realms of Computer Science, Networking, Electronic Engineering and System Architecture is Internet of Things. Various papers were sighted before proceeding with the project, but one most vital is “Internet of Things (IoT): A vision, architectural element, and future directions” by a group of researchers in the autumn of September 2013. It discusses visions and motivations of IoT, applications, cloud centric orientation and future towards a more immensely webbed internet [3] to go onto that subject in detail.

For the project proposed, it was necessary that the team also researched on the effects of indoor plants on human health. A research paper on the Indoor and Built Environment headed “The basic roles of indoor plants in human health and comfort” on the December of 2018. The beautifully written paper discusses on how humans have a close relationship with nature, and integrating this nature world into the indoor space induce peoples’ enhancement with nature, benefiting health and comfort.[4] in the bibliography section to know more.

CHAPTER 3

PROJECT DESCRIPTION

3.1 Existing Systems

3.1.1 Description

A very prominent and very well documented dive into the concept is a Kick-starter project titled “Raspberry Pi Smart Garden System (SGS)”. It is a Raspberry Pi-based smart watering system for DIY explorers and individuals with minimalist knowledge too. Although it is backed by many people and has also raised over $9000, the system is very minimalist in design and even offer 3D Printed Stands to go with the DIY toolkit. [5].

The company “Mobiloitte” offers smart garden solutions that help the users with a hassle-free solution to maintain a garden with minimalist approach. They also help design a custom project just to sooth the customers’ demands and use cases.[6].

A paper cited on the IJITEE Journal labeled “Smart Gardening System” on the March of 2019 discusses the same without an active application of IoT into the diode. [7].

It should be noted that all the existing system go ahead to the similar fixture with methodologies very industry standard.

3.1.2 Disadvantages

The following are the leading disadvantages cited in the preexisting concepts and systems:

1. The overall end-user cost was very high in regards to the technologies used.
2. Assembly was difficult and not at-all user friendly in nature.
3. The system architecture was neat only on paper, the final product looked messy, at ease to falter.
4. Advances in technologies such as IoT and ML were not inclusive in any of the products.
5. Plant-type based optimization was missing.

3.2 Proposed Systems

3.2.1 Description

An indoor plant favorable to home-like environment is placed in the pot. This plant demands care and nurture over time so that it can grow into a beautiful creation making the environment indoor much more pleasing and healthier.

There are various sensors connected to the plant system and environment surrounding the plant itself. These sensors collect data-feed from the soil, air, water, and send it over to the cloud for further processing by the means of two different yet fully connected micro-controllers.

The cloud computes and generates results in accord to the data-feed by the previously mentioned sensors and relay it to the users' mobile device. The user(s) can use this data to fully automate the system or take over controls of the plant whenever she pleases using a User Interface very simple in conduction.

The user(s) may also use already widely available API and Smart Assistant Systems (SAS) for interacting with the plant device system – Herb Cloud – like Google Assistant, Alexa, Siri.

This is a summarized description of the proposed system, and a detailed interception is followed in the next chapter.

3.2.2 Advantages

The advantages using the proposed system are as follows:

1. User-Friendly approach to customer
2. A reliable Plant-Human interface creation
3. Application of new and emerging technologies into the system – not ruining it obsolete
4. The price of the end product is kept low with a marginal ₹2500 while research and development phases – refining even lower cost in production.
5. Enables future plant profiling and advances.

3.3 Feasibility Study

A feasibility study was conducted on the proposed system onto various factors such as the economical, technological, social realms to conjugate a final charge into the project.

3.3.1 Economic Feasibility

A study was carried out to check the economic impact of the system proposed would have on the market, its competitors and how much fund the team can pour into the research and development since these resources were limited. The expenditures were justified and development of the system within a prescribed budget was achieved successfully.

3.3.2 Technological Feasibility

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. All the technologies used were freely available and could be easily integrated into the system in favor of the project, without any infringement acts.

3.3.3 Social Feasibility

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity.

Demography this product is targeted towards range the early-working and other mature classes with cannot find time to maintain a plant, but never the less desire to.

Geographic factors do not play much heed into this analysis as the project is feed to indoor conditions – which tends to remain overall constant even over a varied geography.

Physiological is to make the product so intrinsic and well-off on-and-itself that the users maintain a sort of relationship with the plant like a living pet inside the house, maturing the industry forward.

Behavioral model of the product is to establish this aforementioned relationship by means of engagement between the user and product over a varied means of interfaces such as Graphic UI, Voice Command and also the plant pot itself.

3.4 System Specification

The following are the system specification that the final product follows and a detail view of each is shortly followed in the next chapter.

3.4.1 Hardware Specification

1. ESP 8266 Microcontroller Module [8]
   - Processor - L106 32-bit RISC Microprocessor 80MHz
   - Memory - 32 KB instruction RAM
     32 KB instruction cache RAM
     80 KB user-data RAM
     6 KB ETS system-data RAM
     External QSPI flash: up to 16 MB is supported
   - Wi-Fi - IEEE 802.11 b/g/n
   - I^2S - Interfaces with DMA
   - GPIO - 16 Pins

2. Arduino UNO R3 [9]
   - Processor - ATmega328 16MHz
   - Flash Memory - 32 KB of which 0.5 KB used by boot loader
   - EEPROM - 1 KB
   - SRAM - 2 KB
   - DIO - 14 Pins
   - Input Voltage - 7 – 12 V (recommended)

3. Moisture Sensor FC-28
   - Input Voltage - 3.3 – 5 V
   - Output Voltage - 0 – 4.2 V
   - Input Current - 35mA
   - Output Signal - both Analog and Digital

4. LDR
   - Input Voltage - 3.3 – 5 V
   - Output Voltage - 0 – 4.2 V
   - Input Current - 35mA
   - Output Signal - both Analog and Digital

5. Temperature and Humidity Sensor DTH11
   - Input Voltage - 3.3 – 5 V
   - Output Voltage - 0 – 4.2 V
   - Input Current - 35mA
   - Output Signal - only Analog

6. Gas Sensor MQ2
   - Input Voltage - 3.3 – 5 V
   - Output Voltage - 0 – 4.2 V
   - Input Current - 35mA
   - Output Signal - both Analog and Digital

3.4.2 Software Specification

1. I^{2}C Software Implementation [10]
2. Arduino IDE [11]
3. Ada fruit Server [12]
4. IFTTT [13]
5. Google Assistant [14]

CHAPTER 4

MODULE DESCRIPTION

4.1 System Modules

4.1.1 ESP 8266 Microcontroller

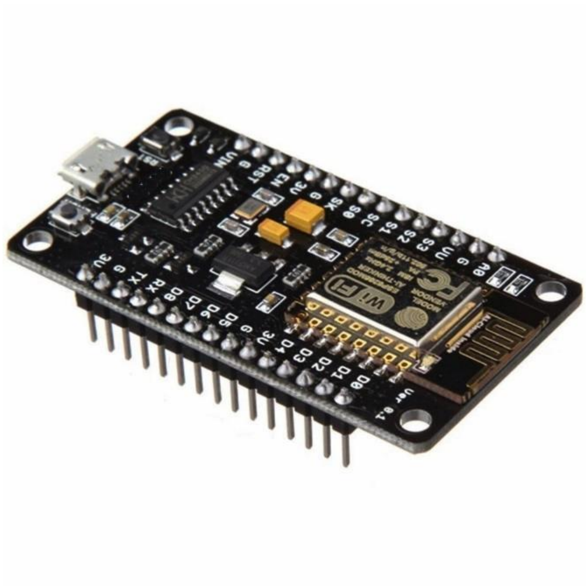

Figure 4.1.1 – ESP 8266 Microcontroller

The ESP-01 ESP8266 Serial WIFI Wireless Transceiver Module is a self-contained SOC with integrated TCP/IP protocol stack that can give any microcontroller access to your Wi-Fi network. The ESP8266 is capable of either hosting an application or offloading all Wi-Fi networking functions from another application processor. Each ESP8266 module comes preprogrammed with an AT command set firmware, meaning, you can simply hook this up to your Arduino device and get about as much Wi-Fi-ability as a Wi-Fi Shield.

4.1.2 Arduino UNO R3

Figure 4.1.2 – Arduino UNO R3

Arduino UNO R3 SMD is the open source Embedded Development board launched by Arduino based on Atmega328 SMD Package Microcontroller. Because Atmel is moving more and more of their production capacity to surface mount ICS, the DIP packaged ATmega is becoming more and more difficult to get. To keep up with demand, we now offer the Arduino UNO R3 with an SMD ATmega.

4.1.3 Moisture Sensor FC-28

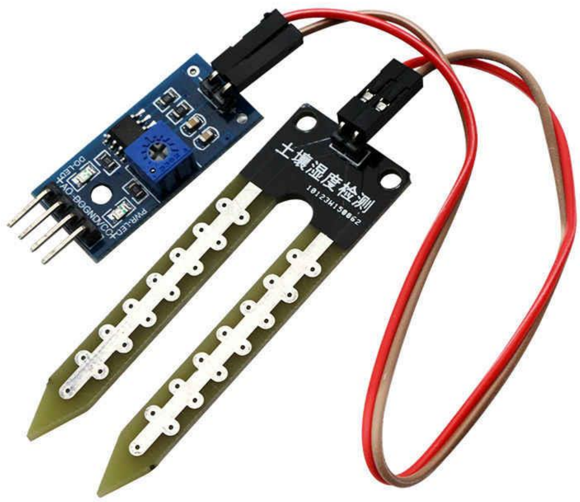

Figure 4.1.3 – Moisture Sensor FC-28

The sensor includes a potentiometer to set the desired moisture threshold. When the sensor measures more moisture than the set threshold, the digital output goes high and an LED indicates the output. When the moisture in the soil is less than the set threshold, the output remains low. The digital output can be connected to a micro controller to sense the moisture level. The sensor also outputs an analog output which can be connected to the ADC of a micro controller to get the exact moisture level in the solid.

4.1.4 LDR

Figure 4.1.4 – LDR

The *LDR* is a special type of resistor that allows higher voltages to pass through it (low resistance) whenever there is a high intensity of light, and passes a low voltage (high resistance) whenever it is dark.

4.1.5 Temperature and Humidity Sensor DTH11

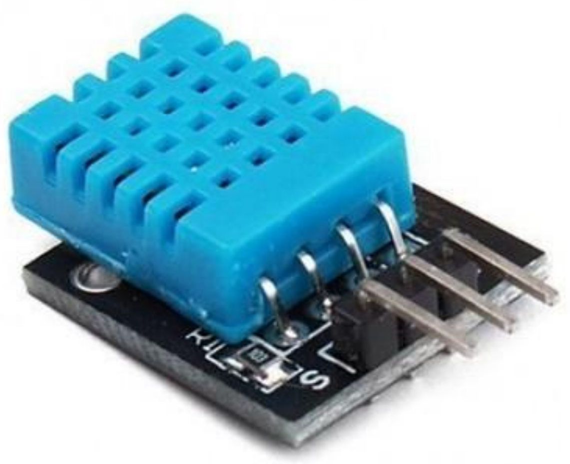

Figure 4.1.5 – Temperature and Humidity Sensor DTH11

DHT11 is a Humidity and Temperature Sensor, which generates calibrated digital output. DHT11 can be interface with any microcontroller like Arduino, Raspberry Pi, etc. and get instantaneous results. DHT11 is a low-cost humidity and temperature sensor which provides high reliability and long-term stability. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air, and outputs a digital signal on the data pin (no analog input pins needed). It's very simple to use, and libraries and sample codes are available for Arduino and Raspberry Pi.

This module makes is easy to connect the DHT11 sensor to an Arduino or microcontroller as includes the pull up resistor required to use the sensor. Only three connections are required to be made to use the sensor - VCC, G_{nd} and Output. It has high reliability and excellent long-term stability, thanks to the exclusive digital signal acquisition technique and temperature & humidity sensing technology.

4.1.6 Gas Sensor MQ2

Figure 4.1.6 – Gas Sensor MQ2

This is a very easy to use low cost semiconductor Gas sensor Module with analog and digital output. This module uses MQ2 Smoke & Flammable gas sensor as a gas sensing element. It requires no external components just plug in VCC & ground pins and you are ready to go. For Digital output the threshold value can be easily set by an on-board potentiometer. Using this module, you can easily interface MQ2 Smoke & Combustible gas Sensor to any Microcontroller, Arduino or even Raspberry Pi. Since this Gas Sensor module is sensitive to smoke it can be used in for fire detection. MQ2 Gas Sensor is also sensitive to flammable/combustible gasses like LPG, Propane & Hydrogen.

4.1.7 Submergible Water Pump

Figure 4.1.7 – Submergible Water Pump

Micro DC 3-6V Micro Submersible Pump Mini water pump For Fountain Garden Mini water circulation System DIY project. This is a low cost, small size Submersible Pump Motor which can be operated from a 3 ~ 6V power supply. It can take up to 120 liters per hour with very low current consumption of 220mA. Just connect tube pipe to the motor outlet, submerge it in water and power it. Make sure that the water level is always higher than the motor. Dry run may damage the motor due to heating and it will also produce noise. [15].

4.2 General Architecture

4.2.1 System Architecture

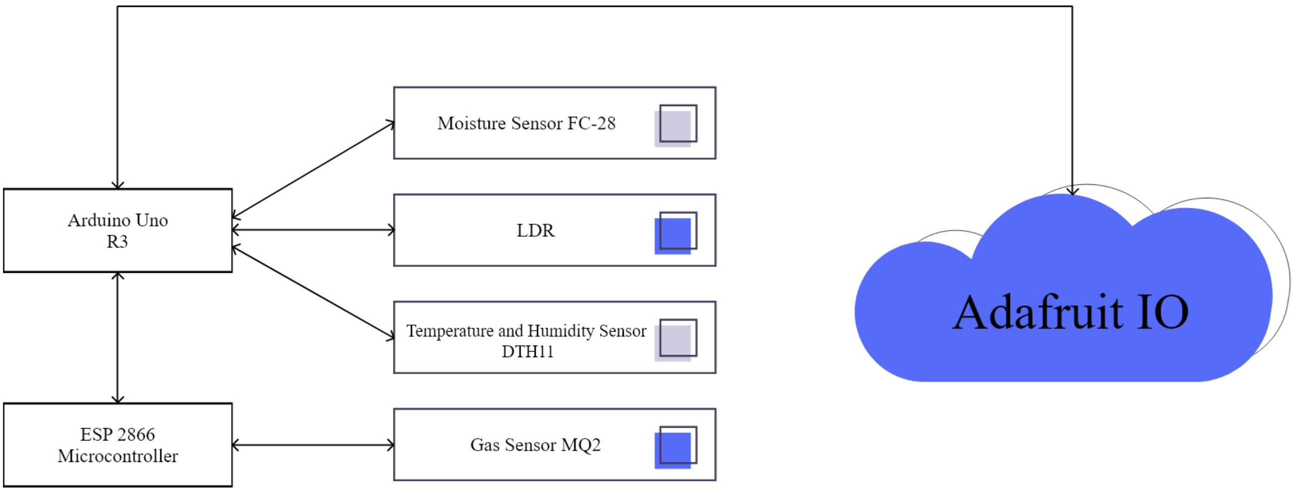

Figure 4.2.1 – Herb Cloud System Architecture

4.2.2 Architecture Description

The above diagram represents a block diagram of the entire architecture of the Herb Cloud system. The system uses two major microcontroller devices, namely the ESP 8266 Module and Arduino UNO R3, of which the first is connected to the internet as it supports easy connectivity using its integrated Wi-Fi module. The two MC are also connected to each other using a hard-wire cable.

The reason of using the Arduino UNO R3 on top of the ESP8266 is because UNO module can support numerous i/o pins and is much more compatible with the sensors than ESP 8266.

The moisture sensor FC-28 is earthed near the plant in order to measure the moisture content in the soil. Other sensors such as the LRD and DTH11 are placed near the plant, without any direct contact to it. All these sensors communicate with the UNO module, which in turn sends data collected over to the internet via ESP8266.

It should be noted that the gas sensor is directly connected to the ESP8266 module, since the main objective of inclusion of this Smoke and Gas sensor is to sent alert directly to the user without any delay whatsoever.

All the data collected from the sensors are send over to the Adafruit IO server, for a visual UI to the user. The feeds that track and operate this communication of data between Adafruit IO and the homely modules can be changed to soothe the use case.

The user uses this Adafruit provided interface to keep track and communicate with the Herb Cloud System remotely. Another way of communication is via the IFTTT protocol, which allows the user to call on commands such as “Turn ON Motor” and “Turn OFF Motor” to communicate and control the water motor in the plant system Herb Cloud.

This is the overall architecture description of Herb Cloud.

4.3 Block Diagram

4.3.1 Diagram

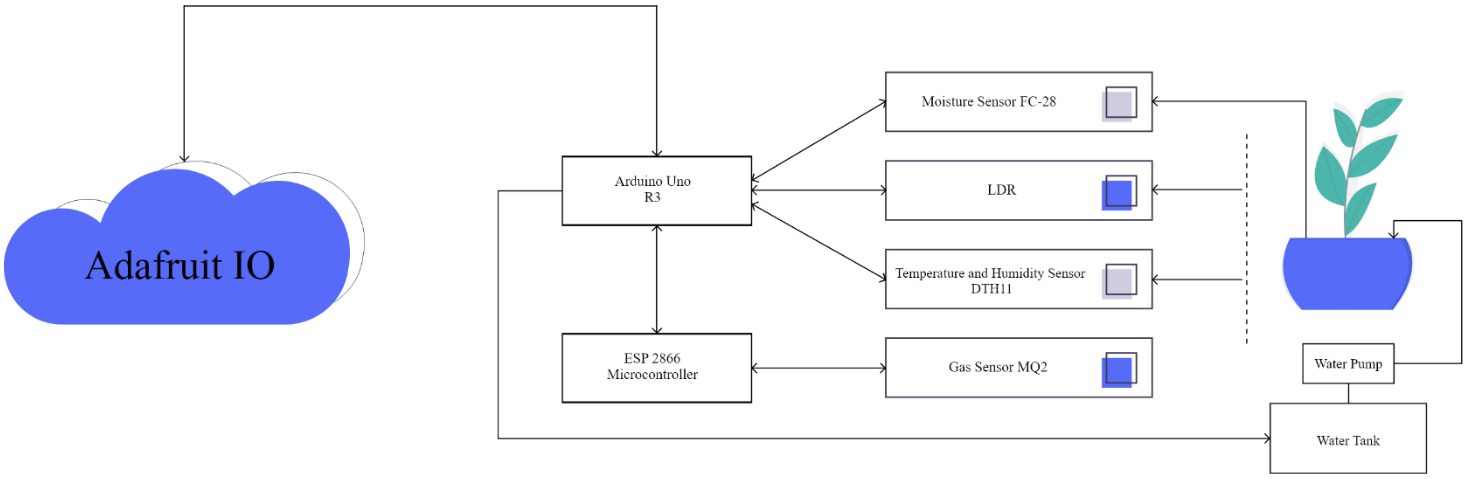

Figure 4.3.1 – Block Diagram

4.3.2 Description

As per the configuration of the system, developer choices which factor in the environment and location of the plant, and the plant itself, the variables configuring the sensor data may change. When the conditions meet as coordinated, in this example instance say moisture sensor and the water pump, the microcontroller turns the Water Pump switch ON, which pumps water from the tank it is placed in to the plant root directly via a tube.

As per the aforementioned description of the system, the two sensors LDR and DTH11 are not in direct contact with the plant, but near in its environment (represented by the marching-ants line). The gas sensor is also placed in the vicinity of the plant, directly connected the ESP8266 module.

The overall function and system architecture description is ted up here. For further intrinsic details and queries about Herb Cloud, [20].

CHAPTER 5

IMPLEMENTATION AND TESTING

5.1 Description

Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus, it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective. The implementation stage involves careful planning, investigation of the existing system and its constraints on implementation, designing of methods to achieve changeover and evaluation of changeover method.

5.2 Input Design

The initial design ideology was to, from scratch, to design the entire project module, the cloud storage and all of the various holding facilities using 3D design tool such as CATIA, AutoCAD Part Design and Microsoft Workbench.

Due to circumstances led by insufficient resources in 3D printing and various other time constraints factors, this idea of developing each part of the system in house was dropped and instead more focus was given in developing and optimizing technologies using the sensing, collection, distribution and analysis of data for an efficient factorial output.

5.3 Output Design

The final output of the project was done by using the aforementioned components, a module consisting of the plant pot, a cloud water storage and other minor holdings.
The images below show the final resulted output of the project.

Figure 5.3.1 – Module Front View

Figure 5.3.2 – Module Side View

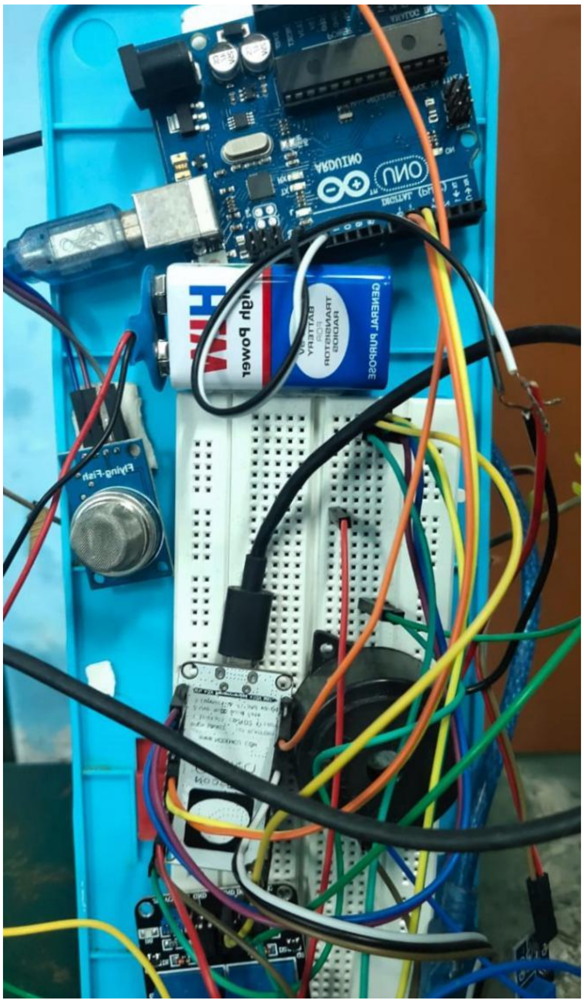

Figure 5.3.3 – Module Back View

5.4 Code

5.4.1 ESP8266

```
#include <ESP8266WiFi.h>
#include <Adafruit_Sensor.h>
#include <DHT.h>
#include <DHT_U.h>
#define DHTPIN D1
#include <SoftwareSerial.h>
#include "Adafruit_MQTT.h"
#include "Adafruit_MQTT_Client.h"
SoftwareSerial s(D6, D5);
#define DHTTYPE    DHT11
#define analogpin A0
#define Relay1 D0
#define WLAN_SSID       "UMAR"
#define WLAN_PASS       "password"
#define AIO_SERVER      "io.adafruit.com"
#define AIO_SERVERPORT  1883                   // use 8883 for SSL
#define AIO_USERNAME    "Umar_7w4"
#define AIO_KEY         "60c2ab0758c94a88bf0e789b63c44b47"
int data;7
DHT_Unified dht(DHTPIN, DHTTYPE);

uint32_t delayMS;
WiFiClient client;
Adafruit_MQTT_Client mqtt(&client, AIO_SERVER, AIO_SERVERPORT,
AIO_USERNAME, AIO_KEY);
Adafruit_MQTT_Subscribe Light1 = Adafruit_MQTT_Subscribe(&mqtt,
AIO_USERNAME"/feeds/motor");
Adafruit_MQTT_Publish sensor_1 = Adafruit_MQTT_Publish(&mqtt,
AIO_USERNAME "/feeds/moisture");
Adafruit_MQTT_Publish sensor_2 = Adafruit_MQTT_Publish(&mqtt,
AIO_USERNAME "/feeds/gas");
```

```
Adafruit_MQTT_Publish sensor_3 = Adafruit_MQTT_Publish(&mqtt, AIO_USERNAME "/feeds/temperature");
Adafruit_MQTT_Publish sensor_4 = Adafruit_MQTT_Publish(&mqtt, AIO_USERNAME "/feeds/hum");
void MQTT_connect();
void setup() {
   dht.begin();
  Serial.println(F("DHTxx Unified Sensor Example"));
  // Print temperature sensor details.
  sensor_t sensor;
  dht.temperature().getSensor(&sensor);
  Serial.println(F("------------------------------------"));
  Serial.println(F("Temperature Sensor"));
  Serial.print  (F("Sensor Type: ")); Serial.println(sensor.name);
  Serial.print  (F("Driver Ver:  ")); Serial.println(sensor.version);
  Serial.print  (F("Unique ID:   ")); Serial.println(sensor.sensor_id);
  Serial.print  (F("Max Value:   ")); Serial.print(sensor.max_value); Serial.println(F("°C"));
  Serial.print  (F("Min Value:   ")); Serial.print(sensor.min_value); Serial.println(F("°C"));
  Serial.print  (F("Resolution:  ")); Serial.print(sensor.resolution); Serial.println(F("°C"));
  Serial.println(F("------------------------------------"));
  // Print humidity sensor details.
  dht.humidity().getSensor(&sensor);
  Serial.println(F("Humidity Sensor"));
  Serial.print  (F("Sensor Type: ")); Serial.println(sensor.name);
  Serial.print  (F("Driver Ver:  ")); Serial.println(sensor.version);
  Serial.print  (F("Unique ID:   ")); Serial.println(sensor.sensor_id);
  Serial.print  (F("Max Value:   ")); Serial.print(sensor.max_value); Serial.println(F("%"));
  Serial.print  (F("Min Value:   ")); Serial.print(sensor.min_value); Serial.println(F("%"));
  Serial.print  (F("Resolution:  ")); Serial.print(sensor.resolution);
```

```
Serial.println(F("%"));
  Serial.println(F("------------------------------------"));
  // Set delay between sensor readings based on sensor details.
  delayMS = sensor.min_delay / 1000;
  Serial.begin(115200);
  s.begin(9600);
  delay(10);
  pinMode(Relay1, OUTPUT);
  pinMode(analogpin, INPUT);
Serial.println(F("Adafruit MQTT demo"));
Serial.println(); Serial.println();
Serial.print("Connecting to ");
Serial.println(WLAN_SSID);
  WiFi.begin(WLAN_SSID, WLAN_PASS);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println();
Serial.println("WiFi connected");
  Serial.println("IP address: "); Serial.println(WiFi.localIP());
  mqtt.subscribe(&Light1);
}
uint32_t x=0;
void loop() {
  delay(delayMS);
  sensors_event_t event;
  dht.temperature().getEvent(&event);
  if (isnan(event.temperature)) {
    Serial.println(F("Error reading temperature!"));
  }
  else {
    Serial.print(F("Temperature: "));
    Serial.print(event.temperature);
```

```
  Serial.println(F("°C"));
}
// Get humidity event and print its value.
dht.humidity().getEvent(&event);
if (isnan(event.relative_humidity)) {
  Serial.println(F("Error reading humidity!"));
}
else {
  Serial.print(F("Humidity: "));
  Serial.print(event.relative_humidity);
  Serial.println(F("%"));
}
  MQTT_connect();
  Adafruit_MQTT_Subscribe *subscription;
  s.write("s");
if (s.available()>0)
{
  data=s.read();
  Serial.println(data);
}
while ((subscription = mqtt.readSubscription(20000))) {
  if (subscription == &Light1) {
    Serial.print(F("Got: "));
    Serial.println((char *)Light1.lastread);
    int Light1_State = atoi((char *)Light1.lastread);
    digitalWrite(Relay1, Light1_State);


  }
Serial.print(F("\nSending Sensor's Value "));
Serial.print("Sensor 1 ");Serial.println(analogRead(analogpin));
Serial.print("...");
int Value = analogRead(analogpin);
if (! sensor_1.publish(Value)) {
  Serial.println(F("Failed"));
```

```
  } else {
    Serial.println(F("OK!"));
  }
  if (! sensor_2.publish(data)) {
    Serial.println(F("Failed"));
  } else {
    Serial.println(F("OK!"));
  }
  if (! sensor_3.publish(event.temperature)) {
    Serial.println(F("Failed"));
  } else {
    Serial.println(F("OK!"));
  }
  if (! sensor_4.publish(event.relative_humidity)) {
    Serial.println(F("Failed"));
  } else {
    Serial.println(F("OK!"));
  }


  delay(2000);
  }}
void MQTT_connect() {
  int8_t ret;
if (mqtt.connected()) {
    return;
  }
Serial.print("Connecting to MQTT... ");
  uint8_t retries = 3;
  while ((ret = mqtt.connect()) != 0) { // connect will return 0 for connected
       Serial.println(mqtt.connectErrorString(ret));
       Serial.println("Retrying MQTT connection in 5 seconds...");
       mqtt.disconnect();
       delay(5000);  // wait 5 seconds
```

```
        retries--;
        if (retries == 0) {
                while (1);
        }
  }
  Serial.println("MQTT Connected!");
}
```

5.4.2 Arduino UNO R3

```
#include <SoftwareSerial.h>
#define gas_sensor A0
SoftwareSerial s(5,6);
int gas_data=0;

void setup() {
s.begin(9600);
pinMode(gas_sensor,INPUT);
pinMode(13,OUTPUT);
}

void loop() {
gas_data= analogRead(gas_sensor);
if(s.available()>0)
{
 s.write(gas_data);
}
if(gas_data>400)
{
  digitalWrite(13,HIGH);
}          }
```

5.5 Sample Screenshots

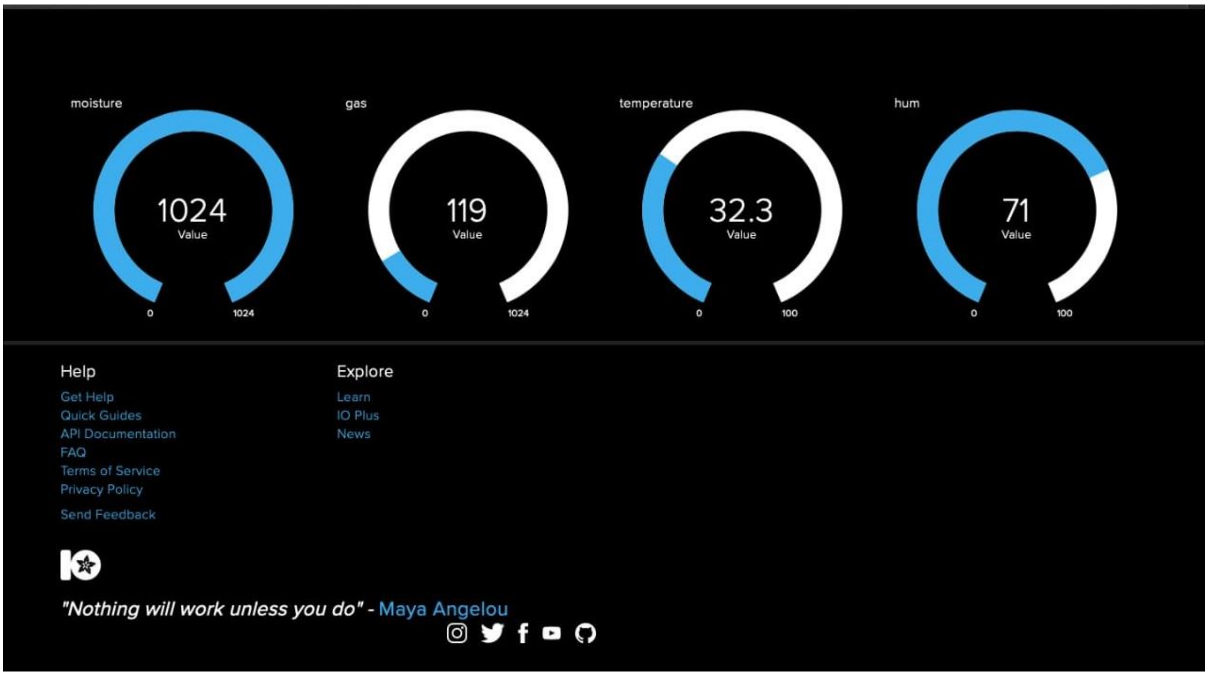

Figure 5.5.1 – Desktop Dashboard

Figure 5.5.2 – Mobile Dashboard

Figure 5.2.3 – Google Assistant View

5.6 Testing

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product.

The final testing was done with the entire module set up. Various reading feed from the sensors to the Adafruit server was compared, and responses from Google Assistant Voice control, and Adafruit UI described controls.

Minor adjustments to sensor data feed and calculation methodologies were made in order to soothe the given testcase plant and the environment. In real-time usage, these adjustments will be needed according to a predefined plant profiling module, described in further detail in the forthcoming section of Future Enhancements.

CHAPTER 6

RESULTS AND DISCUSSION

6.1 Results and Efficiency of the Proposed System

The final system worked seamlessly as intended during the later and final stages of the testing procedure. The team was able to send and receive requests and information from the system with minor complications in heed.

One of the major complications was the UI service, which the team used, namely Adafruit IO, provided with only three input/output feeds for a given dashboard and limit the number of signals via its cloud services to about one hundred per minute. This limited the density of information the team had to work with. For further insight, look into Adafruit IO Services [16].

6.2 Advantages over the preexisting system

As discussed earlier, the preexisting system are confined with the overall end-user cost was very high in regards to the technologies used. Assembly was difficult and not at-all user friendly in nature. The system architecture was neat only on paper, the final product looked messy, at ease to falter. Advances in technologies such as IoT and ML were not inclusive in any of the products. Plant-type based optimization was missing.

The designed system has the following leading advantages over the older, preexisting systems:

1. It is very simple to use and requires minimalistic user-intervention
2. The system can configure modes for a more personalized interaction with the user.
3. Overall design was made with consideration of home-aesthetics and thus looks a lot friendly, in contrast with systems full of wires, and unrecouped components.
4. It takes advantages of modern tools in computation such as IoT and Machine-learning techniques.
5. The cost for production and assembly is low.
6. Advances and future enhancements can be easily made due to the abstraction mode of programming implemented.
7. Inclusion of emergency-alert services as well.

CHAPTER 7

CONCLUSION AND FUTURE ENHANCEMENTS

7.1 Conclusion

This project gave insights into many reals of both electronics and computer science paradigms to the team member. By the application of various techniques learned in this course, and others for different sources, the team believes the end result produce is very viable and such a project is feasible for future considerations.

The team will continue optimizing and introducing new features to the project so as it can be really viable in the long run, to one day, see this very project as a real product in hands and homes of the consumers.

7.2 Future Enhancements

The module and all its programming have been done with abstract thinking in mind, thus later additions to the code can be done with ease. A personalized system with profiling for each and every plant can be maintained and keep structured both by the admins and the users, for the benefit of the entire committee.

Another future enhancement would be addition on Machine Learning techniques into the Already preexisting system architecture as show in the diagram below. This can vastly affect the performance of the system and its efficiency as the system will now be capable of learning and adapting within its own boundaries. Use of Google TensorFlow is suggested for this application.

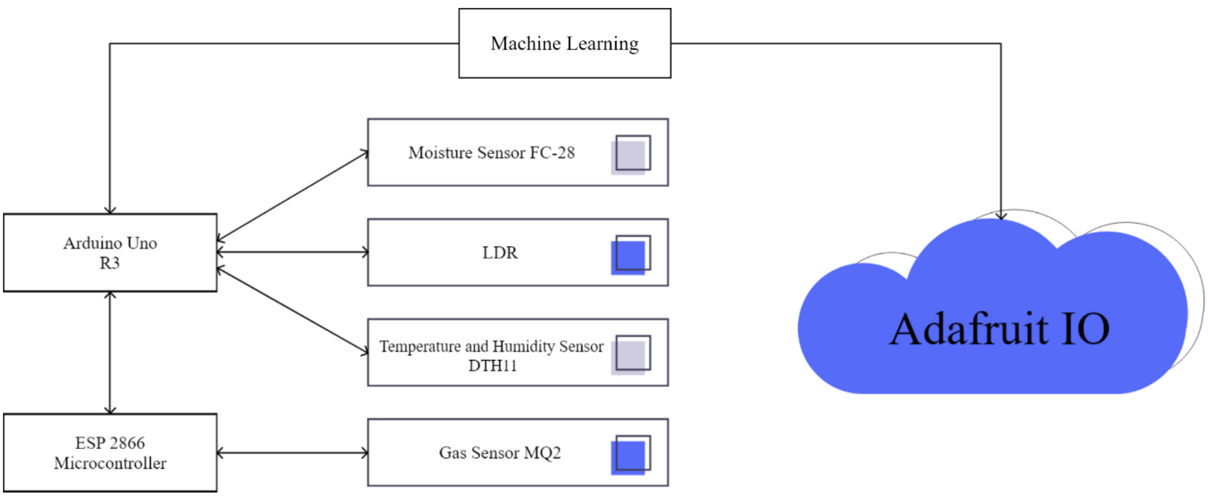

Figure 7.2 – Enhanced System Architecture

One of the most promising features in enhancing the system would be integrating and optimizing this system for large-scale agricultural farms to reduce wastage, increase labor efficiency and the quality in-gross quantity of the produce.

APPENDIX

I) LIST OF ABBREVIATIONS

Abbreviations	Explanation
1. API	Application Program Interface
2. AT	At Time-Interval
3. DIO	Direct Input/output
4. DIP	Direct Interaction Programming
5. DIY	Do It Yourself
6. EEPROM	Electronically Erasable Programmable Read Only Memory
7. GPIO	Ground Pins Input/output
8. GUI	Graphics User Interface
9. IFTTT	If This…Then That
10. IJITEE	International Journal of Innovative Technology and Exploring Engineering
11. IoT	Internet of Things
12. LDR	Light Dependent Resistor
13. SAS	Smart Analytical System
14. SMD	Surface Mount Device
15. SRAM	Static Random-Access Memory
16. TCP/IP	Transmission Control Protocol/Internet Protocol
17. UI	User Interface

II) LIST OF FIGURES

BIBLOGRAPHY

[1] Wireless home automation networks: A survey of architectures and technologies, Publisher: IEEE, Carles Gomez ; Josep Paradells , Published in: IEEE Communications Magazine (Volume: 48 , Issue: 6 , June 2010)

[2] The design of general purpose autonomous agricultural mobile-robot: "AGROBOT", Publisher: IEEE, Halil Durmuş ; Ece Olcay Güneş ; Mürvet Kırcı ; Burak Berk Üstündağ , Robotics and Intelligent Sensors (IRIS) 2016 IEEE International Symposium on, pp. 194-198, 2016.

[3] Internet of Things (IoT): A vision, architectural elements, and future directions, ScienceDirect, Jayavardhana Gubbi ; Rajkumar Buyyab ; Slaven Marusica ; Marimuthu Palaniswamia, Volume 29, Issue 7, September 2013, Pages 1645-1660

[4] The basic roles of indoor plants in human health and comfort, Springer-Verlag GmbH Germany, part of Springer Nature 2018, Linjing Deng ; Qihong Deng, December 2018, Volume 25, Issue 36, pp 36087–36101

[5] Raspberry Pie Smart Garden System – Kickstarter – https://www.kickstarter.com/projects/sunair/raspberry-pi-smart-garden-system-sgs – 23rd October 2019

[6] Smart Garden Solutions – Mobiloitte – https://www.mobiloitte.com/solutions/smart-garden – 23rd October 2019

[7] Smart Gardening System,International Journal of Innovative Technology and Exploring Engineering (IJITEE)ISSN: 2278-3075, Volume-8 Issue-5S March, 2019, Mithya V, Aishwarya M, Gayathri S, Mahalakshmi L S, Pavithra S, Volume-8 Issue5S March, 2019

[8] ESP8266 – Wikipedia – https://en.wikipedia.org/wiki/ESP8266 – 23rd October 2019

[9] Arduino UNO R3 – Hobbytronics – http://www.hobbytronics.co.uk/arduinouno-r3 – 23rd October 2019

[10] I2C – Wikipedia – https://en.wikipedia.org/wiki/I%C2%B2C – 23rd October 2019

[11] Software – Arduino – https://www.arduino.cc/en/main/software – 23rd October 2019

[12] Homepage – Adafruit – https://www.adafruit.com/ – 23rd October 2019

[13] Homepage – IFTTT – https://ifttt.com/ – 23rd October 2019

[14] Google Assistant Home Page – Google – https://assistant.google.com/#?modal_active=none – 31st October 2019

[15] Homepage – ElectronicsComp – https://www.electronicscomp.com/ – 31st October 2019

[16] References – Adafruit – https://learn.adafruit.com/welcome-to-adafruitio/what-is-adafruit-io – 1st November 2019