FOOD SPOILAGE DETECTION USING ARDUINO

A Project Report submitted to

JAWAHARLAL NEHRU TECHNOLOGICAL UNVERSITY ANANTAPUR.

In Partial Fulfillment of the Requirements for the Award of the degree of

BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE AND ENGINEERING
BY

P KRISHNAMURALI	**17121A05G9**
N HARISHA	**17121A05D7**
R SANKETH KUMAR	**17121A05H5**
P YASWANTH REDDY	**17121A05G3**

Under the Guidance of

Dr. K. Suresh
Associate professor
Dept of CSE, SVEC

Department of Computer Science and Engineering
SREE VIDYANIKETHAN ENGINEERING COLLEGE
(Affiliated to JNTUA, Anantapuramu)
Sree Sainath Nagar, Tirupathi – 517 102

2017-2021

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VISION AND MISSION

VISION

To become a Centre of Excellence in Computer Science and Engineering by imparting high quality education through teaching, training and research.

MISSION

The Department of Computer Science and Engineering is established to provide undergraduate and graduate education in the field of Computer Science and Engineering to students with diverse background in foundations of software and hardware through a broad curriculum and strongly focused on developing advanced knowledge to become future leaders.

Create knowledge of advanced concepts, innovative technologies and develop research aptitude for contributing to the needs of industry and society.

Develop professional and soft skills for improved knowledge and employability of students.

Encourage students to engage in life-long learning to create awareness of the contemporary developments in computer science and engineering to become outstanding professionals.

Develop attitude for ethical and social responsibilities in professional practice at regional, National and International levels.

Program Educational Objectives (PEO's)

1. Pursuing higher studies in Computer Science and Engineering and related disciplines

2. Employed in reputed Computer and I.T organizations and Government or have established startup companies.

3. Able to demonstrate effective communication, engage in team work, exhibit leadership skills, ethical attitude, and achieve professional advancement through continuing education.

Program Specific Outcomes (PSO's)

1. Demonstrate knowledge in Data structures and Algorithms, Operating Systems, Database Systems, Software Engineering, Programming Languages, Digital systems, Theoretical Computer Science, and Computer Networks. (PO1)

2. Analyze complex engineering problems and identify algorithms for providing solutions (PO2)

3. Provide solutions for complex engineering problems by analysis, interpretation of data, and development of algorithms to meet the desired needs of industry and society. (PO3, PO4)

4. Select and Apply appropriate techniques and tools to complex engineering problems in the domain of computer software and computer based systems (PO5)

Program Outcomes (PO's)

1. Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems (**Engineering knowledge**).

2. Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences (**Problem analysis**).

3. Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations (**Design/development of solutions**).

4. Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions (**Conduct investigations of complex problems**).

5. Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations (**Modern tool usage**)

6. Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice (**The engineer and society**)

7. Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development (**Environment and sustainability**).

8. Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice (**Ethics**).
9. Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings (**Individual and team work**).

10. Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions (**Communication**).

11. Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments (**Project management and finance**).

12. Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change (**Life-long learning**).

Course Outcomes

CO1. Knowledge on the project topic (PO1)

CO2. Analytical ability exercised in the project work.(PO2)

CO3. Design skills applied on the project topic. (PO3)

CO4. Ability to investigate and solve complex engineering problems faced during the project work. (PO4)

CO5. Ability to apply tools and techniques to complex engineering activities with an understanding of limitations in the project work. (PO5)

CO6. Ability to provide solutions as per societal needs with consideration to health, safety, legal and cultural issues considered in the project work. (PO6)

CO7. Understanding of the impact of the professional engineering solutions in environmental context and need for sustainable development experienced during the project work. (PO7)

CO8. Ability to apply ethics and norms of the engineering practice as applied in the project work.(PO8)

CO9. Ability to function effectively as an individual as experienced during the project work. (PO9)

CO10. Ability to present views cogently and precisely on the project work. (PO10)

CO11. Project management skills as applied in the project work. (PO11)

CO12. Ability to engage in life-long leaning as experience during the project work. (PO12)

CO-PO Mapping

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO10	PO11	PO12	PSO1	PSO 2	PS O3	PSO 4
CO1	3												3			
CO2		3												3		
CO3			3												3	
CO4				3											3	
CO5					3											3
CO6						3										
CO7							3									
CO8								3								
CO9									3							
CO10										3						
CO11											3					
CO12												3				

(Note: 3-High, 2-Medium, 1-Low)

DECLARATION

We hereby declare that this project report titled **"FOOD SPOILAGE DETECTION USING ARDUINO"** is a genuine project work carried out by us, in **B.Tech *(Computer Science and Engineering)*** degree course of **Jawaharlal Nehru Technological University Anantapur** and has not been submitted to any other course or University for the award of any degree by us.

Signature of the student

1. P. Krishnamurali
2. N. Harisha
3. R. Sanketh kumar
4. P. Yaswanth reddy

SREE VIDYANIKETHAN ENGINEERING COLLEGE

(Affiliated to Jawaharlal Nehru Technological University Anantapur)

Sree Sainath Nagar, A. Rangampet, Tirupati – 517 102, Chittoor Dist., A.P.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Project Work entitled

"FOOD SPOILAGE DETECTION USING ARDUINO"

is the bonafide work done by

P KRISHNAMURALI	**17121A05G9**
N HARISHA	**17121A05D7**
R SANKETH KUMAR	**17121A05H5**
P YASWANTH REDDY	**17121A05G3**

In the Department of Computer Science and Engineering, Sree Vidyanikethan Engineering College, A. Rangampet. is affiliated to JNTUA, Anantapuramu in partial fulfillment of the requirements for the award of Bachelor of Technology in Computer Science and Engineering during 2017-2021.

This is work has been carried out under my guidance and supervision.

The results embodied in this Project report have not been submitted in any University or Organization for the award of any degree or diploma.

Internal Guide

Dr. K. Suresh

Associate professor

Dept of CSE

Sree Vidyanikethan Engineering College

Tirupathi

Head

Dr. B. Narendra Kumar Rao

Prof & Head

Dept of CSE

Sree Vidyanikethan Engineering College

Tirupathi

INTERNAL EXAMINER

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

We are extremely thankful to our beloved Chairman and founder **Dr. M. Mohan Babu** who took keen interest to provide us the infrastructural facilities for carrying out the project work.

We are highly indebted to **Dr. B. M. Satish**, Principal of Sree Vidyanikethan Engineering College for his valuable support and guidance in all academic matters.

We are very much obliged to **Dr. B. Narendra Kumar Rao,** Professor & Head, Department of CSE, for providing us the guidance and encouragement in completion of this project.

We would like to express our indebtedness to the project coordinator, **Dr. K. Suresh**, assoicate professor, Department of CSE for his valuable guidance during the course of project work.

We would like to express our deep sense of gratitude to **Dr. K. Suresh**, assoicate professor, Department of CSE, for the constant support and invaluable guidance provided for the successful completion of the project.

We are also thankful to all the faculty members of CSE Department, who have cooperated in carrying out our project. We would like to thank our parents and friends who have extended their help and encouragement either directly or indirectly in completion of our project work.

ABSTRACT

Food safety and hygiene is a major concern in order to prevent food wastage. The Quality of the food needs to be monitored and it must be prevented from rotting and decaying by the atmospheric factors like temperature, humidity and dark. Therefore, it is useful to deploy quality monitoring devices at food stores. These quality monitoring devices keep a watch on the environmental factors that cause or pace up decay of the food. Later, the environmental factors can be controlled like refrigeration, vacuum storage etc. A similar food quality monitoring device will be designed that will keep watch of environmental factors like temperature, humidity, alcohol content and exposure to light. The device is built on Arduino UNO which is a popular prototyping board. The Arduino board is interfaced with various sensors like DHT-11 to monitor temperature and humidity, MQ3 to detect alcohol content and LDR to measure exposure to light. This is an IoT device and sends the measured sensor data to an IoT platform. The ESP8266 Wi-Fi Modem is interfaced with the Arduino to connect it to the internet via Wi-Fi router. The sensor data is also displayed on a character LCD interfaced with the Arduino UNO. The IoT platform used for logging and monitoring of sensor data is embedded spot.

TABLE OF CONTENTS

CHAPTER-1

INTRODUCTION

1.1. INTRODUCTION:

Food spoilage is the process where a food product becomes unsuitable to ingest by the consumer. The cause of such a process is due to many outside factors as a side-effect of the type of product it is, as well as how the product is packaged and stored. Due to food spoilage, one-third of the world's food produced for the consumption of humans is lost every year[1]. Bacteria and various fungi are the cause of spoilage and can create serious consequences for the consumers, but there are preventive measures that can be taken.

Food being the main source of energy for the living beings, the food quality and safety has been the highest demand throughout the history. Technologies like Internet of things (IoT) connects anything at anywhere and anytime. By utilizing the IoT in the food supply chain (FSC), it helps to enhance the life quality by tracking and tracing the condition of the food and live by sharing the data obtained from the consumers.

Currently, the entire application of the IoT technology in FSC is still in the initial stage and a huge distance for improvements. The food hygiene and safety is the major concern to prevent the food from wastage. The food quality must be monitored and also should be prevented from decaying and routing the atmospheric factors such as darkness, humidity and temperature. Therefore, by deploying quality monitoring devices at food stores the wastage can be minimized. These kind of quality monitoring devices check on the atmospheric factors that are capable of causing food decay.

Previously, the atmospheric factors were controlled by the techniques such as refrigeration, vacuum storage etc., the contamination of food shall

occur in the process of production, but mostly caused because of inefficient food handling practices due to the inappropriate ambient conditions during the food transportation and storage. There are various factors that leads to food poisoning, like humidity and temperature changes. So that the monitoring system which is capable of measuring the humidity and temperature difference during the food storage and transportation is of prime importance.

Currently, almost everyone is getting affected by the foods they consume daily, not because of the junk food, but also the canned vegetables, food products that are consumed in daily-life, since their oxygen, temperature, and moisture content varies from time to time, they do not offer great quality. As most of consumers only pays attention towards the data provided on the packets, such as, the quantity of ingredients that are used and the nutritional value, they are blindly risking on their health by ignoring the atmospheric conditions to which these packages are subjected. To ensure the safety of the food, at every stage of supply chain, it must be monitored.

It employs the purpose of protection to the consumer health by maintaining the required ambient conditions that are essential for food quality maintenance. The analysis and performance of the routine measurements, aims at the changes detection does not guarantee in the nutritional quality of the food. For the purposes of policy analysis, trend forecasting, program evaluation and planning, the information are collected by monitoring and surveillance should be analyzed and sent to the decision makers in an appropriate format to be of real value.

Distribution of data must be an interactive process. The foremost job of the monitoring and control systems is to monitor a particular activity to make certain that it rests in the desired manner. By using various

electronic sensors monitoring can be attained. Further for the controlling purpose, these recorded values shall be used. The information that are obtained from the Arduino-based sensors will be compared to the desired values. If the readings of the sensor are found to be differing from the desired values, then the control circuit would come into action to influence the allotted activity to maintain it in the desired quality. We propose the use of this principle to build a system that can preserve the raw food materials. The smart food monitoring system is primarily aimed to monitor and control the food items and thus preventing it from the spoilage that may occur due to climatic and atmospheric changes. Also the inappropriate storing of the food items shall lead to food wastage.

Thus the Smart food monitoring system mainly focus on the safe food storage by monitoring and controlling numerous parameters that are affecting the food items. This monitoring system uses the storage units that are embedded with various Arduino-based sensors that will read the parameters that are affecting the quality of the food items. The Control circuits are designed to overcome the issue of unenviable food storage conditions are the important part of this proposal. This project proposes an IoT foundation for easing the monitoring of the food items, so that it do not get spoiled due to the atmospheric conditions during storage and transportation.

The main cause of food spoilage is invasion by microorganisms such as fungi and bacteria.

1.2. TYPES OF FOOD SPOILAGE:

1.2.1 MICROBIAL SPOILAGE:

Microbial spoilage is caused by microorganisms like fungi (moulds, yeasts) and bacteria. They spoil food by growing in it and producing substances that change the color, texture and odour of the food. Eventually the food will be unfit for human consumption.

Fig 1.1 spoiled bread

When food is covered with a furry growth and becomes soft and smells bad, the spoilage is caused by the growth of moulds and yeasts (look back at Figure 1.1). Microbial spoilage by moulds and yeasts includes souring of milk, growth of mould on bread and rotting of fruit and vegetables. These organisms are rarely harmful to humans, but bacterial contamination is often more dangerous because the food does not always look bad, even if it is severely infected. When microorganisms get access to food, they utilise the nutrients found in it and their numbers rapidly increase. They change the food's flavour and synthesise new compounds that can be harmful to

humans. Food spoilage directly affects the color, taste, odour and consistency or texture of food, and it may become dangerous to eat. The presence of a bad odour or smell coming from food is an indication that it may be unsafe. But remember that not all unsafe food smells bad.

The term **contact spoilage** is used when microbial spoilage is the result of direct contact or touching between the food and any contaminated or unclean surface such as shelves, food preparation boards or unwashed hands. It also includes food-to-food contact, for example between cooked meat and raw meat or between rotting fruit and sound fruit.

1.2.2 PHYSICAL SPOILAGE:

Physical spoilage is due to physical damage to food during harvesting, processing or distribution. The damage increases the chance of chemical or microbial spoilage and contamination because the protective outer layer of the food is bruised or broken and microorganisms can enter the foodstuff more easily. For example you may have noticed that when an apple skin is damaged, the apple rots more quickly.

1.2.3 CHEMICAL SPOILAGE:

Chemical reactions in food are responsible for changes in the color and flavor of foods during processing and storage. Foods are of best quality when they are fresh, but after fruits and vegetables are harvested, or animals are slaughtered, chemical changes begin automatically within the foods and lead to deterioration in quality. Fats break down and become rancid (smell bad), and naturally-occurring enzymes promote major chemical changes in foods as they age.

Enzymic spoilage (autolysis)

Every living organism uses specialized proteins called **enzymes** to drive the chemical reactions in its cells. After death, enzymes play a role in the decomposition of once-living tissue, in a process called **autolysis** (self-destruction) or **enzymic spoilage**. For example, some enzymes in a tomato help it to ripen, but other enzymes cause it to decay (Figure 1.2). Once enzymic spoilage is under way, it produces damage to the tomato skin, so moulds can begin to can attack it as well, speeding the process of decay.

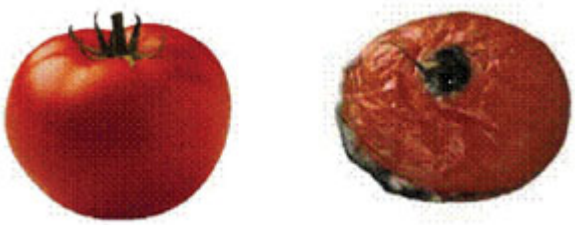

Fig 1.2 spoiled tomato due to enzymes

Enzymic browning

When the cells of fruits and vegetables such as apples, potatoes, bananas and avocado are cut and exposed to the air, enzymes present in the cells bring about a chemical reaction in which colourless compounds are converted into brown-coloured compounds. This is called **enzymic browning**. If the food is cooked very soon after cutting, the enzymes are destroyed by heat and the browning does not occur. For example, apples are prone to discolouration if cut open when raw, but when cooked they do not go brown.

1.2.4 APPEARANCE OF SPOILED FOOD:

Spoiled food is generally more a problem of appearance than a problem of disease causing. In food spoilage, the changes in appearance or texture of the food, such as rottenness, softness and change in colour, taste or odour are usually obvious, whereas in contaminated food such characteristics may not be noticed. A large majority of the microorganisms responsible for food spoilage are not pathogenic to humans. However, you should advise people in your community that they should not eat food that is spoiled because it is not nutritious and may make them sick (cause vomiting).

1.3. FACTORS AFFECTING FOOD SPOILAGE:

Finally, we turn to the factors that can increase or delay the process of food spoilage. They include its water content, environmental conditions, packaging and storage.

1.3.1 WATER CONTENT:

The amount of water available in a food can be described in terms of the water activity (a_w).

The water activity of most fresh foods is 0.99. This means that they have a very high water content and can support a lot of microbial growth.

1.3.2 ENVIRONMENTAL CONDITIONS:

No matter whether food is fresh or processed, the rate of its deterioration or spoilage is influenced by the environment to which it is exposed. The exposure of food to oxygen, light, warmth or even small amounts of moisture can often trigger a series of damaging chemical and/or microbial reactions. Changing the environment can help to delay spoilage. For example, storing foods at low temperatures reduces spoilage because both microbial and enzymic decay is faster at higher temperatures.

1.3.3 PACKAGING AND STORAGE:

Packaging is a means of safeguarding food when it is raw, or after it has been processed or prepared. It helps to protect food against harmful contaminants in the environment or conditions that promote food spoilage including light, oxygen and moisture. The type of packaging is a key factor in ensuring that the food is protected. Packaging of foods in cans, jars, cartons, plastics or paper also serves to ensure food safety if it is intact, because it provides protection against the entry of microorganisms, dust, dirt, insects, chemicals and foreign material.

1.4. PREVENTIVE MEASURES:

A number of methods of prevention can be used that can either totally prevent, delay, or otherwise reduce food spoilage. A food rotation system uses the first in first out method, which ensures that the first item purchased is the first item consumed.

Preservatives can expand the shelf life of food and can lengthen the time long enough for it to be harvested, processed, sold, and kept in the consumer's home for a reasonable length of time. One of the age old

techniques for food preservation, to avoid mold and fungus growth, is the process of drying out the food or dehydrating it. While there is a chance of it developing a fungus targeted towards dried food products, the chances are quite low.[2]

Other than drying, other methods include salting, curing, canning, refrigeration, freezing, preservatives, irradiation, and high hydrostatic pressure:[3] Refrigeration can increase the shelf life of certain foods and beverages, though with most items, it does not indefinitely expand it. Freezing can preserve food even longer, though even freezing has limitations. Canning of food can preserve food for a particularly long period of time, whether done at home or commercially. Canned food is vacuum packed in order to keep oxygen, which is needed by bacteria in aerobic spoilage, out of the can. Canning does have limitations, and does not preserve the food indefinitely.[3] Lactic acid fermentation also preserves food and prevents spoilage.

Food like meat, poultry, milk and cream should be kept out of the Danger Zone (between 4°C / 40°F to 60°C / 140°F). Anything between that range is considered dangerous and can cause pathogenic toxins to be emitted, resulting in severe illness in the consumer.[4] Another way to keep food from spoiling is by following a four step system: Clean, Separate, Cook, Chill.[4]

1.5. CONSEQUENCES:

Spoilage bacteria do not normally cause "food poisoning"; typically, the microorganisms that cause foodborne illnesses are odorless and flavourless, and otherwise undetectable outside the lab.[5][6] Eating deteriorated food could not be considered safe due to mycotoxins or microbial wastes. Some pathogenic bacteria, such as Clostridium perfringens and Bacillus cereus, are capable of causing spoilage.[7]

Issues of food spoilage do not necessarily have to do with the quality of the food, but more so with the safety of consuming said food. However, there are cases where food has been proven to contain toxic ingredients. 200 years ago, Claviceps purpurea, a type of fungus, was linked to human diseases and 100 years ago in Japan, yellow rice was found to contain toxic ingredients.[12]

Some spoiled foods are harmless to eat, and may simply be diminished in quality. But foods exhibiting certain types of spoilage may be harmful to consume. Uncooked or under-cooked animal flesh that spoils is typically quite toxic, and consumption can result in serious illness or death. The toxic effects from consuming spoiled food are known colloquially as "food poisoning", and more properly as "foodborne illness".

1.6. STATEMENT OF THE PROBLEM:

The problem statement is to detect the spoiled food and to send emails to the corresponding user.

1.7. OBJECTIVES:

- The primary objective is to detect the spoiled food using arduino uno and nodemcu module
- This system also helps us to store fruits and vegetables for longer period of time as the period of freshness can be monitored and determined.
- Email alerts.
    - Email should be sent when the air quality is not good.
    - Email should be sent when the food is spoiled
- Sensor data is stored in the cloud, that is in thinkspeak.com
    - sensor data should be stored in the form of json
- Data visualization using freeboard.io
    - To view last feed data into the cloud

1.8. SCOPE:

- The scope of the project is to establish a algorithm that can detect the spoiled food ahead of time.
- The data stored in the cloud can be downloaded via csv file and analytics can be done using ML and DL algorithms
- Air quality is also detected so that the device can sent an email to the user saying that the food is about to get spoiled. By this, Food spoilage can be detected earlier.

1.9. APPLICATIONS:

- Food spoilage can be detected
- Dataset can be made using the data stored in the cloud and ML, DL algorithms can be performed on this dataset.
- Air quality is detected around the food which means that the air quality across the room is also determined
- Light dependent sensor is used by which the amount of light can be measured.
- Temperature and humidity of the room can also be calculated using the sensor which we are using.

1.10. LIMITATIONS:

- Slow updates on the thingspeak because the free version is used.
- freeboard.io will be expired after 30 days if the free version is used.
- There might be outliers because of the delays.
- Some of the sensors used in the project are old. So, sometimes the values are not captured.
- Only email is being sent. If we want to send text message, etc. paid version of ifttt.com should be used.

CHAPTER-2

LITERATURE SURVEY

[1] LoneGram, MariaRasch, Jesper Bartholin Bruhn, AllanBChristensen, MichaelGivskov

Food spoilage is a complex process and excessive amounts of foods are lost due to microbial spoilage even with modern day preservation techniques. Despite the heterogeneity in raw materials and processing conditions, the microflora that develops during storage and in spoiling foods can be predicted based on knowledge of the origin of the food, the substrate base and a few central preservation parameters such as temperature, atmosphere, aw and pH. Based on such knowledge, more detailed sensory, chemical and microbiological analysis can be carried out on the individual products to determine the actual specific spoilage organism. Whilst the chemical and physical parameters are the main determining factors for selection of spoilage microorganisms, a level of refinement may be found in some products in which the interactive behavior of microorganisms may contribute to their growth and/or spoilage activity. This review gives three such examples. We describe the competitive advantage of Pseudomonas spp. due to the production of iron-chelating siderophores, the generation of substrates for spoilage reactions by one organism from another microorganism (so-called metabiosis) and the up-regulation of phenotypes potentially involved in spoilage through cell-to-cell communication. In particular, we report for the first time the widespread occurrence of N-acyl homoserine lactones (AHL) in stored and spoiling fresh foods and we discuss the potential implications for spoilage and food preservation.

[2] Sazonov E and Fontana J M:

Objective monitoring of food intake and ingestive behavior in a free-living environment remains an open problem that has significant implications in study and treatment of obesity and eating disorders. In this paper, a novel wearable sensor system (automatic ingestion monitor, AIM) is presented for objective monitoring of ingestive behavior in free living. The proposed device integrates three sensor modalities that wirelessly interface to a smartphone: a jaw motion sensor, a hand gesture sensor, and an accelerometer. A novel sensor fusion and pattern recognition method was developed for subject-independent food intake recognition. The device and the methodology were validated with data collected from 12 subjects wearing AIM during the course of 24 h in which both the daily activities and the food intake of the subjects were not restricted in any way. Results showed that the system was able to detect food intake with an average accuracy of 89.8%, which suggests that AIM can potentially be used as an instrument to monitor ingestive behavior in free-living individuals.

[3] Ashish Kumar Singh and Neelam Verma:

Today's biosensor technology applied as a most promising tool for analytical purpose. In this contest, we focus on biosensing approaches for determination of L-arginine. L-arginine, which is found in most proteins in our daily diet, has been considered the most potent nutraceutical ever discovered, due to its powerful healing properties, and is being referred to by scientists as the Miracle Molecule. The two most major part of the biosensor that need to be optimized, one is the immobilization of biological components and other is the relevant transducer.so it can be

detected by using such kind of transducer as: pH sensing electrode, ammonia gas sensing, ammonium ion-selective, conductometric and amperometric electrodes are applied. Critical significance of this review is the fact detection is important as a control measure for quality ensure in foods such as beverages, juices and wine. The dietary sources of L-arginine include red meat, nuts, spinach, lentils, whole grains, soy, sea foods and eggs etc. Other more importants in human physiology with the enormous inherent versatility of L-arginine in various pathways, involved with normal growth and maintenance of the body. The estimation of L-arginine is take place through its metabolic products such as urea, ornithine and citrulline. L-arginine is a precursor of several important intermediates so its determination is imperative as a marker for several diseases.

[4] Kong Xiansheng and Sun Jing:

Food spoilage has detrimental effects both in terms of health risks for consumers as well as increased costs for producers. This prompted the development of a system for detecting food spoilage that was compact, low maintenance, and simple to manufacture. The design of said system is discussed in this paper. This paper proposes a design and implementation of food intelligent monitoring system based on pH sensor. The results of our paper show that it is possible to use pH sensor technology and near field communication technology to detect changes in pH to avoid food spoilage.

[5] Syeda Erfana Zohora , A. M. Khan and Arvind K. Srivastava:

In the last few decades there has been a tremendous amount of research on synthetic emotional intelligence related to affective computing that has significantly advanced from the technological point of view that refers to academic studies, systematic learning and developing knowledge and affective technology to a extensive area of real life time systems coupled with their applications. The objective of this paper is to present a general idea on the area of emotional intelligence in affective computing. The overview of the state of the art in emotional intelligence comprises of basic definitions and terminology, a study of current technological scenario. The paper also proposes research activities with a detailed study of ethical issues, challenges with importance on affective computing. Lastly, we present a broad area of applications such as interactive learning emotional systems, modeling emotional agents with an intention of employing these agents in human computer interactions as well as in education.

[6] Julian W.Gardner and Philip N.Bartlett:

The human nose is still the primary 'instrument' used to assess the smell or flavour of various industrial products today, despite considerable and sustained attempts to develop new electronic instrumentation capable of mimicking its remarkable ability. In this paper we review the research effort that has been carried out over the past 25 years or so to create an electronic nose. In doing so, we first provide a definition for the term electronic nose, and then discuss some of the technologies that have been explored in what is essentially an intelligent chemical array sensor system. Finally, we summarize the applications of electronic noses to date and suggest where future applications may lie.

[7] M. Khojastehnazhand, M. Omid and A. Tabatabaeefar:

Grading systems give us many kinds of information such as size, color, shape, defect, and internal quality. Among these color and size are the most important features for accurate classification and/or sorting of citrus such as oranges, lemons and tangerines. Basically, two inspection stages of the system can be identified: external fruit inspection and internal fruit inspection. The former task is accomplished through processing of color images, while internal inspection requires special sensors for moisture, sugar and acid contents. In this paper, an efficient algorithm for grading lemon fruits is developed and implemented in visual basic environment. The system consists of two CCD cameras, two capture cards, an appropriate lighting system, a personal computer and other mechanical parts. The algorithm initially extracts the fruit from the background. The samples of different grades of lemon are situated in front of the cameras and are calibratted off-line. Then information on the HSI color values and estimated volumes of fruits are extracted and saved in a database. By comparing the information during sorting phase with the available information inside the database, the final grade of the passing fruits are determined. This algorithm can be easily adapted for grading and/or inspection of other agricultural products such as cucumber and eggplant.

[8]ALEXANDRA SIFFERLIN:

Food production has been industrialized and its trade and distribution have been globalized, WHO Director-General Dr Margaret Chan said in a statement. “These changes introduce multiple new

opportunities for food to become contaminated with harmful bacteria, viruses, parasites, or chemicals. The WHO notes that foods can contain unsafe viruses and bacteria, which can be contracted from consuming under cooked animal meat, fruit and vegetables contaminated with feces, and toxin-filled shellfish. The agency says governments on the global and national level need to implement measures to protect against food contamination, and respond quickly to food-related outbreaks. Consumers should make sure they practice properly food handing and cooking hygiene.

[9] Ee Lim Tan, Wen Ni Ng, Ranyuan Shao, Brandon D. Pereles and Keat Ghee Ong:

the fabrication of a wireless, passive sensor based on an inductive-capacitive resonant circuit, and its application for in situ monitoring of the quality of dry, packaged food such as cereals, and fried and baked snacks. The sensor is made of a planar inductor and capacitor printed on a paper substrate. To monitor food quality, the sensor is embedded inside the food package by adhering it to the package sinner wall; its response is remotely detected through a coil connected to a sensor reader. As food quality degrades due to increasing humidity inside the package, the paper substrate absorbs water vapor, changing the capacitor's capacitance and the sensor's resonant frequency. Therefore, the taste quality of the packaged food can be indirectly determined by measuring the change in the sensor's resonant frequency. The novelty of this sensor technology is its wireless and passive nature, which allows in situ determination of food quality. In addition, the simple fabrication process and inexpensive sensor

material ensure a low sensor cost, thus making this technology economically viable.

[10] Arghya Sett , Suradip Das , Pragya Sharma and Utpal Bora:

Biosensors have been developed using various types of sensing elements like biomacromolecules (viz. enzymes, antibodies, receptors, nucleic acids, etc.) organelles, tissues, intact cells of both microorganisms and higher organisms. A recent trend is the emergence of aptamers as sensing elements that has the potential to replace all the above ligands. This is possible due to the unique features of aptamers (sensitivity, specificity, reusability, stability, non-immunogenic- ity), which can be easily exploited in biosensor technology. Aptasensors are thus basically biosensors based on aptamers as ligand molecules. Here we review the various applications of aptasensors in health (specifically in diagnostics), food industry and environmental monitoring.

CHAPTER-3

ANALYSIS

3.1. INTRODUCTION:

This phase is important because in here each component which is used is analyzed. By which, the cost can be analyzed properly. And also this phase discusses about the existing system and its limitations and how the proposed system eliminates the limitations.

3.2. HARDWARE AND SOFTWARE REQUIREMENT ANALYSIS:

3.2.1. HARDWARE REQUIREMENT ANALYSIS:

The main aim of the project is to detect spoiled food from arduino and nodemcu modules. The components required are:

S No.	Component	Description
1	Arduino uno	reads the sensor data
2	Nodemcu	used for Wi-Fi connection
3	MQ2 sensor	reads Methane, Butane, LPG, smoke
4	MQ3 sensor	reads Alcohol, Ethanol, smoke
5	MQ4 sensor	reads Methane, CNG Gas
6	MQ135 sensor	reads air quality around food
7	DHT11 sensor	To read temperature and humidity
8	Light dependent register	To read light

Table 3.1 components

3.2.1.1 ARDUINO UNO:

Arduino Uno is a microcontroller board based on the ATmega328P. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.. You can tinker with your Uno without worrying too much about doing something wrong, worst case scenario you can replace the chip for a few dollars and start over again.

"Uno" means one in Italian and was chosen to mark the release of Arduino Software (IDE) 1.0. The Uno board and version 1.0 of Arduino Software (IDE) were the reference versions of Arduino, now evolved to newer releases. The Uno board is the first in a series of USB Arduino boards, and the reference model for the Arduino platform; for an extensive list of current, past or outdated boards see the Arduino index of boards.

Fig 3.1 Arduino uno

3.2.1.2 NODEMCU:

NodeMCU is an open source firmware for which open source prototyping board designs are available. The name "NodeMCU" combines "node" and "MCU" (micro-controller unit).[8] The term "NodeMCU" strictly speaking refers to the firmware rather than the associated development kits. Both the firmware and prototyping board designs are open source.[8]

The firmware uses the Lua scripting language. The firmware is based on the eLua project, and built on the Espressif Non-OS SDK for ESP8266. It uses many open source projects, such as lua-cjson[9] and SPIFFS.[10] Due to resource constraints, users need to select the modules relevant for their project and build a firmware tailored to their needs. Support for the 32-bit ESP32 has also been implemented.

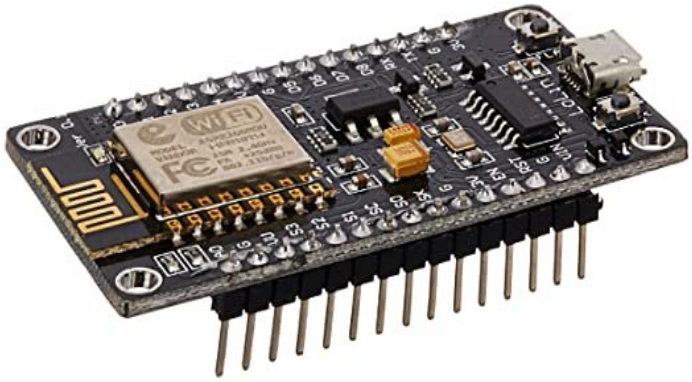

Fig 3.2 Nodemcu

3.2.1.3 MQ2 SENSOR:

MQ2 is one of the commonly used gas sensors in MQ sensor series. It is a Metal Oxide Semiconductor (MOS) type Gas Sensor also known as **Chemiresistors** as the detection is based upon change of resistance of the sensing material when the Gas comes in contact with the material.

MQ2 Gas sensor works on 5V DC and draws around 800mW. It can detect LPG, Smoke, Alcohol, Propane, Hydrogen, Methane and Carbon Monoxide concentrations anywhere from 200 to 10000ppm.

Fig 3.3 MQ2 sensor

3.2.1.4 MQ3 SENSOR:

MQ3 is one of the most commonly used sensors in the MQ sensor series. It is a Metal Oxide Semiconductor (MOS) type of sensor. Metal oxide sensors are also known as Chemiresistors, because sensing is based on the change of resistance of the sensing material when exposed to alcohol. So by placing it in a simple voltage divider network, alcohol concentrations can be detected.

Fig 3.4 MQ3 sensor

3.2.1.5 MQ4 SENSOR:

This semiconductor gas sensor detects the presence of methane (CNG) gas at concentrations from 300 ppm to 10,000 ppm, a range suitable for detecting gas leaks. The sensor's simple analog voltage interface requires only one analog input pin from your microcontroller.

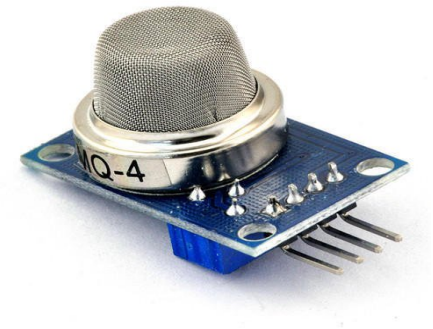

Fig 3.5 MQ4 sensor

3.2.1.6 MQ135 SENSOR:

Air quality sensor for detecting a wide range of gases, including NH3, NOx, alcohol, benzene, smoke and CO2. Ideal for use in office or factory. MQ135 gas sensor has high sensitivity to Ammonia, Sulfide and Benze steam, also sensitive to smoke and other harmful gases. It is with low cost and particularly suitable for Air quality monitoring application.

Fig 3.6 MQ135 sensor

3.2.1.7 DHT11 SENSOR:

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 (no analog input pins needed).

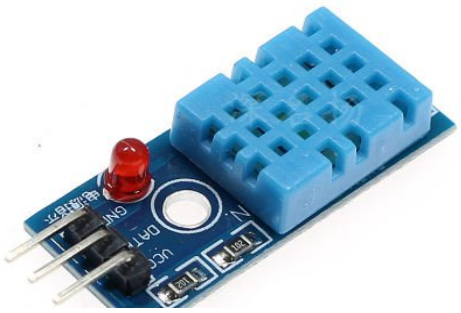

Fig 3.7 DHT11 sensor

3.2.1.8 LIGHT DEPENDENT RESISTOR:

A Light Sensor generates an output signal indicating the intensity of light by measuring the radiant energy that exists in a very narrow range of frequencies basically called “light”, and which ranges in frequency from “Infra-red” to “Visible” up to “Ultraviolet” light spectrum.

Fig 3.8 LDR

3.2.2. SOFTWARE REQUIREMENT ANALYSIS:

- Arduino IDE is required. It is open-source so no cost involoved
    - The software also verifies the code and uploads the code in arduino uno.
    - The software has many inbuilt libraries which makes the developer easy to use and we have also used modules like “DHT-11”, “MQunifiedsensor”, etc...
    - The main objective of software is to read, process the data and send the output
- Operating system: Linux, Mac or Windows
- Web browser (Chrome is best suited).
    - For using thingspeak.com
    - For using ifttt.com
    - **o** For using freeboard.io

3.3. EXISTING SYSTEM:

The block diagram of the design is as shown below. It consists of power supply unit, Arduino microcontroller, Wi-Fi modem, Gas sensor, LDR, DTH11 sensor, LCD.

Hardware implementation deals in drawing the schematic on the plane paper according to the application, testing the schematic design over the breadboard using the various IC's to find if the design meets the objective, carrying out the PCB layout of the schematic tested on breadboard, finally preparing the board and testing the designed hardware.

The system is very limited and it is very easy to implement.

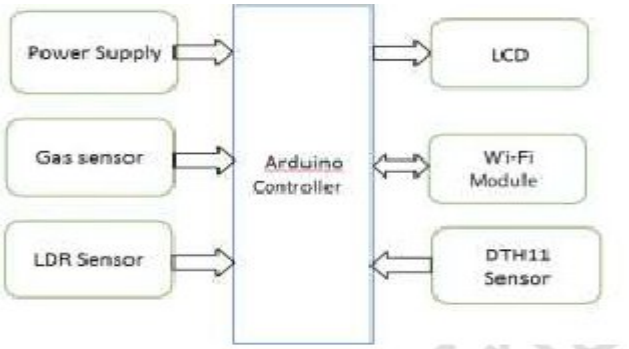

Fig -1: Block Diagram

3.9 Block diagram of existing system

There is no concept of cloud in the existing system. And the sensors used is only one that is MQ3 sensor. The greatest disadvantage of this system is it doesnot detect the food items which doesnot release the alcohol gas into air. This disadvantage has been overcome by the proposed system.

3.4. PROPOSED SYSTEM:

We have used the same existing system with little modifications. In order to detect all kinds of food spoilage, we are using a series of MQ sensors. In the existing system, they have used only MQ2 sensor which detects Methane, Butane, LPG, smoke. But, in the proposed system we are using MQ-3, MQ-4, MQ-135.

System also detects the air quality surrounded by the food using MQ-135 sensor. By this sensor, we can predict whether the food is spoiled or not and we can send an alert message saying that the food about to be spoiled.

System also sends email notification, and also updates the reading of the sensor on server so that the customer can see review anytime.

3.5. APPROACH USED:

Sensors reads the values from the food and environment. A certain threshold is maintained for each sensor and if the sensor reads the value above the threshold value, then an alert message is send to the customer.

Firstly, the system always check the air quality, temperature, humidity and light, if there is any unbalance, then the system sends and alert message saying that the air quality around the food is not good and the food is about to spoil.

If the customer neglects the alert message, then after a certain time gases will be released from the spoiled food and the corresponding MQ sensor detects it and informs the customer that the food is spoiled.

The system also reads the temperature and humidity constantly.

CHAPTER-4

DESIGN

4.1. DESIGN:

The design is very simple. The arduino has three set of pins namely analog pins, power pins and digital pins. Analog pins are six in total namely A0, A1, A2, A3, A4, A5. Digital pins are fourteen in total. They are: 0 to 13. Power pins are six in total. They are: IOREF, RESET, 3.3v, 5v, GND and GND. Look at the figure for more understanding.

The half breadboard can be used to connection sensors and arduino by using jumper cables. Each row in the main section (i.e center) is denoted by capital alphabet and each column is denoted by number. In half breadboard, there are ten rows divided to two sets (A to E and F to J) and 30 columns (1 to 30) .

Nodemcu module is like the heart of this project. This module connects to WiFi. In this project, two pins are mainly used that is D1 and D2.

4.2 ARDUINO CONNECTIONS:

Look at the following table for the arduino connections.

S No.	Component	Component- pins	Arduino	Breadboard
1	MQ2	A0	A0	
		GND		GND
		VCC		5v
2	MQ3	A0	A1	
		GND		GND
		VCC		5v
3	MQ135	A0	A3	
		GND		GND

		VCC		5v
4	DHT11	DATA	D8	
		GND		GND
		VCC		5v
5	LDR			H22 — H24
6	Jumper			+ — J22
7	Jumper			i28 — minus
8	Resistor 1k ohm			H24 — H28
9	Jumper		A4	G24

Table 4.1 Arduino connections

4.3 NODEMCU CONNECTIONS:

Insert the nodemcu module on the breadboard from B1 to B15 and i1 to i15. Look at the following table for connections:

S No.	Component	Component- pins	Arduino	Breadboard
1	Nodemcu	D1	D1	A14
		D2	D0	A13

Table 4.2 Nodemcu connections

The D0 and D1 pins in the arduino uno acts as RX and TX pins respectively. Receiving (RX) and Transmitting (TX) pins of Arduino used for Serial communication.

The D1 and D2 pins of nodemcu acts as RX and TX pins respectively and these pins are used for serial communication.

The connections of arduino and nodemcu should be well established. Use solderless breadboard and use jumbers with pins attached to it. By this, the error will b reduced. Since the communication between the arduino and nodemcu is serial communication, sometimes due to delays the nodemcu module can't read the values from arduino properly. There might be some errors.

And also for updating the data into the cloud and sending the emails, the delays should be added in the nodemcu project. By this, the updation will be slow and immediate responses will be not possible.

The design is very important in this system because if one connection is misplaced, then whole system will not work.

If there is any voltage leakage, then the arduino will not work properly. The baud rate for arduino and nodemcu module is important. The nodemcu baud rate should be higher than arduino because the serial monitor of nodemcu displays more bits compared to arduino's serial monitor.

Some nodemcu's won't work on baud rate higher than 1152000. If the baud rate is set higher than that, then the garbage values will be shown on the serial monitor.

By this, moderate baud rates should be fixed and moderate difference should be maintained between arduino and nodemcu.

4.4 THINGSPEAK DESIGN:

Thingspeak.com is a website used to store the collected sensor data in the cloud. It is an IoT analytics platform service that allows you to aggregate, visualize and analyze live data streams in the cloud. ThingSpeak provides instant visualizations of data posted by your devices to ThingSpeak. With the ability to execute MATLAB code in ThingSpeak you can perform online analysis and processing of the data as it comes in. ThingSpeak is often used for prototyping and proof of concept IoT systems that require analytics.

ThingSpeak is IoT Cloud platform where you can send sensor data to the cloud. You can also analyze and visualize your data with MATLAB or other software, including making your own applications.

The ThingSpeak service is operated by MathWorks. In order to sign up for ThingSpeak, you must create a new MathWorks Account or log in to your existing MathWorks Account.

ThingSpeak is free for small non-commercial projects.

ThingSpeak includes a Web Service (REST API) that lets you collect and store sensor data in the cloud and develop Internet of Things applications. It works with Arduino, Raspberry Pi and MATLAB (premade libraries and APIs exists) But it should work with all kind of Programming Languages, since it uses a REST API and HTTP.

ThingSpeak has many limitations and it costs money if you want to use it for real and professional applications.

Note: For more details, visit chapter-6

4.5 IFTTT DESIGN:

If This Then That is a service that allows a user to program a response to events in the world.

ifttt.com is a website used to create applets. This website is useful for sending mails to the user. Applet connects the nodemcu module and email.

IFTTT helps you connect all of your different apps and devices. When you sign up for a free account, you can enable your apps and devices to work together to do specific things they couldn't do otherwise. For example, you can back up your Instagram photos to Dropbox, have your lights turn on when you enter your home, or automatically remind a Slack channel about a meeting.

IFTTT has partnerships with different service providers that supply event notifications to IFTTT and execute commands that implement the responses. Some event and command interfaces are simply public APIs.

The programs, called applets, are simple and created graphically.

User can create programs and otherwise control IFTTT with a web interface or iOS or Android application.

- IFTTT can automate web-application tasks, such as posting the same content on several social networks.
- Marketing professionals can use IFTTT to track mentions of companies in RSS feeds.
- IFTTT also is used in home automation, for instance switching on a light when detecting motion in a room (with associated compliant devices).

4.6 FREEBOARD DESIGN:

Freeboard.io is an open source dashboard project with optional hosted subscriptions, easy to integrate with various data sources, ready for production and very well designed.

Allows users to create and customize real-time interactive dashboards

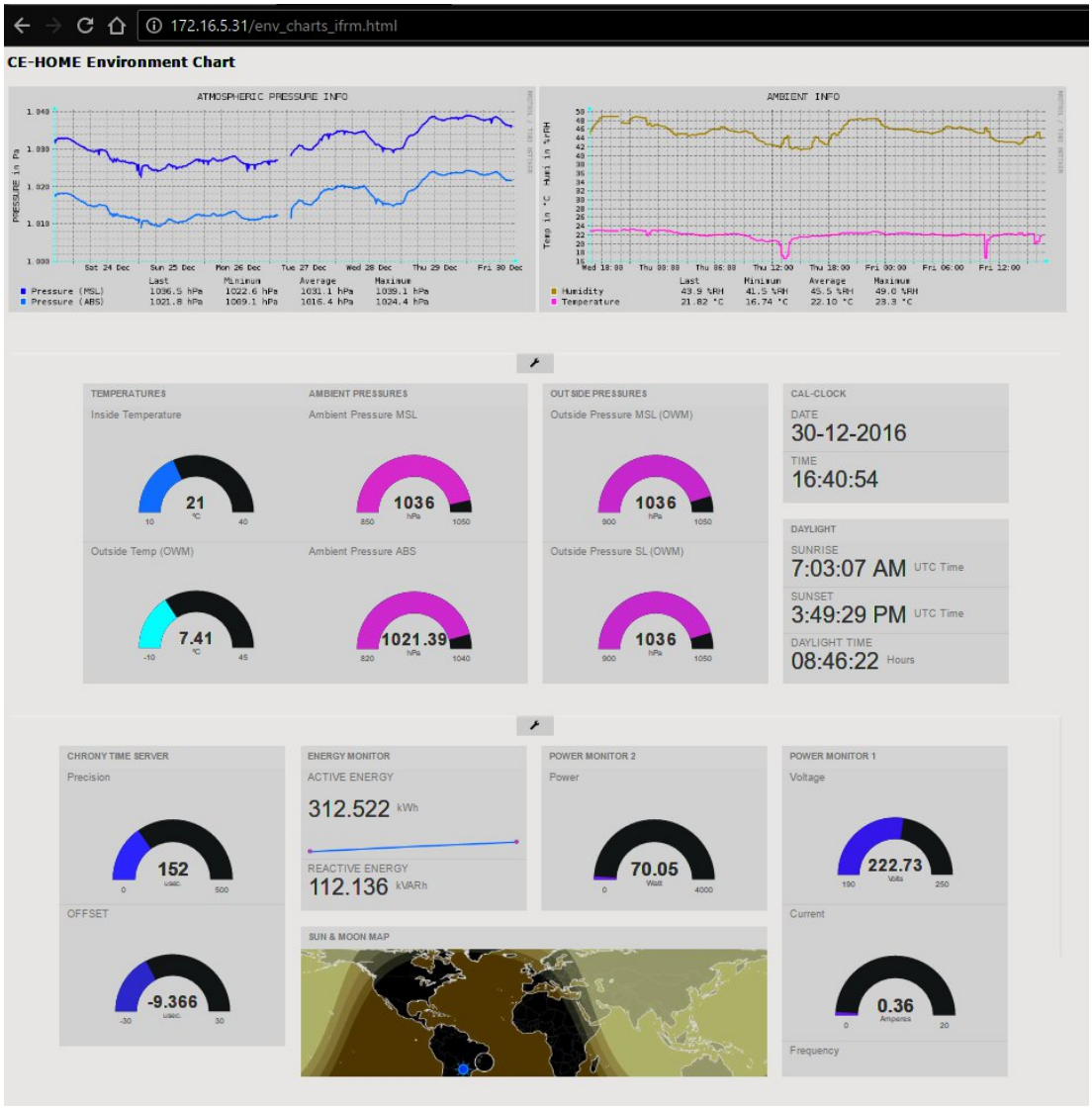

Fig 4.1 Freeboard design

Note: For more details, visit chapter-6

4.7 BLOCK DIAGRAM:

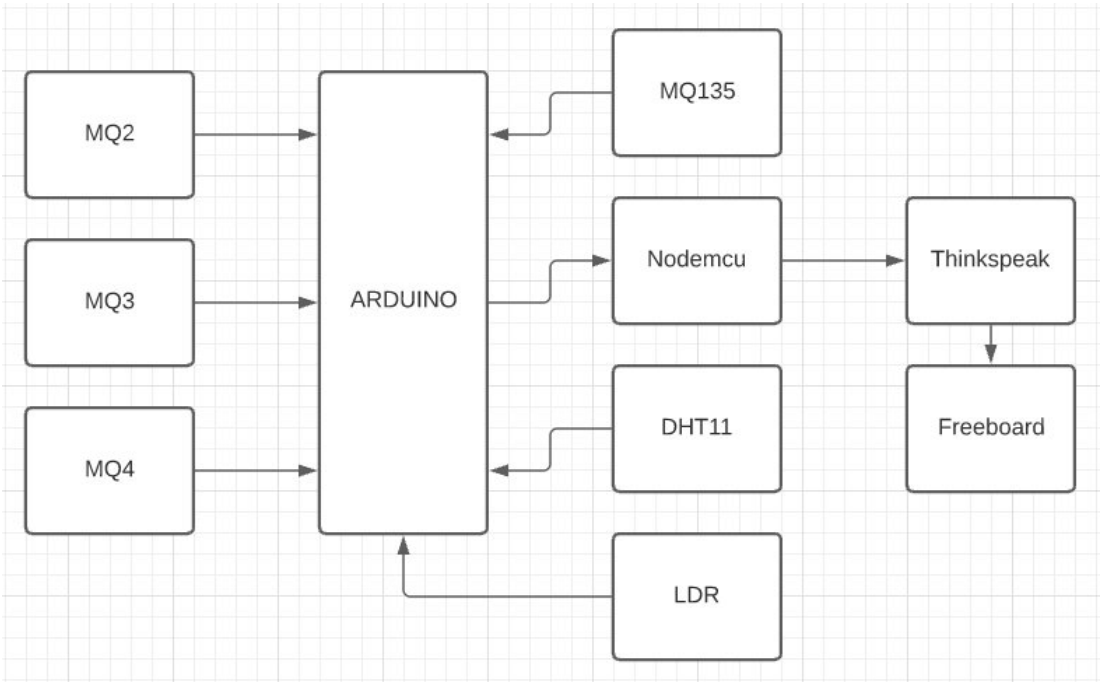

Fig 4.2 block diagram of proposed system

4.8 CIRCUIT DIAGRAM:

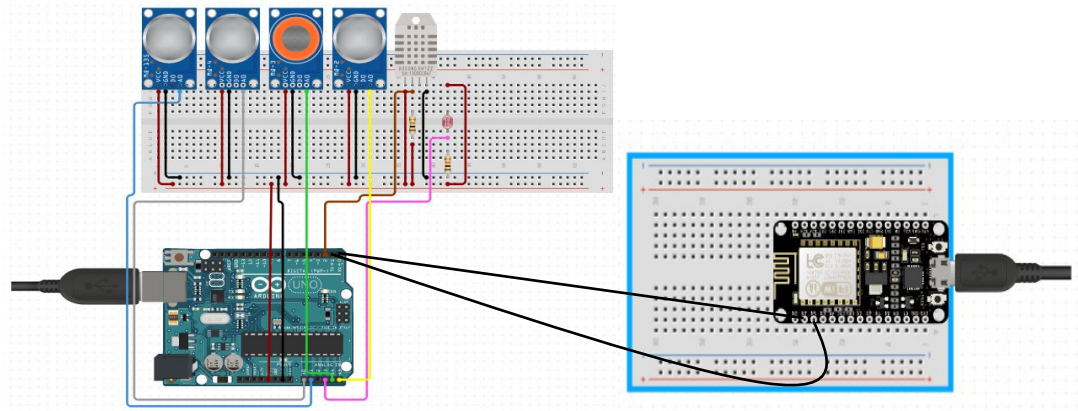

Fig 4.3 Circuit diagram of proposed system

CHAPTER 5

IMPLEMENTATION

There is no standard implemented algorithm. The task of arduino uno is very simple. It connects all the sensors and stores the data in its memory.

5.1 Arduino code snippets:

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

The library used for reading temperature and humidity via DHT11 sensor

```
2. #define MQ2_PIN A0

3. #define MQ3_PIN A1

4. #define MQ4_PIN A2

5. #define MQ135_PIN A3

6. #define LDR A4

7. #define DHT11_PIN 8

8. #define DHTTYPE DHT11
```

Line-2 to Line-7 defines the constants says which sensor maps to arduino pins

Line-8 defines the type of DHT sensor

```
9. float MQ2_read=0, MQ3_read=0, MQ4_read=0, MQ135_read=0;

10. float temperature=0, humidity=0;

11. float LDRReading = 0;

12. DHT dht(DHT11_PIN, DHTTYPE);
```

Line-9 to Line-11 declares the variables which are used to read the sensor data. All variables are initialized to zero so that it will display zero if no data is collected.

Line-12 initializes the dht object and this object is used to read the temperature and humidity values.

```
13. void setup() {

  14. Serial.begin(9600);

  15. dht.begin();

  16. pinMode(MQ2_PIN, INPUT);

  17. pinMode(MQ3_PIN, INPUT);

  18. pinMode(MQ4_PIN, INPUT);

  19. pinMode(MQ135_PIN, INPUT);

20. }
```

The setup() function is called when a sketch starts. Use it to initialize variables, pin modes, start using libraries, etc. The setup() function will only run once, after each powerup or reset of the Arduino board.

Line-14 passes the value 9600 to the speed parameter. This tells the Arduino to get ready to exchange messages with the Serial Monitor at a data rate of 9600 bits per second.

Line-15 enables or begins the DHT11 sensor

Line-16 to Line-19 configures the specified pin to behave as input.

```
21. void loop() {
  22. MQ2_read = analogRead(MQ2_PIN);
  23. MQ3_read = analogRead(MQ3_PIN);
  24. MQ4_read = analogRead(MQ4_PIN);
  25. MQ135_read = analogRead(MQ135_PIN);
  26. LDRReading = analogRead(LDR);


  27. temperature = dht.readTemperature();
  28. humidity = dht.readHumidity();
```

The loop() function (Line-21 to Line-42) does precisely what its name suggests, and loops consecutively, allowing your program to change and respond.

Line-22 to Line-26 reads the analog values from the corresponding sensors and stores accordingly.

Line-27 to Line-28 reads the temperature and humidity values and the dht object as predefined functions to do the task.

```
29. Serial.print(LDRReading);

30. Serial.print(" ");

31. Serial.print(MQ2_read);

32. Serial.print(" ");

33. Serial.print(MQ3_read);

34. Serial.print(" ");

35. Serial.print(MQ4_read);

36. Serial.print(" ");

37. Serial.print(MQ135_read);

38. Serial.print(" ");

39. Serial.print(temperature);

40. Serial.print(" ");

41. Serial.println(humidity);

42. delay(40000);
```

Line-29 to Line-41 displays the recorded values on the serial monitor in a single line.

Line-42 pauses the execution for 40 seconds.

5.2 Nodemcu code snippets:

The stored sensor data is send to nodemcu module via serial communication. The nodemcu is like the heart of the project. Nodemcu's task is to connect to Wifi, send data to cloud that is thingspeak and sending emails.

```
1. #include<SoftwareSerial.h>

2. #include <ESP8266WiFi.h>

3. #include <ThingSpeak.h>
```

The SoftwareSerial library (Line-1) has been developed to allow serial communication on other digital pins of the Arduino.

Line-2 library is used for nodemcu module. It is various features which makes our task easy.

Line-3 library is used for thingspeak connection and data uploading to cloud.

```
4. SoftwareSerial mySerial(D1, D2);
```

Line-4 initializes softwareSerial object and D1, D2 denotes the RX and TX pins of the nodemcu module.

```
5. char* ssid = "*******";

6. char* password = "*******";
```

Line-5 denotes the WiFi SSID and Line-6 denotes the wifi password.

```
7. const char* host = "maker.ifttt.com";
```

Line-7 defines the website name which is used to create an applet for send emails.

```
8. WiFiClient  client;

9. const int httpPort = 80;
```

Line-8 creates a client that can connect to a specified internet IP address and port

Line-9 denotes the standard http port number

```
10. unsigned long myChannelNumber = 1400396;

11. const char* myWriteAPIKey = "08E4IMSBJX21ZBPP";
```

Line-10 and Line-11 denotes the channel number and write api key which was created (in thingspeak.com)

```
12. String message, LDR, MQ2, MQ3, MQ4, MQ135, temperature,
        humidity;

13. int count_spoiled=0,count_air_quality=0, count=0;
```

Line-12 and Line-13 are the variables which are used to read the sensor data and counters to control the flow of emails

```
12. void setup() {
  13. Serial.begin(115200);
  14. mySerial.begin(9600);
  15. ThingSpeak.begin(client);
```

Line-13 enables the serial monitor for nodemcu module.

Line-14 enables and begins connection to arduino uno with the speed parameter as 9600 bitsPerSecond.

Line-15 enables and begins the thingspeak connection.

```
  16. WiFi.mode(WIFI_STA);
  17. WiFi.begin(ssid,password);
  18. while(WiFi.status()!=WL_CONNECTED) {
  19. Serial.print(".");
  20. delay(500);
  21. }
  22. Serial.println("");
  23. Serial.print("IP Address: ");
  24. Serial.println(WiFi.localIP());
25. }
```

Line-16 denotes the mode of Wifi which is station mode

Line-17 starts the connection with respect to the given wifi ssid and wifi password.

Line-18 to Line-21 loop terminates until the wifi is connected to nodemcu.

Line-22 to Line-25 prints the IP address of the WiFi connected.

```
26. void loop() {
  27. message = mySerial.readStringUntil('\n');
  28. LDR = getValue(message,' ',0);
  29. MQ2 = getValue(message,' ',1);
  30. MQ3 = getValue(message,' ',2);
  31. MQ4 = getValue(message,' ',3);
  32. MQ135 = getValue(message,' ',4);
  33. temperature = getValue(message,' ',5);
  34. humidity = getValue(message,' ',6);
  35. Serial.println(message);
  36. thingspeak(temperature, humidity, MQ2, MQ3, MQ4, MQ135, LDR);
  37. sending_email();
  38. delay(40000); }
```

Line-27 reads the sensor data send from arduino in an single line.

Line-28 to Line-34 getValue function splits the message and the corresponding sensor data is being stored into variables accordingly.

Line-36 calls the thingspeak function which sends the data to cloud.

Line-37 calls the sending_email function which sends the mails to the user.

```
39. String getValue(String data, char sep, int index) {
 40. int found = 0;
 41. int strIndex[] = {0, -1};
 42. int maxIndex = data.length()-1;

 43. for(int i=0; i<=maxIndex && found<=index; i++){
  44. if(data.charAt(i)==sep || i==maxIndex){
      45. found++;
      46. strIndex[0] = strIndex[1]+1;
      47. strIndex[1] = (i == maxIndex) ? i+1 : i;
   48. }
 49. }
 50. return found>index ? data.substring(strIndex[0], strIndex[1]) : "";
51. }
```

Line-39 to Line-51 takes data, sep and index as parameters and returns the string which has corresponding index word. So, by this function we can split the data by seperator which is provided.

```
52. void thingspeak(String temperature, String humidity, String MQ2, String MQ3, String MQ4, String MQ135, String LDR) {
53. ThingSpeak.setField(1, temperature);
54. ThingSpeak.setField(2, humidity);
55. ThingSpeak.setField(3, MQ2);
56. ThingSpeak.setField(4, MQ3);
57. ThingSpeak.setField(5, MQ4);
58. ThingSpeak.setField(6, MQ135);
59. ThingSpeak.setField(7, LDR);

60. int x = ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);
61. if(x == 200){
62. Serial.println("Channel update successful."); }
63. else {
64. Serial.println("Problem updating channel. HTTP error code " + String(x)); } }
```

Line-53 to Line-59 sets which field maps to sensor variable.

Line-60 writes the sensor data to corresponding fields and returns value

Line-61 to Line-64 checks whether the data is send to cloud or not. If all is properly done, Line-60 returns 200 otherwise returns -210

```
65. void sending_email() {
66. if(!client.connect(host, httpPort)) {
67.     Serial.println("email connection failed");
68.     return;    }


69. if(count == 10) {
70.     count_spoiled = 0;
71.     count_air_quality = 0;
72.     count=0;         }


73. if(count_spoiled!=3 && MQ2.toInt() >= 470 && MQ3.toInt() >= 345 &&
        MQ4.toInt() >= 280)          {
74.     String url1 =
"/trigger/food_spoiled/with/key/oOAEkILKPPinkD1S3IJ5I7FCYNdDNmWFzl
gsNFoqBD";
75.     Serial.print("Requesting url: ");
```

```
76.    Serial.println(url1);
77.    client.print(String("GET ") + url1 + " HTTP/1.1\r\n" +
                    "Host: " + host + "\r\n" +
                    "Connection: close\r\n\r\n");
78.    count_spoiled += 1;
79.    count+=1;    }
80. else {
81.    Serial.println("email not sent because food is not spoiled yet."); }
82. delay(5000);
83. if(count_air_quality!=3 && LDR.toInt() > 300 && MQ135.toInt() > 270)
       {
84.  String url =
"/trigger/air_quality_and_light_around_the_food_is_not_good-
Move_your_food_to_a_darker_place/with/key/oOAEkILKPPinkD1S3IJ5I7FC
YNdDNmWFzlgsNFoqBD";
85.    Serial.print("Requesting url: ");
86.    Serial.println(url);
87.    client.print(String("GET ") + url + " HTTP/1.1\r\n" +
                    "Host: " + host + "\r\n" +
                    "Connection: close\r\n\r\n");
88.    count_air_quality += 1;
```

```
89.     count+=1; }}
90. else {
91.     Serial.println("email not sent because air quality is good.");          }
92. count+=1;         }
```

Line-66 to Line-68 is used to check whether the email service is connected or not.

Line-69 to Line-78 restricts the emails. Otherwise the emails will be send continuously.

Line-73 to Line-79 is responsible for sending emails when the food is spoiled.

line-83 to line-89 is responsible for sending emails when the air quality around the food is not good.

The urls for triggering the emails are present in the ifttt.com website.

CHAPTER-6

EXECUTION PROCEDURE AND TESTING:

For executing the code, First an account should be created on thingspeak, ifttt.com and freeboard.io. Steps to create each of these is mentioned below.

In order to run the code, arduino IDE is required. The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. This software can be used with any Arduino board. Web version is also available. Steps to install arduino IDE into local device.

6.1 INSTALLATION GUIDE FOR ARDUINO IDE:

1. goto https://www.arduino.cc/en/software and download the file as per your requirement.

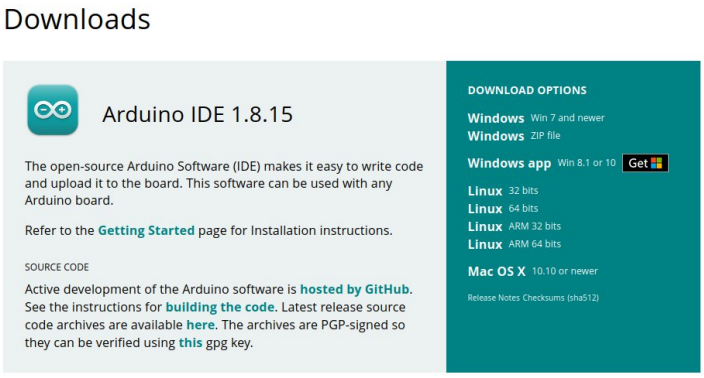

Fig 6.1 download options

2. For windows users, simply run the file downloaded from the software page and DONE.

   Download stable version that is 1.8.15 and same version is being used for this project.

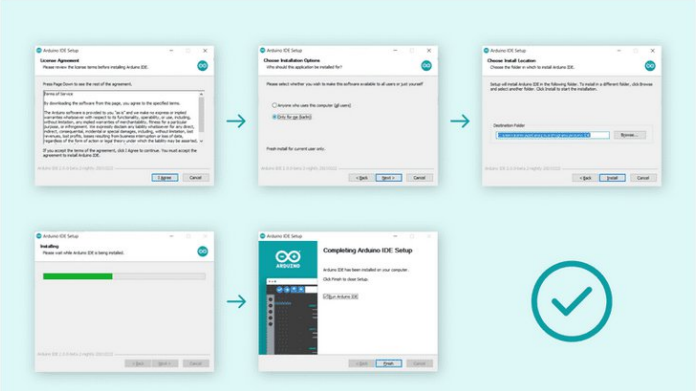

Fig 6.2 windows installation

3. For mac users, simply copy the downloaded file into your application folder and DONE.

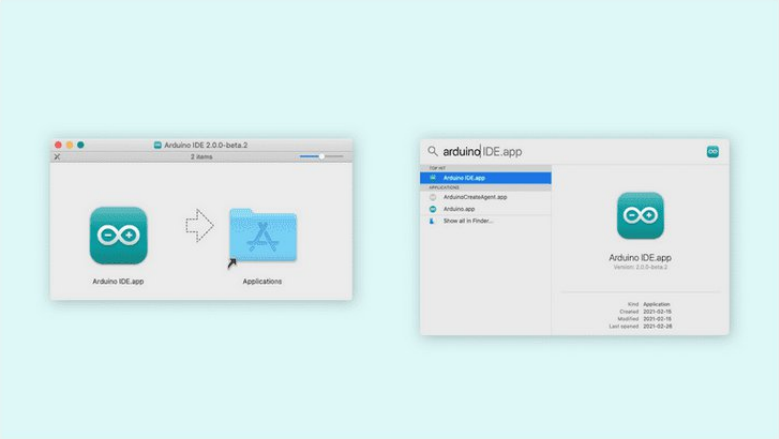

Fig 6.3 Mac installation

4. For linux users, simply extract the downloaded file to a suitable folder. Now, navigate to the folder, right click on it and select run in terminal. Finally, to launch the editor type ./arduino-ide and DONE.

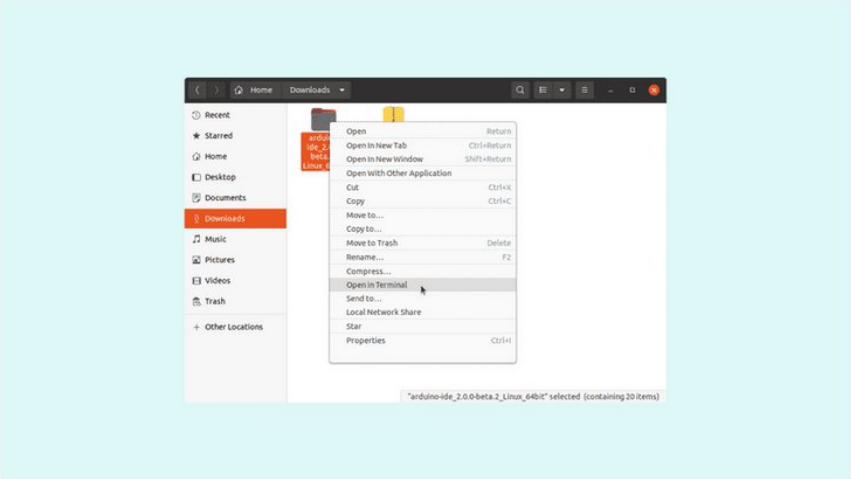

Fig 6.4 Linux installation

After installing arduino ide, launch the arduino IDE in your local system. Create two new projects and name them as arduino-DATASEND and nodemcu-RECEIVED.

1. Open arduino-DATASEND project and copy the arduino code which is written and paste it.
2. Goto Tools and click on boards. Inside boards, select arduino uno. Now, inside the Tools, goto port and choose the port from which the arduino uno board is connected.
3. Now, goto sketch and click on verify/compile. The code will be verified and compiled and displays if there are any errors.
4. Finally, click on upload to load the code into the arduino board

On the other hand,

1. goto nodemcu-RECEIVED project and copy the nodemcu code and paste it here
2. Goto Tools and click on boards. Inside boards, select nodemcu 1.0 board. Now, inside the Tools, goto port and choose the port from which the nodemcu board is connected.

3. Now, goto sketch and click on verify/compile. The code will be verified and compiled and displays if there are any errors.
4. Finally, click on upload to load the code into the nodemcu board

Note:

- If the nodemcu board is not available, then goto board manager and download it.
- Before uploading the code, remove TX and RX connections and connect it after both codes are being uploaded.
- Some of the libraries are not pre-installed

6.2 THINGSPEAK GUIDE:

Thingspeak is the website used to store the sensor data into cloud. Steps to create a free account and how it being used.

1. Goto https://thingspeak.com/ and click on sign in. If account already exists, login with your credentials. Or create a new account.

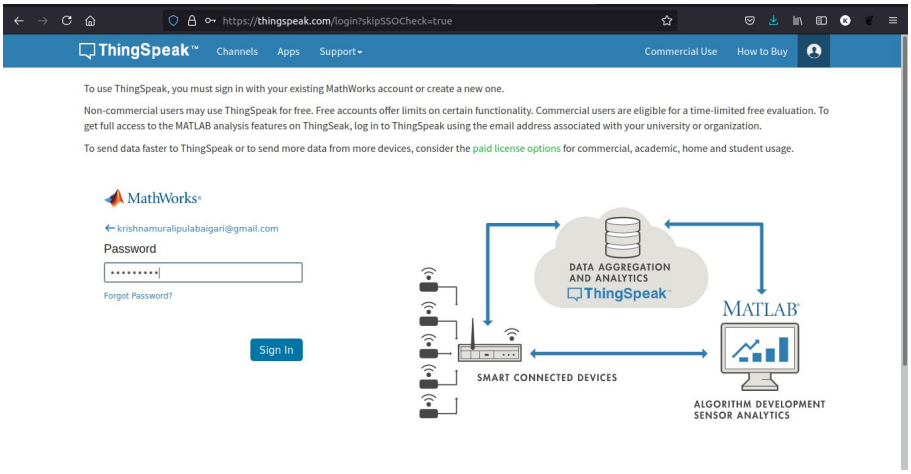

Fig 6.5 thingspeak account creation

2. Goto channels and create a new channel. Type the details as mentioned below:
    a. Name as Food Spoilage Detection

b. Field 1 as temperature
c. Field 2 as humidity
d. Field 3 as MQ2
e. Field 4 as MQ3
f. Field 5 as MQ4
g. Field 6 as MQ135
h. Field 7 as LDR

3. Click on save channel.

6.3 IFTTT guide:

IFTTT means IF This Then This. It is a website which creates applets and used to send emails from nodemcu to user. Steps to create applet in ifttt.com

1. goto ifttt.com and click on create.
2. click on add on the If this label. Search for webhooks service.
3. Click on request a web request
4. Type event name as food spoiled
5. Click on add button on the Then This label. Search for email service.
6. Click on send me an email.
7. Finally, Click on create action
8. Now, goto explore and search for webhooks. The applet created will be shown. On the top bar, click on documentation.
9. You can see url and key of your applet

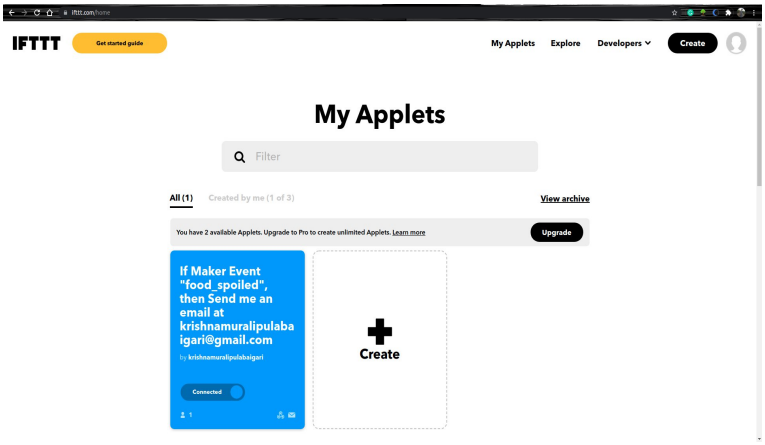

Fig. 6.6 ifttt applets

Two applets were created. One for sending email if the air quality is not good and other for sending if the food is spoiled.

6.4 FREEBOARD GUIDE:

It is a visualization website and best suited for IoT devices. The latest data which is recorded in the cloud will be displayed in an neat manner. Steps to create a free account on freeboard.

1. goto freeboard.io and click on sign-in
2. If account already exists, then login with your credentials. Or create a new account which is very easy.
3. Now, enter the name of the project and click on create new.
4. On the left top corner, below datasources label. Click on add.
5. As we are using thingspeak, select type as JSON.
6. Type url as api.thingspeak.com/channels/CHANNEL_ID/feeds/last
7. Click on add pane.

8. A pane will be created, click on plus icon and select the type as Gauge. Select datasource and add field at the end.

Fig 6.7 Freeboard

CHAPTER - 7

RESULTS AND PERFORMANCE EVALUATION:

7.1. RESULTS:

Nodemcu serial monitor displays the sensor data, whether the data is uploaded to cloud or not, and whether the mails are being sent or not.

Thingspeak gets the sensor data for every fourty seconds and each sensor value is displayed in a graph with timeline on x-axis and values of sensor on y-axis. If the code works properly, the graph displays the value of sensor. By which we can assume that the sensor is being uploaded to the cloud properly.

The latest recorded values can be used via freeboard.io. It is a simple visualization website with rich features. The user can check the latest values by viewing this website. This website has mobile compatibility.

2. NODEMCU SERIAL MONITOR:

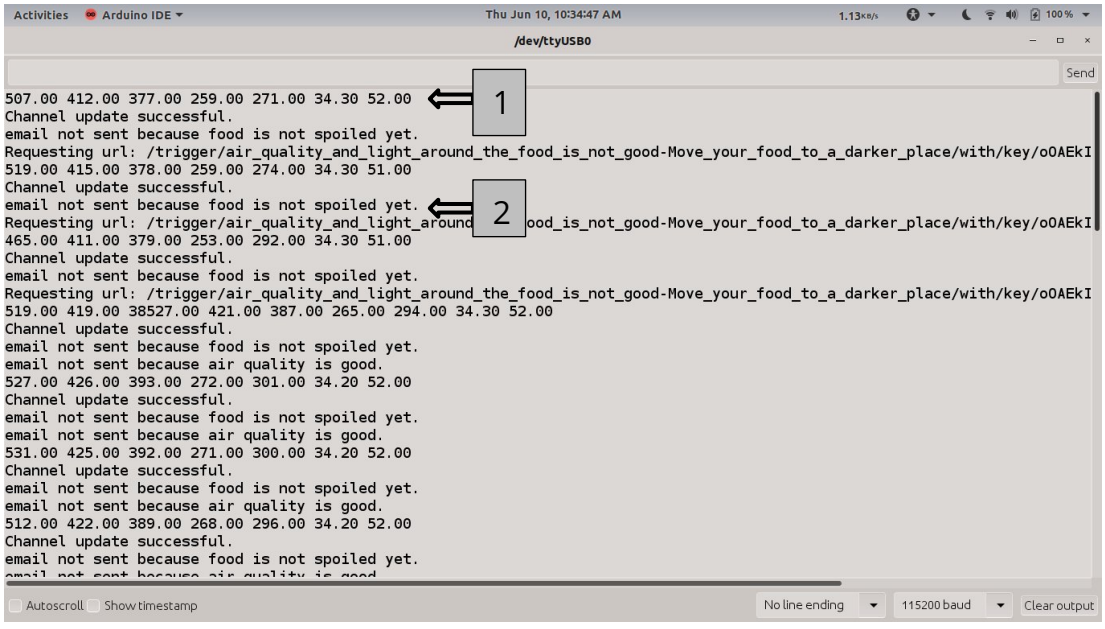

Fig 7.1 Nodemcu serial monitor output-1

The sensor data which is send from arduino is being displayed in statement 1[see above image].

The air quality and light around the food was not good. So, the device triggers the email. After trigging the first email, the food was placed [for experimentation]. Due to slow updates, the device immediately will not respond saying that the food is spoiled. From the image, the email corresponds to food spoilage is not triggered yet.

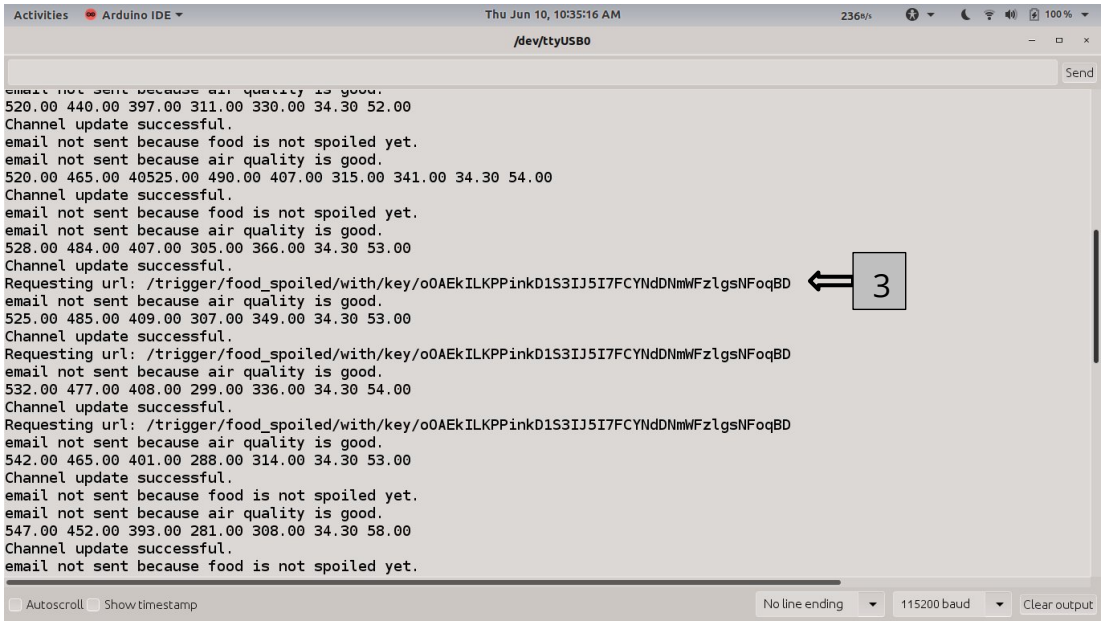

Fig 7.2 Nodemcu serial monitor output-2

After four to five updates, the email corresponding to food spoilage is triggered (statement 3). After the first email sent, the food was removed. Even though, the email was triggered because of slow updates.

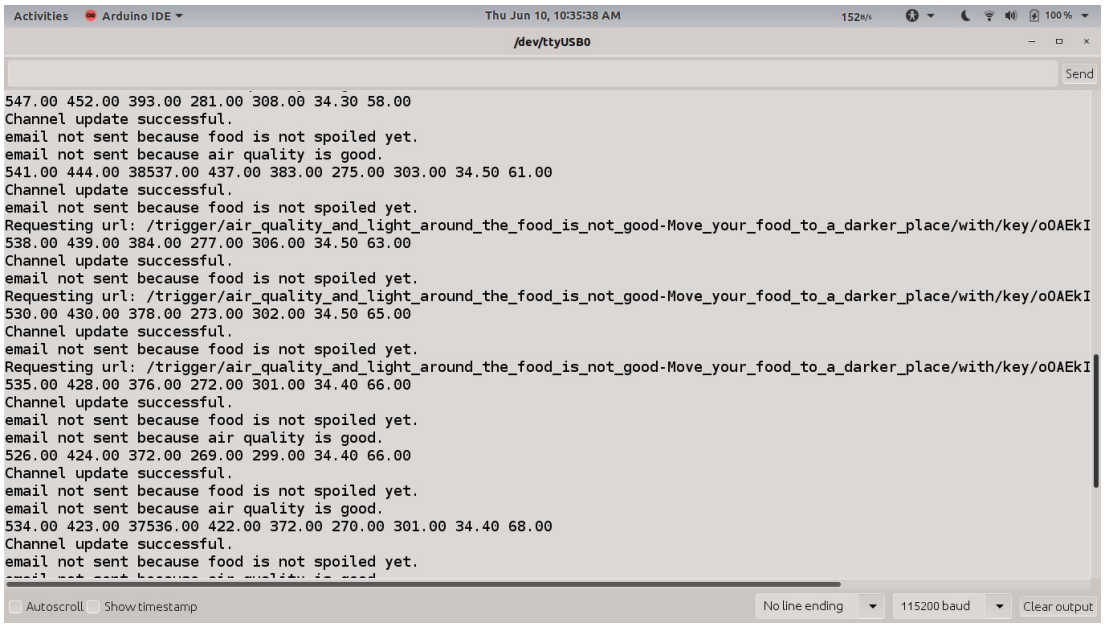

Fig 7.3 Nodemcu serial monitor output-3

After a few minutes the device comes back to its initial state saying that the air quality around the food is not food.

Note: Thingspeak free version is being used. So, the updates the very slow and the performance will decrease slightly.

7.2. Emails:

For every few updates, the emails will be send three times. Until the user resets the device and power off the arduino, the emails will be send continuously.

If the air quality around the food is not good, the email will be sent and the if the food is spoiled, then another email will be sent.

Due to slow updates in thingspeak, there might some delay of 40 seconds to One minute. Look at the below images for more clarity.

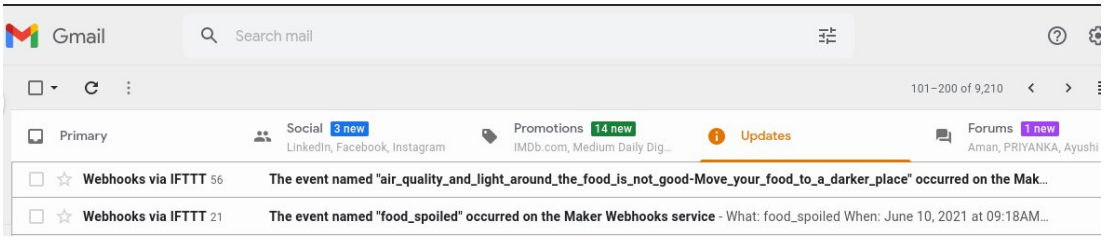

Fig 7.4 emails

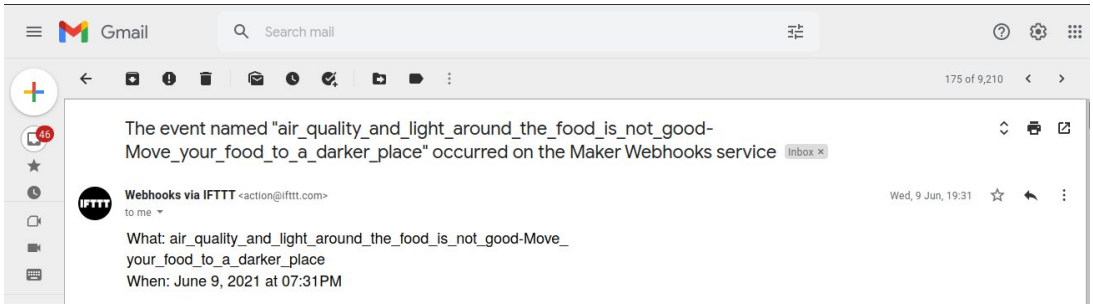

Fig 7.5 air quality email

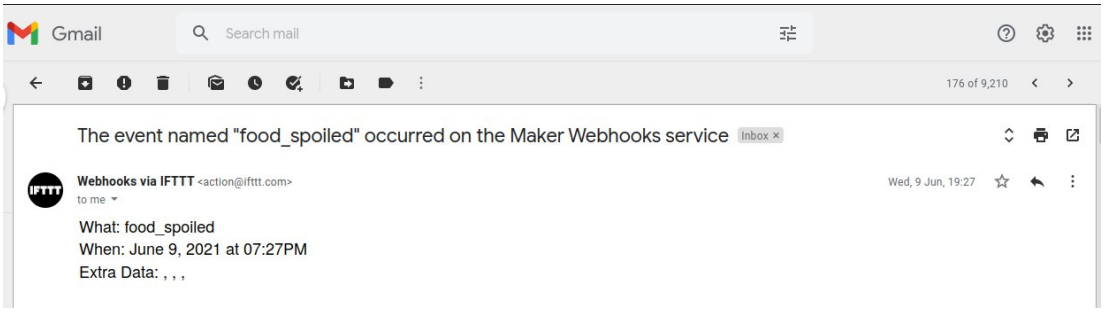

Fig 7.6 food spoiled email

7.3. Thingspeak updation:

For every fourty seconds, the sensor values are being updated and the values in the thingspeak looks like this.

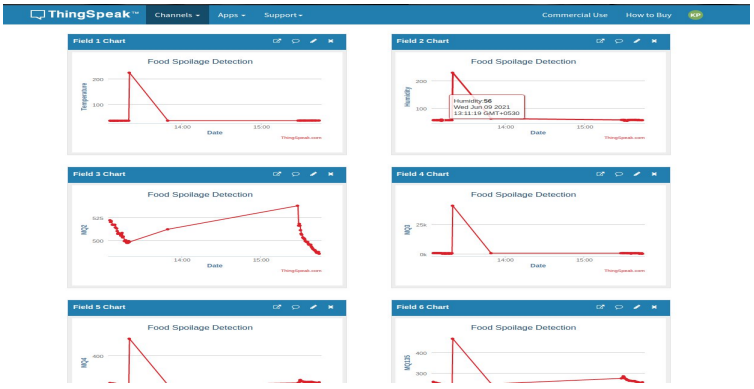

Fig. 7.7 thingspeak updation

7.4 PERFORMANCE EVALUATION:

There is no standard performance evaluation metric to check the accuracy. The threshold values of each sensor is calculated by experimentation. The sensor is recorded in the cloud for every fourty seconds. So, fast and immediate results is not possible. Internet connection should be stable otherwise the data will be uploaded slowly which might decrease the performance of the system.

The performance should be evaluated by experimentation only.

CHAPTER-8

CONCLUSION AND FUTURE WORK

8.1 CONCLUSION:

Food poisoning has been the source of innumerable diseases, to reduce and avoid illness we use bio sensors and electrical sensors to determine the freshness of household food items like diary, fruits, and meat, to expend the device for more items by adding new sensors and by using existing sensors.

Detecting the naturally emitted gases such as Methane, Ammonia and Ethylene as food decay can be used to detect food spoilage. Even before the presence of any visible signs of spoilage, Arduino gas sensors are able to detect gas emissions from food items. Using sensors to detect the presence of these gases among foods can help detect food spoilage early and prevent consumption of spoiled food. Different types of sensors can be linked to further extend the usage of system and to increase the sensitivity of such detection methods.

As per research conducted by World Health Organization an estimated 600 million – almost 1 in 10 people in the world – fall ill after eating contaminated food and 420,000 die every year, resulting in the loss of 33 million healthy life years (DALYs).

8.2 FUTURE WORK:

- The sensor data is recorded in the cloud and can be downloaded in csv format. So that it can be used for deploying machine learning and deep learning algorithms and this data can be used for further development.
- We can add more sensors like MQ136 and MQ137 for detecting hydogen sulfide and ammonia gases. Meat, fish, etc. releases sulfide gases when there are spoiled.
- Voice command can be provided
- GPS module can be added if there are two or more food spoilage devices.
- Detection of alcohol level in food substance to expand the sensing fields.
- For application in liquid process food streams, inclusion of Nano-Detect process will be used to develop on-line and off-line monitoring systems (sensors) which combine the expertise of sensitive molecular biological processes with the potency of nanotechnology.
- Usage of high precision sensors to increase area of sensing.
- Integration of two or more sensors for foods which display dual parameters. Based on amount of calorie consumption a pressure sensor is included which helps to maintain a balanced diet.

APPENDIX

SOURCE CODE

ARDUINO UNO:

```
#include<DHT.h>  // An arduino library for reading DHT-11 sensor

#define MQ2_PIN A0

#define MQ3_PIN A1

#define MQ4_PIN A2

#define MQ135_PIN A3

#define LDR A4

#define DHT11_PIN 8

#define DHTTYPE DHT11


float MQ2_read=0, MQ3_read=0, MQ4_read=0, MQ135_read=0;

float temperature=0, humidity=0;

float LDRReading = 0;


DHT dht(DHT11_PIN, DHTTYPE); // initializing DHT object


void setup() {

  Serial.begin(9600);
```

```
  dht.begin();
  pinMode(MQ2_PIN, INPUT);
  pinMode(MQ3_PIN, INPUT);
  pinMode(MQ4_PIN, INPUT);
  pinMode(MQ135_PIN, INPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
  MQ2_read = analogRead(MQ2_PIN);
  MQ3_read = analogRead(MQ3_PIN);
  MQ4_read = analogRead(MQ4_PIN);
  MQ135_read = analogRead(MQ135_PIN);
  LDRReading = analogRead(LDR);
  temperature = dht.readTemperature();
  humidity = dht.readHumidity();

  Serial.print(LDRReading);
  Serial.print(" ");
  Serial.print(MQ2_read);
```

```
  Serial.print(" ");
  Serial.print(MQ3_read);
  Serial.print(" ");
  Serial.print(MQ4_read);
  Serial.print(" ");
  Serial.print(MQ135_read);
  Serial.print(" ");
  Serial.print(temperature);
  Serial.print(" ");
  Serial.println(humidity);
  delay(40000);
}
```

NODEMCU CODE:

```
#include<SoftwareSerial.h>

#include <ESP8266WiFi.h>                // library used for nodemcu module

#include <ThingSpeak.h>                 // library used to upload data to thingspeak.com


SoftwareSerial mySerial(D1, D2);        // rx, tx

// rx - which receives serial data

// tx - transmit serial data


char* ssid = "Kmurali";                 // WiFi - SSID

char* password = "spectre99";           // WiFi - password

const char* host = "maker.ifttt.com";   // created an applet to send emails


WiFiClient  client;                     // Creates a client that can connect to to a specified internet IP address and port

const int httpPort = 80;

unsigned long myChannelNumber = 1400396; // channel number for thinkspeak

const char* myWriteAPIKey = "08E4IMSBJX21ZBPP";
```

```
String message, LDR, MQ2, MQ3, MQ4, MQ135, temperature, humidity;

int count_spoiled=0,count_air_quality=0, count=0;


void setup() {

  // put your setup code here, to run once:

  Serial.begin(115200);              // enables the serial monitor

  mySerial.begin(9600);              // enables connection to arduino uno


  WiFi.mode(WIFI_STA);               // station mode

  ThingSpeak.begin(client);

  WiFi.begin(ssid,password);


  while(WiFi.status()!=WL_CONNECTED)

  {

    Serial.print(".");

    delay(500);

  }

  Serial.println("");

  Serial.print("IP Address: ");
```

```
  Serial.println(WiFi.localIP());  }


void loop() {
   message = mySerial.readStringUntil('\n');
   LDR = getValue(message,' ',0);
   MQ2 = getValue(message,' ',1);
   MQ3 = getValue(message,' ',2);
   MQ4 = getValue(message,' ',3);
   MQ135 = getValue(message,' ',4);
   temperature = getValue(message,' ',5);
   humidity = getValue(message,' ',6);
   Serial.println(message);


  thingspeak(temperature, humidity, MQ2, MQ3, MQ4, MQ135, LDR); //
sends the sensor data to the thingspeak.com
  sending_email(); // sends an email via an applet created in ifttt.com
  delay(40000);
}
String getValue(String data, char sep, int index) {
  int found = 0;
  int strIndex[] = {0, -1};
```

```
  int maxIndex = data.length()-1;

  for(int i=0; i<=maxIndex && found<=index; i++){
    if(data.charAt(i)==sep || i==maxIndex){
        found++;
        strIndex[0] = strIndex[1]+1;
        strIndex[1] = (i == maxIndex) ? i+1 : i;
    }
  }

  return found>index ? data.substring(strIndex[0], strIndex[1]) : "";
}

void thingspeak(String temperature, String humidity, String MQ2, String MQ3, String MQ4, String MQ135, String LDR) {
  ThingSpeak.setField(1, temperature);
  ThingSpeak.setField(2, humidity);
  ThingSpeak.setField(3, MQ2);
  ThingSpeak.setField(4, MQ3);
  ThingSpeak.setField(5, MQ4);
  ThingSpeak.setField(6, MQ135);
```

```
  ThingSpeak.setField(7, LDR);


  // write to the ThingSpeak channel
  int x = ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);
  if(x == 200){
    Serial.println("Channel update successful.");
  }
  else {
   Serial.println("Problem updating channel. HTTP error code " + String(x));
  }
}


void sending_email() {
  if(!client.connect(host, httpPort)) {
      Serial.println("email connection failed");
      return;
    }


  if(count == 15) {
      count_spoiled = 0;
```

```
    count_air_quality = 0;
    count=0;
   }

 if(count_spoiled!=3 && MQ2.toInt() >= 470 && MQ3.toInt() >= 345 && MQ4.toInt() >= 280) {
   String url1 = "/trigger/food_spoiled/with/key/oOAEkILKPPinkD1S3IJ5I7FCYNdDNmWFzlgsNFoqBD";
   Serial.print("Requesting url: ");
   Serial.println(url1);
   client.print(String("GET ") + url1 + " HTTP/1.1\r\n" +
                "Host: " + host + "\r\n" +
                "Connection: close\r\n\r\n");

   count_spoiled += 1;
   } else {
    Serial.println("email not sent because food is not spoiled yet.");
   }
 delay(5000);
 if(count_air_quality!=3 && LDR.toInt() > 300 && MQ135.toInt() > 270) {
```

```
  String url =
"/trigger/air_quality_and_light_around_the_food_is_not_good-
Move_your_food_to_a_darker_place/with/key/oOAEkILKPPinkD1S3IJ5I7FC
YNdDNmWFzlgsNFoqBD";

  Serial.print("Requesting url: ");

  Serial.println(url);

  client.print(String("GET ") + url + " HTTP/1.1\r\n" +

                "Host: " + host + "\r\n" +

                "Connection: close\r\n\r\n");

 count_air_quality += 1;

 } else {

  Serial.println("email not sent because air quality is good.");

 }

count+=1;

}
```

LIST OF ABBREVIATIONS/NOMENCLATURE:

S No	SHORT FORM	FULL FORM
1	FSC	Food Supply chain
2	IoT	Internet of things
3	AIM	automatic ingestion monitor
4	LDR	Light Dependent Resistor
5	USB	Universal Serial Bus
6	ICSP	In-Circuit Serial Programming
7	MOS	Metal Oxide Semiconductor
8	IOREF	Input Output Reference
9	CSV	Comma Seperated Values
10	IDE	Integrated Development Environment

LIST OF FIGURES:

LIST OF TABLES:

REFERENCES

[1] Garcha, S (September 2018). "Control of food spoilage molds using lactobacillus bacteriocins". Journal of Pure and Applied Microbiology. 12 (3): 1365–1373. doi:10.22207/JPAM.12.3.39.

[2] Pitt, John I.; Hocking, Ailsa D. (2009). Fungi and Food Spoilage. Bibcode:2009ffs..book.....P. doi:10.1007/978-0-387-92207-2. ISBN 978-0-387-92206-5.

[3] Jeanroy, Amelia; Ward, Karen (2009-07-31). Canning & Preserving for Dummies. p. 41. ISBN 9780470555453.

[4] "What is Food Spoilage? | FoodSafety.gov". www.foodsafety.gov. 2016-03-08. Retrieved 2019-04-07.

[5] Food spoilage and food pathogens, what's the difference? October 22, 2015, Michelle Jarvie, Michigan State University .

[6] Jeanroy, Amelia; Ward, Karen (2009-07-31). Canning & Preserving for Dummies. p. 39. ISBN 9780470555453.

[7] Magoulas, Argyris (February 22, 2016). "What is Food Spoilage?". Foodsafety.gov. U.S. Department of Health & Human Services. Retrieved October 25, 2018.

[8] https://developer.ibm.com/tutorials/iot-nodemcu-open-why-use/

[9] Mpx. "Lua CJSON is a fast JSON encoding/parsing module for Lua". Github. Retrieved 2 April 2015.

[10] Pellepl. "Wear-leveled SPI flash file system for embedded devices". GitHub. Retrieved 2 April 2015.