**IOT BASED SMART CONTAINER USING NODE-MCU**

A

Mini Project Report

Submitted in the Partial Fulfillment of the

Academic Requirements

For the Award of the Degree of

**Bachelor of Technology**

**In**

**Electronics and Communication Engineering**

By

**NALLA NARESH 20AG5A0419**

Under The Esteemed Guidance Of

**Mr. B. GIRI RAJU**

**PROFESSOR**

****

**DEPARTMENT OF ELECTRONICS & COMMUICATION ENGINEERING  
ACE ENGINEERING COLLEGE**

**(NBA ACCREDITED B.TECH COURSES, ECE, EEE, CSE, MECH & CIVIL)**

**(NAAC “A” GRADE)**

**An Autonomous Institution**

**Ankushapur(V), Ghatkesar(M), R.R.Dist - 501 301**

2022 -2023

** ACE**

**Engineering College**

**(NBA ACCREDITED B.TECH COURSES ECE, EEE, CSE, MECH & CIVIL) (NAAC “A” GRADE)**

**An Autonomous Institution**

**Ankushapur (V), Ghatkesar (M), Madchal.Dist – 501 301**

**CERTIFICATE**

This is to certify that the project work entitled **“IOT BASED SMART CONTAINER USING NODE-MCU”** is carried out by **NALLA NARESH** with Roll no.**20AG5A0419** .Partial fulfillment for the award of degree of **Bachelor of Technology** in Electronics and Communication Engineering to the **Jawaharlal Nehru Technological University**, **Hyderabad** is a record of bonafide work carried out by them under our guidance and supervision during the academic period 2022-2023.

**Mr. B. GIRI RAJU** **Mr. Dr. P SATISH KUMAR**

Professor (Professor and HOD)

(Project Supervisor)

**EXTERNAL EXAMINER**

I

**ACKNOWLEDGMENT**

The satisfaction that accompanies the successful completion of the task would be put incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crown all the efforts with success.

I would like to express our heart-felt gratitude to our parents without whom we would not have been privileged to achieve and fulfill our dreams. A special thanks to our secretary .**Prof. Y.V. GOPALA KRISHNA MURTHY**, for having founded such an esteemed institution.

I show gratitude to our Principal **Dr. B.L RAJU** for providing all the facilities and support.

I am thankful to **Mr. Dr. P SATISH KUMAR**, Head of Department, Electronics and Communication Engineering for his intense support and encouragement which helped me to mound my project into a successful one.

I would like to thank to **Mr. B GIRI RAJU** my deep sense of gratitude, Our guide and Project Coordinator, Department of Electronics and Communication Engineering, ACE Engineering College for his expert guidance and useful suggestions which helped me in completing the project work in time.

I also thank all the faculty members of Electronics and Communication Engineering Department for their valuable support and generous advice. Finally thanks to all my friends and family members for their continuous support and enthusiastic help.

With gratitude

**NALLA NARESH (20AG5A0419)**

II

II

**ABSTRACT**

This paper is about building a smart container using Node-MCU and ultrasonic sensor to track the contents in the container dynamically from anywhere in the world with the help of internet. This smart container allows us to keep track of the stocks, and it is easily accessible from using the internet. The Container includes an ultrasonic sensor at the top of it and uses the ultrasonic reflected waves to figure out at what extent the Container is filled and how much space is left inside the container. Whenever the amount of content changes in the jar, it is sensed by the Node-MCU, and the same is updated on the web server. This can be helpful to track supplies and plan for restocking from anywhere in the world. The container is smart enough to send the level of container filled through email and warns when the container is empty or less than the threshold value set as per the requirement. It would be helpful for the user to track the details of contents in the container without their actual presence. The smart jar enables us to keep track of the medicines stocks with the help of an android app easily accessible anywhere with a simple internet connection. The jar contains an ultrasonic sound waves emitter and sensor which uses the reflected ultra-sonic waves to find out what level the jar is filled to and how much empty space remains in jar yet to be filled. This sensor is also connected to the internet and interfaced with the application so that as soon as the level of content in the jar changes the data is updated to us in the application without any delay. This helps us in monitoring the stocks and prepare for restocking from anywhere, additionally it also provides important data such as the expiry date and the manufactured data of the content in the jar. .

**Keywords: Node-MCU ESP8266 Wi-Fi Module, HC-05 Ultrasonic Sensor, Smart container, Arduino IDE, IOT architecture**

III

**TABLE OF CONTENTS**

**Page No**

Certificate I

Acknowledgement II

Abstract III

Table of content IV

List of Tables VII

List of Figures VIII

**CHAPTER 1 INTRODUCTION**

* 1. Introduction 1
  2. Reason Of The Project 1
  3. Methodology 2
  4. Role Of IOT 2
  5. System Block Diagram 3
  6. Component 3

**CHAPTER 2 BLOCKS DIAGRAM & DESCRIPTION**

1. Introduction To Node MCU 4
   1. Specification Of Node MCU 5
   2. Pin Diagram Of Node MCU 6
   3. Node MCU Development Board Configuration 8
   4. Esp8266 Node Mcu With Arduino IDE 9

**CHAPTER 3 HARDWARE DESCTIPTION**

1. Introduction Of Ultrasonic Sensor 10
   1. Principal Of Ultrasonic Sensor 11
   2. How Ultrasonic Sensor Works? 11
   3. Ultrasonic Sensor Timing Diagram 12
   4. Ultrasonic Sensor Specification 13
   5. Pin Configuration Of Ultrasonic Sensor 13
   6. Limitation Of Ultrasonic Sensor 14
   7. Introduction To LM35 Sensor 14
   8. LM35 Sensor Features 15
   9. LM35 Sensor Working Principle 15
   10. LM35 Circuit Configuration 15

IV

* 1. LM35 Temperature To Voltage Calculation 16
  2. LM35 Sensor Pin Diagram 17
  3. The Accuracy Level Of Lm35 17

**CHAPTER 4 SOFTWARE DESCTIPTION**

1. Introduction To Software 18
   1. IOT Software 18
   2. Software’s Used In Project 19
      1. Arduino IDE 19
      2. Setup Node MCU Board To Arduino IDE 20

4.3.3 IFTTT (IF THIS THEN THAT) 24

* 1. How Does IFTTT Works? 24
  2. Creation Of Applet In IFTTT 25
  3. Programming Code Explanation 27
  4. Code 29

**CHAPTER 5 CONFIGURATIONS & IMPLEMENTATION**

1. Circuit Diagram Of Project 34
   1. Interconnection Details 34
      1. Ultrasonic Sensor 34
      2. LM35 Sensor 35
      3. Power Supply 35
   2. Flowchart 35
   3. Workflow 36
   4. Results 37
      1. Case-1 37
      2. Case-2 39
   5. Advantages 40
   6. Disadvantages 40
   7. Applications 40

**CHAPTER 6 RESULTS & CONCLUSIONS**

6.1 Conclusion 41

6.2 Future scope 41

V

**BIBLIOGRAPHY**

Text Books 42

Websites 42

VI

**LIST OF TABLES**

**Table Title page**

1. Node MCU Development Board 8
2. Pin Configuration Of Ultrasonic Sensor 13
   1. Pin Configuration Of LM35 Sensor 17
   2. LM35 Accuracy Table 17

VII

**LIST OF FIGURES**

**Figure Title page**

1.1 Block Diagram 3

1. Node MCU ESP8266 4
   1. Pin Diagram Of Node MCU 6
2. Ultrasonic Sensor 10
   1. Ultrasonic Sensor Wave Diagram 11
   2. Ultrasonic Sensor Timing Diagram 12
   3. LM35 Sensor 14
   4. LM35 Circuit Configuration 16
   5. Pin Configuration Of Lm35 17
3. IOT Software 18
   1. Arduino IDE 19
   2. IFTTT 24
   3. Web hooks & Key 25
   4. Creation Of IFTTT 25
   5. Email In Web hooks 26
   6. Event Creation 26
4. Circuit Configuration Diagram 34
   1. Flowchart 35
   2. Empty Container 37
   3. Output Of Web Server 37
   4. Email Notification 38
   5. Email Notification Content 38
   6. IFTTT Connecting Server 38
   7. Partially Filled Container 39
   8. Web Server Output 39

VIII

**CHAPTER 1**

**INTRODUCTION**

* 1. **INTRODUCTION**

Now days, In our day to day life, people became more busy such that they are much more immersed in their work schedule and are unable to concentrate or take care of their household things. Most of the time people find it difficult to track their kitchen items, because nowadays it has become a room of least visit. Ina scenario, where people find hard time to prepare their own food, it is even more difficult for them to track their grocery items in their room. They even waste most of their time in ordering and tracking the remaining items in kitchen room. In this project, I made an IOT based Smart jar using Node-MCU ESP8266 Module and ultrasonic sensor that tracks the exact percentage of the contents in the container. This helps the user to know the quantity of content in the jar and can access the information through the Internet from anywhere in the world. It takes the speed of the emitted waves and time taken to travel as the parameters and gives the actual content of the container.

Node-MCU and ultrasonic sensor to track the contents in the jar dynamically from anywhere in the world with the help of internet. This smart Jar allows us to keep track of the stocks, and it is easily accessible from using the internet. The Jar includes an ultrasonic sensor at the top of it and uses the ultra- sonic reflected waves to figure out at what extent the Jar is filled and how much space is left inside the jar. Whenever the amount of content changes in the jar, it is sensed by the Node-MCU, and the same is updated on the web server. This can be helpful to track supplies and plan for restocking from anywhere in the world. The jar is smart enough to send the level of jar filled through email and warns when the jar is empty or less than the threshold value set as per the requirement .It would be helpful for the user to track the details of contents in the jar without their actual presence.

* 1. **REASON OF THE PROJECT**

The main aim of the project is to design and develop a system to monitor the amount of filled content in the container without his/her presence at a container and to get notified when container is empty also updating of container filled amount in percentage form to web server with temperature, by using internet service .this all happens without manual operation.

* 1. **METHODOLOGY**

The project is developed on concept of Ultrasonic Sensor. We use major component for this project is Ultrasonic Sensor strategy, as it is used for calculation of distance of the object in front of the sensor. Based on this calculation the overall circuit function is takes place.LM35 is also used for taking care of temperature in container. Thus based on this, we used to get output result

* 1. **ROLE OF IOT**

When compared to other technologies, is one of its kinds, which has the maximum capability. Other advantages include less human effort in the work field, reduction of cost, enhancement in data collection and analysis.IOT is also capable in using the resources effectively. The embedded system can be of type microcontroller or type microprocessor. Both of these types contain an integrated circuit (IC). The essential component of the embedded system is a RISC family microcontroller like Motorola 68HC11, PIC 16F84, Atmel 8051 and many more. The most important factor that differentiates these microcontrollers with the microprocessor like 8085 is their internal read and writable memory. The essential embedded device components and system architecture are specified below. The IOT ecosystem is not easy to define. It is also difficult to capture its proper image due to the vastness and emerging possibility and the rapidity with which it is expanding in the entire sector. However, the IOT ecosystem is a connection of various kinds of devices that sense and analyze the data and communicates with each other over the networks.

In the IOT ecosystem, the user uses smart Electronic copy available at: https://ssrn.com/abstract=3915539 devices such as smart phones, tablet, sensors, etc. to send the command or request to devices for information over the networks. The device response and performs the command to send information back to the user through networks after analyzed. The typical IOT ecosystem is shown in below image, where the smarter devices send and receive data from the devices themselves in the environment that are integrate over network and Cloud Computing.

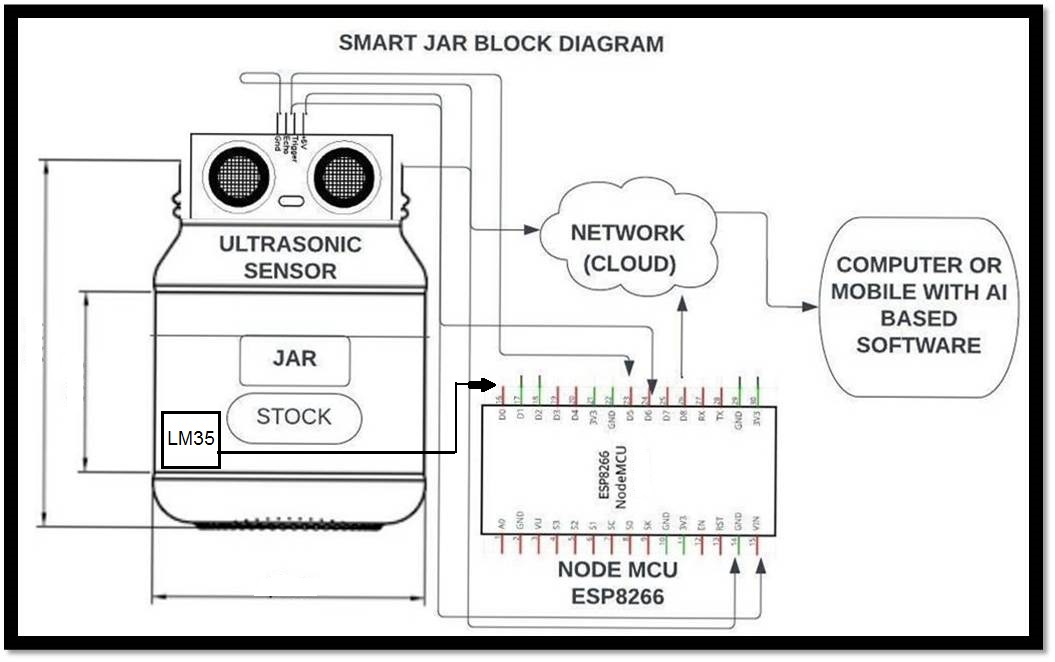
* 1. ** SYSTEM BLOCK DIAGRAM**

Fig.1.1 Block Diagram of Smart Container

Here is a block diagram of the project with basic input and output functions of each block. Complete set project measurements that are required, are mentioned in above figure .It illustrate the basic operation that takes place in project and also the hardware (components) are also mentioned in each block with ,that input and output supply pipe line

* 1. **COMPONENTS USED IN PROJECT**
* Node MCU ESP8266
* Ultrasonic Sensor HC-05
* LM35 Temperature sensor
* 9V Battery
* Container (jar)
* Jumper wires
* Switch

**CHAPTER 2**

**NODE-MCU ESP8266**

1. **INTRODUCTION TO NODE-MCU ESP8266**

The Node-MCU (**N**ode**M**icro**C**ontroller **U**nit*)* is open-source software and hardware development environment built around an inexpensive System-on-a-Chip (SOC) called the ESP8266. The ESP8266 designed and manufactured by Espress if Systems, contains the crucial elements of a computer: CPU, RAM, networking (Wi-Fi), and even a modern operating system and SDK. That makes it an excellent choice for Internet of Things (IOT) projects of all kinds. However, as a chip, the ESP8266 is also hard to access and use. You must solder wires, with the appropriate analog voltage, to its pins for the simplest tasks such as powering it on or sending a keystroke to the “computer” on the chip. You also have to program it in low-level machine instructions that can be interpreted by the chip hardware. This level of integration is not a problem using the ESP8266 as an embedded controller chip in mass-produced electronics. It is a huge burden for hobbyists, hackers, or students who want to experiment with it in their own IOT projects.



Fig.2.1 Node MCU ESP8266

But, The Arduino project created an open-source hardware design and software SDK for their versatile IOT controller. Similar to Node-MCU, the Arduino hardware is a microcontroller board with a USB connector, LED lights, and standard data pins. It also defines standard interfaces to interact with sensors or other boards. But unlike Node-MCU, the Arduino board can have different types of CPU chips (typically an ARM or Intel x86 chip) with memory chips, and a variety of programming environments. There is an Arduino reference design for the ESP8266 chip as well.

It’s perfect for IOT projects, especially other Wireless connectivity projects as Arduino does not work wirelessly. We either need to connect it to a Bluetooth or RF module this chip has a great deal in common with the Arduino – they’re both microcontroller-equipped prototyping boards that can be programmed using the Arduino IDE. The ESP8266 is more updated and younger than Arduino, and therefore the ESP has stronger specifications than Arduino.

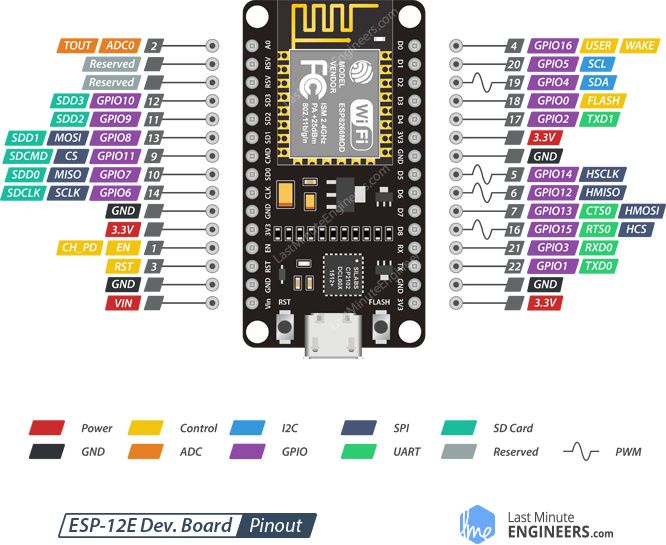
* 1. **SPECIFICATIONS**
* Operating Voltage: 3.0-3.6 V
* Operating Current: 80mA
* Operating temperature: -40 to 125 degree Celsius
* 32-bit MCU
* Integrated 10-bit ADC
* 802.11 b/g/n
* Integrated TCP/IP protocol
* 2.4 GHz Wi-Fi that supports WPA/WPA2
* It supports UART, SPI, I2C, IR remote, PWM, SDIO 2.0
* It has 20 I/O ports
* Digital I/O Pins (DIO): 16
* Analog Input Pins (ADC): 1
* Flash Memory: 4 MB
* SRAM: 64 KB
* Clock Speed: 80 MHz
* PCB Antenna
* USB-TTL based on CP2102 is included onboard, Enabling Plug n Play
* UARTs: 1
* SPIs: 1
* I2Cs: 1
  1. ** PIN DIAGRAM OF NODE-MCU**

Fig. 2.2 Pin Diagram of Node-MCU

**POWER PINS**: There are four power pins viz. one VIN pin & three 3.3V pins. The VIN pin can be used to directly supply the ESP8266 and its peripherals, if you have a regulated 5V voltage source. The 3.3V pins are the output of an on-board voltage regulator. These pins can be used to supply power to external components.

**GND** is a ground pin of ESP8266 Node-MCU development board.

**I2C PINS:** To hook up all sorts of I2C sensors and peripherals in your project. Both I2C Master and I2C Slave are supported. I2C interface functionality can be realized programmatically, the clock frequency is 100 kHz at a maximum. It should be noted that I2C clock frequency should be higher than slowest clock frequency of the slave device.

**GPIO PINS :**ESP8266 Node-MCU has 17 GPIO pins which can be assigned to various functions such as I2C, I2S, UART, PWM, IR Remote Control, LED Light and Button programmatically. Each digital enabled GPIO can be configured to internal pull-up or pull-down, or set to high impedance. When configured as an input, it can also be set to edge-trigger or level-trigger to generate CPU interrupts.

**ADC CHANNEL** The Node-MCU is embedded with a 10-bit precision SAR ADC. The two functions can be implemented using ADC viz. Testing power supply voltage of VDD3P3 pin and testing input voltage of TOUT pin. However, they cannot be implemented at the same time.

**UART PINS:** ESP8266 Node-MCU has 2 UART interfaces, i.e. UART0 and UART1, which provide asynchronous communication (RS232 and RS485), and can communicate at up to 4.5 Mbps. UART0 (TXD0, RXD0, RST0 & CTS0 pins) can be used for communication. It supports fluid control. However, UART1 (TXD1 pin) features only data transmit signal so; it is usually used for printing log.

**SPI PINS:** ESP8266 features two SPIs (SPI and HSPI) in slave and master modes. These SPIs also support the following general-purpose SPI features:

* 4 timing modes of the SPI format transfer
* Up to 80 MHz and the divided clocks of 80 MHz
* Up to 64-Byte FIFO

**SDIO PINS:** ESP8266 features Secure Digital Input/output Interface (SDIO) which is used to directly interface SD cards. 4-bit 25 MHz SDIO v1.1 and 4-bit 50 MHz SDIO v2.0 are supported.

**PWM PINS:** The board has 4 channels of Pulse Width Modulation (PWM). The PWM output can be implemented programmatically and used for driving digital motors and LEDs. PWM frequency range is adjustable from 1000 μs to 10000 μs, i.e., between 100 Hz and 1 kHz.

**CONTROL PINS:** are used to control ESP8266. These pins include Chip Enable pin (EN), Reset pin (RST) and WAKE pin.

* EN pin – The ESP8266 chip is enabled when EN pin is pulled HIGH. When pulled LOW the chip works at minimum power.
* RST pin – RST pin is used to reset the ESP8266 chip.
* WAKE pin – Wake pin is used to wake the chip from deep-sleep.
  1. **NODE-MCU DEVELOPMENT BOARD CONFIGURATION**

|  |  |  |
| --- | --- | --- |
| **Pin** | **Name** | **Description** |
| Power | Micro-USB, 3.3V, GND, Vin | **Micro-USB:** Node-MCU can be powered through the USB port  **3.3V:** Regulated 3.3V can be supplied to this pin to power the board  **GND:** Ground pins  **Vin:**External Power Supply |
| Control Pins | **EN, RST** | The pin and the button resets the microcontroller |
| Analog Pin | A0 | Used to measure analog voltage in the range of 0-3.3V |
| GPIO Pins | GPIO1 to GPIO16 | Node-MCU has 16 general purpose input-output pins on its board |
| SPI Pins | SD1, CMD, SD0, CLK | Node-MCU has four pins available for SPI communication. |
| UART Pins | TXD0, RXD0, TXD2, RXD2 | Node-MCU has two UART interfaces, UART0 (RXD0 & TXD0) and UART1 (RXD1 & TXD1). UART1 is used to upload the firmware/program. |
| I2C Pins |  | Node-MCU has I2C functionality support but due to the internal functionality of these pins, you have to find which pin is I2C. |

* 1. **ESP8266 NODE-MCU WITH ARDUINO IDE**

If you want to use the ESP8266 Node-MCU with Arduino IDE, you must first add it to the software. Luckily it’s easy to do this. First, copy the code URL and follow the following steps

* Step 1: Open your Arduino IDE software and navigate to preference in your file menu. Then, input the code URL you copied in ‘Additional Board Manager URLs.”
* Step 2: Navigate to Tools menu **>** Boards **>** Boards manager and search for ESP8266. Install the available ESP8266 boards; you should see the “INSTALLED” label.

You should see boards based on ESP8266 like the Node-MCU on your Arduino IDE. Afterward, you can choose your preferred board from the list to upload your code.

**CHAPTER 3**

**ULTRASONIC SENSOR & LM35 SENSOR**

* 1. **INTRODUCTION OF ULTRASONIC**

An ultrasonic sensor is a device that transmits and receives ultrasonic waves and converts them into electrical signals. It is mainly used to tell how far away the object is. The ultrasonic waves travel faster than the sound waves. The ultrasonic sensor has basically two parts: one is transmitter and the other is receiver. The transmitter emits the ultrasonic waves towards the targeted object. The waves hit the target object and return towards the sensor with in a specific time. The receiver of the ultrasonic sensor receives these reflected waves. The sensor calculates the time taken between the transmission and receiving of the signal waves. The distance is calculated by the formula distance=time\*(speed of the wave).



Fig. 3.1 Ultrasonic Sensor

To know the distance between the target and the sensor, the sensor calculates the amount of time required for sound emission to travel from transmitter to receiver. The calculation is done as follows:

**D = 1/2 T \* C**

* Where ‘T’ corresponds to time measured in seconds
* ‘C’ corresponds to sound speed = 343 measured in mts/sec
  1. **PRINCIPLE OF ULTRASONIC SENSOR**

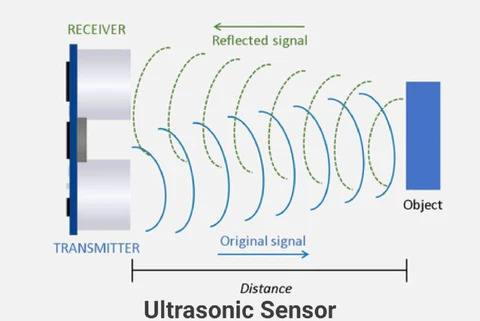
 Working principle is either similar to sonar or radar which evaluates the target/object attributes by understanding the received echoes from sound/radio waves correspondingly. These sensors produce high-frequency sound waves and analyze the echo which is received from the sensor. The sensors measure the time interval between transmitted and received echoes so that the distance to the target is known.

Fig. 3.2 Ultrasonic Sensor Waves Diagram

They are all bounders of technology and can be used in any industrial application. There are several types of objects that can be detected, including solids, liquids, granules, and powders. They reliably detect transparent or glossy objects, as well as objects whose colors change

* 1. **HOW ULTRASONIC SENSOR WORKS?**

An ultrasonic sensor is an electronic device that measures the distance to an object by emitting ultrasonic waves and converting the reflected sound into electrical signals. Ultrasound travels faster than audible sound (that is, sound that humans can hear). An ultrasonic sensor consists of two main components: a transmitter (which uses a piezoelectric crystal to emit sound) and a receiver.

While some sensors use separate sound emitters and receivers, it is also feasible to merge both functions into a single device by using an ultrasonic element to switch between sending and receiving signals in a continuous cycle. The transmitter of the module transmits an ultrasonic sound. This sound will be reflected if an object is present in front of the ultrasonic sensor. The reflected sound is received by the receiver present in the same module. An ultrasonic signal is propagated by a wave at an angle of 30°. The above-depicted Figure illustrates how the ultrasonic signal propagates from the transmitter. Measuring angles should be at least 15° for maximum accuracy. In this case, external objects that fall under this measurement angle interfere with determining the distance to the desired object.

The distance is determined by measuring the travel time of ultrasonic sound and its speed.

Distance = Time x Speed of sound / 2

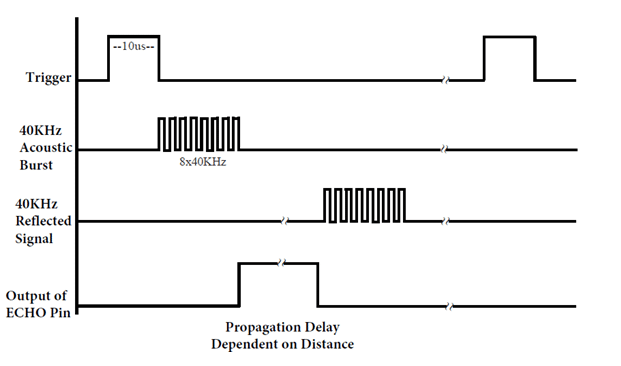
* 1. **ULTRASONIC SENSOR TIMING DIAGARAM**

Fig. 3.3 Ultrasonic Sensor Timing Diagram

**Procedure to analyze the timing diagram:**

1. To the Trig pin, the trigger pulse should be supplied for at least 10 µsec.
2. Then the device automatically transmits eight pulses of 40 kHz and waits for the rising edge to appear on the output pin.
3. When the echo pin observed a rising edge, start the timer and observe the time required appearing falling edge at the echo pin.
4. When the echo pin shows a falling edge, observe the timer count. The count of the timer indicates the time taken by the sensor for object detection and getting back from the object.
   1. **ULTRASONIC SENSOR SPECIFICATION**

* The sensing range lies between 40 cm to 300 cm
* The response time is between 50 milliseconds to 200 milliseconds.
* The Beam angle is around 50.
* It operates within the voltage range of 20 VDC to 30 VDC
* Preciseness is ±5%
* The frequency of the ultrasound wave is 120 kHz
* Resolution is 1mm
* The voltage of sensor output is between 0 VDC – 10 VDC
* The ultrasonic sensor weight nearly 150 grams
* Ambient temperature is -250C to +700C
* The target dimensions to measure maximum distance is 5 cm × 5 cm
  1. **PIN CONFIGURATION OF ULTRASONIC SENSOR**

|  |  |
| --- | --- |
| **PIN NAME** | **DESCRIPTION** |
| Vcc | power supply +5 V |
| Gnd | Common ground |
| Trigger pin | To start the sensor |
| Eco pin | Receive the signal |

Table.3.1 Pin configuration

* **VCC** – This pin has to be connected to a power supply +5V.
* **TRIG** – This pin is used to receive controlling signals from the Arduino board. This is the triggering input pin of the sensor.
* **ECHO** – This pin is used for sending signals to the Arduino board where the Arduino calculates the pulse duration to know the distance. This pin is the ECHO output of the sensor.
* **GND** – This pin has to be connected to the ground.
  1. **LIMITATION OF ULTRASONIC SENSOR**

Ultrasonic sensors like the HC-SR04 can efficiently measure distances up to 400cm with a tight tolerance of 3mm. However, if the target is positioned in such a way that the ultrasonic signal is deflected rather than reflected to the ultrasonic sensor, the calculated distance may be incorrect. In some cases, the object is too small to detect enough reflected ultrasound signals to accurately measure distance. Additionally, objects such as fabrics and carpets can absorb acoustic signals. If the signal is absorbed at the edge of the target, it cannot be reflected to the sensor, so range cannot be measured.

* 1. **INTRODUCTION TO LM35 SENSOR**

LM35 is a precession Integrated circuit Temperature sensor, whose output voltage varies, based on the temperature around it. It is a small and cheap IC which can be used to measure temperature anywhere between -55°C to 150°C. It can easily be interfaced with any Microcontroller that has ADC function or any development platform like Arduino.

Power the IC by applying a regulated voltage like +5V (VS) to the input pin and connected the ground pin to the ground of the circuit. Now, you can measure the temperature in form of voltage as shown below.

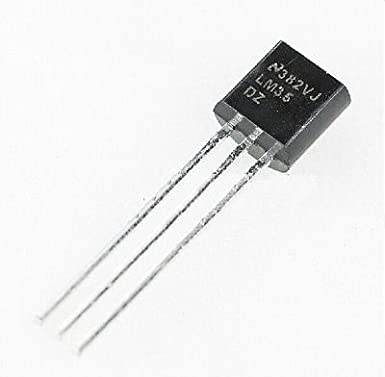
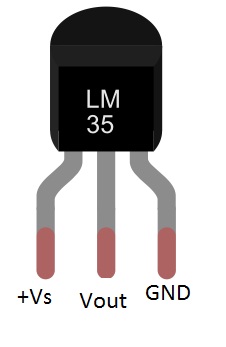
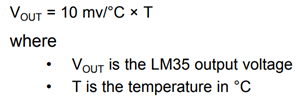


Fig. 3.4 LM35 Temperature Sensor

If the temperature is 0°C, then the output voltage will also be 0V. There will be rise of 0.01V (10mV) for every degree Celsius rise in temperature. The voltage can converted into temperature using the below formulae.

**Voltage Temperature Converter Formula**

****

* 1. **LM35 FEATURES**
* Minimum and Maximum Input Voltage is 35V and -2V respectively. Typically 5V.
* Can measure temperature ranging from -55°C to 150°C
* Output voltage is directly proportional (Linear) to temperature (i.e.) there will be a rise of 10mV (0.01V) for every 1°C rise in temperature.
* ±0.5°C  Accuracy
* Drain current is less than 60uA
* Low cost temperature sensor
* Small and hence suitable for remote applications
* Available in TO-92, TO-220, TO-CAN and SOIC package
  1. **LM35 WORKING PRINCIPLE**

In order to understand the working principle of LM35 temperature sensor we have to understand the linear scale factor. In the features of LM35 it is given to be +10 mills volt per degree centigrade. It means that with increase in output of 10 mills volt by the sensor Vout pin the temperature value increases by one. For example, if the sensor is outputting 100 mills volt at Vout pin the temperature in centigrade will be 10-degree centigrade. The same goes for the negative temperature reading. If the sensor is outputting -100 mills volt the temperature will be -10 degrees Celsius.

* 1. **LM35 SENSOR CIRCUIT CONFIGURATION**

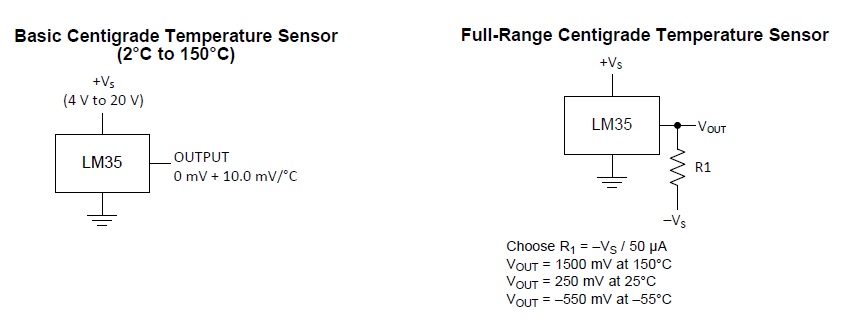
LM35 can be used in two circuit configurations. Both yield different results. In the first configuration, you can only measure the positive temperature from 2 degrees Celsius to 150 degrees Celsius. In this first configuration, we simply power lm35 and connect the output directly to analog to digital converters. In the second configuration, we can utilize all the sensor resources and can measure the full range temperature from -55 degree centigrade to 150-degree centigrade. This configuration is a little complex but yields high results. We have to connect an external resistor, in this case, to switch the level of negative voltage upwards.

Fig. 3.5 LM35 configuration circuit

* 1. **LM35 TEPERATURE TO VOLTAGE CALUCALATION**

The following steps will help you use an LM35 sensor to calculate the temperature.

Start by building the circuit. In the circuit, ensure you connect GND to the ground, then power LM35 VCC with +5 operating voltage (Vs). Thirdly, connect the VOUT to an ADC input (Analog-to-Digital Converter). After which you proceed by sampling the reading from ADC of the output voltage (VOUT).

Finally, finish by converting the output voltage to temperature.

Use the formula below to convert voltage to temperature.

**Centigrade temperature = Voltage read by ADC / 10 mV (mills volt).**

Alternatively, you can use;

**VOUT = -10 mV/°C × T**

Whereby;

VOUT = LM35 output voltage

T = Temperature in centigrade

10mV represents the linear scale factor of LM35.

* 1. **LM35 SENSOR PIN CONFIGURATION**

|  |  |  |
| --- | --- | --- |
| **Pin Number** | **Pin Name** | **Description** |
| 1 | Vcc | Input voltage is +5V for typical applications |
| 2 | Analog Out | There will be increase in 10mV for raise of every 1°C. Can range from -1V(-55°C) to 6V(150°C) |
| 3 | Ground | Connected to ground of circuit |

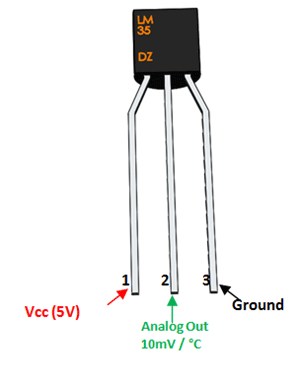


Fig. 3.6 LM35 Pin configuration Diagram

Table.3.2 LM35 Pin configuration

* 1. **THE ACCURACY LEVEL OF LM35**

|  |  |
| --- | --- |
| **Parameter** | **Value** |
| 25°C accuracy level | ±0.5°C |
| -55°C to 150°C accuracy range | ±1°C |
| Temperature slope | 10-mV/°C |

Table.3.3 Accuracy level of LM35

**CHAPTER 4**

**SOFTWARE DESCRIPTION**

* 1. **INTRODUCRION TO SOFTWATRE**

Software is a set of instructions, data, or programs used to operate a computer and execute specific tasks. In simpler terms, software tells a computer how to function. It’s a generic term used to refer to applications, scripts, and programs that run on devices such as PCs, mobile phones, tablets, and other smart devices. Software contrasts with hardware, which is the physical aspects of a computer that perform the work. Without software, most computers would be useless. For example, a web browser is a software application that allows users to access the internet. Without the web browser software, reading a web page on any site wouldn’t be possible. An operating system (OS) is a software program that serves as the interface between other applications and the hardware on a computer or mobile device. TCP/IP is built into all major operating systems to allow computers to communicate over long distance networks. Without the OS or the protocols built into it, it wouldn’t be possible to access a web browser.

* 1. **IOT SOFTWARE**

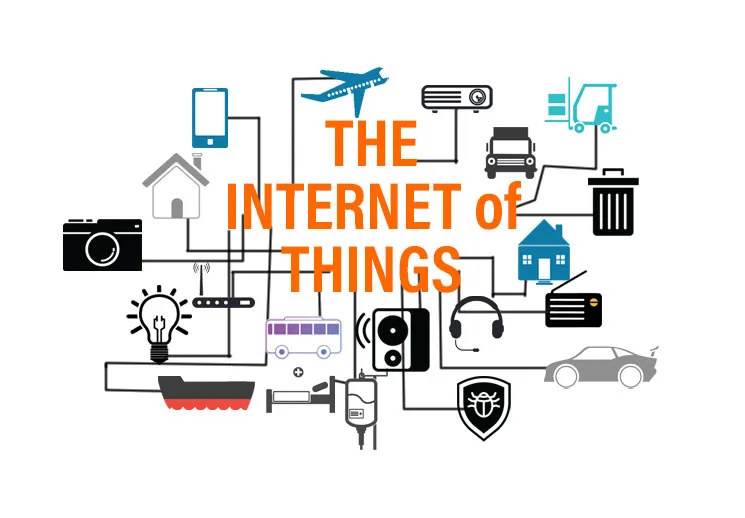
 The following article provides an outline for IOT Software. IOT is about [Internet + (Sense and Communication)]. Things of things are connected through the internet. Let’s take the example of the Mi fitness band- To use this band, a user should install the Mi-fit app to connect to a phone. Band calculates total hours of sleep, the number of calories burn, heart rate monitoring, and sleep analysis. Users can see all the information on mobile phones. Every device, area, software, and sensor are connected, and we can access it through phone or computer.

Fig. 4.1 IOT Software

* 1. **SOFTWARES USED IN PROJECT**
* ARDUINO IDE
* IFTTT (IF THIS THEN THAT)
  + 1. **ARDUINO IDE**

****

Fig. 4.2 Arduino IDE

Arduino designs, manufactures, and supports electronic devices and software, allowing people around the world to easily access advanced technologies that interact with the physical world. Our products are straightforward, simple, and powerful, ready to satisfy users’ needs from students to makers and all the way to professional developers The development of electronics is now easier thanks to Arduino software (IDE), and Arduino boards (hardware) . This set help to build digital and interactive devices with the help of other components. In Previous article we talk about Arduino boards. In this article we will recognize what is Arduino software (IDE), and how use it.

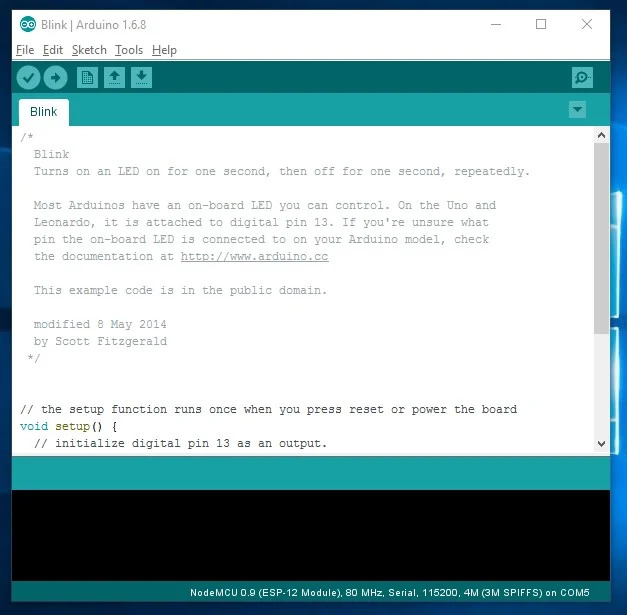
The Arduino software (IDE) is open source software, which is used to program the Arduino boards, and is an integrated development environment, developed by arduino.cc. Allow to write and upload code to Arduino boards. And it consisted of many libraries and a set of examples of mini projects. Arduino software (IDE) is compatible with different operating systems (Windows, Linux, Mac OS X), and supports the programming languages (C/C++)

* + 1. **SETUP NODE-MCU BOARD TO ARDUINO IDE**

**Step 1:** Installing Arduino IDE Software

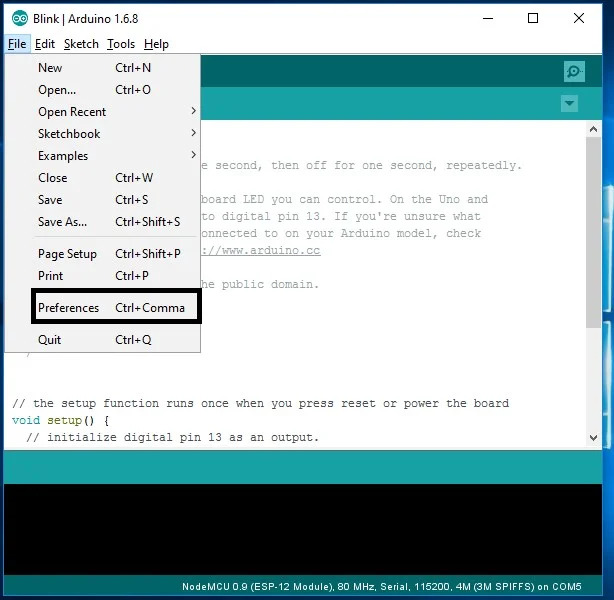
**Step 2:** Opening Arduino IDE

Click on the Icon to open the Arduino window as shown in the figure

****

**Step 4:** Preferences

Open the File and click on the Preferences as shown in the figure

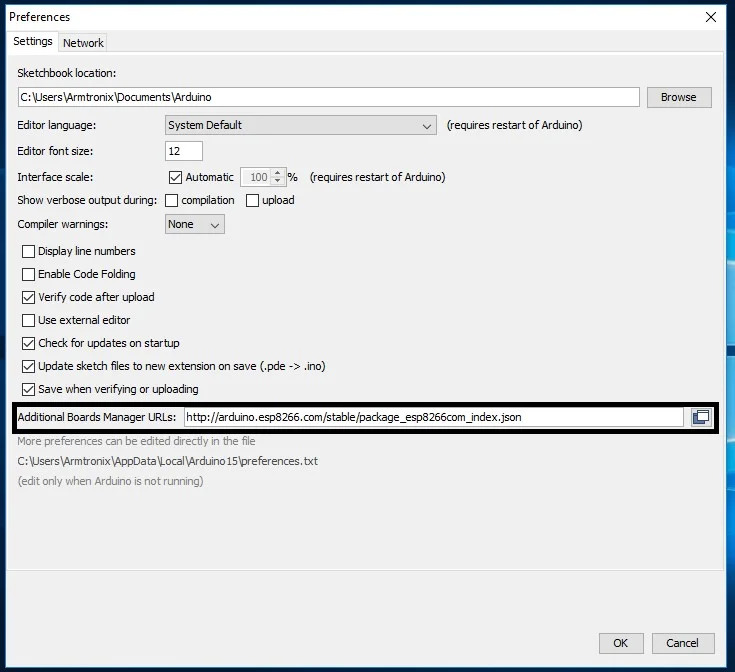
****

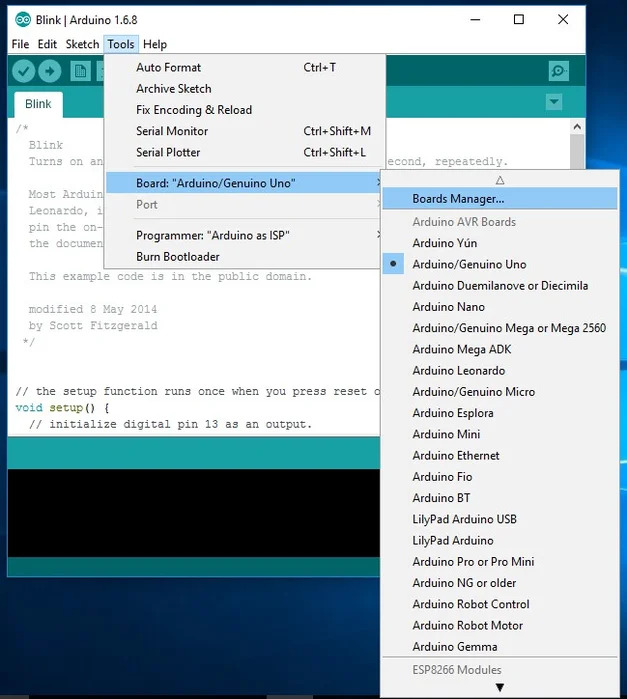
**Step 5**: Adding ESP8266 Board Manager

In the Additional Boards Manager enter below URL.

http://arduino.esp8266.com/stable/package\_esp8266com\_index.json

As highlighted in the figure and enter OK

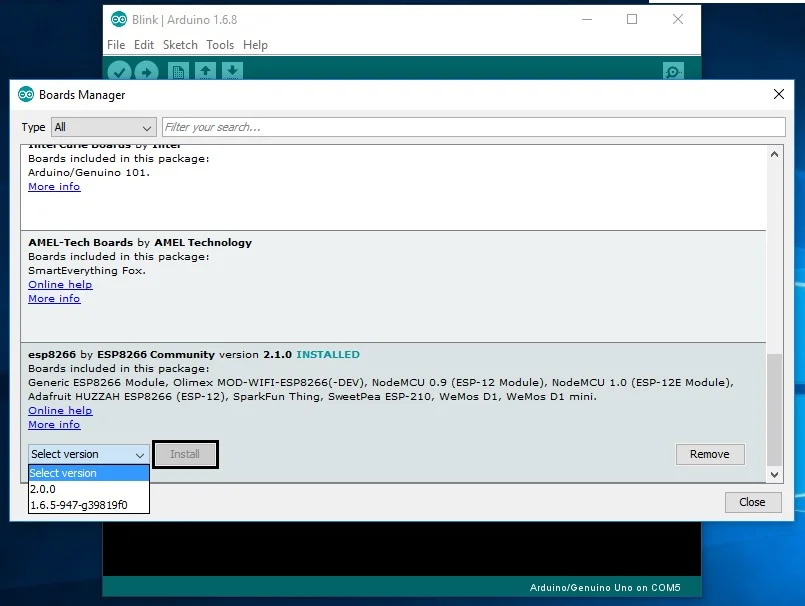
**  
Step 6:** Selecting Board

Now open the tools in that select board: “Arduino/Genuino Uno” and click on the Boards Manager as shown in the figure

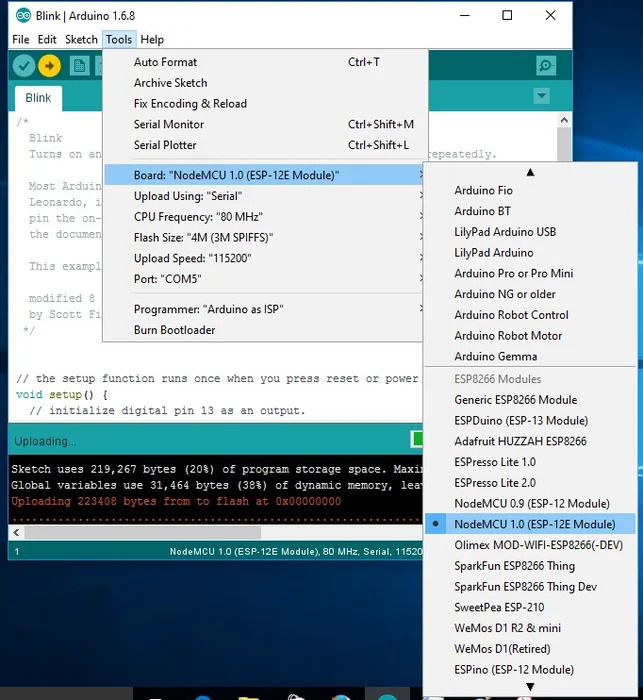
**Step 7**: ESP8266 Board Package

The Boards Manager window opens, scroll the window page to bottom till you see the module with the name ESP8266. Once we get it, select that module and select

version and click on the Install button. When it is installed it shows Installed in the module as shown in the figure and closes the window.

****

**Step 8:** Selecting ESP8266 Arduino Board

To run the esp8266 with Arduino we have to select the Board: “Arduino/Genuino Uno” with Node-MCU 1.0 (ESP-12E Module) by scrolling down, as shown in the figure

**Step 9:** Connecting ESP8266 to the PC

Now let’s connect the ESP8266 module to your computer through USB cable as shown in the figure. When module is connected to the USB, COM port is detected ex:

Here COM5 shown in the figure.



**Step 10:** Selecting Program in Arduino IDE

Now open the File tab in that, go to the New file, a empty file opens, Now enter the code and the debug and upload it to Board.

* + 1. **IFTTT (IF THIS THEN THAT)**

Fig. 4.3 IFTTT

* IFTTT allows you to do more with over 700 different apps and services, including Twitter, Dropbox , Ever note, Fit bit, Amazon Alexa , and Google Assistant
* On IFTTT, we call these services. A list of all services on IFTTT can be found here
* We bring services together into Applets, automations that allow you to do things your apps and devices can't do on their own.
* IFTTT is short for 'If This Then That', and is pronounced like 'Gift' without the 'G'.
* We used to be called 'if this, then that' because Applets would have one trigger and one action. If this happens — then that happens.

# HOW DOES IFTTT WORK?

IFTTT helps connect different apps and devices. When you sign up for a free account, you can enable your apps and devices to work together to do things they couldn't otherwise do. 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.

1. Here's how it works:
2. Create a free account.
3. Browse the IFTTT website or app to find something that interests you.
4. Connect the services that are involved in the Applet or connection.
5. Find more Applets and connections, and repeat!
6. Applets and connections can be built by services or users from the Developer Dashboard. You also have the power to create something custom for yourself! Go to ifttt.com/create to combine two services and make your own Applet!
   1. **CREATION OF APPLETE IN IFTTT**

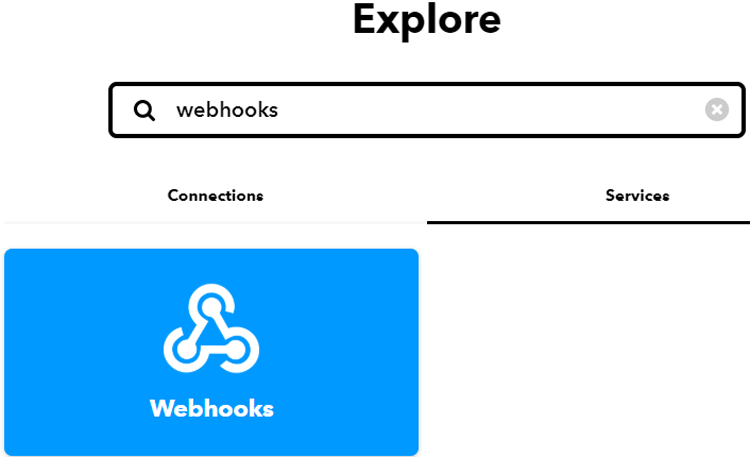
* First login to IFTTT and search for ‘Web hooks’.
* Now to get the Private key, click on ‘Documentation’. Copy this key somewhere; it will be used in the code.

Fig. 4.4 Web hooks & key

* we have the private key, we will create an applet using Web hooks and Email services. To create an applet , click on your profile and then click on **‘Create’** from available options.
* Now in the next window, select If This Then That.

 Fig. 4.5 Creation of IFTTT

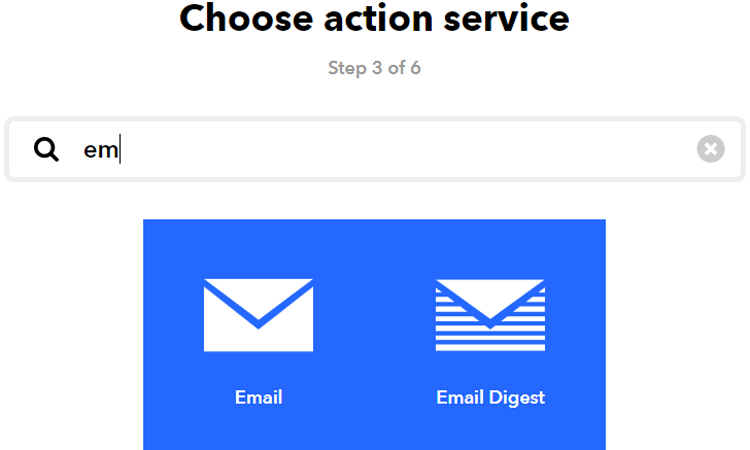
* In ‘This’ field we will use web hooks to get the web requests from the Node-MCU.
* Now select **‘Receive a Web Request’**trigger and then enter the event name as **jar event**and then click on **‘Create Trigger.**
* After this, click on ‘Then that’ and then click on Email.
* Now in Email, click on ‘send me an email’ and enter the email subject and body and then click on create action.

Fig. 4.6 Email in Web hooks

* In the last step, click on ‘Finish’ to complete the Applet setup.



Fig. 4.7 Event creation

* 1. **PROGRAMMING CODE EXPLATION**

We are using Arduino IDE to program Node-MCU. So, make sure you have downloaded Node-MCU board files. Complete code is given at the end of the document. Here we are explaining the code step by step.

So start your code by including all the required library files. The ultrasonic sensor doesn’t require a library file, so we only need ESP8266WiFi.h library file.

#include <ESP8266WiFi.h>

After that, define the pins where you connected the Trig and Echo pins ,also LM35 pin out/input and also define two variables for calculating distance and duration.

const int trigPin = D5;

const int echoPin = D6;

float temp\_celsius = 0;

float temp\_fahrenheit = 0;

long duration;

int distance;

After that, make instances for Wi-Fi name, Wi-Fi password, IFTTT hostname, and private key.Here is the Hotspot name with password should me given in code , with privet key , that we get in documentation.

const char\* ssid = "Wi-Fi Name";

const char\* password = "Password";

const char \*host = "maker.ifttt.com";

const char \*privateKey = "Private key";

Now to access the Wi-Fi Server, we declared an object WifiServer library. 80 is the default port for HTTP.

WiFiServer server(80);

Now inside the *void loop* function, calculate the time between triggered and received signal. Also the temperature values and there calculation this time will be used to calculate the distance.

duration = pulseIn(echoPin, HIGH);

distance = duration \* 0.0340 / 2;

After that, distance is converted into a percentage to show the Jar’s occupancy.

level =((14-distance)/14.0)\*100;

temp\_celsius = (analogRead(A0) \* 330.0) / 1023.0; // To convert analog values to Celsius We have 3.3 V on our board and we know that output voltage of LM35 varies by 10 mV to every degree Celsius rise/fall. So , (A0\*3300/10)/1023 = celsius

temp\_fahrenheit = temp\_celsius \* 1.8 + 32.0;

Then we compared the jar’s occupancy, and if the occupancy level is less than 5, then it will trigger an IFTTT event to send warning Email.

if ( level <= 10) {

send\_event("jar\_event");

}

* 1. **CODE**

#include <ESP8266WiFi.h>

const int trigPin = D5;

const int echoPin = D6;

float temp\_celsius = 0;

float temp\_fahrenheit = 0;

long duration;

int distance;

float level;

const char\* ssid = "vivo 1901";

const char\* password = "78907890";

void send\_event(const char \*event);

const char \*host = "maker.ifttt.com";

const char \*privateKey = "cOQOncyaPtL20voA\_FZ86h";

WiFiServerserver(80);

void setup() {

pinMode(trigPin, OUTPUT);

pinMode(echoPin, INPUT);

pinMode(A0, INPUT);

Serial.begin(9600);

Serial.print("Connecting to Wifi Network");

Serial.println(ssid);

WiFi.begin(ssid, password);

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

delay(500);

Serial.print(".");

}

Serial.println("");

Serial.println("Successfully connected to WiFi.");

Serial.println("IP address is : ");

Serial.println(WiFi.localIP());

server.begin();

Serial.println("Server started");

}

void loop() {

digitalWrite(trigPin, LOW);

delayMicroseconds(2);

digitalWrite(trigPin, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin, LOW);

duration = pulseIn(echoPin, HIGH);

distance = duration \* 0.0340 / 2;

Serial.println("Distance");

Serial.println(distance);

level =((14-distance)/14.0)\*100;

Serial.println("level");

Serial.println(level);

delay(1000);

temp\_celsius = (analogRead(A0) \* 330.0) / 1023.0; // To convert analog values to Celsius We have 3.3 V on our board and we know that output voltage of LM35 varies by 10 mV to every degree Celsius rise/fall. So , (A0\*3300/10)/1023 = celsius

temp\_fahrenheit = temp\_celsius \* 1.8 + 32.0;

Serial.print(" Temperature = ");

Serial.print(temp\_celsius);

Serial.print(" Celsius, ");

Serial.print(temp\_fahrenheit);

Serial.println(" Fahrenheit");

WiFiClient client = server.available();

if (client)

{

Serial.println("Web Client connected ");

String request = client.readStringUntil('\r');

client.println("HTTP/1.1 200 OK");

client.println("Content-Type: text/html");

client.println("Connection: close"); // the connection will be closed after completion of the response

client.println("Refresh: 10"); // update the page after 10 sec

client.println();

client.println("<!DOCTYPE HTML>");

client.println("<html>");

client.println("<style>html { font-family: Cairo; display: block; margin: 0px auto; text-align: center;color: #333333; background-color: ##f3ffee;}");

client.println("body{margin-top: 50px;}");

client.println("h1 {margin: 50px auto 30px; font-size: 50px; text-align: center;}");

client.println(".side\_adjust{display: inline-block;vertical-align: middle;position: relative;}");

client.println(".text1{font-weight: 180; padding-left: 5px; font-size: 50px; width: 170px; text-align: left; color: #3498db;}");

client.println(".data1{font-weight: 180; padding-left: 1px; font-size: 50px;color: #3498db;}");

client.println(".data{padding: 1px;}");

client.println("</style>");

client.println("</head>");

client.println("<body>");

client.println("<div id=\"webpage\">");

client.println("<h1>IoT Based Jar</h1>");

client.println("<div class=\"data\">");

client.println("<div class=\"side\_adjust text1\">Status:</div>");

client.println("<div class=\"side\_adjust data1\">");

client.print(level);

client.println("<div class=\"side\_adjust text1\">% filled</div>");

client.println("</div>");

client.println("</div>");

client.println("</body>");

client.println("</html>");

client.println("HTTP/1.1 200 OK");

client.println("Content-Type: text/html");

client.println("Connection: close"); // the connection will be closed after completion of the response

client.println("Refresh: 10"); // update the page after 10 sec

client.println();

client.println("<!DOCTYPE HTML>");

client.println("<html>");

client.print("<p style='text-align: center;'><span style='font-size: 50px;'><strong>Digital Thermometer</strong></span></p>");

client.print("<p style='text-align: center;'><span style='color: #0000ff;'><strong style='font-size: 50px;'>Temperature (\*C)= ");

client.println(temp\_celsius);

client.print("<p style='text-align: center;'><span style='color: #0000ff;'><strong style='font-size: 50px;'>Temperature (F) = ");

client.println(temp\_fahrenheit);

client.print("</p>");

client.println("</html>");

//delay(5000);

//client.println("<h1>Level Indicator</h1>");

if ( level<= 10) {

send\_event("jar\_event");

}

}

}

void send\_event(const char \*event)

{

Serial.print("Connecting to ");

Serial.println(host);

// Use WiFiClient class to create TCP connections

WiFiClient client;

const int httpPort = 80;

if (!client.connect(host, httpPort)) {

Serial.println("Connection failed");

return;

}

// We now create a URI for the request

String url = "/trigger/";

url += event;

url += "/with/key/";

url += privateKey;

Serial.print("Requesting URL: ");

Serial.println(url);

// This will send the request to the server

client.print(String("GET ") + url + " HTTP/1.1\r\n" + "Host: " + host + "\r\n" + "Connection: close\r\n\r\n");

while(client.connected())

{

if(client.available())

{

String line = client.readStringUntil('\r');

Serial.print(line);

} else {

// No data yet, wait a bit

delay(50);

};

}

Serial.println();

Serial.println("closing connection");

client.stop();

}

**CHAPTER 5**

**SYSTEM CONFIGURATION & IMPLEMENTATION**

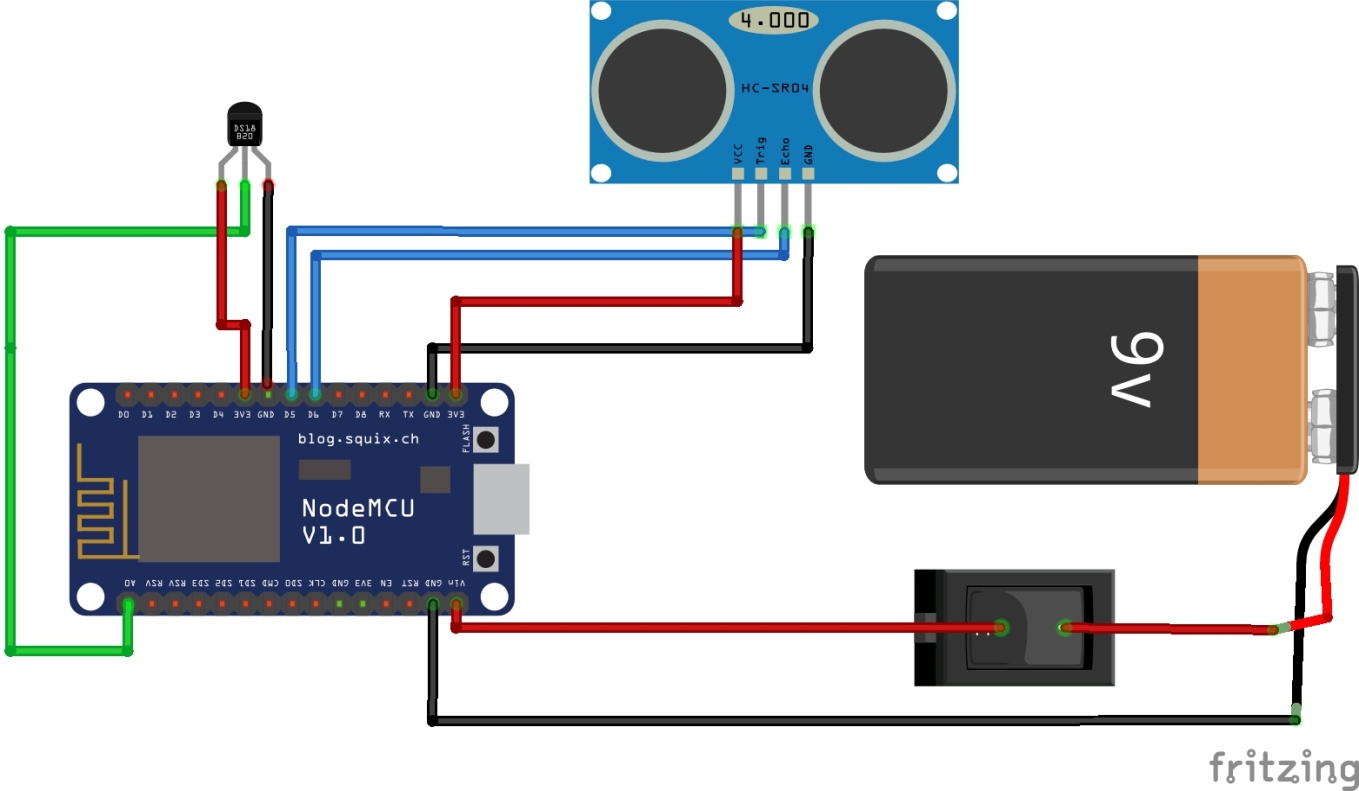
* 1. **CIRCUIT DIAGRAM**

Fig. 5.1 Circuit Diagram Of Project

Here is the interconnection configuration of the system. The Sensors, Power supply and Loads, are the above figure shows where to connect and how to connect. It is just the connections how are made on to the Board (NODE MCU), but reality is that this varies by container sizes and there zones. This all setup is fixed at back of container.

So that the circuit connection can’t visible and also occupies unwanted space of the container.

* 1. **INTERCONNECTIONS DEATAILS** 
     1. **ULTRASONIC SENSOR**
* Vcc to the 3V3 pin of Node-MCU
* Trigger to the D5 pin of Node-MCU
* Echo to the D6 pin of Node-MCU
* GND to the GND pin of Node-MCU
  + 1. **LM35 TEMPERATURE SENSOR**
* Vcc (1st Pin) to the 3ve pin of Node-MCU
* Output (2nd Pin) to the A0 (ADC) pin of Node-MCU
* GND (3rd pin) to the GND pin of Node-MCU
  + 1. **POWER SUPPLY**
* Here ,we going to give Power supply by 9V battery with switch
* Positive terminal of battery is given to switch
* From switch to Vin Pin of Node-MCU
* Another terminal of battery (GND) is directly given to GND pin of Node-Mcu
  1. **FLOWCHART**

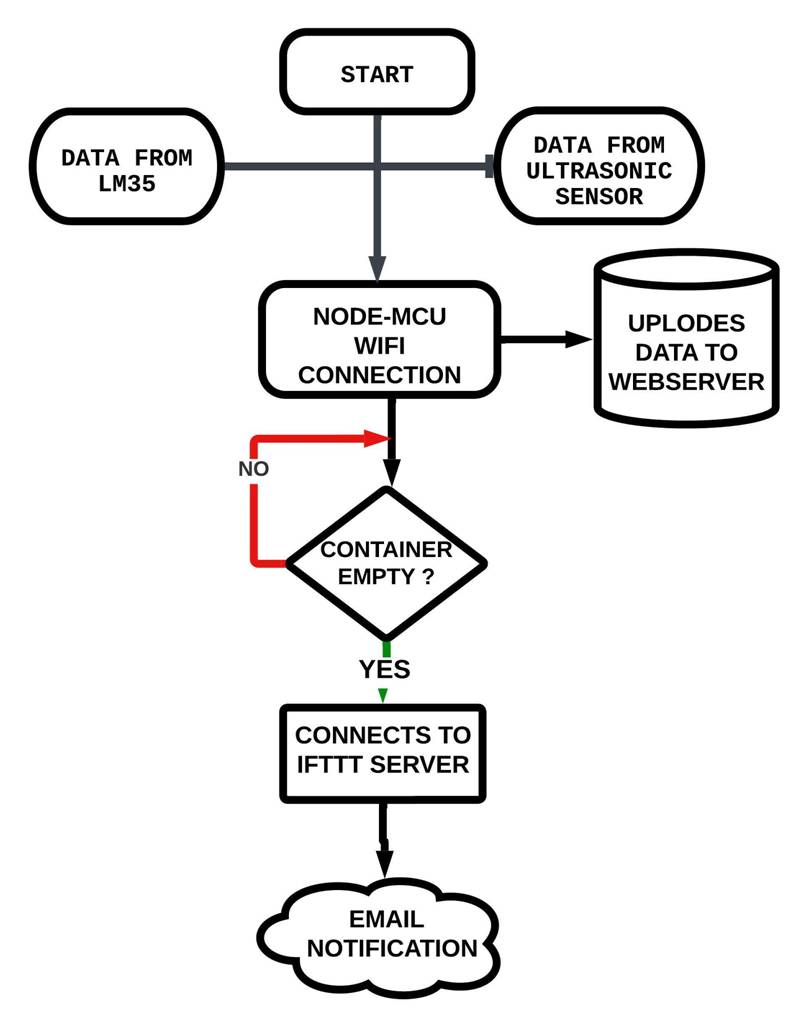
****

Fig. 5.2 Flowchart

* 1. **WORKFLOW**
* It starts from collecting data from Ultrasonic sensor and LM35 sensor , That data is transfer to Node-MCU .Meanwhile the data collection task also happens in Node-MCU that conversion or any calculations needed to take place.
* All the Data transfers to Node-MCU then it connect to Wi-Fi router. As we mentioned that Wi-Fi name and password need to be given in code.
* Now there is internet connection to Node-MCU from Wi-Fi ,so the data will upload to Web Serve to a dynamical IP address.
* Every time the Ip address is changes as it is Dynamical .So you need to find it by Arduino IDE serial monitoring.
* Continuously it uploads data to server for every 10 second of time. Whenever the given condition is satisfied the respected action takes place, that fallowed by program code
* Here is a decision maker by why which the output depends on level of the container filled i.e. Ultrasonic Sensor value.
* If the level is 0 % then it goes connect the IFTT server, from there we get a email Notification .That your container is empty. Please refill it.
* If the level is more than 10% the loop repeats again.
* Even you can see data in any web server as it is updated information in server
  1. **RESULTS**
     1. **CASE -1**

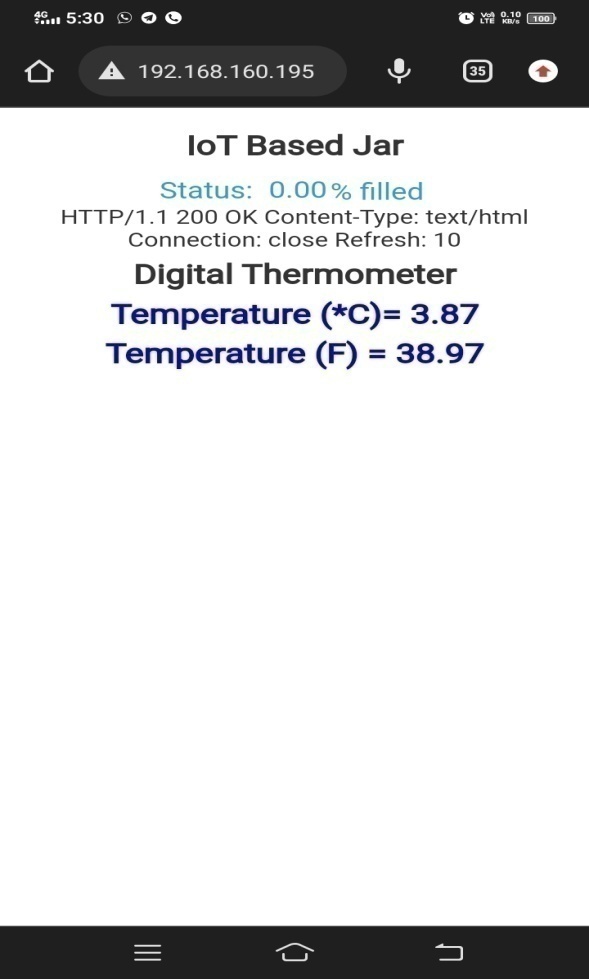
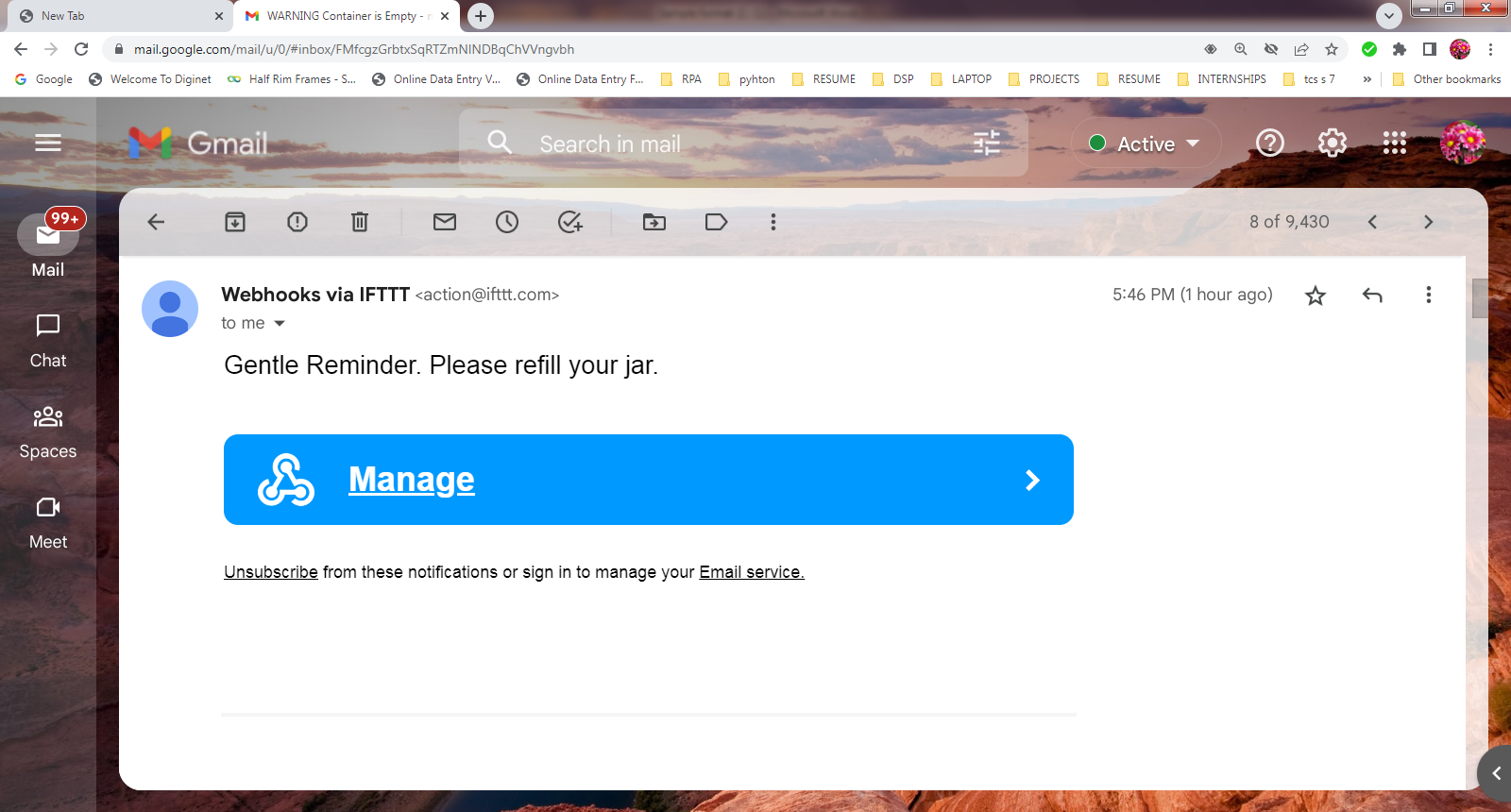
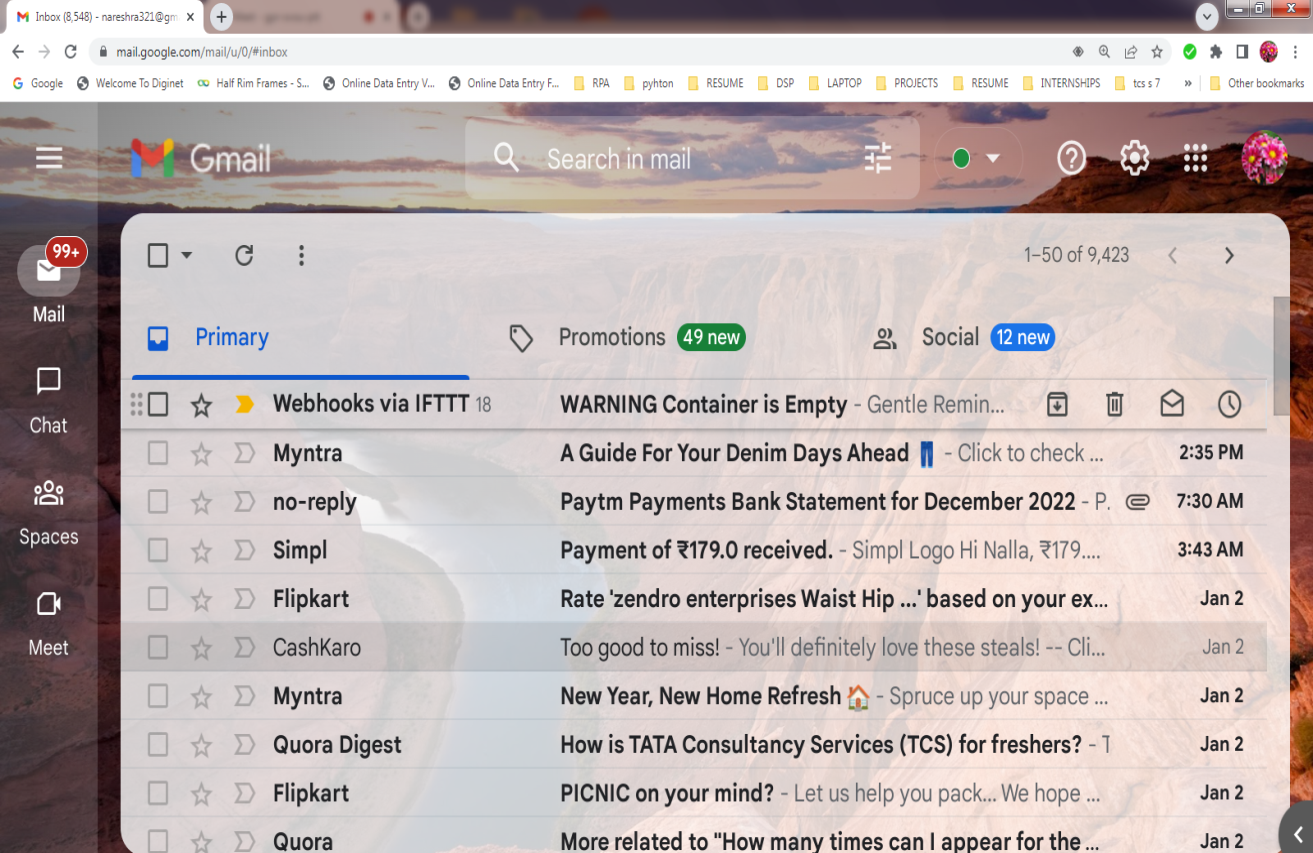
The container is empty i.e. No content is present in container, so the output will be as shown in figure.

Fig. 5.3 Empty Container Fig. 5.4 Web server Output

Here there is no objects present in container .The waves transmitted from ultrasonic sensor is reached at the end of container, So the percentage filled output should be 0 % in web server

Also the condition where we given that, if the container is empty or a given criteria is satisfied, then it should be connect to IFTTT server and Notifies to the mail that “**Container is Empty Please Refill It**”. As it can be notified in the below shown figure.

 Fig. 5.5 Email Notification

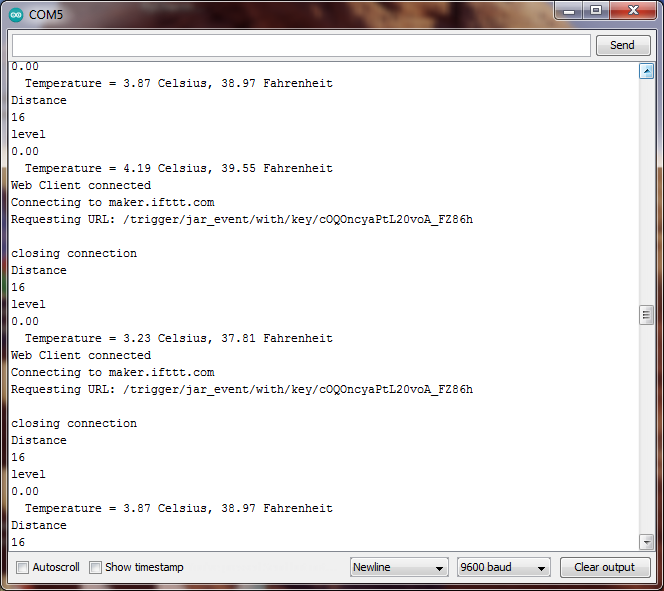
Fig. 5.6 Email Notification Content

Fig. 5.7 IFTTT Connecting server

* + 1. **CASE-2**

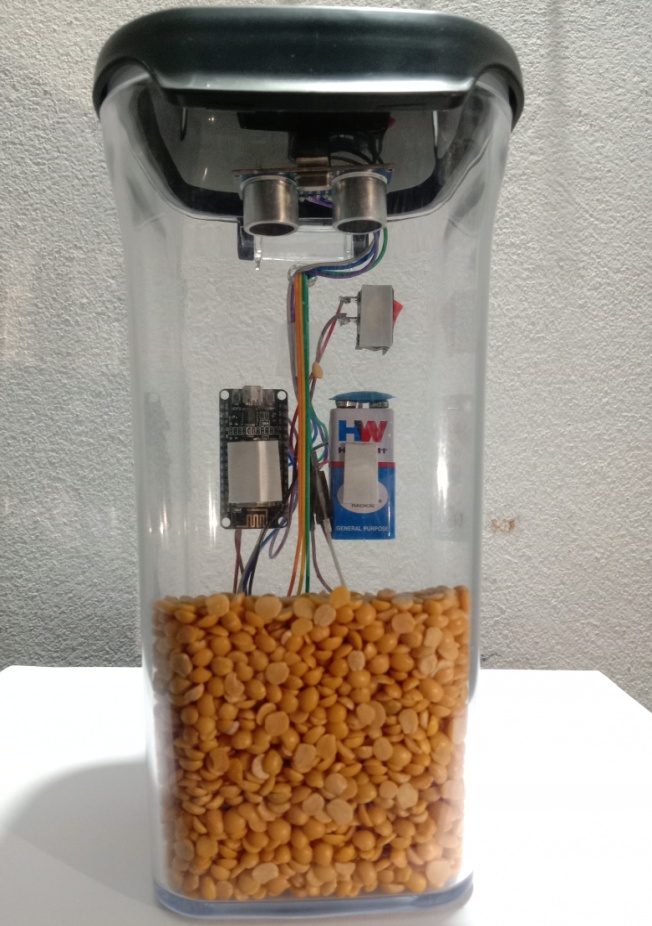
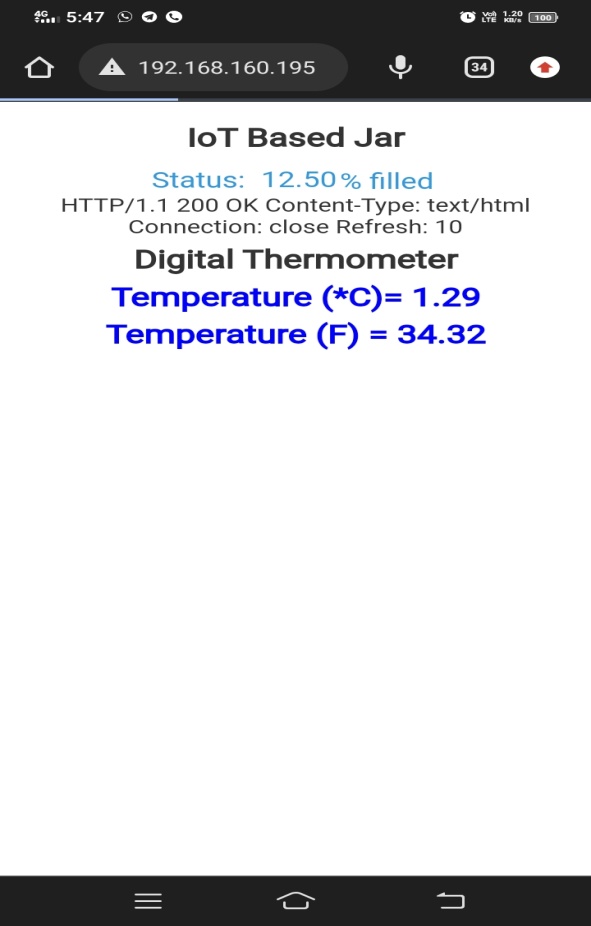
The container is partially filled with some amount of content.So the output will be as shown in figure

Fig. 5.8 Partially Filled Container Fig. 5.9 Web Server Output

Here there is a objects present in container .The waves transmitted from ultrasonic sensor is reached at the top of content present in container, So the percentage filled in container will be output should be based on the calculation of ultrasonic sensor waves and that percentage value will be updated in web server

Also the condition where we given that, if the given criterion is satisfied, then it notifies to the mail .If the given criteria are not satisfied then the given loop repeated again and again.

* 1. **ADVANTAGES**

1. We tract the percentage of filled amount in container without presence of him/her.
2. With the help of Web server we can accessible anywhere any time
3. It consumes less space for circuit connection.
4. As major components are 2 , so power consumption also less.
5. By the Smart jar the house will be more modern smart house.
6. It adds a comfort to the people who work long distance from home.
   1. **DISADVANTAGES**
7. All the time it need an internet connection.
8. A continuous power supply is required.
   1. **APPLICATIONS**
9. The smart container can be used to track the amount of contents of the container from anywhere.
10. It is mostly useful for those who find difficult to know the amount of conents filled in container.
11. It is useful for the busy schedule persons.
12. It can be also used in general stores.

**CHAPTER 6**

**CONCLUSION & FUTURESCOPE**

* 1. **CONCLUSION**

The IOT based Smart Jar helps to track the availability of stock from anywhere in the world and saves a lot of manual labor and also time. The quantity of the jar cautions the user and hence the user can automatically know the data without actually being there. Especially, in medical industry, where people deal with quantity y of medicine, it is more useful. It ultimately saves time and effort of the consumer in this modernized world.

* 1. **FUTURESCOPE**

The jar can be developed for multi-purpose usage such as measuring the liquid quantity of some chemical solutions in a container. It can be made completely wireless with the availability of latest technology then. Synchronizing of one or more jars of similar kind would help the user optimally access various range of stock sand buy according to availability

This technology can also be used to track the quantity of medicine for patients who suffer from diseases like diabetes and who need to have regular medicine update. This jar saves them falling short of medicine and helps them keep their medicines in the jar updated. For merchants who own general store and who are away from their shop, this technology helps them to instantly order the amount of stock required for them to order by modifying the jar formula for bigger containers. The percentage of container filled would make their work easier.

**BIBLIOGRAPHY**

* **TEXT BOOKS**

**[1]**T. K. Gannavaram V and R. Bejgam, "Brief Study and Review on the Next Revolutionary Autonomous Vehicle Technology," 2021 International Conference on Advance Computing and Innovative Technologies in Engineering(ICACITE),2021,pp.34-37

**[2]**D. Singh, G. Tripathi and A. J. Jara, "A Survey of Internet-of-Things: Future Vision Architecture Challenges and Services", Proc. IEEE World Forumon Internet of Things 2014, pp. 287-292, 2014.

[6] Sagnik Kanta, Srinjoy Jash and Himadri Nath Saha, "Internet of Things Based Garbage Monitoring System", IEEE Conference, pp. 127-130, 2017.

* **WEBSITES**

**[3]** Circuit Digest for Node-MCU,

<https://circuitdigest.com/microcontroller-projects/iot-based-smart-jar-using-nodemcu-esp8266-and-ultrasonic-sensor>

**[4]** Electronic Clinic for programming <https://www.electroniclinic.com/nodemcu-esp8266-web-browser-programming-and-circuit-diagram>

**[5]** Circuit Digest for LM35 sensor [https://circuitdigest.com/microcontroller-projects/iot-digital-thermometer-using-nodemcu-esp12-and-lm35](Music)