# IoT BASED PRESSURE COOKER AUTOMATION

by

**Tiruveedhula Uday-19BEC1452**

**Keerthivasan J-19BLC1045**

**Rufus Sam John Immanvel J-19BLC1031**

A project report submitted to

**Dr. BERLIN HENCY V**

**SCHOOL OF ELECTRONICS ENGINEERING**

in partial fulfilment of the requirements for the course of

**ECE3501 – IoT FUNDAMENTALS**

In

**B.Tech. ELECTRONICS AND COMMUNICATION ENGINEERING**

**Vandalur – Kelambakkam Road**

**Chennai – 600127**

**DECEMBER 2021**

# BONAFIDE CERTIFICATE

Certified that this project report entitled "**IoT BASED PRESSURE COOKER AUTOMATION**" is a bonafide work of **RUFUS SAM JOHN IMMANVEL J-19BLC1031, KEERTHIVASN J-19BLC1045 & TIRUVEEDHULA UDAY-19BEC1452** who carried out the Project work under my supervision and guidance for **ECE3501-IoT Fundamentals**

**Dr. BERLIN HENCY V**

School of Electronics Engineering (SENSE)

VIT University, Chennai

Chennai – 600127.

# ABSTRACT

Our project titled **IoT Based Pressure Cooker Automation** emphasizes the need to improve and prioritize the reduction of wastage of food due to overcooking. This project aims at developing an automatic system which would turn off the gas as soon as the required number of whistles is reached.

A sound sensor is used to detect whistle sound and a NodeMCU is used to send the data to the mobile application using Wi-Fi module ESP8266. It also counts the number of whistles and controls the DC motor accordingly. The DC motor is used to rotate the gas stove knob, in order to turn it off when the required number of whistles are reached.

The result which we hope to obtain upon completion of this project is a system that will automatically turn off the gas as soon as the required number of whistles is reached. The results of this experiment can be validated from home.

# ACKNOWLEDGEMENT

We wish to express our sincere thanks and deep sense of gratitude to our project guide, **Dr.Berlin Hency V, Associate Professor Senior, School of Electronics Engineering,** for his consistent encouragement and valuable guidance offered to us in a pleasant manner throughout the course of the project work.

We are extremely grateful to **Dr.Sivasubramanian. A, Dean of School of Electronics Engineering,** VIT Chennai, for extending the facilities of the school towards our project and for her unstinting support.

We express our thanks to our **Head of the Department Dr.Vetrivelan. P,** for his support throughout the course of this project.

We also take this opportunity to thank all the faculty of the school for their support and their wisdom imparted to us throughout the course.

We thank our parents, family, and friends for bearing with us throughout the course of our project and for the opportunity they provided us in undergoing this course in such a prestigious institution.

Ruf. J. Keerthivasan T. uday

RUFUS SAM KEERTHIVASAN J TIRUVEEDHULA UDAY

# TABLE OF CONTENTS

# INTRODUCTION

## 1. OBJECTIVES AND GOALS

- To design an IoT based system that would automatically turn off the knob of a gas stove based on the number of whistles detected.
- The system should notify the user.

## 2. APPLICATIONS

- To Turn Off the Pressure cooker's stove when the number of whistles is predetermined by the user.
- The system would send notifications and email to the user.

## 3. FEATURES

- The User sets the whistle count using the Blynk app in mobile which will be then shown as notification.
- The sound sensor will then monitor the pressure cooker for its sound, and then it persists for a period of time it will then notify the NodeMCU and reduce the count of the whistles from the initialized count .
- When the counter reaches zero it tuns off the stove using the DC gearmotor.
- It will indicate that the stove is turned off in the mobile through a notification from Blynk app that the "Food is ready".

## 1.3 LITERATURE SURVEY

### PAPER 1: Automatic Gas Controller

AUTHORS: R. Dhianeswar, S. Sumathi, K. L. Joshitha

SUMMARY:

- This paper deals with solving the problem of memory loss in the kitchen which is faced by many people(especially elders).
- The paper is about developing an automatic gas controller, which cuts the supply of gas when the person is not present at the environment.
- This project will be a solution to many people who forget to turn off the gas stove and prevents fire accidents and wastage of food and gas.

## PAPER 2: Implementation of automatic safety gas stove

AUTHORS: A. Yalmar, M. Parihar, V. Kadam, K. Kharat.

SUMMARY:

- This research paper deals with automatic safety gas stove that concentrates on additional safety as a fundamental argument towards the usage of domestic LPG. The paper focuses on modes developed to automatically prevent any kind of threat from usage of the gas stove. This project is AT89C51 microcontroller based embedded application which deals only with turning OFF mechanism of kitchen gas stove in various situations. It turns off the gas when it is not in use.
- The project is a precise division working in three modes namely the detection mode, the timer mode and the remote access mode. The detection of the utensil is done by an IR sensor placed near the burner of the stove. Timer mode turns off the gas after a particular interval of time and a remote access mode can turn off the stove knob using the cellular mobile phone.

## PAPER 3: Pressure Cooker Whistle Notification Using Arduino Based Wireless Sensor Networks

AUTHORS: JK Adarsh, R Kishore,˒ R Arul

SUMMARY:

- The sound sensor input consists of sound waves as a frequency and sends them to the microcontroller via one of the pins. The microcontroller then processes the received signal and searches for the frequency range of the tone.

- The gadget detects the pressure cooker whistle which falls in the frequency range and sends the notification directly to the user's smartphone and also the whistle count is updated in real-time via WiFi.

### PAPER 4: Cooker Automation

AUTHORS: **sourav, Sounak Mondal**

SUMMARY:

- Whenever the cook is not around, this device will record the number of times the pressure cooker whistles using a sound sensor. And whenever that number matches with predefined number, this device will start beeping thus alerting the user.
- Whenever the pressure cooker whistles, the counter increases. And if it matches with the preset number (we set before using the push button switch), the buzzer will get triggered and alert the user that desired count has been reached.

### PAPER 5: Design and Analysis of Smart Whistle for Pressure Cooker

AUTHORS: Prathamesh Pandit , Harshkumar Patel , Jayesh Bhoir , Prashant Sathe , Ashish J. Chaudhari

SUMMARY:

- The present study comprises of mechanical pressure regulator device for pressure cookers. This system utilizes the vertical motion of dead weight of pressure regulator during whistling.
- The system also alerts the user about the end of the cooking process after the number of whistles as set by the user for cooking has been attained. Some distinctive whistling sound for the last whistle of the process could be heard that occurs by manipulating the direction and path of steam flow from cooker to surrounding.
- The important feature of this system that it stops the pressure built up after the end of cooking process. This has been achieved by manipulating the path and direction of steam flow from cooker to surrounding. The major advantage is the pressure whistle line blockages could never happened with this system.

# 2.DESIGN

## 2.1 FUNCTIONAL COMPONENTS

### 2.1.1 Functional Sensors Incorporated in System

1. Sound Sensor: A sound sensor is defined as a module that detects sound waves through its intensity and converting it to electrical signals.

### 2.1.2 Importance of Sensors mentioned

When we constantly detect the sound from the cooker in order not to miss any whistle, otherwise it may lead to wastage of food.

## 2.2 BLOCK DIAGRAM

Following is the block diagram of our System. There are four main parts in this system: things, actuators, gateway and an application layer. Things include sensors that we have incorporated into our system namely: Sound sensor while our board includes Node MCU. Our main Gateway would be Wi-Fi. Application layer of our system includes App/Control Room. Actuators: Motor driver and motor to manually turn off the stove.

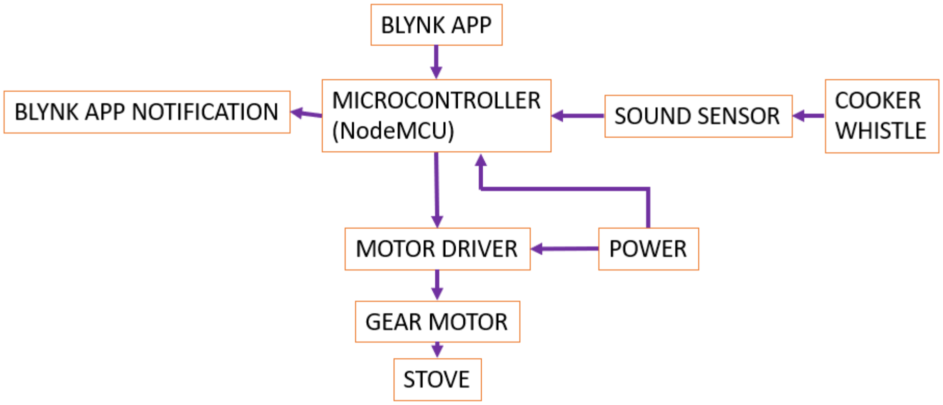


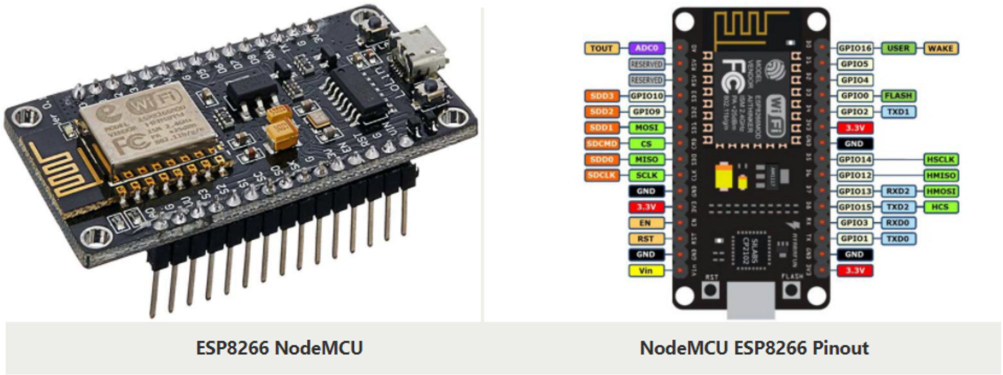
Fig 1: Block diagram of Pressure cooker automation system

## 2.3 CIRCUIT DIAGRAM

## 2.4 HARDWARE ANALYSIS

### Node MCU

NodeMCU is an open-source Lua based firmware and development board specially targeted for IoT based Applications. It includes firmware that runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which is based on the ESP-12 module.



ESP8266 NodeMCU

NodeMCU ESP8266 Pinout

Fig 3: NodeMCU development board

### NodeMCU Development Board Pinout Configuration

| PIN CATEGORY | NAME | DESCRIPTION |
|---|---|---|
| Power | Micro-USB, 3.3V, GND, Vin | **Micro-USB:** NodeMCU can be powered through the USB port.<br>**3.3V:** Regulated 3.3V can be supplied to this pin to power |

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

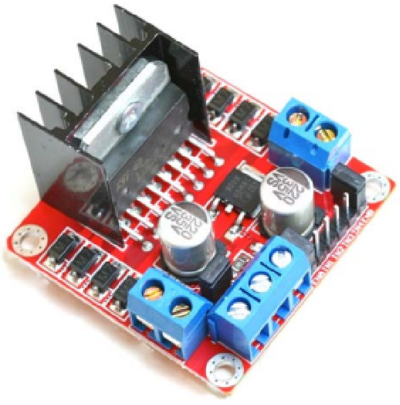
## NodeMCU ESP8266 Specifications & Features

- Microcontroller: Tensilica 32-bit RISC CPU Xtensa LX106
- Operating Voltage: 3.3V
- Input Voltage: 7-12V
- Digital I/O Pins (DIO): 16

- Analog Input Pins (ADC): 1
- UARTs: 1
- SPIs: 1
- I2Cs: 1
- Flash Memory: 4 MB
- SRAM: 64 KB
- Clock Speed: 80 MHz
- USB-TTL based on CP2102 is included onboard, Enabling Plug n Play
- PCB Antenna
- Small Sized module to fit smartly inside your IoT projects

### 2.4.2 L298 dual H- bridge motor driver

This dual bidirectional motor driver, is based on the very popular L298 Dual H-Bridge Motor Driver Integrated Circuit. The circuit will allow you to easily and independently control two motors of up to 2A each in both directions.It is ideal for robotic applications and well suited for connection to a microcontroller requiring just a couple of control lines per motor. It can also be interfaced with simple manual switches, TTL logic gates, relays, etc. This board equipped with power LED indicators, on-board +5V regulator and protection diodes.



**Pin out configuration**

**Specifications**

- Driver Model: L298N 2A
- Driver Chip: Double H Bridge L298N
- Motor Supply Voltage (Maximum): 46V
- Motor Supply Current (Maximum): 2A
- Logic Voltage: 5V
- Driver Voltage: 5-35V
- Driver Current:2A
- Logical Current:0-36mA
- Maximum Power (W): 25W
- Current Sense for each motor
- Heatsink for better performance
- Power-On LED indicator

**Dual shaft BO motor 100 RPM**

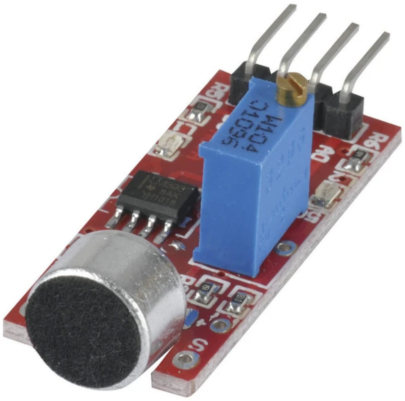
The 100 RPM Dual Shaft BO Motor - Straight motor gives good torque and rpm at lower operating voltages, which is the biggest advantage of these motors. Small shaft with matching wheels gives an optimized design for your application or robot. Mounting holes on the body & light weight makes it suitable for in-circuit placement. This motor can be used with 69mm Diameter Wheel for Plastic Gear Motors. It is an alternative to our metal gear DC motors. It comes with an operating voltage of 3-12V and is perfect for building small and medium robots. The motor is ideal for DIY enthusiasts. This motor set is inexpensive, small, easy to install, and ideally suited for use in a mobile robot car. They are commonly used in our 2WD platforms.

**Specifications**

- Shaft length: 7 mm
- Motor Design: Straight Dual Shaft
- Shaft Diameter: 5.5 mm
- Size: 55 x 48 x 23 mm.
- Operating Voltage: 3 to 12V.
- Current (without loading): 40-180mA.
- RPM: 100 rpm.
- Output Torque: 0.35 kg cm.

### <u>Sound Detection Sensor Module (KY-037)</u>

Sound Detection Module Sensor for Intelligent Vehicle Compatible With Arduino is a Single channel signal output Sensor. The output is effective to the low-level sound signal with good fidelity, When there is sound, outputs low level and signal light. It can be used for Acoustic control light, give sound and light alarm working with the Photosensitive sensor, and sound control, sound detect. (The Maximum induction distance is 0.5M).



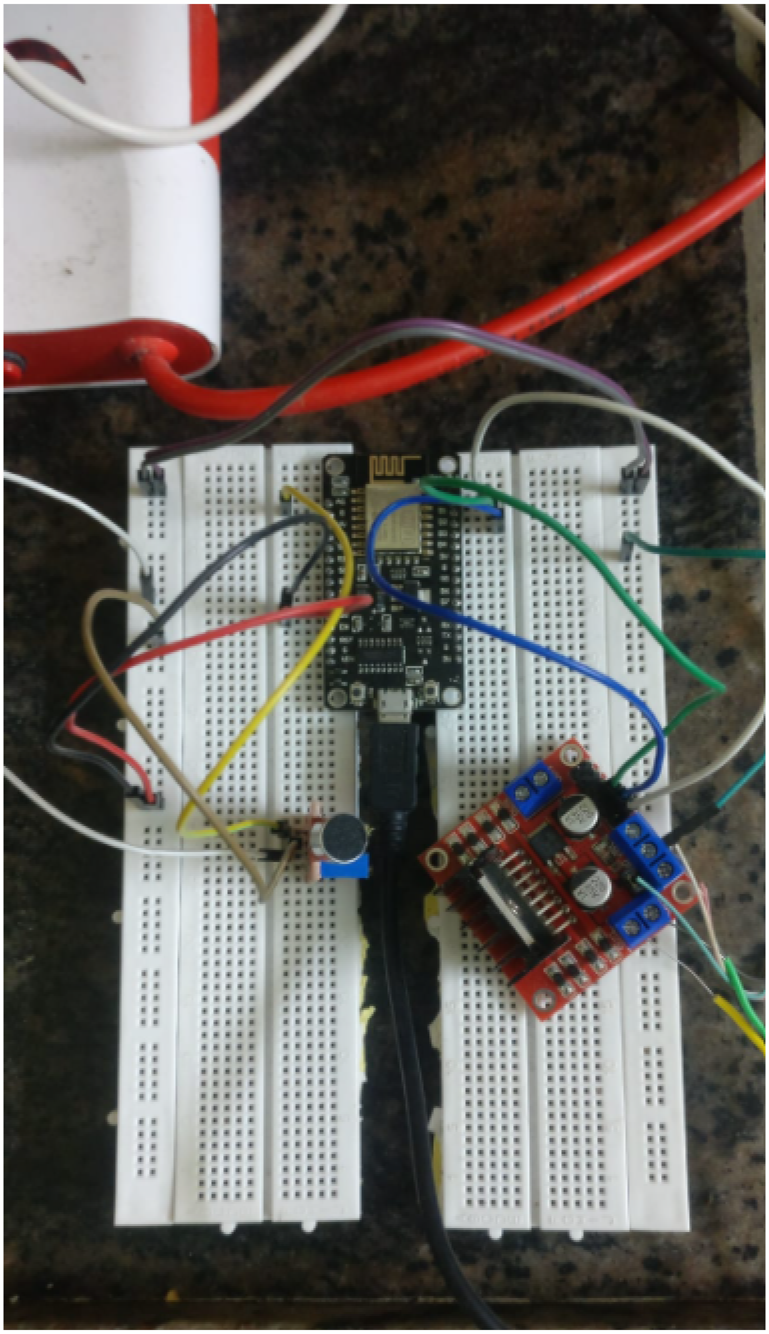
### Pin out configuration

### Specifications

- IC Chip: LM393
- Operating Voltage(VDC): 3.3 to 5V
- Induction Distance:0.5 Meter
- Length (mm): 43
- Width (mm): 17
- Height (mm): 8
- Weight (gm):3

## 2.5 HARDWARE IMPLEMENTATION

This whole system consists of the sound sensor and motor driver as sensor and actuator. Our board in this system is Node MCU. Motor driver is connected to the BO motor.



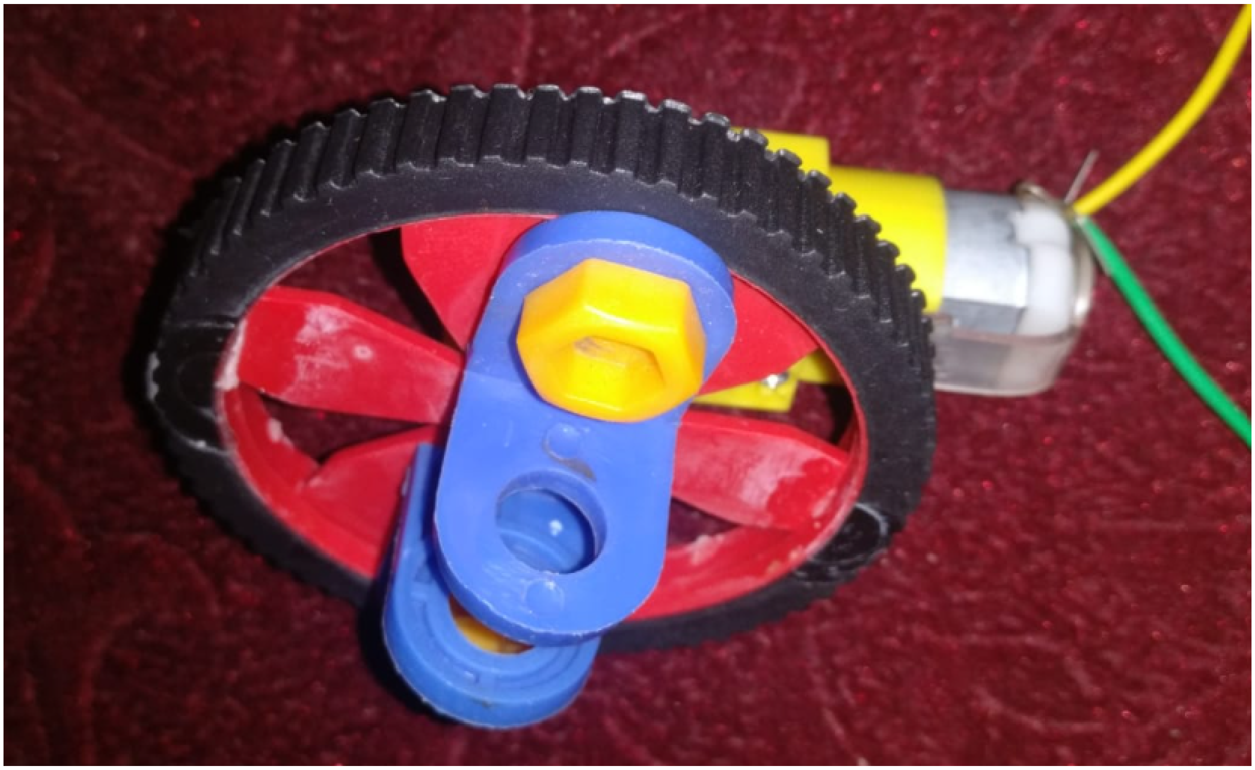


Figure 9: Hardware implementation of Pressure cooker automation.

# 3. SOFTWARE OVERVIEW

## 3.1 API KEYS USED:

**pinMode(pin, mode) -** Configures the specified pin to either input or output.

**- pin:** the Arduino pin number to set the mode of.

**- mode:** INPUT, OUTPUT, or INPUT_PULLUP.

**digitalWrite(pin, value) -** Write a HIGH or a LOW value to a digital pin**.**

**Serial.begin(speed) -** Configures a serial port with specified baud rate**.**

**-Serial:** To access serial port in Arduino Uno.

**-begin:** To configuring serial port.

**-speed:** To specify the baud rate for serial communication.

**analogRead(pin) –** Reads the value from a specified analog pin and gives value.

**millis() -** to record number of milliseconds passed since the current program began the execution.

**Serial.println(val) -** Prints data on serial port as human-readable ASCII text.

**- println:** To perform printing text followed by a newline character.

**- val:** Specify value or text to be printed on serial monitor.

**delay(ms) -** Pauses the program for the amount of time (in milliseconds)

**- ms:** the number of milliseconds to pause.

**BlynkTimer timer;**

BlynkTimer **allows you to send data periodically with given intervals not interfering with** Blynk library routines Blynk Timer inherits SimpleTimer Library, a well-known and widely used library to time multiple events on hardware.

**void setup()**

Used to initialize the required variables, pinmodes, libraries etc.

**Blynk.begin(auth, ssid, pass);**

It gives the authentication key, SSID = Wi-Fi name, Pass=password for initialising Blynk.

**Blynk.run();**

It is an essential housekeeping command that keeps the background library running.

**timer.run();**

**Blynk.virtualWrite(V1, t);**

Used to write the data using virtual pins.

## PRESSURE COOKER AUTOMATION CODE:

```
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>

BlynkTimer timer;
```

```
char auth[] = "Vz7jlXR1mpWsMSN08zOvOSnT7y8Z4MBh";


char ssid[] = "Latha";

char pass[] = "Rehobogg";


int current_whistle = 0, seconds_elapsed=0,f=1,gapend=0,gapstart=0, _minutes=0,
_seconds=0,starting_point=0;

int sound = A0;

int threshold = 200;

int k=0;

// Motor A connections

int enA = 16;

int in1 = 4;

int in2 = 5;

void setup()

{

  // Set all the motor control pins to outputs

  pinMode(enA, OUTPUT);

  pinMode(in1, OUTPUT);

  pinMode(in2, OUTPUT);



  // Turn off motors - Initial state

  digitalWrite(in1, LOW);

  digitalWrite(in2, LOW);
  Serial.begin(9600);
  pinMode(sound,INPUT);
  Blynk.begin(auth, ssid, pass);
  timer.setInterval(1000L, myTimerEvent);   // Setup a function to be called every second
}

BLYNK_WRITE(V0)
{
  int pinValue = param.asInt(); // assigning incoming value from pin V1 to a variable
  current_whistle = pinValue;
  starting_point = millis()/1000;
  Serial.println(pinValue);
}

void myTimerEvent()
{
  // Current whistle value
  Blynk.virtualWrite(V1, current_whistle);

  // time
  Blynk.virtualWrite(V2, _minutes);
  Blynk.virtualWrite(V3, _seconds);
}
```

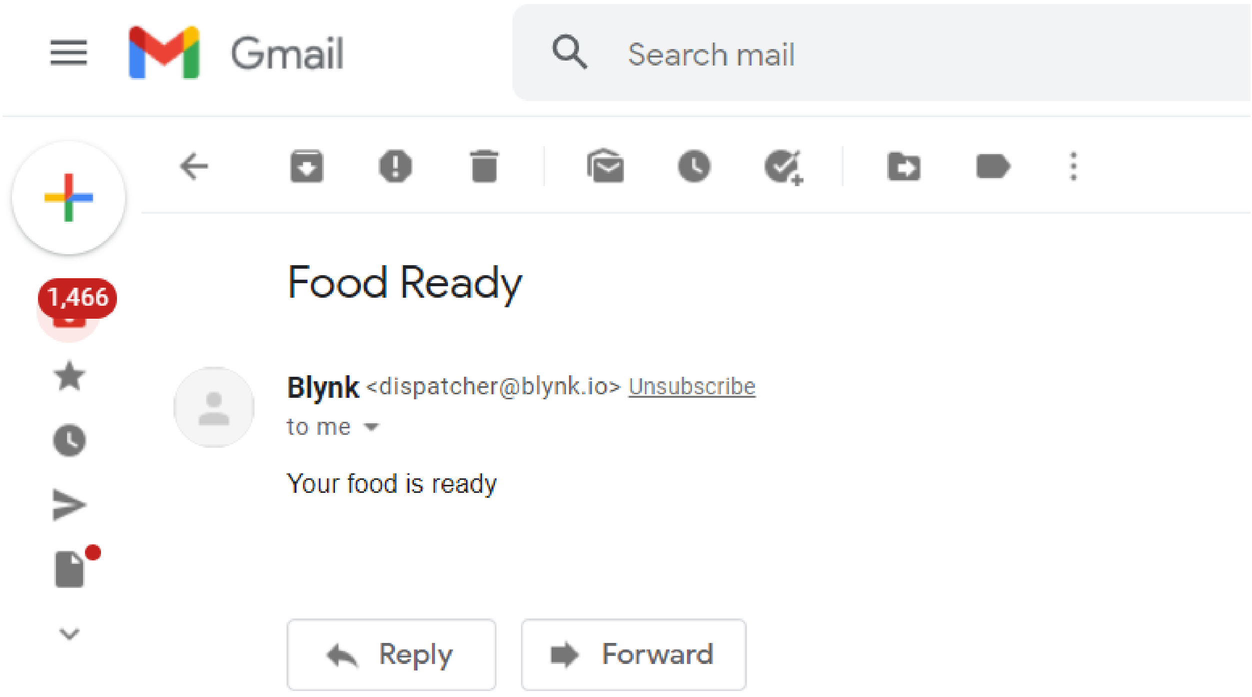
```
void trigger_motor(){
  // switch on motor to off the stove.
  digitalWrite(in1, HIGH);
  digitalWrite(in2, LOW);
  digitalWrite(enA, 100);
  delay(1000);
  digitalWrite(in1, LOW);
  digitalWrite(in2, LOW);
  digitalWrite(enA, 0);
}

void sound_trigger(){
  if(current_whistle > 0){
    current_whistle--;
    String hello = "Whistles remaining: " + String(current_whistle);
    if(current_whistle == 0){
      trigger_motor();
      Blynk.notify("Food is ready");
      Blynk.email("Food Ready","Your food is ready");
    }else{
      Blynk.notify(hello);
    }
  }
```

```
}

void check_whistle_from_sound_sensor(){
  int soundsens = analogRead(sound);
  if(soundsens>=threshold && f==1 ){
    sound_trigger();
    f=0;
    gapstart = millis();
    delay(1000);
  }else{
    gapend = millis();
    if(gapend-gapstart >= 7000) f=1;
  }
  Serial.println(soundsens);
}

void loop()
{

    Blynk.run();
    timer.run(); // Initiates BlynkTimer
    check_whistle_from_sound_sensor();
    // time elapsed
    seconds_elapsed = millis() / 1000 - starting_point;
```

```
   _seconds = seconds_elapsed % 60;

   _minutes = seconds_elapsed / 60;


  delay(100);

}
```

## 5. RESULTS

**Video demo :** https://photos.app.goo.gl/hg7ZTzhEUvGrB4VFA

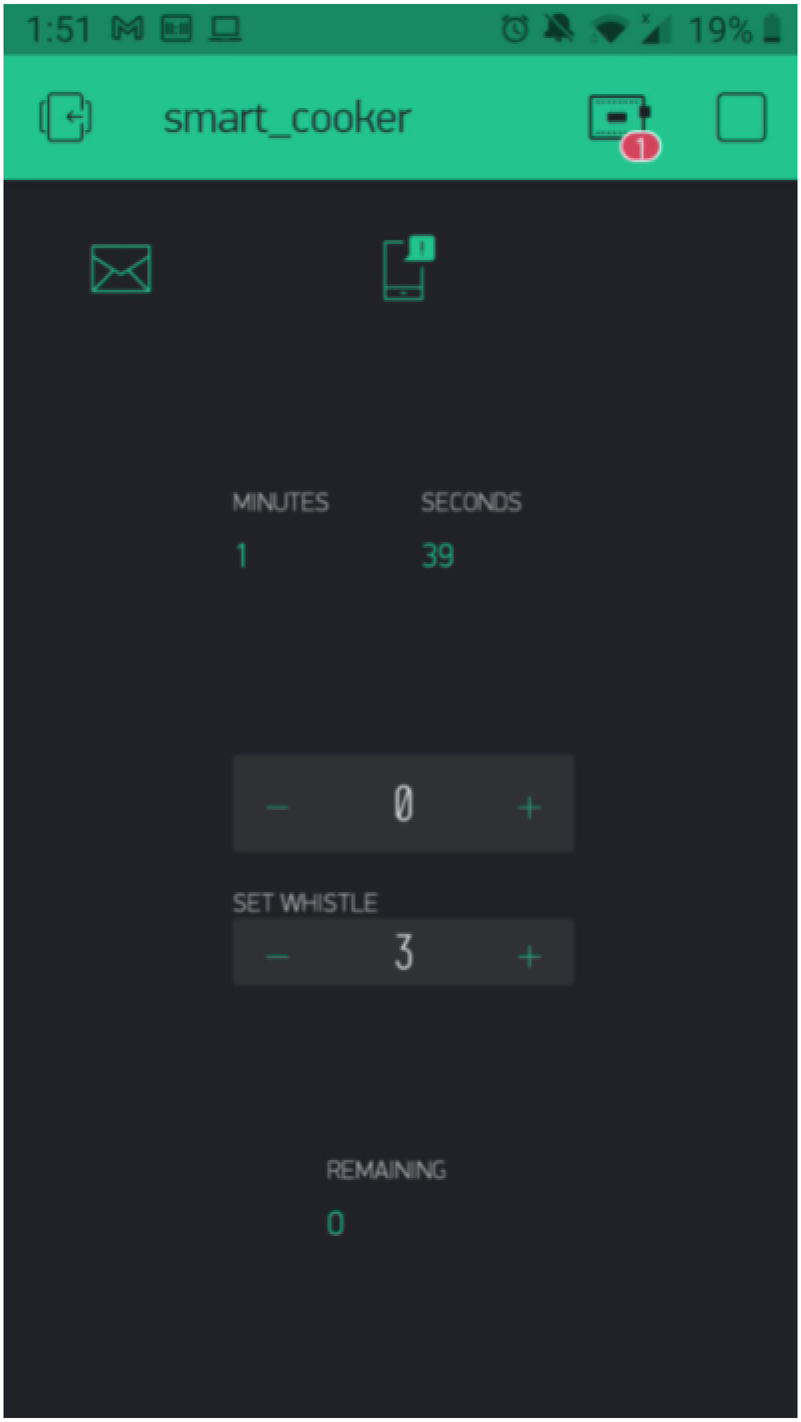


Food Ready

**Blynk** <dispatcher@blynk.io> Unsubscribe
to me

Your food is ready

Fig 10: Blynk email notification



MINUTES 1

SECONDS 39

0

SET WHISTLE 3

REMAINING 0

Fig 10: Blynk mobile app

# 6. CONCLUSION AND FUTURE WORK

## 6.1 CONCLUSION

Thus we have designed a system which performs the following operations for achieving the objectives

- Gets the number of whistles from the user from mobile through the Blynk app.
- Using the sound sensor detects each whistle and updates the remaining number of whistles to the users as a notification.
- Once everything is done it turns off the stove, sends notification to the mobile and also sends email to the user's account.

## 6.2 FUTURE WORK

In the future we would like to make this system more foolproof, by that we mean that we would include a timer and gas sensor to make sure even if the system fails there won't be any damage or accidents. Also if possible we would like to increase the quality of the sound detection system.

In the future, we can make options for cooking various varieties of food and we can automatically set whistles according to it.

# 7. REFERENCES

1. R. Dhianeswar, S. Sumathi and K. L. Joshitha, "Automatic Gas Controller," 2018 International Conference on Communication, Computing and Internet of Things (IC3IoT), 2018, pp. 215-218, doi: 10.1109/IC3IoT.2018.8668178.)

2. A. Yalmar, M. Parihar, V. Kadam and K. Kharat, "Implementation of automatic safety gas stove," 2015 Annual IEEE India Conference (INDICON), 2015, pp. 1-6, doi: 10.1109/INDICON.2015.7443474.)

3. JK Adarsh, R Kishore, R Arul, "Pressure Cooker Whistle Notification Using Arduino Based Wireless Sensor Networks," in *IEEE Access*, vol. 9, pp. 51393-51402, 2021, doi: 10.1109/ACCESS.2021.3069588.

4. Imran, N. Iqbal, S. Ahmad, and D. H. Kim, “” *Sustainability*, vol. 13, no. 5, p. 2461, Feb. 2021 [Online]. Available: http://dx.doi.org/10.3390/su13052461

5. Prathamesh Pandit, Harshkumar Patel, Jayesh Bhoir, Prashant Sathe, Ashish J. Chaudhari Assistant Professor Department of Mechanical Engineering Vidyavardhini’s College of Engineering and Technology,”Design and Analysis of Smart Whistle for Pressure Cooker”

## BIODATA

**Name:** Rufus Sam John Immanvel

**Mobile Number:** 6381935464

**Email ID:** rufussam.johnimmanvelj2019@vitstudent.ac.in

**Permanent Address: 14/1**2A, Satya sai nagar, kolathur, Chennai- 600099

**Name:** Keerthivasan J

**Mobile Number:** 8778205789

**Email ID:** keerthivasan.j2019@vitstudent.ac.in

**Permanent Address:** B1-6,Abhinayam Phase-3,Sriram Nagar,Nolambur, Chennai-600095

**Name:** Tiruveedhula uday

**Mobile Number:** 6300994345

**Email ID:** tiruveedhula.uday2019@vitstudent.ac.in

**Permanent Address:** ganganammapeta **;**Tenali, Andhra Pradesh - 522201