**Smart Cooking**

Prepared By

DEV CHINAI IU1641050008

JAYDEV RAVAL IU1641050018

VIDHI PANCHAL IU1641050033

**Table of Contents**

[Acknowledgment 3](#_TOC_250004)

[Abstract 4](#_TOC_250003)

[Chapter 1 - Introduction 5](#_TOC_250002)

* 1. [Introduction **5**](#_TOC_250001)
  2. [System Functionality **5**](#_TOC_250000)
  3. Project Overview **6**
  4. Flow Diagram

### Chapter 2 - Hardware Design & Description

* 1. Hardware Requirements
  2. Arduino Uno
  3. Temperature Sensor
  4. Servo Motor
  5. ESP8266
  6. Voltage Sensor

### Chapter 3 - Software Description

* 1. Software Requirements

**Chapter 4 - Implementation**

**Chapter 5 - Challenges**

**Chapter 6 - Conclusion**

**Chapter 7 - Future Scope**

# Abstract

# People in today’s generation are living a hectic life and finds rare time to spend for themselves. This also affects to the cooking and eating habits of a busy individual, for example, students, people from certain profession and etc.

# This is a serious issue to look upon, as we cannot completely rely on external sources of food, which can be harmful and non-pocket friendly.

# Smart Cooking is newbie friendly, easy to handle, economical and efficient, IoT based, piece of electronic which has implementation for automated cooking, fresh and homemade ready to serve.

# 

# Chapter 1 - Introduction

## 1.1 Introduction

## Modern busy lifestyles are often at odds with the healthy aspirations of consumers who want to achieve wellness through good nutrition and exercise. Convenience and limited time to exercise and prepare meals or dislike of food preparation and cooking, shape many people’s lifestyles today. Time seems to be an important factor in whether families are willing to cook and prepare healthy meals or choose pre-prepared foods, fast foods, and take-out, or eat snack foods instead of meals

## According to the studies, eating out frequently could be 5 times more expensive than cooking at home. Also it is found that people in this century are spending about 60 min compared to 120 min in 1960s. Frequently eating outside is not healthy as ingredients used by the restaurants may not fresh or processed. Another factor could be living alone without any cooking experience, for those it’s very difficult to get homemade food.

## So our aim of the project does not just deal with the modern challenges but also promises to improve the lifestyle of future adaptation.

## 1.2 System Functionality

## Smart Cooking has the following functionality:

## This whole process is automated, so the smart cooker is attached with temperature sensor which monitors internal temperature.

## This temperature are sent to the Arduino UNO at regular interval, so it can check for the credentials which are preset by the users.

## If the cooker’s temperature is low then, the servo will rotate clockwise to certain degree to rotate knob of the PSU, and power increase, if temperature is high then vice versa.

## Along with constant temperature monitoring, it also take keep track about pressure inside, as the pressure drops during whistle, then again servo change angle, and whistle counts are maintained for efficient cooking

## 1.3 Project overview

Smart

Cooker

Arduino UNO with

components

Wi-Fi

Module

Electronic Induction Plate

PSU

15A/220V

Power Supply

Smart

Phone

# Smart Cooker consists of utensil with a sensor attached inside to measure temperature and send it to the Arduino UNO, this Arduino make use of written algorithm to determine the voltage flow toward the induction plate which consists of copper coil, which heats up accordingly, this voltage is directly from wall supply so, it is very important to keep the voltage under control because we are dealing with over 4000 W, which can be vulnerable. After the process is complete, smartphone connected via Bluetooth will receive a message/notification that food is prepared.

# 1.4 Flow Diagram

START

Initial Configuration

tempCooker = 0

tempSet = 0

watt = 0

time = 0

User Connects via Bluetooth

And Gives Inputs

Change Watt Supply by Changing Angle of Servo till desired tempSet

NO

Timer Runs and tempCooker Monitored

YES

Is tempCooker == tempSet ?

Sensor Starts to Sense Temperature of Cooker

Is Timer Over

NO

Notification Send to Phone.

Induction Plate turned off

YES

STOP

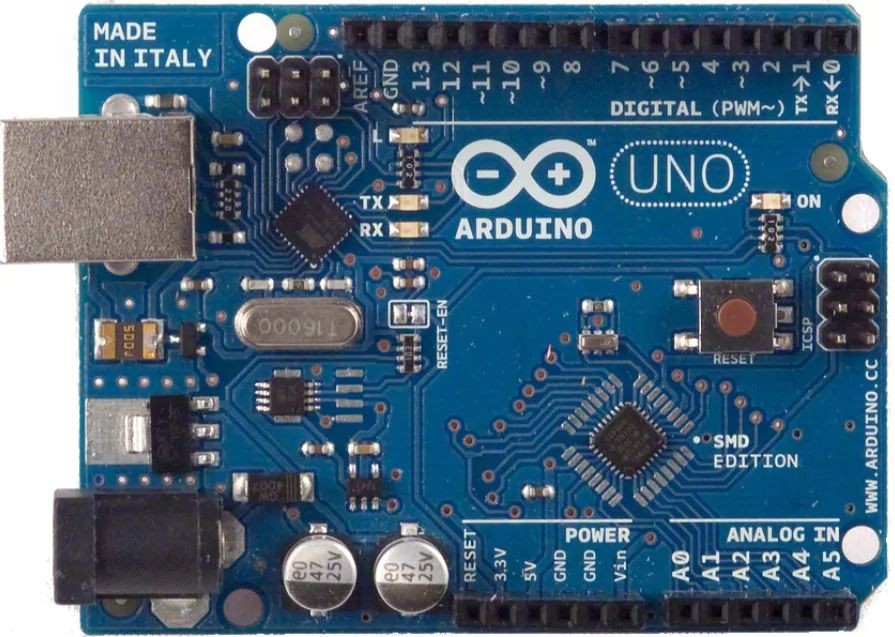
# Chapter 2 - Hardware Design & Description

* This chapter describes the hardware that is being used in the project.

2.1 Hardware Requirement

* Arduino UNO
* Waterproof DS18B20 Temperature Sensor
* Micro Servo Motor MG90s
* ESP8266
* Android Smartphone
* Multimeter
* Jumper Wire
* Breadboard
* DC Power Supply Unit with regulator
* ZMPT101B Voltage Sensor

2.2 Arduino UNO



Arduino is a tool for making computers that can sense and control more of the physical world than your desktop computer. It's an open-source physical computing platform based on a simple microcontroller board, and a development environment for writing software for the board. Arduino can be used to develop interactive objects, taking inputs from a variety of switches or sensors, and controlling a variety of lights, motors, and other physical outputs. Arduino projects can be standalone, or they can communicate with software running on your computer. The boards can be assembled by hand or purchased pre-assembled; the open-source IDE can be downloaded for free. The Arduino programming language is an implementation of Wiring, a similar physical computing platform, which is based on the Processing multimedia programming environment.

### Specification

Microcontroller Microchip ATmega328P

Operating Voltage 5 Volts

Input Voltage 7 - 20 Volts

Digital I/O Pins 14 (of which 6 provide PWM output) Analog Input Pins 6

DC Current per I/O Pin 20 mA DC Current for 3.3V Pin 50 mA

Flash Memory 32 KB of which 0.5 KB used by bootloader SRAM 2 KB

EEPROM 1 KB

Clock Speed 16 MHz

Length 68.6 mm

Width 53.4 mm

Weight 25 g

2.3 Waterproof DS18B20 Temperature Sensor



This is a pre-wired and waterproofed version of the DS18B20 sensor. Handy for when you need to measure something far away, or in wet conditions. While the sensor is good up to 125°C the cable is jacketed in PVC so we suggest keeping it under 100°C. Because they are digital, you don't get any signal degradation even over long distances!

These 1-wire digital temperature sensors are fairly precise (±0.5°C over much of the range) and can give up to 12 bits of precision from the onboard digital-to-analog converter. They work great with any microcontroller using a single digital pin, and you can even connect multiple ones to the same pin, each one has a unique 64-bit ID burned in at the factory to differentiate them. Usable with 3.0-5.0V systems.

As cooking at higher temperature in moist condition, it is important to pick a sensor with such material which can helps to avoid rust and sensor does not fails to work in harsh condition, by breaking down.

2.4 Micro Servo Motor MG90s



MG90S Metal Gear Mini Servo is tiny and lightweight with high output power, this tiny servo is perfect for RC Airplane, Helicopter, Quadcopter or Robot. This servo has metal gear for added strength and durability. Servo can rotate approximately 180 degree(90 in each direction), and works just like the standard kinds but smaller.

Specifications

• Weight: 13.4 g

• Dimension: 22.5 x 12 x 35.5

• Stall torque: 1.8 kgf·cm (4.8V )

• Operating speed: 0.1 s/60 degree

• Operating voltage: 4.8 V -

• Dead band width: 5 µs

This servo has capacity to rotate a knob of PSU to the induction plate to the desire voltage, instead of having a separate voltage regulator for changing the power. This is the most reliable way to change to power.

It has 3 wires: Brown – Ground

Red - 5 V supply

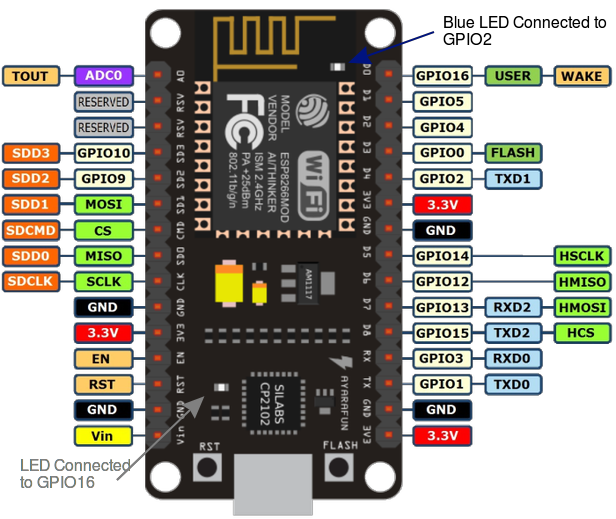
Orange – PWM signal is given in through wire to drive the motor

2.5 NodeMCU (ESP8266)

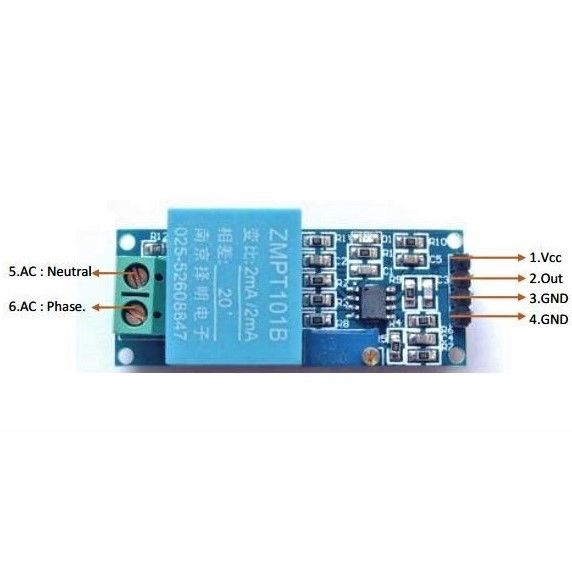


NodeMCU is an open-source firmware and development kit that helps you to prototype or build IoT product. It includes firmware which runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which is based on the ESP-12 module. The firmware uses the Lua scripting language. It is based on the eLua project, and built on the Espressif Non-OS SDK for ESP8266.

MCU stands for MicroController Unit - which really means it is a computer on a single chip. A microcontroller contains one or more CPUs (processor cores) along with memory and programmable input/output peripherals. They are used to automate automobile engine control, implantable medical devices, remote controls, office machines, appliances, power tools, toys etc.



2.6 ZMPT101B VOLTAGE SENSOR

Single-phase ac active output voltage transformer module and voltage sensor module

ZMPT101B voltage sensor module is a voltage sensor made from the ZMPT101B voltage transformer. It has high accuracy, good consistency for voltage and power measurement and it can measure up to 250V AC. It is simple to use and comes with a multi turn trim potentiometer for adjusting the ADC output.

FEATURES :

• within 250 V AC voltage can be measured.

• onboard micro-precision voltage transformer

• Installation : PCB mounting (Pin Length> 3mm)

• Operating temperature : 40ºC ~ + 70ºC

ADVANTAGES :

• Analog output corresponding quantity can be adjusted.

• Pcb board size: 49.5 (mm) x19.4 (mm)

• Good consistency, for voltage and power measurement

• Very efficient and accuracy

**Chapter 3 - Software Description**

* This chapter describes the software that is being used in the project.

3.1 Software Requirements

➔ Arduino IDE

➔ Android Studio

➔ Firebase Realtime Database

3.2 Arduino IDE

The Arduino integrated development environment (IDE) is a cross-platform application (for Windows, macOS, Linux) that is written in the programming language Java. It is used to write and upload programs to Arduino compatible boards, but also, with the help of 3rd party cores, other vendor development boards.

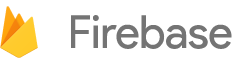
The source code for the IDE is released under the GNU General Public License, version 2. The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures.

The Arduino IDE employs the program argued to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware.

3.3 Android Studio

Android Studio is the official integrated development environment (IDE) for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. It is available for download on Windows, macOS and Linux based operating systems. It is a replacement for the Eclipse Android Development Tools (ADT) as the primary IDE for native Android application development.

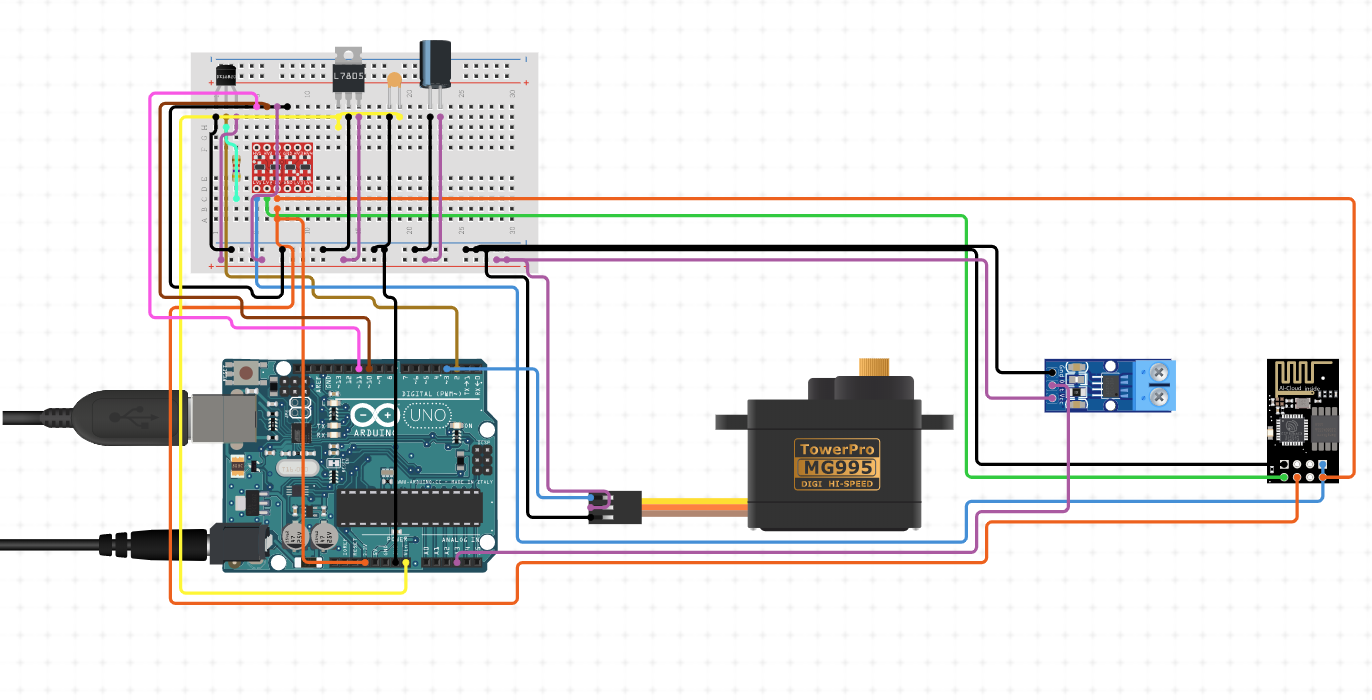
Android Studio was announced on May 16, 2013 at the Google I/O conference. It was in early access preview stage starting from version 0.1 in May 2013, then entered beta stage starting from version 0.8 which was released in June 2014. The first stable build was released in December 2014, starting from version 1.0.

3.4 Firebase Real-time Database

Firebase is a mobile and web application development platform developed by Firebase, Inc. in 2011, then acquired by Google in 2014. As of October 2018, the Firebase platform has 18 products, which are used by 1.5 million apps.

Firebase provides a realtime database and backend as a service. The service provides application developers an API that allows application data to be synchronized across clients and stored in Firebase's cloud. The company provides client libraries that enable integration with Android, iOS, JavaScript, Java, Objective-C, Swift and Node.js applications. The database is also accessible through a REST API and bindings for several JavaScript frameworks such as AngularJS, React, Ember.js and Backbone.js. The REST API uses the Server-Sent Events protocol, which is an API for creating HTTP connections for receiving push notifications from a server.

**Chapter 4 – Implementation**

****

4.2 Mapping:

* DS18B20 Temperature Sensor

Input Digital Signal to D2

5 V Supply from Arduino

Ground

* High Torque Servo

Output Analog Signal to D3

5 V Supply from Arduino

Ground

* Voltage Sensor

Output Analog Signal to A3

5 V Supply from Arduino

Ground

Live Wire from PSU to Phase PIN

Neutral Wire from PSU to Neutral PIN

* ESP8266 with Logic Level Converter

ESP8266 RXD to LogicLevelConverter LV1

ESP8266 TXD to LogicLevelConverter LV2

ESP8266 VCC to LogicLevelConverter LV

LogicLevelConverter LV to Arduino 3.3 V

4.3 Servo Mapping and Whistle Count:

Servo is fitted to the knob of the variable dc power supply unit, to it can rotate the knob precisely to the desire voltage output.

Since the servo is 360 degree rotating, so it is mandatory to control the speed and the angles of rotation, along with that it is also important to have a variable psu with minimum of 270 degree rotation.

So, the algorithm could look like this with knob position:

**void loop()**

**{**

**Int currentPosition = 120;**

**If (tempCooker > tempSet)**

**Servo.write(currentPosition - 10**); //Servo position shift anti-clockwise by 10 degree and since it is in loop it will continue to rotate until desired temperature of induction plate is not set.

**else If (tempCooker < tempSet)**

**Servo.write(currentPosition + 10);** //Servo position shoft clockwise by 10 degree

**delay(1000);**

**}**

This will manage the temperature inside the cooker despite the number of whistle blows.

Since mentioned in Gay-Lussac’s law, with constant volume, change is pressure is directly proportional to the change in temperature, So, when whistle blows, steam escapes and pressure and temperature inside drops, this is monitored by the temperature sensor for a long period of time. What is important to observe is the trend where there is sudden drop in temperate for a brief period and stable at that temperature for about couple of second.

So, proposed algorithm for this is:

**int seconds;**

**int tempBlows;**

**int tempCurrent;**

**int tempMax;**

**int whistleCount;**

**void setup()**

**{**

**tempMax = tempCooker;**

**whistleCount = 0;**

**}**

**void loop()**

**{**

**if(tempMax < tempCurrent)**

**{**

**tempMax = tempCurrent;**

**}**

**if(tempMax > tempCurrent)**

**{**

**seconds++;**

**}**

**if (seconds == 5)**

**{**

**whistleCount++;**

**seconds = 0;**

**}**

4.4 Wi-Fi Module Setup

Wi-Fi module helps to take a step forward in IoT level. Wi-Fi module can come in handy when dealing with such system, because our main purpose is automation, without being physically present, and get a notification on your smart phone for cooked food, along with real-time data being broadcasted to your device so that in case of any uncertainty, we could have control over the process. Wi-Fi implementation can also be useful for getting user input, so user can have a better user interface experience.

Setup Code:

const char \*SSID = "WIFI-SSID"; // Enter your Wi-Fi name

const char \*PASSWORD = "PASSWORD" ; // Enter your Wi-Fi password

ESP8266 wifi(WIFI\_PIN\_RX,WIFI\_PIN\_TX);

const int timeout = 10000; //define timeout of 10 sec

char menuOption = 0;

long time0;

void setup()

{

Serial.begin(9600);

while (!Serial) ;

Serial.println("start");

wifi.init(SSID, PASSWORD);

menuOption = menu();

}

void loop()

{

if(menuOption == '1') {

wifi.httpGet(host, hostPort);

char\* wifiBuf = wifi.getBuffer();

char \*wifiDateIdx = strstr (wifiBuf, "Date");

for (int i = 0; wifiDateIdx[i] != '\n' ; i++)

Serial.print(wifiDateIdx[i]);

}

if (millis() - time0 > timeout)

{

menuOption = menu();

}

}

char menu()

{

Serial.println(F("\nWhich component would you like to test?"));

Serial.println(F("(1) ESP8266-01 - Wifi Module"));

Serial.println(F("(menu) send anything else or press on board reset button\n"));

while (!Serial.available());

while (Serial.available())

{

char c = Serial.read();

if (isAlphaNumeric(c))

{

if(c == '1')

Serial.println(F("Now Testing ESP8266-01 - Wifi Module"));

else

{

Serial.println(F("illegal input!"));

return 0;

}

time0 = millis();

return c;

}

}

}

**Chapter 5 - Challenges**

Since we are dealing with such **high voltage and current** setup, safety was the most important aspect. Tuning the power and keeping the temperature steady was another challenge as with constant volume and steady temperature from coil increases pressure as steam generates, so whistle blows and again the power knob has to be rotated, so high quality PSU is important as we may be need to change the power supply frequently.

**Failure of the temperature sensor** is also another challenge, as it is being exposed to harsh condition so it was important for us to choose something which could withstand this condition and does not poison the food by leaking any of it material.

**Blow off valve** needed to be attached to the cooker, in case of whistle getting stuck, or by any means a sudden generation of stream due to PSU malfunction, where power surges immediately.

Fuses are implemented for each equipment, where power surge could be a potential threat.

An **emergency kill mechanism** has to be implemented which can only be controlled by a Wi-Fi module and actuator, independent to the whole setup. Reason for keeping it independent is because it will work despite of any damage to the system by cutting off the main current supply from PSU.

**Chapter 6 – Conclusion**

Communication model for this setup is:

Device to Device: Smart Cooker and Arduino

Device to Cloud: Arduino and ESP8266

This setup represents IOT Level 3, since data storage and processing is taking place on cloud.

After a series of test and minor changes made to the setup, we have concluded that the system is functional as per expected and can be improved after more case study and further research by adding more components and more functionality to the setup. There might be few flaws in design but it is important to note them during the whole process and bring chance to them.

**Chapter 7 - Future Scope**

There need to be more addition of the sensors such as water sensor, which can sense the level of water inside the pressure cooker, hence we can avoid issue blowing off safety valve.

More efficient and simplified form of utensil, which consist of a single cooking appliance, with integrated copper coil, integrated PSU, so that maintains minimalistic look.

A more user friendly user interface which would help to provide more accessibility to the system.