BRAC UNIVERSITY
Inspiring Excellence

CSE360 Theory Project-based Assignment

Project Title: Smart Kitchen System

Team Members:

Sanzida Akter , 19101584
Sabrin Akhter, 18301098
Akash Ghosh, 19101425
Aniruddho Saha, 18201117

Date of Submission : 25th August, 2022

Introduction:

It is undeniable that a smart kitchen system will show to be a very effective and user-friendly invention in the current technological era when the globe is adapting to smart solutions for every household task. A smart kitchen will guarantee a safe and comfortable atmosphere for cooking. The three elements—a smoke detector, a fire alarm buzzer, and a motion sensor—will provide a safe and intelligent working environment in the kitchen. Additionally, our solution is incredibly cost-effective, so putting it into place won't break the bank. As a result, it becomes a practical, user-friendly, and reasonably priced solution.

Application Area:

We frequently observe in our daily lives that when we cook, smoke is produced from our cooking, making it challenging for us to breathe effectively in the smoke. So, a smart kitchen system is useful in this situation. Because smoke contains various harmful gases that a gas sensor can easily detect, we utilize a gas sensor in our smart kitchen system to detect smoke. If the smoke level is low, the smoke is sucked out and ejected outside using an exhaust fan.

This system replaces a kitchen hood, although an exhaust fan is far more affordable, smaller, and simpler to clean than a kitchen hood, which may be expensive and typically very large and difficult to maintain. As a result, our suggested technique is far more effective and economical. Additionally, since the exhaust fan operates automatically, we won't need to bother about manually turning it on, making our lives easier.

Another important factor that must be considered in order to prevent risks while working in the kitchen is safety. It's important to locate and immediately put out any uncontrolled fires in the kitchen. To detect an uncontrolled fire and activate the sprinkler, we are utilizing a flame sensor in our smart kitchen. Furthermore, it will alert us to the presence of a fire in the kitchen via a buzzer. Additionally, the system uses a GSM module that, whenever a fire is discovered, it begins sending an SMS to 999. So, this is how this technique can guarantee kitchen safety.

Temperature and humidity are two essential factors to keep the foods fresh because bacteria can grow in food at high temperature and high humidity. Therefore, we have used a DHT22 sensor here to track the temperature and humidity. The temperature and humidity values will be shown on an LCD display. So, if the user finds that the temperature and humidity are very high on the LCD display then, the user can understand that the foods can get spoiled. So, he/she can keep the foods which are prone to spoilage, in the fridge or can put it in a cool and dry place.

Furthermore, in the smart kitchen system we have used a PIR sensor which will detect motion and thus, when someone will enter the kitchen, a LED light bulb will turn on.

Any kitchen can use smart home technologies, such as a smart kitchen system. It is quite simple to adopt this smart kitchen system in all kitchens because it is such a low-cost technology. We will be able to deal with the issue of kitchen safety by employing this system, and we won't have to cook in the smoke any more because the smoke will be pulled out of the kitchen. We can monitor the kitchen's temperature and humidity from this smart kitchen in order to preserve the food's freshness. As a result, this technique improves kitchen efficiency and simplifies our lives.

Technology & Tools :

- **Sensors:**
  1. IR Flame Sensor
  2. MQ-2 Gas Sensor
  3. PIR Motion Sensor
  4. DHT22

- **General Purpose I/O Devices:**
  1. Buzzer
  2. DC motor
  3. Servo motor
  4. Push button
  5. 16 x 2 LCD Display (12IC)

- **Interfacing IC:**
    1. Arduino Mega

- **Other tools:**
    1. Wires
    2. Breadboard
    3. Exhaust fan
    4. Sprinkler
    5. TSBC33 Transistor
    6. Resistor
    7. Battery
    8. L298 Motor Driver
    9. GSM module
    10. Voltage regulator

- **Software :**
    1. Arduino Integrated Development Environment Software

Programming Language:

A processing unit is required for our system to function efficiently as it will collect the signals from the sensors as input and analyze them properly to produce the output using the peripheral devices as output. An Arduino Mega board, an open-sourced Arduino microcontroller board, will operate as the processing unit. C++ will be the programming language we apply, and the Arduino IDE software will be implemented to execute the code. Afterward, using a USB cable, the compiled code will be transmitted from our PC to the Arduino board.

Working Mechanism of Sensors:

- **MQ-2 Gas Sensor:**

  We are using this sensor to detect the presence of smoke inside the kitchen. If low levels of smoke, which are produced during cooking, are detected by the sensor, it will send a digital pulse high signal to the Arduino Mega board which is pre-programmed to turn on an exhaust fan when it gets a digital pulse high signal from this MQ-2 gas sensor.

  With a 5V DC operating voltage, this sensor consumes about 800mW. Because the detection is dependent on the change in resistance of the sensing material when the gas comes into contact with the material, it is a Metal Oxide Semiconductor (MOS) type Gas Sensor, also known as a Chemiresistor. Tin dioxide, an n-type semiconductor, absorbs oxygen when heated to high temperatures in the atmosphere. Donor electrons in tin dioxide are drawn to oxygen that has been adsorbed on the surface of the detecting material when there is no smoke and the air is clear. It stops the flow of electric current because there are no available free electrons. As a result, the resistance increases.

  Adsorbed oxygen reacts with reducing gases to reduce its surface density when there are flammable gases or smoke present. The tin dioxide is subsequently given up its electrons, allowing current to flow freely through the sensor. The amount of free electrons in SnO_2 that are available determines this conduction. More free electrons are available when there are more noxious gases in the air. Gas concentrations can be discovered by utilizing a straightforward voltage divider network. Depending on what kind of gas is being used, the sensor's resistance varies.

  The MQ-2 smoke sensor is sensitive to smoke and the following flammable gases: LPG, methane, alcohol, butane, propane and hydrogen. The smoke sensor has a built-in potentiometer that allows us to adjust the sensor’s sensitivity according to the concentration of smoke at which we want our sensor to detect it. To calibrate the gas sensor we can hold the gas sensor near the amount of smoke at which we want our sensor to send a digital pulse high output signal to the arduino and we need to keep turning the potentiometer until the Red LED on the module starts glowing. To increase sensitivity, we need to turn the screw clockwise and to decrease the sensitivity, we need to turn it anti clockwise. The voltage that the sensor outputs changes according to the smoke level that exists in the atmosphere. The sensor outputs a voltage that is proportional to the concentration of smoke. That is, the relationship between voltage and smoke concentration is as follows:

1. The greater the smoke concentration, the greater the output voltage
2. The lower the smoke concentration, the lower the output voltage

The output can be an analog signal sent via A0 pin that can be read with an analog input of the Arduino or a digital output sent via the D0 pin that can be read with a digital input of the Arduino. If smoke is detected by the sensor, the resistance of the sensing material changes according to the concentration of the smoke and then the corresponding analog output voltage reading is sent to the arduino board which will process the reading according to the way it was programmed. Since we want to send a digital output to the Arduino board, we used the on board LM393 High Precision Comparator to digitalize the analog output voltage signal. The LM393 takes the output from the sensor and compares it with a reference voltage and sends a digital high or low signal accordingly.

- **Flame Sensor:**

This sensor allows us to determine whether there are significant amounts of flame present in the kitchen. There are two different kinds of flame sensors, each with three and four pins, respectively. Any micro-controller can easily be interfaced with one of these sensors. Here, a four-pin flame sensor is being used in our smart kitchen system. An IR receiver, a power indication (LED), a potentiometer, OP-Amp circuits, and a led indicator make up the module. This sensor can operate between 3.3V and 5V. It can detect fires up to 100 cm away and can detect flames with wavelengths between 760 and 1100 nm. It can output digital signal (logic 0 for absence of flame and logic 1 for presence of flame) via the D0 pin (pin 4) and can also output analog voltage signal via the A0 pin (pin 3).

This sensor uses an infrared flame flash technique, which enables it to penetrate layers of ice, water vapor, dust, and grease. The photodiode-based IR receiver only picks up low frequency flickering IR radiations generated by the flame. When IR radiation strikes an IR receiver, both the output voltage and resistance vary in proportion to the amount of IR radiation that was received.An on board LM393 op amp is used as a comparator to adjust the sensitivity level of the sensor and to make the readings stable. Since the sensor needs to distinguish between an actual flame and the flame from the stove, a certain threshold level needs to be set for its sensitivity. If the reading exceeds this threshold value, then the flame sensor will give a digital pulse high as output and if the reading is below the threshold value then a digital pulse low will be sent as output. After sensing the flame, the flame sensor will then give an indication through an LED attached at its top. The threshold value can be changed by rotating the top of the potentiometer. To calibrate the flame sensor and set the threshold value, we can hold the flame sensor near the intensity of flame at which we want the sensor to give a digital pulse high output and we need to keep turning the potentiometer until the LED indicator on the module starts glowing.

- **DHT22**

We are using in our smart kitchen system is a DHT22 sensor which can measure both temperature and humidity. This sensor is used in our system to measure the temperature and humidity of the atmosphere of our kitchen and send it to the pre-programmed Arduino board which will process it and display the humidity and temperature readings on a 16 x 2 12IC LCD Display. It can measure temperatures in the range of -40° C to +125° C with +-0.5 degrees accuracy and has a humidity measuring range, from 0 to 100% with 2-5% accuracy. This sensor consists of a humidity sensing component, a NTC temperature sensor (or a thermistor) and an IC on the back side of the sensor.

In order to measure the humidity of the kitchen atmosphere, it uses a humidity sensing component which has two electrodes with moisture holding substrate between them. As the humidity changes, the conductivity of the substrate changes and so does the resistance between these electrodes. This change in resistance is measured and processed by the IC which makes it ready to be read by the Arduino board to which this reading is sent.

The DHT22 sensor uses a thermistor or an NTC temperature sensor to detect the kitchen's temperature. This thermistor is a variable resistor, meaning that as the temperature changes, so does its resistance. NTC stands for "Negative Temperature Coefficient," which denotes that resistance lowers as temperature rises. On the other hand, the resistance rises as the temperature falls. The IC also measures and processes the change in resistance, preparing it for reading by the Arduino board that receives this reading. To achieve greater variations in resistance with only slight changes in temperature, these sensors are created by sintering semi-conductive materials like ceramics or polymers.

- **PIR Sensor:**

This sensor is being used to find movement in the kitchen. Every time someone enters the kitchen building, the motion sensor will detect motion and provide output to the Arduino board. The Arduino is set up to automatically switch on the LED light in the kitchen anytime the PIR motion sensor sends a digital pulse high to it.

Every living thing, including humans and animals, emits infrared radiation, commonly referred to as a heat signature. PIR sensors are made of pyroelectric sensors, which can detect various intensities of these infrared rays without emitting any energy of its own. Instead, it merely accepts it without comment. The human body's surface temperature ranges from 27 to 36 degrees Celsius, and the majority of the IR radiation it emits is focused in the 8 to 12 um wavelength region.

This sensor has two slots made of a special material that is sensitive to infrared. When a human body or any animal passes by, it intercepts the first slot of the PIR sensor. This causes a positive differential change between the two slots. When a human body leaves the sensing area, the sensor generates a negative differential change between the two slots, which is detected as “movement”. But the sensor cannot detect motion over a large distance. Thus, to lengthen the detection distance of the detector, an optical system is added to collect the infrared radiation. A plastic optical reflection system or plastic Fresnel lens is usually used as a focusing system for infrared radiation. In the detection area, which is the kitchen in our case, the lens of the detector receives the infrared radiation from the human body through the clothing and which then focuses it on the pyroelectric sensor. When the human body moves in this surveillance mode, it enters a certain field of view of the PIR sensor in sequence and then walks out of the field of view. The pyroelectric sensor senses the moving human body for a while and then does not when the body leaves its detection area, so the infrared radiation of the human body constantly changes the temperature of the pyroelectric material. As a result, it outputs a corresponding digital pulse high signal, which is the alarm signal. This signal then goes to the pre-programmed Arduino mega board. We are using this sensor because it is small, needs low power, easy to use, and, most importantly, inexpensive.

Connection With ICs:

In our system we are going to use Arduino Mega as the interfacing IC. We will connect the negative terminal of a power supply battery with the ground bus of the breadboard and positive terminal of the battery with the Vin pin of a voltage regulator. Since the input voltage can be varied so we are using a L7805 voltage regulator with the power supply. Then we will use a push button to power on the Arduino Mega. One terminal of the button will be connected to the positive bus on the breadboard and the other terminal will be connected to Arduino pin 12 and another pin will be connected to the ground. Then we connected the GND pin of the Arduino to the ground bus of the breadboard and we have connected the Vin pin of the voltage regulator to the Vin or 12V pin of the arduino. We interfaced the IR flame sensor with the Arduino board by connecting the D0 pin of the flame sensor with the A10 pin of Arduino through the breadboard.

The GND pin of the flame sensor is connected with the ground bus of the breadboard and VCC pin is connected to the positive bus of the bread board.

The D0 pin of the MQ-2 gas sensor is connected to the A0 pin of the Arduino via breadboard. The VCC pin of the sensor is connected to the positive bus of the breadboard and the GND pin of the sensor is connected to the ground bus of the breadboard. We have used a buzzer as an alarm. The positive pin of this buzzer is connected to the positive bus of the breadboard and negative terminal of the buzzer is connected to the collector of an NPN transistor (TSBC33). Base of this transistor is connected to one terminal of the 1Kohm resistor. Another terminal of the resistor is connected to Arduino pin 4. The emitter of the transistor is connected to the ground bus of the breadboard. We have connected the transistor with the buzzer because the buzzer may operate in 5V or 9V or 12V but the Arduino Mega board can operate from 5V. So by connecting the transistor we are ensuring that the buzzer won't be able to draw more than 5V from the Arduino. The PIR motion sensor communicates with the Arduino via SIG pin which is a PIR output pin. We have connected this SIG pin with the Arduino pin 11 through the breadboard. Also, we connected the GND and VCC pin of the sensor with the ground bus and positive bus of the bread board respectively. Next, we connected an LED bulb to show that when someone enters the kitchen it will turn on. The DIN pin of the LED is connected to the Arduino pin 10 and GND and VCC pins are connected to the ground bus and positive bus of the bread board respectively.

We have connected a GSM (SIM800L) module to send emergency messages text to 999 if the fire breaks out. Here the Tx pin of GSM is connected to the PWM enable pin 19 (Rx) of Arduino and Rx pin of GSM is connected to the PWM enable pin 18 (Tx) pin of the Arduino. We connected the VCC and GND pin of the module to the 2 bus (positive and ground) respectively. Also, we connected the RST (reset) pin of the module to arduino pin 13 so that it can work even if the AT command does not work for some reason.

We have connected a servo motor with our IC which works based on the PWM (pulse width modulation) principle. It will help to rotate the sprinkler in 360 degree and to spray the water when the fire breaks out. The SIG pin of the motor which is the control pin is connected to the

Arduino pin 14 via breadboard. A power supply is needed for the motor to turn on and so the VCC and GND pin of the motor is connected to the positive bus and ground bus of the breadboard respectively.

We need to turn on the exhausting fan automatically when smoke is detected which is why we have connected a DC motor in our circuit. One coil of the DC motor is connected to the Output pin 2 of L298 Motor Driver and another coil is connected to the output pin 1 of the driver. Input pin 1 of the driver is connected to the Arduino pin 6 and input pin 2 is connected to the arduino pin 5. Through these input pins arduino will provide the signal to control the motor. This driver controls the rotation speed and direction of the DC motor to turn on the exhausting fan. To give a constant voltage to the motor we connected the Vin pin of a voltage regulator (L7805) with the 12V VSS pin of the driver. Then we connected the GND pin of the driver with the ground bus of the breadboard and the 5V Vs pin of the driver is connected with the positive bus on the breadboard to keep the H bridge, potentiometer and circuits active. The ENA pin of the driver is connected to Arduino pin 2 to enable the motor which is connected to the left side. To measure humidity and temperature, we connected the data pin of DHT22 with arduino mega pin 9. We connected the GND pin with the bus ground and the VCC with the positive bus through a 10Kohm resistor. At last, we connected an LCD (16x2 I2C) to show the temperature and humidity values to the user. Here, we connected the SDA pin and SCL pin of the I2C to the arduino pin 20 and 21 respectively. Ground and VCC of the LCD are connected to the ground bus and positive bus of the breadboard respectively.

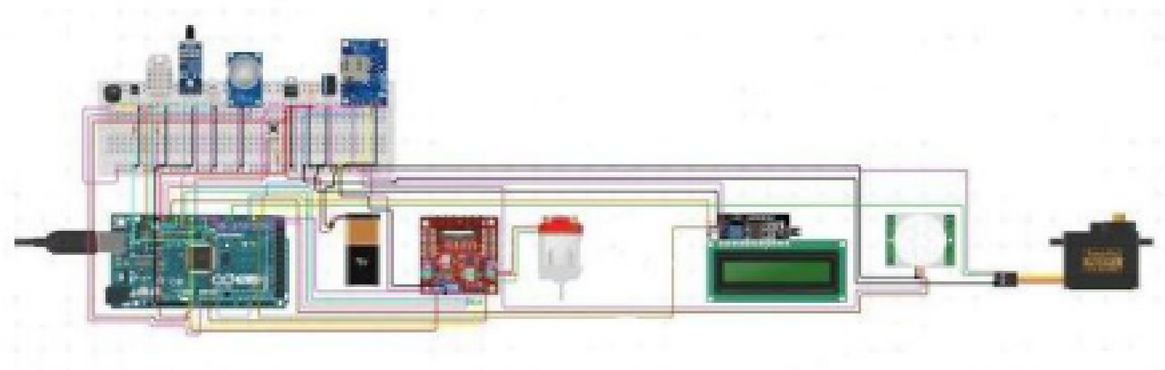

Figure: Circuit Diagram

Data Flow From Sensors Through ICs to I/O Devices:

We are using a push button which is an input device to give power to the Arduino Mega. When the push button is not pressed, due to the pull-up resistor, the input pin (Arduino pin 12) of the IC will read a high state and the circuit will not get established. When the button is pressed, it connects the input pin directly to ground. So, the current flows through the resistor to ground and the input pin of the IC reads a low state. After reading the low state, Arduino Mega will get the power and turn on. Also, as we have connected the Vin pin of the voltage regulator to the Vin or 12V pin of the arduino so, from this Vin pin arduino will get a constant voltage.

In the MQ2 gas sensor, the digital output signal sent via D0 pin will be read by the A0 pin of the Arduino Mega. Here, we have connected the analog pin of the arduino with the digital pin of the MQ2 gas sensor. The analog pin is taking a digital input because analog pins of arduino can take both analog inputs and digital inputs. Then, the IC will read the input that it got from the D0 pin of the gas sensor. If the input is high, the IC will enable the motor by giving a high signal from the arduino pin 2 to the ENA pin of the motor driver. To control the DC motor, the arduino will send signals from arduino pin 6 to the input pin 1 of the driver and also, from arduino pin 5 to the input pin 2 of the L298 motor driver. This driver will control the rotation speed and direction of the DC motor. The voltage regulator has a Vin pin that gets the power from the power supply and this voltage regulator will convert the voltage into a constant voltage. Then, from this Vin pin, this constant voltage is supplied to the 12V VSS pin of the motor driver. After that, the motor driver will send electrical signals to the 2 coils of the DC motor through the output pin 1 and output pin 2. For this, the DC motor will start rotating and so, the exhaust fan turns on. In the flame sensor, the digital output will be sent via D0 pin. This D0 is connected with the pin no. A10 of the arduino Mega. D0 pin will give a digital signal and the input pin of the IC will read the digital signal and if it is a high signal, then it will be identified that a fire has occurred in the kitchen. So, the arduino will send a high signal through pin 4 and pin 14. This pin 4 is connected with a resistor. So, the current will pass through this resistor and then it will flow in the base terminal of the transistor. As a result, the transistor will turn on. Current will then flow through the collector. As the buzzer is connected with the collector terminal of the transistor so, the buzzer will turn on and make a buzzing sound. We are using a resistor here so that the transistor does not burn. A servo motor is also connected with the IC. As its VCC and GND pin is connected to the positive bus and ground bus of the breadboard respectively, it will get the power supply from there. Then the arduino will send a high signal from the arduino pin 14 towards the control pin of the servo motor. Thus, the servo motor will turn on and the sprinkler will start sprinkling water.

We have also used a GSM module, GSM is an international standard for mobile telephones. It allows the Arduino board to connect to the internet, send and receive SMS using the GSM library. Here, we have to mount a SIM 800L to the GSM. And we have set a message in the arduino beforehand. Here the Tx pin and Rx pin of GSM is used for Serial communication. The Rx pin of GSM receives commands from the PWM enable pin 19 (Rx) of Arduino and the Tx pin of GSM is used for sending out the data to the PWM enable pin 18 (Tx) of the Arduino.When from the flame detector arduino gets a digital high signal, then the arduino sends the pre written sms “EMERGENCY !! A fire has occurred!” to the emergency number 999 using GSM.

The output of the PIR motion detection sensor is connected directly to the Arduino pin 11. If any motion is detected by the PIR sensor, this arduino pin value will be set to ‘1’. When this pin no.

11 is set to high, a current will flow from the arduino pin 10 to the DIN pin of the LED. As the current is flowing in the LED, the LED will turn on.

Lastly, we have a DHT22 sensor in our system. This DHT22 sensor sends humidity and temperature reading via its data pin to pin 9 of our Arduino Mega board. This data pin outputs the value of both temperature and humidity as serial data. The output given out by the data pin will be in the order of 8bit humidity integer data + 8bit the Humidity decimal data +8 bit temperature integer data + 8bit fractional temperature data +8 bit parity bit. While connecting an LCD display with an Arduino, the LCD display consumes a lot of pins on the Arduino. So, the solution is to use a I2C LCD Display. It will only consume 4 pins of the arduino. The SDA pin of the 12C LCD is a serial Data pin. After getting the temperature and humidity data from DHT22, the arduino pin 20 will send the data to the SDA pin of the 12C LCD. This SDA pin is used for both transmit and receive. At the heart of the 12C LCD adapter is an 8-Bit I/O Expander chip – PCF8574. This chip converts the I2C data from the Arduino into the parallel data required by the LCD display. SCL is a Serial Clock pin. Here, a timing signal is supplied by the Bus Master device from the pin no. 21. After doing the necessary code, the humidity and temperature value will be shown on the 12C LCD display.

Code:

```
#include<SoftwareSerial.h>
#include<Servo.h>
#include <Wire.h>
#include <DHT.h>
#include <Adafruit_Sensor.h>
#include<DC_Motor.h>
#include<LiquidCrystal_I2C.h>
//LCD
LiquidCrystal_I2Clcd(0x27,16,2);
//GSM Module
SoftwareSerial mySerial(18, 19);
//Constants
//FLAME SENSOR
#define flamePin A10
#define buzzer 4
```

```
//GAS SENSOR
#define MQ2pin A0
int MQ2sensorValue = 0;
// DC Motor connections
int enA = 2;
int in1 = 6;
int in2 = 5;
//DHT22
#define DHTPIN 9
#define DHTTYPE DHT22
DHT dht(DHTPIN, DHTTYPE); // Initialize DHT sensor
float hum; //Stores humidity value
float temp; //Stores temperature value
// Servo motor for sprinkler
#define servoPin = 14;
Servo Servo1;
//PIR
int PIRsensor = 11;
int LED = 10;
unsigned long t=0;
void setup() {
Serial.begin(9600);
mySerial.begin(9600); // Setting the baud rate of GSM Module
Servo1.attach(servoPin);
//LCD
lcd.begin();
lcd.backlight(HIGH);
lcd.clear();
//DHT
dht.begin();
//FLAME
// declare the ledPin and buzzer as an OUTPUT:
pinMode(buzzer, OUTPUT);
//GAS
pinMode(enA, OUTPUT);
pinMode(in1, OUTPUT);
pinMode(in2, OUTPUT);
// Turn off dc motor - Initial state
digitalWrite(in1, LOW);
digitalWrite(in2, LOW);
```

```
//PIR
pinMode(PIRsensor, INPUT);
pinMode(LED, OUTPUT);
digitalWrite(LED,LOW);
while(millis()<13000)
{
digitalWrite(LED,HIGH);
delay(50);
digitalWrite(LED,LOW);
delay(50);
}
digitalWrite(LED,LOW);
delay(100);
}
int globalCount =0;
void loop()
{
//FLAME SENSOR
flameSensor= digitalRead(FlamePin);
Serial.println(flameSensor);
if (flameSensor == HIGH)
{
Serial.println("Fire Detected");
digitalWrite(buzzer, HIGH);
delay(1000);
if (Serial.available()>0){
switch(Serial.read())
{
case 's':
if(globalCount<1){
SendMessage();
globalCount++;
}
}
delay(1000);
// Make servo go to 0 degrees
Servo1.write(0);
delay(1000);
// Make servo go to 90 degrees
Servo1.write(90);
```

```
delay(1000);
// Make servo go to 180 degrees
Servo1.write(180);
delay(1000);
}
else
{
globalCount = 0;
digitalWrite(buzzer,LOW);
delay(sensorValue);
}
//GAS SENSOR
MQ2sensorValue = digitalRead(MQ2pin);
if (MQ2sensorValue == HIGH){
Serial.println("Smoke Detected");
directionControl();
delay(1000);
speedControl();
delay(1000);
}
//PIR SENSOR
digitalWrite(LED,LOW);
if(digitalRead(PIRsensor)==HIGH)
{
t=millis();
while(millis()<(t+5000))
{
digitalWrite(LED,,HIGH);
if((millis()>(t+2300))&&(digitalRead(PIRsensor)==HIGH))
{
t=millis();
}
}
}
//DHT22
int chk = DHT.read22(DHT22_PIN);
//Read data and store it to variables hum and temp
hum = dht.humidity;
temp= dht.temperature;
//Print temp and humidity values to LCD
```

```
lcd.setCursor(0,0);
lcd.print("Humidity: ");
lcd.print(hum);
lcd.print("%");
lcd.setCursor(0,1);
lcd.print("Temp: ");
lcd.print(temp);
lcd.println("Celcius");
delay(2000); //Delay 2 sec between temperature/humidity check.
}
void directionControl() {
// Set motors to maximum speed
// For PWM maximum possible values are 0 to 255
analogWrite(enA, 255);
// Turn on motor A
digitalWrite(in1, HIGH);
digitalWrite(in2, LOW);
delay(2000);
// Turn off motor
digitalWrite(in1, LOW);
digitalWrite(in2, LOW);
}
// This function lets us control speed of the motor
void speedControl() {
// Turn on motor
digitalWrite(in1, HIGH);
digitalWrite(in2, LOW);
// Accelerate from zero to maximum speed
for (int i = 0; i < 256; i++) {
analogWrite(enA, i);
delay(20);
}
// Decelerate from maximum speed to zero
for (int i = 255; i >= 0; --i) {
analogWrite(enA, i);
delay(20);
}
// Now turn off motor
digitalWrite(in1, LOW);
digitalWrite(in2, LOW);
```

```
}
void SendMessage()
{
mySerial.println("AT+CMGF=1"); //Sets the GSM Module in Text Mode
delay(1000); // Delay of 1000 milli seconds or 1 second
mySerial.println("AT+CMGS=\"999\"\r");
delay(1000);
mySerial.println("EMERGENCY !! A fire has occurred! Address : xxxxx ");// the
emergency SMS...xxxx represents the location wherethe fire has occurred
delay(100);
mySerial.println((char)26);// ASCII code of CTRL+Z
delay(1000);
}
```

Estimated cost analysis:

Component	Quantity	Estimated price
Arduino Mega	1pc	545
IR Flame sensor	1pc	85
PIR sensor	1pc	110
MQ-2 Gas sensor	1pc	140
Rotating Sprinkler	1pc	650
Buzzer	1pc	30
Exhaust fan	1pc	800
DHT22	1pc	360
GSM module	1pc	350
Wires	1 Feet	70
Push button	1pc	5
Servo Motor	1pc	150
DC Motor	1pc	165
Breadboard	1pc	80
1k resistor	1pc	15
10 k resistor	1pc	55
L298 Motor Driver	1pc	155
Transistor	1pc	5
Battery	1pc	25
16*2 LCD display(12IC)	1pc	350
Voltage regulator	1pc	20
		Total price: 4165 BDT

Responsibilities of Each Member:

Student ID	**Responsibility**
19101584	• Introduction • Technology & Tools • Workplan (Gantt Chart) • Conclusion • References
18301098	• Application Area • Programming language • Estimated cost analysis • Responsibilities of Each Member • References
19101425 ,18201117	• Working mechanism of Sensors • Connection with ICs • Data flow from sensors through ICs to I/O devices • References

Workplan (Gantt Chart):

Card	List	Labels	Members	Due date
Planning	WorkPlan		SA	Aug 30
Research	WorkPlan		SA	Sep 3
Working on Sensor	WorkPlan		SA	Sep 7
Working on Introduction and Application Area	WorkPlan		SA	Sep 12
Working on IC Connection	WorkPlan		SA	Sep 16
Working on Dataflow	WorkPlan		SA	Sep 20
Working on Estimated Cost	WorkPlan		SA	Sep 22
Working on Code	WorkPlan		SA	Sep 25

Conclusion:

In conclusion, a smart kitchen has the greatest potential to become one of the most important pieces of infrastructure for contemporary households. It will continue to be a low-cost system that can reach more people thanks to our planned system. As a result, it will not only enable more people to work in a risk-free, secure, and effective kitchen environment, but it will also increase the system's commercial viability. It goes without saying that this approach will make our everyday repetitive work the least annoying and also the least time consuming in 2021 when everyone is always in a hurry.

References:

1. https://www.circuitstoday.com/interface-gsm-module-with-arduino
2. https://robu.in/sim800l-interfacing-with-arduino/
3. https://www.allaboutcircuits.com/projects/control-a-motor-with-an-arduino/
4. https://startingelectronics.org/beginners/circuits/arduino-buzzer/#:~:text=The%20transistor%20allows%20the%20buzzer,the%20buzzer%20can%20be%20used.&text=When%20the%20external%20power%20supply,the%20GND%20of%20the%20Arduino
5. https://lastminuteengineers.com/mq2-gas-senser-arduino-tutorial/
6. https://kookye.com/2018/11/16/arduino-lesson-flame-sensor/?fbclid=IwAR37bkbfNDYUc27b15f2g_fgogJWEUIvDGdWoPdII1k3JhIUSbp68SpyPno#:~:text=It%20gives%20us%20both%20analog,rotating%20the%20top%20of%20potentiometer.&text=The%20detection%20angle%20of%20the%20flame%20sensor%20module%20is%20about%2060%20degrees
7. http://rogerbit.com/wprb/wp-content/uploads/2018/01/Flame-sensor-arduino.pdf
8. https://youtu.be/Ojsu7SMW9Yg
9. https://youtu.be/QBwBlLffdgM
10. https://www.arrow.com/en/research-and-events/articles/how-motion-sensors-work
11. https://robu.in/pir-sensor-working-principle/#:~:text=PIR%20is%20made%20of%20a,energy%20but%20passively%20receives%20it.&text=When%20a%20human%20body%20leaves,change%20between%20the%20two%20bisects
12. https://create.arduino.cc/projecthub/trijalsrimal/fire-gas-and-smoke-detector-8241dc?fbclid=IwAR1hhfo5zm5ubYp6OyCMN0_2j9WmA0PCBFm5dr13WWNgqKCN6Twe3S6Pd6g
13. https://circuitdigest.com/microcontroller-projects/send-data-to-web-server-using-sim9
14. https://arduinolearn.github.io/gsm.html
15. https://theorycircuit.com/arduino-flame-sensor-interface/
16. https://www.arduino.cc/en/tutorial/pushbutton?fbclid=IwAR19Ur4kuzSZ3zJGE3UosxmgKbzin5LVStZmRUQgCML6n7Soq4fLSZs5J-c
17. https://components101.com/ics/7805-voltage-regulator-ic-pinout-datasheet
18. https://www.electronicshub.org/arduino-i2c-tutorial/
19. https://lastminuteengineers.com/i2c-lcd-arduino-tutorial/
20. https://components101.com/sensors/dht22-pinout-specs-datasheet