NSS COLLEGE OF ENGINEERING

PALAKKAD, KERALA, INDIA

PINCODE:678008

SUBJECT CODE: ICT305

MICROCONTROLLER PROJECT REPORT

SMART KITCHEN AUTOMATION

SUBMITTED BY:

1.ANASWARA .A ROLL NO:13

2.ANASWARA.M.S ROLL NO:14

3.ANEESH.T ROLL NO:15

4.ANSHAD.A ROLL NO:16

STAFF IN CHARGE:

ACKNOWLEDGEMENT

It is our privilege to express our sincere gratitude and whole Hearted indebtedness to all the people who had helped us in making this project a reality. First of all, we thank Almighty God, who has Showered his grace upon us to make this work a successful one.This project would have been incomplete without proper guidance And constant encouragement. On this behalf, we wish to express our Gratitude to all our teaching and non teaching staffs of **the Department of Instrumentation and Control Engineering, NSS College of Engineering, Palakkad** for their valuable help and Support. We also thank our friends for their whole hearted support and valuable suggestions. Lastly, we express our sincere thanks to All those who have either directly or indirectly supported us in Making this project a reality.

CONTENTS

INTRODUCTION

In the era of automation and the Internet of Things (IoT), the concept of a “Smart Kitchen” has emerged as a practical solution for making household management more efficient, safe, and user-friendly. A Smart Kitchen integrates advanced technologies to offer convenience, energy efficiency, and safety. This project utilizes Arduino, an open-source microcontroller platform, to develop a system that incorporates two smart technologies: motion-sensing-based lighting and emergency detection with automatic exhaust opening. These technologies ensure seamless control over kitchen operations, reduce human effort, and enhance safety.

1. **Motion-Sensing-Based Lighting**
   One of the primary objectives of this project is to implement energy-efficient lighting that responds to human presence. Using motion sensors like Passive Infrared (PIR) sensors connected to Arduino, the lights automatically turn on when movement is detected within the kitchen space. When no movement is sensed for a predefined period, the system switches the lights off to conserve energy. This technology not only promotes sustainability by minimizing electricity wastage but also offers

convenience by eliminating the need for manual operation of light switches.

2. **Emergency Detection with buzzer** Kitchens are prone to hazards such as gas leaks, excessive smoke, or fire, making safety an essential aspect of smart design. In this project, gas and smoke sensors (such as MQ-2 or MQ-135 sensors) are integrated with Arduino to detect dangerous situations. If the system detects an unusual concentration of gas or smoke, it triggers an emergency alarm to indicate harmful gases or smoke. This automated response ensures a safer environment, preventing potential accidents and alerting occupants promptly to take appropriate action.

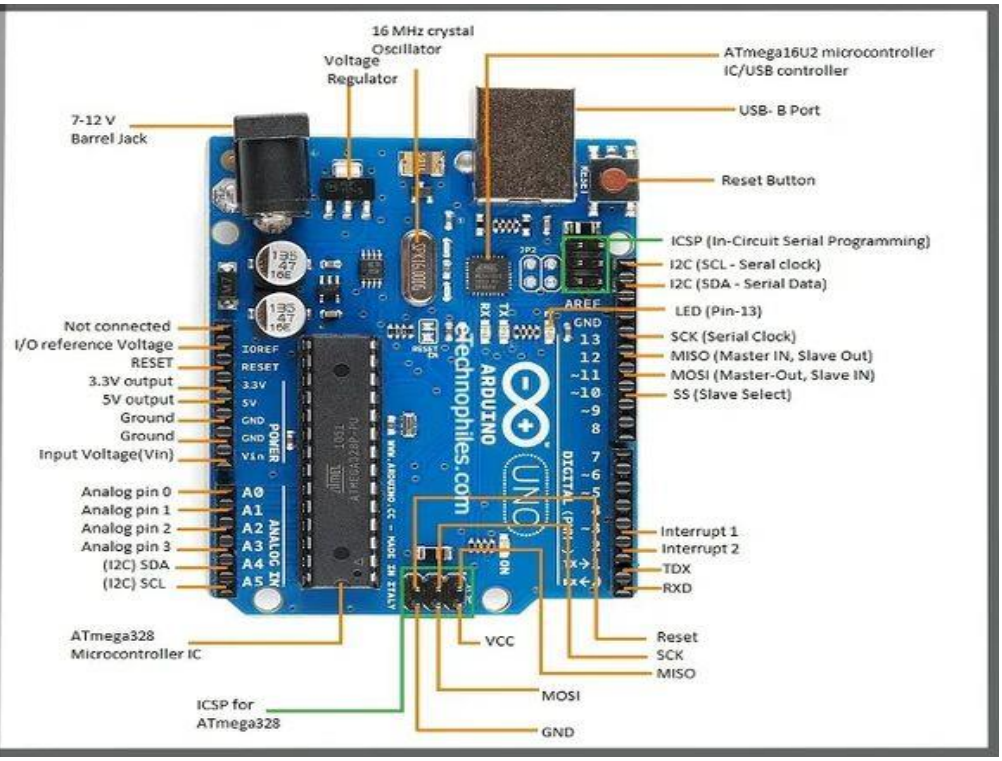
The Smart Kitchen using Arduino represents a step towards smarter homes, offering a blend of automation, convenience, and safety. By integrating motion-sensing lighting and emergency detection this project not only enhances user experience but also addresses critical safety concerns. As energy efficiency and safety continue to be at the forefront of modern design, innovations like this play a crucial role in transforming traditional kitchens into intelligent spaces.

INTRODUCTION TO ARDUINO

The Arduino UNO is a key component of the Smart Kitchen project, serving as the central microcontroller that manages and coordinates various smart functionalities. Known for its versatility, ease of use, and open-source platform, Arduino UNO is a microcontroller board based on the ATmega328P microprocessor. This compact yet powerful board is widely used in electronics projects because it can interface with numerous sensors and modules, making it ideal for IoT applications such as smart home systems.

In the Smart Kitchen project, the Arduino UNO is responsible for executing both motion-sensing-based lighting and emergency detection with automatic exhaust control. With its programmable I/O pins, the Arduino UNO can communicate with motion sensors, gas and smoke detectors, and control output devices like lights and exhaust fans. By using Arduino's coding environment, these components can be efficiently calibrated, enabling the kitchen to respond intelligently to real-time events, such as turning lights on and off based on motion or ventilating the space in case of gas leaks or smoke.

Arduino UNO Pinout



Arduino function					Arduino function
reset	(PCINT14/RESET) PC6	1	28	PC5 (ADC5/SCL/PCINT13)	analog input 5
digital pin 0 (RX)	(PCINT16/RXD) PD0	2	27	PC4 (ADC4/SDA/PCINT12)	analog input 4
digital pin 1 (TX)	(PCINT17/TXD) PD1	3	26	PC3 (ADC3/PCINT11)	analog input 3
digital pin 2	(PCINT18/INT0) PD2	4	25	PC2 (ADC2/PCINT10)	analog input 2
digital pin 3 (PWM)	(PCINT19/OC2B/INT1) PD3	5	24	PC1 (ADC1/PCINT9)	analog input 1
digital pin 4	(PCINT20/XCK/T0) PD4	6	23	PC0 (ADC0/PCINT8)	analog input 0
VCC	VCC	7	22	GND	GND
GND	GND	8	21	AREF	analog reference
crystal	(PCINT6/XTAL1/TOSC1) PB6	9	20	AVCC	VCC
crystal	(PCINT7/XTAL2/TOSC2) PB7	10	19	PB5 (SCK/PCINT5)	digital pin 13
digital pin 5 (PWM)	(PCINT21/OC0B/T1) PD5	11	18	PB4 (MISO/PCINT4)	digital pin 12
digital pin 6 (PWM)	(PCINT22/OC0A/AIN0) PD6	12	17	PB3 (MOSI/OC2A/PCINT3)	digital pin 11(PWM)
digital pin 7	(PCINT23/AIN1) PD7	13	16	PB2 (SS/OC1B/PCINT2)	digital pin 10 (PWM)
digital pin 8	(PCINT0/CLKO/ICP1) PB0	14	15	PB1 (OC1A/PCINT1)	digital pin 9 (PWM)

Digital Pins 11,12 & 13 are used by the ICSP header for MOSI, MISO, SCK connections (Atmega168 pins 17,18 & 19). Avoid low-impedance loads on these pins when using the ICSP header.

SMART KITCHEN USING ARDUINO

AIM

To set up a smart kitchen or kitchen automation using Arduino.

OBJECTIVE

The primary objectives of the Smart Kitchen project are:

- To automate kitchen lighting based on human presence, reducing manual effort and energy usage.
- To ensure kitchen safety through the detection of gas leaks or excessive smoke, triggering with a buzzer indication to prevent accidents

PROJECT SCOPE

This project focuses on integrating Arduino UNO with motion sensors for lighting control and gas/smoke sensors for emergency response. By implementing these two core features, this project establishes a foundation for further smart kitchen enhancements, including potential integration

with mobile applications, voice control, or additional kitchen appliance management systems.

COMPONENTS AND MATERIALS

1.**Arduino UNO**

2.**IR Proximity Sensor**

3.Ultrasonic Sensor

4.**MQ-2 or MQ-135 Gas and Smoke Sensors**

5.**LED Light**

6.**Relay Module**

7.Jumper wires

8.**Buzzer**

9**.LDR**

10**.Breadboard**

WORKING PRINCIPLE

The Smart Kitchen project leverages Arduino technology to transform conventional kitchen spaces into intelligent, responsive environments. By integrating motion-sensing lighting and emergency detection with automatic exhaust control, this project addresses both convenience and safety concerns. Below is a detailed breakdown of the working principles and the operational framework of each component within the project.

1. Working Principle of Motion-Sensing-Based Lighting

The motion-sensing lighting system operates on the principle of detecting infrared radiation from moving objects, particularly humans. In this system, a Infrared (IR) and Ultrasonic sensors plays a crucial role. The IR sensor is calibrated to recognize the infrared heat emitted by human bodies and subsequently triggers an electrical response.

Operational Framework:

Sensor Activation: When an individual enters the kitchen, the IR sensor detects the change in infrared radiation levels. This change is processed by the Arduino microcontroller, which then activates the connected lighting circuit.

Automation Control: The lights remain active as long as the sensor detects movement. When no motion is sensed for a predetermined period, typically adjustable within the Arduino code, the system automatically switches off the lights, conserving energy.

Energy Efficiency: This motion-sensing mechanism ensures lighting is only used when necessary, minimizing electricity waste and enhancing the energy efficiency of the kitchen.

2. Working Principle of Emergency Detection with Buzzer

The emergency detection component operates based on gas and smoke sensor technology. By integrating MQ-series sensors (such as MQ-2 or MQ-135) with Arduino, this system continuously

monitors air quality for signs of dangerous gases or smoke.

Operational Framework:

Continuous Monitoring: The gas and smoke sensors consistently assess the levels of various gases and smoke in the air. These sensors are calibrated to detect specific gases, like methane or propane, which are commonly used in kitchens.

Emergency Triggering: Once the detected gas or smoke levels surpass the safety threshold, the sensors send a signal to the Arduino Microcontroller. The Arduino then initiates an action which is triggering an alarm to alert the occupants.

PROGRAM

```
#include <NewPing.h>
// Define pins
#define TRIG_PIN 2
#define ECHO_PIN 3
#define LDR_PIN A0
#define GAS_SENSOR_PIN A1
// IR sensor pins
#define IR_SENSOR_1_PIN 11
#define IR_SENSOR_2_PIN 12
#define IR_SENSOR_3_PIN 13
// LED pins for ultrasonic sensor distances
#define LED_ULTRASONIC_10_20_PIN 4
#define LED_ULTRASONIC_21_30_PIN 5
#define LED_ULTRASONIC_31_40_PIN 6
// LED pins for IR sensors
#define LED_IR_SENSOR_1_PIN 7
#define LED_IR_SENSOR_2_PIN 8
#define LED_IR_SENSOR_3_PIN 9
```

```
#define BUZZER_PIN A2
 // Pin for the red LED
// Ultrasonic sensor setup
int distance;
NewPing sonar(TRIG_PIN, ECHO_PIN);
// Distance thresholds
const int DISTANCE_THRESHOLD_1 = 15; // in cm
const int DISTANCE_THRESHOLD_2 = 30; // in cm
const int DISTANCE_THRESHOLD_3 = 40; // in cm
const int GAS_THRESHOLD = 600; // threshold for gas detection
const int LDR_THRESHOLD = 1; // threshold for LDR (adjust as needed)
void setup() {
  // Set up LED pins as outputs
  pinMode(LED_ULTRASONIC_10_20_PIN, OUTPUT);
  pinMode(LED_ULTRASONIC_21_30_PIN, OUTPUT);
  pinMode(LED_ULTRASONIC_31_40_PIN, OUTPUT);
  pinMode(LED_IR_SENSOR_1_PIN, OUTPUT);
  pinMode(LED_IR_SENSOR_2_PIN, OUTPUT);
```

```
    pinMode(LED_IR_SENSOR_3_PIN, OUTPUT);
    pinMode(BUZZER_PIN, OUTPUT);

     // Set up red LED pin

    pinMode(IR_SENSOR_1_PIN, INPUT);
    pinMode(IR_SENSOR_2_PIN, INPUT);
    pinMode(IR_SENSOR_3_PIN, INPUT);

    Serial.begin(9600);
    analogWrite(LED_ULTRASONIC_10_20_PIN, 1023);
          analogWrite(LED_ULTRASONIC_21_30_PIN,
1023);
          analogWrite(LED_ULTRASONIC_31_40_PIN,
1023);

}

void loop() {
    // Read the LDR value
```

```
    int ldrValue = digitalRead(LDR_PIN);

    // Only operate the ultrasonic sensor if LDR is high
    if (ldrValue == LDR_THRESHOLD) {
        delay(50); // Wait for a moment
        int distance = sonar.ping_cm();

        // Distance-based LED activation for ultrasonic sensor
        if (distance > 0 && distance <= DISTANCE_THRESHOLD_1) {
            analogWrite(LED_ULTRASONIC_10_20_PIN, 255);
            analogWrite(LED_ULTRASONIC_21_30_PIN, 0);
            analogWrite(LED_ULTRASONIC_31_40_PIN, 0);
        } else if (distance > DISTANCE_THRESHOLD_1 && distance <= DISTANCE_THRESHOLD_2) {
            analogWrite(LED_ULTRASONIC_10_20_PIN, 0);
            analogWrite(LED_ULTRASONIC_21_30_PIN, 255);
            analogWrite(LED_ULTRASONIC_31_40_PIN, 0);
```

```
      } else if (distance > DISTANCE_THRESHOLD_2 &&
distance <= DISTANCE_THRESHOLD_3) {
        analogWrite(LED_ULTRASONIC_10_20_PIN, 0);
        analogWrite(LED_ULTRASONIC_21_30_PIN, 0);
        analogWrite(LED_ULTRASONIC_31_40_PIN,
255);
      } else {
        // Turn off all ultrasonic distance LEDs if out of
range
        analogWrite(LED_ULTRASONIC_10_20_PIN, 0);
        analogWrite(LED_ULTRASONIC_21_30_PIN, 0);
        analogWrite(LED_ULTRASONIC_31_40_PIN, 0);
      }
    } else {
      // Turn off all ultrasonic distance LEDs if LDR is low
      analogWrite(LED_ULTRASONIC_10_20_PIN, 0);
      analogWrite(LED_ULTRASONIC_21_30_PIN, 0);
      analogWrite(LED_ULTRASONIC_31_40_PIN, 0);
    }

    // Check IR sensors and activate corresponding LEDs
```

```
  int irValue1 = digitalRead(IR_SENSOR_1_PIN);
  int irValue2 = digitalRead(IR_SENSOR_2_PIN);
  int irValue3 = digitalRead(IR_SENSOR_3_PIN);

  digitalWrite(LED_IR_SENSOR_1_PIN, irValue1 ==
LOW ? HIGH : LOW);
  digitalWrite(LED_IR_SENSOR_2_PIN, irValue2 ==
LOW ? HIGH : LOW);
  digitalWrite(LED_IR_SENSOR_3_PIN, irValue3 ==
LOW ? HIGH : LOW);
  delay(300);


  // Gas detection
  int gasValue = analogRead(GAS_SENSOR_PIN);
  if (gasValue > GAS_THRESHOLD) {

    digitalWrite(BUZZER_PIN, HIGH);
    digitalWrite(BUZZER_PIN, HIGH);

    delay(50);
```

```
digitalWrite(BUZZER_PIN, LOW);
digitalWrite(BUZZER_PIN, LOW);
delay(25);
digitalWrite(BUZZER_PIN, HIGH);
digitalWrite(BUZZER_PIN, HIGH);
delay(100);
digitalWrite(BUZZER_PIN, LOW);
digitalWrite(BUZZER_PIN, LOW);
delay(25);
  digitalWrite(BUZZER_PIN, HIGH);
digitalWrite(BUZZER_PIN, HIGH);
delay(50);
digitalWrite(BUZZER_PIN, LOW);
digitalWrite(BUZZER_PIN, LOW);
delay(25);
digitalWrite(BUZZER_PIN, HIGH);
digitalWrite(BUZZER_PIN, HIGH);
delay(100);
digitalWrite(BUZZER_PIN, LOW);
digitalWrite(BUZZER_PIN, LOW);
```

```
        delay(25);
       // Activate buzzer
        ; // Turn on red LED for gas leak
    } else {
        digitalWrite(BUZZER_PIN, LOW);  // Deactivate
buzzer
         // Turn off red LED if no gas detected
    }
    // Print sensor values for debugging
    Serial.print("LDR Value: ");
    Serial.print(ldrValue);
    Serial.print(", Gas Value: ");
    Serial.println(gasValue);
    Serial.println(distance);

    delay(500); // Delay for stability
}
```

CONCLUSION

The "Smart Kitchen Automation Using Arduino" project successfully demonstrates how automation can streamline kitchen activities, enhance energy efficiency, and improve overall convenience. . This system not only reduces manual labor but also conserves resources by optimizing energy usage based on real-time data.

The system's modular design allows for easy upgrades and customization, accommodating additional smart appliances or functionalities as required. By leveraging Arduino and a suite of sensors and actuators, we developed a system capable of automating tasks such as lighting, gas leak detection.

In conclusion, the project demonstrates that Arduino-based smart kitchen automation can be both practical and impactful. This prototype lays a foundation for future innovations in smart home technology, highlighting the potential for more advanced, interconnected, and user-friendly kitchen environments. This work serves as a

stepping stone toward fully automated and efficient households, encouraging continued exploration and adoption of smart technologies for a sustainable and convenient future.