

Unlock your Potential

with

“bharat intern”

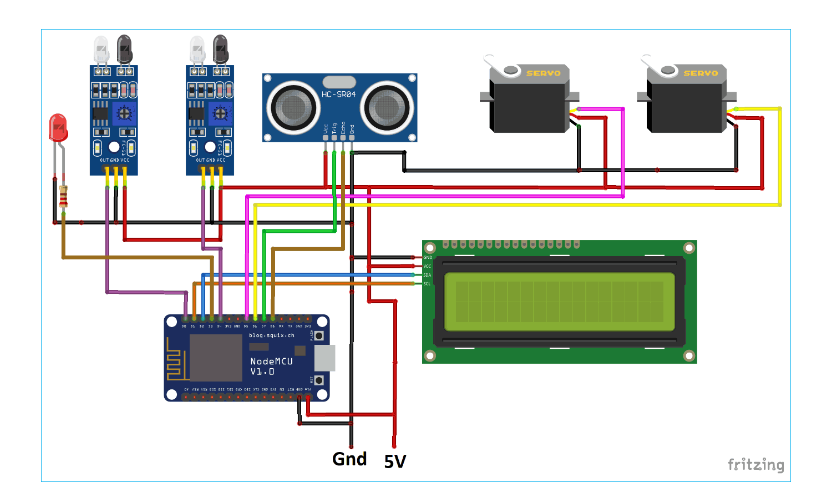
NAME-SOMYA SHISHODIA

PHONE NO.-7678284982

EMAIL ID.-SHISHODIASOMYA31@GMAIL.COM

Question 1 **Smart parking system:**

IOT-based car parking management system help in efficient parking space utilization only Arduino code and components used



Required components include:

1. Arduino board (for example, Arduino Uno)

2. Ultrasonic distance sensor (for example, HC-SR04)

3. LEDs (to indicate the availability of parking)

4. jumper wires

5. Optional breadboard

***Arduino Code***

// Include necessary libraries

#include <NewPing.h>

// Define the pins for ultrasonic sens

#define TRIGGER\_PIN 2

#define ECHO\_PIN 3

// Define the number of parking spaces available

#define TOTAL\_PARKING\_SPACES 5

// Define the pins for LED indicators

int parkingLEDs[] = {4, 5, 6, 7, 8};

// Create an instance of the NewPing library

NewPing sonar(TRIGGER\_PIN, ECHO\_PIN);

// Array to store parking space occupancy status

bool parkingStatus[TOTAL\_PARKING\_SPACES];

void setup() {

  // Initialize LED pins as output

  for (int i = 0; i < TOTAL\_PARKING\_SPACES; i++) {

    pinMode(parkingLEDs[i], OUTPUT);

    parkingStatus[i] = false; // Set initial status as vacant

  }

  Serial.begin(9600);

}

void loop() {

  // Loop through each parking space

  for (int i = 0; i < TOTAL\_PARKING\_SPACES; i++) {

    // Measure the distance using ultrasonic sensor

    int distance = sonar.ping\_cm();

    // Check if a car is parked in the current space

    if (distance < 30) {

      // Car detected, update status if previously vacant

      if (!parkingStatus[i]) {

        parkingStatus[i] = true;

        digitalWrite(parkingLEDs[i], HIGH); // Turn on LED

        Serial.print("Car parked at space ");

        Serial.println(i + 1);

      }

    } else {

      // No car detected, update status if previously occupied

      if (parkingStatus[i]) {

        parkingStatus[i] = false;

        digitalWrite(parkingLEDs[i], LOW); // Turn off LED

        Serial.print("Space ");

        Serial.print(i + 1);

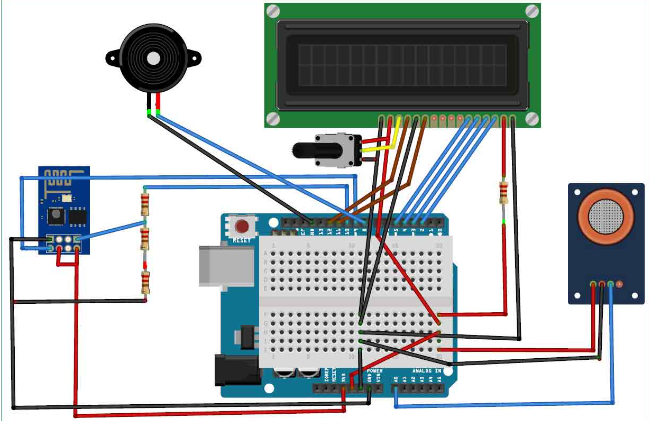
        Serial.println(" is vacant");

      }

    }

Question 2 Air pollution monitoring system :

The system consists of air sensors that sense the presence of harmful gases and compounds in the air Arduino code and components used.



We will need the following components to build an air sensor system that can detect dangerous chemicals and components in the air using Arduino:

1. Arduino board (for example, Arduino Uno)

2. Gas sensors (for example, MQ-2 for detecting numerous gases or particular sensors for individual gases such as MQ-6 for LPG, MQ-7 for CO, and so on).

3. Jumper cables

4. (Optional) breadboard

Arduino Code:

#include "MQ135.h"

#include <SoftwareSerial.h>

#define DEBUG true

SoftwareSerial esp8266(9,10); // This makes pin 9 of Arduino as RX pin and pin 10 of Arduino as the TX pin

const int sensorPin= 0;

int air\_quality;

#include <LiquidCrystal.h>

LiquidCrystal lcd(12,11, 5, 4, 3, 2);

void setup() {

pinMode(8, OUTPUT);

lcd.begin(16,2);

lcd.setCursor (0,0);

lcd.print ("circuitdigest ");

lcd.setCursor (0,1);

lcd.print ("Sensor Warming ");

delay(1000);

Serial.begin(115200);

esp8266.begin(115200); // your esp's baud rate might be different

  sendData("AT+RST\r\n",2000,DEBUG); // reset module

  sendData("AT+CWMODE=2\r\n",1000,DEBUG); // configure as access point

  sendData("AT+CIFSR\r\n",1000,DEBUG); // get ip address

  sendData("AT+CIPMUair\_quality=1\r\n",1000,DEBUG); // configure for multiple connections

  sendData("AT+CIPSERVER=1,80\r\n",1000,DEBUG); // turn on server on port 80

pinMode(sensorPin, INPUT);        //Gas sensor will be an input to the arduino

lcd.clear();

}

void loop() {

MQ135 gasSensor = MQ135(A0);

float air\_quality = gasSensor.getPPM();

if(esp8266.available()) // check if the esp is sending a message

  {

    if(esp8266.find("+IPD,"))

    {

     delay(1000);

     int connectionId = esp8266.read()-48; /\* We are subtracting 48 from the output because the read() function returns the ASCII decimal value and the first decimal number which is 0 starts at 48\*/

     String webpage = "<h1>IOT Air Pollution Monitoring System</h1>";

       webpage += "<p><h2>";

       webpage+= " Air Quality is ";

       webpage+= air\_quality;

       webpage+=" PPM";

       webpage += "<p>";

     if (air\_quality<=1000)

{

  webpage+= "Fresh Air";

}

else if(air\_quality<=2000 && air\_quality>=1000)

{

  webpage+= "Poor Air";

}

else if (air\_quality>=2000 )

{

webpage+= "Danger! Move to Fresh Air";

}

webpage += "</h2></p></body>";

     String cipSend = "AT+CIPSEND=";

     cipSend += connectionId;

     cipSend += ",";

     cipSend +=webpage.length();

     cipSend +="\r\n";

     sendData(cipSend,1000,DEBUG);

     sendData(webpage,1000,DEBUG);

     cipSend = "AT+CIPSEND=";

     cipSend += connectionId;

     cipSend += ",";

     cipSend +=webpage.length();

     cipSend +="\r\n";

     String closeCommand = "AT+CIPCLOSE=";

     closeCommand+=connectionId; // append connection id

     closeCommand+="\r\n";

     sendData(closeCommand,3000,DEBUG);

    }

  }

lcd.setCursor (0, 0);

lcd.print ("Air Quality is ");

lcd.print (air\_quality);

lcd.print (" PPM ");

lcd.setCursor (0,1);

if (air\_quality<=1000)

{

lcd.print("Fresh Air");

digitalWrite(8, LOW);

}

else if( air\_quality>=1000 && air\_quality<=2000 )

{

lcd.print("Poor Air, Open Windows");

digitalWrite(8, HIGH );

}

else if (air\_quality>=2000 )

{

lcd.print("Danger! Move to Fresh Air");

digitalWrite(8, HIGH);   // turn the LED on

}

lcd.scrollDisplayLeft();

delay(1000);

}

String sendData(String command, const int timeout, boolean debug)

{

    String response = "";

    esp8266.print(command); // send the read character to the esp8266

    long int time = millis();

    while( (time+timeout) > millis())

    {

      while(esp8266.available())

      {

        // The esp has data so display its output to the serial window

        char c = esp8266.read(); // read the next character.

        response+=c;

      }

    }

    if(debug)

    {

      Serial.print(response);

    }

    return response;

}

Question3 Health monitoring system:

This system allows users to keep track of their body vitals like blood pressure, sugar level,and heartbeat and remotely send the data to healthcare professionals.

we need the following components to build a health monitoring system that allows users to watch their body vitals such as blood pressure, sugar level, and heartbeat and remotely submit the data to healthcare professionals:

1. Arduino board (such as the Arduino Uno or Arduino Nano)

2. A blood pressure sensor (for example, the BMP180 or a specialized blood pressure sensor).

3. Glucometer sensor for detecting blood sugar levels

4. Heart rate sensor (such as the Pulse Sensor or MAX30102)

5. Wireless communication module (e.g., ESP8266) or Bluetooth module (e.g., HC-05 or HC-06)

6. Jumper cables

7. Optional breadboard

Arduino Code:

#include <Wire.h>

#include <Adafruit\_Sensor.h>

#include <Adafruit\_BME280.h>

#include <ESP8266WiFi.h>

#include <ESP8266HTTPClient.h>

// Replace with your Wi-Fi credentials

const char\* ssid = "YOUR\_WIFI\_SSID";

const char\* password = "YOUR\_WIFI\_PASSWORD";

// Replace with your healthcare professional's server address

const char\* serverAddress = "HEALTHCARE\_SERVER\_ADDRESS";

const int serverPort = 80;

Adafruit\_BME280 bme;

const int pulsePin = A0;

void setup() {

  Serial.begin(9600);

  WiFi.begin(ssid, password);

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

    delay(1000);

    Serial.println("Connecting to WiFi...");

  }

  if (!bme.begin(0x76)) {

    Serial.println("Could not find a valid BME280 sensor, check wiring!");

    while (1);

  }

}

void loop() {

  float temperature = bme.readTemperature();

  float pressure = bme.readPressure() / 100.0F;

  int heartbeat = analogRead(pulsePin);

  Serial.print("Temperature: ");

  Serial.print(temperature);

  Serial.print(" °C, Pressure: ");

  Serial.print(pressure);

  Serial.print(" hPa, Heartbeat: ");

  Serial.println(heartbeat);

  sendDataToServer(temperature, pressure, heartbeat);

  delay(5000); // Adjust the delay as needed for your application

}

void sendDataToServer(float temperature, float pressure, int heartbeat) {

  if (WiFi.status() == WL\_CONNECTED) {

    WiFiClient client; // Create a WiFiClient object

    HTTPClient http;

    http.begin(client, "http://" + String(serverAddress) + ":" + String(serverPort) + "/update");

    http.addHeader("Content-Type", "application/x-www-form-urlencoded");

    String postData = "temperature=" + String(temperature) +

                      "&pressure=" + String(pressure) +

                      "&heartbeat=" + String(heartbeat);

    int httpCode = http.POST(postData);

    if (httpCode > 0) {

      Serial.printf("HTTP POST request sent with status code: %d\n", httpCode);

    } else {

      Serial.println("HTTP POST request failed.");

    }

    http.end();

  } else {

    Serial.println("WiFi connection lost. Data not sent.");

  }

}