Automatic Hand Sanitizer Dispenser with COVID19 Live Updates Using ESP32

INDEX

Abstract:

In this project, we will build an Auto Hand Sanitizer Dispenser with an LCD which also shows the live count of Coronavirus cases. This project will use ESP32, Ultrasonic Sensor, 16x2 LCD Module, Water pump, and Hand Sanitizer. We are using Esri's API Explorer to get the live data of Covid19 infected people. An ultrasonic sensor is used to check the presence of hands below the outlet of the sanitizer machine. It will continuously calculate the distance between the sanitizer outlet and itself and tells the ESP to turn on the pump whenever the distance is less than 15cm to push the sanitizer out.

ESP32 is used as the main controller, it is a Wi-Fi module that can easily connect to the internet. Many people use it to build many IoT based projects using ESP32.

Introduction:

Corona Virus (Covid19) is wreaking havoc in the world. Almost every country is suffering from the Corona Virus. WHO has already announced it a Pandemic disease and many cities are under. lockdown situation, people can't step out of their homes, and thousands have lost their lives.

Viruses such as COVID-19 are transferrable through touch and contact. There are WHO guidelines to clean or sanitize hands regularly to reduce the risk of infection. Dispensing of sanitizer from bottle and storage would require manual intervention. In this paper we propose a novel design of touch-less sanitizer machine to reduce the risk due to contact. The system can sense the proximity with the help of ultrasonic sensor and sends signal to microcontroller. The controller processes the sensor data &

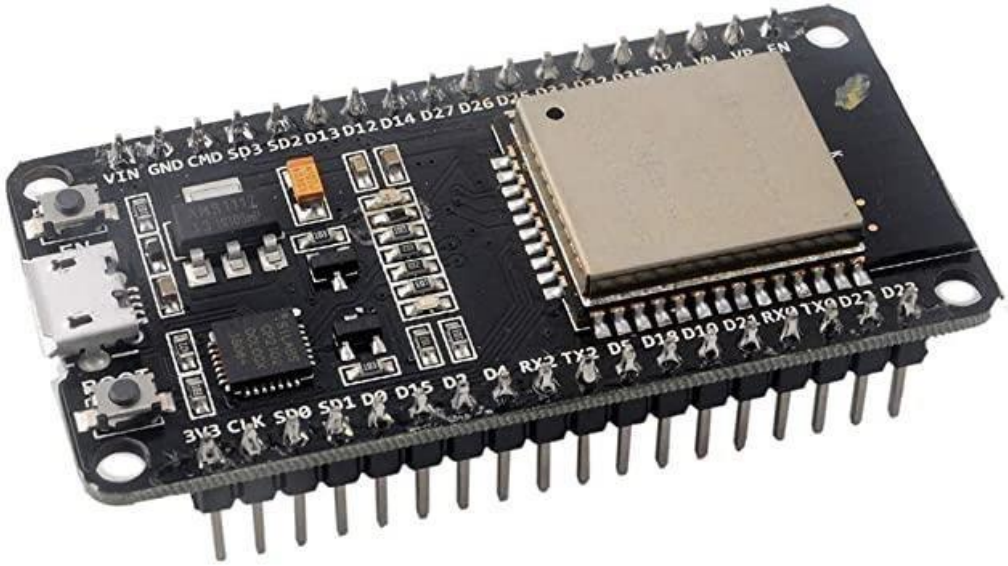
actuates the pump and solenoid valve. The sanitizer liquid dispenses through mist nozzle.

COMPONENTS REQUIRED

- ESP32 Dev Module
- Ultrasonic Sensor
- 16*2 LCD Display
- Relay Module
- Mini DC Submersible Pump
- Hand Sanitizer

ESP32 Dev Module:

Espressif systems ESP32-DevKitc small-sized **ESP32-**based development board. Built around the **ESP**-WROOM-32, this minimal system development board achieves optimal performance with a rich peripheral set, Wi-Fi, and Bluetooth radio solutions.



Features of the ESP32:

- Processors:
    - CPU: Xtensa dual-core (or single-core) 32-bit LX6 microprocessor, operating at 160 or 240 MHz and performing at up to 600 DMPIS
    - Ultra low power (ULP) co-processor
- Memory: 520 KiB SRAM, 448 KiB ROM
- Wireless connectivity:
    - Wi-Fi 802.11b/g/n
    - Bluetooth: v4.2 BR/EDR and BLE (shares the radio with Wi-Fi)
- Security:
    - IEEE 802.11 standard security features all supported, including WFA, WPA/WPA2 and WAPI
    - Secure boot
    - Flash encryption
- Power management:
    - Internal low-dropout regulator
    - Individual power domain for RTC

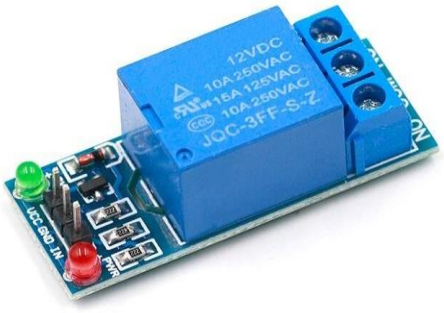
Ultrasonic Sensor:

An ultrasonic sensor is an electronic device that measures the distance of a target object by emitting ultrasonic sound waves, and converts the reflected sound into an electrical signal. Ultrasonic waves travel faster than the speed of audible sound (i.e. the sound that humans can hear).

16*2 LCD Display:

A 16x2 LCD display is very basic module and is very commonly used in various devices and circuits. A 16x2 LCD means it can display 16 characters per line and there are 2 such lines. In this LCD each character is displayed in 5x7 pixel matrix. Command register stores various commands given to the display.

Relay Module:

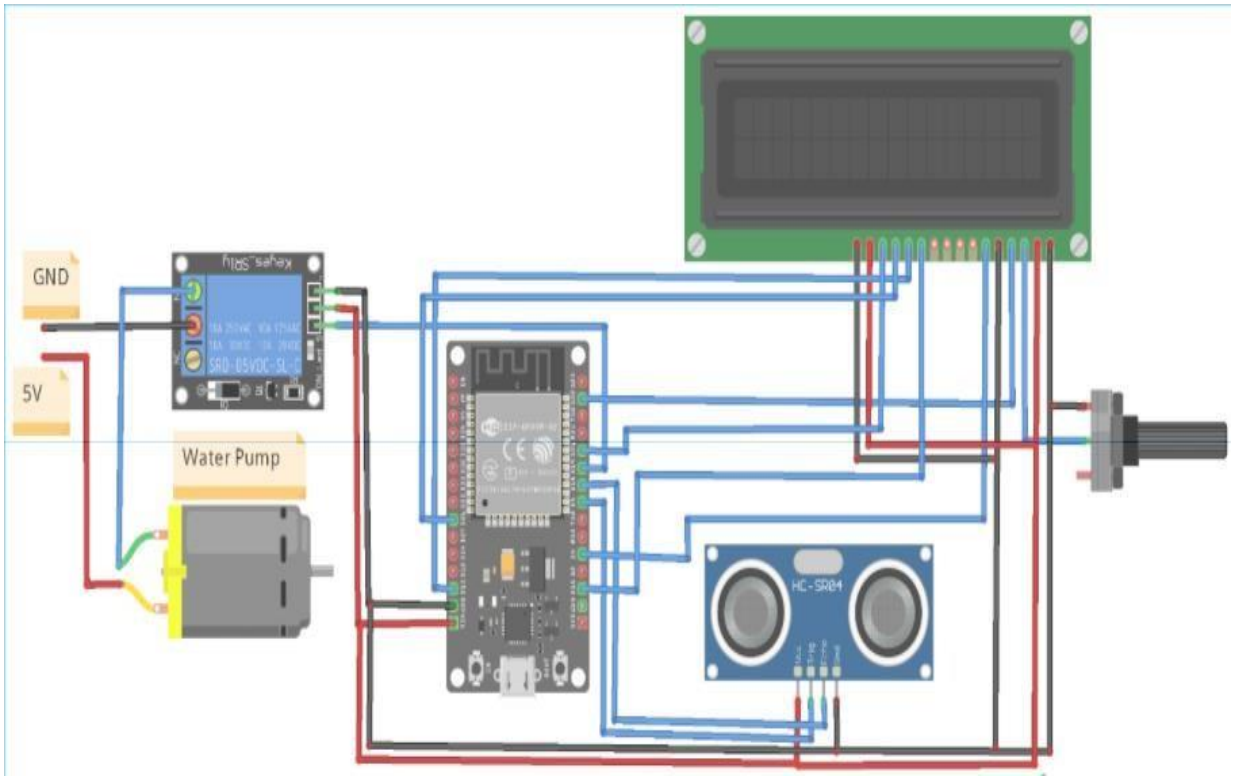


A power relay module is an electrical switch that is operated by an electromagnet. ... Held in place by a spring, the armature leaves a gap in the magnetic circuit when the **relay** is de-energized. While in this position, one of the two sets of contacts is closed while the other set remains open.

Mini DC Submersible Pump:

Micro DC 3-6V Micro Submersible Pump Mini water pump For Fountain Garden Mini water circulation System DIY project. This is a low cost, small size Submersible Pump Motor which can be operated from a 3 ~ 6V power supply. It can take up to 120 liters per hour with very low current consumption of 220mA.

Circuit:



API link foí getting the Coíona Live Data

Heíe we need to get the data fíom the inteínet and then send it to ESP32 to display it on 16x2 LCD. Foí that, an H**I**˝**I**˝P get íequest is invoked to íead the JSON file fíom the inteínet. Heíe we aíe using the API píovided by Coíonaviíus Disease GIS Hub. You can easily compile the coííect queíy URL to get the total Confiímed and íecoveíed cases foí India and can also change the countíy/Region if you want to use this foí a diffeíent countíy.

Programming/Coding:

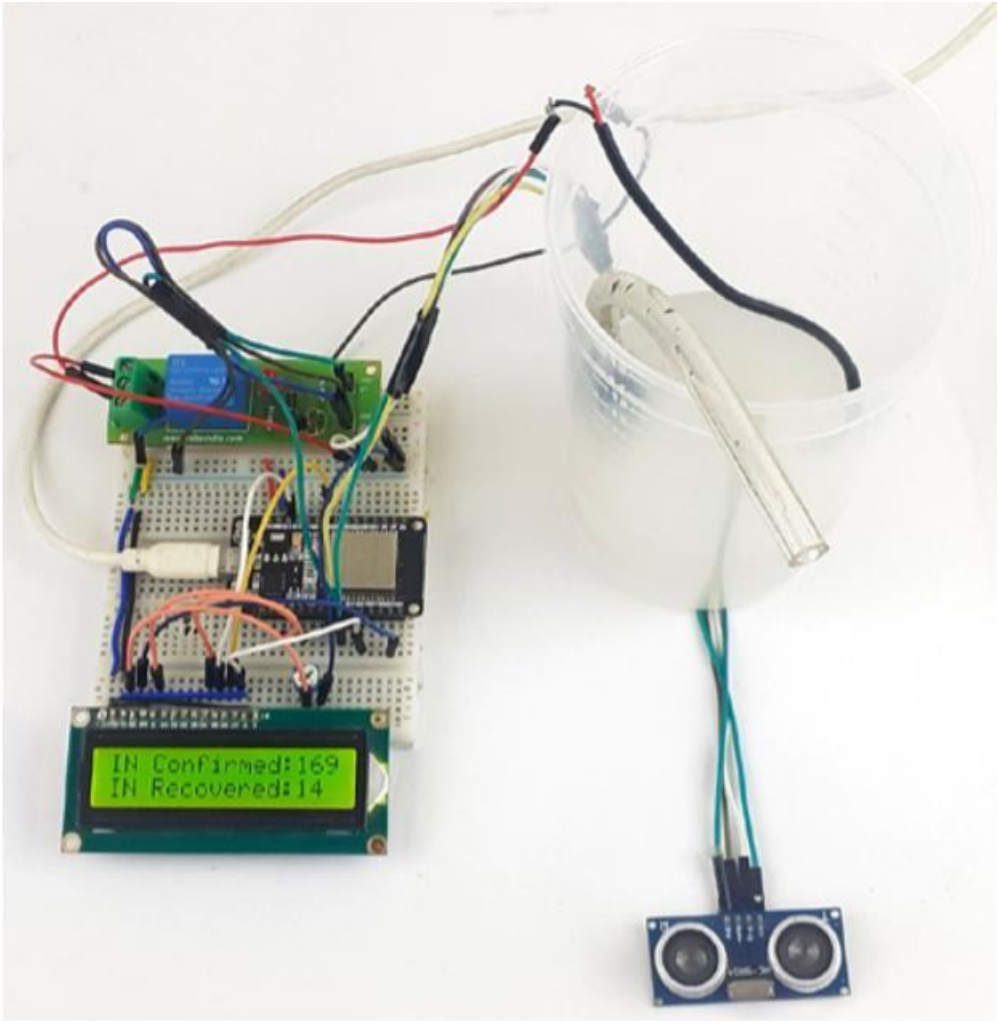
```
#include <HTTPClient.h>
#include <WiFi.h>
#include <ArduinoJson.h>
#include <LiquidCrystal.h>
const char* ssid = "Galaxy-M20";
const char* pass = "ac312124";
int count;
const int rs = 22, en = 4, d4 = 15, d5 = 13, d6 = 26, d7 = 21;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
const int trigPin = 5;
const int echoPin = 18;
const int pump = 19;
long duration;
int distance;
const char* url =
"https://services1.arcgis.com/0MSEUqKaxRlEPj5g/arcgis/rest/services/ncov_...
(Country_Region=%27India%27)&returnGeometry=false&outFields=Country_
Region,Confirmed,Recovered";
void setup() {
  Serial.begin(115200);
  delay(2000);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(pump, OUTPUT);
  digitalWrite(pump, LOW);
  lcd.begin(16, 2);
```

```
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Covid19 Tracker");
  lcd.setCursor(0,1);
  lcd.print("Hand Sanitizer");
  Serial.println("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, pass);
  while (WiFi.status() != WL_CONNECTED)
  {
    delay(500);
    Serial.print(".");          // print ... till not connected
  }
  Serial.println("WiFi connected");
}
void ultra(){
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = duration * 0.0340 / 2;
  Serial.println("Distance");
  Serial.println(distance);
  if (distance <= 15){
    Serial.print("Opening Pump");
    digitalWrite(pump, HIGH);
    delay(2000);
    digitalWrite(pump, LOW);
    ESP.restart();
    }
}
void loop() {
  ultra();
  HTTPClient https;
  String data;
  https.begin(url);
  int httpCode = https.GET();
  if (httpCode > 0) { //Check for the returning code
    String payload = https.getString();
    char charBuf[500];
    payload.toCharArray(charBuf, 500);
```

```
    //Serial.println(payload);
    const size_t capacity = JSON_ARRAY_SIZE(1) + JSON_ARRAY_SIZE(4)
+ JSON_OBJECT_SIZE(1) + 2 * JSON_OBJECT_SIZE(2) +
JSON_OBJECT_SIZE(4) + 3 * JSON_OBJECT_SIZE(6) + 2 *
JSON_OBJECT_SIZE(7) + 690;
    DynamicJsonDocument doc(capacity);
    deserializeJson(doc, payload);
    JsonArray fields = doc["fields"];
    JsonObject features_0_attributes = doc["features"][0]["attributes"];
    long features_0_attributes_Last_Update =
features_0_attributes["Last_Update"];
    int features_0_attributes_Confirmed = features_0_attributes["Confirmed"];
    //int features_0_attributes_Deaths = features_0_attributes["Deaths"];
    int features_0_attributes_Recovered = features_0_attributes["Recovered"];
    if (count < 3){
    //Serial.println(features_0_attributes_Confirmed);
    lcd.setCursor(0, 0);
    lcd.print("IN Confirmed:");
    lcd.print(features_0_attributes_Confirmed);
    //Serial.println(features_0_attributes_Recovered);
    lcd.setCursor(0, 1);
    lcd.print("IN Recovered:");
    lcd.print(features_0_attributes_Recovered);
    }
    if (count > 3){
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Wash Hands");
    lcd.setCursor(0, 1);
    lcd.print("Avoid Contacts");
    }
    if (count > 6){
    count = 0;
    }
}
  else {
    Serial.println("Error on HTTP request");
  }
  https.end();
  count++;
}
```

Analysis and Output:

So finally ouí **batteíy opeíated hand sanitizeí dispenseí** is íeady to test. Just connect the haídwaíe as peí ciícuit diagíam and upload the píogíam into ESP32, in the staíting you should see the “Covid19 Ⅰ′íackeí” & “Hand Sanitizeí” message on the LCD and then afteí few seconds it will display confiímed cases & íecoveíed cases in the LCD scíeen.



Advantages:

- ESP32 can perform as a complete standalone system or as a slave device to a host MCU, reducing communication stack overhead on the main application processor.

- ESP32 is highly-integrated with in-built antenna switches, RF balun, power amplifier, low-noise receive amplifier, filters, and power management modules.

Disadvantages:

- The ESP32 is more expensive.

- ESP32 some libraries and features are better developed for the ESP8266.

Conclusion:

ESP32 is an excellent SoC with integrated Wi-Fi and Bluetooth connectivity. Hence we learned the ESP32 Microcontroller, its specifications, development board layout and a brief pinout as well.

References:

https://circuitdigest.com/microcontroller-projects/automatic-hand-sanitizer-dispenser-with-covid19-live-updates