```
#include <ESP8266WiFi.h>
#include <Firebase_ESP_Client.h>
#include "addons/TokenHelper.h"
#include "addons/RTDBHelper.h"
#include <SPI.h>
#include <U8g2lib.h>
#ifdef U8X8_HAVE_HW_SPI
#include <SPI.h>
#endif
#ifdef U8X8_HAVE_HW_I2C
#include <Wire.h>
#endif

#define TRIGGER_PIN D6
#define ECHO_PIN D5
#define MAX_DISTANCE 200
#define RGB_RED D7
#define RGB_GREEN D8
#define RGB_BLUE D3

U8G2_SSD1306_128X64_NONAME_F_SW_I2C u8g2(U8G2_R0, /* clock=*/ D1, /* data=*/ D2, /* reset=*/
U8X8_PIN_NONE);

#define WIFI_SSID "123456789"
#define WIFI_PASSWORD "123456789"
#define API_KEY "AIzaSyBhsbCD3MOJt43Q0AV1-7_h89v3BKsDePU"
#define DATABASE_URL "https://smart-inhaler-32e40-default-rtdb.firebaseio.com/"

FirebaseData fbdo; FirebaseAuth auth;
FirebaseConfig config; unsigned long
sendDataPrevMillis = 0; bool signupOK =
false;

unsigned long authorizedStartTime = 0; bool
authorizedDisplay = false;

void setup() {  Serial.begin(115200);
pinMode(TRIGGER_PIN, OUTPUT);
pinMode(ECHO_PIN, INPUT);
pinMode(RGB_RED, OUTPUT);
pinMode(RGB_GREEN, OUTPUT);
pinMode(RGB_BLUE, OUTPUT);

 u8g2.begin();
 WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
Serial.print("Connecting to Wi-Fi");  while
```

```
(WiFi.status() != WL_CONNECTED) {
Serial.print("."); delay(300);
 }
 u8g2.clearBu                                    er();
u8g2.setFont(u8g2_font_ncenB08_tr);
u8g2.drawStr(0,    30,    "THANK    YOU!!");
u8g2.sendBu er();

 Serial.println();
 Serial.print("Connected with IP: ");
 Serial.println(WiFi.localIP()); Serial.println();
config.api_key = API_KEY;
config.database_url = DATABASE_URL;

 if (Firebase.signUp(&config, &auth, "", "")) {
Serial.println("Sign-up successful"); signupOK = true;
 } else {
 Serial.printf("Sign-up error: %s\n", config.signer.signupError.message.c_str());
 }

 config.token_status_callback = tokenStatusCallback; // See addons/TokenHelper.h
 Firebase.begin(&config, &auth);
 Firebase.reconnectWiFi(true);
}

void loop() { long duration, distance;
digitalWrite(TRIGGER_PIN, LOW);
delayMicroseconds(2);
digitalWrite(TRIGGER_PIN, HIGH);
delayMicroseconds(10);
digitalWrite(TRIGGER_PIN, LOW); duration
= pulseIn(ECHO_PIN, HIGH);
 distance = (duration / 2) / 29.1; // Convert the distance to centimeters

 Serial.print("Distance: ");
 Serial.print(distance); Serial.println("
cm");

 if (distance < 5) {
 // Display red light digitalWrite(RGB_RED,
HIGH); digitalWrite(RGB_GREEN, LOW);
digitalWrite(RGB_BLUE, LOW); updateDisplay("I
AM FULL"); authorizedStartTime = millis();
authorizedDisplay = true;
 } else if (distance >= 5 && distance <= 8) {
```

```
 // Display blue light  digitalWrite(RGB_RED,
LOW);  digitalWrite(RGB_GREEN, LOW);
digitalWrite(RGB_BLUE, HIGH);  updateDisplay("I
AM NOT SO FULL");
 } else {
 // Display green light  digitalWrite(RGB_RED,
LOW);  digitalWrite(RGB_GREEN, HIGH);
digitalWrite(RGB_BLUE, LOW);  updateDisplay("I
AM HUNGRY");
 }

 if (Firebase.ready() && signupOK && (millis() - sendDataPrevMillis > 1000 || sendDataPrevMillis == 0)) {
sendDataPrevMillis = millis();

 if (Firebase.RTDB.setInt(&fbdo, "main/distance", distance)) {
 Serial.println("Data sent successfully!");
 Serial.println("PATH: " + fbdo.dataPath());
 Serial.println("TYPE: " + fbdo.dataType());
 } else {
 Serial.println("Failed to send data. REASON: " + fbdo.errorReason());
 }
 }

 delay(500); // Delay between readings
}

void    updateDisplay(const    char*    message)    {
u8g2.clearBu                                  er();
u8g2.setFont(u8g2_font_ncenB08_tr);
u8g2.drawStr(0, 30, message);  u8g2.sendBu er();
}
```