ACCELEROMETER BASED ALCOHOL DRINKING ALERT SYSTEM

CODE:

```
#include <ESP8266WiFi.h>
#include <Firebase_ESP_Client.h>
#include "addons/TokenHelper.h"
#include "addons/RTDBHelper.h"
#include <Wire.h>
#include <U8g2lib.h>

#define ADXL335_X A0
#define ADXL335_Y A0
#define ADXL335_Z A0

#define BUZZER_PIN D4

#define SHAKING_THRESHOLD 470 // Adjust this threshold according to your needs

U8G2_SH1106_128X64_NONAME_F_HW_I2C u8g2(U8G2_R0, /* reset=*/
U8X8_PIN_NONE);

#define WIFI_SSID "123456789"
#define WIFI_PASSWORD "123456789"
#define API_KEY "AIzaSyC0gPSHesz3RxIsbFM48OkKK_zCBhfbtmc"
#define DATABASE_URL "https://test-26075-default-rtdb.firebaseio.com/"

FirebaseData fbdo;
FirebaseAuth auth;
FirebaseConfig config;
```

```
unsigned long sendDataPrevMillis = 0;
bool signupOK = false;
String intValue;

void setup() {
  Serial.begin(115200);
  u8g2.begin();
  pinMode(BUZZER_PIN, OUTPUT);
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("Connecting to Wi-Fi");
  while (WiFi.status() != WL_CONNECTED){
    Serial.print(".");
    delay(300);
  }
  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("ok");
    signupOK = true;
  }
  else{
    Serial.printf("%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() {
  int xAccel = analogRead(ADXL335_X);
  int yAccel = analogRead(ADXL335_Y) + 30;
  int zAccel = analogRead(ADXL335_Z) + 28;

  Serial.print("X: "); Serial.print(xAccel);
  Serial.print("\t");
  Serial.print("Y: "); Serial.print(yAccel);
  Serial.print("\t");
  Serial.print("Z: "); Serial.println(zAccel);

  u8g2.clearBuffer();
  u8g2.setFont(u8g2_font_ncenB08_tr);
  u8g2.setCursor(0, 13);
  u8g2.print("X: "); u8g2.print(xAccel);
  u8g2.setCursor(0, 23);
  u8g2.print("Y: "); u8g2.print(yAccel);
  u8g2.setCursor(0, 33);
  u8g2.print("Z: "); u8g2.print(zAccel);
  u8g2.sendBuffer();
  delay(800); // Adjust delay according to your requirements

  // Check if any of the axis values are above the shaking threshold
  if ((xAccel < SHAKING_THRESHOLD) || (yAccel < SHAKING_THRESHOLD) || (zAccel < SHAKING_THRESHOLD)) {
    // If shaking detected, turn on the buzzer
    digitalWrite(BUZZER_PIN, HIGH);
    Serial.println("Drinking Alert!!!....");
    u8g2.setFont(u8g2_font_ncenB08_tr);
    u8g2.setCursor(0, 44);
```

```
   u8g2.print("Drinking Alert!!!....");
   u8g2.sendBuffer();
  } else {
   digitalWrite(BUZZER_PIN, LOW);
   Serial.println("Condition Normal!!!...");
   u8g2.setFont(u8g2_font_ncenB08_tr);
   u8g2.setCursor(0, 40);
   u8g2.print("Condition Normal!!!...");
   u8g2.sendBuffer();
  }
  delay(50);


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


   if (Firebase.RTDB.setFloat(&fbdo, "mainbucket/xAccel",xAccel)){
    Serial.println("PATH: " + fbdo.dataPath());
    Serial.println("TYPE: " + fbdo.dataType());
   }
   else {
    Serial.println("Failed REASON: " + fbdo.errorReason());
   }
   delay(100);
   if (Firebase.RTDB.setFloat(&fbdo, "mainbucket/xAccel",xAccel)){
    Serial.println("PATH: " + fbdo.dataPath());
    Serial.println("TYPE: " + fbdo.dataType());
   }
   else {
    Serial.println("Failed REASON: " + fbdo.errorReason());
   }
```

```
    delay(100);
    if (Firebase.RTDB.setFloat(&fbdo, "mainbucket/yAccel",yAccel)){
      Serial.println("PATH: " + fbdo.dataPath());
      Serial.println("TYPE: " + fbdo.dataType());
    }
    else {
      Serial.println("Failed REASON: " + fbdo.errorReason());
    }
    delay(100);
    if (Firebase.RTDB.setFloat(&fbdo, "mainbucket/zAccel",zAccel)){
      Serial.println("PATH: " + fbdo.dataPath());
      Serial.println("TYPE: " + fbdo.dataType());
    }
    else {
      Serial.println("Failed REASON: " + fbdo.errorReason());
    }
    delay(1000);
}
}
```