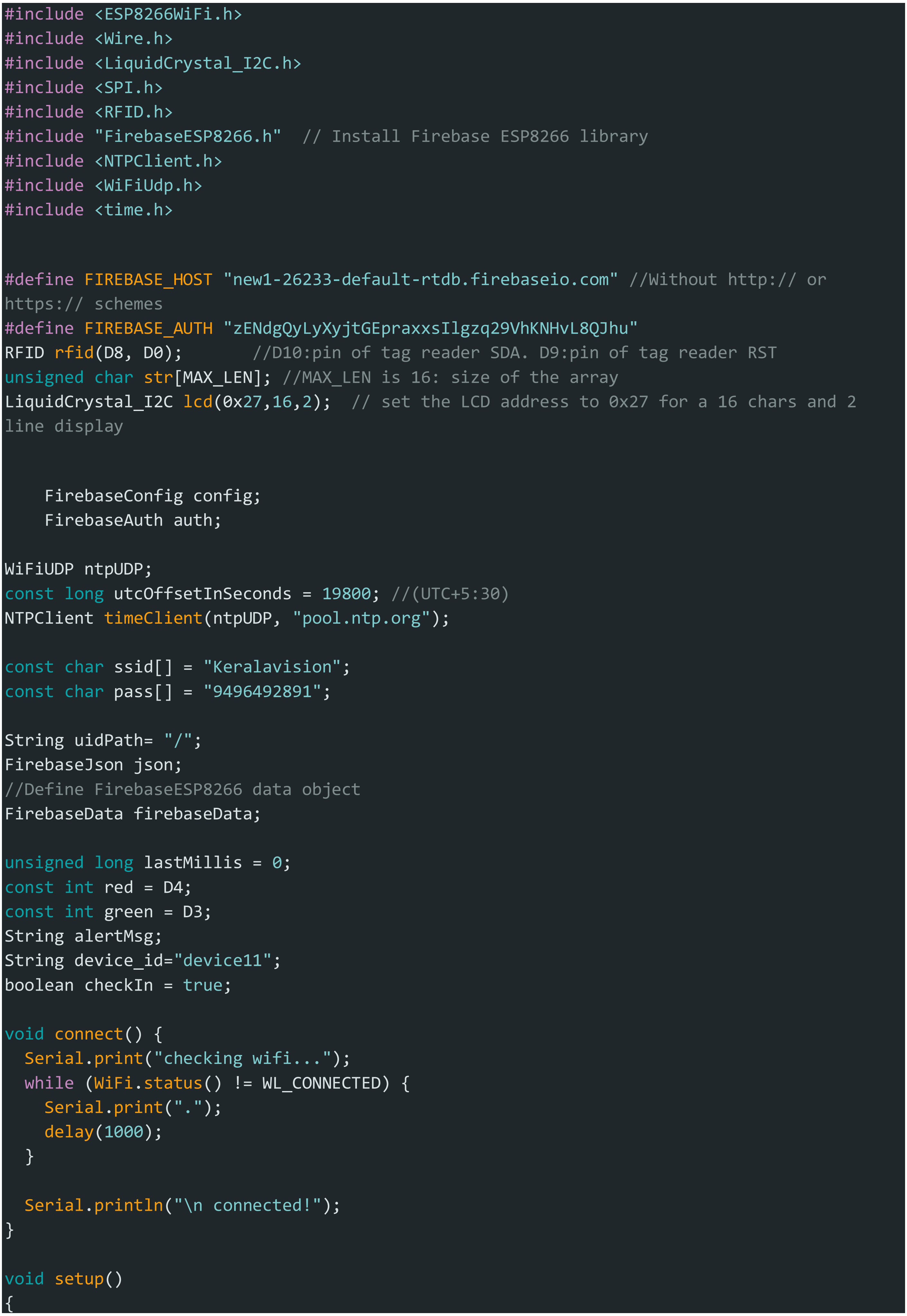

```
#include <ESP8266WiFi.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <SPI.h>
#include <RFID.h>
#include "FirebaseESP8266.h"  // Install Firebase ESP8266 library
#include <NTPClient.h>
#include <WiFiUdp.h>
#include <time.h>


#define FIREBASE_HOST "new1-26233-default-rtdb.firebaseio.com" //Without http:// or https:// schemes
#define FIREBASE_AUTH "zENdgQyLyXyjtGEpraxxsIlgzq29VhKNHvL8QJhu"
RFID rfid(D8, D0);       //D10:pin of tag reader SDA. D9:pin of tag reader RST
unsigned char str[MAX_LEN]; //MAX_LEN is 16: size of the array
LiquidCrystal_I2C lcd(0x27,16,2);  // set the LCD address to 0x27 for a 16 chars and 2 line display


    FirebaseConfig config;
    FirebaseAuth auth;

WiFiUDP ntpUDP;
const long utcOffsetInSeconds = 19800; //(UTC+5:30)
NTPClient timeClient(ntpUDP, "pool.ntp.org");

const char ssid[] = "Keralavision";
const char pass[] = "9496492891";

String uidPath= "/";
FirebaseJson json;
//Define FirebaseESP8266 data object
FirebaseData firebaseData;

unsigned long lastMillis = 0;
const int red = D4;
const int green = D3;
String alertMsg;
String device_id="device11";
boolean checkIn = true;

void connect() {
  Serial.print("checking wifi...");
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(1000);
  }

  Serial.println("\n connected!");
}

void setup()
{
  Serial.begin(115200);
  WiFi.begin(ssid, pass);

  pinMode(red, OUTPUT);
  pinMode(green, OUTPUT);
  lcd.init();                      // initialize the lcd
  lcd.clear();
  lcd.backlight();

  SPI.begin();
  rfid.init();

  timeClient.begin();
  timeClient.setTimeOffset(utcOffsetInSeconds);
  connect();
  //Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);

   config.database_url = FIREBASE_HOST;
  config.signer.tokens.legacy_token = FIREBASE_AUTH;
  Firebase.begin(&config, &auth);

  Firebase.reconnectWiFi(true);
}
void checkAccess (String temp)    //Function to check if an identified tag is registered to allow access
{
    lcd.setCursor(1,0);
    lcd.print("SCAN YOUR RFID");

    if(Firebase.getInt(firebaseData, uidPath+"/users/"+temp)){

      if (firebaseData.intData() == 0)         //If firebaseData.intData() == checkIn
      {
          alertMsg="CHECKING IN";
          lcd.setCursor(2,1);
          lcd.print(alertMsg);
          delay(1000);

          json.add("time", String(timeClient.getFormattedDate()));
          json.add("id", device_id);
          json.add("uid", temp);
          json.add("status",1);

          Firebase.setInt(firebaseData, uidPath+"/users/"+temp,1);

          if (Firebase.pushJSON(firebaseData, uidPath+ "/attendence", json)) {
            Serial.println(firebaseData.dataPath() + firebaseData.pushName());
          } else {
            Serial.println(firebaseData.errorReason());
          }
      }
      else if (firebaseData.intData() == 1)   //If the lock is open then close it
      {
          alertMsg="CHECKING OUT";
          lcd.setCursor(2,1);
          lcd.print(alertMsg);
          delay(1000);

          Firebase.setInt(firebaseData, uidPath+"/users/"+temp,0);

          json.add("time", String(timeClient.getFormattedDate()));
          json.add("id", device_id);
          json.add("uid", temp);
          json.add("status",0);
          if (Firebase.pushJSON(firebaseData, uidPath+ "/attendence", json)) {
            Serial.println(firebaseData.dataPath() + firebaseData.pushName());
          } else {
            Serial.println(firebaseData.errorReason());
          }
      }

    }
    else
    {
      Serial.println("FAILED");
      Serial.println("REASON: " + firebaseData.errorReason());
    }
}
void loop() {
  timeClient.update();
  if (rfid.findCard(PICC_REQIDL, str) == MI_OK)    //Wait for a tag to be placed near the reader
  {
    Serial.println("Card found");
    String temp = "";                              //Temporary variable to store the read RFID number
    if (rfid.anticoll(str) == MI_OK)               //Anti-collision detection, read tag serial number
    {
      Serial.print("The card's ID number is : ");
      for (int i = 0; i < 4; i++)                  //Record and display the tag serial number
      {
        temp = temp + (0x0F & (str[i] >> 4));
        temp = temp + (0x0F & str[i]);
      }
      Serial.println (temp);
      checkAccess (temp);     //Check if the identified tag is an allowed to open tag
    }
    rfid.selectTag(str); //Lock card to prevent a redundant read, removing the line will make the sketch read cards continually
  }
  rfid.halt();

  lcd.setCursor(1,0);
  lcd.print("SCAN YOUR RFID");
  lcd.setCursor(2,1);
  lcd.print("GATE CLOSE");
  delay(500);
  lcd.clear();
}
```