Humidor per conservazione di sigari

Francesco Bizzarri

Ottobre 2022

Relazione del progetto svolto per il corso di Laboratorio di Internet

1 Il progetto

Il progetto consiste nel creare una scatola umidificata per la conservazione dei sigari. I sigari, per essere conservati al meglio, dovrebbero essere tenuti in un ambiente a temperatura e umidità controllata. Generalmente la temperatura è compresa tra 18° e 25° mentre il tasso di umidità tra 68% e 74%. L'idea del progetto è creare un humidor "smart", che permette di verificare i valori di temperatura e umidità da remoto.

1.1 Materiale usato

- Scatola in legno
- Sensore di temperatura e umidità DHT11
- Scheda di sviluppo ESP32
- Cavi jumper
- Powerbank

1.2 Risultato finale

2 Il codice

Tutto il codice è stato scritto utilizzando Arduino IDE.

2.1 Librerie

Con questo frammento di codice vengono importate le librerie usate nel progetto. Come è intuibile dal nome le librerie *WiFi.h* e *FirebaseESP32.h* servono per la connessione alla rete Wi-Fi e al database real-time Firebase. La libreria *DHT.h* invece fornisce le funzioni per interfacciarsi al sensore.

```
#include <DHT.h>
#include <WiFi.h>
#include <FirebaseESP32.h>
```

2.2 Variabili globali

In questa sezione vengono definiti: il pin da cui ricevere i dati del sensore, i dettagli del database, i dettagli della rete a cui connettersi e, infine, il range di valori ideali per temperatura e umidità.

```
#define DHTPIN 5
#define DHTTYPE DHT11
#define FIREBASE_HOST "my_firebase_link"
#define FIREBASE_AUTH "my_firebase_password"
const char* SSID = "my_wifi_name";
const char* PASSWORD = "my_wifi_password";
int MIN_IDEAL_TEMPERATURE = 16;
int MAX_IDEAL_TEMPERATURE = 25;
int MIN_IDEAL_HUMIDITY = 68;
int MAX_IDEAL_HUMIDITY = 75;
```

2.3 Creazione oggetti

```
WiFiServer server(80); // Web server
String request; // Stringa per richiesta HTTP
DHT dht(DHTPIN, DHTTYPE); // Creazione istanza sensore
FirebaseData fData; // Per comunicare con il database
FirebaseJson json; // Per comunicare con il database
size_t updateTime; // Per aggiornare i valori nel database
```

2.4 Setup

Il codice di setup è quello che viene eseguito una sola volta quando la scheda viene accesa. Si occupa di aprire le connessioni necessarie a Wi-Fi e database e avviare il Web Server ed il sensore.

```
void setup() {

  // Connessione Wi-Fi
  Serial.begin(115200);
  Serial.print("Connecting to: ");
  Serial.print(SSID);
  WiFi.begin(SSID, PASSWORD);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
```

```
  // Web server
  server.begin();

  // Connessione al database
  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
  Firebase.reconnectWiFi(true);
  Firebase.setReadTimeout(fData, 1000*60);
  Firebase.setwriteSizeLimit(fData, "tiny");
  updateTime = millis();

  // Avviamento sensore
  dht.begin();
}
```

2.5 Loop

In questa sezione c'è il codice che viene eseguito ciclicamente dalla scheda.

```
void loop() {
  // Lettura valori sensore
  float t = dht.readTemperature();
  float h = dht.readHumidity();

  // Aggiornamento database ogni 2 secondi
  if (millis() - updateTime > 2000) {
    json.set("/temperature", t);
    json.set("/humidity", h);
    Firebase.set(fData, "/values", json);
    updateTime = millis();
  }

  // Ascolto per eventuali nuovi clients
  WiFiClient client = server.available();

  if (client) {
    while (client.connected()) {
      if (client.available()) {

        // Lettura richiesta
        char c = client.read();
        request += c;

        // Codice che gestisce la nostra semplice richiesta
        if (c == '\n') {
          String temperature = String(t, 1);
          String humidity = String(h, 1);

          // Codice HTML
```

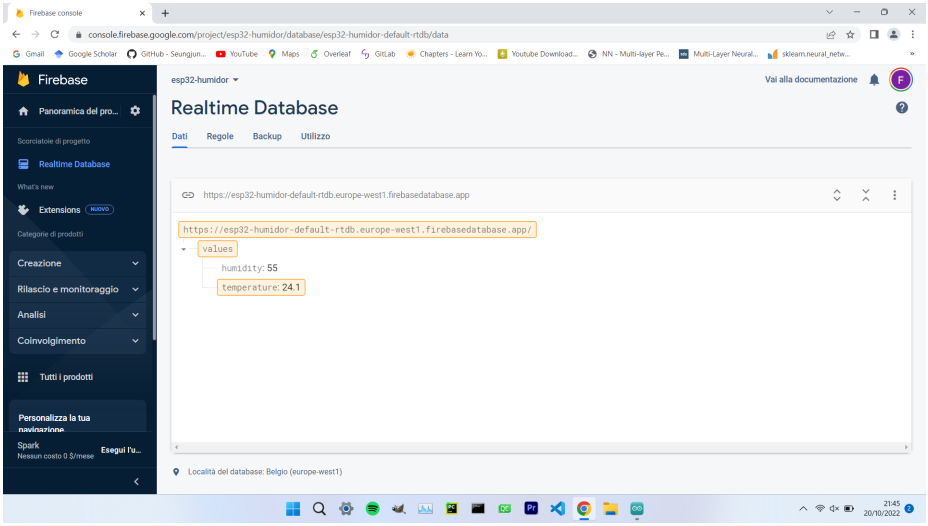
```
            client.println("<!DOCTYPE html><html>");
            client.println("<head><meta name=\"viewport\"
                content=\"width=device-width, initial-scale=1\">");
            client.println("<link rel=\"icon\" href=\"data:,\">");
            client.println("<style>html { font-family: Helvetica; display:
                inline-block; margin: 0px auto; text-align: center;}");
            client.println(".button {background-color: green; border: none;
                color: white; padding: 16px 40px;");
            client.println("text-decoration: none; font-size: 30px; margin:
                2px;}");
            client.println(".buttonRed {background-color:
                red;}</style></head>");
            client.println("<body><h1>HUMIDOR WEB SERVER</h1>");
            client.println("<p>TEMPERATURE</p>");

            // Cambia il colore del bottone a rosso se i valori non sono
                nel range ideale
            if (t <= MAX_IDEAL_TEMPERATURE && t >= MIN_IDEAL_TEMPERATURE) {
              client.println("<p><button
                  class=\"button\">"+temperature+"</button></p>");
            } else {
              client.println("<p><button class=\"button
                  buttonRed\">"+temperature+"</button></p>");
            }
            client.println("<p>HUMIDITY</p>");
            if (h <= MAX_IDEAL_HUMIDITY && h >= MIN_IDEAL_HUMIDITY) {
              client.println("<p><button
                  class=\"button\">"+humidity+"</button></p>");
            } else {
              client.println("<p><button class=\"button
                  buttonRed\">"+humidity+"</button></p>");
            }

            client.println("</body></html>");
            break;
          }
        }
      }
      client.stop();
    }
}
```

3 Database real time

È possibile visualizzare i valori attuali di temperatura e umidità connettendosi a Firebase da remoto. Questa è la schermata visualizzata:



4 Web Server

È anche possibile visualizzare gli stessi risultati connettendosi al web server attraverso l'indirizzo IP stampato in console. L'importante è esseri collegati alla stessa rete Wi-Fi a cui è collegato il sensore. Il bottone su cui viene visualizzato il valore è verde se il valore ricade dentro dal range ideale, rosso altrimenti.