

# InstantMETEO

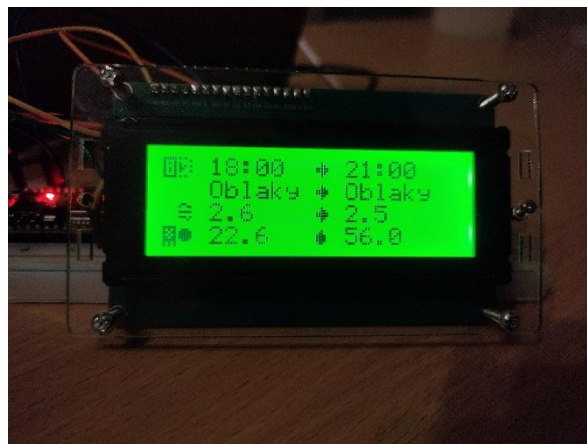
**Autor:** Mgr. Tomáš Krško

**Ročník:** 2RŠI

**Rok:** 2024/2025

## Cieľ projektu

Projekt InstantMETEO je navrhnutý na rýchle zobrazovanie predpovede počasia na nasledujúce 3 až 6 hodín. Umožňuje človeku, ktorý sa ponáhl'a z domu / kancelárie, okamžite vidieť dôležité informácie o počasi bez potreby vytiahnuť mobil alebo hľadať informácie online. Displej zobrazí teplotu, tlak, vlhkosť, vietor a popis počasia. Zároveň DHT sensor sa postará o zmeranie teploty v priestore, kde sa človek nachádza.

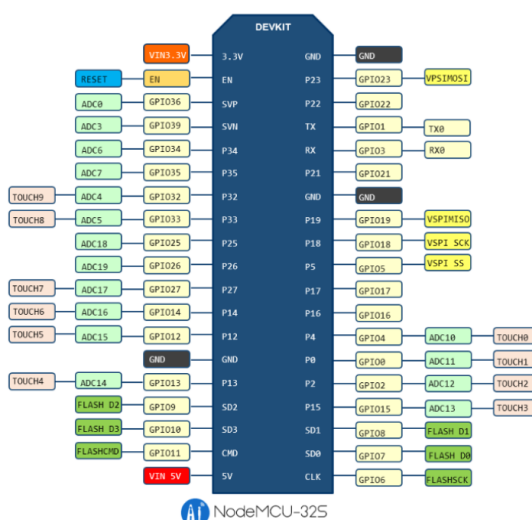
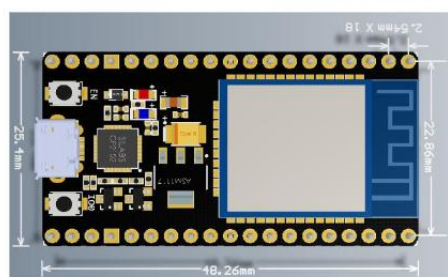


## Použitie v praxi

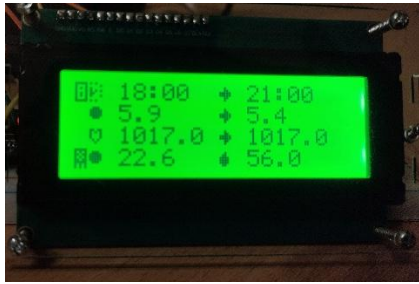
- Rýchla kontrola počasia pri odchode z domu
- Možnosť lepšieho rozhodovania o vhodnom oblečení a vybavení
- Vhodné pre domácnosti, kancelárie

## Komponenty (hardware):

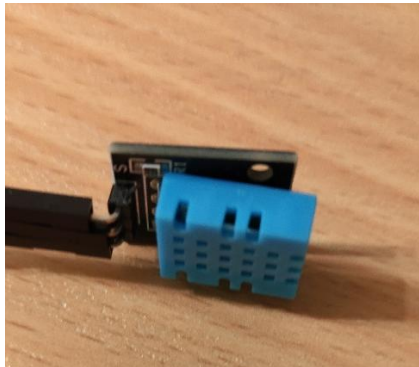
### 1. ESP32 (NodeMCU-32S) ([Datasheet](#))



2. LCD displej 20x4 s I2C ([Datasheet](#))



3. DHT11 (teplota a vlhkosť) – prípadne DHT22

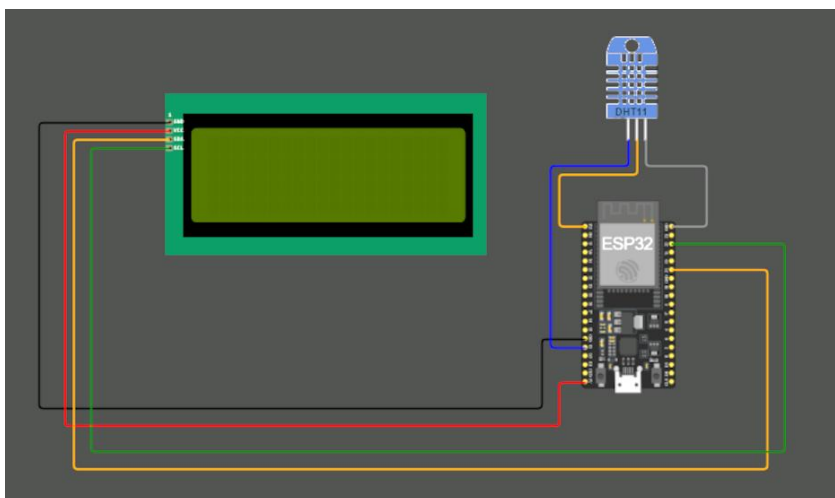


4. BreadBoard

5. USB kábel

6. Prepojovacie vodiče

**Zapojenie komponentov:**



Používame ESP32, DHT11 senzor na meranie teploty, vlhkosti a LCD displej.

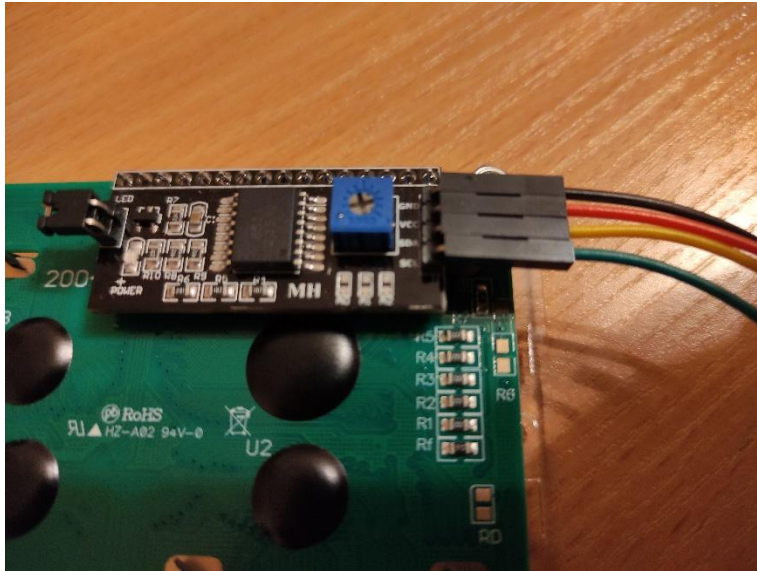
## Zapojenie LCD panel I2C

GND (čierny vodič) → GND na ESP32

VCC (červený vodič) → 5V na ESP32

SDA (oranžový / žltý vodič) → Pin GPIO 21 na ESP32

SCL (zelený vodič) → Pin GPIO 22 na ESP32

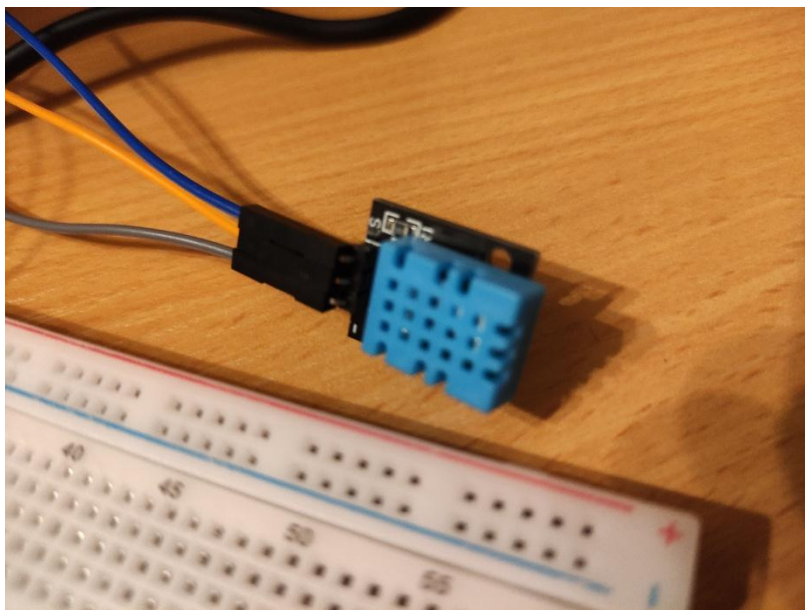


## DHT11 I2C

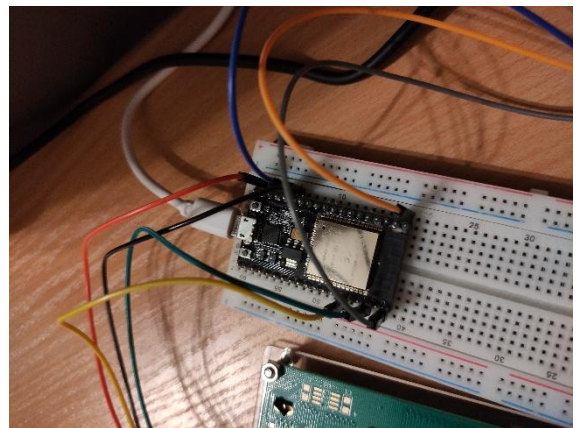
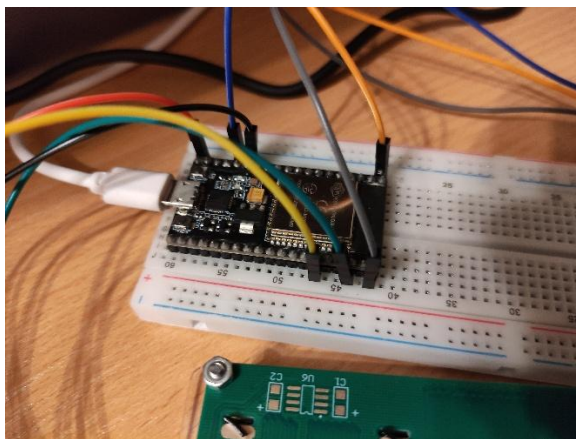
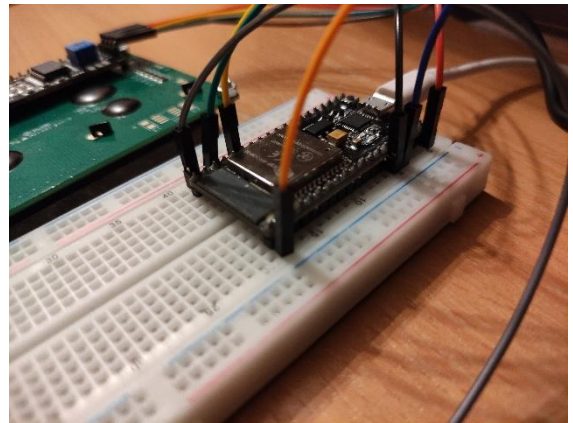
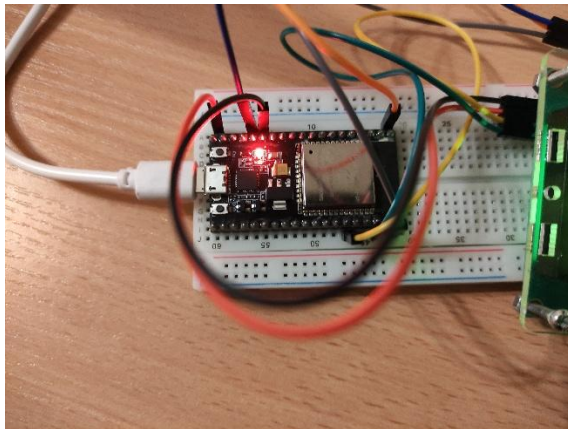
GND (sivý vodič) → GND na ESP32

VCC (oranžový) → 3.3V na ESP32

DATA (modrý vodič) → Pin GPIO 13 na ESP32



## Zapojenie ESP32



### Software:

Pred nahratím programu (a spustením) je potrebné vykonať 3. kroky

1. zadať prístupové údaje pre WIFI sieť (riadky 11 a 12)
2. získať API Key z openweather.com (zadarmo registrácia) (riadok 15)
3. určiť si, pre ktoré mesto ideme zisťovať údaje (riadok 16) – nezabudnúť doplniť kód krajiny za mesto (SK pre slovensko)



Rozpis programu:

Riadky 1 až 8 – definícia podporných knižníc

```
1  #include <Wire.h>
2  #include <LiquidCrystal_I2C.h>
3  #include <WiFi.h>
4  #include <HTTPClient.h>
5  #include <ArduinoJson.h>
6  #include "DHT.h"
7  #include "lcd_fonty.h"
8  #include "preklad.h"
9
```

Riadky 10 – 22 – parametre ako názov a heslo pre WIFI či kľúč k API

```
1  #include <Wire.h>
2  #include <LiquidCrystal_I2C.h>
3  #include <WiFi.h>
4  #include <HTTPClient.h>
5  #include <ArduinoJson.h>
6  #include "DHT.h"
7  #include "lcd_fonty.h"
8  #include "preklad.h"
9
10 // Wi-Fi nastavenia
11 const char* ssid = "ZADAJ-SSID-SIETE";
12 const char* password = "ZADAJ-HESLO-SIETE";
13
14 // OpenWeatherMap API
15 const String apiKey = "ZADAJ-API-KLÚČ"; // Získaj na OpenWeatherMap
16 const String city = "Martin,SK"; // Mesto, ktoré chceš sledovať (nezabudnuť na kód k
17 const String units = "metric"; // Jednotky: "metric" pre °C, "imperial" pre °F
18 const String lang = "en"; // Jazyk odpovede (napr. "sk" pre slovenčinu - pozd
19
20 // URL pre API
21 String weatherUrl = "http://api.openweathermap.org/data/2.5/forecast?q=" + city +
22                    "&appid=" + apiKey + "&units=" + units + "&lang=" + lang + "&cnt=3";
23
```

Riadky 24 – 31 – Definovanie parametrov pre senzor DHT11 a LCD display

```
24 // DHT Senzor
25 #define DHTPIN 13 // Ktorý pin ovláda DHT senzor
26 #define DHTTYPE DHT11 // Typ senzora DHT (DHT22 alebo DHT11)
27 DHT dht(DHTPIN, DHTTYPE); // zadefinovanie DHT senzora
28
29 // Inicializácia LCD displeja (16x4)
30 LiquidCrystal_I2C lcd(0x27, 16, 4);
31
```

Riadky 32 – 56 – „Procedúry“ – clearLCDLine – vymaže riadok na LCD a LCDWriteLine  
vypíše riadok

```
31
32 // Procedúra na vymazanie riadku s parametrom
33 void clearLCDLine(int line)
34 {
35     lcd.setCursor(0,line);
36     for(int n = 0; n < 20; n++) // 20 je počet symbolov v jednom riadku (20x2 alebo 20x1)
37     {
38         lcd.print(" ");
39     }
40 }
41
42 // Procedúra na výpis jednotlivých hľadnot na display s parametrami
43 void LCDWriteLine(int line, int pic, float param1, String param2, float param3)
44 {
45     clearLCDLine(line);
46     lcd.setCursor(1, line); // nastavenie kurzora na 0, riadok
47     lcd.write(byte(pic)); // vykreslenie symbolu (pic)
48     lcd.setCursor(3, line); // nastavenie kurzora pre zaciatok vypisu hodnoty 1. pa
49     lcd.print(param1, 1); // napisanie parametra
50     // lcd.print(param2); // vypisanie jednotky
51     lcd.setCursor(10, line); // nastavenie na poziciu oddelovaca (sipky)
52     lcd.write(byte(3)); // napisanie oddelovaca (sipky)
53     lcd.setCursor(12, line); // nastavenie kurzora na zaciatok pisania 2. udaju
54     lcd.print(param3,1); // 2. udaj
55     // lcd.print(param2); // jednotky druhého udaju
56 }
```

Riadky 59 – 77 – Inicializácia LCD a DHT – zároveň pre LCD sa definujú špeciálne znaky

```
58 // Štart
59 void setup() {
60     Serial.begin(115200);
61
62     // DHT senzor inicializácia
63     dht.begin();
64
65     // Inicializácia LCD
66     lcd.init();
67     lcd.backlight();
68     lcd.createChar(0, Time);
69     lcd.createChar(1, Sun);
70     lcd.createChar(2, Humid);
71     lcd.createChar(3, Arrow);
72     lcd.createChar(4, Internet);
73     lcd.createChar(5, Wind);
74     lcd.createChar(6, Heart);
75     lcd.createChar(7, Sensor);
76 }
```

Riadky 77 – 100 – Vypísanie úvodnej správy na LCD a pripájanie na internet

```
77 // Úvodný výpis na LCD
78 lcd.setCursor(0, 0);
79 lcd.print("*** InstantMeteo ***");
80 lcd.setCursor(4,2);
81 lcd.print("Pripajanie...");
82 delay(5000);
83
84 // Pripojenie na Wi-Fi
85 WiFi.begin(ssid, password);
86 while (WiFi.status() != WL_CONNECTED) {
87     delay(500);
88     Serial.print(".");
89 }
90 Serial.println("\nWi-Fi pripojené!");
91 lcd.setCursor(2, 2);
92 lcd.print("WiFi pripojene! ");
93 lcd.setCursor(1, 3);
94 lcd.print("IP:");
95 lcd.print(WiFi.localIP());
96 delay(5000);
97 }
98
```

Riadky 100 – 134 – Preberanie JSON súboru a jeho čítanie do premenných, ktoré sa neskôr  
použijú

```
99 void loop() {
100     // Načítanie údajov z API
101     if ((WiFi.status() == WL_CONNECTED)) {
102         HTTPClient http;
103         http.begin(weatherUrl); // Inicializácia požiadavky
104         int httpCode = http.GET(); // Odoslanie GET požiadavky
105
106         if (httpCode > 0) { // Ak je odpoveď kladná
107             String payload = http.getString();
108             Serial.println("Úspešná odpoveď z API:");
109             Serial.println(payload);
110
111             // Spracovanie JSON dát
112             DynamicJsonDocument doc(1024 * 5);
113             DeserializationError error = deserializeJson(doc, payload);
114
115             if (!error) {
116                 // Načítanie predpovede na nasledujúce 3 hodiny
117                 String time = doc["list"][0]["dt_txt"]; // Čas predpovede (xx:yy)
118                 float temp = doc["list"][0]["main"]["temp"]; // Teplota (celcius)
119                 float pressure = doc["list"][0]["main"]["pressure"]; // Tlak (hPa)
120                 float humidity = doc["list"][0]["main"]["humidity"]; // Vlhkosť (%)
121                 const char* description = doc["list"][0]["weather"][0]["main"]; // Popis
122                 float wind = doc["list"][0]["wind"]["speed"]; // Vietor (m/s)
123
124                 // Načítanie predpovede na nasledujúcich 6 hodín
125                 String timeForw = doc["list"][1]["dt_txt"]; // Čas predpovede
126                 float tempForw = doc["list"][1]["main"]["temp"]; // Teplota
127                 float pressureForw = doc["list"][1]["main"]["pressure"]; // Tlak (hPa)
128                 float humidityForw = doc["list"][1]["main"]["humidity"]; // Vlhkosť
129                 const char* descriptionForw = doc["list"][1]["weather"][0]["main"]; // Popis
130                 float windForw = doc["list"][1]["wind"]["speed"]; // Vietor (m/s)
131
132                 // Preklad počasia
133                 const char* popisPocasia = translate(description);
134                 const char* popisPocasiaForw = translate(descriptionForw);
135
```

Riadky 136 až 158 – vypisujú sa do konzoly zistené údaje

```
136 // Vypis do konzoly
137 Serial.println("\nPredpoveď na 3 hodiny dopredu:");
138 Serial.println("Čas: " + time + " -----> " + timeForw);
139 Serial.print("Teplota: ");
140 Serial.print(temp);
141 Serial.print(" C -----> ");
142 Serial.print(tempForw);
143 Serial.println(" C");
144 Serial.print("Tlak: ");
145 Serial.print(pressure);
146 Serial.print(" hPa -----> ");
147 Serial.print(pressureForw);
148 Serial.println(" hPa");
149 Serial.print("Vietor: ");
150 Serial.print(wind);
151 Serial.print(" m/s -----> ");
152 Serial.print(windForw);
153 Serial.println(" m/s");
154 Serial.print("Popis:");
155 Serial.print(popisPocasia);
156 Serial.print("----->");
157 Serial.println(popisPocasiaForw);
158
```



```

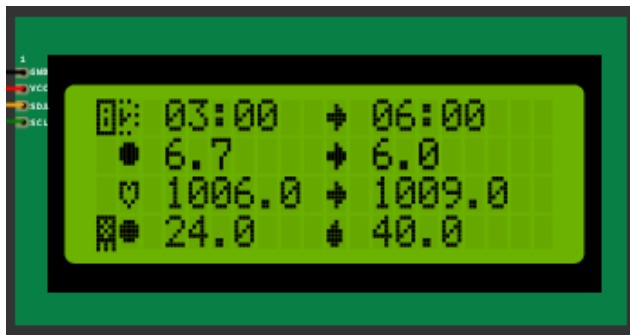
158
159 // Výstup na LCD displej
160 lcd.clear();
161
162 // Zobrazenie casu (0. staly riadok)
163 lcd.setCursor(0, 0);
164 lcd.write(byte(4));
165 lcd.write(byte(0));
166 lcd.print(" " + time.substring(11, 16));
167 lcd.setCursor(10, 0);
168 lcd.write(byte(3));
169 lcd.print(" " + timeForw.substring(11, 16)); // skrat casovy udaj aby sa zmestil
170
171 // Zobrazenie senzora (3. staly riadok)
172 float humidityDHT = dht.readHumidity();
173 float temperatureDHT = dht.readTemperature();
174 clearLCDLine(2);
175 lcd.setCursor(0,3);
176 lcd.write(7);
177 lcd.write(1);
178 lcd.print(" ");
179 lcd.print(temperatureDHT,1);
180 lcd.setCursor(10,3);
181 lcd.write(2);
182 lcd.print(" ");
183 lcd.print(humidityDHT,1);
184
185 // opakovanie 20x
186
187 for (int i = 0; i <= 20; i++) {
188 // zamena udajov (1. a 2. riadok) - teplota a vlhkost
189 LCDWriteLine(1,1, temp, "", tempForw);
190 LCDWriteLine(2,2, humidity, "", humidityForw);
191 delay(10000);
192 // zamena udajov (1. a 2. riadok) - popis pocasia a vietor
193 clearLCDLine(1);
194 lcd.setCursor(3,1);
195 lcd.print(popisPocasia);
196 lcd.setCursor(10,1);
197 lcd.write(3);
198 lcd.print(" ");
199 lcd.print(popisPocasiaForw);
200 LCDWriteLine(2,5, wind, "", windForw);
201 delay(10000);
202 // zamena udajov (1. a 2. riadok) - teplota a tlak
203 LCDWriteLine(1,1, temp, "", tempForw);
204 LCDWriteLine(2,6, pressure, "", pressureForw);
205 delay(10000);
206 }
207 } else {
208 Serial.println("Chyba pri dekódovaní JSON.");
209 lcd.clear();
210 lcd.setCursor(0, 0);
211 lcd.print("Chyba JSON");
212 }
213 } else {
214 Serial.print("HTTP požiadavka zlyhala. Kód: ");
215 Serial.println(httpCode);
216 lcd.clear();
217 lcd.setCursor(0, 0);
218 lcd.print("HTTP chyba:");
219 lcd.setCursor(0, 1);
220 lcd.print(httpCode);
221 }
222
223 http.end(); // Ukončenie požiadavky
224 }
225
226 }

```

## POPIS ROZHRANIA A FUNKCIÍ:

Po úspešnom nahratí do ESP32 Vás privíta obrazovka s názvom aplikácie a oznamom o pripájaní sa na WIFI sieť. V prípade, že sa ESP32 nepripojí do WIFI siete, program ďalej nepokračuje. Naopak, ak by náhodou nefungoval DHT senzor, tak program ide ďalej, ale bude pri senzore písať n/a.

Po úspešnom pripojení dôjde k prvému načítaniu dát a ich zobrazenie na display.



1. riadok – čas o 3 až 6 hodín (pod časom sú hodnoty jednotlivých parametrov v danú hodinu)

2. a 3 riadok – hodnoty

4. riadok – hodnoty zo senzora DHT

1006.0 + 1009.0

tlak v hPa

6.7 + 6.0

teplota v stupňoch celzia

72.0 + 85.0

vlhkosť vzduchu v percentách

1.6 + 2.2

rýchlosť vetra v m/s

Dazd + Dazd

počasie

K obnoveniu informácií prichádza okolo každých 10 minút.

Možné problémy:

Otázka: Počas obnovovania nastala chyba http

Odpoveď: ESP sa bude stále snažiť pripájať, aby došlo k načítaniu dát

Otázka: Prvý časový údaj už nie je posunutý o 3 hodiny, ale len o 2 (alebo 1), či o hodinu dozadu.

Odpoveď: Treba počkať na obnovenie dát (1 hodina dozadu znamená, že dáta idú podľa letného času).

Zdroje:

**OpenWeatherMap** - <https://openweathermap.org/>

**Dokumentácia ESP32** - [https://docs.ai-thinker.com/\\_media/esp32/docs/nodemcu-32s\\_product\\_specification.pdf](https://docs.ai-thinker.com/_media/esp32/docs/nodemcu-32s_product_specification.pdf)

**NexTech Arduino**: [https://www.nextech.sk/clanky/tag\\_search/ESP32](https://www.nextech.sk/clanky/tag_search/ESP32)

**ESP32 Tutorials**: <https://esp32io.com/esp32-tutorials>

**DFRobot ESP32 and Andruino**: <https://www.dfrobot.com/blog-1578.html>