Katolícka univerzita v Ružomberku

Pedagogická fakulta

Ovládaný teplomer cez web na

reguláciu teploty

**Študijný program**: Učiteľstvo informatiky

**Akademický rok**: 2023/2024

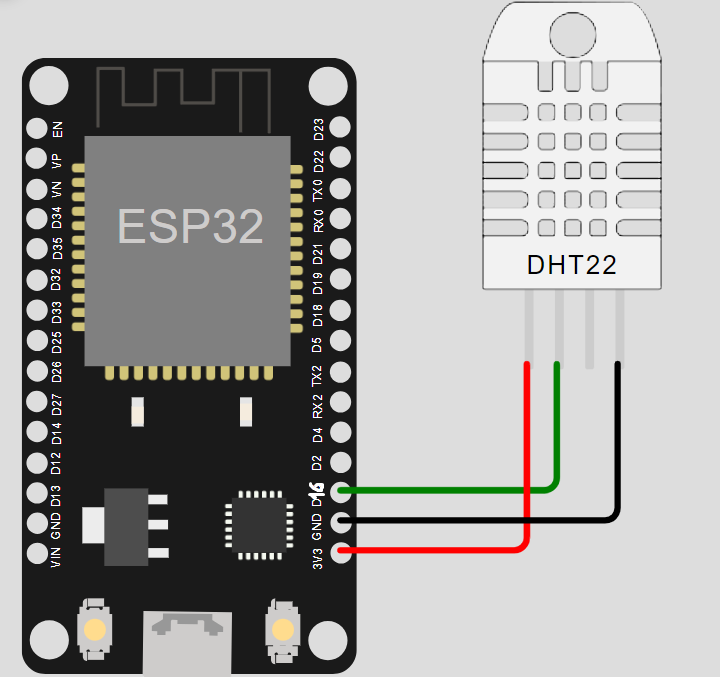
**Ročník**: Druhý

**Predmet**: Internet vecí 2

**Študent**: Sebastián Šatan a Jozef Blaško

**Dátum**: 30. 4. 2024

Tento program meria teplomer a vlhkomer s webovým rozhraním pomocou platformy ESP32. Zahrnuje čítanie teploty a vlhkosti zo senzoru DHT22, ukladanie posledných hodnôt a ich zobrazenie na webovej stránke pomocou asynchrónneho webového servera.



**Použité knižnice**

**WiFi.h:** Knihovna pro prácu s bezdrôtovým pripojením Wi-Fi.

**ESPAsyncWebServer.h:** Knižnica pre asynchrónny webový server na platforme ESP32.

**Adafruit\_Sensor.h:** Knihovna obsahujúca spoločné rozhranie pre senzory Adafruit.

**DHT.h:** Knižnica pre prácu so senzorom teploty a vlhkosti DHT.

**Globálne premenné a konštanty**

**ssid:** Názov Wi-Fi siete.

**password:** Heslo Wi-Fi siete.

**DHTPIN:** Pin pripojenia senzoru DHT k mikrokontroléru.

**DHTTYPE:** Typ senzoru DHT (v tomto prípade DHT22).

**lastTemperature:** Circulárny buffer pre ukladanie posledných 5 teplotných hodnôt.

**lastHumidity:** Circulárny buffer pre ukladanie posledných 5 hodnôt vlhkosti.

**currentIndex:** Index aktuálnej pozície v circulárnom buffri.

**Funkcie**

**readDHTTemperature():** Čítanie teploty zo senzoru DHT. Vracia hodnotu teploty a ukladá ju do circulárneho buffra.

**readDHTHumidity():** Čítanie vlhkosti zo senzoru DHT. Vracia hodnotu vlhkosti a ukladá ju do circulárneho buffra.

**processor(const String& var):** Nahradenie placeholdrov v HTML šablóne aktuálnymi hodnotami teploty, vlhkosti a poslednými záznamami.

**Inicializácia servera**

**WiFi.softAP():** Nastavuje mikrokontrolér ESP32 ako Access Point s definovaným SSID a heslom.

**server.begin():** Spúšťa asynchrónny webový server na definovanom porte.

**server.on():** Definuje spracovanie HTTP GET požiadaviek na rôzne URL cesty. Pri príchode GET požiadavky na každú z týchto ciest sa vykoná príslušná funkcia obsluhujúca požiadavku.

**Senzor teploty a vlhkosti (DHT22)**

**DHTPIN:** Pin, na ktorom je pripojený senzor DHT22.

**DHTTYPE:** Typ senzora DHT22.

**DHT dht(DHTPIN, DHTTYPE):** Inicializácia objektu senzora DHT s definovaným pinom a typom.

**Web server (AsyncWebServer)**

**server(80):** Inicializácia asynchrónneho webového servera na porte 80.

**server.on():** Definuje spracovanie HTTP požiadaviek na jednotlivé URL cesty.

**Funkcie pre čítanie hodnôt teploty a vlhkosti**

**readDHTTemperature():** Číta teplotu zo senzora DHT a ukladá ju do premenných. V prípade chyby vracia textový reťazec "--".

**readDHTHumidity():** Číta vlhkosť zo senzora DHT a ukladá ju do premenných. V prípade chyby vracia textový reťazec "--".

**HTML šablóna (index\_html)**

Obsahuje štruktúru HTML stránky s vloženými štýlmi a JavaScriptom.

Používa placeholdery (%TEMPERATURE%, %HUMIDITY%, %READINGS%), ktoré sú nahrádzané aktuálnymi hodnotami teploty, vlhkosti a predošlými záznamami.

**JavaScript**

**setInterval():** Periodicky odosiela HTTP požiadavky na server pre aktualizáciu údajov o teplote, vlhkosti a predošlých záznamov.

**Zdrojový kód:**

#include "WiFi.h"

#include "ESPAsyncWebServer.h"

#include <Adafruit\_Sensor.h>

#include <DHT.h>

const char\* ssid = "pocasie-info";

const char\* password = "123456789";

#define DHTPIN 16

#define DHTTYPE    DHT22

DHT dht(DHTPIN, DHTTYPE);

AsyncWebServer server(80);

float lastTemperature[5] = {0};

float lastHumidity[5] = {0};

int currentIndex = 0;

String readDHTTemperature() {

  float t = dht.readTemperature();

  if (isnan(t)) {

    Serial.println("Failed to read temperature from DHT sensor!");

    return "--";

  } else {

    Serial.println(t);

    lastTemperature[currentIndex] = t;

    currentIndex = (currentIndex + 1) % 5;

    return String(t);

  }

}

String readDHTHumidity() {

  float h = dht.readHumidity();

  if (isnan(h)) {

    Serial.println("Failed to read humidity from DHT sensor!");

    return "--";

  } else {

    Serial.println(h);

    lastHumidity[currentIndex] = h;

    currentIndex = (currentIndex + 1) % 5;

    return String(h);

  }

}

const char index\_html[] PROGMEM = R"rawliteral(

<!DOCTYPE HTML>

<html>

<head>

  <meta name="viewport" content="width=device-width, initial-scale=1">

  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/milligram/1.4.1/milligram.min.css">

  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.4/css/all.min.css">

  <style>

    html {

      font-family: Arial;

      display: flex;

      justify-content: center;

      align-items: center;

      height: 100%;

      background-color: #e0f7fa;

    }

    body {

      text-align: center;

    }

    h2 {

      font-size: 3.5rem;

      font-weight: bold;

    }

    p {

      font-size: 2.5rem;

      font-weight: bold;

    }

    .units {

      font-size: 1.5rem;

      font-weight: bold;

    }

    .sensor-container {

      display: flex;

      flex-direction: column;

      justify-content: center;

      align-items: center;

      margin-bottom: 20px;

    }

    .sensor-info {

      width: 100%;

    }

    .progress-bar {

      width: 100%;

      background-color: #ddd;

      border-radius: 20px;

      padding: 3px;

    }

    .progress-bar-fill {

      height: 20px;

      text-align: center;

      line-height: 20px;

      color: white;

      width: 0%;

      border-radius: 20px;

    }

    .readings {

      font-size: 2rem;

      text-align: left;

      margin-top: 20px;

    }

    ul {

      list-style-type: none;

      padding: 0;

      margin: 0;

    }

    li {

      background-color: #e0f7fa;

      padding: 10px;

      margin-bottom: 5px;

    }

  </style>

</head>

<body>

  <h2>Teplomer a vlhkomer</h2>

  <!-- Temperature and Humidity section -->

  <div class="sensor-container">

    <div class="sensor-info">

      <p>

        <i class="fas fa-thermometer-half" style="color:#059e8a;"></i>

        <span id="temperature">%TEMPERATURE%</span>

        <sup class="units">&deg;C</sup>

      </p>

      <div class="progress-bar">

        <div class="progress-bar-fill" id="temperature-bar"></div>

      </div>

    </div>

    <div class="sensor-info">

      <p>

        <i class="fas fa-tint" style="color:#00add6;"></i>

        <span id="humidity">%HUMIDITY%</span>

        <sup class="units">&percnt;</sup>

      </p>

    </div>

  </div>

  <!-- Previous Readings section -->

  <div class="readings">

    <h3>Predchadzajuce hodnoty:</h3>

    <ul id="readings">%READINGS%</ul>

  </div>

</body>

<script>

setInterval(function ( ) {

  var xhttp = new XMLHttpRequest();

  xhttp.onreadystatechange = function() {

    if (this.readyState == 4 && this.status == 200) {

      document.getElementById("temperature").innerHTML = this.responseText;

      var temperature = parseFloat(this.responseText);

      var temperatureBar = document.getElementById("temperature-bar");

      if (temperature >= 0 && temperature <= 20) {

        temperatureBar.style.backgroundColor = "#FBBD06";

      } else if (temperature >= 21 && temperature <= 26) {

        temperatureBar.style.backgroundColor = "#FBAD06";

      } else if (temperature >= 27 && temperature <= 35) {

        temperatureBar.style.backgroundColor = "#FB7C06";

      } else {

        temperatureBar.style.backgroundColor = "#FB4B06";

      }

      temperatureBar.style.width = ((temperature / 50) \* 100) + "%";

    }

  };

  xhttp.open("GET", "/temperature", true);

  xhttp.send();

}, 10000 ) ;

setInterval(function ( ) {

  var xhttp = new XMLHttpRequest();

  xhttp.onreadystatechange = function() {

    if (this.readyState == 4 && this.status == 200) {

      document.getElementById("humidity").innerHTML = this.responseText;

    }

  };

  xhttp.open("GET", "/humidity", true);

  xhttp.send();

}, 10000 ) ;

setInterval(function ( ) {

  var xhttp = new XMLHttpRequest();

  xhttp.onreadystatechange = function() {

    if (this.readyState == 4 && this.status == 200) {

      document.getElementById("readings").innerHTML = this.responseText;

    }

  };

  xhttp.open("GET", "/readings", true);

  xhttp.send();

}, 10000 ) ;

</script>

</html>)rawliteral";

String processor(const String& var){

  if(var == "TEMPERATURE"){

    return readDHTTemperature();

  }

  else if(var == "HUMIDITY"){

    return readDHTHumidity();

  }

  else if(var == "READINGS"){

    String readingsList;

    for (int i = 0; i < 5; ++i) {

      int idx = (currentIndex - i - 1 + 5) % 5;

      readingsList += "<li>Teplota: " + String(lastTemperature[idx]) + "°C, Vlhkost: " + String(lastHumidity[idx]) + "%</li>";

    }

    return readingsList;

  }

  return String();

}

void setup(){

  Serial.begin(115200);

  dht.begin();

  IPAddress local\_ip(192, 168, 1, 1);

  IPAddress gateway(192, 168, 1, 1);

  IPAddress subnet(255, 255, 255, 0);

  WiFi.softAP(ssid, password);

  WiFi.softAPConfig(local\_ip, gateway, subnet);

  Serial.print("Connect to My access point: ");

  Serial.println(ssid);

  server.begin();

  Serial.println("HTTP server started");

  Serial.println(WiFi.localIP());

  server.on("/", HTTP\_GET, [](AsyncWebServerRequest \*request){

    request->send\_P(200, "text/html", index\_html);

  });

  server.on("/temperature", HTTP\_GET, [](AsyncWebServerRequest \*request){

    request->send\_P(200, "text/plain", readDHTTemperature().c\_str());

  });

  server.on("/humidity", HTTP\_GET, [](AsyncWebServerRequest \*request){

    request->send\_P(200, "text/plain", readDHTHumidity().c\_str());

  });

  server.on("/readings", HTTP\_GET, [](AsyncWebServerRequest \*request){

    String readingsList;

    for (int i = 0; i < 5; ++i) {

      int idx = (currentIndex - i - 1 + 5) % 5;

      readingsList += "<li>Teplota: " + String(lastTemperature[idx]) + "°C, Vlhkost: " + String(lastHumidity[idx]) + "%</li>";

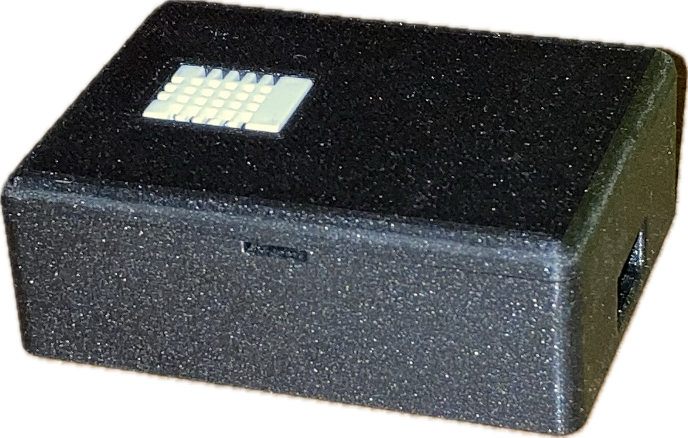
    }

    request->send(200, "text/html", readingsList);

  });

}

void loop(){

}

**Link na stiahnutie 3D modelu pre krabičku pre ESP32 so senzorom DHT22:**

<https://www.printables.com/model/177480-dht22-am2302-case>

