SŠPU
STŘEDNÍ ŠKOLA PRŮMYSLOVÁ
A UMĚLECKÁ, OPAVA

ZÁVĚREČNÁ STUDIJNÍ PRÁCE
dokumentace

Meteorologická stanice

Karolína Říčná

Obor:	18-20-M/01 INFORMAČNÍ TECHNOLOGIE se zaměřením na počítačové sítě a programování
Třída:	IT4
Školní rok:	2022/2023

Poděkování

Děkuji panu učiteli Ing. Petru Grussmannovi za cenné rady v průběhu celé práce a panu učiteli Mgr. Marcelu Godovskému za pomoc se součástkami.

Prohlašuji, že jsem závěrečnou práci vypracovala samostatně a uvedla veškeré použité informační zdroje.

Souhlasím, aby tato studijní práce byla použita k výukovým účelům na Střední průmyslové a umělecké škole v Opavě, Praskova 399/8.

V Opavě 31. 12. 2022

podpis autora práce

Anotace

Projekt se zabývá tvorbou meteorologické stanice se zobrazováním hodnot měřených veličin v systému pro domácí automatizaci – Home Assistant. Projekt se skládá z hardwarové a softwarové části. Základ hardwarové části tvoří vývojová deska WeMos D1 Mini. Zařízení pomocí senzoru BME280 měří teplotu, atmosférický tlak, vlhkost vzduchu a absolutní nadmořskou výšku. Program reaguje na změnu měřených veličin, jakmile ji zaznamená změna se projeví v grafech. Programová část stanice, která obstarává správu měřícího senzoru a odesílání zpráv je řešena v jazyce Arduino, což je kombinace jazyků C a C++. Další konfigurace jsou provedeny v konfiguračním souboru Home Assistantu – configuration.yaml nebo přímo v uživatelském rozhraní Home Assistantu či jeho add-onů.

Klíčová slova

WeMos D1 Mini; BME280; ESP8266EX; MQTT; Wi-Fi; add-on; InfluxDB; Grafana; Home Assistant

Annotation

The project deals with the creation of meteorological station with the display of the values of measured quantities in the system for home automation – Home Assistant. The project consists of a hardware and software part. The base of the hardware part is the We-Mos D1 Mini development board. Using the BME280 sensor, the device measures temperature, atmospheric pressure, air humidity and absolute altitude. The program reacts to a change in the measured values, as soon as it is recorded, the change is reflected in the graphs. The program part, which provides sensor management and sending messages of the measuring, is solved in the Arduino language, which is a combination of the C and C++ languages. Other configurations are made in the Home Assistant configuration file – configuration.yaml or directly in the Home Assistant user interface or its add-ons.

Keywords

WeMos D1 Mini; BME280; ESP8266EX; MQTT; Wi-Fi; add-on; InfluxDB; Grafana; Home Assistant

OBSAH

ÚVOD

Pro svůj závěrečný projekt jsem se rozhodla vytvořit meteorologickou stanici se zobrazováním hodnot v systému pro domácí automatizaci – Home Assistant. Jelikož mě práce se senzory vždy zajímala, tak mi problematika Internetu věcí, tedy propojení hardwarových zařízení s internetem, připadala jako vhodné téma mé závěrečné práce. Na podobném principu jako má meteostanice totiž mohou fungovat různé automatizace chytrých domácností, kde je možné třeba z mobilního telefonu zapnout světla, počítač, pračku apod.

Hlavním cílem projektu bylo pochopit a prostudovat technologii MQTT, pomocí ní propojit zařízení s Home Assistantem a v grafech vykreslit hodnoty měřených veličin měnící se v čase. Hlavní hardwarovou součástí je vývojová deska WeMos D1 Mini s wifi modulem ESP8266EX a k němu připojené další komponenty, které budou blíže popsány v dalších částech dokumentace. Stanice je naprogramována v jazyce Arduino, který je kombinací jazyků C a C++. Komunikace mezi stanicí a MQTT brokerem probíhá přes Wi-Fi.

V této dokumentaci podrobně popisuji výrobu stanice a princip jejího fungování, pokračuji popisem technologií nezbytných k její výrobě i k její současné funkčnosti a rozebírám, jak funguje nejen samotná stanice, tak i MQTT protokol a samotný Home Assistant, popisuji, jak funguje jejich vzájemná komunikace a jednotlivé úkony obou částí systému. V závěru se zabývám podobou stanice a hodnotím odvedenou práci.

1 VÝROBA METEOROLOGICKÉ STANICE

První částí mého projektu představovalo sestavení samotného zařízení. Jelikož se jednalo o mou první zkušenost s projektem této velikosti, výběru a nákupu součástek předcházel zdlouhavý výzkum dané problematiky. Po diskusi s učiteli jsem se rozhodla jako základ mého zařízení použít vývojovou desku WeMos D1 Mini s Wi-Fi modulem ESP8266EX.

Vzhledem k tomu, že jsem s většinou nově nakoupených součástek dříve nepracovala, rozhodla jsem se postupovat po malých krocích. Zapojovala jsem jednu součástku po druhé do nepájivého pole a zkoušela jejich funkčnost pomocí jednoduchých příkladů. Po vyzkoušení všech součástek jsem začala dávat dohromady samotnou sestavu.

Po zkompletování celkové sestavy jsem přešla k programování samotné stanice. Pro tento účel jsem zvolila jazyk pro Arduino což je kombinace jazyků C a C++. Když byla první část projektu plně funkční, vypisování měřených hodnot na LCD displej, čekala mě ta pravá výzva – připojit ESP k Wi-Fi a pomocí MQTT protokolu začít zobrazovat hodnoty v Home Assistantu, což byl hlavní cíl mého projektu.

2 VYUŽITÉ TECHNOLOGIE

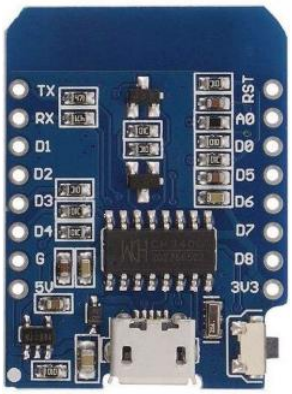
2.1 Hardware

2.1.1 Seznam součástek

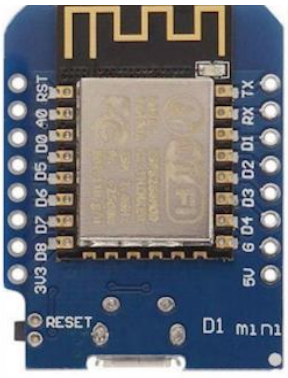
- Vývojová deska WeMos D1 Mini
- Senzor tlaku, teploty a vlhkosti BME 280
- LCD displej 1602
- LCD I2C sériové rozhraní
- Nepájivé kontaktní pole s 830 kontakty

2.1.2 WeMos D1 Mini

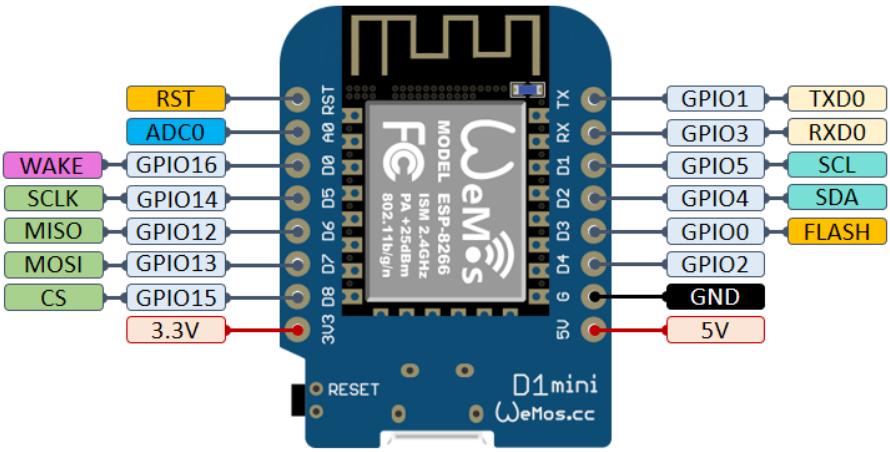
Základ projektu tvoří vývojová deska WeMos D1 Mini (obrázek č. 1). Firmware obstarává Wi-Fi modul ESP8266EX (obrázek č. 2) od společnosti Espressif Systems, který používá bezdrátový komunikační protokol 802.11 b/g/n. Deska má 11 digitálních vstupně/výstupních pinů z nichž všechny kromě D0 jsou schopny využívat pulzně šířkové modulace (PWM), přerušení, I2C a 1 – Wire rozhraní. Dále má 1 vstupní analogový pin, který může přijmout napětí až 3,3 V. Více o pinech viz obrázek č. 3. Pro program je k dispozici Flash paměť o velikost 4 MB.



Obrázek č. 1 WeMos D1 Mini



Obrázek č. 2 ESP8266EX



Obrázek č. 3 Piny WeMos D1 Mini

2.1.3 Senzor tlaku, teploty a vlhkosti BME280

Jako hlavní senzor mého zařízení jsem použila BME280 (obrázek č. 4), který se stará o měření všech veličin zobrazovaných v Home Assistantu. Senzor BME280 je modernějším nástupcem senzorů BMP180, BMP183 atd. Senzor obsahuje vestavěný napěťový regulátor LM6206, který umožňuje používat senzor jak s napětím 3,3 V tak s napětím 5 V. Dále obsahuje rozhraní I2C. Senzor při chodu spotřebovává méně než 1 mA a při nečinnosti pouze 5 µA, díky tak nízké spotřebě je vhodný pro použití i v bateriově napájených zařízeních.

Obrázek č. 4 BME280

Měření teploty probíhá v rozsahu od -40 °C do 85 °C. V teplotním rozsahu 0 až 65 °C je přesnost ±1,0 °C; mimo tento rozsah přesnost klesá na ±1,5 °C. Měření relativní vlhkosti probíhá v rozsahu od 0 do 100 % s přesností ±3 %. Maximální měřitelná vlhkost však klesá při extrémně vysokých nebo nízkých teplotách. Podle datasheetu senzor může měřit až 100 % vlhkosti v teplotním rozsahu 0 až 60 °C. Měření atmosférického tlaku probíhá v rozsahu od 300 Pa do 1100 hPa s přesností ±1 hPa. V teplotním rozsahu 0 až 65 °C

je dosaženo plné přesnosti. Senzor dokáže měřit atmosférický talk s takovou přesností, že může zároveň sloužit jako výškoměr (altimetr) s přesností ±1 m.

Senzor nepracuje absolutní nadmořskou výškou, tedy výšku nad hladinou moře (MSL), ale výškou nad úrovní terénu (AGL). Senzor sám AGL přímo neměří, ale odhaduje ji pomocí měření tlaku. Pokud chceme měřit absolutní nadmořskou výšku, senzor potřebuje znát aktuální atmosférický tlak u hladiny moře. Použila jsem symbolickou proměnnou SEALEVELPRESSURE_HPA nastavenou na hodnotu 1013.25 hPa, což je průměrný atmosférický tlak u hladiny moře.

```
#define SEALEVELPRESSURE_HPA (1013.25)

Adafruit_BME280 bme;

void setup() {
    Serial.begin(9600);

    if (!bme.begin(0x76)) {
        Serial.println("Could not find a valid BME280 sensor, check
wiring!");
        while (1);
    }
}


void loop() {
    Serial.print("Temperature = ");
    Serial.print(bme.readTemperature());
    Serial.println("*C");
    delay(1000);
}
```

2.1.4 LCD displej 1602

Jako výstupní zařízení jsem použila LCD displej 1602, který dokáže zobrazit až 32 znaků celkem na dvou řádcích (16 x 2). Pro jednodušší provedení jsem pro připojení zvolila model s I2C sériovým rozhraním, které umožňuje displej připojit pouze dvěma vodiči pro přenos dat. Funkčnost displeje obstarává knihovna LiquidCrystal_I2C.

2.2 Napájení

Napájení vývojové desky zatím řeším z počítače pomocí Micro USB kabelu, který modulu dodává 5 V, toto se zatím hodí pouze pro vnitřní použití stanice, pro venkovní použití mám do budoucna v plánu zařídit externí zdroj. Maximální výstupní proud je 500 mA, což je dostačující pro napájení všech součástek.

Všechny součástky jsem k vývojové desce připojila pomocí propojovacích kabelů na nepájivém kontaktním poli s 830 kontakty.

2.3 Software

2.3.1 Jazyk Arduino

Program sloužící k ovládání hardwarových součástek jsem napsala v jazyce Arduino, který je, až na drobné úpravy, velmi podobný jazykům C a C++. Jazyk Arduino byl přímo vytvořen k programování integrovaných obvodů. Jako vývojové prostředí jsem pro svůj projekt zvolila Visual Studio Code s rozšířením PlatformIO. Jazyk do něj připojíme pomocí modifikace souboru platformio.ini a modifikace zdrojového souboru, v mém případě main.cpp.

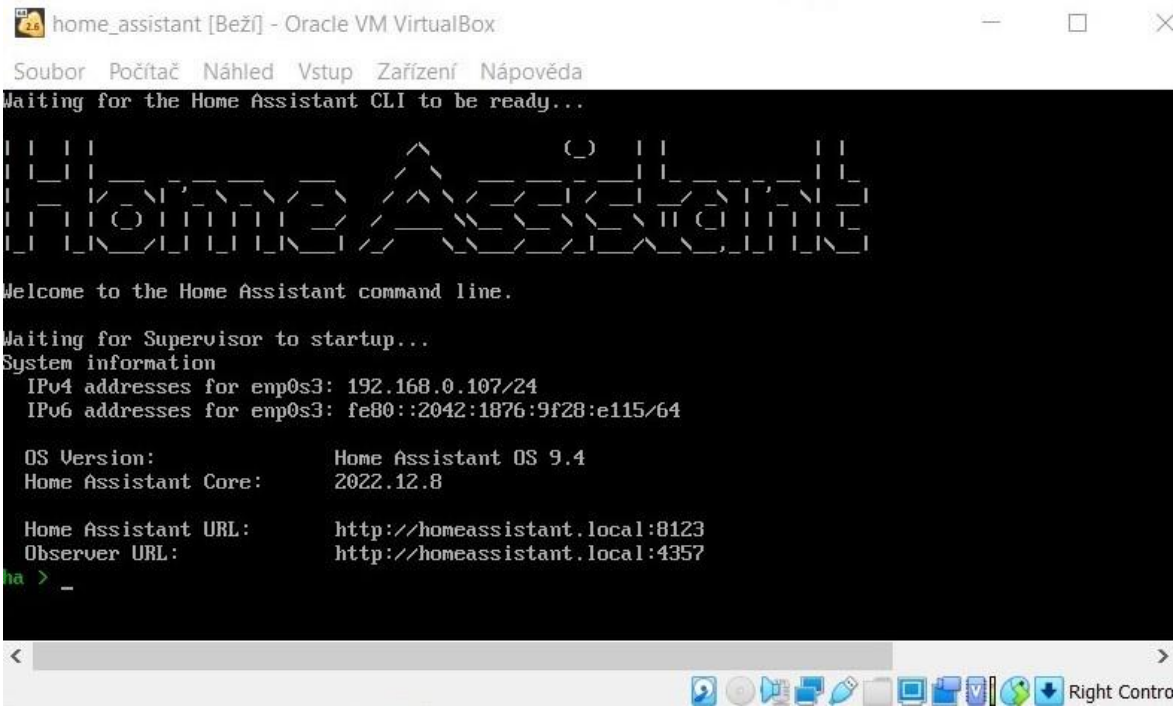
```
[env:d1_mini_lite]
platform = espressif8266
board = d1_mini_lite
framework = arduino
```

```
#include <Arduino.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <Adafruit_Sensor.h>
```

2.3.2 VirtualBox

Nezbytnou součástí mého projektu je Open source virtualizační nástroj VirtualBox, který umožňuje vytvářet virtuální stroje, které dokážou simulovat skutečné počítače a ty umožňují instalaci OS a dalších programů. V mém projektu jsem využila VirtualBox k vytvoření virtuálního stroje s doporučenými parametry a nastaveními, na který jsem nahrála můj

Home Assistant Operating System (HassOS), o něm se budu blíže zmiňovat v další části dokumentace. Po nahrání HassOS se nám na virtuálním stroji zobrazí textové uživatelské rozhraní (obrázek č. 5), kde se vypíšou systémové informace včetně IPv4 adresy a portu, kde se můžeme k Home Assistantu připojit.



Obrázek č. 5 Textové uživatelské rozhraní HassOS s výpisem systémových informací

2.3.3 Home Assistant

Home Assistant je Open Source software pro domácí automatizaci navržený jako centrální řídicí systém pro inteligentní domácí zařízení. Home Assistant má několik různých typů instalací, já jsem zvolila instalaci Home Assistant Operating System (HassOS), protože umožňuje správu add-onů (doplňků), které jsou klíčovou částí mého dalšího postupu. Po instalaci HassOS a připojení k IPv4 adrese a portu se dostaneme do grafického uživatelského rozhraní Home Assistanta, kde jsem si vytvořila uživatelský účet a provedla základní konfigurace. V tomto stavu jsem mohla přejít k instalaci add-onů.

2.3.4 Studio Code Server

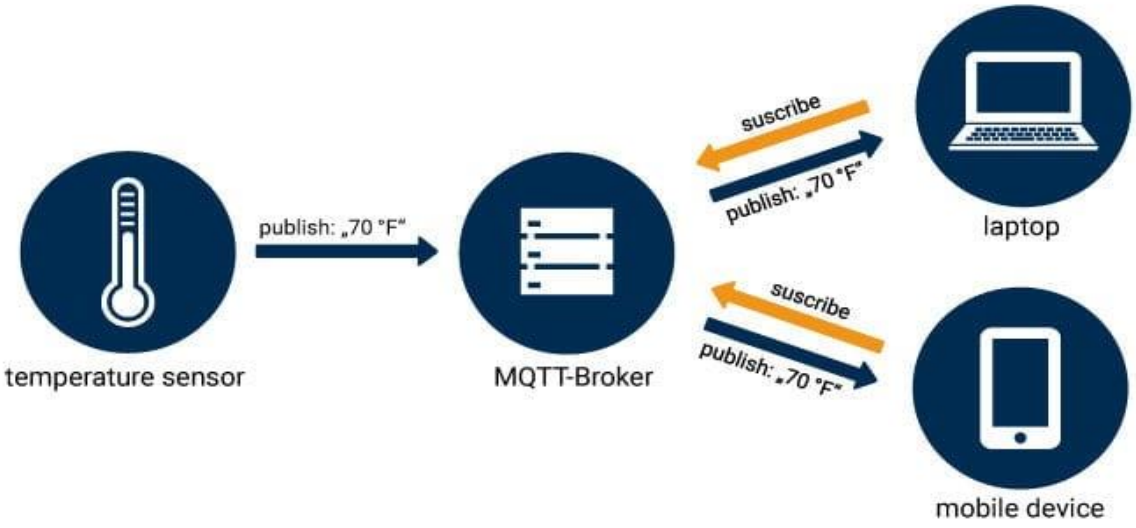
Tento Home Assistant add-on umožňuje v grafickém uživatelském rozhraní Home Assistantu používat Visual Studio Code pro úpravu zdrojových souborů. Já jsem tento add-on používala k úpravě souboru configuration.yaml, tento soubor obsahuje integrace spolu s jejich konfiguracemi.

2.3.5 ESP Home

Okrajově bych chtěla zmínit add-on ESP Home, který umožňuje jednoduchou správu vašich ESP8266/ESP32 zařízení skrze Home Assistant pouze s minimálními nároky na programování. Tento add-on jsem využila pouze pro inspiraci, jak by mohl můj výsledný projekt vypadat.

2.3.6 Mosquitto broker

Hlavním add-onem, který jsem ve svém projektu použila je Mosquitto broker. Tento add-on umožňuje instalaci Open Source brokeru využívajícího MQTT protokol. Tento protokol funguje na principu publish-subscribe, kde broker funguje jako prostředník pro předávání zpráv mezi klienty. Zprávy jsou tříděny do témat (topic) a zařízení v daném tématu buď publikuje (publish) nebo odebírá (subscribe), grafické zobrazení na obrázku č. 6.



Obrázek č. 6 Grafické zobrazení fungování MQTT brokeru

2.3.7 InfluxDB, Grafana

Add-on InfluxDB je Open Source databáze optimalizovaná pro velký objem zápisu. Z toho důvodu je ideální pro záznam dat ze senzorů. Díky HTTP API pro komunikaci s klientem jsem ji využila v kombinaci s dalším add-onem, a to vizualizačním nástrojem Grafana.

Tento add-on mi umožňuje jednoduše zobrazit naměřené hodnoty v přehledných a lehce editovatelných grafech a dashboardech.

3 ZPŮSOBY ŘEŠENÍ A POUŽITÉ POSTUPY

3.1 Hardwarové zařízení

3.1.1 Popis fungování stanice

Po spuštění virtuálního stroje s HassOS a připojení vývojové desky k počítači v souboru configuration.hpp pozměníme klíčové údaje, které umožní připojení k vaší Wi-Fi síti a MQTT brokeru.

```
#define wifi_ssid "Your-wifi-ssid"
#define wifi_password "Your-wifi-password"

#define mqtt_server "Mqtt-ip"
#define mqtt_user "Mqtt-user-name"
#define mqtt_password "Mqtt-user-password"
```

Po zkompilování a nahrání kódu se nejprve spustí funkce setup(), která inicializuje sériovou linku, připojí se k Wi-Fi prostřednictvím funkce setup_wifi, připojí se k MQTT bokreru, pomocí knihovny Wire zahájí komunikaci po I2C sběrnici, zkontroluje přítomnost měřícího senzoru (BME280), inicializuje LCD displej, a nakonec zavolá funkci loop().

Funkce loop() vyvolá funkci vypisHodnotLCD(), pro výpis hodnot na LCD displej, zkontroluje připojení pomocí knihovny PubSubClient. Pokud je čas od spuštění nebo poslední kontroly větší než 1000 milisekund (1 sekunda) aktualizuje měřené veličiny. Pokud se hodnota veličiny liší oproti poslední kontrole pošle zprávu s aktualizovanou hodnotou k příslušnému tématu (topicu). Každých 5 minut probíhá vynucené publikování (forced publish).

3.1.2 Získávání hodnot měření

Funkčnost hlavního senzoru obstarávají knihovny Wire, Adafruit Unified Sensor a Adafruit BME280 Library. Pomocí třídy Adafruit_BME280 jsem vytvořila objekt bme, díky kterému získáme přístup k jednotlivým funkcím knihovny Adafruit BME280 Library.

Získávání teploty probíhá funkcí readTemperature(), získávání vlhkosti funkcí readHumidity(), získávání tlaku funkcí readPressure() – pro získání výsledku v hPa vydělíme získa-

nou hodnotu 100, získávání absolutní nadmořské výšky probíhá funkcí readAltitude(), do které pošleme hodnotu symbolické proměnné SEALEVELPRESSURE_HPA.

```
  temperature = bme.readTemperature();
  pressure = bme.readPressure() / 100.0F;
  altitude = bme.readAltitude(SEALEVELPRESSURE_HPA);
  humidity = bme.readHumidity();
```

3.1.3 Odesílání hodnot měření

Odesílání měřených hodnot probíhá ve funkci publish(), která nejprve vyvolá funkci forcedPublish(), která se stará o vynucené publikování po nastaveném čase, v mém případě po 300000 milisekundách (5 minutách).

```
void forcedPublish(int publishAfterMs){
 long now = millis();
 // Když je momentální čas – čas poslední zprávy větší než 1000.
  if (now - lastMsg > 1000)
  {
    // Čas poslední zprávy = momentální čas.
    lastMsg = now;
  // Když je momentální čas – čas poslední vynucené zprávy větší než námi
nastavený čas pro vynucené publikování.
  if (now - lastForceMsg > publishAfterMs) {
      lastForceMsg = now;
      forceMsg = true;
      Serial.println("Forcing publish every 5 minutes...");
    }
  }
}
```

Pak získá nové hodnoty měřených veličin a uloží je do proměnných „new“. Následně u každé veličiny zkontroluje pomocí funkce checkDiff(), zdali se liší nová hodnota oproti staré nebo zdali je hodnota proměnné forceMsg true. Pokud podmínka projde nastaví se hodnota originální proměnné na hodnotu nové proměnné, informace o změně se vypíše na serial monitor a zpráva se publikuje k příslušnému tématu.

```
void publish(){

    // Vyvolání funkce pro vynucené publikování.
    forcedPublish(300000);

    // Získání nových hodnot do proměnných.
    newTemp = bme.readTemperature();
    newHum = bme.readHumidity();
    newPres = bme.readPressure() / 100.0F;
    newAlt = bme.readAltitude(SEALEVELPRESSURE_HPA);

    // Kontrola rozdílu nové a minulé hodnoty nebo vynucené publikování.
    if (checkDiff(newTemp, temp, diff) || forceMsg)
    {
      // Minulá hodnota = nová hodnota.
      temp = newTemp;
      // Výpis a publikování.
      Serial.print("New temperature:");
      Serial.println(String(temp).c_str());
      client.publish(temperature_topic, String(temp).c_str(), true);
    }
          forceMsg = false;
  }
```

3.1.4 Připojení k Wi-Fi

V momentální situaci řeším připojení k Wi-Fi pomocí knihovny ESP8266WiFi, to mám ovšem v plánu do budoucna vyměnit za knihovnu WiFiManager, díky které bude možné připojit ESP k Wi-Fi z telefonu a ukládat konfigurace různých Wi-Fi sítí.

```
void setup_wifi()
{
  // Výpis ssid sítě, ke které se připojujeme.
  delay(10);
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(wifi_ssid);

  // Začátek komunikace.
  WiFi.begin(wifi_ssid, wifi_password);
```

```
  // Pomyslné načítání, dokud nejsme připojeni.
  while (WiFi.status() != WL_CONNECTED)
  {
    delay(500);
    Serial.print(".");
  }

  // Výpis IP adresy nově připojeného zařízení.
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}
```

3.2 Home Assistant konfigurace

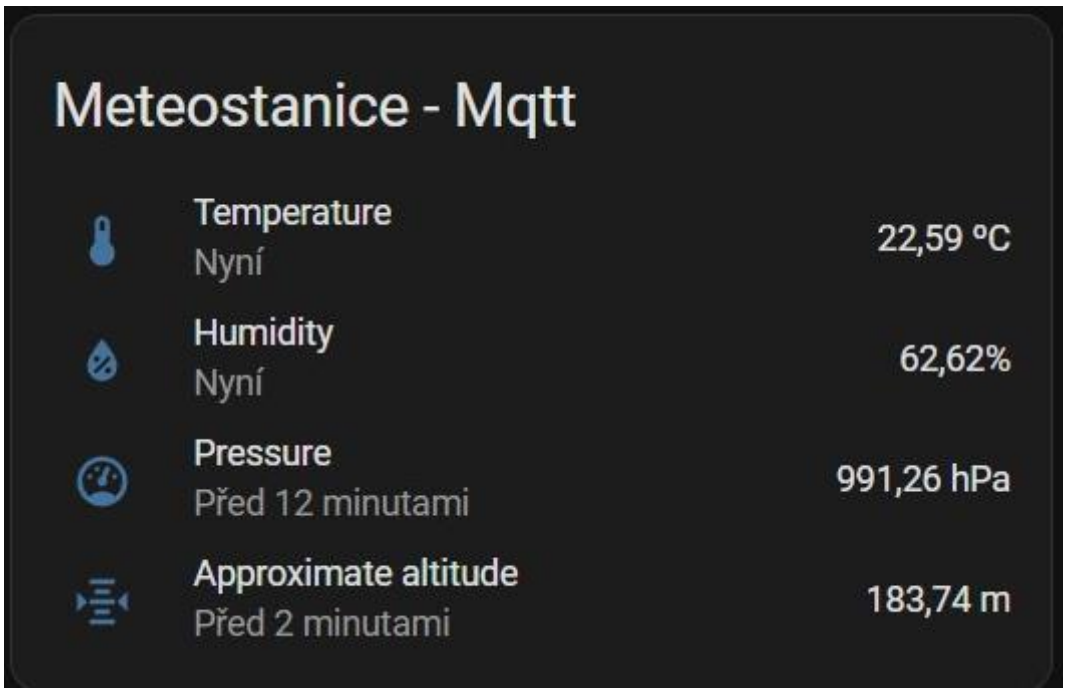
3.2.1 Konfigurace MQTT

V Home Assistantu jsem do konfiguračního souboru configuration.yaml přidala inicializaci mého senzoru a veškerých entit, které budu chtít zobrazovat.

```
mqtt:
  sensor:
    - name: "Temperature"
      state_topic: "sensor/temperature"
      qos: 0
      unit_of_measurement: "ºC"

    - name: "Humidity"
      state_topic: "sensor/humidity"
      qos: 0
      unit_of_measurement: "%"

    - name: "Pressure"
      state_topic: "sensor/pressure"
      qos: 0
      unit_of_measurement: "hPa"

    - name: "Approximate altitude"
      state_topic: "sensor/altitude"
      qos: 0
      unit_of_measurement: "m"
```

3.2.2 Home Assistant Overview dashboard

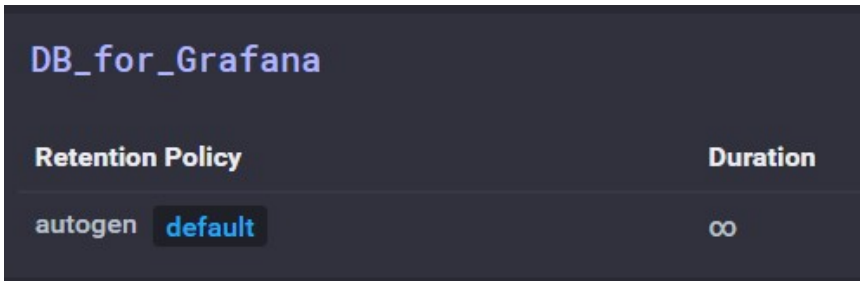
V rámci testování, zda vše funguje, jak má, jsem si v Overview Home Assistanta vytvořila jednoduchý dashboard s entitami které, jsem nakonfigurovala v configuration.yaml. Výsledkem jednoduchého klikání v uživatelském rozhraní je dashboard na obrázku č. 7.



Obrázek č. 7 Dashboard vytvořený v Home Assistant Overview

3.2.3 Konfigurace InfluxDB

Nejprve bylo třeba vytvořit novou InfluxDB databázi určenou pro Grafenu (obrázek č. 8), poté vytvořit uživatele této databáze (obrázek č. 9) a nakonec přidat konfiguraci databáze do souboru configuration.yaml.



Obrázek č. 8 Nově vytvořená databáze

User	Password	Permissions
chronograf	Change	ALL
kapacitor	Change	ALL
grafana-user	Change	ALL

Obrázek č. 9 Nově vytvořený uživatel

```
influxdb:
  host: 192.168.0.1
  port: 8086
  database: DB_for_Grafana
  username: grafana-user
  password: Grafana-user-password
  max_retries: 3
  default_measurement: state
```

3.2.4 Konfigurace Grafana

V Grafickém uživatelském rozhraní Grafany nejprve nakonfigurujeme nový data source, a to InfluxDB. V konfiguraci nastavíme správně jméno databáze, jméno uživatele pro tuto databázi a správnou URL. Poté můžeme v jednoduchém grafickém rozhraní vytvářet dashboardy a k nim i komplexnější grafy (obrázek č. 10).

Obrázek č. 10 Grafy vytvořené za pomoci Grafany

4 VÝSLEDKY ŘEŠENÍ

4.1 Podoba Hardwarového zařízení

Meteorologická stanice je v současné době plně funkční, a i po náročném testování splňuje mé počáteční očekávání. Z důvodu častých úprav je zařízení umístěno na nepájivém poli. Do budoucna by bylo vhodné vytvořit pevný plošný spoj na poli pájivém, zejména kvůli křehkosti současné sestavy. Na místě je také přidání ochranné krabičky a dalších senzorů, například senzory typu MQ pro detekci plynů v ovzduší nebo čidlo směru větru či anemometr pro detekci rychlosti větru. Jak jsem již dříve zmiňovala napájení pomocí kabelu není vhodné při venkovním použití, proto by bylo vhodné napájet zařízení baterií, díky tomu bychom v budoucnu přestali být závislí na délce kabelu či dostupnosti napájení z elektrické sítě.

4.2 Bezdrátová komunikace

Bezdrátová komunikace mezi hardwarovým zařízením a MQTT brokerem funguje spolehlivě. Jak jsem již dříve zmiňovala hlediska bezpečnosti a jednoduchosti bych do budoucna chtěla využít pro připojování k Wi-Fi knihovnu WiFiManager, díky které bude možné připojit ESP k Wi-Fi z telefonu a ukládat konfigurace různých Wi-Fi sítí.

4.3 Vykreslování měřených hodnot

Vykreslování hodnot v grafech plní svůj účel, po přihlášení do Home Assistanta tak přehledně vidíme změny měřených veličin v čase a sami můžeme grafy editovat dle našich potřeb. Do budoucna bych ráda přidala více statistik například nejvyšší či nejnižší naměřená hodnota a podobně.

Obrázek č. 11 Grafy vytvořené pomocí Grafany

ZÁVĚR

Cílem projektu bylo vytvořit funkční sestavu, pochopit a prostudovat technologii MQTT, propojit zařízení s Home Assistantem a v grafech vykreslit hodnoty měřených veličin měnící se v čase. Vyčtené cíle byly naplněny, stanice i propojení s Home Assistantem je plně funkční. Při vývoji mě nicméně napadl nespočet změn, které by mé zařízení vylepšily, několik z nich už jsem vyčetla dříve jako například vytvoření pevného plošného spoje na pájivém poli, napájení pomocí baterie, přidání ochranné krabičky nebo připojování se k Wi-Fi pomocí knihovny WifiManager. Kromě dříve zmíněných bych také ráda vyměnila spouštění Home Assistantu z VirtualBoxu za Raspberry Pi, toto je ovšem velmi nákladné a jedná se o vylepšení do opravdu daleké budoucnosti.

SEZNAM POUŽITÝCH INFORMAČNÍCH ZDROJŮ

[1] Which Home Assistant install is right for you?. *YouTube* [online]. 2022 [cit. 2022-12-30]. Dostupné z: https://www.youtube.com/watch?v=i72K1wyuTfg&t=433s

[2] MQTT and Home Assistant. *YouTube* [online]. 2022 [cit. 2022-12-30]. Dostupné z: https://www.youtube.com/watch?v=cZV2OOXLtEI&t=547s

[3] Home Assistant. *Home Assistant* [online]. 2022 [cit. 2022-12-30]. Dostupné z: https://www.home-assistant.io/

[4] Install Home Assistant Operating System. *Home Assistant* [online]. [cit. 2022-12-30]. Dostupné z: https://www.home-assistant.io/installation/windows

[5] MQTT. *Home Assistant* [online]. [cit. 2022-12-30]. Dostupné z: https://www.home-assistant.io/integrations/mqtt/

[6] Report the temperature with ESP8266 to MQTT. *Home Assistant* [online]. [cit. 2022-12-30]. Dostupné z: https://www.home-assistant.io/blog/2015/10/11/measure-temperature-with-esp8266-and-report-to-mqtt/

[7] Hello-Future CZ, playlist Home assistant. *YouTube* [online]. 2019 [cit. 2022-12-30]. Dostupné z: https://www.youtube.com/watch?v=XUv89aVMZNE&list=PLmEu9kI4avpRZ4o1WXfSlVWO4LYC2PyLs

[8] Grafana Explained in Under 5 Minutes. *YouTube* [online]. 2020 [cit. 2022-12-30]. Dostupné z: https://www.youtube.com/watch?v=lILY8eSspEo

[9] Statistics in Home Assistant with InfluxDB and Grafana. *YouTube* [online]. 2022 [cit. 2022-12-30]. Dostupné z: https://www.youtube.com/watch?v=KM6UC4tMVYo&list=LL&index=2

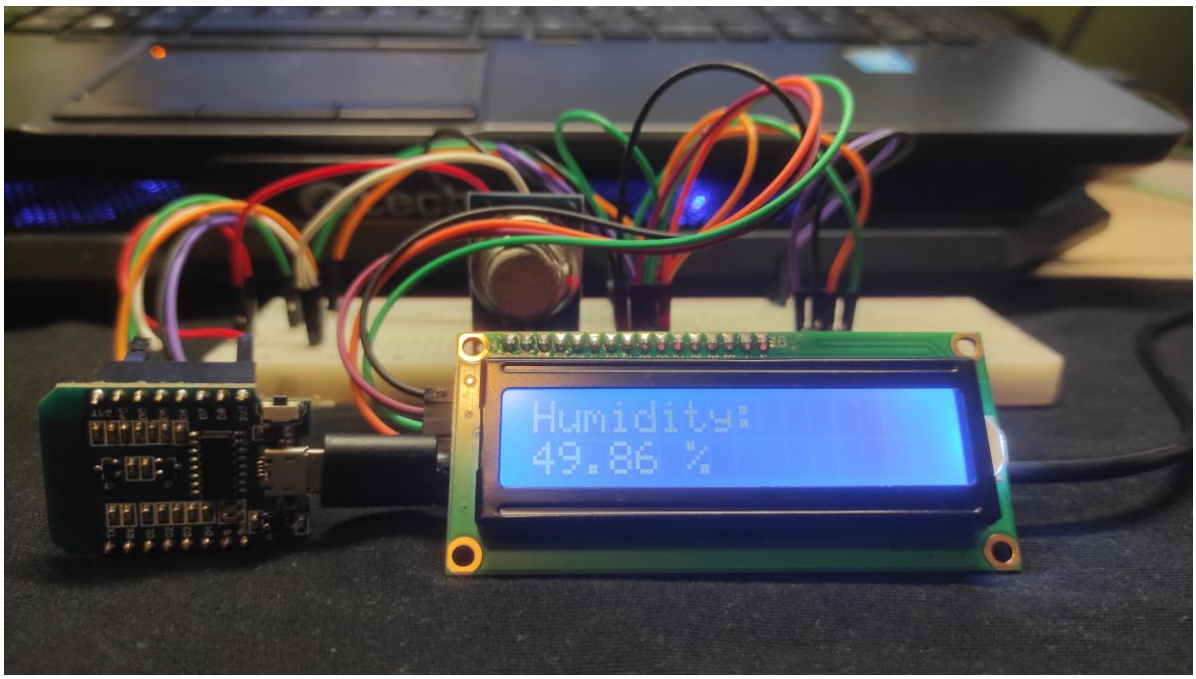
[10] Android #27 Wifimanager + OTA for ESP32/ESP8266 (ESPAsyncWifimanager / ESPAsyncElegantOTA). *YouTube* [online]. 2021 [cit. 2022-12-30]. Dostupné z: https://www.youtube.com/watch?v=UlRLTvl4DRc&list=LL&index=5

[11] Interface BME280 Temperature, Humidity & Pressure Sensor with Arduino. *Last Minute ENGINEERS* [online]. c2022 [cit. 2022-12-30]. Dostupné z: https://lastminuteengineers.com/bme280-arduino-tutorial/

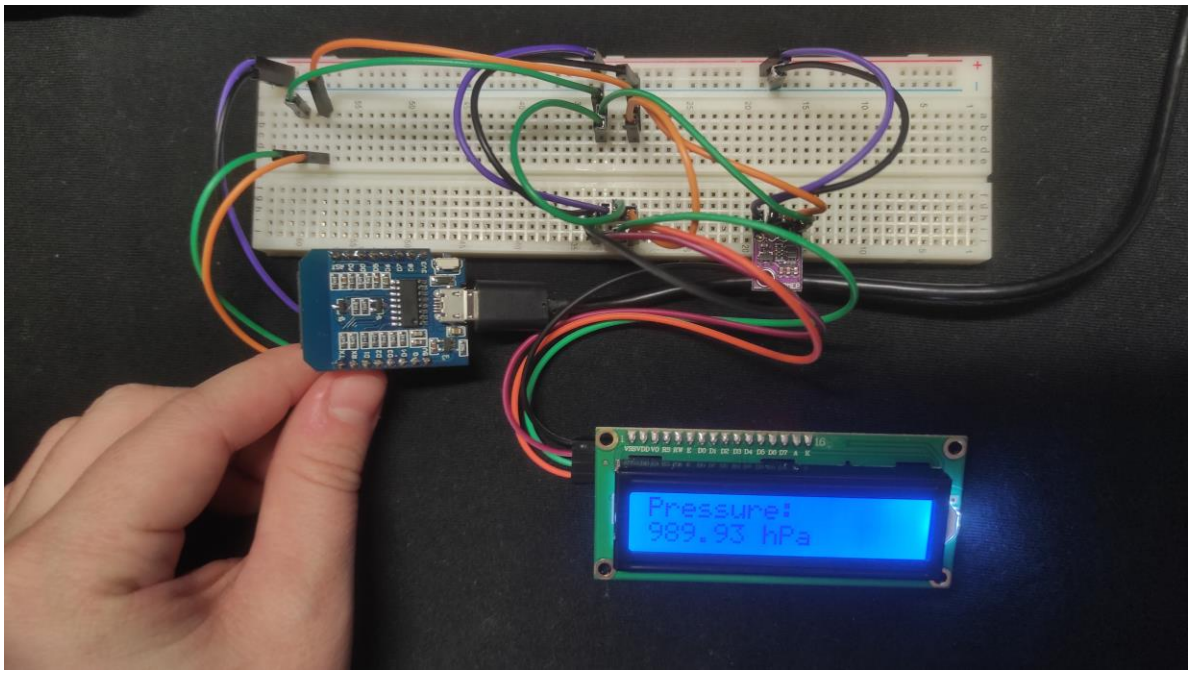
[12] ESP8266 NodeMCU MQTT – Publish BME280 Sensor Readings (Arduino IDE). *Random Nerd Tutorials* [online]. c2013-2022 [cit. 2022-12-30]. Dostupné z: https://randomnerdtutorials.com/esp8266-nodemcu-mqtt-publish-bme280-arduino/

SEZNAM PŘÍLOH

Příloha č. 1: Fotodokumentace



Přední detail na sestavu v průběhu práce



Pohled shora na konečné zapojení