

**ZÁVĚREČNÁ STUDIJNÍ PRÁCE**

**dokumentace**

**Thermistor Thermometer**

Jakub Komenda

**Obor:**

18-20-M/01 INFORMAČNÍ TECHNOLOGIE

se zaměřením na počítačové sítě a programování

**Třída:**

IT4

**Školní rok:**

2020/2021

*Poděkování*

*Rád bych poděkoval panu učiteli Ing. Petru Grussmannovi za cenné rady, za pomoc se součástkami a řešení některých problémů. Také bych rád poděkoval panu učiteli Mgr. Marceli Godovskému, který mi také pomohl a hodně věcí vysvětlil.*

Prohlašuji, že jsem závěrečnou práci vypracoval samostatně a uvedl 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. 2020 |



*podpis autora práce*

**Anotace**

Cílem mého závěrečného projektu bylo vytvořit funkční teploměr, který pomocí termistoru změří napětí a následně ho převede na stupně Celsia. Tuto hodnotu posílám skrz MQTT broker na vlastní Home Assistant v podobě grafu. Projekt tvoří zapojený hardware na nepájivém poli a Home Assistant nainstalovaný skrz Docker na ubuntu serveru. Komunikace probíhá pomocí Wi-Fi. Teplota se začne posílat do HA po zapsání příkazu do Visual Studio Code a vytvářet graf.

**Klíčová slova**

Teploměr; Termistor; Wi-Fi; Home Assistant; MQTT; ESP8266; Docker

**OBSAH**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ÚVOD ........................................................................................................................................................... | | | | [5](#page6) |
| 1[VÝROBA A VÝVOJ TEPLOMĚRU ..................................................................................](#page6) | | | | 6 |
| [2](#page7) | [PRINCIP FUNGOVÁNÍ TEPLOMĚRU .....................................................................](#page7) | | | [7](#page7) |
| [3](#page8) | [VYUŽITÉ TECHNOLOGIE .......................................................................................](#page8) | | | [8](#page8) |
|  | [3.1](#page8) | [HARDWARE .......................................................................................................](#page8) | | [8](#page8) |
|  | [3.1.1](#page8) | | [Seznam součástek ............................................................................................](#page8) | [8](#page8) |
|  | [3.1.2](#page8) | | [ESP-8266 D1 Mini .........................................................................................](#page8) | [8](#page8) |
|  | [3.1.3](#page9) | | [Termistor NTC Vishay 10k …….....................................................................](#page9) | [9](#page9) |
|  | [3.1.4](#page9) | | [Rezistor 10k ...................................................................................................](#page9) | [9](#page9) |
|  | [3.2](#page10) | [SOFTWARE .......................................................................................................](#page10) | | [10](#page10) |
|  | [3.2.1](#page10) | | [Oracle VM VirtualBox 6.1.16 ......................................................................](#page10) | [10](#page10) |
|  | [3.2.2](#page10) | | [Docker Engine 20.10.1 .................................................................................](#page10) | [10](#page10) |
|  | [3.2.3](#page10) | | [Eclipse Mosquitto 2.0.4 ................................................................................](#page10) | [10](#page10) |
|  | 3.2.4 | | Home Assistant ........................................................................................... | 10 |
|  | 3.2.5 | | Visual Studio Code ..................................................................................... | 11 |
| [4](#page11) | ZPŮSOBY ŘEŠENÍ A[VYUŽITÉ TECHNOLOGIE .................................................](#page11) | | | [12](#page11) |
|  | [4.1](#page11) | [HARDWAROVÉ ZAŘÍZENÍ ...........................................................................](#page11) | | [1](#page11)2 |
|  | [4.1.1](#page11) | | [Základní funkce jazyka Arduino ..................................................................](#page11) | [12](#page11) |
|  | [4.1.2](#page11) | | [Připojení k WiFi ……...................................................................................](#page11) | [12](#page11) |
|  | [4.1.3](#page11) | | [Připojení k MQTT brokeru ...........................................................................](#page11) | 13 |
|  | [4.1.4](#page12) | | [Měření teploty ….........................................................................................](#page12) | [13](#page12) |
|  | [4.1.5](#page12) | | [Posílání hodnot na Home Assistant ..............................................................](#page12) | [14](#page12) |
|  | [4.2](#page13) | [WEBOVÁ APLIKACE ......................................................................................](#page13) | | [15](#page13) |
| [5](#page15) | [VÝSLEDKY ŘEŠENÍ ...............................................................................................](#page15) | | | [16](#page15) |
| ZÁVĚR............................................................................................................................... | | | | [17](#page17) |
| SEZNAM POUŽITÝCH INFORMAČNÍCH ZDROJŮ .................................................... | | | | 18 |

**ÚVOD**

Cílem tohoto projektu bylo vytvořit teploměr, který bude posílat teplotu na webový server a já to budu moct sledovat na různých zařízeních. Jako první nápad mě napadlo postavit meteostanici, která bude měřit více hodnot najednou. Přišlo mi to jako fajn projekt, ale po čase stráveném přemýšlení nad kódem jsem zjistil, že to je nad moje síly. Poté mě napadlo, že bych vytvořil teploměr pomoci DHT11 nebo BME280, ale přišlo mi to obyčejné. Tak jsem se rozhodl, že teplotu budu měřit termistorem.

Jako první myšlenka byla posílat teplotu na nějakou mobilní aplikaci, skrze kterou bych kontroloval teplotu mého pokoje. Poté mě však napadlo posílat teplotu na svůj vlastní Home Assistant, kde bych do budoucna mohl vytvořit další přístroje na automatizování domácnosti. Celý systém Home Assistantu funguje přes Docker na ubuntu serveru. Celý teploměr je naprogramován v jazyce Arduino, což je kombinace jazyků C a C++.

Ve své dokumentaci se zmiňuji o použitých technologiích, principu fungování teploměru a posílání hodnot přes MQTT. V další části popisuji technologie potřebné k výrobě a současné funkčnosti, to znamená potřebný hardware a software. V další části popisuji jednotlivé úkony systému.

5

1. **VÝROBA A VÝVOJ TEPLOMĚRU**

Prvním krokem bylo objednání součástek na internetu a následné sestavení, které není vůbec těžké, jelikož zapojujete pouze 3 součástky. Zapojení hardwaru nebylo moc složité, jelikož stačilo zapojit 3 drátky. Po zapojení by to bylo k hardwaru asi všechno, zbytek projektu se skládá ze softwarových částí jako programování teploměru a Home Assistantu.

Vzhledem k tomu, že jsem s potřebnými softwarovými aplikacemi ještě nesetkal, musel jsem si zjistit co přesně dělají a jak fungují. Poté jsem nainstaloval ubuntu server, kde jsem musel nainstalovat Docker Engine bez kterého bych vlastně nezprovoznil celý Home Assistant a MQTT systém. Následovalo nainstalování a nastavení už řečeného Home Assistantu a MQTT. MQTT jsem si prvně zkoušel používat přes terminály a až poté do Home Assistantu.

Kód jsem upravoval a kontroloval postupně. Prvně jsem musel napsat kód na změření teploty. Měření teploty spočívá v tom, že termistor neměří ze začátku teplotu ale napětí a vy to musíte převést na teplotu. Vzoreček na tenhle výpočet jsem si našel na internetu. Poté následuje připojení ESP8266 na WiFi, MQTT broker a zaslání změřené teploty na Home Assistant, kde se to vykresluje do grafu. Do budoucna bych chtěl zařízení přesunout na PCB desku a vložil do krabičky, abych ho mohl mít venku nebo v jinačích částech domu. Také mě napadlo v budoucnu spojit teploměr s termostatem a fungovalo by to tak, že by se v domě topilo do doby než by přesáhl zadanou teplotu a následně by termostat vypl. Dále se dají připojit další zařízení na měření hodnot (např. tlak, vlhkost).

6

1. **PRINCIP FUNGOVÁNÍ TEPLOMĚRU**

Během vývoje se fungování přístroje několikrát změnilo v závislosti na nápadech.

Systém funguje jednoduše, když se teploměr zapojí do zdroje tak začne měřit napětí, které následně pomocí vzorce vypočítá na teplotu. Dále se připojí na WiFi a broker. MQTT pošle hodnoty na Home Assistant do grafu nebo jiného zobrazení, které si můžete přednastavit. Uživatel, který je připojen na síti, kde Home Assistant běží, si může prohlídnout graf teplot, které, teploměr změřil. Uživatel také nepotřebuje do procesu nějak zasahovat.

7

1. **VYUŽITÉ TECHNOLOGIE 3.1 Hardware**

**3.1.1 Seznam součástek**

* ESP-8266 D1 Mini
* Termistor NTC Vishay 10k
* Rezistor 10k
* Nepájivé pole se 400 kontakty

**3.1.2 ESP-8266 D1 Mini**

Základ projektu tvoří vývojová deska ESP-8266. ESP-8266 je nízkonákladový mikročip Wi-Fi s plnou schopností protokolu TCP/IP a schopností mikrokontroléru vyráběný společností Espressif Systems. Na čipu se nachází 2x8 pinů. 11 pinů je digitálních, napětí výstup 5V a 3,3V a analogový vstup.

**Parametry**

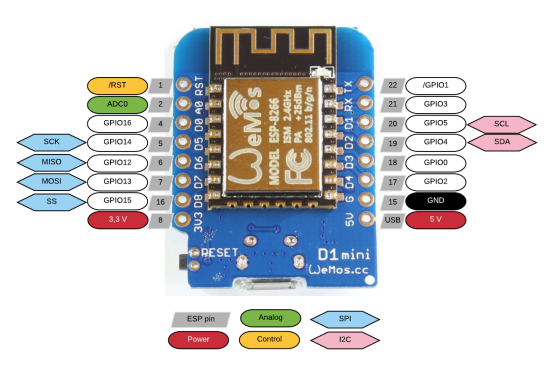
**-** Pracovní napětí: 3.3V

-Frekvence procesoru: 80MHz/160MHz

-Micro USB konektor

-Flash: 4MB



*ESP-8266*  *Piny ESP-8266*

8

**3.1.3 Termistor NTC Vishay 10k**

Termistor NTC pro měření teploty

**Parametry**

- Odpor: 10k Ohm

- Výkon P: 500mW

- Rozteč: 2.54mm

- Teplotní rozsah: - 40°C až 125°C

- Výrobce: Vishay

**3.1.4 Rezistor 10k**

je pasivní [elektrotechnická součástka](https://cs.wikipedia.org/wiki/Elektrotechnick%C3%A1_sou%C4%8D%C3%A1stka) projevující se v [elektrickém obvodu](https://cs.wikipedia.org/wiki/Elektrick%C3%BD_obvod) v ideálním případě jedinou vlastností – [elektrickým odporem](https://cs.wikipedia.org/wiki/Elektrick%C3%BD_odpor) (jednotka [Ohm](https://cs.wikipedia.org/wiki/Ohm), značka Ω).

**Parametry**

- Odpor: 10k Ohm

- Výkon: 250mW

- Tolerance rezistence:  ±0.1%

9

**3.2** **Software**

**3.2.1 Oracle VM VirtualBox 6.1.16**

Oracle VM VirtualBox je [multiplatformní](https://cs.wikipedia.org/wiki/Multiplatformn%C3%AD_software) [virtualizační nástroj](https://cs.wikipedia.org/wiki/Virtualizace) distribuovaný jak pro [Linux](https://cs.wikipedia.org/wiki/Linux)/[Unix](https://cs.wikipedia.org/wiki/Unix) tak pro [Windows](https://cs.wikipedia.org/wiki/Microsoft_Windows) a [Mac OS](https://cs.wikipedia.org/wiki/MacOS). Ten jsem potřeboval pro instalování ubuntu serveru ve kterém běží Docker Engine, Home Assistant a MQTT.

**3.2.2 Docker Engine 20.10.1**

Docker je v [informatice](https://cs.wikipedia.org/wiki/Informatika) název pro [otevřený software](https://cs.wikipedia.org/wiki/Otev%C5%99en%C3%BD_software) (open source projekt), jehož cílem je poskytnout jednotné rozhraní pro izolaci aplikací do [kontejnerů](https://cs.wikipedia.org/wiki/Virtualizace_na_%C3%BArovni_opera%C4%8Dn%C3%ADho_syst%C3%A9mu) v prostředí [macOS](https://cs.wikipedia.org/wiki/MacOS), [Linuxu](https://cs.wikipedia.org/wiki/Linux) i [Windows](https://cs.wikipedia.org/wiki/Windows_NT) („odlehčená virtualizace“). Docker Engine jsem instaloval z důvodu Home Assistant a poté MQTT. Předtím než jsem měl Docker tak jsem spouštěl Home Assistant přes VDI a následně jsem zjistil problém, že mi nepůjde nainstalovat MQTT, proto jsem použil Docker.

**3.2.3 Eclipse Mosquitto 2.0.4**

Eclipse Mosquitto je zprostředkovatel zpráv s otevřeným zdrojovým kódem, který implementuje protokol MQTT verze 5.0, 3.1.1 a 3.1. MQTT je potřeba abych mohl posílat změřenou teplotu na Home Assistant.

**3.2.4 Home Assistant**

Home Assistant je bezplatný a open-source software pro automatizaci domácnosti navržený jako centrální systém řízení domácí automatizace pro ovládání technologie inteligentních domů. Pomoci Home Assistantu můžu vidět hodnoty z teploměru. Určitě bych do budoucna chtěl s Home Assistantem více pracovat a vytvořit si chytrou domácnost.

10

**3.2.5 Visual Studio Code**

Visual Studio Code je bezplatný editor zdrojového kódu vytvořený společností Microsoft pro Windows, Linux a macOS. Vybíral jsem mezi Arduino IDE a Visual Studio Code, ale díky tomu, že jsme s VS Codem v minulosti více pracovali a měl jsem s ním více zkušeností tak jsem si ho vybral jako editor kódu.

11

1. **ZPŮSOBY ŘEŠENÍ A VYUŽITÉ TECHNOLOGIE**

**4.1** **Hardwarové zařízení**

**4.1.1 Základní funkce jazyka Arduino**

Jazyk Arduino obsahuje dvě základní funkce. Funkce setup() a loop(). Až na drobné úpravy je velmi podobný jazyku C a C++. Funkce setup(), slouží k inicializaci proměnných a nastavení potřebných hodnot. Je zavolá-na při spuštění nebo po restartu. Funkce loop() je hlavní funkce programu, která se stále opakuje.

**4.1.2 Připojení k WiFi**

Pro snadnější připojení k WiFi jsem využil knihovnu *ESP8266WiFi.h*. Díky této knihovně se zařízení samo pokusí o připojení k přednastavené WiFi, jméno a heslo od WiFi je nutno předem zapsat do zdrojového kódu.

void setup\_wifi() {

delay(10);

// Připojení k WiFi

Serial.println();

Serial.print("Připojování k ");

Serial.println(ssid);

WiFi.mode(WIFI\_STA);

WiFi.begin(ssid, password);

while (WiFi.status() != WL\_CONNECTED) {

delay(500);

Serial.print(".");

}

randomSeed(micros());

Serial.println("");

Serial.println("WiFi připojena");

Serial.println("IP address: ");

Serial.println(WiFi.localIP());

*Připojení k WiFi*

12

**4.1.3 Připojení k MQTT brokeru**

Abych dostal hodnoty do Home Assistantu, tak se musím připojit na vytvořený MQTT broker. K tomu mi slouží funkce *reconnect,* která se po vyvolání snaží připojit k přednastavené IP adrese od vytvořeného MQTT brokeru. Pokud se připojení nepovede, funkce se o to pokusí znova po 5 vteřinách a tak se to opakuje dokud se nepřipojí.

void reconnect() {

  // Opakování dokud se nepřipojí

  while (!client.connected()) {

    Serial.print("Zkouška připojení k MQTT...");

    // Vytvoření náhodného ID klienta

    String clientId = "ESP8266Client-";

    clientId += String(random(0xffff), HEX);

    // Pokus o připojení

    if (client.connect(clientId.c\_str())) {

      Serial.println("připojeno");

      client.subscribe("inTopic");

    } else {

      Serial.print("nepřipojeno, rc=");

      Serial.print(client.state());

      Serial.println(" zkus znova za 5 sekund");

      // Čekej 5 sekund na opakování

      delay(5000);

    }

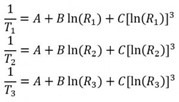
  }

}

*Připojení k MQTT*

**4.1.4 Měření teploty**

Samotný termistor nedokáže změřit teplotu ale pouze napětí. K tomu slouží Steinhart-Hartova rovnice termistoru, která vypočítá pomoci parametrů rovnice termistoru samotnou teplotu. Problém je v tom, že se ale teplota vypočítá v Kelvinech a proto aby mohla být ve stupních, tak se musí od vypočítané hodnoty odečíst 273.15.

137562035_218051283201901_15523901717404913_n 

*Parametry rovnice termistoru Steinhart-Hartova rovnice*

13

double Vout, Rth, teplota, adc\_value;

  adc\_value = analogRead(A0);

  Vout = (adc\_value \* VCC) / adc\_resolution;

  Rth = (VCC \* R2 / Vout) - R2;

  teplota = (1 / (A + (B \* log(Rth)) + (C \* pow((log(Rth)),3))));

  teplota = teplota - 273.15;

*Samotný výpočet teploty*

**4.1.5 Posílání hodnot na Home Assistant**

Pomocí knihovny PubSubClient.h mohu jednoduše publikovat zprávy se serverem, který podporuje MQTT. V kódu si nastavíte topic, do kterého budete zasílat hodnoty a na MQTT serveru budete tento topic poslouchat a hodnoty z něho brát.

unsigned long now = millis();

  if (now - lastMsg > 2000) {

    lastMsg = now;

    ++value;

    snprintf (msg, MSG\_BUFFER\_SIZE, " %lf", teplota);

    Serial.println(teplota);

    client.publish("/testtopic/teplota", msg);

*Posílání teploty na mqtt*

V mém případě posílám hodnoty z funkce *teplota* na topic */testtopic/teplota.*

14

**4.2** **Webová aplikace**

Jak jsem už psal, instalace probíhá v dockeru na ubuntu serveru. Po instalaci se vytvoří konfigurační soubor *configuration.yaml* ve kterém je potřeba aby se nám teplota vypisovala do grafu, nakonfigurovat sensor, který se poté vybere při tvorbě karty grafu.V konfiguraci se nastavuje platforma, jméno, topic a jednotky ve kterých měříme teplotu.

sensor: - platform: mqtt

name: "Měření teploty"

state\_topic: "/testtopic/teplota"

unit\_of\_measurement: 'C'

# value\_template: "{{ value\_json.RSSI }}"

*Konfigurace grafu*

15

1. **VÝSLEDEK ŘEŠENÍ**

S výsledkem jsem velice spokojen a s momentální funkčností taky, po několikadenním testování funguje, jak má a splňuje mé očekávání. Během vytváření mě napadalo mnoho vylepšení mého zařízení. Do budoucna mám rozhodně hodně plánů s mým teploměrem jako třeba ho dát do vytvořené krabičky, aby byl přenosný a mohl bych měřit teplotu ve všech místnostech. Také by bylo dobré napájet teploměr baterií a nemuset ho pořád zapojovat do počítače, což má svoje výhody. Možné by bylo taky propojení teploměru s termostatem aby se zapnul nebo vypnul podle naměřené teploty a takhle udržoval stálou teplotu. Rozhodně mám v plánu s tímhle pokračovat a budovat si chytrou domácnost. Myslím si, že tenhle projekt je dobrý pro začátek k takové záležitosti jako je tvorba chytré domácnosti.

16

**ZÁVĚR**

Cílem projektu bylo vytvořit funkční teploměr, který bude posílat teplotu na Home Assistant a já se tak mohl sledovat teplotu mého pokoje. Všechny cíle, které jsem měl jsem splnil, zařízení i aplikace funguje jak mají a nejsou s teploměrem žádné problémy. Můj pokoj je nejchladnější místnost v domě a takhle se můžu já a rodiče dívat na aktuální teplotu pokoje a rozhodnout se, jestli zatopit nebo ne. V budoucnu svůj výtvor vylepším o spojení s termostatem a ovládání teploty na dálku, ale to je jenom jeden z mála vylepšení, které se můžou provést. S chytrou domácností jsem neměl v minulosti žádné zkušenosti a tohle je si myslím dobrý začátek.

Link na můj Github: https://github.com/Rawenexik/Projekt

17

**SEZNAM POUŽITÝCH INFORMAČNÍCH ZDROJŮ**

1. *ESP8266 MQTT* [online].[cit.2021-1-9].Dostupnéz:https://iotdesignpro.com/projects/how-to-connect-esp8266-with-mqtt
2. *Thermistor* [online].[cit.2021-1-9].Dostupnéz <https://navody.dratek.cz/arduino-projekty/mereni-teploty-s-termistorem.html>
3. *PubSubClient* [online].[cit.2021-1-9].Dostupnéz https://github.com/knolleary/pubsubclient
4. *Docker HA* [online].[cit.2021-1-9].Dostupnéz <https://www.home-assistant.io/docs/installation/docker/>
5. *Docker MQTT* [online].[cit.2021-1-9].Dostupnéz <https://hub.docker.com/_/eclipse-mosquitto>
6. Sensor [online].[cit.2021-1-9]. Dostupné z https://www.home-assistant.io/integrations/template/

18