

|  |
| --- |
| **Závěrečná studijní práce**  **Dokumentace** |
| **Detektor deště a vlhkosti s webovou aplikaci** |
| *Jan Spiška* |

|  |  |
| --- | --- |
| **Obor:** | 18-20-M/01 INFORMAČNÍ TECHNOLOGIE  se zaměřením na počítačové sítě a programování |
| **Třída:**  **Školní rok:** | IT4  2020/2021 |

###### Poděkování

*Děkuji panu učiteli Ing. Petru Grussmannovi za pomoc při zapojování obvodu a propůjčení pájné soustavy a panu učiteli Mgr. Marcelu Godovskému za pomoc s ESP8266.*

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**

Projekt se zabývá získáváním informací o dešti a vlhkosti které se posílají na webovou aplikaci pomocí wifi. Skládá se z hardwarové a softwarové části. Základ hardwarové části tvoří deska ESP8266. Zařízení pomocí senzoru vlhkosti pošle informaci do webové aplikace o vlhkosti vzduchu a také informaci, zda prší nebo ne, což je věc kterou ocení npř. Majitelé zahrádek nebo polí, kteří se musí starat o neustálé zalívaní rostlin, a nebo by mohly deště zničit úrodu. Programová část je řešena v jazyce Arduino, což je kombinace jazyků C a C++. Druhou část projektu tvoří webová aplikace vyvinutá v jazyce HTML. Tato aplikace vypisuje měřené hodnoty, kdykoliv dostupné pro uživatele.

**Klíčová slova**

Senzory; počasí; ESP8266; WiFi; webová aplikace;

**Obsah**

[**Úvod** 5](#_Toc61865935)

[**1.** **Výroba detektoru deště a vlhkosti** 6](#_Toc61865936)

[**2.** **Využité technologie** 7](#_Toc61865937)

[**2.1.** **Hardware** 7](#_Toc61865938)

[**2.1.1.** **Seznam součástek** 7](#_Toc61865939)

[**2.1.2.** **NodeMcu** 8](#_Toc61865940)

[**2.1.3.** **Rain Sensor** 8](#_Toc61865941)

[**2.2.** **Napájení** 9](#_Toc61865942)

[**2.3.** **Software** 9](#_Toc61865943)

[**2.3.1.** **Jazyk Arduino** 9](#_Toc61865944)

[**2.3.2.** **Webová aplikace** 9](#_Toc61865945)

[**2.3.3.** **Visual Studio Code** 9](#_Toc61865946)

[**2.3.4.** **KiCad** 9](#_Toc61865947)

[**3.** **Způsoby řešení a použité postupy** 10](#_Toc61865948)

[**3.1.** **Hardwarové zařízení** 10](#_Toc61865949)

[**3.1.1.** **Nahrávání programu** 10](#_Toc61865950)

[**3.1.2.** **Popis fungování detektoru deště a vlhkosti.** 10](#_Toc61865951)

[**3.1.3.** **Připojení k wifi** 10](#_Toc61865952)

[**3.1.4.** **Komunikace se serverem** 10](#_Toc61865953)

[**3.1.5.** **Webová aplikace** 11](#_Toc61865954)

[**4.** **Výsledky řešení** 11](#_Toc61865955)

[**4.1.** **Hardwarové zařízení** 11](#_Toc61865956)

[**4.2.** **Funkčnost webové aplikace** 12](#_Toc61865957)

[**4.3.** **Schéma** 13](#_Toc61865958)

[**Závěr** 14](#_Toc61865959)

[**Zdroje** 15](#_Toc61865960)

# **Úvod**

Rozhodl jsem se jako ročníkový projekt vytvořit detektor deště a vlhkosti s webovou aplikaci, protože mě vždycky bavilo pracovat s různými senzory (když jsem byl mladší, chodil jsem do elektro-kroužku a hráli jsme si s detektory pohybu, což mě vždy hodně bavilo), proto jsem si chtěl opět nějaký senzor vyzkoušet.

Hlavním cílem projektu bylo propojit zařízení s webovou aplikací, aby si uživatel mohl kdykoliv a odkudkoliv zkontrolovat měřené hodnoty a zjistit, zda npř. prší nad jeho polem či zahradou, když zrovna není přítomen.

Jelikož se jednalo o můj první projekt v této oblasti, zvolil jsem pro jeho zhotovení známé a jednoduché technologie. Hardwarovou část tvoří hlavně deska NodeMcu, která již obsahuje WiFi modul ESP8266, k ní připojené další komponenty, o kterých se zmíním v dalších částech dokumentace. Zařízení je naprogramováno v jazyce Arduino, jedná se o kombinaci jazyků C a C++. Webová aplikace je dělána v jazyce HTTP. Komunikace mezi senzorem a aplikaci probíhá přes wifi.

V této dokumentaci podrobně popisuji výrobu detektoru deště a vlhkosti a princip jeho fungování. Na začátku se zmiňuji o problémech, se kterými jsem se setkal při vývoji, pokračuji popisem technologií nezbytných k jeho výrobě i k jeho současné funkčnosti a rozebírám, jak funguje nejen samotné zařízení, ale i webová aplikace. V další části vysvětluji, na jakých principech funguje jejich vzájemná komunikace, a popisuji jednotlivé úkony obou částí systému. V závěru se zabývám současnou podobou projektu a hodnotím odvedenou práci.

# **Výroba detektoru deště a vlhkosti**

První část mého projektu byla o sestavení zařízení. Když přišla fáze nakupování a objednávání součástek, objevila se druhá vlna koronaviru, tudíž jsme museli pracovat z domu bez rozsáhlé komunikace s panem Grussmannem. Nejdříve jsem chtěl projekt dělat na desce Arduino od firmy Atmel, ale po konverzaci s panem Grussmannem přes BlueButton jsem se rozhodl použít NodeMcu s ESP8266.

Vzhledem k tomu, že jsem se součástkami nikdy nepracoval bylo pro mě jejich zapojování hodně obtížné, proto jsem postupoval po malých krocích a každou součástku hned testoval. Při testování jsem zjistil že moje ESP je zastaralé a aby fungovalo s technologii kterou používám musel bych si dokupovat adaptér na který bych musel zase necelý měsíc čekat. Zašel jsem tedy za panem Godovským, který mi hned poradil a prodal ESP které pro mě bude mnohem lepši. Finální verzi projektu jsem nejdříve navrhl v programu KiCad, kde jsem si odzkoušel rozvržení součástek na nepájivém poli.

Po zapojení a odzkoušení všech součástek jsem přešel k programování samotného senzoru. Pro tento účel jsem zvolil programovací jazyk přímo pro Arduino, což je kombinace jazyků C a C++. Když byla první část projektu plně funkční, čekala mě ta pravá výzva – spojit zařízení přes WiFi s webovou aplikací a umožnit tak jeho ovládání na dálku, což byl, jak jsem již zmiňoval, hlavní cíl mého projektu.

Webovou aplikaci jsem se rozhodl vyřešit pomocí http, jelikož nás s ní seznamoval pan učitel Lučný.

# **Využité technologie**

# **Hardware**

# **Seznam součástek**

* Vývojová deska NodeMcu
* Rain Sensor
* Nepájivé pole
* LE dioda

# **NodeMcu**

Základ projektu tvoří NodeMcu s WiFi modulem ESP8266 od společnosti Espressif Systems. NodeMcu obsahuje 12 digitálních GPIO pinů a jeden analogový pin. Pro program je k dispozici paměť o velikosti 128 kB. NodeMcu pracuje s napětím 3,3 V.

# **Rain Sensor**

Podstatnou součástí mého projektu je Dešťový senzor (též snímač nebo detektor deště) je spouštěcí zařízení reagující na déšť. Zařízení se používá především v automobilovém průmyslu a v závlahových systémech.

void loop() {

int rainAnalogVal = analogRead(rainAnalog);

  boolean bIsRaining = !(digitalRead(rainDigital));

  Serial.println(rainAnalogVal);

  Serial.print("\t");

  Serial.println(bIsRaining);

  delay(1000);

}

*Př. Senzor vody v autě*

Nejběžnější moderní dešťové senzory jsou založeny na principu úplného vnitřního odrazu. Infračervené světlo je po celou dobu paprskem vyzařováno pod úhlem 45 stupňů do čelního skla z interiéru. Pokud je sklo suché, je kritický úhel pro celkový vnitřní lom kolem 42 °. Tato hodnota je získána z celkového vzorce interního lomu

kde n = 1 je přibližná hodnota indexu lomu vzduchu pro infračervené záření a n = 1,5 je přibližná hodnota indexu lomu skla, rovněž pro infračervené záření. V takovém případě, když je úhel dopadu světla 45 °, se veškeré světlo odráží a detektor přijímá maximální intenzitu.

Pokud je sklo mokré, kritický úhel se změní na přibližně 60 °, protože index lomu vody je vyšší než vzduch (n = 1,3). V takovém případě, když je úhel dopadu je 45 °, se nezíská celkový vnitřní odraz. Část světelného paprsku prochází sklem a intenzita měřená při odrazu je nižší: systém detekuje vodu a zapnou se stěrače.

# **Napájení**

Napájení je řešeno pomocí napájecího modulu nepájivého pole, který dokáže připojit napájení do dvou napěťových větví. Modul vyžaduje vstupní napětí od 6,5 V do 12 V, které lze přivést kabelem z elektrické sítě nebo pomocí USB z počítače. Maximální výstupní proud je 700 mA, což je dostačující všechny součástky

Všechny komponenty jsem připojil na nepájivá pole.

# **Software**

# **Jazyk Arduino**

Program jsem napsal v jazyce Arduino, který je velmi podobný jazyku C nebo C++. Jazyk Arduino byl vytvořen přímo k programování integrovaných obvodů.

# **Webová aplikace**

Webovou aplikaci jsem řešil v jazyce http. Při tvorbě uživatelského rozhraní jsem použil framework Bootstrap v4.5.3, který mi usnadnil práci s HTML5 a CSS, hlavně v oblasti responzibility. Vzhled uživatelského rozhraní jsem řešil pomocí kaskádových stylů CSS3 a také pomocí javascriptové knihovny jQuery. Celá aplikace běží na Apache HTTP Serveru.

# **Visual Studio Code**

Jako vývojové prostředí jsem pro program zvolil Visual Studio Code, jelikož jsem v něm zvyklý pracovat a přijde mi graficky přijemnější.

# **KiCad**

Pro návrh bezpečnostního zařízení jsem použil open-source program KiCad ve verzi 5.1.9. Ten umožňuje graficky sestavit zapojení jednotlivých součástek na nepájivém poli, což pomáhá k představě o výsledném rozložení.

# **Způsoby řešení a použité postupy**

# **Hardwarové zařízení**

# **Nahrávání programu**

Veškeré zapojení v nepajivém poli je umístěno v uzavřené krabičce(senzor je na vrchu krabičky), po zapojení by měl systém bezpežně pracovat.

# **Popis fungování detektoru deště a vlhkosti.**

Když zapneme NodeMcu spustí se setup, poté se modul připojí k WiFi pomocí knihovny WiFiManager a zavolá loop, funkčnost senzoru si ověříme pomocí led, která by měla svítit jen pod podmínkou kontaktu s vodou. Pokud zjistíme že senzor funguje, můžeme celý přistroj umístit na libovolné místo v dosahu wifi a přesunout se k webové aplikaci, kde stačí pomocí tlačítka **zjistit hodnoty deště a vlhkosti**, získat uvedené hodnoty.

# **Připojení k wifi**

Pro připojení k WiFi jsem využil knihovnu WiFiManager. Pomocí ní se zařízení po zapnutí automaticky pokusí o připojení k síti. Pokud se to nepodaří, vytvoří se web server, na kterém uživatel zadá heslo ke své WiFi. Tato síť bude nadále použitá jako defaultní.

# **Komunikace se serverem**

Pro ovládání zařízení přes wifi je nutná jeho komunikace se serverem. Tu zajišťuji pomocí HTTP GET a POST požadavků z klienta na server. Funkce getValues se připojí k serveru a získá JSON s daty. Pomocí knihovny ArduinoJson dekóduje JSON data. Ta se pak uloží do proměnných, se kterými lze dále pracovat. Odesílání dat na server zajišťuje funkce postValues. Zbytek obstarává web. aplikace.

# **Webová aplikace**

Při řešení designu aplikace mi pomohl framework Bootstrap, díky kterému je uživatelské rozhraní plně responzivní. Důležitou částí aplikace je databázový systém MySQL, do kterého prostřednictvím Nette ukládám data, ke kterým pak můžu přistupovat.

# **Výsledky řešení**

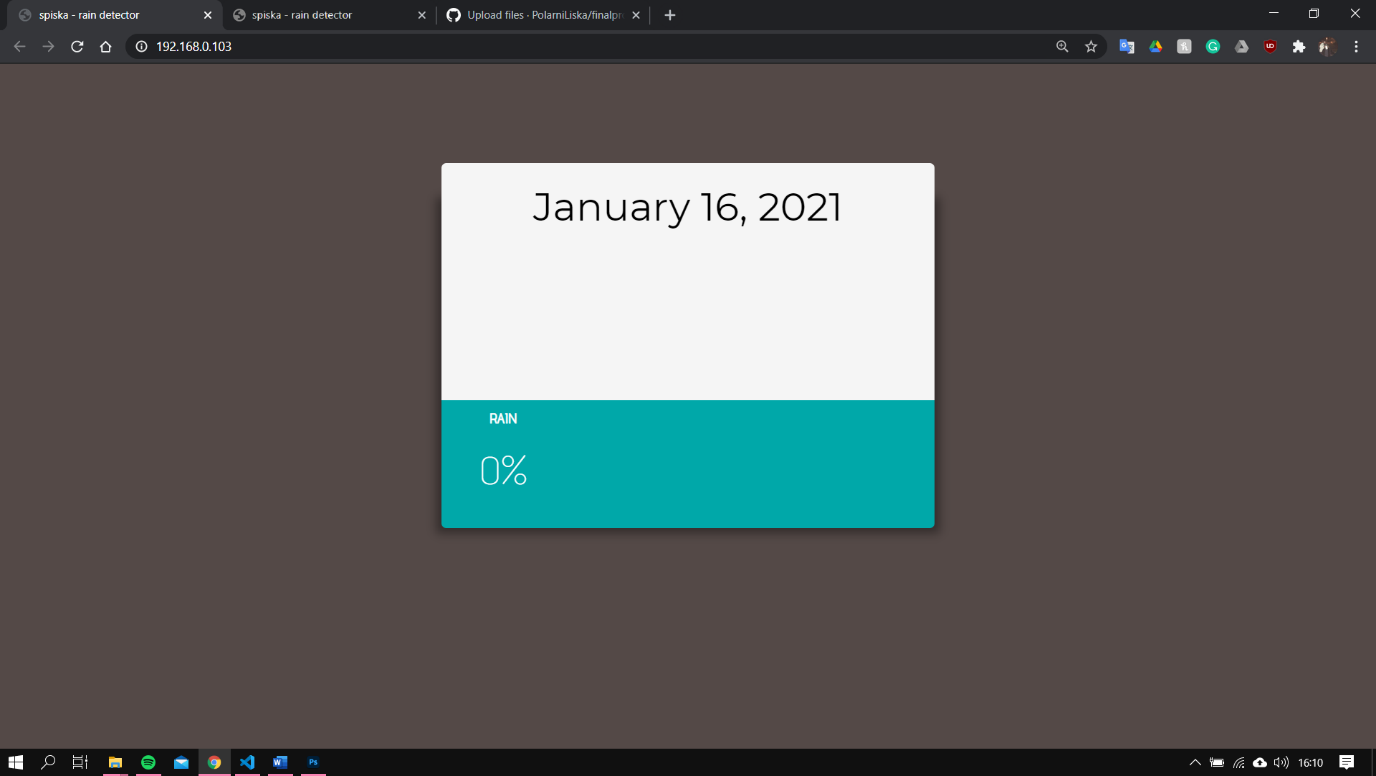
# **Hardwarové zařízení**

Detektor deště a vlhkosti je v současné době funkční, všechny jeho funkce fungují. Zařízení je umístěno na dvou nepájivých polích. Do budoucna by bylo vhodné vytvořit pevný plošný spoj na poli pájivém. Systém jsem umístil do jednoduché krabičky se senzorem na vrchu.

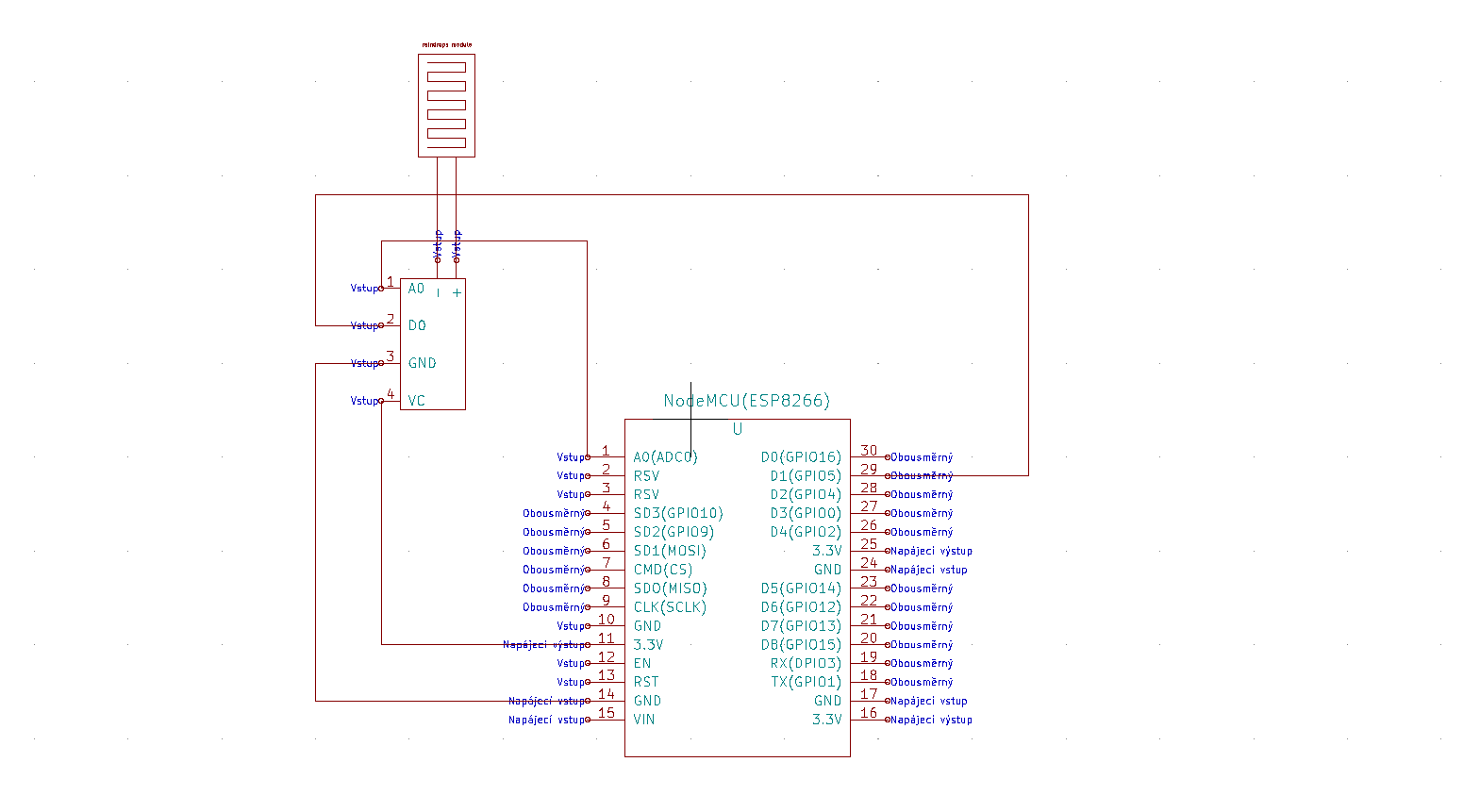
****

# **Funkčnost webové aplikace**

Webová aplikace splňuje svůj účel, uživatelé ví přesně co může v aplikaci dělat. Aplikace je také plně responzivní, lze používat i na počítači, mobilu i tabletu. Do budoucna bych rád přidal více měřených hodnot.



# **Schéma**



# **Závěr**

Cílem projektu bylo vytvořit detektor deště s možností zobrazovaní hodnot na dálku prostřednictvím webové aplikace. Vytyčené cíle byly splněny, detektor i webová aplikace bez problému fungují. Při vývoji mě napadlo do budoucna přidat rain gauge na měření hodnoty kolik deště napršelo. Na tato vylepšení mi již bohužel nestačil čas před termínem odevzdání práce, proto se jim budu věnovat později. Samotné součástky bych rád umístil na desku plošných spojů místo dvou nepájivých polí a zařízení umístil do vymodelované krabičky.

# **Zdroje**

[1] <https://www.hackster.io/KaustubhAgarwal/rain-warning-display-on-mobile-for-your-home-garden-86c691>  
[2] <https://www.easyduino.cz/Destovy-senzor-detektor-deste-kompatibilni-s-Arduino-d169.htm>  
[3] <https://www.thingiverse.com/thing:2163969>  
[4] <https://www.instructables.com/id/NodeMCU-Hall-Effect-Rain-Gauge/>  
[5] <https://create.arduino.cc/projecthub/MisterBotBreak/how-to-use-a-rain-sensor-bcecd9>  
[6] <https://www.instructables.com/id/Arduino-Rain-Gauge-Calibration/>  
[7] [https://www.allaboutcircuits.com/projects/build-a-wireless-tipping-bucket-rain-gauge- part-1assembling-the-base/](https://www.allaboutcircuits.com/projects/build-a-wireless-tipping-bucket-rain-gauge-%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20part-1assembling-the-base/)  
[8] <https://arduino-shop.cz/arduino/1352-esp8266-bezdratovy-modul-esp-12f-ap-sta.html>