

Abstrakt

Semestrální práce se zabývá vývojem chytrého zásuvkového termostatu, který je schopen komunikovat s uživatelem a dalšími aplikacemi pomocí wifi rozhraní v pásmu 2,4 GHz. Související problematikou je elektrické měření teploty. Dalšími oblastmi, na které se dokument zaměřuje, jsou IoT online cloudy, které se v dnešní době používají pro logování a vizualizaci naměřených dat.

Bibliografická citace

VYMAZAL, Tomáš. Regulátor teploty s ESP8266. Brno, 2020. Dostupné také z: <https://www.vutbr.cz/studenti/zav-prace/detail/130294>. Semestrální práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedoucí práce Ivo Lattenberg.

Prohlášení autora o původnosti díla

Jméno a příjmení studenta:	Tomáš Vymazal
VUT ID studenta:	214411
Typ práce:	semestrální práce
Akademický rok:	2020/21
Téma závěrečné práce:	Regulátor teploty s ESP8266

Prohlašuji, že svou závěrečnou práci jsem vypracoval samostatně pod vedením vedoucí/ho závěrečné práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené závěrečné práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne: 8. prosince 2020

podpis autora

Obsah

SEZNAM OBRÁZKŮ	9
SEZNAM TABULEK.....	10
ÚVOD	11
1. PROBLEMATIKA MĚŘENÍ TEPLOTNÍMI SENZORY	12
1.1 ANALOGOVÉ TEPLOTNÍ SENZORY [1].....	12
1.1.1 Kovová čidla	12
1.1.2 Polovodičová teplotní čidla	13
1.2 DIGITÁLNÍ TEPLOTNÍ SENZORY	16
1.2.1 DS18B20 [6].....	16
1.2.2 AM2302 [7].....	17
1.3 TERMOČLÁNKY [1].....	18
2. ROZBOR IOT CLOUDŮ	19
2.1 THINGSPEAK.....	20
2.2 TMEP [10].....	21
2.3 OPENHAB	22
2.4 MQTT.....	22
3. HARDWARE.....	23
3.1 MIKROKONTROLÉR A JEHO PERIFERIE	24
3.1.1 ESP8266.....	25
3.2 OVLÁDÁNÍ SPÍNANÉ ZÁSUVKY	26
3.3 USB TO UART PŘEVODNÍK	28
3.4 I2C EEPROM PAMĚŤ	29
3.4.1 I2C	29
3.5 VÝROBA A NÁVRH DPS.....	30
3.6 OSAZENÍ A OŽIVENÍ DPS	34
3.6.1 Deska zařízení verze 1.....	34
3.6.2 Oživení zařízení verze 1.	35
3.6.3 Deska zařízení verze 2.....	36
3.6.4 Oživení zařízení verze 2.	37
3.6.5 Deska zařízení verze 3.....	39
3.6.6 Oživení zařízení verze 3.	40
4. FRIMWARE.....	42
4.1 POUŽITÉ KNIHOVNY	44
4.2 VYTVOŘENÉ KNIHOVNY	45
4.3 POUŽITÉ KNIHOVNY	46
5. ROZHRANÍ ZAŘÍZENÍ	47
5.1 WEBOVÉ ROZHRANÍ	47
5.2 TERMINÁL	47
5.3 SIGNALIZAČNÍ DIODY	47

6. ZÁVĚR.....	49
LITERATURA.....	50
SEZNAM PŘÍLOH.....	52
OBSAH PŘILOŽENÉHO ARCHÍVU	53

SEZNAM OBRÁZKŮ

1.1	Teplotní senzor PT100	13
1.2	NTC termistor 640-10K	14
1.3	NTC termistor B57164K0472K000	14
1.4	Foto senzoru KTY81 110	15
1.5	Jednoduchý způsob odečítání hodnot z analogového teplotního čidla.	15
1.6	Foto teplotní senzoru DS18B20	16
1.7	Foto teplotní senzoru DS18B20	17
1.8	Graf odchylky při měření teploty pomocí AM2302[7]	17
1.9	Graf odchylky při relativní vlhkosti pomocí AM2302	18
1.10	Tabulka přehledu termočlánků [8]	18
2.1	Ukázka z uživatelského rozhraní ThingSpeak [13]	20
2.2	Mapa čidel vyexportovaná 6.12.2020 v 17:15 [10]	21
3.1	Schéma zapojení obvodu relé.	26
3.2	Schéma zapojení USB převodníku na UART.	28
3.3	Schéma desky mikrokontroleru ESP [18]	25
3.4	3D model navržené desky, export KiCad	31
3.5	Vyrobená deska - přední strana	34
3.6	Vyrobená deska - zadní strana	35

SEZNAM TABULEK

2.1	Tabulka nejpoužívanějších IoT cloudu	19
-----	--	----

ÚVOD

Zadáním semestrální práce je vývoj a výroba elektrického regulátoru teploty s určitými parametry, jako jsou například možnost připojení zařízení do internetové sítě nebo schopnost spínat síťové napětí 230 V s proudem 16 A. Pod regulátorem teploty si každý může představit něco jiného. Zařízení vyvíjené v této práci se často označuje jako elektronický termostat, jedná se o zařízení, které je schopné spínat topení a docílit tak požadované/nastavené teploty na termostatu. Regulátorem teploty tedy rozumíme zařízení, které obsahuje jak část termostatu, tak samotného topného tělesa. Tato práce se zabývá pouze návrhem části termostatu, jako topné zařízení se zde uvažuje libovolné elektrické topné zařízení, které lze připojit do zásuvky a je schopno vytápět nějaký prostor v okolí našeho termostatu.

Nedílnou částí této práce je také analýza potřebné teorie ohledně měření teploty, způsobu měření a rozbor typů jednotlivých čidel, které se používají v elektrotechnice pro měření teploty. Problematika klade důraz na využitelnost pro vyvíjené zařízení tak, aby bylo schopno měřit teplotu v intervalu -20 až 50 °C.

Dalším tématem, kterým se práce zabývá je problematika IoT cloudu jakožto možného uložště dat získaných z termostatu. Výběr vhodného cloudu se nesmí podcenit, protože v této době existuje mnoho služeb, které by se daly k tomu účelu použít. Každé, pro projekt použitelné uložště, má specifické požadavky na komunikaci se zařízeními, tudíž komplikuje možnost migrace do jiné služby. Jediným a častým společným jmenovatelem je podpora MQTT protokolu, jakožto možnosti komunikace zařízení s cloudovou službou. Z toho důvodu je zde tento protokol důkladně prostudován a v budoucnu bude i implementován do konečného zařízení.

V poslední části práce je navrhován požadovaný regulátor teploty tak, aby vyhověl všem požadavkům ze zadání. Práce se především zabývá návrhem DPS, jejím osazením a oživením. Součástí práce zatím není výsledný firmware, který bude řídit celé zařízení. Oživením je proto chápáno pouhé otestování periférií, které jsou na desce osazeny a ověření jejich funkcí.

1. PROBLEMATIKA MĚŘENÍ TEPLOTNÍMI SENZORY

Elektrický senzor pro měření teploty je součástka, u které se mění její vlastnosti, většinou vodivost, v závislosti na okolní teplotě. V praxi se můžeme setkat i s jinými typy senzorů, například senzory pro bezdotykové měření teploty, které ale nejsou součástí této studie. Jejich použití pro tuto práci by postrádalo smysl, a navíc by bylo zbytečně dražší a složitější než použití dotykového čidla.

Dotyková čidla se dále dělí na elektrická, u kterých se mění fyzikální veličina, kterou lze v elektrotechnice snadno změřit, a na neelektrická. Typickým zástupcem neelektrických čidel je rtuťový teploměr, u kterého se mění objem a tím pádem je kapalina vytlačována vzhůru do kapiláry, na které je stupnice. Automatizované odečítání hodnot z takového neelektrického čidla by bylo velice náročné, nehledě na nutnost druhého čidla, které by do systému vneslo další chybu.

Dotykové elektrické senzory můžeme rozdělit do dvou kategorií, a to podle toho, jakým způsobem z nich vyčítáme naměřenou hodnotu do komponenty, která signál dále zpracovává. První kategorií jsou analogové teplotní senzory, u nichž dochází ke změně elektrických vlastností v závislosti na teplotě. Druhou kategorií jsou digitální teplotní senzory. Ty jsou konstrukčně mnohem komplikovanější. Ve vnitřním zapojení takového digitálního senzoru opět najdeme analogový teplotní senzor, který je doplněn další periferií, která zpracovává naměřené hodnoty a ukládá aktuální hodnotu teploty do vnitřních registrů součástky.

1.1 Analogové teplotní senzory [1]

Jedná se o nejobecnější kategorii teplotních senzorů, proto je potřeba ji dále rozčlenit především podle použitých materiálů, ze kterých je dané čidlo vyrobeno. Základními skupinami jsou:

- kovová čidla,
- polovodičová čidla,
- termočlánky.

1.1.1 Kovová čidla

Jedná se o čidla vyrobená z velmi čistých kovů, například platina, nikl, molybden nebo měď. Uvnitř čidla můžeme nalézt tenký drátek určité délky, který je stočený do cívky. Jako jádro této cívky jsou použity nevodivé materiály, jako například sklo nebo keramika. Typickým zástupcem této kategorie je teplotní čidlo Pt100, z jehož označení můžeme odvodit, že se jedná o platinové čidlo (Pt) a jeho odpor při 0 °C je 100 Ω.

Pro každý typ čidla existuje tabulka naměřených hodnot, která udává závislost odporu na teplotě. Odečítání teploty z kovového čidla probíhá tak, že musíme přesně změřit

odpor čidla, a pak pomocí tabulky vyčteme hodnotu teploty, která náleží právě změřenému odporu.

Obrovskou nevýhodou těchto čidel je jejich křehkost a cena. Další nevýhodou je, že na čidlu dochází k relativně malé změně odporu při pokojových teplotách, proto je potřeba čidlo provozovat v zapojení s pomocným zesilovačem, což jeho použití opět prodražuje. Jako výhodu čidla můžeme uvést, že je velice přesné a pracuje na velkém teplotním rozsahu (-50 až 400 °C pro Pt100). [2]

Při návrhu zesilovače pro Pt100 nesmíme zapomenout, že čidlem musí protékat proud menší než 1 mA, proto je třeba dbát zvýšené pozornosti při výběru odporů.



Obrázek 1.1 Teplotní senzor PT100

1.1.2 Polovodičová teplotní čidla

Základem každého polovodiče je velmi čistý monokrystal křemíku (Si), který je následně obohacen o příměsi prvků ze III. skupiny (B, Al, Ga, In). V takovém případě hovoříme o polovodiči typu P. Pokud se k monokrystalu přidá příměs prvků z V. skupiny (F, As, Sb), pak hovoříme o polovodiči typu N. [1][3]

Polovodičové teplotní senzory se dělí na termistory a na monokrystalické odporové senzory. Termistory můžeme dále dělit na pozistory (PTC) a na negastory (NTC).

Výroba termistorů je trochu rozdílná. Místo N a P přechodu se spíše využívají práškové oxidy kovů (Fe_2O_3 , TiO_2 , CuO , MnO , NiO , CoO , BaO), které se následně spékají. [4] Z toho se nám může zdát, že se nejedná o polovodiče v pravém slova smyslu, přesto se termistory řadí do polovodičových senzorů teploty, protože v nich nastává podobný děj jako v polovodičích. Se změnou teploty dochází k uvolnění nosičů elektrického proudu, tím pádem se materiál stává vodivým. [5]

PTC (*positive temperature coefficient*) termistory mají pozitivní teplotní součinitel odporu, zjednodušeně řečeno s rostoucí teplotou roste odpor. Jejich využití je hlavně v tom, že se dají použít jako vratné teplotní pojistky, například ve vinutí motoru. Měření teploty pomocí PTC termistoru není dobrou volbou, protože teplotní závislost je velice strmá. Častěji se používají pro kvantitativní měření, kde nejsou velké nároky na přesnost.

NTC (*negative temperature coefficient*) termistory mají negativní teplotní součinitel odporu, neboli s rostoucí teplotou klesá odpor termistoru. Jeho teplotní charakteristika se může aproximovat exponenciálou, proto odečítání umožňuje měřit malé změny teploty bez větších potíží. Malou nevýhodou je, že každá součástka vykazuje trochu jiný průběh teplotní charakteristiky, který se může časem mírně měnit. Výhodou NTC termistoru je jeho cena. Díky tomu se často používá pro bodové měření teploty. [4]



Obrázek 1.2 NTC termistor 640-10K



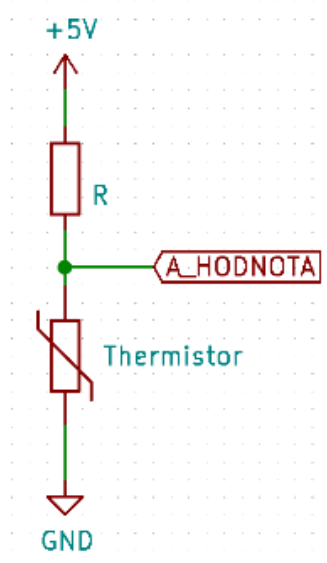
Obrázek 1.3 NTC termistor B57164K0472K000

Monokrystalické senzory se vyrábějí z křemíku (Si), přesněji pomocí polovodiče typu N. S rostoucí teplotou u nich dochází k rozptylu nosičů, to má za následek snížení vodivosti neboli růst odporu. Oproti PTC termistorům mají mírnější strmost teplotní charakteristiky, proto jsou vhodné pro měření teplot v rozsahu od -55 do 150 °C. Typickým zástupcem této skupiny je teplotní senzor s označením KTY. [5]



Obrázek 1.4 Foto senzoru KTY81 110

Nejjednodušším způsobem, jak můžeme analogové čidlo připojit k nějakému mikrokontroleru, je zapojit ho do série s rezistorem a pomocí analogového převodníku vyčítat úbytek napětí na čidle, viz. Obrázek 1.5. Jedná se o opravdu jednoduché zapojení, kde nemůžeme klást velké nároky na přesnost. Přesně můžeme měřit pouze teploty blízké pokojové teplotě. Navíc vyčítaná hodnota není lineární. To je způsobeno použitím AD převodníku. Přesnost měření obzvláště ovlivňuje počet jeho kvantizačních úrovní.



Obrázek 1.5 Jednoduchý způsob odečítání hodnot z analogového teplotního čidla.

1.2 Digitální teplotní senzory

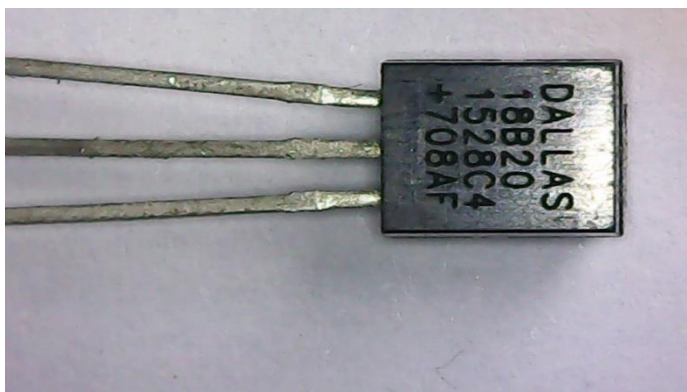
Digitálními teplotními senzory se chápou součástky, které v sobě mají implementováno měření teploty pomocí analogového teplotního senzoru. Takto naměřená teplota je následně digitálně zpracována a uložena do registru uvnitř součástky. Pro získání dat musíme senzor připojit na sběrnici. Poté můžeme se součástkou komunikovat a vyčítat z ní data, nebo nastavit v jakých jednotkách (°C nebo F) má data posílat. Nejčastějšími typy sběrnic/komunikačních protokolů jsou I2C, 1-Wire, SPI. Typickými zástupci v této kategorii jsou DS18B20 nebo AM2302.

1.2.1 DS18B20 [6]

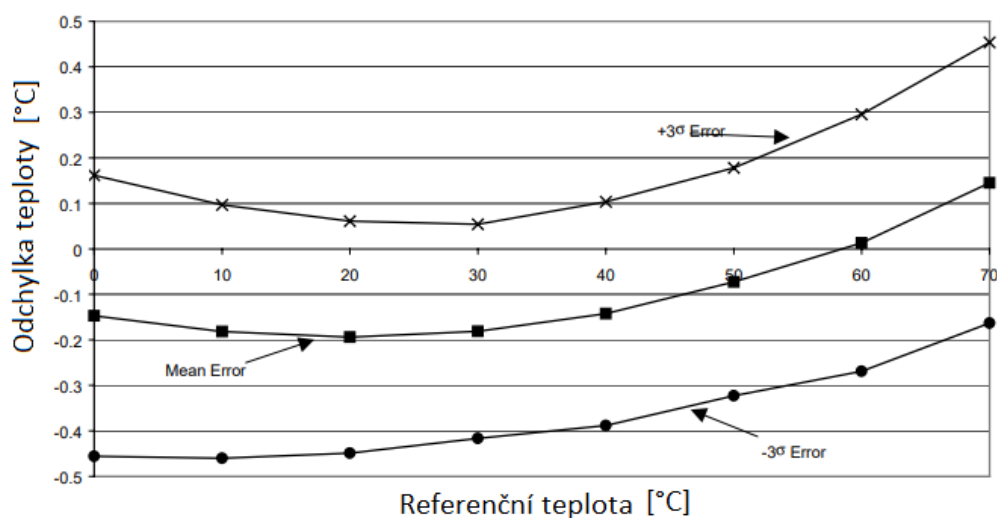
Digitální teplotní senzor měří teplotu v rozsahu od -55 do 125 °C. Poprvé ho vyrobila firma DALLAS SEMICONDUCTOR, proto se mu lidově říká „dallas“. Senzor komunikuje pomocí 1-Wire sběrnice. Teplota je v senzoru uložena jako 12 bitové slovo.

Dalšími možnostmi, které čidlo nabízí, je nastavení přesnosti měření, a to v rozsahu 9 až 12 bitů. Snížení přesnosti měření urychluje měřicí proces. Díky tomu můžeme několikanásobně urychlit měření, které je při nejvyšší přesnosti velice pomalé.

Čidlo je dále vybaveno interní logikou, díky které se samo může chovat jako termostat.



Obrázek 1.6 Foto teplotní senzoru DS18B20

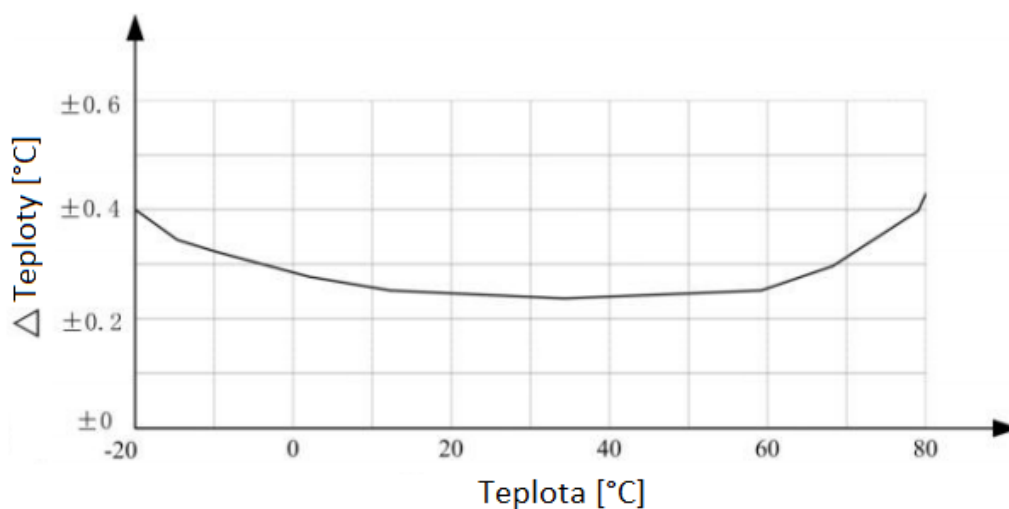


Obrázek 1.7 Foto teplotního senzoru DS18B20

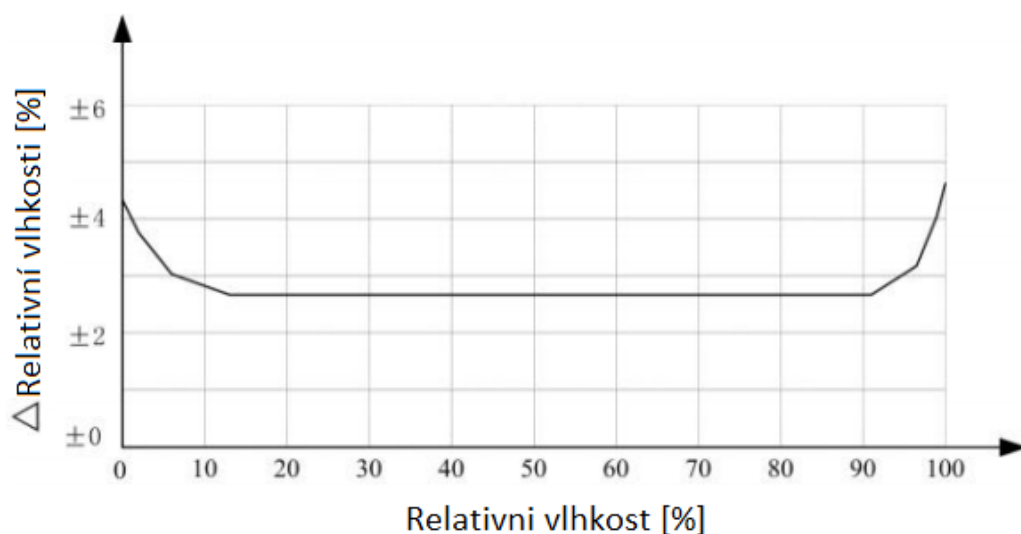
1.2.2 AM2302 [7]

Jedná se o další často využívané čidlo především proto, že je součástí DHT22 teplotního senzoru. Obrovskou výhodou čidla je, že dokáže s teplotou měřit i relativní vlhkost vzduchu. Jeho nevýhoda je přesnost (viz. Obrázek 1.8) oproti předchozímu DS18B20 (Obrázek 1.7), přestože AM2302 je schopno rozlišit měřenou hodnotu teploty na 16 bitů.

Pro komunikaci čidlo využívá Single-bus komunikační protokol. Jedná se o upravenou verzi 1-Wire protokolu, přesto je nelze zaměňovat.



Obrázek 1.8 Graf odchylky při měření teploty pomocí AM2302[7]



Obrázek 1.9 Graf odchylky při relativní vlhkosti pomocí AM2302

1.3 Termočlánky [1]

Termočlánek využívá termoelektrického jevu. U tohoto jevu vzniká na rozhraní dvou různých kovů malé napětí, které závisí na teplotě (Seebeckův jev). Díky tomu lze termočlánek použít pro měření velkého rozsahu teplot s relativně velkou přesností. Bezespornou výhodou tohoto typu teplotního sensoru je jeho časová stálost.

Termočlánky se dělí podle kombinace kovů a nebo podle jejich kalibrace. Nejčastěji se můžeme setkat s kalibracemi, které se značí J, K, E, T (viz obrázek 1.8, kde je přehled jejich vlastností).

Označení termočlánku dle IEC 584	Původní označení	Měřicí rozsah [°C]
T	Cu-CuNi, Cu-ko	- 200 až 350
J	Fe-CuNi	- 200 až 750
E	NiCr-CuNi, ch-ko	- 100 až 900
K	Ni-Cr-Ni, ch-a	- 200 až 1200
N	NiCrSi-NiSi	- 200 až 1200
S	PtRh10-Pt	0 až 1600
R	PtRh13-Pt	0 až 1600
B	PtRh30-PtRh6	300 až 1700

Obrázek 1.10 Tabulka přehledu termočlánků [8]

2. ROZBOR IOT CLOUDŮ

Cloudové služby pro IoT jsou v dnešní době rychle se rozvíjející služby, jejichž cílem je logování a vizualizace dat. Každá služba je specifická. Především se liší v tom, jak můžeme data zobrazovat, kolik můžeme připojit čidel, jak často můžeme data do cloudu posílat nebo z jakých čidel můžeme vysílat.

Důležitým faktorem při výběru IoT cloudu je především to, pro jaké zařízení nebo pro jaký typ čidel je služba určena. Některé služby jsou vyhrazeny pouze pro originální Arduino zařízení. Existuje služba, která je určena především pro venkovní senzory. Kupodivu se v tomhle případě nejedná o nějaký nesmyslný zákaz, ale služba se snaží chránit samotné uživatele.

Společným průsečíkem většiny služeb je podpora MQTT protokolu, který se využívá pro odesílání dat do cloudové služby. Tento protokol je vysvětlen dále v této práci.

V této době existuje obrovské množství těchto služeb, seznam nejpoužívanějších služeb můžeme nalézt v Tabulka 2.1.

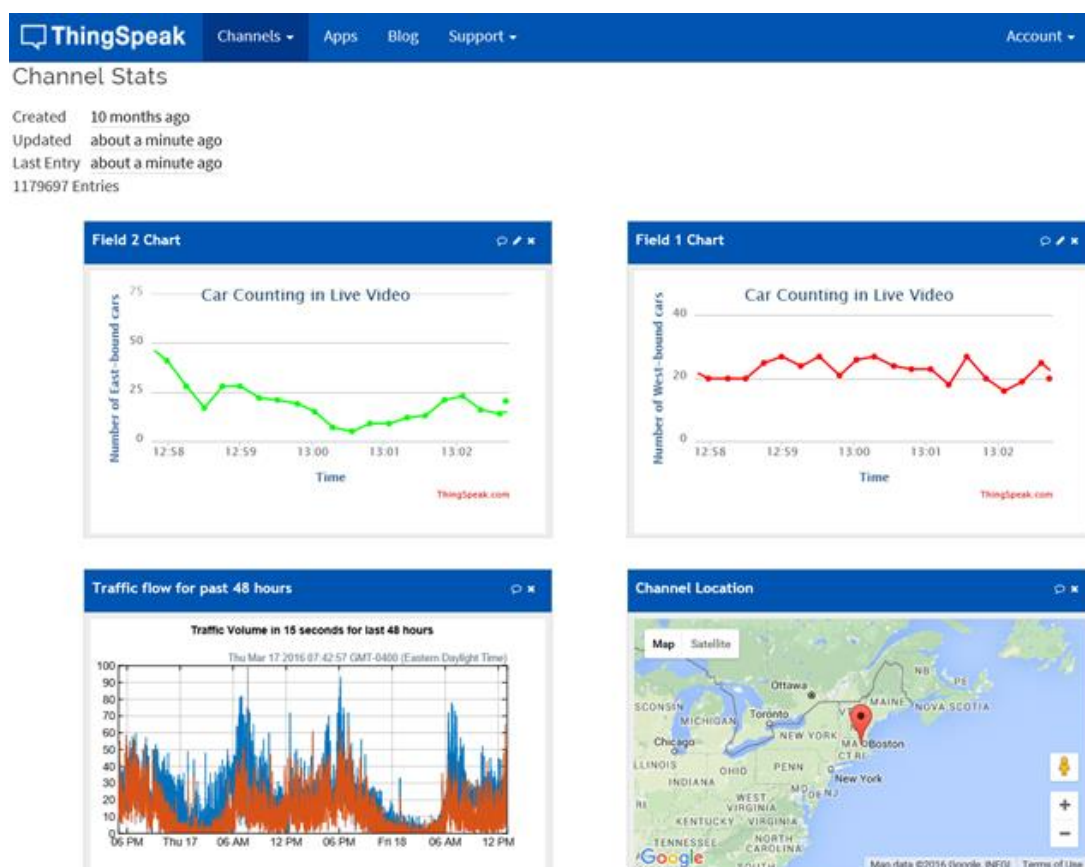
Tabulka 2.1 Tabulka nejpoužívanějších IoT cloudů

Název cloudu	Webová stránka projektu
ThingSpeak	https://thingspeak.com/
TMEP	https://tmep.cz/
IoT GURU cloud	https://iotguru.live/
ArduinoIoTcloud	https://login.arduino.cc/
openHAB	https://www.openhab.org/
IBMCloud	https://cloud.ibm.com/login
Cloud IoT Core	https://cloud.google.com/iot-core
Fogwing	https://www.fogwing.io/
Axonize	https://www.axonize.com/platform/
Blynk	https://blynk.io/
DATADOG	https://www.datadoghq.com/product/
Bevywise	https://www.bevywise.com/iot-platform/

2.1 ThingSpeak

Jedná se o online aplikaci, která je v poslední době velice rozšířená. K jejím přednostem patří především to, že ji lze využívat bezplatně, ovšem s určitými omezeními. Bezplatná verze nabízí zprávu maximálně 4 kanálů s celkovým maximálním počtem 3 milionů zpráv za rok. Dalším kritériem je limit intervalu aktualizace zpráv, a to 15 sekund. Data uložená v cloudu jsou omezena u bezplatných účtů na 10 milionů. Z toho vyplývá, že data v cloudu budou uložena něco málo přes 3 roky s minimálním intervalem aktualizace. [9].

Důvodem, proč je tato služba v poslední době velice rozšířená je, že kolem aplikace existuje docela velká komunita lidí, která vydává mnoho návodů, jak to používat. Obrovskou výhodou je možnost zpracovávat přijatá data pomocí matlabu, díky kterému, můžeme data přepočítávat, vyvozovat z nich nějaké závěry a zobrazovat nejruznější průběhy v grafech.



Obrázek 2.1 Ukázka z uživatelského rozhraní ThingSpeak [13]

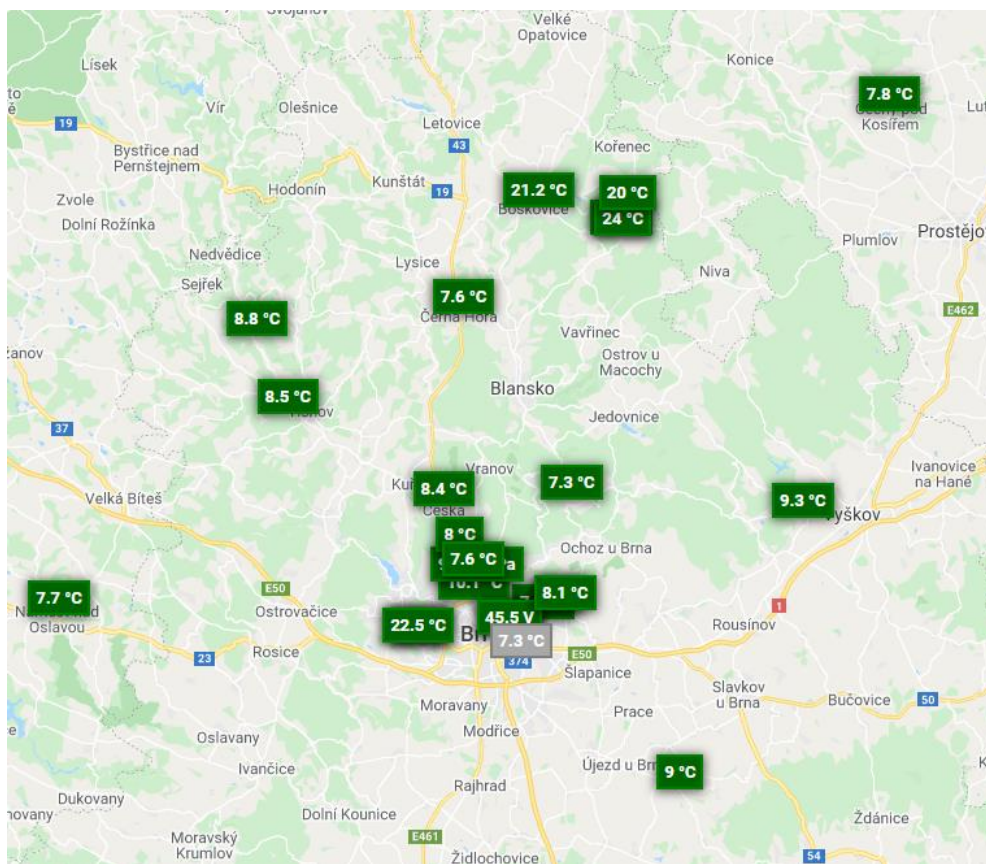
2.2 TMEP [10]

Cloudová služba TMEP je ryze českou záležitostí, protože v ní můžeme nalézt pouze venkovní čidla na území České republiky a blízkého okolí. Motivací této služby je shromažďovat meteorologická data z celého území ČR, proto je potřeba sdílet i GPS souřadnice čidla. Většina uživatelů vystavuje svoje čidla veřejně tak, aby se k jejím naměřeným datům mohl kdokoli dostat.

Provozování 3 čidel bez komerčního užití je zcela zdarma, dokonce údaje z čidel nemusí být veřejně vystaveny. Stačí, když u čidla nejsou uvedeny GPS souřadnice. Podmínkou pro zajištění správné činnosti je potřeba posílat data každou minutu, v nejhorším případě každých 5 minut. Zásadní informací je, že služba nepodporuje MQTT protokol. Místo toho jsou data odesílána pomocí HTTP GET dotazu na port 80.

Teplota není jediný parametr, který lze posílat. Často se teplota kombinuje s vlhkostí vzduchu nebo atmosférickým tlakem. Výjimkou není ani zasílání hodnoty napětí na baterii.

Další výhodou je, že zjednodušená verze celého projektu podporující jedno čidlo je vystavena jako open source a dá se stáhnout z GitHubu.



Obrázek 2.2 Mapa čidel vyexportovaná 6.12.2020 v 17:15 [10]

2.3 openHAB

Jak už je patrné z názvu, jedná se o open source projekt, který je primárně určený k automatizaci chytrých domů. Cloudová služba předpokládá provoz vlastního serveru v lokální síti nebo serveru ve veřejném internetu, která bude jenom pro účely daného chytrého domu. S vidinou připojení vyvíjeného termostatu by taková služba byla hodně velký a finančně nákladný luxus. Ovšem komunita z openHAB foundation provozuje i online službu myopenHAB, která je zdarma. [11]

MyopenHAB je sice především určen k prezentačním a testovacím účelům, přesto nezakazuje využití této aplikace pro provoz osobních čidel. Ale má to malé riziko. Služba negarantuje to, že bude stále fungovat nebo že se data po čase nesmažou. Použití je na vlastním uvážení, ale nesmíme zapomenout na omezení.

Tato služba je pro účely vyvíjeného termostatu nebo na testování během vývoje optimální. Hlavně pokud budeme chtít termostat využívat spíše nárazově, jednou za čas. Poslední dobou není problém použít malý deskový počítač např. raspberryPi, který lze v domácnosti používat jako server, na kterém by mohla běžet tato aplikace. V dnešní době je mnoho domů, které už takový malý domácí server mají a v budoucnu se zvyšujícím se zájmem o inteligentní domy bude takový server standard, proto by bylo dobré, aby byl termostat kompatibilní a snadno připojitelný i na tento cloudový systém. Podpora MQTT protokolu je u této aplikace samozřejmostí.

2.4 MQTT

Protokol můžeme najít pod označením ISO/IEC 20922 a jeho zkratka znamená „*Message Queuing Telemetry Transport*“, v překladu: „*zprávu dotazující telemetrický přenos*“. MQTT pracuje na sedmé aplikační vrstvě a využívá TCP/IP protokolu. Je navržen tak, aby byl lehký a moc nezatěžoval datový přenos. Hlavním cílem je publikování a přijímání zpráv mezi zařízeními. Vyznačuje se obousměrností, bezztrátovostí a neprohazováním pořadím zpráv. Díky tomu se skvěle hodí pro komunikaci mezi zařízeními stroj-stroj nebo k nasazení do IoT zařízení, protože nezatěžuje běh zařízení a nepotřebuje nebo nemá velkou šířku pásma, dokonce je odolný proti velkým latencím v síti.

V dnešní době existuje mnoho verzí MQTT, nejaktuálnější je MQTT version 5.0. I přesto je potřeba dodat, že protokol neřeší žádnou autentizaci a bezpečnost, proto je dobré ho zkombinovat ještě s TLS. Protokol dále podporuje QoS, díky kterému můžeme zprávě přiřadit prioritu podle toho, jak je důležitá. Priorita zpráv se dělí do tří kategorií: doručit maximálně jednou, doručit alespoň jednou a doručit pouze jednou. [12]

3. HARDWARE

Ze zadání projektu vyplývá, co všechno musí zařízení obsahovat a jaké parametry musí jednotlivé součástky splňovat. V tomto případě máme zadané, že zařízení má obsahovat ESP8266, což je mikrokontroler, který má v sobě implementované wifi rozhraní, kterého máme využít ke komunikaci se zařízením a pomocí kterého mají být odesílána data do IoT cloudu. Celé schéma zapojení je v přílohách tohoto dokumentu.

Termostat by neplnil svoji funkci, kdyby neuměl měřit teplotu, zařízení proto obsahuje konektor, do kterého můžeme připojit jedno z výše popsanych digitálních čidel. Pro univerzálnost je na desce osazen konektor RJ13 (samice 4P4C), do kterého můžeme zasunout konektor RJ10 (samec 4P4C). Z toho vyplývá, že k desce bude teplotní čidlo připojeno pomocí kabelu, kde na jedné straně bude čidlo a na druhém konci konektor RJ10.

Důležitým blokem toho zařízení je externí I2C EEPROM, na kterou jsou ukládány aktuální nastavení a konfigurace, především informace o wifi sítích, ke kterým se má zařízení připojovat. Nastavení síťového rozhraní v režimu AP a klíče pro odesílání dat na IoT cloud.

Dalším funkčním blokem je převodník USB to UART, pomocí kterého je možné zařízení naprogramovat a ladit při vývoji Firmware. Jedná se obvod s převodníkem CH340G a s pomocnými součástkami, díky kterým celý blok správně pracuje.

Při návrhu desky plošných spojů se objevilo mnoho problémů, to vedlo k několika verzím DPS, ve kterých byly postupně odstraňovány chyby. V posledních verzích došlo k rozdělení desky na dvě části, zdrojovou a výpočetní. Tento koncept se zachoval i v konečné verzi. Hlavní motivací k tak radikální změně byla implementace bezpečného, zato rozměrově většího zdroje MYRRA 47162. Rozdělení na dvě části přineslo i zvýšení bezpečnosti zařízení, protože umožnila dostat do zařízení druhé relé, díky kterému zařízení spíná všechny pracovní vodiče. To zajistí že na spínané zásuvce se ve vypnutém stavu neobjeví fázové napětí.

Nesmíme zapomenout na fázi oživení zařízení, která je nedílnou součástí vývoje každého zařízení. V této fázi jsou většinou odhaleny nejzásadnější chyby, kterých jsme se dopustili při návrhu. V mém případě tomu bylo jinak. V drtivé většině se jednalo o chybějící cesty, špatně zvolená pouzdra součástek, anebo nevhodně zvolené hodnoty odporů. Odhalování těchto chyb je zdlouhavý a vyčerpávající proces. Mnohdy zabere mnohem více času než samotný návrh. Přesto je často opomenut, přestože je nedílnou součástí každého vývoje ať hardwareového tak softwarového.

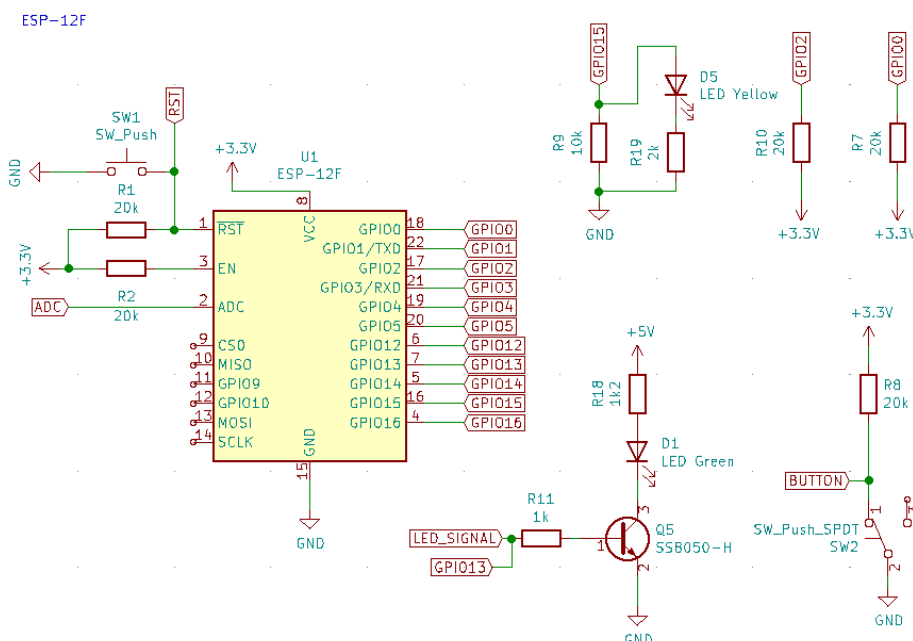
3.1 Mikrokontrolér a jeho periferie

Návrh tohoto bloku je především zaměřen na periferie modulu ESP8266. Hlavním úkolem této části je zajistit potřebné napěťové úrovně na pinech GPIO15, GPIO0 a GPIO2, aby modul mohl nabootovat z interní SPI flash paměti nebo provést nahrání nového firmwaru, viz tabulka 3.1. Připojení pinu na určitou logickou úroveň (H = 3.3 V) a (L = 0 V), musí být provedeno pomocí odporů, které omezují proud.

Mezi další komponenty patří resetovací tlačítko, které při stisknutí přemostí rezistor R1 a přivede log 0 na pin RST, čímž resetuje ESP zařízení. Dále jsou k MCU připojeny dvě kontrolní LED diody. První LED dioda zelené barvy je spínaná pomocí tranzistoru a druhá žlutá je připojená přímo na GPIO15 pin, který je možné po startu zařízení použít jako výstupní pin. Poslední součástka, kterou zbývá popsat je přepínač, pomocí kterého je možné přepínat režimy zařízení. Popiska BUTTON dále je ve schématu připojena k GPIO16. V předchozích verzích DPS byla k přepínači připojena kontrolní LED, která indikovala, v jakém režimu se zařízení nachází. Od této myšlenky bylo upuštěno, protože byla potřeba ovládat tuto diodu softwarově, aby bylo možno rozpoznávat více stavů než dva. Díky této změně teď můžeme rozpoznávat stavy vypnuto – blikající - zapnuto.

Tabulka 3.1 Tabulka potřebných napěťových úrovní pro ESP8266

GPIO15	GPIO0	GPIO2	Mód
L	H	H	Boot z SPI flash paměti
L	L	H	Programování pomocí UART



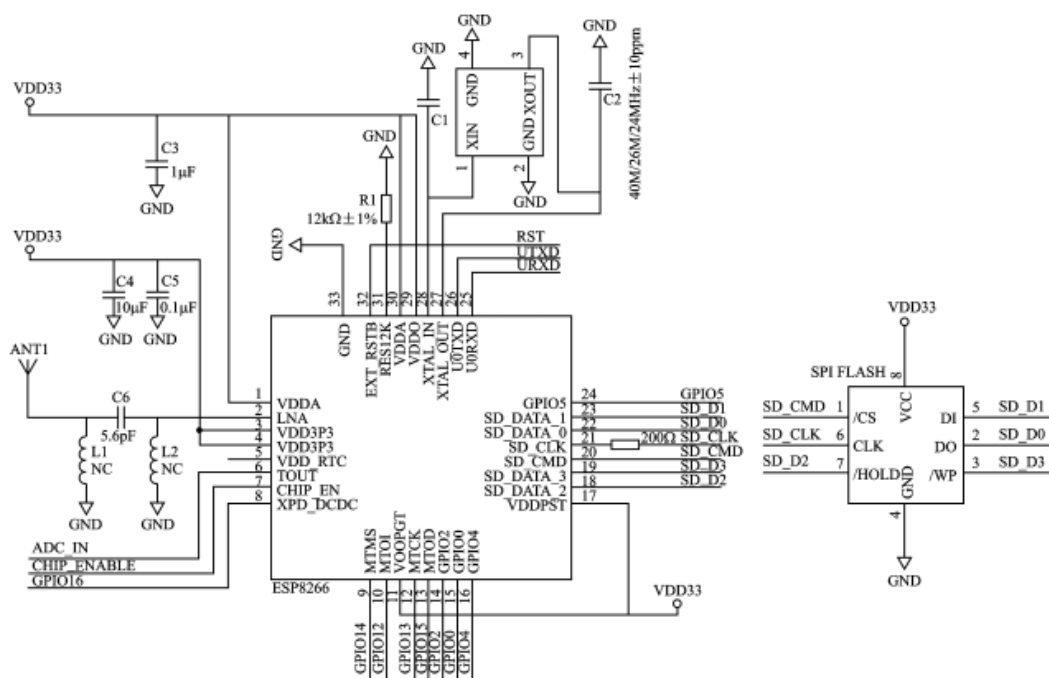
Obrázek 3.1 Blokové schéma zapojení MCU a jeho periférií.

3.1.1 ESP8266

Jedná se o relativně malou destičku, kterou je možné použít místo mikrokontroleru. V dnešní době existuje mnoho variant a provedení od ESP-01 po ESP-12. Rozdíl mezi jednotlivými typy je především v počtu vyvedených pinů a velikosti interní EEPROM paměti, do které se ukládá program. Dalšími parametry jsou rozměry modulu, uspořádání pinů, tvar a provedení antény.

Provedení ESP-12 se dále dělí na ESP-12E a ESP-12F, jediným rozdílem v těchto verzích je velikost a tvar antény. Modul verze 12F má parametrově lepší integrovanou anténu, z toho důvodu byl pro daný projekt zvolen právě tento modul. Dalším důvodem je velká EEPROM paměť, která má velikost 4 MB, dokonce jsou na pouzdro modulu vyvedeny piny SPI sběrnice na kterou, je připojena více zmíněná interní EEPROM paměť. V neposlední řadě je zpětně kompatibilní s ESP-07. ESP-07 má jednu obrovskou výhodu, a to je možnost připojení externí antény. To v projektu umožní, že v případě malého pokrytí můžeme pouze vyměnit desku ESP modulu a přidat externí anténu bez nutnosti návrhu celé nové DPS.

ESP obecně má velmi malé nároky na spotřebu proudu, k běhu potřebuje pouze 80 mA a díky tomu je vhodný pro nasazení do aplikací, kde je použito napájení z baterií. Nevýhodou je spotřeba při startu, při kterém může ESP potřebovat až 500 mA. Pokud by při startu nebyl dodán dostatečný proud, který čip potřebuje, může se stát, že nenaběhne správně. Mohlo by se pak stát, že se bude chovat náhodně nebo nebudou fungovat některé periferie. Z tohoto důvodu je v projektu použit dostatečně výkonný zdroj, který je schopen dodávat až 700 mA dlouhodobě s krátkodobým zatížením až na 1 A.



3.2 Ovládání spínané zásuvky

Na obrázku 3.3 je vidět zapojení obvodu pro spínání dvou relé. Každé relé je připojeno na zdroj 5 V a spínáno tranzistorem NPN MMBT5551L. Každý z těchto tranzistorů je spínám přes optočlen EL817, který je zde použit jen jako galvanický oddělovač. Optočleny jsou ovládány vstupním signálem označeným jako RELE ten je připojen přímo na pin ESP modulu. Dalšími součástkami, které jsou v obvodu použity jsou ochranné diody, díky kterým při sepnutí/rozepnutí relátka dochází k demagnetizaci

cívky relé a tím chrání spínací tranzistory proti napět'ovým špičkám. Další důležitý prvek je LED dioda, která indikuje, jestli je relé sepnuto. Při návrhu DPS byla tato indikační LED dioda přesunuta ze zdrojové části do výpočetní, a to z důvodu, aby byly všechny kontrolní LED diody na jednom místě. Přesto ve schématu zůstala umístěna zde, a to hlavně kvůli tomu, aby bylo jasné vidět, jakým způsobem je stav LED diody vyčítám.

Bezpečnost tohoto bloku je nejvyšší prioritou, proto v klidovém stavu (bez úrovně 3,3 V na signálu RELE) jsou relátka vypnuta. Díky tomu se v případě poruchy se na výstupní zásuvce neobjeví síťové napětí a dojde k odstavení ovládaného tepelného spotřebiče. Dalším bezpečnostním prvkem je použití dvou relátek pro spínání všech pracovních vodičů. (díky tomu vypneme vždy i fázový vodič).

Volba tranzistoru byla provedena následovně: napětí napájecího zdroje U_{VCC} je 5 V, proto je potřeba hledat tranzistor, který snese alespoň trojnásobek tohoto napětí, tedy 15 V, což splňuje skoro každý tranzistor. Dalším důležitým parametrem je maximální proud spínaný tranzistorem. Ten je vypočten v rovnici 3.1. $R_{cívky}$ je katalogová hodnota pro zvolené relé a značí odpor cívky.

$$I_{c, tranz} = \frac{U_{VCC}}{R_{cívky}} = \frac{5}{62} = 81 \text{ mA} \quad (3.1)$$

Vypočtená hodnota je 81 mA, ale stejně jako u pracovního napětí je potřeba tuto hodnotu ztrojnásobit. Je hledán tranzistor s hodnotou maximálního proudu alespoň 250 mA. Proud procházející tranzistorem může být chvilkově vyšší než teoreticky spočtený, například při zapnutí nebo vypnutí.

Výpočet potřebného proudu pro sepnutí relé je uveden v rovnici 3.2, kde β_{tranz} je zesílení tranzistoru a U_{VCC} je 5 V. Poslední hodnota $R_{cívky}$ je katalogová hodnota pro zvolené relé a značí odpor cívky.

$$I_{b, tranz} = \frac{I_{c, tranz}}{\beta_{tranz}} = \frac{U_{VCC}}{\beta_{tranz} \cdot R_{cívky}} = \frac{5}{80 \cdot 62} = 1 \text{ mA} \quad (3.2)$$

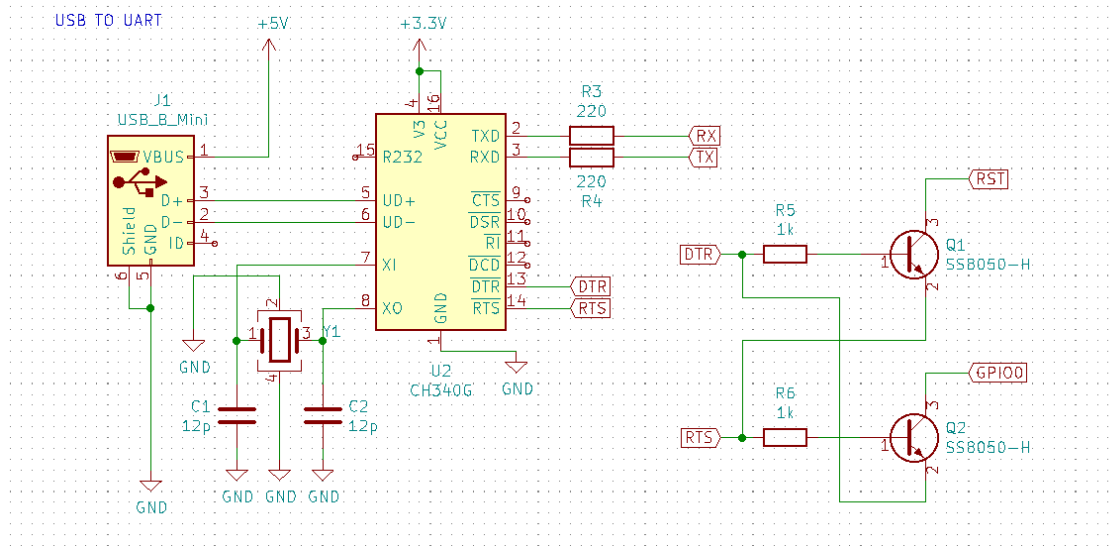
Z předchozího výpočtu (rovnice 3.2) jasné plyne, že tranzistor je možné spínat přímo pinem na mikrokontroleru, kde je pro ESP8266 povolen proud 12 mA. Přesto je zde ještě začleněn optočlen, proto je hodnota rezistorů R12 a R13 určena experimentálně na 1 K Ω .

Hodnota odporu R15 u LED diody je vypočtena pomocí rovnice 3.3, kde U_{VCC} je napájecí napětí 5 V, U_d a I_d je pracovní bod diody.

$$R_{15} = \frac{U_{VCC} - U_d}{I_d} = \frac{5 - 1,7}{0,007} = 470 \text{ } \Omega \quad (3.3)$$

3.3 Převodník z USB na UART

Další důležitou částí návrhu je převodník z USB na UART viz Obrázek 3.4. Jeho hlavním prvkem je integrovaný obvod CH340G, který se využívá i v mnoha jiných vývojových deskách s ESP [15]. Jeho výhodou je to, že z jeho pouzdra jsou vyvedeny i ovládací piny pro RS232. Dva z těchto pinů se využívají jako řídicí signály pro přepnutí ESP do režimu bootu nebo běhu. Díky tomu je možné mikrokontroler programovat bez nutnosti nějaké jiné akce na desce, jako například přepnutí tlačítka nebo přidání propojky.



Obrázek 3.4 Schéma zapojení převodníku z USB na UART.

Důležitou součástí tohoto schématu je klopný obvod díky, který má na starosti ovládat logickou hodnotu na GPIO0 pinu, tak aby v klidové stavu byl tento pin připojen na Log. 1 a případně nahrávání nového frimwaru vyresetovat MCU a nastavit na pin GPIO0 Log. 0. viz tabulka 3.1.

Integrovaný obvod CH340G nemůže fungovat bez externího 12 MHz krystalu, který celý systém taktuje. Pro napájení IO je použito napětí 3,3 V a to hlavně kvůli kompatibilitě UART protokolu s ESP modulem tak, aby oba pracovali na stejné napěťové hladině. Pokud by to tak nebylo, musel by se použít pomocný obvod, který by převáděl 5 V UART na 3,3 V a obráceně. K tomuto účelu by se dala použít Zenerova dioda pro snížení napěťové úrovně.

Mini USB zde bylo použito hlavně kvůli jeho robustnosti, a protože se snadněji osazuje na DPS než USB Micro. Jeho nevýhodou je jeho velikost, na desce zabírá poměrně hodně místa. Tato nevýhoda se projevila v prvních návrzích desek, ovšem od verze s více DPS není tato nevýhoda tak zásadní.

3.4 I2C EEPROM paměť

Další součástí, kterou obvod obsahuje je EEPROM paměť. V původním návrhu byla použita M24512, která má 512 kbit. V konečné verzi byla nahrazena pamětí 24LC16B s 8x255 bajtů, a protože obě paměti fungují koncepčně jinak, bylo potřeba drobných úprav.

Obě paměti jsou připojeny pomocí I2C sběrnice. Paměť je zamýšlena jako úložiště konfiguračních dat, jako například jméno a heslo na wifi síť.

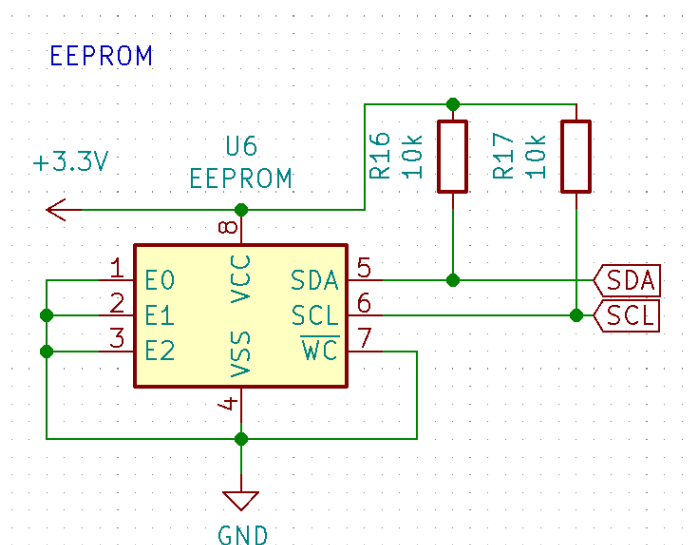
Na obrázku 3.4 je vidět piny E0, E1 a E2. Těmito piny je možné u běžné paměti přesněji nastavit poslední 3 bity v adrese zařízení. V bakalářské práci jsou všechny tyto piny nastaveny na hodnotu logické 0, neboli jsou připojeny na GND. Navíc u paměti 24LC16B jsou takto nastavené hodnoty ignorovány, protože je členěna na 8 bloku po 255 bajtech. Výběr bloku je realizován v adrese. Z toho vyplývá, že EEPROM se tváří jako 8 EEPROM zařízení na jedné sběrnici, a to bez ohledu na to, jak jsou zapojeny piny E0-2. [a2]

3.4.1 I2C

Jedná se o datovou sběrnici, využívající pro obousměrný přenos dat pouze dva vodiče, SDA a SCL. Sběrnice byla vyvinuta firmou Philips v roce 1982 [a1], dnes pod firmou NXP Semiconductors. Jedná se o chráněnou značku, proto se můžeme setkat i s označením TWI, což je téměř identická sběrnice jako I2C.

Vodič s označením SCL přenáší hodinový signál s frekvencí 100 kHz nebo 400 kHz a vodič SDA přenáší data. Na sběrnici je v ideálním případě možné umístit až 128 zařízení. Pro přenos je potřeba stanovit, které zařízení bude master a které jsou slave.

Pro zajištění funkčnosti sběrnice je potřeba připojit oba vodiče SCL a SDA pomocí rezistorů R16 a R17 k napájení, viz obrázek 3.5.



Obrázek 3.5 Schéma zapojení EEPROM paměti.

3.5 Výroba a návrh DPS

Před samotným návrhem DPS je potřeba vybrat krabičku. Pro tento projekt byla vybrána krabička od firmy KRADEX s výrobním označením Z27J, která už obsahuje dva konektory, jeden typu samec, druhý typu samice, oba pro síťové napětí 230 V, které se používají na území České republiky.

Po výběru krabičky bylo přikročeno k návrhu samotné desky. Rozměry desky byly přímo přizpůsobeny tak, aby se vlezla do vybrané krabičky a taky, aby seděly montážní otvory na předpřipravené plošky.

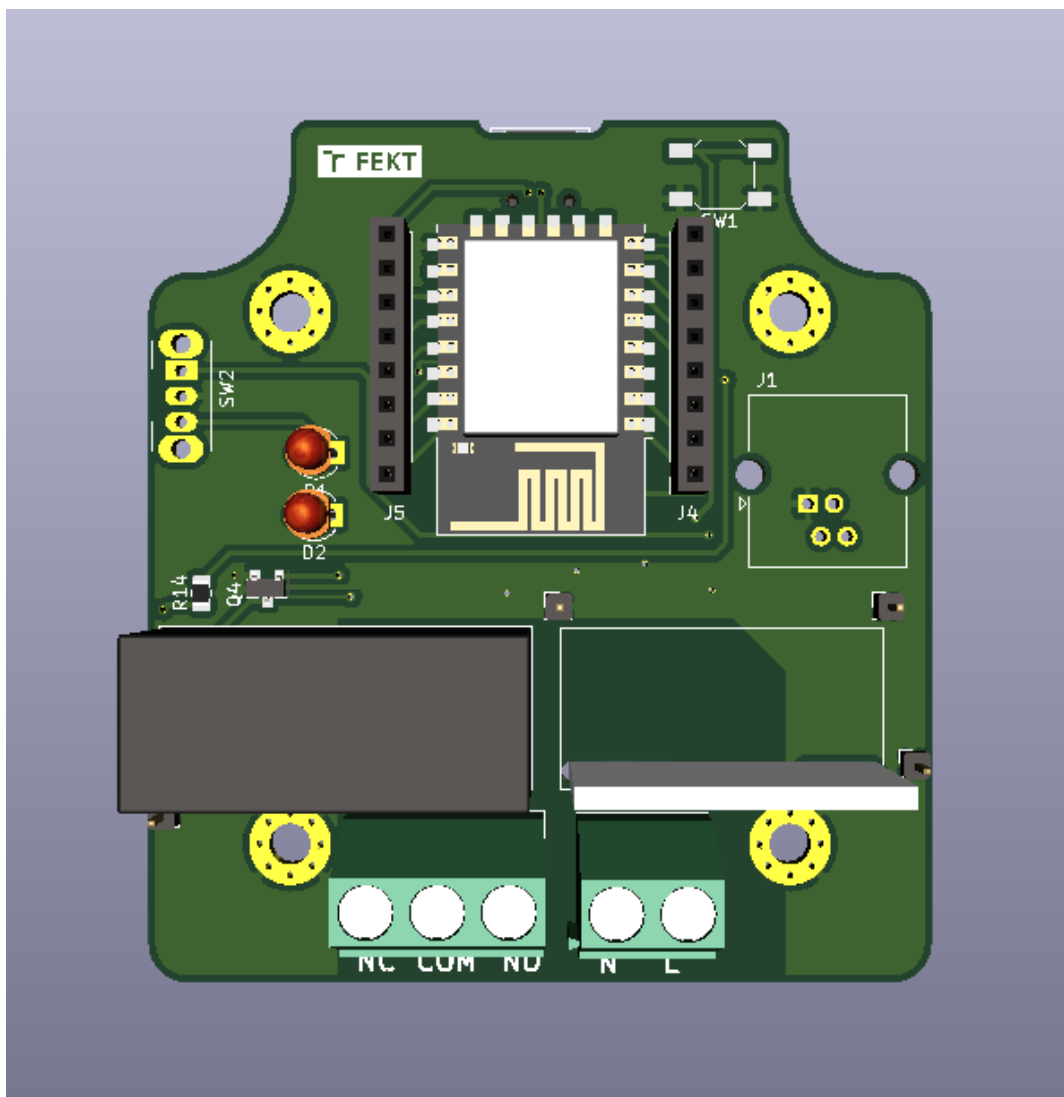
Návrh desky a kreslení schémat bylo provedeno v programu KiCAD, který byl zvolen hlavně kvůli tomu, že umožňuje exportování 3D modelu navržené desky. Dalšími výhodami programu je velká knihovna součástek a jejich snadná editace. Celý program je distribuován jako open source. Díky tomu existuje velká komunita lidí, kteří jsou ochotni poradit s jakýmkoli problémem, který je potřeba v programu vyřešit.

Návrh počítá s použitím SMD a vývodových součástek, a to hlavně kvůli tomu, aby se ušetřilo místo na desce, rozměry desky jsou limitovány rozměrem krabičky. U některých prvků, především u relé, přepínače, zdrojové desky a konektorů není možné použít povrchovou montáž. U signalizačních LED diod je také uvažována vývodová montáž a to z důvodu, aby se LED diody mohly namontovat do krytu krabičky a nemusely být řešeny světlovody, které by světlo přenášely z desky na povrch krabičky.

Do projektu také byla přidána patice na připojení dalších prvků. Pro tuto patici byla zvolena rozteč a uspořádání pinů, jako na desce Wemos [15]. Díky tomu při vývoji není nutné osazovat desku přímo ESP modulem, ale může se využít toho, že se na tuto patici připojí deska Wemos. Bohužel z důvodu rozdělení zařízení na dvě desky už není možné namontovat výpočetní desku s Wemos zařízením, ale je potřeba DPS osadit ESP modulem, jinak se zařízení nevleze do krabičky.

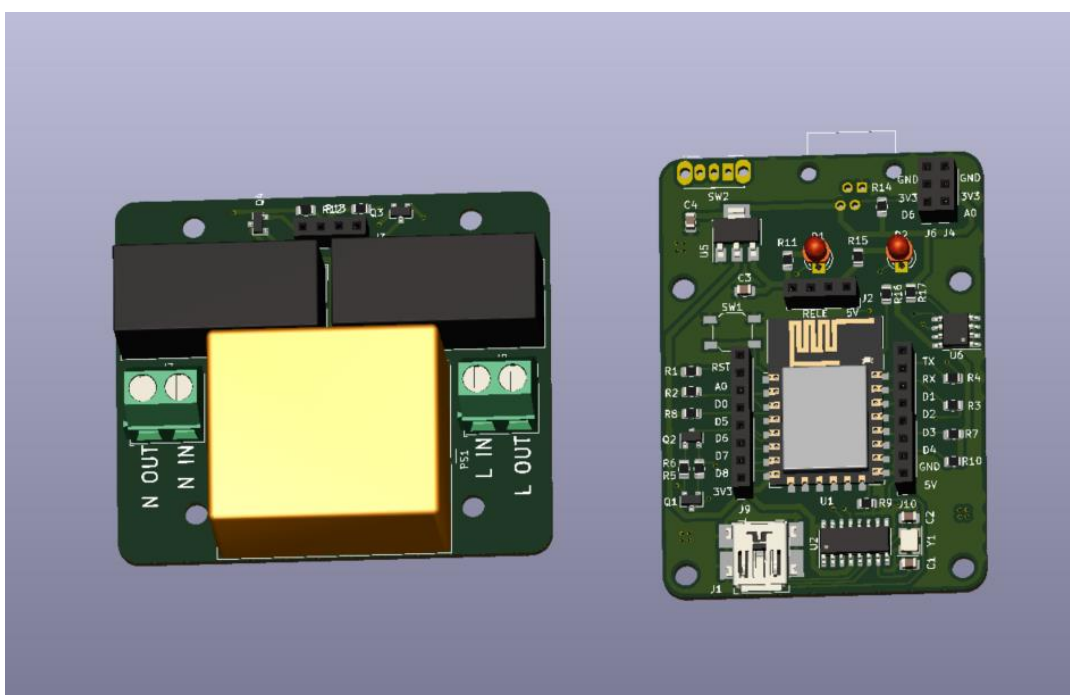
Postupný vývoj zařízení vyústil ve tři rozdílné verze zařízení, první verze (černá) je navržena pouze na jedné desce. Důvodem k opuštění této verze byla nevhodná volba zdroje, který nevyhovoval evropským standardům. Nesplňoval bezpečnostní odstupy síťového napětí na PDS zdroje, navíc zdroj podstatným způsobem rušil ESP, což vedlo ke snížení citlivosti přijímaného wifi signálu. Další verze byla doplněna o MYRRA zdroj, kvůli kterému bylo potřeba zařízení rozdělit na dvě DPS. Díky tomu v zařízení vznikl dostatek místa na to, aby se do něj mohlo umístit ještě jedno relé. Tato verze byla mnohonásobně bezpečnější než první verze. Protože při návrhu druhé verze jsem zapomněl nakreslit jednu cestu, musel jsem přistoupit k návrhu třetí verze, která obsahuje pouze novou výpočetní DPS. Spodní zdrojová DPS byla zachována z verze 2. Pro snadnější montáž a vycentrování desky byl potřeba navrhnout speciální distanční blok, díky kterému je možné desky spojit a namontovat do krabičky. Tento distanční blok byl

navržen s programu SOLIDWORKS a následně vytisknut na 3D tiskárně z transparentního PLA s výškou vrstvy 0,2 mm.

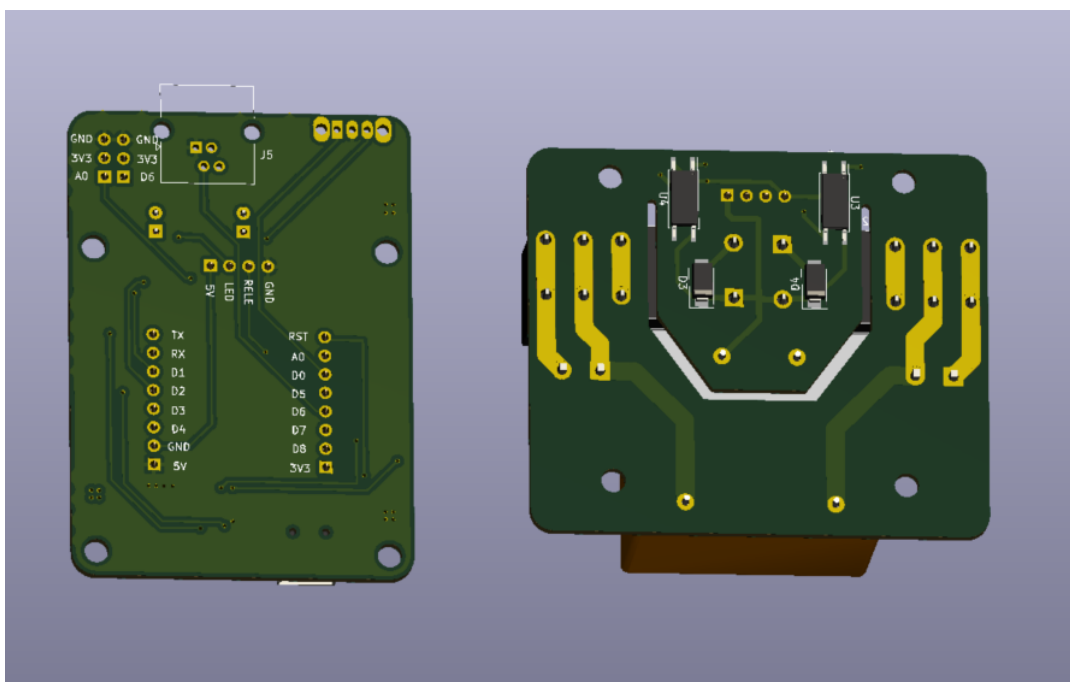


Obrázek 3.6 3D model navržené desky verze 1, export KiCad.

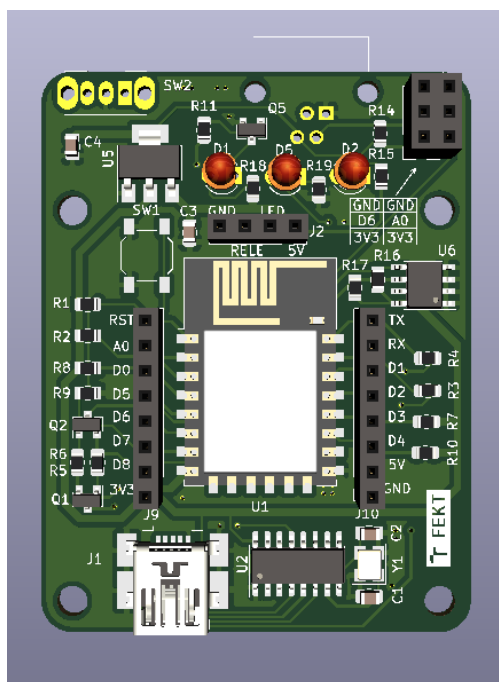
Výroba DPS byla provedena společností JLCPCB [16]. Jedná se o zahraniční firmu, která se zabývá prototypovou výrobou DPS. Stačí desku vyexportovat ve formátu GERBER a nahrát ji do jejich portálu a zaplatit. Výroba 5 desek stojí \$2, co se týče poštovného, tak to je dražší, ale výroba touto formou je stále levnější a kvalitnější než u tuzemských firem. Doba, než je deska vyrobena, je zpravidla 3 dny, u dopravy garantují, že nejpozději do 10 dnů deska dojde na vámi zadanou adresu.



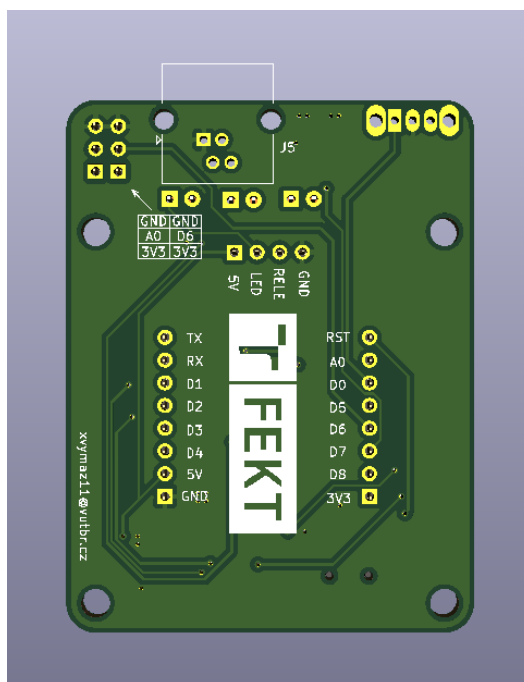
Obrázek 3.7 3D modely DPS verze 2 přední pohled, export KiCad.



Obrázek 3.8 3D modely DPS verze 2 zadní pohled, export KiCad.



Obrázek 3.9 3D model DPS verze 3 přední pohled, export KiCad.



Obrázek 3.10 3D model DPS verze 3 zadní pohled, export KiCad.

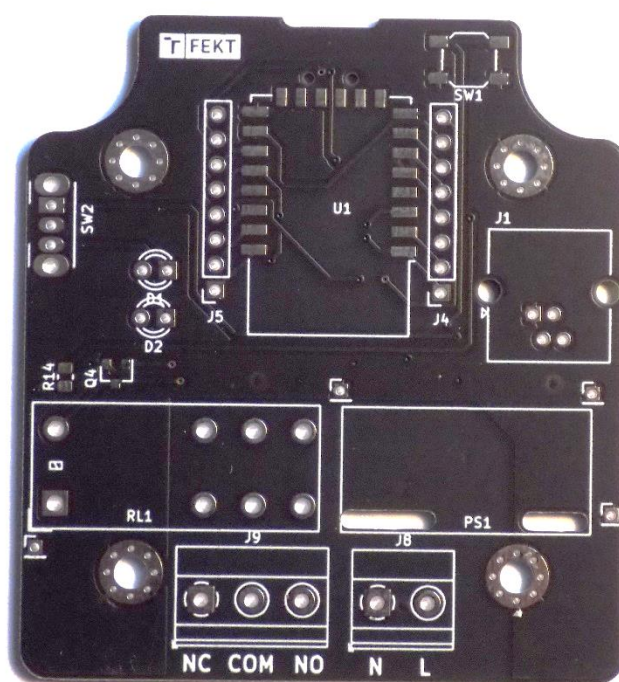
Zásadním rozdílem mezi verzí 2 a 3 je doplnění chybějící cesty, otočení pinů pro připojení čidel (vpravo nahoře na přední straně). Další úpravou je softwarové ovládání LED diod (zelená, žlutá), dodání popisek a log (změna popisu vývodů Wemos, prohození 5 V a GND) a v posledním případě kompletní změna tras, obzvláště u I2C.

3.6 Osazení a oživení DPS

K osazení byl použit bezolovnatý cín. Hlavní motivací k použití bezolovnaté pájky byla směrnice Evropského parlamentu a Rady 2011/65/EU, která zakazuje používání olovnatých pájek v elektrozařízeních, kromě výjimek pro zdravotnictví, kosmický průmysl a vojenské účely [17].

Dalšími nástroji použitými k osazení DPS byla mikropájka a horkovzdušná pistole na osazení krystalu u obvodu USB to UART převodníku.

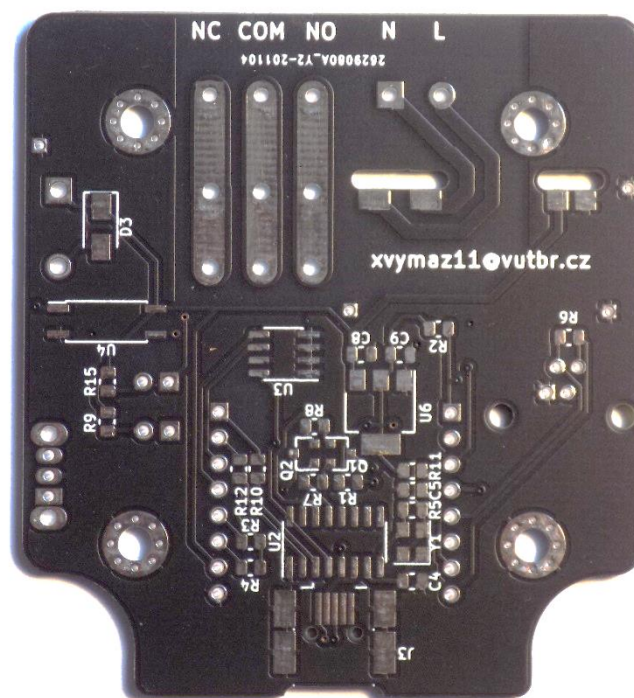
3.6.1 Deska zařízení verze 1



Obrázek 3.11 Vyrobená deska verze 1 - přední strana

Na obrázku 3.12 je pohled na zadní stranu desky. Zde jsou připraveny kontakty na připojení relé a svorkovnice, na které bude připojena spínaná zásuvka s proudem 16 A. Z norem pro návrh desek se dá vyčíst, že při tloušťce mědi 35 μm by požadovaná šířka musela mít více jak 2,5 mm. Z toho důvodu se uvažuje přidání pomocného spojení pomocí cínu, a to po celé délce cest.

Na spodní části desky si také můžeme povšimnout dvou montážních otvorů pro chycení zdroje, který byl hlavním důvodem proč tato verze desky nakonec nebyla použita.



Obrázek 3.12 Vyrobená deska verze 1 - zadní strana

3.6.2 Oživení zařízení verze 1

Desku jsem osazoval postupně a jednotlivé periferie jsem rovnou testoval. Díky této metodě jsem jednoduše mohl nalézat chyby. První chyba, na kterou jsem narazil, se projevovala tak, že deska komunikovala s počítačem, ale nešel do mikrokontroleru nahrát frimware. Chybu se mi povedlo odhalit, byla způsobena použitím špatného pouzdra u tranzistorů Q1 a Q2, které mají za úkol při nahrávání programu přepnout ESP do režimu bootu.

Další periferie byly také odzkoušeny. Komunikace s EEPROM paměť fungovala bez problémů, stejně tak spínání relé. Bohužel komunikace mezi DS18B20 ze začátku nechtěla fungovat. Tato chyba byla způsobena špatnou verzí knihovny.

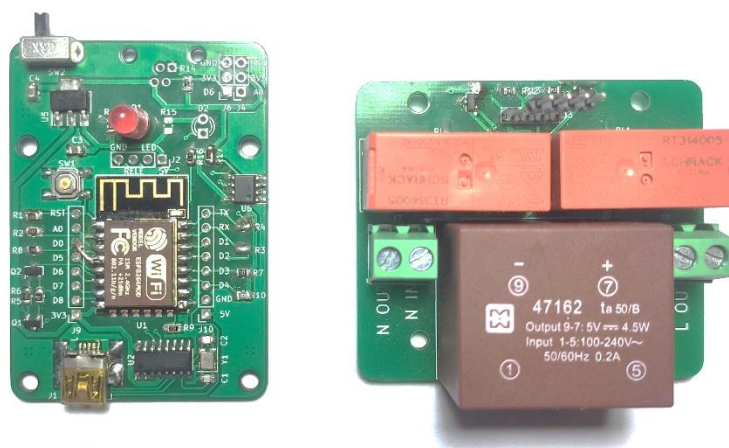
V dnešní době, bohužel nemám k dispozici osazenou desku s verzi 1 tu jedinou, kterou jsem měl jsem musel rozebrat, abych součástky mohl použít na další verze. Přesto jsem objevil několik fotek, na kterých ji můžeme vidět, ale z důvodu horší kvality jsem je přiložil do elektronické přílohy.

3.6.3 Deska zařízení verze 2

Součástí návrhu zařízení není zdroj. Ve verzi 2 a 3 je použit již zpracovaný zdroj určený k osazení na DPS. Jedná se o zdroj od firmy MYRRA s označením 47162 s výstupní napětí 5 V DC, s maximálním výkonem 4,5 W, s ochranami proti přehřátí, proti přetížení a proti zkratu. Ten byl hlavním důvodem pro návrh nové verze zařízení, které je rozděleno na dvě jednotlivé desky, na zdrojovou a výpočetní desku.

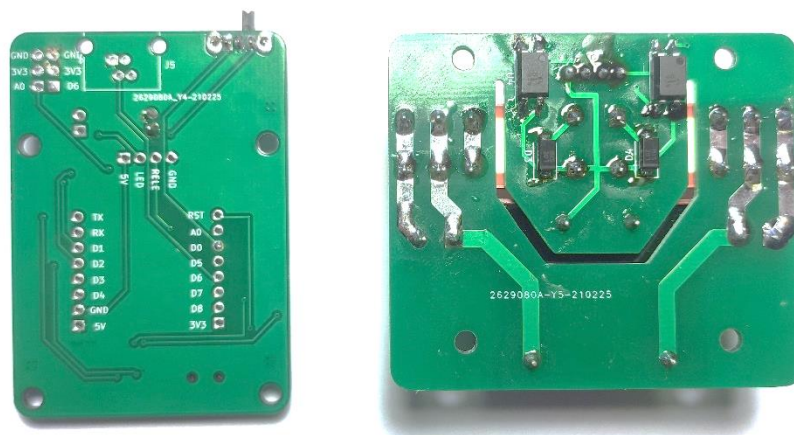
Zdrojová deska obsahuje výše zmíněný MYRRA zdroj, dvě relé a ovládací periferie jako jsou optočleny, tranzistory a pomocné rezistory. Pro spojení s horní výpočetní deskou je použit hřebíkový konektor se čtyři piny (GND, RELE, LED, 5 V). GND a 5 V je napájení pro řídicí – výpočetní desku. Signálem RELE je možné ovládat spínání relátek na zdrojové desce (3,3 V zapnuto a 0 V vypnuto). Signál LED je uzemněn, pokud jsou relé sepnuty. Díky tomu může spínat červenou LED diodu na výpočetní desce, která indikuje sepnutí výstupní zásuvky.

Výpočetní deska obsahuje zbylé komponenty: modul ESP, EEPROM, stabilizátor, USB převodník a kontrolní LED diody. Desky jsou spojeny pomocí konektorů a speciálního distančního sloupku, viz obrázek 3.15



Obrázek 3.13 Vyrobená a osazená deska verze 2 – přední strana

Na obrázku 3.13 je pohled na přední stranu desek verze 2, vlevo se nachází výpočetní deska, na které je osazena pouze jedna signalizační LED dioda a chybějící cesta je improvizovaně nahrazena drátkem (levá strana modulu ESP). Vpravo se nachází zdrojová deska, která je stejná i pro verzi 3. Na zdrojové desce můžeme vidět MYRRA zdroj (velká hnědá krabička), dvě relátka (dva oranžové objekty) a dva konektory pro připojení síťové napájení.



Obrázek 3.14 Vyrobená a osazená deska verze 2 – zadní strana

3.6.4 Oživení zařízení verze 2

Při ožiování zdrojové desky jsem nenarazil na žádný problém vše fungovalo hned po osazení. Jedním z problémů byla chybějící cesta od přepínače do MCU, kterou jsem nahradil, propojkou. Další problémem byly LED diody, které sice svítily ale vůbec nic neříkaly o tom, v jakém stavu se nachází program. Při špatném wifi signálu se může stát, že ESP není schopné se připojit do sítě, z toho důvodu je potřeba indikovat nejen do kterého režimu je výrobek přepnut, ale jestli se ESP připojilo. Při poruše čidla nebo při rozbití poškození kabelu čidla je potřeba, aby ESP hned signalizovalo poruchový stav. Proto byla další verze o tyto indikační LED diody rozšířena.

V neposlední řadě jsem se zaměřil na externí EEPROM paměť (M24512), kterou jsem následně nahradil 24LC16B. Důvod, proč korektně nefungovala EEPROM paměť M24512 přikládám tomu, že se pravděpodobně jednalo o vadnou dávku. Ze tří pamětí, které jsem měl, se každá chovala jinak. Jedna nefungovala hned od začátku, další fungovala chvíli a pak přestala zapisovat a poslední nechtěla zapisovat na určitá místa. Prvotním plánem byla myšlenka, že by veškeré textové proměnné (texty na webu) byly taktéž uloženy v EEPROM paměti a za běhu programu by se dal editovat obsah webu. Tato myšlenka se později ukázala jako zbytečná.

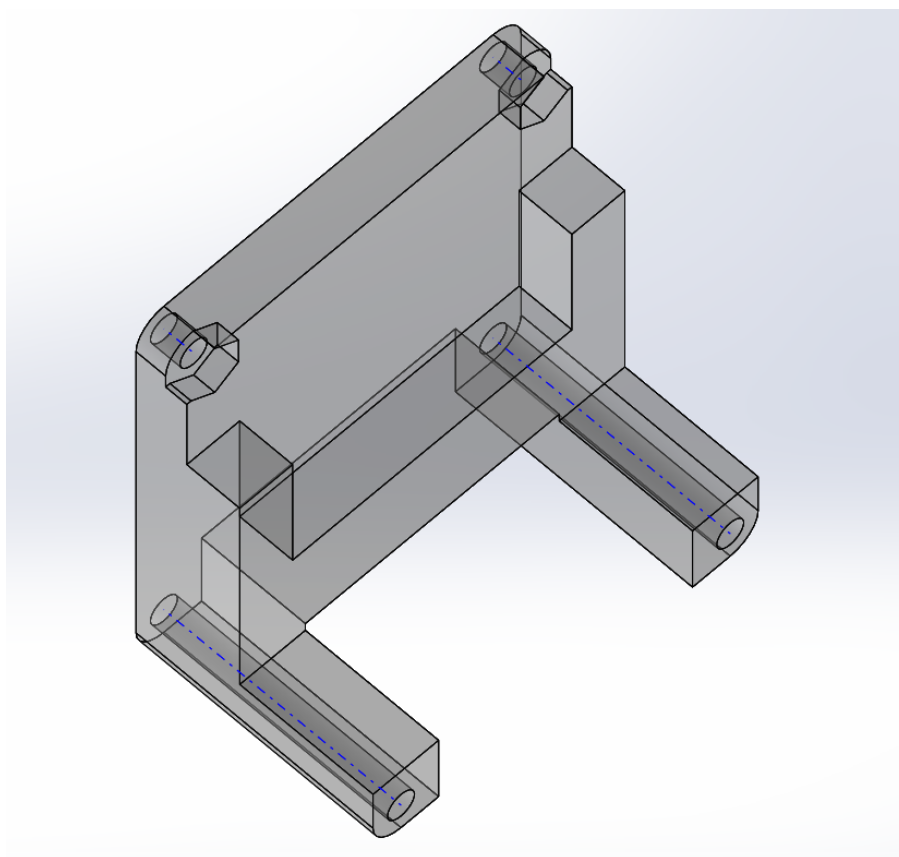
Další spíše kosmetickou chybu jsem našel na popisku Wemos rozhraní, kde byly prohozeny popisky na 5 V a GND. V další verzi jsou popisky už správně. Ne zcela zjevnou změnu prodělali i piny pro připojení digitálního a analogového čidla (vpravo nahoře při čelním pohledu). Prohozením pořadí těchto výstupů jsem umožnil elegantnější připojení DS18B20 v pouzdře TO92.

Nejzávažnější problémem, proč nemohla být použita výpočetní deska verze 2, byla její velikost. Při pokusu dostat celé zařízení do krabičky jsem zjistil, že deska přesahuje

asi o 3,5 mm mimo prostor krabičky. Nejjednodušším a nejelegantnějším řešením bylo vytvořit novou verzi.

Při té příležitosti jsem vytvořil speciální distanční sloupek, pomocí kterého jsem mohl desky k sobě uchytit a tím zjistit o kolik musím novou desku zkrátit. Tenhle sloupek se mi tak osvědčil, že jsem se rozhodl ho zakomponovat do celého výroku, aby byly desky pevně přichyceny k sobě.

K výrobě sloupku byl použit materiál PLA, což není nejvhodnější volba. Mnohem vhodnějším materiálem by bylo například ABS, které je časově stálejší. Protože při použití toho distančního sloupku by mohlo dojít k akumulaci tepla, především na relátkách, nechal jsem jednu stranu co nejvíce odkrytou, aby zde mohlo docházet k úniku tepla do okolí. Jestli je tohle tahle metoda dostačující nemohu řídit, bylo by potřeba zařízení podrobit náročné zkoušce, při které by se sledovalo nárůst tepla za jednotku času. Naštěstí hned vedle relátek je umístěn zdroj, který by se také od nich zahříval. Tenhle zdroj je vybaven tepelnou ochranou, takže by se při zahřátí na 50 °C měl odstavit čímž by došlo i k vypnutí relátek a zařízení by tak mělo čas zchladnout, dokud by zdroj zase nezačal fungovat.

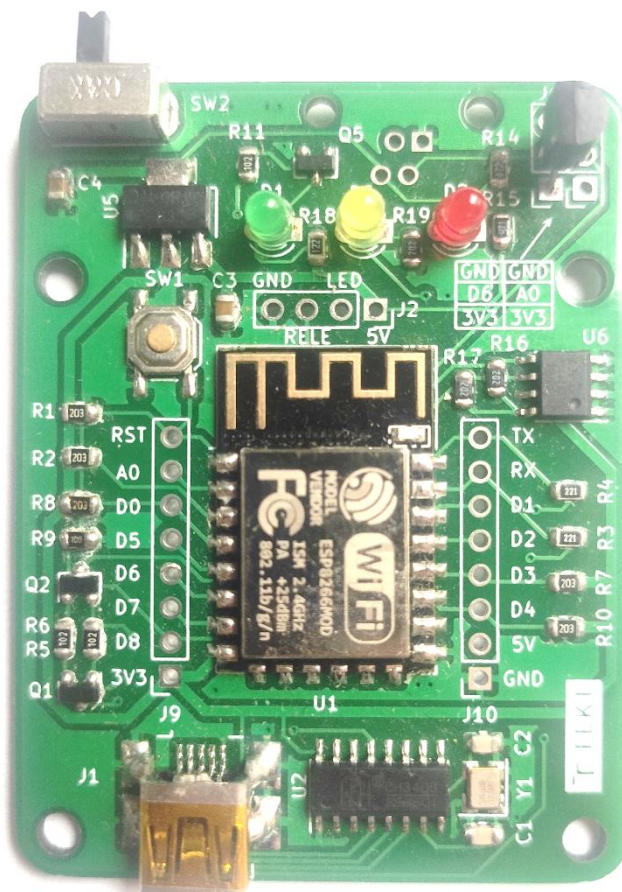


Obrázek 3.15 Speciální distanční sloupek pro propojení desek.

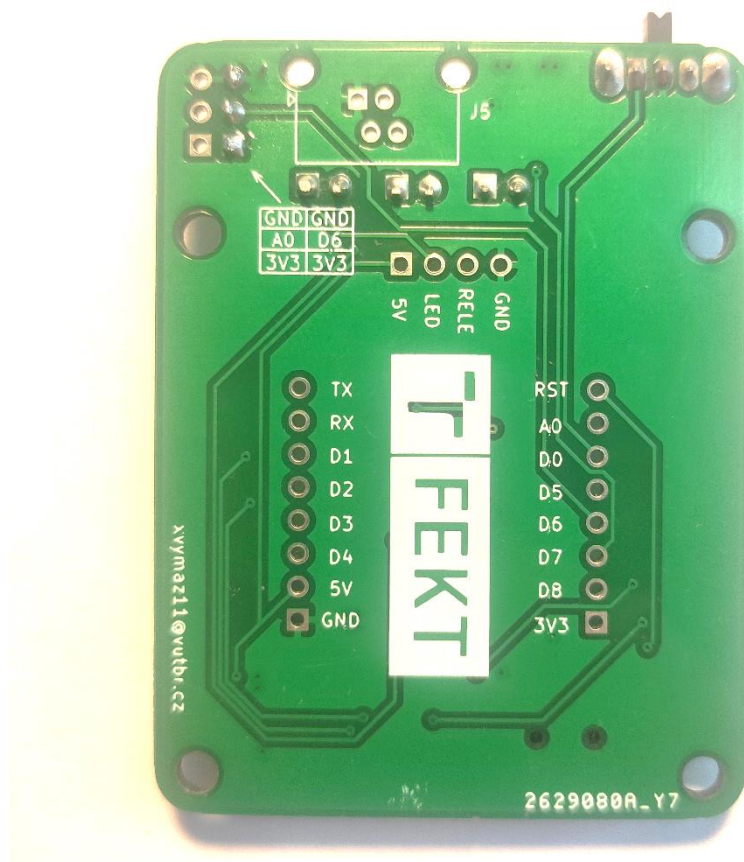
3.6.5 Deska zařízení verze 3

V poslední (3. verzi) se povedlo, odstranit veškeré, problémy, na které jsem narazil při testování předchozích verzí. Při rychlém pohledu se může zdát, že se tato poslední verze se příliš neliší od svého předchůdce, ale opak je pravdou. Jediné, co bylo zachováno je přibližné rozmístění součástek. Občas bylo potřeba součástku pootočit nebo nepatrně posunout. Čeho si každý na první pohled hned všimne je rozdílný počet LED diod, ta zásadnější změna je tom, že zelenou se žlutou diodu je možné ovládat softwarově.

Na obrázku 3.16 můžeme vidět již osazenou desku verze 3, kde místo RJ konektoru pro připojení čidla je Dallas přímo připájen na vývody vpravo nahoře. To je z důvodu, aby deska bylo skladnější při následném vývoji Frimwaru. V budoucnu je také možnost zde připojit další interní čidlo, které by mohlo sledovat teplotu uvnitř krabičky. Hned vedle jsou další pájecí body, kde je možnost připojit analogové teplotní čidlo.



Obrázek 3.16 Vyrobená a osazená výpočetní deska verze 3. – přední strana.



Obrázek 3.17 Vyrobená a osazená výpočetní deska verze 3. – zadní strana.

3.6.6 Oživení zařízení verze 3.

Při ožívování finální verze desky se objevil pouze jeden problém, který se povedlo poměrně rychle vyřešit. Při snaze co nejméně snížit spotřebu zařízení jsem se rozhodl, použít místo 20 k Ω rezistorů 200 k Ω , bohužel tato změna způsobila problém, ESP nechtěl nastartovat a donekonečna se restartoval. Při vrácení 20 k Ω rezistorů tato závada zmizela.

U této verze desky také bylo potřeba správně navrhnout rezistory u LED diod. Protože jsem použil diody, které jsem měl doma, bylo potřeba stanovit jejich pracovní bod. K tomu jsem využil regulovatelný zdroj a ampérmetr. Diodu jsem zapojil sériově k ampérmetru a tuto sestavu jsem připojil k regulovatelnému zdroji. Postupně jsem zvyšoval napětí na zdroji, dokud se mi měřená dioda nerozsvítila s požadovanou intenzitou. Pak jsem si poznamenal napětí na zdroji a proud protékající diodou. Tuto metodu jsem opakoval pro všechny barvy použitých LED diod. Naměřená data jsou uvedena v následující tabulce 3.2.

Tabulka 3.2 Tabulka potřebných napěťových úrovní pro ESP8266

Barva LED diody	Ubytek na diodě [V]	Proud diodou [mA]	Referenční napětí [V]
červená	1,7	7	5
žlutá	1,8	0,7	3,3
zelená	3,2	1,5	5

Výpočet potřebné hodnoty rezistoru jsem provedl podle následujícího vzorce 3.5. Ve kterém referenční napětí U_r je napěťová úroveň, ze které bude LED dioda napájena. R značí hodnotu hledaného rezistoru, I_d a U_d značí pracovní bod diody.

$$R_{barva} = \frac{U_r - U_d}{I_d} \quad (3.4)$$

Postupně jsem provedl výpočty pro všechny použité barvy. Protože rezistory jsou vyráběny pouze v určitých hodnotových řadách je nutné získaný výsledek převést na hodnotu rezistoru, která je běžně vyráběna. Díky tomu sice dojde k posunu pracovního bodu, ale tenhle posun bude zanedbatelný.

$$R_{červená} = \frac{U_r - U_d}{I_d} = \frac{5 - 1,7}{0,007} = 471,43 \, \Omega \quad (3.5)$$

$$R_{žlutá} = \frac{U_r - U_d}{I_d} = \frac{3,3 - 1,8}{0,0007} = 2142,86 \, \Omega \quad (3.6)$$

$$R_{zelená} = \frac{U_r - U_d}{I_d} = \frac{5 - 3,2}{0,0012} = 1200,00 \, \Omega \quad (3.7)$$

Z vypočtených hodnot jsem vydrál pro červenou diodu hodnotu $470 \, \Omega$, pro žlutou $2 \, k\Omega$ a pro zelenou $1200 \, \Omega$. Po osazení a otestování vše bez problému fungovalo.

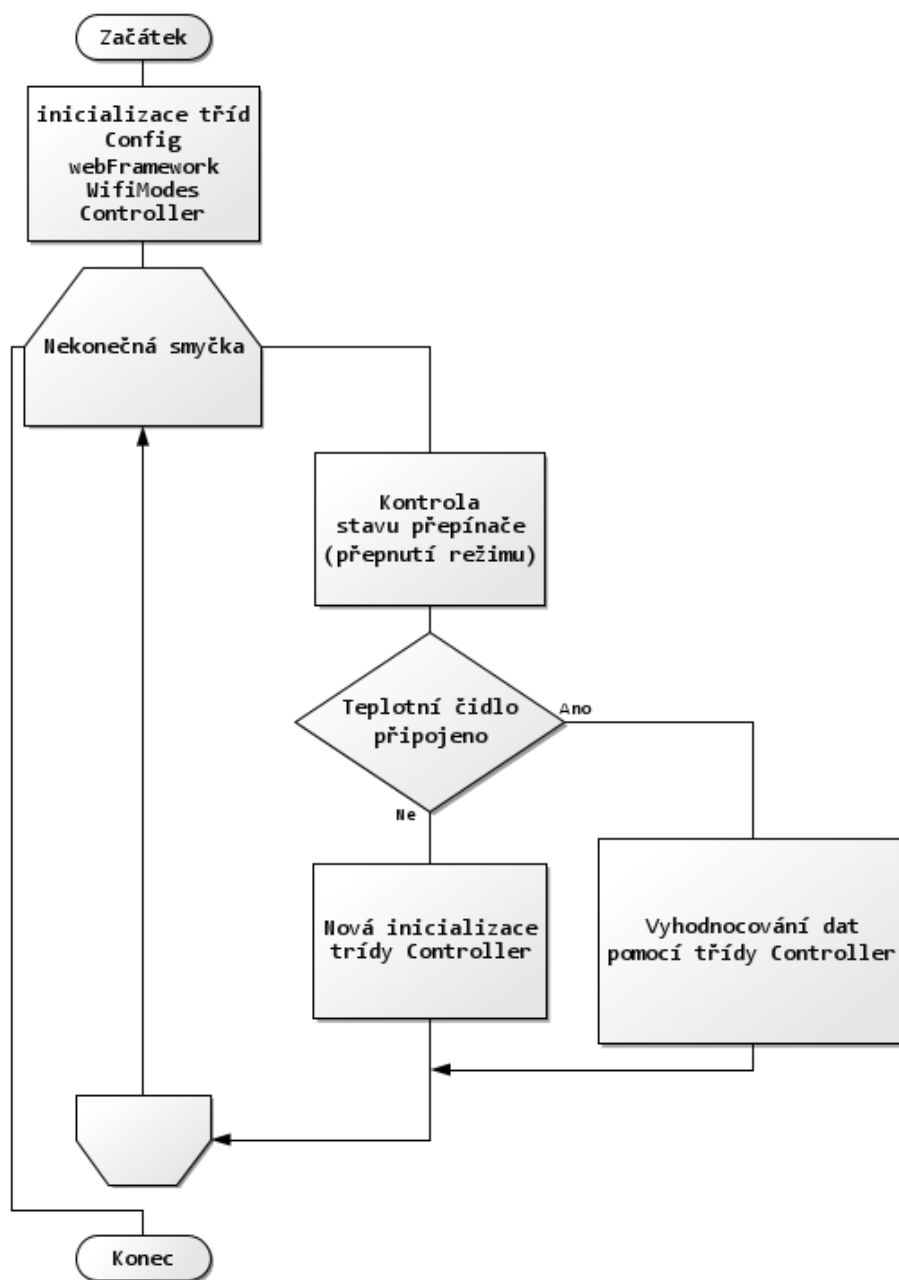
4. FRIMWARE

Pro projekt byl použit programovací jazyk C/C++. Pro usnadnění práce bylo použito Arduino studio, obzvláště při kompilaci a uploadu frimwaru. Kód byl napsán v jiném textovém editoru, Arduino studio není nejvhodnější textový editor pro vývoj větších projektů. Jako externí editor byl využit program Atom.

Sofwarová část projektu byla rozdělena na dílčí části aby, byl přehlednější a snadno editovatelný. Snahou také bylo využít výhod objektové programování. Projekt je členěn do funkčních bloků (tříd), každá třída má za úkol jinou specifickou činnost.

Požadavky na tento projekt byly, regulace teploty, ukládání naměřených dat na IoT cloud, možnost přepínat v režimech Wifi AP a Wifi Client a možnost ovládat o konfigurovat zařízení pomocí webového rozhraní. Z toho důvodu byl projekt rozdělen na následující třídy.

Základní strukturu frimwaru můžeme vidět na následujícím obrázku 4.1. Na kterém jsou vidět dva základní bloky ze kterých se skládá většina frimwaru. Jedná se o část kódu, která se vykoná pouze jednou v main souboru ji můžeme najít pod označením setup. Druhá část kódu pojmenována loop, je vykonávána stále dokola dokud nedojde k vytnutí nebo resetování zařízení.



Obrázek 4.1 Vyrobená a osazená výpočetní deska verze 3. – zadní strana.

4.1 Použité knihovny

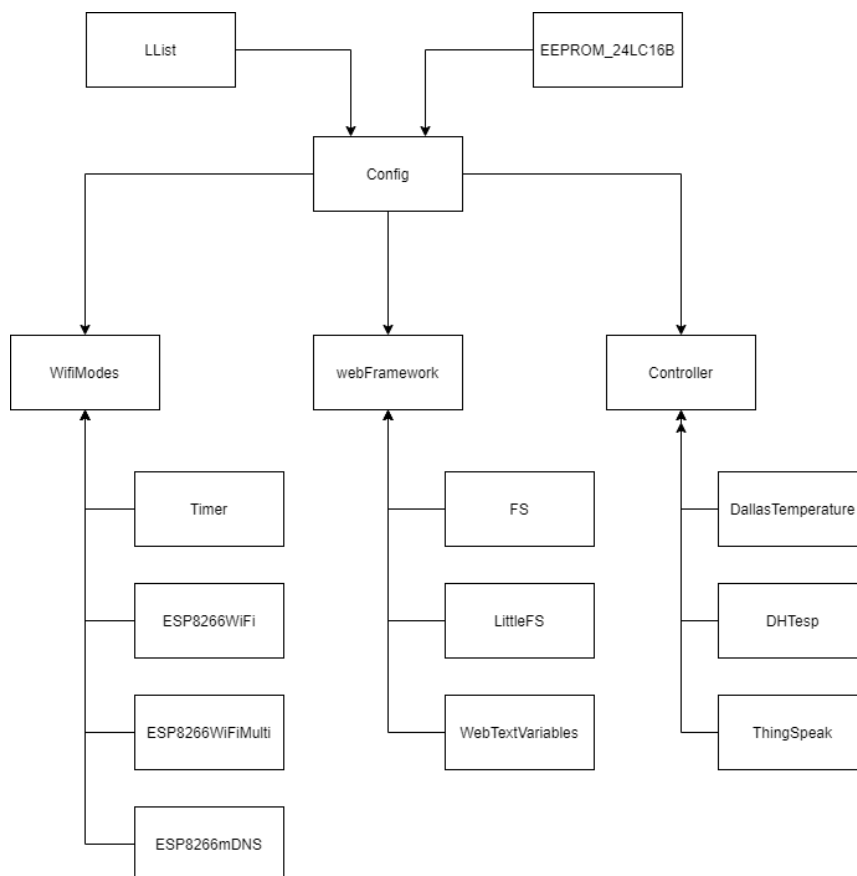
- **Arduino.h** – jedná se o základní knihovnu pro arduino core, obsahuje základní definice jednoduchých funkcí jako je digitalRead nebo digitalWrite. Dokonce obsahuje definice jmen pinů. Tato knihovna sice automaticky přidávána ale pouze do main souboru, pokud potřebuje napsat knihovnu pracující s jejími funkcemi musíme ji přidat.
- **ESP8266WiFi.h** – knihovna, která umožňuje přístup k wifi periférii. Především přes ní nastavujeme síťové rozhraní a nastavuje režim v jakém má zařízení fungovat. Díky této knihovně je možné i získávat data o okolních wifi sítích.
- **ESP8266WiFiMulti.h** – nadstavba nad ESP8266WiFi.h, díky této knihovně je možné nastavit zařízení seznam wifi sítí na který se má připojit. Knihovna se připojí k první možné wifi síti ze seznamu.
- **WiFiClient.h** – pomocí této knihovny je možné realizovat TCP/IP spojení. V projektu je především použita jako podpora pro ThingSpeak.h knihovnu, díky které je možné ukládat data na IoT Cloud.
- **ESP8266mDNS.h** – tato knihovna realizuje multicast DNS server, díky němuž je možné zařízení nalézt v místní síti pomocí jeho mDNS jména. V reálných sítích a na některých zařízeních, tenhle protokol bohužel nefunguje. Přesto byla snaha ho implementovat.
- **Wire.h** – knihovna implementující I2C/TWI komunikaci, která je potřeba pro komunikaci s EEPROM pamětí.
- **OneWire.h** – implementuje 1-Wire sběrnici, jedná se o upravenou I2C sběrnici. Díky této knihovně je možné komunikovat s DS18B20 čidlem.
- **stdint.h** – jedná se o standardní C knihovnu, v projektu je především použita pro import datových typů jako je například uint8_t atd.
- **stdarg.h** – opět se jedná o standardní C knihovnu, díky ní je možné vytvořit funkce s proměnným počtem parametrů.
- **DallasTemperature.h** – knihovna realizující interface mezi uživatelem a OneWire knihovnou při komunikaci s teplotním čidlem DS18B20. Knihovna především obsahuje funkce pro vyčítání hodnot a převod mezi °C a °F.
- **ThingSpeak.h** – jedná se o rozhraní, které ulehčuje komunikaci se stejnojmenným IoT Cloudem.
- **ESPAsyncTCP.h** – knihovna pro asynchronní TCP dotazy, je využívána především níže popsaným Webovým serverem, který ji využívá pro svoji práci.

- **ESPAsyncWebServer.h** – knihovna realizující Webový server, nejedná se zrovna o nejrozšířenější knihovnu realizující webový server na ESP2866, ale knihovna má velice podrobně zpracovanou technickou dokumentaci se spoustu ukázek. Obrovskou výhodou toho serveru je jeho funkcionalita a jednoduchost.
- **FS.h** – umožňuje zprávu a přístup do virtuálního souborového systému, který je možné na ESP vytvořit a provozovat. V tohle souborovém systému jsou například uloženy HTML šablony, CSS soubory a veškeré JavaScript funkce, která využívá webové rozhraní.
- **LittleFS.h** – rozšiřuje funkcionalitu FS knihovny

4.2 Vytvořené knihovny

- **webFramework.h** – knihovna starající se o načítání a webové stránky, zasílání JSON data a odchytávání GET a POST dotazů.
- **ProjectConfig.h** – soubor ve kterém jsou uloženy globální proměnné potřebné pro sestavení projektu, například definice jmen pinů a deklarace funkcí usnadňují tisk do konzole.
- **Config.h** – knihovna ve které jsou uloženy veškeré proměnné potřebné pro běh zařízení, uložení wifi sítě, síťová konfigurace, mDNS jméno atd.
- **WifiModes.h** – stará se nastavení wifi rozhraní, připojuje se do sítě nebo vytváří vlastní Wifi síť, a to v závislosti na tom do jaké pozice je přepnut přepínač na DPS. V této knihovně je také implementován mDNS server.
- **LList.h** – knihovna realizující spojovou strukturu.
- **EEPROM_24LC16B.h** – implementuje rozhraní pro práci s EEPROM pamětí.
- **Controller.h** – knihovna zodpovědná za vyčítání teploty a rozhodnutí, jestli má být vstupní zásuvka sepnuta. Dále zodpovídá za odesílání dat do IoT Cloudu.
- **Buffer.h** – pomocná knihovna umožňující tisk do bufferu, především pro zjednodušení výpisu dat na konzoli.
- **Timer.h** – umožňuje časovat akce, které je potřeba spustit jednou za dlouhý časový úsek.
- **WebTextVariables.h** – soubor ve které je uloženo pole ve kterém jsou definovány veškeré textové proměnné, které jsou vypisovány na webové stránce.

4.3 Použité knihovny



Obrázek 4.2 Vyrobená a osazená výpočetní deska verze 3. – zadní strana.

5. ROZHRAŇÍ ZAŘÍZENÍ

5.1 Webové rozhraní

Pro konfiguraci a vizualizaci naměřených data je možné použít webové rozhraní. Jedná se dvě jednoduché webové stránky, na první z nich je jsou zobrazena měřená data a také je zde možné nastavit parametry potřebné pro regulaci. Jako jsou například porádovaná teplota, hystereze regulace a možnost kompletně odstavit regulaci. Na této stejné stránce je v režimu klient zobrazován graf s naměřenými hodnotami. Další zajímavostí je grafická vizualizace signalizačních LED diod, které nám mohou usnadnit určit v jakém stavu se zařízení právě nachází.

Na další webové stránce je možné provádět konfiguraci síťového rozhraní, konfiguraci spojení na IoT cloud nebo zprávu wifi sítí ke kterým se může ESP připojovat. Pro uložení takto nastavených dat je potřeba změnu konfigurace uložit na EEPROM. Toho se docítí tlačítkem, které se také nachází na této stránce viz následující obrázek.

5.2 Terminál

5.3 Signalizační diody

Posledním a zároveň nejdůležitějším rozhraním jsou signalizační LED diody, které jsou umístěny na DPS a pomocí otvorů v krabici vyvedeny až ven. Zařízení obsahuje 3 diody s následujícími barvami, zelená, žlutá a červená. Podle chování těchto diod můžeme detekovat poruchové stavy, nebo jednoduše určit v jakém režimu se zařízení nachází, nebo jestli se připojilo do místní sítě.

Červená LED dioda indikuje, jestli jsou sepnuta výstupní relé neboli jestli je na výstupní zásuvku přivedeno 230 V. Volba červené diody nebyla náhodná, volba barvy byla inspirována barevným označením jističe v sepnutém stavu. Na rozdíl od jističe došlo k malé změně při indikování vypnutého stavu. V tohle případě je indikován tak, že červená led nesvítí.

Žlutá LED dioda oznamuje poruchový stav, nebo start zařízení. Při startu zařízení v režimu client se jedná o první diodu, která se musí rozsvítit následně po ní se rozbliká i zelená dioda. Blikání diody se nemusí projevit, protože frekvence blikání je 0,5 Hz. Následně poté by měla zhasnou žlutá dioda a zelená zůstane svítit, to značí že se zařízení připojilo do místní sítě a identifikovalo teplotní čidlo. V případě, kdy by zůstali svítit obě diody po delší čas než dvě sekundy, můžeme usuzovat, že se zařízení připojilo do sítě, ale nebylo schopno detekovat teplotní čidlo.

Zelená dioda indikuje stav sítě a režim zařízení ve kterém se nachází. Tato dioda, jak už bylo zmíněno výše, se v režimu klient rozbliká se zároveň rozsvícenou žlutou diodou. Režim AP je indikován blikáním zelené diody (frekvence 0,5 Hz) bez toho, aby se rozsvítila i žlutá dioda. Jednotlivé stavy ledek a k nim posaný popis stavu, je uveden v následující tabulce

Tabulka 5.1 Tabulka potřebných napěťových úrovní pro ESP8266

Barva LED diody	Ubytek na diodě [V]	Proud diodou [mA]	Referenční napětí [V]
červená	1,7	7	5
žlutá	1,8	0,7	3,3
zelená	3,2	1,5	5

6. ZÁVĚR

Semestrální práce se v úvodu zabývá způsoby měření teploty. Ta je zde hodnocena z pohledu použitelnosti pro vyvíjený výrobek. Tato teorie nepřináší nic nového, spíše shrnuje celou problematiku a hledá vhodného adepta, který by se dal použít k měření teploty. Zde byly nakonec vybrány digitální čidla ds18b20 a DHT22.

V další části se práce zaměřuje na rozbor několika IoT cloudů, shrnuje jejich klady a zápory. Z této části by měla být vybrána nějaká služba, která bude následně použita pro ukládání dat ze zařízení. Svázání zařízení s konkrétní službou není dobrý nápad, protože není garantováno, jak dlouho budou tyto cloudy nabízeny. Na trhu se také může objevit nová, která bude vhodnější než ta, se kterou by zařízení bylo svázáno. Z toho důvodu je snaha, aby termostat byl univerzálnější a byl schopen migrovat ze služby na službu, proto je zde nastíněn protokol MQTT, který tyto služby podporují a s velkou pravděpodobností budou podporovat i v budoucnu.

V poslední části se práce věnuje samotnému vývoji desky, kterou se povedlo vyrobit a částečně osadit. Testováním periferií byly objeveny určité problémy. Ty se ale dají snadno vyřešit, jako například zrcadlově otočené tranzistory, které je potřeba při pájení převrátit, aby deska fungovala správně.

Práce nepřináší myšlenku elektrického termostatu, ale rozvíjí možnost připojení a regulace takového zařízení pomocí bezdrátového spojení. Většina dnešních podobných termostatů tuto možnost nemá a pro jejich regulaci k nim člověk musí přijít a ručně je přenastavit.

Budoucností tohoto projektu je naprogramování řídicího mikrokontroleru a vývoj potřebných podpůrných programů, které budou potřeba pro získání plnohodnotného funkčního zařízení. Projekt by dále mohl být rozšířen možností regulace bez použití bezdrátového spojení, kde by se důležitá data zobrazovala na display, který by se dal připojit na patici wemos.

LITERATURA

- [1] BRZOBOHATÝ Lukáš: *Měření teploty*. Brno, 2011. 19s., CD. FSI VUT v Brně, Ústav strojírenské technologie, Vedoucí práce Ing. Kamil Podaný, Ph.D.
- [2] *Čidla teploty Pt 100: datasheet* [online]. SENSIT HOLDING [cit. 2020-12-07]. Dostupné z: <http://remaxcz.com/old/katalog/snimace/teplota/kl/char/Pt100.pdf>
- [3] EARNSHAW, Alan a Norman Neill GREENWOOD. *Chemie prvků*. Svazek I. Přeložil František JURŠÍK. Praha: Informatorium, 1993. ISBN 80-85427-38-9.
- [4] *Polovodiče Termistory: závody práškové metalurgie Šumperk* [online]. Šumperk: obchodně technická služba a propagace s. p., 1990 [cit. 2020-12-07]. Dostupné z: <http://www.ferity.cz/termistory/negohm.pdf>
- [5] ŠPRINGL, Vít. Měření teploty: polovodičové odporové senzory teploty. *Vyvoj.hw.cz: teorie-a-praxe* [online]. 19. Srpen 2004 [cit. 2020-12-07]. Dostupné z: <https://vyvoj.hw.cz/teorie-a-praxe/dokumentace/mereni-teploty-polovodicove-odporove-senzory-teploty.html>
- [6] *DS18B20 Programmable Resolution 1-Wire® Digital Thermometer: datasheet* [online]. [cit. 2020-12-07]. Dostupné z: <https://cdn.sparkfun.com/datasheets/Sensors/Temp/DS18B20.pdf>
- [7] *Temperature and humidity module AM2302 Product Manual: datasheet* [online]. Aosong(Guangzhou) Electronics Co. [cit. 2020-12-07]. Dostupné z: <http://robotchip.ru/download/datasheet/AM2302-Datasheet.pdf>
- [8] HYRŠ, Jan: *Měření teploty termočlánky*. Brno, 2012. 32 s., CD. FSI VUT v Brně, Ústav strojírenské technologie, Odbor technologie tváření kovů a plastů. Vedoucí bakalářské práce Ing. Kamil Podaný, Ph.D.
- [9] *ThingSpeak™: Licensing FAQ* [online]. 2020 [cit. 2020-12-07]. Dostupné z: https://thingspeak.com/pages/license_faq
- [10] *Portál TMEP.cz: Mapa čidel* [online]. [cit. 2020-12-06]. Dostupné z: <https://tmep.cz/mapa/>
- [11] *Portál: myopenHAB* [online]. 2020 [cit. 2020-12-07]. Dostupné z: <https://www.myopenhab.org>
- [12] ICS : 35.100.70 APPLICATION LAYER. *ISO/IEC 20922:2016: Information technology — Message Queuing Telemetry Transport (MQTT) v3.1.1*. 2016. 2016.
- [13] SHURE, Loren. THE MATHWORKS, INC. *Explore your IoT data with ThingSpeak and MATLAB: blog* [online]. 2016 [cit. 2020-12-07]. Dostupné z: <https://blogs.mathworks.com/loren/2016/03/23/explore-your-iot-data-with-thingspeak-and-matlab/>

- [14] *Portál: espressif.com* [online]. SHANGHAI: ESPRESSIF SYSTEMS, 2020 [cit. 2020-12-08]. Dostupné z: <https://www.espressif.com/en/products/socs/esp8266>
- [15] *Portál: EscapeQuotes* [online]. 2016 [cit. 2020-12-07]. Dostupné z: <https://escapequotes.net/esp8266-wemos-d1-mini-pins-and-diagram/>
- [16] *Portál: JLCPCB.COM* [online]. 2020 [cit. 2020-12-07]. Dostupné z: <https://jlcpcb.com/>
- [17] Olovnaté a bezolovnaté pájky. *Portál: KONDIK.cz* [online]. 01.02.2019 [cit. 2020-12-07]. Dostupné z: <https://www.kondik.cz/olovnata-a-bezolovnata-pajka>
- [18] *Portál: itead.cc* [online]. [cit. 2020-12-08]. Dostupné z: https://www.itead.cc/media/wysiwyg/Products/ESP-01_Schematic.png

SEZNAM PŘÍLOH

PŘÍLOHA A - SCHÉMA ZAPOJENÍ VYVÍJENÉ DPS 54

PŘÍLOHA B – Projekt programu KiCAD vyvíjené DPS je uložen v přiloženém archívu.

PŘÍLOHA C – Schéma zapojení vyvíjené DPS je uložen v přiloženém archívu.

PŘÍLOHA D –GERBER soubory pro výrobu DPS jsou uloženy v přiloženém archívu.

OBSAH PŘÍLOŽENÉHO ARCHÍVU

—PŘÍLOHA B

logo_lib_vut.lib

—Temperature-controller-with-ESP8266

fp-E644.tmp

fp-info-cache

logo_lib_vut.lib

sestava.FCStd

sestava.FCStd1

Temperature-controller-with-ESP8266-cache.lib

Temperature-controller-with-ESP8266.kicad_pcb

Temperature-controller-with-ESP8266.kicad_pcb-bak

Temperature-controller-with-ESP8266.net

Temperature-controller-with-ESP8266.pro

Temperature-controller-with-ESP8266.sch

Temperature-controller-with-ESP8266.sch-bak

Temperature-controller-with-ESP8266.wrl

Temperature-controller-with-ESP8266.xml

test-cache.lib

—PŘÍLOHA C

schéma_zapojení.jpg

—PŘÍLOHA D

—Gerbel

Temperature-controller-with-ESP8266-B_Cu.gbr

Temperature-controller-with-ESP8266-B_Mask.gbr

Temperature-controller-with-ESP8266-B_SilkS.gbr

Temperature-controller-with-ESP8266-Edge_Cuts.gbr

Temperature-controller-with-ESP8266-F_Cu.gbr

Temperature-controller-with-ESP8266-F_Mask.gbr

Temperature-controller-with-ESP8266-F_SilkS.gbr

Temperature-controller-with-ESP8266-job.gbrjob

Temperature-controller-with-ESP8266.drl

Příloha A - Schéma zapojení vyvíjené DPS

