



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV BIOMEDICÍNSKÉHO INŽENÝRSTVÍ

DEPARTMENT OF BIOMEDICAL ENGINEERING

BEZDRÁTOVÝ DATALOGGER PRO MONITOROVÁNÍ PROSTŘEDÍ UCHOVÁVANÝCH LÉČIV – SERVEROVÁ ČÁST

WIRELESS DATA LOGGER FOR ENVIRONMENTAL MONITORING - SERVER-SIDE

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Jiří Bártek

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Jiří Sekora, MBA

BRNO 2024

Bakalářská práce

bakalářský studijní program **Biomedicínská technika a bioinformatika**

Ústav biomedicínského inženýrství

Student: Jiří Bártek

ID: 237410

Ročník: 3

Akademický rok: 2023/24

NÁZEV TÉMATU:

Bezdrátový datalogger pro monitorování prostředí uchovávaných léčiv – serverová část

POKYNY PRO VYPRACOVÁNÍ:

1) Navrhněte zařízení klient-server pro automatické zaznamenávání teploty prostředí pro uložení léčiv. Prostudujte příslušnou legislativu SÚKL k dané problematice. 2) Proveďte systémový návrh serveru pro příjem informace z dataloggeru prostřednictvím bezdrátové sítě WiFi. 3) Zařízení musí umožňovat záznam ukládání naměřených hodnot do protokolu, který bude uvádět identifikátor zařízení, datum, čas a naměřenou hodnotu. 4) Server bude automaticky generovat protokoly se záznamem za stanovený interval (např. denní / týdenní / měsíční log.). Napájení bude pomocí elektrické sítě se zálohou baterií. 5) Sestavte navržené zařízení a ve spolupráci s klientskou částí ověřte správnou funkci.

DOPORUČENÁ LITERATURA:

[1] WILSON, Jon S. The digital signal processing handbook: theory, design, and implementation. 2nd ed. Boca Raton: CRC Press, 2010, 1 sv. The electrical engineering handbook series. ISBN 978-1-4200-4604-5.

[2] WILSON, Jon S. Sensor technology handbook: theory, design, and implementation. Boston: Elsevier, c2005, ix, 691 p. Engineering. ISBN 07-506-7729-5.

Termín zadání: 5.2.2024

Termín odevzdání: 29.5.2024

Vedoucí práce: Ing. Jiří Sekora, MBA

doc. Ing. Jana Kolářová, Ph.D.
předseda rady studijního programu

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být 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.

ABSTRAKT

Tato práce popisuje návrh a zapojení serverové části, která je součástí komplexního úkolu pro měření teploty v lékařských skladovacích prostorech a odesílání naměřené teploty v určitých časových intervalech. Vytvořené zařízení bude přijímat naměřená data a následně posílat na e-mailovou adresu. Úvodní část se zabývá legislativou spojenou s uchováváním léčiv a dále čtenáře seznamuje teoreticky s použitými zařízeními a použitými protokoly. Další část práce uvádí realizaci zařízení a programové řešení. Výsledkem práce je zařízení, které je schopno komunikovat s druhým mikropočítačem a obdržená data posílat na zvolenou e-mailovou adresu pro následnou archivaci.

KLÍČOVÁ SLOVA

ESP8266, Arduino, mikrokontrolér, SÚKL, WiFi, Přenos dat , SMTP, NTP

ABSTRACT

This work describes the design and implementation of the server-side component, which is part of a comprehensive task for measuring temperature in medical storage spaces and sending the measured temperature at specified time intervals. The created device will receive the measured data and subsequently send it to an e-mail address. The introductory part deals with the legislation related to the storage of medicines and further provides the reader with theoretical information about the used devices and protocols. The next section of the work introduces the implementation of the device and software solution. The outcome of the work is a device capable of communicating with another microcontroller and sending the received data to a selected e-mail address for subsequent archiving.

KEYWORDS

ESP8266, Arduino, microcontroller, SIDC, WiFi, Data Transmission, SMTP, NTP

BÁRTEK, Jiří. *Bezdrátový datalogger pro monitorování prostředí uchovávaných léčiv – serverová část*. Bakalářská práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav biomedicínského inženýrství, 2024. Vedoucí práce: Ing. Jiří Sekora, MBA

Prohlášení autora o původnosti díla

Jméno a příjmení autora: Jiří Bártek
VUT ID autora: 237410
Typ práce: Bakalářská práce
Akademický rok: 2023/24
Téma závěrečné práce: Bezdrátový datalogger pro monitorování prostředí uchovávaných léčiv – serverová část

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/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno

.....

podpis autora*

*Autor podepisuje pouze v tištěné verzi.

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu bakalářské práce panu Ing. Jiřímu Sekorovi, MBA za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci. Děkuji také za poskytnutí podmínek a prostředků pro realizaci praktické části. Tato práce je jen část komplexního zařízení, které se skládá ze serverové a klientské části, a tak bych chtěl poděkovat i svému kolegovi Vlastimilovi Štoplovi za spolupráci při sestavování konečného zařízení.

Obsah

Úvod	11
1 Právní normy	12
1.1 Měření teploty pro skladování léčiv	12
1.1.1 Rozmístění teplotních čidel	12
1.1.2 Požadavky na měřicí zařízení	12
1.1.3 Kalibrace	12
1.2 Skladovací podmínky	12
1.3 Záznamy teploty	13
2 Model TCP/IP	14
2.1 Vrstva síťového rozhraní	14
2.2 Síťová vrstva	15
2.3 Transportní vrstva	15
2.3.1 TCP	15
2.3.2 UDP	15
2.4 Aplikační vrstva	16
2.4.1 SMTP protokol	16
2.4.2 Synchronizace času	17
3 Systémový návrh	18
4 Návrh zařízení	19
4.1 Vývojová platforma	19
4.1.1 Mikrokontrolér ESP8266	19
4.1.2 Vývojové prostředí	21
4.2 Zadané požadavky	21
4.3 Blokové schéma	22
4.4 Napájení	23
4.5 Protokol pro správu dat	23
4.6 Návrh řídicího programu	24
5 Realizace zařízení	26
5.1 Řídicí jednotka	26
5.2 Napájení	26
5.2.1 Step down/up měnič	27
5.3 Displej	28
5.4 Nastavení e-mailového účtu	29

6 Programové řešení	31
6.1 Knihovny	31
6.2 Hlavní část programu	32
6.2.1 Funkce setup C.1	33
6.2.2 Funkce loop C.2	33
6.3 Pomocné funkce	34
6.3.1 Zpracování dat pro varovné e-maily	35
6.3.2 Dekódování dat	35
6.3.3 Uložení a čtení dat	36
6.3.4 Správa odesílání e-mailů	36
7 Ověření funkčnosti zařízení a diskuze	39
7.1 Normální provoz	39
7.2 Provoz při změně podmínek	40
7.2.1 Výpadek WiFi	40
7.2.2 Výpadek WiFi při posílání e-mailu	41
7.2.3 Výpadek napájení ze sítě	42
7.2.4 Pokles či růst teplot	43
7.2.5 Vypojený senzor	44
7.2.6 Slabá baterie u klienta	45
7.2.7 Slabá baterie u serveru	45
7.2.8 Náhodné odpojení od síťového napájení	45
7.3 Diskuze	46
Závěr	48
Literatura	49
Seznam symbolů a zkratk	52
Seznam příloh	53
A Vývojový diagram	54
B Schéma zapojení hardwaru	56
C Příklady zdrojových kódů	57
C.1 Výpis funkce setup	57
C.2 Výpis části funkce loop	58
D Fotografie výsledného zapojení	59

E	Ukázka výsledků měření	61
F	Obsah elektronické přílohy	63

Seznam obrázků

2.1	Model TCP/IP	14
2.2	SMTP protokol	16
2.3	Hierarchie serverů pro NTP	17
3.1	Blokové schéma celého zařízení, server — klient	18
4.1	Vývojová deska WeMos D1 s integrovaným ESP8266	20
4.2	Ukázka prostředí Arduino IDE	21
4.3	Blokové schéma serverové části	22
4.4	Ukázka protokolu na posílaných datech	24
4.5	Návrh řídicího programu	25
5.1	Elektrické schéma zapojení napájení a detekce zdroje	27
5.2	3,3V regulátor napětí U1V10F3	28
5.3	zapojení displeje	29
5.4	Popis I2C modulu	29
6.1	Schéma návaznosti funkcí	34
7.1	Formát e-mailu při příjmu dat ze tří stanic	39
7.2	Výpis ze sériového monitoru při testu výpadku wifi	41
7.3	Ukázka e-mailu při testu výpadku wifi	41
7.4	Ukázka sériového monitoru, když je výpadek WiFi při odesílání e-mailu	42
7.5	Ukázka e-mailu, když je výpadek WiFi při odesílání e-mailu	42
7.6	Ukázka e-mailu při testu výpadku elektrické sítě	43
7.7	Výpis ze sériového monitoru při testu výpadku elektrické sítě	43
7.8	Ukázka varovného e-mailu při různých teplotních výkyvech	44
7.9	Varovný e-mail při odpojení senzoru	44
7.10	Displej při odpojení senzoru	44
7.11	Ukázka e-mailu při detekci nízkého stavu baterie	45
7.12	Varovný e-mail při poklesu napětí na záložním napájení serveru	45
7.13	Varovný e-mail při odpojení síťového napájení u klientské části	45
7.14	Varovný e-mail při odpojení síťového napájení u serverové části	46
A.1	Vývojový diagram	55
B.1	Elektrické schéma zapojení celého zařízení	56
D.1	Zapojení serverové části	59
D.2	Zapojení klientské části	60
E.1	Data první den	61
E.2	Data druhý den	62
E.3	Varovné e-maily v průběhu testů	62

Úvod

Při uchovávání některých léčiv je velmi důležité hlídat teplotu, ve které jsou skladovány. Úkolem práce je seznámení se s legislativou o uchovávání léčiv a následně navrhnout zařízení, které automaticky bude přijímat, zpracovávat a následně odesílat kontrolní e-maily obsahující naměřenou teplotu a další důležité parametry. Zařízení bude pracovat pomocí WiFi modulu ESP8266, který bude přijímat data z druhého mikropočítače, který bude sbírat data z teplotního senzoru. Celá komunikace bude probíhat přes bezdrátovou síť WiFi . Tato práce systematicky postupuje od seznámení s legislativou přes teoretické základy a popis jednotlivých komponent až k návrhu zapojení a řídicímu programu. V první části práce jsou podrobně rozebrány teoretické aspekty spojené s projektem. Následuje detailní popis každé komponenty zařízení a navrhovaného zapojení. Další fáze práce se věnuje implementaci teorie do praxe. Praktická část obsahuje popis celého procesu výroby zařízení, včetně programování. V této části jsou zdokumentovány kroky, které vedou k vytvoření funkčního zařízení. Nakonec je věnována pozornost ověření efektivity navrženého zařízení při různých krizových stavech. Závěrečná část práce shrnuje dosažené výsledky a případně navrhuje možné oblasti pro další vylepšení.

1 Právní normy

1.1 Měření teploty pro skladování léčiv

1.1.1 Rozmístění teplotních čidel

Pro zařízení jako jsou mrazicí zařízení, chladničky a chladicí boxy pro skladování léčiv se teplota ověří v prostoru takzvanou teplotní mapou. Poté je nutné provést ověření teplotní mapy po zásahu do chladicích či klimatizačních jednotek, jenž mohou způsobit změnu teploty. Měřicí body pro vznik teplotní mapy se určují podle velikosti a tvaru prostoru a také podle umístění chladicích jednotek. Teploměry se umísťují podle teplotní mapy na kritická místa, což jsou například místa s nejvyšší teplotou.

1.1.2 Požadavky na měřicí zařízení

Nutným požadavkem zařízení, pro měření teploty v těchto podmínkách, je mít vlastní zdroj napájení v případě výpadku elektrického proudu. Dále by toto zařízení mělo obsahovat výstražnou signalizaci, která se spustí po překročení horního nebo spodního limitu. Je nutné provádět pravidelné kontroly tohoto výstražného zařízení.

1.1.3 Kalibrace

Zákonem není přímo stanoveno, jak často by měla probíhat kalibrace měřidel v lékárnách. Státní ústav pro kontrolu léčiv (dále jen SÚKL) však doporučuje provádět pravidelnou kalibraci alespoň jednou za pět let. Tato opatření jsou klíčová pro udržení přesnosti a spolehlivosti výsledků. Při kalibraci by měla být zjištěna chyba měření. U termolabilních látek, což jsou látky náchylné k degradaci při vysoké teplotě, by mělo měření dosahovat maximální chyby 0,5 °C a u ostatních látek se pohybuje chyba do 1 °C.

1.2 Skladovací podmínky

Důležitým prvkem u skladování je seznam uchovávaných léčiv pro jednotlivé skladovací prostory, jako jsou chladničky, mrazničky, skladování surovin a pomůcek pro výrobu léčiv, skladovací prostory pro nebezpečné látky a v neposlední řadě skříně a regály. Skladované látky by se neměly nacházet v bezprostřední blízkosti topných těles a chladicích desek a také by neměly být vystaveny slunečnímu záření. Všechny

specifikace ohledně uchování by měly být zobrazeny na obalu dané látky nebo v příbalovém letáku. Při nedodržení podmínek skladování může dojít ke znehodnocení léčiva a jeho použití může být neefektivní nebo nebezpečné.

Nejčastější podmínky uchování léčiv jsou dle SÚKL [13] následující:

- Uchovávejte při teplotě do 25 °C nebo až do 30 °C
- Uchovávejte v chladničce (2 °C až 8 °C)
- Chraňte před chladem
- Chraňte před mrazem
- Uchovávejte v původním obalu
- Uchovávejte v dobře uzavřeném vnitřním obalu
- Uchovávejte tak, aby byl přípravek chráněn před světlem
- Uchovávejte tak, aby byl přípravek chráněn před vlhkostí

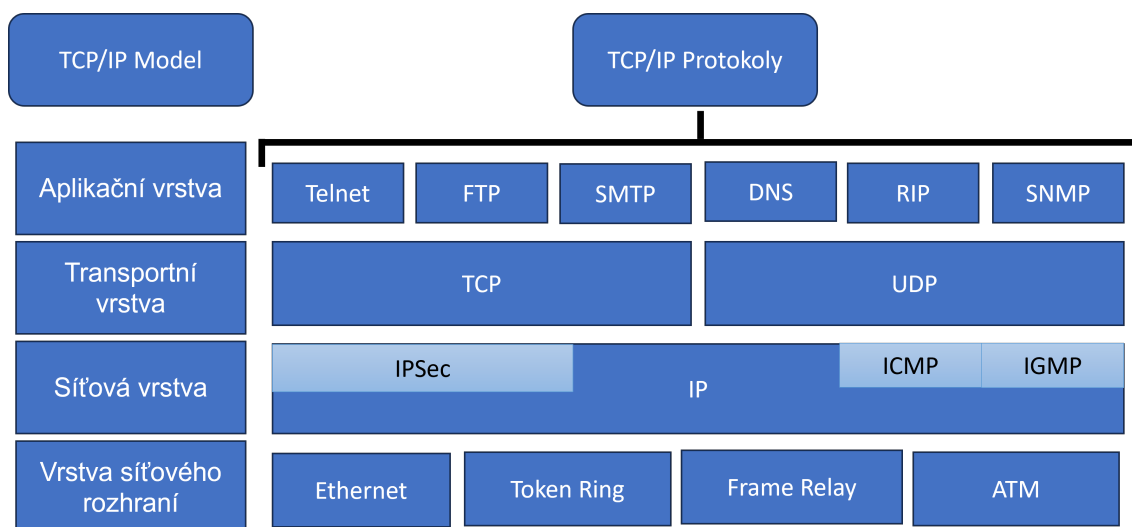
1.3 Záznamy teploty

Je nezbytné pravidelně zaznamenávat teploty, za kterých jsou léčiva skladována, alespoň jednou denně. Toto by mělo být prováděno v době, kdy se očekává, že budou podmínky skladování léčiv nejvíce nepříznivé. U termolabilních léčiv se pořizují záznamy teplot kontinuálně pomocí vhodného zařízení, jako je například teplotní datalogger. Záznamy se poté zálohují a i v dnešní době se stále uchovávají jako výtisk.

Zákony nařizují, aby se během výroby a distribuce léčiv vedly záznamy, buď písemně nebo pomocí elektronických zařízení. Tyto záznamy dokazují, že se postupovalo podle stanovených postupů a pokynů a zaznamenávají se do nich i jakékoli odchylky. Nic nesmí být zamlčeno. Záznamy týkající se výroby, distribuce i skladování léčiv musí být uchovávány v přístupné a srozumitelné podobě. Přesná dokumentace je klíčovou součástí výroby, skladování a přepravy léčiv. K dokumentaci patří instrukce a záznamy. Nejjednodušší forma dokumentace je klasická papírová, ale dnes se běžně využívají elektronické záznamy. Dokumentace sleduje a zaznamenává všechny aspekty spojené s výrobou, skladováním a přepravou léků. Každý dokument musí mít kopii. Aby byla zajištěna přesnost a dostupnost dokumentů, provádějí se pravidelné kontroly. Dokumentace zahrnuje popisy činností výrobce, instrukce pro balení, testování, výrobu, přepravu a skladování, postupy, specifikace, protokoly, záznamy o činnostech, zprávy, analytické certifikáty a technické smlouvy.[6][5][23]

2 Model TCP/IP

TCP/IP model je referenční model sítí, který specifikuje, jakým způsobem jsou data přenášena prostřednictvím počítačové sítě. Model je členěn do celkem čtyř vrstev, kde každá vrstva má specifickou funkci a odpovědnost. Tento model je závazným pro návrh, implementaci a správu moderních sítí. Jednotlivé vrstvy modelu zajišťují, aby síťová komunikace byla efektivní a spolehlivá.



Obr. 2.1: Model TCP/IP (převzato z [21])

2.1 Vrstva síťového rozhraní

Někdy se můžeme dočíst, že se této vrstvě říká linková. Jedná se o první vrstvu v modelu TCP/IP. Tato vrstva se stará o fyzické propojení a definuje, jak mají být data odesílána přes síť. Existuje spousta přenosových technologií, mezi které patří například Ethernet, Token Ring nebo FDDI.

V této práci bude preferován přenos prostřednictvím bezdrátové sítě WiFi. WiFi je bezdrátová komunikační technologie postavená na standardech rodiny IEEE 802.11, která využívá rádiové vlny pro přenos dat mezi zařízeními. Jedním z hlavních přínosů této technologie je eliminace potřeby fyzických kabelů, což přináší uživatelům pohodlí a flexibilitu, která se hodí v našem návrhu. Důvodem zvolení této technologie je mikrokontrolér ESP8266, který je schopen jak sám generovat síť WiFi, tak i připojovat se k jiným vysílačům.

2.2 Síťová vrstva

Jedná se o druhou vrstvu. Jejím hlavním úkolem je zajišťovat přenos datových paketů nejen mezi sousedními uzly v síti, ale také mezi všemi ostatními uzly. Hlavním nástrojem pro tuto funkci je Internet Protocol (IP). IP protokol operuje na principu přenosu dat ve formě IP-paketů, které obsahují IP hlavičku s adresou příjemce a směrovacími informacemi. Jednou z klíčových vlastností IP je, že v případě výpadku některého uzlu, paket najde alternativní cestu k cíli. To vede k možnosti doručení dat v různém pořadí a je na vyšších vrstvách TCP/IP, jako například Transmission Control Protocol (TCP), aby zajistily správné seřazení přijatých dat. Tato vrstva nezaručuje stoprocentní správnost přijatých dat. V případě poškozených dat se neprovádí oprava a přijímání dat pokračuje dál. Tato vlastnost může být výhodná například při živých přenosech obrazu nebo zvuku, kde je důležitější minimalizovat zpoždění než garantovat absolutní správnost dat.

2.3 Transportní vrstva

Tato vrstva má na starosti přenos dat mezi koncovými zařízeními. Dva hlavní protokoly v této vrstvě jsou Transmission Control Protocol (TCP) a User Datagram Protocol (UDP). Nelze říci jaký protokol je lepší, protože se každý používá na jiné přenosy.

2.3.1 TCP

U tohoto protokolu se před přenosem dat mezi zařízeními vytvoří spojení. TCP poskytuje spolehlivý a zajištěný přenos dat. Obsahuje mechanismy pro detekci a opravu chyb při přenosu dat a také řídí správnost pořadí, v jakém mají data po sobě následovat. TCP je vhodný pro aplikace, které vyžadují spolehlivý a řízený přenos dat, jako například webové prohlížeče, e-mailové klienty a aplikace pro přenos souborů.

2.3.2 UDP

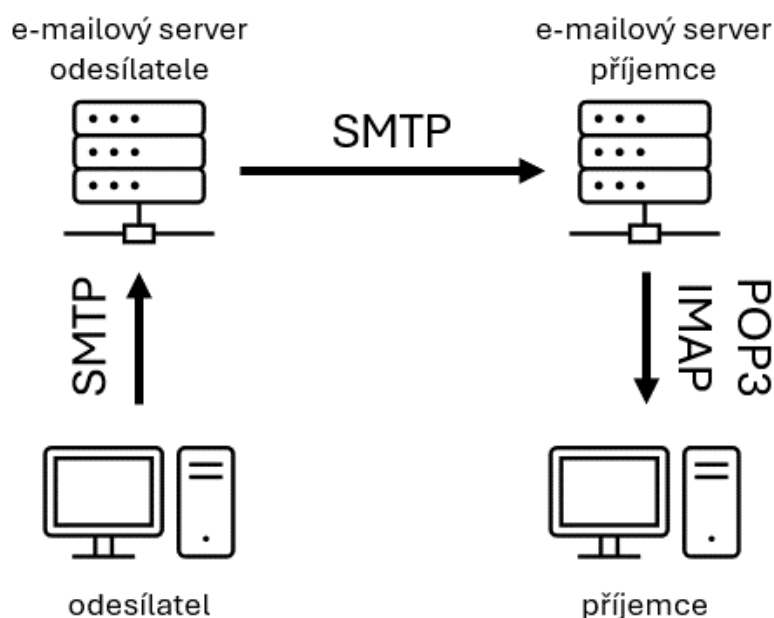
Zde se oproti TCP nevytváří spojení mezi jednotlivými zařízeními. Klíčový rozdíl je ten, že je méně spolehlivý, protože neobsahuje stejné mechanismy na opravu dat. Naproti tomu je mnohem rychlejší a méně zatěžuje síťovou komunikaci. UDP se často používá v aplikacích, kde je důležitější rychlost než spolehlivost. Využití se najde v hlasové a video komunikaci, online hrách a dalších aplikacích, kde malá ztráta dat není kritická, ale reakční čas je důležitý.

2.4 Aplikační vrstva

Aplikační vrstva je poslední a zároveň nejvyšší vrstva v architektuře TCP/IP modelu. Představuje klíčový prvek, který umožňuje koncovým uživatelům komunikovat a využívat různé síťové služby. Tato vrstva se zaměřuje na poskytování uživatelsky příjemných aplikací a služeb, které mohou být využívány v rámci počítačových sítí. Od e-mailových klientů po webové prohlížeče a další komunikační nástroje, aplikační vrstva zahrnuje širokou škálu protokolů, které umožňují efektivní a spolehlivou komunikaci mezi zařízeními. V následujících podkapitolách jsou zmíněny protokoly, které jsou využity v navrhovaném systému.

2.4.1 SMTP protokol

SMTP (Simple Mail Transfer Protocol) je standardní internetový protokol pro přenos elektronické pošty (e-mailu). Jeho hlavním účelem je přenášet a doručovat e-mailové zprávy mezi počítači nebo mailovými servery. SMTP pracuje ve spolupráci s jiným protokolem nazývaným POP3 (Post Office Protocol) nebo IMAP (Internet Message Access Protocol), které slouží k příjmu e-mailů na straně příjemce. SMTP pracuje na principu klient-server, kde klient (například e-mailový klient na vašem počítači) odesílá e-mailovou zprávu serveru SMTP. Tento server pak předává zprávu k doručení cílovému serveru, který zase zprávu uloží do doručovací schránky příjemce.



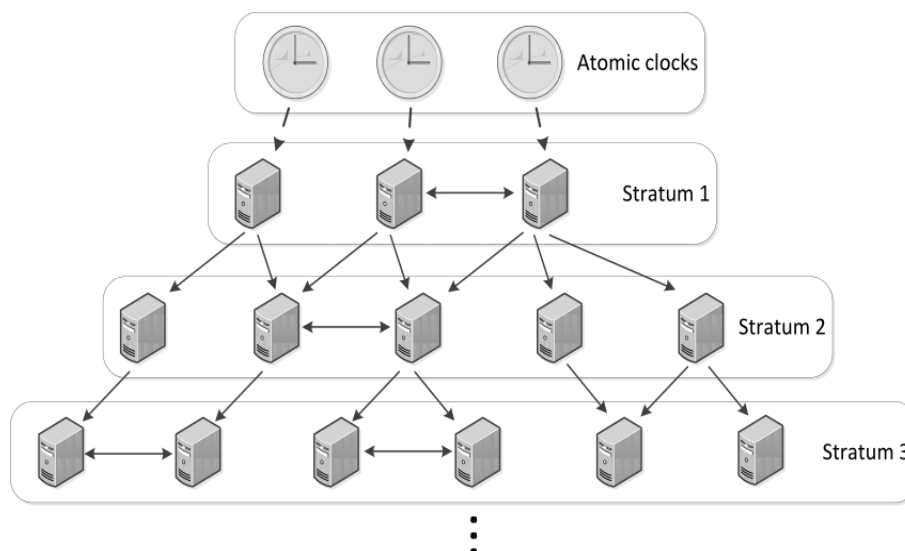
Obr. 2.2: SMTP protokol (převzato z [2])

2.4.2 Synchronizace času

Protokol pro synchronizaci času se nazývá Network Time Protocol (NTP). Jedná se o internetový protokol používaný k synchronizaci se zdroji času počítačových hodin v síti. Synchronizace hodin je velmi důležitá pro dosažení shody času mezi klientem a serverem, čímž eliminujeme rozdíl v místních hodinách. Umožňuje správnou koordinaci a časovou shodu mezi různými zařízeními, což má důsledky na správnou funkci, bezpečnost a řádnou evidenci časově citlivých událostí.

Proces synchronizace času

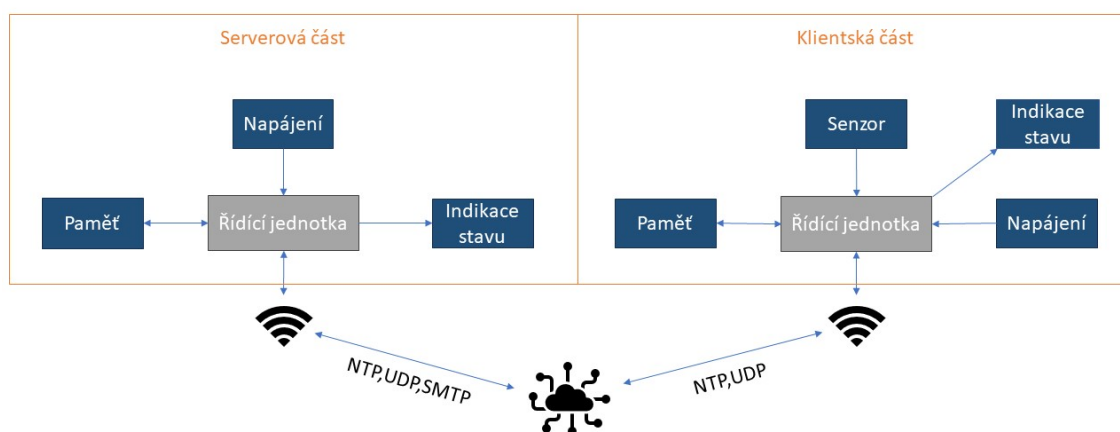
Klient se dotáže na časové informace serveru NTP. Klienti se mohou dotazovat i více serverů a snaží se najít nejlepší časový údaj. NTP servery jsou hierarchicky rozděleny do tzv. "strata". Server s nejvyšším číslem (např. stratum 1) je přímo spojen se zdrojem času, zatímco servery s nižším stratem jsou synchronizovány s vyššími straty. Jako poslední krok klient dopočítá zpoždění a následně nastaví odpovídající čas.



Obr. 2.3: Hierarchie serverů pro NTP (převzato z [24])

3 Systémový návrh

Základním cílem celého zařízení je měření teploty teplotním senzorem v prostorách pro uchovávání léčiv a následné odeslání naměřené teploty. S tím je posláno i datum a čas, kdy byly hodnoty naměřeny a také stav baterie. Server tyto data zpracuje a odešle na zvolený e-mail. Celá komunikace probíhá za pomoci dvou mikrokontrolérů ESP8266 přes bezdrátovou síť. Toto je stručný popis funkce celkového zařízení a dále se budeme věnovat pouze návrhu serverové části. Hardwarová část je složena z mikroprocesoru ESP8266, který je vestavěn na vývojovou desku WeMos D1. Deska je napájena ze sítě. Paměť postačí ta, která je zabudována v mikrokontroléru. Pro komunikaci je použita síť WiFi a data jsou směřována pomocí UDP protokolu. A obě části, jak klient, tak i server jsou synchronizovány pomocí NTP protokolu. Po obdržení naměřených dat dojde k uložení a dekodování zprávy a, následně za pomoci SMTP protokolu, po určitém časovém úseku jsou zaslána na příslušný e-mail. Popis hardwaru a použitých vývojových platforem a přenos dat je rozepsán v následujících kapitolách.



Obr. 3.1: Blokové schéma celého zařízení, server — klient

4 Návrh zařízení

Pro návrh je využit koncept Internet of Things, který se týká propojení fyzických zařízení, objektů a systémů s internetem, aby mohly vzájemně komunikovat a vyměňovat data. Tím se vytváří rozsáhlá síť zařízení, které mohou spolupracovat, sdílet informace a vykonávat úkoly v reálném čase. Internet of Things se využívá v mnoha oborech, od zemědělství, přes průmysl, až po zdravotnictví a mnoho dalších. Mezi základní prvky patří připojení k internetu, což v této práci bude obstaráno pomocí WiFi. Dále tyto zařízení obsahují nějaký typ senzoru pro snímání okolí, jako například teplotní senzor. Důležitým a zásadním prvkem je automatizace procesů, která se v této práci soustředí na pravidelný příjem a odesílání naměřených hodnot pro následnou archivaci.

4.1 Vývojová platforma

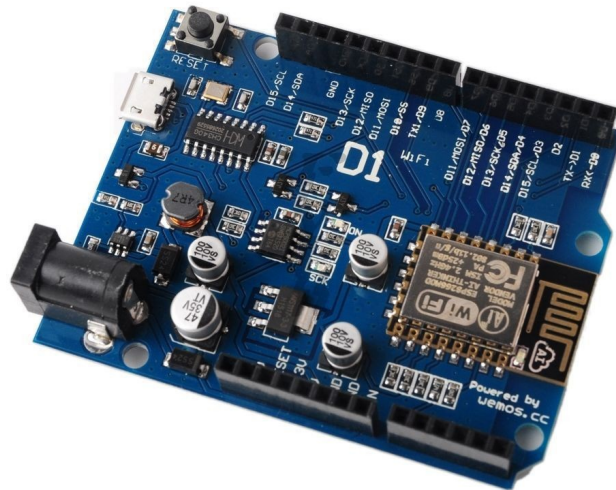
Jako vývojová základna pro návrh serverové části byl použit mikrokontrolér ESP8266 umístěný na desce WeMos D1. Programování a nahrání programu se realizuje pomocí vývojového prostředí Arduino IDE.

4.1.1 Mikrokontrolér ESP8266

V této práci pro realizaci daného problému je použit mikrokontrolér ESP8266 od firmy espressif. Toto zařízení bylo zvoleno z mnoha důvodů, jedním z nich je integrovaný WiFi modul spolu s TCP/IP protokolem, který umožňuje zařízení připojit se k bezdrátové síti a komunikovat přes internet. Další výhodou je velikost čipu a také jeho nízká spotřeba energie, a má také schopnost konzistentně fungovat v průmyslovém prostředí díky širokému rozsahu provozních teplot. Důležitý parametr, který stojí za zmínku, je provozní pásmo 2,4 GHz, které je určeno pro bezlicenční provoz bezdrátových zařízení v průmyslu, vědě a medicíně. Kvůli nepotřebě získat licenci od regulačních orgánů se urychluje a usnadňuje vývoj a inovace na různých bezdrátových zařízeních.

Pro ESP8266 byla zvolena vývojová deska WeMos D1. Tato deska umožňuje snadné připojení k WiFi. Obsahuje také spoustu vstupně-výstupních pinů pro připojení senzorů a dalších periférií. Důležité je zmínit, že piny pracují na napětí 3,3 V a v případě připojení 5 V může dojít ke zničení modulu. Pokud tedy chceme propojit s WeMos deskou ostatní moduly, je nutné vyhledat kompatibilitu s napájením na 3,3 V. Hlavní výhodou a důvod, proč byla tato platforma zvolena, je ten, že je kompatibilní k Arduino IDE, což je jednoduché a příjemné prostředí pro uživatele. Velká výhoda je v tom, že uživatelé, kteří v minulosti pracovali se zařízením Arduino, tak

nemají žádný problém pracovat s touto deskou, protože za prvé je velmi podobná vzhledově a rozložením pinů a za druhé je kompatibilní k vývojovému prostředí Arduino. Dalším neoddiskutovatelným plusem je cena, výkon platformy a také možnost nabíjet celou desku přes USB kabel.



Obr. 4.1: Vývojová deska WeMos D1 s integrovaným ESP8266 (převzato z [7])

Mezi klíčové parametry vývojové platformy patří:

- piny: 11 vstupně-výstupních pinů a jeden analogový pin (pracují pouze na napětí +3,3 Voltů)
- Procesor: Tensilica L106 32bitový mikrokontrolér pracující na frekvenci až 160 MHz.
- Paměť Flash: od 512 kB do 16 MB
- Krystalový oscilátor: pro generaci hodinového signálu
- Rozměry desky: 68,8 x 53,4
- Rozměry mikrokontroléru: 24 × 16 mm
- Napájení: 3.3V, externě se stabilizátorem max. 5V
- Operační teplotní rozsah: -40°C až 80°C.
- Komunikační rozhraní: Podpora UART, SPI, I2C, 1Wire
- Provozní proud: 80 mA

4.1.2 Vývojové prostředí

Volba vývojového prostředí Arduino IDE byla jednoduchá, vzhledem k tomu, že je perfektně kompatibilní k naší desce a zároveň je velmi uživatelsky přívětivé. Pro základní práci s prostředím se používá panel s ikonami, které jsou přizpůsobeny rychlé práci s prostředím.

- ✓ Ověření, zda kód neobsahuje žádné chyby.
- ➔ Kompilace kódu a nahrání
- 📄 Otevření nového sketchu
- 📁 Otevření již existujícího sketchu
- 📄 Uložení kódu
- 🔧 Otevření sériového monitoru

Další příkazy se nacházejí v pěti nabídkách: File , Edit , Sketch , tools , Help.



Obr. 4.2: Ukázka prostředí Arduino IDE (převzato z [3])

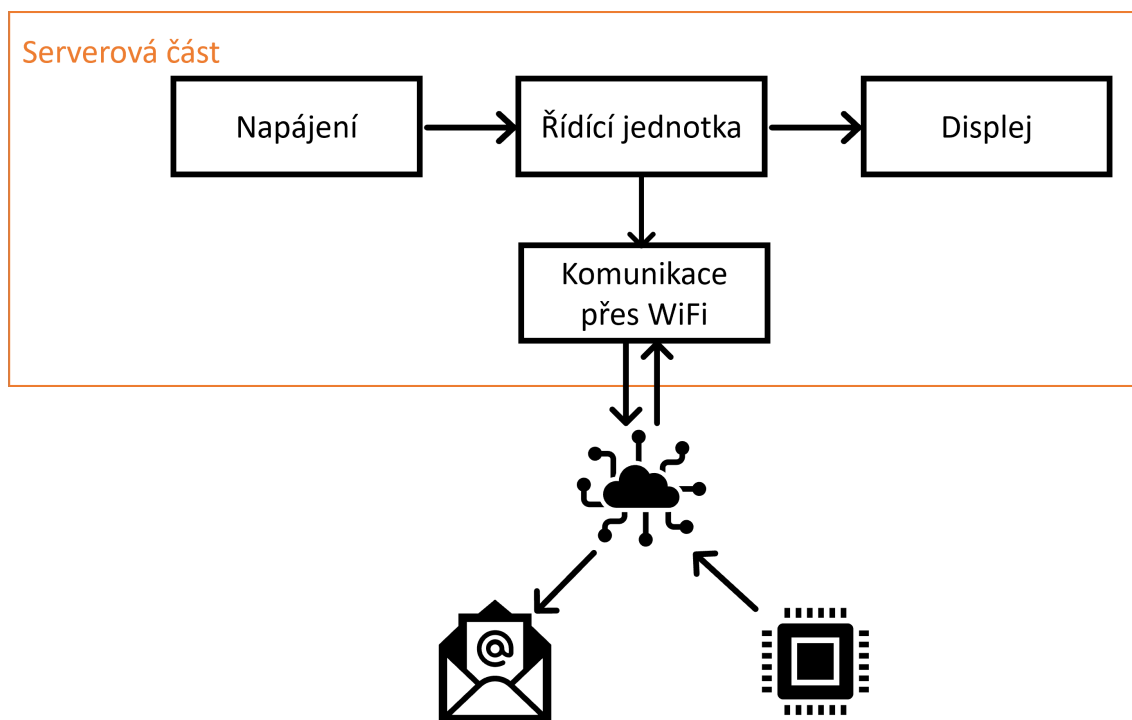
4.2 Zadané požadavky

Základem práce je návrh zařízení, které by mohlo být použito ve skladovacích prostorech pro různé druhy léčiv, za účelem sběru dat ze zařízení, která měří teplotu. Hlavním požadavkem je pravidelný a spolehlivý sběr dat z měřících stanic. Dále by toto zařízení mělo umět vyhodnotit různé stavy, jako je například zvyšování či pokles teplot ve skladovacích prostorech. Další stav, který by měl indikovat a upozorňovat

na něj, je stav baterie, a to u klienta tak i u serveru. Nejdůležitějším požadavkem je odesílání těchto varování na e-mailovou adresu ihned po doručení hodnot, a pak jednou za den poslat v jednom e-mailu všechny naměřené hodnoty. Tento posílaný záznam by měl obsahovat identifikační číslo, z kterého jsou data měřena, teplotu a čas, kdy bylo měření pořízeno, a nakonec by měl obsahovat stav baterií a zmínku o tom, jestli dané zařízení je napájeno ze sítě nebo právě ze záložních baterií.

4.3 Blokové schéma

Blokové schéma znázorňuje strukturu systému a propojení mezi jeho klíčovými komponentami. Základem je mikroprocesor, který zajišťuje centrální řízení a zpracování dat. Jeho zdrojem napájení je samostatná část, která zajišťuje přísun elektrické energie. Pro komunikaci s externími zařízeními a službami využívá WiFi, která umožňuje připojení k internetu. Díky WiFi systém umožňuje komunikovat s dalšími částmi prostřednictvím různých protokolů, jako jsou SMTP a UDP. SMTP se používá pro odesílání e-mailů na určenou e-mailovou adresu, zatímco UDP slouží k výměně informací s klientskou částí. Celé schéma poskytuje přehled o tom, jak různé části systému spolupracují a ukazuje klíčovou roli Řídící jednotky jako spojovacího bodu pro všechny ostatní komponenty.



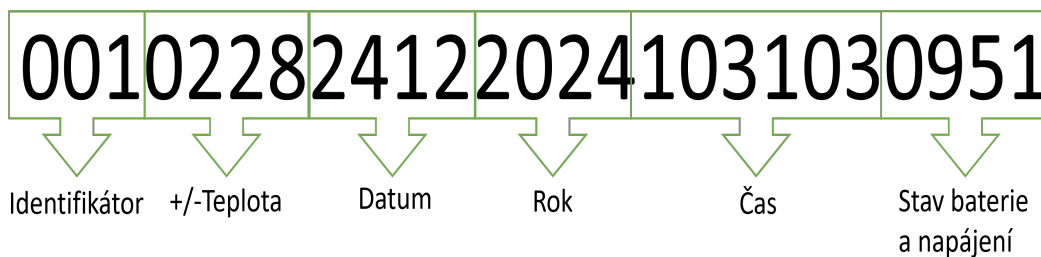
Obr. 4.3: Blokové schéma serverové části

4.4 Napájení

Celé zapojení serverové části spočívá v přivedení napájecího napětí na desku WeMos D1. Důležité je použít převodník úrovní napětí, který bude převádět napětí ze sítě na nižší hodnoty, na kterých deska pracuje. K zařízení se přidá i záložní baterie, ale na funkci zařízení to nemá vliv, protože v případě výpadku proudu nebude fungovat ani internetové připojení. Baterie je zde přidána jen kvůli záloze, kdyby došlo k nějakému neúmyslnému odpojení od sítě a nebo při nějaké poruše elektrické instalace. V tom případě, pokud by nebyla poškozena i elektroinstalace pro napájení WiFi, tak by se poslal varovný e-mail o stavu napájení a zařízení by pracovalo bez omezení po dostatečnou dobu, než by byl problém odstraněn.

4.5 Protokol pro správu dat

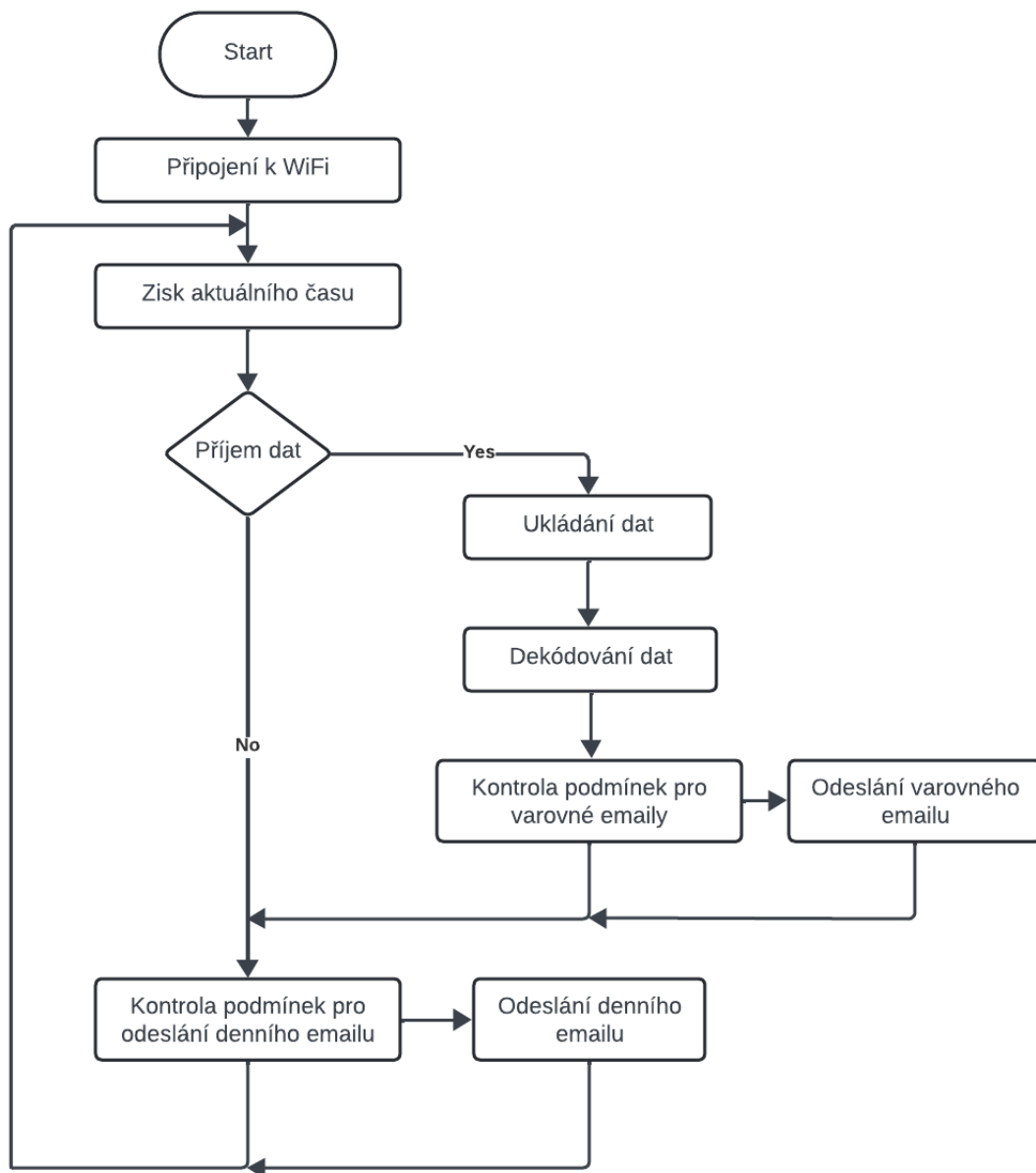
Hlavní funkce programu je příjem naměřených dat od klientského zařízení. Data jsou posílána protokolem, který byl ustanoven ve spolupráci s odesílatelem dat tak, aby se později dekodovala do e-mailové zprávy. Protokol obsahuje identifikátor zařízení, z kterého měřené hodnoty přišly a ten je reprezentován třemi čísly zprávy. Pak je jedna pozice vyhrazena znaménku před samotnou teplotou. Znaménko minus je reprezentováno číselnou hodnotou jedna a znaménko plus zase hodnotou nula. Poté následuje samotná naměřená teplota, která zabírá tři znaky, protože je násobena číslem deset, aby se dosáhlo celého čísla. Následuje číselný řetězec o délce 14 znaků udávající, kdy byly hodnoty naměřeny a jsou zapsány v pořadí den, měsíc, rok, hodiny, minuty a sekundy. Na další třech pozicích v protokolu je informace o bateriovém stavu u jednotlivých měřících stanic. A na posledním místě se nachází zápis o tom, jestli je zařízení nabíjeno ze sítě nebo používá energii z baterií. Napájení ze sítě je reprezentováno číselnou hodnotou jedna, a když používáme energii z baterií, tak v protokolu je zasílána hodnota nula. Na posledním místě protokolu se ještě může objevit číslo tři, které reprezentuje také nabíjení z baterie, ale pomáhá rozlišit stavy, jestli data byla naměřena při celkovém výpadku napájení v budově (tzn. WiFi nefunguje) nebo došlo jen nedopatřením k odpojení od síťového napájení a zařízení běží na baterie.



Obr. 4.4: Ukázka protokolu na posílaných datech

4.6 Návrh řídicího programu

Program by měl začít připojením k WiFi síti, čímž si zajišťuje možnost komunikace, buď s měřicími stanicemi nebo se SMTP a NTP servery. Poté přijímá data a ukládá je do předem určeného úložiště. Jakmile jsou data přijata a uložena, program je dekóduje, aby je bylo možné dále zpracovávat. Po dekódování program kontroluje podmínky, které pokud jsou splněny, jako je například překročení teploty, tak program odešle varovný e-mail, aby upozornil na potenciální problém nebo podal výstrahu. Nakonec by vždy jednou za den měla být poslána všechna nasbíraná data na určenou e-mailovou adresu a zde by mělo dojít k určitému typu archivace.



Obr. 4.5: Návrh řídicího programu

5 Realizace zařízení

Zařízení je vytvořeno ze tří stavebních bloků, které se člení do menších částí. První blok je samotný čip ESP8266. Další je napájení a poslední je signalizace v podobě displeje. Celkové zapojení těchto částí je znázorněno v příloze B.1 a v další příloze D.1 je foto výsledného zapojení.

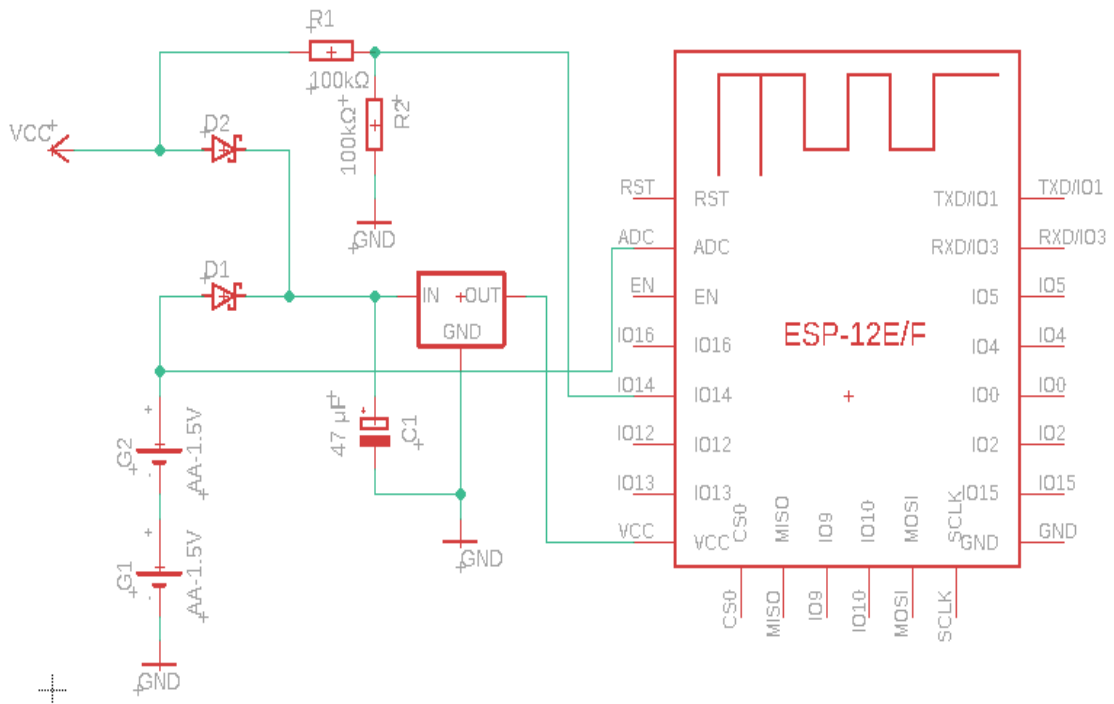
5.1 Řídicí jednotka

Jako řídicí jednotka byl použit čip ESP8266, který je popsán i s jeho parametry a s vývojovou deskou v kapitole Vývojová platforma 4.1. Nabízí se zde možnost nepoužívat vývojovou desku WeMos D1 a použít samotný čip jen s vývody pro napájení a připojení displeje. Ale jelikož se jedná jen o prototyp a serverová část nemusí hledět úplně na velkou úsporu místa, vzhledem k tomu, že může být uložena kdekoliv, kde je dostupné napájení, tak pro konečné řešení byla zvolena právě deska WeMos D1. Deska nám poskytuje jednoduchou práci s připojováním napájení a dalších komponent. A v konečném důsledku nedosahuje ani obrovských rozměrů.

5.2 Napájení

Na obrázku 5.1 je vidět napájení čipu ESP8266. Napájení z USB kabelu je reprezentováno značkou VCC a záložní napájení je provedeno spojením dvou 1,5V baterií. Proud z obou napájecích větví putuje přes svou schottkyho diodu, která je zapojena v propustném směru, ale při opačném směru toku proudu je zapojena v závěrném směru a slouží pro ochranu před zpětnými proudy a tím účinně odděluje napájecí zdroje. Následně zapojení směřuje do měniče napětí 5.2.1, který převede 5V na stabilních 3.3V. Před samotným měničem je mezi vstupem VIN a GND připojen elektrolytický kondenzátor, který by podle výrobce měl mít minimální hodnotu 33 μ F (v zapojení je použita hodnota 47 μ F) a slouží jako ochrana před napětovými špičkami, které mohou být mnohem vyšší než vstupní napětí a mohly by tak poškodit měnič. Z výstupu VOUT vychází 3.3V a je přiváděno na desku. Pro programovou část je nutné ještě detekovat zdroj napětí a u baterií také jejich hodnotu. Za bateriemi je vyveden vodič, který směřuje do ADC převodníku, který má 10 bitů, což znamená, že může měřit napětí v rozsahu od 0 do 1023. Vstupní rozsah napětí na pinu A0 je od 0 do 3.3V, což je splněno, protože dvě baterie nám dají dohromady 3V. U napájení ze sítě není potřeba znát hodnotu, ale je potřeba jen detekovat jestli je deska napájena ze sítě nebo z baterií a to se dá provést pomocí GPIO pinů, ale je nutné zapojit dělič napětí, který sníží napětí pod 3.3V. Dělič je vytvořen z rezistoru

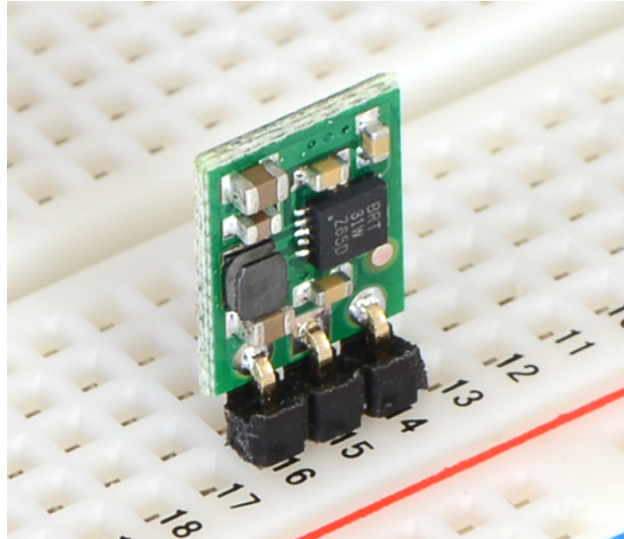
R1 a R2 s hodnotou $100\text{k}\Omega$, tudíž na piny bude přicházet 2.5V a tím pádem nedojde k poškození AD převodníku na vstupu.



Obr. 5.1: Elektrické schéma zapojení napájení a detekce zdroje

5.2.1 Step down/up měnič

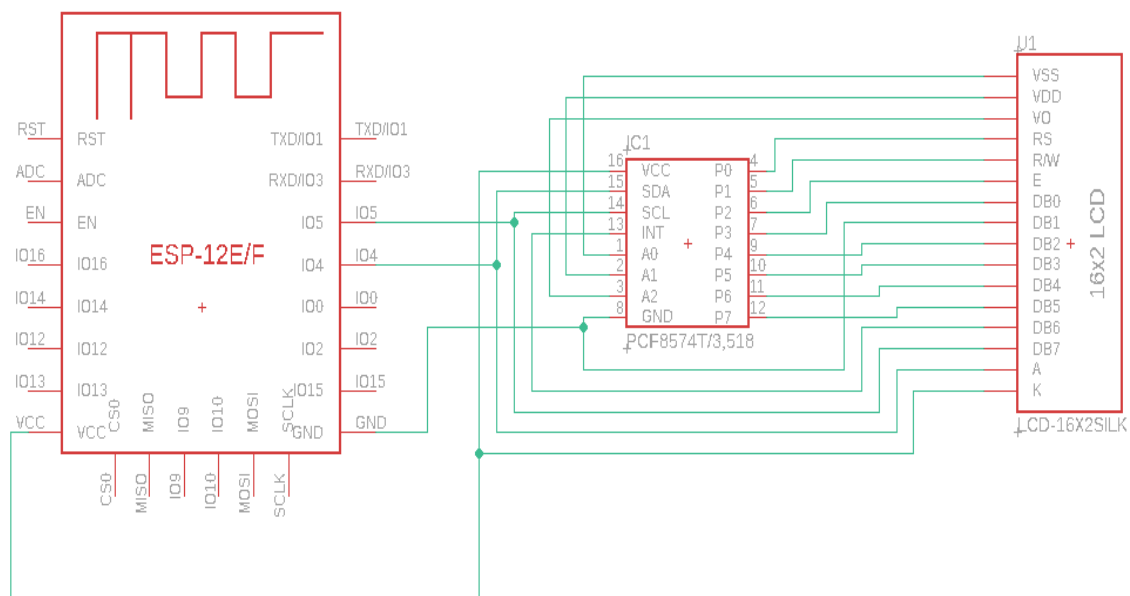
- **Model:** Pololu 3,3V regulátor napětí U1V10F3
- **Výstupní napětí (V_{out}):** $3,3\text{ V}$
- **Výstupní proud (I_{out}):** až $1,2\text{ A}$
- **Vstupní napětí (V_{in}):** $0,5\text{ V}$ až $5,5\text{ V}$
- **Účinnost:** až 65% až 85% v závislosti na vstupním napětí a zatížení
- **maximální Klidový proud:** 1 mA při nízkém vstupním napětí
- **Spínací frekvence:** 1 MHz
- **Provozní teplota (T_J):** $-40\text{ }^\circ\text{C}$ až $+140\text{ }^\circ\text{C}$



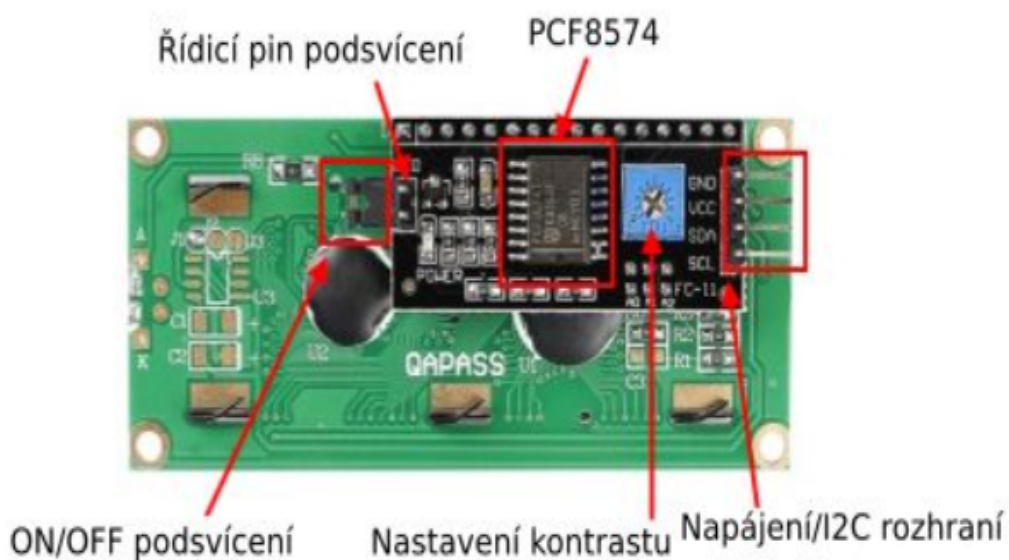
Obr. 5.2: 3,3V regulátor napětí U1V10F3 (převzato z [15])

5.3 Displej

Aby při provozu bylo poznat na první pohled, jestli zařízení správně funguje, tak je použit LCD displej 1602. Jedná se o displej 16 x 2, tj. dva řádky a šestnáct sloupců s modrým podsvícením a bílými znaky. Díky přídatnému I2C konvertoru PCF8574 je možné s displejem komunikovat přes I2C sběrnici. Na přídatném modulu 5.4, kde je již zmiňovaný konvertor, se nachází i trimr pro nastavení kontrastu a také je zde možnost vypnutí podsvícení. Adresa displeje je nastavena z výroby na 0 x 27 nebo 0 x 3F. Napájecí napětí, které je nutné přivést, je 5V a rozměry jsou 80 x 36 x 17 mm. Díky konvertoru PCF8574 stačí displej k desce WeMos D1 připojit pouze čtyřmi dráty. Jeden je přiveden na napětí 5V, další na GND a zbylé dva jsou označeny SDA a SCL a ty jsou přivedeny na stejnojmenné GPIO piny.



Obr. 5.3: zapojení displeje



Obr. 5.4: Popis I2C modulu

(převzato z [1])

5.4 Nastavení e-mailového účtu

Pro tuto bakalářskou práci bylo zapotřebí založit dva nové e-mailové účty. Byly zvoleny účty na bezplatné e-mailové službě jménem Gmail od společnosti Google

LLC. První e-mail slouží jako příjemce posílaných e-mailů s daty. Do tohoto účtu by měla mít přístup pověřená osoba, která má na starosti archivaci a také kontrolu poruchových hlášení. Druhý e-mail byl založen pro odesílání těchto e-mailů. U tohoto e-mailu bylo potřeba zapnout dvoufázové ověření a následně bylo zapotřebí si vygenerovat heslo aplikací, díky kterému má zařízení přístup k tomuto e-mailovému účtu a může jeho prostřednictvím odesílat sesbíraná data.

6 Programové řešení

V této kapitole budou detailně popsány klíčové části zdrojového kódu zařízení, které implementuje algoritmus diskutovaný v kapitole týkající se návrhu řídicího programu.4.6

6.1 Knihovny

Použity jsou knihovny důležité pro vývoj aplikací na platformě ESP8266, které vyžadují komunikaci přes WiFi, práci se soubory a další funkce.

ESP8266WiFi:

Tato knihovna umožňuje připojení a správu WiFi sítě na čipu ESP8266. Poskytuje funkce pro připojení k WiFi síti, získání IP adresy, nastavení režimu (klient nebo přístupový bod) a další operace související s WiFi.

WiFiUdp:

Tato knihovna umožňuje komunikaci přes UDP (User Datagram Protocol) pomocí WiFi na čipu ESP8266. UDP je lehký protokol pro odesílání zpráv (datagramů) mezi zařízeními v síti bez navazování spojení.

FS (File System):

Knihovna pro práci s file systémem na ESP8266. Umožňuje čtení a zápis souborů na flash paměť zařízení. FS je obecná knihovna a může být použita pro různé souborové systémy jako je SPIFFS nebo LittleFS.

NTPClient:

Tato knihovna je určena pro synchronizaci času pomocí NTP (Network Time Protocol). Umožňuje získávat aktuální čas z NTP serverů přes internet, což je užitečné pro aplikace, které potřebují přesný čas.

E-MailSender:

Knihovna pro odesílání e-mailů pomocí ESP8266. Umožňuje konfiguraci e-mailového účtu a odesílání zpráv na zadané e-mailové adresy. Může být použita pro různé notifikace nebo alarmy v projektech.

time:

Standardní knihovna pro práci s časem v jazyce C. Poskytuje funkce pro získání aktuálního času, konverzi mezi různými formáty času a práci s časovými zónami.

ArduinoJson:

Knihovna pro práci s JSON (JavaScript Object Notation) na platformě Arduino. Umožňuje snadné vytváření, parsování a manipulaci s JSON daty, což je užitečné pro komunikaci s webovými službami a API.

Wire:

Tato knihovna umožňuje komunikaci přes I2C (Inter-Integrated Circuit) sběrnici. I2C je sériová komunikační sběrnice, která umožňuje připojení více zařízení (např. senzory, displeje) k mikrokontroléru pomocí dvou vodičů (SDA a SCL).

LiquidCrystal_I2C:

Knihovna pro ovládání LCD displejů, které jsou připojeny přes I2C sběrnici. Umožňuje snadné zobrazení textu na LCD displejích s použitím pouze dvou vodičů pro komunikaci.

6.2 Hlavní část programu

Program je psaný v prostředí Arduino IDE ve verzi 1.8.19. Hlavní část kódu je zpracována jako vývojový diagram v příloze A.1. Kód začíná načtením knihoven, které jsou popsány v kapitole knihovny 6.1. Poté následuje část, kde se provádí definice všech potřebných proměnných, jako jsou přihlašovací údaje k WiFi síti nebo údaje pro přístup k e-mailovým účtům. Další proměnné definují NTP server. Pro tuto práci byl zvolen NTP server `europa.pool.ntp.org`. V kódu je tento klient specifikován dvěma parametry. První parametr posouvá čas do požadované časové zóny a druhý určuje, po jakém intervalu se má čas aktualizovat. Program dále pokračuje definicí adresy k displeji 5.3. Je nutné také nadefinovat pin, který snímá stav baterie(A0), a který bude číst, jestli je zařízení napájeno ze sítě(pin D5). Další část definuje, jak bude probíhat UDP přenos a nakonec následují různé časové proměnné pro správné odesílání dat. Po všech konfiguracích a definicích různých proměnných následuje funkce *setup* 6.2.1, která se vykoná jen jednou při spuštění zařízení nebo při jeho restartu. Je určena k inicializaci proměnných, nastavování režimů pinů, startování komunikace a dalších nastavení, která potřebujeme provést před hlavním cyklem

programu. A na funkci *setup* pak navazuje funkce *loop* 6.2.2, která běží stále dokola v nekonečné smyčce.

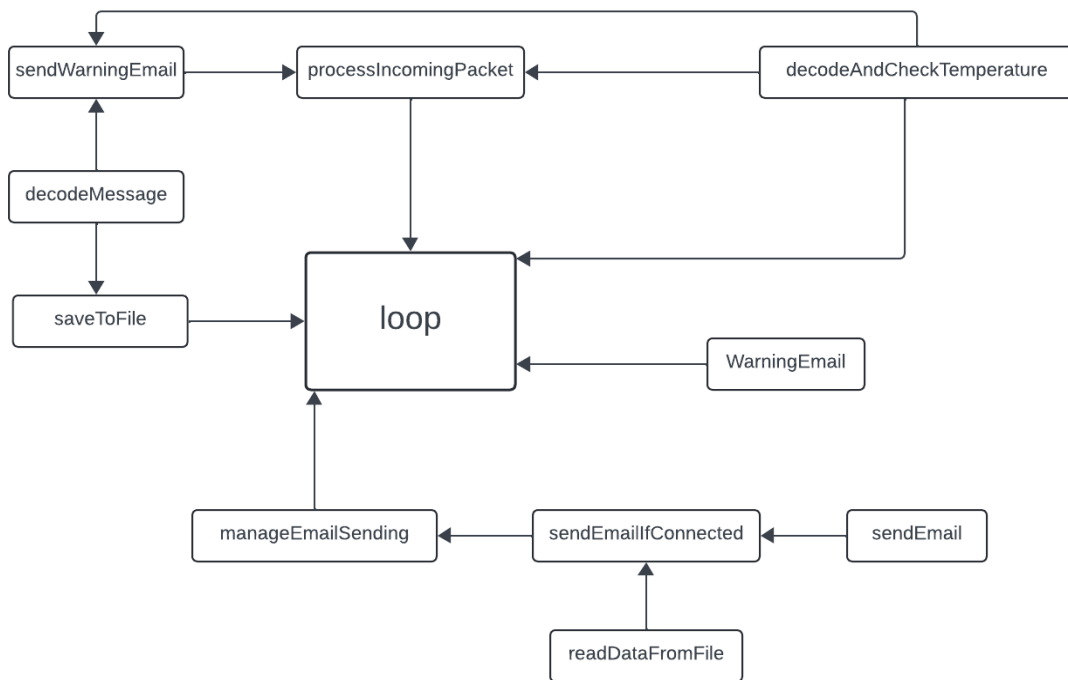
6.2.1 Funkce *setup* C.1

Funkce začíná inicializací sériového monitoru s rychlostí přenosu dat 115200. Pomocí sériového monitoru se odesílají a přijímají textové zprávy, což je užitečné pro ladění a diagnostiku. Další řádky inicializují LCD displej a zapnou jeho podsvícení. Poté se nastavují piny pro měření napětí baterie (*batteryPin*) a pro detekci zdroje napájení (*powerPin*) jako vstupní. Funkce *pinMode* konfiguruje určený pin jako vstup nebo výstup. Na řádcích kódu 9 až 15 v příloze C.1 se nastavuje statická IP adresa pro zařízení, včetně adresy brány, masky podsítě a DNS serveru. Následně se zařízení pokusí připojit k WiFi síti pomocí zadaného *ssid* (názvu sítě) a hesla. Následuje blok kódu, který kontroluje stav připojení k WiFi. Pokud zařízení není připojeno, čeká půl sekundy a poté vypíše tečku na sériový monitor. Jakmile je zařízení připojeno, vypíše zprávu "Připojeno k WiFi". Poté se inicializuje SPIFFS (SPI Flash File System), UDP komunikace a NTP klient, který synchronizuje čas zařízení s časovým serverem. Tím končí tato funkce a přechází se na hlavní smyčku kódu.

6.2.2 Funkce *loop* C.2

První věc, která se provádí v této funkci, je nastavení kurzoru na začátek prvního řádku LCD displeje. Následně se kontroluje stav připojení k WiFi. Pokud je zařízení připojeno k WiFi, zobrazí se na LCD zpráva "WiFi připojeno". Pokud není připojeno, zobrazí se zpráva "WiFi odpojeno". Poté se aktualizuje čas a kurzor se nastaví na začátek druhého řádku LCD displeje a zobrazí se aktuální čas ve formátu "HH:MM". Program pokračuje přečtením analogové hodnoty napětí baterie z pinu *batteryPin* a převede se na reálné napětí. Současně se detekuje zdroj napájení ze vstupního pinu *powerPin*, kde hodnota 0 znamená napájení z baterie a hodnota 1 napájení z USB. Následují podmínky pro zaslání varování na e-mailovou adresu ohledně napájení a stavu baterie u serveru 6.3.4. První varovný e-mail se posílá, pokud poklesne napětí na baterie pod 2V 7.2.7. Toto varování je ošetřeno tak, aby se neposílalo pořád, ale poslalo se jen jednou za den. Druhé varování ohledně napájení přijde, když je zařízení napájeno z baterií a zároveň je WiFi připojena 7.2.8. To indikuje, že nejspíš došlo k odpojení od síťového zdroje a obsluha je upozorněna každou hodinu, pokud problém přetrvává, aby problém odstranila. A nyní se v kódu vykoná nejdůležitější část, kterou je příjem dat. První se zkontroluje, zda byl přijat nový UDP paket. Pokud ano, přečte se obsah paketu do bufferu *incomingPacket*. Následně se vypíše informace o přijatém paketu a jeho obsah na sériový monitor. Poté se zavolá funkce *saveToFile* 6.3.3 a pak se volá další funkce *decodeAndCheckTemperature*

6.3.1, která zde slouží pro to, aby se aktuální naměřená teplota vypsala na LCD displej. Následně se data proženou přes funkci *processIncomingPacket* 6.3.1, která vyhodnocuje nepříznivé stavy a vyvolává odesílání varovných e-mailů 6.3.4. Poslední funkce, která je volaná, je funkce *manageEmailSending* 6.3.4, která ve spolupráci ještě s dalšími funkcemi má na starosti odesílání sesbíraných dat. Pro jednodušší orientaci, jaká funkce navazuje na kterou, je na obrázku 6.1 vytvořena mapa těchto funkcí, jak na sebe jednotlivě navazují. Tam kde, směřuje šipka, tak to znamená, že se ta funkce volá do toho místa. Například funkce *decodeMessage* se volá do funkce *saveToFile* a *sendWarningEmail*



Obr. 6.1: Schéma návaznosti funkcí

6.3 Pomocné funkce

Tyto funkce pomáhají především udržet v kódu přehlednost a pořádek. Funkce jsou rozděleny na čtyři hlavní bloky, kterými jsou Zpracování dat pro varovné e-maily 6.3.1, Dekódování dat 6.3.2, Uložení a čtení dat 6.3.3, Správa odesílání e-mailů 6.3.4. Tyto bloky jsou rozděleny na samotné funkce, které se různě doplňují nebo se přímo volají do hlavní smyčky, jak jde vidět na obrázku 6.1.

6.3.1 Zpracování dat pro varovné e-maily

processIncomingPacket

Tato funkce analyzuje přijatý paket dat, kontroluje stav teploty, baterie a napájení a v případě problémů odešle varovný e-mail. První, co provádí tato funkce, tak je extrakce hodnot na pozici 24 (indexace od 0), aby zjistila zdroj napájení. Dále volá funkci *decodeAndCheckTemperature* 6.3.1, která dekóduje teplotu z řetězce a vrátí její hodnotu. Výsledek uloží do proměnné `temperature`. Ještě se extrahuje a převede na celé číslo podřetězec s indexem (21, 24) , a tím se získá stav baterie. Pak se zkontroluje, zda je senzor odpojený pomocí funkce *isnan*, která ověřuje, zda je teplota rovna NAN. Výsledek uloží do proměnné `isSensorDisconnected`. Nakonec proběhne kontrola, jestli je teplota v povoleném rozmezí, zda je stav baterie nízký, a zda je odpojený zdroj napájení, což je indikováno hodnotou "3" v proměnné `powerSource`. A pokud je splněna nějaká z podmínek teplotní varování, nízký stav baterie, odpojený zdroj napájení, nebo odpojený senzor, funkce zavolá *sendWarningEmail* a odešle varovný e-mail s údaji.

decodeAndCheckTemperature

Tato funkce má vstupní parametr řetězec znaků, z kterého extrahuje podřetězec od indexu 4 do indexu 7, čímž získá část řetězce, který obsahuje teplotu ve formě textu. Zkontroluje, zda extrahovaný podřetězec je rovný "999". Pokud ano, vrátí hodnotu NAN, což indikuje, že senzor je odpojený. Pokud podřetězec není "999", převede ho na celé číslo, které pak vydělí desíti, aby získal hodnotu teploty s jednou desetinnou čárkou a vrátí vypočítanou hodnotu teploty jako výsledek funkce.

6.3.2 Dekódování dat

decodeMessage

Funkce má za atribut poslaný řetězec znaků. Jako první ověří, zda délka řetězce je přesně 25 znaků. Pokud ne, vrátí prázdný řetězec. Poté postupně dekóduje všechny znaky dle stanoveného protokolu 4.5. U teploty, pokud je hodnota "999", znamená to, že je senzor odpojený a nastaví teplotu na "—". Jinak převede teplotu, aby měla jedno desetinné číslo a před ni dopíše znaménko. Datum se naformátuje do formátu "DD.MM." a čas do formátu "HH:MM:SS". Poslední znak řetězce z číselné hodnoty 1 nastaví jako "sít", jinak na "baterie". Nakonec funkce naformátuje všechny extrahované údaje do výstupního HTML řetězce a vrátí naformátovaný HTML řetězec jako výstup.

6.3.3 Uložení a čtení dat

saveToFile

Tato funkce slouží k uložení dat do souboru ve formátu JSON na paměťovém systému SPIFFS. Když zařízení přijme data, volá tuto funkci s daty jako její argument. Na začátku se vytvoří dynamický JSON dokument s velikostí 8192 bajtů. Funkce zkontroluje, zda soubor `udp_data.json` existuje v paměti SPIFFS. Pokud soubor existuje, otevře se pro čtení a deserializuje se. To znamená, že obsah souboru, který je ve formátu JSON, se převede zpět do objektu "doc" v paměti, se kterým můžeme pracovat. Poté se přijatý řetězec dekóduje 6.3.2, aby se z něj získaly konkrétní hodnoty a hlavně se získají první tři znaky, které slouží jako identifikátor zařízení. V souboru se zjistí, zda již existuje pole pro toto zařízení (`deviceId`) a pokud ne, vytvoří nové pole. Poté přidá dekódovanou zprávu do tohoto pole. Pak se otevře soubor pro zápis, což přepíše jeho obsah. Serializuje se JSON dokument "doc" zpět do formátu JSON a zapíše se do souboru. Tímto způsobem se uloží nová data a soubor se zavře.

readDataFromFile

Funkce *readDataFromFile* slouží k načítání dat ze souboru ve formátu JSON na paměťovém systému SPIFFS. První se vytvoří dynamický JSON dokument s velikostí 8192 bajtů, který bude použit pro načítání a manipulaci s daty ve formátu JSON. Otevře se soubor na zadané cestě (`path`) pro čtení. Načte se obsah souboru a deserializuje se do JSON dokumentu "doc". Vytvoří se prázdný řetězec `data`, do kterého se budou ukládat výsledná data. Následně se projde každý pár klíč-hodnota v JSON objektu. Každý pár klíč-hodnota představuje ID zařízení a odpovídající pole zpráv. Například klíč je identifikátor zařízení (např. "001") a hodnota je pole zpráv pro toto zařízení. Pak se do řetězce `data` přidá identifikace zařízení a pod každý identifikátor se vypíše příslušné hodnoty. Oddělení dat a zařízení je provedeno příkazem `
`. Na konci funkce se vrátí sestavený řetězec, který obsahuje všechna načtená data ze souboru ve formátu vhodném pro zobrazení .

6.3.4 Správa odesílání e-mailů

manageEmailSending

Funkce získává aktuální čas pomocí `timeClient.getEpochTime()` a převádí jej na strukturu `tm` pomocí `localtime`. Dále získá aktuální stav WiFi připojení a uloží ho do proměnné `currentWifiStatus`, která je `true`, pokud je WiFi připojena, a `false`, pokud není. Funkce nastavuje příznak pro odeslání e-mailu po obnovení připojení,

pokud měl být e-mail odeslán. Pokud není WiFi připojena a aktuální čas je po plánovaném čase odeslání (hodina je větší než `emailHour` nebo je hodina stejná a minuta větší než `emailMinute`), nastaví se příznak `attemptEmailAfterReconnect` na `true`. Následuje podmínka pro odeslání e-mailu, která zní, že pokud je WiFi připojena a aktuální čas je přesně v plánovaném čase odeslání e-mailu a e-mail ještě nebyl dnes odeslán nebo je nastaven příznak `attemptEmailAfterReconnect` a e-mail ještě nebyl dnes odeslán, zavolá se funkce `sendEmailIfConnected` 6.3.4. Po úspěšném odeslání se příznaky `emailSentToday` a `attemptEmailAfterReconnect` resetují.

sendEmailIfConnected

Tato funkce načte obsah e-mailu ze souboru `udp_data.json` pomocí funkce `readDataFromFile` 6.3.3. Pak už se jen zkontroluje podmínka, že pokud obsah e-mailu není prázdný, tak se zavolá funkce `sendEmail` 6.3.4, která má za atribut právě tento zkontrolovaný obsah.

sendEmail

Funkce vytvoří nový objekt `EMailSender::EMailMessage` a pojmenuje ho `message`. Pak se nastaví předmět e-mailu a vytvoří se počáteční HTML obsah e-mailu. To znamená, že e-mail je napsaný a naformátovaný pomocí HTML (HyperText Markup Language). To je stejný jazyk, který se používá k tvorbě webových stránek. Použití HTML v e-mailech umožňuje vytvářet vizuálně atraktivní a interaktivní zprávy, které mohou obsahovat různé prvky. Na dalším řádku se přidá hlavička e-mailu, která obsahuje popisky pro jednotlivé sloupce údajů a hned za tím, na nový řádek, se přiloží naměřená data. Řádek `message.message = emailContent` nastaví obsah e-mailu na právě vytvořený HTML obsah `emailContent`. Za pomoci knihovny `EmailSender` se odešle e-mail se všemi nasbíranými hodnotami a jakmile se odešle, tak se vymaže soubor `udp_data.json` a resetují se příznaky `emailSentToday` a `lowBatteryWarningSentToday`.

sendWarningEmail

Na začátku jsou dekodována data, která byla shledána, že překračují určité meze. Výsledek se pak uloží do nové proměnné. Poté jsou z řetězce extrahována důležitá data, jako je teplota a stav napájení, a jsou podrobeny podmínkám, které by měly splňovat. Tím se zjistí, jaké hodnoty jsou mimo stanovené rozsahy a začne se sestavovat e-mail, který sestává z varovné zprávy nebo i více zpráv, jež jsou psané vždy na nový řádek. Pak následuje hlavička, pod kterou se vypíše data do svých kolonek. Toto formátování lze vidět například na obrázku 7.9. Pak už se jen vytvoří instance

`EMailSender::EMailMessage`, nastaví se předmět e-mailu a obsah e-mailu, zavolá se metoda `send` a tím se odešle varovný e-mail na uvedenou e-mailovou adresu.

WarningEmail

Tato funkce je volaná jen v případě, že dojde k nějaké poruše napájení u serverové části a vypisuje jen varovnou hlášku. Nejprve definuje proměnnou `emailMessage` jako instanci `EMailSender::EMailMessage`. Nastaví předmět e-mailu na "Varování" a obsah e-mailu na hodnotu předanou v parametru `message`. Poté zavolá metodu `send` na objektu `emailSender`, čímž odešle e-mail na adresu uloženou v proměnné `mailRecipient`.

7 Ověření funkčnosti zařízení a diskuze

Tato kapitola je zaměřena na testování a shrnutí výsledků. Testování probíhalo v různých podmínkách, jako je třeba vložení měřících stanic se senzory do venkovního prostředí, kde mohou probíhat náhlé změny teplot. Další zařízení bylo vloženo do chladničky a třetí do mrazáku. Poslední snímač byl ponechán v normální místnosti s pokojovou teplotou. Při testování byl nasimulován i výpadek proudu a další nežádoucí okolnosti, které by se mohly v reálném provozu vyskytnout.

7.1 Normální provoz

Při normálním provozu se přijímají zprávy a ukládají se do paměti a jednou za den se odešlou na e-mail, ze kterého se data archivují v tištěné formě.

Identifikátor	Teplota[°C]	Datum	Rok	Čas	Stav baterie[%]	Napájení
Zařízení 001:						
001	+23.1	22.05.	2024	22:16:09	60	sít
001	+3.4	22.05.	2024	22:46:09	60	sít
001	+3.4	22.05.	2024	23:16:09	60	sít
001	+6.6	22.05.	2024	23:46:09	60	sít
001	+2.7	23.05.	2024	00:16:09	60	sít
001	+2.7	23.05.	2024	00:46:09	60	sít
001	+6.2	23.05.	2024	01:16:09	60	sít
Zařízení 002:						
002	+15.8	22.05.	2024	22:17:27	50	sít
002	-19.3	22.05.	2024	22:47:27	50	sít
002	-22.8	22.05.	2024	23:17:27	50	sít
002	-21.0	22.05.	2024	23:47:28	50	sít
002	-21.0	23.05.	2024	00:17:28	50	sít
002	-22.0	23.05.	2024	00:47:28	50	sít
002	-18.8	23.05.	2024	01:17:28	50	sít
Zařízení 003:						
003	+24.2	22.05.	2024	22:18:26	55	sít
003	+24.2	22.05.	2024	22:48:26	55	sít
003	+24.3	22.05.	2024	23:18:26	55	sít
003	+24.2	22.05.	2024	23:48:26	55	sít
003	+24.3	23.05.	2024	00:18:26	55	sít
003	+24.1	23.05.	2024	00:48:26	55	sít
003	+23.9	23.05.	2024	01:18:26	55	sít

Obr. 7.1: Formát e-mailu při příjmu dat ze tří stanic

Na obrázku 7.1 je vidět ukázka, jak zařízení pracuje a vypisuje hodnoty při sběru dat z více zařízení. Pro tento test bylo první zařízení vloženo do chladničky, kde můžeme pozorovat teploty od 2 do 7°C. Jediná první hodnota je vyšší z důvodu, že

zařízení bylo spuštěno hned po vložení, tudíž se nestihla ustálit teplota. Druhé zařízení bylo vloženo do mrazničky. Tam se teploty pohybují kolem -20°C a první měření je stejný případ, jako u lednice. A nakonec bylo provedeno měření při pokojové teplotě okolo 24°C . Všechna zařízení se držela ve stanovených teplotních rozmezích, tudíž jely bez vyvolání jakéhokoli varovného teplotního stavu. Celé denní záznamy, které jsme testovali, jsou v příloze E. Tam se nachází obrázky, které byly pořízeny z e-mailové adresy při testovacím měření, které probíhalo po dobu dvou dnů. První den se poslalo 23 naměřených hodnot. V čase 21:02 nedošlo k správnému příjmu dat, nejspíš kvůli špatnému připojení k WiFi. Druhý den se již poslala všechna data a došlo zde i k překročení teploty u tří měření, které jsou v podobě varovných e-mailů zobrazeny na obrázku E.3

7.2 Provoz při změně podmínek

Pokud dojde k nějakým nežádoucím stavům, tak celé zařízení funguje úplně stejně jako při normálním provozu, až na to, že přejde třeba na jiný zdroj napájení nebo oddálí odeslání e-mailu a nebo jen upozorní na vzniklý problém.

7.2.1 Výpadek WiFi

V případě výpadku WiFi se přestanou posílat data z měřících stanic a ty si v tu chvíli ukládají měřené hodnoty do své paměti a po obnovení spojení jsou v dalším odesílacím čase všechna data poslána a uložena normálně do paměti. Na obrázku 7.2 jde vidět, jak se první tři data posílají po 30 sekundách a pak je odmlka, při které se data jen ukládají u klientské části a po znovupřipojení jde vidět, jak se co sekundu odešlou celkem 4 záznamy, a poté běh zařízení pokračuje, jako na začátku.


```

Aktuální čas: 21:28:43
Obsah UDP paketu: 0010244220520242128410591
Aktuální čas: 21:29:13
Obsah UDP paketu: 0010244220520242129120591
Aktuální čas: 21:29:43
Obsah UDP paketu: 0010243220520242129420601
Aktuální čas: 21:32:14
Obsah UDP paketu: 0010243220520242130130601
Aktuální čas: 21:32:15
Obsah UDP paketu: 0010243220520242130490601
Aktuální čas: 21:32:16
Obsah UDP paketu: 0010243220520242131190601
Aktuální čas: 21:32:17
Obsah UDP paketu: 0010243220520242131490611
Aktuální čas: 21:32:18
Obsah UDP paketu: 0010243220520242132120611
Aktuální čas: 21:32:44
Obsah UDP paketu: 0010243220520242132420611
Aktuální čas: 21:33:14
Obsah UDP paketu: 0010244220520242133120611

```

Obr. 7.2: Výpis ze sériového monitoru při testu výpadku wifi

Identifikátor	Teplota[°C]	Datum	Rok	Čas	Stav baterie[%]	Napájení
Zařízení 001:						
001	+24.4	22.05.	2024	21:28:41	59	sít
001	+24.4	22.05.	2024	21:29:12	59	sít
001	+24.3	22.05.	2024	21:29:42	60	sít
001	+24.3	22.05.	2024	21:30:13	60	sít
001	+24.3	22.05.	2024	21:30:49	60	sít
001	+24.3	22.05.	2024	21:31:19	60	sít
001	+24.3	22.05.	2024	21:31:49	61	sít
001	+24.3	22.05.	2024	21:32:12	61	sít
001	+24.3	22.05.	2024	21:32:42	61	sít
001	+24.4	22.05.	2024	21:33:12	61	sít

Obr. 7.3: Ukázka e-mailu při testu výpadku wifi

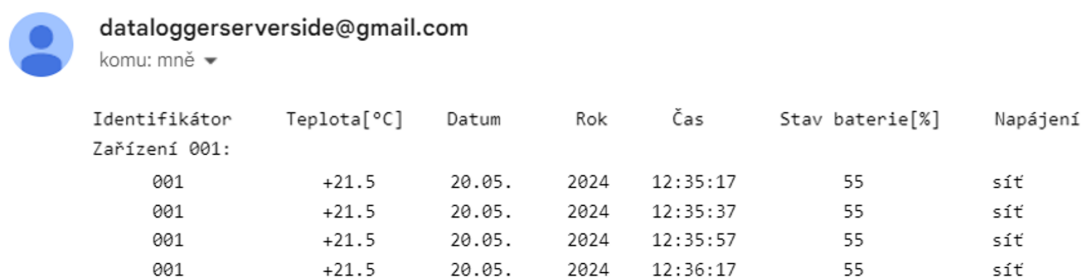
7.2.2 Výpadek WiFi při posílání e-mailu

Jakmile dojde k výpadku sítě v momentě, kdy by se měl posílat denní souhrn naměřených hodnot, tak je kód ošetřen podmínkou, že pokud nedošlo ten den ještě k odeslání, ale čas už překročil odesílací dobu, tak po opětovném připojení se odešle e-maile se všemi nasbíranými daty. Na obrázku 7.4 je znázorněn výpis ze sériového monitoru, kde vidíme poslední poslaná data, předtím než došlo k odpojení od WiFi sítě. Čas odeslání e-mailu byl nastaven na 12:38 hodin a jak jde vidět, e-mail se odešlal až po znovupřipojení k WiFi, ke kterému došlo někdy po dosažení času 12:39

a následně se poslal i samotný e-mail s daty. Pro toto testování byl zvolen interval měření a posílání dat 20 sekund. Nabízí se tu otázka, proč nejsou v e-mailu poslána i data, která byla naměřena i při odpojení od internetu (tento mechanismus je vidět na obrázku 7.2). A odpovědí je, že naměřená data u klientské části se posílají až při dalším měřicím cyklu. V takových časových intervalech jako je 20 sekund to není ideální, protože může chybět hodně hodnot ten den, ale jelikož se sbírají data každou hodinu, tak v tom není problém, že budou chybět třeba dvě měření, která se stejně pro účely archivace odešlou v dalším každodenním e-mailu.

```
Aktuální čas: 12:36:17
Obsah UDP paketu: 0010215200520241236170551
E-mail byl úspěšně odeslán
Aktuální čas odeslání emailu: 12:39:28
```

Obr. 7.4: Ukázka sériového monitoru, když je výpadek WiFi při odesílání e-mailu



The screenshot shows an email interface with a profile icon and the sender's name 'dataloggerserverside@gmail.com'. Below the header is a table with the following data:

Identifikátor Zařízení 001:	Teplota[°C]	Datum	Rok	Čas	Stav baterie[%]	Napájení
001	+21.5	20.05.	2024	12:35:17	55	sít
001	+21.5	20.05.	2024	12:35:37	55	sít
001	+21.5	20.05.	2024	12:35:57	55	sít
001	+21.5	20.05.	2024	12:36:17	55	sít

Obr. 7.5: Ukázka e-mailu, když je výpadek WiFi při odesílání e-mailu

7.2.3 Výpadek napájení ze sítě

Při celkovém výpadku proudu chladicí zařízení přejdou na záložní zdroj a senzory jedou na bateriové napájení, ale nedochází k posílání dat, protože je přerušena internetová komunikace. V tomto případě se data u měřicích zařízeních ukládají do vlastní paměti a po znovupřipojení k WiFi dojde k poslání všech nahromaděných dat. Oproti jen výpadku Wifi 7.2.1 je zde rozdíl v datech, že místo napájení ze sítě, se po dobu výpadku přepne zařízení na bateriové napájení. V e-mailu 7.6 jde zřetelně vidět v kolonce napájení, že došlo k výpadku celkem na tři měřicí cykly. To samé jde vidět u sériového monitoru 7.7, ale tam je spíš zřetelně naznačeno, jak se data ukládala a po znovupřipojení se naráz odeslala.

Identifikátor	Teplota[°C]	Datum	Rok	Čas	Stav baterie[%]	Napájení
Zařízení 001:						
001	+24.7	22.05.	2024	21:49:02	53	síť
001	+24.7	22.05.	2024	21:49:32	55	síť
001	+24.8	22.05.	2024	21:50:10	31	baterie
001	+24.9	22.05.	2024	21:50:40	30	baterie
001	+24.8	22.05.	2024	21:51:10	29	baterie
001	+24.8	22.05.	2024	21:51:33	53	síť
001	+24.8	22.05.	2024	21:52:03	54	síť

Obr. 7.6: Ukázka e-mailu při testu výpadku elektrické sítě

```

Aktuální čas: 21:49:04
Obsah UDP paketu: 0010247220520242149020531
Aktuální čas: 21:49:34
Obsah UDP paketu: 0010247220520242149320551
Aktuální čas: 21:51:34
Obsah UDP paketu: 0010248220520242150100310
Aktuální čas: 21:51:35
Obsah UDP paketu: 0010249220520242150400300
Aktuální čas: 21:51:36
Obsah UDP paketu: 0010248220520242151100290
Aktuální čas: 21:51:37
Obsah UDP paketu: 0010248220520242151330531
Aktuální čas: 21:52:04
Obsah UDP paketu: 0010248220520242152030541

```

Obr. 7.7: Výpis ze sériového monitoru při testu výpadku elektrické sítě

7.2.4 Pokles či růst teplot

S každými přijatými daty se vždy kontroluje podmínka, jestli je teplota ve vymezeném rozmezí. Toto rozmezí je nastaveno na dvě různé úrovně, pokud by se přijímala data z různých skladovacích prostor. Standardně se uchovávají léčiva při teplotách od 2 do 8°C nebo při pokojové teplotě 15 až 25°C a poté také určité typy léčiv v mrazničkách, kde se pohybujeme v teplotách pod -15°C . Na obrázku 7.8 jdou vidět tři varovné e-maily pro různé teploty.



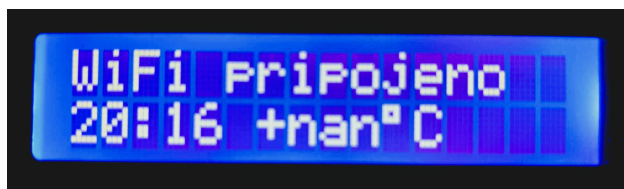
Obr. 7.8: Ukázka varovného e-mailu při různých teplotních výkyvech

7.2.5 Vypojený senzor

Když se vypojí senzor pro měření teploty, tak v posílané zprávě se zobrazí místo teploty, číselná kombinace tří devítek, což vyvolá podmínku pro odeslání varovného e-mailu, a tím upozorní obsluhu na vzniklé potíže. Na obrázku 7.9 jde vidět v jaké formě se posílají data, pokud došlo právě k odpojení nebo nefunkčnosti senzoru. Na obrázku 7.10 je výpis, který se zobrazí na displeji při této chybě¹.



Obr. 7.9: Varovný e-mail při odpojení senzoru



Obr. 7.10: Displej při odpojení senzoru

¹nan, angl. *Not a Number* je stavová chyba vyjadřující hodnotu mimo číselný datový formát.

7.2.6 Slabá baterie u klienta

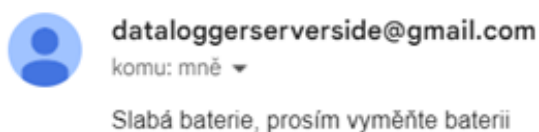
V protokolu je zasílána procentuální hodnota záložní baterie a je určena určitá hranice, kdy dojde k odeslání varovného e-mailu, který upozorní na nutnost výměny těchto baterií.



Obr. 7.11: Ukázka e-mailu při detekci nízkého stavu baterie

7.2.7 Slabá baterie u serveru

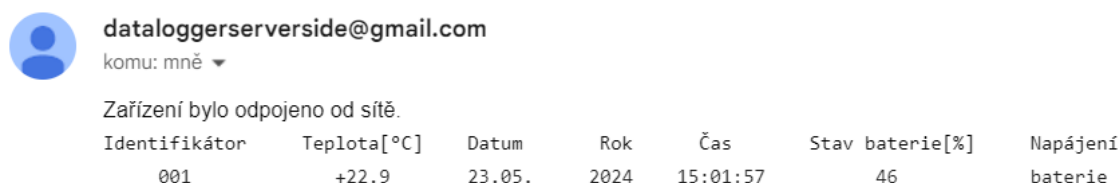
U serverové části se nikde pravidelně nevypisuje stav baterie, ale je jen monitorováno napětí, a pokud baterie klesne pod napětí 2V, tak se na monitorovací e-mail odešle varování o nízkém stavu baterie a daná obsluha by měla provést její výměnu. Práh pro poslání varování je nastaven tak, aby byla dostatečně dlouhá doba na výměnu baterií.



Obr. 7.12: Varovný e-mail při poklesu napětí na záložním napájení serveru

7.2.8 Náhodné odpojení od síťového napájení

Občas se stane, že při neopatrné manipulaci může dojít k odpojení napájení ze síťového adaptéru, a proto jsou kódy ošetřeny tak, aby v případě tohoto náhodného odpojení upozornily na vzniklý problém.



Obr. 7.13: Varovný e-mail při odpojení síťového napájení u klientské části



dataloggerserverside@gmail.com

komu: mně ▾

Zařízení bylo odpojeno od sítě, zkontrolujte správnost připojení

Obr. 7.14: Varovný e-mail při odpojení síťového napájení u serverové části

7.3 Diskuze

Tato diskuze je zaměřena na hodnocení a analýzu výsledků testování jednotlivých zařízení v různých provozních podmínkách. Testy byly navrženy tak, aby simulovaly různé scénáře, které mohou nastat během reálného provozu, a umožnily posoudit spolehlivost měřicích stanic se senzory, a také samotný server pro ukládání a zasílání dat. Během normálního provozu server sbírá data a odesílá je na určený e-mail pro archivaci. První zařízení, umístěné v chladničce, vykazovalo teploty v rozmezí 2–7°C, což odpovídá očekáváním. Podobně zařízení v mrazničce správně měřilo teploty kolem -20°C. Měření při pokojové teplotě (okolo 24°C) rovněž odpovídalo předpokladům. Z tohoto testu vyplývá, že všechna zařízení fungují správně v rámci stanovených teplotních rozmezí. Zařízení bylo testováno i při různých nežádoucích stavech, jako jsou výpadky WiFi, výpadky napájení a teplotní výkyvy. V případě výpadku WiFi se ukázalo, že měřicí zařízení správně ukládají data do paměti a po obnovení spojení je odesílají. Podobně při výpadku napájení zařízení přešla na záložní bateriový zdroj a po obnovení napájení správně poslaly nasbíraná data. Během testů bylo zjištěno, že zařízení správně reagují na kritické situace jako jsou poklesy či růsty teplot mimo nastavené rozmezí, odpojení senzoru, slabé baterie u klienta i serveru, a náhodné odpojení od síťového napájení. Každý z těchto stavů byl správně detekován a zařízení odeslalo varovné e-maily, čímž byla zajištěna informovanost obsluhy o vzniklých problémech. Celkově lze říci, že měřicí stanice se senzory prokázaly určitou míru spolehlivosti a schopnosti přizpůsobit se různým provozním podmínkám. I přes simulaci různých nežádoucích stavů zůstala funkčnost zařízení nezměněna a byly schopny obnovit svůj normální provoz po vyřešení vzniklých problémů. Ale při některých měřeních bylo zjištěno, že ve správném fungování zařízení hraje velkou roli síla a stabilita WiFi připojení, jelikož se přes internet aktualizuje čas, posílají se data a hlavně e-maily. Při špatném WiFi připojení se stávalo, že když se zařízení není schopno připojit k NTP serverům, tak nastaví výchozí čas na 1.ledna v 00:00:00 hodin a rok 1970. To způsobí neodesílání e-mailů v daný čas a v posílaných datech bude špatný čas, kdy byla data naměřena, tudíž nejsou vhodná pro archivaci. Pokud se ale i přes špatné připojení aktualizuje správně čas, tak je problém v zasílání e-mailu, který se odesílá dlouho a stalo se, že odeslán

selhalo. V tom případě však data zůstanou uložena a pošlou se další den. Právě při této události, kdy se sbírají data a neodešlou se ten den, ale až při dalším cyklu, byla zjištěna největší vada zařízení a to ta, že je serverová část schopna ukládat velký počet dat, ale jen do určitého objemu se odešle e-mail a jinak při překročení počtu nasbíraných dat nedochází k odeslání. Tento problém byl konzultován a byly vyzkoušeny různé metody pro zlepšení, kdy došlo k jistému zlepšení, ale na úkor čitelnosti dat pro uživatele. Ale jinak zařízení pracuje spolehlivě. Jako možné zlepšení se nabízí přidat větší displej, aby bylo zobrazováno více dat přímo na místě, a také připadá v úvahu uživatelská aplikace na mobilní telefon, přes kterou by se mohly bez nutnosti upravovat kód a přehrávat zařízení měnit určité parametry. A hlavně do této aplikace by se mohla odesílat data přímo a odpadl by problém s posíláním velkého množství dat, a poté by se jen jednou za čas data stáhla a vytiskla pro papírovou archivaci.

Závěr

Tato práce se zaměřila na návrh a implementaci klient-server zařízení pro monitorování teploty léčiv s ohledem na požadavky legislativy SÚKL. V teoretické části byl prostudován legislativní rámec týkající se uchovávání léčiv a jeho specifické požadavky na monitorování teploty. Hlavním cílem práce bylo navrhnout a sestavit zařízení, které by bylo schopno automaticky zaznamenávat teplotu prostředí léčiv a ukládat tato data a následně posílat ke zpracování na server, odkud by putovala na zvolenou e-mailovou adresu pro následnou kontrolu a archivaci. Praktická část práce se soustředila na testování zařízení v různých podmínkách. Testování prokázalo, že navržený systém je schopen spolehlivě fungovat v různých prostředích, včetně chladniček, mrazniček a běžných pokojových teplot. Zařízení správně reagovalo na změny podmínek, jako jsou výpadky napájení, výpadky WiFi připojení, a teplotní výkyvy, čímž se potvrdila jeho robustnost a spolehlivost. Zařízení splňuje všechny legislativní požadavky na monitorování teploty léčiv, což je klíčové pro jejich správné uchovávání a zajištění jejich účinnosti. Implementace varovných e-mailů, v případě detekce nežádoucích stavů, poskytuje rychlou a efektivní možnost reakce na vzniklé problémy, což je nezbytné pro minimalizaci rizik spojených s nesprávným uchováváním léčiv. V budoucnosti by bylo vhodné pokračovat v optimalizaci zařízení, aby bylo možné uchovávat větší počet dat, a tím pádem by zařízení bylo odolné vůči všem nežádoucím situacím. Celkově lze konstatovat, že dosažené výsledky odpovídají cílům práce a navržené zařízení představuje efektivní řešení pro monitorování teploty léčiv v souladu s legislativními požadavky.

Literatura

- [1] 1487765909: LCD displej 16X2 znaků s I2C rozhraním. Online. ECLIPSERA, 2021.
- [2] ALANI, Mohammed M. a ALANI, Mohammed M. TCP/IP Model. Online. In: *Guide to OSI and TCP/IP Models*. SpringerBriefs in Computer Science. Cham: Springer International Publishing, 2014, s. 19-50. ISBN 978-3-319-05151-2. Dostupné z: https://doi.org/10.1007/978-3-319-05152-9_3. [cit. 2023-11-28].
- [3] *Arduino.cc*. Online. ©2023. Dostupné z: <https://www.arduino.cc/>. [cit. 2023-12-28].
- [4] ASHISH, B. Temperature monitored IoT based smart incubator. Online. In: *2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*. IEEE, 2017, s. 497-501. ISBN 978-1-5090-3242-6. Dostupné z: <https://doi.org/10.1109/I-SMAC.2017.8058400>. [cit. 2023-12-03].
- [5] ČESKÁ REPUBLIKA. Vyhláška č. 229/2008 Sb.: vyhláška o výrobě a distribuci léčiv. In: *Sbírka zákonů*. 2008, 73/2008.
- [6] *DIS-15 verze 4 Sledování a kontrola teploty při skladování a přepravě léčiv*. Online. 4. SÚKL, 2019.
- [7] ECLIPSERA S.R.O. *Drátek.cz*. Online. Drátek.cz. 2016. Dostupné z: <https://dratek.cz/arduino/1457-esp8266-esp-12e-ota-wemos-d1-ch340-wifi.html>. [cit. 2024-04-26].
- [8] *ESP-WROOM-02D/02U: Datasheet*. Online. In: ESPRESSIF. © 2023. Dostupné z: https://www.espressif.com/sites/default/files/documentation/esp-wroom-02u_esp-wroom-02d_datasheet_en.pdf. [cit. 2023-12-05].
- [9] KODALI, Ravi Kishore a SAHU, Archana. An IoT based weather information prototype using WeMos. Online. In: *2016 2nd International Conference on Contemporary Computing and Informatics (IC3I)*. IEEE, 2016, s. 612-616. ISBN 978-1-5090-5256-1. Dostupné z: <https://doi.org/10.1109/IC3I.2016.7918036>. [cit. 2023-12-03].
- [10] LANGER, Martin; TEICHEL, Kristof; SIBOLD, Dieter a BERMBACH, Rainer. Time synchronization performance using the network time security protocol. Online. In: *2018 European Frequency and Time Forum (EFTF)*. IEEE,

- 2018, s. 138-144. ISBN 978-1-5386-5620-4. Dostupné z: <https://doi.org/10.1109/EFTF.2018.8409017>. [cit. 2023-12-30].
- [11] *Low drop power Schottky rectifier: 1N5817, 1N5818, 1N5819*. Online. 5. STMicroelectronics, 2011.
- [12] MESQUITA, Joao; GUIMARAES, Diana; PEREIRA, Carlos; SANTOS, Frederico a ALMEIDA, Luis. Assessing the ESP8266 WiFi module for the Internet of Things. Online. In: *2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA)*. IEEE, 2018, s. 784-791. ISBN 978-1-5386-7108-5. Dostupné z: <https://doi.org/10.1109/ETFA.2018.8502562>. [cit. 2023-12-03].
- [13] O léčích.cz: Encyklopedie SÚKL. Podmínky skladování léků [online]. [cit. 2024-4-15]. Dostupné z: <http://www.olecich.cz/encyklopedie/podminky-skladovani-lecivych-pripravku>.
- [14] PARIDA, Debapriya; BEHERA, Asish; NAIK, Jagdish Kumar; PATTANAIK, Soumyaranjan a NANDA, Rajat Sekhar. Real-time Environment Monitoring System using ESP8266 and ThingSpeak on Internet of Things Platform. Online. In: *2019 International Conference on Intelligent Computing and Control Systems (ICCS)*. IEEE, 2019, s. 225-229. ISBN 978-1-5386-8113-8. Dostupné z: <https://doi.org/10.1109/ICCS45141.2019.9065451>. [cit. 2023-12-03].
- [15] POLOLU. *Pololu 3.3V Step-Up Voltage Regulator U1V11F3*. Online. POLOLU. Pololu robotics & electronics. 2001, 2024. Dostupné z: <https://www.pololu.com/product/2561/pictures>. [cit. 2024-05-18].
- [16] SCHWARTZ, Marco. Internet of Things with ESP8266. In: *Internet of Things with ESP8266*. Packt, 2016, s. 1-13. ISBN 9781786468024.
- [17] Státní ústav pro kontrolu léčiv. VYR-32: kapitola 3 verze 4 [online]. [cit. 2024-4-24]. Dostupné z: <https://www.sukl.cz/leciva/vyr-32-kap-3-verze-4>.
- [18] Státní ústav pro kontrolu léčiv. VYR-32: verze 3 kapitola 4 [online]. [cit. 2024-4-26]. Dostupné z: <https://www.sukl.cz/leciva/vyr-32-verze-3-kapitola-4>.
- [19] SURESWARAN, R.; BAZAR, Hussein Al; ABOUABDALLA, O.; MANASRAH, Ahmad M. a EL-TAJ, Homam. Active e-mail system SMTP protocol monitoring algorithm. Online. In: *2009 2nd IEEE International Conference on Broadband Network and Multimedia Technology*. IEEE, 2009, s. 257-260.

ISBN 978-1-4244-4590-5. Dostupné z: <https://doi.org/10.1109/ICBNMT.2009.5348490>. [cit. 2023-12-30].

- [20] Tcp-ip model in data communication and networking. *American Journal of Engineering Research*. 2015, roč. 4, č. 10, s. 102-107. ISSN e-ISSN: 2320-0847 p-ISSN : 2320-0936.
- [21] *Transport Layer Protocols*. Online. Geeksforgeeks. 2008, 2024-01-25. Dostupné z: <https://www.geeksforgeeks.org/transport-layer-protocols/>. [cit. 2024-05-28].
- [22] TR INSTRUMENTS SPOL. S R.O. *Naše řešení pro: měření teplot pro zdravotnické aplikace*. Online. TR INSTRUMENTS SPOL. S R.O. TR instruments spol. s r.o. měřicí přístroje a systémy. © 2024, 2024. Dostupné z: <https://www.trinstruments.cz/komplexni-dodavky/teploty-nemocnice>. [cit. 2024-05-24].
- [23] Vyhláška č. 84/2008 Sb.: Vyhláška o správné lékařské praxi, bližších podmínkách zacházení s léčivý v lékárnách, zdravotnických zařízeních a u dalších provozovatelů a zařízení vydávajících léčivé přípravky. In: *Sbírka zákonů*. 2008, 25/2008, s. 14-17.
- [24] *Weock*. Online. Weock: What is NTP/SNTP Protocols, How NTP works. © 2023. Dostupné z: <https://iot.samteck.net/>. [cit. 2023-12-28].

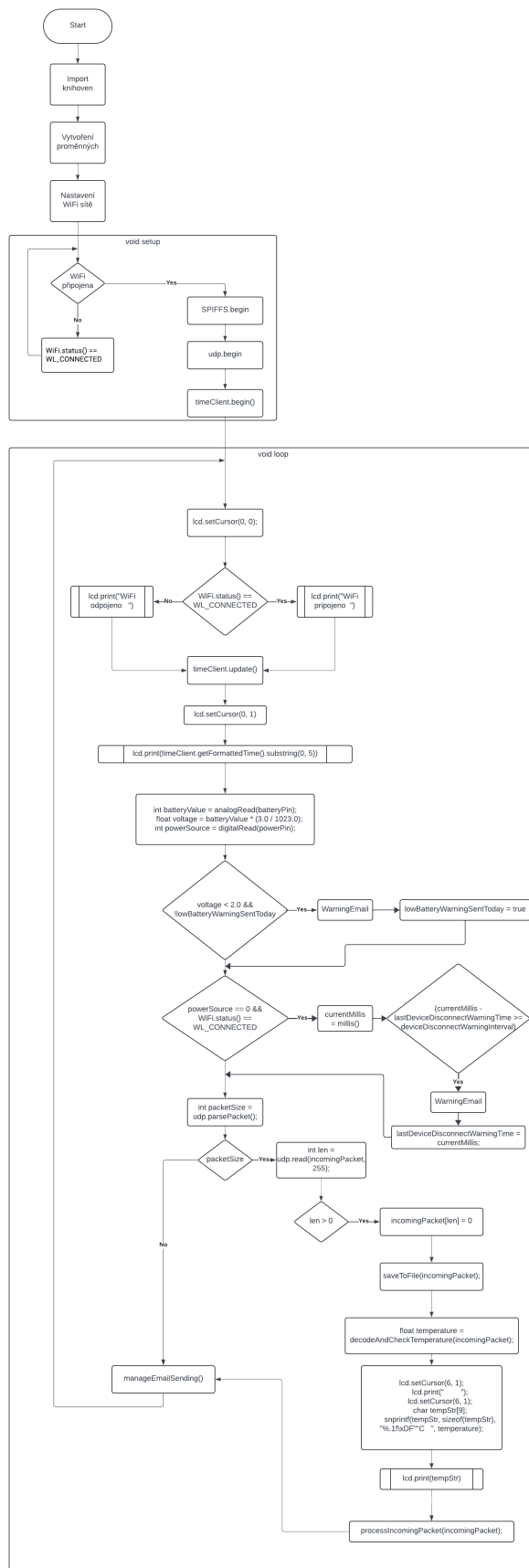
Seznam symbolů a zkratek

SÚKL	Státní Ústav pro kontrolu léčiv
UDP	User Datagram Protocol – Uživatelský datagramový protokol
NTP	Network Time Protocol – Síťový protokol času
SMTP	Simple Mail Transfer Protocol – Jednoduchý protokol pro přenos elektronické pošty
TCP	Transmission Control Protocol – Protokol řízení přenosu
IP	Internet Protocol – Internetový protokol
USB	Universal Serial Bus – Univerzální sériová sběrnice
POP3	Post Office Protocol version 3 – Protokol poštovní kanceláře verze 3
IMAP	Internet Message Access Protocol – Internetový protokol pro přístup k zprávám
JSON	JavaScript Object Notation – Javascriptová objektová notace
I2C	Inter Integrated Circuit – Interní sériová sběrnice
API	Application Programming Interface – Rozhraní pro programování aplikací
ADC	Analog-to-digital converter – Analogově-digitální převodník
GPIO	General Purpose Input/Output – Všeobecné vstupy/výstupy
SPIFFS	SPI Flash File System – SPI Flash File System
SPI	Serial Peripheral Interface – Sériové periferní rozhraní
HTML	HyperText Markup Language – Hypertextový značkový jazyk

Seznam příloh

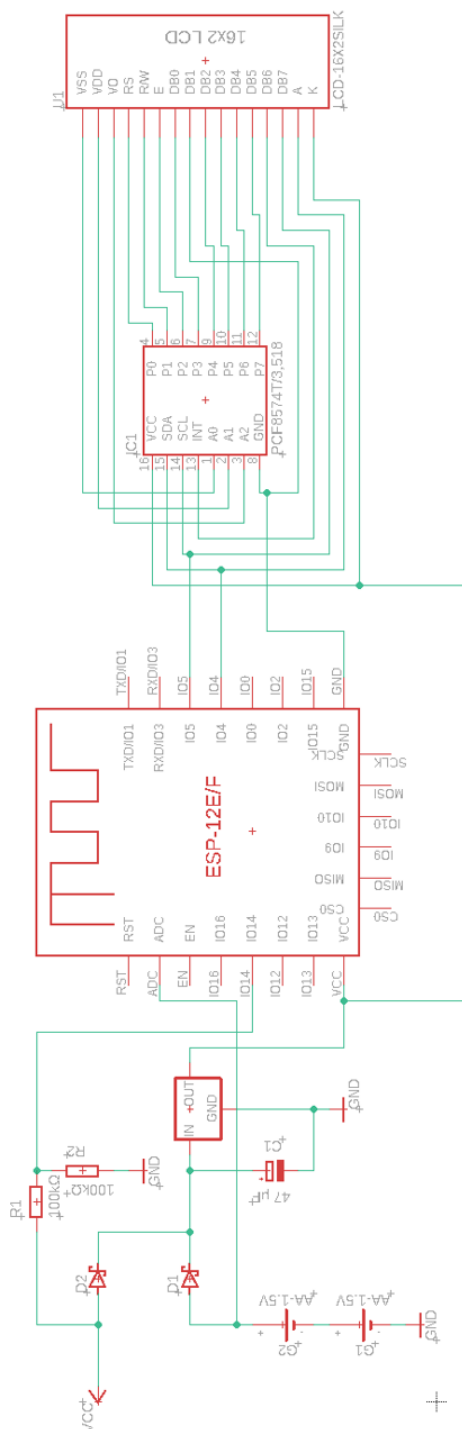
A	Vývojový diagram	54
B	Schéma zapojení hardwaru	56
C	Příklady zdrojových kódů	57
C.1	Výpis funkce setup	57
C.2	Výpis části funkce loop	58
D	Fotografie výsledného zapojení	59
E	Ukázka výsledků měření	61
F	Obsah elektronické přílohy	63

A Vývojový diagram



Obr. A.1: Vývojový diagram

B Schéma zapojení hardwaru



Obr. B.1: Elektrické schéma zapojení celého zařízení

Mikrokontrolér ESP8266 je ve schématu uveden jako funkční blok s připojením jednotlivých pinů.

C Příklady zdrojových kódů

C.1 Výpis funkce setup

Výpis C.1: funkce setup

```
void setup() {
  Serial.begin(115200);
  lcd.init();
  lcd.backlight();

  pinMode(batteryPin, INPUT);
  pinMode(powerPin, INPUT);

  IPAddress staticIP(192, 168, 137, 243);
  IPAddress gateway(192, 168, 137, 1);
  IPAddress subnet(255, 255, 255, 0);
  IPAddress dns(8, 8, 8, 8);

  WiFi.config(staticIP, gateway, subnet, dns);
  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("\nPřipojeno k Wi-Fi");
  if (SPIFFS.begin()) {
    Serial.println("SPIFFS inicializováno");
  } else {
    Serial.println("Selhalo inicializování SPIFFS");
    return;
  }

  udp.begin(localUdpPort);

  timeClient.begin();
  timeClient.update();
}
```

C.2 Výpis části funkce loop

Výpis C.2: Část funkce loop bez snímání napájení

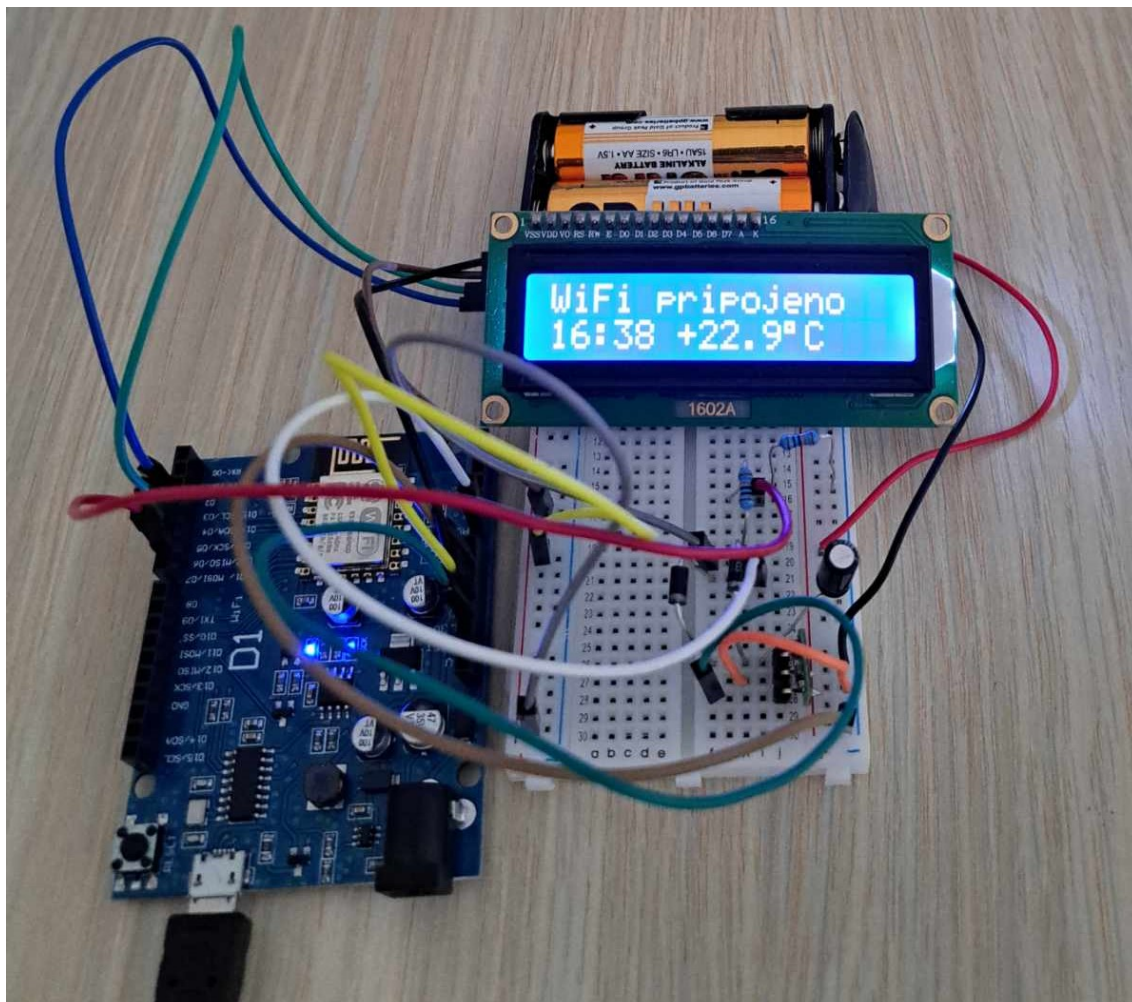
```
void loop() {
    lcd.setCursor(0, 0);
    if (WiFi.status() == WL_CONNECTED) {
        lcd.print("WiFi_pripojeno_");
    } else {
        lcd.print("WiFi_odpojeno_");
    }
    timeClient.update();
    lcd.setCursor(0, 1);
    lcd.print(timeClient.getFormattedTime().substring(0, 5));
    lcd.print("_");
    int packetSize = udp.parsePacket();
    if (packetSize) {
        int len = udp.read(incomingPacket, 255);
        if (len > 0) {
            incomingPacket[len] = 0;

            Serial.printf("Obsah_UDP_paketu:%s\n", incomingPacket);

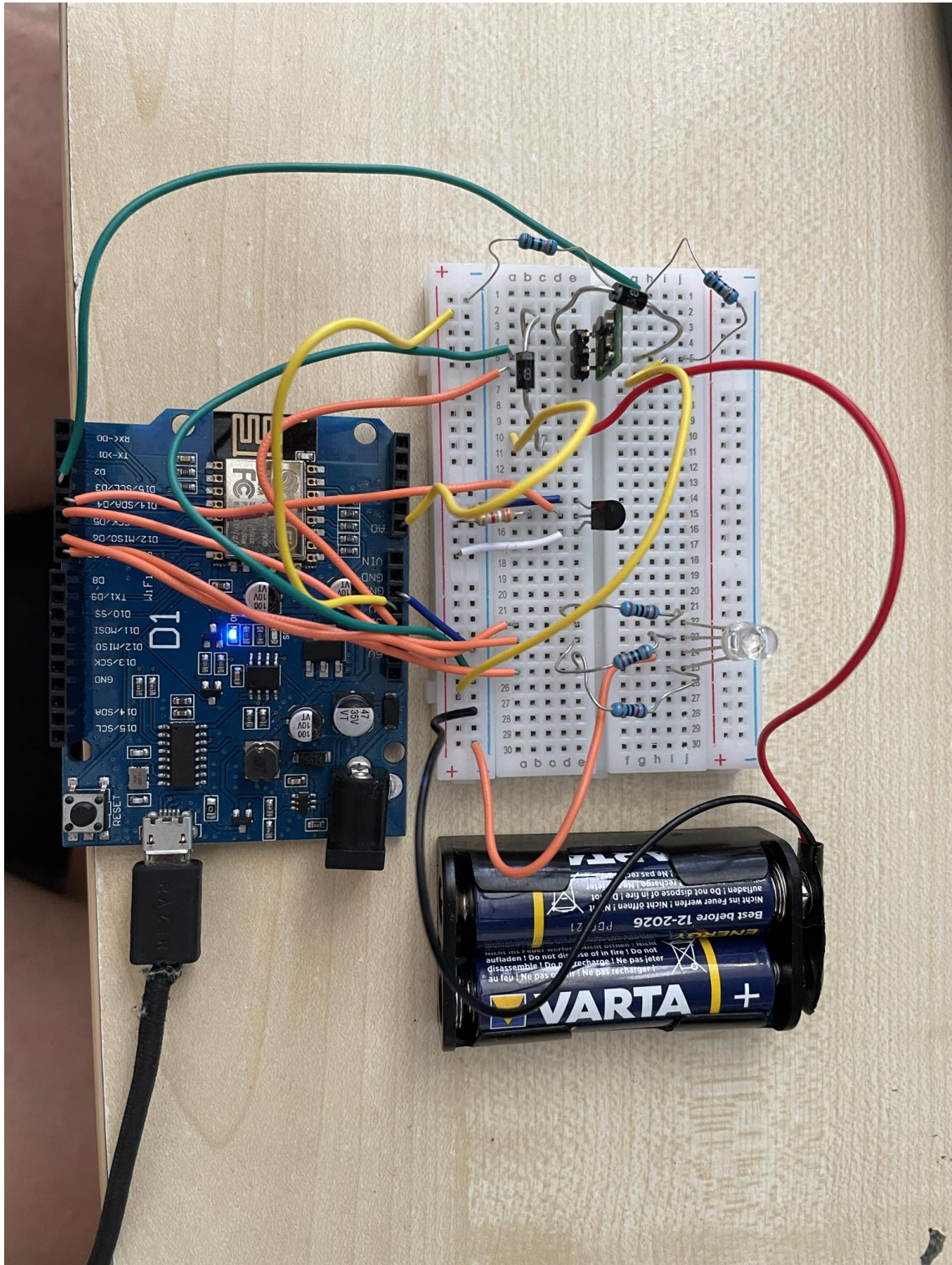
            saveToFile(incomingPacket);
            Serial.println("Soubor_ulozen");
            float temperature =
            decodeAndCheckTemperature(incomingPacket);
            lcd.setCursor(6, 1);
            lcd.print("          ");
            lcd.setCursor(6, 1);
            char tempStr[9];
            char sign = incomingPacket[3] == '1' ? '-' : '+';
            snprintf(tempStr, sizeof(tempStr), "%c%.1f\xDF"C_");
            sign, fabs(temperature));
            lcd.print(tempStr);
            processIncomingPacket(incomingPacket);
        }
    }
    manageEmailSending();
}
```

D Fotografie výsledného zapojení

V příloze se nachází dvě fotky, které ukazují jak vypadá výsledné zapojení obou částí.



Obr. D.1: Zapojení serverové části



Obr. D.2: Zapojení klientské části

E Ukázka výsledků měření

V této příloze se nachází obrázky, které byly pořízeny z e-mailové adresy, při testovacím měření, které probíhalo dva dny.



dataloggerserverside@gmail.com


komu: mně ▾

Identifikátor	Teplota[°C]	Datum	Rok	Čas	Stav baterie[%]	Napájení
Zařízení 001:						
001	+23.1	23.05.	2024	17:02:46	60	sít
001	+24.5	23.05.	2024	18:02:46	60	sít
001	+24.0	23.05.	2024	19:02:46	60	sít
001	+24.1	23.05.	2024	20:02:46	60	sít
001	+23.4	23.05.	2024	22:02:47	60	sít
001	+23.4	23.05.	2024	23:02:47	60	sít
001	+23.5	24.05.	2024	00:02:47	60	sít
001	+23.4	24.05.	2024	01:02:47	60	sít
001	+23.4	24.05.	2024	02:02:47	60	sít
001	+23.6	24.05.	2024	03:02:48	60	sít
001	+23.5	24.05.	2024	04:02:48	60	sít
001	+23.6	24.05.	2024	05:02:48	60	sít
001	+23.6	24.05.	2024	06:02:48	60	sít
001	+23.6	24.05.	2024	07:02:48	60	sít
001	+23.6	24.05.	2024	08:02:49	60	sít
001	+23.3	24.05.	2024	09:02:49	60	sít
001	+23.4	24.05.	2024	10:02:49	60	sít
001	+23.3	24.05.	2024	11:02:49	60	sít
001	+23.4	24.05.	2024	12:02:49	60	sít
001	+23.3	24.05.	2024	13:02:49	60	sít
001	+23.4	24.05.	2024	14:02:50	60	sít
001	+23.4	24.05.	2024	15:02:50	60	sít
001	+23.5	24.05.	2024	16:02:50	60	sít

Obr. E.1: Data první den


Identifikátor Zařízení 001:	Teplota[°C]	Datum	Rok	Čas	Stav baterie[%]	Napájení
001	+23.4	24.05.	2024	17:02:50	60	síť
001	+23.4	24.05.	2024	18:02:50	60	síť
001	+23.4	24.05.	2024	19:02:51	60	síť
001	+24.2	24.05.	2024	20:02:51	60	síť
001	+25.1	24.05.	2024	21:02:51	60	síť
001	+25.6	24.05.	2024	22:02:51	60	síť
001	+25.5	24.05.	2024	23:02:51	60	síť
001	+24.8	25.05.	2024	00:02:52	60	síť
001	+24.4	25.05.	2024	01:02:52	60	síť
001	+24.2	25.05.	2024	02:02:52	60	síť
001	+24.2	25.05.	2024	03:02:53	60	síť
001	+23.5	25.05.	2024	04:02:53	60	síť
001	+23.3	25.05.	2024	05:02:53	60	síť
001	+23.8	25.05.	2024	06:02:53	60	síť
001	+23.8	25.05.	2024	07:02:53	60	síť
001	+23.9	25.05.	2024	08:02:53	60	síť
001	+23.8	25.05.	2024	09:02:53	60	síť
001	+23.9	25.05.	2024	10:02:53	60	síť
001	+23.5	25.05.	2024	11:02:54	60	síť
001	+24.3	25.05.	2024	12:02:54	60	síť
001	+24.7	25.05.	2024	13:02:54	60	síť
001	+24.6	25.05.	2024	14:02:54	60	síť
001	+24.3	25.05.	2024	15:02:54	60	síť
001	+23.8	25.05.	2024	16:02:55	60	síť

Obr. E.2: Data druhý den

 **dataloggerserverside@gmail.com** pá 24. 5. 21:02
 komu: mně ▾


Teplota mimo povolené rozmezí.

Identifikátor	Teplota[°C]	Datum	Rok	Čas	Stav baterie[%]	Napájení
001	+25.1	24.05.	2024	21:02:51	60	síť

 **dataloggerserverside@gmail.com** pá 24. 5. 22:02
 komu: mně ▾

Teplota mimo povolené rozmezí.

Identifikátor	Teplota[°C]	Datum	Rok	Čas	Stav baterie[%]	Napájení
001	+25.6	24.05.	2024	22:02:51	60	síť

 **dataloggerserverside@gmail.com** pá 24. 5. 23:02
 komu: mně ▾

Teplota mimo povolené rozmezí.

Identifikátor	Teplota[°C]	Datum	Rok	Čas	Stav baterie[%]	Napájení
001	+25.5	24.05.	2024	23:02:51	60	síť

Obr. E.3: Varovné e-maily v průběhu testů

F Obsah elektronické přílohy

V příloze k závěrečné práci je ve formátu ZIP vložen zdrojový kód, který je psán ve vývojovém prostředí Arduino IDE ve verzi 1.8.19. Pro správnou funkci programu je nutné zadat vlastní parametry WiFi sítě (`ssid,password`). A také je nutné upravit ip adresu(`staticIP`) a podle toho i `gateway`. Dále je nutné založit dva e-maily, které jsou potřeba k zasílání dat. Podrobně k čemu slouží a co je potřeba nastavit je zmíněno v kapitole Nastavení E-mailového účtu. 5.4.