VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

ZABEZPEČENÝ SYSTÉM NEUTRALIZACE ZÁPACHU

SECURE ODOR NEUTRALIZATION SYSTEM

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE — Martin Děcký

AUTHOR

VEDOUCÍ PRÁCE — Ing. Radek Možný

SUPERVISOR

BRNO 2023

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Bakalářská práce

bakalářský studijní program **Informační bezpečnost**

Ústav telekomunikací

Student: Martin Děcký ***ID:*** 230800

Ročník: 3 ***Akademický rok:*** 2022/23

NÁZEV TÉMATU:

Zabezpečený systém neutralizace zápachu

POKYNY PRO VYPRACOVÁNÍ:

Cílem této práce je vytvoření konceptu zabezpečeného smart zařízení, které bude navrženo za účelem neutralizace zápachu z odpadků pomocí ozonu a monitoringu vnitřního prostředí úložiště odpadů. Zařízení bude umístěno v rámci odpadového uložiště (např. popelnice) a bude umožňovat bezúdržbový provoz kombinací fotovoltaického článku a bateriového úložiště. Součástí návrhu je i představení systému monitoringu prostředí – měření teploty, vlhkosti apod.

Navržený systém bude integrovat možnost vzdálené komunikace a ovládání. Musí být schopen vytvářet statistiky o provozu a tyto statistiky přenášet vzdáleně na webový server zabezpečeným způsobem.

Mezi klíčové aspekty řešení práce patří výběr vhodné technologie pro komunikaci, výběr vhodného komunikačního protokolu pro monitoring a řízení a zabezpečení celého komunikačního řetězce na jednotlivých úrovních.

Zařízení musí mj. umožnit synchronizaci reálného času tak, aby nedocházelo ke generaci ozonu v době, kdy je vyšší riziko otevření odpadového úložiště (v denní době) a nedošlo tak k nežádoucímu úniku.

Výstupem bakalářské práce je sestrojený prototyp, který je schopen generovat ozon a současně přenášet tato data zabezpečeným způsobem na vzdálený server, kde budou data bezpečným způsobem zpracována, analyzována a vyhodnocována.

DOPORUČENÁ LITERATURA:

Podle pokynů vedoucího práce.

Termín zadání: 6.2.2023 ***Termín odevzdání:*** 26.5.2023

Vedoucí práce: Ing. Radek Možný

doc. Ing. Jan Hajný, Ph.D.
předseda rady studijního programu

ABSTRAKT

Tato bakalářská práce se zabývá neutralizací zápachu pomocí ozonu a přenosu získaných dat na webový server. Na základě výpočtů byly zvoleny vhodné komponenty, aby bylo celé zařízení bezúdržbové a byl sestaven systém. Odesílání dat na server byl řešen pomocí HTTP (Hypertext Transfer Protocol) POST metody. Sestavený systém by měl být schopný fungovat na základě sesbíraných a vyhodnocených dat, která ale není schopen data poslat na server, který je zabezpečen nejnovějším protokolem TLS (Transport Layer Security).

KLÍČOVÁ SLOVA

Internet věcí, ESP32, ozon, webový server, solární napájení

ABSTRACT

This bachelor thesis deals with the neutralization of odour using ozone and the transfer of the acquired data to a web server. Based on the calculations, suitable components were selected to make the whole device maintenance free and the system was built. The sending of data to the server was dealt with HTTP (Hypertext Transfer Protocol) POST method. The assembled system should be able to function based on the collected and evaluated data, but it is not able to send the data to the server which is secured with the latest TLS (Transport Layer Security) protocol.

KEYWORDS

Internet of Things, ESP32, ozone, web server, solar power

DĚCKÝ, Martin. *Zabezpečený systém neutralizace zápachu*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2023, 51 s. Bakalářská práce. Vedoucí práce: Ing. Radek Možný

Prohlášení autora o původnosti díla

Jméno a příjmení autora:	Martin Děcký
VUT ID autora:	230800
Typ práce:	Bakalářská práce
Akademický rok:	2022/23
Téma závěrečné práce:	Zabezpečený systém neutralizace zápachu

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. Radku Možnému za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Obsah

Seznam obrázků

Seznam výpisů

Úvod

Problém zápachu popelnic je všeobecně rozšířený problém. Největší potíže s pachem mají oblasti s vysokou průměrnou roční teplotou a pak dále obecně většina měst, kde se odpad hromadí velmi rychle a nestíhá se likvidovat, což vede k znečišťování ovzduší pachy. Právě této problematice se věnuje tato práce.

Cílem této práce je vytvořit zabezpečený systém neutralizace zápachu pomocí generátoru ozonu. Systém bude nainstalován v již zmíněných kontejnerech na odpad. Systém bude sbírat data ze senzoru teploty a vlhkosti vzduchu, vyhodnocovat zdali jsou vhodné podmínky pro ozonování a následně tyto data odesílat na webový server. Zařízení bude bezúdržbové, což znamená nezávislé na člověku, tudíž bude použito napájení pomocí solárních panelů. Další součástí práce je přenášená data zabezpečit.

Práce je rozdělena do dvou hlavních částí, teoretické a praktické. V teoretické části proběhne nejprve v kapitole 1 přiblížení pojmu internetu věcí, možností senzorů v IoT a také je zde popsán přenos a zabezpečení dat. V kapitole 2 se čtenář seznámí s ozonem. V poslední kapitole teoretické části 3 jsou popsány jednotlivé komponenty zařízení. Kapitola 4 se věnuje praktické části práce a to konkrétně postupem při výběru komponent a následné sestavování systému.

1 Internet věcí

Internet of Things (IoT) popisuje síť fyzických objektů, které jsou osazeny senzory, softwarem a dalšími technologiemi za účelem propojení a výměny dat s jinými zařízeními a systémy přes internet. Tato zařízení sahají od běžných domácích předmětů až po sofistikované průmyslové nástroje [1].

1.1 Senzory v IoT

Senzory v IoT se používají pro snímání věcí a zařízení atd. Zařízení, které poskytuje použitelný výstup v reakci na zadané měření. Senzor získá fyzikální parametr a převede jej na signál vhodný pro zpracování (např. elektrických, mechanických, optických) charakteristik jakéhokoli zařízení nebo materiálu pro detekci přítomnosti konkrétní fyzikální veličiny. Výstupem snímače je signál, který je převeden do podoby čitelné pro člověka, jako jsou změny charakteristik, změny odporu a kapacity nebo třeba změny teploty a vlhkosti vzduchu [2].

1.2 Přenos dat v IoT

IoT zařízení komunikují desítkami různých způsobů pomocí nejrůznějších protokolů. To jak komunikují a jaký protokol využívají závisí na tom, jakého jsou typu, kde jsou implementovány, s jakými jinými zařízeními komunikují, jak nasbíraná data posílají a jakou přenosovou rychlostí posílají. Neexistuje jediný nejlepší protokol, což je v podstatě běžný způsob používaný ke směrování zpráv z jednoho zařízení IoT do druhého. Správná volba vždy závisí na konkrétních potřebách aplikace [3].

Asi nejpoužívanějšími způsoby přenosu dat v IoT jsou:

- GSM (Groupe Spécial Mobile),
- LTE (Long Term Evolution),
- Bluetooth,
- Wi-Fi (Wireless Fidelity),
- LoRa(Long Range)/LoRaWAN(Long Range Wide Area Network),
- Telemetrie SigFox [4].

1.2.1 Zabezpečený přenos dat

FTP (File Transfer Protocol) se používá ke komunikaci a přenosu souborů mezi počítači v síti TCP/IP (Transmission Control Protocol/Internet Protocol), neboli internetu. Uživatelé, kterým byl udělen přístup, mohou přijímat a přenášet soubory na serveru File Transfer Protocol (FTP host/site) [5].

FTP spojení je navázáno mezi dvěma systémy a komunikují spolu pomocí sítě. Takže pro připojení může uživatel získat povolení poskytnutím přihlašovacích údajů serveru FTP nebo může použít anonymní FTP. Když je navázáno připojení FTP, jsou také vytvořeny dva typy komunikačních kanálů, které se nazývají příkazový kanál a datový kanál. Příkazový kanál se používá k přenosu příkazů a odpovědí z klienta na server a ze serveru na klienta. FTP používá ke komunikaci přes řídicí připojení stejný přístup jako TELNET nebo SMTP. Ke komunikaci používá znakovou sadu NVT ASCII. Používá port číslo 21. Zatímco datový kanál se používá ke skutečnému přenosu dat mezi klientem a serverem. Používá port číslo 20 [6].

FTP klient používající URL dává příkaz FTP spolu s adresou FTP serveru. Jakmile se server a klient připojí k síti, uživatel se přihlásí pomocí ID uživatele a hesla. Pokud uživatel není registrován na serveru, může také přistupovat k souborům pomocí anonymního přihlášení, kde heslo je e-mailová adresa klienta. Server ověří přihlášení uživatele a umožní klientovi přístup k souborům. Klient přenese požadované soubory a ukončí připojení [6].

Při přenosu dat na server je nutné, aby data byla nějakým způsobem chráněna, aby se útočník nedokázal do přenosu dostat a získat tak informace. K tomuto zabezpečení slouží nejrůznější zabezpečovací protokoly. Mezi nejznámější patří SSL nebo také TLS.

SSL

SSL neboli Secure Sockets Layer je internetový bezpečnostní protokol založený na šifrování. Poprvé byl vyvinut společností Netscape v roce 1995 za účelem zajištění soukromí, ověřování a integrity dat v internetové komunikaci. SSL je předchůdcem moderního šifrování TLS, které se dodnes používá [7].

TLS

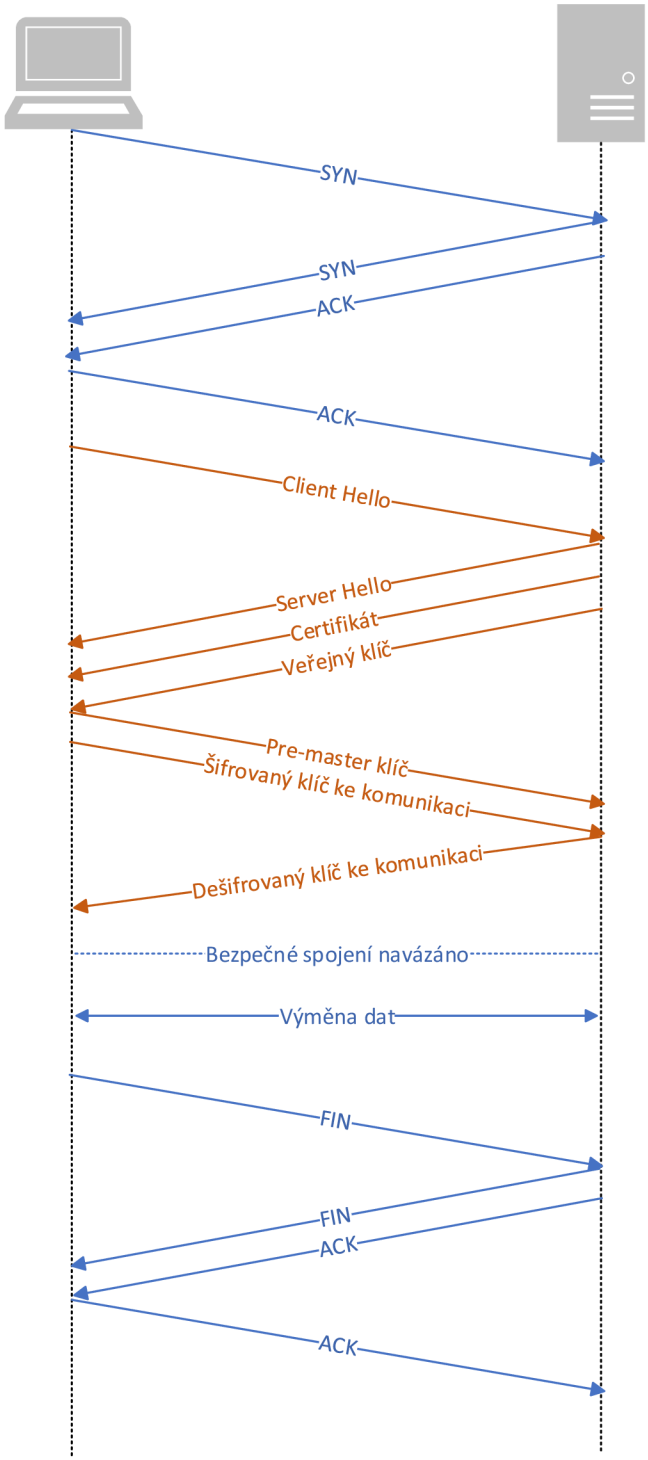
Transport Layer Security neboli TLS je široce rozšířený bezpečnostní protokol navržený pro usnadnění soukromí a zabezpečení dat pro komunikaci přes internet. Primárním případem použití TLS je šifrování komunikace mezi webovými aplikacemi a servery, jako jsou webové prohlížeče načítající web. TLS lze také použít k šifrování jiné komunikace, jako je e-mail, zprávy a hlas přes IP (VoIP). TLS navrhla Internet Engineering Task Force (IETF), mezinárodní organizace pro standardy, a první verze protokolu byla zveřejněna v roce 1999. Nejnovější verzí je TLS 1.3 , která byla publikována v roce 2018 [8].

Navázání TLS spojení začíná klasickým TCP trojcestným handshakem. V prvním kroku, kde chce klient navázat spojení se serverem, se odešle segment SYN (Synchronized sequence number), který informuje server o nadcházející komunikaci a určuje počáteční pořadové číslo segmentů ze strany klienta. V druhém kroku server odpovídá na požadavek klienta obdobným segmentem ACK, tedy jakým pořadovým číslem začíná server. Dále posílá zprávu ACK (Acknowledgement), která značí, že úspěšně přijal segmenty od klienta. Ve třetím kroku TCP handshaku už jen klient potvrzuje serveru, že spojení proběhlo úspěšně [9].

Po navázání TCP spojení je třeba toto spojení ještě zabezpečit. Zabezpečení probíhá právě pomocí TLS protokolu ve čtyřech krocích. Prvním krokem je počáteční zpráva ze strany klienta, která obsahuje verzi a šifrovací sady, které klient podporuje a 32 bajtové náhodné číslo známé jako Client Random. V druhém kroku server odpovídá zprávou Server Hello, ve které odesílá vybranou verzi TLS a šifru ze seznamu poskytnutého klientem. Také generuje 32bajtové číslo, ale jiné než klient a společně s certifikátem a svým veřejným klíčem zasílá klientovi [10].

Certifikáty obsahují veřejný klíč vygenerovaný serverem a digitální podpis podepsaný soukromým klíčem důvěryhodné třetí strany známé jako certifikační autorita (CA). Většina webových prohlížečů a operačních systémů je dodávána s veřejnými klíči od důvěryhodných CA, které se používají k ověření, že CA vydala certifikát. Ve třetím kroku, po ověření certifikátu, klient vygeneruje pre-master klíč, kterým zašifruje veřejný klíč od serveru a posílá ho na stranu serveru. Dále také klient odešle šifrovaný klíč ke komunikaci. V posledním kroku server dešifruje pre-master klíč a klíč ke komunikaci a tím, že pošle zpět dešifrovaný klíč prokáže klientovi, že má ty správné klíče k bezpečné komunikaci [10].

Po výměně dat klient ukončuje spojení TCP segmentem FIN (Finish) nastaveným na hodnotu 1 a čeká na odpověď serveru. Ten odpovídá segmentem ACK a následně také odesílá segment FIN s hodnotou 1 zpět klientovi. Když klient obdrží bitový segment FIN od serveru, potvrdí ho segmentem ACK a spojení je úspěsně ukončeno. Všechny kroky navázání spojení jsou přehledně zpracované na obrázku 1.1 [11].



Obr. 1.1: Navázání TLS spojení.

1.3 Přenos pomocí Wi-Fi

Technologie Wi-Fi (Wireless-Fidelity) je bezdrátová technologie, která k přenosu využívá rádiové vlny v sítích WLAN. Pro přenos využívá šířku pásma 2,4 a 5 GHz s přenosovou rychlostí až 150 Mbps. V současnosti je Wi-Fi nejrozšířenější technologií bezdrátového přenosu dat. Základním prvkem sítě je tzv. přístupový bod neboli „hotspot“, který vysílá signál, který je počítač schopen rozpoznat a zpracovat [4].

1.3.1 Zabezpečení Wi-Fi

Bezpečnostní protokoly Wi-Fi využívají k zabezpečení sítí a ochraně dat svých klientů šifrovací technologii. Bezdrátové sítě jsou však často méně bezpečné než kabelové, takže protokoly bezdrátového zabezpečení jsou zásadní pro vaši bezpečnost online. Nejběžnějšími protokoly zabezpečení Wi-Fi jsou dnes WEP (Wired Equivalent Privacy), WPA (Wi-Fi Protected Access) a WPA2. Existuje také protokol WPA3, ale je velmi málo rozšířený, protože ho podporuje velmi malé množství zařízení a případný přechod ze staršího protokolu WPA2 je velice nákladný. Všechny bezpečnostní protokoly používají kryptografické klíče k náhodnému rozdělení dat, aby byla nedešifrovatelná. Protože systémy Wi-Fi používají symetrické šifrování, používá se k šifrování a dešifrování dat stejný klíč [12].

WEP

WEP (Wired Equivalent Privacy) je nejstarší protokol zabezpečení Wi-Fi. Jednalo se o součást ochrany osobních údajů stanovenou v IEEE 802.11, sadě technických norem, jejichž cílem bylo poskytnout bezdrátové místní síti (WLAN) srovnatelnou úroveň zabezpečení jako kabelová místní síť (LAN). Wi-Fi Alliance ratifikovala WEP jako bezpečnostní standard v roce 1999. Jakmile bylo WEP nabízeno jako stejné bezpečnostní výhody jako kabelové připojení, bylo v průběhu let sužováno mnoha bezpečnostními chybami. A jak se zvýšil výpočetní výkon, tyto zranitelnosti se zhoršily. Navzdory snahám o zlepšení WEP je stále zranitelný vůči narušení bezpečnosti. V roce 2004 byl provoz WEP ukončen a všechny systém používající WEP byly aktualizovány nebo nahrazeny [12].

WPA

WPA (Wi-Fi Protected Access) je bezdrátový bezpečnostní protokol vydaný v roce 2003, aby řešil rostoucí zranitelnosti svého předchůdce WEP. Protokol WPA je bezpečnější než WEP, protože k šifrování používá 256 bitový klíč, což je hlavní vylepšení 64 bitových a 128 bitových klíčů, které používal WEP. WPA také používá protokol Temporal Key Integrity Protocol (TKIP), který dynamicky generuje nový klíč pro

každý paket nebo jednotku dat. TKIP je mnohem bezpečnější než systém s pevným klíčem používaný WEP [12].

WPA ovšem stále není bezchybný protokol. TKIP, základní komponent WPA, byl navržen pro implementaci do systémů s podporou WEP prostřednictvím aktualizací firmwaru. To vedlo k tomu, že WPA stále spoléhá na snadno zneužitelné prvky [12].

WPA2

WPA2 je druhá a mezi uživateli nejrozšířenější generace bezdrátového bezpečnostního protokolu WPA. Stejně jako jeho předchůdce byl WPA2 navržen k zabezpečení a ochraně sítí Wi-Fi. WPA2 zajišťuje, že data odesílaná nebo přijímaná přes vaši bezdrátovou síť jsou šifrována a přístup k nim mají pouze lidé s vaším síťovým heslem [12].

Výhodou systému WPA2 bylo, že zavedl Advanced Encryption System (AES), který nahradil zranitelnější systém TKIP používaný v původním protokolu WPA. AES, který používá vláda USA k ochraně utajovaných dat, poskytuje silné šifrování [12].

Bohužel, stejně jako jeho předchůdce, jsou přístupové body s podporou WPA2 zranitelné vůči útokům prostřednictvím WEP. Pro eliminaci této zranitelnosti je třeba úplně zakázat WPA, aby na tento protokol již router nespoléhal [12].

WPA3

WPA3 je nejnovější bezpečnostní protokol pro Wi-Fi sítě, který Wi-Fi Alliance poprvé představila v roce 2018. WPA3 byl vytvořen, aby poskytoval větší bezpečnost a ochranu proti útokům, jako jsou například útoky na odposlech, krádeže přístupových hesel a další bezpečnostní hrozby. Mezi klíčové vlastnosti WPA3 patří vylepšené šifrování, používání individuálních klíčů pro každé připojení a ochrana proti "bruteforce"útokům. WPA3 také zahrnuje další bezpečnostní prvky, jako jsou například ochrana proti zneužití při připojování k veřejným Wi-Fi sítím [13].

Dalším vylepšením je protokol Wi-Fi Device Provisioning Protocol (DPP), který nahrazuje snadno zneužitelné Wi-Fi Protected Setup (WPS). Zařízení lze ověřit pro připojení k síti bez hesla pomocí DPP včetně QR kódů nebo značek NFC [14].

Přestože WPA3 představuje významný pokrok v oblasti zabezpečení Wi-Fi sítí a řeší všechny zranitelnosti WPA a WPA2, stále se jedná o relativně novou technologii a zatím není podporován na všech zařízeních.

Módy WPA

Všechny generace bezpečnostního protokolu WPA jsou rozděleny na dva módy. WPA-Personal (WPA-PSK) a WPA-Enterprise (WPA-EAP).

WPA-Personal je vhodný pro většinu domácích sítí. Pokud je na bezdrátovém směrovači nebo přístupovém bodu (AP) nastaveno heslo, musí jej uživatelé při připojení k síti Wi-Fi zadat [15]. V režimu PSK nelze bezdrátový přístup spravovat individuálně ani centrálně. Pro všechny uživatele platí jedno heslo, které je třeba ručně změnit u všech bezdrátových klientů, jakmile je ručně upraveno na původním bezdrátovém směrovači nebo přístupovém bodu. Heslo je uloženo v bezdrátových klientech. Proto se může kdokoli na počítači připojit k síti a heslo také vidět [15].

WPA-Enterprise poskytuje zabezpečení potřebné pro bezdrátové sítě v podnikovém prostředí. Jeho nastavení je složitější a nabízí individuální a centralizované řízení přístupu k síti Wi-Fi. Při pokusu o připojení k síti musí uživatelé předložit své přihlašovací údaje [15].

Tento režim podporuje ověřování 802.1x RADIUS a je vhodný v případech, kdy je nasazen server RADIUS. Režim WPA-Enterprise by se měl používat pouze v případě, že je pro ověřování klientů připojen server RADIUS. Uživatelé nikdy nepřijdou so styku se skutečnými šifrovacími klíči. Jsou bezpečně vytvořeny a přiděleny pro každou uživatelskou relaci na pozadí poté, co uživatel předloží své přihlašovací údaje. Tím se zabrání tomu, aby někdo získal síťový klíč z počítačů [15].

1.4 Přenos dat na webový server

1.4.1 MQTT protokol

MQTT (Message Queuing Telemetry Transport) je protokol pro zasílání zpráv založený na standardech nebo soubor pravidel, který se používá pro komunikaci mezi stroji. Inteligentní senzory, nositelná zařízení a další zařízení IoT musí obvykle přenášet a přijímat data přes síť s omezenými zdroji a omezenou šířkou pásma. K tomuto přenosu dat přes síť lze využít protokol MQTT. Tento protokol dokáže efektivně komunikovat data IoT. MQTT podporuje zasílání zpráv mezi zařízeními do cloudu a cloudem do zařízení [16].

Protokol MQTT byl vynalezen v roce 1999 pro použití v ropném a plynárenském průmyslu. Inženýři potřebovali protokol pro minimální šířku pásma a minimální ztrátu baterie pro sledování ropovodů přes satelit. V roce 2010 IBM vydala MQTT 3.1 jako bezplatný a otevřený protokol, který byl poté v roce 2013 předložen k údržbě orgánu pro specifikaci Organisation for the Advancement of Structured Information Standards (OASIS). V roce 2019 vydala OASIS aktualizovanou verzi MQTT 5. Nyní již MQTT není zkratka, ale je považován za oficiální název protokolu [16].

Protokol MQTT funguje na principech modelu publish/subscribe (publikování/odebírání). V tradiční síťové komunikaci spolu klienti a servery komunikují přímo. Klienti požadují zdroje nebo data ze serveru, poté server zpracuje a odešle odpověď.

MQTT však používá vzor publish/subscribe k oddělení odesílatele zprávy (vydavatele) od příjemce zprávy (odběratele). Namísto toho se o komunikaci mezi vydavateli a odběrateli stará třetí komponent, nazývaný broker (zprostředkovatel) zpráv. Úkolem brokera je filtrovat všechny příchozí zprávy od vydavatelů a správně je distribuovat odběratelům [16].

Klient MQTT

Klient MQTT je jakékoli zařízení od serveru po mikrokontrolér, které provozuje knihovnu MQTT. Pokud klient odesílá zprávy, chová se jako vydavatel, a pokud zprávy přijímá, chová se jako odběratel. V podstatě každé zařízení, které komunikuje pomocí MQTT po síti, lze nazvat klientským zařízením MQTT [16].

Broker MQTT

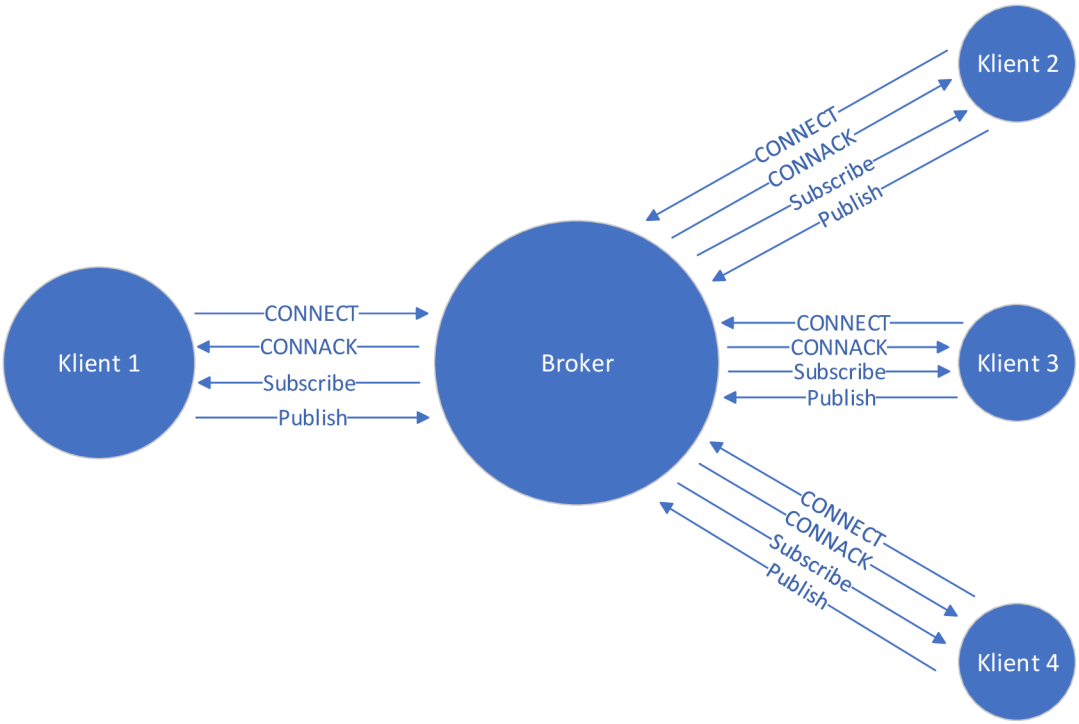
Broker MQTT je backendový systém, který koordinuje zprávy mezi různými klienty. Mezi povinnosti zprostředkovatele patří přijímání a filtrování zpráv, identifikace klientů přihlášených k odběru každé zprávy a odesílání zpráv. Je také zodpovědný za další úkoly, jako například autorizace a ověřování klientů MQTT, předávání zpráv dalším systémům pro další analýzu nebo zpracování zmeškaných zpráv a klientských relací a také ukládání zpráv na server [16].

Obecně existují dva typy brokerů, spravovaný a samoobslužný broker. Spravovaní brokeři nevyžadují nastavování na serveru, aby byla umožněna komunikace MQTT. Služby spravovaných brokerů vám umožňují používat jejich hostované brokery pro systém. Dobrým příkladem spravovaného MQTT brokeru je AWS IoT Core [17].

Samoobslužný broker vyžaduje aby si uživatel nainstaloval broker na svůj vlastní VPS (Virtual Private Server) server se statickou IP. Instalační proces není obtížný, ale správa, zabezpečení a škálování brokerů vyžaduje důkladnou znalost systému. Existuje několik open-source implementací brokerů MQTT, například hivemq nebo mosquitto [17].

Spojení klienta s brokerem pomocí MQTT

Klienti a brokeři začnou komunikovat pomocí připojení MQTT. Klienti zahájí připojení odesláním zprávy CONNECT zprostředkovateli MQTT. Broker potvrdí navázání spojení odpovědí zprávou CONNACK . Klient MQTT i zprostředkovatel vyžadují ke komunikaci TCP/IP. Klienti se nikdy nespojují mezi sebou, pouze s brokerem, jak je možné vidět na obrázku 1.2, kde je znázorněno celé spojení mezi brokerem a jednotlivými klienty [16].



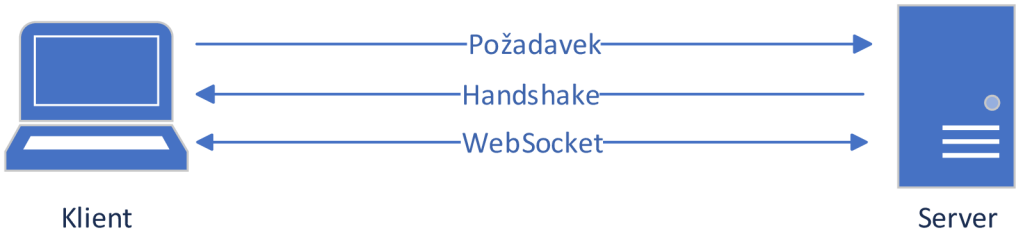
Obr. 1.2: Spojení pomocí MQTT.

1.4.2 WebSocket

WebSocket je obousměrný, plně duplexní protokol, který se používá ve scénáři komunikace klient-server. Jedná se o stavový protokol, což znamená, že spojení mezi klientem a serverem zůstane aktivní, dokud nebude ukončeno kteroukoli stranou (klientem nebo serverem). Po ukončení spojení klientem a serverem je spojení ukončeno z obou konců [18].

Kdykoli je tedy zahájeno spojení mezi klientem a serverem, klient provede handshake a rozhodne se vytvořit nové spojení, které zůstane otevřené, dokud je některá z komunikujících stran neukončí. Když je spojení navázáno a aktivní, komunikace probíhá pomocí stejného spojovacího kanálu, dokud není ukončeno [18].

Po handshaku se klient a server dohodnou na novém připojení, toto nové připojení bude známé jako WebSocket. Jakmile je navázání komunikačního spojení otevřeno, bude výměna zpráv probíhat v obousměrném režimu, dokud spojení mezi klientem a serverem přetrvává. Pokud se některý z nich neočekávaně odpojí nebo se rozhodne spojení ukončit, tak je spojení uzavřeno oběma stranami [18].



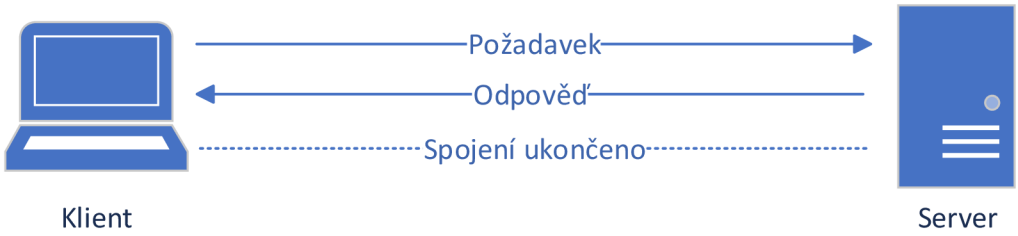
Obr. 1.3: Spojení pomocí WebSocket protokolu.

1.4.3 HTTP POST a REST API

HTTP (Hypertext Transfer Protocol) je jednosměrný komunikační protokoly na bázi klient-server, kde klient odešle požadavek a server odešle odpověď. Například, když uživatel odešle požadavek na server, tento požadavek má podobu HTTP nebo HTTPS, po obdržení požadavku server odešle odpověď klientovi, každý požadavek je spojen s odpovídající odpovědí, po odeslání odpovědi se spojení uzavře. Každý HTTP nebo HTTPS požadavek pokaždé naváže nové spojení se serverem a po obdržení odpovědi se spojení samo ukončí. HTTP je bezstavový protokol, který běží nad TCP, což je protokol orientovaný na spojení, který zaručuje doručení přenosu datových paketů pomocí třícestného handhaku a ztracené pakety znovu přenáší [18].

HTTP může běžet nad jakýmkoli spolehlivým protokolem orientovaným na připojení, jako je TCP, SCTP (Stream Control Transmission Protocol). Když klient odešle HTTP požadavek na server, je mezi klientem a serverem otevřené TCP spojení a po obdržení odpovědi se TCP spojení ukončí, např. pokud klient odešle 10 požadavků na server se otevře 10 samostatných TCP spojení. a následně se zavře po obdržení odpovědi [18].

Na straně serveru se HTTP POST požadavky zpracovávají v REST API (Representational State Transfer Application Programming Interface). REST API je rozhraní pro programování aplikací, které vyhovuje omezením architektonického stylu REST a umožňuje interakci s webovými službami RESTful. Informace, které na server přijdou prostřednictvím HTTP jsou dále zpracovávány pomocí programovacího jazyka PHP nebo Python [19].



Obr. 1.4: Spojení pomocí HTTP protokolu.

1.5 Webový server

Webový server je počítač, který provozuje webové stránky. Je to počítačový program, který distribuuje webové stránky tak, jak jsou požadovány. Základním cílem webového serveru je ukládat, zpracovávat a doručovat webové stránky uživatelům. Tato komunikace se provádí pomocí HTTP protokolu. Tyto webové stránky jsou většinou statického obsahu, který zahrnuje HTML (Hypertext Markup Language) dokumenty, obrázky, styly, testy atd. Webový server kromě HTTP podporuje také protokoly SMTP a FTP pro zasílání e-mailů a pro přenos souborů a skladování [20].

Největší výzvou webového serveru je obsluhovat mnoho různých webových uživatelů současně tzn, každý z nich požaduje jiné stránky. Webové servery zpracovávají soubory napsané v různých programovacích jazycích, jako je PHP, Python nebo Java [21].

Webový server je požádán, aby představil webovou stránku s obsahem prohlížeči uživatele. Všechny webové stránky na internetu mají jedinečný identifikátor ve smyslu IP adresy. Tato adresa internetového protokolu se používá ke komunikaci mezi různými servery přes internet. V dnešní době je server Apache nejběžnějším webovým serverem dostupným na trhu [20].

Webový server Apache

Apache je software s otevřeným zdrojovým kódem, který zpracovává téměř 70 procent všech dnes dostupných webových stránek. Většina webových aplikací používá jako výchozí prostředí webového serveru Apache. Dalším obecně dostupným webovým serverem je Internet Information Service (IIS) [20].

Webový server IIS

Internet Information Service neboli IIS, je webový server společnosti Microsoft, který běží na operačním systému Windows a používá se k výměně statického a dynamic-

kého webového obsahu s uživateli internetu. IIS lze použít k hostování, nasazení a správě webových aplikací pomocí technologií jako ASP.NET a PHP [22].

Porovnání Apache a IIS

Hlavním rozdílem mezi webovými servery Apache a IIS je bezpochyby jejich kompatibilita. Zatímco Apache může běžet na jakémkoli operačním systému, včetně Linux, iOS nebo UNIXu, tak IIS pracuje pouze na operačním systému Windows a šance, že by pracoval stabilně na jiném systému je velmi nízká. S tím souvisí další rozdíl a tím je bezpečnost. Protože IIS pracuje pouze na Windows, který je sám o sobě náchylný k nejrůznějším virům a malwarům, tak je méně bezpečný než Apache, který má velmi dobré zabezpečení [23].

Třetím rozdílem je licencování. Apache nevyžaduje pro hostování webových stránek nebo pro komerční provoz žádnou licenci vývojáře. Naproti tomu straně IIS není open-source server a tak ke komerčnímu užití je potřeba licence od vývojáře [23].

Poslední rozdíl souvisí s množstvím webových stránek, které mohou servery hostit. Jak již bylo zmíněno výše, IIS pracuje pouze na Windows a tak i podporuje pouze stránky kompatibilní s Windows. Apache tedy může hostit více webových stránek [23].

2 Ozon

Ozon je namodralý plyn. Jedná se o alotropickou modifikací kyslíku. Molekuly ozonu mají tři atomy kyslíku (O_3), na rozdíl od kyslíku ve vzduchu (O_2). Ozon je silný, protože nadbytečný atom kyslíku může snadno uniknout a připojit se k jiným látkám. Tato vysoce reaktivní kvalita může potenciálně změnit chemické složení některých látek ve vzduchu a ovlivnit naše buňky, pokud ji vdechneme [24].

Při prvním setkání s ozonem se každému vybaví ozonová vrstva, která chrání veškerý život před nebezpečným UV zářením slunce. To je člověku nezávadný ozon, který je přítomen ve stratosféře. Ale ozon na úrovni země je nežádoucí, protože je zdraví nebezpečný a je klasifikován podle Agentury pro ochranu životního prostředí ze Spojených států Amerických (EPA) jako látka znečišťující ovzduší. Ozon na úrovni země vzniká v přírodě interakcí slunečního záření s určitými chemikáliemi, které se uvolňují do životního prostředí, včetně emisí z vozidel a průmyslových závodů. Vzniká také při úderu blesku, proto je po bouřce vždy cítit ve vzduchu ozonový zápach. Ozon, ať už čistý nebo smíšený s jinými chemikáliemi, může být zdraví škodlivý [24].

2.1 Vznik a výskyt ozonu

Ozon (O_3) vzniká, když je dvouatomový kyslík (O_2) vystaven elektrickému poli nebo ultrafialovému (UV) světlu. Vystavení těmto vysokým úrovním energie způsobí, že se část dvouatomových molekul kyslíku rozdělí na jednotlivé atomy kyslíku. Tyto volné atomy kyslíku se následně tzv. rekombinačním procesem spojují s dvouatomovými molekulami kyslíku za vzniku ozonu. Molekuly kyslíku a ozonu jsou porovnány na obrázku 2.1 [25].

Ozon je nestabilní molekula kvůli slabým vazbám držícím třetí atom kyslíku, což z ozonu dělá přirozeně silné oxidační a dezinfekční činidlo. Ozon dodává oxidační sílu buď přímo, nebo prostřednictvím tvorby hydroxylových volných radikálů při rozkladu rozpuštěného ozonu na vodu [25].

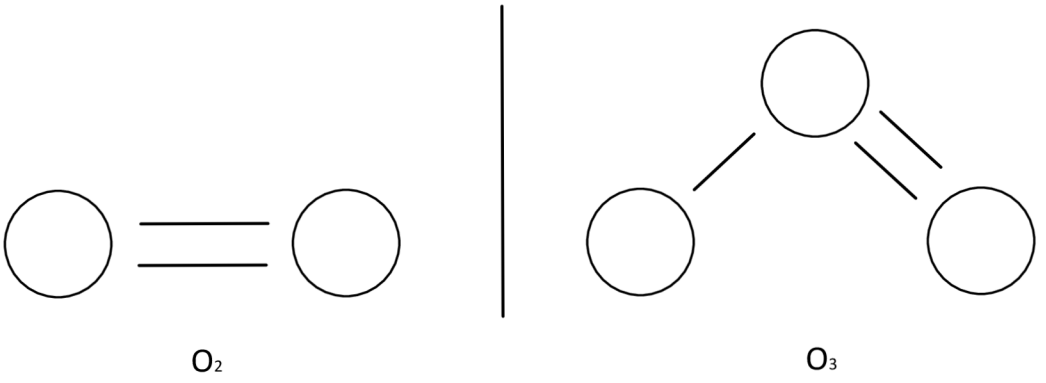
Tato reakce poskytuje tři simultánní procesy: oxidaci, dezinfekci a rozklad. Během oxidačního procesu může ozon, přímo a prostřednictvím vysoce reaktivních hydroxylových radikálů, rozbít chemické vazby organických sloučenin. Například složky buněčných stěn mikroorganismů mohou být oxidovány a rozkládány ozonem. Tento proces usnadňuje dezinfekci narušením a lýzou (rozkladem) buněčných stěn, čímž je obsah buňky vystaven další oxidaci a inaktivaci. Nakonec se ozón rozloží na dvouatomový kyslík a nezanechá žádnou nežádoucí zbytkovou chuť ani zápach [25].

Vznik ozonu v přírodě:

- výboji (blesky) při bouřce,
- slunečním (UV) zářením,
- fotochemicky za současného působení smogu a záření.

Vznik technicky pomocí pomocí:

- výbojů dielektrické bariéry,
- UV výbojky,
- plazmy [25].



Obr. 2.1: Porovnání molekul O_2 a O_3.

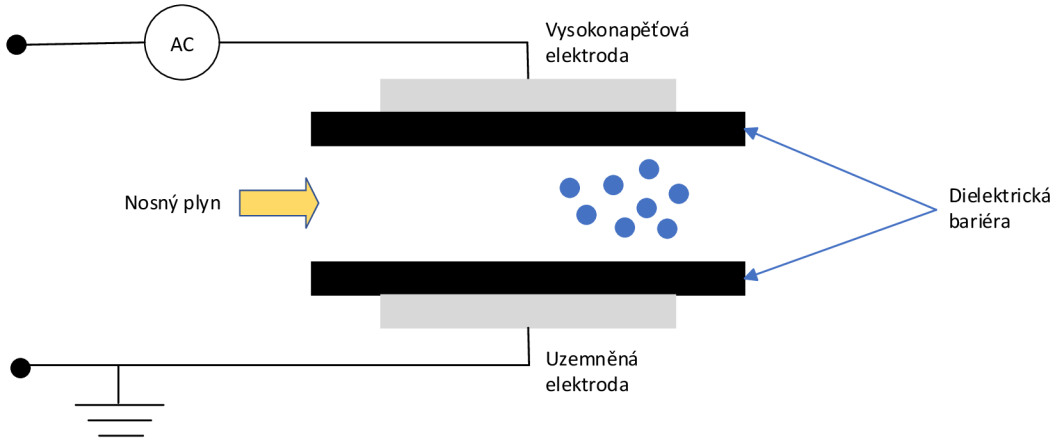
3 Základní stavební bloky

V následující kapitole jsou obecně popsány komponenty potřebné pro sestavení systému a poté i konkrétní typy, použity v této práci. Výběr jednotlivých částí je zde rozebrán a objasněn.

3.1 Generátor ozonu

Generátor ozonu, nebo také ozonizér, je přístroj, který dokáže vyčistit vzduch v jakkoli velkém uzevřeném prostoru pomocí aktivního kyslíku (ozonu) a to efektivně a ekologicky a za malý časový úsek. Ozon vyrábí pomocí dielektrického barierového výboje [26].

Jedná se o jeden z nejefektivnějších způsobů, jakými lze ozon technicky vyrobit. Na obrázku 3.1 je konfigurace elektrod, používaná právě k dielektrickému výboji. Skládá se dvou plochých kovových elektrod pokrytých dielektrickým materiálem, z vysokonapěťové elektrody a uzemněné elektrody. Mezi nimi se pohybuje nosný plyn který je přeměňován na ozon pomocí výbojů vytvořených elektrickým polem mezi deskami. V jiných konfiguracích mohou být elektrody jiného tvaru, např. válce a také může být dielektrický materiál pouze na jedné z elektrod [26].



Obr. 3.1: Dielektrický bariérový výboj.

Generátory ozonu se dělí hlavně podle velikosti a s tím spojeným množstvím vyprodukovaného ozonu za hodinu. Ty nejmenší ozonovače dokáží vyrobit za hodinu jen desítky miligramů ozonu za hodinu. Středně velké generátory, které jsou asi nejběžnější vyprodukují od 1 g do 10 g za hodinu a ty největší generátory dokáží vytvářet až 100 gramů ozonu za hodinu [24].

Samotný proces ozonového čištění vzduchu trvá od pár minut do 2 až3 hodin. Záleží na výkonu ozonizéru a velikosti prostoru. Interiér vozu můžete mít vydezinfikovaný za 10 až 30 minut, zatímco hala o výměře 200 m^2 a více bude vyčištěná za již zmíněné 2 až 3 hodiny, nebo po několika dvouhodinových cyklech. Co se týká konkrétně popelnice, tak takto malý prostor je vyčištěn do pěti minut, protože velikost prostoru nepřesáhne 1 m^3 [24].

3.2 Mikrokontroler

Vývojová deska je deska s plošnými spoji používaná pro vývoj vestavěného systému, včetně řady hardwarových komponent, jako je centrální procesorová jednotka, paměť, vstupní zařízení, výstupní zařízení, datová cesta/sběrnice a rozhraní externích zdrojů [27].

Vývojové desky jsou obecně přizpůsobeny vývojáři vestavěných systémů podle potřeb vývoje. Vývojáři mohou také sami zkoumat a navrhovat vývojovou desku. Vývojová deska je pro začátečníky, aby se naučili hardware a software systému. Na některých vývojových deskách je k dispozici základní integrované vývojové prostředí a také zdrojový kód softwaru a schémata hardwaru [27].

V obecném procesu vývoje vestavěného systému je hardware obecně rozdělen do dvou platforem, jedna je vývojová platforma (hostitel) a druhá je cílová platforma (cíl), tedy vývojová deska. Vývojová a cílová platforma se mezi sebou propojují prostřednictvím přenosového rozhraní, jako je sériový port, USB, paralelní port nebo síť (Ethernet) [27].

3.2.1 Mikrokontroler ESP32

ESP32 je řada levných a nízkoenergetických mikrokontrolérů System on a Chip (SoC) vyvinutých společností Espressif, které zahrnují bezdrátové funkce Wi-Fi, Bluetooth classic i Bluetooth Low Energy (BLE). Poháněn může být několika druhy procesorů. Procesor Tensilica Xtensa LX6 v jednojádrové nebo dvoujádrové variantě, dvoujádrový Xtensa LX7 nebo jednojádrový RISC-V. Většinou se ale jedná o dvoujádrovou variantu procesoru Xtensa LX6 [28].

Ke komunikaci s počítačem přes port COM používá čip CP2102 (USB to UART). Na ESP32 se může objevit také čip CH340, ale ten není tak častý [28].

Dále je na desce (kromě 30 pinů) také vestavěná anténa, která podporuje Wi-Fi v pásmu 2,4 GHz, regulátor napětí, tlačítko RESET, sloužící pro restart vývojové desky a tlačítko BOOT pro uvedení desky do režimu, ve kterém přijímá naprogramovaný kód [28].

Tab. 3.1: Specifikace ESP32 [28].

Počet jader:	2 (dvoujádrové)
Wi-Fi:	2,4 GHz až 150 Mbit/s
Bluetooth:	BLE a Bluetooth 4.2
Architektura:	32 bitů
Takt:	Až 240 MHz
RAM:	512 kB
Flash paměť:	4 MB
Počet pinů:	30
Provozní napětí:	3,3 V

Na desce jsou přítomny také dvě vestavěné LED diody. Jedna je modrá, vnitřně připojena k GPIO 2. Ta je užitečná při ladění, aby poskytla vyzuální fyzický výstup. Druhá LED dioda je červená a slouží k indikaci napájení desky [28].

Vstupní a výstupní piny

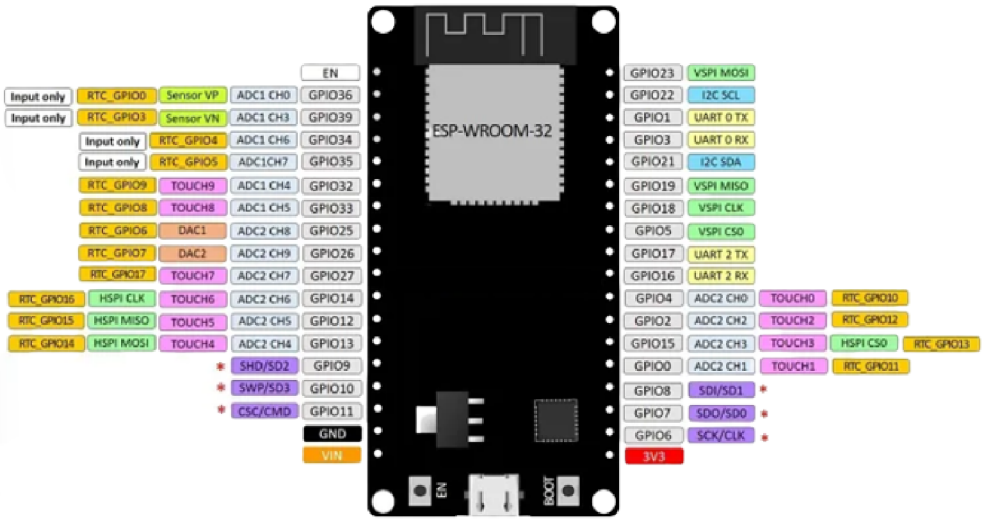
Na vývojové desce ESP32 se nachází 30 exponovaných GPIO (General-purpouse input/output) pinů, jak je možné vidět na obrázku 3.2 [29].

- Napájecí piny – Obvykle na všech typek ESP32. Tyto piny lze použít k napájení desky, když není využito napájení přes USB port. V případě napájení přes USB, lze piny použít k napájení dalších periferií. Konkrétně se jedná o piny 3V3, GND a VIN.
- GPIO piny – Tyto piny lze využít k realizaci projektu. Standardně jsou vlastnosti pinů již předdefinované, ale je možné je definovat jinak v softwaru. Některé piny mají specifické vlastnosti a proto jsou vhodné pouze pro konkrétní projekty.

3.3 Solární panel

Solární panely (také známé jako fotovoltaické panely) se používají k přeměně světla ze slunce na elektřinu, kterou lze použít k napájení elektrických spotřebičů nebo k uchování v nabíjecích akumulátorech.

Solární panely shromažďují čistou obnovitelnou energii ve formě slunečního světla, které se skládá z částic energie nazývaných fotony, a přeměňují toto světlo na elektřinu, kterou lze následně použít k napájení. Solární panely se skládají z několika jednotlivých solárních článků, které jsou samy složeny z vrstev křemíku, fosforu (poskytuje záporný náboj) a boru (poskytuje kladný náboj). Solární panely absorbují



Obr. 3.2: Piny ESP – 32 a jejich vlastnosti [29].

fotony a tím iniciují elektrický proud. Výsledná energie generovaná z fotonů dopadajících na povrch solárního panelu umožňuje elektronům vyrazit z jejich atomových drah a uvolnit je do elektrického pole generovaného solárními články, které pak tyto volné elektrony přitáhnou do směrového proudu. Celý tento proces je známý jako „fotovoltaický efekt“ [31].

V dobře vyvážené konfiguraci připojené k akumulátoru generuje solární pole energii během dne, ta se uloží na baterii a je k dispozici i v noci. Solární pole posílá stejnosměrný proud (DC) přes regulátor nabíjení do bateriové banky. Energie je poté odebírána z baterie do střídače, který převádí stejnosměrný proud na střídavý proud (AC), který lze použít pro spotřebiče bez stejnosměrného proudu. S pomocí měniče lze pole solárních panelů dimenzovat tak, aby splňovaly nejnáročnější požadavky na elektrickou zátěž. Střídavý proud lze použít k napájení zátěží v domácnostech nebo komerčních budovách, rekreačních vozidlech a lodích, vzdálených chatkách, chatách nebo domech, dálkových řízeních dopravy, telekomunikačních zařízeních, monitorování toku ropy a plynu, RTU, SCADA a mnoho dalších [31].

3.3.1 Výhody

Mezi všemi výhodami solárních panelů je nejdůležitější, že solární energie je skutečně obnovitelným zdrojem energie. Lze jej využít ve všech oblastech světa a je k dispozici každý den. Sluneční energie je nevyčerpatelná a tak nám na rozdíl od jiných zdrojů energie nemůže dojít. [32].

Druhá výhoda solárních panelů jsou jejich nízké náklady na údržbu. Stačí je pouze udržovat relativně čisté. Také výdrž je velmi dobrá, protože zde nejsou žádné

pohyblivé části, takže nedochází k mechanické u opotřebení. Životnost panelu bývá obvykle 10 let a poté stačí pouze vyměnit panel a systém může fungovat dále [32].

3.3.2 Nevýhody

Asi hlavní nevýhodou solární energie je závislost na počasí. I když lze solární energii stále shromažďovat během zatažených a deštivých dnů, účinnost solárního systému klesá. Solární panely jsou závislé na slunečním světle, aby efektivně chytaly sluneční svit a přeměňovaly ho na energii. Proto může mít zamračená obloha znatelný vliv na energetický systém. Je také důležité zmínit, že sluneční energie nemůže být sbírána v noci, tudíž se čas, za který můžeme dobíjet akumulátory znatelně sníží [32].

Z toho vyplývá druhý zápor sluneční energie, skladování. Solární energii je třeba využít hned , nebo ji lze ukládat do baterií. Tyto baterie, používané v solárních systémech mimo síť, lze nabíjet během dne. Nejideálnější způsob je ve dne používat solární energii a v noci používat energii ze sítě, což je ale možné jen v případě, že je systém do sítě připojen [32].

Poslední nevýhodou je velikost solárního panelu. Pro vyrobení dostatku energie je většinou potřeba hodně místa. Když ale nemáme potřebné místo a panely jsou proto menší, tak vyrábí daleko méně energie a dobití akumulátoru je proto mnohonásobně delší [32].

3.4 Akumulátor

Dobíjecí baterie je zařízení pro ukládání energie, které lze po vybití znovu nabít přivedením stejnosměrného proudu na jeho svorky. Dobíjecí baterie je obecně ekologičtější a udržitelnější náhradou jednorázových baterií, které jsou dodávány plně nabity a po vyplýtvání jsou na vyhození.Proto jsou dobíjecí baterie šetrnější k přírodě. Jednorázové baterie generují proud pomocí chemických reakcí, při kterých se spotřebovává reaktivní anoda. Anoda v dobíjecí baterii se spotřebovává také, ale pomaleji, což umožňuje mnoho nabíjení a vybíjení. Při používání jsou dobíjecí baterie stejné jako běžné. Po vybití jsou však baterie umístěny do nabíječky nebo v případě vestavěných baterií připojen AC/DC adaptér nebo jiný zdroj energie [33].

I když dobíjecí baterie nabízejí lepší dlouhodobé náklady a snižují množství odpadu, mají několik nevýhod. Mnoho typů dobíjecích článků vytvořených pro spotřebitelská zařízení, včetně „tužkových“ baterií AA a AAA produkuje nižší napětí 1,2 V na rozdíl od 1,5 V. I když toto nižší napětí nebrání správnému provozu ve správně navržené elektronice, může to znamenat, že jedno nabití nevydrží tak dlouho nebo nenabídne stejný výkon v relaci. To však neplatí pro lithium-polymerové (Li-Pol)

a lithium-iontové (Li-Ion) baterie, asi nejznámější a nejčastěji používané typy dobíjecích akumulátorů. Některé další typy baterií, jako je nikl-kadmium a nikl-metal hydrid, mohou při pouze částečném vybití vyvinout paměťový efekt baterie, což snižuje výkon následných nabíjení a tím i životnost baterie v daném zařízení [33].

3.4.1 Rozdíly Li-Pol a Li-Ion baterií

Nejzásadnějším rozdílem mezi lithium-polymerovými a lithium-iontovými bateriemi je biochemický elektrolyt v záporných a kladných elektrodách. Zatímco v lithium-iontové baterii je tekutý elektrolyt, lithium-polymerová baterie používá suchý, pevný elektrolyt [34].

Co se výkonu týče, oba typy baterií jsou vysoce výkonné. Lithium-iontové baterie jsou ale o něco účinější a používají se více než lithium-polymerové. Lithium-polymerové baterie jsou ale na druhou stranu zase lehčí než lithium-iontové a díky tomu, že jsou robustní a pevné, tak mají nižší riziko vytečení. Celková životnost baterie je ale vyšší u lithium-iontových baterií, ale postupem času se z nich vytrácí jejich úložná kapacita [34].

4 Návrh

4.1 Výběr komponent

Pro výběr jednotlivých hlavních komponent bylo nejprve nutné si určit, při jakém napětí bude celý systém pracovat. Kvůli velikosti a kapacity různých nabíjecích akumulátorů a také velikosti a výkonu generátorů ozonu bylo rozhodnuto pro 12 V napětí. Baterie i generátory jsou kompaktních rozměrů a navíc jsou generátory dostatečně výkonné a baterie zase objemné.

Jaký konkrétní generátor zvolit se odvíjí hlavně od velikosti místa určeného pro čištění. Pro velké místnosti je třeba zvolit ozonovač, který vyrobí stovky gramů ozonu za hodinu. Pro průměrnou popelnici o velikosti $1{,}5 \times 0{,}5 \times 0{,}5$ m postačí opravdu malý generátor s vyprodukovaným množstvím 10 g/h. Příkon mého generátoru je 10 Wh, ale za jeden den by celý cyklus netrval déle než půl hodiny, spíše by se čas ozonování pohyboval při této velikosti popelnice v jednotkách minut, tudíž budu počítat s denní spotřebou 5 Wh.

Zvolený mikrokontroler ESP32 je velmi úsporný spotřebič a jeho spotřeba by dle technických parametrů neměla být vyšší než 0,5 W za hodinu, takže za celý den provozu to vychází na 12 Wh, tzn. celková denní spotřeba systému se pohybuje okolo 17 Wh.

Pro napájení celého systému byl vybrán solární panel s výstupním výkonem $6\,\mathrm{W_p}$ pro napětí 12 V. Když budeme uvažovat 6 hodin denního svitu, tak celkovou vyprodukovanou energii za ideálních podmínek můžeme vypočítat pomocí vzorce na výpočet výkonu:

$$\mathrm{E} = \mathrm{P} * \mathrm{t} = 6\,\mathrm{W_p} * 6\,\mathrm{h} = 36\,\mathrm{Wh}. \tag{4.1}$$

Za 6 hodin slunečního svitu je tedy solární panel schopen vyprodukovat 36 Wh energie, což je asi dvojnásobek toho, co je potřeba na jeden ozonovací cyklus.

Jako poslední je třeba zvolit dostatečně velký úložný prostor pro vyprodukovanou energii. Pro udržení systému při chodu i za nevhodných podmínkách slunečního svitu byl vybrán akumulátor kapacitou 6800 mAh. Při denní spotřebě 17 Wh, která byla určena výše, je možné vypočítat, kolik energie z 12 V baterie vezme jeden cyklus:

$$\mathrm{C_1} = \frac{\mathrm{P}}{\mathrm{U}} = \frac{17\,\mathrm{Wh}}{12\,\mathrm{V}} = 1,42\,\mathrm{Ah}. \tag{4.2}$$

Z výpočtu (4.2) je patrné, že jeden cyklus spotřebuje z baterie 1,42 Ah, což je 1400 mAh. V baterii o kapacitě 6800 mAh, je tedy k dispozici necelých pět cyklů ozonování, což by mělo pokrýt i dny bez nebo s minimem slunečního svitu.

4.2 Realizace systému

Před kompletním sestavením systému, jehož schéma je k vidění na obrázku 4.1 je třeba si určit jaké všechny díly budou potřeba. Kromě hlavních komponentů jako jsou solární panel, dobíjecí baterie, mikrokontroler ESP32 a generátor ozonu jsou to různá spínací relé, rezistory, čidlo na čtení teploty a vlhkosti vzduchu nebo měniče napětí.

Celý systém sice pohání baterie, která je 12V, i solární panel a generátor mají napětí 12 V, ale mikrokontroler 12 V napětí nevydrží a pro bezproblémový chod potřebuje napětí o velikosti 3,3 V. Proto byl do zapojení zvolen nastavitelný step–down měnič napětí (LM2596), který dokáže vstupní napětí v rozmezí od 4,5–35 V snížit na 3–33 V. V mém případě jsem měnič nastavil na snižování napětí z 12 V na potřebných 3,3 V. Ochrana přebití baterie je již její součástí, tudíž nebylo třeba instalovat ochranu externě.

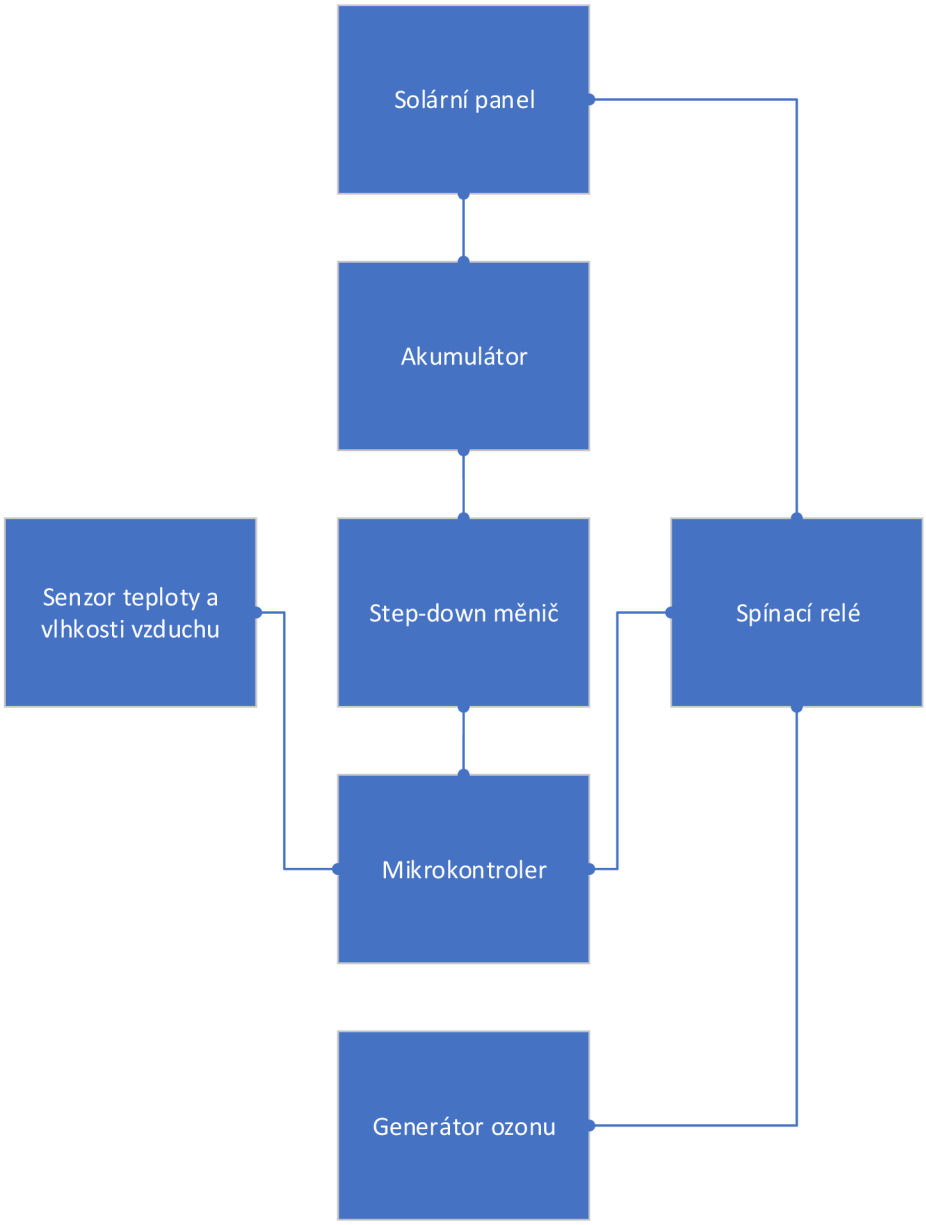
Dalším nezbytným dílem je spínací relé (SV-5A1R1P-N), které bylo vybráno ze stejného důvodu jako step-down měnič, jen s tím rozdílem, že byl problém opačný. Bylo potřeba mikrokontrolerem s napětím 3,3 V sepnout generátor ozonu s napětím 12 V. Spínací relé dokáže tato dvě napětí od sebe rozdělit, takže když mikrokontroler vyšle signál, tak relé sepne obvod s generátorem.

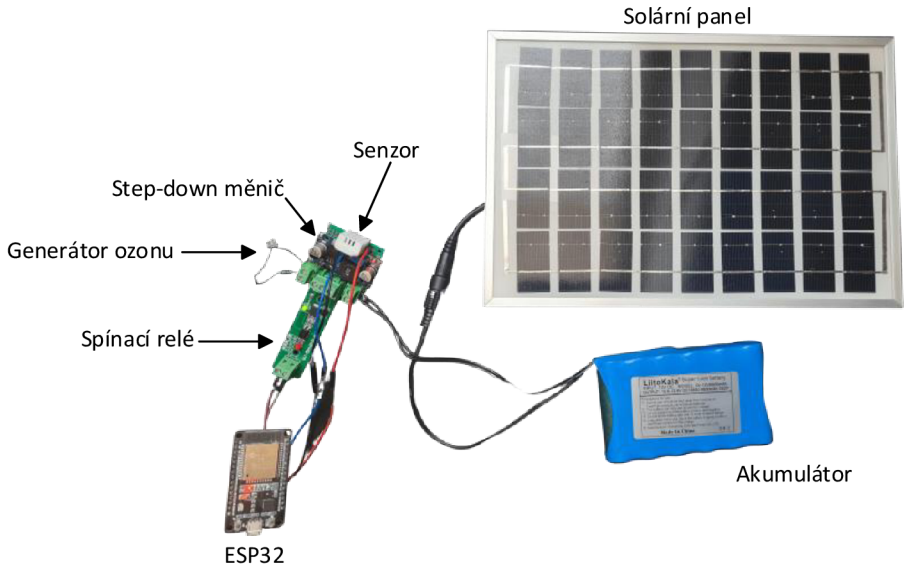
Poslední část je senzor teploty a vlhkosti vzduchu (DHT22). Je třeba senzor použít, protože čištění ovzduší ozonem je účinné jen za určitých podmínek. Když teplota vzduchu klesne pod úroveň 14 stupňů nebo vlhkost vzduchu stoupne nad 75 procent, tak nemá smysl ozonovat, protože nedosáhneme potřebné hustoty ozonu ve vzduchu [35].

Zapojení senzoru probíhá pomocí tří kontaktů. Jeden je mezi uzemňovacími piny ESP32 a senzoru, druhý mezi napájecími piny a třetí mezi datovými piny a mezi datovým pinem senzoru a napájecím pine mikrokontroleru s použitím 10 kΩ rezistoru.

V reálném zapojení, které se nachází na obrázku 4.2, jsou všechny vybrané komponenty, až na generátor ozonu, který je v zapojení nahrazen LED diodou, kvůli jednodušší kontrole funkčnosti systému.

Obr. 4.1: Schéma zapojení systému.





Obr. 4.2: Reálné zapojení systému pro neutralizaci zápachu.

4.3 Vývojové prostředí Arduino IDE

Arduino IDE je open-source software, který se používá k zápisu a nahrávání kódu na desky Arduino. Aplikace je vhodná pro různé operační systémy, jako jsou Windows, MacOS X a Linux. Podporuje programovací jazyky C a C++, ale jazyky jsou zde lehce modifikovány pro programování vývojových desek. Zkratka IDE znamená Integrated Development Environment, neboli integrované vývojové prostředí.

Program nebo kód napsaný v Arduino IDE se často nazývá skicování. Potřebujeme propojit desku Arduino, ESP32 nebo jinou podporovanou desku s IDE, abychom nahráli kód napsaný v softwaru Arduino IDE. Projekt je uložen s příponou „.ino“.

V novém projektu se nachází dvě předepsané funkce, jak je možné vidět na obrázku 4.3. První z nich je „setup“, která má za úkol inicializovat proměnné, nastavovat módy vstupních a výstupních pinů nebo určit jaké knihovny se používají. Tato funkce proběhne vždy pouze jednou a to bezprostředně po nahrání kódu do mikrokontroleru. Druhá funkce je funkce „loop“, která se pouští opakovaně se zpožděním, jaké si může uživatel nastavit. V této funkci se běžně nachází hlavní program. Tato funkce se dá přirovnat k funkci „main“, která se používá v jiných programovacích jazycích, takže je doporučováno si vytvořit různé funkce, které se poté v hlavní funkci postupně vyvolávají.

Obr. 4.3: Vývojové prostředí Arduino IDE.

4.4 Programové řešení

V následující podkapitole je popsáno použité vývojové prostředí a také funkce, které jsou součástí systému. Jsou popsány potřebné knihovny a imlplementace celé funkce. Dále je zde popsán vývojový diagram.

4.4.1 Popis jednotlivých funkcí a výpisy terminálu

Základní funkce setup a loop

Ve funkcích setup a loop, dostupné ve výpisu A.1, jsou pouze nezbytné příkazy pro stanovení pinů na přijímací nebo odesílací. Dále jsou zde vyvolávány jednotlivé funkce nutné pro běh systému.

Funkce připojení k Wi-Fi

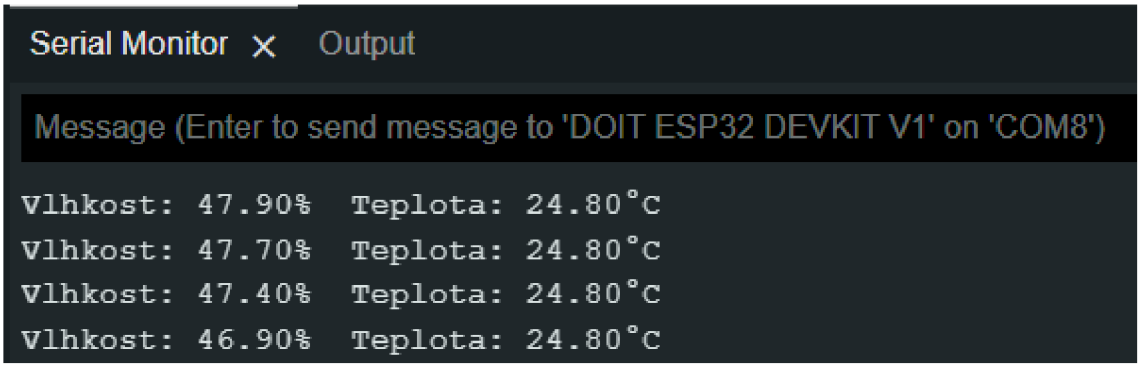
Kvůli této funkci je nejprve nutné importovat knihovnu **WiFi.h**, která zajišťuje možnost připojení mikrokontroleru k Wi-Fi.

V samotné funkci, kterou je možné vidět ve výpisu A.2, je poté zpoždění 3 sekundy, po kterém se ESP32 připojí k předem známé síti, kterou si určíme pomocí konstant. Když by se možnost připojení ztratila a po nějaké době opět obnovila, tak se mikrokontroler znovu připojí.

Funkce čtení dat a následného ozonování

K běhu této funkce je třeba si naimportovat knihovnu **DHT22**, kterou je třeba do vývojového prostředí přidat, jelikož se narozdíl od knihovny pro přpojení k Wi-Fi v základní verzi nenachází.

Ve funkci se nejprve spojí mikrokontroler se senzorem a vyžádá si vlhkost a teplotu vzduchu. Tyto údaje následně použije v podmínce pro ozonování. Jak bylo již uvedeno v kapitole 4.2, nejvhodnější podmínky pro ozonování jsou při vlhkosti vzduchu pod 75 % a teplotě vzduchu nad 14 °C. Pro kontrolu těchto hodnot je ve funkci implementován výpis do terminálového okna 4.4. Celý kód této funkce je k dispozici ve výpisu A.3.



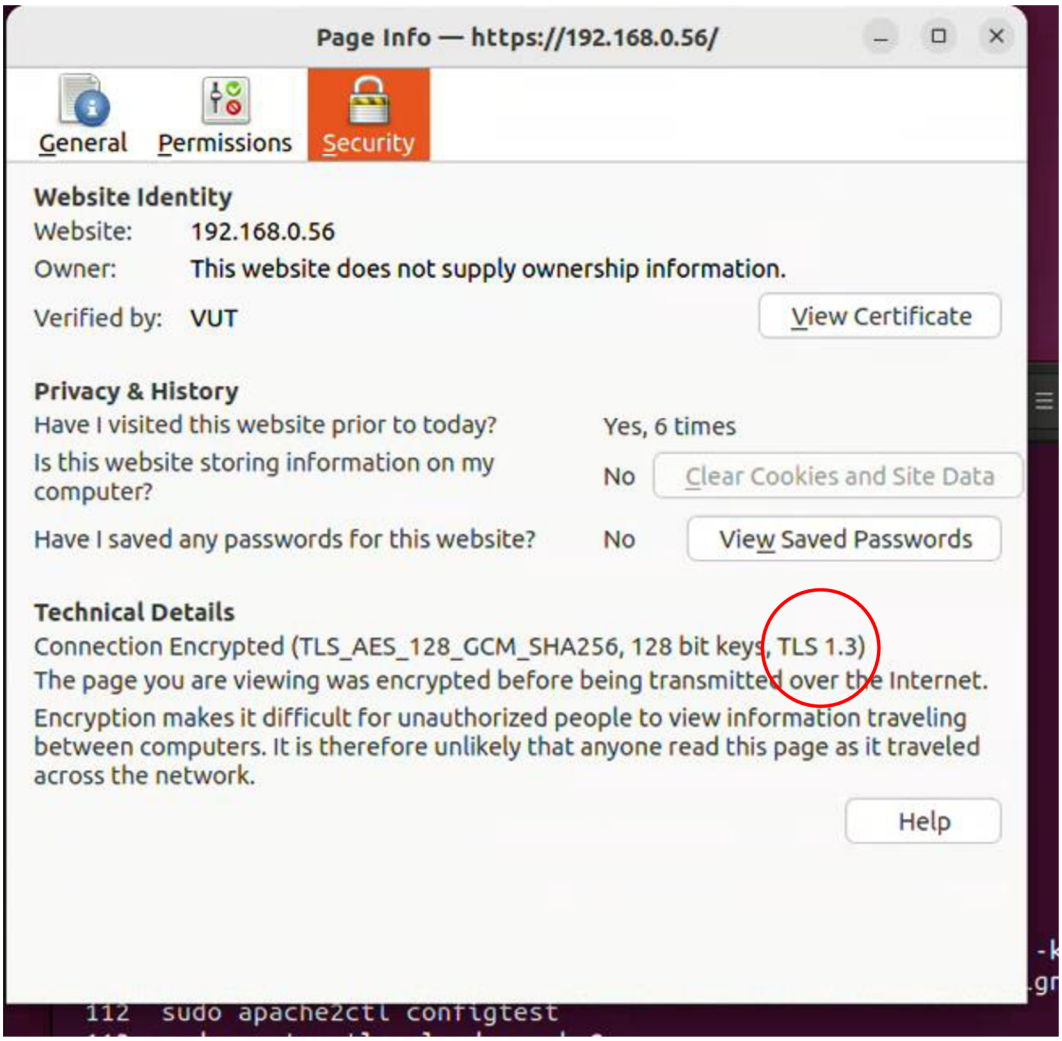
Obr. 4.4: Výpis hodnot do terminálu.

Funkce odesílání dat na server

Pro správné fungování funkce pro odesílání sesbíraných dat na stranu serveru je potřeba importovat knihovnu **HTTPClient.h**, která zajišťuje navázání spojení mezi klientem, v tomto případě mikrokontrolerem ESP32 a serverem, v tomto případě Apache, který je zabezpečen nejnovějším protokolem TLS 1.3, jak ej možné vidět na obrázku 4.5.

Samotný přenos je provozován pomocí funkce na mikrokontroleru a PHP skriptem, běžícím na serveru. Ve výpisu A.4 je k vidění celá funkce. První je třeba začít komunikaci a definovat adresu serveru společně s cestou k PHP souboru, kde se data budou následně zpracovávat. Dále je přidána hlavička paketu a data, která je třeba dostat na server. Při úspěšném přenosu se do kontrolního terminálového okna vypíše potvrzení, že vše proběhlo jak mělo. V opačném případě se vypíše chybový kód. Při řešení práce jsem nejčastěji natrefoval na kód chyby *404* (nedostupný server) nebo kód *-1* (chybné připojení mikrokontroleru k WiFi).

Druhá část potřebná k přenosu dat na server je PHP skript zpracovávající HTTP POST zprávy. Tento skript je k nahlédnutí ve výpisu A.5, kde se ze začátku definuje, že se jedná o PHP skript. Dále při příchozí zprávě typu POST se data z požadavku vezmou a vloží do textového souboru, ze kterého se následně zobrazí na webové stránce. Při úspěšném i neúspěšném pokusu o přijetí dat je vypsána kontrolní zpráva.

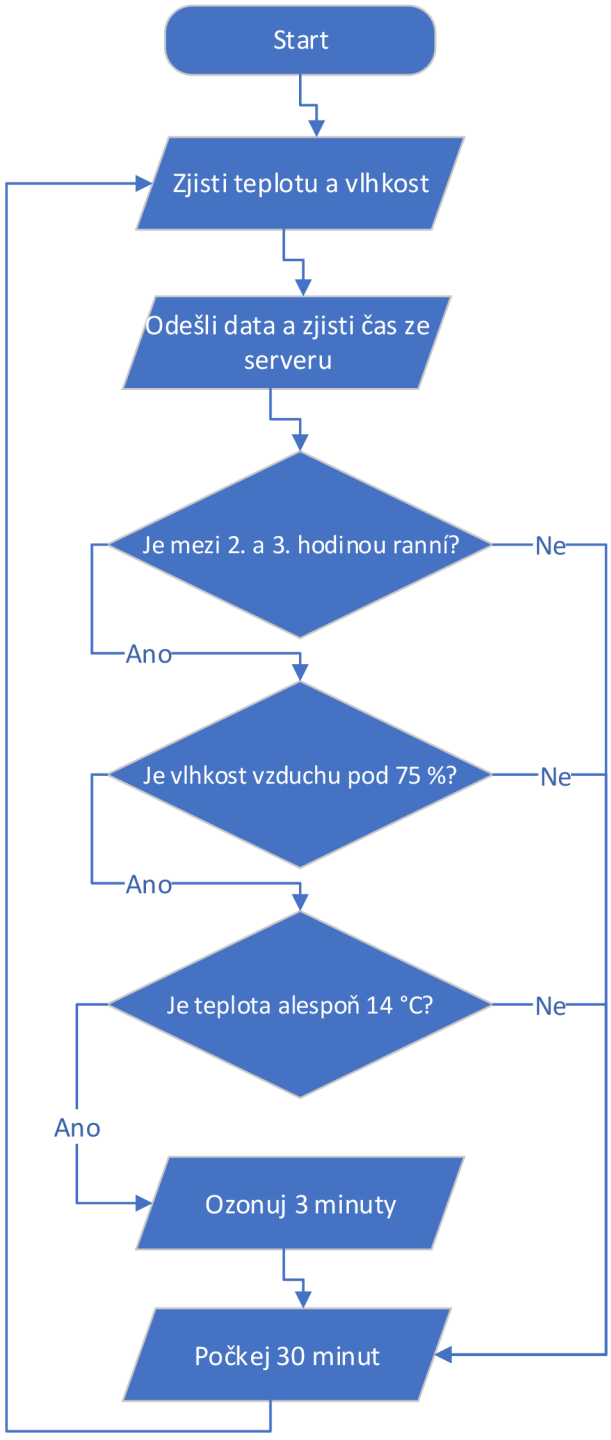


Obr. 4.5: Zabezpečení serveru protokolem TLS.

4.4.2 Vývojový diagram

V programové části práce bylo třeba zajistit, kdy se bude generátor ozonu spouštět a jak a také kdy se budou odesílat data na webový server. Na obrázku 4.6 je vývojový diagram ze kterého je patrné fungování systému. Nejprve se zjistí teplota a vlhkost vzduchu ze senzoru a tato data jdou k odeslání na webový server. S odeslanými daty jde na server také dotaz na aktuální čas, který se využívá v další funkci, která postup pustí dál, jen když je brzo ráno, tudíž je velmi malá šance na otevření popelnice a způsobení nějaké škody na zdraví. Když je tato podmínka splněna, tak přichází další již probírané podmínky v kapitole 4.2, vlhkost a teplota vzduchu. Když i tyto podmínky jsou splněny, spustí se generování ozonu na určitou dobu nutnou k neutralizaci zápachu v popelnici. Po dokončení ozonování nebo při nesplnění jedné

z podmínek se v cyklu počká 30 minut a celý postup jede znovu od začátku. Je tedy zřejmé, že na webový server se data o teplotě a vlhkosti dostanou jednou za půl hodiny a synchronizace času probíhá také jednou za půl hodiny.



Obr. 4.6: Vývojový diagram.

Závěr

Cílem práce bylo aplikovat zabezpečený systém neutralizace zápachu. V teoretické části bylo popsán pojem Internet of Things a možnost snímání dat v IoT. Dále byl popsán přenos dat pomocí protokolů MQTT (Message Queue Telemetry Transport), WebSocket a HTTP (Hypertext Transfer Protocol) a možnosti zabezpečení pomocí protokolů SSL (Secure Sockets Layer) a TLS (Transport Layer Security). Následně byl popsán asi nejrozšířenější způsob pro přenos dat (Wireless Fidelity) a její zabezpečení WEP (Wired Equivalent Privacy), WPA (Wi-Fi Protected Access), WPA2 a WPA3. Také byl ujasněn pojem webový server a dva jeho nejběžnější zástupci, Apache a IIS (Internet Information Services). V práci je také popsán vznik a výskyt ozonu. Jako poslední součást praktické části práce je popis jednotlivých komponent. V praktické části byl popsán postup při výběru komponent, sestavení systému a také zde bylo popsáno vývojové prostředí s jednotlivými funkcemi.

Systém je schopný snímat data, analyzovat je, a v případě splnění podmínek teoreticky začít generovat ozon. Limity systému tkví v přenášení dat na webový server. Systém se bez problémů připojí k Wi-Fi, ale data následně již nepošle dále na server. Další limity by mohly být ve spotřebě zařízení, ale ta se projeví až při testování funkčnosti. Do budoucna by se tedy mohl vylepšit přenos dat na server a případně vyšší odolnost zařízení před přírodními, ale hlavně lidskými faktory. Do budoucna by se kromě spotřeby zařízení mohlo implementovat posílání dat pomocí protokolu MQTT místo protokolu HTTP.

Literatura

[1] What is IoT. *Oracle* [online]. 2022 [cit. 8. 12. 2022]. 7 Dostupné z URL: <https://www.oracle.com/internet-of-things/what-is-iot/>

[2] Sensors in Internet of Things(IoT). *GeeksForGeeks* [online]. Noida, 2021 [cit. 8. 12. 2022]. Dostupné z URL: <https://www.geeksforgeeks.org/sensors-in-internet-of-thingsiot/>

[3] FALUDI, Rob. How Do IoT Devices Communicate. *Digi* [online]. 2021 [cit. 8. 12. 2022]. Dostupné z URL: <https://www.digi.com/blog/post/how-do-iot-devices-communicate>

[4] Internet of Things – Wireless data transfer technologies. *SOS electronic* [online]. 2017 [cit. 8. 12. 2022]. Dostupné z URL: <https://www.soselectronic.com/articles/sos-supplier-of-solution/internet-of-things-wireless-data-transfer-technologies-part-2-2043>

[5] GUDELIAUSKAS, Domantas. What is FTP. *Hostinger* [online]. 2022 [cit. 9. 12. 2022]. Dostupné z URL: <https://www.hostinger.com/tutorials/what-is-ftp#What_Is_FTP>

[6] File Transfer Protocol (FTP). *GeeksForGeeks* [online]. 2021 [cit. 9. 12. 2022]. Dostupné z URL: <https://www.geeksforgeeks.org/file-transfer-protocol-ftp/>

[7] What is SSL. *Cloudfare* [online]. [cit. 9. 12. 2022]. Dostupné z URL: <https://www.cloudflare.com/learning/ssl/what-is-ssl/>

[8] What is Transport Layer Security (TLS). *Cloudfare* [online]. [cit. 9. 12. 2022]. Dostupné z URL: <https://www.cloudflare.com/learning/ssl/transport-layer-security-tls/>

[9] TCP 3-Way Handshake Process. *GeeksForGeeks* [online]. 2021 [cit. 8. 5. 2023]. Dostupné z URL: <https://www.geeksforgeeks.org/tcp-3-way-handshake-process/>

[10] The TLS Handshake Explained. *Auth0 - blog* [online]. 2023 [cit. 8. 5. 2023]. Dostupné z URL: <https://auth0.com/blog/the-tls-handshake-explained/>

[11] TCP Connection Termination. *GeeksForGeeks* [online]. 2022 [cit. 8. 5. 2023]. Dostupné z URL: <https://www.geeksforgeeks.org/tcp-connection-termination/>

[12] GHIMIRAY, Deepan. Wi-Fi Security: WEP vs WPA or WPA2. *Avast* [online]. 2022 [cit. 10. 12. 2022]. Dostupné z URL: <https://www.avast.com/c-wep-vs-wpa-or-wpa2#topic-6>

[13] Discover Wi-Fi: Security. *Wi-Fi Alliance®* [online]. [cit. 9. 5. 2023]. Dostupné z URL: <https://www.wi-fi.org/discover-wi-fi/security>

[14] What is Wi-Fi Protected Access (WPA). *Sunny Valley Networks* [online]. [cit. 9. 5. 2023]. Dostupné z URL: <https://www.sunnyvalley.io/docs/network-security-tutorials/what-is-wpa>

[15] The differences between WPA-Personal and WPA-Enterprise. *TP-LINK* [online]. 2013 [cit. 9. 5. 2023]. Dostupné z URL: <https://www.tp-link.com/cz/support/faq/500/>

[16] What Is MQTT. *Amazon Web Services* [online]. 2023 [cit. 17. 5. 2023]. Dostupné z URL: <https://aws.amazon.com/what-is/mqtt/>

[17] Network Admin's Guide to Synthetic Monitoring: MQTT Broker. *Catchpoint Systems* [online]. 2023 [cit. 17. 5. 2023]. Dostupné z URL: <https://www.catchpoint.com/network-admin-guide/mqtt-broker>

[18] What is web socket and how it is different from the HTTP. *GeeksforGeeks* [online]. 2023 [cit. 17. 5. 2023]. Dostupné z URL: <https://www.geeksforgeeks.org/what-is-web-socket-and-how-it-is-different-from-the-http/>

[19] What is a REST API. *Red Hat* [online]. 2020 [cit. 17. 5. 2023]. Dostupné z URL: <https://www.redhat.com/en/topics/api/what-is-a-rest-api>

[20] What is Web Server. *The Economic Times* [online]. 2022 [cit. 10. 12. 2022]. Dostupné z URL: <https://economictimes.indiatimes.com/definition/web-server>

[21] B., Richard. What Is Apache. *Hostinger* [online]. 2022 [cit. 10. 12. 2022]. Dostupné z URL: <https://www.hostinger.com/tutorials/what-is-apache#What_Is_a_Web_Server>

[22] What Is IIS Server. *SolarWinds* [online]. 2022 [cit. 10. 12. 2022]. Dostupné z URL: <https://www.solarwinds.com/resources/it-glossary/iis-server>

[23] OGUUJIOFOR, Kamso. Apache vs IIS – What's the Difference between Web Servers. *Cloud Infrastructure Services* [online]. 2022 [cit. 11. 12. 2022].

Dostupné z URL: <https://cloudinfrastructureservices.co.uk/apache-vs-iis-whats-the-difference-between-web-servers/>

[24] KUČEROVÁ, Lída. Ozonový generátor. *NanoSPACE* [online]. Domažlice: nanoSPACE, 2021 [cit. 16. 11. 2022]. Dostupné z URL: <https://www.nanospace.cz/blog/ozonovy-generator/>

[25] Mastering the Fundamentals of Ozone: Ozone Generation. *Wqpmag* [online]. Benicia: wqpmag, 2020 [cit. 24. 11. 2022]. Dostupné z URL: <https://www.wqpmag.com/water-disinfection/ozone-systems/article/10954859/mastering-the-fundamentals-of-ozone-ozone-generation>

[26] HOFFMANN, Clotilde, BERGANZA Carlos a ZHANG John. Atmospheric Plasma: methods of production and application in dentistry and oncology [online]. 2013, 3(21) [cit. 24. 11. 2022]. Dostupné z URL: <https://doi.org/10.1186/2045-9912-3-21>

[27] An Overview of Development Board. *Utmel* [online]. 2021 [cit. 28. 11. 2022]. Dostupné z URL: <https://www.utmel.com/blog/categories/pcb/an-overview-of-development-board>

[28] Getting Started with the ESP32 Development Board. *Random Nerd Tutorials* [online]. [cit. 28. 11. 2022]. Dostupné z URL: <https://randomnerdtutorials.com/getting-started-with-esp32/>

[29] ESP32 Pinout Reference. *Random Nerd Tutorials* [online]. [cit. 28. 11. 2022]. Dostupné z URL: <https://randomnerdtutorials.com/esp32-pinout-reference-gpios/>

[30] How does solar power work. *National Grid* [online]. [cit. 29. 11. 2022]. Dostupné z URL: <https://www.nationalgrid.com/stories/energy-explained/how-does-solar-power-work>

[31] What is a solar panel. *MrSolar* [online]. [cit. 29. 11. 2022]. Dostupné z URL: <https://www.mrsolar.com/what-is-a-solar-panel/>

[32] VOURVOULIAS, Aris. Pros & Cons of Solar Energy. *GreenMatch* [online]. 2022 [cit. 29. 11. 2022]. Dostupné z URL: <https://www.greenmatch.co.uk/blog/2014/08/5-advantages-and-5-disadvantages-of-solar-energy>

[33] Rechargeable battery. *WhatIs* [online]. [cit. 29. 11. 2022]. Dostupné z URL: <https://www.techtarget.com/whatis/definition/rechargeable-battery>

[34] . VASHISTA, Sumit. Lithium-ion vs Lithium-Polymer Batteries. *LinkedIn* [online]. 2022 [cit. 29. 11. 2022]. Dostupné z URL: <https://www.linkedin.com/pulse/lithium-ion-vs-lithium-polymer-batteries-sumit-vashista>

[35] Čištění ozónem. *OzonGeneratory* [online]. [cit. 6. 12. 2022]. Dostupné z URL: <https://www.ozongeneratory.cz/cisteni-ozonem>

Seznam symbolů a zkratek

°C	stupeň Celsia
%	procento
AC	Alternating Current
AES	Advanced Encryption Standard
Ah	ampérhodina
AWS	Amazon Web Services
BLE	Bluetooth Low Energy
CA	Certificate Authority
DPP	Device Provisioning Protocol
DC	Direct Current
EPA	Environmental Protection Agency
FTP	File Transfer Protocol
g	gram
g/h	gram za hodinu
GHz	gigahertz
GND	ground
GSM	Groupe Spécial Mobile
GPIO	General-purpose input/output
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
ID	Identification
IDE	Integrated Development Environment
IEEE	Institute of Electrical and Electronics Engineers

IETF	Internet Engineering Task Force
IoT	Internet of Things
IP	Internet Protocol
IIS	Internet Information Services
kΩ	kiloohm
kB	kilobyte
LAN	Local Area Network
LED	Light-Emitting Diode
Li-Ion	Lithium-ion
Li-Pol	Lithium-polymer
LTE	Long Term Evolution
LoRa	Long Range
LoraWAN	Long Range Wide Area Network
m	metr
mAh	miliampérhodina
MB	megabyte
Mbit/s	megabit za sekundu
MHz	megahertz
MQTT	Message Queue Telemetry Transport
NVT	Network Virtual Terminal
NFC	Near Field Communication
OASIS	Organization for the Advancement of Structured Information Standards
PHP	Hypertext Preprocessor
QR	Quick Response

RADIUS	Remote Authentication Dial In User Service
RAM	Random Access Memory
REST	Representational State Transfer
REST API	Representational State Transfer Application Programming Interface
RTU	Remote Terminal Unit
SCADA	Supervisory Control And Data Acquisition
SCTP	Stream Control Transmission Protocol
SMTP	Simple Mail Transfer Protocol
SoC	System on a Chip
SSL	Secure Sockets Layer
TCP/IP	Transmission Control Protocol over Internet Protocol
TELNET	Teletype Network
TKIP	Temporal Key Integrity Protocol
TLS	Transport Layer Security
UART	Universal asynchronous receiver-transmitter
URL	Uniform Resource Locator
USA	United States of America
USB	Universal Serial Bus
UV	Ultraviolet
V	volt
VoIP	Voice over Internet Protocol
VPS	Virtual Private Server
W	Watt
WEP	Wired Equivalent Privacy
Wh	Watthodina

Wi-Fi	Wireless Fidelity
WLAN	Wireless Local Area Network
WPA	Wi-Fi Protected Access
WPA-EAP	Wi-Fi Protected Access Enterprise
WPA-PSK	Wi-Fi Protected Access Personal
WPS	Wi-Fi Protected Setup

A Výpisy kódu

Výpis A.1: Základní metody mikrokontroleru ESP–32.

```
void setup() {
    Serial.begin(115200);
    pinMode(ozoPin, OUTPUT);
}

void loop() {
    wifiReconnect();
    senzorData();
}
```

Výpis A.2: Připojení k WiFi.

```
void wifiReconnect() {
    delay(3000);
    if(WiFi.status() != WL_CONNECTED) {
        WiFi.begin(ssid, password);
        Serial.print("[*] Informace o síti ");
        Serial.println(ssid);
        Serial.print("[+] Výchozí brána : ");
        Serial.println(WiFi.gatewayIP());
        Serial.print("[+] Maska : ");
        Serial.println(WiFi.subnetMask());
        Serial.print("[+] ESP32 IP : ");
        Serial.println(WiFi.localIP());
      }
}
```

Výpis A.3: Sběr dat a ozonování.

```
void dataOzone() {
  delay(2000);

  dht.begin();

  float h = dht.readHumidity();
  float t = dht.readTemperature();

  Serial.print(F("Vlhkost: "));
  Serial.print(h);
  Serial.print(F("%  Teplota: "));
  Serial.print(t);
  Serial.println(F("C "));

if (h<75 && t>14) {
    digitalWrite(ozoPin, HIGH);
    delay(180000);
    digitalWrite(ozoPin, LOW);
  }
  return;
}
```

Výpis A.4: Odesílání dat na server – část mikrokontroleru.

```
HTTPClient http;
http.begin("https://192.168.0.56/receiver.php");
http.addHeader("Content-Type", "text/plain");
int httpResponseCode = http.POST(h,t);
if (httpResponseCode == 200) {
   Serial.println("Data poslána.");
} else {
    Serial.print("Chyba při přenosu. Kód chyby: ");
    Serial.println(httpResponseCode);
}
http.end();
```

Výpis A.5: Odesílání dat na server – část serveru.

```
<?php
if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    $data = file_get_contents('php://input');
    file_put_contents('data.txt', $data);
    echo "Data␣prijata.";
} else {
    echo "Neplatná␣zadost.";
}
?>
```