

Česká zemědělská univerzita v Praze
Provozně ekonomická fakulta
Katedra informačního inženýrství



Diplomová práce
Chytrá domácnost

Jan Jakubík

ZADÁNÍ DIPLOMOVÉ PRÁCE

Bc. Jan Jakubík

Systémové inženýrství a informatika
Informatika

Název práce

Chytrá domácnost

Název anglicky

Upgrade to smart home

Cíle práce

Hlavním cílem práce je vytvoření hardwarového a softwarového řešení monitorujících a řídicích prvků pro chytrou domácnost. Dílčím cílem je vytvoření webového rozhraní pro komunikaci mezi uživatelem a chytrou domácností.

Metodika

V diplomové práci bude provedena analýza požadavků konkrétního řešení. Na základě této analýzy bude sestaven návrh na řešení. Řešení bude realizováno pomocí mikropočítače Raspberry Pi a jednotlivých mikrokontrolerů. Bude provedeno sestavení, naprogramování a tvorba rozhraní. Jednotlivé komponenty budou komunikovat přes bezdrátovou síť s řídicí jednotkou. Komunikace mezi uživatelem a chytrou domácností bude zprostředkována pomocí webové stránky.

Doporučený rozsah práce

50-60 stran

Klíčová slova

Raspberry Pi, chytrá domácnost, automatizace, web, HTML

Doporučené zdroje informací

BURIAN, P. Internet inteligentních aktivit. Praha: Grada, 2014. ISBN 978-80-247-5137-5.

GOODWIN, Steven. Smart home automation with Linux and Raspberry Pi. Berkeley: Apress, 2013. ISBN 978-1430258872.

SHOVIC, John C. Raspberry Pi IoT projects: prototyping experiments for makers. New York: Apress, [2016]. Technology in action series. ISBN 1484213785

Předběžný termín obhajoby

2020/21 LS – PEF

Vedoucí práce

Ing. Marek Pícka, Ph.D.

Garantující pracoviště

Katedra informačního inženýrství

Elektronicky schváleno dne 23. 2. 2021

Ing. Martin Pelikán, Ph.D.

Vedoucí katedry

Elektronicky schváleno dne 23. 2. 2021

Ing. Martin Pelikán, Ph.D.

Děkan

V Praze dne 30. 03. 2021

Čestné prohlášení

Prohlašuji, že svou diplomovou práci "Chytrá domácnost" jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu použitých zdrojů na konci práce. Jako autor uvedené diplomové práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne 31.3.2021

Poděkování

Rád bych touto cestou poděkoval panu Ing. Marek Píčka Ph.D. za vedení mé práce. Také bych rád poděkoval své rodině a přítelkyni za jejich podporu v mém studiu.

Chytrá domácnost

Abstrakt

Diplomová práce se zabývá problematikou návrhu a realizace systému pro chytrou domácnost. Hlavní náplní práce chytré domácnosti je sběr dat naměřených veličin, ovládání zařízení a automatizace.

V rámci teoretické části práce jsou představeny jednotlivé komunikační protokoly využívané v chytrých domácnostech a průmyslových řešeních v oblasti internetu věcí. V této části jsou také popsány jednotlivá softwarová vybavení potřebná ke zprovoznění centrální řídicí jednotky a zajištění její funkčnosti i jako file serveru a web serveru. Poslední část teoretické části obsahuje představení jednotlivých hardwarových komponent používaných pro realizaci chytré domácnosti.

Na základě teoretických poznatků byl navržen a realizován systém. Tento proces je popsán v praktické části diplomové práce. Nejprve byla provedena analýza požadavků na konkrétní systém. Na základě výstupu analýzy požadavků bylo navrženo řešení systému. Další částí praktické části byla realizace navrhovaného systému a ta obsahovala instalaci a konfiguraci jednotlivého programového vybavení, sestavení jednotlivých částí systému a jejich programování. Poslední částí realizace bylo programování serverových aplikací pro zajištění chodu chytré domácnosti a vytvoření webové aplikace pro její ovládání.

Klíčová slova: Raspberry Pi, Arduino, Arduino IDE, Python, C, Javascript, HTML, MQTT, chytrá domácnost, automatizace, web

Upgrade to smart home

Abstract

This diploma thesis is focused on designing and implementation of a system for smart home. The main tasks of smart home is data collection of measured quantities, device control and automation.

The theoretical part of the thesis presents individual communication protocols used in smart homes and industrial solutions in the field of the Internet of Things. This section also describes the individual software equipment needed to put the central control unit into operation and ensure its functionality as a file server and web server. The last part of the theoretical part contains an introduction of individual hardware components used for implementation of the smart home.

Based on theoretical knowledge, a system was designed and implemented. This process is described in the practical part of the thesis. First, an analysis of the requirements for a specific system was performed. Based on the output of the requirements analysis, a system solution design was created. Another part of the practical part was the implementation of the design and that included the installation and configuration of individual software, assembly of individual parts of the system and their programming. The last part of the implementation was the programming of server applications to ensure the operation of a smart home and creation of a web application for its control.

Keywords: Raspberry Pi, Arduino, Arduino IDE, Python, C, Javascript, HTML, smart home, automation, web

Obsah

1 Úvod	11
2 Cíl práce a metodika	12
2.1 Cíl	12
2.2 Metodika	12
3 Teoretická východiska	13
3.1 Protokoly	13
3.1.1 Wifi	13
3.1.2 Bluetooth	16
3.1.3 ZigBee	16
3.1.4 MQTT	20
3.1.5 Z-Wave	22
3.1.6 SMB	24
3.1.7 FTP	26
3.2 Software	27
3.2.1 Arduino IDE	27
3.2.2 DietPi	27
3.2.3 Apache	27
3.2.4 NGINX	28
3.2.5 Samba	29
3.3 Hardware	30
3.3.1 Raspberry Pi	31
3.3.2 Arduino	31
3.3.3 ESP 8266	32
3.3.4 CC2531	32
3.3.5 BME280	32
3.3.6 DHT22	33
4 Analýza požadavků	34
4.1 Use case	34
4.2 Use case diagram	34
	35
4.3 Use case scénáře	35
4.3.1 Spuštění webové aplikace	35
4.3.2 Nastavení automatického zhasínání světla	36
4.3.3 Nastavení automatického spuštění režimu	36
4.3.4 Rozsvícení světla a nastavení jasu a teploty	37
4.3.5 Nahrání souboru na sdílený disk SMB	37

5 Návrh řešení	38
5.1 Model systému	38
5.2 Wireframy	39
5.3 Databázový model	40
6 Realizace řešení	42
6.1 Operační systém	43
6.2 Access point	44
6.3 Web server	45
6.4 MQTT broker	45
6.5 Zigbee2mqtt	45
6.6 Databáze	46
6.7 File server	47
6.8 Venkovní senzorová stanice	47
6.9 Serverové aplikace	49
6.9.1 Zapisovač naměřených dat	50
6.9.2 Program vnitřního teploměru	50
6.9.3 Program časovač	51
6.9.4 Hlavní aplikace	51
6.10 Webová aplikace	52
7 Výsledky	54
7.1 Hodnocení vytvořeného systému	54
7.1.1 Klady	54
7.1.2 Zápory	54
7.2 Finální podoba webového rozhraní	55
7.3 Ukázky kódů	57
8 Závěr	59
9 Seznam použitých zdrojů	61

Seznam obrázků

Obrázek 1: Zigbee stack layer.....	18
Obrázek 2: Use case diagram.....	35
Obrázek 3: Model systému.....	39
Obrázek 4: Wireframy webové aplikace.....	40
Obrázek 5: Návrh datbázových tabulek.....	42
Obrázek 6: Centrální řídicí jednotka Raspberry Pi.....	43
Obrázek 7: Venkovní senzorová stanice.....	49

Obrázek 8: Webové rozhraní Hlavní panel pro desktop.....	55
Obrázek 9: Webové rozhraní Automatizace pro desktop.....	55
Obrázek 10: Webové rozhraní Automatizace pro mobil.....	56
Obrázek 11: Webové rozhraní Hlavní panel pro mobil.....	56
Obrázek 12: Program senzorové stanice.....	57
Obrázek 13: Program vnitřního teploměru.....	58
Obrázek 14: Ukázka Javascriptu Webové aplikace.....	58

1 Úvod

Tato práce pojednává o problematice navrhování a realizování chytrých domácností. V dnešní uspěchané době jsou kladeny větší a větší nároky na automatizaci všech možných činností. V našich domácnostech se nachází spousta elektronických zařízení, které musíme v případě potřeby zapínat a vypínat každé zvlášť. Řešení chytrých domácností a budov nabízí řada firem. Firmy nabízejí buď již hotová řešení s možností minimálních změn, které jsou cenově dostupné a nebo řešení na míru, která jsou řádově dražší. V momentě, kdy si zvolíme jednoho konkrétního distributora řešení jsme odkázáni pouze na jeho technickou podporu a jeho produkty. Tato práce řeší problematiku již vybaveného bytu či postaveného domu ve kterém není v zájmu předělávat elektroinstalaci. Jde tedy o přechod z klasické domácnosti na domácnost chytřejší za užití primárně bezdrátových technologií. Výhodou bezdrátového řešení je jeho jednoduchá přizpůsobitelnost a přenosnost. V případě rekonstrukce či stěhování lze řešení pouze odpojit a přemístit. Nastavení zůstanou stejná a není je tedy nutno přeprogramovávat či znovu připojovat a odpojovat dráty. Odpadávají i problémy s řešením kudy dráty vést, aby nepůsobili neesteticky a z jakých materiálů dráty používat a s jakou ochranou, aby se navzájem neovlivňovali či nekorodovaly.

2 Cíl práce a metodika

2.1 Cíl

Hlavním cílem práce je navržení, vytvoření a implementování hardwarového a softwarového řešení monitorujících a řídicích prvků pro chytrou domácnost. Dílčím cílem je vytvoření webového rozhraní pro komunikaci mezi uživatelem a chytrou domácností.

2.2 Metodika

Na počátku práce je provedena analýza požadavků na základě které, budou vyvozeny případy užití a jejich scénáře. Výsledky počáteční analýzy nároků na řešení budou výchozím bodem pro vytváření návrhu řešení. V rámci návrhu řešení bude navržena architektura systému, databázový model a webové rozhraní v podobě wireframů. Na základě návrhu řešení bude systém realizován. Prvním krokem bude instalace operačního systému pro řídicí jednotku a jeho konfigurace. V dalším kroku bude nainstalován software pro vytvoření přístupového bodu na centrální řídicí jednotce a jeho konfigurace. Na centrální jednotce bude zprovozněn webový server nutný pro realizaci webového rozhraní a MQTT broker zajišťující zasílání zpráv v rámci systému. Dalším krokem bude sestavení a naprogramování jednotlivých částí systému. V další fázi realizace budou naprogramovány serverové aplikace, které zajistí chod celého systému. Poslední částí bude realizace dílčího cíle a to webové aplikace. Webová aplikace bude realizována na základě wireframů a bude vytvořena v jazyce HTML a Javascript. Jednotlivé serverové aplikace budou naprogramovány v jazyce Python. Mikrokontrolery v systému budou naprogramovány v jazyce C a budou k tomu využity nástroje Arduino IDE. V závěru práce bude tento systém zhodnocen včetně návrhů na budoucí vývoj.

3 Teoretická východiska

3.1 Protokoly

Komunikace jednotlivých zařízení v chytré domácnosti probíhá přes síť a to jak pevnou drátovou, tak i přes bezdrátovou a nebo kombinací obou. Jednotlivá zařízení spolu komunikují pomocí protokolů, které byly navrženy tak aby spotřebovávaly co nejméně energie.

3.1.1 Wifi

Wifi je standard označován IEEE 802.11. Tento bezdrátový standard je v dnešní době běžně používán v domácnostech. Technologie Wifi využívá bezlicenčního frekvenčního pásma. Využívaná frekvenční pásma jsou 2,4 GHz a 5 Ghz. Dosah signálu je přibližně 50 až 100 metrů. Maximální rychlost je 600 Mb/s. Rychlost se nejčastěji pohybuje mezi 150-200 Mb/s. Záleží na frekvenci kanálu, který je používán a na počtu antén. S využitím nejrozšířenějšího standardu 802.11-ac by mohlo být dosahováno rychlostí 500 Mb/s až 1 Gb/s.

Wifi je jeden z nejpoužívanějších protokolů pro internet věci. Tento druh připojení je vhodný pro lokální síť a umožňuje rychlý přenos velkých dat. Toto se může hodit pro různé domácí file servery a streamovací služby. Na druhou stranu spotřeba energie je příliš vysoká a proto není vhodné používat wifi protokol na všechny aplikace.

Roku 1997 byla publikována specifikace standardu bezdrátové sítě s označením IEEE 802.11. Specifikaci vydal mezinárodní institut pro standardizaci IEEE což je anglická zkratka Institute of Electrical and Electronic Engineers. O dva roky později byl standard rozšířen o dvě specifikace a to 802.11a a 802.11b. Ve stejném roce byla založena i aliance WECA (Wireless Ethernet Compatibility Alliance). WECA byla v roce 2002 přejmenována na Wi-Fi Alliance. Wi-Fi Alliance je zodpovědná za přidělování oprávnění používat logo Wi-Fi pro zařízení, která prokáží schopnost plně fungovat se sítí Wi-Fi. V roce 2003 byl publikován standard 802.11g díky kterému, se teoretické přenosové rychlosti zvedly na 54 Mbit/s a to v pásmu 2,4 Ghz. Reálná rychlost je přibližně poloviční. Standard

802.11n se ocitl na světle světa roku 2008 a mimo zvýšení rychlosti přenosu přinesl podporu frekvence 5 Ghz a také technologii MIMO (Multiple Input Multiple Output). Technologie MIMO umožňuje komunikaci zařízeníů současně na více frekvencích a to podle počtu antén. Roku 2013 byl vydán standard 802.11ac a o rok později 802.11ad.

Původní ideou Wi-Fi bylo připojení k lokální síti mobilních zařízeníů pomocí bezdrátového přenosu například v rámci firmy. Později se však začala používat i k připojení do sítě internet za pomoci takzvaných hotspotů. S technologií se lze setkat téměř ve všech mobilních zařízeníů jako jsou například telefony, tablety a notebooky. Konkurencí pro Wi-Fi měl být WiMAX ten se ale neuchytil i když měl nabízet zlepšení dosahu sítě.

Nevýhodami a bezpečnostními riziky jsou paradoxně bezdrátový přenos. Útočník, který chce napadnout síť nemusí reálně vstoupit do zabezpečené budovy a připojit se do fyzické sítě v reálném světě. Místo toho stačí, když útočník je v dosahu signálu a odtud si může nerušeně sledovat přenos na síti, vytvářet ho či rušit. Další možností využití i zneužití je lokalizace zařízeníů pomocí více přístupových bodů. Toto se dá však použít i obráceně a tedy zjistit kdo kde je v budově. Program inSSIDer a NetStumbler (pro Windows) jsou programy, které zjišťují informace o okolních bezdrátových sítích. Proces shromažďování informací o umístění bezdrátových sítích se nazývá Warchalking nebo také Wardriving. Program Kismet umožňuje sledovat provoz na bezdrátové síti. Je tedy užitečným programem pro boj s narušiteli sítě. Na druhou stranu třeba program Aircrack-ng slouží ke zjištění zranitelností sítě.

SSID (Service Set Identifier) je řetězec až 32 ASCII znaků. SSID slouží k rozlišení jednotlivých sítí. Přístupový bod sítě vysílá SSID v pravidelných intervalech a díky tomu si každý potenciální klient může zobrazit základní informace o síti. Pokud chceme bezdrátovou síť schovat lze broadcast SSID vypnout. To má za následek, že klient, který se chce k síti připojit již musí znát SSID dopředu. Jedná se o nejjednodušší zabezpečení sítě. SSID je však přenášeno v čitelné podobě a proto ho lze odposlechnout z komunikace mezi jiným klientem a přístupovým bodem. Komunikaci mezi klientem a přístupovým bodem lze i vyprovokovat zasíláním rámců, které donutí klienty k opětovné asociaci s přístupovým bodem.

Propojení ad-hoc je propojení dvou klientů a pohlíží na sebe jako sobě rovní (peer-to-peer). Klienti se navzájem identifikují pomocí SSID. Druhým a spíše typickým propojením sítě je takzvané infrastrukturní. V takové síti se nachází jeden nebo více přístupových bodů, kteří vysílají každý SSID. SSID v rámci jedné sítě mohou být rozdílné a nebo stejné. V případě stejných SSID lze měnit automaticky, ke kterému přístupovému bodu je klient připojen v závislosti na síle signálu (roaming).

Hlavním problémem bezpečnosti bezdrátové sítě je tedy ona bezdrátovost a fakt, že zařízení využívající bezdrátové připojení se dříve prodávala bez základního nastavení tedy naprosto nezabezpečená. Starší zařízení chráněná pomocí WEP v dnešní době také představují bezpečnostní riziko, protože tato technologie je zastaralá a její bezpečnostní díry již jsou dobře známy. Zabezpečení bezdrátových sítí lze rozdělit do dvou základních kategorií a to šifrování což je zabezpečení přenášených dat, a autorizace což je ověření oprávněných uživatelů.

Zabezpečit síť lze i pomocí kontroly MAC adres. Přístupový bod pracuje s takzvaným white listem a black listem což jsou seznamy povolených a blokových MAC adres. Toto zabezpečení lze však obejít tak, že se zařízení bude prezentovat jinou MAC adresou než je jeho vlastní ideálně tou, která se nachází na white listu nebo která není na black listu.

802.1.X je protokol pro autentizaci, který umožní klientovi komunikaci s třetí stranou, která provede ověření. Se serverem třetí strany komunikuje program, který je nazýván prosebníkem. Příkladem serveru třetí strany je třeba Radius server. Pomocí protokolu 802.X lze odstranit nedostatky zabezpečení pomocí WEP. Zabezpečení WEP je realizováno na základě statických klíčů symetrického šifrování. Klíče jsou nastavovány ručně na přístupovém bodu i na straně klienta. Při použití programu aircrack-ng lze odchytnout konkrétní rámce a po jejich analýze získat klíč poměrně snadno. Kvůli výše zmíněným nedostatkům vzniklo WPA (Wi-Fi Protected Access), které používá PSK (Pre-shared key) nebo Radius server a nebo další varianty EAP (Extensible Authentication Protocol) protokolu. Kvůli zpětné kompatibilitě však nejsou WEP klíče odstraněny, ale

jsou dynamicky měněny na základě protokolů TKIP a CCMP. PSK klíč je prolomitelný pouze slovníkovou metodou aplikovanou na odposlechnutou úspěšnou autentizaci. WPA2 má bezpečnější šifrování AES, které je však výpočetně náročnější. Další variantou jak lze připojit zařízení do sítě je pomocí WPS (Wi-Fi Protected Setup). To probíhá tak, že na přístupovém bodě je tlačítko a po jeho zmáčknutí se lze do sítě přihlásit bez autentizace po určitou dobu a nebo pomocí osmimístného PINu. Lze také kombinovat obojí. PIN je však náchylný k brute force útokům. Proto je vhodné omezit počet pokusů o připojení.

Nejúčinnější metodou pro bezpečné připojení k bezdrátové síti je šifrování veškerých přenášených dat. To má za následek, že útočník i když si odposlechne nějaké informace, tak nemá možnost jak z nich získat například MAC adresy, IP adresy či nešifrované hlavičky HTTPS. V dnešní době připojení k nešifrované síti není ohrožením sítě nebo uživatele, protože komunikace probíhá většinou pomocí HTTPS nebo SSH, které jsou šifrovány už samy o sobě. Stejně tak i integrita dat, která jsou přenášena je hlídána pomocí DNSSEC, elektronického podpisu a jinými metodami.[1][2]

3.1.2 Bluetooth

Standard Bluetooth 4.2 core specification má frekvenci 2,4 GHz a dosah signálu je v průměru 50 až 150 metrů pro Smart a BLE zařízení. Rychlost přenosu dat je 1 Mb/s.

Bluetooth je jedna z nejpoužívanějších technologií pro komunikaci na krátkou vzdálenost. Je vhodná pro posílání malých kousků dat jaké třeba produkují chytré hodinky.

Téměř každé chytré zařízení využívá technologii bluetooth a proto se na trhu objevuje nový Bluetooth Low-Energy (BLE) a Bluetooth Smart. Tyto technologie spotřebovávají mnohem méně energie a proto jsou vhodné pro použití pro internet věcí.[3]

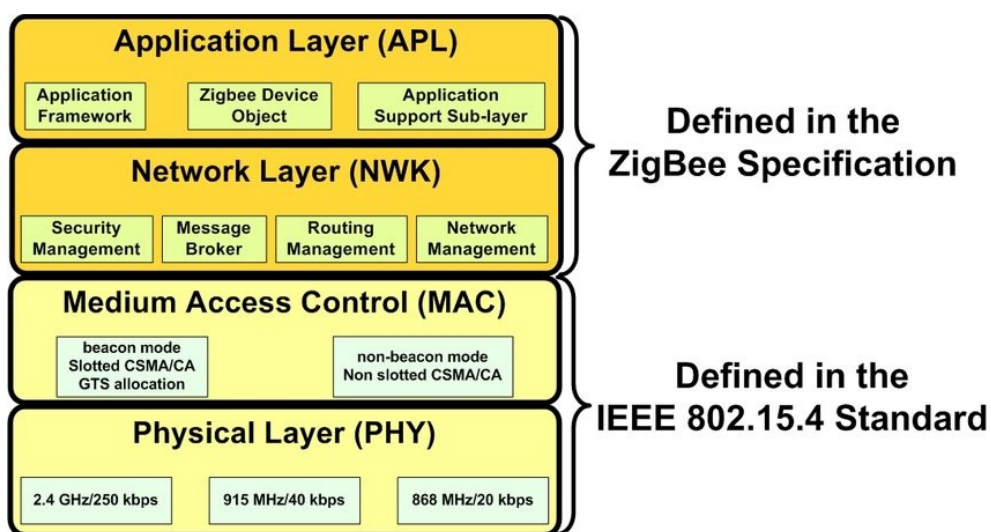
3.1.3 ZigBee

Jedná se o bezdrátovou komunikační technologii založenou na standardu IEEE 802.15.4. ZigBee je platným komunikačním standardem již od roku 2004 a počátky jeho vývoje sahají do roku 1998. Stejně jako již dříve zmíněný Bluetooth je určený pro osobní síť PAN, do kterých se připojují zařízení s nízkým výkonem a na malé vzdálenosti a to do

75 metrů. Avšak komunikace může probíhat i na delší vzdálenosti mezi dvěma zařízeními pokud se mezi nimi nachází další. ZigBee používá takzvané multikioskové ad-hoc směrování. ZigBee je primárně určeno pro průmyslová řešení a pro senzorové sítě. ZigBee operuje v bezlicenčních pásmech a to na 868 MHz, 902-928 MHz a 2,4 Ghz. Rychlost přenosu dat je 20, 40 a 250 kbit/s.

Narozdíl od Bluetooth, které je využíváno hlavně ve spotřební elektronice je ZigBee orientováno pro průmyslové užití. Kvůli nedostatkům Bluetooth standardu byla založena ZigBee aliance, která měla za úkol vytvořit nový komunikační bezdrátový standard právě pro průmyslové a automatizační užití. V současné době se na rozvíjení standardu podílejí i velké nadnárodní firmy z oboru automatizace jako jsou například Siemens, Motorola, Honeywell, Phillips, Samsung a další. Hlavními přednostmi ZigBee jsou jednoduchost, flexibilita, spolehlivost, nízká spotřeba energie. Díky těmto přednostem se lze setkat s touto technologií u automatizace budov, ve spotřební elektronice, ve zdravotnictví a v průmyslové automatizaci. ZigBee má tři základní režimy přenosu dat. A to pro periodicky se opakující přenosy dat například z čidel, pro asynchronní přenosy, kterým může být třeba stisknutí tlačítka a pro opakující se přenosy s rychlou odezvou pro bezdrátové počítačové periferie.

Při vytváření ZigBee bylo cílem vytvořit standard, který lze implementovat i do málo výkonných zařízení jako jsou 8bitové mikrokontrolery. Díky tomu se struktura protokolů vejde i do paměti o velikosti 30kb. Protokol má tři základní vrstvy. Nejnižší vrstva je vrstva IEEE 802.15.4, nad ní se nachází vrstva NWK (Network Layer) a nejvyšší je vrstva APL (Application Layer). Nejnižší vrstva má podvrstvy MAC (Medium Access Control) a PHY (Physical Layer). Součástí APL jsou podvrstvy APS (Application Support Sublayer), Application Framework a ZigBee Device Object. NWK obsahuje Security Management, Message Broker, Routing Management a Network Management. Vrstva standardu IEEE 802.15.4 definuje základní rádiová pásma pro využití v různých částech světa, protože národní předpisy a normy jednotlivých zemí nejsou sjednoceny a mají různé definice rádiových pásem. Na základě této problematiky byly definovány tři základní pásma a to pro Evropu, Ameriku plus Australii a globální.



Obrázek 1: Zigbee stack layer

Použitelný dosah v praxi bývá 10 až 50 metrů a samozřejmě záleží na podmínkách kudy signál prochází. Signál se moduluje metodou O-QPSK a přenos probíhá pomocí DSSS (Direct Sequence Spread Spectrum). K přístupu na médium je používána metoda CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance and optional time slotting).

MAC vrstva definuje komunikační protokol, který funguje na principu přenosu datových rámců. Rámce jsou čtyři a využívají se k řízení, posílání dat a správě sítě. Data Frame slouží k přenosu dat. Acknowledgement Frame se používá k posílání potvrzovacích informací. Beacon Frame je využíván koordinátorem pro posílání takzvaných beacons, které slouží k uvádění připojených zařízení do spánkového módu. Posledním datovým rámcem je Frame Command Frame, který slouží k řízení a nastavování připojených zařízení.

Díky tomuto přístupu lze dosáhnout velice úsporných řešení. Při průběžnému uspávání připojených zařízení na základě předem definovaných dob probouzení lze výdrž koncového zařízení napájeného alkalickou baterií prodloužit až na dva roky. Proces přeposílání dat funguje v síti tak, že koordinátor přijme zprávu z jednoho zařízení, která je

určená pro jiné zatím uspané a vyčká na předem domluvenou dobu, kdy se zařízení probudí a tehdy mu zprávu pošle.

Protože je ZigBee postaveno na IEEE 802.15.4 podporuje tři typologie sítě. Typologie může být typu hvězda, což znamená, že koordinátor je uprostřed a každé koncové zařízení je připojeno ke koordinátoru a pouze k němu. Druhým typem zapojení je do stromu. To znamená, že plně funkční zařízení může být zapojeno mezi koordinátor a koncovým zařízením a tím zvětšit dosah sítě. Mezi koordinátorem a každým zařízením vede sítí pouze jedna cesta. Poslední typologií je zapojení do sítě (mesh). Zapojení do sítě je takové zapojení, při kterém aspoň jedno zařízení může přistupovat ke koordinátoru dvěma cestami.

ZigBee rozděluje zařízení na dvě kategorie a to FFD (Full Functional Device), která mají plnou funkčnost a na RFD (Reduced Functionality Device) s omezenou funkčností. Zařízení označené jako FFD je zařízení, které implementují kompletní rámec protokolu a dokáží zajistit veškeré služby stanovené standardem ZigBee. Zařízení označená jako RFD mohou být pouze koncová zařízení a mohou komunikovat pouze s koordinátorem nebo se zařízeními FFD. RFD zařízení mohou být tedy jen koncovými větvemi na rozdíl od FFD a koordinátoru.

Každé zařízení v síti je adresováno kódem o délce 64 bitů a nebo v případě zkrácené podoby 16 bitů. Každá síť má svojí identifikaci také pomocí 16 bitového PAN ID pro rozlišení od jiných sítí založených na standardu IEEE 802.15.4. V každé síti přiděluje PAN ID koordinátor sítě a ostatní zařízení jsou buď směrovače nebo jen koncová zařízení. Síťová vrstva (NWK) má na starost zabezpečení, směrování a synchronizaci. V případě koordinátoru i start sítě a přidělování adres nově připojeným zařízením.

ZigBee je v základu zabezpečeno AES (Advanced Encryption Standard) 128bit dlouhým klíčem, který implementován přímo v síťové vrstvě. Lze zabezpečit i rámce v MAC vrstvě, tak je lze také zabezpečit pomocí AES, která je taktéž implementována přímo v ní.

Aplikační vrstva (APL) je tvořena z pomocné aplikační podvrstvy (APS), ZigBee objektů (ZDO) a aplikačních objektů definovaných uživatelem. O párování se stará pomocná aplikační vrstva na základě požadavků a poskytnutých služeb. K tomu je využívána vázací tabulka (binding). ZDO definuje role jednotlivých zařízení v síti. Role jsou jak bylo už zmíněno dříve koordinátor, směrovač a koncové zařízení. Další funkcí, kterou má ZDO na starost je vyhledávání nových zařízení a jaké služby poskytují. Dále je zodpovědný za zabezpečení, protože volí jeho způsob. Aplikační objekty pak aplikují požadavky aplikace do definovaného ZigBee profilu. Na základě specifikace ZigBee aliance je každý profil unikátně identifikovatelný pomocí 16bitového klíče.[4]

3.1.4 MQTT

Message Queuing Telemetry Transport je standardem ISO/IEC 20922. Lze pomocí protokolu odesílat balíčky dat až o velikosti 256 Mb. Je jednoduchým protokolem pro posílání zpráv mezi senzory a aplikací a middlewarem. Protokol pracuje v nejvyšší vrstvě TCP/IP pro zajištění jednoduchých, ale spolehlivých proudů dat. MQTT by mělo fungovat s jakoukoliv sítí, která poskytuje seřazené, bez ztrátové a obousměrné připojení. V MQTT protokolu vystupují tři hlavní elementy a těmi jsou subscriber, publisher a broker.

Je známo, že MQTT protokol je dobrou možností pro bezdrátové sítě, ve kterých se častěji vyskytují nespolehlivá připojení. Společnost Facebook používala protokol MQTT ve svém produktu pro online chatování Facebook Messenger.

Autory první verze MQTT z roku 1999 byli Andy Stanford-Clark z IBM a Arlen Nipper, který tehdy pracoval pro Eurotech Inc. Tato verze byla použita k monitorování ropovodů společně s frameworkem SCADA. Cílem bylo vytvořit protokol, který by využíval málo energie, byl jednoduchý a efektivní co se týče přenosu po sítích s nízkou propustností.

V roce 2013 zaslala IBM do OASIS specifikaci MQTT verze 3.1 s listinou, že pouze drobné úpravy jsou možné. MQTT-SN (MQTT for Sensor Networks) je variací hlavního protokolu, který je zaměřený hlavně na nízkou spotřebu energie zabudovaných zařízeních v sítích bez TCP/IP jako například ZigBee.

MQTT protokol definuje dva typy síťových entit a to message broker a client. MQTT broker je server, který přijímá zprávy vyslané od všech klientů a poté je rozesílá klientům, pro které zprávy byly určeny. MQTT klientem může být jakékoliv zařízení od mikrokontroleru po server, které je připojené k brokeru po síti a disponuje knihovnama MQTT.

Informace jsou řazeny do témat hierarchicky. V případě, že zařízení chce publikovat data, tak je odešle brokeru společně s tématem. Každému zařízení, které je u brokeru přihlášeno k odběru tohoto téma je zaslána zpráva s daty. Publikující zařízení nemusí mít žádné informace o počtu a lokacích příjemců, příjemci nemusí mít informace o odesílatelích.

V případě, že broker obdrží zprávu, ale nemá ji kam poslat, tak ji prostě zahodí. Broker zprávu nezahodí v případě, že odesílatel stanovil zprávu jako retained message. Retained message je normální MQTT zpráva jako každá jiná, ale liší se tím, že má nastavený retained flag na true. V tomto případě broker zprávu uloží a počká dokud se nějaké zařízení nepřihlásí do tématu a v momentě kdy se tak stane je mu ihned odeslána uložená zpráva. Broker ukládá vždy pouze jednu retained message na jeden topic. Když se zařízení poprvé připojí do sítě může nastavit zprávu jaká se má poslat všem odběratelům v případě, že dojde k odpojení zařízení od brokeru. Klienti mohou komunikovat pouze s brokerem avšak pokud systém obsahuje více brokeru, tak ti si mohou mezi sebou vyměňovat informace ve svých topicích.

MQTT spoléhá na TCP protokol pro přenos dat. Pro variantu MQTT-SN jsou používány jiné protokoly jako UDP a Bluetooth. MQTT zasílá přihlašovací údaje v plain textu a neobsahuje žádné bezpečnostní prvky. Lze však komunikaci zabezpečit pomocí TLS enkrypcí. Výchozím portem bez enkrypcí je port 1883 a ekryptovaným portem je port 8883.

U každého připojení k brokeru lze nastavit QoS (Quality of service). Tímto se nastavuje jak bude ověřováno a zda-li vůbec, jestli zpráva dorazila k příjemci. QoS může

býti nastaveno na At most once což znamená, že se zpráva pošle a už dále neřešíme co se s ní děje. Dalším nastavením může být At least once a to zapříčiní, že se zpráva posílá tak dlouho dokud nepřijde potvrzení o přijetí. Poslední možností je Exactly once a to pracuje tak, že se provede dvoustupňový handshake mezi odesílatelem a příjemcem k zaručení, že se odešle pouze jedna kopie zprávy a ta bude doručena. [5][6]

3.1.5 Z-Wave

Jedná se o bezdrátový komunikační protokol využívaný hlavně pro domácí automatizaci. Z-Wave funguje na bázi mesh sítě, ve které jednotlivá zařízení spolu komunikují za pomoci nízkoenergetických rádiových vln. Díky tomu lze ovládat bezdrátově různé domácí spotřebiče, osvětlení, garážová vrata a jiné. Stejně jako i u jiných protokolů pro domácí a kancelářskou automatizaci lze systém na bázi Z-Wave ovládat přes internet z mobilního telefonu, tabletu či počítače. Lokálně lze síť ovládat bezdrátovým ovladačem, chytrým reproduktorem nebo na zdi zavěšeným panelem s bránou Z-Wave a nebo centrálním řídicím zařízením, které slouží jako ovladač hubu, ale i jako portál ven. Z-Wave poskytuje interoperabilitu aplikační vrstvy mezi systémy od různých výrobců, kteří jsou součástí aliance.

Protokol Z-Wave byl vyvinut v roce 1999 dánskou společností Zensys v Kodani. V tomto roce společnost představila spotřebitelský systém pro ovládání světla, který se později vyvinul do Z-Wave jako proprietární SoC pro protokol domácí automatizace na bezlicenčním frekvenčním pásmu v rozsahu 900 Mhz. Chipset řady 100 byl vydán v roce 2003, řada 200 v květnu 2005 společně s čipem ZW0201, který nabízí vysoký výkon za nízkou cenu. Řada 500 známá jako Z-Wave plus byla vypuštěna roku 2013 v březnu. Paměť u řady 500 byla čtyřnásobná než doposud společně s vylepšeným dosahem bezdrátové sítě a lepší životností baterií. Technologie Z-Wave se uchytila v Severní Americe v roce 2005 kdy ji přijalo pět společností včetně Danfoss, Ingersoll-Rand a Levito Manufacturing. Tyto společnosti utvořili Z-Wave alianci, jejímž cílem bylo podporování používání technologie Z-Wave. Všechny produkty společností v alianci jsou interoperabilní.

Interoperabilita Z-Wave na aplikační vrstvě zajišťuje, že zařízení mohou sdílet informace, a umožňuje spolupráci veškerého Z-Wave hardwaru a softwaru. Díky tomu, že tento protokol využívá bezdrátové mesh síťové technologie může každý uzel komunikovat přímo či nepřímo s jiným uzlem. Uzly, které jsou v dosahu, komunikují přímo a ty které jsou mimo dosah používají pro svou komunikaci jiný uzel, který je v dosahu a jejich zprávy přeposílá. V září roku 2016 byly některé části technologie Z-Wave veřejně zpřístupněny tehdejším majitelem Sigma Designs, když vydal veřejnou verzi vrstvy interoperability Z-Wave společně se softwarem Z-Wave, který byl přidán do open-source knihovny.

Z-Wave je navržen tak, aby poskytoval spolehlivý přenos malých datových paketů s nízkou latencí při datových rychlostech až 100kbit/s. Propustnost je 40kbit/s (9,6 kbit/s u starých čipů) a je vhodná pro řídicí a senzorové aplikace, na rozdíl od Wi-Fi a dalších bezdrátových LAN systémů založených na IEEE 802.11, které jsou určeny primárně k vysokým přenosovým rychlostem. Maximální komunikační vzdálenost mezi dvěma uzly je 30 metrů a 40 metrů s čipy řady 500. Díky schopnosti zprávy přeposlat až čtyřikrát mezi uzly je pokrytí dostatečné pro většinu obytných domů. Modulace je FSK (Frequency-shift keying) s Manchesterským kódováním.

Z-Wave využívá nelicensované ISM (Industrial Scientific Medical) pásmo a to konkrétně část 15. Pracuje na 868,42 Mhz v Evropě, na 908,42 Mhz v Severní Americe a podle předpisů v jiných zemích využívá i jiné frekvence. Toto pásmo je společné pro některé bezdrátové telefony a jinou spotřební elektroniku, ale není tak vytíženo jako pásmo 2,4 Ghz, ve kterém operují protokoly Wi-Fi, Bluetooth a jiné. Spodní vrstvy, MAC a PHY, jsou popsány v ITU-T G.9959 a jsou plně zpětně kompatibilní. V roce 2012 zahrnuje ITU (International Telecommunication Union) ve svém standardu G.9959 pro bezdrátová zařízení do 1 Ghz vrstvy Z-Wave PHY a MAC. Datové rychlosti zahrnují 9600 b/s a 40 kb/s s výstupním výkonem 1 mW nebo 0 dBm. Z-Wave transceiver čipy dodává společnost Silicon Labs.

Z-Wave síť může obsahovat až 232 zařízení. Pokud je potřeba připojit více zařízení lze toho docílit pomocí přemostění k další síti Z-Wave. Každá síť Z-Wave je identifikována

pomocí Network ID a každé zařízení v síti pomocí Node ID. Network ID také nazýváno Home ID je běžnou identifikací každého uzlu v síti. Network ID má délku 4 bytů (32 bitů) a je přiřazeno každému zařízení hlavním kontrolerem. Zařízení s rozdílným Network ID spolu nemohou komunikovat. Node ID je adresa konkrétního uzlu v síti a má délku 1 byt (8 bitů) a musí být v síti unikátní.

Dne 17. listopadu roku 2016 oznámila Z-Wave aliance přísnější bezpečnostní standardy pro zařízení, která obdržela certifikaci Z-Wave od 2. dubna roku 2017. Známa jako Security 2 (nebo S2) a poskytuje pokročilé zabezpečení pro inteligentní domácí zařízení, brány a rozbočovače. Podporuje šifrovací standardy pro přenosy mezi uzly a nařizuje nové postupy párování pro každé zařízení s jedinečným PIN nebo QR kódem na každém zařízení. Nová vrstva ověřování má zabránit hackerům převzít kontrolu nad nezajištěnými nebo špatně zabezpečenými zařízeními. Podle Z-Wave aliance je nový bezpečnostní standard nejpokročilejším zabezpečením na trhu pro chytrá domácí zařízení a řadiče, brány a rozbočovače. Bohužel kvůli nutnosti zpětné kompatibility jsou zařízení S2 během párování stále zranitelná.

Čip pro uzly Z-Wave je ZW0500, postavený na mikrokontroleru Intel MCS-51 s interním systémovým taktem 32 Mhz. RF část čipu obsahuje transceiver GisFSK pro softwarově volitelnou frekvenci. Při napájení 2,2-3,6 voltů spotřebuje v režimu přenosu 23 mA. Mezi jeho funkce patří šifrování AES-128, bezdrátový kanál 100 kb/s, souběžný poslech na více kanálech a podpora USB VCP.[7]

3.1.6 SMB

SMB (Server Message Block) je také znám jako CIFS (Common Internet File System) je síťový protokol 7. vrstvy zvané application layer v síťovém modelu OSI, který zajišťuje přístup ke sdíleným souborům, tiskárnám a jiným vybavení sítě. Poskytuje také zabezpečenou komunikaci. SMB je nejvíce rozšířeno na počítačích s platformou Windows.

Protokol byl vyvinut ve firmě IBM s cílem zpřístupnit lokální soubory po síti. Protokol byl vyvinut Barrym Feigenbaumem a poté co Microsoft provedl určité změny, tak ho obsáhl do produktu LAN Manager. Důležitým aktem byla také svobodná implementace

protokolu za pomoci reverzního inženýrství v rámci projektu Samba. Druhá verze protokolu tedy SMB2 byla představena Microsoftem společně s Windows Vista.

Protokol pracuje na modelu klient-server. Server poskytuje přístup ke sdíleným souborům na základě požadavků klienta. Klient má za úkol formulovat požadavek, který odešle serveru ve formě paketu SMB. Server požadavek rozebere na jednotlivé příkazy a zkontroluje zda-li má klient dostatečná práva k realizaci těchto příkazů. V případě, že jsou práva dostatečná, tak server spustí požadované příkazy a výsledek vrací klientovi ve formě identického paketu.

Přístup ke sdíleným prostředkům je řízen dvěma způsoby. Sdílení může být řízeno na základě share level, tedy jaká úroveň sdíleného prostředku je přístupná a na základě user level, tedy jaké úrovni je uživatel. U prvního způsobu je vyžadováno heslo k přístupu ke sdílenému prostředku. V případě správného hesla je klientu přidělen NID (Network ID), který mu umožňuje přístup k prostředku. Druhý způsob probíhá tak, že se klient přihlásí uživatelským jménem a heslem a na základě toho mu je přiděleno UID (User ID) a na základě UID server ví zda má uživatel právo na jednotlivé požadavky pro manipulaci se sdílenými prostředky.

Protokol SMB2 byl představen v roce 2006 spolu s operačním systémem Windows Vista a jeho specifikace byly zveřejněny a díky tomu i ostatní systémy mohli komunikovat s produkty Microsoft. SMB2 je oproti SMB1 mnohem menší, protože byl zredukován počet příkazů ze stovek na pouhých devatenáct. SMB2 podporuje pipelining (řetězení) tedy, že lze odeslat další příkazy, aniž bychom obdrželi odpovědi na předešlé. To má za následek zvýšení výkonu a také v rámci jednoho požadavku odeslat více akcí. Dále má SMB2 podporu pro symbolické adresy. Další významnou změnou je přechod z pouhých 16 bitů offsetu na 32, 64 či 128 bitů, což má za následek, že v rámci jednoho bloku lze přenést větší množství dat a díky tomu se zrychluje přenos dat. Využití tohoto vylepšení se významně podepisuje na době potřebné k přenosu větších souborů po síti.[8]

3.1.7 FTP

FTP je zkratka z angličtiny z File Transfer Protocol. Jak název napovídá, jedná se o protokol pro přenos souborů mezi počítači po počítačové síti. Protokol FTP využívá protokol TCP z rodiny TCP/IP. Protokol je platformě nezávislý, tedy může být používán na různých platformách. K jeho definování došlo v RFC 959 a to již roku 1985 a jeho rozšíření se uskutečnilo v roce 1997 v RFC 2228. FTP protokol je podporován webovými prohlížeči a nebo speciálními programy zvaných FTP klient. Příkladem FTP klienta lze uvést třeba software zvaný Filezilla.

První specifikaci sepsal Abhay Bhushan, která byla publikována 16. dubna 1971 v RFC 114, kterou později nahradila novější RFC 765 z června roku 1980. I ta byla později nahrazena novější a to již dříve zmiňovanou RFC 959 v říjnu roku 1985, která je aktuální specifikací protokolu FTP. Specifikace byla několikrát upravována a to z důvodu bezpečnosti v RFC 2228 a z důvodu přidání podpory Ipv6 v RFC 2428.

FTP patří k nejstarším protokolům. FTP používá protokoly TCP/20 a TCP/21. Port 20 je využíván k přenosu dat v 8mi bitové podobě. Po portu 21 jsou pak posílány příkazy. Přenos dat může být buď binární nebo textový v ASCII. Pro textový přenos musí dojít ke konverzi konců řádků. U windows jsou konverzovány CR a LF konce řádků. CR značí přesunutí kurzoru na začátek řádku a LF přesun na další řádek. U linuxu je prováděna pouze konverze LF znaku. Konverze je prováděna jen za předpokladu, že koncové systémy jsou rozdílné. V případě binárního přenosu dat se do dat nijak nezasahuje.

Protokol umožňuje řízení přístupu přihlašování pomocí loginu a hesla. Také umožňuje výpis vzdáleného adresáře a popis formátu přenášeného souboru. V RFC 2228 byla pro protokol definována rozšíření, protože v dnešní době ho nelze považovat za bezpečný.

Protokol pracuje na konceptu modelu klient-server. FTP server umožňuje přístup k datům ostatním počítačům v síti. Po připojení klienta může uživatel provádět různé operace se soubory. Práva k operacím, které může uživatel se soubory provádět jsou mu přidělena na základě loginu, kterým se na FTP server přihlásil.

Nevýhodou FTP je, že hesla a soubory jsou přenášeny v textové podobě bez šifrování, což velice snižuje bezpečnost. V sázce jsou díky této nevýhodě přihlašovací údaje i přenášená data. Existují však rozšíření, která tento nedostatek odstraní. Další nevýhodou FTP je, že server má delší odezvu. Toto se děje díky tomu, že nelze sloučit více menších souborů do jednoho a to má za následek vyšší časovou režii a zátěž serveru. V případě použití firewallu je vyžadován aktivní FTP přenos. U šifrovaného řídicího spojení však aktivní přenos nefunguje. Výhodou na druhou stranu je, že serverová část je jednodušší oproti běžným HTTP serverům. Další výhodou oproti HTTP je širší paleta možností jako jsou například nastavení práv, mazání, upload a další. FTP protokol v dnešní době je používán méně. V některých sítích není ani povolen jiný protokol než HTTP.[8]

3.2 Software

Software je programové vybavení počítače – tedy programy a aplikace v počítači. Software rozdělujeme na aplikační a systémový. Systémový software zajišťuje chod počítače a aplikační software jsou programy, s nimiž pracuje uživatel. Software je opakem hardware.

3.2.1 Arduino IDE

Arduino IDE je cross-platformní aplikace pro Windows, macOS a Linux napsaná v jazycích C, C++ a Java. Používá se pro psaní a nahrávání programů do desek od firmy Arduino, ale i do desek od jiných výrobců. Zdrojový kód je vydán pod licencí GNU General Public License. IDE podporuje jazyky C a C++ a přináší softwarové knihovny od projektu Wiring, které poskytují spousty častých vstupních a výstupních procedur. Součástí Arduino IDE je také program avrdude, který slouží ke konvertování spustitelného kódu do textového souboru v hexadecimálním kódování, který je posléze nahrán do desky. [9]

3.2.2 DietPi

DietPi je extrémně odlehčený operační systém Debian a je co nejvíce optimalizován pro minimální vytěžování procesoru a operační paměti. To má za následek nižší provozní teplotu a více výpočetního výkonu pro uživatelské aplikace.[10]

3.2.3 Apache

Apache je multiplatformní softwarový webový server. Je dostupný pod operačními systémy GNU/Linux, BSD, Solaris, Windows a mac OS. Apache byl do nedávna nejvíce rozšířeným webovým serverem nyní však zaujímá druhé místo s 25,87% za NGINX, který vládne s 36,45%.

Vývoj Apache započal v roce 1993 na Illionské univerzitě. Původní název projektu byl NCSA HTTPd. První zveřejnění proběhlo v roce 1995 ve verzi 0.6.2. Webový server je vyvíjen v jazyce C a vývoj má na starosti Apache Software Foundation. Znak indiánského

pera a název Apache jsou odkazem na původní obyvatelstvo Ameriky a stejnojmenný indiánský kmen.

Portfolio podporovaných funkcí je velké. Mnoho z těchto funkcí jsou implementované jako kompilované moduly, které rozšiřují jádro. Server podporuje spoustu různých programovacích jazyků jako jsou například Python, Perl, PHP. Dále podporuje autentizační schémata jako jsou například mod_auth, mod_access, mod_digest a jiné. Dalšími funkcemi serveru jsou SSL, TLS, proxy modul, konfigurace logů, URL přepisovač a další.

Co se výkonu týče Apache není nejrychlejším webovým serverem. Hlavním důvodem je, že server není založen na eventech, ale na MultiProcessing modulech zkráceně MPM. Není zde tedy implementace jedné architektury. Na druhou stranu díky MPM lze webový server přizpůsobit systému, na kterém je spuštěn. Hlavním cílem je zajistit spolehlivé obslužení požadavků a konzistenci. Podle vývojářů je nejvýhodnější verzí kombinace více procesů a vláken. [11]

3.2.4 NGINX

Jedná se softwarový webový server s otevřeným zdrojovým kódem v jazyce C. NGINX operuje s protokoly HTTP, HTTPS, SSL, IMAP, POP3 a SMTP. Disponuje také load managementem a reverzní proxy. Webový server je dostupný pro Linux, Unix a jiných podobných systémech pod BSD. Jsou nabízeny i varianty pro Windows, macOS a Solaris. Hlavním cílem tohoto webového serveru jsou především vysoký výkon a nízká paměťová náročnost.

NGINX byl vyvinut Igorem Sysoevem jako reakce na nespokojenost s funkčností a škálovatelností Apache. Vývoj byl zahájen v roce 2002 a o dva roky na podzim roku 2004 byla vydána první oficiální verze. Hlavními cíli nového řešení bylo zajistit obslužnost 10 000 požadavků najednou bez problémů a s nízkými nároky na operační paměť.

NGINX cílí na rychlou distribuci statického obsahu a také na rozdělení zátěže na další servery v závislosti na jejich prioritě. Lze také nastavit záložní server, který zkusí

odpovědět na požadavek v případě, že hlavní server neodpověděl do časového limitu. Na rozdíl od Apache, který požadavky vyřizuje za pomoci vláken a procesů, tak NGINX zpracovává požadavky asynchronně. NGINX se při příchozím požadavku nejdříve podívá, jestli odpověď na stejný požadavek už nedávala a není uložena v cashi, v tom případě rovnou odpoví. V případě, že se odpověď v cashi nenachází, tak je požadavek přeposlán na servery s nižší prioritou a případně i na záložní server. To jak dlouho se má požadavek v cash nacházet lze nastavit. Další funkcí, kterou systém disponuje je omezení počtu připojení z jedné adresy IP, to poskytuje alespoň nějakou ochranu proti DOS útokům. NGINX se skládá z modulů, které lze přidávat a odebírat, avšak často je rekompilace nutností. Příklady modulů jsou třeba GEO lokace pro vyřizování požadavků na základě země ze které pochází, modul pro zabezpečení stránky heslem, modul pro přesměrování, modul pro podporu komprese gzip, modul pro streamování FLV a MP4 a jiné.[12]

3.2.5 Samba

Jedná se o síťový souborový systém distribuovaný pod licenci GNU GPLv3. Samba je multiplatformní a lze ji zprovoznit jak na Windows tak i na UNIXových operačních systémech jako například GNU/Linux, Solaris, BSD, macOS. Samba může sloužit i jinak než pouze ke sdílení souborů a tiskových služeb. Lze ji využít i pro integraci do domény Windows jako primární doménový řadič nebo jako člena v doméně. Lze ji zahrnout i do Active Directory. Od verze 4 může být Samba řadičem domény Active Directory. Samba pracuje se síťovým protokolem SMB (Server Message Block) a v novějších verzích s CIFS (Common Internet File System)

Samba byla vyvinuta Andrewem Tridgellem a to nejdříve pro operační systém UNIX. Tým Samba se dostal ke kompletní dokumentaci SMB protokolu roku 2007, kterou jim poskytl Microsoft. To je důležitým milníkem, protože právě přístup k dokumentaci umožnil open source programátorům implementovat řešení sdílení a jiných služeb v síti Microsoft a zajistit tím kompatibilitu. Důsledkem bylo vydání další verze Samba a to Samba 4, která již plně podporovala dříve zmíněný Active Directory.

První verze Samba byla vyvinuta již roku 1992 na Australské národní univerzitě, kde Andrew Tridgell studoval doktorské studium. Pomocí reverzního inženýrství aplikovaného na DEC Pathworks vytvořil projekt, který tehdy nazýval souborovým UNIX serverem pro DOS Pathworks a to až do verze 1.0, kde došel k uvědomění si, že úspěšně implementoval NetBIOS protokol. V roce 1993 vydal software, který obsahoval jak klientskou, tak i serverovou část. Tento software byl pojmenován nserver tedy netbios pro unix. Pro projekt byla vybrána license GPLv2 a projekt se zaměřoval na interoperabilitu s LAN Managerem od Microsoft. Během verze 1.5 došlo ke změně názvu na smbserver na základě regulárního výrazu '^s.*m.*b' aplikovaného na /usr/share/dict/words pomocí

programu grep. Tridgell uvádí, že samba tým vznikl pravděpodobně díky vzniku CVS v roce 1996. CVS (Concurrent Version System) je systém, který slouží ke správě verzí projektů. Verze Samba 2 byla vydána v roce 1999 a verze Samba 2.2 v roce 2001.

Dalším důležitým milníkem je rok 2003, kdy byla vydána důležitá aktualizace Samby na verzi 3.0, která byla již schopna se připojit do Active Directory a to zatím pouze jako člen nikoliv řadič domény. S verzí 3.2 se přešlo z licence GPLv2 na GPLv3 plus některé části jsou šířeny pod licencí LGPL3. Verze 4.0 byla vydána 11. prosince 2012. Tato verze již dokáže plnit úlohu řadiče domény Active Directory a být jejím plnohodnotným členem.

Samba umožňuje sdílení souborů vybraných unixových adresářů a včetně všech jejich podadresářů. Celá větev se uživatelům Windows ukáže jako složky Windows sdílené po síti. Uživatelé přistupující z operačního systému Unix pak mohou k souborům přistupovat tak, že buď si adresář připojí do systému pomocí smbmount a nebo mohou k datům přistupovat pomocí smb klienta smbclient, který pracuje jako klasický FTP klient. Každý adresář má nastavená práva přístupu, která jsou nadřazená běžným právům v systému Unix.

Samba je spouštěna jako dva démoni. Démon smbd má na starost poskytování přístupu ke sdíleným souborům a tiskárnám. Démon nmbd pak zajišťuje překlad NetBIOS na IP adresu. Nastavení Samby lze provést editací souboru smb.conf, který se nachází typicky v adresáři /etc/ a nebo jeho podadresáři /etc/samba/. Přihlašovací skripty a nastavení práv je pak konfigurováno pomocí souboru poledit. Samba také disponuje webovým rozhraním pro její správu a to se nazývá SWAT (Samba Web Administration Tool). Toto rozhraní je však nutno aktivovat, protože z bezpečnostních důvodů je ve výchozím nastavení vypnuto. [13]

3.3 Hardware

Hardware je veškeré technické vybavení počítače. Volba správného hardwaru je základním kamenem pro tvorbu chytré domácnosti. Je důležité zvolit hardware, který je energeticky nenáročný, nezabírá příliš prostoru, je cenově dostupný a spolehlivý. Je důležité zvážit co každá konkrétní jednotka bude vykonávat a co vše k tomu potřebuje za hardwarové vybavení.

3.3.1 Raspberry Pi

Malý jednodeskový počítač Raspberry Pi byl vytvořen britskou nadací Raspberry Pi Foundation za účelem výuky informační techniky na školách ve Velké Británii. Cílem bylo žákům ukázat, že lze ovládat různá zařízení v domácnosti pomocí počítače.

Raspberry Pi bylo vyvinuto v roce 2012 a primárním operačním systémem je Raspbian, který je založen na operačním systému Debian. Raspbian je optimalizován pro ARM procesory použité v Raspberry Pi. Desktopovým prostředím Raspbianu je PIXEL což je zkratka anglických slov Pi, Improved, Xwindows, Enviroment, Lightweight. PIXEL tvoří LXDE a Openbox.

K Raspberry lze připojit běžné periferie jako tomu je u normálních počítačů. Na tomto jednodeskovém počítači lze nalézt slot na SD kartu, HDMI výstup, DSI display port, CSI camera port, 3,5 mm audio jack, USB porty, LAN. Raspberry také disponuje Wifi a bluetooth moduly.

Operační systém Raspberry Pi je nahrán na SD kartě a při spuštění se nahraje do operační paměti RAM.

Častými projekty k domácímu užití ke kterým je Raspberry využíváno jsou náhrady NAS serverů, chytré televize například za pomoci softwarového multimediálního centra KODI , retro gaming a nebo vytvoření přístupového bodu Wifi.[14]

3.3.2 Arduino

Arduino stejně jako Raspberry je jednodeskový počítač, který taktéž měl podpořit výuku na školách v rámci informačních technologií. Na rozdíl od Raspberry nedisponuje periferiemi pro připojení monitoru, klávesnice či myši. Jedná se tedy o mikrokontroler a ne o mikroprocesor jakým je Raspberry.

Arduino se liší i vývojem řídicího programu. Na rozdíl od Raspberry je program vyvíjen na jiném počítači a až po sléze nahrán do Arduina. Na Arduinu je tedy spuštěn jen ten program, který do něj byl nahrán. Tento program většinou obsahuje nekonečnou

smyčku, která se pořád opakuje a získává data, která poté ukládá a případně posílá dále. Výsledkem tohoto konceptu je nízká energetická náročnost a proto lze Arduino napájet i menšími bateriemi.[15]

3.3.3 ESP 8266

Jedná se o mikrokontroler vytvořený firmou Espressif Systems. ESP 8266 disponuje plnou podporou TCP/IP. Mikročip vyžaduje vstupní napětí 3.3 V je osazen 32 bitovým RISCovým procesorem L106 běžícím na 80 nebo 160 Mhz. Paměť je rozdělená na 32 KiB pro instrukce a 80 KiB pro uživatelská data. Jako vstup slouží 16 GPIO pinů. Mikrokontroler zvládá Wifi IEEE802.11 b/g/n s autentizací WEP, WPA/WPA2 a nebo jako otevřená síť.

Mikrokontroler ESP 8266 je vhodný pro rozšíření jiných mikrokontrolerů a mikroprocesorů, kteří sami o sobě nedisponují možností připojení k bezdrátové síti Wifi. V momentě kdy zařízení disponuje možností se připojit k bezdrátové síti se stává mobilním a to je v určitých řešeních vyžadováno. [16]

3.3.4 CC2531

CC2531 je system on chip připojovaný přes USB. Je určený pro IEEE 802.15.4, Zigbee a RF4CE. Umožňuje vytvářet USB sniffery nebo síťové uzly s nízkými celkovými náklady. CC2531 využívá výkon RF transceiveru s vylepšeným 8051 MCU. Má programovatelnou flash paměť a operační paměť o velikosti 8 KB. CC2531 má různé provozní režimy a to jej dělá vhodným pro systémy, kde je vyžadována velmi nízká energetická náročnost .

Tento kontroler je využíván pro přemostění mezi Zigbee a MQTT protokolem. [17]

3.3.5 BME280

Senzor BME280 disponuje digitálním teploměrem, tlakoměrem a vlhkoměrem. Senzor lze připojit k různým mikroprocesorům pomocí I2C sběrnice. Napájecí napětí je od 1,8 voltů do 5 voltů. I2C adresy jsou 0x76 pro SDO LOW a 0,77 pro SDO HIGH. Co se rozsahu měření týče, tak teplota má rozsah měření mínus 40 až plus 85 stupňů Celsia.

Vlhkost má rozsah 0 až 100 procent vlhkosti. Tlak má rozsah 300 až 1100 hektopaskalů. Přesnost měření jednotlivých veličin je pak následující. U teploty se může hodnota lišit o jeden stupeň Celsia. U vlhkosti se může naměřená hodnota lišit od skutečné o 3 jednotky procent. Tlak může být naměřen s nepřesností jednoho paskalu. Rozlišení měření jednotlivých veličin je pro teplotu 0,01 stupně Celsia, 0,008 procent vlhkosti a 0,18 paskalů.[18]

3.3.6 DHT22

DHT22 je nízkonákladový digitální senzor pro měření tlaku a teploty. Využívá kapacitního senzoru vlhkosti a termistor pro měření teploty okolního vzduchu. Naměřené hodnoty posílá po digitálním pinu. Další dva piny jsou pro vstupní napětí a uzemění. Nevýhodou tohoto zařízení je frekvence čtení jednou za dvě sekundy. Rozsah měření vlhkosti je od 0 do 100 procent s maximální odchylkou 2 procenta. Rozsah měření teploty je od -40 do +80 stupňů Celsia s maximální odchylkou půl stupně Celsia. [19]

4 Analýza požadavků

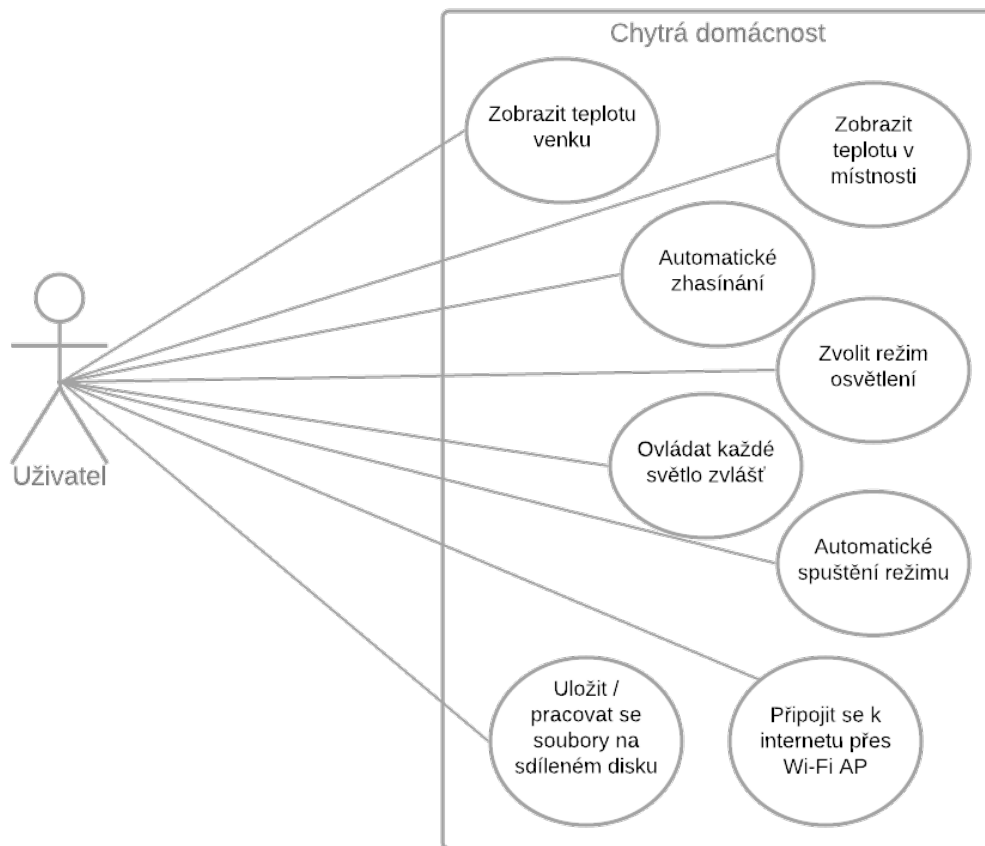
Cílem práce bylo navrhnout a vytvořit systém pro chytrou domácnost, který bude využíván více uživateli. Každý z budoucích uživatelů sdělil své nároky, co od systému požaduje za funkcionality a co od něj očekává. Na základě výsledků dotazování budoucích uživatelů byl sestaven seznam případů užití.

4.1 Use case

- Systém bude schopen zobrazit teplotu venkovního vzduchu, jeho vlhkost a tlak.
- Systém bude schopen zobrazit teplotu a vlhkost uvnitř místnosti.
- Systém bude schopen sdílet soubory na disku po síti.
- Systém bude umožňovat nastavení jasu, barvy a teploty světla pro každou chytrou žárovku v domácnosti zvlášť.
- Systém bude umožňovat nastavení automatického zhasínání pro každou chytrou žárovku zvlášť
- Systém bude umožňovat zvolit z rodílných režimů osvětlení, které nastaví jas a teplotu světla.
- Systém bude umožňovat nastavení automatického spuštění režimu osvětlení.
- Systém bude umožňovat připojení k lokální síti pomocí Wi-Fi.
- Uživatel bude se systémem komunikovat přes webové rozhraní.
- Webové rozhraní určené pro komunikaci mezi uživatelem a systémem bude mít responzivní design.

4.2 Use case diagram

Na základě případů užití byl sestaven use case diagram.



Obrázek 2: Use case diagram

4.3 Use case scénáře

Use case scénář popisuje posloupnost kroků popisujících interakci mezi uživatelem a systémem. Základními atributy use case scénáře jsou název, který dostatečně reprezentuje cíl use case a hlavní scénář, kterým je číslovaný seznam kroků a krokem je myšlena jedna konkrétní interakce mezi účastníkem a systémem. Use case scénář může také obsahovat seznam rozšíření.

4.3.1 Spuštění webové aplikace

1. Uživatel zadá do internetového prohlížeče IP adresu nebo doménu řídicí jednotky ručně nebo za pomoci QR kódu.

2. Systém zobrazí webovou aplikaci chytré domácnosti včetně ovládacích komponent a ty nainicializuje podle aktuálního stavu systému. V horní liště zobrazí údaje o teplotě, vlhkosti a tlaku vzduchu venku a teplotu a vlhkost naměřenou uvnitř místnosti.

4.3.2 Nastavení automatického zhasínání světla

1. Uživatel spustí webovou aplikaci.
2. Systém zobrazí hlavní ovládací panel a inicializuje ho.
3. Uživatel stiskne tlačítko Automatizace.
4. Systém uživatele přesměruje na webovou stránku obsahující ovládací panel pro automatizaci a jednotlivé komponenty nainicializuje na základě stavu systému.
5. Uživatel na ovládacím panelu v sekci Automatické zhasínání navolí pomocí droplistu zařízení, kterému chce nastavit automatické zhasínání. Poté nastaví hodnotu políčka Čas na požadovanou hodnotu v sekundách po jaké době se má světlo zhasnout. V droplistu Aktivní pak navolí uživatel, jestli se má nebo nemá zhasínat po uplynutí doby. Po vyplnění těchto parametrů uživatel stiskne tlačítko OK patřící do této sekce.
6. Systém zpracuje požadavek a nastaví automatické zhasínání podle zadaných parametrů.

4.3.3 Nastavení automatického spuštění režimu

1. Uživatel spustí webovou aplikaci.
2. Systém zobrazí hlavní ovládací panel a inicializuje ho.
3. Uživatel stiskne tlačítko Automatizace.
4. Systém uživatele přesměruje na webovou stránku obsahující ovládací panel pro automatizaci a jednotlivé komponenty nainicializuje na základě stavu systému.
5. Uživatel na ovládacím panelu v sekci Automatické spuštění režimu navolí pomocí droplistu režim, kterému chce nastavit automatické spuštění. Poté nastaví hodnotu políčka Spustit v, do kterého uživatel nastaví v kolik hodin a minut se má režim automaticky aktivovat. V droplist poli poté zvolí jestli automatické spuštění bude aplikováno či ne. Po vyplnění těchto parametrů uživatel stiskne tlačítko OK patřící do této sekce.

6. Systém zpracuje požadavek a nastaví automatické spouštění režimu podle zadaných parametrů.

4.3.4 Rozsvícení světla a nastavení jasu a teploty

1. Uživatel spustí webovou aplikaci
2. Systém zobrazí hlavní ovládací panel a inicializuje ho.
3. Uživatel stiskne tlačítko pro světlo, které chce ovládat a které je zabarveno bíle.
4. Systém rozsvítí požadované světlo a zaznamená změnu do systému. Barva stisknutého tlačítka je změněna na žlutou barvu.
5. Uživatel posune posuvník pod tlačítkem na požadovanou úroveň jasu.
6. Systém upraví jas podle uživatelem zadané úrovně. Změnu zaznamená do systému.
7. Uživatel v sekci nastavování barvy a teploty v droplistu zvolí světlo, kterému chce nastavit teplotu.
8. Systém inicializuje panel pro nastavení hodnot RGB a teploty podle hodnot uložených v systému.
9. Uživatel na posuvníku Temp navolí požadovanou teplotu světla.
10. Systém upraví teplotu světla na úroveň zadanou uživatelem a změnu zaznamená do systému.

4.3.5 Nahrání souboru na sdílený disk SMB

1. Uživatel otevře správce souborů, který podporuje SMB protokol a v rámci něj vyhledá dostupná zařízení v lokální síti.
2. Systém se ohlásí, že je dostupný v rámci lokální sítě.
3. Uživatel klikne na ikonu sdílené složky file serverem.
4. Systém vyzve uživatele k zadání přihlašovacích údajů.
5. Uživatel zadá své přihlašovací údaje a odešle je.
6. Systém v případě správně zadaných údajů udělí uživateli přístup. A zobrazí uživateli obsah sdílené složky.
7. Uživatel zvolí ve sdílené složce umístění kam chce uložit soubor a poté pomocí drag and drop či pomocí copy-paste dá příkaz k nahrání souboru.
8. Systém uživatele informuje o průběhu a výsledku nahrání souboru.

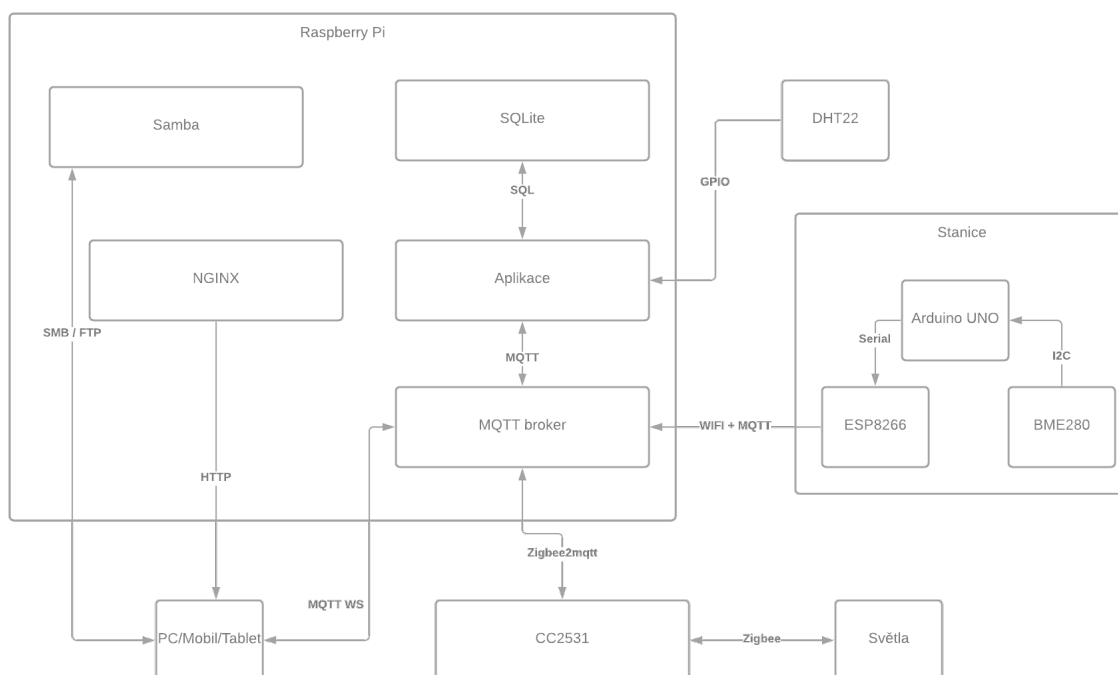
5 Návrh řešení

Po přezkoumání požadavků na systém bylo navrženo schéma zapojení a komunikace jednotlivých částí systému. Řídící jednotkou se stal mikropočítač Raspberry Pi, který má mít na starost sdílení souborů po síti, pracuje jako webový server, obsahuje také systém pro řízení báze dat a je také schopen pracovat jako MQTT broker. Jednotlivá zařízení se připojují k centrální řídicí jednotce pomocí bezdrátové technologie Wi-Fi nebo Zigbee. Důraz byl kladen primárně na spolehlivost systému a jeho jednoduchou mobilnost a přenositelnost.

5.1 Model systému

Hlavní součástí systému je řídicí jednotka, kterou je jednodeskový počítač Raspberry Pi. Na Raspberry je nainstalován file server Samba, který sdílí soubory na síť za použití protokolů SMB a FTP. Za provoz webové aplikace je zodpovědný webový server NGINX, který je také nainstalován v řídicí jednotce. Dále je součástí také systém řízení báze dat, kterým je SQLite. Dále se zde nachází aplikace Eclipse Mosquitto MQTT broker, která má na starost přeposílání MQTT zpráv v rámci celého systému. Součástí systému jsou také čtyři serverové aplikace. K řídicí jednotce je také připojen digitální senzor DHT22 přes GPIO piny. Data z tohoto senzoru jsou zpracována pomocí serverové aplikace a následně uložena do databáze. Jako poslední zařízení, které je připojené k řídicí jednotce je kontroler CC2531, který posílá MQTT zprávy do Zigbee sítě a naopak.

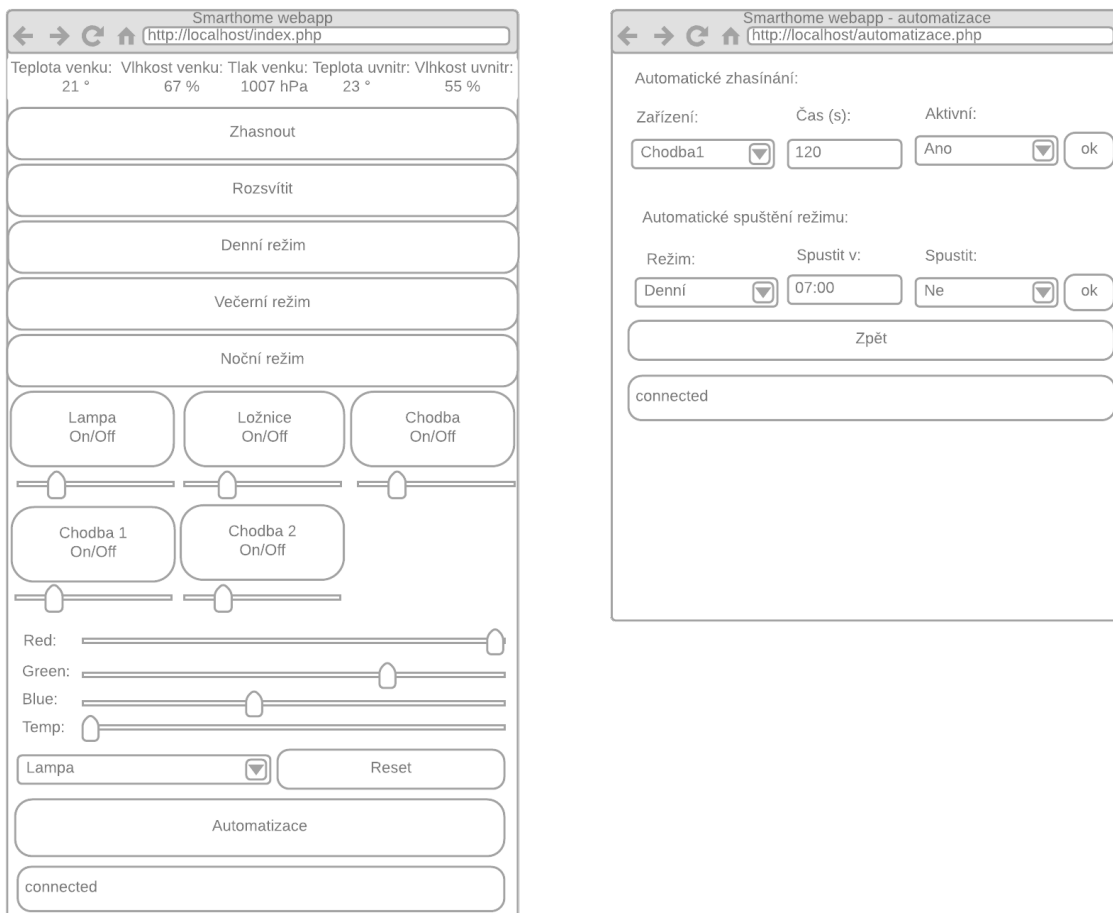
Venkovní senzorová stanice je realizována Arudinem Uno, ke kterému je připojen senzor BME280. Arduino zpracovává data ze senzoru a po sériové sběrnici je předává ESP8266, který je poté posílá přes Wi-Fi jako MQTT zprávu řídicí jednotce.



Obrázek 3: Model systému

5.2 Wireframy

Návrh webové aplikace byl vypracován pomocí wireframů. Jednotlivé ovládací prvky v rámci webové aplikace jsou seřazeny podle očekávané četnosti užití jednotlivých funkcí systému. Webová aplikace slouží také k zobrazení dat o naměřených veličinách. Dále je v aplikaci zahrnuta část pro nastavování automatických činností systému. Je předpokládáno, že světla v domácnosti budou spouštěna a vypínána automaticky systémem nebo klasickými spínači.

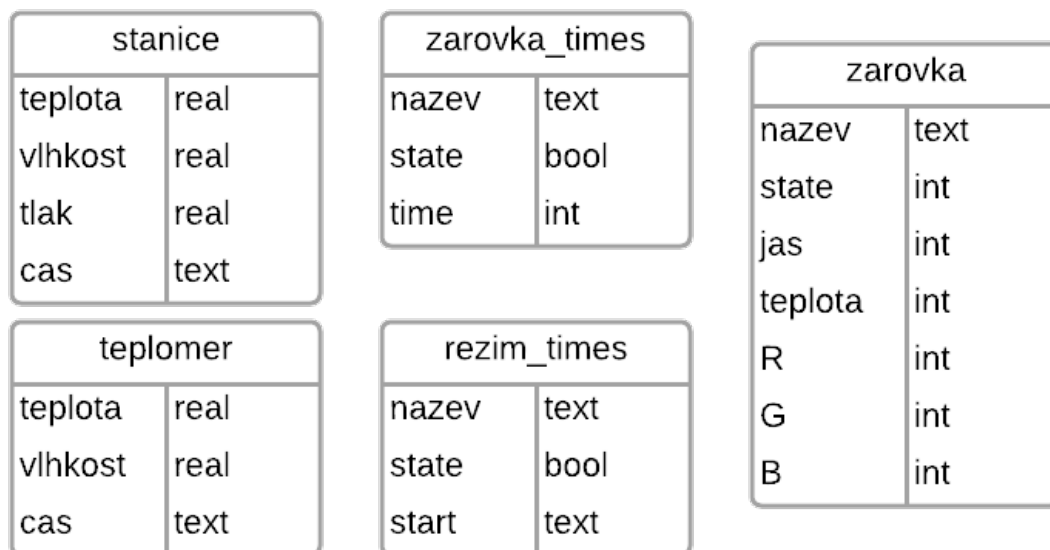


Obrázek 4: Wireframe webové aplikace

5.3 Databázový model

Součástí systému je databáze, která slouží k uchovávání naměřených dat a také k ukládání stavu jednotlivých komponent chytré domácnosti. Dále databáze uchovává nastavení uživatele. Tabulka stanice byla navržena pro ukládání naměřených hodnot venkovní sensorovou stanicí. Tabulka obsahuje čtyři atributy a to teplotu ukládanou jako desetinné číslo, vlhkost ukládanou jako desetinné číslo a i tlak je ukládán jako desetinné číslo. Jednotky teploty jsou stupně Celsia, vlhkost je ukládána v procentech a tlak v hekto-paskalech. Posledním atributem je čas, který je ukládán jako text, protože SQLite nedisponuje datovým typem pro datum. Podobnou strukturu má tabulka pro ukládání dat

vnitřního teploměru avšak tato tabulka nemá atribut tlak, protože senzor DHT22 nedisponuje hardwarem pro měření této veličiny. Tabulka pro ukládání stavů jednotlivých žárovek v systému má atributy název v datovém typu text. Ostatní atributy jsou již v datovém typu int, tedy celé číslo. Atribut state může nabývat hodnot 0 pro zhasnuto, 1 pro rozsvíceno a 2 pro odpojeno ze sítě Zigbee. Atribut jas je pro ukládání hodnoty jasu nastavené pro žárovku a nabývá hodnot od 0 do 254. Atributy R.G a B mají stejné vlastnosti jako atribut jas. Do těchto tří proměnných se bude ukládat barva světla. Posledním atributem v tabulce zarovka je atribut teplota. Do tohoto atributu se bude ukládat hodnota teploty světla nastavená žárovce. Rozmezí je od 250 do 454. Další tabulkou je tabulka zarovka_times, ve které se budou uchovávat uživatelem definované intervaly, po kterých dojde ke zhasnutí konkrétní žárovky. Atribut nazev uchovává název konkrétní žárovky, pro kterou je daný záznam určen. Atribut state uchovává informaci o tom, jestli je tento interval platný nebo jestli nemá být aplikován. Tedy, že světlo má svítit dokud ho někdo nevypne. Atribut time uchovává délku definovaného intervalu v jednotkách sekund. Tabulka rezim_times slouží k ukládání uživatelem nastaveného automatického spouštění jednotlivých režimů osvětlení. Atribut nazev obsahuje název režimu, state jestli je aktivní a má-li na něj být v daný čas přepnuto a atribut start obsahuje čas kdy se tomu tak má stát.



Obrázek 5: Návrh datbázových tabulek

6 Realizace řešení

V rámci realizace navrženého řešení byly jednotlivé komponenty sestaveny, zapojeny a naprogramovány. Během realizace byl zvolen a nainstalován operační systém řídicí jednotky. Byly vytvořeny programy pro venkovní sensorovou stanici. Byl nainstalován a nastaven webový server na Raspberry. Byl zvolen MQTT broker a nainstalován na řídicí jednotku. Proběhla také instalace systému řízení báze dat a vytvoření databáze a tabulek v ní. Bylo vytvořeno přemostění mezi protokoly MQTT a Zigbee. Byly zapojeny žárovky a připojeny do sítě Zigbee. Proběhla instalace a nastavení file serveru. Byly naprogramovány serverové aplikace a webová aplikace.



Obrázek 6: Centrální řídicí jednotka Raspberry Pi

6.1 Operační systém

V rámci realizace musel být na Raspberry Pi nainstalován operační systém. Jako operační systém byl zvolen DietPi. Výhodou tohoto systému je jeho malá velikost, která je menší než 500MB a také minimální vytížení operační paměti a procesoru. Systém obsahuje pouze to nezbytné pro jeho chod a tedy zbývá více operační paměti a výkonu procesoru pro uživatelské aplikace. DietPi nedisponuje žádným desktopovým prostředím a jeho ovládání probíhá tedy pouze přes příkazovou řádku a v případě některých systémových aplikací přes Whiptail menu.

Instalace operačního systému na Raspberry Pi byla provedena za pomoci aplikace balenaEtcher. Nejprve je nutno si stáhnout image DietPi z oficiálních stránek tohoto projektu. Dále je nutno naformátovat microSD kartu o velikosti alespoň 2Gb. Poté lze pomocí balenaEtcher obraz operačního systému nahrát na paměťovou kartu. Po nahrání softwaru na kartu lze na úložišti nalézt soubor dietpi.txt, který slouží k dalším nastavením,

keré se provedou při prvním spuštění systému. Karta byla poté připojena do Raspberry a to bylo spuštěno. Při prvním spuštění se vše automaticky nastavilo podle nastavení v souboru dietpi.txt. Po prvním načtení operačního systému byla nastavena přístupová práva uživatelům a systém byl otevřen pro přístup po lokální síti. Poté byla zjištěna IP adresa Raspberry v síti a od této chvíle dále bylo Raspberry programováno a ovládáno vzdáleně přes ssh připojení.

6.2 Access point

K vytvoření přístupového bodu Wi-Fi byla využita řídicí jednotka Raspberry. Díky tomu, že zvolený model obsahuje jak port pro Ethernet, tak rozhraní pro Wi-Fi, lze vytvořit přístupový bod z Raspberry bez použití jakýchkoliv dalších komponent. Existují dvě varianty přístupových bodů a to takzvané přemostění a samostatná síť. Při přemostění má Raspberry na starost pouze rozšiřování signálu a veškerá přijatá data ze zařízení posílá routeru a data z routeru opět zařízení. Veškerou administraci sítě má na starost router. Veškerou autentizaci a přidělování IP adres provádí tedy router. Druhá varianta samostatné sítě, která je použita i v rámci realizace chytré domácnosti spočívá v tom, že síť Wi-Fi poskytovaná Raspberry je spravovaná právě Raspberry. To má na starost autentizaci a přidělování IP adres.

Při vytváření přístupového bodu byl fyzicky zapojen do Raspberry Ethernet kabel, který ho připojit k routeru. Následně byl obnoven seznam dostupných balíčků a poté pomocí příkazu `sudo apt install hostapd` byl nainstalován balíček pro zprovoznění přístupového bodu. Příkazy k přidání `hostapd` jako služby `sudo systemctl unmask hostapd` a `sudo systemctl enable hostapd` byly provedeny a díky tomu je program pro přístupový bod spouštěn vždy při startu systému automaticky. Dalším krokem bylo nainstalovat balíček `dnsmasq`, který umožňuje služby jako DNS či DHCP. Balíček byl opět nainstalován za pomoci `apt`. Dalšími balíčky, které byly nainstalovány byly `netfilter-persistent` a jeho rozšíření `iptables-persistent`. Tato utilita pomáhá tím, že ukládá pravidla firewallu a znovu je načítá při restartu systému.

Nastavení přístupového bodu je realizováno načtením konfiguračního souboru `hostapd.conf`. V tomto souboru bylo nastaveno SSID sítě, byl zde také zvolen protokol

IEEE80211n, byla zvolena frekvence 2,4 Ghz, kanál byl nastaven na 6, bylo zde také nastaveno heslo a autentizace pomocí WPA2-PSK. K tomu, aby nastavení přístupového bodu bylo uplatněno je zapotřebí restartovat službu hostapd. V dnsmasq.conf je pak nastaven rozsah přidělovaných IP adres, rozhraní, přes které se bude přistupovat do sítě bylo nastaveno na wlan0 a lokální bezdrátová DNS doména na wlan.

6.3 Web server

Jako webový server byl zvolen NGINX na základě toho, že je hardwarově nenáročný a to je potřeba u málo výkoného hardwaru. Instalace byla provedena pomocí správce balíků apt. Nejprve bylo nutno provést aktualizaci seznamu dostupných balíků pomocí příkazu `sudo apt update`. Následně byla provedena instalace webového serveru za užití příkazu `sudo install nginx`. Po instalaci byl spuštěn server příkazem `sudo /etc/init.d/nginx start`. Otestování funkčnosti proběhlo zadáním IP adresy Raspberry do prohlížeče. Výchozí složkou v systému pro webovou stránku je `/etc/www/html/`. Do této složky byly později přidány webové stránky aplikace.

6.4 MQTT broker

Funkcí MQTT brokeru je přeposílání MQTT zpráv v síti. Jako broker byla zvolena aplikace Eclipse Mosquitto. Stejně jako při předchozí instalaci byl nejdříve aktualizován seznam dostupných balíků a poté pomocí příkazu `sudo apt install -y mosquitto mosquitto-clients` byl broker nainstalován. Po instalaci byl broker aktivován jako služba a to má za následek, že ho není nutno spouštět ručně při každém restartu. Příkaz byl `sudo systemctl enable mosquitto.service`. Testování funkčnosti brokeru bylo prováděno za pomoci dvou přihlášených terminálů. Na jednom byl zahájen odběr na dané téma a z druhého byla poslána do tohoto tématu.

6.5 Zigbee2mqtt

Pro komunikaci centrální řídicí jednotky a zařízení v síti Zigbee bylo potřeba vytvořit přemostění. Přemostění bylo vytvořeno za použití kontroleru CC2531, který je připojený do USB portu na Raspberry. Po připojení bylo potřeba zjistit kde se v systému adaptér nachází a jaká uživatelská oprávnění vyžaduje. Poté byla provedena instalace

balíčků potřebných ke správné funkčnosti Zigbee2mqtt. Jako první byl nainstalován Node.js. Poté došlo na instalaci samotného Zigbee2mqtt. Jako poslední krok v rámci instalace byly ještě prověřeny závislosti a jestli jsou všechny splněny.

Nastavení Zigbee2mqtt se provádí pomocí konfiguračního souboru `configuration.yaml`. V konfiguračním souboru bylo potřeba nastavit kořenový topic pro mqtt komunikaci a server. Dále byl nastaven sériový port CC2531, který se nachází na `/dev/ttyACM0`. Dále byl definován klíč sítě a umožnění přihlašování nových zařízení do sítě nastavením proměnné `permit_join` na `true`. Po připojení požadovaných zařízení do sítě zigbee je `permit_join` nastaven zpět na `false` a tím je síť méně zranitelná. Dalším krokem bylo nastavení takzvaných `friendly_name` jednotlivých zařízení v síti pro člověka přívětivější. Například žárovka s původním označením `0x5c0272fffe57864f` byla přejmenována na `loznice`. Tento krok není nutný, nicméně je obecně doporučován pro jednodušší a přehlednější následné programování.

6.6 Databáze

Pro ukládání naměřených hodnot, stavů jednotlivých zařízení systému a nastavení uživatele bylo nutno nainstalovat systém řízení báze dat. Jako systém řízení báze dat byl zvolen `sqlite3`. `Sqlite` je nejrozšířenějším databázovým enginem, protože díky jeho velikosti je součástí téměř všech chytrých telefonů a často je i součástí jiných desktopových aplikací. `Sqlite` je open-source a cross-platformní.

Instalace se tentokrát provedla pomocí správce balíčků `apt-get`. Po instalaci samotného systému řízení báze dat byly vytvořeny jednotlivé tabulky podle návrhu. Byly tedy vytvořeny tabulky `stanice` a `teplomer` pro ukládání naměřených hodnot. Byla také vytvořena tabulka `zarovka` pro zaznamenání údajů o stavech jednotlivých žárovek. Na konec byly vytvořeny ještě tabulky `rezim_times` a `zarovka_times`, do kterých jsou ukládána nastavení uživatele.

Komunikaci mezi databází a zbytkem systému mají na starosti serverové aplikace, o kterých bude pojednáno později.

6.7 File server

V rámci realizace byl nainstalován file server Samba. Instalace opět proběhla za pomoci manažera balíčků apt. Po instalaci Samby bylo nutno nakonfigurovat, které soubory se mají sdílet a také bylo potřeba nastavit uživatele samby, jejich oprávnění a jejich přístupové údaje. Nastavení sdílení je možno konfigurovat pomocí konfiguračního souboru `smb.conf`, který se nachází v adresáři `/etc/samba/`. V souboru se nachází sekce, která je určena pro globální nastavení. V této části byl nastaven název serveru, kterým se bude server prezentovat. Dále byla nastavena část sdílené větve jako veřejná. Její popisek byl nastaven na `Share` a cesta na `/mnt/seagate/share/`. Zapisování a jakákoliv jiná práce než čtení byla zakázána. Přístup nepřihlášeného uživatele byl povolen. Druhou sdílenou větví je větev, která je soukromá. Popisek byl zvolen `private` a cesta `/mnt/seagate/`. Proměnná `valid users` obsahuje pouze jednoho uživatele. Proměnná `public` je nastavena na `no` a přihlášený uživatel, který má přístup do této sdílené větve má práva na čtení i zápis souborů. V rámci nastavení si můžeme všimnout, že sdílené větve se navzájem překrývají. To má za následek, že pokud vyhledáme na síti file server a rozklikneme ho, tak se nám zobrazí dvě složky, jedna s názvem `Share` a druhá s názvem `Private`. Rozdíl mezi nimi je v tom, že po přihlášení ať už jako uživatel nebo host se lze dostat do složky s názvem `Share` avšak do složky `Private` se lze dostat pouze pokud jsme přihlášení jako povolený uživatel. Po dokončení nastavování konfiguračního souboru je potřeba sambu restartovat, aby došlo k načtení změněných hodnot.

V rámci nastavování samby bylo za potřebí přidat uživatele samby. To bylo provedeno příkazem `smbpasswd -a username` spuštěným jako správce. Poté systém vyzval k zadání hesla pro uživatele. Tímto bylo nastavení file serveru dokončeno.

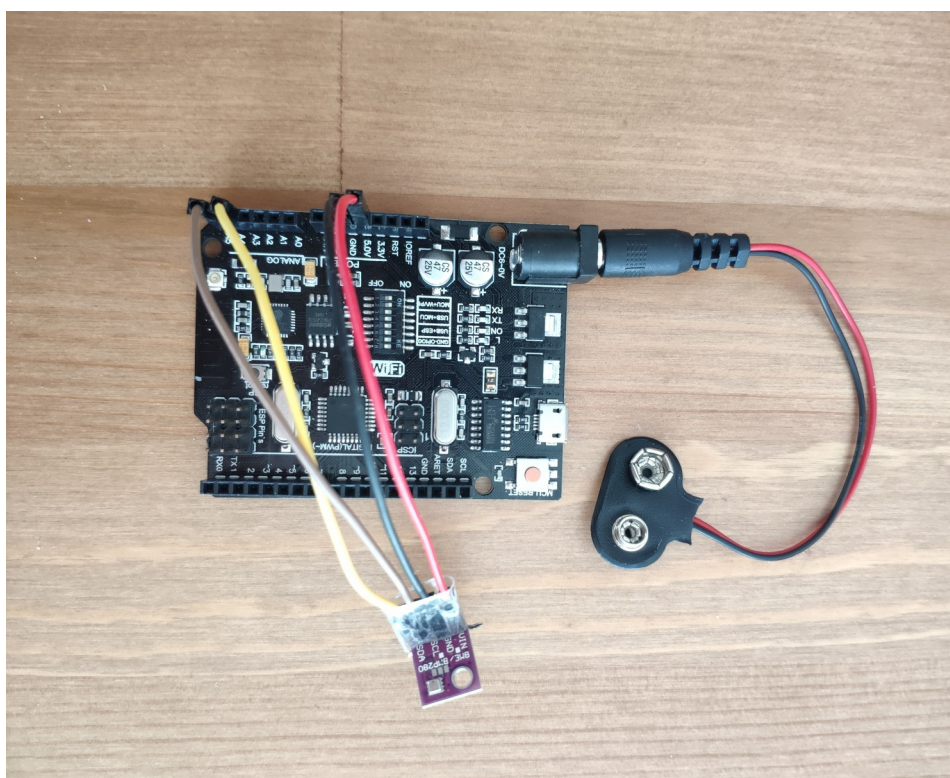
6.8 Venkovní senzorová stanice

Venkovní senzorová stanice se skládá z Arduina Uno, ESP8266 a digitálního multi senzoru BME280. Senzor BME280 je schopen měřit tlak, vlhkost a teplotu. Pro vytvoření venkovní senzorové stanice byl použit upravený klon Arduina Uno. Tento upravený klon disponuje i ESP8266 a není tedy potřeba nic pájet dohromady. Tento jednodeskový počítač

obsahuje switch pomocí kterého je nastaveno, které součásti jsou aktivní a případně jestli mají fungovat spolu či každá zvlášť. To samé platí i pro programování jednotlivých částí.

Nejprve bylo provedeno fyzické zapojení senzoru BME280 do klonu Arduino Uno a to následovně. VIN byl připojen na 5V, GND na GND, SCL na A5 a SDA na A4. Připojení bylo realizováno pomocí dupont prototypovacích nepájecích kabelů samec-samec. Následně bylo Arduino připojeno k počítači, na kterém bylo nainstalováno Arduino IDE určené k vývoji programů. Nejprve bylo potřeba stáhnout rozšíření pro jednotlivé platformy a to pro Arduino Uno a ESP8266. Po základním nastavení Arduino IDE bylo zahájeno programování Arduina. Program se sestává ze zahrnutí knihoven potřebných ke komunikaci se senzorem, zahájení komunikace po seriové lince, navázání spojení se senzorem a poté kódování do json struktury, která je odeslána po sériové lince. Po napsání programu bylo potřeba nahrát program na Arduino. Na desce byl na přepínači nastaven režim pro zapisování programu pro arduino uno. Poté byla deska připojena pomocí USB portu k počítači a po kompilaci programu ho Arduino IDE nahrálo na desku. Dalším krokem bylo programování ESP8266. Úkolem ESP8266 je komunikace s MQTT brokerem přes síť Wi-Fi a odesílání přijatých dat ze sériové linky od Arduina. Na začátku programu se nachází zahrnutí knihoven nezbytných pro MQTT komunikaci a knihovna pro modul ESP8266, který bude zajišťovat připojení k Wi-Fi síti. Dalším krokem v programu je nastavení proměnných, které obsahují SSID sítě, ke které se bude stanice připojovat a heslo. Dále je zde proměnná, do které je nastavena adresa MQTT serveru. Po inicializování těchto proměnných se program pokusí o navázání spojení se sítí Wi-Fi. V případě úspěchu je dále proveden pokus o navázání kontaktu s MQTT brokerem. V případě výpadku nebo neúspěšného přihlášení se do sítě nebo k brokeru je pokus o navázání spojení opakován. Je-li navázáno úspěšné spojení k síti i brokeru, začne program odposlouchávat ze sériové linky a čeká až se k němu dostanou informace zaslané arduinem. V momentě kdy převezme celou zprávu z arduina, tak ji pošle přes síť wifi jako MQTT zprávu. Zpráva je posílána jako json objekt a je posílána do topicu home/stanice/data. V případě, že se stane nějaká chyba jako například odpojení od sítě či rozpojení kabelů mezi senzorem a arduinem je vyslána chybová hláška do topicu home/stanice/error. Po doprogramování je potřeba nastavit desku do módu nahrávání programu pro ESP8266, což je zase provedeno pomocí přepínače na desce. V Arduino IDE je jako vývojová deska zvoleno ESP8266 a

poté lze spustit kompilaci. Po úspěšném sestavení programu je program připraven k nahrání na desku. Po nahrání programu na desku byla deska odpojena od počítače a pomocí přepínače byla nastavena tak, aby běželo jak Arduino Uno tak i ESP8266 a aby společně komunikovali přes sériovou linku. To je nastaveno tak, že první dva přepínače na desce jsou nastaveny na 1 a zbytek na nula. Poté byl k desce připojen akumulátor a deska byla spuštěna. Funkčnost desky byla otestována pomocí klientského programu na straně centrální jednotky, kde byl přihlášen odběr na téma home/stnice/data a v momentě kdy stanice odeslala naměřená data a broker je přijal, byla zobrazena na obrazovce terminálu.



Obrázek 7: Venkovní senzorová stanice

6.9 Serverové aplikace

Tato část pojednává o aplikacích, které běží na centrální jednotce a zajišťují chod chytré domácnosti. Aplikace jsou naprogramovány v programovacím jazyce python3. A jsou spuštěny operačním systémem jako služby. Serverové aplikace zastřešují komunikaci mezi systémem a databází. Na základě dat z databáze a událostem objevujícím se v systému vyhodnocují co se má stát dále.

6.9.1 Zapisovač naměřených dat

První aplikace, která byla naprogramována je aplikace pro zápis naměřených dat do databáze. Tato aplikace má za úkol se přihlásit k odběru tématu, do kterého jsou zasílány naměřené hodnoty. V případě venkovní stanice je téma pro data, která stanice naměřila `home/stanice/data` a v případě vnitřního teploměru to je téma `home/stanice/uvnitr`. Přijatá data, pokud se jedná o naměřené hodnoty a ne chybové hlášky jsou pak uložena do databáze. Aby mohla aplikace fungovat potřebuje nainportovat knihovny `paho.mqtt.client` pro možnost komunikování s mqtt brokerem. Další důležitou knihovnou je knihovna `sqlite3`, díky které může aplikace komunikovat se systémem řízení báze dat. A v neposlední řadě je potřeba nainportovat knihovnu pro práci s json objekty. V aplikaci je potřeba nastavit kde má hledat databázový soubor, adresu mqtt brokeru a port, přes který se k němu bude přistupovat. Program fuguje tak, že po spuštění se brokeru přihlásí jako odběratel tématu `home/stanice/#`. Mřížka znamená cokoliv co se nachází na této úrovni. Aplikace tedy během svého chodu poslouchá na dané téma a při přijetí zprávy na základě tématu, do kterého přišla s ní naloží. V případě, že se jedná o zprávu od venkovní stanice jsou naměřená data zapsána do tabulky stanice a v případě, že se jedná o zprávu od vnitřní stanice, jsou data zanesena do tabulky `teplomer`. Díky tomu, že stanice vysílají data jako json objekty je práce s nimi mnohem přívětivější. Do databáze je i ukládáno v rámci záznamu, kdy byl záznam přidán.

6.9.2 Program vnitřního teploměru

Jako vnitřní teploměr byl zvolen digitální teploměr DHT22, který je připojen k Raspberry Pi pomocí GPIO pinů. První pin zleva na DHT22 je připojen k prvnímu pinu na Raspberry, který slouží jako přívod napětí. Druhý pin je datový a ten je potřeba spojit s pinem GPIO4, který se nachází jako pin číslo sedm na desce Raspberry. Třetí pin se nepřipojuje. A čtvrtý pin slouží k uzemnění a tedy ho připojíme na šestý pin, který slouží k uzemnění. Teploměr umí měřit vlhkost vzduchu a jeho teplotu. Aplikace ke své funkci využívá knihovny `paho.mqtt.publish` pro publikování mqtt zpráv, knihovnu `board` pro komunikaci s GPIO piny a knihovnu `adfruit_dht` pro čtení dat senzoru. Naměřené hodnoty jsou poté poslány jako mqtt zpráva do vlákna `home/stanice/uvnitr`. Na toto téma poslouchá

již výše zmíněná aplikace pro zapisování naměřených dat, která si zprávu převezme a naměřená data uloží do databáze. Aplikace vysílá naměřená data jednou za deset minut, protože se očekává, že se teplota nemění příliš rychle.

6.9.3 Program časovač

Tento program má na starost činnosti systému, které jsou založené na nějakém časovém údaji. Může se jednat o pravidelný impuls pro hlavní program, aby zkontroloval připojení zařízení a jejich stavy, nebo se také může jednat o vyslání impulsu ke zhasnutí světla po uplynutí stanoveného intervalu uživatelem a nebo vyslání impulsu k aktivování daného režimu ve stanovený čas uživatelem. Aplikace využívá k publikování těchto impulsů knihovnu `paho.mqtt.publish`, díky které je schopna zaslat zprávu do tématu na kterém odposlouchává hlavní program. Dále program pracuje s knihovnou `datetime` a `sqlite3`. Aplikace se připojuje na databázi, ve které z tabulek získává data o uživatelském nastavení a o stavu jednotlivých zařízení, které dále využívá ke svojí funkci.

6.9.4 Hlavní aplikace

Hlavní aplikace je v podstatě ve středu celého systému. Jejím úkolem je získávat data o jednotlivých částech systému a ty zanášet do databáze a odpovídat na dotazy jiných aplikací v rámci systému. V momentě kdy nastane nějaká změna, například se rozsvítí světlo, tak hlavní program tuto skutečnost zaznamená do databáze. Aplikace má nastavené reakce na různé zprávy v různých tématech. Zprávy lze rozdělit na dvě základní kategorie podle reakce na ně. Jednou z možných reakcí je, že hlavní program zašle zprávu s požadovaným obsahem například poslední data z teploměru. A druhou možností je, že hlavní program po přijetí zprávy vykoná nějaký úkon. Například dostane zprávu, aby došlo ke zhasnutí všech světel v domě. Program tedy vyšle všem světlům zprávu o tom, že se mají zhasnout a poté zaznamená do databáze stavů skutečnost, že všechna světla jsou zhasnuta. Funkcí hlavního programu tedy je informovat o aktuálním stavu systému a realizovat ovládání chytré domácnosti na základě přijatých zpráv obsahujících příkazy.

6.10 Webová aplikace

Webová aplikace je naprogramována v jazycích HTML a javascript. Jako framework byl využit Bootstrap 4. Webová aplikace je umístěna na webovém serveru NGINX. Do aplikace se lze dostat zadáním IP adresy Raspberry do webového prohlížeče. Komunikace mezi prohlížečem a webovým serverem probíhá za pomoci protokolu HTTP. Po otevření stránky se zobrazí hlavní ovládací panel. Webová aplikace se přihlásí k MQTT brokeru a k odběru zpráv v rámci systému. Poté zašle žádost hlavní serverové aplikaci o to aby ji zpět zaslala celkový stav systému a jeho aktuální nastavení a také je požádáno o aktuální naměřená data ze senzorových stanic. Po přijetí zpráv o stavu systému a aktuálních datech ze stanic jsou jednotlivé části hlavního ovládacího panelu inicializovány. V horní liště s nám tedy zobrazí údaje o tlaku, teplotě a vlhkosti vzduchu venku a o teplotě a vlhkosti uvnitř místnosti. Dále hlavní panel obsahuje tlačítka, pro zhasnutí či rozsvícení všech světel a spuštění předdefinovaných režimů. U těchto tlačítek k žádné inicializaci nedochází, protože jejich funkcionality je pevně dána. Na druhou stranu tlačítka, která vyžadují inicializaci jsou tlačítka pro jednotlivá světla. Tlačítka pro ovládání jednotlivých světel mohou nabývat tři barev podle stavu ve kterém se nacházejí. Pokud je tlačítko žluté, znamená to, že světlo svítí. Pokud je tlačítko bílé, znamená to, že světlo nesvítí. A pokud je tlačítko červené znamená to, že světlo není připojeno do sítě zigbee. Pod tlačítky se nachází posuvníky, pomocí kterých lze nastavit jas konkrétního světla. I ty jsou nastaveny podle dat obdrženy od hlavního serverového programu na pozici odpovídající úrovni nastaveného jasu. Dalšími ovládacími prvky jsou další posuvníky a to pro hodnotu červené, zelené, modré a teploty světla pro konkrétní světlo zvolené za pomocí droplistu. Dále se zde nacházejí ještě tlačítka reset a automatizace. Tlačítko reset nastaví hodnoty pro červenou, zelenou, modrou a teplotu světla na tovární pro zvolené světlo v droplistu. Tlačítko automatizace slouží k přesměrování na stránku pro konfiguraci automatických činností. Na konci stránky lze nalézt stavový řádek, který nás informuje o tom jestli je webová aplikace připojena k MQTT brokeru. Pokud ano, tak je stavový řádek zelený a hlásí, že je připojen. V případě, že dojde k odpojení od brokeru, tak o tom aplikace informuje prostřednictvím právě tohoto stavového řádku tak, že zčervená a nabídne možnost, že po kliknutí na stavový řádek bude zahájen nový pokus o připojení. Na stránce pro automatizaci jsou dvě sekce. První je pro nastavování automatického zhasínání

konkrétního světla po určitém intervalu. A druhá sekce slouží k nastavení automatického spuštění daného režimu v určitou dobu. Aplikace po jakékoliv interakci s uživatelem odesílá data o změně parametru a zobrazí je. Například uživatel stiskne tlačítko lampa. V ten moment aplikace zasílá informaci o stisknutí tlačítka hlavní serverové aplikaci. Ta požadavek vyhodnocuje a podle stavu vyše světlu příkaz, aby se zhaslo nebo rozsvítilo. Výsledek pak zaznamená do databáze a informuje o aktuálním stavu mqtt zprávou v daném tématu. To má za následek, že webové aplikace otevřené na jiných zařízeních se také dozví o změně v systému a budou ji prezentovat uživateli. Komunikace mezi webovou aplikací a MQTT brokerem probíhá pomocí MQTT websockets. Web je tedy dynamický i bez použití PHP a není nutno při aktualizování obsahu znovu načítat celou stránku. Webová aplikace je přihlášená k odběru zpráv v kořenovém tématu, kterým je home/ a proto je schopna přijmout a reagovat na zprávy vyslané jak z venkovní senzorové stanice, tak i na zprávy vysílané programem pro vnitřní teploměr. To má za následek, že lišta s naměřenými hodnotami na webové stránce je aktualizována v momentě, kdy jsou publikována nová data stanicí nebo vnitřním teploměrem a není nutno čekat na odezvu hlavní serverové aplikace. O responzivitu webového rozhraní se stará framework Bootstrap. Díky němu je webová aplikace prezentována jinak na displeji počítače či tabletu. A jinak na displeji mobilního telefonu.

7 Výsledky

7.1 Hodnocení vytvořeného systému

Cílem této práce bylo navrhnout a vytvořit hardwarové a softwarové řešení pro chytrou domácnost a dílčím cílem bylo navrhnout a vytvořit webové rozhraní určené pro komunikaci mezi uživatelem a chytrou domácností. Nástroje využívané k vytvoření řešení chytré domácnosti byly Arduino IDE, Sublime Text a nano. Programovacími jazyky pak byly Python, C, HTML a javascript.

Výsledkem diplomové práce je funkční systém pro chytrou domácnost, který je schopen konat úkony podle definovaných požadavků. Avšak jako každý systém i systém vytvořený v rámci této práce má svá slabší místa.

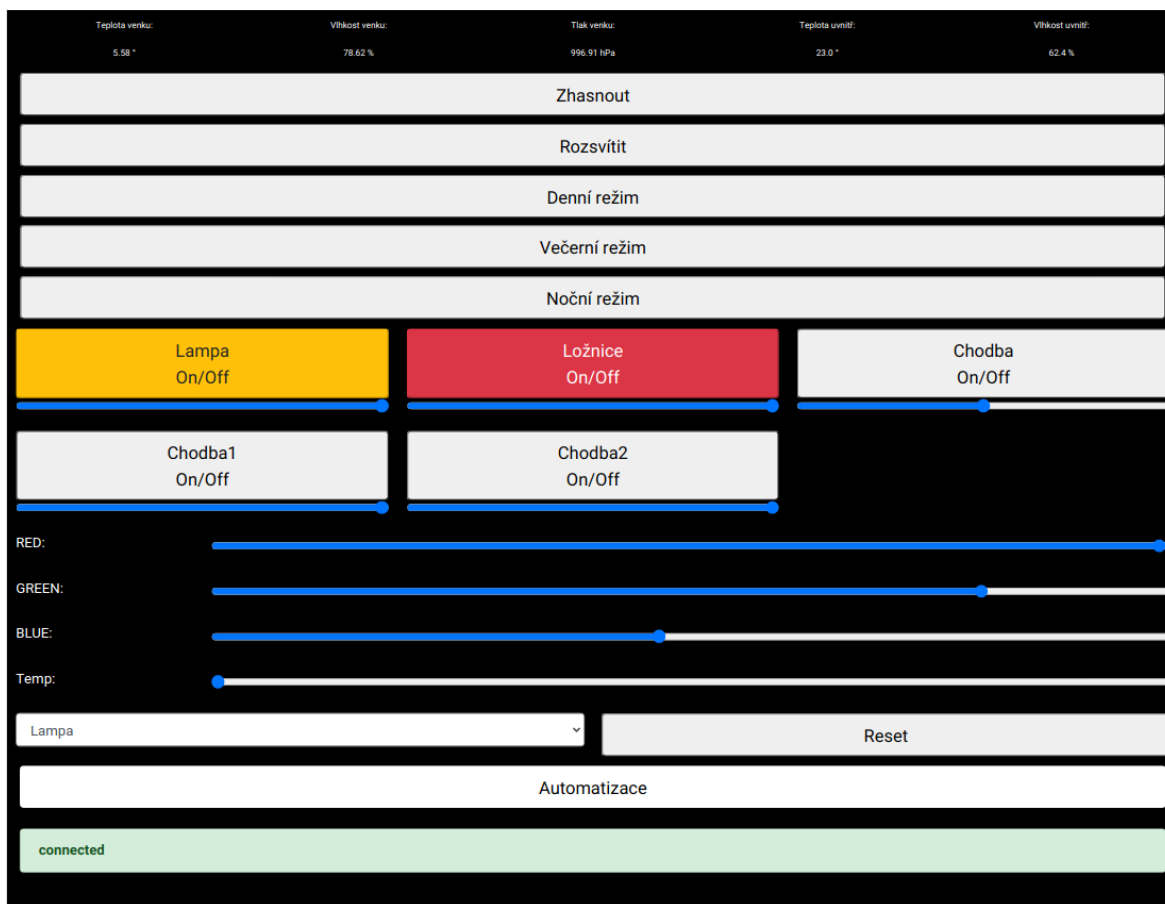
7.1.1 Klady

Užívání webové aplikace pro ovládání systému je jednoduché a intuitivní. K návrhu rozložení jednotlivých ovládacích a informačních prvků v rámci webové aplikace byly uplatněny znalosti nabyté v rámci studia předmětů Interakčního designu. Systém je robustní a spolehlivý. Další zásadní výhodou je, že systém může komunikovat se zařízeními zigbee od více výrobců.

7.1.2 Zápory

Mezi zápory systému je nutné zařadit delší odezvu při ovládání skrze webovou aplikaci oproti klasickému spínači. Další nevýhodou systému je absence hardwarových ovládacích prvků k nastavení jasu, teploty a barvy světla. Pro nastavení těchto parametrů je nutno použít webovou aplikaci, a to může být problémem v situaci, kdy člověk nemá po ruce žádné zařízení, na kterém by se mohl připojit k ovládacímu panelu.

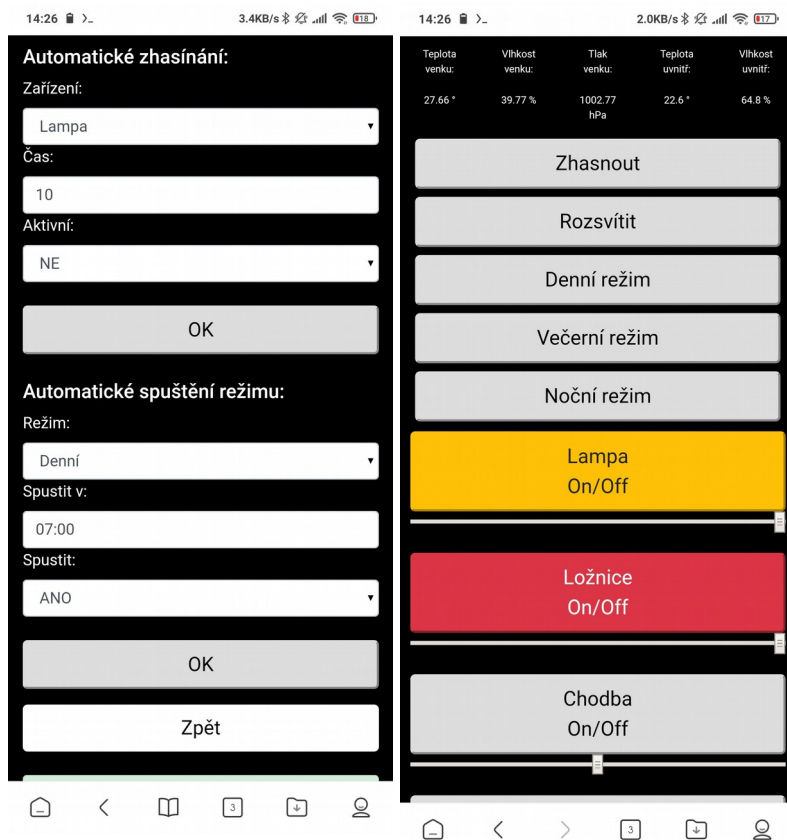
7.2 Finální podoba webového rozhraní



Obrázek 8: Webové rozhraní Hlavní panel pro desktop



Obrázek 9: Webové rozhraní Automatizace pro desktop



Obrázek 10: Webové rozhraní Automatizace pro mobil

Obrázek 11: Webové rozhraní Hlavní panel pro mobil

7.3 Ukázky kódů

```
#include <SPI.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_BME280.h>
// nastavení adresy senzoru
#define BME280_ADRESA (0x76)
// inicializace senzoru BME z knihovny
Adafruit_BME280 bme;

void setup() {
  Serial.begin(115200);

  if (!bme.begin(BME280_ADRESA)) {
    Serial.println("!ERROR");
    while (1);
  }
}

void loop() {
  Serial.print("{ ");
  Serial.print("\"Sensor_ID\":\");
  Serial.print("BME280_1");
  Serial.print("\", ");
  Serial.print("\"Temperature\":\");
  Serial.print(bme.readTemperature());
  Serial.print("\", ");
  Serial.print("\"Humidity\":\");
  Serial.print(bme.readHumidity());
  Serial.print("\", ");
  Serial.print("\"Pressure\":\");
  Serial.print(bme.readPressure()/100.0F);
  Serial.print("\n }");
  delay(600000);
}
```

Obrázek 12: Program senzorové stanice

```

#!/usr/bin/python3
import adafruit_dht
import time
import board
import paho.mqtt.publish as publish

dhtSensor = adafruit_dht.DHT22(board.D4)

while True:
    try:
        humidity = dhtSensor.humidity
        temp_c = dhtSensor.temperature
    except RuntimeError:
        print("error")
        continue

    msg = "{ \"Temperature\": \"" + str(temp_c) + "\", \"Humidity\": \"" + str(humidity) + "\" }"
    publish.single("home/stanice/uvnitr",msg,hostname="10.0.1.27")
    #print(msg)
    time.sleep(600)

```

Obrázek 13: Program vnitřního teploměru

```

function onMessageArrived(msg) {

    if(msg.destinationName=="home/zhong/dataVenku") {
        var out_msg=msg.payloadString;
        var obj = JSON.parse(out_msg);
        document.getElementById("temperature").innerHTML = obj.Teplota+" °";
        document.getElementById("humidity").innerHTML = obj.Vlhkost+" %";
        document.getElementById("pressure").innerHTML = obj.Tlak+" hPa";
    }
    else if(msg.destinationName=="home/stanice/data") {
        var out_msg=msg.payloadString;
        var obj = JSON.parse(out_msg);
        document.getElementById("temperature").innerHTML = obj.Temperature+" °";
        document.getElementById("humidity").innerHTML = obj.Humidity+" %";
        document.getElementById("pressure").innerHTML = obj.Pressure+" hPa";
    }
    else if(msg.destinationName=="home/zhong/dataUvnitr") {
        var out_msg=msg.payloadString;
        var obj = JSON.parse(out_msg);
        document.getElementById("inside_temperature").innerHTML = obj.Teplota+" °";
        document.getElementById("inside_humidity").innerHTML = obj.Vlhkost+" %";
    }
    else if(msg.destinationName=="home/stanice/uvnitr") {
        var out_msg=msg.payloadString;
        var obj = JSON.parse(out_msg);
        document.getElementById("inside_temperature").innerHTML = obj.Temperature+"
        °";
        document.getElementById("inside_humidity").innerHTML = obj.Humidity+" %";
    }
}

```

Obrázek 14: Ukázka Javascriptu Webové aplikace

8 Závěr

Cílem diplomové práce bylo vytvořit systém pro chytrou domácnost. Dílčím úkolem bylo vytvořit webovou aplikaci pro ovládání chytré domácnosti. Byla provedena formulace požadovaných funkcionalit na základě nároků budoucích uživatelů a vyvození souvisejících funkčních požadavků. Na základě požadavků byl sestaven use case diagram a poté byly sepsány use case scénáře. Celý proces je popsán v kapitole 4. - Analýza požadavků. Návrh modelu zapojení celého systému na základě use casů je popsán v části 5.1. - Model systému. Návrh webové aplikace na úrovni wireframů je popsán v části 5.2: - Wireframy. Databázový model, který sloužil jako předloha pro vytvoření databáze je zobrazen a popsán v části 5.3. - Databázový model. Instalace operačního systému na centrální řídicí jednotku je popsána v části 6.1. - Operační systém. Vytváření přístupového bodu Wi-Fi je popsáno v části 6.2. - Access point. O instalaci a nastavení webového serveru se píše v části 6.3. - Web server. Instalace MQTT brokeru a jeho testování je popsána v části 6.4. - MQTT broker. Vytvoření přemostění se sítí Zigbee pomocí kontroleru CC2531 a konfigurace tohoto přemostění je sepsána v části 6.5. - Zigbee2mqtt. Instalace systému řízení báze dat a vytvoření databáze je popsáno v části 6.6. - Databáze. Realizace File serveru včetně jeho konfigurace a nastavení uživatelských práv je sepsáno v části 6.7. - File Server. Sestavení, zaopjení a programování programů pro venkovní sensorovou stanici je popsáno v části 6.8 - Venkovní sensorová stanice. Vytváření systémových aplikací pro zapisování dat do databáze, získávání naměřených hodnot, spouštění časově řízených událostí a hlavní serverová aplikace jsou popsány v části 6.9. - Serverové aplikace. Vytvoření webové aplikace a jejích funkcionalit na základě wireframu z části 5.2. - Wireframy je popsáno v části 6.10. - Webová aplikace. Realizace se odvíjela od návrhu modelu systému popsaném v části 5.1. - Model systému, a na základě případů užití popsaných v části 4.1. - Use case. Lze tedy konstatovat, že byly naplněny všechny požadavky a tudíž i cíle práce.

Potencionální budoucí rozvoj systému spočívá v obsažení více zařízení a zautomatizování dalších částí běžného života. S přibývajícimi zařízeními bude potřeba vždy kód více zobecnit a zajistit jeho opakované užití. Toho by mohlo být docíleno pomocí refactoringu. I když byl systém pro momentální potřeby navržen co nejlépe, tak nemusí v budoucnosti vyhovovat. Požadavky lidí na systémy se s časem mění a je tedy možné, že

časem se změní natolik, že bude potřeba systém předělat k nepoznání. Nabízí se třeba možnosti automatického zavlažování rostlin a krmení domácích mazlíčků. Otázkou však zůstává jestli hobby pokud ho příliš zautomatizujeme jestli stále je tím stejným koníčkem jako předtím.

9 Seznam použitých zdrojů

- [1] The History of WiFi: 1971 to Today - CableFree. CableFree: 4G & 5G 10 Gigabit Wireless Technology [online]. Dostupné z: <https://www.cablefree.net/wireless-technology/history-of-wifi-technology/>
- [2] Jak zapojíme síť: WiFi bez tajemství - Zabezpečení sítě | Svět hardware. Svět hardware | homepage [online]. Copyright © 1998 [cit. 29.03.2021]. Dostupné z: <https://www.svethardware.cz/jak-zapojime-sit-wifi-bez-tajemstvi/12953-6>
- [3] What Is Bluetooth?. Lifewire: Tech News, Reviews, Help & How-Tos [online]. Dostupné z: <https://www.lifewire.com/what-is-bluetooth-2377412>
- [4] KRAJICEK, T. Moderní bezdrátová technologie – ZigBee: bakalářská práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2009. 51 s. Vedoucí semestrální práce Ing. Michal Skořepa.
- [5] Co je MQTT a k čemu slouží ve IIoT? Popis protokolu MQTT— iPC2U s.r.o. [online]. Dostupné z: https://ipc2u.tech/blogs/news/mqtt-protokol?gclid=Cj0KCQjw9YWDBhDyARIsADt6sGbhtkWIHy1CYFF4z3RWaMJoZQOve2yJl7dC31H5PHFgVqGUY7Zp0YaAsrREALw_wcB
- [6] MQTT: univerzální protokol nejen pro cloudové aplikace | Automatizace.HW.cz. Automatizace.HW.cz | Elektronika v automatizaci [online]. Dostupné z: <https://automatizace.hw.cz/mqtt-univerzalni-protokol-nejen-pro-cloudove-aplikace.html>
- [7] What is Z-Wave? - Definition from WhatIs.com. Internet of Things (IoT) news, blogs and analysis - IoTAgenda.com [online]. Dostupné z: <https://internetofthingsagenda.techtarget.com/definition/Z-Wave>
- [8] DEPIAK, P. Realizace víceúčelového serveru na bázi WINDOWS. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2008. 49 s. Vedoucí bakalářské práce Ing. Václav Pfeifer.
- [9] Arduino - Environment . Arduino - Home [online]. Dostupné z: <https://www.arduino.cc/en/guide/environment>
- [10] DietPi - FriendlyARM Wiki. [online]. Dostupné z: <https://wiki.friendlyarm.com/wiki/index.php/DietPi>
- [11] What Is Apache Web Server? What It Is and How It Works. Kinsta - Managed WordPress Hosting for All, Large or Small [online]. Copyright © 2021 Kinsta Inc. All

rights reserved. [cit. 29.03.2021]. Dostupné z: <https://kinsta.com/knowledgebase/what-is-apache/>

[12] What is NGINX? - NGINX. NGINX | High Performance Load Balancer, Web Server, & Reverse Proxy [online]. Copyright © F5, Inc. All rights reserved. [cit. 29.03.2021]. Dostupné z: <https://www.nginx.com/resources/glossary/nginx/>

[13] Samba - opening windows to a wider world [online]. Dostupné z: <https://www.samba.org/samba/docs/SambaIntro.html>

[14] Raspberry Pi: what is it and why do i need one? - Liquid Light. Liquid Light - Award winning web design & digital agency [online]. Copyright © Liquid Light Ltd 2021 [cit. 29.03.2021]. Dostupné z: <https://www.liquidlight.co.uk/blog/raspberry-pi-what-is-it-and-why-do-i-need-one/>

[15] Arduino - Introduction . Arduino - Home [online]. Dostupné z: <https://www.arduino.cc/en/guide/introduction>

[16] ESP8266 Introduction. Fab Academy [online]. Copyright © 2015 [cit. 29.03.2021]. Dostupné z: <http://fabacademy.org/archives/2015/doc/networking-esp8266.html>

[17] CC2531 data sheet, product information and support | TI.com. Analog | Embedded processing | Semiconductor company | TI.com [online]. Dostupné z: <https://www.ti.com/product/CC2531#product-details###description>

[18] Humidity Sensor BME280 | Bosch Sensortec. Bosch MEMS Technology | Bosch Sensortec [online]. Copyright © Bosch Sensortec GmbH 2021, all rights reserved [cit. 29.03.2021]. Dostupné z: <https://www.bosch-sensortec.com/products/environmental-sensors/humidity-sensors-bme280/>

[19] DHT22 temperature-humidity sensor + extras : ID 385 : \$9.95 : Adafruit Industries, Unique & fun DIY electronics and kits. Adafruit Industries, Unique & fun DIY electronics and kits [online]. Dostupné z: <https://www.adafruit.com/product/385>