

UNIVERZITA PARDUBICE
Fakulta elektrotechniky a informatiky

Senzorická síť
Ondřej Ludvík

Bakalářská práce
2022

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: 2021/2022

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Ondřej Ludvík**
Osobní číslo: **I19201**
Studijní program: **B0714P060001 Aplikovaná elektrotechnika**
Téma práce: **Senzorická síť**
Zadávací katedra: **Katedra elektrotechniky**

Zásady pro vypracování

Pomocí platformy ESP8266/ESP32 vytvořte zabezpečenou senzorickou síť pro min. 7 senzorů a jeden master. Implementujte řízení spotřeby, obousměrnou komunikaci, režim Master-multislave / mesh síť. Ošetřete stavy přidání nového senzoru a odebrání senzoru. Oměřte příkonovou náročnost senzorů během komunikace. Komunikaci zabezpečte a popište způsob zabezpečení.

Rozsah pracovní zprávy: **30-40**
Rozsah grafických prací:
Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam doporučené literatury:

- [1] Matoušek D., Práce s mikrokontroléry Atmel AVR, BEN technická literatura, Praha 2006, ISBN: 80-7300-209-4
- [2] Joseph Yiu., The Definitive Guide to the ARM Cortex-M3 (Embedded Technology) 1st Edition, ISBN-13: 978-0750685344
- [3] Herout Pavel., Učebnice jazyka C – 1. díl, Kopp, ISBN-13: 978-80-7232-383-8

Vedoucí bakalářské práce: **Ing. Martin Dobrovolný, Ph.D.**
Katedra elektrotechniky

Datum zadání bakalářské práce: **15. listopadu 2021**
Termín odevzdání bakalářské práce: **13. května 2022**

Ing. Zdeněk Němec, Ph.D. v.r.
děkan

L.S.

Ing. Jan Pidanič, Ph.D. v.r.
vedoucí katedry

V Pardubicích dne 31. ledna 2022

Prohlášení autora

Prohlašuji:

Tuto práci jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Beru na vědomí, že v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů, a směrnicí Univerzity Pardubice č. 9/2012, bude práce zveřejněna v Univerzitní knihovně a prostřednictvím Digitální knihovny Univerzity Pardubice.

V Pardubicích dne 14. 12. 2022

Ondřej Ludvík

Poděkování

Rád bych poděkoval vedoucímu mé práce Ing. Martinu Dobrovolnému, Ph.D. za jeho čas, pomoc, rady a trpělivost při zpracovávání mé práce. Děkuji Ing. Janu Pidaničovi, Ph.D. za cenné rady k psaní této práce a jeho velmi vlídný přístup. Dále děkuji svým spolužákům a kamarádům za nespočet rad a připomínek při tvorbě mé bakalářské práce. V neposlední řadě děkuji mé rodině, díky které jsem měl vůbec možnost studovat na Univerzitě Pardubice.

Anotace

Cílem práce je sestrojít zabezpečenou senzorickou síť pomocí platformy ESP32. Součástí sítě je implementace řízení spotřeby, obousměrná komunikace, režim master-multislave / mesh síť. Změřením příkonové náročnosti jednotlivých senzorů během komunikace se ověří účinnost řízení spotřeby. Dále je potřeba ošetřit stavy přidání a odebrání nového senzoru.

Klíčová slova

ESP32, senzorická síť, zabezpečení, ESP-NOW, mesh

Title

Sensor network

Annotation

This bachelor work is focused on building a secured sensor network using the ESP32 platform. The network includes implementation of power management, two-way communication, master-multislave mode / mesh network. By measuring the power consumption of individual sensors during communication, the effectiveness of power management is verified. It is also necessary to handle the states of adding and removing a new sensor.

Keywords

ESP32, sensor network, security, ESP-NOW, mesh

Obsah

Seznam zkratk.....	8
Seznam obrázků.....	9
Seznam tabulek.....	9
Úvod.....	10
1 ÚVOD DO PROBLEMATIKY	11
1.1 Co je senzorická síť	11
1.2 Druhy sítí.....	11
1.2.1 Sběrnice (bush).....	11
1.2.2 Hvězda (star).....	11
1.2.3 Kruh (ring).....	12
1.2.4 Mesh síť.....	12
2 Obvyklá řešení a dostupné technologie	13
3 Návrh sítě	17
3.1 Technologie přenosu.....	18
3.1.1 Bluetooth	18
3.1.2 Wifi.....	18
3.1.3 ESP-NOW	18
3.2 Zabezpečení.....	19
3.3 Očekávaná spotřeba a možnosti jejího snížení.....	19
3.3.1 Úsporné režimy	20
3.4 Časová synchronizace.....	22
3.4.1 Synchronizace času přes NTP	22
3.4.2 Millis.....	22
3.4.3 Použití knihoven pro synchronizaci a editaci času.....	22
3.5 Přidávání a odebírání uzlů	23
4 Praktická část návrhu	24
4.1 Architektura sítě	24
4.2 Nastavení přístupového bodu	24
4.3 Inicializace ESP-NOW	24
4.4 Vyhledání klientů v síti	25
4.5 Připojení klienta do sítě.....	27

4.6 Způsob distribuce paketů skrz síť	29
4.7 Zapnutí a vypnutí úsporného módu	30
4.8 Časová synchronizace.....	32
4.9 Použití vláken a časovače watchdog	33
4.10 Použití EEPROM.....	33
4.11 Formát odesílaných zpráv.....	34
4.12 Odpojení klientů ze sítě	34
4.13 Porovnání očekávané spotřeby s reálnou.....	35
4.14 Zobrazení dat	39
Závěr	41
Literatuta.....	42
Příloha A.....	44

Seznam zkratek

IoT	Internet of Things
IEEE	Institute of Electrical and Electronics Engineers
PAN	personal area network
AES	Advanced Encryption Standard
IPv6	Internet Protocol version 6
ECDH	Elliptic-curve Diffie–Hellman
MITM	Man-in-the-middle
LED	Light-Emitting Diode
BSS	Basic service set
PMK	Private Master Key
LMK	Local Master Key
WPA2	Wi-Fi Protected Access 2
TKIP	Temporal Key Integrity Protocol
RC4	Rivest Cipher 4
ULP	Ultra Low Power
RTC	Real-time clock
SRAM	Static Random Access Memory
EEPROM	Electrically Erasable Programmable Read-Only Memory
NTP	Network Time Protocol
MAC	Media Access Control
SSID	Service Set Identifier
RSSI	Received Signal Strength Indication
BSSID	Basic service set identifier
TTL	Time to live
CCMP	Counter Cipher Mode with Block Chaining Message Authentication Code Protocol

Seznam obrázků

Obrázek 1 Topologie mesh sítě	12
Obrázek 2 Launchpad SensorTag CC1352R [1]	13
Obrázek 3 Sonoff ZigBee Bridge [20]	14
Obrázek 4 Eve Motion [4]	14
Obrázek 5 NRF52840 Dongle [8]	15
Obrázek 6 SLWSTK6050B [10]	15
Obrázek 7 ESP32-DevKit přehled základních funkcí jednotlivých pinů [19]	17
Obrázek 8 Úsporné režimy ESP32 [22]	21
Obrázek 9 Diagram funkce pro najetí okolních zařízení pomocí WiFi	26
Obrázek 10 Diagram způsobu distribuce šifrovacích klíčů	28
Obrázek 11 Způsob distribuce paketu skrz síť	29
Obrázek 12 Diagram zapínání/vypínání úsporného režimu	31
Obrázek 13 Diagram způsobu výměny časových značek	33
Obrázek 14 Rozložení použitých adres paměti EEPROM u masteru	34
Obrázek 15 Rozložení použitých adres paměti EEPROM u slavu.....	34
Obrázek 16 ESP32 na kterém probíhalo měření	35
Obrázek 17 Oscilogram měření času přenosu jednoho paketu.....	35
Obrázek 18 Oscilogram měření času.....	36
Obrázek 19 Oscilogram měření napětí v nejvyšším bodě	36
Obrázek 20 Oscilogram měření času od přenosu po konec poklesu napětí	37
Obrázek 21 Oscilogram měření napětí po přenosu	37
Obrázek 22 Oscilogram měření napětí mezi přenosy.....	38
Obrázek 23 Oscilogram měření času mezi přenosy	38
Obrázek 24 Aplikace k přehlednému zobrazení dat.....	40
Obrázek 25 Zapojení ESP32 s periferiemi (DHT22, LED dioda, 3x tlačítko).....	40

Seznam tabulek

Tabulka 1 Formát rámce [16]	19
Tabulka 2 Specifický obsah [16]	19
Tabulka 3 Spotřeba energie pro přenos a příjem [13]	20
Tabulka 4 Spotřeba energie v úsporných režimech [13]	21

Úvod

Zadáním této práce je praktická implementace senzorické sítě s důrazem na zabezpečení, řešení problematiky přidání a odebrání uzlů sítě za chodu a předávání dat mezi uzly s nepřímou viditelností. Práce si klade za cíl vytvořit síť s co možná největším polem využití. Síť bude vytvořena tak, aby vyžadovala co nejmenší zásahy obsluhy a vydržela co nejdéle na externím zdroji elektrické energie. Na energetickou náročnost se bude klást důraz při budování sítě.

Práce je koncipovaná tak, že nejdříve proběhne rešerše dostupných řešení bezdrátových sítí od více výrobců a firem. Jednotlivá řešení jsou porovnávána mezi sebou a vyzdvihnuty jsou jak jejich klady, tak jejich zápory. Klady se zapracují do sítě popsané v této práci. Součástí této kapitoly je také popis, jakým způsobem zmiňované technologie pracují a jak navazují komunikaci.

V teoretickém návrhu se popisují některé problémy, které je nutné vyřešit ještě před praktickým návrhem sítě. Také jsou zde vybrány moduly schopné vytvořit bezdrátovou síť a je popsáno, jakými technologiemi přenosu tyto moduly disponují a jaké jsou jejich vlastnosti. Z dostupných technologií přenosu se zvolí ta nejvhodnější a budou podrobněji popsány její vlastnosti. Kapitola se věnuje také způsobu zabezpečení sítě, očekávané spotřebě a možnosti jejího snížení.

V poslední praktické části návrhu senzorické sítě je realizován vzorek sítě s vybranými moduly. Také jsou zde popsány konkrétní funkce nezbytné pro správné fungování sítě pomocí zjednodušujících vývojových diagramů a zdrojových kódů. Dojde zde i k porovnání očekávané spotřeby s reálnou. Reálná energetická náročnost se změří až při úplném sestavení sítě.

1 ÚVOD DO PROBLEMATIKY

V této kapitole budou vysvětleny základní pojmy používající se při budování bezdrátové komunikační sítě. Také proběhne srovnání topologií sítí a zvolí se ta, která se jeví jako nejvhodnější pro stav sítě.

1.1 Co je senzorická síť

Předpokládáme, že senzorická síť je složena z několika senzorů, které získávají data v různých místech a posílají je na předem určené místo, kde se data dále zpracovávají pro snadnější analýzu, například ve formě grafu. Forma přenosu dat je buďto drátovou nebo bezdrátovou formou v závislosti na druhu, funkci a velikosti sítě. Drátová síť propojí jednotlivé senzory fyzicky, což zvyšuje spolehlivost dodání dat. Nevýhodou je složitost konstrukce. Bezdrátová síť používá pro propojení senzorů například Bluetooth nebo Wi-Fi. Jedním z výhod bezdrátové senzorické sítě je snadné přemístování jednotlivých senzorů. Komunikace senzorů může být buďto jednosměrná (senzor pouze odesílá data), nebo obousměrná (senzor je schopen přijímat i posílat data). Jednotlivá zařízení schopná přijímat a odesílat data se nazývají uzly.

1.2 Druhy sítí

Jak již bylo popsáno výše, jednotlivé uzly mohou komunikovat jednosměrně nebo obousměrně. Pokud je komunikace jednosměrná, jsou pevně dané role master a slave. Master získává data a zasílá je do slavu, který informace dále zpracovává. Jeden master může mít hned několik slavů. Jejich počet je pouze limitovaný použitou technologií. Další možností je mít více masterů, které zasílají informace do jedno slavu. Ten se pak označuje jako multislave. Jako nejvhodnější se jeví obousměrná komunikace mezi uzly.

1.2.1 Sběrnice (bush)

Používá se primárně pro kabelové sítě. V této síti je každý uzel připojen v sérii na hlavní vedení. Dojde-li k poruše na hlavním vedením, může se více uzlů ocitnout bez přístupu k síti. Tato topologie tedy není vhodná pro účely této práce.

1.2.2 Hvězda (star)

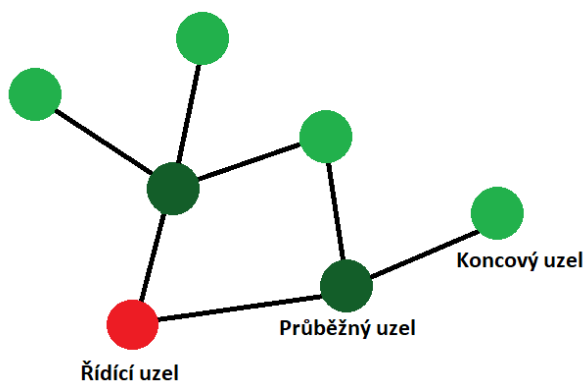
V tomto druhu sítě je každý uzel spojen s každým pomocí centrálního bodu (hub, switch). To má tu nevýhodu, že pokud selže centrální bod (například z důvodu výpadku elektřiny), dojde ke kolapsu celé sítě. Naopak, dojde-li k výpadku jednoho uzlu či k poruše jeho datové cesty, ostatní zařízení to nijak neovlivní. Výhodou je také snadné přidávání nového zařízení do systému. Máme-li více hvězdicových sítí, lze jejich centrální body propojit a tím vznikne uskupení nazývaní se rozšířená hvězda. Díky těmto vlastnostem se jedná o nejpoužívanější způsob propojování většího počtu počítačů. Tato topologie se také nejvíce jeví jako vhodná pro implementaci do sítě popsané v této práci.

1.2.3 Kruh (ring)

V této síti jsou uzly zapojeny v sérii postupně za sebou. Následně dojde k propojení koncových zařízení, čímž se vytvoří kruh. Konstrukce je jednodušší, než je tomu u hvězdicové sítě. Na druhou stranu, dojde-li k poruše jednoho uzlu nebo jedné přístupové cesty, celý systém se stane nefunkčním. Také pokud chceme přidat nový uzel, musíme kruh na nějaký čas přerušit.

1.2.4 Mesh síť

Při obousměrné komunikaci nejsou role pevně dané. Každé zařízení je zároveň master i slave a je tedy schopno informace přijímat i posílat. To je základ pro vytvoření mesh sítě. Tato síť má tu vlastnost, že není nutné mít napřímo propojené zařízení, ze kterého informace vychází, a cílové zařízení, do kterého informace směřují. Jinými slovy mohou data putovat přes několik zařízení na místo jejich určení. Konstrukce této sítě má velikou výhodu v tom, že jednotlivé uzly lze umísťovat do větší vzdálenosti, než je schopno cílové zařízení dosáhnout v případě bezdrátového přenosu. Další výhodou je, že pokud dojde k absenci jednoho nebo více uzlů, je síť dál schopna provozu.



Obrázek 1 Topologie mesh sítě

Z výše popsaných topologií se jeví mesh jako nejvhodnější pro bezdrátovou síť realizovanou v této práci. Před její implementací je vhodné ověřit její vlastnosti a funkčnost analýzou dostupných řešení.

2 Obvyklá řešení a dostupné technologie

V současné době je na trhu větší množství komerčně dostupných řešení. Např. firma Texas Instruments vyrábí a nabízí k prodeji bezdrátovou sadu Launchpad SensorTag kit [1]. Tato sada je založena na vývojovém kitu SimpleLink CC1352R od stejného výrobce. Je určena především jako vývojová platforma, kdy se při návrhu nových produktů pro IoT nemusí vyrábět hardware a software. Software stačí případně pouze upravit. Sada disponuje například technologiemi Bluetooth či ZigBee, což je technologie vytvořena podle mezinárodního standardu IEEE 802.15.4 a umožňuje vytvářet malé osobní sítě PAN podobně jako Bluetooth. Díky protokolu ZigBee lze vytvořit topologii typu sítě mesh. Používá se například pro automatizaci průmyslového sektoru a u aplikací, na které není použití Bluetooth vhodné. Tento produkt pracuje ve skupině bezlicenčních pásem pod 1 GHz, ale také v 2,4 GHz a jeho velkou výhodou je, že lze tyto pásma přepínat v průběhu přenosu. Obsahuje čtyři zabudované senzory okolního světla, tepla/vlhkosti, hallovu sondu a akcelerometr. Jako další výhodu výrobce udává nízkou energetickou náročnost. V aktivním režimu udává spotřebu energie 2,89 mA a v nejúspornějším režimu standby pak 0,85 μA [2]. Mezi další výhody pak patří ochranný kryt, externí anténa a aplikace do mobilního zařízení sloužící k zobrazení přijatých dat. Hlavními nevýhodami této bezdrátové sady je dostupnost výrobku a také cena, která se pohybuje okolo 1000 Kč za jedno zařízení.



Obrázek 2 Launchpad SensorTag CC1352R [1]

Dalšími produkty, které mohou vytvořit senzoričnou síť, je produkční řada od firmy Sonoff [3]. Na rozdíl od výše popsané vývojové sady jsou tyto produkty určeny pro běžného spotřebitele, který si chce doma pomocí vzájemně komunikujících čidel vytvořit jednoduchou síť chytré domácnosti. Zařízení splňují standard pro bezdrátovou komunikaci WiFi IEEE 802.11 b/g/n 2,4 GHz a také technologii Zigbee, popsanou v předešlém odstavci, která je základním kamenem ekosystému výrobce, a je také topologií typu mesh. V síti má každé zařízení svoji roli. Koordinátor zajišťuje řízení sítě, funguje jako přemostění mezi sítěmi WiFi a Zigbee a je schopen komunikovat se všemi zařízeními v jeho dosahu. Pro roli koordinátora výrobce nabízí zařízení jménem Sonoff ZigBee Bridge, které přijatá data může pomocí WiFi zobrazit v mobilní aplikaci. Jeho nevýhoda je, že potřebuje síťové napájení. Další uzel může být v roli podřízeného zařízení. Může komunikovat se všemi v dosahu a dokáže zastoupit funkci koordinátora. Jako poslední jsou jednotlivé koncové senzory, například senzory kouře, teploty, vlhkosti, pohybu nebo zjištění stavů oken a dveří. Výhodou tohoto

uzavřeného ekosystému je snadné použití a přehledné zobrazení přijatých dat. Mezi nevýhody pak patří pořizovací cena jednotlivých složek sítě, nutnost připojení k internetu, nemožnost připojit zařízení jiného výrobce a malý počet typů senzorů.



Obrázek 3 Sonoff ZigBee Bridge [20]

Produkty schopné vytvořit bezdrátovou síť nabízí i společnost Eve [4]. Stejně jako u předchozího ekosystému je i tento zaměřen na vytvoření do jisté míry automatického a informačního systému v domácnostech a není určen jako platforma pro vývojáře. Tyto zařízení používají ke komunikaci technologie WiFi, Bluetooth, ale především také síťový protokol Thread [5] vytvořený stejnojmennou společností. Stejně jako ZigBee je založen na standardu IEEE 802.15.4, vytváří bezdrátovou topologii sítě mesh s nízkou spotřebou energie a je určen pro produkty IoT. Stejně jako protokol ZigBee využívá šifrování AES 128. Na rozdíl od něj však podporuje komunikační protokol IPv6. Ačkoli se jedná o otevřený standard, zařízení jednotlivých výrobců spolu nemusí být kompatibilní. Firma Eve nabízí produkty například pro zabezpečení, osvětlení, nebo hlídání spotřeby energie a vody. Přijaté informace mohou nahrávat na internet přes síť WiFi a dokáží spolupracovat s aplikacemi jiných výrobců. Hlavní nevýhodou je nedostupnost na území Evropy.



Obrázek 4 Eve Motion [4]

Bluetooth mesh [6] je další technologie vhodná pro vytvoření osobní senzorické sítě. Vychází z bezdrátové sítě typu PAN Bluetooth Low Energy. Jak už název napovídá, topologie sítě je typu mesh. To znamená, že každé zařízení se připojí na ostatní uzly, které jsou v dosahu a všem jim pošle data. Zprávy jsou šifrovány symetrickým klíčem a poté je klíč distribuován pomocí protokolu ECDH, kdy tento protokol zabraňuje odposlouchávání komunikace pomocí útoku MITM. Díky konstrukci spolu mohou komunikovat stovky až

tisíce jednotlivých uzlů. Na rozdíl od ZigBee není vhodné pro průmyslové použití, ale je primárně určeno pro domácí aplikaci, jako je automatické osvětlení a tak dále. Stejně jako jiné generace Bluetooth pracuje na frekvenci 1,4 GHz. Oproti jiným protokolům má menší dosah a používá jednodušší mechanismus posílání zprávy [7]. Bluetooth mesh používá mnoho mobilních zařízení či produkty patřící do tzv. chytré domácnosti. Dále jsou také dostupné vývojové kity, například NRF52840 [8] od výrobce Nordic Semiconductor podporující mimo jiné Thread i ZigBee.



Obrázek 5 NRF52840 Dongle [8]

Další bezdrátový komunikační protokol schopný vytvářet osobní síť PAN nese název Z-Wave [9]. Stejně jako ostatní protokoly je primárně určen pro bezdrátové propojení elektronických zařízení disponujících touto technologií. Není určen pro průmyslové použití. Pracuje v pásmu v rozsahu 800 až 900MHz podle lokace použití, například v Evropě je to 868,4 MHz. Je zkonstruován pro přenášení menšího množství dat oproti ZigBee. Data přenáší i menší rychlostí, výrobce udává rychlost 100 kilobitů za vteřinu. Díky tomu je vhodnější spíše na posílání povelů typu zhasni/rozsviť. Šifrování je shodné se systémem Thread, používá tedy AES 128. Shodná je i topologie sítě a to mesh. Oproti ostatním protokolům vyžaduje méně elektrické energie a pracuje na jiné frekvenci, což zabrání rušení s ostatními systémy jako je třeba WiFi nebo Bluetooth. Zařízení podporující Z-Wave protokol je například SLWSTK6050B [10] určený pro vývojáře tohoto systému. Je schopno fungovat jako vstupní i koncové zařízení. Pro usnadnění práce obsahuje i vzorový kód.



Obrázek 6 SLWSTK6050B [10]

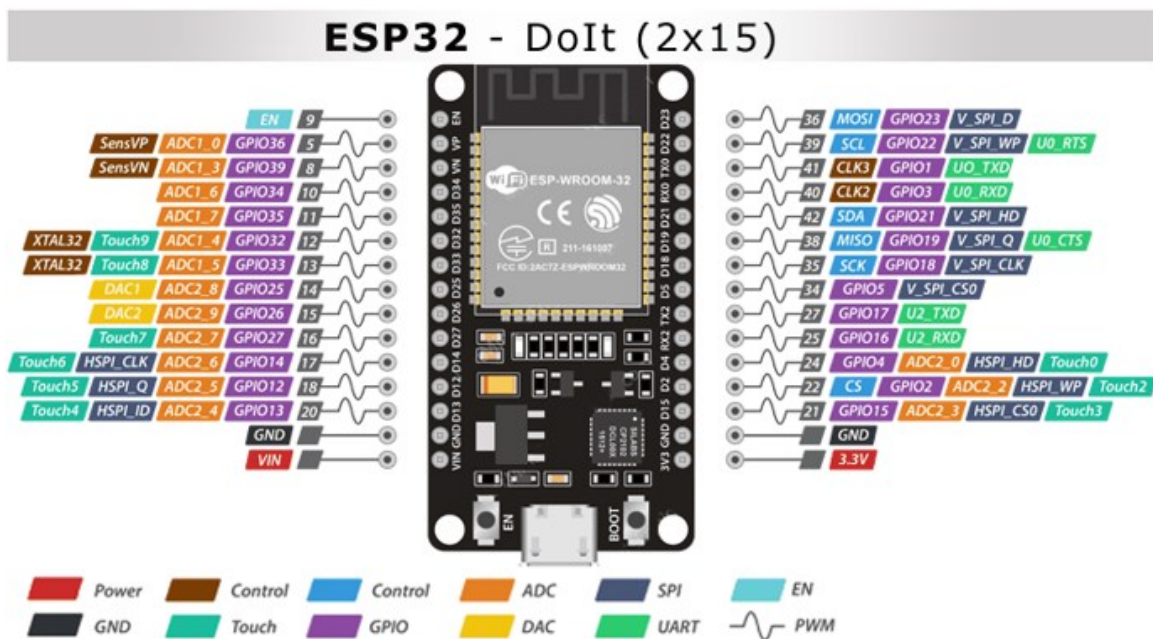
Dalším protokolem pro komunikaci se zařízeními chytré domácnosti a platformami IoT je Matter [11]. Na rozdíl od předešlých se nejedná o zcela nový protokol, ale je vytvořen na základech systému Thread a WiFi jako aplikační vrstva. Matter si neklade za cíl vytvořit náhradu za již existující platformy, které spolu nedokáží komunikovat, ale snaží se je „sjednotit“, aby bylo možné všechny zařízení ovládat pomocí jednoho ovladače. Jelikož Matter vyvíjejí velké technologické společnosti (Google, Apple, Amazon atd), jako ovladač

může sloužit spoustu produktů od různých výrobců. To je velká výhoda i pro vývojáře. Místo toho aby museli vyvíjet a vyrábět moduly pro různé komunikační platformy, bude jim stačit pouze jeden modul podporující Matter. Další výhodou je, že některá již existující zařízení lze upgradovat, aby byly schopny podporovat Matter, ale záleží na typu výrobce. Výrobci si také sami mohou zvolit způsob zabezpečení sítě s ohledem na povahu jejich produktů. Systém Matter byl představen teprve před několika měsíci, proto ještě neexistuje mnoho produktů, které ho podporují. Stejně tak o něm neexistuje mnoho informací, co se funkčnosti a spolehlivosti týče.

3 Návrh sítě

Z dostupných technologií určených pro tvorbu bezdrátové komunikační sítě je patrné, že nejvhodnější a zdaleka nejpoužívanější je topologie sítě typu mesh. Dále je potřeba zvolit, jakým stylem se budou distribuovat odesílaná data skrz síť, tedy způsob, jakým se má odesílaný paket dostat do místa určení. Existují dva základní druhy distribuce, záplavový algoritmus (flooding) a směrování (routing). Některé protokoly podporují oba způsoby, například ZigBee. Směrování doručí paket co nejkratší možnou cestou, paket tedy přijmou a odešlou jen některé uzly v síti. Opakem je zaplavování, kdy se paket odešle na všechny sousední uzly. To představuje riziko opětovného šíření jednoho paketu skrz síť. Jedna zpráva tedy může přijít několikrát, čemuž se musí zabránit. Výhodou je jednoduchá implementace a spolehlivost. Díky těmto vlastnostem byla zvolena právě tato metoda. Většina systémů používá pro zobrazení přijatých dat mobilní aplikace, které samozřejmě vyžadují přístup k internetu. Jelikož síť popsaná v této práci by měla být co nejvšestranější a mělo by ji jít vytvořit v jakémkoli prostředí, berme nutnost přístupu k internetové síti jako mínus.

Pro potřeby zadání bude v této práci realizována mesh síť a budou popsány praktické dopady realizace na modulech ESP32. Dále bude zvolena metoda bezdrátového přenosu pomocí vývojové platformy ESP32 vytvořené firmou Espressif Systems. Tento vývojový kit byl zvolen z důvodu jednoduché vývojářské manipulace, ceny, dostupnosti a nízké energetické náročnosti. Pro bezdrátovou komunikaci obsahují moduly ESP32 Wifi podle standardu 802.11 b/g/n v pásmu 2,4 GHz a dále pak Bluetooth. Pro řadu ESP byl vyvinut bezdrátový komunikační protokol ESP-NOW. Technologie přenosu budou popsány v další kapitole.



3.1 Technologie přenosu

3.1.1 Bluetooth

ESP32 Verze Bluetooth se liší dle verze modulu (např. ESP32-S2 Bluetooth neobsahují, ESP32-C3 mají v5.0) [12]. Pracuje na stejné frekvenci jako WiFi. Bluetooth u ESP32 je nevhodnější pro komunikaci mezi dvěma zařízeními. Například v odesílání dat z ESP na mobilní aplikaci. Výrobce udává vysílací výkon +9 dBm [13]. Pokud chceme komunikaci více zařízení bez klienta, je řešení daleko složitější. Výhoda je poměrně snadné zobrazení dat například v mobilním zařízení.

3.1.2 Wifi

Pro bezdrátovou komunikaci moduly ESP32 obsahují Wifi podle standardu 802.11 b/g/n v pásmu 2,4 GHz [12]. Podporuje také BSS, to znamená, že každé zařízení může být nastaveno buď jako přístupový bod (access point), režim stanice (station mode), nebo jako obojí najednou. V roli klienta lze ESP32 připojit do již existující sítě. V roli přístupového bodu se může připojit k ESP32 jakékoli zařízení podporující Wifi pomocí metody SoftAP. Lze tedy pomocí ESP32 vytvořit svoji vlastní Wifi síť. Velikou nevýhodou pro naše použití je nutnost internetového připojení na všech bodech v síti. Jednotlivá zařízení by tedy nekomunikovaly přímo spolu, ale přes internet. To by značně omezovalo použití senzorické sítě z hlediska přístupu k internetu na tak velické ploše.

3.1.3 ESP-NOW

ESP-NOW [14] je komunikační protokol vyvinutý společností Espressif, která stojí i za výrobou vývojářské desky ESP32. ESP-NOW je tedy dostupné pouze na některých zařízeních z produktové řady ESP. Je založen na standardu IEEE 802.11n, to znamená, že obsahuje zabezpečovací protokol CCMP. Komunikace může probíhat i nezabezpečenou formou. Také je možné si vybrat, s jakými spárovanými zařízeními budou komunikace probíhat šifrovanou formou, a se kterými nikoli [15]. Přenosová cesta je na frekvenci 2,4 GHz stejně jako u WiFi a Bluetooth. Toto pásmo je tedy poměrně vytížené. Do budoucna plánuje firma Espressif vyrábět kity schopné podporovat pásmo 5 GHz, avšak žádné konkrétní datum stanoveno není.

Komunikace je typu peer-to-peer, to znamená, že spolu dokážou komunikovat jednotlivé uzly a nepotřebují k tomu centrální nadřazený prvek skrz který by komunikace probíhala. Existuje zde i možnost zpětného volání stavu přenosu paketu. Tato technologie přenosu má i svá omezení, a to například nemožnost posílání zašifrované broadcast zprávy. U šifrovaného paketu je podporován pouze unicast. U nešifrované komunikace je podporován broadcast i unicast. Maximální velikost přenášených užitečných dat je 250 bajtů. Asi největší limitaci představuje omezení spárovaných zařízení (peerů). Maximální počet spárovaných zařízení je 20. Maximální počet zašifrovaných peerů je 10 [15]. Tento protokol je tedy pro naši síť zcela dostačující, a to díky výše uvedeným vlastnostem. Rozhodující je hlavně vývojářská jednoduchost, rychlost a podpora zašifrování.

3.1.3.1 Formát rámce ESP-NOW

ESP-NOW využívá pro přenos dat na míru upravený rámec:

Tabulka 1 Formát rámce [16]

Název	Velikost (bajty)	Definice
MAC záhlaví	24	Definuje MAC formát rámce
Kód kategorie	1	Nastaven na 127, indikuje kategorii vendor-specific
Identifikátor organizace	3	První tři bajty MAC adresy (unikátní)
Náhodná hodnota	4	Zabraňuje "štafetovému útoku"
FCS	4	Sekvence kontroly rámce
Specifický obsah	7~255	Tabulka 2

Tabulka 2 Specifický obsah [16]

Název	Velikost (bajty)	Definice
ID prvku	1	Nastaven na 221, indikuje vendor-specific element
Délka	1	Celková délka organizace, typu, verze a dat
Identifikátor organizace	3	První tři bajty MAC adresy (unikátní)
Typ	1	Nastaven na 4, indikuje ESP-NOW
Verze	1	verze ESP-NOW
Tělo	0~250	Uživatelská data

3.2 Zabezpečení

Jak již bylo zmíněno, ESP-NOW podporuje protokol pro bezdrátovou komunikaci CCMP ze standardu IEEE 802.11. CCMP využívá šifrovací algoritmus AES 128, který pomocí primárního klíče (PMK) o velikosti 128 bitů zašifruje rámec tím, že zamění byty, prohodí řádky, zkombinují se sloupce atd. Zašifrovaná zpráva je posléze zaslána na spárované zařízení, u kterého musí být zapnuta šifrovaná komunikace. K dešifrování v místě určení je pak využit druhý lokální klíč (LMK) o stejné velikosti jako primární klíč. V rámci standardu WPA2 je šifrovací protokol CCMP s AES bezpečnější než TKIP s RC4, na druhou stranu vyžaduje vyšší výpočetní výkon. V praktické implementaci bude zapotřebí vytvořit mechanismus předávání klíčů mezi jednotlivými klienty sítě.

3.3 Očekávaná spotřeba a možnosti jejího snížení

Ačkoli je spotřeba jádra ESP32 velmi nízká, při využití všech periférií může spotřeba dosáhnout až cca 240 mA [17]. To představuje problém při napájení z baterie, kdy provoz zařízení by byl omezen jen na pár hodin. Energetickou náročnost lze tedy snížit několika způsoby:

- Rozumným využitím (např. není třeba snímat teplotu 100x za 1 s)
- Navazování spojení Bluetooth/WiFi trvá dlouho, vyplatí se tedy poslat co nejvíce dat najednou
- Deaktivovat nepoužívané periferie
- Využit ULP koprocesoru k řízení jednoduchých činností
- Přechod do některého z úsporných režimů [17]

Rozumné využití může být například frekvence snímání teploty ze senzoru. Snímat teplotu dvakrát za vteřinu je nerozumné, pokud se data zobrazují jednou za vteřinu. Dále je potřeba zvolit rozumný poměr času, kdy je ESP v aktivním stavu, tedy kdy má nejvyšší spotřebu energie, a kdy je ESP v jednom z úsporných režimů. Je tedy žádoucí, aby aktivní režim byl co nejkratší. Délka aktivního režimu závisí hlavně na rychlosti probuzení ESP a dále pak na rychlosti, jakým je schopno data odeslat a přijmout. Doba trvání úsporného režimu závisí pouze na tom, jak často chceme dostávat informace z ostatních uzlů. Maximální doba usnutí v podstatě není omezena, ESP může spát i několik dní.

3.3.1 Úsporné režimy

- Active mode – V aktivním režimu je vše zapnuté, včetně všech periférií.

Tabulka 3 Spotřeba energie pro přenos a příjem [13]

Mode	Min	Typ	Max	Unit
Transmit 802.11b, DSSS 1 Mbps, POUT = +19.5 dBm	-	240	-	mA
Transmit 802.11g, OFDM 54 Mbps, POUT = +16 dBm	-	190	-	mA
Transmit 802.11n, OFDM MCS7, POUT = +14 dBm	-	180	-	mA
Receive 802.11b/g/n	-	95 ~ 100	-	mA
Transmit BT/BLE, POUT = 0 dBm	-	130	-	mA
Receive BT/BLE	-	95 ~ 100	-	mA

- Modem-sleep mode – V tomto režimu dojde k vypnutí komunikačních periférií, WiFi, Bluetooth a rádia.
- Light-sleep mode – Na rozdíl od režimu modem-sleep je v tomto režimu navíc pozastaven hodinový signál, obsah paměti a registru je zachován.
- Deep-sleep mode – Pouze ULP koprocesor, RTC paměť a časovač jsou zapnuty. Obsah RAM a registru je ztracen. Pro uložení některých dat lze využít RTC SRAM. Probuzení lze provést resetem, RTC časovačem, nebo externím přerušením, stejně jako v režimu Light-sleep.
- Hibernation mode – Vnitřní 8 MHz oscilátor, stejně jako ULP koprocesor, jsou vypnuty. Pouze časovač RTC a některé GPIO piny jsou funkční a schopny probudit zařízení z úsporného módu.



Obrázek 8 Úsporné režimy ESP32 [22]

Tabulka 4 Spotřeba energie v úsporných režimech [13]

Power mode	Description		Power consumption	
Modem-sleep	The CPU is powered on.	240 MHz *	Dual-core chip(s)	30 mA ~ 68 mA
			Single-core chip(s)	N/A
		160 MHz *	Dual-core chip(s)	27 mA ~ 44 mA
			Single-core chip(s)	27 mA ~ 34 mA
		Normal speed: 80 MHz	Dual-core chip(s)	20 mA ~ 31 mA
Single-core chip(s)	20 mA ~ 25 mA			
Light-sleep	-		0.8 mA	
Deep-sleep	The ULP coprocessor is powered on.		150 μ A	
	ULP sensor-monitored pattern		100 μ A @1% duty	
	RTC timer + RTC memory		10 μ A	
Hibernation	RTC timer only		5 μ A	
Power off	CHIP_PU is set to low level, the chip is powered off.		1 μ A	

Zdroje probuzení:

- Čítač (Watchdog)
- Kapacitní touch senzor
- Externí přerušení
- Sériový port

Jelikož je žádoucí dosáhnout co možná nejmenší spotřeby energie a tím dosáhnout co nejdélejší životnosti baterie, která bude napájena z jednoho uzlu sítě, je zapotřebí zvolit vhodný úsporný režim, který bude aktivní v době, kdy spolu nebudou komunikovat jednotlivé uzly. Režimy modem-sleep a light-sleep mají oproti režimu deep-sleep výrazně vyšší spotřebu, ale obsah RAM zůstane zachován. Existuje i způsob, jak uchovat data i v režimu deep sleep, a to tak, že se nahrají do EEPROM (512 B), která je emulována v NOR-FLASH paměti, lze i použít paměť RTC o velikosti 8 KB. EEPROM je nevolatilní paměť, to znamená, že po odpojení napájení zůstanou data zachována. Z důvodu velmi malé energetické náročnosti bude pro tvorbu senzorické sítě popsané v této práci zvolen režim deep sleep. K synchronizovanému probuzení všech klientů bude využit časovač.

3.4 Časová synchronizace

Aplikovat řízení spotřeby na celou síť je poměrně náročné. Kvůli úspoře elektrické energie bude použit úsporný režim, jak bylo popsáno v předešlé kapitole. Během tohoto režimu spolu jednotlivé uzly v síti nemohou komunikovat, proto je nutné, aby jednotlivá zařízení byla uspána na stejně dlouhou dobu. Probuzení provede RTC časovač, který je i během režimu deep sleep aktivní. RTC hodiny mají pět možných zdrojů: XTL_CLK, XTL32K_CLK, RC_FAST_CLK, RC_FAST_DIV_CLK a RC_SLOW_CLK. Jako defaultní zdroj hodinového signálu pro RTC časovač je RC_SLOW_CLK. „Vyznačuje se nejnižší spotřebou během úsporného režimu deep sleep a není závislý na žádných externích součástkách, frekvence zdroje je však ovlivněna teplotou [18].“ Stejně jako u CPU hodin lze vybrat „rychlé“ zdroje, nebo pomalejší. Dále je nutné, aby jednotlivá ESP byla uspaná ve stejný okamžik. Je tedy potřeba „sjednotit“ čas na všech zařízeních.

Jelikož se režim uspaní a aktivní režim budou periodicky opakovat, znamená to komplikace při sestavování sítě. Pokud bude nějaký uzel vyhledávat ostatní zařízení, mohou být některá zařízení v režimu spánku, tím pádem nedojde k navázání komunikace a bezdrátová síť nebude spolehlivě fungovat. Při sestavování sítě (rozmisťování senzorů) je tedy žádoucí, aby byly v aktivním režimu a řízení spotřeby se zapnulo, až když jsou všechny klienti sítě na svém místě.

3.4.1 Synchronizace času přes NTP

NTP se využívá pro synchronizaci hodin na všech zařízeních v síti. Pro naše použití by tedy byl ideální, kdyby nevyžadoval internetové připojení. Jak již bylo napsáno v kapitole technologie přenosu, je vhodnější vyvarovat se nutnosti přístupu k internetu z hlediska použitelnosti senzorické sítě.

Bude tedy třeba synchronizovat časy na jednotlivých uzlech pomocí námi použitého komunikačního protokolu.

3.4.2 Millis

ESP nabízí možnost získat čas po jak dlouhou dobu je zařízení zapnuto. Tento čas jde pouze číst, nelze ho nijak editovat. Millis by tedy šlo využít pro časovou synchronizaci pouze za předpokladu, že by se všechna ESP v síti zapnula ve stejný čas. To v praktickém využití nedává smysl.

3.4.3 Použití knihoven pro synchronizaci a editaci času

Další možností je využít knihovny pro zobrazení a editaci času, schopné pracovat v režimu bez NTP serveru. Tuto možnost nabízí knihovna „time.h“ [19]. Teorie je tedy taková, že na všech klientech v síti se spustí časovač, který se vynuluje při vstupu do sleep módu. Pokud se některý uzel probudí později nebo dříve je zapotřebí jednotlivé časové obvody znovu synchronizovat s dostatečnou přesností, neboť jednotlivé časovače na zařízeních nemusí mít stejnou hodnotu.

Pro synchronizaci času budou použity knihovny `time.h` a z ní vycházející knihovna „DS1307RTC.h“ [19]. Jako zdroj časového signálu knihovny využívají RTC časovač. Jednotlivé uzly se budou přizpůsobovat Masteru neboli zařízení, které řídí komunikaci. Synchronizace proběhne tak, že se stav jednotlivých časovačů bude posílat přes komunikační protokol. Master odešle informace o stavu svého časovače na spárovaná zařízení, stavy časovačů se následně porovnají a přizpůsobí se podle Masteru. Zařízení, která porovnaly svůj stav s Mastrem obsahují informaci, že mají aktualizován stav svého časovače (mají tedy tzv. „MasterTime“). Tyto klienti následně porovnávají svůj stav s dalšími zařízeními, které se zase přizpůsobí jim.

3.5 Přidávání a odebrání uzlů

Aby jednotlivé ESP navázaly komunikaci přes protokol ESP-NOW, potřebují získat svoje MAC adresy. Jelikož nechceme najít všechny adresy, které se vyskytují v okolí, ale chceme pouze adresy těch zařízení, které spadají do námi budované senzorické sítě, je zapotřebí tento stav ošetřit. To lze provést nastavením SSID. Do SSID se kromě MAC adresy místního zařízení přidá „předpona“. Každý klient v síti bude mít stejnou předponu, tím je zajištěno spárování námi vybraných zařízení.

Poté, co dojde ke spárování klientů sítě, komunikace ještě nebude probíhat tak, jak má, jelikož komunikace je zabezpečená. Každý uzel musí mít klíče k zašifrování a odšifrování paketů. V našem případě klíči disponuje pouze master a ostatním klientům síť musí být odeslán od masteru. Slave tedy musí zažádat o zaslání klíčů. Je tedy zapotřebí zvolit algoritmus zaslání klíče do zařízení, které si o něj požádá. Jako uživatelsky nejprívětivější se nabízí zmačknutí dvou tlačítek. Jedno tlačítko se stiskne na ESP, které žádá o klíč a druhé na Masteru, kterému zmačknutí tlačítka umožní odeslání klíče na příslušné ESP32. Po obdržení klíče již může komunikace pobíhat tak, jak má. Jelikož dojde ke ztrátě dat při režimu `deep sleep`, bylo by nutné klíče posílat každý úsporný cyklus, což je samozřejmě velice neefektivní. Klíče se tedy musí po obdržení neprodleně zapsat do paměti EEPROM, která zůstane zachována i po zapnutí úsporného režimu.

Odpojení ESP od sítě se může řešit obdobně přes hardwarové tlačítko. Toto řešení bude uživatelsky přívětivější. Také je vhodné realizovat odpojení klienta pomocí řídicího prvku sítě (master). Bude tedy stačit udělat opak, než je popsáno v odstavci pro připojení tlačítka do sítě. Pokud tedy zařízení obsahuje klíč uložený v paměti EEPROM, jednoduše se paměť vymaže/přepíše. Komunikace bez klíče není možná. Také je vhodné odstranit již spárovaná zařízení, aby se zamezilo posílání nešifrovaných paketů do bezdrátové sítě.

4 Praktická část návrhu

4.1 Architektura sítě

Každý uzel sítě komunikuje obousměrně. Klienti v síti mají dvě role. V síti je přítomen jeden master a více slave zařízení. Master slouží k řízení komunikace a synchronizaci času. Může odesílat zprávy obsahující instrukce určené konkrétnímu klientovi. Master také obsahuje klíče k šifrované komunikaci, o které si slave musí před vstupem do komunikace zažádat. Uživatel sítě si pomocí něho může zobrazit příchozí data z rozmístěných senzorů. Primární funkcí uzlů v roli slave je získat a odeslat data získaných z připojených senzorů. Také distribuuje přijatá data sousedních uzlů podle topologie typu mesh. I na slavu lze zobrazit získaná data.

4.2 Nastavení přístupového bodu

Jako komunikační protokol je použito ESP-NOW. Jelikož je primárně určen na komunikaci stacionárních zařízení, adresy jednotlivých zařízení je potřeba znát ještě před zahájením komunikace. Z uživatelského hlediska je velice nepohodlné a časově náročné zjišťovat MAC adresy všech klientů sítě. Je tedy zapotřebí tento problém odstranit a to tak, že jednotlivá ESP32 dokážou přečíst MAC adresy klientů v jejich dosahu. Abychom tohoto docílili, musíme nejdříve nastavit přístupový bod a inicializovat ESP-NOW.

ESP-NOW využívá ke své funkčnosti WiFi. Nejprve je zapotřebí nastavit WiFi mód. Za předpokladu, že víme MAC adresu zařízení, se kterým se lokální ESP chce spárovat, stačilo by zařízení dát do módu stanice, a to i při obousměrné komunikaci. V tomto případě ho ale nevíme, takže musíme všechny klienty sítě nastavit do kombinovaného modu WIFI_AP_STA, tedy jako stanici a zároveň přístupový bod. K nastavení WiFi módu je samozřejmě nutná knihovna WiFi.h. Dále do indikátoru bezdrátové sítě SSID nahrajeme unikátní MAC adresu lokálního ESP a také předponu, v tomto případě pojmenovanou "MySlave:", která slouží tomu, aby ESP vyhledalo pouze indikátory s tímto indexem. Zamezí se tím získání identifikátoru klienta cizí sítě. Uvedená operace obsahuje funkce s názvem SSID_AP_STA.

```
void SSID_AP_STA() {  
  WiFi.mode(WIFI_AP_STA);           //nastavení AP a STA  
  String MAC = WiFi.macAddress();    //MAC adresa  
  String Predpona = "MySlave: ";     //předpona  
  String SSID = Predpona + MAC;      //nahraní do SSID  
}
```

4.3 Inicializace ESP-NOW

Samozřejmě musíme inicializovat samotný komunikační protokol. Od výrobce Espressif je doporučeno to udělat až po nastavení Wifi. Nejdůležitější je zde funkce `esp_now_init()`, pokud se inicializace z nějakého důvodu nepodaří, celá funkce se opakuje.

```

void ESPnow() {
  if (esp_now_init() == ESP_OK) {
    Serial.println("ESP-NOW Init");
  }
  else {
    Serial.println("ESP-NOW chyba");
    ESPnow();
  }
}

```

Dále se musí nastavit primární klíč sloužící k zašifrování odesílaných paketů a také zpětné volání k zobrazení stavu odeslané zprávy a zpětné volání k příchozí zprávě od jiného ESP s protokolem ESP-NOW.

```

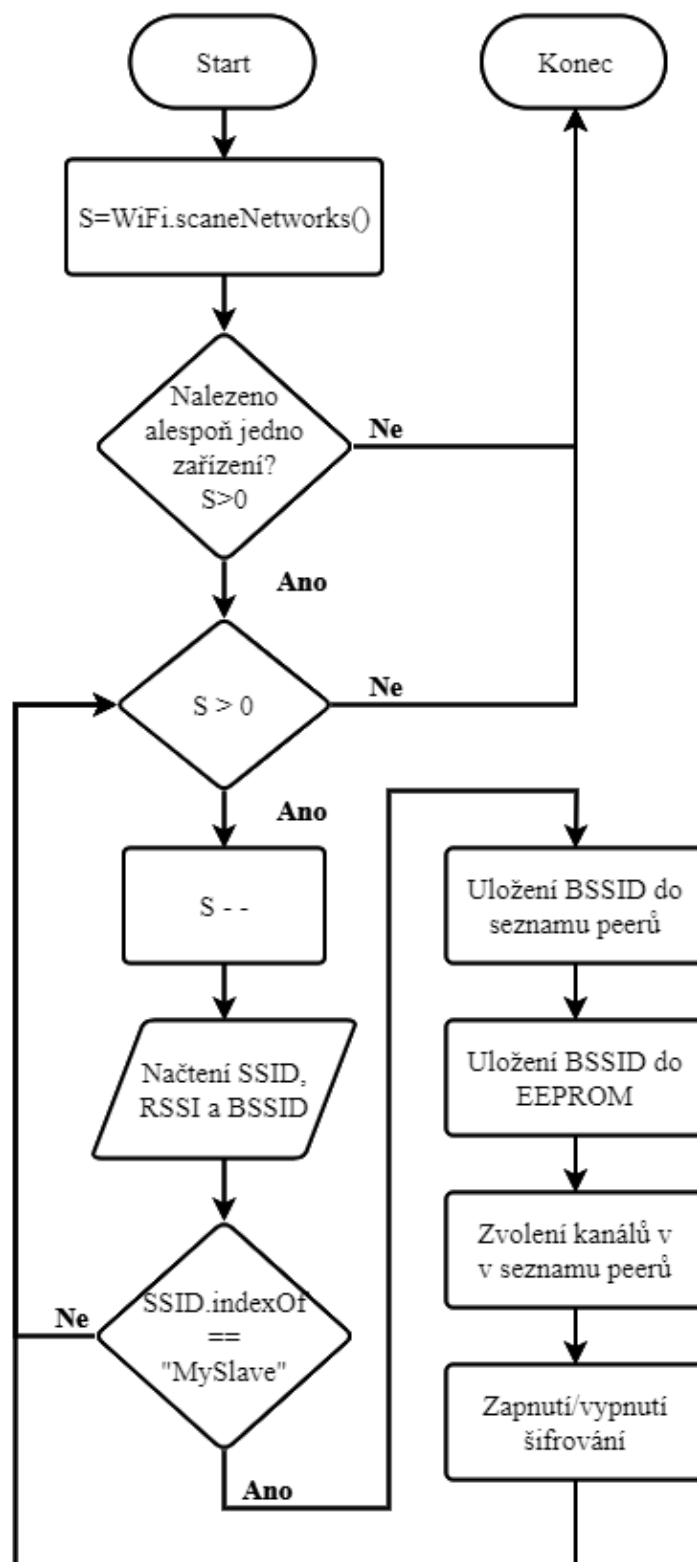
esp_now_set_pmk((uint8_t *)PMK_KEY_STR); // Nastavení PMK klíče
esp_now_register_send_cb(sendState); // zpětné volání stavu odeslané
zprávy
esp_now_register_recv_cb(recv); // zpětné volání příchozí zprávy

```

4.4 Vyhledání klientů v síti

Nyní je již nastaveno vše potřebné k vyhledání adres potřebných ke spárování. K vyhledání ostatních zařízení slouží funkce *ScanForSlave* [20]. Funkce nejprve najde dostupné WiFi sítě pomocí *WiFi.scanNetworks*. Tato funkce najde pouze ta zařízení, která jsou nastavena jako přístupový bod. Najdou-li se nějaké sítě, tak se z každé uloží indikátor SSID, síla příchozího signálu RSSI (slouží pouze pro zobrazení jako zajímavost) a také MAC adresa obsažena v BSSID.

Jak již bylo napsáno, chceme spárovat pouze ty adresy, které mají námi zvolený index (MySlave). To ověří funkce *SSID.indexof*. Pokud takovou přeponu obsahují, lze konečně získat MAC adresu z *BSSIDstr* pomocí funkce *k* přečtení *scanf*. Nyní se adresy uloží do seznamu spárovaných zařízení a do paměti EEPROM. Dále se u každé adresy zvolí komunikační kanál. Použití více kanálu je zapotřebí, když se u zařízení používá ESP-NOW a WiFi současně. Toto se této práci netýká, takže jednoduše se všechny adresy nastaví na stejnou úroveň 0. Už zbývá pouze zvolit, zda se bude jednat o šifrovanou nebo nešifrovanou komunikaci a zapsat počet spárovaných adres do paměti. To bude potřeba při režimu šetřící spotřebu elektrické energie. Nyní, když byly získány adresy, lze spárovat nalezené sousední klienty. To zajišťuje funkce *AddPeer()*, která periodicky ověřuje stav jednotlivých peerů. Funkce vyhledávání zařízení se může spustit třemi způsoby. Buď zmáčknutím příslušného tlačítka, nebo v případě špatného uložení MAC adresy (uloží-li se samé nuly) a také při určitém probuzení ESP32 z úsporného režimu (například se spustí každé páté probuzení).



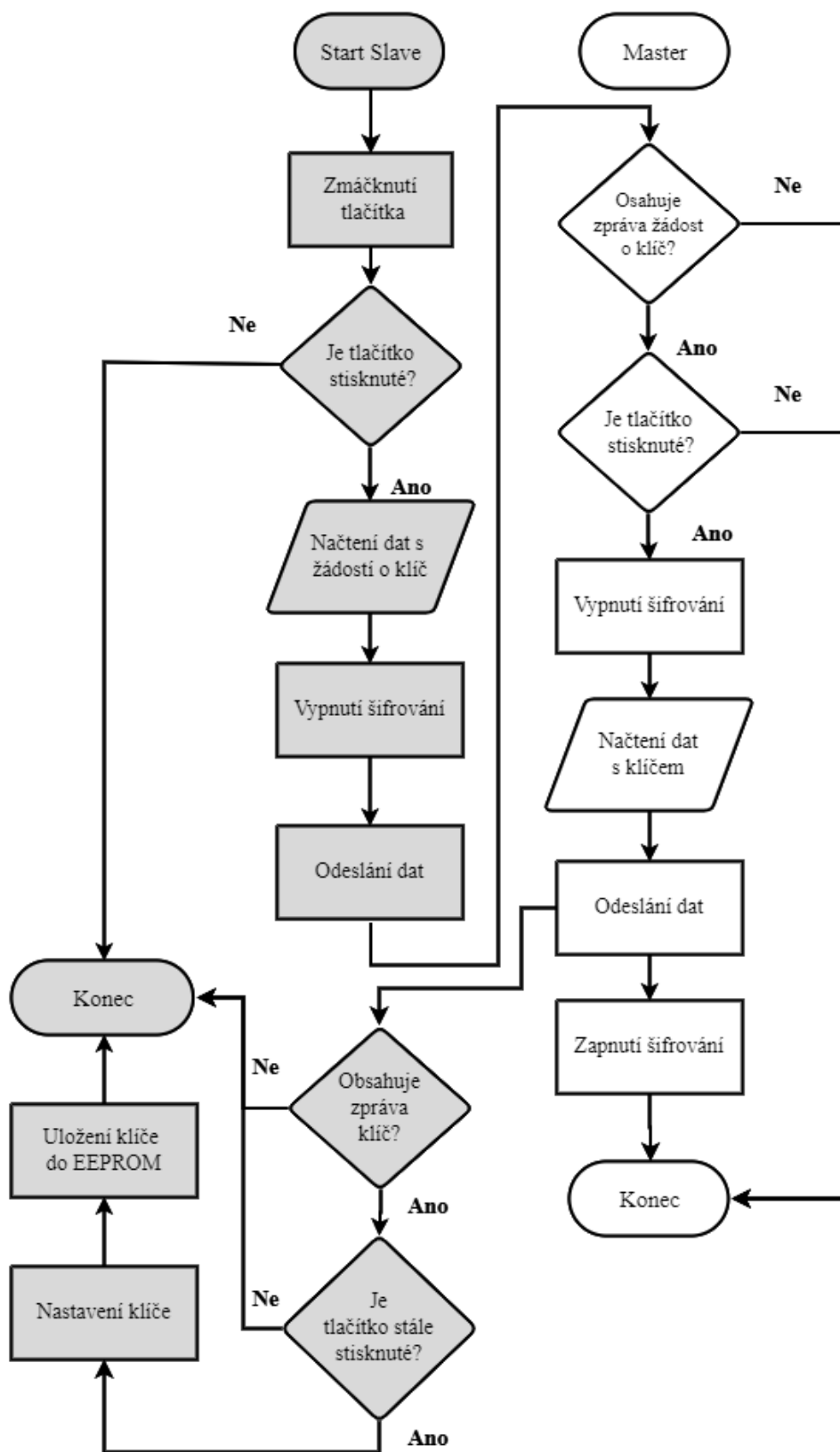
Obrázek 9 Diagram funkce pro najití okolních zařízení pomocí WiFi

4.5 Připojení klienta do sítě

Ačkoli lokální ESP32 disponuje MAC adresami sousedních zařízení, komunikace ještě proběhnout nemůže z důvodu zapnutého šifrování. Klíči potřebnými k šifrované komunikaci disponuje pouze ESP označené jako master. Master tedy může hned ze začátku komunikace nastavit primární klíč k šifrování odesílaných paketů. Způsob nastavení je popsán v předešlé kapitole. Lokální klíč sloužící k dešifrování příchozích paketů je nastaven před spárováním adres sousedních klientů. Zařízení označené jako slave ze začátku nemá k dispozici ani jeden klíč. Klíče potřebné ke komunikaci tedy musí získat z masteru.

Aby byl spuštěn mechanismus odeslání klíčů, musí být stisknuta tlačítka na obou ESP. Ve chvíli, kdy je na slavu stisknuto tlačítko, se vypne šifrovaná komunikace. Pokud je šifrování vypnuto, nelze posílat běžné zprávy s daty ze senzorů, lze odeslat pouze žádost o klíče. U Masteru dojde k vypnutí šifrování až ve chvíli, kdy je doručena zpráva s žádostí o klíče. Tato zpráva obsahuje MAC adresu klienta, který o klíče žádá, a také potvrzení toho, že je vypnuté šifrování. Následně master načte a odešle zprávu obsahující klíče a také adresu klienta, který o ně zažádal. Adresa ve zprávě zamezuje tomu, aby klíče neuložilo zařízení, kterému není určeno.

Po odeslání zprávy master obnoví šifrovanou komunikaci. Poté, co slave obdrží zprávu od masteru, dojde k ověření, že zpráva skutečně obsahuje klíče, o které zažádal. Následně dojde k nastavení klíčů a jejich zapsání do EEPROM, aby nebyly ztraceny během úsporného režimu. Nyní již komunikace může probíhat šifrovanou formou.

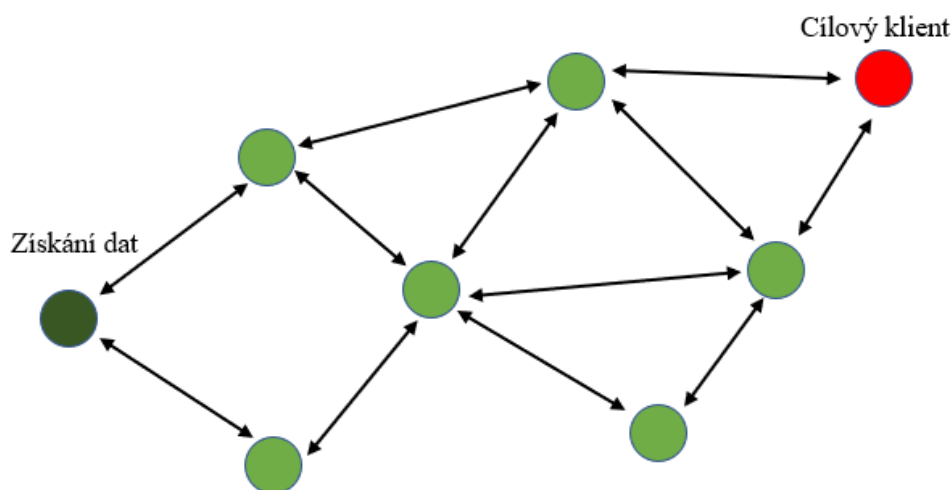


Obrázek 10 Diagram způsobu distribuce šifrovacích klíčů

4.6 Způsob distribuce paketů skrz síť

Jako způsob šíření zprávy byl zvolen tzv. záplavový algoritmus (flooding algorithm) kvůli jednoduchosti jeho implementace do sítě. Funguje tak, že místní zařízení odešle paket na všechny klienty, které jsou v jeho okolí, a ty následně udělají to samé. Jeden paket s daty se odešle na všechny klienty sítě. Tento způsob distribuce přináší jeden problém. Jedna konkrétní zpráva se může šířit po síti opakovaně a došlo by k její duplicitě. Tuto situaci lze vyřešit poměrně jednoduše a to tak, že se do každé zprávy přidá unikátní znak nebo číslo. Použití náhodného čísla se jeví jako lepší možnost. Při odesílání zprávy každý klient vygeneruje pseudonáhodné číslo a vloží ho do odesílaného paketu. Při příjmu zprávy toto číslo přečte a uloží ho do volatelné paměti. Tím pádem získá přehled o tom, zda zprávu již dostal či nikoli. Pokud ano, zprávu klient ignoruje, to znamená, že ji nezobrazí a ani ji dále neodesílá do sítě.

Každá zpráva v síti má také omezený počet přeskoků. Do zprávy se vloží maximální počet klientů sítě. Při průchodu každým klientem se číslo zmenší o jedna. Pokud číslo dosáhne nuly, celá zpráva zanikne podobně, jako tomu je u položky TTL v IPv4.



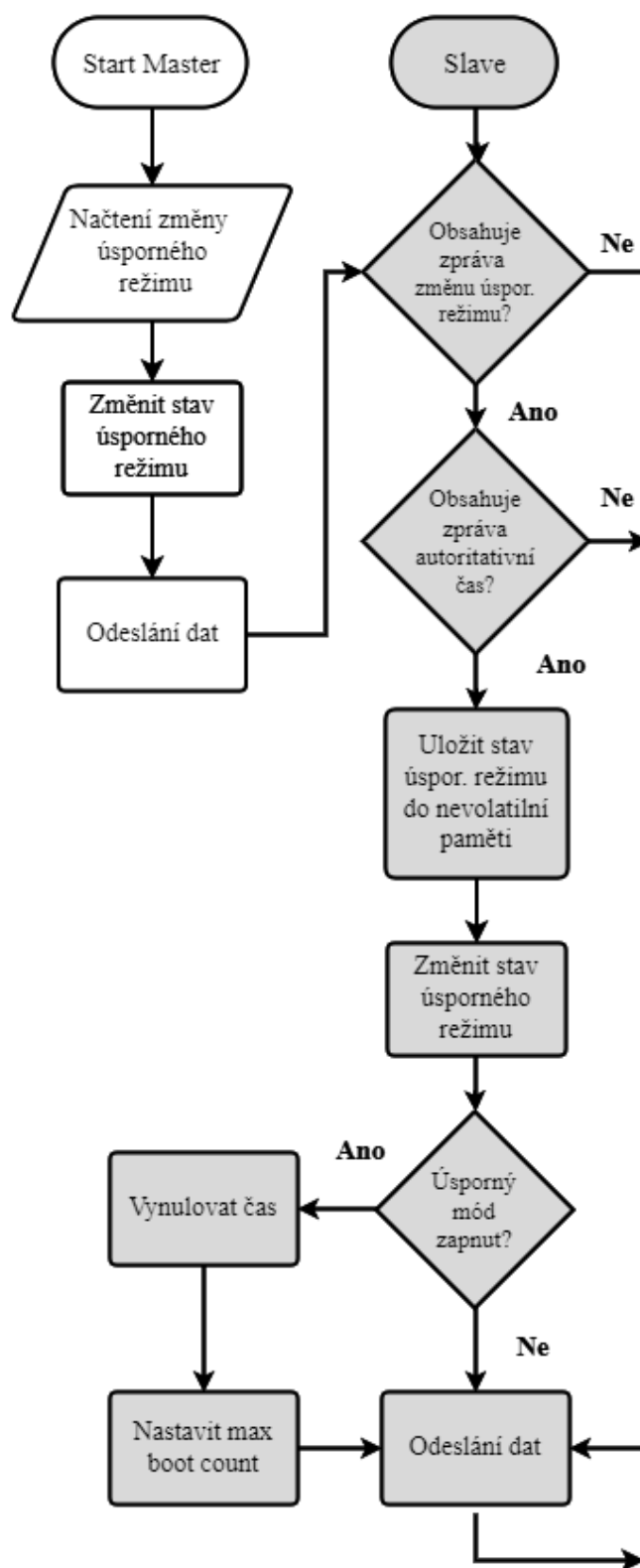
Obrázek 11 Způsob distribuce paketu skrz síť

4.7 Zapnutí a vypnutí úsporného módu

Úsporným módem se myslí to, že zařízení bude periodicky přecházet do režimu deep sleep. Je-li úsporný mód vypnutý, ESP32 zůstává v aktivním módu a nepřechází do deep sleep. Vypnutí úsporného módu bylo zavedeno pro snazší sestavení sítě, kdy se rozmisťují jednotlivé zařízení a následně se na každém musí aktivovat funkce pro získání adres klientů v jejich dosahu. Pokud by někteří klienti byly uspani, jejich adresu by nebylo možné získat.

Uživatel provede změnu stavu úsporného režimu na ESP32, které je v roli master, a to pomocí zapsání písmena „S“ do sériového monitoru nebo pomocí ovládací aplikace. Master následně začne odesílat právy, kde je aktualizovaný stav úsporného režimu. Tato zpráva je následně distribuována po síti, jak bylo popsáno dříve. Zařízení typu slave také odesílají stav svého úsporného režimu, takže by mohlo dojít k tomu, že se na některých klientech budou opakovaně měnit stavy úsporného režimu. Tudíž je zapotřebí jednotlivým klientům dát informaci, že se jedná o aktuální stav z masteru. Master tedy současně se změnou stavu úsporného režimu odešle i informaci, že zpráva pochází od něho, a tím se předejde tomuto problému.

Po obdržení zprávy se zapnutým úsporným režimem se také vynuluje čas na příchozím ESP32, tím dojde k prvotní synchronizaci času v síti. Dále také dojde ke změně proměnné (boot count), která se postupně přičítá po zapnutí ESP32. Ta zde slouží k zapínání funkce pro vyhledání ostatních zařízení v dosahu. Funkce se například aktivuje každé páté zapnutí vývojového kitu. Proměnná se nastaví na maximální možné číslo, aby po zapnutí byla tato funkce aktivována. Proč se tato funkce zapíná jen někdy, je popsáno později.



Obrázek 12 Diagram zapínání/vypínání úsporného režimu

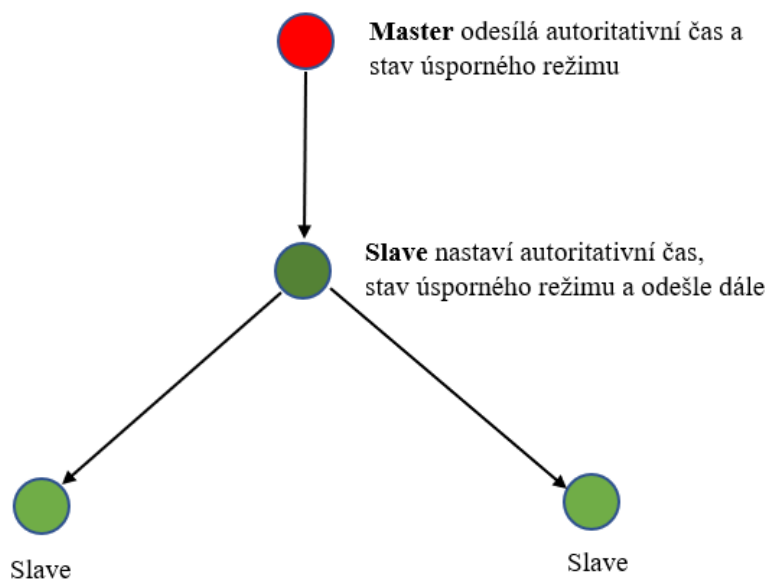
4.8 Časová synchronizace

Časová synchronizace se spustí hned, jakmile přijde zpráva o zapnutí úsporného režimu. V ten moment dojde k vynulování a spuštění časovače (místní čas). V aktivním režimu si zařízení posílají stavy jejich časovačů. Proces posílání časových značek je popsán v kapitole návrh sítě. Čas, který uplyne od odeslání zprávy z prvního zařízení do okamžiku přijetí na sousedním ESP32, se pohybuje v řádech desítek milisekund. Nejvyšší naměřená hodnota byla 53 ms. Jelikož technologie použita pro synchronizaci času pracuje v řádech jednotek vteřin, lze čas přenosu datového paketu zanedbat.

Jakmile na časovači uplyne předem stanovená doba (čas bdělosti), přejde do režimu deep sleep, ve kterém bude také předem stanovenou dobu. Důležité je, aby na všech klientech sítě byly tyto doby stejné, jinak by klienty nešlo synchronizovat. K probuzení z úsporného režimu je použit čítač (watchdog) s přesností na mikrosekundy. Čas, po kterou je zařízení uspano, se zadává ve vteřinách a není v návrhu SW omezen. Stačí si určit, jak často chceme dostávat data ze senzorů. U určení času, po kterou jsou klienti v aktivním režimu, je to trochu složitější, co se týče dolní hranice. ESP32 musí mít dostatek času na provedení všech nutných operací na odeslání zprávy.

Je tedy zapotřebí změřit, kolik času zmíněné operace potřebují. Měření bylo provedeno pomocí funkce *Millis*. Funkce *setup* trvá 150 až 160 ms a první zpráva se odešle 350 ms od zapnutí. Stejně je tomu u první přijaté zprávy, které odešlou sousední klienti sítě. Pokud námi sledované ESP32 nemá přímou viditelnost s některým z klientů sítě, zpráva z tohoto klienta putuje delší dobu. Putuje-li zpráva přes jednoho klienta, doba od odeslání po přijetí zprávy je cca 350 ms. Čas probuzení musí být zaokrouhlen na celé vteřiny, kvůli použité technologii pro synchronizaci času.

Funkce pro vyhledání ostatních zařízení trvá od 4800 ms do 5400 ms. Je-li tato funkce spuštěna, musí stav probuzení trvat minimálně cca 5,5 s. Používat funkci každé probuzení by bylo značně nevhodné. Z tohoto důvodu se MAC adresy z této funkce zapíší do EEPROM a při zapnutí zařízení se čtou z této paměti, to zabere mnohem méně času (50 ms). Jelikož může docházet k přesunu senzorů či k výpadku některého ze senzorů, vyplatí se čas od času vyhledat sousední zařízení. Jak často by se měla tato funkce zapnout není definováno časově, nýbrž počtem zapnutí ESP32. Například se funkce aktivuje každé páté probuzení.



Obrázek 13 Diagram způsobu výměny časových značek

4.9 Použití vláken a časovače watchdog

Použitý způsob synchronizace času nenabízí vyvolání přerušení po uplynutí stanovené doby. Aby byla synchronizace co nejpřesnější, je nutné, aby se ESP32 uspalo v co možná nejkratší době po uplynutí zmiňovaného času. Proto je vhodné použít více vláknové programování, které ESP podporuje. Vytvoří se dvě vlákna, z nichž každé poběží na jiném jádře procesoru (vytvořit více vláken lze i na jedno jádrových CPU, díky systému FreeRTOS). Jedno vlákno bude použito pro hlídání přetečení časovače spouštějící deep sleep. Toto vlákno se bude opakovat s co největší frekvencí. Druhé vlákno bude mít na starost odesílání paketu a obsluhu periférií.

Watchdog je použit pro probouzení z úsporného režimu deep sleep. Dále také hlídá jednotlivá vlákna proti zacyklení. V případě přetečení vyvolá čítač reset celého CPU.

4.10 Použití EEPROM

Kvůli maximální úspoře energie je použit úsporný režim deep sleep. Během tohoto režimu dojde ke ztrátě krátkodobé paměti, proto se pro úschovu důležitých dat musí použít nevolatilní paměť. Jak již bylo popsáno, do paměti EEPROM se uloží MAC adresy (zelená), aby se funkce ScanForSlave používala co nejméně z úsporných důvodů. Současně s nimi se také uloží počet najitých zařízení (modrá). Dále se ukládá stav úsporného režimu (žlutá), který určuje, zda se může aktivovat režim deep sleep či nikoli. Do paměti se také ukládá stav LED diody (oranžová) použité pro kontrolu komunikace mezi klienty sítě. U slavu se navíc ukládají klíče pro šifrovanou komunikaci (šedá a hnědá) a také další stav LED diody (tmavě oranžová) sloužící k ověření příjmu zprávy určené pouze pro jednoho klienta sítě.



Obrázek 14 Rozložení použitých adres paměti EEPROM u masteru



Obrázek 15 Rozložení použitých adres paměti EEPROM u slavu

4.11 Formát odesílaných zpráv

Při běžné komunikaci obsahují odeslané zprávy MAC adresu zařízení, ze kterého byla odeslána. Ačkoli komunikační protokol nabízí možnost zobrazit MAC adresu ze zařízení, ze kterého zpráva byla přijata, tato zpráva může putovat přes několik klientů a mohlo by dojít k záměně místa odeslání zprávy, proto je MAC adresa obsažena v uživatelských datech. Dále zpráva obsahuje, zda odesílatel komunikuje v šifrované formě, unikátní číslo (popsáno v kapitole způsob distribuce skrz síť), časovou značku, boot count, stav LED diody (signalizace) a stav úsporného režimu. Zpráva také obsahuje číslo indikující přes kolik klientů prošla. Při průchodu klientem se číslo odečte. Pokud je číslo rovno nule, zpráva zaniká. Odesílané zprávy samozřejmě obsahují data ze senzorů. V tomto případě je ke každému uzlu připojen senzor DHT11/22 snímající okolní teplotu. Pro připojení senzoru k ESP32 je nutná knihovna „DHT.h“ [21].

Pokud se z masteru odesílá zpráva určená pro nějakého klienta sítě, formát zprávy je trochu odlišný. Místo svojí MAC adresy master vloží do zprávy adresu místa určení. Aby to bylo možné, master si uloží každou novou příchozí adresu do seznamu. Z tohoto seznamu pak přečte požadovanou adresu. Ostatní zařízení následně porovnají svoji adresu s příchozí adresou. Pokud se shodují, klient ví, že zpráva je určena pro něho. Aby si tuto zprávu nespletl se svojí vlastní, která k němu může doputovat, má zpráva své specifické pseudonáhodné číslo, které je obsaženo v každé odeslané zprávě kromě té s údaji pro šifrovanou komunikaci.

Pokud slave žádá o klíče umožňující šifrovanou komunikaci, odešle pouze svoji adresu a potvrzení, že má vypnuté šifrování. Master následně odešle zprávu obsahující adresu zařízení, které žádost poslalo a šifrovací klíče. Podrobnější popis je popsán v kapitole připojení klienta do sítě.

4.12 Odpojení klientů ze sítě

K odpojení klienta ze sítě lze využít dva způsoby. První způsob je zmáčknutí tlačítka na klientovi, podobně jako tomu je u připojení do sítě. Po zmáčknutí dojde k vymazání (přepsání) paměti EEPROM. U zařízení v roli slave dojde tedy ke ztrátě klíčů umožňujících šifrovanou komunikaci. Dále dojde k vypnutí komunikačního protokolu ESP-NOW a je vymazán seznam spárovaných ESP32. Po opětovném stisknutí tlačítka se komunikační protokol znovu inicializuje a klient se může opětovně připojit do sítě, jak je tomu popsáno v kapitole o připojení klienta do sítě. Druhý způsob je odeslání příkazu k ukončení

komunikace z masteru. Klient po obdržení zprávy provede stejné operace, jako tomu je u zmáčknutí tlačítka. Při ztrátě nebo odcizení klienta je vhodné změnit šifrovací klíče. To ztracenému zařízení znemožní opětovný návrat do probíhající komunikace.

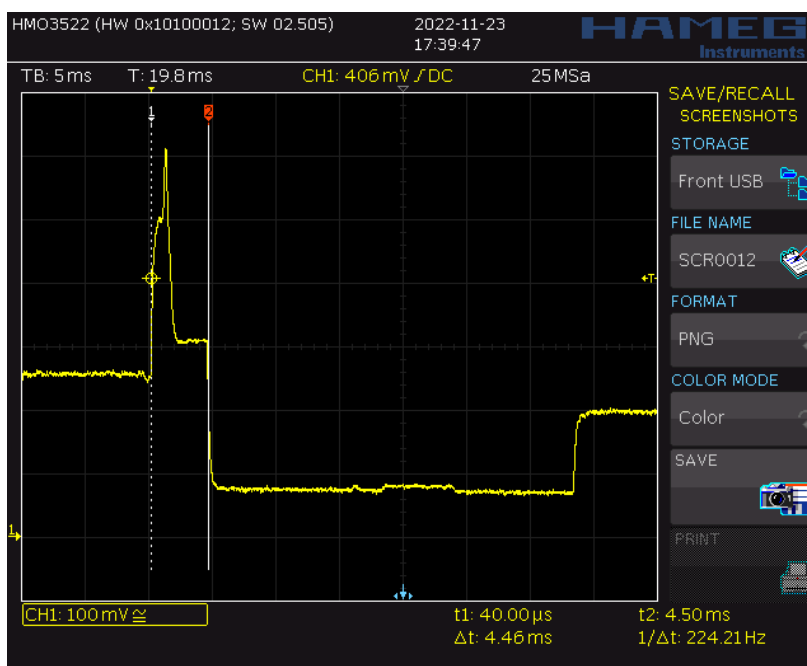
4.13 Porovnání očekávané spotřeby s reálnou

Očekávaná spotřeba byla popsána již dříve v kapitole návrhu. Důležité jsou hlavně tabulky 3 a 4 s hodnotami od výrobce čipů Espressif, se kterými budeme porovnávat naměřené hodnoty. Praktické měření bylo provedeno na upraveném vývojovém kitu ESP 32, na kterém bylo přerušeno napájení regulátoru napětí, aby bylo možné připojit bočník a k němu sondy osciloskopu. Měření bylo realizováno pomocí osciloskopu s bočníkem o velikosti $2,5 \Omega$.



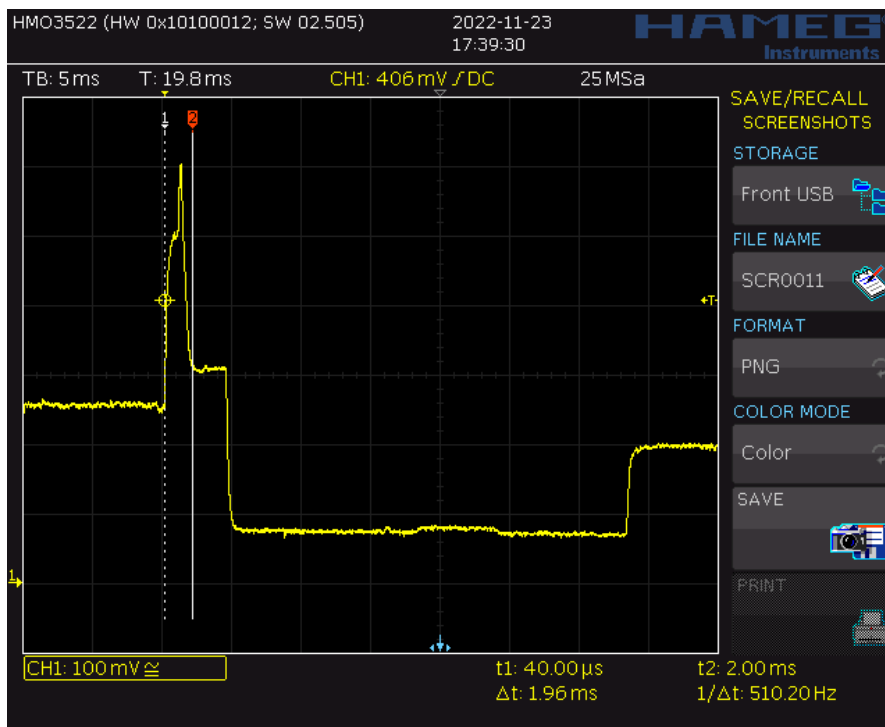
Obrázek 16 ESP32 na kterém probíhalo měření

Nejprve se změřila doba přenosu jednoho paketu. Z oscilogramu lze vidět, že doba od nárustu napětí po pokles trvá 4,46 ms.



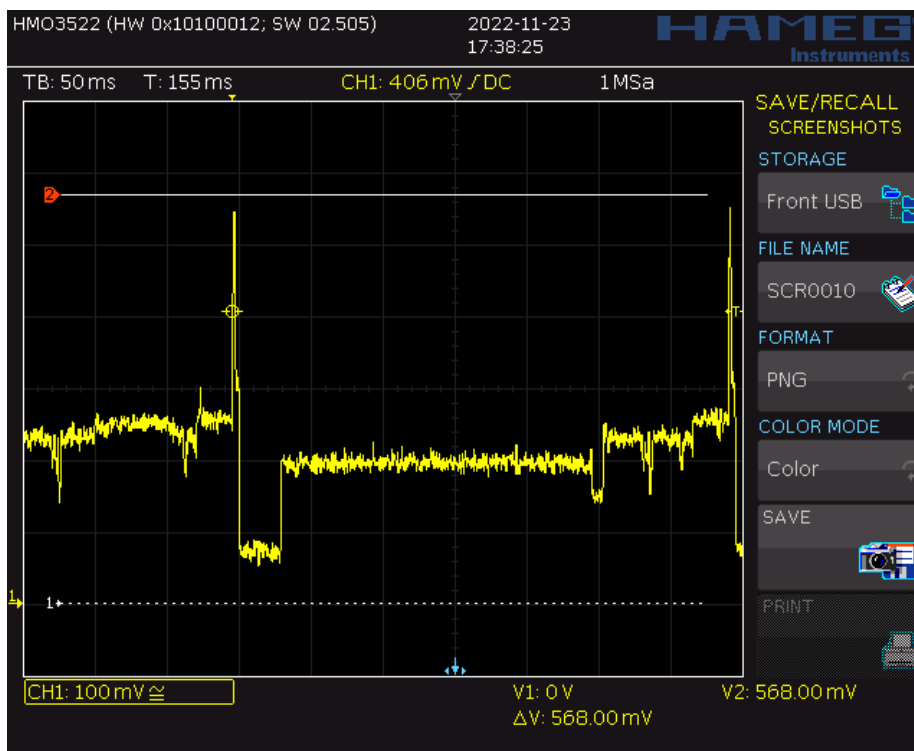
Obrázek 17 Oscilogram měření času přenosu jednoho paketu

Doba špičkového odběru proudu trvá 1,96 ms.



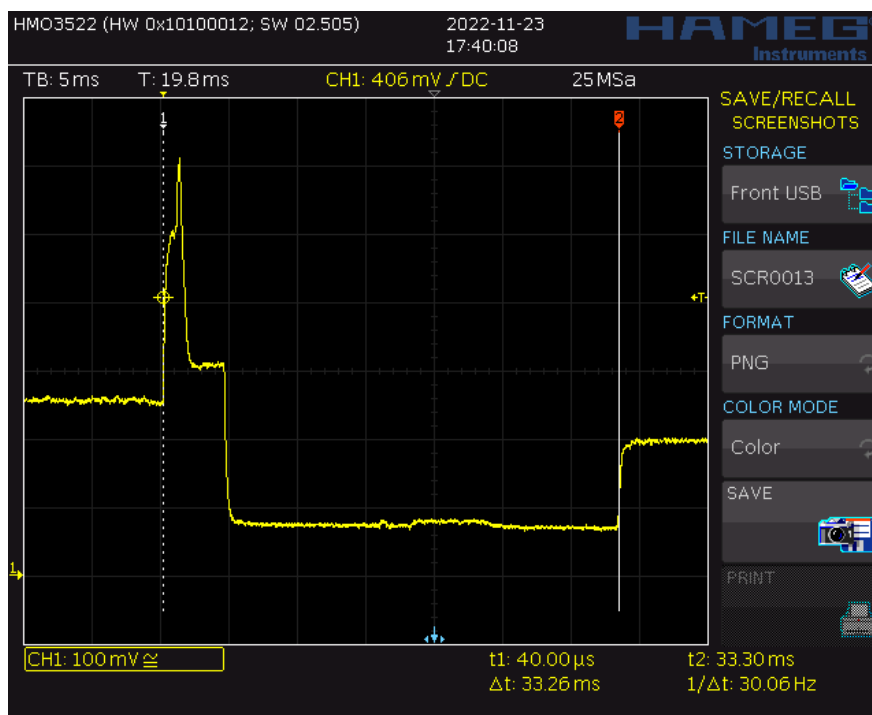
Obrázek 18 Oscilogram měření času

Podle Ohmova zákona prochází bočníkem proud 227,2 mA.



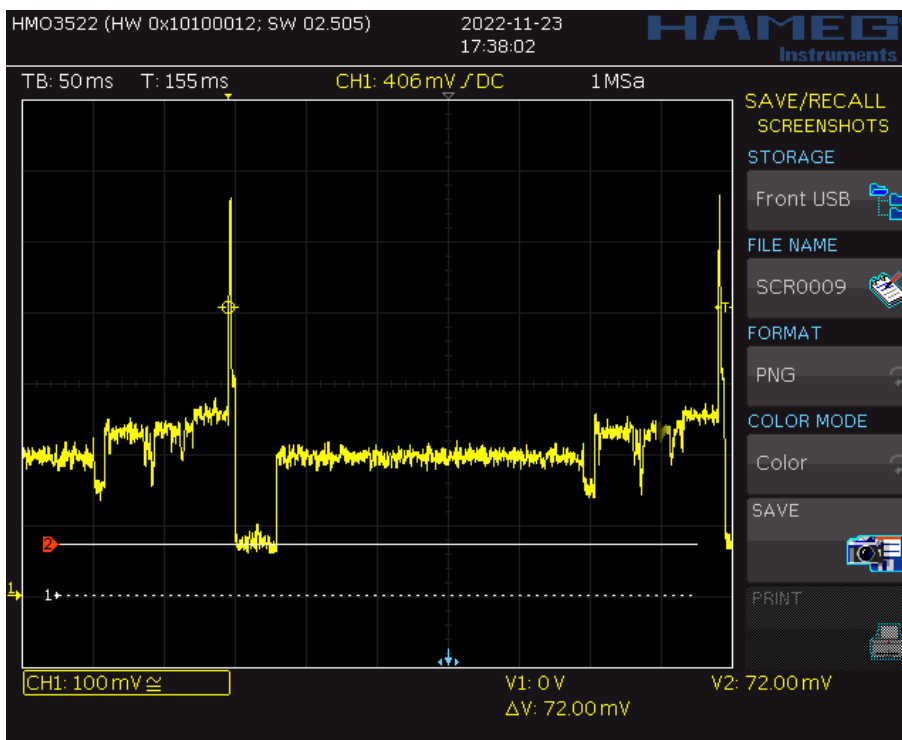
Obrázek 19 Oscilogram měření napětí v nejvyšším bodě

Po přenosu dojde k poklesu napětí. Následující oscilogram zobrazuje měření času od počátku přenosu po konec poklesu napětí.



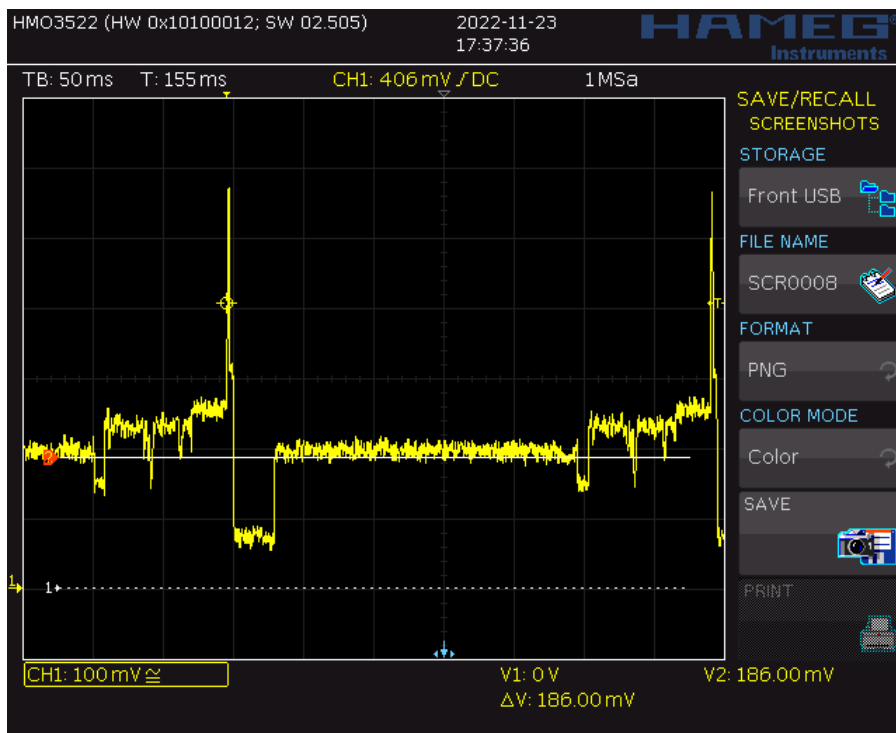
Obrázek 20 Oscilogram měření času od přenosu po konec poklesu napětí

V tomto útlumu dosahuje hodnota proudu 28,8 mA.



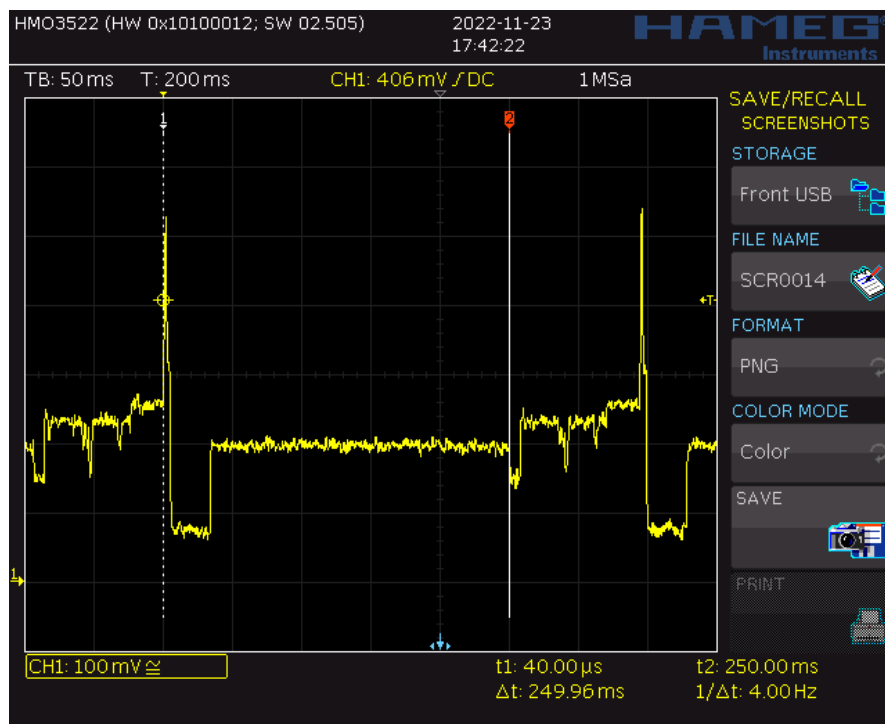
Obrázek 21 Oscilogram měření napětí po přenosu

Mimo vysílání je proud ustálen na hodnotě 74 mA.



Obrázek 22 Oscilogram měření napětí mezi přenosy

Dále je změřen čas od konce přenosu jednoho paketu po začátek přenosu dalšího.



Obrázek 23 Oscilogram měření času mezi přenosy

Při měření příkonové náročnosti v režimu deep sleep byla naměřena hodnota 0,1 mA. Tato naměřená hodnota je shodná s hodnotou, kterou udává výrobce pro zapnutý ULP koprocesor (tabulka 4). Při měření přenosu je nejvyšší naměřená hodnota 227,2 mA po dobu 1,96 ms. Udávaný proud výrobcem pro přenos se pohybuje v rozmezí od 180 mA do 240 mA. Mimo vysílání se proud pohybuje okolo 74 mA.

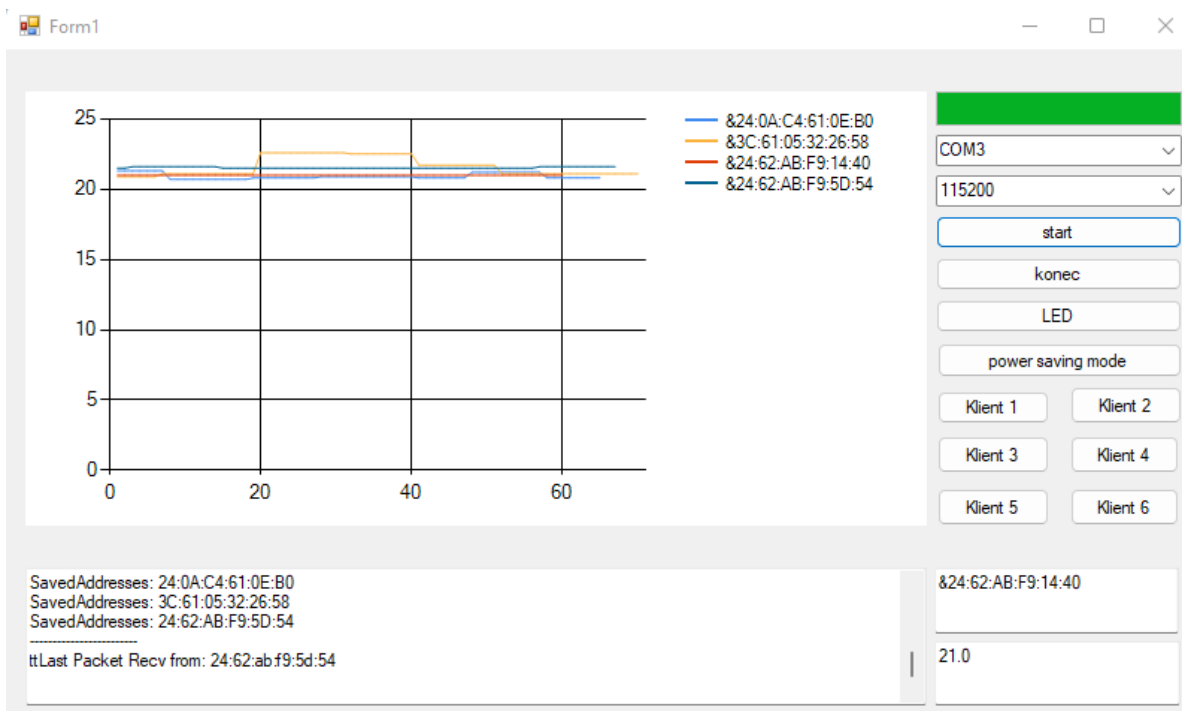
Střední hodnota proudu jedné periody je 42,369 mA. Hodnota byla vypočítaná z vygenerovaných dat osciloskopem (soubor TRC v příloze A). Zařízení jsou uspána na jednu minutu a aktivní jsou 2 s. Aktivní režim tedy trvá 3,22 % z jedno cyklu. Celková střední hodnota proudu je 1,463 mA. Například při použití bateriového napájení o kapacitě 3400 mAh zařízení vydrží v provozu cca 2 323 h tj. 96 dní bez periferií (LED diody, senzoru).

4.14 Zobrazení dat

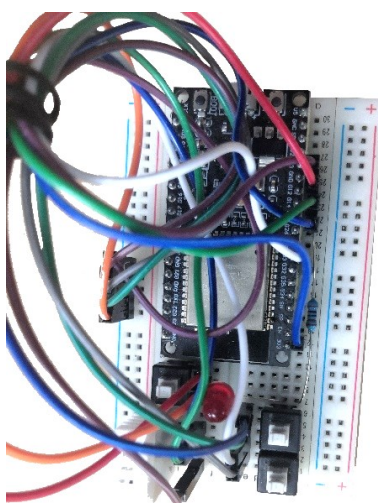
K zobrazení přijatých dat slouží aplikace vytvořená ve Windows Form. Data mohou být zobrazena na jakémkoli zařízení v síti. Aplikace ale obsahuje i tlačítka pro řízení komunikace a tyto povely je schopný zpracovat pouze master. V aplikaci si uživatel nejdříve zvolí, na kterém dostupném rozhraní sériového portu je připojeno ESP32. Následuje zvolení komunikační rychlosti. Po zmáčknutí tlačítka start aplikace začne zobrazovat data ze sériového monitoru. Pomocí znaku „&“ na začátku řádku aplikace pozná, která data jsou určena k zobrazení, vybere je a zobrazí v grafu. Není tedy nutné posílat do sériového monitoru jen data určená zobrazovací aplikací. Data se zobrazují v grafu, který se aktualizuje s každou přijatou zprávou určenou pro aplikaci (na obrázku je na ose Y teplota a na ose X je počet přijatých vzorků). Měřítko grafu se přizpůsobí podle přicházejících dat. Osy grafu nejsou popsány, aby nedošlo k mýlce v případě výměny senzorů. Aplikace automaticky přidá do grafu nově příchozí adresu, není tedy nutné ji zadávat manuálně. Napravo od grafu jsou vidět všechny adresy klientů sítě včetně masteru. Pod grafem je okno sloužící k zobrazení všeho, co je posláno do sériového monitoru. Vpravo dole se vypíše právě příchozí adresa a jaké data obsahuje.

Kromě zobrazení dat lze aplikací posílat příkazy do sítě skrz master. Po zmáčknutí některého z tlačítek aplikace odešle do sériového monitoru písmeno. Zařízení master poté příslušné písmeno přečte a spustí odpovídající operaci.

Tlačítko LED změní stavy všech diod na každém klientovi sítě. Power saving mode zapne úsporný režim v síti. Tlačítka klient 1 až 6 slouží k odpojení konkrétního klienta ze sítě, nebo mohou být případně použity i pro jinou funkci.



Obrázek 24 Aplikace k přehlednému zobrazení dat



Obrázek 25 Zapojení ESP32 s periferiemi (DHT22, LED dioda, 3x tlačítko)

Závěr

Cílem této práce bylo navrhnout a sestavit zabezpečenou senzorkou síť s ošetřením stavu přidání a odebrání komunikačních uzlů a s možností řízení spotřeby. Klienti spolu musí dokázat komunikovat i s nepřímou viditelností a komunikace musí být obousměrná. Každé zadané věci byl věnován prostor a bylo zdůvodněno, z jakého důvodu bylo použito právě popsané řešení.

Jako vývojová platforma pro tvorbu bezdrátové sítě se použilo ESP32. Tato platforma nabízí několik technologií pro bezdrátový přenos, ze které se vybrala ta nejvhodnější pro tuto práci. Topologie sítě je typu mesh, tím pádem klienti spolu dokážou komunikovat i s nepřímou viditelností mezi nimi. Všechny uzly sítě dokážou komunikovat obousměrně.

Dále byl popsán způsob snížení a řízení spotřeby. Kvůli tomu bylo nutné implementovat na všechny klienty sítě synchronizaci času umožňující probouzet a uspávat jednotlivé uzly ve stejný okamžik. Díky tomu mohl být použit nejúspornější režim, jaký ESP32 nabízí. Efektivitu toho režimu ověřilo kontrolní měření. Výsledky toho měření se porovnály s hodnotami od výrobce vývojového kitu. Kvůli použitému úspornému režimu bylo také nutné zapisovat data do nevolatilní paměti, aby nedošlo k jejich ztrátě.

Při tvorbě sítě se brala v úvahu její použitelnost v místech bez internetového připojení, proto je síť vytvořena tak, aby byla schopna fungovat v off-line režimu. Také byla věnována pozornost způsobu zabezpečení senzorké sítě a algoritmu předávání klíčů mezi jednotlivými uzly. Stav přidání a odebrání jednotlivých uzlů byly realizovány použitím mechanických tlačítek. Odpojit klienty ze sítě lze i na dálku pomocí uzlu v roli master. Pro zpřehlednění příchozích dat od jednotlivých klientů sítě se vytvořila aplikace vytvářející graf, který se aktualizuje s nově příchozími daty. Aplikace také vypíše MAC adresy všech uzlů v síti. Při tvorbě této práce byla pozornost zaměřena především na funkčnost a efektivitu bezdrátové sítě. V práci je možné pokračovat a potenciál je zejména v části obsahující řízení spotřeby. Zajímavá je rovněž problematika způsobu distribuce paketu přes síť, kde je prostor pro zdokonalení zabezpečení sítě, například omezením maximální doby existence datagramu. Všechny stanovené cíle byly splněny.

Literatuta

- [1] Texas Instruments: LPSTK-CC1352R. In: *Texas Instruments* [online]. [cit. 2022-10-31]. Dostupné z: <https://www.ti.com/tool/LPSTK-CC1352R#description>
- [2] *CC1352R SimpleLink™ High-Performance Multi-Band Wireless MCU: CC1352R* [online]. In: . s. 96 [cit. 2022-10-31]. Dostupné z: https://www.ti.com/lit/ds/symlink/cc1352r.pdf?ts=1667926371527&ref_url=https%253A%252F%252Fwww.google.com%252F
- [3] *Sonoff: SONOFF PRODUCTS* [online]. [cit. 2022-11-01]. Dostupné z: <https://sonoff.tech/products/>
- [4] *Evehome: eve* [online]. [cit. 2022-11-02]. Dostupné z: <https://www.evehome.com/en>
- [5] *Threadgroup: thread* [online]. [cit. 2022-11-02]. Dostupné z: <https://www.threadgroup.org/What-is-Thread/Thread-Benefits>
- [6] Mesh Networking: Mesh networking is blue. In: *Bluetooth* [online]. 2022 [cit. 2022-11-02]. Dostupné z: <https://www.bluetooth.com/learn-about-bluetooth/recent-enhancements/mesh/>
- [7] YADAV, Rahul. How to choose between Zigbee and BLE mesh for your IoT application?. *Linkedin* [online]. [cit. 2022-11-02]. Dostupné z: <https://www.linkedin.com/pulse/how-choose-between-zigbee-ble-mesh-your-iot-rahul-yadav>
- [8] *NRF52840 Dongle: Development dongle for Bluetooth 5/Bluetooth mesh/Thread/Zigbee/802.15.4/ANT/2.4GHz* [online]. In: . s. 2 [cit. 2022-11-02]. Dostupné z: <https://www.farnell.com/datasheets/3166596.pdf>
- [9] *Z-wave: Smart home control on one app* [online]. 2022 [cit. 2022-11-02]. Dostupné z: <https://www.z-wave.com/>
- [10] SLWSTK6050B: Z-Wave Long Range 700 Starter Kit. In: *Silabs* [online]. 2022 [cit. 2022-11-03]. Dostupné z: <https://www.silabs.com/development-tools/wireless/z-wave/z-wave-long-range-700-starter-kit?tab=techdocs>
- [11] Matter: The Foundation for Connected Things. In: *Csa-iot: Building the Foundation and Future of the IoT* [online]. 2022 [cit. 2022-11-03]. Dostupné z: <https://csa-iot.org/all-solutions/matter/>

- [12] DOBROVOLNÝ, Martin. *Aplikace mikroprocesorů 2: WiFi [přednáška]*. UPCE, 2021.
- [13] *ESP32 Series: Datasheet* [online]. In: . s. 69 [cit. 2022-11-09]. Dostupné z: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf
- [14] ESP-NOW. In: *Espressif* [online]. 2022 [cit. 2022-11-06]. Dostupné z: <https://www.espressif.com/en/products/software/esp-now/overview>
- [15] *ESP-NOW: User Guide* [online]. In: . s. 10 [cit. 2022-11-06]. Dostupné z: https://www.espressif.com/sites/default/files/documentation/esp-now_user_guide_en.pdf
- [16] ESP-NOW. In: *Espressif: ESP-IDF Programming Guide* [online]. 2022 [cit. 2022-11-06]. Dostupné z: https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/network/esp_now.html
- [17] DOBROVOLNÝ, Martin. *Aplikace mikroprocesorů 2: Úsporné režimy [přednáška]*. UPCE, 2021.
- [18] System Time. In: *Espressif: ESP-IDF Programming Guide* [online]. 2022 [cit. 2022-11-16]. Dostupné z: https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/system/system_time.html
- [19] MARGOLIS, Michael. *Arduino_time*. In: *Github* [online]. 2022 [cit. 2022-12-09]. Dostupné z: https://github.com/michaelmargolis/arduino_time
- [20] M., Luboš. *Bezdrátová komunikace ESP-Now s ESP32*. In: *Navody drátek: Webový magazín o ARDUINU* [online]. [cit. 2022-11-14]. Dostupné z: https://navody.drateg.cz/navody-k-produktum/bezdratova-komunikace-esp-now-s-esp32.html?gclid=CjwKCAiA68ebBhB-EiwALVC-NgngzEEYwVfJH6c1LRNAuz0DNFpkmYxrMVjOZptBq1_FcFTSaZyVBxoCrKYQAvD_BwE
- [21] DHT sensor library. In: *Github* [online]. 2022 [cit. 2022-12-09]. Dostupné z: <https://github.com/adafruit/DHT-sensor-library>
- [22] ESP32: Insight Into ESP32 Sleep Modes & Their Power Consumption. In: *Lastminuteengineers: Learn Electronics* [online]. 2022 [cit. 2022-11-09]. Dostupné z: <https://lastminuteengineers.com/esp32-sleep-modes-power-consumption/>

Příloha A

- Zdrojový kód pro zařízení „master“ v jazyce Wiring.
- Zdrojový kód pro zařízení „slave“ v jazyce Wiring.
- Aplikace pro zobrazení přijatých dat.
- Soubor se souřadnicemi měřeného časového průběhu.