



**Fakulta elektrotechnická
Katedra měření**

Samostatný projekt

Využití Wifi v IoT aplikacích a testování specifických vlastností

Petr Kučera
Otevřená informatika

Září 2022, leden 2023

<https://github.com/petrkucerak/rafting-button>

Vedoucí práce: doc. Ing. Jiří Novák, Ph.D.

Poděkování / Prohlášení

Chtěl bych poděkovat svému vedoucímu práce doc. Ing. Jiřímu Novákovi, Ph.D. za trpělivost při hledání důvodů poškození modulů a pomoc se směřováním projektu. Svému bratrovi Jakubovi za pomoci při měření vlastností protokolu ESP-NOW v prostředí divoké přírody a parádní fotografie. Mým kamarádům, spolubydlícím a tátovi za dlouhé noční diskuze, které mne vždy v projektu nasměrovali dalším směrem.

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

Ve Spojil dne 20. 1. 2023

.....

Abstrakt / Abstract

Cílem samostatného projektu je se seznámit s možnostmi Wifi technologie pro IoT aplikace. Následně vyspecifikovat kritické parametry pro aplikaci v navazující bakalářské práci, zvolit správné zařízení a otestovat požadované vlastnosti.

Klíčová slova: IoT; Wifi; embedded; mikrokontroler; samostatný projekt; bezdrátová komunikace; distribuovaný systém.

The goal of the individual project is to familiarize oneself with the possibilities of Wifi technology for IoT applications. Then, to specify critical parameters for application in a subsequent bachelor thesis, to select the appropriate device and test the required properties.

Keywords: IoT; Wifi; embedded; microcontroller; individual project; wireless communication; distributed system.

Title translation: Utilization of Wifi in IoT applications and testing specific properties

/ Obsah



Kapitola 1

Úvod

V první kapitole bych chtěl vysvětlit motivace, které mne vedli k volbě tohoto téma. Rád bych popsal téma bakalářské práce, k níž tato práce směřuje. V poslední části úvodní kapitoly popíšu plány, jak bych rád v práci postupoval a jakých výsledků dosáhl.

1.1 Motivace

Už od útlých dětských let jezdím na letní tábory a přes rok organizuji různé mládežnické akce. Na jedné z akcí, Hudebním týdnu, vždy hráváme hru, v které jsou různé týmy, jenž mezi sebou soutěží. Mají za úkol uhádnout nějakou otázku nebo např. poznat film dle hudby. Nikdy jsme nevymysleli efektivní způsob, jak poznat, kdo se o slovo přihlásil jako první. Vždy to skončí na zvonečku. V situaci, kdy se současně o slovo přihlásí více týmů, nejsme schopni určit pořadí.

1.2 Popis cíle bakalářské práce

Rád bych vytvořil hlasovací zařízení (tlačítko), které by fungovalo jako autonomní systém, který by společně s dalšími tlačítky tvořil distribuovanou síť. Síť by následně řešila problematiku konsenzu. Respektive pomocí vhodně navrženého a implementovaného algoritmu by bylo možné s určitostí tvrdit, jaké z tlačítek bylo stisknuto první.

Představa uživatelského zážitku je taková, že uživatel zapne hlasovací zařízení, zařízení se spojí s dalšími zařízeními v distribuované síti, uživatel má na tlačítku možnost aktivovat lokální server pro zobrazení pořadí stisknutí.

1.3 Samostatný projekt

Před navrhnutím samotného algoritmu pro řešení konsenzu je ale třeba specifikovat klíčové parametry, které bude muset síť zvládat. Proto jsem se v tomto samostatném projektu rozhodl vybudovat síťovou infrastrukturu pro následnou aplikaci při samotné bakalářské práci.

1.4 Stanovení cílů a plánů

Jelikož se jedná o komplexní problém, který čítá mnoho oblastí problematik, rozhodli jsme se, po diskuzi s vedoucím práce, práci rozdělit na několik fází.

- Proveďte rešerši využití Wifi technologie pro IoT aplikace.
- Definujte kritické vlastností Wifi sítě (spolehlivost, odezva, propustnost sítě, ...).
- Zvolte vhodný Wifi modul pro implementaci.
- Experimentálně ověřte dosažení výše definovaných kritických parametrů.

Kapitola 2

Rešerše

Druhá kapitola se věnuje trendům technologií aplikovaných v IoT řešení. Popisuje největší výzvy a nejčastější problémy, s kterými se v praxi setkáváme. Dále se zaměřuje na hierarchickou strukturu IoT sítě.

2.1 Vymezení základních pojmů

Než přejdeme k samotné diskuzi problematiky domnívám se, že je důležité v rychlosti definovat základní pojmy jako je IoT či Industry 4.0 v kontextu tohoto projektu.

2.1.1 Internet věcí

Pojem **internet věcí** (*Internet of Things* či zkráceně IoT) označuje síť fyzických zařízení. Každé zařízení měří data pomocí sensorů nebo zpracovávají již naměřená data či ovládají reakční člen (např. DC motor). Zařízení si mezi sebou vyměňují informace skrze komunikační síť jako je např. *Internet*. [??][??][??]

2.1.2 M2M komunikace

Pojem **Machine to Machine** communications (M2M) označuje bezdrátovou i drátovou komunikaci mezi dvěma a více zařízeními napřímo bez jakéhokoliv zásahu člověka. *M2M* je součástí *IoT*. Jedná se však o pojmy označující zcela jiné skutečnosti. [??]

2.1.3 Industry 4.0

Neexistuje žádná přesná definice pro **Průmysl 4.0** (*Industry 4.0*), která by dle mého objektivně popisovala vše, co se pod tímto pojmem skrývá. Tato okolnost je pravděpodobně způsobena tím, že se jedná o stále probíhající období, na které se nemůžeme dívat s odstupem. Obecně ale můžeme tvrdit, že při specifikování průmyslu čtvrté generace, se definice shodují v hlavních rysech [??][??] popisujících tuto *evoluci*.¹

- **Aditivní výroba** (opak obrábění, předmět vzniká přidáváním materiálu²)
- **Digitální transformace** (smíšená realita, digitální dvojčata atd.)
- **IoT** (síť fyzických zařízení spojující stroje, lidi a věci)
- **Automatizace** s prvky **AI** a **prediktivní analýzy** na základě naměřených dat
- **Cloud Computing** umožňující horizontální a vertikální škálování
- **Kybernetická bezpečnost** systémů

2.1.4 Wireless communication

Pojem **bezdrátová komunikace** (*Wireless communication*) označuje přesun informace z místa A do místa B jinak než pomocí elektrických či optických vláken. Typicky se jedná o vlnění přenášené vzduchem. Jméno a typ vlnění závisí na konkrétních parametrech. Nejčastěji se jedná o *rádiové vlny*. [??]

¹ Pojem evoluce jsem použil schválně. Chtěl jsem tak poukázat na to, že z našeho pohledu současné situace se opravdu nejedná o revoluci nýbrž průběžný a postupný rozvoj technologií.

² Více informací na: https://cs.wikipedia.org/wiki/Aditivní_výrobní_proces

2.2 Trendy ve světě IoT

Zavádění IoT zařízení do denního osobního života a fungování firem přináší výhody, jako je zvýšení efektivity, produktivity a nutnosti nedělat rutinní věci. Trendem je sbírat, co největší množství dat, aby bylo možné data efektivně vytěžit. S tímto trendem souvisí zvyšování množství IoT zařízení, jako jsou například drobné MCU se senzorem pro měření jednoho parametru a modulem pro bezdrátovou komunikaci.

Takovéto zařízení typicky funguje tak, že se jednou za určitou dobu vzbudí. Pomocí senzoru naměří data, která zpracuje do podoby, aby byla interpretovatelná.³ Data uloží a pokračuje ve spánku. Po naměření a elementárním zpracování většího množství dat, dojde k odeslání do hierarchicky nadřazeného zařízení.⁴ K odeslání se obvykle využívá bezdrátového spojení. Po odeslání dat zařízení opět usne a celý cyklus se opakuje.

Takováto zařízení se těší velké oblibě. Jsou drobná, lze je umístit na jakékoliv místo. Jsou optimalizována pro minimální spotřebu energie a jsou napájena z baterie. Díky tomu nezávisí na externím zdroji energie.

Aby mohl být popsán provoz realizován, je nutné všechny vrstvy maximálně optimalizovat pro úsporu energie. Nejvíce dominantní při celém životním cyklu je vysílání zprávy. Při ní je nutné vyzařovat energii. Proto byly navrženy speciální technologické postupy, které pomáhají snížit spotřebu.

Pro základní představu je v tabulce ?? specifikována orientační hodnota spotřeby proudu.⁵

REŽIM ČINNOSTI	SPOTŘEBA
aktivní provoz (měření, ukládání do paměti, ...)	5mA
režim vysílání (odesílání, případně příjem dat)	0.045A
režim spánku (čekání vzbuzení)	1μA

Tabulka 2.1. Orientační hodnoty pro porovnání režimů činností dle spotřeby energie.

Je možné, že pokrok technologií nám přinese řešení napájení. Jedna z možných cest je, že MCU bude odebírat potřebou energii z vysílaného signálu.⁶ V současnosti není zvykem takováto zařízení využívat a ani se nedomnívám, že se jedná o univerzálně správné řešení. To především kvůli tomu, že IoT zařízení jsou umístěna i na místech, kde se nepředpokládá přítomnost dostatečně silného signálu.

2.2.1 Problematika spotřeby energie při bezdrátové komunikaci

Jak bylo výše popsáno, největší energetickou spotřebu na činnosti IoT zařízení má ve většině případů bezdrátová komunikace, konkrétně vysílání. Při hledání řešení, jak vybudovat síťovou infrastrukturu pro samostatný projekt, jsem se setkával s řešením této problematiky. Pro moji aplikaci není v počáteční fázi nikterak kritická. Rozhodl jsem se ale prozkoumat základní technologie, které jsou v dané oblasti k dispozici. V podstatě mapují technologie používané pro bezdrátovou komunikaci.

³ Zpracováním dat je myšleno interpretovat data tak, aby dávala smysl i mimo kontext senzoru. Například při měření vzdálenosti pomocí vířivých proudů, je třeba naměřené napětí vyčistit *lock-in detekcí* a kalibrovat, aby vzdálenost byla v jednotkách délky.

⁴ Gateway či router, které data odesílají dále nebo koncentrátor. Který data ukládá a dále zpracovává.

⁵ Hodnota proudu není dynamicky specifikována v závislosti na čase činnosti. Pro stanovení přesné hodnoty pro spotřebu po dobu n s by bylo nutné hodnotu vynásobit n , kde $[n] = As$.

⁶ <https://cdr.cz/clanek/intel-ukazal-iot-cip-ktery-si-vystaci-s-energii-z-radiovyh-vln>

Podle požadavků fungování IoT zařízení je vhodné volit danou technologii. Pokud budou odesílaná data v desítkách až stovkách bajtů a zařízení bude nuceno fungovat na větší vzdálenost, pravděpodobně zvolíme technologii označované jako **Low Power Wide Area Network (LPWAN)**. Mezi ně patří například LoRaWAN⁷, Sigfox⁸ nebo NB-IoT.

LoRaWAN je komunikační protokol a systémová architektura postavená na technologii LoRa. Zkratka LoRa pochází z anglického *long range*, v překladu tedy dlouhý dosah. [??] LoRa používá modulaci CSS (*chirp spread spectrum*), která v podstatě rozprostře signál po celém vysílacím kanálu s lineární změnou kmitočtu. LoRa funguje v pásmech ISM 433, 868 a 915 MHz. LoRaWAN umožňuje tři módy provozu, které se liší podle energetické náročnosti a počtu přenesených dat. [??]

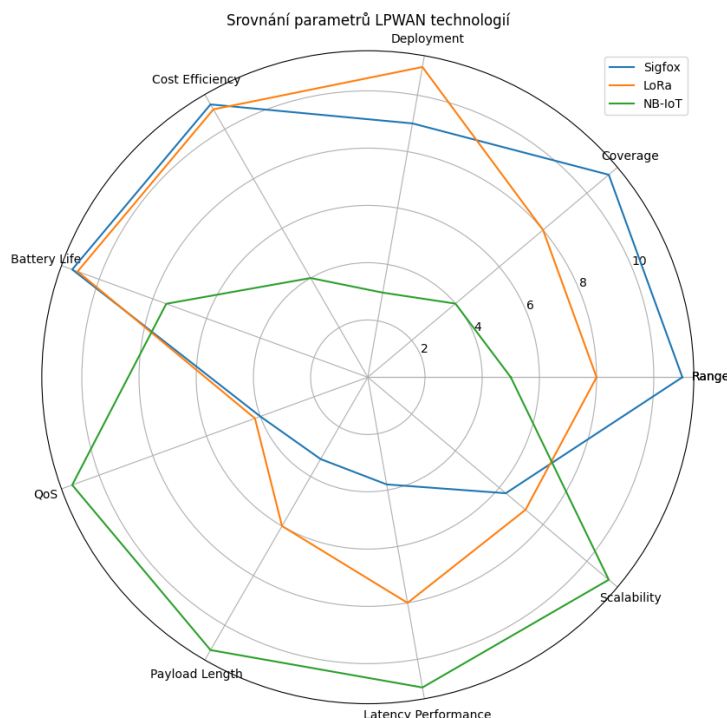
Sigfox je francouzská společnost, která se věnuje komunikacím s nízkou spotřebou energie. Poskytují službu Sigfox, která podobně jako LoRaWAN slouží především k sbírání dat z IoT zařízení. Sigfox se v praxi používá například v elektroměrech či chytrých pračkách, které jednou za čas odesílají malé množství dat. [??] Na rozdíl od technologie LoRa používá UNB (*Ultra Narrow-Band*). Řešení Sigfox se liší také v tom, že neumožňuje data odesílat do vlastní infrastruktury, nýbrž je třeba využívat unifikovaný cloud přímo od společnosti. [??]

Narrowband IoT (NB-IoT) je technika, kdy úzkopásmová síť využívá již vybudované síťové infrastruktury poskytovatelů mobilní signálu. Pásmo, v kterém NB-IoT může běžet, je veliké 200kHz. [??] Přenosová rychlost se liší dle generace komunikace.⁹ Trendu užití již vybudovaných IoT sítí se budu věnovat v další kapitole. [??]

⁷ Webové stránky organizace starající se o udržování LoRaWAN <https://loro-alliance.org/>.

⁸ Webové stránky společnosti Sigfox: <https://www.sigfox.com/>

⁹ Detailní přehled jednotlivých typů NB-IoT s informací o přenosové rychlosti: https://en.wikipedia.org/wiki/Narrowband_IoT#3GPP_LPWAN_standards.



Obrázek 2.1. Srovnání parametrů LPWAN technologií. [??]

Technologie Sigfox vyniká dosahem, pokrytím a nízkou energetickou náročností. LoRa je stejně jako Sigfox vhodná pro aplikace s extrémně nízkou energetickou náročností. Navíc dokáže odeslat více zpráv. NB-IoT nevyniká v dosahu. Zvládne ale dosahovat mnohem větších přenosových rychlostí, kapacit a jeho latence je ze všech doposud zmiňovaných technologií, nejnižší. [??]

Pokud jsou požadavky na IoT technologii takové, že vyžadujeme vysoce energeticky nenáročnou komunikaci, která je na malou vzdálenost, připadají v úvahu především technologie jako Bluetooth, ZigBee či Wifi, které jsou definovány v rodině standardů *IEEE 802*. [??] Analogicky podél LPWAN bychom skupinu takových zařízení mohli pojmenovávat jako **Low Power Personal Area Network (LPPAN)**.

Fyzická vrstva **ZigBee** je popsána standardem *IEEE 802.15.4*. Jedná se o bezdrátovou komunikaci, která funguje do dosahu 75 m na frekvencích ISM 868 MHz, 902 - 928 MHz a 2,4 GHz. Pro směrování používá multiskokové *ad-hoc* směrovací metody. Díky tomu zvládá dosahovat velkých vzdáleností. Na rozdíl od Bluetooth jsou jeho typickou aplikací průmyslové senzorové sítě. [??]

Technologie **Bluetooth** původně měla nahradit drátové sériové rozhraní *RS-232*. Je definovaná standardem *IEEE 802.15.1*. Pracuje v ISM pásmu na frekvenci 2,4 GHz. Nejnovější verze Bluetooth přenáší data rychlostí až 2 Mbit/s na vzdálenost 200 m a velikost zprávy může být až 255 bajtů. Bluetooth se těší velké oblibě pro komunikaci s drobnou elektronikou, jako jsou například bezdrátová sluchátka. Domácími spotřebiči či elektrickými hudebními nástroji, u kterých umožňuje provádět konfiguraci. [??]

Wifi technologie je skupina bezdrátových protokolů, která vychází ze standardu *IEEE 802.11*. V počátcích svého vývoje by se dala nazvat jako dvojče *IEEE 802.3* (Ethernet). V průběhu času nové verze Wifi přinášejí novinky, které v Ethernetu nejsou a z podstaty

ani nikdy nebudou. Wifi funguje v pásmu ISM 2,4 GHz a 5 GHz. [??] Široká veřejnost zná Wifi jako nástroj, který slouží pro komunikaci s Internetem. V praxi má ale mnohem širší využití.

Zajímavou demonstrací postupných technologických novinek je vývoj Wifi jako technologie v čase. [??] Jednotlivé verze ukazují na trendy jako je zvyšování výpočetního výkonu a požadavků na propustnost sítě. Možnosti protokolů a vznik speciálních nových protokolů, například i s IoT technologiemi.¹⁰ [??]

Ze všech tří vyjmenovaných technologií rozsahu osobní sítě (PAN) se Wifi těší největší oblibě. Dle mého názoru je tato skutečnost způsobena především univerzálností, která pramení z:

- rozšíření Wifi sítě (77% domácností v rámci svého domu nebo bytu rozvádějí Wifi pomocí *routeru* nebo *modemu*¹¹),
- a vysokými rychlostmi přenosu a nízkou latencí.

Jak uvádí autoři v úvodní části [??], výzva je vybudovat robustní ale zároveň energeticky nenáročnou síť. Takovouto aplikaci technologie Wifi označujeme pojmem **Low-Power Wifi**.

2.2.2 Low-Power Wifi

Pokud se podíváme na vývoj verzí *IEEE 802.11*, je si vhodné položit otázku, jaká verze je pro IoT aplikace nejvíce vhodná. Dle článku [??] je *802.11n* (Wifi-4) optimálnější pro aplikaci v IoT zařízení než *802.11ac* (Wifi-5). Verze *802.11n* totiž nabízí 2 pásma pro komunikaci (2,4 GHz a 5 GHz). Oproti tomu pozdější verze *802.11ac* pouze jeden kanál na 5 GHz. Pro IoT aplikace je užitečnější 2,4 GHz, protože signál lépe proniká materiálem a má větší dosah. Navíc verze *802.11ac* má vyšší energetická náročnost, která je způsobena komplikovanějším algoritmem sloužícím k zrychlení přenosu, které je pro většinu IoT aplikací nepotřebné.

Nově do hry také přichází nejnovější verze **Wifi-6** definovaná ve standardu *802.11ax*. Slibuje hned několik novinek, které by měli IoT zařízeními rapidně pomoci.

Nejčastější překážka, kterou v této práci zmiňuji, je úspora energie. Wifi-6 implementuje **Target Wait Time (TWT)** mechanismus, který umožňuje IoT zařízení při nečinnosti usnout a tak ušetřit značné množství energie (viz 2.1).

Druhou novinkou je **OFDMA**,¹² který umožňuje odesílat stejná data do více zařízení ve stejný čas.

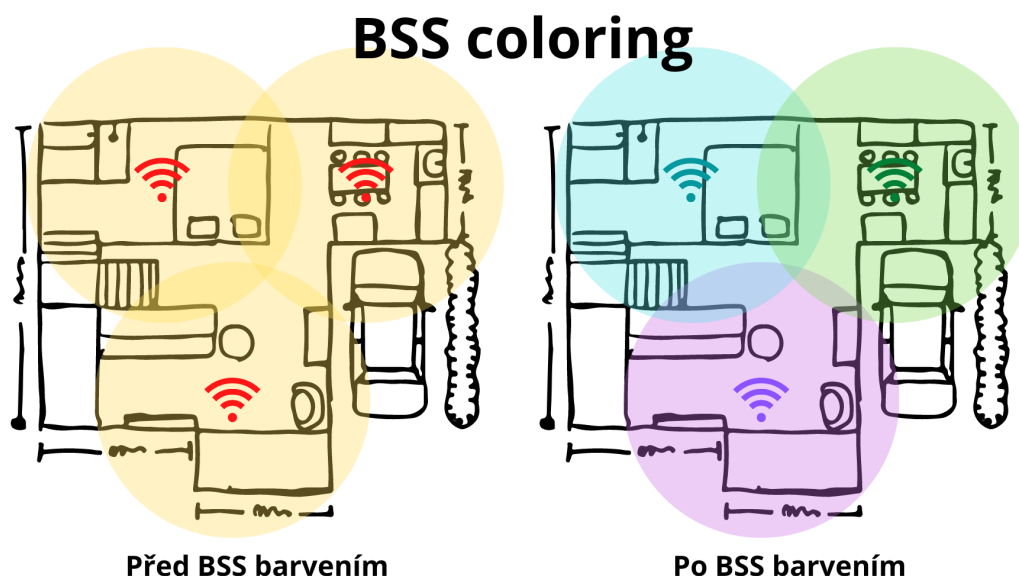
Poslední novinkou, na kterou bych rád upozornil je **BSS**¹³ **coloring**. Novinka umožňuje *obarvit* signál z různých *access points* a tak předejít překrývání signálů. [??]

¹⁰ Rychlý průlet verzemi Wifi technologie je například na webové stránce: <https://www.pcmag.com/encyclopedia/term/80211-versions>.

¹¹ Hodnoty v ČR k roku 2022 [??]

¹² Orthogonal frequency-division multiple access

¹³ Basic Service Set coloring



Obrázek 2.2. Vizualizace fungování BSS barvení ve Wifi-6.

2.2.3 Optimalizace nízké spotřeby na zařízení

Energii pro delší fungování zařízení lze především ušetřit správnou volbou MCU a vhodnou konfigurací. Existuje několik pravidel, které mohou pomoci snížit spotřebu energie a vytvořit umožnit, tak IoT zařízení fungovat lépe.

- volba vhodného modulu či MCU
- uvedení zařízení do *deep sleep módu*¹⁴
- vypínání WiFi při neaktivitě
- vhodná optimalizace kódu

Na první pohled by se mohlo zdát, že se vyplatí použít všechna pravidla. To ale nemusí být vždy nejvhodnější řešení. Především kombinace vypínání WiFi a přecházení do *deep sleep módu*, jak ukázal experiment Přeučila a Martin Novotného. [??]

2.3 Struktura sítě

Pro IoT řešení je typická **hierarchická struktura sítě**. Tato povaha je dána účelem řešení, nejtypičtěji vytěžování dat. Hierarchická struktura sítě je podobná architektuře internetu.¹⁵

- embeded zařízení, senzory, akční členy
- koncentrátoři, routery a gateway
- server, cloud computing

První úroveň se stará o měření dat pomocí senzorů, která jsou zpracovávána v embeded zařízeních nebo o vykonávání činností na základě přijatých dat.¹⁶ V IoT scénářích

¹⁴ Procesor přejde do tzv. *deep sleep módu* a vzbudí se až při zachycení interruptu.

¹⁵ Popsaná struktura neplatí absolutně ve všech případech. Snaží se pouze popsat typický scénář IoT projektu.

¹⁶ Například pomocí generovaného PWM signálu rozsvícení adresovatelné LED diody

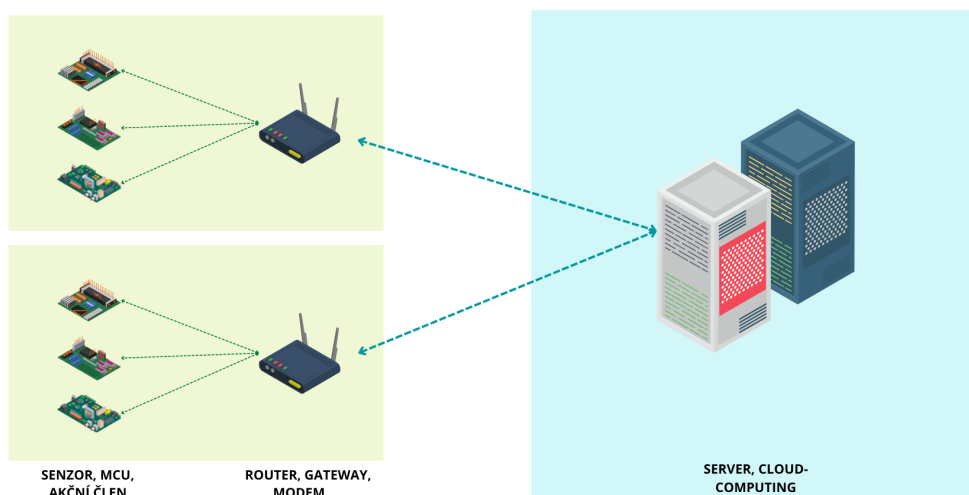
se ale častěji jedná o měření. Tato úroveň následně informace odesílá do hierarchicky nadřazené struktury.

Druhá úroveň sbírá data z více embedded zařízení a zprostředkovává odeslání do úrovně třetí. Data mohou být na této úrovni i předzpracována. Typické zařízení pro tuto úroveň je router, koncentrátor, které bývají vstupní bránou (gateway) do třetí úrovně.

Na **třetí úrovni** typicky běží server, který může být onpremisového či cloudového charakteru. Výhodou onpremisu je to, že mám zařízení u sebe. To může přinášet ekonomickou či bezpečnostní výhodu. Výhodami cloudu je naopak možnost *horizontálního* a *vertikálního škálování*¹⁷ či *Pay As You Go*¹⁸ subskripce. Server data zpracovává. Ukládá a připravuje vizualizace pomocí nástrojů, jako jsou například open-source *Kibana*¹⁹ a *Grafana*²⁰. Kromě vizualizace jsou data na této úrovni analyzována pomocí algoritmů. Mezi ně může i patřit strojové učení. Cílem je přeměnit data na zanalost, aby je bylo možné interpretovat.

Data jsou přenášena skrz strukturu oběma směry. Pro IoT je ve většině případech spíše vytěžován směr nahoru, respektive z embedded zařízení na server.

Tradiční struktura IoT řešení



Obrázek 2.3. Vizualizace typické (hierarchické) struktury IoT projektu.

¹⁷ Horizontální škálování znamená přidávání nových instancí. Vertikálním se myslí navyšování a snižování výkonu.

¹⁸ Zaplat jen to, co utratíš.

¹⁹ <https://www.elastic.co/kibana/>

²⁰ <https://grafana.com/>

Kapitola 3

Specifikace požadavků

Třetí kapitola specifikuje a zdůvodňuje jednotlivé parametry, které jsou požadovány při budování síťové infrastruktury.

3.1 Uživatelské požadavky

Zařízení by mělo:

- být jednoduše ovladatelné,
- být odolné vůči úhozů a pádu na zem,
- řešit, kdo první stiskl tlačítko,
- mít dobře přístupné spouštěcí tlačítko,
- být identické s ostatními,
- být možné napájet pomocí baterií,
- být možné přepnout do módu *PREZENTÉR*, v kterém bude ukazovat výsledky,
- fungovat v hospodě ale i v přírodě.

3.2 Softwarové požadavky

Software by měl:

- fungovat jako distribuovaná síť, která bude řešit volbu konsenzu,
- umět vytvořit síť,
- umět přenášet informace do velikosti 200 bajtů s nízkou latencí,
- být zdokumentovaný,
- fungovat bez chyb,
- být testován.

3.3 Požadavky na hardware

Hardware by měl:

- být optimalizovaný pro minimální spotřebu energie,
- být napájen z baterií, které lze měnit,
- mít modul pro komunikaci pomocí Wifi,
- být dokumentován,
- být odolný pádu ze stolu.

Kapitola 4

Realizace

Čtvrtá kapitola popisuje průběh samotné praktické práce na samostatném projektu. Mapuje celý proces od volby modulu přes základní konfiguraci vývojového prostředí až po užití knihovny a tvorbu softwaru.

Veškerý proces vývoje je zaznamenán v repozitáři na GitHubu¹.

4.1 Architektura projektu

Přesnou podobu projektu jsem vytvářel agilně při vývoji. Vycházel jsem ovšem z požadavků sepsaných v kapitole 3.

4.1.1 Komunikační protokol

Při studiu možností IEEE 802.11 pro IoT zařízení, jsem se setkal pouze s hierarchickou síťovou strukturou. Rozhodl jsme se pro distribuovanou síť, kde si budou všechna zapojená zařízení na počátku rovna.

Při tvorbě řešení jsem objevil nemalé množství protokolů, s nimiž by mohla zařízení mezi sebou komunikovat. Nejvhodnější pro mou aplikaci byly MQTT protokol či klasický TCP/IP. Po diskuzi s vedoucím práce jsem se rozhodl pro experimentálnější způsob. Nepoužiji žádný ze zmiňovaných protokolů, nýbrž se pokusím implementovat vlastní protokol, který by dle definice IEEE 802.11,² posílal MAC frames. Celá síť by byla adresovaná na úrovni MAC adres.³

Při pozdějším řešení implementace pro konkrétní typ modulu jsme objevil řešení, protokol EPS-NOW, který splňuje všechny mé požadavky.

4.2 Volba vhodného modulu

Prvním reálným krokem v práci bylo vybrat konkrétní typ modulu. Z požadavků specifikovaných v kapitole 3 jsem sestavil kritéria, které by měl modul splňovat:

- podpora standardů IEEE 802.11 (b/g/n)
- mít možnost pracovat s MAC frames
- pin pro zapojení tlačítek
- možnost napájení z baterie

Po rychlém průzkumu trhu jsem vybral prvně 5 následně 3 zařízení, které splňovali moje požadavky.

¹ <https://github.com/petrkucerak/rafting-button>

² konkrétně 2 úrovně síťového modelu OSI (linkové)

³ Nepovedlo se mi totiž žádný takový existující protokol vyhledat. Ani po diskuzi s odborníky na telekomunikační technologie z řad učitelů, jsem se nedozvěděl o jakémkoliv již funkčním způsobu posílání dat pouze na 2 úrovni síťového OSI modelu.

- ESP32
- ESP8266
- PI Pico
- CC3200
- ATSAMW25

Velice rychle jsem ale zavrhnul poslední dva jmenované a vybíral mezi třemi výše zmiňovanými. Nakonec jsem se rozhodl pro architekturu ESP32. To především z důvodů dostupnosti zařízení na trhu.

Objednal jsem si moduly z rodiny ESP32 od firmy Espressif,⁴ konkrétně 2x *ESP32-S2-pico* a 1x *ESP32-S2-LCD-0.96inch* s displejem pro účely debugování.

Jedná se o Wifi vývojové desky s základními periferiemi jako je ADC převodník, I2C a SPI komunikace či UART. Deska integruje *low-power Wifi System on Chip* (SoC). Oproti od ESP32, které má 2 jádra procesoru, ESP32-S2 má pouze jeden Xtensa single-core 32-bit procesor, označován jako ESP32-S2FH4, který podporuje frekvenci hodin až do 240 MHz.

Další specifikace je možné najít na stránkách dodavatele.⁵ Do přílohy ?? přikládám relevantní schémata.

4.3 Vývojové prostředí a konfigurace

MCU z rodiny ESP32 a ESP8266 lze programovat v podstatě 2 způsoby, respektive 2 frameworky. Já se rozhodl postavit základ na Arduino frameworku, který je pro počáteční konfiguraci méně náročný a používat knihovny z ESP-IDF frameworku.

Jako vývojové prostředí jsem si vybral VS Code s PlatformIO. Dle mého se jedná o uživatelsky přívětivé prostředí. Navíc má vysokou podporu komunity a disponuje již spoustou připravených balíčků a konfigurací pro vývojové desky.

Založil jsem repozitář, pro celý projekt. Naklonoval si složku do PC a vytvořil nový PlatformIO projekt s konfigurací.⁶

4.4 Konfigurace komunikace pomocí USB-C

Pro otestování toho, zdali zařízení správně funguje, bylo třeba prvně nakonfigurovat komunikaci skrze USB zařízení. Při této implementaci jsem vycházel z ukázky kódu, které poskytuje dodavatel.⁷

Projekt s konfigurací pouze USB-C lze najít v release verzi 3.0.1.⁸

⁴ <https://www.espressif.com/>

⁵ <https://www.waveshare.com/wiki/ESP32-S2-Pico>

⁶ `[env:esp32dev];` podmínka prostředí, lze využít v budoucnu
`platform = espressif32`
`board = esp32-s2-saola-1;` vývojový bord se stejným procesorem
`framework = arduino;` volba frameworku
`board_build.mcu = esp32s2;` volba typu procesoru
`upload_protocol = esptool;` nástroj pro stahování softwaru do zařízení
`board_build.flash_cpu = 240000000L;` definování frekvence
`board_build.flash_mode = dio;` rychlostní mód nahrávání do paměti

⁷ Adresa pro stažení demo kódů: https://www.waveshare.com/wiki/ESP32-S2-Pico#Examples_2

⁸ <https://github.com/petrkucera/rafting-button/tree/3.0.1>

4.5 Konfigurace display a tvorba základních funkcí

Při prozkoumávání možnostech modulů jsem chtěl zprovoznit LCD displej, který by mohl sloužit k debugování a zobrazování dat nezávisle na terminálu. Opět jsme vyšel z ukázkového kódu na stránkách dodavatele.⁹ Tentokrát bylo ale třeba implementovat více funkcí. Přidal jsme metody pro vykreslování barev, čtverečků a textu v pěti velikostech.

Veškeré zdrojové kódy k ovládání displeje jsou ve složce `src/display`.¹⁰

4.6 Implementace ESP-NOW

Během práce a seznamování s modulem od Espressif jsem objevil protokol ESP-NOW, který implementuje odesílání informací na 2 úrovni.¹¹ Jedná se o univerzální protokol, který je možný upravit pro svoje konkrétní potřeby. Umí odesílat informace o maximální velikosti 250 bajtů (dáno vlastností rámů). Má šifrovanou i nezabezenou verzi. Adresu se na úrovni MAC adres.

Protokol není nikterak robustní. Je poměrně jednoduchý. Tato skutečnost podmiňuje, že pro komplexnost je třeba implementovat více funkcí sám. Rozhodl jsem se, že, pro užívání tohoto protokolu ve svém projektu, zvolím následující pravidla:

- před spuštěním protokolu, je třeba registrovat *callback* funkce,
- informace z *callback* funkcí budou ukládány do pomocné struktury `ESP_NOW_HANDLER`,
- struktura bude kromě informací ukládat MAC adresu příjemce.

Díky přidání takto jednoduché struktury jsem docílil toho, že jsem schopen efektivně měřit a zpracovávat data.

Projekt v této fázi si je možné prohlédnout ve verzi 3.0.2¹².

4.7 Poškození čipů

Během tvorby programu pro testování round-time trip, se začal jeden z modulů přehřívat. I přes to Modul fungoval dál. Pravděpodobně došlo k poškození části procesoru a proudová spotřeba stoupla o řád výš.

Později při vyladování uživatelského zážitku z měření s modulem, který má displej, se začal přehřívat i čip druhý. Nejednalo se o stejnou chybu. Zařízení přestalo úplně odpovídat. Došlo k poškození, které jsme nebyli schopni s vedoucím práce, ani po dlouhé investigaci, přesně rozklíčovat či napravit.

Tato skutečnost do jisté míry ovlivnila moji následující práci. Naštěstí jen tím, že u měření nešlo sledovat progress bar na displeji, který jsem implementoval a také tím, že jsme nemohl měřit komunikace mezi více zařízeními ve stejný čas.

4.8 Program pro testování round-time trip

V závěrečné části jsem navrhu a implementoval program pro měření round-time tripu. Fungování je detailněji popsáno v kapitole ??.

Kód pro měření včetně Python skriptu pro vizualizaci výsledků je dostupný ve verzi 3.0.3¹³.

⁹ Adresa pro stažení demo kódu: https://www.waveshare.com/wiki/ESP32-S2-Pico#Examples_2

¹⁰ <https://github.com/petrkucerak/rafting-button/tree/main/src/display>

¹¹ síťového modelu ISO/OSI

¹² <https://github.com/petrkucerak/rafting-button/tree/3.0.2>

¹³ <https://github.com/petrkucerak/rafting-button/tree/3.0.3>

Kapitola 5

Měření

Kapitola popisuje prostředí, v kterých byla měřena data a způsob jejich měření. Vykresluje jednotlivé scénáře a interpretuje naměřená data v souvislostech.

5.1 Úvod k měření

Abych byl schopen zajistit plynulé fungování distribuované sítě, je třeba otestovat vlastnosti protokolu ESP-NOW. Proto jsem v závěru samostatného projektu vytvořil jednoduchý program pro testování. Funguje tak, že:

1. zaregistruje USB callback a spustí sériovou komunikaci,
2. inicializuje struktury nutné pro správné fungování protokolu ESP-NOW¹ a zaregistruje callback pro příchozí a odchozí komunikaci.
3. V případě veze MCU s displejem,² inicializuje displej a vypíše na něj vlastní MAC adresu.
4. Ukončí inicializační metodu. V případě MCU bez displeje³ ohlásí konec konfigurace bliknutím červené LED diody, která je vestavěná na desce.
5. Spustí měření, které má za úkol simulovat reálný provoz. Zde je možné konfigurovat základní parametry⁴. Program měří *round-time trip*. Respektive dobu, poslání informace ze zařízení A, jejího uložení do patřičné struktury v zařízení B, odeslání zpět do zařízení A a uložení do patřičné struktury v zařízení A. Pokud se informace nevrátí ze zařízení B do určitého deadline, zařízení A registruje zprávu jako chybovou a odesílá rovnou další. V případě použití MCU s displejem, je proces měření zobrazován ve formě *progress baru*.
6. Po ukončení měření, se data vypíší do konzole pro následné zpracování.

5.2 Scénáře měření

Aby bylo možné otestovat různé parametry v různých prostředích, rozhrnul jsem měření na několik scénářů.

Při měření byly sledovány následující parametry:

- prostředí, v němž bylo měření prováděno,
- překážka mezi zařízeními,
- vzdálenost mezi zařízeními,
- velikost zprávy,^{5 6}
- počet odeslaných zpráv,

¹ Jako je například 'ESP_NOW_HANDLER'

² ESP32-S2-pico-LCD

³ ESP32-S2-pico

⁴ Více je specifikováno v sekci ??.

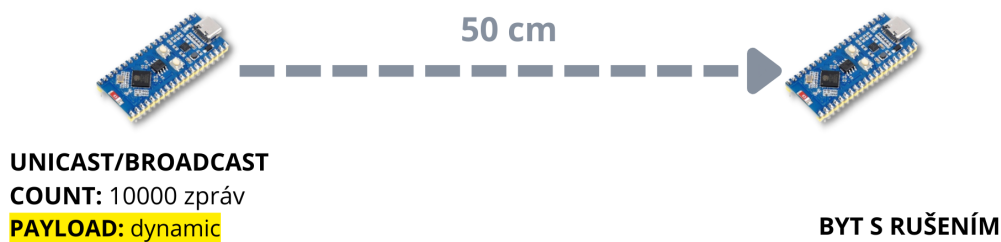
⁵ Někdy označováno také jako *payload*.

⁶ Maximální velikost zprávy je 250 bajtů.

- typ odesílání (*broadcast/unicast*),
- počet chybných zpráv.⁷

5.2.1 Scénář A

Scénáře Axy



Obrázek 5.1. Vizualizace parametrů pro měření scénářů A.

V scénářích typu A se snažím otestovat to, jak změna parametru typu odesílání, tj. přepínání mezi *broadcastem* a *unicastem*, ovlivní *round-time trip* v závislosti na velikosti zprávy. Měření se odehrává v prostředí bytu v činžovním domě, kde dochází k rušení několika okolními Wifi sítěmi.

SCÉNÁŘE	A11	A12	A13	A14	A15
prostředí	byt	byt	byt	byt	byt
překážka	vzduch	vzduch	vzduch	vzduch	vzduch
vzdálenost	50 cm	50 cm	50 cm	50 cm	50 cm
velikost	1 B	10 B	50 B	120 B	250 B
počet zpráv	10 000	10 000	10 000	1 000	5 000
typ vysílání	broadcast	broadcast	broadcast	broadcast	broadcast

Tabulka 5.1. Přehled parametrů pro A1x scénáře.

SCÉNÁŘE	A21	A22	A23	A24	A25
prostředí	byt	byt	byt	byt	byt
překážka	vzduch	vzduch	vzduch	vzduch	vzduch
vzdálenost	50 cm	50 cm	50 cm	50 cm	50 cm
velikost	1 B	10 B	50 B	120 B	250 B
počet zpráv	10 000	10 000	10 000	10 000	10 000
typ vysílání	unicast	unicast	unicast	unicast	unicast

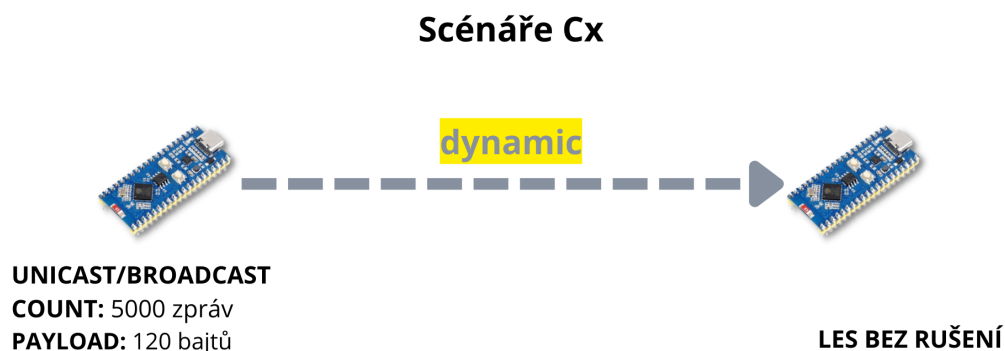
Tabulka 5.2. Přehled parametrů pro A2x scénáře.

Výsledky měření těchto scénářů si je možné prohlédnout v grafech ?? a ??. Odesílání se chová dle očekávání. Zprávy větší velikosti trvají déle než zprávy té menší. Zajímavé je srovnání *broadcastu* a *unicastu*. *Broadcast* je nepatrně pomalejší.

Také je zajímavé si povšimnout nakumulovaných odpovědí v jeden čas. Tuto skutečnost si vysvětlují implementací *broadcastu* v protokolu ESP-NOW.

⁷ Měření tohoto parametru bylo přidáno od scénáře A, kde ovšem k žádné ztrátě nedocházelo.

5.2.2 Scénář C



Obrázek 5.2. Vizualizace parametrů pro měření scénářů C.

Sadou scénářů Cx se snažím pozorovat vlastnosti v signálově čistém prostředí⁸ v závislosti na vzdálenosti a typu vysílání.

SCÉNÁŘE	C1	C2	C3	C4	C5	C6
prostředí	les	les	les	les	les	les
překážka	vzduch	vzduch	vzduch	vzduch	vzduch	
vzdálenost	0,5 m	25 m	50 m	100 m	100 m	50 m
velikost	125 B	125 B	125 B	125 B	125 B	125 B
počet zpráv	5 000	5 000	5 000	5 000	5 000	5 000
počet chybných zpráv	0	3	15	35	15	6
typ vysílání	unicast	unicast	unicast	unicast	broadcast	broadcast

Tabulka 5.3. Přehled parametrů pro Cx scénáře.

Výsledky měření těchto scénářů si je možné prohlédnout v grafu ???. Zde se opět prokol chová dle očekávání. Nedochází k takovému zpoždění, jako například při měření scénářů typu A. Také si je možné povšimnout toho, že se jednou za čas nějaká zpráva opozdí.

Při tomto měření jsem zaznamenával také chybovost počet chybných zpráv.⁹ Jejich četnost si ji možné prohlédnout v tabulce 5.3.

Během tohoto měření jsem zjistil, že je důležité, aby na větší vzdálenosti,¹⁰ nestála signálu v cestě žádná překážka.

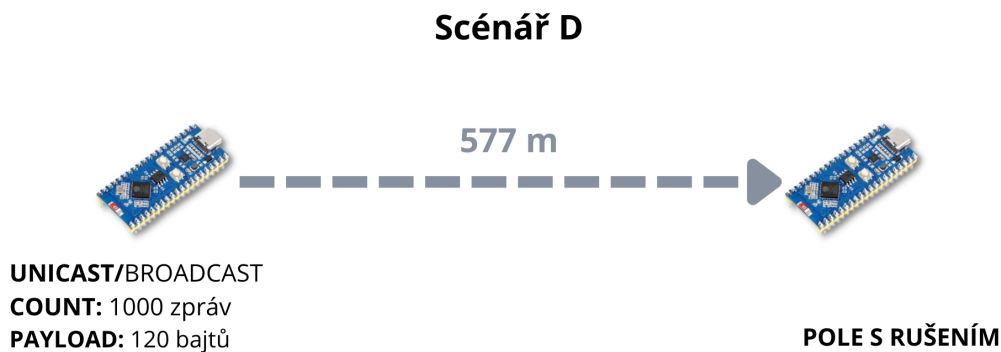
Zajímavé je také srovnání scénáře A a C. Můžeme pozorovat, že **vliv vzdálenosti ovlivňuje především ztrátovost paketů. Naopak velikost ovlivňuje rychlost přenosu.**

⁸ Nejedná se o laboratorně čisté prostředí. Měření bylo prováděno v prostředí lesa, který je od nejbližší obce vzdálen asi 5 km a v okolí mého bydliště nejvíce čisté od 2,4 GHz rušení.

⁹ Chybnou zprávou se myslí taková zpráva, která nedorazí do specifikovaného deadline, tedy mimo graf.

¹⁰ 25 m a více

5.2.3 Scénář D



Obrázek 5.3. Vizualizace parametrů pro měření scénářů D.

Scénář D byl oproti ostatním měřením odlišný v tom, že jsem se nejprve snažili stanovit hranici, kdy je zařízení ještě schopno přijímat zprávy a kdy už ne. Experimentálně jsem dospěl k hranici 580 m. Následně jsem odeslal 1000 zpráv s cílem zjistit, jak je velká ztrátovost. Měření bylo realizováno na poli, přes které může procházet signál na 2,4 GHz.

SCÉNÁŘE	D1
prostředí	pole (s 2,4 GHz)
překážka	vzduch
vzdálenost	577 m
velikost	125 B
počet zpráv	1 000
počet chybných zpráv	50
typ vysílání	unicast

Tabulka 5.4. Přehled parametrů pro D scénář.

Při měření jsme zjistil, že při odesílání na velkou vzdálenost je třeba dbát na orientaci čipu. Pokud nebylo zařízení správně natočeno, nešlo odeslat žádné zprávy.

Výsledek měření je vizualizován grafem ???. Při tomto scénáři bylo ovšem mnohem zajímavější pozorovat četnost úplné ztráty dat.

Při odeslání 1000 zpráv, se ztratilo 50. Můžeme tedy jednoduchým výpočtem zjistit jaká je procentuální ztrátovost na dlouhé vzdálenosti.

$$loss = error / sent$$

V našem případě se při odeslání 1000 zpráv objevilo 50 chyb. Chybovost je tedy 5 %.¹¹

5.3 Hodnocení měření

Veškerá naměřená data se skripty pro vyrenderování grafů je možné najít v repozitáři projektu.¹²

¹¹ $50/1000 = 0,05$

¹² <https://github.com/petrkucerak/rafting-button/tree/main/measure>

Uvědomuji si, že měření nebylo prováděno za ideálních podmínek. Přesnost měření mohlo ovlivnit především to, že:

- se jeden čipů měl větší odběr proudu a mírně se při měření přehříval,
- větší vzdálenosti¹³ byly měřeny s přesností na jednotky metrů a největší vzdálenosti¹⁴ s přesností na 10 m,
- les nebyl absolutně odstíněný od 2,4 GHz,
- během měření různých scénářů jsme upravoval kód,
- v některých scénářích by větší množství dat, mohlo říci více.

Zajímavé jsou ještě 2 výše zmíněné poznání. Konkrétně to, že při přenosu na delší vzdálenost je nutné, aby signálu nebránilo nic v cestě a aby byly moduly správně orientovány.

Výsledky prokazují, že protokol ESP-NOW splňuje očekávané požadavky a je vhodný pro použití v budoucí implementaci.

¹³ více jak 10 m

¹⁴ více jak 100 m

Kapitola 6

Závěr

6.1 Průběh projektu

Při práci na tomto projektu jsem udělal spoustu chybných rozhodnutí, které mě zavedli na slepou cestu. Během studia možností jsem z počátku nevěděl, kde hledat vhodné informace. Následně jsem něco našel, ale neuměl jsem se v literatuře orientovat.

Věřím, že jsem se i poučil v dalších oblastech. Například příště při objednávání modulů, se pokusím rovnou sehnat více zařízení, abych byl připraven na to, že nějaké zařízení může přestat fungovat. Velice dlouhou dobu jsem také probděl nad tím, jak správně nakonfigurovat USB callback, aby fungovala komunikace s PC. Až po delší době mne napadlo se podívat do DEMO programů.

I měření mne poučilo. Především tím, že je si sebou do terénu potřeba vzít opravdu vše, co vás napadne.

6.2 Budoucnost projektu

Rád bych si prvně celý projekt detailně zhodnotil se svým vedoucím, protože by mne lákalo v tématu pokračovat i během bakalářské práce. Tou bych rád navázal na tento samostatný projekt.

Literatura

- [1] *Internet of Things Global Standards Initiative*.
<https://www.itu.int/en/ITU-T/gsi/iot/Pages/default.aspx>.
- [2] Alexander S. Gillis. *What is the internet of things (IoT)?* 2022.
<https://www.techtarget.com/iotagenda/definition/Internet-of-Things-IoT>.
- [3] *Internet of things*. 2001-.
https://en.wikipedia.org/wiki/Internet_of_things.
- [4] *Machine to Machine Communications*. 2022.
<https://dot.gov.in/machine-machine-communications>.
- [5] *Průmysl 4.0*. 2022.
<https://www.plm.automation.siemens.com/global/cz/our-story/glossary/industry-4-0/29278>.
- [6] *Co je průmysl 4.0?*
<https://www.sap.com/cz/insights/what-is-industry-4-0.html>.
- [7] *Wireless*. 2001-.
<https://en.wikipedia.org/wiki/Wireless>.
- [8] *LoRa*. 2001-.
<https://en.wikipedia.org/wiki/LoRa>.
- [9] *LoRa Alliance®*. 2023.
<https://loro-alliance.org/>.
- [10] *Sigfox*. 2001-.
<https://cs.wikipedia.org/wiki/Sigfox>.
- [11] *Sigfox*. 2023.
<https://www.sigfox.com/>.
- [12] *Narrowband IoT*. 2001-.
https://en.wikipedia.org/wiki/Narrowband_IoT.
- [13] *Vodafone*. 2023.
<https://www.vodafone.cz/podnikatele/internet-veci/nb-iot1/>.
- [14] Kais Mekki, Eddy Bajic, Frédéric Chaxel a Fernand Meyer. A comparative study of LPWAN technologies for large-scale IoT deployment. 2019, 5 1-7. DOI 10.1016/j.ict.2017.12.005.
- [15] *IEEE 802*. 2001-.
https://en.wikipedia.org/wiki/IEEE_802.
- [16] *ZigBee*. 2001-.
<https://cs.wikipedia.org/wiki/ZigBee>.
- [17] *Bluetooth*. 2001-.
<https://cs.wikipedia.org/wiki/Bluetooth>.

- [18] *Wi-Fi*. 2001-.
<https://en.wikipedia.org/wiki/Wi-Fi>.
- [19] *802.11 versions*. 1981- 2023.
<https://www.pcmag.com/encyclopedia/term/80211-versions>.
- [20] *Využívání informačních a komunikačních technologií v domácnostech a mezi osobami - 2022*. 2022.
<https://www.czso.cz/documents/10180/164606768/0620042201.pdf/5699654d-a722-44c9-a5e8-80443c89be18?version=1.1>.
- [21] Win Hlaing, Somchai Thepphaeng, Varunyou Nontaboot, Natthanan Tangsunantham, Tanayoot Sangsuwan a Chaiyod Pira. Implementation of WiFi-based single phase smart meter for Internet of Things (IoT). *2017 International Electrical Engineering Congress (iEECON)*. 2017, 1-4. DOI 10.1109/IEECON.2017.8075793.
- [22] *Low-Power Wi-Fi Ideal for Energy Efficient IoT Devices*. 2023.
<https://www.silabs.com/blog/low-power-wi-fi-Ideal-for-energy-efficient-iot-devices>.
- [23] Benny Har-Even. *Wi-Fi 6 For Low-Power IoT*. 2013-2023.
<https://semiengineering.com/wi-fi-6-for-low-power-iot/>.
- [24] Tomáš Přeučil a Martin Novotný. *Evaluation of power saving methods for low-power WiFi environment sensors*. In: *2022 11th Mediterranean Conference on Embedded Computing (MECO)*. 2022. 1-5.

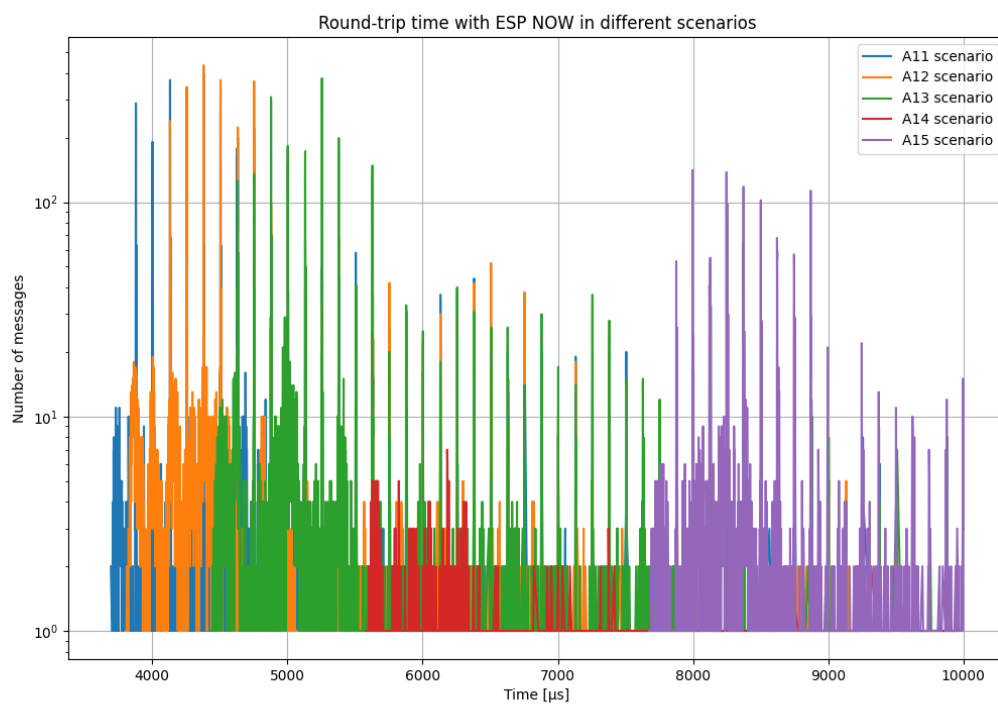
Příloha A

Slovníček pojmů

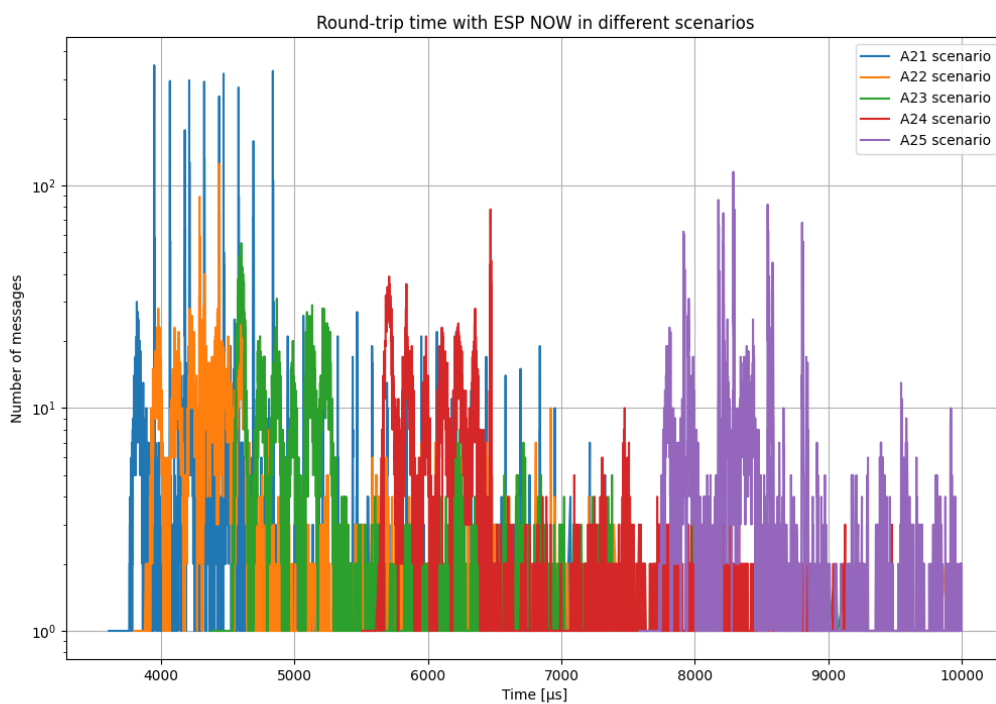
AI	■	Artificial Intelligence, umělá inteligence
AP	■	access point
CSS	■	Chirp Spread Spectrum
IEEE	■	Institute of Electrical and Electronics Engineers
IoT	■	Internet of Things, Internet věcí
ISM	■	industrial, scientific and medical
LPPAN	■	Low-power privat-area network
LPWAN	■	Low-power wide-area network
MCU	■	mikrokontroler
SoC	■	System on Chip
UNB	■	Ultra Narrow-Band

Příloha B

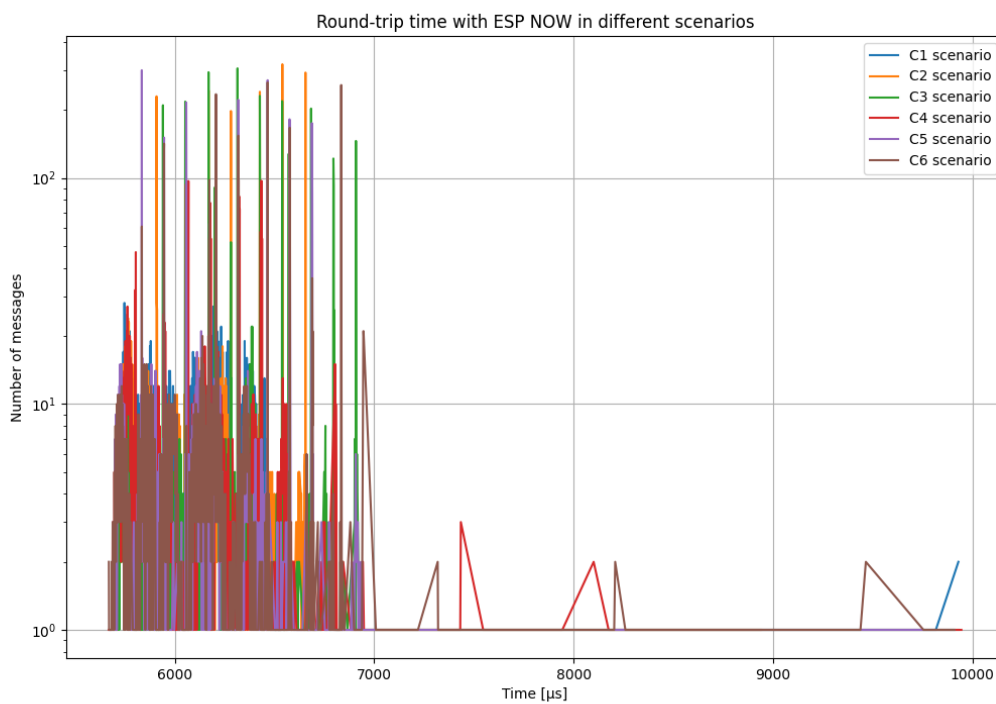
Grafy z měření



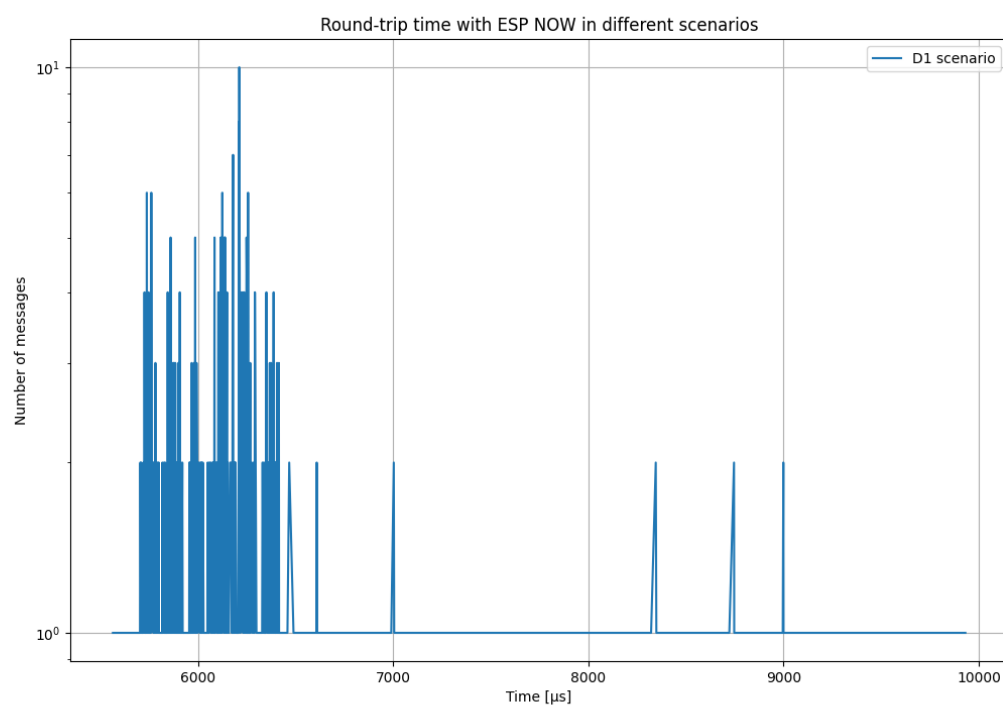
Obrázek B.1. Graf výsledků měření scénářů A1.



Obrázek B.2. Graf výsledků měření scénářů A2.



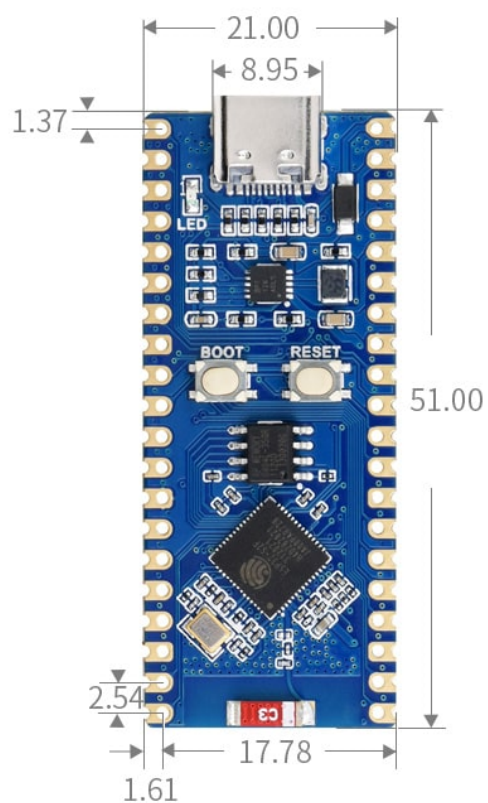
Obrázek B.3. Graf výsledků měření scénářů C.



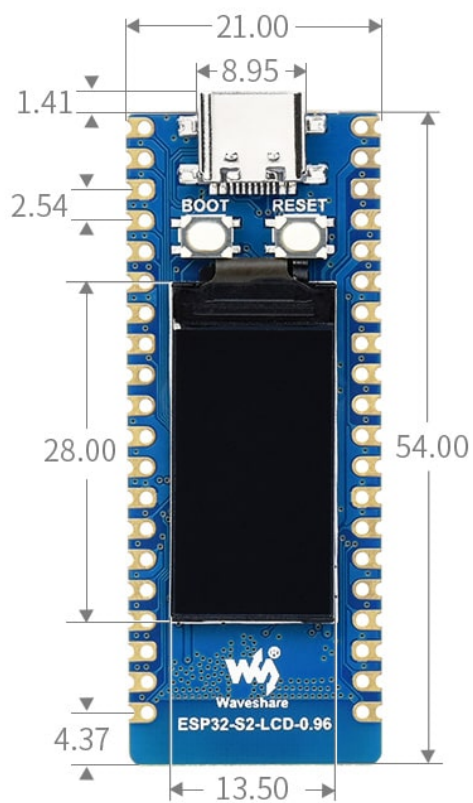
Obrázek B.4. Graf výsledků měření scénáře D.

Příloha C

Schéma modulů



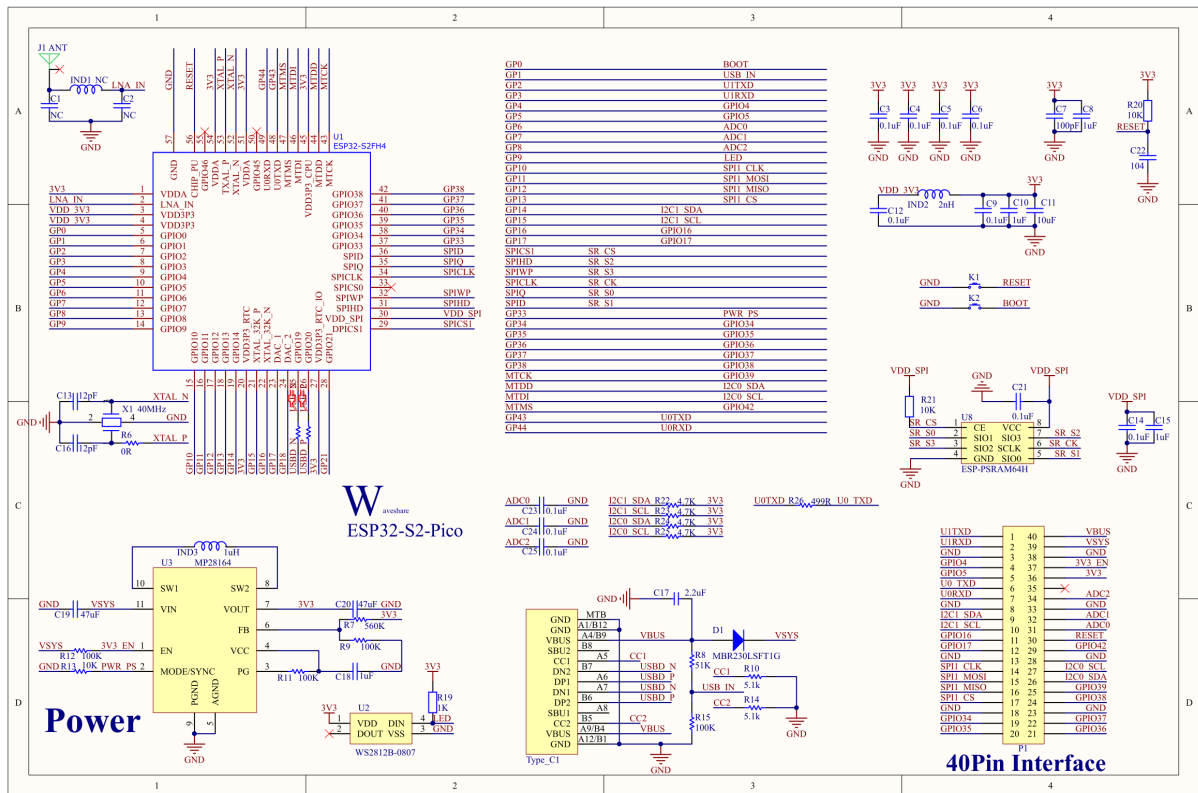
▲ ESP32-S2 Standard Ver.



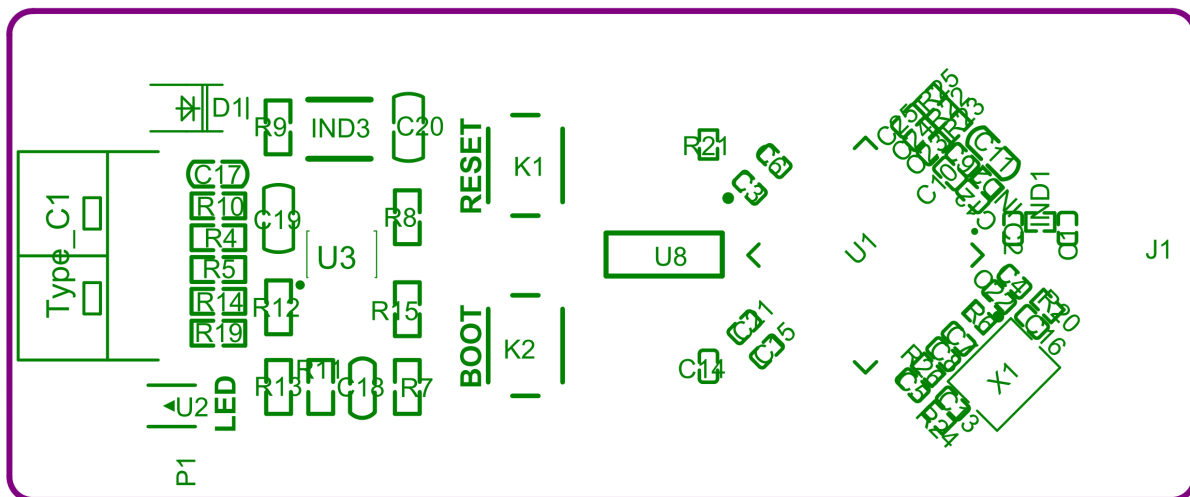
▲ ESP32-S2 LCD Ver.

Unit: mm

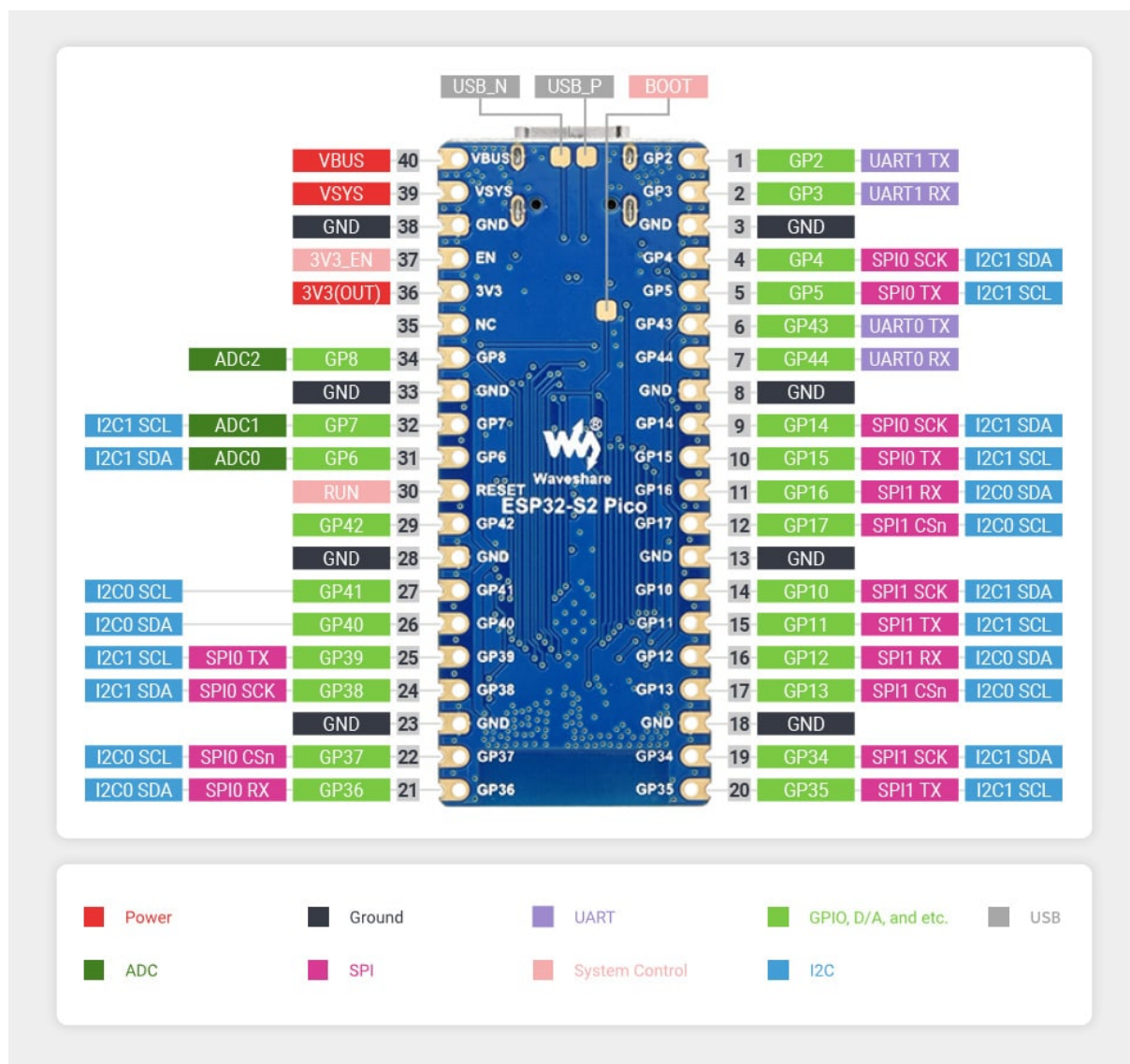
Obrázek C.5. Rozměry ESP32-S2-Pico [??].



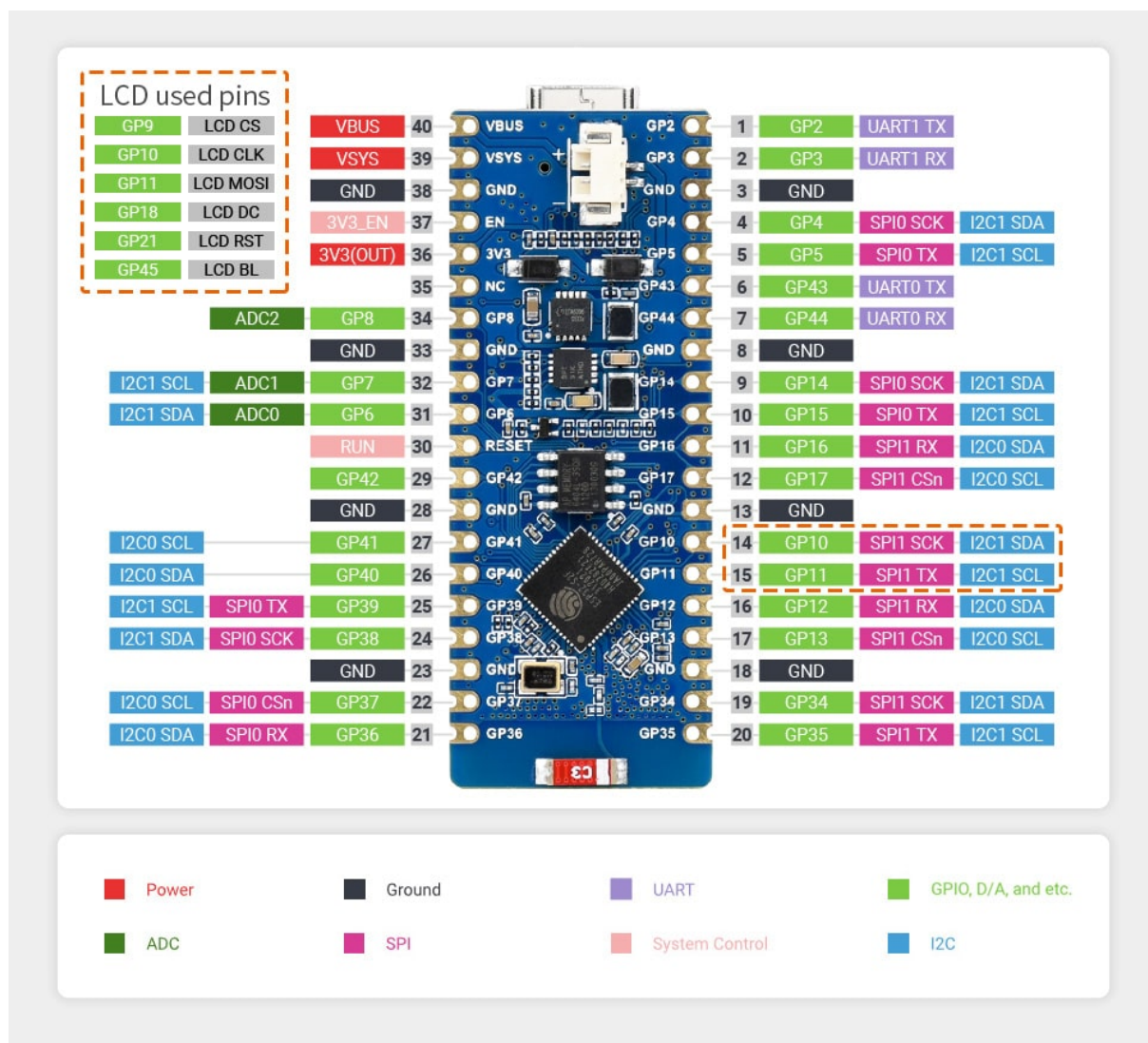
Obrázek C.6. Schéma ESP32-S2-Pico [??].



Obrázek C.7. Schéma rozložení ESP32-S2-Pico [??].



Obrázek C.8. Schéma pinu ESP32-S2-Pico [??].



Obrázek C.9. Schéma pinu ESP32-S2-LCD [??].