



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INFORMAČNÍCH SYSTÉMŮ

DEPARTMENT OF INFORMATION SYSTEMS

**NÁVRH ÚSPORNÝCH IOT SENZORŮ A SÍŤE CHYTRÉ
DOMÁCNOSTI**

POWER EFFICIENT SMART HOME IOT NETWORK AND SENSORS

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

VEDOUcí PRÁCE

SUPERVISOR

GABRIELA CHMELAŘOVÁ

Ing. JAN PLUSKAL

BRNO 2019

Zadání bakalářské práce



21338

Studentka: **Chmelařová Gabriela**
Program: Informační technologie
Název: **Návrh úsporných IoT senzorů a sítě chytré domácnosti**
Power Efficient Smart Home IoT Network and Sensors
Kategorie: Počítačové sítě

Zadání:

1. Vyhledejte a teoreticky porovnejte dostupné HW i SW platformy pro vytvoření IoT senzorů. Zaměřte se na možnosti úspory energie při provozu senzorů na baterii, jako jsou např. úsporné režimy MCU a různé typy síťových rozhraní.
2. Navrhněte alespoň čtyři typy senzorů určených pro domácí síť na základě prozkoumaných platform, zvolených po konzultaci s vedoucím.
3. Navrhněte síťové topologie, alespoň tři, pro testovací množinu zařízení s cílem minimalizovat spotřebu energie při zasílání informací ze senzorů.
4. Implementujte senzory na navržených vývojových platformách a síťových topologiích.
5. Proveďte optimalizace řešení vzhledem ke spotřebě energie. Změřte spotřebu a vyhodnoťte teoretickou výdrž vzhledem ke kapacitě baterií, které se s danými typy zařízení běžně používají.

Literatura:

- Blum, J. (2013). *Exploring Arduino: tools and techniques for engineering wizardry*. John Wiley & Sons.
- Gill, K., Yang, S. H., Yao, F., & Lu, X. (2009). A zigbee-based home automation system. *IEEE Transactions on Consumer Electronics*, 55(2).
- Gomez, C., & Paradells, J. (2010). Wireless home automation networks: A survey of architectures and technologies. *IEEE Communications Magazine*, 48(6).
- Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. *Future generation computer systems*, 29(7), 1645-1660.

Pro udělení zápočtu za první semestr je požadováno:

- Body 1, 2 a 3.

Podrobné závazné pokyny pro vypracování práce viz <http://www.fit.vutbr.cz/info/szz/>

Vedoucí práce: **Pluskal Jan, Ing.**
Vedoucí ústavu: Kolář Dušan, doc. Dr. Ing.
Datum zadání: 1. listopadu 2018
Datum odevzdání: 15. května 2019
Datum schválení: 28. října 2018

Abstrakt

Práce pojednává o nalezení vhodného řešení při konstrukci úsporných IoT senzorů pro domácí síť. Řešení se zabývá návrhem a sestrojením zařízení provozovaných na bateriích, s účelem zajištění jejich dlouhodobého provozu. Součástí textu je popis technologií umožňující implementaci úsporných senzorů. Dále je v práci popsán návrh a sestrojení úsporných zařízení a topologií. Poslední částí je popis optimalizace senzorů za účelem úspory energie, přehled naměřených hodnot a porovnání dosažených výsledků.

Abstract

This thesis deals with finding a suitable solution for the construction of energy saving IoT sensors for home network. The solution deals with design and construction of devices operated on batteries in order to ensure their long-term service. The text includes an overview of technologies enabling implementation of power efficient sensors. Furthermore, the thesis describes the design and construction of energy-saving devices and topologies. The last part deals with sensor optimization for energy saving, overview of measured values and comparison of achieved results.

Klíčová slova

IoT senzor, chytrá domácnost, vestavěné systémy, mikrokontrolér, ESP32, ESP8266, síťové topologie, úspora energie, bezdrátová senzorová síť, WSN, MQTT, ESP-NOW, LoRa, BLE, 6LoWPAN, LoRaWAN, mesh topologie, star topologie

Keywords

IoT sensor, smart home, embedded systems, microcontroller, ESP32, ESP8266, network topology, energy saving, wireless sensor network, WSN, MQTT, ESP-NOW, LoRa, BLE, 6LoWPAN, LoRaWAN, mesh topology, star topology

Citace

CHMELAŘOVÁ, Gabriela. *Návrh úsporných IoT senzorů a sítě chytré domácnosti*. Brno, 2019. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Jan Pluskal

Návrh úsporných IoT senzorů a sítě chytré domácnosti

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracovala samostatně pod vedením pana Ing. Jana Pluskala. Uvedla jsem všechny literární prameny a publikace, ze kterých jsem čerpala.

.....
Gabriela Chmelařová
31. července 2019

Poděkování

Děkuji vedoucímu práce Ing. Janu Pluskalovi za jeho cenné rady, důsledný přístup a vstřícnost při konzultacích, které značně přispěly k dokončení této práce.

Obsah

1	Úvod	3
2	Současné HW a SW platformy pro konstrukci IoT senzorů	4
2.1	Zařízení pro tvorbu IoT senzorů	5
2.1.1	Úsporné režimy mikrokontrolérů	5
2.1.2	Vývojové platformy pro práci se senzory	6
2.1.3	Komerční desky	10
2.1.4	Zařízení určená k napájení senzorů	15
2.1.5	Snímače veličin	16
2.2	Síťové rozhraní, protokoly a jejich vazba na úsporu energie	22
2.2.1	Vícevrstevní protokoly	22
2.2.2	Protokoly aplikační vrstvy	23
2.2.3	Protokoly síťové vrstvy	24
2.2.4	Protokoly linkové vrstvy	24
2.2.5	Protokoly fyzické vrstvy	25
2.3	Topologie sensorových sítí	25
3	Návrh senzorů určených pro domácí síť a úsporných síťových topologií	29
3.1	Senzory určené pro snímání veličin prostředí	29
3.2	Návrh úsporných síťových topologií	34
4	Implementace zvolených senzorů nad navrženými topologiemi	38
4.1	Implementace topologie č. 1	40
4.2	Implementace topologie č. 2	42
4.3	Implementace topologie č. 3	43
5	Měření a optimalizace spotřeby, výpočet teoretické výdrže zařízení na bateriích	46
5.1	Optimalizace implementovaných senzorů vzhledem k spotřebě energie	46
5.2	Měření spotřeby při provozu senzorů	49
5.3	Výpočet teoretické výdrže zařízení na základě naměřených hodnot	52
6	Závěr	59
	Literatura	60
	Přílohy	63
A	Tabulka výsledných naměřených hodnot senzorů	64

B	Tabulka aplikace optimalizací zařízení	65
C	Tabulka hodnot naměřených při podmíněném zasílání zpráv	66
D	Obsah přiloženého CD	67
	Slovník pojmů	68

Kapitola 1

Úvod

V dnešní uspěchané době, roste potřeba kontrolovat a ovládat okolní prostředí. Uspokojení této potřeby umožňuje zavedení chytrých senzorů do domácností. Často se ale potýkáme se špatnými podmínkami pro připojení těchto zařízení do rozvodů elektrické energie, či rozvod el. energie není k dispozici vůbec. Řešení je v použití senzorů napájených alternativními zdroji elektrické energie. K využití takového řešení je třeba sestrojít senzory šetrné k provozu na zdrojích s omezenou kapacitou.

Cílem mé práce je najít způsob, jak snížit spotřebu energie u senzorů provozovaných na baterii a tím zvýšit jejich použitelnost ve větším rozsahu. K dosažení tohoto cíle je třeba senzory vhodně navrhnout po hardwarové i softwarové stránce a zajistit u nich co nejúspornější vzájemnou komunikaci [2]. Je tedy kritické se obeznámit se zařízeními používanými pro tvorbu úsporných senzorů a se současnými technologiemi zajišťujícími úsporný provoz a komunikaci zařízení.

Struktura práce je blíže popsána v následujícím odstavci. V kapitole 2 jsou porovnány současné hardwarové platformy a komponenty, vhodné pro sestrojení úsporných senzorů. Dále tato kapitola poskytuje přehled existujících úsporných technologií zajišťujících vytvoření senzorové sítě zařízení provozovaných na bateriích. Kapitola 3 se zabývá návrhem zařízení a konceptů sítí, založených na poznacích popsaných v kapitole 2. Obsahem kapitoly 4 je implementace zvolených senzorů a realizace navržených sítí s popisem využitých technologií pro jejich sestrojení. Kapitola 5 se zabývá měřením spotřeby elektrické energie implementovaných senzorů a následně optimalizací s cílem maximalizace provozuschopnosti vytvořených zařízení a sítí. V této kapitole jsou také shrnuty výsledky měření a na jejich základě jsou vytvořeny odhady teoretické výdrže senzorů při provozu na baterii.

Kapitola 2

Současné HW a SW platformy pro konstrukci IoT senzorů

Současné technologické pokroky v oblasti nízkonapěťových integrovaných obvodů a bezdrátové komunikaci nám umožňují sestavit úsporná, levná a malá zařízení, která jsou aplikovatelná ke snímání vlastností prostředí. Tato zařízení jsou schopna vedle měření také vykonávat požadované operace a na krátké vzdálenosti si mezi sebou zasílat zprávy. Jednotlivá zařízení nazýváme *uzly* (anglicky *nodes*), které jako celek tvoří bezdrátové sítě senzorů (*WSN* — *Wireless Sensor Networks*) [22]. V rámci sítě jsou **senzory** propojeny do **topologií**, které jim zajišťují různé atributy, jako je například stabilita, nízká spotřeba nebo dosah sítě. Tyto vlastnosti jsou vybírány na základě požadavků na síť a možností v závislosti na použitých zařízeních.

Základem monitorovací sítě jsou hardwarové komponenty. Ty jsou zastoupeny jednotlivými uzly, které se skládají z výpočetní jednotky, vysílacích prvků, jednotlivých snímacích rozhraní a zdroje elektrické energie. Tyto komponenty lze zakoupit zvlášť a zařízení z něj sestavit, nebo si lze pořídit již hotové vývojové desky. Tyto platformy mohou obsahovat již všechny potřebné součásti, nebo je lze případně rozšířit o další přípojné moduly, které zařízení poskytují přidanou funkcionalitu.

Druhou neodmyslitelnou částí při konstrukci WSN je vytvoření propojení jednotlivých uzlů a ustavení jejich komunikace. V tomto kroku je klíčová volba správné technologie, na jejímž základě budou zařízení komunikovat. Dalším důležitým opatřením je volba vhodného rozložení jednotlivých uzlů v prostoru a cest, kterými budou zasílány zprávy od koncových zařízení k centrálnímu bodu sítě. Volba vhodné topologie zajistí dlouhodobou životnost a dobrou rozšiřitelnost sítě.

Uzly ve vytvořené síti potřebují mezi sebou komunikovat a zasílat data na jednotlivé stanice vzdálené jeden či více tzv. „hopů“. **Protokolový zásobník** použité technologie nebo případná kombinace technologií by měla být schopna komunikovat mimo lokální síť přes propojovací uzel (též nazýván jako *gateway*), který slouží jako spojení mezi lokální podsítí WSN a vnější sítí [20]. Je třeba myslet na to, že vytvořená síť může dynamicky měnit svou kvalitu kvůli možnému přesunu uzlů, náhlým překážkám, vzniku rušení nebo při vybití zařízení [19]. Často dochází k odpojení uzlu, které způsobí zhoršenou dostupnost ostatních zařízení v síti. Na základě této skutečnosti je třeba volit vhodné propojení zařízení, které omezí případné nedostatky sítě.

V následující podkapitole 2.1 jsou uvedeny charakteristiky zařízení, které jsou základem pro vytvoření jednotlivých uzlů. Pohled je přiblížen na hotové vývojové platformy

a jejich přídavné komponenty. Součástí popisu je sekce 2.1.1, která zahrnuje seznámení se základními úspornými režimy hardwarových platform. Podkapitola 2.2 obsahuje přehled bezdrátových komunikačních technologií a protokolů využívaných při komunikaci zařízení v sensorových sítích. Na závěr první kapitoly je v podkapitole 2.3 uveden přehled topologií, běžně využívaných pro vytvoření bezdrátové sensorové sítě.

2.1 Zařízení pro tvorbu IoT senzorů

Senzory určené ke snímání fyzikálních vlastností lze získat několika způsoby. Nejrychlejší variantou jsou předem sestavené komerční produkty, které stačí pouze zprovoznit. Ty ale nemusí vyhovovat specifickým požadavkům a většinou jsou drahé. Kromě hotových produktů, je možnost vytvořit vlastní senzor dle potřeb uživatele. Základem pro sestavení vlastního zařízení je **mikrokontrolér** (zkráceně MCU). Mikrokontrolér je programovatelnou hardwarovou součástí, která umožňuje ovládat elektrické, mechanické nebo softwarové systémy [5]. Senzory jsou dále pro účely komunikace vybavené úspornými **rádio frekvenčními (RF) vysílači**, které umožňují zařízením mezi sebou komunikovat. Výhoda RF komunikace spočívá v jednoduchém přidávání a odebrání zařízení v síti bez potřeby fyzického propojení kabely. Bezdrátová komunikace tím zajišťuje nižší náklady na zavedení senzorů [21]. Dále je třeba, aby zařízení obsahovala paměť pro uložení a vykonávání řídicího programu. Zařízení obsahují také rozhraní (například v podobě mikro USB), která umožňují napájení senzoru a případné zavedení řídicího programu. Zařízení mohou být napájena také jiným způsobem, například přes baterie, k tomu je ale zapotřebí speciálních konektorů pro jejich připojení.

Výsledná zařízení lze buď sestavit z výše popsaných komponent, nebo lze použít sestavené vývojové platformy. K hotovým platformám lze dále připojovat tzv. **moduly** nebo **shieldy**. Jsou to elektronické součástky, které poskytují zařízením určitou funkcionalitu, například v podobě Wi-Fi připojení nebo umožnění napájení přes baterie. Ve formě modulů se často vyskytují také jednotlivé snímače hodnot.

V následujících podkapitolách jsou popsány konkrétní zařízení umožňující sestavení chytrých senzorů. Úvodem v podkapitole 2.1.1 jsou popsány základní režimy spánku mikrokontrolérů, které při zavedení šetří energii zařízením. Podkapitola 2.1.2 obsahuje popis dvou hlavních vývojových platform, od firem Arduino.cc a Espressif Systems. Podkapitola 2.1.3 poté popisuje a srovnává konkrétní hotové vývojové platformy, použitelné k sestavení úsporných IoT senzorů. V podkapitole 2.1.4 jsou sepsány komponenty, které lze použít pro zajištění zdroje napájení platformy. V poslední podkapitole 2.1.5 je vypsán přehled modulů obsahujících snímače vlastností prostředí a charakteristiky samotných snímačů.

2.1.1 Úsporné režimy mikrokontrolérů

U většiny modulů nelze upravovat hardware za účelem úspory energie, je tedy možné modifikovat pouze software zařízení. Pro tento účel slouží tzv. režimy spánku (anglicky *sleep modes*). Jde o úsporné režimy, kdy dojde k dočasnému přerušení činnosti určitých částí mikrokontroléru, za účelem snížení spotřeby energie zařízením. Využívání úsporných režimů mikrokontrolérů je jedním z hlavních prostředků pro šetření baterie při provozu senzorů. Všechny dostupné vývojové platformy popsané v podkapitole 2.1.3 umožňují alespoň některé ze spánkových módů. Úroveň úsporných režimů může být několik, liší se množstvím deaktivovaných procesů mikrokontroléru.

Modem-sleep

Modem-sleep je standardním úsporným režimem využívaným mikrokontroléry. Při tomto režimu je deaktivován Wi-Fi modul, dokud není znovu „probuzen“ přijetím požadavku. Pokud není tento úsporný mód dále modifikován, je zařízení nastaveno na automatické buzení, není tedy třeba dodatečné uživatelské konfigurace. Režim lze také vynutit vypnutím **RF vysílače**. Modem-sleep mód je využíván především u aplikací, které potřebují, aby byl procesor zařízení stále v provozu, například **PWM** světlo. Odběr proudu se při použití modem-sleep režimu se například u mikrokontroléru ESP8266 pohybuje kolem 10–20 mA [14].

Light-sleep

Úsporný režim light-sleep umožňuje při aktivaci kromě odpojení Wi-Fi také deaktivaci většiny paměti RAM, CPU, hodin digitálních periférií a je sníženo napájecí napětí. Během light-sleep módu je odpojeno CPU a nereaguje tedy na signály a přerušení z hardwarových periférií. Proto je třeba pro aktivaci zařízení nastavit tzv. *trigger*, tedy událost, která vyvolá přerušení spánkového režimu. U čipu ESP8266 lze zařízení převést z režimu spánku například přes externí pin „**GPIO**“. Je ale třeba tuto službu aktivovat příslušnou funkcí. ESP32 lze „budit“ například zaznamenáním přerušení od snímače dotyku. Po ukončení light-sleep módu spustí CPU a periférie opět svoje operace a mohou pokračovat v činnosti ze stejného místa, odkud byly převedeny do režimu spánku (vnitřní stav je zachován). Průměrný odběr proudu v light-sleep módu je například u čipu ESP8266 0,5 mA [14].

Deep-sleep

Nejvíce úsporným režimem je deep-sleep mód, do kterého nemůže zařízení, na rozdíl od předchozích dvou módů, přejít automaticky. U čipů ESP (blíže popsanych v sekci 2.1.2) je možno tento mód aktivovat po zavolání odpovídající funkce. V režimu deep-sleep vypíná čip CPU, většinu paměti RAM a všechny digitální periférie. V provozu zůstává pouze **RTC** modul a jeho paměť, který je zodpovědný za periodickou aktivaci zařízení. Pokud je třeba zachovat důležitá data z předchozího provozu zařízení, je možné je uložit do paměti RTC. U čipů ESP je možné nastavit různé způsoby deaktivace úsporných režimů, například po uběhnutí zvoleného časového intervalu, či aktivací přes přivedený signál na pin „**GPIO**“. Co se spotřeby v režimu deep-sleep týče, například u čipu ESP8266 je zaznamenán odběr proudu 10–20 μ A [14].

2.1.2 Vývojové platformy pro práci se senzory

Tato podkapitola se věnuje dvěma rodinám hardwarových platform, jejichž zařízení a komponenty jsou základem většiny chytrých senzorů. Jedná se o platformy firem Arduino.cc a Espressif Systems, které jsou charakterizované v následujících sekcích. Oba výrobci poskytují hardwarové a softwarové zázemí pro tvorbu chytrých zařízení určených například ke snímání fyzikálních vlastností prostředí.

Vývojová platforma Arduino

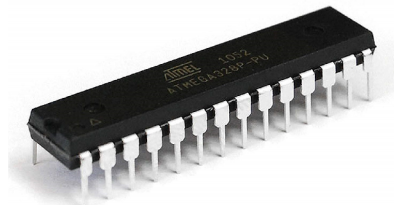
Společnost *Arduino.cc* navrhuje a vytváří vývojové desky, které slouží pro vlastní konstrukci chytrých zařízení. Fyzické platformy, které produkují, se nazývají jako firma Arduino. Platformy Arduino jsou založené na mikročipu od firmy Atmel. Většina vývojových desek používá čip ATMega 328p [10], který používá například i jeden z jejich nejznámějších produktů

Arduino UNO (viz sekce 2.1.3). Mikročip skýtá 32 kB paměti flash, která pracuje při napětí 1,8–5,5 V. Mikročip ATmega 328 podporuje sběrnici SPI a I2C, užitečnou při komunikaci se snímači veličin. Komponenta disponuje 28 piny (viz obrázek 2.1). Mikročip pracuje na napětí 2,7–5,5 V. Jeho rychlost je udávána na 8 MHz při napětí 2,7–5,5 V a 16 MHz při napětí 4,5–5,5 V. Při provozní frekvenci 4 MHz a napětí 3 V odebírá při aktivním režimu 1,5 A a při režimu spánku 1 μ A.

Firma se dále podílí na vývoji multiplatformního grafického rozhraní *Arduino IDE*¹, které slouží k vytvoření řídicího programu hardwarových platforem, primárně zařízení Arduino. Díky možnosti nahrávání knihoven pro správu platforem a přípojných modulů do IDE je výrazně usnadněn vývoj chytrých zařízení. Tím je také umožněno spravování řídicích programů platforem i od jiných výrobců, což značně přidává na popularitě tohoto nástroje při vývoji chytrých zařízení.

Vývojové platformy Arduino jsou **open-source** [5], což umožňuje dalším firmám vytvářet tzv. **klony**, tedy hardwarové kopie těchto zařízení. Různé firmy také vydávají rozšiřovací hardware na vývojové desky nazývané **shield**. Mezi nejznámější Arduino shieldy patří například Wi-Fi Shield, rozšiřující Arduino o možnost Wi-Fi komunikace. Dále je využíván například Ethernet Shield, který umožňuje desce Arduino připojení k internetu přes ethernetový kabel.

Vývojová deska Arduino je programovatelná přes mikro USB port, přes který lze zároveň platformu napájet. Tím je značně usnadněn vývoj chytrých zařízení, jelikož odpadá potřeba speciálního na míru vytvořeného hardware se speciálním přípojným vodičem [10]. Společnost poskytuje různé druhy dodávaného hardware. Jedním typem jsou tzv. jednodeskové mikrokontroléry, kdy jsou všechny komponenty integrovány na jednu vývojovou platformu. Další možností je mikrokontrolérový kit, kdy je s vývojovou deskou dodáváno množství dalších komponent, například snímačů, které lze k desce dle potřeby připojovat. Tím je umožněno snadno sestavit senzory dle potřeby z navzájem kompatibilních zařízení.



Obrázek 2.1: Mikročip ATmega 328P. (Obrázek přejat z: <https://robu.in/product/atmega328p-pu-pdip-28-microcontroller/>)

Vývojová platforma Espressif Systems

Společnost Espressif Systems vytváří hardwarové a softwarové prostředky pro tvorbu chytrých zařízení. Po hardwarové stránce se zabývá především vývojem mikročipů ESP a na nich založených platforem. Mezi zástupce patří například platforma ESP32-DevKit, která je dále popsána v sekci 2.1.3. Dále firma po hardwarové stránce produkuje přípojné moduly ke svým platformám. Co se týče softwaru, vyvíjí společnost vlastní **SDK** (*Software Development Kit*) k vývoji programů nad čipy ESP. Dále také společnost vyvíjí vlastní komunikační protokol ESP-NOW, který je dále popsán v sekci 2.2.1.

¹Arduino IDE — <https://www.arduino.cc/en/main/software>

Velmi důležitým produktem společnosti jsou čipy ESP. Ty v základním složení obsahují moduly umožňující RF komunikaci, čímž se liší od mikrokontrolérů platforem Arduino. Firma vyvinula dva přední zástupce čipů ESP — ESP8266 a ESP32. Tyto mikrokontroléry jsou součástí značného množství volně prodejných vývojových platforem. Čipy ESP umožňují zařízením komunikovat přes technologie Wi-Fi (ESP8266) nebo i Bluetooth (ESP32). Je možné u nich při provozu využít úsporné režimy popsané v sekci 2.1.1. V následujících sekcích jsou jednotlivé čipy blíže specifikovány, především z hlediska poskytovaných služeb a realizace úsporných režimů.

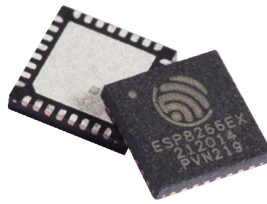
ESP8266

Mikročip ESP8266 je úsporným čipem používaným při konstrukci chytrých senzorů. Jeho hlavní předností je zprostředkování komunikace přes technologii Wi-Fi. Je součástí komerčních vývojových platforem, například je základem desky **WeMos D1 mini**. Jelikož Arduino platformy v základním složení neobsahují Wi-Fi modul (například deska **Arduino UNO**), je možné využít čip ESP8266 ve formě přípojného modulu ke zprostředkování Wi-Fi komunikace, čímž ve výsledku poskytuje levnější řešení oproti oficiálnímu Arduino Shield. Čip ESP8266, který nese také označení ESP8266EX (viz obrázek 2.2), pracuje na frekvenci 80 MHz, kterou lze zvýšit až na 160 MHz.

Za účelem úspory energie při provozu je možné použít u mikrokontroléru ESP8266 následující úsporné režimy. V základním nastavení čip přechází do automatického modem-sleep režimu (viz sekce 2.1.1). K úspoře energie se zavádí režim light-sleep, kdy jsou navíc u čipu vypínány systémové hodiny a je pozastaveno CPU, které je uvedeno do stavu nečinnosti nebo je úplně odstaveno. Druhým zaváděným režimem je deep-sleep, kdy zůstane v provozu pouze RTC (*Real Time Clock*).

Deep-sleep mód je nejvíce efektivním způsobem úspory energie při provozu platforem s čipem ESP8266. Délka převádění mikročipu do režimu spánku se pohybuje v řádu mikrosekund. Důležitým faktem je, že dle knihovny ESP8266 nelze mikročip převést do úsporného režimu na více než 71 minut [18]. Průběh vykonávání programu při zavedení úsporného režimu je následující. Nejprve zařízení pracuje v aktivním režimu, poté přejde do režimu spánku, odkud znovu přejde do aktivního režimu. Rozdíl je při použití light-sleep a deep-sleep režimu ten, že při „probuzení“ zařízení z light-sleep módu pokračuje řídicí program ve vykonávání tam, kde přestal, zatímco při použití deep-sleep režimu se řídicí program začne vykonávat od začátku.

Existuje několik obecně doporučovaných způsobů, jak efektivně využívat deep-sleep mód. v technickém dokumentu od společnosti Espressif Systems je jich popsáno celkem 8. Obecným doporučením ohledně spánkových módů je, že pokud má být zařízení usnáno na méně než 2 vteřiny, je pro něj vhodnější light-sleep mód a pro více jak 2s tak deep-sleep mód [14].



Obrázek 2.2: Mikročip ESP8266EX. (Obrázek přejat z <https://www.amazon.com/ESP8266EX-Tiny-Wireless-802-11-GC-ESP8266EX-10/dp/B078VQ5G65>)

ESP32

Dalším čipem společnosti Espressif je mikročip ESP32, který je nástupcem mikročipu **ESP8266**. Čip obsahuje navíc oproti ESP8266 podporu Bluetooth verze 4.2 a technologie BLE (blíže popsané v sekci 2.2.4), která umožňuje zařízení minimalizovat nároky na spotřebu energie během komunikace.

Mezi konkrétní zástupce čipů, které jsou součástí platform vyžitých v této práci je čip ESP32-D0WDQ6 (viz obrázek 2.3). Jedná se o dvoujádrový mikročip, který neobsahuje žádnou vestavěnou paměť flash. Jeho maximální rychlost je 240 MHz. Čipy ESP32-D0WDQ6 mohou být součástí aplikovatelných modulů. Patří sem například modul ESP-WROOM-32 (zobrazený na obrázku 2.4).

ESP32 umožňuje také provoz úsporných režimů, popsanych v sekci 2.1.1. Základním módem je modem-sleep a mezi aplikovatelné režimy patří light-sleep a deep-sleep. Módy zde fungují podobným způsobem, jako u ESP8266 [13].

V light-sleep a deep-sleep módu jsou vypnuty bezdrátové periferie. Před přechodem do úsporných režimů musí aplikace vypnout Wi-Fi a Bluetooth příslušnými funkcemi. Avšak bezdrátová spojení nejsou udržována ani pokud nejsou tyto funkce zavolány. Pokud je potřeba mít aktivní Wi-Fi připojení, je třeba zapnout Wi-Fi modem-sleep režim a k němu aktivovat automatický light-sleep režim. Toto nastavení umožní systému automatické aktivování z režimu spánku vyvolané ovladačem Wi-Fi, čímž se udržuje spojení s přístupovým bodem.

Z režimu deep-sleep může být zařízení aktivováno několika způsoby. První možností je probuzení pomocí časovače. Před uvedením zařízení do režimu spánku je nastaven interval délky úsporného režimu v řádu milisekund. Tato varianta nevyžaduje zapnuté RTC periferie nebo RTC paměti. Další možnosti se vztahují na RTC vstupně výstupní modul. Je možné vyvolat probuzení pomocí signálu přerušeni zasláného dotykovým senzorem. Nebo je možné sledovat požadovanou hodnotu zvoleného „GPIO“ pinu. Lze také monitorovat více pinů a aktivovat zařízení na základě funkcí, které nad výstupy vybraných pinů vykonají logické operace OR nebo AND.



Obrázek 2.3: Na obrázku je vidět čip ESP32-D0WDQ6, který je základem platformy firmy Espressif Systems. (Obrázek přejat z: <https://www.macnica.eu/node/267>)



Obrázek 2.4: Modul ESP-WROOM-32. Uprostřed modulu lze vidět, kde je uvnitř zakomponovaný čip ESP32-D0WDQ6. (Obrázek přejat z: <http://www.electronics-lab.com/getting-started-esp32/>)

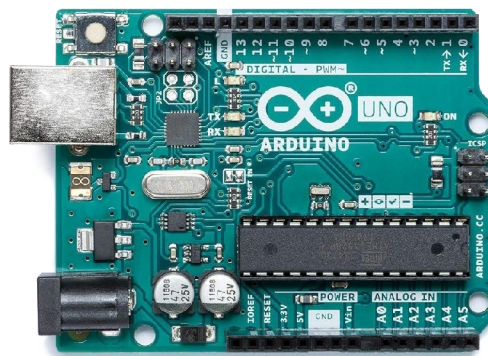
2.1.3 Komerční desky

V následující části je vytvořen přehled konkrétních vývojových platform. Jedná se o základní prvky senzorů. Platformy jsou založeny na mikročipech z podkapitoly 2.1.2. Důležitými aspekty porovnávání vývojových desek jsou například povolené napájecí napětí, možnosti napájení a podpora komunikačních technologií.

Arduino UNO

Arduino UNO je jedním ze základních zástupců vývojových desek od firmy Arduino.cc. Základem desky je mikrokontrolér ATmega. Vývojová deska disponuje 14 digitálními piny — z čehož 6 jich může být použito jako výstupy PWM. Dále obsahuje 6 analogových vstupů. Umožňuje připojení přes USB, což je vhodné zejména pro napájení a nahrání řídicího programu do zařízení. Platforma může být napájena 6–20 V a disponuje regulovanými výstupy poskytujícími 5 V a 3,3 V.

Co se týče komunikace umožňuje Arduino UNO sériovou komunikaci přes rozhraní UART, dostupnou na pinech 0 a 1. Existuje varianta s jiným typem mikrokontroléru, který umožňuje zasílat sériová data přes USB, které se pak zobrazí na virtuálním COM portu v aplikaci na PC. Odběr proudu činí 40 mA pro každý I/O pin a 50 mA pro 3,3 V pin [17].



Obrázek 2.5: Vývojová deska Aruino UNO – na obrázku lze vidět rozložení pinů na desce. (Obrázek přejat z: https://articulo.mercadolibre.com.ar/MLA-775167821-curso-arduino-desde-cero-a-avanzado-ejercicios-y-teorias-_JM?quantity=1)

ESP32-DevKit

Vývojová platforma ESP32 DevKit, vyrobená firmou MH-ET LIVE, je založena na mikročipu ESP32. Čip je součástí vestavěného modulu ESP32-WROOM-32² (viz obrázek 2.6), který je produkován přímo firmou Espressif. Verze ESP čipu v modulu je ESP32-D0WDQ6³, což je dvoujádrový mikročip pracující na frekvencích 80–240 MHz. Mikročip umožňuje vysílání dat přes Wi-Fi, Bluetooth a BLE. K napájení platformy je v desce zabudovaný mikro USB port typu B, který slouží také pro naprogramování platformy. V platformě je také zaveden regulátor napětí AMS1117⁴ a převodník USB/UART CP2104⁵. Deska obsahuje dvě diody (viz obrázek 2.6), z čehož jedna je nekontrolovatelná napájecí dioda a druhá dioda je ovladatelná a je připojená k pinu „GPIO2“.

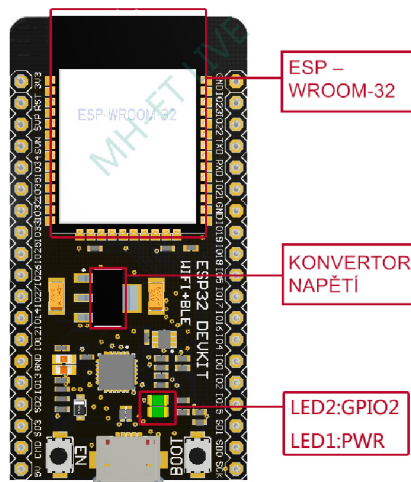
Vývojová deska dále disponuje tlačítkem pro reset (EN) systému. Dále obsahuje tzv. Boot tlačítko, které když je stisknuto zároveň s tlačítkem reset, tak se zařízení převede do stahovacího módu. Poté je možno stahovat firmware do zařízení přes sériový port. Dále se na desce nachází USB pro komunikaci s PC a pro napájení. Poskytuje také množství pinů, které umožňují využít různé funkcionality v podobě například ADC, DAC, PWM, I2C, a další. Popsané komponenty jsou vyobrazeny na obrázku 2.6.

²https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf

³https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf

⁴<http://www.advanced-monolithic.com/pdf/ds1117.pdf>

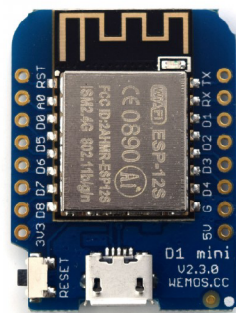
⁵<https://www.silabs.com/documents/public/data-sheets/cp2104.pdf>



Obrázek 2.6: Schéma vývojové platformy ESP32-DevKit — hlavním prvkem desky je ESP WROOM-32, dále deska obsahuje regulátor napětí a 2 LED diody. (Obrázek přejat z: <http://mh.nodebb.com/topic/2/mh-et-live-esp32-devkit>)

WeMos D1 mini

Jedná se o desku založenou na modulu obsahující čip ESP8266 (viz sekce 2.1.2). Tím umožňuje využití úsporných módů za účelem snížení spotřeby energie při napájení na baterii. Platformu s tímto čipem lze také využít díky téměř identickému rozložení pinů jako alternativu platformy Arduino UNO (viz podkapitola 2.1.3). Disponuje také zabudovanou flash pamětí o velikosti 4 MB⁶. Platforma obsahuje 11 digitálních vstupně-výstupních pinů a jedním analogovým vstupním pinem označeným „A0“. Pro komunikaci s PC a napájení obsahuje mikro USB.



Obrázek 2.7: Vývojová deska WeMos D1 mini (Obrázek přejat z: <https://www.roboterbausatz.de/974/wemos-d1-mini-arduino-kompatibles-board-mit-Wi-Fi-esp8266>)

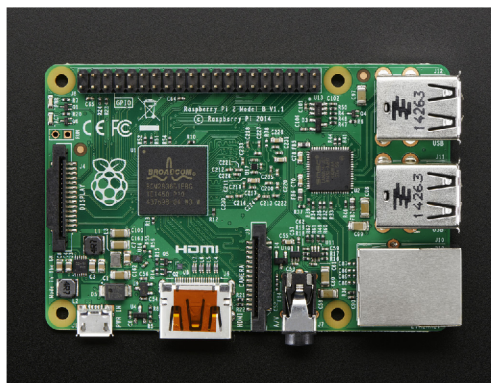
Raspberry Pi 2 - Model B

Raspberry Pi je levným malým jednodeskový počítač, který může v senzorových sítích zastupovat několik funkcí. Raspberry Pi 2 je založen na multijádrovém procesoru ARMv7,

⁶https://wiki.wemos.cc/products:retired:d1_mini_v3.0.0

obsahuje 1 GB paměti RAM a umožňuje připojení pomocí ethernetového kabelu, HDMI a čtyř USB portů. Dále umožňuje navýšit paměť přes SD kartu. Jeho spotřeba se pohybuje kolem 200 mA, v případě aktivní Wi-Fi se zvyšuje o 170 mA a při připojení ethernetu je navýšení spotřeby o 40 mA [1]. Je tedy vhodný pro zprostředkování například centrálního bodu v senzorové síti.

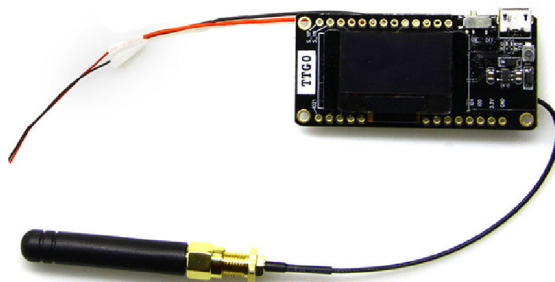
V senzorových sítích je Raspberry Pi vhodně využitelné jako server pro přijímání naměřených hodnot, nebo může sloužit sám jako senzor, díky několika zabudovaným snímačům, kterými disponuje.



Obrázek 2.8: Raspberry Pi 2 — Model B, přehled slotů. (Obrázek přejat z: <https://learn.adafruit.com/introducing-the-raspberry-pi-2-model-b?view=all>)

LILYGO TTGO LORA32

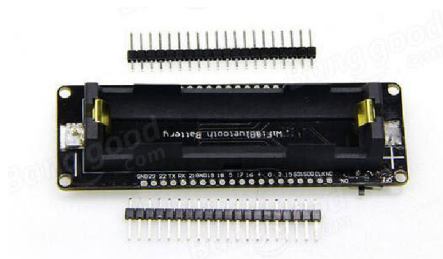
Jde o vývojovou desku založenou na čipu ESP32. Hlavním prvkem platformy je čip SX1276, který umožňuje komunikaci na vysoké vzdálenosti přes technologii LoRa, která je blíže specifikovaná v sekci 2.1.3. Komunikace pak probíhá na frekvenci 868 MHz pro Evropu. Zařízení obsahuje také anténu pro zvýšení dosahu signálu. V této verzi platformy lze desku napojit přímo na baterii, ale nelze přes ni baterii nabíjet. Dále disponuje komunikačním rozhraním SPI. Hardwarová platforma obsahuje také OLED display například k zobrazení měřených veličin. Zařízení pracuje na napětí 3,3–7 V [6].



Obrázek 2.9: LILYGO TTGO LORA32 — deska s připojenou anténou a napájením na baterii (Obrázek přejat z: <https://www.sensorae.com/eu/inicio/2363-modulo-ttgo-lora-v20-esp32.html>)

WeMos Battery ESP32 Development Tool

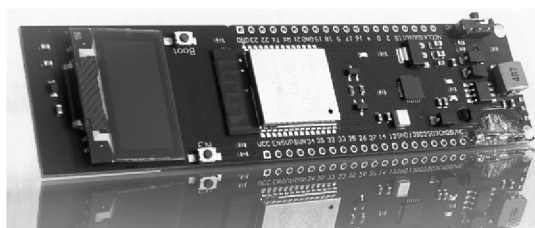
Vývojová deska od společnosti WeMos poskytuje připojení přes Wi-Fi, Bluetooth nebo BLE díky vestavěnému čipu ESP32 v modulu ESP-WROOM-32. Výhodou této vývojové desky je již vestavěný modul pro vložení a napájení na baterii. Například 3000 mAh baterie může zajišťovat funkčnost čipu ESP na 17 a více hodin. Obsahuje také přepínač ON/OFF na kontrolu napájení desky. Je schopná nabíjet svou baterii přes externí napájení, zároveň také při vykonávání jiných operací. Obsahuje také LED indikátor stavu nabíjení baterie (svítí zeleně pro nabitou baterii, červeně pro nabíjení) [8].



Obrázek 2.10: Zadní část WeMos Battery ESP32 - slot pro napájení baterií (Obrázek přejat z: <https://www.ezgiz.com/product/wemos-wi-fi-bluetooth-battery-esp32-development-tool/>)

TTGO Wi-Fi + Bluetooth Battery

Vývojová platforma společnosti TTGO, obsahuje propojení hned několika funkcionalit již zmíněných vývojových desek. Jejím základem je čip ESP32, čímž umožňuje Wi-Fi, Bluetooth a BLE konektivitu. Má OLED display jako **TTGO LoRa32**, který je připojen přes sběrnici I2C. Má také slot pro baterii, takže nepotřebuje externí napájení jako vývojová platforma **WeMos Development Tool**. Je také schopná baterii nabíjet pomocí externího napájení přes USB port. Při této aktivitě poskytuje stejnou identifikaci stavu nabíjení jako u **WeMos Development Tool**[7]. Poskytuje tedy potřebné funkční části z ostatních desek.

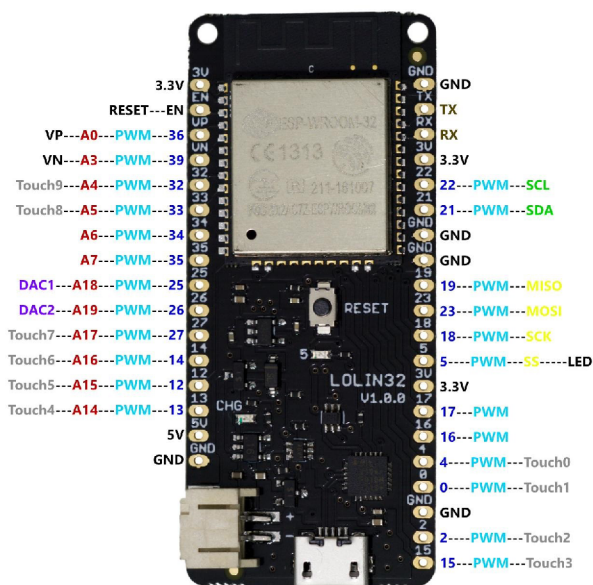


Obrázek 2.11: Vzhled vývojové platformy TTGO Wi-Fi + Bluetooth Battery (Obrázek přejat z: <http://forums.4fips.com/viewtopic.php?f=3&t=6905>)

WeMos Lolin32

Platforma Lolin32 patří pod vývojové desky společnosti WEMOS.CC. Hlavním prvkem desky je modul ESP-WROOM-32, který svým zabudovaným čipem ESP32 poskytuje podporu Wi-Fi a Bluetooth technologie. Platforma dále disponuje 4 MB flash paměti. Obsahuje 26 digitálních a 12 analogových pinů. Pracovní napětí platformy je stanoveno na 3,3 V.

Deska disponuje konektorem PH-2,0 mm⁷ pro napájení zařízení na baterii. Uprostřed desky je zabudované tlačítko pro restart platformy. Na desce jsou také dvě LED diody, jedna ovladatelná přes pin „IO5“ a druhá sloužící k detekci připojení platformy k napájení. Deska umožňuje komunikaci přes sběrnici I2C a SPI, přesné rozmístění pinů umožňujících tuto komunikaci je zobrazeno na obrázku 2.12.



Obrázek 2.12: Rozdělení pinů na vývojové desce Lolin32 od firmy WEMOS.CC. Na desce jsou zvláště piny pro komunikaci se sběrnici I2C (zeleně) a SPI (žlutě). Mimo jiné jsou zde zvláště červeně analogové piny a tmavě modře digitální piny. (Obrázek přejat z: <https://forum.micropython.org/viewtopic.php?t=5181>)

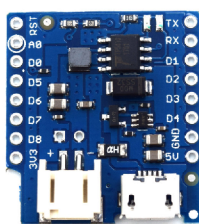
2.1.4 Zařízení určená k napájení senzorů

V následující sekci je vytvořen přehled zařízení, která slouží k umožnění napájení senzorů na baterii. Prvním zástupcem je WeMos Battery Shield, který je využíván především u platformy WeMos D1 mini (viz sekce 2.1.3). Druhým zařízením je modul aplikovatelný na nepájivé pole, na které lze následně připojovat jednotlivá zařízení. Bližší specifikace obou zařízení jsou popsána v následujících sekcích.

WeMos Battery Shield

Tento modul umožňuje vývojovým deskám užívat napájení z baterie. Rozložení pinů přesně sedí na vývojovou desku WeMos D1 mini, je tedy jednoduše aplikovatelný přímo pro tuto platformu. Umožňuje také případné nabíjení připojené baterie pomocí externího zdroje připojeného přes mikro USB. Baterii je možné připojit i přímo pomocí konektoru PH2-2,0 mm. Je možné i baterii přes shield nabíjet, a to napětím max 10 V (doporučuje se 5 V). Vhodné baterie na napájení jsou lithiové s napětím 3,3–4,2 V[9].

⁷<https://wiki.wemos.cc/products:lolin32:lolin32>



Obrázek 2.13: WeMos Battery Shield — ukázka zařízení, konektorů (Obrázek přejet z: <https://tronixlabs.com.au/esp8266/wemos-esp8266/wemos-d1-mini-lipo-battery-shield-australia/>)

Napájecí modul kontaktního pole

Jde o zařízení, kterým lze napájet nepájivá pole. Lze jej tedy využít především testování a měření spotřeby proudu při provozu. Zařízení obsahuje několik slotů umožňujících napájení nepájivého pole — a to klasické USB typu B, mini USB a 5,5 mm 2,1 mm konektor. Je možné jej napájet napětím 7–12 V a disponuje výstupním napětím 3,3 V a 5 V mezi kterými je možno přepínat. Deska obsahuje také 6 stálých výstupních pinů — 3 pro 3,3 V a 3 pro 5 V (viz obrázek 2.14). Na zařízení je zabudována LED dioda indikující, zdali je zařízení napájeno.



Obrázek 2.14: Na obrázku lze vidět výstupní konektory, které jsou realizovány formou přepínačů a nebo stálých výstupních pinů. Při připojení zařízení je důležitá jeho orientace. (Obrázek přejet z: <https://ja.aliexpress.com/item/32821750004.html>)

2.1.5 Snímače veličin

Poslední skupinou v rámci zařízení potřebných pro tvorbu IoT senzorů jsou konkrétní snímače hodnot, které jsou potřeba k měření vlastností prostředí. Dnešní doba poskytuje různorodá řešení, co se týče funkcionality a cenového rozpětí. Sensory mohou být již zabudované do vývojových desek, mohou být vestavěny do přípojných modulů, nebo je možné je pořídit samostatně.

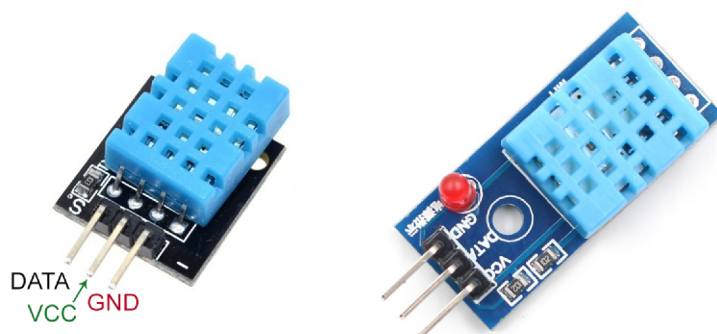
V následující podkapitole jsou uvedené příklady několika základních zařízení pro snímání hodnot. Jako základní snímače pro zaznamenání teploty a vlhkosti vzduchu jsou uvedeny senzory **DHT11** a **DHT22**. Mezi zástupce snímačů vestavěných do vývojové platformy patří

například půdní senzor **HiGrow**, který poskytuje údaje o vlhkosti půdy, teplotě a vlhkosti vzduchu. Za účelem získání informací o intenzitě srážek je jako využitelné zařízení uveden **MH Sensor**. V případě potřeby měření intenzity osvětlení je níže popsán **Light Sensor**. Dále pro měření teploty konkrétního objektu a prostředí je uveden **Melexis MLX90614**.

DHT11 – Teplotní a vlhkostní snímač

Teploměr a vlhkoměr DHT11 je často dodáván již zakomponovaný v přípojném modulu. Stačí ho pouze připojit k naprogramované vývojové platformě a bez dodatečné kalibrace umožňuje okamžitě číst naměřené hodnoty. Vhodná perioda pro zaslání vzorků je kolem 1 s, ideálně by neměla klesnout pod 1 s. Značnou výhodou snímače oproti jiným teplotním sensorům je schopnost měřit kromě teploty i vlhkost. Umožňuje měřit na stupnici teplot 0–50 stupňů Celsia, s přesností ± 1 stupeň Celsia, a vlhkost je měřena v rozmezí 20–90 % RH (Relative Humidity), s přesností ± 4 % RH [4]. Je schopný rozlišit hodnoty po 1 stupni dané veličiny. Existuje také přesnější varianta tohoto teploměru DHT22, viz sekce 2.1.5.

Maximální odběr proudu senzorem činí 2,5 mA při zpracovávání dat. Disponuje čtyřmi piny (schéma na obrázku 2.15), z čehož je jeden určen pro napájení 3–5,5 V (nejčastěji na 5 V), druhý slouží pro přenos dat, a čtvrtý k uzemnění. Třetí pin není běžně využíván. Pokud připojujeme datový kabel na malé vzdálenosti (pod 20 metrů), doporučuje se k němu připojit **pull-up rezistor**. DHT11 zůstává v úsporném režimu, dokud mu není od MCU zaslán startovací signál. Průměrná spotřeba činí 0,5–2,5 mA v aktivním módu a 100–150 μ A v režimu pohotovosti [28].

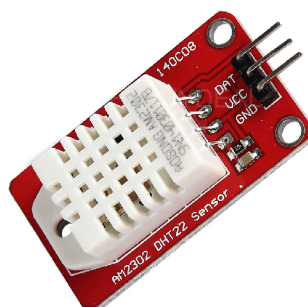


Obrázek 2.15: Na obrázku jsou zachyceny dva typy DHT11 modulů a rozložení jejich pinů. Modul na pravé straně obsahuje dodatečnou LED diodu, indikující aktivní napájení modulu. (Obrázky přežaty z: <https://happyec.in/product/dht11-temperature-and-humidity-sensor-module/>, <https://robu.in/product/dht11-digital-relative-humidity-temperature-sensor-module/>)

DHT22 – Teplotní a vlhkostní snímač

DHT22, označovaný také názvem AM2302, je nástupcem teplotního senzoru DHT11. Napájecí napětí snímače je stanoveno na 3,3–6 V, obvykle je napájen na 5 V. Naměřené hodnoty jsou zaslány výstupním signálem v digitální podobě, který je veden po jediné sběrnici. Rozsah teplot, které snímač umožňuje naměřit se pohybuje v rozmezí -40 – 80 °C. Vlhkost je možno naměřit v rozsahu 0–100 % RH (Relative Humidity). Přesnost měření vlhkosti je ± 2 % RH a teploty $\pm 0,5$ °C. DHT22 rozlišuje teplotu a vlhkost na stupnici po 0,1 stupni

dané veličiny. Jeho průměrný interval měření je 2 s, ideálně by měl být delší než 2 s. V pohotovostním režimu odebírá 40–50 μA a 1–1,5 A při měření [3].



Obrázek 2.16: Na obrázku je zobrazen snímač DHT22, připojený k modulu s dodatečným rezistorem a kapacitorem. (Obrázek přejet z z: <https://potentiallabs.com/cart/buy-dht22-module-online-hyderabad-india>)

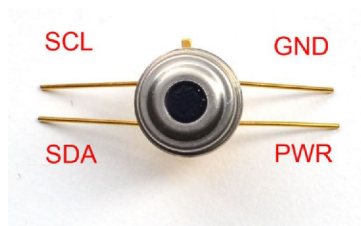
Dallas DS18B20 - Teplotní snímač

Dallas je standardní teplotní snímač bez rozšířené funkcionality. Měří teploty v rozmezí mezi -55 – 125 $^{\circ}\text{C}$. Hodnoty jsou měřeny s přesností $\pm 0,5$ $^{\circ}\text{C}$ při rozmezí naměřených teplot od -10 – 85 $^{\circ}\text{C}$. Obsahuje tři piny, kterými je napojen na vývojovou desku — pin pro napájení, uzemnění a přenos signálu (DQ). Pin DQ může fungovat také v režimu „parasite power“. Při něm je DQ připojen ke sběrnici a odebírá jí napětí, když je sběrnice napájena. V tomto případě musí být pin pro napájení připojen na zem. Teplotní snímač pracuje na napájecím napětí mezi 3–5,5 V. Odběr při nečinném módu je kolem 1 μA a v aktivním režimu (například při zápisu dat) se pohybuje od 1–1,5 mA [26].

Melexis MLX90614 — Infračervený teplotní snímač

Snímač Melexis slouží k bezkontaktnímu snímání teploty objektů a prostředí. Měří teploty prostředí v rozmezí od -40 – 125 $^{\circ}\text{C}$ a teploty objektů v rozsahu mezi -70 – 380 $^{\circ}\text{C}$. Senzor snímá s přesností na $\pm 0,5$ $^{\circ}\text{C}$ i při měření extrémních hodnot. Některé varianty senzoru mohou mezi měřením fungovat v úsporném režimu. Různé varianty snímače pracují na napětí 3 V nebo 5 V. 5 V-varianta má doporučené napájení v rozmezí mezi 4,5–5,5 V a 3 V-varianta od 2,6–3,6 V [27]. Komunikuje přes rozhraní I2C se 7 bitovou adresou. Z toho důvodu obsahuje čtyři piny (viz obrázek 2.17), kde první je určen pro napájecí napětí (pin PWR), druhý slouží pro zaslání dat přes I2C (pin SDA), třetí pro zaslání hodinového signálu v rozhraní I2C (pin SCL) a čtvrtý slouží pro uzemnění (pin GND).

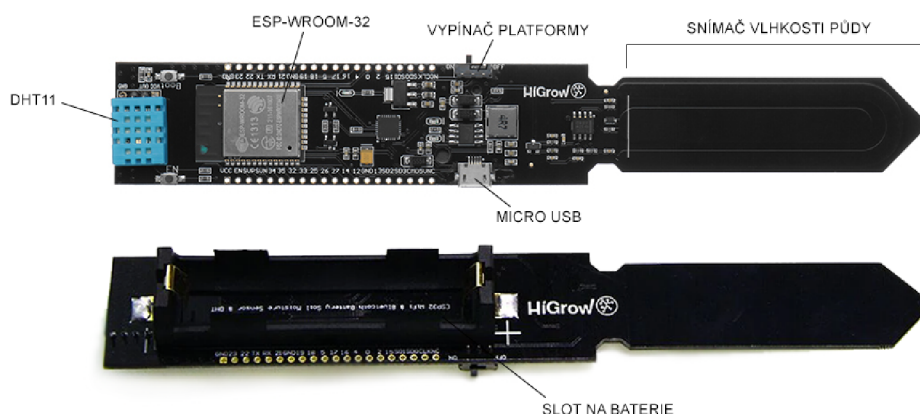
Snímač MLX90614 je schopný spánkového režimu při zaslání příkazu. Tento režim není umožněn ve variantě zařízení s napájením na 5 V. Pro omezení spotřeby na často požadovaných 2,5 mA, je třeba SCL pin držet v nízké hladině napětí při úsporném režimu.



Obrázek 2.17: Piny snímače Melexis MLX90614 — napájecí (PWR), zemní (GND), sloužící pro komunikaci přes I2C (SDA a SCL). (Obrázek přejat z: <https://cdn-learn.adafruit.com/downloads/pdf/using-melexis-mlx90614-non-contact-sensors.pdf>)

HiGrow – Snímač vlhkosti půdy

Senzor HiGrow se může svým složením řadit mezi vývojové platformy. Mimo snímače obsahuje již zabudovaný čip ESP32, umožňující komunikovat pomocí Wi-Fi nebo Bluetooth, který je součástí modulu ESP-WROOM-32. Ve svém základu disponuje hned několika snímači. Prvním je půdní sonda, která slouží pro zaznamenávání vlhkosti půdy. Při instalaci se zavádí do země (viz obrázek 2.18). Dále obsahuje teplotní snímač DHT11, který umožňuje snímat teplotu a vlhkost vzduchu v okolí senzoru HiGrow. Disponuje také zabudovaným slotem na baterie, který slouží k bezdrátovému napájení zařízení. Teplotu a vlhkost vzduchu poskytuje v rozmezí zajištěné již zmíněným senzorem DHT11, hodnoty získané půdním senzorem je třeba nejprve zkalibrovat [15].



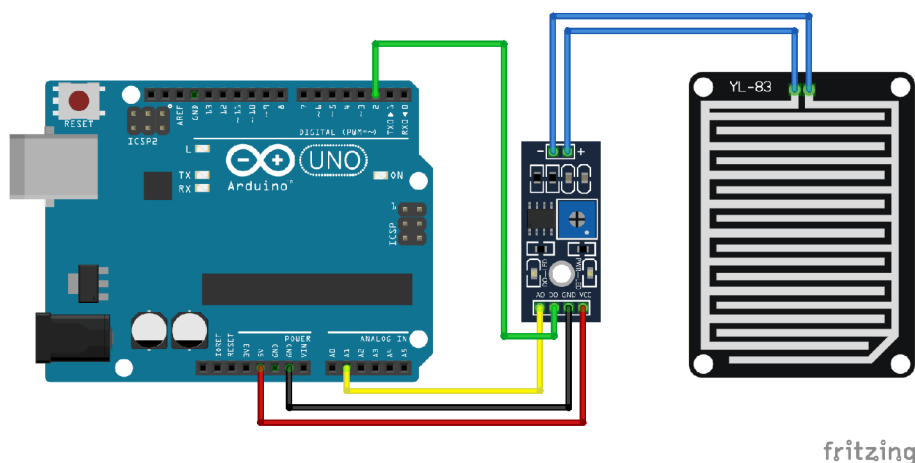
Obrázek 2.18: Senzor HiGrow — na horním obrázku je vidět přední stranu senzoru, na níž je osazen snímač teploty a vlhkosti půdy DHT11, modul ESP-WROOM-32 s čipem ESP32, vypínač celé platformy, mikro USB a celou pravou část tvoří snímač vlhkosti půdy, který je aplikován do půdy. Na zadní straně senzoru (spodní část obrázku) lze vidět slot na baterie. (Obrázky přejaty z: <https://www.tokopedia.com/productkamiimport/wemos-higrow-esp32-wi-fi-bluetooth-battery-dht11-soil>, <https://ja.aliexpress.com/item/32816660142.html>)

Flying Fish MH Sensor – Srážkový snímač

Modul Flying Fish slouží k indikaci množství srážek dopadených na měřicí desku snímače. Senzor se skládá ze dvou částí — řídicí desky, která přijímá data ze snímače v analogové po-

době, a ze snímače, který data měří a poskytuje desce. Řídící deska obsahuje komparátor, který zachytává překročení stanovené hranice (množství srážek) a vrací indikaci překročení v digitální podobě. Jak lze vidět na obrázku 2.19, tak na rozdíl od senzoru **HiGrow** neobsahuje žádné moduly pro vysílání dat, či napájení přes baterii. Proto je třeba ho pro zprovoznění těchto funkcionalit připojit ještě k vývojové platformě [11].

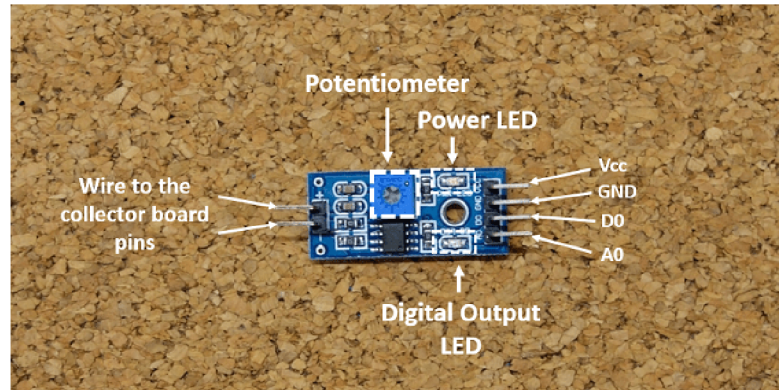
Řídící desku lze využít více způsoby dle připojeného modulu. Za účelem indikace deště je napojena na srážkový modul MH-DR. Ten měří výskyt a případnou intenzitu srážek. V tomto případě vrací snímač naměřenou analogovou hodnotu a řídící deska zasílá převedenou digitální hodnotu jako reprezentaci indikace deště platformě, například k **Arduino UNO**. Napájení je stanoveno na 5V.



Obrázek 2.19: Flying Fish MH Sensor — schéma zapojení (Obrázek přejat z: <http://invent.module143.com/rainfall-sensor-how-to-use-it/>)

YL-38 (se snímačem YL-69) – Snímač vlhkosti půdy

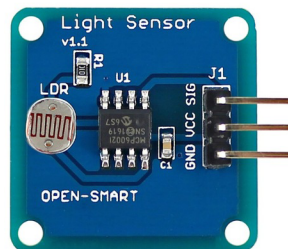
Jeho využití spočívá v indikaci aktuální vlhkosti půdy a následné zpětné detekci vhodné závlivky rostlin. Sensor YL-38 funguje na podobném principu jako senzor **Flying Fish**. Vysílá data v analogové podobě, a to v rozmezí hodnot napětí od 0–4,2 V. Umožňuje také digitální signalizaci překročené stanovené hranice v podobě digitálního výstupu 0 nebo 1. Na rozdíl od snímače vlhkosti půdy **HiGrow**, je třeba jej pro zprovoznění připojit na vývojovou platformu, například na desku **Arduino UNO**. Snímač YL-69 je napájen napětím od 3,3–5 V [30]. Disponuje čtyřmi piny (viz obrázek 2.20), z nichž kromě klasických pinů pro napájení a připojení k uzemnění, má také pin pro výstup digitálního (D0) a analogového (A0) signálu.



Obrázek 2.20: Senzor YL-38 - rozložení pinů, A0 (analogový výstup), D0 (digitální výstup), VCC (vstup napětí), GND (uzemnění). Dále je na desce vidět potenciometr pro regulaci citlivosti - ovlivňuje digitální výstup. (Obrázek přejat z: <https://randomnerdtutorials.com/guide-for-soil-moisture-sensor-yl-69-or-hl-69-with-the-arduino/>)

Light Sensor - Snímač intenzity osvětlení

Senzor intenzity osvětlení slouží pro detekci ambientního světla. Hlavní částí snímače je fotorezistor LDR (*Light Dependent Resistor*). LDR je analogový snímač — poskytuje mikrokontroléru určitou úroveň napětí, která je interpretována jako daná intenzita osvětlení. Obsahuje pouze tři piny (viz obrázek 2.21) — pro napájecí napětí (VCC), zaslání analogového signálu (SIG) a uzemnění (GND). Je možno jej napájet napětím 2,7–5,5 V, typicky využívané napětí je 5 V [34].



Obrázek 2.21: Light Sensor — na obrázku je zobrazen snímač intenzity osvětlení s vestavěným fotorezistorem LDR. Dále jsou na modulu vyobrazeny piny pro napájení, uzemnění a přenos analogového signálu. (Obrázek přejat z: <https://ja.aliexpress.com/item/Light-Intensity-Sensor-Module-5528-Photoresistor-For-Arduino-Hot-New-Original-Product/1825718281.html>)

2.2 Síťové rozhraní, protokoly a jejich vazba na úsporu energie

V případě softwarových rozhraní využitelných při tvorbě domácích senzorů je třeba zvažovat požadované vlastnosti jako je spotřeba energie, dostupnost, robustnost sítě. Musíme vědět k jakému účelu senzory budou sloužit, kolik jich bude, jak často budou komunikovat, jaký objem dat budou zasílat a jak vzdálené od sebe budou. Na základě takto vytvořené představy můžeme uvažovat následující protokoly, umožňující komunikaci mezi senzory a následně jejich propojení do síťových topologií, blíže rozebraných v kapitole 2.3.

Důležitou kategorií patřící mezi softwarová řešení zapojení senzorů jsou síťové protokoly. Protokoly vyvinuté přímo pro komunikaci senzorů a odesílání dat ze snímačů nám mohou umožnit jednodušší přenos a tím pádem nám zaručí značnou úsporu energie. Tyto protokoly jsou představeny v následujících sekcích.

2.2.1 Vícevrstevní protokoly

První skupinou jsou technologie, modifikující komunikační protokoly na několika vrstvách. Jde o modifikace za účelem zjednodušení komunikace a snížení objemu přenášených dat. Mezi zástupce této kategorie patří technologie Zigbee (popsána v podkapitole [Zigbee](#)), která je standardem pro implementaci zařízení komunikujících na krátké vzdálenosti. Druhou technologií je Z-Wave (popsána v sekci [Z-Wave](#)). Třetí technologií modifikující více vrstev komunikačního zásobníku je technologie ESP-NOW, která je blíže specifikována v podkapitole [ESP-NOW](#).

ZigBee

Bezdrátová technologie ZigBee je založena na standardu IEEE 802.15.4. Zigbee definuje sadu komunikačních protokolů pro komunikaci na krátké vzdálenosti, řádem na 10–100 m. Zařízení je tedy orientováno spíše na domácí využití v sítích [PAN](#). Je vhodný pro zařízení, která přenášejí malý objem dat, na nízkou vzdálenost, s potřebou úspory energie. Zařízení založená na této technologii komunikují ve frekvenčním pásmu 868 MHz pro Evropu, 915 MHz pro Severní Ameriku a 2,4 GHz celosvětově. Data jsou přenášena v jednotlivých pásmech rychlostmi 20, 40 a 250 kbit/s [21].

Většina zařízení fungujících pod technologií Zigbee mají omezenou dobu aktivity a většinu času jsou přepnuty do úsporných režimů (popsány v podkapitole 2.1.1). Výsledkem takového chování je výdrž zařízení s technologií Zigbee při provozu na baterii několik let bez potřeby výměny baterie [16].

Protokolový zásobník technologie Zigbee je rozložen na 4 hlavních vrstvách: na fyzické, podvrstvě linkové vrstvy Media Access Control (MAC) vrstvě, síťové vrstvě a aplikační vrstvě. Síťová vrstva podporuje adresování a směrování v topologiích typu [mesh](#). Umožňuje směrování typu ad-hoc, čímž povoluje komunikaci i mimo přímou viditelnost dvou zařízení.

Z-Wave

Je o sadu protokolů, která jsou opět využitelná pro zařízení přenášející malé množství dat a napájená na baterii. Je založena stejně jako [Zigbee](#) na standardu IEEE 802.15.4. v Evropě komunikuje na pásmu 868,4 MHz a 908 MHz v Americe, neruší se tak s technologiemi jako je například Wi-Fi nebo Bluetooth. Rychlost přenosu dat činí 9,6 kbit/s pro Evropu a 40 kbit/s pro Ameriku. Novější typ čipu podporuje i 2,4 GHz pásmo s přenosovou rychlostí

200 kbit/s [21]. Podporuje komunikaci na krátké vzdálenosti, uvnitř kolem 30 m a venku až 100 m. Protokolový zásobník technologie Z-Wave je složen z pěti hlavních vrstev: fyzické, MAC podvrstvy, transportní, síťové a aplikační.

Sít Z-Wave obsahuje dva typy komunikujících uzlů — *controllers* a *slaves*. Controllers zahrnují směrovací tabulku a počítají vhodné cesty mezi routery. Uzly typu slave slouží pouze jako monitorovací zařízení, pracující se vstupy a výstupy. Protokol Z-Wave podporuje připojení až 232 uzlů typu controllers nebo slaves. Z-Wave je propojována do topologií typu **mesh**.

ESP-NOW

Komunikační technologie ESP-NOW zahrnuje protokol, který se napodobuje Wi-Fi bez zprostředkování spojení. V této technologii jsou data aplikace zapouzdřena do rámce specifického pro typ příjemce a zaslána z jednoho Wi-Fi zařízení na druhé bez potřeby navázání spojení. Pro ochranu rámce je využit šifrovací protokol CCMP [12]. Protože jde o nespojovanou komunikaci, je hlavička MAC vrstvy mírně pozměněna oproti podobě v klasickém rámci.

Komunikační technologie ESP-NOW umožňuje zasílání malých objemů dat. Tím zkracuje dobu potřebnou na přenos zprávy a tím dále šetří spotřebu zařízení. Tím se stává úspornější než Wi-Fi. Protokol komunikuje na pásmu 2,4 GHz. Dosahuje do vzdálenosti až třikrát větší než Wi-Fi, tedy až 480 m⁸. Po spárování zařízení je spojení trvalé. Využívá peer-to-peer architektury pro komunikaci.

2.2.2 Protokoly aplikační vrstvy

Protokolů modifikujících aplikační vrstvu v rámci funkcionality v IoT je více. Významným komunikačním standardem pro IoT, který modifikuje aplikační vrstvu, je protokol MQTT. Tento protokol umožňuje zasílání zpráv od snímačů hodnot do zařízení, která si přijímání daných zpráv vyžadují. Tento mechanismus je blíže popsán v následující sekci.

MQTT

Je protokolem aplikační vrstvy, který přenáší data pomocí běžné internetové sítě. MQTT (*MQ Telemetry Transport*) je nenáročný komunikační protokol využitelný v komunikaci M2M (*Machine to Machine*). Jeho principem je zasílání zpráv přes svůj centrální bod, tzv. *broker*. V článku (odkázaném zde [36]) je popsán jako „novinář“, který může mít mnoho poskytovatelů zpráv — uzly nazvané *publisher*, a mnoho čtenářů — uzly pojmenované jako *subscriber*. Jeden novinář může spravovat více poskytovatelů zpráv i více čtenářů zároveň, přičemž čtenář dostane jen ty zprávy, u kterých se přihlásil předem k odběru. V reálném použití jsou publishers i subscribers tvořeny elektronickými součástmi komunikujících jako MQTT klienti, připojeni k jednomu brokeru — MQTT serveru. Teoreticky může být každý klient i publisher a zároveň i subscriber, ale v praxi je častější rozdělení těchto rolí. Publisher bývá v našem případě nějaký snímač či senzor, který odesílá naměřená data na server (broker) a subscriber je řídicí jednotka, která odebírá tyto zprávy ze serveru a dále s nimi vykonává nějakou akci, například je zobrazuje.

⁸<https://www.instructables.com/id/ESP-NOW-Wi-Fi-With-3x-More-Range/>

2.2.3 Protokoly síťové vrstvy

Další skupinou protokolů se zabývá síťovou vrstvou. Jde především o protokol 6LoWPan sloužící ke komunikaci nízkospotřebních zařízení přes sítě založené na standartu IEEE 802.15.4.

6LoWPAN

Tento protokol linkové vrstvy je již podle názvu je propojením internetového protokolu (IPv6) a nízkospotřebních bezdrátových osobních sítí (LoWPAN – Low-power Wireless Personal Area Networks). Tím umožňuje i nejmenším zařízením (v našem případě například jednoduchým sensorům) s omezeným výkonem posílat bezdrátově data s využitím internetového protokolu. Umožňuje tedy těmto dříve opomíjeným zařízením komunikaci přes sítě založené na standartu IEEE 802.15.4.⁹ Odhadovaný dosah sítě s protokolem 6LoWPAN je 116 m [24]. Využívá topologie typu mesh.

2.2.4 Protokoly linkové vrstvy

Následující skupina protokolů modifikuje určitým způsobem linkovou vrstvu komunikačního modelu. Hlavními zástupci v této skupině využitelnými při komunikaci několika sensorů jsou kromě klasické technologie Bluetooth, BLE a LoRaWAN. Každá z nich má přednosti v rozdílných aspektech. v následující části jsou tyto výhody, respektive nevýhody blíže rozebrány. Hlavním účelem těchto protokolů je snížit objem zasílaných dat a zvýšit tím efektivitu komunikace mezi uzly.

Bluetooth

Technologie Bluetooth je v současnosti velmi rozšířená, avšak v oblasti vývoje sensorových sítí existují s ohledem na spotřebu energie dnes i mnohem výhodnější řešení (viz BLE). Bluetooth je využíván v PAN (Personal Area Network), takže nepracuje na příliš velké vzdálenosti — maximum je 100m. Je také omezený co se týče maximálního počtu připojených zařízení, kde maximálním počtem současně propojených zařízení je 7. Jeho spotřeba energie je 1W, ale existuje i mnohem šetrnější alternativa (BLE). Technologie Bluetooth má však kratší čas pro zaslání jedné zprávy, který je 0,625 ms. Jeho čas na probuzení a odeslání zprávy z neaktivního stavu je 100 ms. Při maximálním vytížení odebírá proud přibližně 30 mA [29].

BLE

Jak už sama zkratka značí (*Bluetooth Low Energy*), jde o protokol implementujícím odlehčenou linkovou vrstvu. Je provozován na frekvencích jako Bluetooth 2,4 GHz, jeho výhoda je ale ve výrazném šetření energie u baterií napájených zařízení. Jeho spotřeba energie je oproti Bluetooth značně nižší — 0,01–0,5 W. S technologií Bluetooth je zpětně kompatibilní. Maximální počet připojených zařízení zde závisí na konkrétní implementaci. „Za účelem šetření energie je zde oproti technologii Bluetooth výrazně snížena přenosová rychlost dat na 260 kbit/s [32].“

Dosahuje vzdálenosti nad 100 m (předpokládá 77 m¹⁰), jeho latence (zpoždění) je 6 ms z nepřipojeného stavu, v čemž je tedy mnohem rychlejší než Bluetooth, ale na druhou stranu

⁹<https://www.link-labs.com/blog/6lowpan-vs-zigbee>

¹⁰<https://www.link-labs.com/blog/ble-range>

jeho minimální potřebný čas pro zaslání zprávy jsou 3 ms, v čemž oproti klasickému Bluetooth zaostává. Jeho spotřeba při maximálním vytížení je však oproti Bluetooth poloviční — méně než 15 mA [29]. Topologie připojení BLE zařízení je typu hvězda.

LoRaWAN

LoRaWAN je protokol postavený na LoRa technologii, vytvořený LoRa Alliancí. LoRaWAN (*Long Range Wide Area Network*) je síťový protokol zajišťující zabezpečenou nízkospotřební komunikaci mezi zařízeními obsahujícími čip LoRa. Často je zaměňována s pojmem LoRa (*Long Range*), který značí protokol fyzické vrstvy. Data přenáší v pásmu 1 GHz.

Komunikuje na pásmu 868 MHz a rychlost přenosu činí 0,25–50 kbit/s. Aby byla výdrž baterie koncového zařízení co největší, spravuje server s LoRaWAN RF výstup a přenosovou rychlost pro každé koncové zařízení zvlášť. Dosah činí 40 km v otevřeném terénu a 2–5 km ve městě. Výdrž na bateriích se udává 5–15 let¹¹ v závislosti na hustotě komunikace. Síťovou topologií využívanou při této technologii je hvězda na dvou úrovních.

2.2.5 Protokoly fyzické vrstvy

Poslední významnou skupinou v rámci softwarových platforem využitelných v návrhu senzorové sítě jsou protokoly modifikující pouze fyzickou vrstvu. Mezi nejdůležitějšího zástupce patří LoRa, poskytující řešení pro senzorové sítě realizované na velké vzdálenosti.

LoRa

LoRa je bezdrátová technologie, která slouží pro přenos dat pomocí modulace radiových vln. Modulace dat je generována v čipu vysílače, který je dodáván firmou Semtech. Modulační technika umožňuje přenos malého množství dat na velké vzdálenosti, bez přítomnosti interference. Minimalizuje také spotřebu zařízení.

Technologie zahrnuje protokol fyzické vrstvy, který umožňuje komunikaci na vzdálenosti až 7 km z uzavřených prostor a až 20 km v otevřeném prostoru.¹² Udávaná výdrž na baterii je až 10 let, čímž snižuje náklady na provoz. Komunikace přes LoRa zařízení vykazuje velmi nízkou spotřebu mezi 10–100 mW [23]. v Evropě pracuje na frekvenci 868 MHz. Rychlost odesílání a přijímání dat se pohybuje mezi 0,3–100 kbit/s. Průměrná latence při příjmu zpráv se pohybuje kolem 2 s.

LoRa zařízení mohou komunikovat v klasickém point-to-point spojení, nebo mohou tvořit LoRa síť, za použití například LoRaWAN protokolu (viz **LoRaWAN** podkapitola). Při point-to-point komunikaci si spolu předávají data dvě zařízení s LoRa čipy, za využití RF signálů. V tomto případě má LoRa komunikace mezi dvěma zařízeními mnohem vyšší dosah, než kdyby komunikace probíhala například přes Wi-Fi nebo Bluetooth. Dosah EPS32 s LoRa čipem je u spolehlivé komunikace udáván na 200 m, dosahující až 30 km u některých řešeních.

2.3 Topologie senzorových sítí

Pro účely úsporného chování zařízení v senzorové síti existuje několik topologií, které jsou v IoT přednostně využívané. Mezi nejčastější z nich patří topologie typu **mesh**, **hvězda**

¹¹<https://www.iot-portal.cz/2016/02/29/lorawan/>

¹²<https://www.murata.com/support/faqs/products/lpwa/lorawan/hardware/0008>

a topologie typu **point-to-point** [25]. Každá z nich má jiné přednosti a nedostatky odvíjející se od konkrétních potřeb sítě. V následujícím přehledu jsou shrnuty obecné vlastnosti těchto tří topologií, na základě kterých je možno zvolit vhodnou topologii pro konkrétní případ užití.

Topologie se liší využitelností s ohledem na množství připojených uzlů a na vyžadovanou kvalitu sítě. V případě kvality se jedná o stabilitu sítě při nečekaném odpojení libovolného uzlu. Co se týče množství propojených uzlů, lze obecně říci že nejhůře je na tom topologie typu **point-to-point**. Ta je vhodná pouze pro propojení dvou bodů. Topologie typu hvězda je sice schopná pojmout více zařízení, ale na druhou stranu při výpadku centrálního bodu ztrácí svou stabilitu. Nejvhodnější topologií je v tomto směru decentralizovaná topologie typu **mesh**, která má schopnost regenerace, ale ta pro změnu strádá nejvyššími nároky na spotřebu energie [35].

Point-to-Point

Přímé spojení mezi pouze dvěma uzly. Výhodou je v tomto případě jednoduchost oproti topologii typu **mesh** a hvězdicové topologii a nízká cena. Dosah sítě je limitován na jeden hop, z toho vyplývá i nevýhoda pro využití v IoT, kdy uzly nemohou využít komplikovanější sítě například pro větší dosah. Data mohou být zasílána jedním nebo oběma směry. Využívá se při jednodušších schématech zapojení, kdy není třeba uvažovat širokou komunikující síť, například při spojení telefon–bezdrátová sluchátka. Jeden z uzlů je typicky gateway, aby bylo možné komunikovat mimo lokální síť.



Obrázek 2.22: Schéma propojení v topologii typu **point-to-point** — jednosměrná a obousměrná komunikace (Obrázek přejat z: <http://archive.rtc magazine.com/articles/view/102871>)

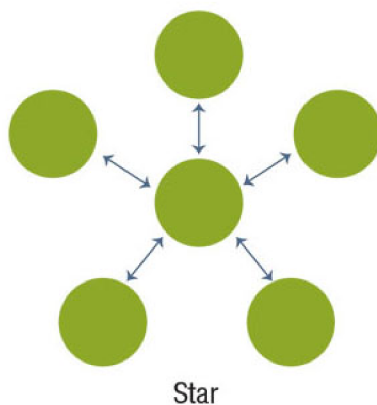
Hvězdicová topologie

Je to způsob rozložení sítě kolem jednoho ústředního bodu. Tím bývá centrální přístupový bod (uzel **gateway**), který je propojen s vnější sítí. Na něj jsou napojeny jednotlivé senzory, které snímají a zasílají hodnoty tomuto zařízení. Všechny uzly mohou tedy zasílat nebo přijímat data pouze skrze centrální bod. Příkladem je například domácí Wi-Fi síť.

Výhodou této topologie je její výkon. Síť je konzistentní, disponuje předvídatelnou kvalitou a je také rychlá. Data v síti typu hvězda putují typicky jeden hop do požadovaného cíle (při cestě mezi zařízením typu hub a senzorem), nebo nanejvýš 2 „hopy“ při komunikaci mezi 2 senzory [31]. Tento fakt způsobuje jejich nízké a předvídatelné zpoždění. Duhou výhodou je vysoká spolehlivost sítě díky snadné detekci chybových zařízení. Protože každý senzor je spojen pouze s centrálním bodem, je možné v případě chyby izolovat zařízení jeden po druhém nezávisle na sobě.

Na druhou stranu má topologie typu hvězda několik nevýhod. Tento způsob propojení komunikujících zařízení, stejně jako topologie **point-to-point** omezuje přenosovou vzdálenost jednotlivých zařízení. Druhou a podstatnější nevýhodou je problém odpojení centrálního

bodů. Pokud dojde k selhání gateway, je odpojena od vnější sítě celá WSN. Na rozdíl od topologie typu **mesh**, kdy po selhání gateway sice WSN není schopná komunikovat s vnější sítí, ale lokálně nadále funguje.



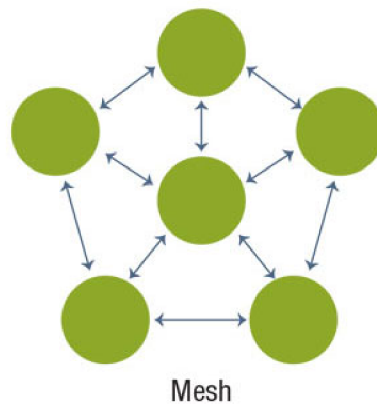
Obrázek 2.23: Schéma propojení v topologii typu hvězda (Obrázek přejat z:<http://archive.rtcmagazine.com/articles/view/102871>)

Mesh

Je typ síťové topologie, kde se všechny uzly podílejí na zasílání dat po síti. Každý uzel je nejen schopný přijímat a odesílat zprávy, ale také směřovat zprávy pro uzly k sobě připojené. Tím je schopen šířit zprávy po celé síti a zároveň komunikovat s centrálním bodem, který je v tomto případě **gateway**. Ta slouží pro komunikaci mimo WSN.

Topologie typu mesh je výhodná pro aplikace vyžadující velký dosah a pokrytí široké oblasti. Tuto výhodu topologie poskytuje, protože není dosahem omezena na vzdálenost pouze mezi dvěma uzly. Nevýhodou v této topologii může být vyšší energetická spotřeba a redundance dat při přenosu.

Při potřebě šetřit energii můžeme zavést princip zohledňující napájení na baterii. Tento způsob staví na teorii, že uzly volí cestu pro zaslání dat podle zjištění stavu baterie sousedních uzlů. Například uzly s nízkou kapacitou baterie by měly být úplně vynechány z hledání dostupné cesty. Hlavním cílem tohoto přístupu je nalezení energeticky nejlevnější trasy mezi požadovanými uzly. Další možností úspory energie je zavádění úsporných režimů v době mezi přenosy [33]. Zařízení jsou přepínána mezi aktivním režimem, light-sleep a deep-sleep módem mezi vlastními měřeními nebo žádostmi o předání zprávy sítí.



Obrázek 2.24: Schéma propojení v topologii typu mesh (Obrázek přejat z: <http://archive.rtc magazine.com/articles/view/102871>)

Kapitola 3

Návrh senzorů určených pro domácí síť a úsporných síťových topologií

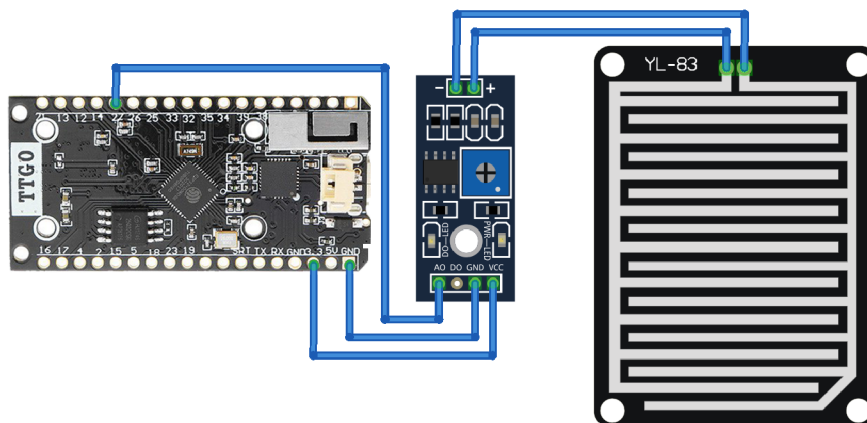
Následující kapitola se zabývá návrhem senzorů určených ke snímání fyzikálních vlastností prostředí. U každého zařízení jsou popsány zvolené komponenty, ze kterých se skládá. Následně jsou rozebrány způsoby napájení a navrženy vhodné aparáty umožňující úsporu energie. Z důvodu potřeby sledovat spotřebu jednotlivých zařízení v provozu, jsou senzory navrženy za účelem jejich napájení přes baterie.

3.1 Senzory určené pro snímání veličin prostředí

Navržené senzory jsou specifikovány dle druhu snímané veličiny. Podle názvu je vždy identifikovatelná platforma a snímaná veličina, které je důležité odlišit pro další srovnávání. Platformy jsou využity z kapitoly 2, kde jsou blíže specifikovány jejich parametry.

Srážkový senzor LoRa_rain

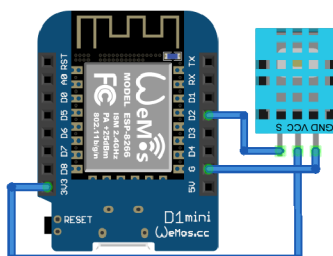
Senzor určený ke snímání intenzity srážek je založen na platformě LILYGO TTGO LoRa32, která je specifikována v sekci 2.1.3. Deska obsahující modul ESP-WROOM-32, který je založen na čipu ESP32, umožňuje provoz komunikace přes protokoly založené na technologii Wi-Fi, bluetooth a BLE. Jako snímací zařízení je využit srážkový snímač **Flying Fish MH Sensor**. Snímač je k platformě připojen přes analogový pin 27 (viz obrázek 3.1) a napájen je z desky přes 3,3 V. Díky zabudovanému konektoru k připojení baterie ji lze připojit přímo k platformě.



Obrázek 3.1: Schéma zapojení senzoru LoRa_rain. Modul ke snímání srážek je napájen pinem označeným „3.3“ (na 3,3 V) a data jsou zaslána na analogový pin 27.

Teplotní a vlhkostní senzor WeMos_DHT11

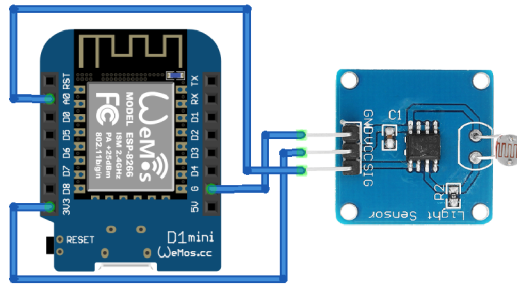
Ke snímání teploty a vlhkosti prostředí slouží zařízení založené na vývojové platformě WeMos D1 mini, která je blíže popsána v části 2.1.3. Platforma je založena na čipu ESP8266, který umožňuje komunikaci přes Wi-Fi technologii. K zařízení je připojen snímač DHT11, sloužící k detekci teploty a vlhkosti prostředí. Snímač je napájen z 3,3 V pinu platformy WeMos D1 mini a signál je přenášen digitálním pinem „D2“ (viz obrázek 3.2). Pro napájení desky přes baterie je možno využít WeMos Battery Shield (blíže specifikován v sekci 2.1.3), který přesně pasuje počtem a rozložením pinů na platformu WeMos.



Obrázek 3.2: Schéma zapojení senzoru WeMos_DHT11, které zobrazuje připojení snímače DHT11 k pinu „D2“ platformy WeMos za účelem zaslání dat a k pinu „3V3“ k napájení snímače.

Senzor na měření intenzity osvětlení WeMos_light

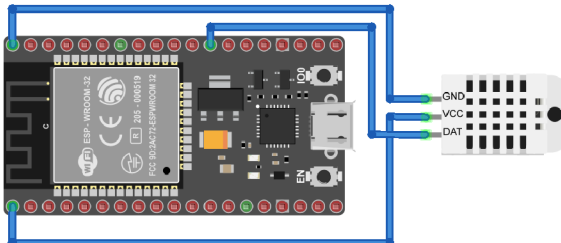
Základem zařízení sloužícího je snímání intenzity světla v prostředí je deska WeMos D1 mini (specifikována v sekci 2.1.3). Vestavěný mikročip ESP8266 umožňuje komunikaci protokolem založeným na technologii Wi-Fi. K platformě je dále připojen modul obsahující LDR fotorezistor, jehož parametry jsou přiblíženy v části 2.1.5. Snímací modul je k platformě připojen přes pin označený jako „3V3“ (viz obrázek 3.3), který poskytuje snímači potřebné napájecí napětí v rozsahu 3,3 V. Pro zaslání dat je modul připojen k analogovému pinu „A0“ desky WeMos. K napájení celého senzoru je využít napájecí modul v podobě WeMos Battery Shield.



Obrázek 3.3: Schéma zapojení senzoru WeMos_light, které zobrazuje připojení modulu s fotorezistorem LDR k pinu „A0“ platformy WeMos za účelem zasílání dat a k pinu „3V3“ k napájení snímače.

Teplotní a vlhkostní senzor DevKit_DHT22

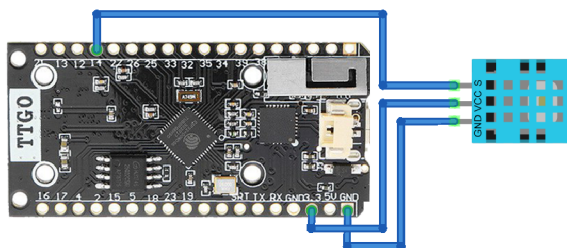
K sestavení senzoru měřícího teplotu a vlhkost prostředí je využita platforma ESP32-DevKit, založená na čipu ESP32 (platforma je blíže popsána v části 2.1.3). Díky čipu ESP32, který je součástí modulu ESP-WROOM-32 je možné u senzoru zahájit komunikaci přes technologie BLE, bluetooth a Wi-Fi. Je možné také díky tomuto mikročipu využít případných režimů spánku. Jako snímač je použita novější verze snímače DHT11 — DHT22. Snímač je napojen na pin „3V3“ pro zajištění jeho napájení na 3,3 V. Pro zasílání naměřených hodnot je DHT22 připojen k platformě k digitálnímu pinu „IO17“ (viz obrázek 3.4). Pro napájení na bateriích je třeba využít napájecí modul kontaktního pole (popsaný v sekci 2.1.4), který je připojen k nepájivému poli a poskytuje platformě napájecí napětí 5 V.



Obrázek 3.4: Schéma zapojení senzoru DevKit_DHT22, na kterém je vidět připojení snímače DHT22 k pinu „IO17“ platformy ESP32-DevKit za účelem zasílání digitálních dat a k pinu „3V3“ k napájení snímače.

Teplotní senzor LoRa_DHT11

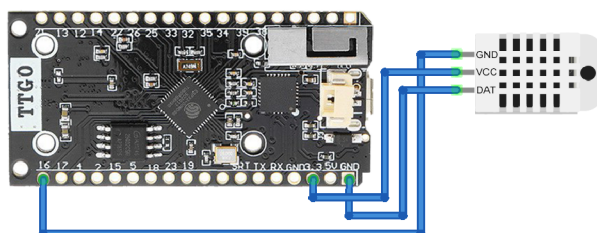
Zařízení je určeno k měření teploty prostředí. Hlavním prvkem je platforma LILYGO TTGO LoRa32, která díky vestavěnému modulu ESP-WROOM-32 obsahuje mikročip ESP32. Tím je zajištěna možnost využití režimů spánku a komunikace přes protokoly založené na technologiích Wi-Fi, BLE a bluetooth. Dále díky vestavěnému čipu SX1276 umožňuje také komunikaci přes technologii LoRa a LoRaWAN. Ke snímání teploty je využit snímač DHT11, který je z desky napájen pinem „3V3“ na 3,3 V (viz obrázek 3.5). Pro přijímání naměřených hodnot ze snímače je modul DHT11 připojen k digitálnímu pinu 16 vývojové desky. K napájení je využit konektor pro připojení baterie, který umožňuje napájet zařízení požadované napětí.



Obrázek 3.5: Schéma zapojení senzoru LoRa_DHT11, které zobrazuje připojení snímače DHT11 k pinu 16 platformy TTGO LoRa za účelem zaslání dat a k pinu „3V3“ k napájení snímače.

Teplotní senzor LoRa_DHT22

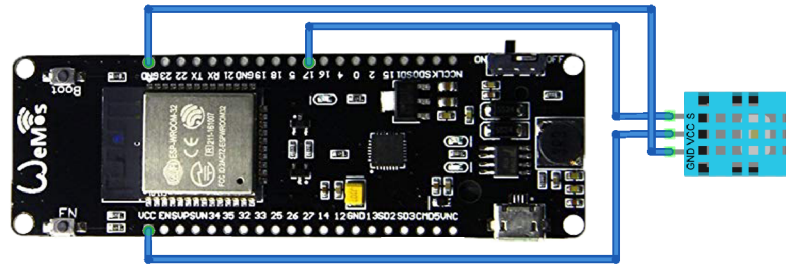
K sestavení senzoru je využita vývojová deska LILYGO TTGO LoRa32. Výpočetní jednotkou je mikročip ESP32 obsažený v modulu ESP-WROOM-32. Zařízení mimo komunikaci přes Wi-Fi, BLE a bluetooth (které jsou zajištěny díky čipu ESP32), umožňuje také díky vestavěnému čipu SX1276 komunikaci přes technologii LoRa. Umožňuje tím komunikovat na vzdálenosti v řádu kilometrů. Zařízení snímá teplotu připojeným teplotním snímačem DHT22. DHT22 je připojen k platformě k pinu označenému jako „3V3“ pro jeho napájení na 3,3 V. Pro přijímání dat ze snímače jsou data zaslána na digitální pin 14 (viz obrázek 3.6). Zařízení je možno provozovat na bateriích bez potřeby připojení externího modulu díky bateriovému konektoru.



Obrázek 3.6: Schéma zapojení senzoru LoRa_DHT22, zobrazující připojení snímače DHT22 k pinu 14 platformy TTGO LoRa za účelem zaslání dat a k pinu „3V3“ k napájení snímače.

Senzor ke snímání teploty objektu WeMos-Battery_DHT11

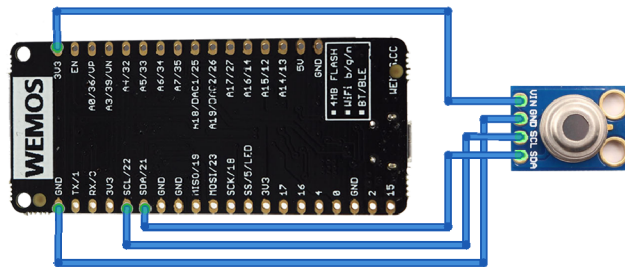
K sestavení senzoru ke snímání teploty je využita deska WeMos D1 mini, blíže popsána v části 2.1.3. Platforma disponuje modulem ESP-WROOM-32, který obsahuje čip ESP32. Díky čipu ESP lze komunikovat s platformou přes technologii Wi-Fi, Bluetooth a BLE. Jako snímač teploty objektu je použit infračervený teplotní snímač. Napájení DHT11 je zajištěno jeho připojením k platformě přes pin označený „3V3“, který zařízení napájí 3,3 V (viz obrázek 3.7). Pro příjem naměřených hodnot ze snímacího modulu je připojen k vývojové desce k digitálnímu pinu 17. Platformu lze přímo napájet bateriemi díky vestavěnému slotu na baterie.



Obrázek 3.7: Schéma zapojení senzoru WeMos-Battery_DHT11.

Senzor na měření teploty Lolin32_Melexis

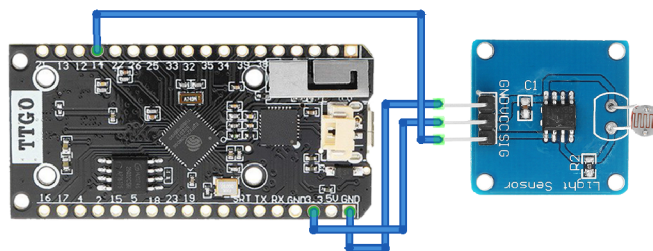
Senzor je založen na vývojové desce Lolin32, který je blíže popsán v sekci 2.1.3. Platforma obsahuje zabudovaný modul ESP-WROOM-32, který umožňuje zařízení komunikaci protokoly využívajícími technologie Wi-Fi, BLE a Bluetooth. Pro snímání intenzity osvětlení je k platformě připojen snímač Melexis MLX90614. Snímač je napájen z pinu „3V3“ vývojové desky. Komunikace se snímačem probíhá přes rozhraní I2C, bylo tedy třeba zvolit platformu, která tuto komunikaci umožňuje. Jak lze vidět na obrázku 3.8, k desce je snímač připojen k pinům SDA (pin 21) a SCL (pin 22). Zařízení lze přímo napájet na baterii díky zabudovanému konektoru na připojení baterie PH-2 2.0mm.



Obrázek 3.8: Schéma zapojení senzoru LoRa_DHT22, které zobrazuje připojení snímače MLX90614 k pinům SDA, SCL a napájecímu pinu platformy WeMos Battery za účelem zprostředkování komunikace sběrnici I2C.

Senzor na měření intenzity osvětlení LoRa_light

Zařízení určené ke snímání intenzity osvětlení pracuje na platformě LILYGO TTGO LoRa32. Deska umožňuje komunikaci přes technologie BLE, Bluetooth a Wi-Fi díky čipu ESP32 obsaženém v modulu ESP-WROOM-32 a přes technologii LoRa, která je zpřístupněna díky vestavěnému čipu SX1276. Ke snímání intenzity světla je použit přípojný modul, s vestavěným LDR fotorezistorem, který je napájen z platformy LoRa přes pin označený „3V3“, který poskytuje snímači napětí 3,3 V. Naměřené hodnoty jsou přijímány na analogový pin 14 (viz obrázek 3.9). Napájení celé platformy na bateriích lze uskutečnit přes připojení baterie na konektor vestavěný do desky LoRa.



Obrázek 3.9: Schéma zapojení senzoru LoRa_DHT22, zobrazující připojení modulu s foto-rezistorem k analogovému pinu 14 platformy TTGO LoRa za účelem zasílání dat a k pinu „3V3“ k napájení snímače.

3.2 Návrh úsporných síťových topologií

Základem každé síťové topologie jsou dříve navržené senzory (viz podkapitola 3, které je třeba zvolit vhodně dle požadované funkcionality. Je třeba se zaměřit především na vestavěné moduly, umožňující komunikaci přes požadované protokoly. Navržené senzory obsahují moduly, které umožňují komunikaci přes protokoly založené na technologii LoRa, Wi-Fi, Bluetooth a BLE. V tomto případě je třeba také zvolit vhodné skupiny senzorů, které budou schopny komunikovat nad alespoň jedním, pro danou skupinu zvoleným, protokolem.

Druhým důležitým aspektem při návrhu sítě založené na úsporných senzorech je volba vhodných protokolů. Pro účely komunikace v síti senzorů je třeba ověřit kompatibilitu s aplikačními protokoly jako je například protokol MQTT. Dále je třeba za účelem úspory energie zvolit vhodné protokoly linkové vrstvy pro zajištění komunikace na této úrovni.

Volba vhodné síťové topologie je pro WSN kritická. Je třeba volit topologie na základě požadavků na danou síť. Ty se odvíjí podle počtu připojených zařízení, vzdálenosti mezi jednotlivými uzly, stability a robustnosti sítě. Často je kladen důraz na nízkou spotřebu zařízení. Dalším požadavkem může být potřeba se vypořádat s výpadkem uzlu. Aby síť mohla dosahovat požadované kvality, je třeba pro ni zvolit odpovídající topologii.

Síťová topologie pro kontrolu počasí vně budovy (Topologie č.1)

Následující síť je sestavena z několika senzorů typu LoRa. Účelem sítě je zasílat údaje nasnímané ze senzoru na větší vzdálenosti. K sestavení jsou proto využity senzory schopné vysílat na vzdálenost v řádu km. Smyslem sítě je sledování parametrů jako teplota vzduchu uvnitř a vně budovy, intenzitu světla ve zvolené oblasti a množství srážek. Kombinací těchto parametrů je možné vytvořit přehled o vnějším prostředí, se srovnáním teplot uvnitř a venku.

Centrální bod sítě

K měření požadovaných veličin v síti slouží následující senzory, blíže popsané v kapitole 3, popisující návrhy úsporných senzorů. Ústředním prvkem je senzor LoRa_DHT11, který slouží jako přístupový bod k vnější síti. U tohoto senzoru není výhodné zavádět úsporné režimy, jelikož je třeba aby přístupový bod neustále očekával zprávy od odesílajících zařízení. Přístupový bod je tedy stále připojen ke zdroji elektrické energie. Je možné přístupový bod zároveň používat jako senzor, při připojení a zprovoznění navrženého snímače DHT11.

Senzory určené ke snímání hodnot

Prvním senzorem, určeným k odesílání naměřených dat do centrálního bodu je LoRa_DHT11, sestavená k snímání teploty uvnitř budovy. Druhý využitý senzor je LoRa_DHT22, který je použit na zaznamenávání teploty vnějšího prostředí. Dále je ve vnějším prostředí nasazen senzor LoRa_light, který slouží pro detekci množství světla v určené oblasti. Posledním senzorem využitým v dané topologii je LoRa_rain, který daném prostředí umožňuje monitorovat množství dopadených srážek.

Specifikace použitých senzorů

Všechny zmíněné senzory (kromě centrálního bodu) jsou napájeny bateriemi přes vestavěné konektory. Jelikož jsou zvolené senzory založeny na čípech ESP, využívají spánkových režimů v době nečinnosti. Za účelem dosažení co největší úspory energie, budou po sestavení, nasazení do prostředí a otestování režimy spánku optimalizovány.

Komunikační protokol

Zařízení s mikročipem SX1276 umožňují použití upravené fyzické vrstvy protokolem LoRa, který zajišťuje komunikaci ve zvoleném pásmu. Komunikace probíhá na vysoké vzdálenosti a zajišťuje přenos malých objemů dat s výsledkem nízké spotřeby. Síť je zapojena do topologie typu hvězda za účelem minimalizovat množství dat zasílaných skrze jedno zařízení a zvyšovat tak výdrž zařízení na bateriích. Centrální uzel této topologie je ale zároveň kritickým prvkem, protože při jeho výpadku přestane komunikovat celá síť. Všechna data jsou tedy přijímána přístupovým bodem v podobě LoRa_DHT11 senzoru, odkud jsou dále odesílána přes technologii Wi-Fi mimo lokální síť a publikována například na serveru MQTT.

Síťová topologie pro kontrolu teplot v domě (Topologie č. 2)

Tato topologie poskytuje vhodné propojení senzorů uvnitř budovy za účelem poskytování informací o stavu teplot v různých pokojích a objektech v nich. Ke snímání teplot vzduchu jsou využity snímače DHT11 a DHT22. Dále pro snímání teploty objektu je využit snímač Melexis MLX90614. U MLX90614 je třeba využít platformu, která podporuje komunikaci přes sběrnici I2C. Výsledkem sítě by měl být přehled o teplotě několika místností a jednoho objektu.

Centrální bod sítě

Přístupovým bodem sítě je v tomto případě senzor DevKit_DHT11, specifikovaný v podkapitole 3.1. U senzoru nejsou využity spánkové módy ani jiné způsoby šetření energie, protože je třeba neustále očekávat zprávy od ostatních senzorů. Senzor je z důvodu vysoké spotřeby připojen ke stálému zdroji elektrické energie.

Senzory určené ke snímání hodnot

Ke snímání teploty jsou určeny senzory založené na rozdílných platformách. Prvním senzorem je Lolin32_DHT11, který už podle názvu snímá teplotu prostředí díky snímači DHT11. Další teplotní senzor je LoRa_DHT11, který rovněž používá k měření teploty senzor DHT11. Dalším využitým senzorem je WeMos-Battery_Melexis, který zprostředko-

vává měření teploty konkrétního objektu. Posledním senzorem je LoRa_DHT22, využívající novější verzi senzoru DHT — DHT22.

Specifikace použitých senzorů

Za účelem spoření energie jsou zavedeny spánkové režimy. Ty jsou umožněny u všech zařízení díky čipům ESP. Napájení senzorů založených na platformě LoRa a Lolin32 je možné uskutečnit pomocí vestavěných konektorů pro připojení baterie. Senzor označený jako WeMos_Battery_Melexis je možné napájet bateriemi přímo díky vestavěnému slotu pro baterie.

Komunikační protokol

Ke komunikaci je využit úsporný protokol založený na technologii BLE. Ten umožňuje zasílat data odlehčeným způsobem, čímž zvyšuje výdrž zařízení na baterii. Jeho odhadovaný dosah, který činí více než 100 m, je dostačující pro využití v rámci jedné budovy. Aby mohla být technologie BLE využita, je třeba pracovat se zařízeními, které ji podporují. Základní podmínkou je tedy, aby byly senzory založené na mikročipech ESP32, což všechny navržené senzory splňují.

Senzory budou rozmístěny rovnoměrně po budově tak, aby byly vždy v dosahu přístupového bodu. Rozmístění v síti je rozvrženo do topologie typu hvězda. Data jsou dále z centrálního bodu zasílány přes Wi-Fi a protokol MQTT na MQTT server, odkud mohou být dále čteny nebo zpracovány uživatelem

Síťová topologie na monitorování stavu pokojové rostliny (Topologie č. 3)

Jedná se o topologii vytvořenou pro sledování podmínek a prostředí pokojové rostliny. Při této topologii je třeba měřit vlhkost půdy, intenzity orosení listů, vlhkost okolního vzduchu, teplotu vzduchu a intenzitu světla dopadajícího na listy.

Centrální bod sítě

Ústředním bodem topologie je senzor DevKit_DHT22. Kvůli neustálé potřebě zařízení očekávat zprávy od ostatních senzorů, je třeba jej udržovat neustále v aktivním módu. Nejsou u něj tedy využity žádné režimy spánku. Zařízení je z důvodu neustálého provozu připojeno stále ke zdroji napájecího napětí.

Senzory určené ke snímání hodnot

Síť umožňuje snímání intenzity orosení listů prostřednictvím zařízení Lora_rain, charakterizovaného v podkapitole 3.1. Ke snímání intenzity osvětlení je využit snímač WeMos_light. Dále je na platformě WeMos založen i další senzor WeMos_DHT11, který umožňuje měřit teplotu a vlhkost vzduchu v okolí rostliny. Posledním zařízením využitým v síti je kontrolní senzor LoRa_DHT11, který taktéž slouží k měření teploty a vlhkosti vzduchu, čímž potvrzuje naměřené hodnoty senzorem WeMos_DHT11.

Specifikace použitých senzorů

Za účelem měření spotřeby a její následné optimalizace jsou senzory napojeny na napájení přes baterie. Z toho důvodu je třeba zajistit senzorům úsporu energie. Ta je zajištěna díky vývojovým deskám senzorů s čipy ESP32, které umožňují nastavit spánkové režimy. Ty jsou

nastaveny v závislosti na vhodném poměru a zaslání dat a uvádění zařízení do úsporného režimu. Je třeba zohlednit přístup buzení zařízení pouze v případě detekované změny či po předem daném časovém intervalu.

Komunikační protokol

Mezi senzory jsou zařízení založená na rozdílných platformách, podporujících rozdílné způsoby komunikace. Jediná společná technologie, kterou podporují všechna zahrnutá zařízení je Wi-Fi. Protokolem na zajišťujícím přenos dat je ESP-NOW. Díky jeho provozu bez potřeby navázat spojení a přenosu menších packetů je úsporným řešením pro zařízení podporující technologii Wi-Fi. Sensorová síť je založena na topologii typu hvězda (viz podkapitola 2.3). Tato topologie je vhodná díky malým vzdálenostem mezi senzory.

Kapitola 4

Implementace zvolených senzorů nad navrženými topologiemi

Následující kapitola se věnuje realizaci jednotlivých senzorů, které byly navrženy v podkapitole 3.1 a jejich následnému propojení do navržených topologií z podkapitoly 3.2. V následujících podkapitolách jsou blíže probrány nástroje využívané k implementaci daných prvků. Jsou zde přiblíženy nástroje užívané pro vývoj řídicích programů senzorů, pro jejich zasílání dat na MQTT server. Dále jsou zde postupně rozebírány sestavení jednotlivých zařízení a jejich vnitřní logika a průběh komunikace.

Vyvojové prostředí PlatformIO a verzovací systém git

Senzory jsou vytvořeny v jazyce C++ za využití vývojového prostředí PlatformIO IDE¹. Tento open source nástroj byl zvolen na základě jeho správy knihoven a nativní podpory všech využitých platforem v této práci. PlatformIO je možné nainstalovat jako plugin jiného vývojového prostředí, v mém případě bylo přidáno jako součást Visual Studio IDE. Další přívětivou funkcí PlatformIO pluginu je integrace s verzovacím systémem git.

Pro správu verzí byl využit verzovací systém git a repozitář GitLab². Repozitář obsahuje složky, kde každá z nich odkazuje na jednu navrženou topologii. Jednotlivé složky topologií obsahují soubory pro konkrétní složky sítě. Samotné projekty dedikované jednotlivým souborům stačí tedy otevřít ve vhodném prostředí (například Platformio), zkompilovat a nahrát přes mikro USB do zařízení.

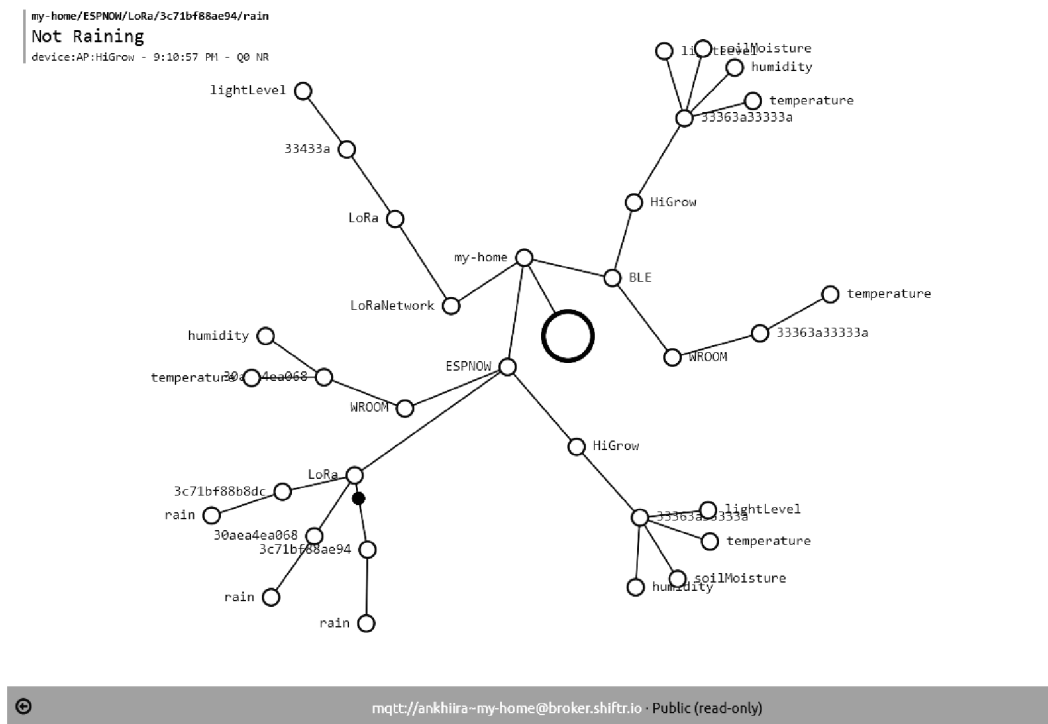
Shiftr.io

Společná část všech senzorových topologií spočívá v komunikaci směrem od přístupového bodu (gateway) zařízení v jednotlivých sítích. Jde o zasílání zpráv na MQTT broker, kde jsou dále jednotlivá data zobrazována v požadovaném formátu. Tento způsob byl využit především pro testování gateway zařízení a k simulaci reálného využití senzorové sítě, jelikož je možno data odsud dále číst, nebo zpracovávat. Jako MQTT broker jsem použila volně dostupný broker od platformy shiftr.io, která zdarma nabízí interaktivní webové rozhraní pro vizualizaci odeslaných a přijatých MQTT zpráv. Jednotlivé zprávy jsou na základě témat (*topic*) řazeny do jednotlivých řetězců a vyobrazují tak strukturu sítě (viz. obrázek 4.1). Tímto způsobem je také možno sledovat topologii včetně zasílání zpráv v reálném čase i

¹<https://platformio.org/>

²<https://about.gitlab.com/>

mimo lokální síť. V dalších záložkách je také možnost sledovat vykreslené grafy časového průběhu přijatých hodnot a aktivitu zařízení, které danou veličinu zaslá.



Obrázek 4.1: Shiftr.io — ukázka zobrazení zpráv MQTT v prostředí shiftr.io. Jednotlivá vlákna značí jednotlivé veličiny, přijaté snímači identifikovatelnými svými **MAC adresami**.

Formát zpráv zasílaných na server MQTT

Z obrázku je zřetelný formát zpráv, která jsou skládána do tzv. témat (*topics*). Téma je definováno na straně odesílatele zprávy (*publisher*) a přijímací zařízení (*subscriber*) se poté může přihlásit k jeho odběru. Mnou vytvořená témata mají následující strukturu. První položkou tématu je označení celého jednotného projektu, v tomto případě označuje soubor sítí nazvaných jako *my-home*. Další částí řetězce je rozlišení na jednotlivé sítě dle jednotlivých protokolů, na kterých jsou založeny. V mém příkladu jde například o síť založenou na technologii ESP-NOW, označenou jako *ESPNOW*. Po určení sítě je dalším uzlem specifikován druh platformy, ze které zpráva přišla. V obrázku 4.1 je vidět například právě přicházející zpráva od zařízení založeného na platformě TTGO LoRa. Další položkou je identifikační číslo sestavené z MAC adresy odesílatěho zařízení. Poté následuje typ proměnné, která byla naměřena a řetězec je zakončen samotnou naměřenou hodnotou. V našem případě byla hodnota převedena na namapovaný stav — „Not Raining“ (neprší).

Struktura pro ukládání a zasílání naměřených veličin

Pro účely zasílání naměřených hodnot je vytvořena struktura, která má pevně daný formát a velikost pro všechna zařízení v síti. Tím je umožněna pohodlná detekce zařízení, typu a hodnot naměřených veličin. Složení struktury je následující. První položkou je *char deviceType*, která slouží pro uložení jednoho charakteru indikujícího typ platformy. Jednot-

livé typy platforem a jejich značení znázorňuje tabulka 4.1. Druhá položka je *char type*, ta slouží pro uchování typu hodnoty, kterou daný senzor měří (viz. tabulka typů naměřených hodnot 4.2). Další položka je *float value* a slouží pro uchování hodnoty naměřené senzorem. Poslední položkou je *char devMac[6]*, která obsahuje šestimístné pole pro uložení MAC adresy zařízení. Struktura je dalších dvou topologií mírně odlišná z důvodu potřeby/nepotřeby zasílání výše zmíněných informací.

```

struct __attribute__((packed)) SENSOR_DATA {
    char deviceType;
    char type;
    float value;
    char devMac[6];
} sensorData;

```

Tabulka 4.1: Zkratky platforem využívané při zasílání dat ze senzorů slouží pro identifikaci jednotlivých zařízení, které jsou do sítě zapojeny.

platforma	zkratka
Wemos D1 mini	w
ESP32-DevKit	d
WeMos Battery ESP32	b
LoRa	l
Lolin32	n

Tabulka 4.2: Zkratky naměřených veličin jsou zasílány ve strukturované zprávě od senzoru, kde identifikují druh naměřené veličiny.

veličina	zkratka
teplota	t
vlhkost vzduchu	h
světlo	l
množství srážek	r
teplota objektu	o

Struktura řídicích programů senzorů

Činnost senzorů je dána řídicím programem následující struktury. Program se skládá z tří hlavních částí — definice pinů a proměnných, inicializační fáze označená jako *setup()*, která je vykonávána pouze jednou po restartu zařízení a periodicky se opakující fáze *loop()*, která je spuštěna bezprostředně po inicializační fázi a je realizována jako nekonečná smyčka. Při vývoji vlastní sítě jsem použila dva odlišné průběhy. Zařízení, která nepřecházejí do režimu spánku (přístupové body), procházejí nejprve inicializační fázi a následně běží v nekonečné smyčce jako je popsáno výše. Sensory, které jsou napájeny baterií a využívají režimy spánku, fungují pouze ve fázi *setup()*, protože jádro programu probíhá většinou pouze jedenkrát. V této fázi inicializují potřebné proměnné, změří veličiny, odešlou data a přejdou do režimu spánku.

4.1 Implementace topologie č. 1

Vytvoření topologie pro kontrolu vnitřního a vnějšího prostředí spočívá v sestavení senzorů pro snímání teploty uvnitř a vně stavby, množství osvětlení a detekci deště vně budovy. Topologie je zapojená do rozložení typu hvězda s jedním centrálním přijímačem. Síť je založená na technologii LoRa.

Sestrojení senzorů

K vytvoření senzorů topologie byly využity vývojové platformy **LILYGO TTGO LoRa32** s čipem SX1276. Dále byly k platformám postupně připojeny senzory ke snímání teploty modul DHT11, modul obsahující senzor DHT22, fotorezistor LDR zabudovaný ve snímači intenzity osvětlení a srážkový modul se snímačem označeným YL-B3. Pro připojení snímačů a platformy bylo využito nepájivé kontaktní pole, snímače byly k platformám připojeny odnímatelnými vodiči. Takto zvolené zapojení je použito z důvodu jeho využitelnosti při budoucím testování spotřeby proudu zařízení. Jako zdroj napětí vývojové desky slouží lithiová baterie, blíže specifikována v sekci 5.2.

Zprostředkování komunikace přes technologii LoRa

Senzory se v komunikaci přes technologii LoRa dělí na dvě skupiny — odesílací (*sender*) a přijímací (*receiver*) zařízení. Zařízení typu *receiver* slouží zároveň jako přístupový bod a předává zprávy dále mimo lokální síť LoRa. V topologii č.1 se tento typ zařízení vyskytuje pouze jednou a slouží pro přijímání zpráv odeslaných všemi ostatními senzory typu *sender*. Všechna odesílací zařízení slouží pro snímání hodnot a následné zasílání do přístupového bodu.

K realizaci komunikace byla využita knihovna *RadioHead*³, která umožňuje zasílání zpráv přes běžná radiová pásma, včetně pásem pro technologii LoRa. K tomu bylo třeba využít ovladač čipu SX1276 — označený jako *RH_RF95*⁴.

Zařízení typu *sender*

Nyní se zaměříme na strukturu programu tvořící jádro senzoru typu *sender*. Řídící program odesílacího zařízení pracuje ve třech fázích — měření hodnot, odesílání dat a režim spánku. První částí programu je konfigurace zařízení a snímače hodnot, jsou nastaveny používané piny, probíhá příprava snímače na čtení hodnot a inicializace komunikace přes technologii LoRa. Dále je třeba pomocí vhodných knihoven obsahujících ovladače snímače veličin přečíst digitální hodnoty senzorů. V našem případě pro práci se snímači DHT11 a DHT22 je využita knihovna *DHT.h*. Analogové hodnoty jsou čteny přímo v pinu, na který je snímač napojen a poté namapovány na jejich odpovídající digitální reprezentaci. Takto získané hodnoty jsou dále uloženy do předem stanovené struktury (viz podkapitola *Struktura pro ukládání a zasílání naměřených veličin*). do struktury jsou poté přidány zkratky identifikující druh platformy, ze které je zpráva odesílána, MAC adresa zařízení a druh zasílané veličiny. Poté je inicializována knihovna pomocí

Zařízení typu *receiver*

Zařízení typu přijímač nejprve nakonfiguruje ovladač RF čipu s využitím knihovny *RadioHead* a poté jej se zvolenými parametry inicializuje ve fázi *setup()*. Následně přechází do fáze *loop()*, kde při odečítání časovače vyčkává požadovanou dobu na příjem zpráv od ostatních senzorů. Pokud přijme zprávu od zařízení, uloží celý její obsah do předem stanovené struktury. Tu poté přidá do vektoru struktur. Po uběhnutí požadované doby přepne senzor z režimu snímání a zpracuje všechny uložené přijaté zprávy z vektoru struktur a připraví je na odeslání do vnější sítě. V této fázi za využití knihovny pro práci s Wi-Fi *Wi-Fi.h*⁵

³<https://github.com/PaulStoffregen/RadioHead>

⁴https://www.airspayce.com/mikem/arduino/RadioHead/classRH_RF95.html

⁵<https://github.com/espressif/arduino-esp32/blob/master/libraries/Wi-Fi/src/Wi-Fi.h>

nastaví připojení k síti Wi-Fi a Wi-Fi. Poté inicializuje spojení s MQTT serverem, ke kterému jsem využila knihovny *PubSubClient*⁶. Po úspěšném přihlášení k serveru se pokusí odeslat zpracovanou zprávu v podobě tématu (*topic*) na MQTT broker. Témata zasílané topologií č. 1 jsou identifikována druhou položkou řetězce, která je v tomto případě nazvána *LoRaNetwork*, pro indikaci sítě využívající technologii LoRa.

4.2 Implementace topologie č. 2

Síť určená pro měření teplot v budově je založena na teplotních senzorech se snímači DHT11 a DHT22, komunikujících za využití protokolu BLE. Síť se skládá z pěti sensorů — čtyři vysílače a jeden přijímač. Sensory jsou rozestavěny do topologie typu hvězda s přístupovým bodem postaveným tak, aby byl v dosahu všem vysílačům. Zařízení jsou rozdělena na jeden přijímač („client“) a čtyři vysílače („server“).

Sestavení jednotlivých zařízení

V topologii č. 2 je přijímacím zařízením DevKit_DHT11. Přijímací zařízení je neustále připojeno ke zdroji elektrické energie. Důvod pro toto stálé zapojení je potřeba neustále skenovat prostor a očekávat přijaté zprávy od sensorů. Zařízení je možné po synchronizaci s ostatními senzory v síti na určitý čas přepnout do režimu spánku, ale při aktuální potřebě častého skenování sítě a posílání zpráv na MQTT server přes Wi-Fi síť je třeba velké množství energie, proto nemá i tak smysl napájet platformu bateriemi.

K sestavení vysílačů teploty LoRa_DHT11 a LoRa_DHT22 byla použita platforma **LLYGO TTGO LoRa32**. Tato deska byla zvolena, protože je založena na čipu ESP32, který podporuje BLE komunikaci. Zároveň deska obsahuje obvod s konektorem pro připojení a napájení baterií. Baterii je také možno přes tento okruh nabíjet, ale pouze u této verze desky. K měření hodnot slouží připojené moduly se snímači DHT11 a DHT22, které zasílají naměřená data v digitální podobě. Modul se snímačem DHT11 obsahuje již zabudovaný odpor a LED diodu. Moduly jsou pro přenos dat připojeny k digitálním pinům „GPIO16“. Senzor Lolin32_Melexis je založen na platformě **WeMos Lolin32**. Ke snímání je využit modul obsahující snímač MLX90614. Posledním senzorem je WeMos-Battery_DHT11 založené na platformě WeMos Battery ESP32, který ke snímání hodnot využívá modul se snímačem DHT11. Všechny platformy umožňují přímé napájení přes baterie, v podobě konektorů nebo slotů. K propojení snímačů a platform jsou využity odnímatelné vodiče.

Zprostředkování komunikace přes technologii BLE

Síť založená na komunikaci přes protokol BLE obsahuje jedno přijímací zařízení, které zároveň slouží jako přístupový bod. Komunikace probíhá na základě navázání spojení mezi přijímačem a vysílačem. Přijímací zařízení se v komunikaci přes technologii BLE nazývá *client*. Sensory snímající hodnoty prostředí jsou rozmístěny vhodně na dosah *client* uzlu a jsou nazývána *server*. V následujících sekcích je popsána činnost obou typů zařízení.

K implementaci byla využita knihovna *ESP32_BLE_Arduino*⁷ jejíž autorem je Neil Kolban. Jelikož je knihovna pro práci s BLE velká, není možné s ní bez případných úprav pracovat. Ve mnou využívaném prostředí PlatformIO jsem tento problém vyřešila zvětšením

⁶<https://github.com/knolleary/pubsubclient/blob/master/src/PubSubClient.h>

⁷https://github.com/nkolban/ESP32_BLE_Arduino

přiděleného prostoru v konfiguračním souboru `platformio.ini` příkazem `board_build.partitions = no_ota.csv`.

Zařízení typu *server*

Každý server nejprve inicializuje proměnné pro zahájení komunikace přes technologii BLE. Vytvoří instanci *BLEserver*, vytvoří *BLEservice*, kterému je přiřazeno náhodně vygenerované UUID. UUID slouží k označení služby, která značí odesílání měření ze serveru. Pro tento service vytvoříme dále dvě charakteristiky, které slouží k identifikaci, zdali se jedná o odeslanou nebo přijatou hodnotu. Vytvořeným charakteristikám se opět přiřadí příslušná UUID, která budou dále sloužit také pro identifikaci zařízení a typu hodnoty, která je zasílána (popřípadě přijímána). Poté spustíme vytvořený service a zařízení přepneme do režimu advertising, ve kterém očekává na spojení při vysílání nabízené služby. Server má také rovnou nastavené chování v případě navázání spojení a přijetí zprávy ve funkcích reagujících na příjem zprávy (tzv. *callbacky*). Senzor poté naměří hodnotu svého snímače, kterou následně uloží do vytvořené charakteristiky a následně ji přes funkci *notify()* zašle na zařízení typu *client*. Po detekci zprávy oznamující ukončení spojení, kterou může zasílat pouze *client*, senzor typu *server* přejde do režimu spánku.

Zařízení typu *client*

Client, neboli cílové přijímací zařízení, funguje následovně. Nejprve vytvoří instanci třídy *BLEScan*, který hledá pod předem stanoveným UUID požadovaný *service*. Tento *service* identifikuje hledanou službu, v našem případě měření ze snímače. *Client* periodicky čeká 20 s na příchozí zprávy a zapíná každé 2 s sken hledajícího zařízení, které požadovanou službu nabízí. Po nalezení se senzorem naváže spojení a zastaví skenování sítě. Dále z nalezeného zařízení získá jméno, které obsahuje předem domluvenou zkratku platformy, na které je senzor založen. Poté přečteme data požadované charakteristiky (určené na příjem dat) a uložíme je do vytvořené struktury. do struktury dále přidáme další požadované údaje jako je MAC adresa odesílacího zařízení a typ přijímané hodnoty a uložíme ji do vektoru struktur. Po přečtení všech požadovaných údajů se odpojí od zařízení s označením *server*. Poté opět zapne skenování a čeká na ustanovení spojení s dalším zařízením typu *server*. Po uběhnutí 20 s se přijímací zařízení, které je zároveň přístupovým bodem, připojí k síti Wi-Fi a pokusí se zaslát zprávu na server MQTT, ve formátu tématu (*topic*).

4.3 Implementace topologie č. 3

Třetí síť, sloužící pro měření veličin kolem pokojové rostliny, využívá k monitorování prostředí hodnoty tepelných, světelných a srážkových senzorů. Zapojení je formou topologie typu hvězda s jedním centrálním uzlem.

Vytvoření jednotlivých senzorů

Centrálním bodem sítě je senzor `DevKit_DHT22`. Zařízení slouží jako *gateway* a je založeno na platformě `ESP32-DevKit`. Jeho přednostní využívání pro přístupový bod je z důvodu absence přímého způsobu napájení senzoru bateriemi. Je neustále připojen ke elektrické energii. Prvním senzorem určeným ke snímání hodnot je `LoRa_rain`, který je implementován za využití platformy **LILYGO TTGO LoRa32**. K němu je připojen modul `MH-RD` se snímačem `YL`. Snímač zasílá analogová data informující o intenzitě srážek na

analogový pin 27. Druhý použitý senzor LoRa_DHT11 je taktéž založený na platformě **LLYGO TTGO LoRa32**, disponující snímačem DHT11. Další dva senzory WeMos_DHT11 a WeMos_light jsou implementovány s využitím platformy WeMos D1 mini v3.0.0 a v2.2.0. Senzor WeMos_DHT11 má připojen modul se snímačem DHT11. Ke snímání intenzity osvětlení je k zařízení WeMos_light připojen analogový modul s fotorezistorem LDR. Platformy jsou propojeny s moduly přes odnímatelné vodiče a napájeny jsou v případě zařízení LoRa_rain přes přímý bateriový konektor a senzory WeMos_DHT11 a WeMos_light přes WeMos Battery Shield.

Zprostředkování komunikace přes technologii ESP-NOW

Komunikace probíhá mezi zařízeními, které vysílají (tzv. *slave*) a přijímacím zařízeními (tzv. *master*). Protokol ESP-NOW je komunikačním protokolem postaveným na technologii Wi-Fi. Je tedy nutné, aby všechna zařízení zúčastněná v komunikaci obsahovala Wi-Fi čip. Jelikož jsou implementované platformy založené na čípech ESP32 a ESP8266, je tato podmínka splněna.

Zařízení typu *master*

Komunikace přes ESPNOW probíhá následujícím způsobem. Nejprve si musí zařízení nastavit MAC adresu, pod kterou bude viditelný pro všechna zařízení. Tato MAC adresa bude poté pevně stanovena i u všech vysílacích zařízení, pro urychlení komunikace. Zabrání se tím potřebě stále adresu zasílat všem zařízením v síti a zároveň potřebě nejprve adresu přijmout na straně koncových zařízení. Poté je třeba v zařízení nastavit reakce na zvolené chování pomocí tzv. *callbacks*, na základě kterých bude zařízení typu *master*, čekat na příchozí zprávy od vysílacích platform typu *slave*. V případě detekce příchozí zprávy si gateway uloží dané zařízení do tabulky peerů, pošle potvrzení danému *slave* a následně uloží zprávu do předem stanovené struktury. Poté uloží zprávu do vektoru struktur přijatých zpráv a dále naslouchá dalším zařízením. Po předem stanoveném časovém intervalu (v našem případě 20 s), přejde do fáze preposílání zpráv mimo lokální síť. V této části se připojí senzor na síť Wi-Fi a k MQTT serveru, poté zpracuje přijaté zprávy do témat (zpráv ve formátu vhodném pro zasílání na server MQTT), která následně odešle na MQTT server.

Zařízení typu *slave*

Snímače (*slaves*) pracují v síti ESPNOW na následujícím principu. Každé zařízení si ve fázi nastavení uloží do tabulky peerů přidá pouze MAC adresu Master zařízení, na které bude odesílat naměřené hodnoty. Zařízení mají tedy předem uloženou MAC adresu přijímače (*master* zařízení). Poté senzor naměří hodnoty snímače a uloží je do předem specifikované struktury. Tuto strukturu doplní o informace sdělující typ zasílané veličiny, typ platformy, ze které je veličina zasílána a MAC adresu odesílacího zařízení. Následně odešle zprávu na dříve uloženou MAC adresu ze seznamu peerů. Po odeslání dat se senzor přepne do režimu spánku.

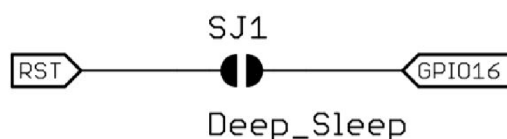
Provoz ESP-NOW a Wi-Fi na jednom čipu

Senzor uskutečňující komunikaci přes protokol Wi-Fi a ESP-NOW, nemůže spolehlivě pracovat oběma technologiemi zároveň. V našem případě, pokud senzor po skončení přijímání zpráv a odeslání dat přes Wi-Fi chce opět přejít na přijímání dat protokolem ESP-NOW,

není mu tato možnost zpřístupněna. ESP-NOW zprávy už nejsou při opětovném zapnutí komunikace přijímány. Tento problém je řešen v následujícím projektu⁸. Existují dvě následující řešení — první možností je sestavit další dodatečný senzor, který pouze přijímá data od zařízení typu *master* po sériové lince, a poté je posílá mimo lokální síť, zatímco *master* komunikuje pouze protokolem ESP-NOW. Druhou možností je restart přijímacího zařízení po každém skončení komunikace přes Wi-Fi technologii. V implementaci této sítě jsem si zvolila možnost s restartováním čipu funkcí *ESP.restart()*, volanou po každém odeslání dat na server MQTT.

Automatický restart WeMos D1 mini

Při snaze o zprovoznění senzorů založených na platformě WeMos D1 mini bylo třeba vyřešit problém s nepřecházením zařízení do režimu, ve kterém je zařízení možno naprogramovat a při provozu bylo třeba vyřešit problém, že se zařízení po dokončení deep-sleep režimu samo nespustilo. Pro napravení těchto problémů a zajištění správné funkcionality platformy jsem zjistila, že je třeba při restartu (popřípadě naprogramování) desky pin „GPIO16“ (označovaný také jako pin „D0“) držet v nízké úrovni napětí. Tento problém je možné vyřešit připojením pinu „GPIO16“ na pin „RST“. Verze 3.0.0 platformy WeMos D1 mini obsahuje dokonce předpřipravené spojení těchto pinů (viz obrázek 4.2), které poté stačí pouze spojit cínem pro umožnění jejich funkcionality.



Obrázek 4.2: Na obrázku lze vidět část schématu platformy WeMos D1 mini, která umožňuje po spojení vodivým kovem propojení mezi piny „GPIO16“ a „RST“. (Obrázek přejat z:https://wiki.wemos.cc/products:d1:d1_mini)

⁸<https://github.com/HarringayMakerSpace/ESP-Now>

Kapitola 5

Měření a optimalizace spotřeby, výpočet teoretické výdrže zařízení na bateriích

Následující kapitola se věnuje měření a optimalizování senzorů s cílem minimalizace jejich spotřeby. Kapitola se následně zabývá výpočtem teoretické výdrže zařízení při provozu na zvolené baterii. První podkapitola se věnuje shrnutí aplikovatelných optimalizací, kde každá sekce popisuje jeden způsob zavedení způsobu snížení spotřeby senzoru. Druhá podkapitola se zaměřuje na aplikaci těchto optimalizací a měření celkové spotřeby zařízení. V této části jsou shrnuty obecné způsoby, jakým byla zařízení měřena a jaké komponenty byly využity k jejich napájení. V sekcích jsou rozebrány účinky optimalizací a v tabulkách shrnuty naměřené hodnoty jednotlivých implementovaných topologií. Třetí podkapitola se zabývá výpočtem teoretické výdrže při provozu implementovaných zařízení na zvolené baterii. Podkapitola se zaměřuje na porovnání použitých zařízení, senzorů a komunikačních technologií. Výsledkem kapitoly je odhad výdrže navržených senzorů

5.1 Optimalizace implementovaných senzorů vzhledem k spotřebě energie

Po prostudování problematiky týkající se snahy o minimalizaci energie spotřebované bateriovými senzory při provozu, jsem našla několik způsobů, kterými je možné snížit spotřebu zařízení. Hlavním prostředkem, který rozhoduje o výdrži baterie, je použití režimů spánku. Díky nim je možné provozovat zařízení na bateriích. Dalším způsobem spoření zařízení při provozu na baterii je odesílání dat pouze, pokud se hodnota měřené veličiny výrazně změní. K úspoře je také možno využít zasílání zprávy přímo koncovému zařízení, místo odesílání zpráv všem zařízením v síti. Po prvotních testech jsem také přišla na problém s USB ovladačem, který je třeba explicitně uvést do režimu nečinnosti. Dále bylo na základě testování zjištěno, že ke snížení spotřeby je také třeba odpojit LED diody na snímačích a platformách. Ke snížení spotřeby je také důležité ukončení činnosti všech vnitřních modulů. Poslední testovanou optimalizací bylo snížení frekvence mikročipů platform. Výsledky použité následujících způsobů jsou poté popsány dále v podkapitole [5.2](#).

Využití režimu spánku

Použití režimů spánku způsobuje výrazný rozdíl v délce provozu zařízení na baterii. Při provozu zařízení stráví senzory většinu času čekáním mezi měřeními a zasíláním hodnot. Je tedy žádoucí snížit co nejvíce aktivitu zařízení v této fázi, což je dosažitelné zavedením režimů spánku. Jelikož jsou všechny použité senzory založeny na čipu ESP, je u nich možno tyto režimy aplikovat. U všech sensorů je při provozu využít deep-sleep mód s cyklickým uspáváním na předem zvolený časový interval. Při využití režimu deep-sleep bylo třeba nejprve zvolit způsob probouzení mikročipu. Ten byl v tomto případě zvolen na probuzení časovačem, který je nastaven funkcí `esp_sleep_enable_timer_wakeup()`, která je součástí knihovny `esp_sleep.h`¹. Funkci je jako parametr předávána délka intervalu režimu spánku v μs . Časový interval a způsob probuzení je třeba nastavit v inicializační fázi (fázi `setup()`) senzoru. Poté ve fázi `loop()` je po provedení čtení a odeslání hodnot zavolána funkce `esp_deep_sleep_start()`, čímž zařízení přejde do režimu spánku.

Odesílání na základě velikosti změny veličin

Dalším způsobem, jak snížit spotřebu energie zařízením, je korigované odesílání dat. To znamená nezasílat data pouze po daném časovém intervalu, ale měřit data po určitých časových intervalech, a na základě velikosti rozdílu hodnot oproti předchozím naměřeným hodnotám se rozhodnout, zda data odeslat či zařízení rovnou převést zpět do režimu spánku. Při využívání režimu spánku při provozu sensorů bylo třeba vymyslet způsob, jak hodnoty uchovávat i poté, co jsou všechny komponenty platformy odpojeny od napájení a data z klasické paměti jsou ztraceny. Tento problém jsem vyřešila uložením dat do paměti RTC, která je součástí čipů ESP a jako jediná po přechodu zařízení do deep-sleep režimu zůstává napájena a uchovává si tedy uložené hodnoty. Pro uložení proměnných do paměti RTC stačí požadovanou proměnnou označit předepsáním `RTC_DATA_ATTR` před datový typ proměnné.

Přepojení z USB na baterii

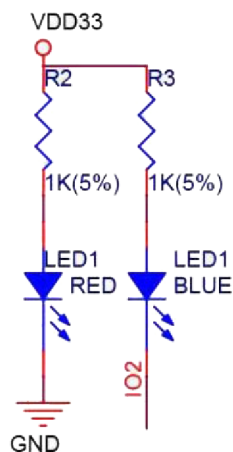
Při měření sensorů jsem se setkala s neočekávaně vysokou spotřebou sensorů při režimu spánku (kolem 12 mA). Po bližším prozkoumání jsem objevila příčinu a odpověď na následujícím fóru². Jako příčina je zde popsán nevyhovující design platformy při provozu na baterii, jelikož při napájení přímo z baterie USB converter zavedený v platformě přejde do `STANDBY_MODE` režimu, kde vyčkává na připojení mikro USB a to i když se zařízení uspí. Jednou možností, jak tento problém obejít bez fyzického zásahu do platformy, je připojit mikro USB i baterie na začátku provozu a poté mikro USB odpojit.

Odpojení LED diody

Další věcí, která způsobuje spotřebu energie v režimu spánku jsou LED diody indikující napájení zařízení. Napájecí diody pracují neustále a jsou připojeny přímo na okruh s napájecím napětím, jak lze například vidět na výseku ze schématu na obrázku 5.1 platformy ESP32-DevKit. Zde je vidět, že modrá dioda je ovladatelná přes pin „GPIO2“ ale červená dioda je připojena přímo k zemi, tudíž ji není možné softwarově vypnout. Je tedy třeba diody fyzicky z desky odstranit nebo alespoň přerušit jejich spojení s platformou.

¹https://github.com/espressif/esp-idf/blob/master/components/esp32/include/esp_sleep.h

²https://github.com/Heltec-Aaron-Lee/Wi-Fi_Kit_series/issues/6#issuecomment-398650440



Obrázek 5.1: Část schématu platformy ESP32-DevKit, která zobrazuje zapojení LED diod na desce. (Obrázek přejat z:<http://mh.nodebb.com/topic/2/mh-et-live-esp32-devkit/12>)

Režim spánku u všech modulů platformy

Při měření senzorů jsem zjistila, že některé moduly platformy nepřejdou, po volání funkce pro spuštění deep-sleep módu, do úsporného režimu. Proto je třeba u některých modulů volat dodatečné funkce pro ukončení jejich činnosti. Přidala jsem tedy do řídicího programu před spuštění režimu spánku následující funkce. Byla přidána funkce pro ukončení sériové komunikace `Serial.end()`, funkce pro odpojení Wi-Fi čipu od současné sítě `Wi-Fi.disconnect()` a pro úplné odpojení Wi-Fi funkce `Wi-Fi.mode(Wi-Fi_OFF)`, dále u senzorů komunikujících přes technologii BLE byla před režimem spánku volána funkce `BLEDevice::deinit(true)`, která deinicilizuje a ukončí bluetooth komunikaci. U zařízení komunikujících v síti LoRa byla před úsporným režimem volána funkce pro ukončení činnosti RF modulu `rf95.sleep()`.

Snížení frekvence MCU

V některých materiálech je uvedeno, že při nižší frekvenci procesoru je možné dosáhnout mnohonásobně nižší spotřeby energie. Snížením frekvence procesoru se ale zároveň prodlouží doba činnosti zařízení. Je tedy nutné otestovat, zdali tento způsob u konkrétních platformech umožní úsporu energie. Platformy využitě při sestavení senzorů jsou založeny na čipech ESP32 a ESP8266. U ESP32 je u platformech použit model ESP32-D0WDQ6, který operuje na základní frekvenci 160 MHz, kterou lze zvýšit na 240 MHz. U platformech s čipem ESP8266 je využit model ESP8266EX, jehož základní operační rychlost je 80 MHz a kterou lze zvýšit až na 160 MHz.

Odesílání cílené zprávy místo broadcast vysílání

Při zasílání dat a udržování senzoru v aktivním módu je spotřeba energie mnohonásobně vyšší. Je proto nutné udržet počet zasílaných zpráv na minimum. Toho lze docílit neodesláním zpráv všem zařízením ale pouze vysláním zpráv typu unicast cílovému zařízení. Tento případ lze řešit například prvotním zasláním zprávy s cílovou adresou všem budoucím měřícím zařízením, a ty poté dále už mohou zasílat data pouze na tuto adresu. Tato optimalizace

má především využití při použití topologie typu mesh. Jelikož jsou mé topologie navrženy do rozložení typu hvězda, není třeba u něj dále optimalizovat zaslání zpráv jednotlivých senzorů.

5.2 Měření spotřeby při provozu senzorů

Následující podkapitola se zabývá měřením sestavených senzorů. K měření proudu spotřebovaného zařízením byl použit multimetr UNI-T UT61E s kontrolou naměřených hodnot digitálním multimetrem SOLID RE-230C (V10). Implementované senzory byly napájeny několika způsoby. Pro měření spotřeby při provozu senzorů bylo třeba využít přímé napájení senzorů bateriemi nebo modul, který by napájení přes baterie simuloval. U zařízení, která přímo podporovaly nabíjení z baterie (tzn. obsahovaly konektor nebo slot pro připojení baterií) byly přímo napájeny lithiovou baterií, specifikovanou v sekci „Referenční zdroj napájení“. Další zařízení byla napájena přes breadboard power module (specifikovaný v sekci 2.1.4, připojený k nepájivému poli. Při měření byl střídavě využit i WeMos Battery Shield, pro potvrzení správnosti výsledků.

Při pokusu o měření přes slot na baterie vestavěný v platformách se začaly senzory nebezpečně přehřívat a začalo se z nich dýmat a hrozilo poškození zařízení, byly proto z bezpečnostních důvodů k napájení využity pouze vestavěné konektory a napájená nepájivá pole.

Zdroje napájení senzorů provozovaných na bateriích

Existuje několik druhů externích zdrojů, přes které je možno senzory napájet. Naším cílem je napájení senzoru, který pro svůj provoz potřebuje napětí kolem 3,3 V. Mezi nejvhodnější zdroje napětí platforem patří lithiové a lithium-polymerové baterie. Je možné také využít baterie typu LiFePO_4 , ale je udáváno, že dodávají o 70 % méně energie než lithiová baterie stejné velikosti³. Na druhou stranu výhodou oproti lithium-polymerovým bateriím je, že při špatném zacházení nebo nízké kvalitě nejsou náchylné k přehřívání.

Naopak nejméně vhodné je napájení senzorů přes powerbanku, protože pracuje na napětí 3,7 V, dále napájí senzor 5 V a ten musí zpět regulovat napětí na 3,3 V. Následkem těchto konverzí je vysoká spotřeba energie. Dalším problémem powerbank je jejich automatické vypínání v případě minimálního odběru proudu, který je těžké rozeznat při malé spotřebě zařízení. Vhodné nejsou ani NiMH baterie, jelikož při vysílání dat například přes Wi-Fi je třeba nárazově dodat několikanásobně vyšší proud v krátkých pulsech, což vede k poklesu napětí na ESP čipu a zařízení se tak resetuje.

Lithiové baterie je vhodné použít především protože zprostředkovávají vhodnou míru napětí — dvě baterie v sérii nebo jedna CRV123 poskytují 3 V, což je vhodné napětí pro napájení senzorů. Toto napětí poskytují stabilně až do přibližně posledních 10 % kapacity baterie. S lithiovou baterií se také senzor bez problémů vyrovná s krátkými impulsy vznikajícími v případě náhlého zaslání dat například přes Wi-Fi. Mají velmi nízké samovolné vybíjení a dokážou fungovat v teplotách až do $-20\text{ }^\circ\text{C}$. Například baterie Varta CRV123, která dodává napětí 3 V při 1700 mAh, může napájet senzor, který je v režimu nečinnosti až 5 let, v závislosti pouze na tom, jak často je senzor uspáván, nebo jak dlouho je připojen k síti a vysílá zprávy.

³<https://www.radioshuttle.de/en/media-en/tech-infos-en/battery-powered-esp32/>

Referenční zdroj napájení

Jako referenční zdroj napájení jsem si zvolila baterii ICR18650-28A⁴, která má stanovenou nominální kapacitu na 2800 mAh. Je třeba také brát v potaz, že tato hodnota udává kapacitu v ideálním stavu, při pracovní teplotě 25 °C. Její reálná kapacita je tedy nižší. Baterie byla připojena k platformám, které obsahovaly konektor k připojení lithiových baterií. Baterie byla nabita na 4,1 V a poskytovala napětí 3,7 V.

Výsledky měření implementovaných topologií

Po propojení zařízení do obvodu umožňujícího měření platformy v provozu (viz podkapitola 5.2) byly odměřeny hodnoty proudu odebíraných platformami. Naměřené hodnoty jednotlivých topologií jsou zahrnuty v následujících tabulkách.

Měření 1. topologie

První testovanou sítí byla topologie č.1 založená na technologii LoRa. Výsledky měření jsou shrnuty v tabulce 5.1. Sensory byly měřeny v základním nastavení při použití režimu deep-sleep. Z naměřených údajů lze vidět, že celková aktivní doba jednoho cyklu řídicího programu trvá průměrně 200 ms, což při nízkém počtu měření může znamenat zanedbatelnou hodnotu napětí v provozu. Spotřeba v režimu spánku byla měřena nejprve klasicky a poté po připojení obou zdrojů napájení (mikro USB a baterie) a následném odpojení baterie (viz podkapitola 5.1). Naměřené hodnoty před a po využití tohoto způsobu jsou zaznamenány ve třetím a čtvrtém sloupci tabulky 5.1.

Tabulka 5.1: V tabulce jsou zaznamenány naměřené hodnoty při provozu topologie č. 1, která je založená na topologii LoRa. První sloupec znázorňuje průměrnou dobu trvání aktivní části řídicího programu jednotlivých senzorů za jeden cyklus běhu, v druhém sloupci jsou naměřené průměrné hodnoty odběru proudu zařízení v provozu a třetí sloupec obsahuje hodnoty odebíraného proudu zařízením při režimu spánku. Ve čtvrtém sloupci jsou zaznamenány hodnoty po využití způsobu zapojení více zdrojů napětí a následném ponechání pouze napájení z baterie.

platforma	doba trvání aktivního režimu (ms)	spotřeba v provozu (mA)	spotřeba v režimu spánku před (mA)	spotřeba v režimu spánku po (mA)
LoRa_DHT11	198	111.8	11	3,16
LoRa_DHT22	197	60	10	2,24
LoRa_light	197	57	9	2,4
LoRa_rain	209	71.2	12	5

Měření 2. topologie

Druhá testovaná síť byla založená na technologii ESP-NOW. Sensory byly v základním nastavení pouze s využitím režimu spánku. Jak lze vidět z údajů naměřených v tabulce, doba běhu řídicího programu se s použitím protokolu ESP-NOW oproti protokolu LoRa

⁴<https://www.bto.pl/pdf/02023/icr18650-28A.pdf>

prodloužila. Spotřeba senzorů více závisela na použitých zařízeních. Z naměřených hodnot je také vidět výrazný rozdíl hodnot odebíraného proudu v režimu spánku senzorů založených na platformě LoRa a WeMos. Platforma WeMos D1 mini se dokonce v úsporném režimu dostala na úroveň mikroampérů, čímž značně prodloužila délku provozu zařízení na baterii. Hodnoty úsporného režimu byly měřeny dvěma způsoby, a to klasicky připojením zařízení na nepájivé pole, napájené přes mikro USB a druhým způsobem pomocí dvojího napájení přes mikro USB a baterii, po následném odpojení mikro USB. U senzorů založených na platformě WeMos nebylo třeba tento způsob testovat, protože už při prvním způsobu napájení nejevily známky aktivní činnosti v režimu spánku. Výsledný přehled naměřených hodnot je zobrazen v tabulce 5.2.

Tabulka 5.2: V tabulce jsou zaznamenány naměřené hodnoty při provozu topologie č. 2 založené na topologii ESP-NOW. První sloupec znázorňuje průměrnou dobu trvání aktivní části řídicího programu jednotlivých senzorů za jeden cyklus běhu, v druhém sloupci jsou naměřené průměrné hodnoty odběru proudu zařízení v provozu a třetí sloupec obsahuje hodnoty odebíraného proudu zařízením při režimu spánku. Ve čtvrtém sloupci je přehled naměřeného odběru proudu po využití způsobu dvojího napájení. Hodnoty u zařízení založených na platformě WeMos D1 mini nebyly měřeny, jelikož jejich konstrukce neprojevila podobné nedostatky.

platforma	doba trvání aktivního režimu (ms)	spotřeba v provozu (mA)	spotřeba v režimu spánku před (mA)	spotřeba v režimu spánku po (mA)
LoRa_rain	314	144	12,14	5
LoRa_DHT11	321	142,1	10	3,22
WeMos_DHT11	389	70.54	0.07	-
WeMos_light	206	80.2	0.20	-

Měření 3. topologie

Při měření 3. topologie byy získány hodnot senzorů založených na technologii BLE. U senzorů byly využity režimy spánku. U senzorů WeMos-Battery_DHT11 a Lolin32_Melexis se bohužel senzory nepodařilo úspěšně naměřit, jelikož multimetr při provozu způsobuje příliš velký pokles napětí a senzory se neustále restartovaly, místo aby pracovaly správně. Tento problém by se dal vyřešit použitím například dalšího regulovatelného zdroje napětí. Po prvotním měření byl zaveden odlišný způsob napájení v podobě připojení mikro USB a zároveň baterie k senzoru a následném odpojení mikro USB (viz podkapitola 5.1).

Tabulka 5.3: V tabulce jsou zaznamenány naměřené hodnoty při provozu topologie č. 3, založené na topologii BLE. První sloupec znázorňuje průměrnou dobu trvání aktivní části řídicího programu jednotlivých senzorů za jeden cyklus běhu, v druhém sloupci jsou naměřené průměrné hodnoty odběru proudu zařízení v provozu a třetí sloupec obsahuje hodnoty odebíraného proudu zařízením při režimu spánku. Ve čtvrtém sloupci se nacházejí hodnoty naměřené po využití způsobu napájení zařízení přes mikro USB a baterii a po následném odpojení mikro USB. Zařízení WeMos-Battery_DHT11 a Lolin32_Melexis nebylo možné naměřit z důvodu příliš vysokého poklesu napájecího napětí při připojení multimetru, které vedlo k neustálým restartům zařízení.

platforma	doba trvání aktivního režimu (ms)	spotřeba v provozu (mA)	spotřeba v režimu spánku před (mA)	spotřeba v režimu spánku po (mA)
LoRa_DHT11	198	111,8	10	2,2
LoRa_DHT22	738	104,1	10,4	3,4
WeMos-Battery_DHT11	-	-	-	-
Lolin32_Melexis	-	-	-	-

5.3 Výpočet teoretické výdrže zařízení na základě naměřených hodnot

Teoretická spotřeba zařízení byla vypočítána na základě následující rovnice. Vstupními veličinami jsou naměřené délky trvání jednotlivých fází běhu řídicího programu — délka provozu t_{act} a délka režimu spánku t_{sleep} v milisekundách. Druhým vstupním parametrem je naměřená spotřeba v aktivní fázi I_{act} a v úsporném režimu I_{sleep} v miliampérech. Na základě těchto hodnot je sestavena následující rovnice, která počítá průměrnou spotřebu I_{avg} proudu zařízení za 24 hodin. Proměnná x je počet měření za jeden den.

$$I_{avg} = \frac{I_{act} * t_{act} * x + I_{sleep} * t_{sleep}}{1000 * 60 * 60 * 24}, \quad (5.1)$$

kde t_{sleep} je vypočítán počtem milisekund za den a odečtením aktivní doby senzoru

$$t_{sleep} = 86400000 - x * t_{act}. \quad (5.2)$$

Teoretickou výdrž zařízení na bateriích lze poté vypočítat pomocí vztahu

$$t_{sen} = \frac{C_{bat}}{I_{avg}}, \quad (5.3)$$

kde C_{bat} značí nominální kapacitu referenční baterie ICR18650-28A a t_{sen} je výsledná teoretická doba provozu senzoru na baterii.

Na základě odvozených rovnic je možné vypočítat, že například senzor WeMos se snímačem DHT s průměrnou spotřebou $67 \mu A$ v režimu spánku, je schopen vydržet na bateriích při měření teploty každé 2,4 h až 4,6 let. Podobně jsou tomu další senzory založené na platformě WeMos, u kterých se liší spotřeba kvůli použití jiných snímačů nebo modulů pro snímání veličin. Spotřeba může být až dvojnásobná, pokud není možnost snímač nebo

modul odepnout v režimu spánku a pokud se sám nevypíná. U DHT11 tento problém nastal, ale například u senzoru pro měření světelné intenzity s modulem LDR, se spotřeba v režimu spánku pohybovala kolem 204 μ A a výsledná výdrž na bateriích byla při stejném počtu měření odhadem pouze 1,5 let.

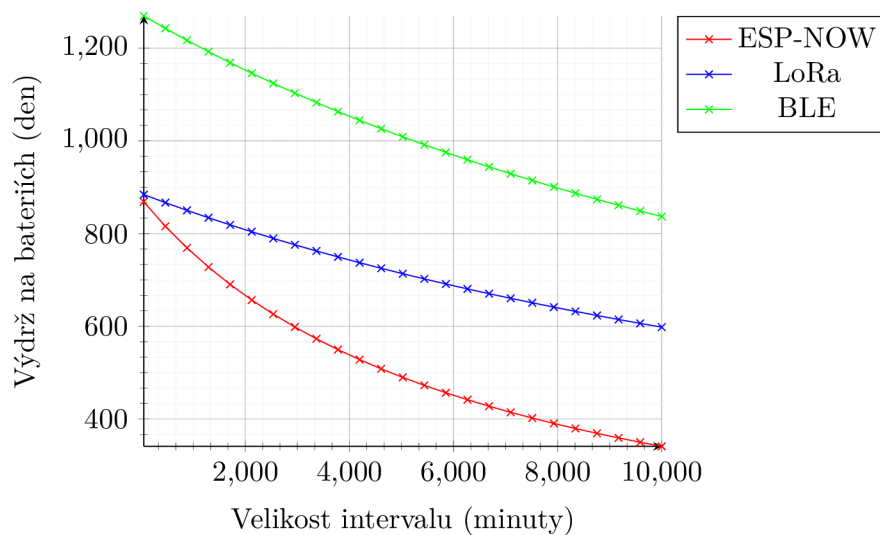
Dále například při měření senzoru LoRa_rain se snímačem intenzity srážek MH-DR byla odhadovaná výdrž zařízení několikanásobně menší. Jelikož spotřeba zařízení v režimu spánku se pohybuje kolem 12 mA, tak by při stanovených pouze 2 měřeních za den senzor vydržel v provozu pouze 9 dní. Na základě těchto měření jsem se pokusila najít další možnosti, které jsou zmíněny v podkapitole 5.1.

Vliv počtu měření na spotřebu

Při pozorování vlivu počtu měření na výdrž baterie, se spotřeba začala značně snižovat při zvýšení intenzity měření. Například při měření každé 2,4 h hodiny, zařízení vydrží na baterii 4,6 let, ale při měření každým 24 minut se doba provozu zařízení zkracuje na 3,3 let, tudíž o necelých 30 %. Při počtu měření je také třeba brát v úvahu zvolenou technologii, přes kterou zařízení komunikuje. Vliv technologií na počet měření je shrnut na grafu 5.2.

Porovnání způsobů komunikace

Z naměřených hodnot vyplývá, že celková spotřeba značně závisí na použitém zařízení a jeho způsobu konfigurace. Pro možnost porovnání jednotlivých komunikačních protokolů jsem použila stejný senzor LoRa_DHT11 ve všech sítích se stejnou konfigurací. Senzor jsem měřila v základním nastavení, s využitím režimu spánku, ze kterého je přiváděn k činnosti periodicky po určitém časovém intervalu.



Obrázek 5.2: Na grafu je zachycený vliv použití různých technologií pro zprostředkování komunikace u stejného fyzického zařízení — platformy TTGO LoRa se snímačem DHT11. Počet měření za den začíná na hodnotě 50 (měření každých 29 minut) až po 10000 měření za den (což znamená měření každých 10 vteřin).

Z grafu 5.2 lze vidět, že při použití stejných hardwarových komponent a konfigurace zařízení, je nejušpornější technologií BLE, při počátečních 50 měření disponuje teoretickou

výdrž zařízení přibližně 1300 hodin, zatímco u technologií LoRa a ESP-NOW je teoretická výdrž pouze přibližných 900 hodin. Dále lze na grafu vyčíst, že i když mají při nízkém počtu měření technologie LoRa a ESP-NOW téměř stejnou výdrž, při větším počtu měření lze vidět, že zařízení komunikující technologií ESP-NOW vydrží na baterii mnohem kratší dobu.

Vliv optimalizací na výslednou výdrž baterie

Jednotlivé optimalizace byly zavedeny do provozu senzorů a následně testovány na různých typech zařízení, topologiích a protokolech. Výsledné hodnoty byly zavedeny do srovnávací tabulky **B** v příloze práce.

Režim spánku

Režim spánku má výrazný vliv na celkovou výdrž zařízení při provozu na baterii. Při použití pouhého režimu nečinnosti by nebylo zařízení možné provozovat na bateriích. Zařízení se většinu provozního času nacházejí v režimu spánku, je tedy důležité tuto fázi držet na hranici co nejnižší spotřeby. Časové intervaly aktivních fází senzorů se pohybují v řádu milisekund (viz tabulky naměřených hodnot).

Důležitým aspektem je také stanovený počet měření. V rozmezí kolem 2 až 10 měření (čili měření například každé 2 hodiny) denně ještě není výdrž baterie znatelně ovlivněna. Ale pokud měříme hodnoty například každých 20 minut, může výdrž zařízení na baterii klesnout klidně o 30 % (viz podkapitola **Spotřeba zařízení**). Je tudíž nutné si předem stanovit požadovaný interval měření.

Připojení USB a baterie

Při řešení tohoto problému jsem použila zmíněný způsob, kdy jsem nejprve zapojila senzor k napájení pomocí mikro USB a zároveň přes konektor připojila baterie, a micro konektor jsem následně vyjmula. Odběr proudu senzorem okamžitě několikanásobně klesl. Pro ilustraci jsem naměřila hodnoty senzorů před a po odpojení mikro USB. Z tabulky **5.2** můžeme vidět, že například při použití senzoru LoRa_rain se snímačem intenzity srážek se mi podařilo snížit spotřebu z původních 12,14 mA o více než polovinu na 5 mA. Výsledný přehled všech naměřených hodnot této optimalizace je uveden v přehledu **B**.

Odesílání dat na základě rozdílu hodnot

Následující optimalizaci jsem implementovala na základě uložení předchozí naměřené hodnoty do `RTC_DATA_ATTR` a testováním této hodnoty před inicializací komunikace. Tuto fázi bylo následně třeba odměřit a výsledky měření pro různé platformy jsou uvedeny v následující tabulce. Hodnoty jsou ve výsledku odesílány až po překročení předem stanovené hranice, tudíž nyní poměrnou část (v závislosti na stanoveném rozdílu hodnot po kterém se hodnoty mají zasílat) tvořila pouze inicializační fáze s měřením, která se projevila mnohem nižší spotřebou (viz tabulka).

Dalším testovaným jevem bylo odesílání dat až při detekované změně naměřených hodnot. Program byl sestaven tak, aby při každém běhu nejprve naměřil hodnotu snímače a porovnal ji s předchozí naměřenou hodnotou. Pokud se tato hodnota lišila o poměrnou vytyčenou část, jediné tehdy by senzor mohl inicializovat komunikaci přes aktuální síť

a poslat novou hodnotu přijímači. Tento případ byl testován u několika zařízení a výsledky měření byly zaneseny do tabulky C.

Odpojení LED diody

K aplikaci této optimalizace jsem odstranila napájecí LED diodu několika platformem a snímačů pro otestování rozdílů spotřeby. LED diody se vyskytovaly na všech vývojových platformách a na několika modulech snímačů, indikující napájení, a to i přes režim spánku. Po odpájení LED diod jsem znovu naměřila spotřebu, která se vlivem odstranění viditelně snížila. Výsledné hodnoty byly zaznamenatelné pouze v úsporném režimu, kdy je hodnota víceméně konstantní. Hodnoty po odstranění diody jsou zaznamenány v tabulce v příloze B. Například zde můžeme vidět, že při odstranění napájecí diody LED ze senzoru LoRa_DHT11 (komunikující protokolem LoRa) se spotřeba snížila z 1,4 mA na 1,3 mA.

Odpojení modulů platformy

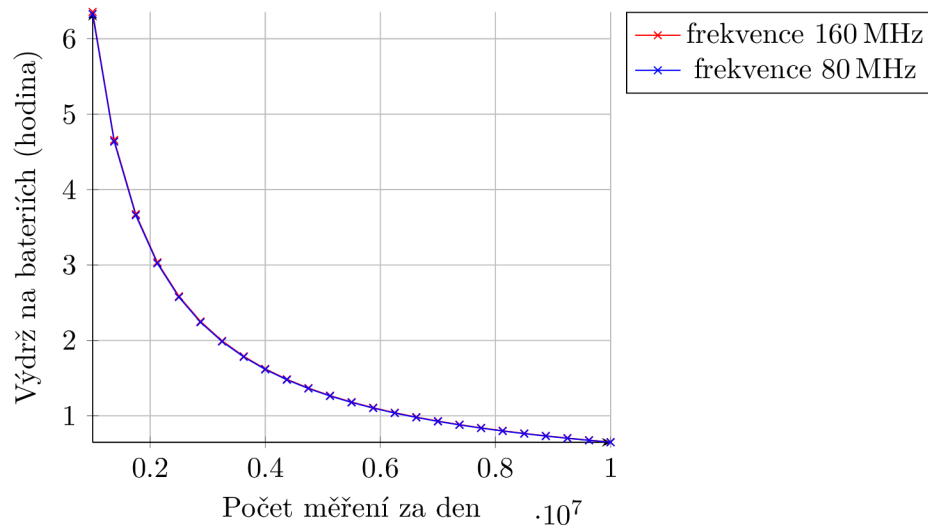
Při odpojení modulů popsaných v podkapitole 5.1 jsem dospěla k následujícím výsledkům. Vysoký vliv aplikace měl příkaz k odstavení činnosti RF modulu `rf95.sleep()`, který byl použit u platformem TTGO LoRa. Při jeho aplikaci klesl odběr proudu zařízení v režimu spánku z původních 3,16 mA na 1,4 mA, tedy téměř na polovinu. Výsledkem se tedy potvrdilo, že při vyvolání úsporného režimu se RF modul sám úplně nedeaktivuje.

Další zmíněné testované funkce bohužel neměly z důvodu výkyvů hodnot při měření natolik signifikantní změny hodnot, aby byly stále zaznamenatelné. Je možné, že tyto funkce pomáhají snížit spotřebu a ujišťují o bezpečném ukončení činnosti modulů, ale to dle mého názoru záleží na konkrétní aplikaci a skladbě řídicího programu.

Snížení frekvence MCU

Snížení frekvence bylo otestováno na dvou mikročipech ESP32. Prvním testovaným čipem byl mikročip ESP32 vestavěný do vývojové platformy LoRa. K porovnání jsem použila hotový otestovaný senzor LoRa_rain, který byl v základním nastavení, využívající pouze deep sleep režim. Při provozu zařízení na základní frekvenci 160 MHz byl v aktivní fázi zařízení naměřen v průměru proud 144 mA a program se vykonával průměrně 314 ms. Z toho vyplývá výdrž baterie, při spotřebě v režimu spánku 12,14 mA a při měření každých 10 s, přibližně 7,18 dne. Po snížení frekvence na 80 MHz, se průměrná spotřeba při provozu snížila na 112,4 mA, ale zároveň se zvýšila doba provozu na průměrných 460,2 ms. Tím pádem se při stejné spotřebě v režimu spánku (12,14 mA) a stejném počtu měření výdrž zařízení na vzorové baterii s kapacitou 2800 mAh snížila na 6,97 dne. Z naměřených hodnot je tedy zřejmé, že spotřeba se sice mírně snížila, ale více se zvýšila doba konání programu a tím se výsledná spotřeba, a především doba vykonávání programu naopak zvýšila. Není tedy výhodné při provozu těchto senzorů snižovat frekvenci mikročipu.

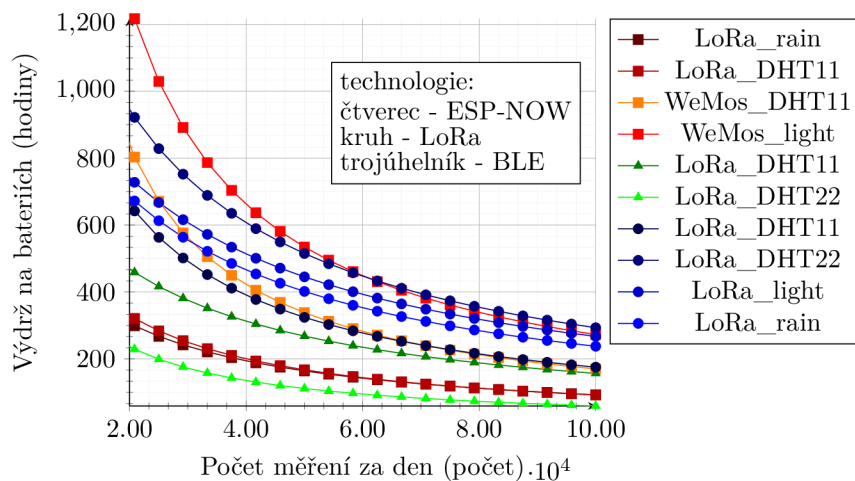
Na obrázku je sestaven graf závislosti počtu měření za jeden den na výdrži baterie. V grafu jsou zaneseny závislosti při různých frekvencích mikročipu. Jak lze vidět s grafu, výsledné křivky se překrývají, z toho vyplývá že při nižší frekvenci bude výdrž zařízení na baterii téměř totožná. Nemá tedy smysl při optimalizaci implementovaných senzorů zavádět snižování frekvence.



Obrázek 5.3: Graf zobrazující rozdílnou výdrž na baterii zařízení LoRa_rain při rozdílných frekvencích mikroprocesoru. Na grafu lze vidět, že výsledné křivky teoretické výdrže se překrývají.

Výsledný přehled úsporných zařízení

Výsledkem aplikace optimalizací na implementované senzory vznikly úsporná zařízení, která se liší způsobem využití, použitými technologiemi a tím také dobou výdrže. Nejúspornější z testovaných zařízení se jeví senzory založené na platformě WeMos D1 mini, u kterých jako jediných se podařilo získat v režimu spánku spotřebu na úrovni mikroampérů. Při vhodné volbě intervalů měření (viz graf 5.4) je zmíněné zařízení možné provozovat na baterii v řádu let. Na druhou stranu ostatní zařízení, založená na čípech ESP32, mohou po určité míře optimalizací pracovat uspokojivě množstvím času na bateriích, i když rozmezí se pohybuje v řádu několika měsíců.



Obrázek 5.4: Na grafu jsou zobrazeny křivky průměrných délek provozu jednotlivých senzorů při napájení na baterii. U zařízení byly použity všechny dostupné způsoby optimalizací, jejichž naměřené hodnoty se nacházejí v přílohách A a B.

U jednotlivých zařízení v grafu 5.4 byl zaveden způsob odesílání dat pouze při detekci výrazné změny veličiny. Potřeba zasílání dat na základě detekce změny byla stanovena na teoretických 50 % případů. Spotřeba při používání způsobu zasílání pouze při změně hodnot je vypsána v příloze C. Vybrané hodnoty výdrže zařízení na bateriích jsou vypsány v tabulce 5.4.

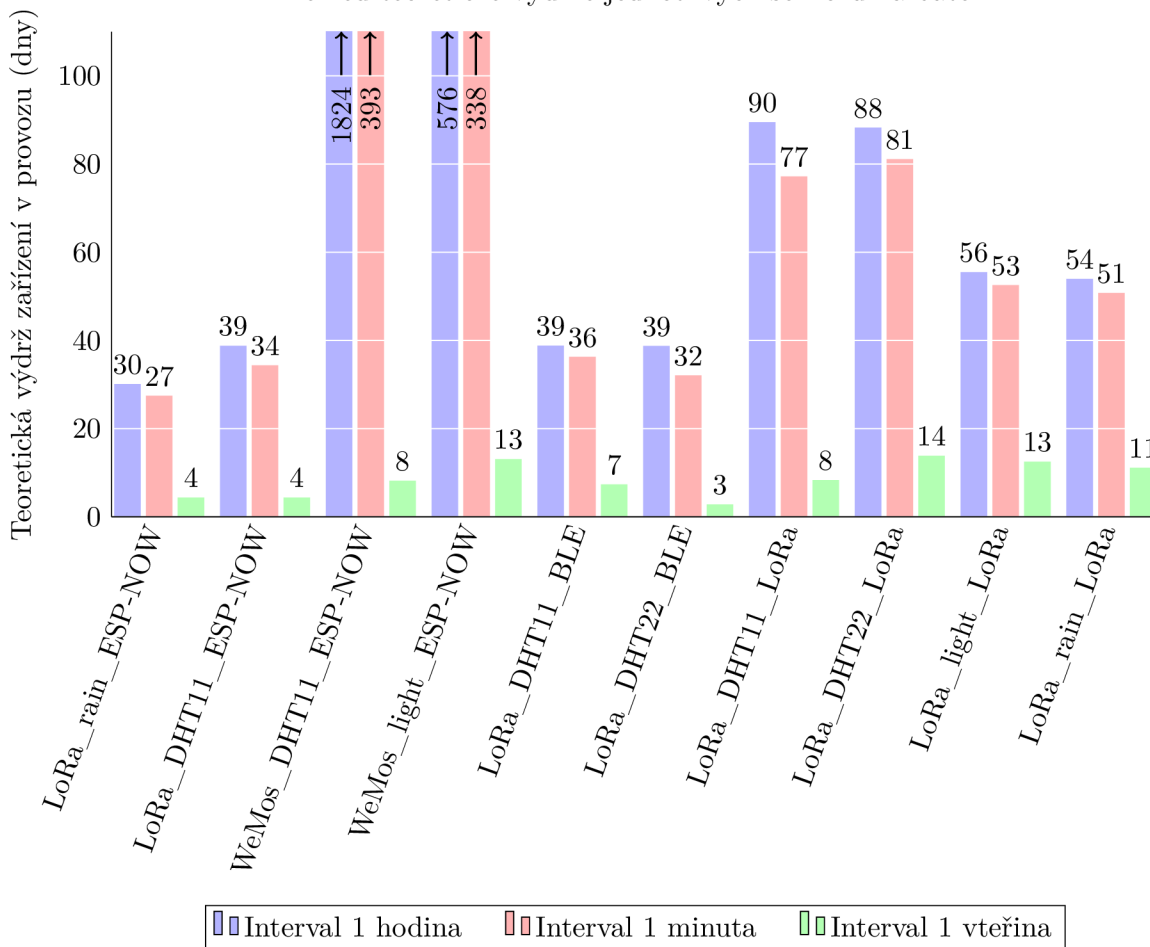
Z grafu 5.4 lze vidět, že u všech měřených zařízení se vyskytuje podobná trajektorie křivky značící snižující se výdrž zařízení při rostoucím počtu měření. Zároveň lze říci, že při vysokém počtu měření denně (v řádu několika tisíců), klesá výdrž všech zařízení přibližně stejným tempem a použité technologie a platformy už nemají na toto tempo velký vliv. Je tedy třeba volit vhodný interval měření, pro zachování úspornosti zařízení.

platforma	technologie	interval 1 hodina (dny)	interval 1 minuta (dny)	interval 1 vteřina (dny)
LoRa_rain	ESP-NOW	30,10	27,45	4,38
LoRa_DHT11	ESP-NOW	38,80	34,36	4,37
WeMos_DHT11	ESP-NOW	1824,54	393,36	8,18
WeMos_light	ESP-NOW	576,37	338,14	13,11
LoRa_DHT11	BLE	38,84	36,30	7,36
LoRa_DHT22	BLE	38,75	32,08	2,83
LoRa_DHT11	LoRa	89,50	77,20	8,35
LoRa_DHT22	LoRa	88,25	81,10	13,84
LoRa_light	LoRa	55,50	52,54	12,50
LoRa_rain	LoRa	53,95	50,76	11,14

Tabulka 5.4: V tabulce jsou vepsány vypočítané hodnoty teoretické výdrže zařízení, vypočítané pro tři různě velké intervaly měření během dne. V prvním sloupci je vypočítána výdrž při měření každých . U měření spotřeby byla brána v úvahu teoretická změna způsobující potřebu odesílání hodnot v 50 % měření.

V tabulce 5.4 lze vidět několik vybraných hodnot naměřených při teoretických intervalech měření. Hodnoty byly měřeny při malém počtu měření za den (tzn. každou hodinu), průměrném měření každou minutu a intenzivním měření každou vteřinu. Hodnoty byly poté vykresleny v grafu 5.5. Na základě hodnot lze vyčíst výrazné rozdíly při malém počtu měření (například porovnání platformy WeMos s ostatními). Na druhou stranu lze vidět že senzory založené na čípech ESP32 mají průměrnou výdrž do 100 dní při měření jednou za hodinu.

Přehled teoretické výdrže jednotlivých senzorů na baterii



Obrázek 5.5: Výsledný graf přehledu naměřených hodnot senzorů při provozu na baterii. Senzory byly rozděleny vždy na tři části podle tří měřených velikostí intervalů — 1 hodina, minuta a vteřina.

Z grafu 5.5 lze vyčíst rozdílné poměry měřených intervalů v závislosti na výdrži zařízení při provozu na baterii. Nejvíce úspornými se staly zařízení založená na platformě WeMos s obsahující čip ESP8266. Lze také vidět, že při volbě intervalu měření každou hodinu nebo minutu nemá výrazný vliv na výdrž baterie. Tento fakt je způsoben velmi krátkými aktivními dobami zařízení (v řádu milisekund). Na druhou stranu lze vyzorovat obrovský propad při snaze o intenzivní měření každou vteřinu a to proto, že velikost intervalu měření se již téměř vyrovnává velikosti intervalu režimu spánku.

Co se komunikačních technologií týče, záleží na vyžadovaných parametrech. Pokud je hlavním cílem dlouhá výdrž zařízení, je nejvhodnější variantou technologie BLE. Ta ovšem nemusí splňovat potřebné podmínky na dosah komunikaci zařízení. Na nejdelší vzdálenost dokáže dle specifikací technologií komunikovat senzory přes protokol LoRa, na který je ale třeba mít vestavěný modul, který není součástí základním ESP čipů. ESP-NOW je sice nejméně úspornou technologií, je ale vhodným obecným řešením například pro senzory s čipem ESP8266, které obsahují pouze Wi-Fi čip. Poté umožňuje úspornější komunikaci než přes klasický Wi-Fi protokol, čímž pomáhá spořit energii.

Kapitola 6

Závěr

Cílem práce bylo vytvoření úsporných senzorů napájených na baterii a úsporných sítí pro domácí použití. Úvodem byly nalezeny existující řešení k sestavení chytrých IoT senzorů. Byly nalezeny platformy vhodné pro využití v úsporných senzorových sítích založené na čipech ESP8266 a ESP32. Dále byl vytvořen přehled zařízení učených ke snímání veličin a byly představeny protokoly a technologie umožňující jejich komunikaci. Důraz byl kladen na nalezení způsobů úspory energie při volbě správných komponent, uzpůsobení provozu a komunikace senzorů. V práci jsou také představeny používané způsoby rozložení uzlů při sestavování IoT sítí.

Na základě získaných informací byly senzory navrženy společně s návrhy jejich zapojení do úsporných sítí. Při tvorbě sítí byl kladen důraz na úsporné rozložení do topologie typu hvězda. Dále byly senzory implementovány a byla zahájena jejich komunikace. Po zprovoznění tří navržených topologií byla odměřena jejich spotřeba. Poté byly do provozu senzorů začleněny navržené způsoby optimalizace a následně byl jejich vliv ověřen pomocí průběžného měření.

Ze získaných hodnot byly vypočítány teoretické doby provozu jednotlivých zařízení na baterii. Z výsledků odhadů bylo možné sestavit přehled rozdílů výdrže jednotlivých zařízení. Dále byl vytvořen přehled vlivu komunikačních protokolů a počtu měření na výdrž zařízení při provozu na baterii. Výsledkem jsou úsporná zařízení schopná provozu na bateriích, z čehož nejúspornější naměřenou platformou pro tvorbu senzorů se stala platforma WeMos D1 mini, založená na mikročipu ESP8266, schopná provozu v řádu let. Na druhou stranu zařízení založená na platformách LoRa s čipem ESP32 se projevila jako méně vhodná pro při stejném způsobu napájení. Jejich teoretická výdrž se při stejné aktivitě pohybovala kolem jednoho až tří měsíců. Co se komunikační technologie týče, nejvíce úspornou se projevila technologie BLE. Nejméně výhodnou technologií pro komunikaci úsporných senzorů se stala technologie ESP-NOW.

Práce byla z důvodu jejího možného využití dalšími uživateli zveřejněna na webovém repozitáři Github¹. Během vývoje jsem také modifikovala konfigurace sítě lora mesh² od nootropicdesign³, která umožnila provoz této sítě čipům ESP.

¹<https://github.com/4Gabby4/ESP-based-efficient-networks>

²<https://github.com/nootropicdesign/lora-mesh/pull/1>

³<https://github.com/nootropicdesign/lora-mesh>

Literatura

- [1] Adafruit: *Introducing the Raspberry Pi 2 - Model B*.
URL <https://cdn-learn.adafruit.com/downloads/pdf/introducing-the-raspberry-pi-2-model-b.pdf>
- [2] Akyildiz, I.; Su, W.; Sankarasubramaniam, Y.; aj.: Wireless sensor networks: a survey. *Computer Networks*, ročník 38, č. 4, 2002: s. 393 – 422, ISSN 1389-1286, doi:[https://doi.org/10.1016/S1389-1286\(01\)00302-4](https://doi.org/10.1016/S1389-1286(01)00302-4).
URL <http://www.sciencedirect.com/science/article/pii/S1389128601003024>
- [3] Aosong Electronics Co.,Ltd: *Digital-output relative humidity & temperature sensor/module*.
URL <https://www.sparkfun.com/datasheets/Sensors/Temperature/DHT22.pdf>
- [4] Arduino návody: *Teploměr a vlhkoměr DHT11 a DHT22*.
URL <https://navody.arduino-shop.cz/navody-k-produktum/teplotni-senzor-dht11.html>
- [5] Arduino.cc: *What is Arduino?* [Online; cit. 10.12.2018].
URL <https://www.arduino.cc/en/Guide/Introduction>
- [6] Banggood: *LILYGO® TTGO LORA32*.
URL https://www.banggood.com/LILYGO-TTGO-LORA32-868Mhz-SX1276-ESP32-0led-Display-bluetooth-WIFI-Lora-Development-Module-Board-p-1248652.html?rmmds=search&cur_warehouse=CN
- [7] Banggood: *Wemos TTGO WiFi + Bluetooth Battery*.
URL https://www.banggood.com/Wemos-TTGO-Wi-Fi-Bluetooth-Battery-ESP32-0_96-Inch-OLED-Development-Tool-p-1213497.html?rmmds=search&cur_warehouse=CN
- [8] Banggood: *WeMos WiFi + Bluetooth Battery*.
URL https://www.banggood.com/WeMos-Wi-Fi-Bluetooth-Battery-ESP32-Development-Tool-p-1164436.html?cur_warehouse=CN
- [9] Banggood: *Wemos® Battery Shield V1.2.0*.
URL https://www.banggood.com/Wemos-Battery-Shield-V1_2_0-For-Wemos-D1-Mini-Single-Lithium-Battery-Charging-Boost-p-1264243.html?cur_warehouse=CN
- [10] Blum, J.: *Exploring Arduino: Tools and Techniques for Engineering Wizardry*. USA: John Wiley & Sons, Inc., 2013, ISBN 978-1-118-54936-0.

- [11] e-Gizmo Mechatronix Central: *Rain Drop Sensor*.
URL <https://www.e-gizmo.net/oc/kits%20documents/Rain%20drops%20sensor/Rain%20Drop%20sensor.pdf>
- [12] Espressif Inc: *ESP-NOW*. [Online; cit. 14.12.2018].
URL https://docs.espressif.com/projects/esp-idf/en/latest/api-reference/network/esp_now.html
- [13] Espressif Inc.: *Sleep Modes*. [Online; cit. 12.12.2018].
URL https://docs.espressif.com/projects/esp-idf/en/latest/api-reference/system/sleep_modes.html
- [14] Espressif Inc.: *ESP8266, Low Power Solutions*. 2016, [Online; cit. 14.12.2018].
URL https://www.espressif.com/sites/default/files/9b-esp8266-low_power_solutions_en_0.pdf
- [15] Fabbri, L.: *HiGrow plants monitoring sensor*.
URL <https://hackaday.io/project/25253-higrow-plants-monitoring-sensor>
- [16] Farahani, S.: *ZigBee Wireless Networks and Transceivers*. Newton, MA, USA: Newnes, 2008, ISBN 0750683937, 9780750683937.
- [17] Farnell: *Arduino Uno*.
URL <https://www.farnell.com/datasheets/1682209.pdf>
- [18] Foxworth, T.: *Making the ESP8266 low-powered with deep sleep*. LOSANT, Březen 2017, [Online; cit. 11.12.2018].
URL <https://www.losant.com/blog/making-the-esp8266-low-powered-with-deep-sleep>
- [19] Ghosh, A.; Das, S. K.: Coverage and connectivity issues in wireless sensor networks: A survey. *Pervasive and Mobile Computing*, ročník 4, č. 3, 2008: s. 303 – 334, ISSN 1574-1192, doi:<https://doi.org/10.1016/j.pmcj.2008.02.001>.
URL <http://www.sciencedirect.com/science/article/pii/S1574119208000187>
- [20] Gill, K.; Yang, S.-H.; Yao, F.; aj.: A ZigBee-Based Home Automation System. *Consumer Electronics, IEEE Transactions on*, ročník 55, 06 2009: s. 422 – 430, doi:[10.1109/TCE.2009.5174403](https://doi.org/10.1109/TCE.2009.5174403).
- [21] Gomez, C.; Paradells, J.: Wireless home automation networks: A survey of architectures and technologies. *IEEE Communications Magazine*, ročník 48, č. 6, June 2010: s. 92–101, ISSN 0163-6804, doi:[10.1109/MCOM.2010.5473869](https://doi.org/10.1109/MCOM.2010.5473869).
- [22] Gubbi, J.; Buyya, R.; Marusic, S.; aj.: Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, ročník 29, č. 7, 2013: s. 1645 – 1660, ISSN 0167-739X, doi:<https://doi.org/10.1016/j.future.2013.01.010>.
URL <http://www.sciencedirect.com/science/article/pii/S0167739X13000241>
- [23] Hladík, Z.: *LORA*. 2016.
URL https://brmlab.cz/_media/event/lora_drnsnik.pdf

- [24] LinkLabs: *6LoWPAN Range: Use Case Calculations*. 2014.
URL <https://www.link-labs.com/blog/6lowpan-range>
- [25] LinkLabs: *Comparing Mesh, Star & Point-To-Point Topology In IoT Networking*. 2018.
URL <https://www.link-labs.com/blog/iot-topology>
- [26] Maxim Integrated Products, Inc.: *DS18B20 Programmable Resolution – 1-Wire Digital Thermometer*. 2018.
URL <https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf>
- [27] Melexis: *MLX90614 family*. 2009.
URL <https://www.sparkfun.com/datasheets/Sensors/Temperature/SEN-09570-datasheet-3901090614M005.pdf>
- [28] Mouser Electronics: *DHT11 Humidity & Temperature Sensor*.
URL <https://www.mouser.com/ds/2/758/DHT11-Technical-Data-Sheet-Translated-Version-1143054.pdf>
- [29] neznámý: *BLE Overview*. Laird Technologies Wireless Connectivity Blog, Únor 2013, [Online; cit. 7.11.2018].
URL <http://www.summitdata.com/blog/ble-overview/>
- [30] Njoroge, K. P.: *MICROCONTROLLER-BASED IRRIGATION SYSTEM*.
URL <http://eie.uonbi.ac.ke/sites/default/files/cae/engineering/eie/MICROCONTROLLER-BASED%20IRRIGATION%20SYSTEM.pdf>
- [31] Pacelle, M.: *3 topologies driving IoT networking standards*. 2014.
URL <http://radar.oreilly.com/2014/04/3-topologies-driving-iot-networking-standards.html>
- [32] Palivec, P.: *Bluetooth Low Energy*. 2012.
URL <https://www.dps-az.cz/soucastky/id:9912/bluetooth-low-energy>
- [33] Prakash, S. P. S.; Nagabhushan, T. N.; Krinkin, K.: Energy aware power save mode management in wireless mesh networks. In *14th Conference of Open Innovation Association FRUCT*, Nov 2013, ISSN 2305-7254, s. 122–131, doi:10.1109/FRUCT.2013.6737954.
- [34] RS Components: *Light dependent resistors*.
URL https://components101.com/sites/default/files/component_datasheet/LDR%20Datasheet.pdf
- [35] Saunders, M.: *Power Management Techniques for Low-Energy IoT Devices*.
URL <https://www.avnet.com/wps/portal/us/resources/technical-articles/article/iot/power-management-techniques-for-%20low-energy-iot-devices/>
- [36] Vojáček, A.: *IoT MQTT prakticky v automatizaci – 1.díl – úvod*. automatizace.HW, Leden 2017, [Online; cit. 8.12.2018].
URL <https://automatizace.hw.cz/iot-mqtt-prakticky-v-automatizaci-1dil-uvod.html>

Přílohy

Příloha A

Tabulka výsledných naměřených hodnot senzorů

platforma	technologie	doba trvání aktivního režimu (ms)	spotřeba v provozu (mA)	spotřeba v režimu spánku (mA)
LoRa_rain	ESP-NOW	314	144	3,87
LoRa_DHT11	ESP-NOW	321	142,1	3,0
WeMos_DHT11	ESP-NOW	389	70,54	0,06
WeMos_light	ESP-NOW	206	80,2	0,20
LoRa_DHT11	BLE	198	111,8	3,0
LoRa_DHT22	BLE	738	104,9	3,0
LoRa_DHT11	LoRa	198	111,8	1,3
LoRa_DHT22	LoRa	197	60	1,32
LoRa_light	LoRa	197	57	2,1
LoRa_rain	LoRa	209	71,2	2,16

Tabulka A.1: Tabulka obsahuje výsledné naměřené optimalizované hodnoty zařízení.

Příloha B

Tabulka aplikace optimalizací zařízení

platforma	technologie	spotřeba v režimu spánku původní (mA)	připojení USB a baterie (mA)	odpojení modulů (mA)	odpojení LED (mA)
LoRa_rain	ESP-NOW	12,14	5	-	3,87
LoRa_DHT11	ESP-NOW	11	3,22	-	3,0
WeMos_DHT11	ESP-NOW	0,07	-	-	0,06
WeMos_light	ESP-NOW	0,2	-	-	0,2
LoRa_DHT11	BLE	10	2,2	-	3,0
LoRa_DHT22	BLE	10,4	3,4	-	3,0
LoRa_DHT11	LoRa	11	3,16	1,4	1,3
LoRa_DHT22	LoRa	10	2,24	1,64	1,32
LoRa_light	LoRa	9	2,4	2,39	2,1
LoRa_rain	LoRa	12	5	3,42	2,16

Tabulka B.1: Tabulka naměřených hodnot při použití zvolených optimalizací.

Příloha C

Tabulka hodnot naměřených při podmíněném zasílání zpráv

platforma	technologie	délka trvání s odesláním (ms)	délka trvání bez odeslání (ms)	odběr proudu s odesláním (mA)	odběr proudu bez odeslání (mA)
LoRa_rain	ESP-NOW	314	35	144	49
LoRa_DHT11	ESP-NOW	321	77	142,1	39
WeMos_DHT11	ESP-NOW	389	89	70,54	11
WeMos_light	ESP-NOW	206	67	80,2	14
LoRa_DHT11	BLE	198	76	111,8	58
LoRa_DHT22	BLE	738	45	104,1	43
LoRa_DHT11	LoRa	198	66	111,8	54
LoRa_DHT22	LoRa	197	47	60	58
LoRa_light	LoRa	197	78	57	49
LoRa_rain	LoRa	209	40	71,2	57

Tabulka C.1: Tabulka obsahuje hodnoty délek časových intervalů aktivní doby zařízení a velikosti odebíraných proudů při běhu programu, který využívá zasílání dat pouze při detekovaném stanoveném rozdílu hodnot. Hodnoty označené jako „s odesláním“ jsou naměřené hodnoty v případě detekce rozdílu a následného odeslání zprávy. Hodnoty v ostatních sloupcích označují výsledky naměřené bez konečného odeslání zprávy.

Příloha D

Obsah přiloženého CD

```
/.....kořenový adresář přiloženého CD
├── text
│   ├── xchmel27-Usporne-senzory-IoT.zip.....zdrojové soubory textu práce
│   └── xchmel27-Usporne-senzory-IoT.pdf.....text práce ve formátu pdf
├── tabulka.....složka s tabulkou k výpočtu spotřeby
│   └── tabulka-spotreba.xlsx.....tabulka k výpočtu spotřeby zařízení
├── src.....zdrojové soubory řídicích programů
│   ├── ESP-NOW.....složka zdrojových souborů topologie založené na ESP-NOW
│   │   ├── ESP32-Devkit - ESPNOW - gateway..složka se zdrojovými soubory senzoru
│   │   │   ├── *.cpp.....zdrojový kód řídicího programu senzoru
│   │   │   └── *.ini.....konfigurační soubor rozhraní PlatformIO
│   │   ├── LoRa - DHT11 - ESPNOW
│   │   ├── LoRa - raindrop - ESPNOW
│   │   ├── WeMos - DHT11 - ESPNOW
│   │   └── WeMos - LightSen - ESPNOW
│   ├── BLE.....složka zdrojových souborů topologie založené na BLE
│   │   ├── Devkit - BLE - gateway
│   │   ├── Lolin32 - Melexis - BLE
│   │   ├── LoRa - DHT11 - BLE
│   │   ├── LoRa - DHT22 - BLE
│   │   └── WeMos-Battery - DHT11 - BLE
│   └── LoRa.....složka zdrojových souborů topologie založené na LoRa
│       ├── LoRa-reciever-gateway
│       ├── LoRa-sender-raindrop
│       ├── LoRa-sender-DHT11
│       ├── LoRa-sender-DHT22
│       └── LoRa-sender-light
```

Slovník pojmů

gateway, přístupový bod Označení síťového uzlu, který zajišťuje propojení mezi dalšími sítěmi, založenými na různých protokolech. [4](#), [26](#), [27](#)

GPIO — General Purpose Input/Output GPIO je typ pinu, který se vyskytuje na integrovaných obvodech. Na rozdíl od specifikovaných signalizačních a datových pinů, nemá předem určený význam. Ten je mu přiřkládán až samotnou aplikací. [6](#)

hop Cesta uskutečněná mezi dvěma fyzickými zařízeními. [4](#)

I2C Komunikační sběrnice, která dělí komunikující zařízení na dva typy - *master* (řídící prvek) a *slave* (prvky ovládané uzlem *master*). [18](#), [35](#)

IoT modul Malá hardwarová součástka obsahující určitou komponentu, která připojenému zařízení zprostředkuje danou funkcionalitu. [5](#)

klon Napodobenina originálního výrobku, za použití oficiálně přístupných zdrojových kódů a popisů komponent zařízení. [7](#)

MAC adresa Číslo sloužící jako jednoznačný identifikátor zařízení v síti. [39](#)

MCU — Mikrokontrolér Jednočipový počítač, který je tvořen jednoduchým integrovaným obvodem. [5](#)

open-source Zdrojové kódy a popis hardware jsou volně přístupné a využitelné kýmkoliv. [7](#)

PAN — Personal Area Network Typ bezdrátové sítě, často označení zařízení v okolí jedné osoby. [22](#)

pin Vývod ze zařízení, kterým jsou k němu připojovány vodiče. [6](#)

protokolový zásobník Sestava protokolů, která je použita v komunikační síti. Tvoří předepsanou hierarchii softwarových vrstev. [4](#)

pull-up rezistor Speciální rezistor, který udržuje logickou hodnotu na vodiči nesoucím informaci v době, kdy není definována žádným jiným zařízením. [17](#)

PWM — Pulse Width Modulation Pulsně šířková modulace slouží k popisu přenosu digitálního signálu. [6](#), [10](#)

- RF vysílač** — **radio frekvenční vysílač** Vysílač fungující na principu komunikace přes radiové vlny. 5, 6
- RTC** — **Real Time Clock** Hodiny, které v zařízení udržují informaci o reálném čase. 6
- SDK** Soubor nástrojů umožňujících vývoj softwaru pro dané zařízení. 7
- senzor** Zařízení schopné detekovat a měřit změny fyzikálních vlastností prostředí a odesílat je sítí. 4
- shield** Zařízení rozšiřující základní vývojovou platformu o další funkcionalitu. Na rozdíl od modulu většinou zaujímá všechny piny desky, na kterou je připojen. 5, 7
- topologie** Vzor propojení jednotlivých síťových prvků mezi sebou. 4
- WSN** — **Wireless Sensor Network** Typ bezdrátové sítě, založené na zařízeních, které snímají fyzikální vlastnosti (jako je například teplota či vlhkost) zvoleného prostředí a zasílají hodnoty do centrálního bodu. 4, 34