

Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra informačního inženýrství



**Česká
zemědělská
univerzita
v Praze**

Diplomová práce

Monitorovací systém včelího úlu za pomoci IoT

Bc. Martin Kozej

© 2024 ČZU v Praze

ZADÁNÍ DIPLOMOVÉ PRÁCE

Bc. Martin Kozej

Informatika

Název práce

Monitorovací systém včelího úlu za pomoci IoT

Název anglicky

IoT bee hive monitoring system

Cíle práce

Cílem práce je návrh a následné vytvoření monitorovacího systému pro včelí úl. Úl bude posílat naměřené hodnoty do databáze, z které budou následně data vizualizována pomocí grafů.

Metodika

Práce se bude skládat z teoretické a praktické části.

Teoretická část se bude zaměřena na studium problematiky včelaření, potřeb včelstva a integrace IoT do úlů. Na základě získaných informací budou vybrány adekvátní čidla pro monitorovací systém. Poznatky z teoretické části budou sloužit jako základ pro vypracování praktické části.

V praktické části bude proveden návrh vlastního monitorovacího systému, který bude následně implementován včelího úlu.

Sběr dat bude obstaráván pomocí senzorů, které budou posílat naměřené hodnoty na Raspberry pi, které je bude dále zpracovávat.

Vytvořený systém bude posléze nasazen a otestován v reálném prostředí.

Doporučený rozsah práce

50-60 stran

Klíčová slova

úl, včelstvo, IoT, monitorování, Raspberry pi

Doporučené zdroje informací

BELL, Charles A. *Beginning sensor networks with Arduino and Raspberry Pi*. [New York, New York]: Apress, 2013. ISBN 1430258241.

BOGUSCH, Petr. *Domečky pro včely a užitečný hmyz*. Praha: Grada, 2019. ISBN 978-80-271-2510-4.

CRAMP, David; PICKETT, Robert; PISOVÁ, Kateřina. *Včelařství : obrazový průvodce : od pořízení včelstev po medobraní : více než 400 návodných fotografií*. Čestlice: Rebo, 2014. ISBN 978-80-255-0831-2.

FLOTTUM, Kim; VOJTĚCHOVÁ, Tereza. *Příručka včelaře : jak chovat včely na dvoře, za domem, na střeše nebo na zahradě*. [Praha]: Slovart, 2019. ISBN 978-80-7529-844-7.

GUSTIN, Yves. *Ilustrované včelařství : nepostradatelná rodinná příručka pro odvážné včelaře*. V Praze: J. Radvan, 2010. ISBN 978-80-87060-27-8.

Včelařství : odborný časopis pro včelařství. Praha: ISSN 0042-2924.

Předběžný termín obhajoby

2023/24 LS – PEF

Vedoucí práce

Ing. Marek Pícka, Ph.D.

Garantující pracoviště

Katedra informačního inženýrství

Elektronicky schváleno dne 28. 11. 2023

Ing. Martin Pelikán, Ph.D.

Vedoucí katedry

Elektronicky schváleno dne 9. 2. 2024

doc. Ing. Tomáš Šubrt, Ph.D.

Děkan

V Praze dne 30. 03. 2024

Čestné prohlášení

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

V Praze dne 30.3.2024

Poděkování

Rád bych touto cestou poděkoval vedoucímu práce Ing. Markovi Píckovi, Ph.D. za cenné rady a připomínky. Dále bych chtěl poděkovat Bc. Janu Pavelkovi za rady ohledně slaboproudu.

Monitorovací systém včelího úlu za pomoci IoT

Abstrakt

Diplomová práce se zabývá návrhem a realizací systému pro vzdálené monitorování včelího úlu s využitím různých přenosových technologií, jako je LoRa nebo Wi-Fi. Pro implementaci tohoto systému jsou využity mikrokontrolery ESP32. Z včelího úlu jsou sbírána data, jako je váha úlu, relativní vlhkost, teplota a množství CO₂. Systém tyto naměřené hodnoty ukládá do databázového systému InfluxDB, ze kterého jsou data vizualizována do interaktivních dashboardů vytvořených v platformě Grafana.

Tato práce si klade za cíl vytvořit efektivní, spolehlivý a cenově dostupný systém pro monitorování včelstva, který umožní včelařům sledovat a analyzovat klíčové parametry prostředí v úlu a jejich vliv na zdraví a chování včel.

Klíčová slova: IoT, včelařství, LoRa, MQTT, úl, monitorování, Raspberry Pi

IoT beehive monitoring system

Abstract

This master's thesis deals with the design and implementation of a system for remote monitoring of beehives using various transmission technologies such as LoRa or Wi-Fi. The ESP32 microcontrollers are used for the implementation of this system. Data collected from the beehive includes hive weight, relative humidity, temperature, and CO2 levels. The system stores these measured values in the InfluxDB database system, from which the data is visualized into interactive dashboards created on the Grafana platform. The aim of this thesis is to create an efficient, reliable, and cost-effective system for beekeeping monitoring, enabling beekeepers to track and analyze key environmental parameters within the hive and their impact on bee health and behavior.

Keywords: IoT, bee keeping, LoRa, MQTT, beehive, monitoring, Raspberry Pi

Obsah

1 Úvod	12
2 Cíl práce a metodika.....	13
2.1 Cíl.....	13
2.2 Metodika.....	13
3 Teoretická východiska	14
3.1 Včelaření.....	14
3.1.1.1 Včelí dělnice	14
3.1.1.2 Matka.....	15
3.1.1.3 Trubec.....	15
3.1.2 Úl.....	15
3.1.3 Včelařské rámečky	17
3.2 Sledovatelná data	17
3.2.1 Teplota.....	17
3.2.2 Vlhkost	18
3.2.3 Váha	18
3.2.4 CO2	18
3.2.5 Hluk.....	19
3.2.6 Počet včel.....	19
3.3 Přenosové technologie.....	20
3.3.1 WiFi.....	20
3.3.2 Bluetooth	21
3.3.3 ZigBee	21
3.3.4 LoRa	22
3.4 Čidla	23
3.4.1 Senzor pro detekci CO2	23
3.4.2 Váhové senzory.....	26
3.4.3 Senzory pro teplotu a vlhkost	27
3.5 Mikrokontrolery	29
3.5.1 ESP32	29
3.5.2 Raspberry pi.....	31
3.6 Komerční řešení	32
3.7 Software.....	33
3.7.1 Node-RED	33
3.7.2 Eclipse Mosquitto	34
3.7.2.1 MQTT.....	34

3.7.3	Influx DB.....	35
3.7.4	PostgreSQL.....	36
3.7.5	MongoDb.....	36
3.7.6	Elasticsearch	36
3.7.7	Grafana	37
3.7.8	Kibana	37
3.7.9	IOTstack	37
3.7.10	OpenMQTTGateway.....	37
4	Vlastní práce	38
4.1	Stanovení požadavků.....	38
4.2	Návrh.....	38
4.2.1	Architektura	39
4.2.2	Komunikace	40
4.2.3	Úlová váha.....	42
4.2.4	Snímání hodnot uvnitř úlu	42
4.2.5	Meteostanice	47
4.2.6	Napájení.....	50
4.3	Implementace.....	51
4.3.1	Senzorická vrstva	51
4.3.1.1	Úlová váha.....	51
4.3.1.2	Úlové senzory	54
4.3.1.3	Meteostanice	63
4.3.2	Síťová vrstva.....	65
4.3.3	Vrstva zpracování dat.....	65
4.3.3.1	Příprava serveru	66
4.3.3.2	Node-red.....	68
4.4	Aplikační vrstva	72
4.5	Napájení.....	75
5	Výsledky	76
5.1	Náklady na vyhotovení.....	76
5.2	Budoucí vývoj.....	76
5.3	Výsledky testování	77
6	Závěr	78
7	Bibliografie.....	79

Seznam obrázků

Obrázek 1 Popis částí včelího úlu (6).....	16
--	----

Obrázek 2 Přehled bezdrátových přenosových technologií (69)	20
Obrázek 3 Princip PASens® senzoru (32)	24
Obrázek 4 Senzor SCD41 (32)	25
Obrázek 5 Princip měření pomocí senzoru NDIR (35).....	25
Obrázek 6 Senzor MH-Z14A (35)	26
Obrázek 7 Tenzometrický senzor váhy (37).....	26
Obrázek 8 AD převodník HX771 (39).....	27
Obrázek 9 Wheatstonův můstek (38)	27
Obrázek 10 Rozdíl DHT senzorů (41)	28
Obrázek 11 ESP32 (44).....	29
Obrázek 12 Raspberry pi (49).....	32
Obrázek 13 Api Cloud (52)	33
Obrázek 14 MQTT struktura zpráv (60).....	35
Obrázek 15 Architektura navrženého systému (vlastní)	39
Obrázek 16 Komunikační schéma (vlastní).....	41
Obrázek 17 Zapojení úlové váhy (vlastní).....	42
Obrázek 18 Schéma zapojení úlového ESP32 (vlastní)	43
Obrázek 19 Vývojový diagram setup() úlového ESP32 (vlastní).....	45
Obrázek 20 Vývojový diagram loop() úlového ESP32 (vlastní)	46
Obrázek 21 Schéma zapojení ESP32 meteostanice (vlastní)	Chyba! Záložka není definována.
Obrázek 22 Vývojový diagram setpu() meteostanice ESP32 (vlastní)	48
Obrázek 23 Vývojový diagram loop() meteostanice ESP32 (vlastní)	49
Obrázek 24 Schéma elektrického zapojení (vlastní).....	50
Obrázek 25 Přehled vodičů tenzometrického senzoru (vlastní)	52
Obrázek 26 HX711 usazený na váze (vlastní).....	53
Obrázek 27 Úlová váha (vlastní).....	53
Obrázek 28 Skript pro vypnutí automatické kalibrace (vlastní)	54
Obrázek 29 Funkce pro odeslání zprávy pomocí LoRa modulu (vlastní).....	55
Obrázek 30 Inicializace LoRa modulu a SCD41 úlové ESP32 (vlastní)	56
Obrázek 31 Inicializace úlové váhy a DHT22 úlové ESP32 (vlastní)	57
Obrázek 32 Cyklus pro měření úlové váhy (vlastní).....	57
Obrázek 33 Měření hodnot uvnitř včelího úlu (vlastní)	58

Obrázek 34 Odeslání naměřených hodnot a uspání úlového ESP32 (vlastní).....	58
Obrázek 35 Upravené schéma úlového ESP32 pro využití BME280 (vlastní)	59
Obrázek 36 Inicializace BME280 úlového ESP32 (vlastní).....	60
Obrázek 37 UTP kabel s napojenými senzory uvnitř upraveného nástavku (vlastní)	61
Obrázek 38 SCD41 mezi nástavky uvnitř úlu (vlastní).....	61
Obrázek 39 BME280 uvnitř včelího úlu (vlastní).....	62
Obrázek 40 Úl s nasazeným systémem (vlastní)	62
Obrázek 41 Funkce pro připojení na WiFi meteostanice ESP32 (vlastní).....	63
Obrázek 42 Funkce setup meteostanice ESP32 (vlastní)	63
Obrázek 43 Cyklus loop() meteostanice ESP32 (vlastní)	64
Obrázek 44 Umístění meteostanice (vlastní)	64
Obrázek 45 Umístění LoRa brány (vlastní).....	65
Obrázek 46 IoT stack menu (vlastní)	67
Obrázek 47 Obsah zprávy odeslané meteostanicí (vlastní)	68
Obrázek 48 Obsah zprávy o míře CO2 odeslané úlovým ESP32 (vlastní)	69
Obrázek 49 Propojení uzlů v Node red pro data z úlu (vlastní).....	69
Obrázek 50 Obsah uzlu change pro teplotu plodiště (vlastní).....	70
Obrázek 51 Script pro očištění váhové hodnoty o fixní položky.....	70
Obrázek 52 Uživatelské rozhraní pro zadání váhových hodnot	71
Obrázek 53 Všechny uzly v Rode red (vlastní)	72
Obrázek 54 Záložka rychlý přehled (vlastní)	73
Obrázek 55 Záložka pro Úl_1 (vlastní).....	74
Obrázek 56 Záložka pro venkovní hodnoty (vlastní).....	74
Obrázek 57 Solární panel s úly (vlastní)	75
Obrázek 58 Uložení baterie a solárního regulátoru (vlastní).....	75

Seznam tabulek

Tabulka 1 spotřeb ESP32 během nízkoenergetických režimů.....	30
Tabulka 2 zapojení pinů jednotlivých senzorů a modulů (vlastní)	43
Tabulka 3 Teoretické spotřeby systému (vlastní)	50
Tabulka 4 Nákladová tabulka	76

1 Úvod

Monitorování včelstev prostřednictvím Internetu věcí (IoT) otevírá bránu k inovativnímu přístupu k péči o včely a podporuje udržitelnost včelařského průmyslu. V dnešní době, kdy globální úbytek včelstev způsobuje znepokojení včelařů, ekologů a zemědělců, se stává klíčovým využití moderních technologií pro monitorování a optimalizaci podmínek, ve kterých včely žijí a pracují. Implementace IoT do včelařské praxe přináší možnost real-time sledování zdraví včelstev, teploty, vlhkosti a dalších důležitých parametrů. Tímto způsobem je možné nejen chránit včelí populace, ale také podporovat udržitelnost a rozvoj biodiverzity v zemědělství.

Monitorování včelstva chytrými zařízeními není žádnou novinkou. Naopak na včelařských fórech je to často diskutované téma. Vždy se ale naráží na stejný problém. Komerční řešení jsou finančně velmi nákladná a ta podomácku vyrobená nebývají příliš dobře popsána. Tato práce je založena na snaze vyrovnat se komerčním systémům při vynaložení minimálních finančních prostředků.

2 Cíl práce a metodika

2.1 Cíl

Cílem této práce je vytvořit IoT řešení pro monitorování včelstva, které umožní včelařům sledovat a analyzovat různé faktory ovlivňující zdraví a produktivitu včelstva. Tato práce je zaměřena na návrh a implementaci systému, který bude schopen monitorovat a sbírat data o teplotě, relativní vlhkosti, váze úlu, a dalších relevantních metrikách úlu ve kterém je systém nasazen. Veškeré naměřené hodnoty budou ukládány do databáze, z které budou následně data vizualizována pomocí grafů, které umožní sledovat stav včelstva v reálném čase. Důraz bude kladen na to, aby všechny části systému byly detailně zdokumentovány, aby se mohly stát inspirací pro včelaře, kteří by chtěli vytvořit podobné řešení. Tímto způsobem bude práce přispívat k ochraně a podpoře včelařství prostřednictvím moderních technologií IoT, a to za dostupnou cenu, což je zvláště důležité vzhledem k vysokým nákladům na komerční řešení.

2.2 Metodika

Práce se bude skládat z teoretické a praktické části. V teoretické části bude práce zaměřena na zkoumání aspektů spojených s včelařstvím, potřebami včelstev a možnostmi integrace technologií IoT do úlů. Bude provedeno seznámení s komerčními řešeními v oblasti monitorování včelstev, aby byl získán přehled o existujících technologiích a možnostech. Na základě této analýzy budou vybrána vhodná sensorová zařízení pro vytvářený monitorovací systém. Informace z teoretické části poslouží jako východisko pro praktickou část. V praktické části práce budou definovány požadavky, na jejichž základě bude navržen vlastní monitorovací systém pro včelí úl. Tento systém bude následně implementován a nasazen v reálném prostředí, kde proběhne jeho testování a ověření funkcionality.

3 Teoretická východiska

V teoretické části práce je částečně nastíněna problematika včelaření a také jsou detailněji popsány jednotlivé části úlu. Jsou zde vypsána teoretická data, která lze pomocí technologií IoT získat ze včelího úlu. Součástí je také rozbor různých bezdrátových přenosových technologií a seznámení se softwarem využitelným v oblasti IoT. Jedná se především o databázové a vizualizační nástroje a o popis již existujících komerční řešení.

3.1 Včelaření

Včela medonosná je příslušníkem řádu blanokřídlí, do kterého se dále řadí například i vosy nebo mravenci. Příslušníci tohoto řádu žijí a pracují ve skupinách. (1) Včelaření je praktika chovu včel s cílem produkovat med, vosk, propolis a další včelí produkty. Včelaři pečují o svá včelstva tím, že zajišťují vhodné životní podmínky, kontrolují zdraví včel a spravují hnízdní prostor. To zahrnuje inspekci úlů, léčení proti nemocem a parazitům a manipulaci s plásty pro stimulaci produkce medu nebo pro zajištění dostatečného prostoru pro rozmnožování kolonie. Když jsou podmínky příznivé, včelaři sklízí med a další produkty, čímž zajistí udržitelnost svých včelstev a přispívají k opylování rostlin v okolí. (2) (3)

3.1.1.1 Včelí dělnice

Dělnice představují nejpočetnější skupinu vzhledem ke všem obyvatelům úlu. Zpravidla dorůstají do velikosti 12 – 14 mm. Jejich úkoly se dělí podle jejich zařazení. Práci v úle obstarávají především dělnice mladušky. Ty se starají o krmení matky, trubců a plodu, hlídají vchod do úlu, čistí jej a také přijímají nektar a zpracovávají jej v med. Jsou také zodpovědné za udržování vhodné teploty a vlhkosti uvnitř úlu. Jiné skupině dělnic se říká létavky. Jejich úlohou je přinášet do úlu vodu, nezpracovaný pyl, nektar a propolis (pryskyřnatý tmel). Z celkové populace uvnitř včelstva představují létavky přibližně polovinu. Létavky a mladušky si v případě potřeby v úlohách navzájem vypomáhají. Jednotlivé úkoly se mohou lišit pro každou dělnici dle dosaženého věku. (4) Nejmladší včely se zabývají čištěním buněk, ale už po 2 – 3 dnech života začínají ošetřovat plod (1) (3)

3.1.1.2 Matka

Dříve často nazývána Královnou. Jejím jediným úkolem je klást vajíčka, kterých je schopna za den naklást až 1500. Matky se dožívají přibližně tří až čtyř let, obvykle jsou však obměňovány včelařem dříve pro prosperitu včelstva. Matka přebývá v plodišti, úl opustí pouze v případě snubních proletů. (4)

Uvnitř včelí kolonie se nachází pouze jedna matka. Na krátkou časovou periodu se může vyskytovat vícero matek ve včelstvu, ale to pouze při rojení anebo při výměně matek. (1)
(3)

3.1.1.3 Trubec

Včelí trubci neboli samci, jsou klíčovou součástí včelstva, ačkoli jejich přítomnost je omezena pouze na letní měsíce, protože v zimním období ve zdravém včelstvu nejsou přítomni. Jsou vylíhnutí z neoplozených vajíček a dosahují velikosti 15 - 17 mm s hmotností kolem 250 mg. V silných včelstvech se obvykle vyskytuje 500 až 3000 trubců.

S jejichž počtem by nemělo být manipulováno, neboť mají své specifické účely a význam v celkové dynamice včelstva. (4)

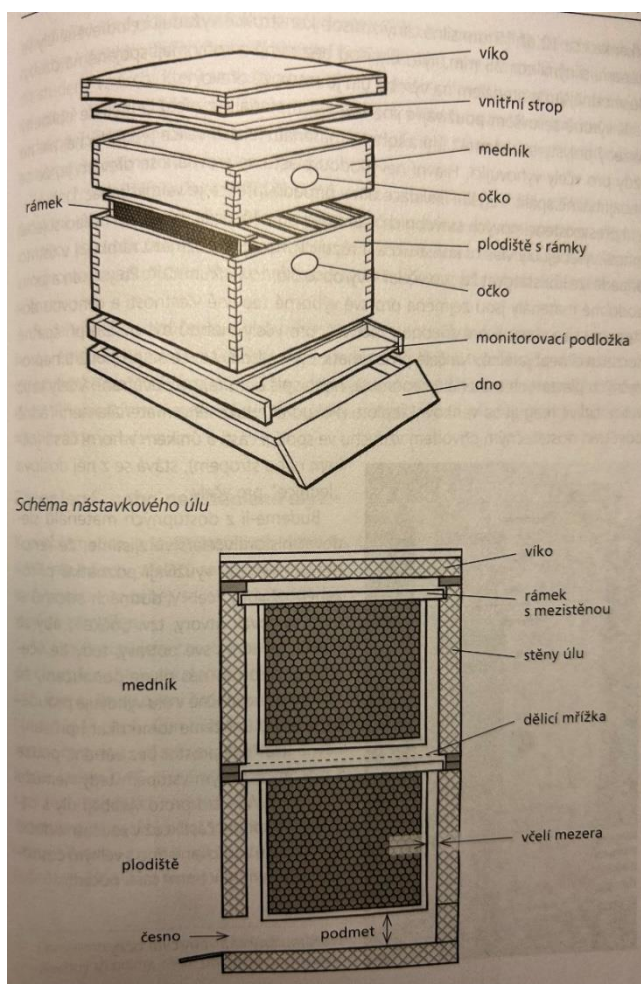
Po skončení období snůšky jsou včelí trubci vypuzeni z úlu a matka přestává klást neoplozená vajíčka, ze kterých by se vylíhnuli trubci. Dělnice již neposkytují trubcům potravu a vyhazují je ven z úlu. Trubci mají jedinou roli, která je produkce spermií a jejich přenos do pohlavních orgánů matky. K oplodnění dochází, když se trubci a matky setkají na určených místech v přírodě, která se nazývají trubčí shromaždiště. Trubci dosahují vrcholu svého sexuálního života mezi 10 a 15 dnem svého života. (4) (1)

3.1.2 Úl

Již od 19. století se na českém území vyráběly úly pro včely. Nebyly však nijak standardizované a každý si je vyráběl po svém. První pokus o zavedení úlu s jednoduchou rámkovou mírou 37 x 30 cm s názvem Čechoslovák proběhl v roce 1960. Ten se však neuchytil. Prvním sériově vyráběným nástavkovým úlem byl takzvaný Tachovský úl. Právě nástavkové úly jsou mezi včelaři v dnešní době nejrozšířenější. Včelí úly jsou v dnešní době nejčastěji vyráběny ze dřeva, mohou se však vyskytovat i například úly plastové. (5)

Nástavkový úl můžeme rozdělit do několika hlavních částí kterými jsou víko, medník, plodiště, a česno. Plodiště a medník jsou identické části úlu, které se nazývají nástavek, jediným rozdílem je jejich účel. Plodiště je určeno pro kladení plodů. Zde se nachází matka, která do rámků klade vajíčka. Medník je určen pro uchovávání medu. Plodiště je od medníku odděleno pomocí kovové mateří mřížky, která je dostatečně široká pro včely dělnice ale příliš úzká pro matku. Tímto způsobem je matce zamezen přístup do medníku a není schopná klást vajíčka do rámků určených pro medování. (1) (6)

Dno neboli podmet je nejspodnější částí úlu, vyskytuje také v různých podobách. Všeobecně je děleno na pevné a Varroa. Pevná dna mají zacelený spodek a jsou vhodnější do chladnějších období zatím co Varroa nemají dno, ale pouze oddělovací mřížku, díky čemu má úl lepší ventilaci. (2) (3) (6)



Obrázek 1 Popis částí včelího úlu (6)

3.1.3 Včelařské rámečky

Jedná se o dřevěnou konstrukci obdélníkového tvaru propletenou kovovým drátem určena pro uložení voskového plástu na výstavbu včelího díla. Stejně jako u úlů, existuje více druhů. Prvním je Hofmanův typ, který je na první pohled rozeznatelný díky bočním loučkům na obou stranách rámečku, díky kterým je dosaženo dostatečné mezery pro pohyb včelích dělnic. Zároveň po usazení do nástavku jsou loučky spojeny, což zajišťuje lepší tepelnou izolaci. Dalším typem je rámeček s mezerníkem. Ten má ploché boční hrany a vymezení mezi rámečky je docíleno pomocí válečků které jsou přibity na jedné straně bočnic. Tento typ je však v dnešní době na ústupu. Takřka identickým typem je “klasický” rámeček, který je ochuzen o mezerníky. Pro správné využívání těchto rámečků je však zapotřebí upravený nástavek, který má na bočnicích hřebeny, do kterých rámeček po usazení zapadne, ty však z pravidla mívají menší kapacitu. Po usazení všech rámečků do nástavku by měla zůstat přibližně 15 centimetrová manipulační mezera na jedné straně nástavku. (1) (3)

3.2 Sledovatelná data

V části sledovatelných dat jsou specifikována data, jež jsou obsažena v publikacích a lze je měřit pomocí IoT senzorů nebo jsou již měřena pomocí komerčních řešení. Jsou zde uvedeny hodnoty, které by se měly u sledovaných dat uvnitř včelího úlu vyskytovat.

3.2.1 Teplota

Včely jsou velmi schopné v oblasti termoregulace. Teplota se může lišit v různých částech úlu, hlavní oblastí zájmu včel je udržování ideálních teplotních a vlhkostních podmínek je plodiště. V plodišti by měla být přibližně 35 °C. (7)

Podmínky se v průběhu roku mění. Teplota uvnitř úlu by se v letních měsících měla pohybovat v rozmezí přibližně 34.8 °C až 35.5 °C. Pokud by překročila hranici 45 °C vystavuje se včelstvo riziku úhynu z důvodu přehřátí. Včely jsou však schopny teplotu samy regulovat. V případě vysoké teploty uvnitř úlu se dělnice rozmístí po úlu a kmitáním křídel do něj vhánějí studený venkovní vzduch. (7) (8) (9)

Pokud venkovní teplota klesne pod 14 °C nebo teplota uvnitř úlu pod 11 °C tak včely začínají vytvářet zimní chomáč. Uvnitř chomáče by se měla teplota pohybovat mezi 15 °C

až 30 °C na povrchu však pouze 10 °C. Teploty v úlu se mohou lišit podle aktuální polohy chomáče. (10)

Pro účely včelaření je však velmi důležitá i venkovní teplota. Vzhledem k tomu, že pro včelstvo je nejlepší minimalizovat jejich rušení, je optimální otevírat úl při teplotách mezi 15 °C až 35 °C kdy je nejvíce včel mimo úl. (7)

3.2.2 Vlhkost

Průměrná vlhkost uvnitř úlu by se ideálně měla pohybovat okolo 60 %. Může se stát, že vlhkost klesne, pokles by však neměl přesáhnout hranici 50 %, stejně tak by vlhkost v úlu neměla nikdy být vyšší než 65 %. (11)

Příliš nízká vlhkost může mít za následek vysušování plodu. Vejce nejsou schopna se vylíhnout a mohou se poškodit, což může mít za následek pokles včelí populace uvnitř úlu. Vysoká vlhkost představuje nebezpečí z důvodů větší pravděpodobnosti výskytu plísní a hub, které mohou včelstvu uškodit a zároveň snížit kvalitu produktů včelího původu jako jsou například med nebo vosk. Při vysoké vlhkosti se taktéž může na víku včelího úlu vytvářet kondenzát, který následně může na včely kapat a v zimních měsících může mít za následek prochladnutí včel, které může vézt až k jejich úhynu. Ideální vlhkost je stanovena na 60 % z důvodu nejlepších podmínek pro vypařování vody z nektaru. Voda je vypařována, dokud v nektaru nezůstane 17 % podíl vody a nestane se z něj med. Vlhkost ve včelím úlu si stejně jako teplotu regulují včely samy. (7) (11)

3.2.3 Váha

Pomocí váhy je možné sledovat přibližný přírůstek medu, což je pro včelaře pravděpodobně jeden z nejzajímavějších údajů. V zimních měsících je hmotnost využitelná především pro odhad síly včelstva, zdravotní stav včelstva, spotřebu zásob pro výživu nové generace bez nutnosti otevírání včelstev a vyrušování. Váha je ovlivňována především počtem nástavků. Většina komerčních úlových vah má maximální zátěžovou nosnost stanovenou na 150 kg, málokteré přesahují nosnost 200 kg. (12) (13)

3.2.4 CO₂

Koncentrace oxidu uhličitého se vyjadřuje v jednotkách ppm (parts per milion), někdy však také v mg/m³ nebo v procentech. Přepočítání těchto jednotek je následující: 1 000

ppm = 1 800 mg/m³ = 0,1 %. Obecně se uvádí, že člověka negativně ovlivňuje překročení koncentrace 1 000 ppm, kdy se cítíme ospale. Předpokládaná hodnota CO₂ na čerstvém vzduchu je 400 ppm. Pokud by během měření v běžných podmínkách čidlo ukazovalo méně než 400 ppm, jedná se pravděpodobně o chybná měření a je nutné čidlo překalibrovat. (14)

Koncentrace CO₂ uvnitř úlu bývá dynamická a vykazuje velké výkyvy během dne. Je zaznamenáno, že hodnoty mohou dosahovat až 23000 ppm, ale také mohou klesnout na hodnoty 450 až 500 ppm. Naměřené množství CO₂ se liší podle místa měření, v plodišti budou vždy naměřeny větší hodnoty než v medníku. (15)

Množství CO₂ je ovlivňováno i druhem dna. Úly s dnem Varroa neboli otevřeným dnem vykazují mnohem větší průměrnou koncentraci CO₂ než úly plné, poměrně o 300 ppt. Pokud však včely nemohou opustit úl z důvodu vysokého větru, koncentrace CO₂ uvnitř úlu s plným dnem dramaticky vzroste, zatím co u úlů s dnem otevřeným není efekt tak velký. (8)

Pomocí monitorování CO₂ je možné zjistit přibližnou velikost včelí kolonie. Na kilogram včel připadá přibližně 8701 ± 928 ppt (15)

3.2.5 Hluk

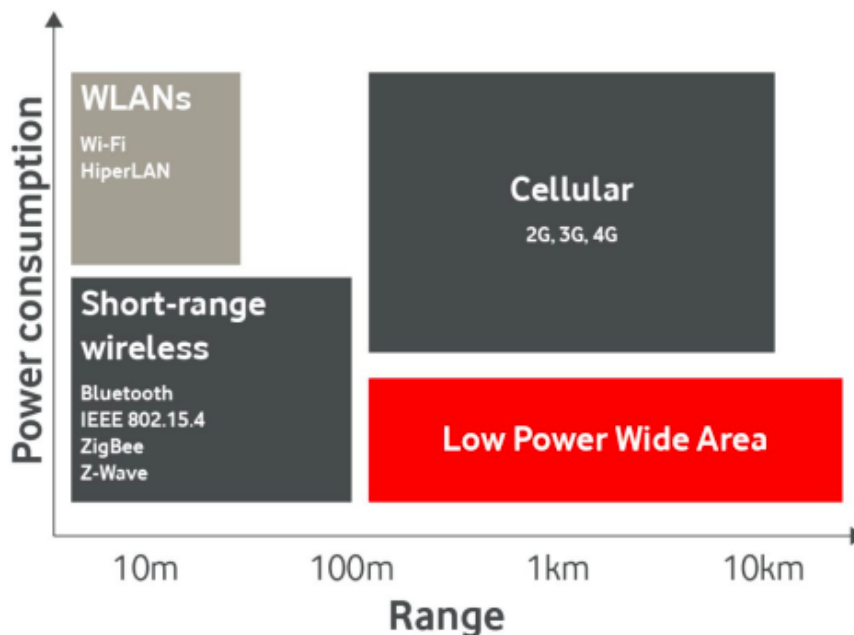
Včelí úl produkuje hluk v frekvenčním rozmezí 100 Hz až 1kHz, nejvíce se však pohybují okolo 300, 410 a 510 Hz Díky zvuku je možné odhadnout co se v daném úlu odehrává. Například před rojením se frekvence pohybuje okolo 150 Hz, ale během rojení je frekvence zvýšena na přibližně 500 Hz. Včelstvo dává pomocí zvuku najevo rojení již 21 dní dopředu Změna frekvence nastane i při zahynutí matky (16) (17)

3.2.6 Počet včel

Uvnitř úlu se nachází přibližně 60 tisíc včel, jednotlivé počítání není možné. Pomocí IoT je však možné pozorovat aktivitu včelstva na základě monitorování počtu včel vstupujících a opouštějících úl. (18)

3.3 Přenosové technologie

Při návrhu IoT řešení, je jednou z nejvýznamnějších věcí výběr správné přenosové technologie a protokoly. Více než 70 % IoT zařízení jsou připojeny bezdrátově. (19) Při výběru je nutné stanovit, jak často budou data vysílána, na jak velkou vzdálenost, jak velký objem dat bude nutné přenést, ale i v jakém prostředí bude IoT zařízení nasazeno. (20)



Obrázek 2 Přehled bezdrátových přenosových technologií (69)

3.3.1 WiFi

Wireless Fidelity neboli WiFi je technologií, která umožňuje bezdrátové připojení k internetu a lokálním sítím. Komunikační protokoly Wi-Fi jsou definovány sadou pravidel a postupů, které umožňují komunikaci mezi jednotlivými zařízeními přes bezdrátové spojení. (21) (22)

Tato technologie je neustále vyvíjena a zdokonalována s cílem poskytnout rychlé, spolehlivé a bezpečné bezdrátové připojení. Rodina standardů 802.11 je základem Wi-Fi protokolů. Tato rodina zahrnuje různé verze, jako jsou 802.11b, 802.11a, 802.11g, 802.11n, 802.11ac a 802.11ax (Wi-Fi 6). Každý standard je charakterizován určitými rychlostmi a frekvenčním pásmem, které ovlivňuje dosah a spolehlivost připojení.

Bezpečností je klíčově zajištěna ve Wi-Fi sítích. Protokoly jako WPA (Wi-Fi Protected Access), WPA2 a novější WPA3 poskytují šifrování dat a metody autentizace, které chrání síť před neoprávněným přístupem a útoky. Service Set Identifier (SSID) je identifikátorem

sítě, který umožňuje zařízením najít a připojit se k určité Wi-Fi síti. Komunikační protokoly Wi-Fi jsou také využívány standardními protokoly pro komunikaci v počítačových sítích, jako je TCP/IP (Transmission Control Protocol/Internet Protocol), kterým jsou definovány způsoby, jak data cestují po síti a jak jsou adresována a směrována. (21) (22)

3.3.2 Bluetooth

Bezdrátová technologie Bluetooth byla vyvinuta pro umožnění komunikace mezi zařízeními na krátkou vzdálenost. Původně sloužila k propojení mobilních telefonů s hands-free zařízeními, ale své využití rozšířila i do různých oblastí elektroniky. Bluetooth pracuje na krátkém dosahu, obvykle několik metrů, ale existují i verze s delším dosahem, jako například Bluetooth Class 1 s možností dosahu až 100 metrů. (20)

Frekvence, na které Bluetooth komunikuje, patří do bezlicenčního pásma ISM ve 2,4 GHz. Toto pásmo je rozděleno do kanálů umožňujících současnou komunikaci několika zařízení. Při párování se navzájem vytváří bezpečné spojení mezi zařízeními a generují se klíče pro šifrovanou komunikaci. (20)

Bluetooth existuje v několika verzích, přičemž novější verze nabízejí lepší rychlost přenosu dat a nižší spotřebu energie. Tato technologie je podporována širokou škálou zařízení, včetně chytrých telefonů, sluchátek, reproduktorů, počítačů a tiskáren. V rámci Bluetooth existuje také speciální verze nazývaná Bluetooth Low Energy (BLE), optimalizovaná pro energeticky úsporné aplikace, například fitness senzory a sledovače aktivity. Bluetooth definuje různé protokoly a profily pro specifické druhy zařízení a aplikací, což umožňuje efektivní a bezpečnou komunikaci mezi nimi. Díky své univerzálnosti a širokému využití se Bluetooth stal neocenitelným nástrojem pro bezdrátovou komunikaci mezi zařízeními v každodenním životě. (20)

3.3.3 ZigBee

ZigBee je technologií vyvinutou organizací ZigBee Alliance, která slouží jako nízkoenergetický komunikační protokol pro vytváření malých sítí typu mesh. Tento přístup umožňuje signálům snadno přeskokovat mezi různými zařízeními ZigBee, aniž by bylo nutné propojení každého z nich přímo s internetem pomocí Wi-Fi. Nicméně udržování komunikace mezi zařízeními vyžaduje použití centrálního rozbočovače. (20)

ZigBee je navržena jako jednoduchá a flexibilní technologií vhodnou jak pro vytváření malých lokálních sítí, tak i pro rozsáhlejší bezdrátové infrastruktury, které nevyžadují vysokou propustnost dat. Mezi hlavní výhody jsou zahrnuty spolehlivost, snadná implementace, minimální energetická náročnost a příznivá cena. (20)

Zigbee se často využívá v oblastech chytré domácnosti (domácí automatizace), průmyslového internetu věcí (IIoT) a dalších aplikacích, kde je zapotřebí efektivní a spolehlivá bezdrátová komunikace mezi zařízeními (20).

Dosah signálu Zigbee je z velké části ovlivněn vysílači jednotlivých zařízení, v otevřeném prostoru je přenášán až do 120 metrů a v uzavřeném prostoru na 50 metrů. (23)

3.3.4 LoRa

„LoRa je technologie, která moduluje data do elektromagnetických vln na fyzické vrstvě (rádio) umožňující komunikaci na velké vzdálenosti, LoRaWAN je komunikační protokol a architektura celé sítě.“ (24)

Long-Range neboli LoRa byla vytvořena a patentována společností Semtech v roce 2012. LoRa technologie využívá specifický modulační formát pro přenos dat, který je založen na principu rozprostřeného spektra. Tento přístup umožňuje zpracování signálů pod hranicí šumu. Klíčovou složkou této modulace jsou širokopásmové lineárně kmitočtově modulované signály známé jako "chirp". Chirp je signál, jehož frekvence se lineárně mění v průběhu času od spodního k hornímu kmitočtu (chirp-up) nebo od horního k spodnímu (chirp-down), a poté se tento proces opakuje. Tyto chirpy pak nesou informaci, která je přenášena. Díky této modulaci je LoRa technologie velmi odolná vůči úzkopásmovým rušením a různým formám vícecestného šíření signálu. (25)

LoRaWAN pracuje v nelicencovaném rádiovém spektru a využívá nižší rádiové frekvence s delším dosahem. Pro používání dálkových frekvencí platí určitá omezení pro jednotlivé země. (26)

Maximální velikost vysílané zprávy je omezena na 243 B ve směru Downlink i Uplink.

LoRa může být provozována v nelicencovaných sub-gigahertzových pásmech, jako jsou 915 MHz, 868 MHz a 433 MHz. Také může být provozována na frekvenci 2,4 GHz, což umožňuje dosažení vyšších přenosových rychlostí ve srovnání se sub-gigahertzovými pásmy, avšak s ústupky v dosahu. Tyto frekvence jsou mezinárodně vyhrazeny pro průmyslové, vědecké a lékařské použití v pásmech ISM. V Evropě LoRaWAN využívá 10 kanálů, z nichž 8 kanálů disponuje různými přenosovými rychlostmi od 250 bps do 5,5

kb/s, 1 kanál má vysokou přenosovou rychlost až 11 kb/s a 1 FSK kanál má rychlost 50 kb/s. (25) (26)

Vysílání s krátkým dosahem na kmitočtech 433 MHz a 868 MHz je možno provádět pouze v souladu s všeobecným oprávněním č. VO-R/10/07.2021-8, které vydal Český telekomunikační úřad. Tato zařízení mohou být používána bez individuálního povolení, avšak s podmínkou použití antény, která zajistí dodržení maximálního vyzářeného výkonu. Tento výkon je limitován s cílem zabránit vzájemnému rušení, zejména pokud více vysílačů vysílá na stejnou frekvenci. Používání přidavných zesilovačů a převaděčů je zakázáno, a navržená stanice nesmí být upravována. Vysílače operují na sdílených frekvencích, což nezajišťuje úplnou ochranu před rušením ze strany jiných stanic na stejné frekvenci. V případě rušení při dodržení maximálního výkonu je řešení závislé na dohodě mezi uživateli. Na frekvenci 433 MHz je povoleno vysílat v pásmu 433,05 až 434,79 MHz s výkonem 10 mW po dobu max. 6 minut za hodinu (s maximálním klíčovacím poměrem 10%) nebo s výkonem 1 mW nepřetržitě. Na frekvenci 868 MHz lze vysílat v celém pásmu 868,0 až 868,6 MHz s maximálně 1% času vysílání, což odpovídá maximálně 36 sekundám během jedné hodiny. (27) (28)

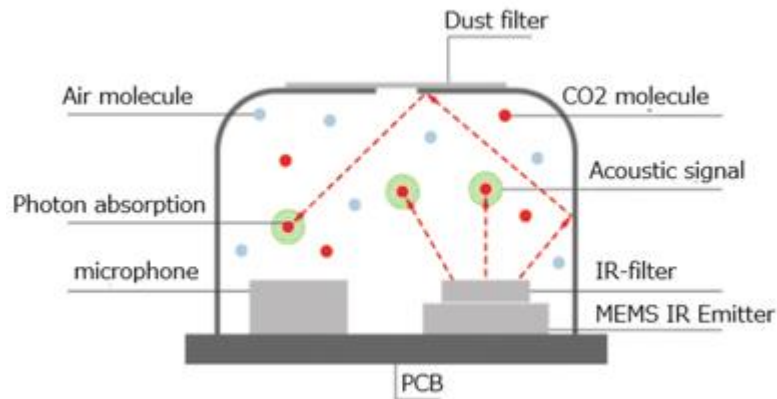
Síť LoRaWAN byla často porovnávána se sítí Sigfox, která měla obdobné charakteristiky a stejně jako LoRaWAN byla využívána v bezlicenčním pásmu 868 MHz avšak byla v mnohém odlišná. Jednalo se o celosvětovou LPWAN síť francouzské stejnojmenné společnosti, v České republice byla provozována firmou SimleCell ve spolupráci se společností T-Mobile, která poskytovala svou infrastrukturu k vybudování základnových stanic pro přenos SigFox signálu. Na rozdíl od LoRaWAN umožňovala odeslání pouze 140 zpráv za den. Společnost Sixfox však vyhlásila v roce 2022 bankrot a následující rok byla síť Sigfox v České republice vypnuta. (29) (30)

3.4 Čidla

3.4.1 Senzor pro detekci CO₂

Pro účely měření koncentrace oxidu uhličitého v ovzduší je možné využít senzory využívající technologii na principu optické detekce neboli NDIR nebo PASens® (Photoacoustic sensor technology). (31)

V případě foto akustického principu se v čidle nachází zdroj infračerveného záření (IR) a mikrofon. Světlo ze zdroje IR je skrze IR filtr očištěno a vyzařováno do uzavřeného čidla. Absorbovaná energie molekul CO₂ způsobí zvýšený tlak v obalu čidla. Tento zvýšený tlak je závislý na koncentraci CO₂ a je jeho změna je měřena mikrofonem. (31)



Obrázek 3 Princip PASens® senzoru (32)

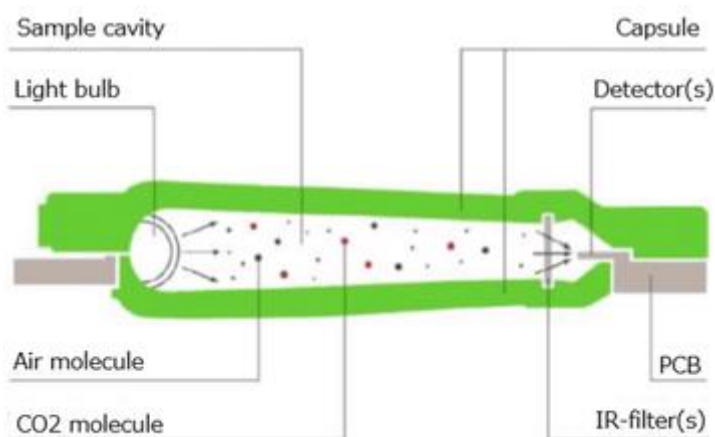
Představitelem technologie foto akustického principu je čidlo SCD41, které je primárně určeno pro měření koncentrace oxidu uhličitého v ovzduší. CO₂ je schopno měřit v rozsahu od 400 do 5000 ppm. Kromě měření CO₂ je SCD41 schopné měřit i teplotu a vlhkost. Životnost senzoru stanovena do 10 let. Čidlo je schopno měření teploty v rozsahu -10 °C až 60 °C s přesností $\pm 0,8$ °C a relativní vlhkosti od 0 do 95 % s přesností 6 %. Doporučená teplota pro dlouhodobou funkčnost je 10 °C – 50 °C. Krátkodobě je možné čidlo uchovávat v teplotách -40 °C až 70 °C. Napětí napájení čidla je 2,4-5,5V tím pádem je možné senzor využít mikrokontrolery ESP a deskami Arduino a jejími klony, čemuž přispívá i fakt, že samotný výrobce senzorů poskytuje knihovnu pro platformu Arduino. Přibližný odběr čidla je přibližně 400 μ A při jednorázovém měření jednou za 5 minut (32)



Obrázek 4 Senzor SCD41 (32)

Čidla využívající NDIR jsou principiálně jednodušší. Na jedné straně trubice se nachází emitor infračerveného záření, na opačné straně je usazen IR detektor.

Molekuly CO₂ absorbují některé vlnové délky z infračerveného spektra. Tuto absorpci detekuje a měří již zmíněný IR detektor. (33)



Obrázek 5 Princip měření pomocí senzoru NDIR (35)

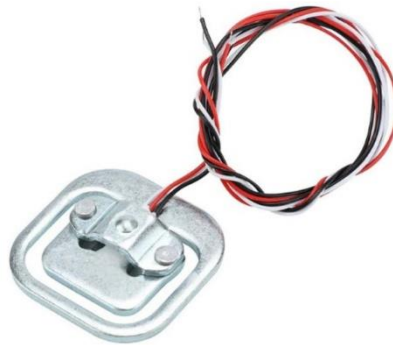
NDIR je technologií kterou využívá senzor MH-Z14. Jeho výstupem je digitální hodnota vyčítaná přes UART nebo PWM, ale také analogova hodnota 0.4-2V. Životnost čidla je stanovena na 5 let. Senzor využívá napětí 4.5 až 5.5V, z toho důvodu může senzor CO₂ fungovat na všech mikrokontrolerech Arduino s provozním napětím 5V. Pro mikrokontrolery ESP8266 a ESP32 s provozním napětím 3,3V je nutné použít externí zdroj s napětím 5V. Během provozu MH-Z14A je odběr proudu kolem 100 mA a proto není vhodný pro projekt napájený baterií. Při počátečním spuštění senzoru CO₂ potřebuje infračervená lampa přibližně 3 minuty na předehřátí, aby vytvořila optimální infračervené záření přes měřený plyn. Rozsah měření CO₂ je mezi 0 až 5000 ppm s měřicí přesností kolem +/- 50 ppm. (34) (35)



Obrázek 6 Senzor MH-Z14A (35)

3.4.2 Váhové senzory

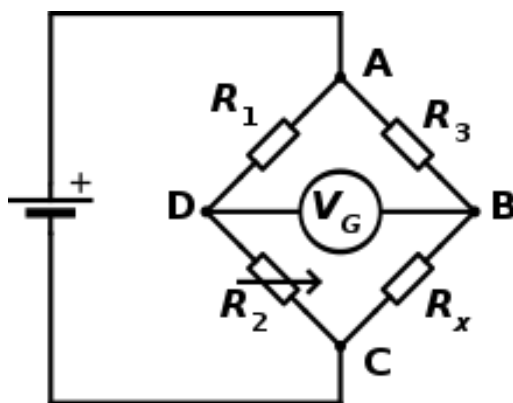
Tenzometrické senzory hmotnosti fungují na principu měření změny odporu tenzometrického materiálu při deformaci pod vlivem zatížení. Tento princip umožňuje přesné měření hmotnosti nebo síly. (36)



Obrázek 7 Tenzometrický senzor váhy (37)

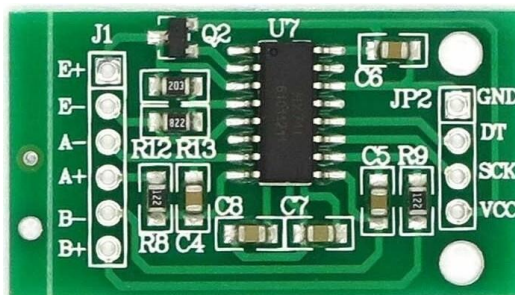
Senzory disponují vnitřním 1000 Ohm polovičním tenzometrickým můstkem s rozsahem až 50 kg. Při měření je aplikována síla na vnější stranu napínací části snímače ve tvaru písmene E a na vnější hrany pro vytvoření smykové síly v opačném směru, tj. uprostřed ohybového tahového napětí může dojít ke změnám pod napětím. (37)

V případě nutnosti navýšení měřicího rozsahu, je možné zapojit více tenzometrických senzorů do Wheatstoneova můstku, díky kterému se měřící kapacita sčítá. (38)



Obrázek 9 Wheatstonův můstek (38)

Standartně by bylo možné připojit váhové senzory přímo do analogových vstupů mikrokontroleru, ale změny napětí uvnitř senzoru jsou tak malé, že je nutné využít převodníkový modul HX771 který je určený pro vysoce přesné měření analogového signálu. Modul převede napěťový signál z váhových senzorů na digitální výstup. Díky pracovnímu napětí modulu, které se pohybuje v rozmezí 2,6-5,5 VDC, je možné váhové senzory připojit jak na desky Arduino a její klony tak i na WiFi mikrokontrolery typu ESP. (39)



Obrázek 8 AD převodník HX771 (39)

3.4.3 Senzory pro teplotu a vlhkost

Na trhu se vyskytuje velká škála senzorů na snímání teploty a vlhkosti, často jsou obsazeny na jedné desce a bývají obohaceny o schopnosti měření dalších enviromentálních ukazatelů jako jsou například tlak, nebo množství CO₂. (40)

Hlavními ukazateli při výběru jsou rozsah a přesnost naměřených hodnot, digitální či analogový výstup z čidla, (40)

Dalším důležitým faktorem, ke kterému je nutné přihlížet je spotřeba čidla, a to především z důvodu přehřívání senzoru, díky čemuž může doházet ke zkresleným měřením.

Jedním z nejrozšířenějších čidel pro měření teploty a vlhkosti je série DTH a to DHT11 a DHT22, především díky nízké ceně a relativně dobré přesnosti, především u DHT22. Oba

senzory obsahují čip, kterým je prováděn analogově-digitální převod a který na řídicí mikrokontroler odesílá digitální signál v podobě aktuální hodnoty teploty a vlhkosti.

Hlavním rozdílem mezi DHT11 a DHT22 je mimo pořizovací ceny především přesnost naměřených hodnot, ale také rozsah měření. (41) (40)

	DHT11	DHT22
Teplotní rozsah	0 až 50 °C (±2 °C)	-40 až 80 °C (±0,5 °C)
Rozsah vlhkosti	20 až 90 % (±5 %)	0 až 100 % (±2 %)
Rozlišení	Vlhkost: 1 % Teplota: 1 °C	Vlhkost: 0,1 % Teplota: 0,1 °C
Provozní napětí	3-5,5 V DC	3-6 V DC
Spotřeba proudu	0,5-2,5 mA	1-1,5 mA
Vzorkovací doba	1 sekunda	2 sekundy

Obrázek 10 Rozdíl DHT senzorů (41)

Senzory DHT se však nehodí pro prostředí s dlouhodobě vysokou vlhkostí, protože ta je schopna čidlo časem poškodit. (40)

Nezanedbatelnou výhodou senzorů DHT je že na přenos dat je využíván pouze jeden vodič, zatím co pro velkou část vlhkoměrů je využíváno rozhraní I2C které k přenosu využívá vodiče dva. (40)

Jedním z nejlépe hodnocených senzorů na trhu pro měření teploty a vlhkosti které využívá rozhraní I2C, je BME280, vyrobený společností Bosch, který mimo výše zmíněných veličin je schopen i snímat atmosférický tlak. Senzor provádí měření teploty v rozmezí -40 až 80 °C, tlaku mezi 300 - 1100hPa a vlhkosti do 100% s absolutní přesností ± 0.5 °C a ±3 % pro vlhkost. (42) Senzor se však má tendence při extenzivním měření přehřívat a měřit teploty o 1 až 2 °C jiné, než jsou reálné hodnoty. Aby se takové situaci předešlo, je zapotřebí redukovat frekvenci měření na jedno za 30 sekund. Levnější alternativou jsou senzory skupiny BMP, ty však jsou ochuzeny o možnost měření relativní vlhkosti. (40)

3.5 Mikrokontrolery

3.5.1 ESP32

Jedná se o sérii bezdrátových Wifi mikro kontrolerů od společnosti Espressif a o novější generaci ESP8266 od stejnojmenné firmy. Čipy jsou schopny pracovat v provozní teplotě $-40\text{ }^{\circ}\text{C}$ až $125\text{ }^{\circ}\text{C}$. Oproti předchozímu čipu ESP8266 obsahuje nově ESP32 vedle WiFi také Bluetooth ve verzi 4.2 s podporou BLE (Bluetooth Low Energy). Mikrokontroler je osazen dvoujádrovým mikroprocesorem 32-bit Xtensa LX6, SRAM pamětí o velikosti 512 kB, ROM pamětí o velikosti 520 kB a 4 MB flash paměti. Mimo komunikačních modulů WiFi a Bluetooth se na čipu nachází i ULP koprocesor, RTC modul a periferie, které jsou vyvedeny do GPIO pinů. (43) (44)



Obrázek 11 ESP32 (44)

Čipy ESP32 v aktivním stavu odebírají přibližně 240 mA, nikdy však odběr nemůže překročit hranici 790 mA. Z toho důvodu disponují čipy čtyřmi uspávacími funkcemi, které jsou určeny pro snížení konzumace elektrické energie mikrokontroleru, primárně napájeného omezeným zdrojem jako je baterie. Uspávací funkce omezí aktivity, které spotřebovávají více energie, ale udržují dostatek energie pro probuzení procesoru, pokud je potřeba využití větších zdrojů ESP. Uspávací funkce dělíme na Modem Sleep, Light Sleep, Deep Sleep a Hibernation. (45)

Modem-sleep	The CPU is powered on.	Max speed 240 MHz: 30 mA ~ 50 mA
		Normal speed 80 MHz: 20 mA ~ 25 mA
		Slow speed 2 MHz: 2 mA ~ 4 mA
Light-sleep	-	0.8 mA
Deep-sleep	The ULP co-processor is powered on.	150 μ A
	ULP sensor-monitored pattern	100 μ A @1% duty
	RTC timer + RTC memory	10 μ A
Hibernation	RTC timer only	5 μ A
Power off	CHIP_PU is set to low level, the chip is powered off	0.1 μ A

Tabulka 1 spotřeb ESP32 během nízkoenergetických režimů (45)

Při využití Modem Sleep je zapnut pouze procesor, ULP koprocessor a RTC modul. Periferie, Wifi a Bluetooth jsou vypnuty, díky čemuž je možné snížit spotřebu přibližně na 2 až 50 mA. Aby však byla pořád udržována komunikace, je čip v pravidelných intervalech probouzen do aktivního stavu. Tento jev se v anglickém jazyce nazývá Association Sleep Pattern. Interval, ve kterém je čip uspáván je většinou mezi 100 až 1000 ms. Stejný způsob probouzení využívá i Light Sleep, jediný rozdíl oproti Modem Sleep je v tom, že procesor má omezenou taktovací frekvenci. RTC a ULP koprocessor zůstávají aktivní, díky čemuž je možné dosáhnout odběru okolo 0.8 mA. Během Deep Sleep stavu se procesor úplně vypnutý, aktivní zůstanou pouze RTC modul a ULP koprocessor, který je schopný zaznamenávat údaje ze senzorů a vzbudit CPU, pokud je potřeba. Tímto způsobem je odběr snížen na přibližně 0.15 mA až 10 μ A. S CPU je vypnuta i hlavní paměť, jediná paměť která je během Deep Sleep modu aktivní je RTC paměť. Do té jsou před usmáním ukládána data o připojení na Wifi a Bluetooth. Zdaleka nejmenšího odběru je možné dosáhnout pomocí Hibernation mode která dokáže být až poloviční oproti Deep Sleep. V hibernačním režimu je jediným aktivním prvkem RTC časovač a několik RTC GPIO výstupů pro vzbuzení čipu. Není zapnuta ani RTC recovery paměť, takže není možné cokoli ukládat. (46)

V případě, že je potřeba ESP vzbudit, je možné využít jeden z následujících metod nebo jejich kombinaci.

- Časovač
- Dotykem
- Externí vzbuzení

K probouzení pomocí časovače je využíván RTC kontrolér, který má zabudovaný časovač, který probouzí ESP po předefinovaných časových intervalech. Tato metoda je ideální pro periodické úkony, Probouzení časovačem je ovládáno funkcí

esp_sleep_enable_timer_wakeup(time_in_us) ve která se definuje za jak dlouhou dobu v mikrosekundách se má ESP vzbudit. Další metodou je využití takzvaných dotykových (kapacitních) pinů, které jsou v diagramech desek označovány popisem „Touch“. Kapacitní dotykové senzory lze využít k detekci jakýchkoli elektrických a magnetických vln okolo, stačí k těmto pinům připojit jakýkoli vodivý materiál. (43) K probouzení dotekem slouží funkce *esp_sleep_enable_touchpad_wakeup()*. Poslední metodou probuzení je externím zdrojem. Existují dva druhy externích spouštěčů, jedním je ext0, který umožní ESP probouzet se specifickým GPIO pinem, druhý je ext1, ten je využíván v případě využití většího počtu GPIO pinů. Pro externí buzení je nutné použít piny které jsou v diagramech desek označovány popisem „RTC_GPIO“, ESP je probuzeno, pokud se na pinu změní logika. Jsou spouštěny funkcemi:

```
esp_sleep_enable_ext0_wakeup(GPIO_PIN, LOGIC_LEVEL)  
esp_sleep_enable_ext1_wakeup(BUTTON_PIN_MASK, LOGIC_LEVEL)
```

(45) (47)

3.5.2 Raspberry pi

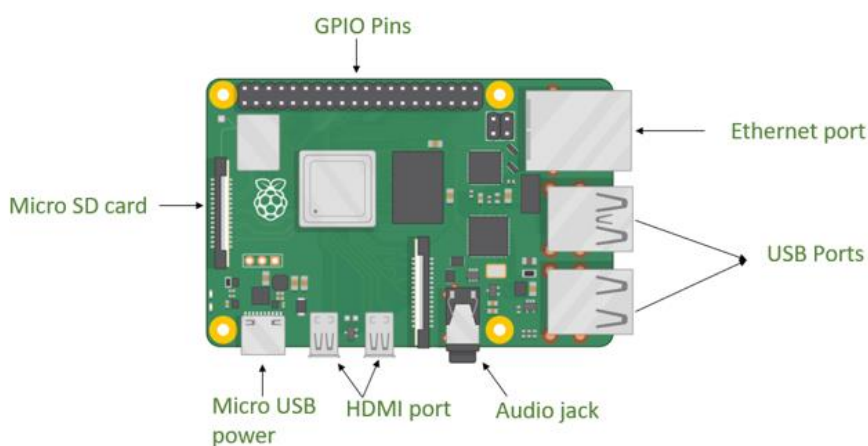
„Raspberry Pi je řada jednodeskových počítačů, které byly určeny pro vzdělávací účely. Konstrukce všech počítačů Raspberry Pi je založena na jedné desce plošných spojů, která obsahuje mikrokontrolér s procesorem a pamětí a také na čtyřiceti pinech GPIO a konektorech umožňujících komunikaci. Původní zařízení a příslušenství vytvořila Raspberry Pi Foundation“ (48)

Existuje mnoho modelů Raspberry Pi, všechny však disponují vestavěným procesorem, pamětí SDRAM nebo SRAM, piny GPIO, konektorem pro paměťovou kartu a portem pro USB napájení. Jednodeskové počítače Raspberry pi jsou postaveny na architektuře ARM. Jednotlivé modely mohou disponovat dodatečnými rozšířeními jako jsou video a audio konektory, nebo možností síťového připojení (Wifi, Ethernet, Bluetooth). Hlavními sériemi jsou Pico, Zero a hlavní série nesoucí pouze název Raspberry Pi. Desky Zero jsou nejlevnější, nejmenší a nejúspornější z minipočítačů a jsou primárně určeny k prototypování a do nenáročných vestavných a přenosných projektů. Pico je programovatelné v jazycích C a MicroPython, tudíž je přizpůsobitelné široké škále aplikací. (49)

Žádná z verzí nedisponuje pevným diskem. Nahrávání operačního systému a ukládání dat je prováděno na přenosná média, primárně na SD kartu. V době vyhotovování této diplomové práce je aktuální vlajkový model Raspberry Pi 5. (49)

Oficiálním podporovaným operačním systémem pro Raspberry Pi je Raspberry Pi OS, dříve nazývaný Raspbian který, je dostupný v několika různých verzích. Jejich hlavními rozdíly jsou přítomnost uživatelského rozhraní, případně obohaceným doporučenými aplikacemi a 32 bit nebo 64 bit verze. Jedná se upravenou distribuci Linuxu Debian. Architektura jednodeskových počítačů Raspberry Pi zůstala díky čemuž je možné provozovat aktuální verzi Raspberry Pi OS i na nejstarším modelu. (50)

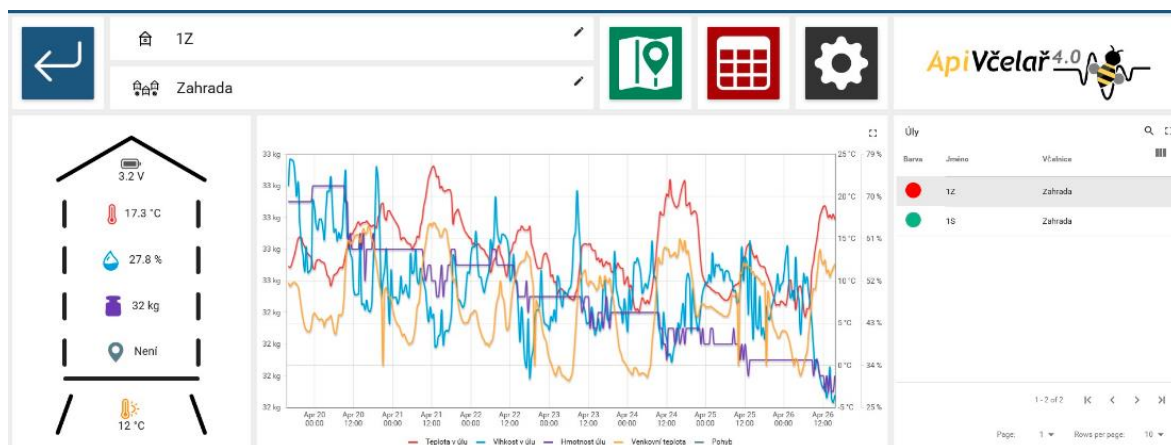
Raspberry Pi OS však není jediným operačním systémem, schopným fungovat na desce Raspberry Pi, existuje velká řada alternativních systémů, jako jsou například Windows 10 IoT Core, Ubuntu mate. RICS nebo systémy které jsou zaměřeny na specifický účel jako Batocera který je určený pro emulaci retro herních konzolí. (50) (51)



Obrázek 12 Raspberry pi (49)

3.6 Komerční řešení

Komerční řešení jsou často prodávána po částech. Těmi hlavními jsou úlová váha, snímací senzor uvnitř úlu, případně dodatkové komunikační moduly nebo externí napájení. Napájení těchto systémů je často řešeno pomocí interní baterie LiFePo4 případně Li-Ion. Například Výrobce ApiVčelař 4.0 na svých stránkách udává, že stanovená výdrž baterie úlové váhy bez dobíjení případně výměny je stanovena na 1 rok. Většina výrobců využívá vlastní proprietární systém pro zobrazení naměřených hodnot. (52)



Obrázek 13 Api Cloud (52)

Vesměs všechna řešení jsou schopna měřit vlhkost, teplotu, a váhu úlu, velký důraz je kladen na určování polohy pomocí GPS modulu, některá řešení jsou pak rozšířena o možnost měření tlaku nebo případně analýzu zvuku. (53) (12)

Jako úložiště je často využívána SD karta přímo u váhy. V případě externí komunikace je ve většině případů využívána GSM brána na spojení pomocí NB-IoT, případně Bluetooth, u kterého sami výrobci uvádějí efektivní dosah do 50 metrů. U komerčních řešení bývají podstatně vyšší provozní náklady, v případě Bee Hive Monitoring je nutné si připočítat přibližně 20 € za roční provoz SIM karty pro bránu GSM a dalších a dalších 19 € za roční licenci, pokud je využívána počítačová aplikace. (18)

3.7 Software

3.7.1 Node-RED

Node-RED byl vyvinut společností IBM. Jedná se o open-source nástroj pro vizuální programování a řízení toku dat primárně pro IoT projekty. Základními prvky jsou uzly, které reprezentují různé funkce a operace. Uzly jsou propojeny linkami, což umožňuje snadnou tvorbu toku dat mezi zařízeními a službami. (54)

Node-RED poskytuje rozsáhlou knihovnu uzlů, která zahrnuje standardní uzly pro manipulaci s daty, komunikaci přes různé protokoly (MQTT, HTTP, WebSocket), integraci s cloudovými službami a mnoho dalšího. (55)

Součástí Node-RED je i modul Node-RED Dashboard který poskytuje sadu uzlů, díky kterým je možné vytvářet dashboardy pro vizualizaci naměřených dat uzly pro rychlé vytvoření uživatelského rozhraní s tlačítky, posuvníky, grafy, měřítky.

3.7.2 Eclipse Mosquitto

Mosquitto byl původně vyvinut pod nadací Eclipse jako open-source projekt a je distribuován pod licencí Eclipse Public License (EPL). Jeho hlavním účelem je poskytovat spolehlivý a škálovatelný broker pro komunikační protokol MQTT, který umožňuje komunikaci mezi různými zařízeními a systémy v IoT sítích. (56)

Mosquitto podporuje různé úrovně kvality služeb (QoS) pro zajištění spolehlivého doručování zpráv, šifrování pomocí TLS/SSL pro zabezpečenou komunikaci, podpora pro autentizaci a autorizaci, a další. Je možné ho nasadit na různých platformách, včetně Linuxu, Windows, a macOS. (56)

Eclipse Mosquitto je často používán v projektech IoT pro správu a zprostředkování zpráv mezi zařízeními v síti, což přispívá k efektivní a spolehlivé komunikaci v rozsáhlých IoT infrastrukturách. (56)

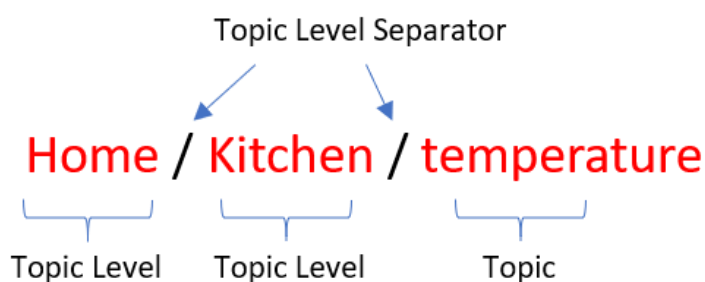
Podporuje všechny verze MQTT. (57)

3.7.2.1 MQTT

Jedná se o nízkonákladový komunikační protokol vyvinutý přímo pro IoT v roce 1999. V těchto dobách bylo standardem, používat pro přenos informací protokol HTTP, ten se však osvědčil jako neefektivní v náročných podmínkách z důvodů vysoké zdrojové náročnosti při komunikaci a složité implementaci což dalo za vznik protokolu Message Queue Telemetry Transport, který původně určen pro monitorování ropovodů přes nespolehlivé satelitní připojení. MQTT je protokol navržený pro efektivní a spolehlivý přenos zpráv mezi zařízeními, zejména v prostředích s omezenými prostředky a díky své stabilitě a nízké spotřebě energie na provoz je jedním z nejrozšířenějších nízkonákladových komunikačních protokolů využívaných IoT zařízeními. MQTT je využíváno i v jiných odvětvích. Například pro přenos zpráv na sociálních sítích jako je Instagram nebo Facebook (58) (59)

Komunikace prostřednictvím MQTT je založena na hvězdicové topologii kdy centrálním prvkem je Broker, jehož úkolem je předávání zpráv mezi klienty. Ti jsou děleny podle jednotlivých témat na poskytovatele zpráv (Publisher) a odběratele (Subscriber), kteří mají přihlášený odběr k danému tématu. Každý klient může být současně poskytovatelem i odběratelem daného tématu. Poskytovatel zpráv obvykle představuje nějaký druh senzoru nebo mikrokontroleru, který vysílá naměřené hodnoty na brokera. (58) (59)

Poskytovatel zpráv přidělí právě jedno téma (topic). Všechny přenášené zprávy jsou systematicky uspořádány podle témat. Každá zpráva je přiřazena právě jednomu tématu (topic). Před tím, než se odběratel stane aktivním účastníkem komunikace, je třeba, aby se předem přihlásil u brokera k odběru konkrétního tématu. Témata mohou mít vrstvy, jež jsou odděleny lomítky. Tato flexibilita umožňuje odběrateli selektivně přijímat zprávy související s určitou oblastí, například specifickou místností. Je důležité si uvědomit, že témata jsou citlivá na velikost písmen. (58) (59) (60)



Obrázek 14 MQTT struktura zpráv (60)

Odběratel obdrží pouze zprávy týkající se témat, ke kterým má předem potvrzený odběr u brokera. V rámci tohoto systému stačí, aby odběratel znal pouze IP adresu brokera a název tématu ke kterému chce přihlásit odběr. (60)

Témata v MQTT jsou kódována ve formátu UTF-8, což znamená, že podporují diakritiku. Neexistuje žádný pevně stanovený standard pro hierarchii témat. Ta závisí na konkrétní aplikaci a návrhu programátora. (60)

Zprávy obsahují binární data. Broker s daty uvnitř zpráv nemanipuluje, a tak není formát dat při přenosu zásadní. Záleží pouze na schopnostech odběratele a jeho schopnosti zprávu interpretovat. (60)

Pro komunikaci standardně využíván TCP/IP port 1883 pro nezabezpečenou komunikaci a port 8883 pro zabezpečenou komunikaci. (61)

Velikost zprávy je omezena na 256 MB. MQTT minimalizuje objem přenášených dat, což byl od začátku hlavní cíl. Tím se MQTT stává ideální volbou pro přenos příležitostných informací a hodnot, které nevyžadují vysokou šířku pásma, což je zvláště vhodné pro účely IoT. (61)

3.7.3 Influx DB

Jedná se o open-source databázi pro ukládání časových řad, která byla navržena speciálně pro ukládání, dotazování a vizualizaci časově citlivých dat. Tato databáze je

optimalizována pro práci s časovými řadami, což znamená data, která jsou označena časovým razítkem, jako jsou telemetrická data, metriky, logy nebo jakékoli časově orientované informace. (62)

InfluxBD je často využívána spolu s vizualizačním nástrojem Grafana. (62)

3.7.4 PostgreSQL

PostgreSQL se řadí mezi robustní databázové servery s řadou pokročilých funkcí. Často je přirovnáván k MySQL. Podobně jako ten je PostgreSQL dostupný pro všechny hlavní platformy a je také šířen jako open source. Oproti MySQL je však PostgreSQL občas mírně pomalejší, ale tuto nevýhodu kompenzuje svou dlouhodobou bezpečností a spolehlivostí, s níž mohou soupeřit pouze poslední verze MySQL. Mezi výhody PostgreSQL patří také jeho vynikající rozšiřitelnost a schopnost spouštět procedury napsané v jazycích jako je Perl, Python a další. (63)

3.7.5 MongoDB

MongoDB je dokumentově orientovaný databázový systém, který patří mezi tzv. NoSQL databáze. Jedná se o open-source databázi, která nabízí flexibilní a škálovatelné úložiště dat. Jedním z hlavních principů MongoDB je koncept dokumentů, což jsou záznamy, které mohou obsahovat různé typy dat a struktury, a které jsou uloženy ve formátu BSON (Binary JSON). Tento formát umožňuje efektivní ukládání a zpracování dat v databázi. (64) (65)

3.7.6 Elasticsearch

Elasticsearch je open-source distribuovaný vyhledávací a analytický systém, který je navržen pro efektivní ukládání, vyhledávání a analýzu velkých objemů dat v reálném čase. Jedná se o jednu z nejpoužívanějších technologií pro fulltextové vyhledávání, indexaci a analýzu strukturovaných i nestruturovaných dat. Elasticsearch je postaven na bázi Apache Lucene a je široce využíván pro různé účely, včetně vyhledávání informací, logování, monitorování, business intelligence a textové analýzy. (66)

3.7.7 Grafana

Grafana je open-source platforma, jejíž hlavním účelem je vizualizace a monitorování dat. Nabízí širokou škálu vizualizací, jako jsou grafy, diagramy, tabulky a jiné. Jedná se o nástroj, který nám umožňuje vytvářet řídicí panely v rámci naší nástěnky (tzv. „dashboard“) a následně v reálném čase dotazovat a zobrazovat metriky a data z různých zdrojů, jako například MySQL, InfluxDB, Prometheus, Gitlab a mnoho dalších. (65)

3.7.8 Kibana

Kibana je open-source nástroj pro vizualizaci a analýzu dat, který je vyvinut firmou Elastic. Je navržen tak, aby spolupracoval s Elasticsearchem a umožňoval uživatelům efektivně vizualizovat a analyzovat data uložená v Elasticsearch indexech. Kibana poskytuje uživatelské rozhraní pro tvorbu interaktivních grafů, tabulek, dashboardů a dalších vizualizací dat, které umožňují uživatelům objevovat vzory, trendy a informace skryté v jejich datech. Mezi další funkce Kibany patří možnost dotazování nad daty, filtrování, agregace a vyhledávání, což umožňuje uživatelům hlouběji prozkoumat a porozumět svým datům. (66)

3.7.9 IOTstack

IOTstack je nástroj pro tvorbu docker-compose prostředí, který umožňuje snadno vytvářet a udržovat IoT kontejnery na Raspberry Pi. IOTstack obsahuje řadu předdefinovaných konfigurací pro běžné IoT aplikace a služby, jako jsou Node-RED, InfluxDB, Grafana, Mosquitto a další. (67)

3.7.10 OpenMQTTGateway

Jedná se o open-source projekt, sloužící jako brána pro převod různých bezdrátových protokolů na MQTT (Message Queuing Telemetry Transport). (68)

OpenMQTTGateway je často používáno v IoT projektech, kde je potřeba integrovat různé typy senzorů a dalších zařízení komunikujících pomocí různých bezdrátových technologií (např. Zigbee, Bluetooth, RF, Infrared atd.) do MQTT. (68)

Firmware je možné provozovat na velké řadě desek jako jsou například ESP32, ESP8266 nebo Arduino MEGA. Pro LoRa bránu je doporučována deska TTGO LoRa32. (68)

4 Vlastní práce

Část vlastní práce je zaměřena na stanovení požadavků na systém vzdáleného monitorování včelstva, Z požadavků je vytvořen návrh architektury celého systému, na který navazuje podrobný popis implementace do reálného prostředí.

4.1 Stanovení požadavků

Systém by se měl sběrem dat vyrovnat komerčním řešením, především by měl měřit váhu úlu, jelikož mezi včelaři se často jedná o hlavní důvod k nasazení IoT monitorovacího systému. Již zmíněným problémem je však cena zařízení, které je schopné nejen váhu naměřit, ale i získaná data poslat. Většina komerčních řešení snímá teplotu a vlhkost.

Systém by měl být schopen naměřená data dlouhodobě uchovávat z důvodu budoucího možného zpracování. To znamená, že data by měla být ukládána do paměti systému nebo na externí úložiště s dostatečnou kapacitou a spolehlivostí, aby bylo zajištěno zachování dat v dlouhodobém horizontu. Toto uchování dat umožní budoucí analýzu nebo další zpracování, které může být pro účely monitorování či optimalizace včelařských stanovišť velmi užitečné.

Včelařská stanoviště jsou často umístěna na odlehlých místech, což může znamenat omezený přístup k elektrické síti. Systém by proto měl využívat jiný energetický zdroj než napájení z elektrické sítě. To může zahrnovat solární panely, baterie nebo jiné zdroje obnovitelné energie, které umožní nezávislý provoz systému i v odlehlých lokalitách.

Dále by systém měl být schopen přenášet naměřená data na delší vzdálenost. Což znamená, že komunikační technologie používaná v systému by měla být schopna dosáhnout spojení na větší vzdálenostech, než je běžné u domácích síťových zařízení.

4.2 Návrh

V rámci návrhu je popsána architektura celého systému, popis komunikace a komunikačních prvků, zapojení jednotlivých čidel a postup měření jednotlivých částí systému.

4.2.1 Architektura

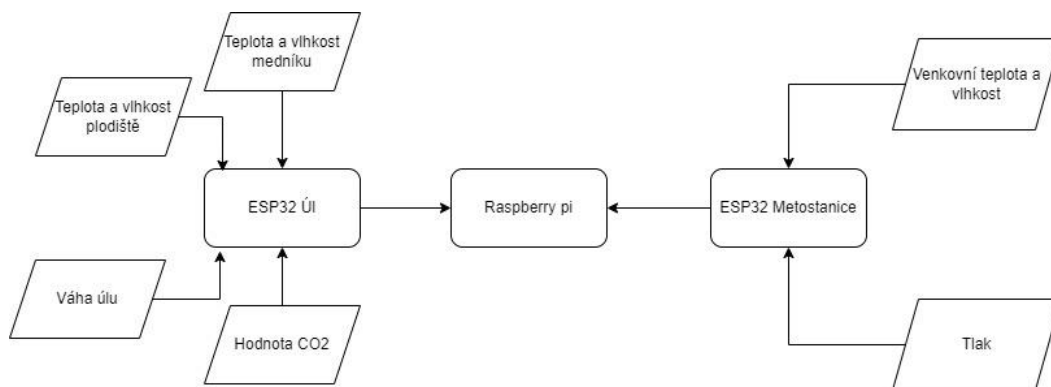
Centrálním prvkem, který uchovává veškerá naměřená data a zajišťuje interakci s uživatelem, je Raspberry Pi. Pro tento projekt byl zvolen model 3B+ zejména pro jeho dostatečný výkon a nižší pořizovací náklady ve srovnání s novějšími modely. Jako operační systém byl vybrán Raspberry Pi OS, který je doporučen výrobcem tohoto zařízení. Ten je uložen na 32GB SD kartě, což zajistí dostatečný prostor pro operační systém a aplikace. Pro zjednodušení práce s aplikacemi je na Raspberry Pi spuštěn kontejner IoTstack, který obsahuje následující aplikace:

- Node-RED – Automatizační nástroj, zpracování dat
- InfluxDB – Databázový systém, ukládání dat
- Grafana – Vizualizační platforma, prezentace dat
- Eclipse Mosquitto – Zprostředkovávání MQTT komunikace

Za sběr a odesílání dat naměřených na včelím úlu je zodpovědný mikrokontrolér ESP32. Tento komponent zajišťuje monitorování klíčových faktorů prostředí v úlu, jako je váha úlu, teplota a vlhkost medníku i plodiště, a množství CO₂ uvnitř úlu. Tyto údaje jsou klíčové pro sledování zdraví a aktivit včelstva a poskytují cenné informace pro včelaře.

Nicméně, aktivita včel je ovlivňována také venkovními podmínkami. Z tohoto důvodu byla vyvinuta jednoduchá venkovní meteostanice, která monitoruje venkovní teplotu, vlhkost a tlak. Tyto informace jsou důležité pro porozumění prostředí, ve kterém se včely pohybují a pracují, a mohou pomoci včelařům lépe porozumět včelím aktivitám v kontextu venkovního prostředí.

S pomocí těchto datových zdrojů mohou včelaři sledovat a analyzovat jak vnitřní, tak i venkovní podmínky, které ovlivňují chování a zdraví včelstva. Tato data mohou být také využita k optimalizaci péče o včely a maximalizaci jejich produktivity a blahobytu.



Obrázek 15 Architektura navrženého systému (vlastní)

4.2.2 Komunikace

Problematika spojená se síťovou vrstvou v rámci IoT zahrnuje dvě klíčová témata: strukturu sítí a specializované komunikační protokoly pro IoT. Úspěch či neúspěch konkrétního řešení je výrazně ovlivněn volbou vhodné architektury pro přenos dat, přičemž tuto volbu determinují vnější faktory instalace senzoričké sítě, jako jsou klimatické podmínky, environmentální vlivy, intenzita provozu a další. Při rozhodování o architektuře sítě je klíčové zohlednit specifika prostředí, ve kterém bude IoT nasazeno. Různé klimatické podmínky a požadavky na spolehlivost mohou vyžadovat odlišné přístupy k síťovému návrhu. Zároveň je nezbytné brát v úvahu intenzitu provozu v daném prostředí, abychom zajistili dostatečnou škálovatelnost a výkonnost sítě. Specializované IoT komunikační protokoly představují další klíčový prvek v síťové vrstvě. Jejich správný výběr může výrazně ovlivnit efektivitu přenosu dat a celkový výkon sítě IoT. Při implementaci těchto protokolů je opět nezbytné brát v úvahu specifika prostředí a požadavky konkrétního nasazení.

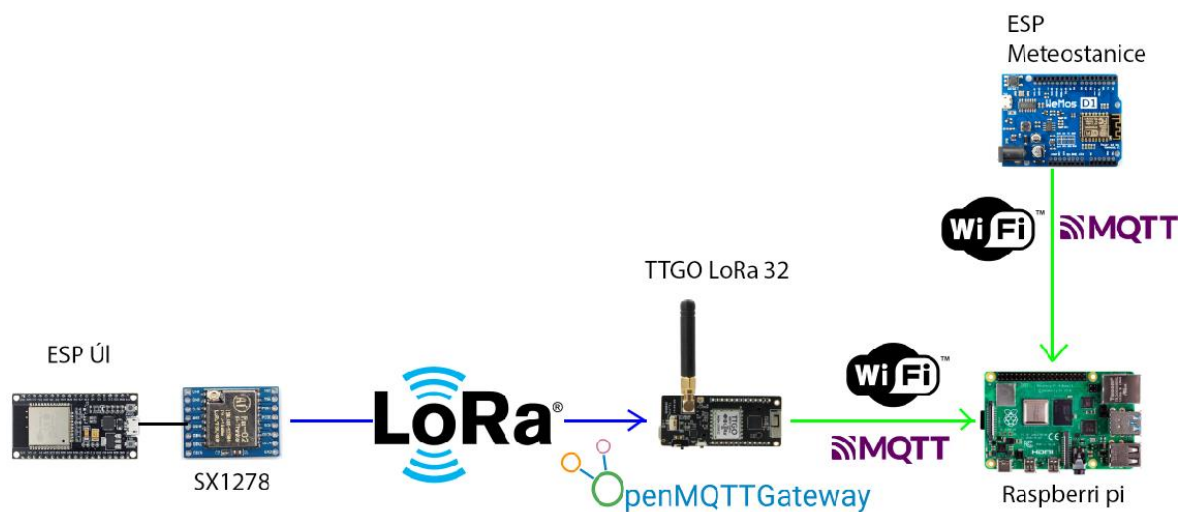
Přesto že pro prostředí, ve kterém je systém nasazen se jevila jako nejvhodnější technologie pro zajištění komunikace Wifi, díky již zavedené infrastruktuře a množství nepřetržitého napájení z elektrické sítě, tak zvolena nebyla z důvodu nedostatečné reprezentace běžných podmínek, ve kterých jsou včelstva často chována. Včelíny a samostatné úly se v mnoha případech nacházejí na odlehlých místech bez elektrické infrastruktury, z čehož je možné stanovit hlavní faktory které ovlivnili výběr přenosové technologie, kterými jsou nízká energetická náročnost na přenos a možnost posílat data na velkou vzdálenost. Z tohoto důvodu byli vybráni komunikační protokol LoRaWAN.

Pro vysílání na síti LoRaWAN je v České republice možné využívat frekvence 863 MHz až 870 MHz nebo 433 MHz. Pro systém byla zvolena frekvence 433MHz z důvodu mnohem lepších vlastností dostupnosti do zalesněných kopcovitějších terénu. V případě, že by se monitorované úly nacházely ve městě, bylo by mnohem vhodnější využití vyššího frekvenčního spektra z důvodů vysokého rušení kolem frekvence 433MHz. Mikrokontroler ESP32 sám o sobě není schopen komunikace přes LoRaWAN. Pro využití LoRa je zapotřebí modul, který umožňuje komunikaci se sítí LoRa, například vysílač SX1278. Na trhu je možné najít ESP desky které už v základu kompilátor obsahují jako je například deska TTGO LoRa 32 nebo Heltec WiFi LoRa v3, ty však mají výrazně vyšší pořizovací

náklady. Z toho důvodu byla zvolena volba „holého“ mikrokontroleru ESP32 s využitím SX1278 pro zachování minimálních pořizovacích nákladů na jednu měřící jednotku.

Mimo komunikačního protokolu LoRaWAN je v systému využita WiFi kterou ke komunikaci využívá Raspberry pi. Aby bylo možný přenos LoRa zpráv odeslaných od úlu do Raspberry pi je nutné využití LoRa brány, která má za úkol překlad příchozích zpráv na formát, který je zpracovatelný pomocí Raspberry pi. Cena LoRa bran se může pohybovat v řádu několika tisíců, pro zachování minimálních pořizovacích nákladů, byla zvolena možnost vytvořit si vlastní za pomoci firmwaru OpenMQTTGateway.

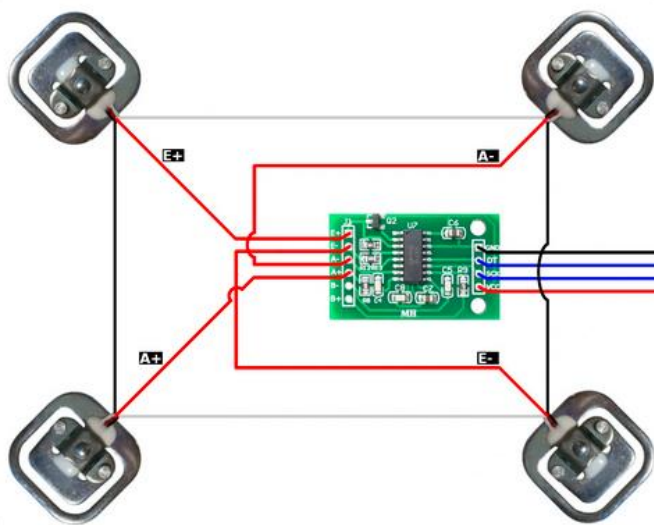
Síť Wifi je využívána pouze mezi Raspberry pi. LoRa bránou a meteostanicí. Z důvodu doporučení vývojářem softwaru byla zvolena výše uvedená deska TTGO LoRa 32.



Obrázek 16 Komunikační schéma (vlastní)

4.2.3 Úlová váha

Pro měření váhy je využívána úlová váha, která využívá tenzometrické senzory. Ideální kapacita váhy by měla být 200 kg, avšak minimální kapacita je 100 kg. Z důvodu finanční náročnosti samostatných senzorů s velkým měřitelným spektrem byla zvolena alternativní možnost porízení většího množství senzorů s menší kapacitou, jež jsou následně zapojeny do Wheatstoneova můstku, čímž je dosaženo navýšení kapacity za nižší náklady. Pro možnost měření hmotnosti do 200 kg bylo vybráno čtyři senzory s kapacitou 50 kg. Všechny senzory jsou vzájemně propojeny a následně připojeny do převodníkového modulu HX771, který zesílené napětí z úlové váhový převádí na digitální podobu. Vstupy na HX771 jsou označeny písmeny E, A, B, za kterými následuje znaménko "-" nebo "+". Při zapojení senzorů do Wheatstoneova můstku je nezbytné dbát na pořadí váhových čidel, kdy musí být ve sloupci vždy stejné znaménko, ale rozdílný znak. Maximální počet tenzometrických senzorů, které lze zapojit na jeden AD převodník HX771, je šest.



Obrázek 17 Zapojení úlové váhy (vlastní)

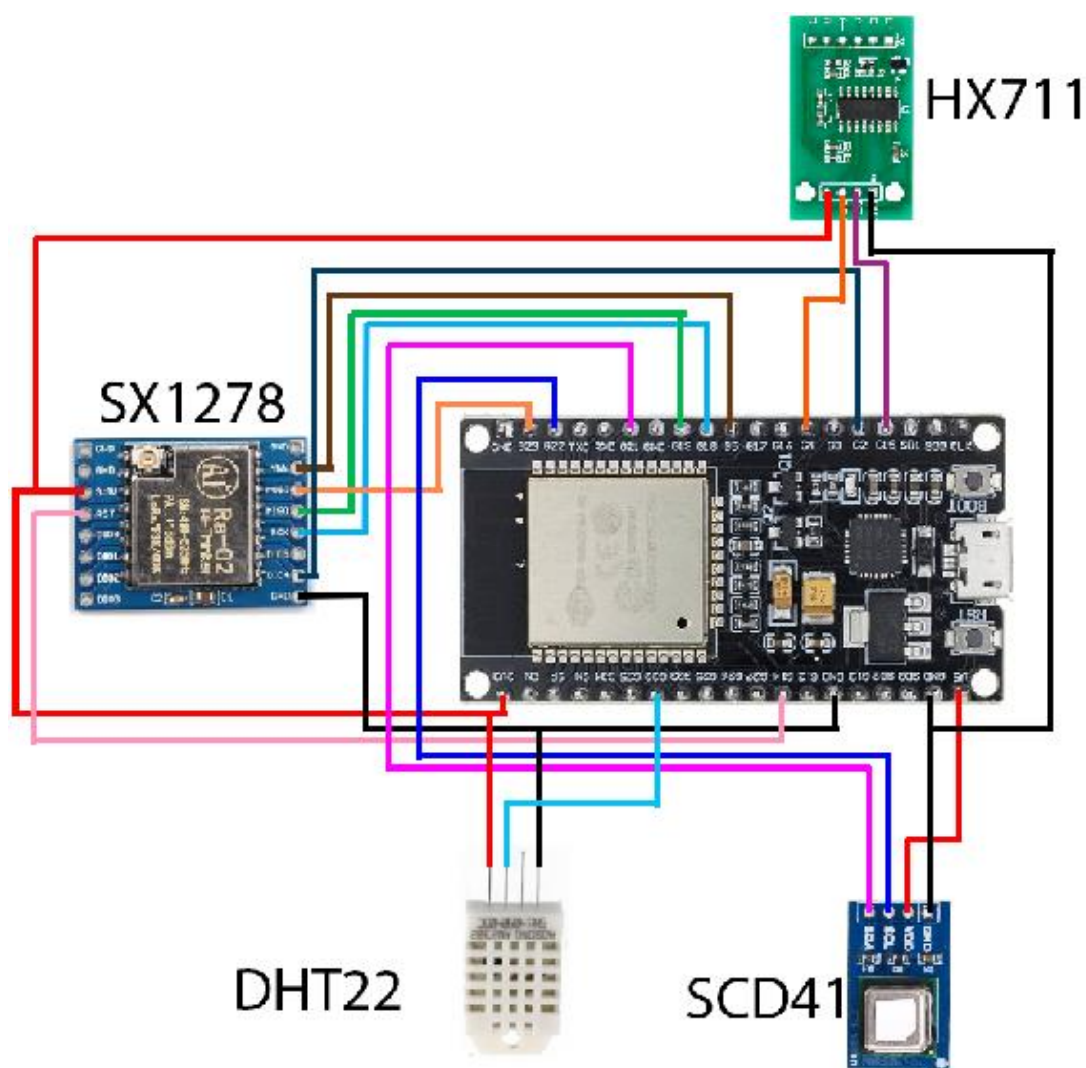
4.2.4 Snímání hodnot uvnitř úlu

Pro sběr hodnot uvnitř včelího úlu bylo vybráno čidlo SDA41 na měření teploty, vlhkosti a množství CO₂ uvnitř plodiště a čidlo DHT22 které snímá teplotu a vlhkost uvnitř medníku. Senzor SDA41 má mnohem menší spotřebu energie než senzory technologie NDIR, které potřebují 3 minuty na zahřátí, což není pro systém, který využívá baterii jako zdroj napájení ideální. DHT22 byl vybrán především z důvodu nízké pořizovací ceny a relativně velké přesnosti. Do ESP je taktéž napojen vysílač SX1278 pro

možnost komunikace pomocí LoRa, dále je zapojen HX711 pro měření váhy pomocí tenzometrických senzorů

ESP32	SX1278	SDA41	HX711	DHT11
3,3 V	VCC		VCC	VCC
GND	GND	GND	GND	GND
GPIO 18	SCK			
GPIO 19	MISO			
GPIO 23	MOSI			
GPIO 5	SS			
GPIO 14	RST			
GPIO 2	DIO			
VCC		VCC		

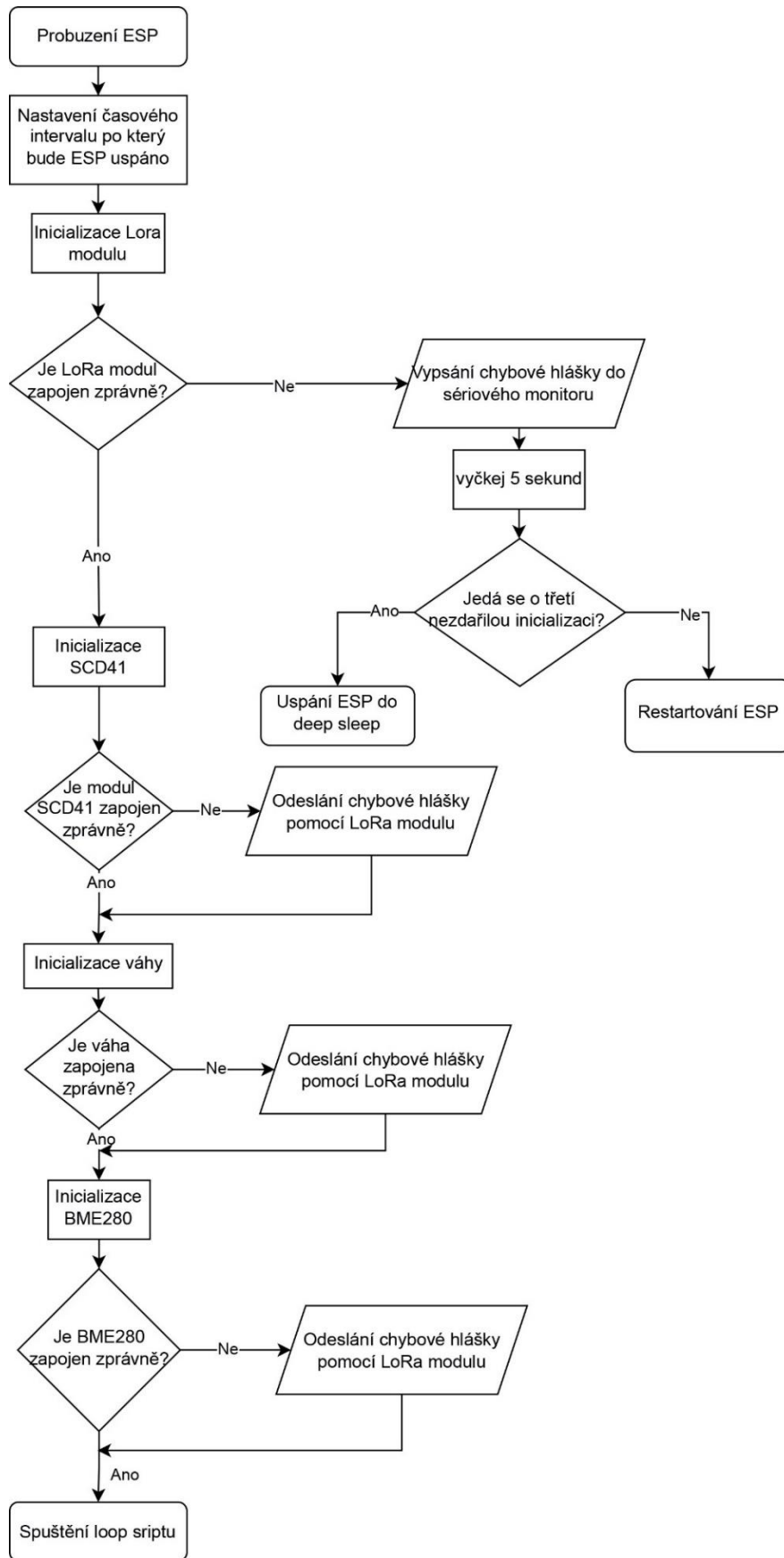
Tabulka 2 zapojení pinů jednotlivých senzorů a modulů (vlastní)



Obrázek 18 Schéma zapojení úložného ESP32 (vlastní)

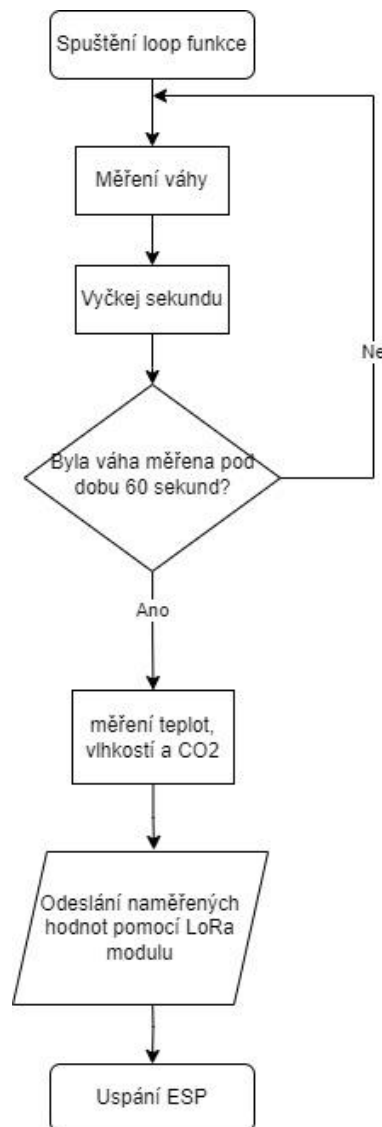
Programy psané v prostředí Arduino ide obsahují dvě hlavní funkce. Setup() která je provedena jednorázově při spuštění mikrokontroleru a funkci loop() která je vykonávána v cyklu.

Uvnitř funkce setup() je nejdříve provedeno nastavení délky uspání ESP a následně jsou inicializovány jednotlivé moduly. Jako první proběhne inicializace LoRa modulu, v případě neúspěšné inicializace, je vypsána chybová hláška do sériového monitoru, a po pětivteřinové pauze je mikrokontroler restartován, Pokud nastane situace, kdy je ESP restartováno třikrát po sobě, je mikrokontroler uveden do deep sleep režimu na následujících 30 minut. V případě neúspěšné inicializace senzorů jsou veškeré chybové hlášky odesílány pomocí LoRa modulu a průběh programu není přerušen.



Obrázek 19 Vývojový diagram `setup()` úložného ESP32 (vlastní)

Po úspěšném dokončení funkce setup() je spuštěn cyklus loop() ve kterém jsou prováděna jednotlivá měření. Úlová váha potřebuje nějaký čas na ustálení měření po probuzení mikrokontroleru, přesněji provedení páru desítek měření, než začne vykazovat správné hodnoty. Z toho důvodu je jako první spuštěn cyklus for, ve kterém je váha měřena šedesátkrát s vteřinovými rozestupy. Po ukončení cyklu jsou změřeny hodnoty uvnitř včelího úlu, které jsou následně odeslány pomocí SX1278 LoRa vysílače. Po odeslání naměřených hodnot je mikrokontroler uveden do deep sleep stavu, ze kterého bude probuzen za 30 minut.



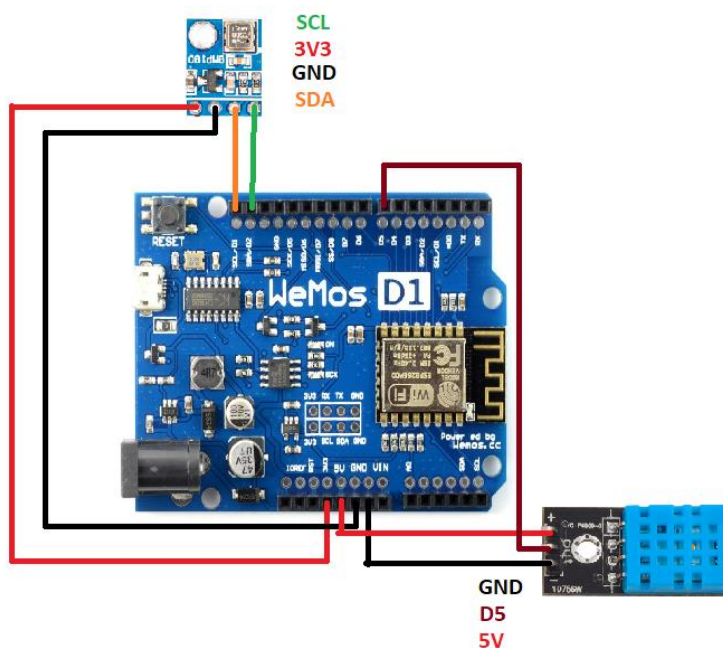
Obrázek 20 Vývojový diagram loop() úlového ESP32 (vlastní)

4.2.5 Meteostanice

Jako mikrokontroler bylo vybráno opět ESP32, pokud by bylo zapotřebí větší snižování nákladů postačil by starší model ESP8266. Meteostanice snímá venkovní teplotu, vlhkost a tlak.

Pro měření tlaku bylo vybráno čidlo BMP180, které je schopné snímat i teplotu, nikoli však vlhkost, z toho důvodu byla stanice obohacena o další senzor. Přestože DHT11 není jeden z nejlepších enviromentálních senzorů, jeho 5% přesnost pro měření relativní vlhkosti je pro orientační data naměřené pomocí meteostanice dostatečná.

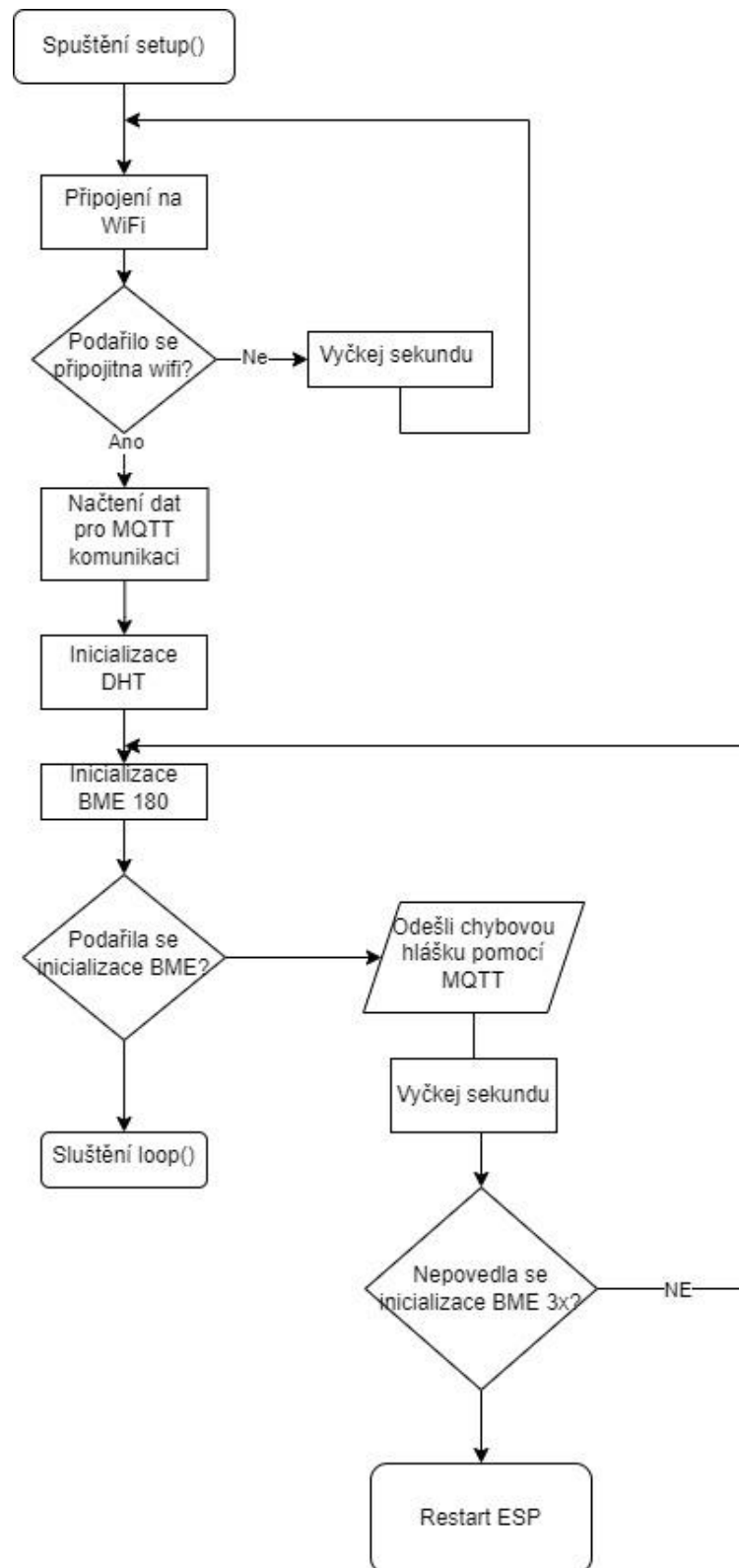
Napájení je zajištěno přímo z elektrické sítě pomocí 10V napájecího zdroje, z toho důvodu není nutné klást velký důraz na spotřebu energie jako je tomu u mikrokontroleru nasazeného u včelího úlu a nebyla nutná konfigurace uspávání, tudíž je ESP permanentně zapnuté a připojené do lokální sítě pomocí WiFi.



Obrázek 21 Schéma zapojení ESP32 meteostanice (vlastní)

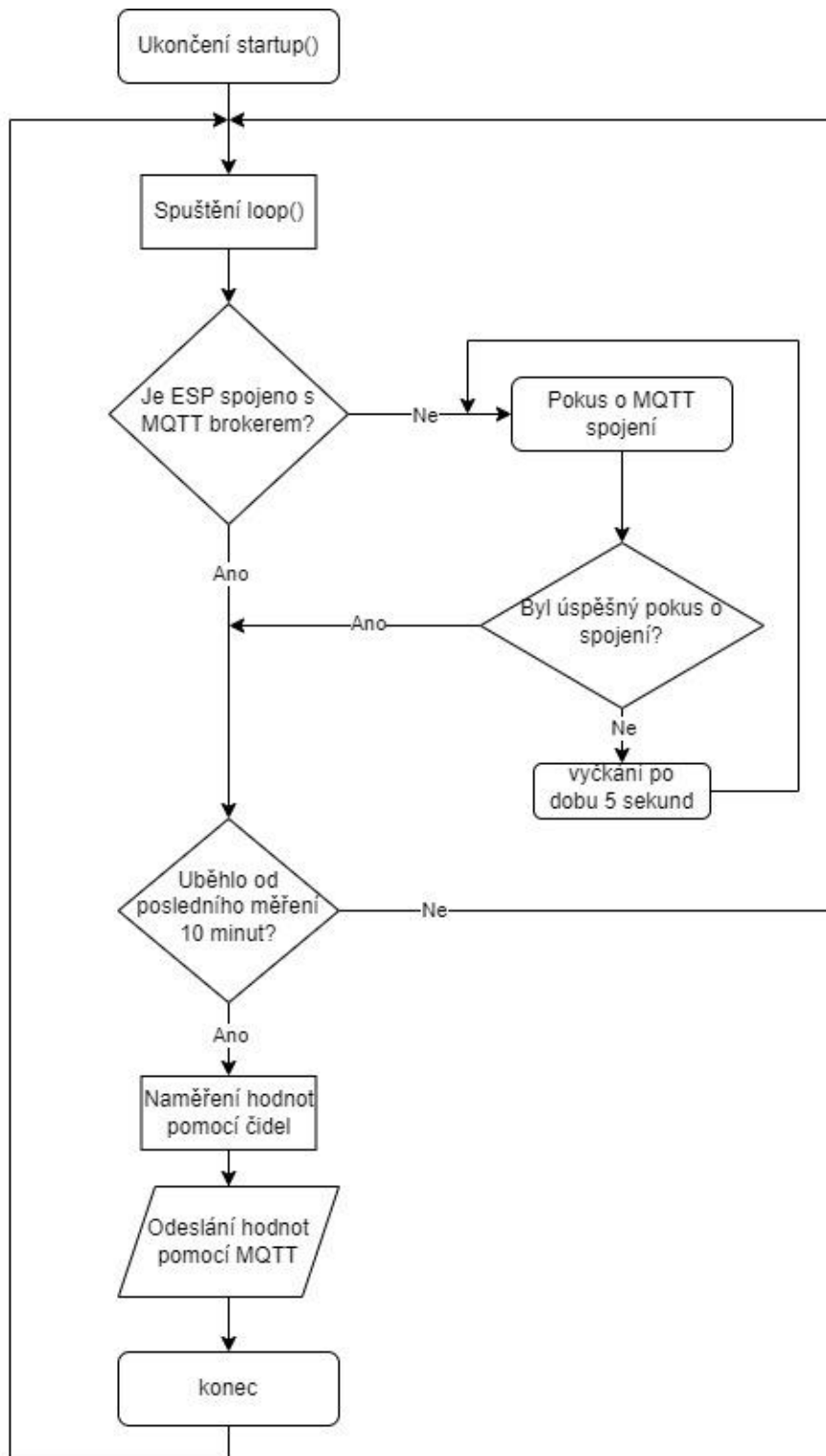
Data jsou snímána každých 10 minut, následně je z naměřených hodnot sestaven Json dokument, který je následně odeslán pomocí MQTT prostřednictvím WiFi.

Při zapnutí mikrokontroleru je spuštěna sekvence `setup()`, při které se ESP nejdříve pokusí připojit na WiFi síť, po úspěšném připojení je načtena IP adresa a port na kterém je provozován MQTT broker, následně je inicializováno čidlo DHT a BMP180. V případě, že se inicializace čidla BMP tři krát po sobě nezdařila, je ESP restartováno a celá sekvence začne znovu. Pokud proběhne inicializace v pořádku je spuštěn cyklus `loop()`.



Obrázek 22 Vývojový diagram setup() meteostanice ESP32 (vlastní)

Uvnitř cyklu loop() je v první řadě zkontrolováno, zda je ESP spojeno s MQTT brokerem

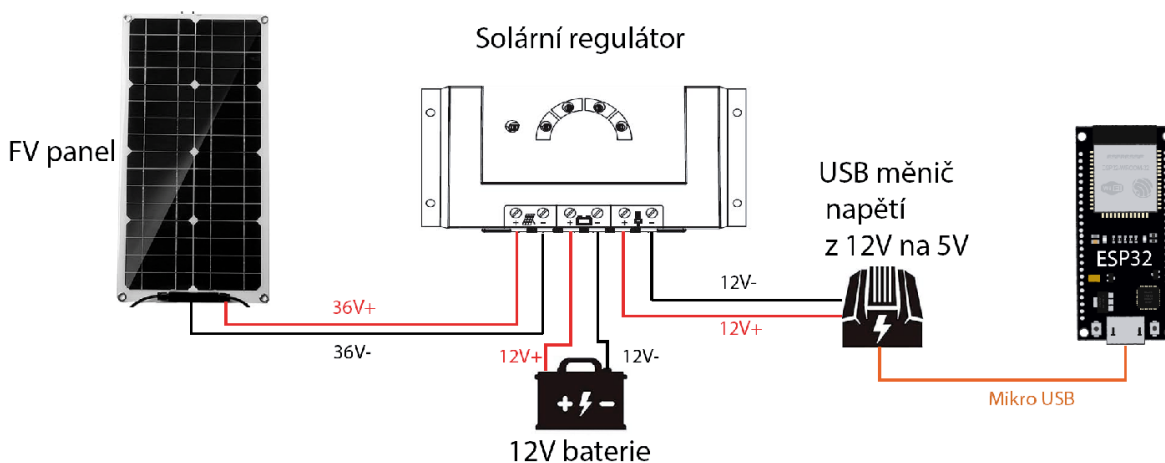


Obrázek 23 Vývojový diagram loop() meteostanice ESP32 (vlastní)

4.2.6 Napájení

Včelí úly jsou ve většině případů umístěny v místech, bez možnosti využití eklektické sítě. Je nutné tedy využít nějakou z forem mobilní energie.

V případě využití solární energie je nutné si vypočítat, jak velký solární panel bude zapotřebí a jak velká baterie bude zapotřebí pro napájení při absenci slunečního svitu především během zimních měsíců, kdy je možné že FV panel bude pod vrstvou sněhu a nebude možné dobíjet baterii. Taktéž je nutné při výběru baterie počítat s tím, že není možné využít její celkovou kapacitu, jinak by bylo riskováno její poškození. Standardně je stav baterie monitorován solárním regulátorem, který při většinovém vyčerpání kapacity akumulátor odpojí. Hranice ochrany proti podbití akumulátoru se liší podle výrobce, většinou se však pohybuje mezi 20% až 30% kapacity baterie.



Obrázek 24 Schéma elektrického zapojení (vlastní)

V tabulce 3 je vypočítána teoretická spotřeba částí systému, která ke svému napájení využívá baterii. Výsledná hodnota musí být brána pouze jako orientační, nejsou v ní totiž zahrnuty ztráty, které nastanou při změně napětí z 12V na 5V. Z toho důvodu, ačkoli v tabulce je výsledná spotřeba přibližně 1 Wh, reálná se pravděpodobně bude blížit ke 2 Wh.

Prvek	Proud	Jednotky	Poznámka	Délka úkonu v hod	Napětí	Příkon	Délka úkonu v sec	Počet sputění za den	Čas provozu za den v sec	Spotřeba
esp32	0,24	A		0,02	5	1,2	60	48	2880	0,96
	0,00015	A	deep sleep	0,50	5	0,00075	1800	48	86400	0,018
CDA41	0,0004	A		0,00	3,3	0,00132	1	48	48	1,76E-05
sx1278	0,093	A	odesílání	0,00	3,3	0,3069	2	48	96	0,008184
	0,0016	A	připraveno	0,02	3,3	0,00528	60	48	2880	0,004224
HX711	0,0015	A		0,02	3,3	0,00495	60	48	2880	0,00396
BME280	0,000281	A		0,00	3,3	0,000927	1	48	48	1,24E-05
										0,994398 Wh

Tabulka 3 Teoretické spotřeby systému (vlastní)

4.3 Implementace

V části implementace je popsán proces přípravy a nasazení jednotlivých částí systému. Od kalibrace čidel po nasazení do včelího úlu.

4.3.1 Senzorická vrstva

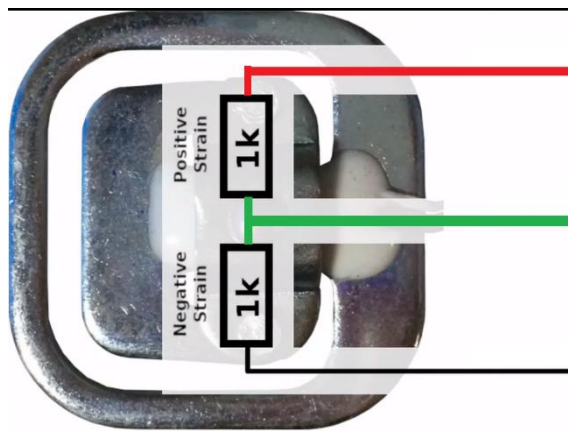
V části senzorické vrstvy je práce zaměřena především na architekturu fyzických zařízení, která jsou využívána k akvizici dat. Tato zařízení se skládají z funkčních modulů, zahrnujících modul pro sběr dat s využitím kalibrovaných čidel, dále z kontroléru pro řízení dalších připojených zařízení (akčních členů) a z komunikační jednotek, které slouží pro přenos a příjem zejména dat. Veškeré moduly jsou řízeny specializovanou řídicí jednotkou. Kromě toho je klíčovým prvkem senzorické vrstvy i fyzická bezpečnost tohoto zařízení a zajištění adekvátního zdroje elektrické energie pro jeho provoz

4.3.1.1 Úlová váha

Jako první byla vyrobena dřevěná konstrukce podle velikosti dna úlu, na které jsou umístěny váhové senzory. Senzory nemají žádný úchytný systém a nebylo možné je pouze přichytit ke konstrukci pomocí lepidla nebo vrutu. Při pohledu na čílo je zřejmě viditelná vnější a vnitřní část obdélníkové části senzoru. Zatímco vnější část se má fixně opírat o konstrukci na které je čidlo připevněno, vnitřní část musí mít pod sebou prostor, aby měla pod zátěží možnost se prohnout. Na internetu je možné najít modely pro 3D tisk do kterých se dá čidlo následně uložit. Jelikož během vypracovávání této práce nebylo možné využití 3D tiskárny, bylo nutné do dřevěné konstrukce vyfrézovat otvory do kterých se vnější části čidel mohou pohybovat. Senzory byly následně umístěny na nově vyfrézované otvory na rozích dřevěné konstrukce a přichyceny vruty. Jednotlivá čidla mají měřící rozsah až 50 kg, dalším krokem tedy bylo zapojení všech čtyř tenzometrických senzorů do Wheatstoneova můstku, aby bylo docíleno posunu rozsahu na 200 kg.

Ze senzorů jsou vyvedeny tři vodiče, barevné označení vodičů se může u různých výrobců lišit, z toho důvodu bylo nutné v první řadě otestovat jaký je odpor mezi vodiči. V případě této práce jsou vodiče označeny červenou, zelenou a bílou barvou. Mezi zeleným a červeným drátem byl naměřený odpor 1 Ohm, mezi červeným a bílým 2 Ohmy a mezi bílým a zeleným 1 Ohm, z čehož vyplývá že zelený vodič je využit pro zapojení do HX711

a černý a červený vodič pro propojení s ostatními sensory. Pro co nejpřesnější měření je zapotřebí, aby všechny vodiče měly stejnou délku

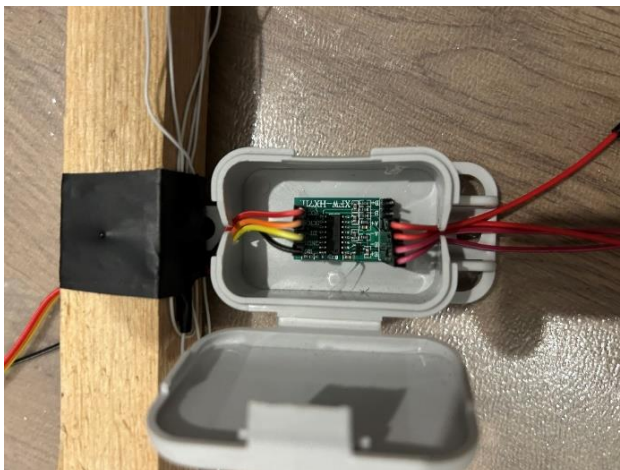


Obrázek 25 Přehled vodičů tenzometrického senzoru
(vlastní)

Váha objektů je měřena pouze změnou odporu, z toho důvodu bylo nutné provést kalibraci pomocí referenční hodnoty. Ta byla provedena pomocí scriptu Calibration obsaženého v příkladech použité knihovny HX711_ADC.h potřebné pro komunikaci s převodníkovým modulem HX711. Kalibrace probíhala následujícím způsobem. Po připojení váhy na ESP32 byl nahrán kalibrační scrip, ten nejdříve zkontroloval zapojení na správné piny, následně byly do sériového monitoru vypsány instrukce na položení váhy do vodorovné polohy na rovný povrch a odstranění veškeré zátěže. Po zadání pokynu ke kalibraci script vyzval k položení referenční hmotnosti na váhu a zadání hmotnosti pomocí sériového monitoru, následně byla váha zkalibrována a kalibrační hodnota byla vypsána na obrazovce a byl položen dotaz, zda hodnotu zapsat do flash paměti ESP32. Jelikož knihovna HX711_ADC.h byla vydána v roce 2017, využívá pro ukládání knihovnu EEPROM, ta je však v dnešní době zastaralá a byla nahrazena uživatelsky přívětivější knihovnou Preferences.h která je automaticky zahrnuta při instalaci ESP32 desek v Arduino IDE. Scrip byl tedy upraven, aby využíval knihovnu Preferences na místo EEPROM.

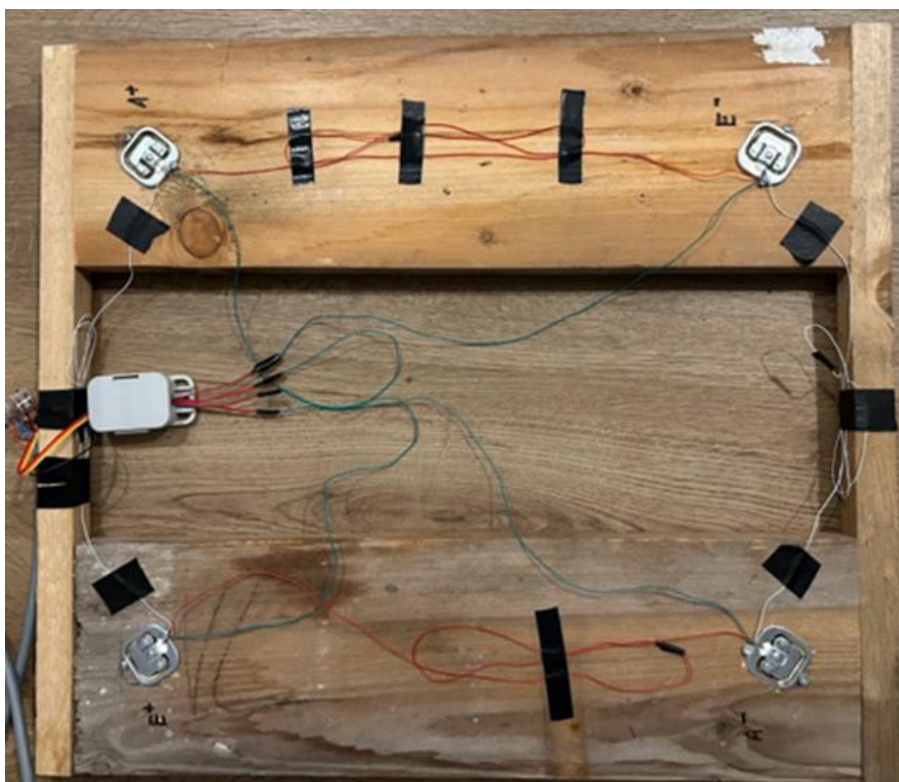
První kalibrace byly prováděny za pomoci nižších hmotností jako 3 až 7 kg, ukázalo se však, že při měření vyšších hmotností vzniká velká odchylka mezi reálnou a naměřenou hodnotou. Například při zátěži přibližně 80 kg ukazovala váha v průměru o 14 kg méně. Byla tedy zvolena druhá množnost, a to provést kalibraci pomocí větší váhy, která vykazovala mnohem lepší výsledky. Kalibrace byla provedena pomocí 75 kilogramů a odchylka od reálných hodnot při testování u nižší i vyšší hmotnosti byla do 40 gramů.

Mimo kalibračního scriptu bylo nutné využít `Persistent_zero_offset` opět obsaženého v knihovně `HX711_ADC.h` pro výpočet vyrovnávací hodnoty, díky které je hmotnost úlové váhy brána jako nulová hodnota.



Obrázek 26 HX711 usazený na váze (vlastní)

Standartně tento výpočet dělá váha sama při inicializaci, z čehož vyplývá problém, pokud je váha pod konstantní zátěží. V praxi by takový případ znamenal, že při každém probuzení nebo zapnutí by si úlová váha myslela, že hmotnost celého úlu je hmotností samotné úlové váhy a veškeré hmotností měření by byli rovny nule.



Obrázek 27 Úlová váha (vlastní)

4.3.1.2 Úlové senzory

Napájení senzoru SCD41 pro měření CO₂ bylo nutné napojit na pin VIN na kterém je napětí 5V. Přesto že samostatné čidlo je podle výrobce množné napájet 2,4-5,5V, tak desky na kterých je senzor napájen mohou obsahovat rezistory díky kterým je nemožné ho provozovat na standartním napětí desek ESP 3,3V.

Senzoru SCD41 disponuje auto kalibračním algoritmem, který bere nejnižší hodnotu naměřenou za poslední týden a usoudí že se jedná o hodnotu 400ppm. Je však zásadní si uvědomit, že tato automatická kalibrace může vést k nepřesným měřením, pokud je senzor umístěn v prostředí, které nedosahuje měření 400 ppm CO₂ v posledních sedmi dnech, což je problém pro senzor, který je permanentně usazen uvnitř úlu. Pro řešení tohoto problému je doporučováno trvale vypnout režim automatické kalibrace. Vypnutí proběhlo jednorázovým spuštěním skriptu vyobrazeném na obrázku 28.

```
#include "scd4x.h"

SCD4X co2;

void setup() {
  // Inicializace SCD4x knihovny
  co2.begin();

  // vypnutí auto-kalibrace
  co2.setCalibrationMode(false);

  // uložení do falsh pameti
  co2.saveSettings();
}

void loop() {
}
```

Obrázek 28 Skript pro vypnutí automatické kalibrace (vlastní)

Použité knihovny

- HX711_ADC.h – Knihovna pro HX711 AD převodník
- Preferences.h – Knihovna pro práci s flash pamětí
- Wire.h – Knihovna pro komunikaci pomocí I2C
- scd4x.h – Knihovna pro senzor SCD41
- LoRa.h – Knihovna pro odesílání a příjem dat pomocí LoRA vln
- SPI.h – Knihovna pro komunikaci pomocí SPI

- DHT.h – Knihovna pro senzor DHT22

Po zapnutí, nebo probuzení ESP je jako první spuštěna funkce `setup()`. Prvním důležitým krokem je zapnutí funkcionality `deep sleep` a nastavení způsobu, kterým bude ESP probuzeno. Byla zvolen způsob probuzení pomocí časovače dobu „`esp_sleep_enable_timer_wakeup`“, která má jako vstupní hodnotu čas, za který bude ESP probuzeno v mikrosekundách. Aby bylo nastavení srozumitelnější, byly mikrosekundy převedeny na sekundy. Samotné definování délky spánku a převodový faktor byl nastaven na počátku Arduino programu pomocí „`#define uS_TO_S_FACTOR 1000000ULL`“ a „`#define TIME_TO_SLEEP 600`“

Následně je inicializován LoRa modul. V případě nezdařilé inicializace je vypsána chybová hláška do sériového monitoru a je vyčkáno 5 sekund, než je mikrokontroler restartován. Pokud se nezdaří ani třetí pokus o inicializaci je ESP usnáno na dobu 30 minut. Aby bylo možné uchovávat počet nezdařilých inicializací bylo nutné hodnotu proměnné `prestart` uchovávat v paměti RTC čehož bylo docíleno deklarací „`RTC_DATA_ATTR int prestart = 0;`“

Po inicializaci je zadáno synchronizační slovo, které je stejné jako na LoRa bráně, aby bylo docíleno úspěšné komunikace mezi bránou a ESP.

Následně proběhne inicializace senzoru SCD41, V případě, že se nezdaří je vygenerována chybová hláška, která je uložena v proměnné `msg` která je následně odeslána pomocí vytvořené funkce na odesílání zpráv pomocí LoRaWAN.

```
// Funkce pro odeslání dat pomocí LoRa modulu
void odeslaniLoRa(String hodnoty) {
    LoRa.beginPacket ();
    LoRa.print (hodnoty);
    LoRa.endPacket ();
    Serial.println (hodnoty);
    delay (2000);
}
```

Obrázek 29 Funkce pro odeslání zprávy pomocí LoRa modulu
(vlastní)

Po inicializaci začne senzor SCD41 měření.

```

void setup() {
  Serial.begin(115200); delay(5000);
  esp_sleep_enable_timer_wakeup(TIME_TO_SLEEP * uS_TO_S_FACTOR); // nastavení délky času usnutí ESP
  Serial.println("ESP32 bude usnuto na " + String(TIME_TO_SLEEP) + " sekund");

  //Statup pro LoRa modul
  SPI.begin(SCK, MISO, MOSI, SS);
  LoRa.setPins(SS, RST, DI0);
  if (!LoRa.begin(BAND)) {
    Serial.println("Start LoRa se nezdařil!");
    delay(5000);
    ++prestart;
    if (prestart == 3) {
      prestart = 0; // vynulování hodnoty, pro další pokusy po probuzení
      esp_deep_sleep_start(); // usnutí po deseti neúspěšných inicializacích
    }
    else {
      ESP.restart(); // restartování ESP
    }
  }
  prestart = 0; // vynulování hodnoty, pokud je modul inicializován
  LoRa.setSyncWord(0x12); // synchronizační slovo
  delay(1500);
  Serial.println();

  //Statup pro SCD41
  Wire.begin(); // Inicializace the I2C komunikace
  co2.begin(); // Inicializace the SCD4x knihovny
  // kontrola zapojení CO2 senzoru
  if (!co2.isConnected()) {
    while (true) {
      msg = "{\"typ\":\"ch\",\"id\":\"\" + NodeId + "\",\"v\":\"SCD setup f\"}"; //Vygenerování chybové hlášky
      odeslaniLoRa(msg); // Odeslání chybové hlášky pomocí lora
    }
  }
  co2.startPeriodicMeasurement(); // Start periodických měření
  delay(10000); // Vyčkání na zahřátí senzoru

```

Obrázek 30 Inicializace LoRa modulu a SCD41 úlové ESP32 (vlastní)

Následně je spuštěna inicializace úlové váhy. Nejdříve je otevřeno úložiště s názvem kalibrace, do kterého během přípravy váhy byla uložena kalibrační hodnota. Úložiště je otevřeno režimu pouze pro čtení. Poté je kalibrační hodnota z úložiště načtena a je zadán stabilizační čas pro ustálení váhy při spuštění. Dále je nastaven tare_offset kterým se určuje hmotnost samotné úlové váhy. V případě neúspěšné inicializace je opět odeslána chybová hláška pomocí LoRa modulu.

Po inicializaci váhy je inicializováno DHT22.


```

//Startup pro váhu
preferences.begin("kalibrační", true); // otevření úložiště v paměti flash
LoadCell.begin();
float calibrationValue; // kalibrační proměnná
calibrationValue = preferences.getFloat("KalHodnota", -26213.60); // načtení kalibrační hodnoty z flash paměti
unsigned long stabilizingtime = 20000; // stabilizační čas pro spřesnění měření
long tare_offset = 8357017; //offset pro váhu, bez kterého by se váha vynulovala během každého restartu
LoadCell.setTareOffset(tare_offset);
boolean _tare = false; // musí být false, jinak by se váha při startu vynulovala
LoadCell.start(stabilizingtime, _tare);
if (LoadCell.getTareTimeoutFlag()) {
    msg = "{\"typ\":\"ch\",\"id\":\"\" + NodeId + "\",\"v\":\"Vaha setup f\"}"; //Vygenerování chybové hlášky
    odeslaniLoRa(msg); // Odeslání chybové hlášky pomocí lora
    Serial.println("Vaha v haji!");
}
else {
    LoadCell.setCalFactor(calibrationValue); // Nastavení kalibrační hodnoty

Serial.println("Funguje start up vahy");
}
preferences.end();

// Sartup pro DHT22
dht.begin();

```

Obrázek 31 Inicializace úlové váhy a DHT22 úlové ESP32 (vlastní)

Po ukončení funkce setup() je spuštěn na loop(). Úlová váha potřebuje provést několik měření, než začne ukazovat správné hodnoty, z toho důvodu je nejdříve spuštěn cyklus for, který se spustí šedesátkrát a trvá přibližně minutu, ve kterém je prováděno měření váhy úlu.

```

void loop() {
/*
-----
Loop pro váhu
-----
*/

for (int i = 0; i < 60; i++)
{
static boolean newDataReady = 0;
const int serialPrintInterval = 5; //interval pro výpis na sériový monitor
now = millis();
if (LoadCell.update()) newDataReady = true;
if (newDataReady) {
if (millis() > τ + serialPrintInterval) {
float i = LoadCell.getData();
Serial.print("Váha: ");
Serial.println(i);
vahafinal = i;
newDataReady = 0;
τ = millis();
delay(1000);
}
}
}
/*
-----
*/
}

```

Obrázek 32 Cyklus pro měření úlové váhy (vlastní)

Po změření váhy je provedeno měření podmínek, které panují uvnitř úlu. Pokud nastane problém při měření podmínek uvnitř plodiště, je opět poslána chybová hláška prostřednictvím LoRaWAN.

```

-----
Výčítání hodnot senzorů uvnitř úlu.
-----
*/

double co2Value, temperature, humidity, tm, vm;

// Vytěžení hodnot z SCD41
uint8_t errorCode = co2.readMeasurement(co2Value, temperature, humidity);

// Kontrola chybových hlášení SCD41
if (errorCode == 0) {
    // Formátování naměřených hodnot z SCD41 do stringu
    char measurementString[100];
    sprintf(measurementString, "CO2: %.0f ppm, Teplota: %.1f °C, Vlhkost: %.0f %%RH", co2Value, temperature, humidity);
    // Vypsání naměřených hodnot do sériového monitoru
    Serial.println(measurementString);
} else {
    // Konvertování chyby do textu
    const char* errorText = co2.getErrorText(errorCode);
    msg = "{\"typ\":\"ch\",\"id\":\"\" + NodeId + "\",\"v\":\"\" + errorText + "}"; //Vygenerování chybové hlášky
    odeslaniLoRa(msg); // Odeslání chybové hlášky pomocí lora
}
tm = dht.readTemperature(); // načtení teploty v medniku pomocí DHT22
vm = dht.readHumidity(); // načtení vlhkosti v medniku pomocí DHT22

```

Obrázek 33 Měření hodnot uvnitř včelího úlu (vlastní)

Následně jsou všechny naměřené hodnoty jednotlivě odesílány. Po odeslání hodnot je ESP uvedeno do režimu deep sleep na 30 minut.

```

-----
odeslání hodnot pomocí LoRa modulu a uspání ESP
-----
*/

// Odesílání teploty plodiště
msg = "{\"typ\":\"tp\",\"id\":\"\" + NodeId + "\",\"v\":\"\" + String(temperature) + "}";
odeslaniLoRa(msg);

// Odesílání vlhkosti plodiště
msg = "{\"typ\":\"vp\",\"id\":\"\" + NodeId + "\",\"v\":\"\" + String(humidity) + "}";
odeslaniLoRa(msg);

// Odesílání co2
msg = "{\"typ\":\"co\",\"id\":\"\" + NodeId + "\",\"v\":\"\" + String(co2Value) + "}";
odeslaniLoRa(msg);

// Odesílání váhy
msg = "{\"typ\":\"va\",\"id\":\"\" + NodeId + "\",\"v\":\"\" + String(vahafinal) + "}";
odeslaniLoRa(msg);

// Odesílání vlhkosti medniku
msg = "{\"typ\":\"vm\",\"id\":\"\" + NodeId + "\",\"v\":\"\" + String(vm) + "}";
odeslaniLoRa(msg);

// Odesílání teploty medniku
msg = "{\"typ\":\"tm\",\"id\":\"\" + NodeId + "\",\"v\":\"\" + String(tm) + "}";
odeslaniLoRa(msg);

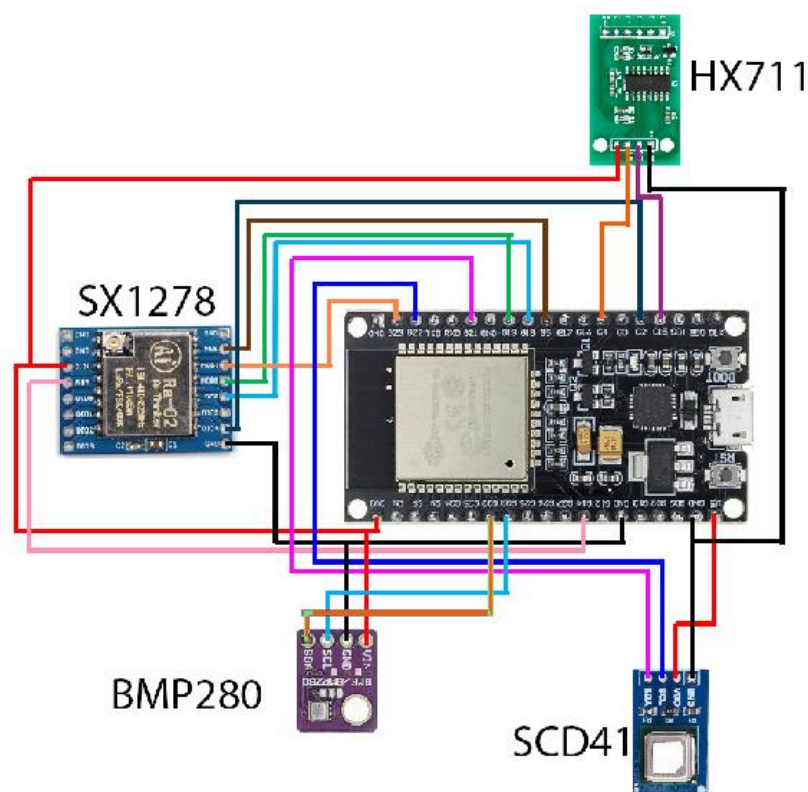
Serial.println("Dobrou noc");
Serial.flush();
esp_deep_sleep_start(); // uspání ESP do deep sleep

```

Obrázek 34 Odeslání naměřených hodnot a uspání úlového ESP32 (vlastní)

Úlové ESP32 bylo před nasazením do včelího úlu otestováno v domácích podmínkách. Na váze bylo položené úlové dno, na kterém se nacházel jeden nástavek s víkem. Uvnitř nástavku se nacházelo ESP32 se zapojenými senzory.

Během testování ve vnitřních prostorech bylo využito pro monitorování teploty a vlhkosti uvnitř mendíku čidlo DHT22 z původního návrhu, při prvních výsledcích však bylo vyzorováno že DHT22 není ideální volbou pro nasazení. Přesto že výrobcem uvedená přesnost čidla činí přibližně 0.5 °C, hodnoty naměřené během testování měly odchylku od reálných hodnot až 6 °C. Stejný scénář nastal i při měření vlhkosti kdy odchylka dosahovala až 55 %. Za nepřesností naměřených hodnot stojí pravděpodobně stáří senzoru nebo jeho uložení v neideálních podmínkách. Z toho důvodu bylo provedeno rozhodnutí místo DHT22 využít mnohem kvalitnější senzor BME280, ten stejně jako senzor SCD41 využívá pro přenos dat sběrnici I2C. Změněné schéma zapojení je zobrazeno v obrázku 35.



Obrázek 35 Upravené schéma úlového ESP32 pro využití BME280 (vlastní)

Bylo nutné knihovnu DHT.h nahradit knihovnami Adafruit_Sensor.h a Adafruit_BME280.h

Aby se předešlo problému, kdy by například v budoucnu došlo k výměně senzorů které by vedlo ke konfliktům adres I2C. Jednoduchým řešením je využití předdefinovaných objektů „Wire()“ a „Wire1()“. Metoda „Wire().begin()“ vytváří komunikaci I2C na první sběrnici I2C pomocí výchozích pinů a výchozí frekvence. Pro Wire1.begin() je nutné předat požadované piny pro SDA a SCL. Využit je možné všechny GPIO piny až na piny 34, 35, 36 a 39, které jsou pouze vstupními porty a nelze je použít jako SCL nebo SDA.

Pro zapojení byly využity piny 32 a 33 které byly definovány na počátku Arduino programu pomocí:

- #define SDA_2 33
- #define SCL_2 32

Definování nových pinů nahradilo původní určení typu DHT a definování pinu pro DHT které v původním programu vypadalo následovně.

- #define DHTTYPE DHT22
- const int DHTPin = 5

Ve funkci setup() byla původní inicializace DHT nahrazena kódem zobrazeným v obrázku 36.

```
Wire1.begin(SDA_2, SCL_2);
bool status;
status = bme.begin(0x77, &Wire1);
if (!status) {
    msg = "{\"typ\":\"ch\",\"id\":\"\" + NodeId + "\",\"v\":\"BME280 setup f\"}"; //Vygenerování chybové hlášky
    odeslaniLoRa(msg); // Odeslání chybové hlášky pomocí lora
}
```

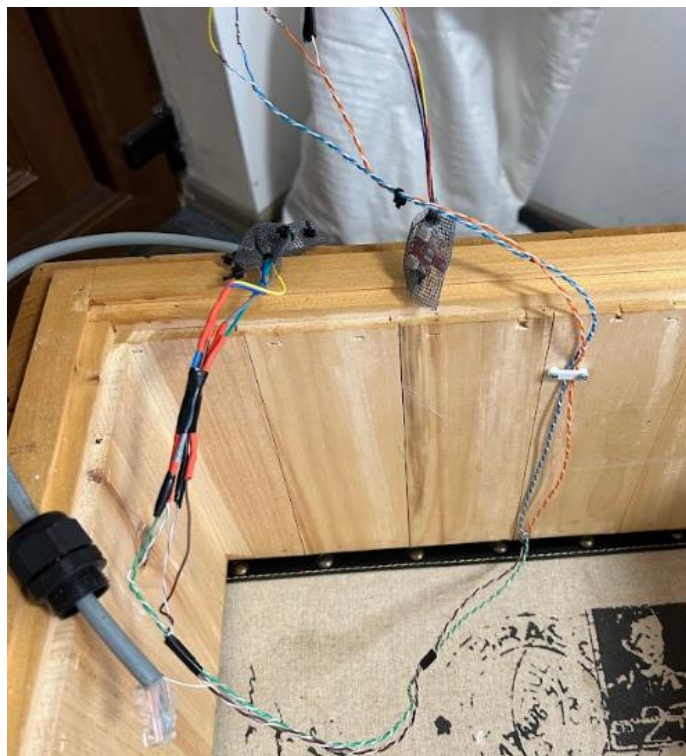
Obrázek 36 Inicializace BME280 úlového ESP32 (vlastní)

V části loop() bylo při vyčítání hodnot z medníku pouze změněno „dht.“ Na „bme.“. Zbytek programu zůstal nepozměněn.

Důležitým faktorem je poloha senzorů uvnitř úlu. Pro co nepřesnější měření, je ideální, aby byla čidla umístěna co nejbližší středu úlu. Z toho důvodu by bylo umístít SCD41 na mateří mřížku, která je uložena na plodišti, BME280 je umístěn pod víkem, aby nebylo bráněno přístupu včelám k rámečkům.

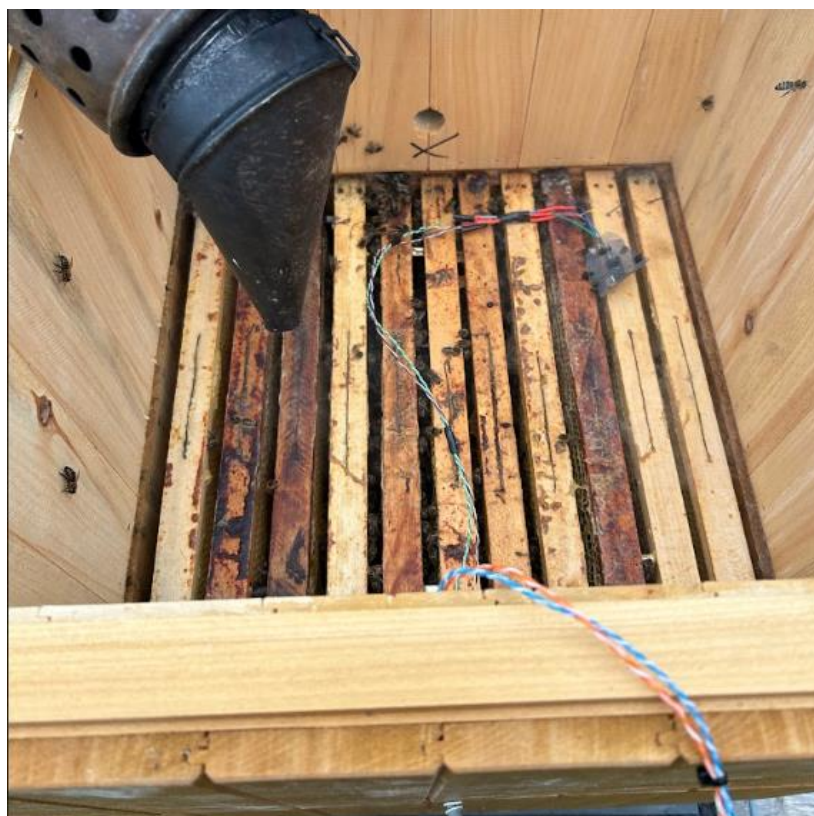
V době testování včelího úlu, nebyl k dispozici žádný úl, který by měl medník. Byly však úly, které měly dvě plodiště, přičemž jediným rozdílem oproti standardnímu úlu s medníkem bylo, to že mezi plodišti nebyla umístěna mateřská mřížka. Systém byl tedy nasazen do úlu s dvěma plodišti kdy pro testovací účely substituuje medník vrchní plodiště. Před nasazením byl připraven jeden nástavek, do spodní části nástavku byla vyvrtána díra, kterou byl protažen UTP kabel, který byl zvolen pro jednodušší zapojení. Z části UTP kabelu, která se nachází uvnitř úlu byla odstraněna vrchní izolace a vnitřní vodiče byly rozděleny do dvou čtveřic, kdy každá čtveřice slouží k připojení jednoho senzoru. Aby bylo možné oddělení nástavku, kterým je UTP vedeno od zbytku úlu byl na konci kapelu přidán konektor RJ45, díky kterému možné nástavek jednoduše připojit a odpojit.

Upravený nástavek byl zaměněn za vrchní plodiště vybraného úlu.



Obrázek 37 UTP kabel s napojenými senzory uvnitř upraveného nástavku (vlastní)

Senzor SDA41 byl umístěn mezi nástavky do míst, kde by se standartně nacházela mateřská mřížka.



Obrázek 38 SCD41 mezi nástavky uvnitř úlu (vlastní)

Senzor BME280 je umístěn mezi rámkou vrchního nástavku a víka. Všechna čidla jsou chráněna pomocí plastové mřížoviny, která zabraňuje znečištění od včel.



Obrázek 39 BME280 uvnitř včelího úlu (vlastní)

ESP je umístěno uvnitř IP 65 ochranné krabičky, aby bylo uchráněno před venkovními podmínkami.



Obrázek 40 Úl s nasazeným systémem (vlastní)

4.3.1.3 Meteostanice

Použité knihovny

- Adafruit_BMP085.h
- ArduinoJson.h
- PubSubClient.h
- DHT.h
- WiFi.h

Po spuštění ESP nejdříve provedena funkce na připojení WiFi, která probíhá, dokud se nepřipojí do lokální sítě.

```
// Funkce pro připojení na wifi
void setup_wifi() {
  delay(10);
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.print(".");
  }
  Serial.println("");
  Serial.print("WiFi připojena - NodeMCU IP addressa: ");
  Serial.println(WiFi.localIP());
}
```

Obrázek 41 Funkce pro připojení na WiFi meteostanice
ESP32 (vlastní)

Následně je natavena adresa a port MQTT brokera, po kterém je provedená inicializace senzoru DHT22 a BMP180. V případě neúspěšné inicializace BMP180 je odeslána chybová hláška pomocí MQTT. Pokud se inicializace nezdaří třikrát je ESP restartováno.

```
void setup() {
  Serial.begin(9600);
  setup_wifi(); // připojení na wifi
  client.setServer(mqtt_server, 1883); // nastavení MQTT brokera
  delay(1000);
  dht.begin(); // inicializace DHT
  while (!bmp.begin()) {
    if (!client.loop()) {
      client.connect("sitiny_meteostanice");
      client.publish("sitiny/metsta/ch", "BMP setup f");
    }
  }
  delay(1000);
  ++pokusy;
  // kontrola nepovedených inicializací
  if (pokusy == 3) {
    ESP.restart();
  }
}
}
```

Obrázek 42 Funkce setup meteostanice ESP32 (vlastní)

Po ukončení funkce `setup()` je spuštěn cyklus `loop()` ve kterém je nejdříve zjištěno zda je meteostanice připojena na WiFi. Pokud není, je spuštěn a funkce `reconnect()` která má za úkol se zpět připojit. Následně je zkontrolováno, zda poslední měření hodnot proběhlo déle jak před deseti minutami a, pokud ano, tak jsou hodnoty teploty, vlhkosti a tlaku změřeny, poskládány do JSON formátu a odeslány pomocí MQTT. Celý cyklus se následně opakuje do nekonečna.

```
void loop() {
  if (!client.connected()) {
    reconnect();
  }
  if(!client.loop())
    client.connect("sitiny_meteostanice");

  now = millis();
  StaticJsonDocument<32> doc;
  char output[55];
  // Odesílání teploty každých 10 minut
  if (now - lastMeasure > 600000) {
    lastMeasure = now;
    // Svyčarání senzorů
    float humidity = dht.readHumidity();
    float temperatureC = bmp.readTemperature();
    float pressure = bmp.readPressure();

    doc["t"] = temperatureC;
    doc["p"] = pressure;
    doc["h"] = humidity;

    serializeJson(doc, output);
    client.publish("sitiny/metsta", output);
    Serial.println("Sent");
  }
}
```

Obrázek 43 Cyklus `loop()` meteostanice ESP32 (vlastní)

Samotná meteostanice je umístěna venku pod okenní římsou a uvnitř ochranné krabice, ven je vyveden senzor DHT11 a pro BMP je v krabici vytvořen otvor, přes který čidlo načítá hodnoty



Obrázek 44 Umístění meteostanice (vlastní)

4.3.2 Síťová vrstva

Z důvodu doporučení vývojářem softwaru byla zvolena výše uvedená deska TTGO, na kterou byl pomocí internetového prohlížeče nainstalován OpenMQTTGateway. Po úspěšné instalaci začala deska sloužit jako wifi přístupový bod. Po připojení na nově vytvořenou wifi síť bylo zobrazeno konfigurační okno bány, ve kterém bylo nutné vyplnit SSID a přihlašovací údaje k síti na kterém bude brána provozována, Jedná se o IP adresu MQTT Brokeru, jméno vytvořené brány a MQTT téma na které budou odesílána veškeré přijaté zprávy a heslo pro přístup do konfigurace. Jakmile byla konfigurace ukončena, AP bylo vypnuto a mikrokontroler se připojil na wifi. Následně byla vypsána na display desky TTGO IP adresa, na které byla provedena následná konfigurace LoRa. Zde byla zvolena frekvence 433MHz, a zadáno synchronizační slovo, aby brána nechytala nevyžádanou komunikaci.

V případě odesílání zpráv přes OpenMQTTGateway je formát MQTT témat následující.

- Základní téma / jméno brány / překlad protokolů / id odesílatele
- sítiny/LORA_GATEWAY/LORaToMQTT/u11

Mikrokontroler s OpenMQTTGateway byl následně umístěn za okno směřující k úlům, kde byl monitorovací systém nasazen. Vzdálenost mezi LoRa bránou a úly je přibližně 100 metrů.



Obrázek 45 Umístění LoRa brány (vlastní)

4.3.3 Vrstva zpracování dat

Vrstva pro zpracování dat (processing layer) je často nazývána jako Middleware. Tato vrstva se stará o příjem, dekodování, extrakci, transformaci, validaci a ukládání dat.

4.3.3.1 Příprava serveru

Zpočátku bylo nutné zvolit operační systém pro Raspberry Pi. Jako ideální vycházel Raspberry Pi OS především díky tomu, že je vyvíjen samotným výrobcem jednodeskových počítačů Raspberry. Jelikož nebyla zapotřebí přítomnost uživatelského rozhraní, byla zvolena i pro menší nároky 64bitová Lite verze. Ta byla nainstalována na 32 GB SD kartu pomocí programu Raspberry Pi Imager.

Při přípravě instalace byla využita možnost přednastavení SSH pro vzdálený přístup.

Pro IoT projekty, ve kterých se vyskytuje centrální prvek v roli serveru je nutné nastavit statickou IP adresu, v opačném případě by při změně adresy pomocí DHCP mohlo v aktuálním projektu například dojít k přerušení mezi LoRa bránou a Raspberry Pi. Z toho důvodu byla provedena konfigurace statické IP adresy editací konfiguračního souboru „dhcpcd.conf“

Následovala aktualizace operačního systému pomocí po sobě následujících příkazů

- *sudo apt update*
- *sudo apt upgrade*

Pro jednodušší nasazení a zprávu, byla zvolena možnost provozovat potřebné aplikace uvnitř kontejnerů díky IoTStack, ten byl stažen a nainstalován následujícím příkazem.

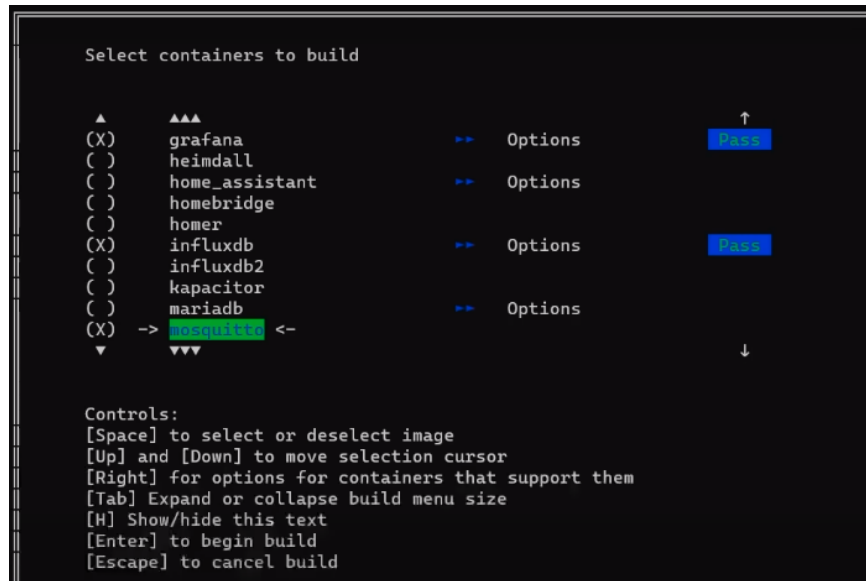
- *curl -fsSL https://raw.githubusercontent.com/SensorsIoT/IOTstack/master/install.sh / bash*

Po instalaci byl spuštěn script menu.sh uvnitř nově vytvořené složce IOTstack. Bylo zobrazeno menu na správu kontejnerů. Jako první byla vybrána možno „Build Stack“, ta je využívána k výběru aplikací, které budou nainstalovány a spuštěny uvnitř kontejnerů.

Zvoleny byly následující aplikace.

- Node-red – Automatizační grafický programovací nástroj
- Grafana – Nástroj pro grafické zobrazení dat z databáze
- Influxdb – Databázový systém pro úchovu dat

- Mosquitto – Zajištění MQTT komunikace



Obrázek 46 IoT stack menu (vlastní)

Po výběru aplikací se vrátil program do menu, kde dalším výběrem byl „Docker Commands“ ve kterém byla zvolena možnost „Build Stack“, která stáhla, nainstalovala a spustila všechny aplikace v kontejnerech.

Po spuštění bylo menu IOTstack opuštěno a do terminálu byl zadán následující příkaz pro možnost zadávání terminálových příkazů uvnitř kontejneru s InfluxBD.

```
docker exec -it influxdb influx
```

Následným příkazem byla vytvořena databáze s názvem *bee_data*.

```
CREATE DATABASE bee_data
```

Dalším prvkem, který bylo zapotřebí nakonfigurovat je platforma Grafana. Do Grafany je přístupováno pomocí internetového prohlížeče po zadání ip adresy, která byla nastavena v konfiguračním souboru *dhcpd.conf* a portu 3000. Výsledný příkaz do prohlížeče může vypadat následujícím způsobem. *http://192.168.1.88:3000*.

Při prvním spuštění je vyžadováno vytvoření uživatelského účtu.

Pro vizualizaci naměřených hodnot bylo nutné nakonfigurovat zdroj dat. V hlavním menu byla vybrána možnost DATA SOURCE ve které jsou vyobrazeny veškeré druhy databází, ze kterých je Grafana schopna zobrazovat data. Byla vybrána InfluxDB v záložce Time series databases. Následně bylo nutné vyplnit URL adresu, jméno databáze a nastavit http metodou na GET. Po provedení těchto úkonů byla Grafana připravena na vytváření dashboardů.

4.3.3.2 Node-red

Pomocí Node-red bylo vypracováno zpracování zpráv které byli odeslány pomocí mikrokontrolerů. Veškeré přijímané zprávy jsou posílány pomocí MQTT protokolu, k zachycení těchto zpráv je využíván uzel „MQTT in“.

Pro funkčnost uzlu je nutné přidání MQTT Brokera, prostřednictvím kterého je komunikace protokolem zprostředkovávána. Po umístění uzlu na pracovní plochu a jeho následném otevření byla v kolonce server zanechána možnost „Add new MQTT-broker...“ a bylo kliknuto na vedlejší ikonu editace. Následně bylo otevřeno konfigurační okno, kde bylo nutné vyplnit IP adresu na které je provozován MQTT Broker. Jelikož všechny aplikace jsou spuštěny v kontejnerech nebylo možné využít adresy localhost, ale bylo nutné zadat IP adresu samotného Raspberry pi. Následně byl vyplněn název pro větší přehlednost a editace Brokeru byla ukončena. Po zvolení nově vytvořeného Brokera v kolonce server bylo nutné vyplnění položky „topic“. V práci se vyskytují pouze dva druhy otevíratelných témat, které jsou následující:

- sitiny/metsta – Data z meteostanice
- sitiny/LORA_GATEWAY/LORAtoMQTT/ul1 – Data z úlu 1

Uvnitř přijímaných se nacházejí data ve formátu JSON. Struktura dat je pro každé téma jiná, zatím co zprávy odeslané meteostanicí obsahují veškeré naměřené hodnoty, data odesílána pomocí mikrokontroleru upevněného na úlu jsou odesílána jednotlivě.

```
sitiny/metsta : msg.payload : Object
  ▼ object
    t: 5.5
    p: 92201
    h: 76
```

Obrázek 47 Obsah zprávy odeslané meteostanicí (vlastní)

U zpráv odeslaných meteostanicí se vyskytují tři proměnné. V proměnné „t“ se nachází naměřená venkovní teplota, v „p“ tlak a v „h“ relativní vlhkost.

```

sitivity/LORA_GATEWAY/LORAtoMQTT/u1 :
msg.payload : Object
  ▾ object
    typ: "co"
    id: "u11"
    v: 1357
    rssi: -88
    snr: 8.25
    pferror: 4538
    packetSize: 35

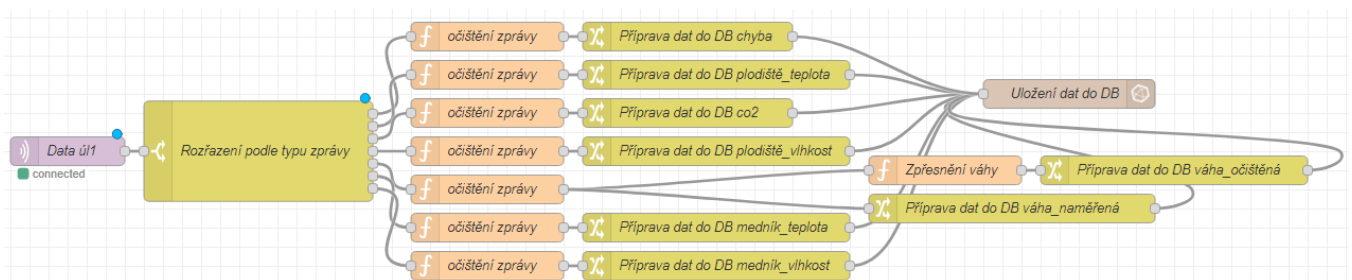
```

Obrázek 48 Obsah zprávy o míře CO2
odeslané úlovým ESP32 (vlastní)

zprávy informující o stavu úlu mají tři hlavní atributy, „**typ**“, který informuje, o který senzor se jedná a co měří, „**id**“ pojednávající o umístění senzoru a „**v**“ ve kterém je uložena naměřená hodnota. Typy zpráv jsou následující:

- **ch** – chybová hláška
- **co** – hodnota CO2
- **tp** – teplota plodiště
- **vp** – relativní vlhkost plodiště
- **tm** – teplota medníku
- **vm** – relativní vlhkost medníku
- **va** – váha úlu

V případě že přijde MQTT zpráva, která je původem odeslaná úlem, je nejdříve rozřazena podle typu zprávy pomocí uzlu switch, následně putuje do uzlu „function“, ve kterém je pomocí scriptu obsah zprávy změněn na hodnotu uloženou v proměnné „**v**“.



Obrázek 49 Propojení uzlů v Node red pro data z úlu (vlastní)

Všechny zprávy putují do uzlu „change“, ve kterém je obsah zprávy obohacen o proměnnou, která je následně využita jako jméno pole, pod kterým má být hodnota uložena v databázi.

```

1  {
2      "plodiště_teplota": msg.payload
3  }

```

Obrázek 50 Obsah uzlu change pro teplotu plodiště (vlastní)

Zvláštním případem je však zpráva typu „va“, která očištění dělí a jedna část putuje do dalšího „function“ uzlu, uvnitř kterého se nachází funkce, která od naměřené hodnoty odečte přibližnou váhu samotného prázdného úlu.

```

1  var pocetnas = global.get("pocetnas_ul1");
2  var vahanas = global.get("vahanas_ul1");
3  var vahadna = global.get("vahadna_ul1");
4  var vahavyko = global.get("vahavyko_ul1");
5  var pramku = global.get("pramku_ul1");
6  var vaharamek = global.get("vaharamek_ul1");
7  var vaha = msg.payload;
8
9  msg.payload = (vaha-(pocetnas*vahanas)-(pramku*vaharamek)-vahadna-vahavyko);
0  return msg;

```

Obrázek 51 Script pro očištění váhové hodnoty o fixní položky

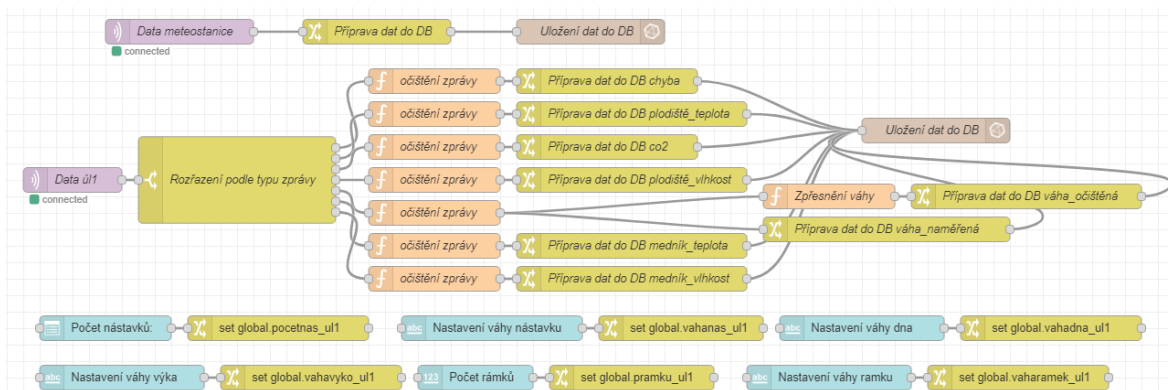
V Node-red dashboard bylo vytvořeno uživatelské rozhraní pro editaci vstupů, které ovlivňují statickou váhu měřeného úlu. Uživatel si tak může editovat počet nástavků, rámků a váhy jednotlivých částí úlu pro přesnější měření hmotnosti očištěné od váhy částí úlů

Obrázek 52 Uživatelské rozhraní pro zadání váhových hodnot

Váha jednoho rámečku s mezerníkem bez mezistěny váží přibližně 0,18 kg, plně nastrojené dno úlu přibližně 4 kg, jeden nastavek 6.2 kg a víko 2 kg. Hodnoty jsou následně uloženy do globálních proměnných, které jsou využity v uzlu s názvem „Zpřesnění váhy“.

Veškeré uzly „change“ jsou následně napojeny do uzlu „influxdb out“, který slouží k uložení naměřených hodnot do již připravené databáze. Stejně jako v případě uzlu MQTT in bylo nutné nakonfigurovat připojení na databázi s rozdílem, že v kolonce Database bylo nutné vyplnit jméno databáze která byla vytvořena při přípravě serveru. Po výběru databáze byl v kolonce „Measurement“ vyplněn název, pod kterým kontejnerem budou data uložena.

Datový tok pro zprávy odeslané meteostanicí je takřka identický s tokem úlu s rozdílem kdy v uzlu „change“ není hlavička přidávána jedné hodnotě, ale třem.



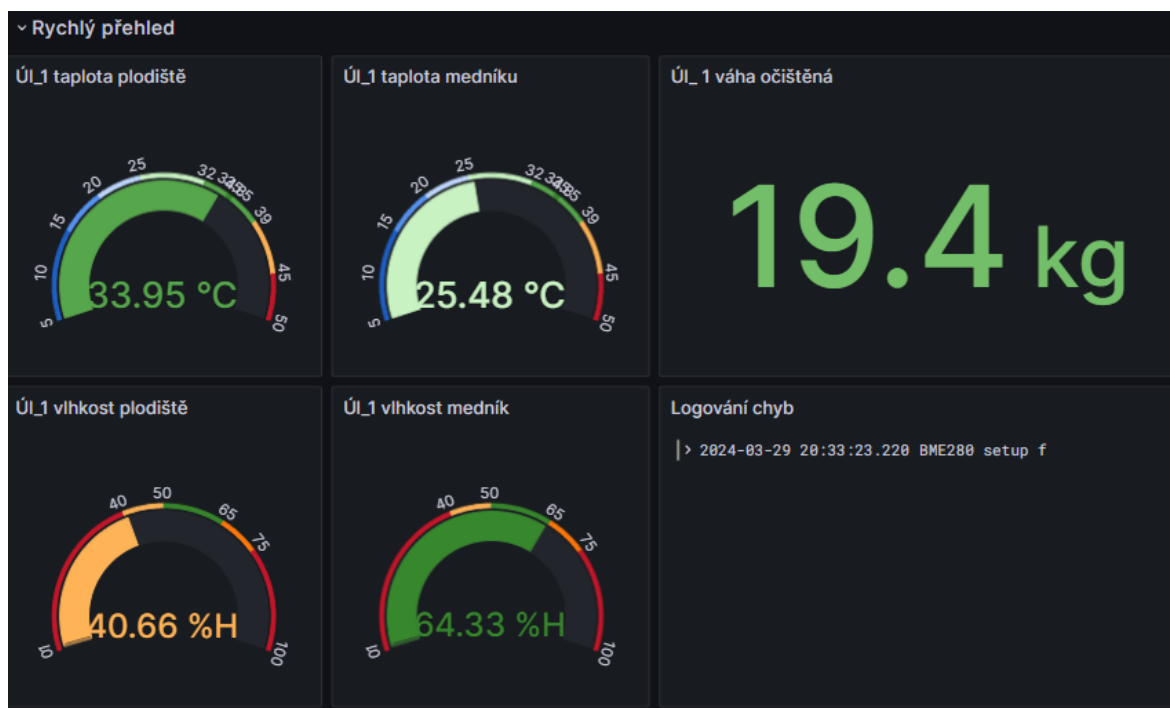
Obrázek 53 Všechny uzly v Rode red (vlastní)

4.4 Aplikační vrstva

Hlavním úkolem aplikační vrstvy (application layer) je zajistit prezentaci a vizualizaci dat. Často je nutné data zpracovat a prezentovat způsobem, který umožní snadnou a přesnou interpretaci, k čemuž se standardně využívají grafické nástroje typu dashboard

Prezentace dat je zajištěna pomocí dashboardů vytvořených v aplikaci Grafana. Jsou zobrazovány veškeré zájmové oblasti, které jsou vyčítány z databázového systému Influx DB. Z úlu jsou vizualizovány teplota a vlhkost jak z medníku, tak z plodiště, celková váha, váha očištěná o hmotnost částí úlu a koncentrace CO₂. Dále jsou v dashboardu zahrnuty venkovní vlhkost, teplota a tlak.

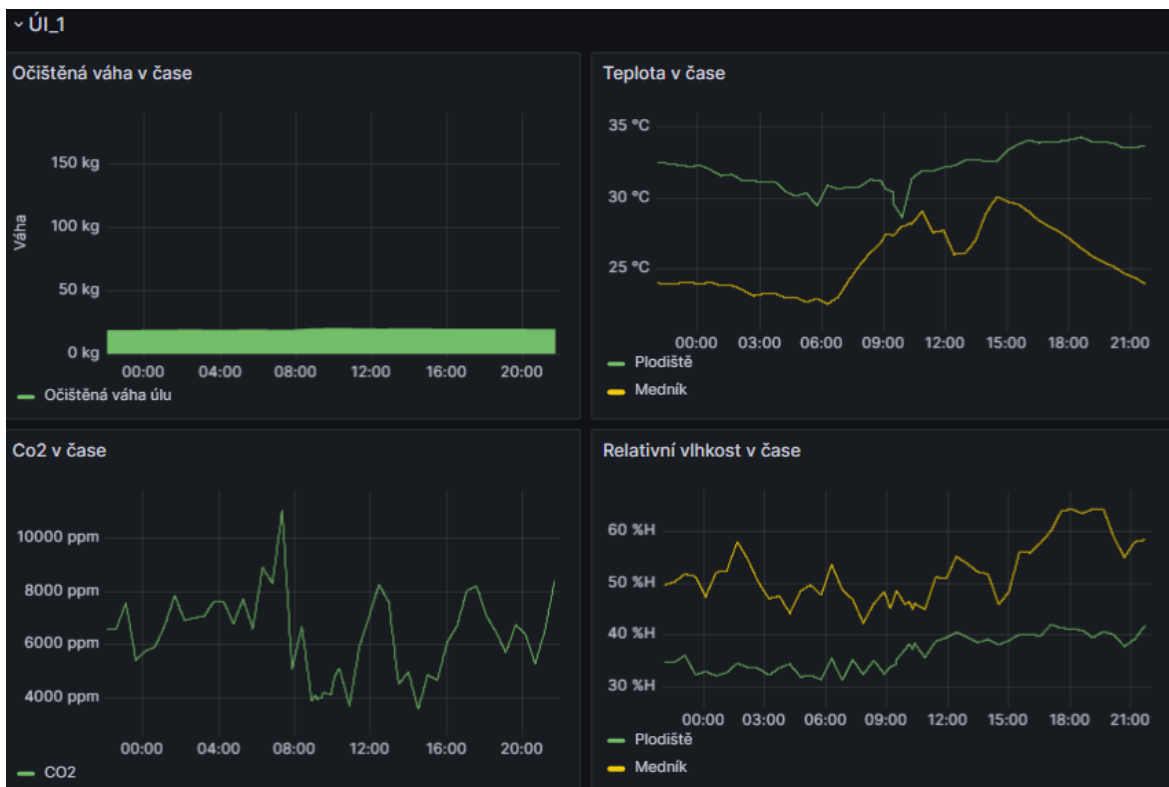
Jednotlivé vizualizace jsou rozděleny do jednotlivých záložek. První záložkou, kterou uživatel vidí je rychlý výběr, zde jsou přehledně zobrazeny poslední uložené hodnoty, aby měl uživatel okamžitou informaci o tom, jaké v úlu panují podmínky a nebylo nutné si prohlížet graf.



Obrázek 54 Záložka rychlý přehled (vlastní)

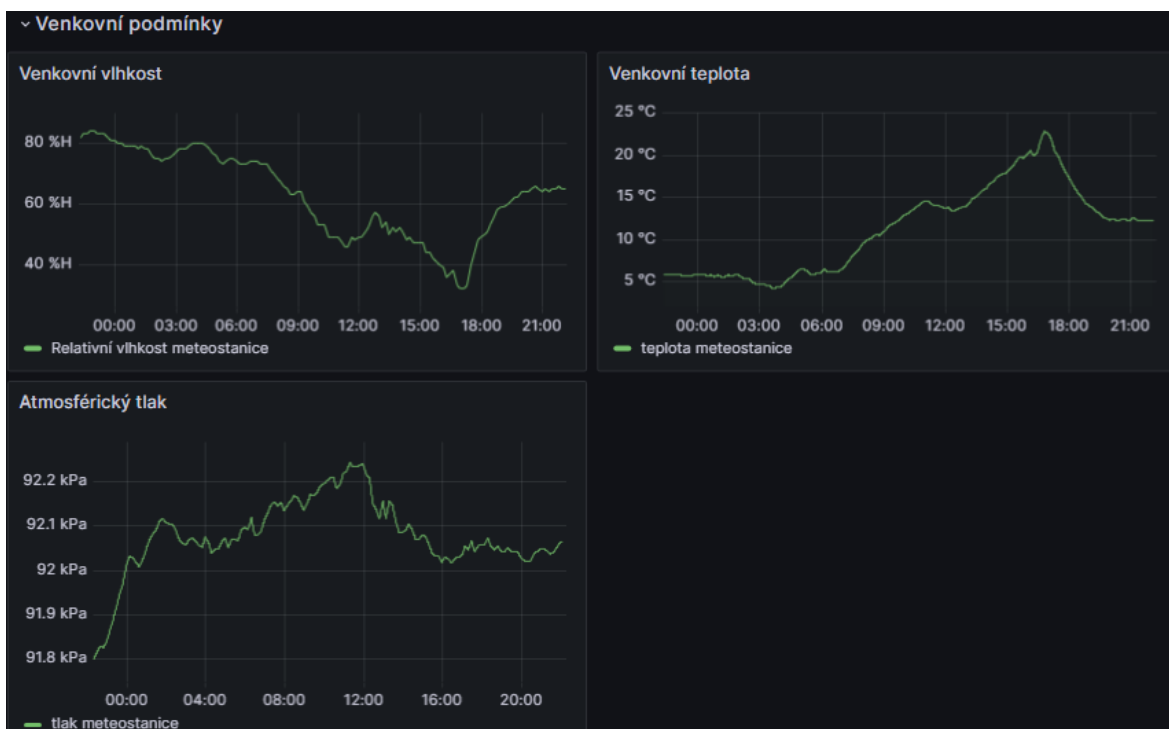
Pro záložku rychlý výběr, byly zvoleny hodnoty teploty a vlhkosti v nastavcích, očištěná váha o hodnoty nastavené v Node red dashboard a také okno, ve kterém jsou vypsány logové chyby. Teploty a relativní vlhkosti jsou zobrazeny pomocí vizuálního prvku gauge, který slouží k zobrazení a monitorování jediné hodnoty v určitém rozsahu. Tento prvek se obvykle používá k prezentaci aktuální hodnoty nějaké metriky nebo ukazatele, jako je teplota, rychlost, výkon apod. Naměřená hodnota je zobrazována na stupnici, kde jsou vyznačeny jednotlivé rozsahy a barevně je reprezentováno, zda je v přijatelných normách. To umožňuje uživatelům rychle porozumět aktuální hodnotě metriky a jejímu poměru k celkovému rozsahu. Dále je číselně zobrazena aktuální váha úlu a logové okno, ve kterém jsou vypsány zaznamenané chybové hlášky, které jsou odeslány mikrokontrolery v případě problému.

Další záložka slouží pro vizualizaci dat naměřených na určitém úlu v čase. Oproti rychlému výběru je přidána grafická reprezentace míry CO₂ uvnitř úlu. Všechna data jsou zobrazena jako časová řada. Teploty uvnitř medníku a plodiště jsou zahrnuty uvnitř jediného grafu, stejný přístup se opakuje i pro relativní vlhkost.



Obrázek 55 Záložka pro Úl_1 (vlastní)

Na poslední záložce jsou vizualizovány venkovní podmínky. Je zde zahrnut atmosférický tlak, venkovní teplota a relativní vlhkost.



Obrázek 56 Záložka pro venkovní hodnoty (vlastní)

4.5 Napájení

Napájení bylo vyhotoveno podle návrhového schématu. Fotovoltaický panel byl napojen na solární regulátor. Napětí solárních panelů je přibližně 36 V, které následně regulátor mění na 12 V. Do regulátoru je napojená gelová baterie s kapacitou 17Ah. Veškerá ochrana baterie a logika dobíjení je řešena právě regulátorem který při vybití akumulátoru pod hranici 30 % baterii odstaví, aby nedošlo k jejímu poškození. Regulátor disponuje i zátěžovým vývodem, který je využit pro napájení ESP32, to však vyžaduje maximální napětí 5 V, zatím co veškeré vývody regulátoru mají námětí 12 V. Tento problém byl vyřešen využitím USB měniče do automobilového zapalování, do kterého je zapojen mikro USB kabel, který napájí ESP32.



Obrázek 57 Solární panel s úly (vlastní)

Kapacita použitého akumulátoru při možném využití 70% je přibližně 142 Wh, což znamená že při spotřebě mikrokontroleru využitého pro měření hodnot úlu za den, vychází že při absenci slunečního svitu by měl systém fungovat přibližně 70 dní.



Obrázek 58 Uložení baterie a solárního regulátoru (vlastní)

5 Výsledy

5.1 Náklady na vyhotovení

Položka	Stav	Cena v Kč
Raspberry pi B3+	použité	800
Solární panel	použité	500
Baterie	použité	310
Solární regulátor	použitý	300
Váhové senzory + HX711	nový	130
TTGO Wifi LoRa ESP32	nový	505
ESP32	nový	198
SCD41	nový	350
BMP180	nový	48
BME280	nový	188
SX1278 LoRa modul	nový	198
DHT11	nový	48
Elektrikářský materiál	nový	1200

Tabulka 4 Nákladová tabulka (vlastní)

Celková pořizovací cena celého systému je přibližně 4800 Kč.

Veškeré komponenty až na SCD41 byly nakoupeny v České republice. Cena nového čidla SCD41 se v tuzemských obchodech v době vyhotovování této práce pohybuje okolo 1000 Kč. Komponenty, které jsou považovány za použité byly pořízeny pomocí inzerčního serveru Bazoš a aukčního portálu Aukro, cena nového zboží se tedy může dramaticky lišit. Cena rozšíření o další úl na stejném místě je přibližně 1080 Kč, při větším počtu připojených zařízení by bylo nutné pořízení akumulátoru s větší kapacitou. Pořizovací cena celého systému by se dala ještě snížit, kdyby se většina materiálu pořídila ze zahraničních internetových obchodů

5.2 Budoucí vývoj

Pozornost bude zaměřena především na snímání a analýzu zvuku, neboť se jedná o důležitý ukazatel, díky kterému je možné mu je možné predikovat chování včel.

Velkým důraz bude také kladen na pokus implementace vytvořeného systému do projektu Včelstva online, aby bylo možné naměřená data sledovat i mimo lokální síť.

Taktéž proběhne pokus o implementaci sledování stavu baterie, aby bylo možné v čas vyřešit případný problém s napájením.

Systém bude nadále nasazen na testovacím místě.

5.3 Výsledky testování

Během testování se ukázalo, že systém je spolehlivý. Všechna data přišla vždy v očekávaný čas a za celou dobu provozu nenastala chyba žádného senzoru. Během měření se ukázalo, že teplota v plodišti je dlouhodobě vyšší než v nástavku, který substituuje medník, což je pochopitelné, jelikož systém byl testován v relativně chladných podmínkách, kdy je většina včel stažena do plodiště. V horním nástavku se naopak po celou dobu měření udržuje o průměrně 15% větší relativní vlhkost. Koncentrace CO₂ má velké spektrum, mezi sedmou až osmou hodinou ranní bývají naměřené hodnoty nejvyšší, kdy se koncentrace pohybuje okolo 11000 ppm, po osmé koncentrace začne rapidně hlesat až se dostane na hodnoty okolo 4000 ppm. Výsledky naměřených dat bude možné zpracovat až z dlouhodobého měření, především během snůšky.

6 Závěr

System zvoleného monitorování včelstva je možné sestavit za mnohem nižší cenu, než za kterou jsou prodávána komerční řešení. Má také velký potenciál pro budoucí rozšiřování, na rozdíl od již koupených sestav. Stavba však vyžaduje alespoň elementární elektrikářské znalosti a zkušenosti s pájením. Tato práce však může posloužit jako inspirace pro vytvoření identického nebo upraveného systému vytvářeného během zhotovování práce. V diplomové práci byly představeny možné způsoby měření hodnot uvnitř medníku pomocí dvou různých senzorů. Přestože původní návrh během vyhotovování práce nevykazoval uspokojivé výsledky, neznamená to, že je špatný. Při použití nových senzorů by fungoval bez problému. Upravený návrh, který využívá BME280 je však lepší ve všech ohledech.

7 Bibliografie

- (1) CRAMP, David. *Včelařství: obrazový průvodce : od pořízení včelstev po medobraní : více než 400 návodných fotografií*. 2. vyd. Čestlice: Rebo, 2014. ISBN 978-80-255-0831-2.
- (2) FLOTTUM, Kim. *Příručka včelaře: jak chovat včely na dvoře, za domem, na střeše nebo na zahradě*. Aktualizované vydání. [Praha]: Slovart, 2019. ISBN 978-80-7529-844-7.
- (3) GUSTIN, Yves. *Ilustrované včelařství: nepostradatelná rodinná příručka pro odvážné včelaře*. První vydání. Kraj (Baobab: GplusG: GplusG. V Praze: Baobab, 2010. ISBN 978-80-87060-27-8.
- (4) KUBIŠOVÁ, Sylvie a HÁSLBACHOVÁ, Hana. *Včelařství*. 1.vyd. Brno: VŠZ, 1992. ISBN 80-7157-024-9.
- (5) BOGUSCH, Petr. *Domečky pro včely a užitečný hmyz*. První vydání. Praha: Grada, 2019. ISBN 978-80-271-2510-4.
- (6) ŠEFČÍK, Jozef. *Začínáme včelařit*. Praha: Grada, 2014. ISBN 978-80-247-4857-3.
- (7) SIMMONDS, Michael. How Honeybees Maintain Temperature and Humidity in a Beehive. online. In: *BeeKeepClub*. 2023. Dostupné z: <https://beekeepclub.com/how-honeybees-maintain-temperature-and-humidity-in-a-beehive/>. [cit. 2024-02-07].
- (8) G. MEIKLE, William; BARG, Aidan a WEISS, Milagra. Honey bee colonies maintain CO2 and temperature regimes in spite of change in hive ventilation characteristics. online. In: *Apidologie*. 2022, s. 2-12. Dostupné z: <https://hal.science/hal-04183757/document>. [cit. 2024-03-17].
- (9) GERSTMEIER, David a MILTENBERGER, Tobias. *Ekologické včelaření: včely na prvním místě*. 1.st. Praha: Grada Publishing, 2020. ISBN 978-802-7120-154.
- (10) MOŽÍŠ, Pavel. Zimní chumáč včel a jeho teplota. online. In: *BeeInfo.cz*. 2018. Dostupné z: <https://beeinfo.cz/zimni-chumac-vcel-a-jeho-teplota/>. [cit. 2024-03-30].
- (11) BIENEFELD, Kaspar. *Včelařství krok za krokem: pro milovníky krásného koníčka*. II. vydání. [Líbeznice]: Víkend, 2010. ISBN 978-80-7433-023-0.
- (12) Úlové váhy. online. In: *Úlové váhy*. 2024. Dostupné z: <https://www.ulovevahy.cz/ulove-vahy/>. [cit. 2024-03-30].

- (13) Weighing your beehive: Why and how: Why is weighing your beehive important?. online. In: *PERFECTBEE*. 2024. Dostupné z: <https://www.perfectbee.com/blog/weighing-your-beehive-why-and-how>. [cit. 2024-03-30].
- (14) Proč měřit koncentraci CO₂?. online. In: *Šetřiče EU*. 2018. Dostupné z: <https://www.energomonitor.eu/blog/proc-merit-koncentraci-co2>. [cit. 2024-03-31].
- (15) BENCSIK, Martin; MCVEIGH, Adam; TSAKONAS, Costas; KUMAR, Tarun; CHAMBERLAIN, Luke et al. A Monitoring System for Carbon Dioxide in Honeybee Hives: An Indicator of Colony Health. online. *Sensors*. 2023, roč. 23, č. 7, s. 1. ISSN 1424-8220. Dostupné z: <https://doi.org/10.3390/s23073588>. [cit. 2024-03-18].
- (16) GUERRERO, Antonio; ANAYA, Tonatiuh; RAMÍREZ, Efrén a TEJADA, Carlos. Frequency Analysis of Honey Bee Buzz for Automatic Recognition of Health Status: A Preliminary Study. *Research in Computing Science*. 2017, roč. 2017-08-07, č. 142, s. 89–98. ISSN 1870-4069.
- (17) TERENCEZI, Alessandro; CECCHI, Stefania a SPINSANTE, Susanna. On the Importance of the Sound Emitted by Honey Bee Hives. online. *Veterinary Sciences*. 2020, roč. 7, č. 4. ISSN 2306-7381. Dostupné z: <https://doi.org/10.3390/vetsci7040168>. [cit. 2023-11-16].
- (18) *BEE HIVE MONITORING*. online. In: *BEE HIVE MONITORING*. 2024. Dostupné z: <https://beehivemonitoring.com/>. [cit. 2024-03-30].
- (19) SINHA, Satyajit. State of IoT 2023: Number of connected IoT devices growing 16% to 16.7 billion globally. online. In: . Dostupné z: <https://iot-analytics.com/number-connected-iot-devices/>. [cit. 2023-10-24].
- (20) IoT Protocols: A Comprehensive Guide to Different Smart Home Protocols. online. In: <https://microcontrollerslab.com/>. 2024. Dostupné z: <https://microcontrollerslab.com/iot-protocols-types/>. [cit. 2024-03-30].
- (21) ZÁKLADNÍ PŘEHLED O TECHNOLOGII WIFI. online. In: *FCC PS*. 2024. Dostupné z: <https://www.fccps.cz/zakladni-prehled-o-technologie-wifi>. [cit. 2024-03-31].
- (22) KLEMENT, Milan. *Technologie bezdrátových sítí základní principy a standardy*. Online. 1.st. 2019.
- (23) Zigbee Range: You Must Know The Truth. online. In: *Reolink*. 2024. Dostupné z:

- <https://reolink.com/blog/zigbee-range/>. [cit. 2024-03-29].
- (24) PECH, Jiří. IOT TECHNOLOGIE: LORA A LORAWAN (3/5). online. In: *Eman*. 2024. Dostupné z: <https://www.eman.cz/blog/iot-technologie-lora-a-lorawan-3-5/>. [cit. 2024-03-30].
- (25) LoRa Technology: Its Difference With LoRaWAN. online. In: *Renke*. 2024. Dostupné z: <https://www.renkeer.com/lora-technology/>. [cit. 2024-03-30].
- (26) LORENC, Tomáš. *MĚŘENÍ VLASTNOSTÍ LORA/LORAWAN KOMUNIKACE*. Bakalářská práce. Brno: VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ, 2020.
- (27) PAPEŽ, JOSEF. *BEZDRÁTOVÝ ZVONEK S DIGITÁLNÍM PŘENOSEM HLASU*. Diplomová práce, vedoucí ING. RADEK VÁCLAVÍK. Brno: VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ, 2008.
- (28) Všeobecné oprávnění č. VO-R/10/07.2021-8 k využívání rádiových kmitočtů a k provozování zařízení krátkého dosahu. In: . 2021.
- (29) DRTINA, Martin. Sigfox v Česku končí, T-Mobile vypíná vysílače. online. In: *Lupa.cz*. 2024. Dostupné z: <https://www.lupa.cz/aktuality/sigfox-se-v-cesku-odmlcel-zda-je-to-konec-teto-site-internetu-veci-jeji-provozovatele-nekomentuji/>. [cit. 2024-03-30].
- (30) Overview of IOT Networks. online. In: *IoT Factory*. 2020. Dostupné z: <https://iotfactory.eu/iot-knowledge-center/overview-of-iot-networks/>. [cit. 2024-03-31].
- (31) Čidlo CO2 SCD41 prakticky. online. In: *Chiptron*. 2021. Dostupné z: https://chiptron.cz/articles.php?article_id=294. [cit. 2024-02-10].
- (32) *SCD4x: Breaking the size barrier in CO2 sensing*. Version 1.4. 2023.
- (33) Alps Alpine Develops Ambient Air Sensor Module Useful for COVID-19 Prevention: Monitoring of the Surrounding Environment with High-Accuracy Sensing of CO2, Temperature, Humidity and Other Parameters. online. In: *Alps Alpine*. 2024. Dostupné z: <https://www.alpsalpine.com/eu/news/detail/2021-0628-01/>. [cit. 2024-03-31].
- (34) MH-Z14A CO2 Meter Tutorial with Arduino, ESP8266 or ESP32. online. In: *DiyOT*. 2021. Dostupné z: <https://diyi0t.com/mh-z14a-co2-meter-arduino-esp8266-esp32/>. [cit. 2024-03-25].
- (35) NDIR senzor plynu MH-Z14 Oxid uhličitý CO2. online. In: *Láskakit*. 2024. Dostupné z: <https://www.laskakit.cz/ndir-senzor-plynu-mh-z14-oxid-uhlicity-co2/#relatedFiles>.

- [cit. 2024-03-26].
- (36) Princip činnosti snímačů. online. In: *Váhy roh*. 2010. Dostupné z: https://www.tenzometricke-snimace.cz/Princip-cinnosti-snimacu-c11_0_1.htm?gad_source=1&gclid=Cj0KCQjw8J6wBhDXARIsAPo7QA-h-y5i6obRT3eaenILPqkTWD511WW7QfjT8rkadicWkcOUCJimkXkaAnCLEALw_wcB. [cit. 2024-03-31].
- (37) Váhový senzor - 50 kg. online. In: *Dárek.cz*. 2024. Dostupné z: <https://dratek.cz/arduino/2202-vahovy-senzor-50-kg.html>. [cit. 2024-03-31].
- (38) Měřicí můstek. online. In: *Střední odborná škola a Střední odborné učiliště Lanškroun*. 2020. Dostupné z: <https://www.spslan.cz/images/PDF1314/rekvalifikace/mustek.pdf>. [cit. 2024-03-31].
- (39) Arduino Weight Measurement using Load Cell and HX711 Module. online. In: *Circuitit Digest*. 2023. Dostupné z: <https://circuitdigest.com/microcontroller-projects/arduino-weight-measurement-using-load-cell>. [cit. 2024-03-31].
- (40) DHT11 vs DHT22 vs LM35 vs DS18B20 vs BME280 vs BMP180. online. In: *Random nerd tutorials*. c2013-2024. Dostupné z: <https://randomnerdtutorials.com/dht11-vs-dht22-vs-lm35-vs-ds18b20-vs-bme280-vs-bmp180/>. [cit. 2024-03-31].
- (41) ESP32 a čidlo teploty a vlhkosti DHT11/DHT22. online. In: *Fyzikální kabinet FyzKAB*. 2024. Dostupné z: <http://kabinet.fyzika.net/ESP32/ESP32-dht/ESP32-a-cidlo-teploty-a-vlhkosti-DHT11.php>. [cit. 2024-03-31].
- (42) BOSH ENZORIC. BME280: Combined humidity and pressure senzor. online. In: BOSH ENZORIC. *Mouser*. 2024. Dostupné z: <https://www.mouser.com/datasheet/2/783/BST-BME280-DS002-1509607.pdf>. [cit. 2024-03-26].
- (43) ESP32 – Jak používat GPIO piny?. online. In: *Fyzikální kabinet FyzKAB*. 2022. Dostupné z: <http://kabinet.fyzika.net/ESP32/ESP32-gpio/vstupy-a-vystupy-ESP32.php>. [cit. 2024-02-08].
- (44) ESP32 DEVKIT ESP32-WROOM-32 Development Board ESP32 ESP-32S 2.4GHz WiFi+BT BLE Dual-core Dev Module CP2102 for Arduino DOIT. online. In: *Amazon*. 2024. Dostupné z: <https://www.amazon.com/ESP32-WROOM-32-Development-ESP-32S-Bluetooth-Arduino/dp/B084KWNMM4>. [cit. 2024-03-31].
- (45) ESP32 Deep Sleep with Arduino IDE and Wake Up Sources. online. In: *Random Nerd*

- Tutorials*. 2019. Dostupné z: <https://randomnerdtutorials.com/esp32-deep-sleep-arduino-ide-wake-up-sources/>. [cit. 2024-02-08].
- (46) Insight Into ESP32 Sleep Modes & Their Power Consumption. online. In: *Last Minute Engineers*. 2023. Dostupné z: <https://lastminuteengineers.com/esp32-sleep-modes-power-consumption/>. [cit. 2024-02-08].
- (47) ESP32 Deep Sleep & Its Wake-up Source. online. In: *Last Minute Engineers*. 2023. Dostupné z: <https://lastminuteengineers.com/esp32-deep-sleep-wakeup-sources/>. [cit. 2024-02-08].
- (48) MRÓZ, Mateusz. Co je Raspberry Pi?. online. In: *Botlandblog*. 2023. Dostupné z: <https://botland.cz/blog/co-je-raspberry-pi/>. [cit. 2024-02-08].
- (49) Architecture of Raspberry Pi. online. In: *Geeks for geeks*. 2020. Dostupné z: <https://www.geeksforgeeks.org/architecture-of-raspberry-pi/>. [cit. 2024-03-31].
- (50) Operating system images. online. In: *Raspberry pi*. 2024. Dostupné z: <https://www.raspberrypi.com/software/operating-systems/>. [cit. 2024-03-31].
- (51) 23 Best Operating Systems You Can Run on Raspberry Pi in 2024. online. In: *GM GeeksMint*. 2024. Dostupné z: <https://www.geeksmint.com/operating-system-in-raspberry-pi/>. [cit. 2024-03-31].
- (52) ApiCloud. online. In: *Api Včelař 4.0*. 2024. Dostupné z: <https://www.apivcelar.cz/>. [cit. 2024-03-30].
- (53) MONITOROVACÍ SENZORY DO ÚLŮ. *Forsage*. online. In: *Forsage.org*. 2020. Dostupné z: <https://www.forsage.net/>. [cit. 2024-03-30].
- (54) *Node-RED*. online. 2024. Dostupné z: <https://nodered.org/>. [cit. 2024-03-31].
- (55) Node RED Programming Guide: Programming the IoT. online. In: *Node RED Programming Guide*. 2023. Dostupné z: <https://noderedguide.com/>. [cit. 2024-03-31].
- (56) *Eclipse Mosquitto: An open source MQTT broker*. online. Dostupné z: <https://mosquitto.org/>. [cit. 2022-11-30].
- (57) HiveMQ vs. Mosquitto. online. In: *HiveMQ*. Dostupné z: https://info.hivemq.com/hivemq-vs-mosquitto?utm_source=adwords&utm_campaign=&utm_term=mosquitto%20mqtt&utm_medium=ppc&hsa_tgt=kwd-414712814116&hsa_cam=18392969014&hsa_src=g&hsa_net=adwords&hsa_kw=mos

- quitto%20mqtt&hsa_ad=623153723239&hsa_grp=144539382154&hsa_ver=3&hsa_ac
c=3585854406&hsa_mt=e&gclid=Cj0KCQiAm5ycBhCXARIsAPldzoViUidmWktbc
22v19OXFODj2ZhUdQ8_pn3GwBDstPUwBYHKOtFQqtOIaAtDIEALw_wcB. [cit.
2022-11-30].
- (58) COPE, Steve. Beginners Guide To The MQTT Protocol: What is MQTT?. online. In: *Steves Internet Guide*. Dostupné z: <http://www.steves-internet-guide.com/mqtt/>. [cit. 2022-11-30].
- (59) MQTT Essentials: The Ultimate Guide to MQTT for Beginners and Experts. online. In: *HiveMQ*. Dostupné z: <https://www.hivemq.com/mqtt-essentials/>. [cit. 2022-11-30].
- (60) COPE, Steve. *Understanding the MQTT Protocol Packet Structure*. online. In: *Steves Internet Guide: Practical Guide to MQTT and Mosquitto*. Dostupné z: <http://www.steves-internet-guide.com/mqtt-protocol-messages-overview/>. [cit. 2022-11-30].
- (61) MQTT V3.1 Protocol Specification. online. s. 42. Dostupné z: http://public.dhe.ibm.com/software/dw/webservices/ws-mqtt/MQTT_V3.1_Protocol_Specific.pdf. [cit. 2022-11-30].
- (62) *Influx data*. online. In: *Influx data*. 2024. Dostupné z: <https://www.influxdata.com/>. [cit. 2024-03-31].
- (63) *PostgreSQL*. online. In: *PostgreSQL*. c1996-2024. Dostupné z: <https://www.postgresql.org/>. [cit. 2024-03-31].
- (64) *MongoDB*. online. In: *MongoDB*. 2024. Dostupné z: <https://www.mongodb.com/>. [cit. 2024-03-31].
- (65) *Grafana Labs*. online. In: *Grafana Labs*. 2024. Dostupné z: <https://grafana.com/>. [cit. 2024-03-31].
- (66) *Elastic*. online. In: *Elastic*. 2024. Dostupné z: <https://www.elastic.co/>. [cit. 2024-03-31].
- (67) *IoT Stack*. online. In: *IoT Stack*. 2020. Dostupné z: <https://www.iotstack.co/>. [cit. 2024-03-31].
- (68) *OpenMQTTGateway v1.7.0*. online. In: *OpenMQTTGateway*. 2024. Dostupné z: <https://docs.openmqttgateway.com/>. [cit. 2024-03-24].
- (69) MASNÝ, Vojtech. *Optimalizace prenosu dat v LPWAN sítích z hlediska energetické*

náročnosti. Bakalářská práce. Praha: České vysoké učení technické v Praze, 2023.