

Jihočeská univerzita v Českých Budějovicích
Přírodovědecká fakulta

Sběr a analýza dat z inteligentního včelího úlu

Diplomová práce

Bc. Lukáš Širhal

Školitel Ing. Miroslav Skrbek, Ph.D.

České Budějovice 2018

ZADÁVACÍ PROTOKOL MAGISTERSKÉ PRÁCE

Student: Bc. Lukáš Širhal
(jméno, příjmení, tituly)

Obor – zaměření studia: Aplikovaná informatika 1802T001

Katedra: Ústav aplikované informatiky

Školitel: .Ing. Miroslav Skrbek, Ph.D.
(jméno, příjmení, tituly, u externího š. název a adresa pracoviště, telefon, fax, e-mail)

Garant z PřF:
(jméno, příjmení, tituly, katedra – jen v případě externího školitele)

Školitel – specialista, konzultant: konzultant: Ing. Václav Křišťůfek, CSc., BC AVCR, v.v.i.
- Ústav pudní biologie
(jméno, příjmení, tituly, u externího š. název a adresa pracoviště, telefon, fax, e-mail)

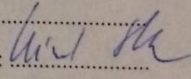
Téma magisterské práce: Sběr a analýza dat z inteligentního včelího úlu

Cíle práce:

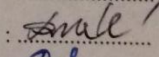
Navrhnete a realizujete technické a programové vybavení pro sběr a analýzu dat z včelího úlu. Zaměřte se na měření vlhkosti a teploty uvnitř i vně úlu, měření hmotnosti úlu a zvuku vydávaného včelami. Data přenášejte na vzdálený server a ukládejte do datového úložiště. Dále navrhnete a realizujete programové vybavení pro analýzu a vizualizaci dat z úlu. Pro analýzu dat zvolte vhodné, data miningové metody. Jak v analýze, tak ve vizualizaci se soustředte na vzájemné vztahy mezi jednotlivými měřenými veličinami, například teplotou v úlu ve vztahu k zvuku vydávaného včelstvem. Použité data miningové metody a vyvinuté programové vybavení řádně zdokumentujte. Rozsah práce upřesněte po dohodě s vedoucím práce.


Základní doporučená literatura : dodá vedoucí práce a konzultant

Financování práce : BC AVCR a UAI

Vedoucí práce: Ing. Miroslav Skrbek, Ph.D.podpis : 

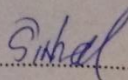
U externích vedoucích fakultní garant prácepodpis :

Garant oboru mag. studiapodpis : 

Vedoucí katedrypodpis 

Případný souhlas vedoucího ústavu AVpodpis :

V Českých Budějovicích dne 3.9. 2018

Převzal/a dne 26.11. 2018podpis : 

Bibliografické údaje

Bc. Lukáš Širhal, 2018: Sběr a analýza dat z inteligentního včelího úlu.

[Collection and analysis of intelligent beehive data, Mgr. Thesis, in Czech.] Faculty of Science, University of South Bohemia, České Budějovice, Czech Republic.

Anotace

Diplomová práce se zabývá vytvořením technického a programového vybavení inteligentního včelího úlu za účelem sběru dat. Měřenými charakteristikami jsou teplota, vlhkost, hmotnost včelího úlu a záznam včelího bzukotu. Tyto charakteristiky jsou měřeny za pomoci dostupných technologií z oboru IoT. Jedná se tedy o jednoduché jednočipové součástky s jednoznačným zaměřením. V práci je také zahrnuto vytvoření programové obsluhy těchto měření a následné komunikaci se vzdáleným serverem. Zdokumentována je také programová vybavenost vzdáleného serveru, jak po stránce analýzy příchozích hodnot, tak také jejich zobrazení. Software je tvořen v programovacím jazyku Python.

Annotation

This thesis deals with creating hardware and software equipment of intelligent beehive for data collection purpose. Measurement characteristics are temperature, humidity, weight of beehive and audio record of bees. They are measured by available IoT technology. They are simple single-chip component with a clear focus. This thesis also include creation of software for measurements this characteristics and communications with server. The software featured of the remote server are also documented. This software analyze incoming measurement and also provides their displays. Software in this thesis is developed in language Python.

Klíčová slova

Inteligentní úl, analýza dat, senzor teploty, senzor vlhkosti, nahrávání audia včel, Frourierova transformace, server, dokumentová databáze, JSON, Raspberry Pi

Keywords

Intelligent beehive, date analyze, temperature sensor, humidity sensor, bee audio record, Fourier transform, remote server, document database, JSON, Raspberry Pi

Prohlašuji, že svoji diplomovou práci jsem vypracoval samostatně pouze s použitím pramenů a literatury uvedených v seznamu citované literatury.

Prohlašuji, že v souladu s § 47b zákona č. 111/1998 Sb. v platném znění souhlasím se zveřejněním své diplomové práce, a to v nezkrácené podobě elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách, a to se zachováním mého autorského práva k odevzdanému textu této kvalifikační práce. Souhlasím dále s tím, aby toutéž elektronickou cestou byly v souladu s uvedeným ustanovením zákona č. 111/1998 Sb. zveřejněny posudky školitele a oponentů práce i záznam o průběhu a výsledku obhajoby kvalifikační práce. Rovněž souhlasím s porovnáním textu mé kvalifikační práce s databází kvalifikačních prací Theses.cz provozovanou Národním registrem vysokoškolských kvalifikačních prací a systémem na odhalování plagiátů.

V Českých Budějovicích dne 10. 12. 2018

.....
Bc. Lukáš Širhal

Poděkování

Na tomto místě bych chtěl poděkovat vedoucímu diplomové práce Ing. Miroslavu Skrbkovi Ph.D. za odborné vedení práce, ochotu pomoci a cenné rady, poskytnuté v průběhu vypracování diplomové práce. Dále bych chtěl poděkovat Ing. Václavu Křišťůfkovi, CSc. za umožnění práce se včelami a cenné rady doprovázející celý proces tvorby inteligentního úlu.

Práce byla finančně podpořena projekty Strategie AV21, ROZE (2017 – 2018) a z projektů Ústavu aplikované informatiky Přírodovědecké fakulty JČU.

Obsah

Úvod a motivace.....	1
1.1 Trocha včelařské historie.....	2
1.2 Význam včelaření pro člověka	3
1.3 Metodika.....	3
Teoretická část práce.....	4
1.4 Návrh architektury.....	4
1.5 Výběr základní desky	5
1.5.1 Arduino.....	5
1.5.2 Raspberry Pi	7
1.6 Komunikace	8
1.6.1 Wifi.....	8
1.6.2 ZigBee	8
1.6.3 GSM/GPRS	9
1.7 Senzory.....	10
1.7.1 Teplota, vlhkost	10
1.7.2 Hmotnost	11
1.7.3 Záznam zvuku.....	12
1.8 Analýza včelího bzukotu	13
1.9 Uchování naměřených hodnot.....	14
1.10 Zobrazení dat.....	15
1.10.1 ElasticSearch	15
1.11 Kibana	16
Praktická část práce	17
1.12 Hardware	17
1.12.1 Základní deska.....	18
1.12.2 Hardwarové vybavení základní desky	19
1.13 Komunikace	20
1.14 Senzory.....	22
1.14.1 Měření teploty a vlhkosti	22
1.14.2 Zvuk.....	27
1.14.3 Váha.....	29
1.15 Zpracování přijatých dat.....	31
1.15.1 HTTPServer.....	31
1.15.2 BaseHTTPRequestHandler.....	32
1.15.3 Application/octet-stream.....	33

1.15.4	Application/json.....	34
1.16	Analýza zvuku.....	36
1.17	Ukládání naměřených dat.....	40
1.18	Zobrazení naměřených dat.....	43
Testování		47
Aktuální využití projektu.....		48
Návrhy na zlepšení		49
1.19	Senzory a mikrofon	49
1.20	Ukládání dat	49
1.21	Analýza zvuku.....	50
Závěr		51
Seznam použité literatury		55
Seznam obrázků.....		57
Seznam tabulek.....		58
Seznam bloků kódů		59

Úvod a motivace

Populace včely medonosné (*Apis mellifera*) v posledních letech dramaticky klesá. Pokles je jak důsledkem zvýšeného infekčního tlaku ze strany včelích patogenů, tak i nevysvětlitelných úhynů celých včelstev v průběhu roku, a především v zimních měsících. Zhoršený zdravotní stav včel, chápaných jako hlavní opylovači většiny klíčových zemědělských plodin, je vnímán s výrazným neklidem. Po světě tedy vznikají četné národní a nadnárodní výzkumné aktivity, jejichž cílem je příčiny úbytku osvětlit.

Hlavním problémem může být směsice faktorů, jako je ztráta přirozených ekosystémů a monotónnost včelí pastvy či používání pesticidů v zemědělství a obecně znečištěné životní prostředí. K významným negativním faktorům patří také nevhodné postupy při ošetřování včelstev, jako je nepatřičná stimulace plodování včelstva podzimmím příkrmováním včelstev v průběhu podletí. V rámci projektu Rozmanitost života a zachování ekosystémů (ROZE), aktivita Zdraví včel, se zaměřuje na vyhodnocení vlivu intenzity plodování a síly včelstva na endokrinní regulaci a aktivitu telomerázy, jakožto markery biologického věku, a posouzení do jaké míry může pozdní příkrmování a nutriční skladba příkrmů vést k urychlenému stárnutí přezimující generace včel a kolapsům včelstev v zimním období.

Motivace vytvoření inteligentního včelího úlu vznikla v rámci realizace projektu Strategie AV 21, (ROZE). Tento projekt je finančně podpořený AV ČR a realizovaný Biologickým centrem AV ČR, v. v. i. v. Č. Budějovicích.

Cílem práce je navrhnout technické řešení pro sběr a analýzu dat včelího úlu. Sledovanými parametry včelstva v úle budou teplota, vlhkost a hmotnost úlu včetně včel. Bude se také pořizovat záznam zvuku vydávaného včelami uvnitř úlu pro možnost analýzy. Pro účely sběru dat a následné analýzy bude vytvořený potřebný software. Software bude také potřeba vytvořit pro komunikace mezi „inteligentním úlem“ a vzdáleným serverem. Server bude umístěn v univerzitní síti. Na serveru bude nutné vytvořit software pro zpracování a uložení příchozích dat.

Uložená data bude možné po zpracování zobrazit v internetové aplikaci, a to pomocí grafů tak, aby bylo možné sledovat vztahy mezi měřenými veličinami. Jak hardware, tak software musí být dobře zdokumentován pro další možné použití nebo rozšíření.

Celá práce bude rozdělena do třech hlavních částí. Teoretická část, praktická část a závěr. První část bude teoretická, zde si poukážeme na možnosti při vytváření inteligentního úlu, výběru základní desky, měřících senzorů, možností komunikací mezi inteligentním úlem a vzdáleným serverem. Do této části bude také zahrnuto uložení dat na serveru a možnosti jejich zobrazení.

V druhé části se řešitel bude věnovat výběru konkrétních součástí inteligentního úlu a jeho kompletací. V této části bude vysvětleno, proč byly vybrány právě tyto součástky. Bude popsána i komunikace mezi inteligentním úlem a vzdáleným serverem a zpracování dat na vzdáleném serveru.

Závěrem bude shrnut vývoj inteligentního úlu postupnou kompletací fyzických součástí a následujícím testováním použitých senzorů.

1.1 Trocha včelařské historie

Včelaření, doložené zápisky nebo poznámkami, sahá až do období 5000 let př.n.l. v zemích Blízkého východu. V této době byly včely chované v hliněných džbánech. Včelí produkt – med, byl nejrozšířenějším sladidlem v evropské kuchyni a náznaky vybírání medu jsou patrné už na jeskynních malbách v Pavoučí jeskyni s odhadovaným stářím 12 tisíc let.

Moderní včelaření je v českých zemích datováno od počátku 19. století, kdy se začínají zakládat včelařské spolky. Taky začíná převažovat chov včel v přenosných úlech nad tzv. brtnictvím, tedy chovu včel ve vykotlaných kmenech stromů. Včelaři v těchto dobách byli zbaveni břemen jako jsou daně nebo nevolnictví, nicméně v těchto dobách se včelaři stávali především kněží. Ti chovali včely v klátech nebo slaměných koších, a to s sebou neslo nevýhody, zničení včelího díla při každém nahlédnutí do včelstva. Švýcar Francois Huber hledal řešení a zkonstruoval rámkový úl, díky kterému se při náhledu do včel neničilo včelí dílo a dalo se se včelami lépe manipulovat.

19. století bylo přelomové ve vícero věcech, byl vynalezen stojanový rámkový úl, ve kterém byly rámkové v míře 39 x 24 cm a byla zhotovena umělá vosková včelí mezistěna.

Ta napomohla při zakládání nových rámků, včely si mohly vytvořit své bezchybné dílo přesně tam, kde ho včelař vytvořit potřeboval a předešlo se tak nechtěným „divočinám“ v úle.

Významnou osobností včelaření v Čechách byl František Adamec, který se zasloužil o prosazení jednotné rámkové míry 39 x 24 cm a Langstrothova systému, přístupu do úlu shora. [1][2]

1.2 Význam včelaření pro člověka

Většina lidí si myslí, že hlavní přínosem a zároveň i produktem pro lidstvo od včel je med. To ale není pravda, protože největším přínosem včel je opylování rostlin, které jsou většinou na opylení závislé pro další produkci semen, aby se mohly rozmnožovat a zároveň plodit. Včelstvo tedy dokáže zvýšit výnos rostlin více jak stonásobně.

Pro zajímavost, v Čechách je téměř na každé vesnici jeden včelař, čímž se dostatečně pokryje opylování zemědělské úrody. Hlavním výnosem se pro včelaře tedy stává produkce medu, ale třeba v Dánsku si opylování lidé váží až padesátkrát více než hodnoty medu.

V Kalifornii je dokonce opylování hlavním zdrojem příjmů včelaře, protože je tam včelstev málo a majitelé plantáží platí nemalé peníze za umístění včelího úlu blízko mandloňových sadů. [1]

1.3 Metodika

Níže jsou uvedené body, které jsou podmínkou pro splnění stanovených cílů. Od rešerše stávajících řešení přes teoretický návrh až po samotnou konstrukci inteligentního úlu.

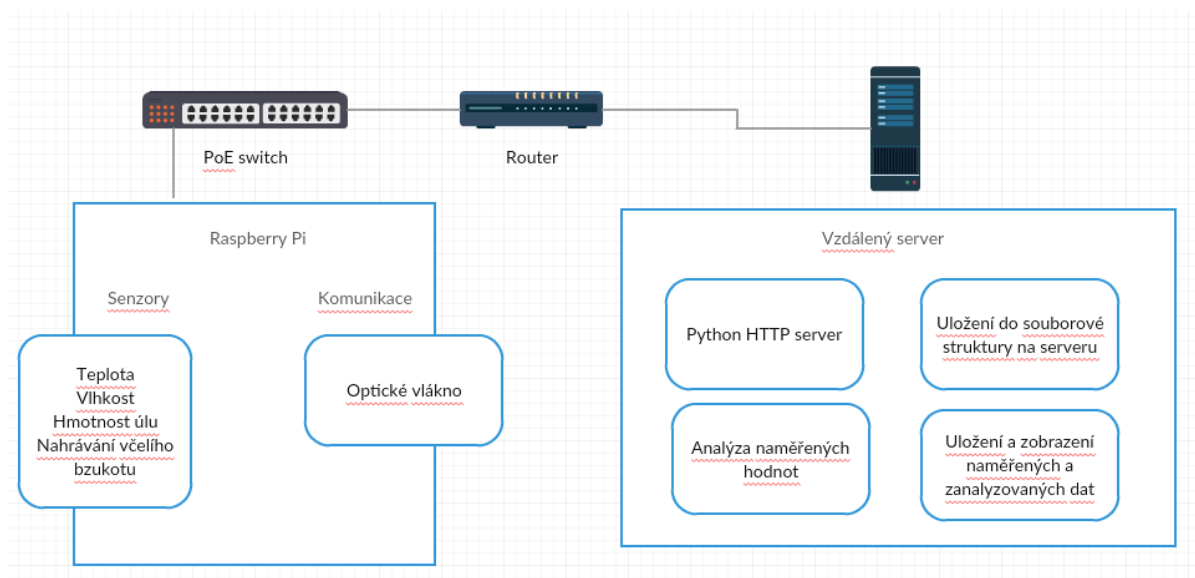
1. Vyhledat dostupné metodiky a jejich řešení.
2. Vybrat metodiku vyhovující nejvíce zadaným potřebám.
3. Vybrat hardware v odpovídající cenové a kvalitativní hladině.
4. Zkompletovat hardware společně s úlem.
5. Vytvořit software pro záznam dat a analýzu
6. Zhodnotit výsledky.

Teoretická část práce

Tato kapitola se věnuje návrhu architektury inteligentního úlu. V úvahu jsou brána možná řešení, založená na prvotní rešerši.

1.4 Návrh architektury

Architektura celého projektu je postavena na komunikaci základní desky se senzory pro záznam zvolených veličin a posléze odesláním sesbíraných dat na server pro další analýzu.



Obrázek 1 Návrh architektury

Senzory budou k základní desce připojené datovým kabelem a budou vždy čekat na výzvu základní desky pro odeslání momentální hodnoty dané veličiny. Senzor se tedy probudí pouze po dobu potřebnou pro záznam. Senzory budou komunikovat pouze se základní deskou, nebudou komunikovat mezi sebou a tímto odpadá nutná režie při komunikaci senzorů navzájem. Základní deska se tedy při zpracování scriptu bude ptát každého senzoru, jednoho po druhém, na záznam aktuální hodnoty dané veličiny.

Senzory budou měřit pouze teplotu a vlhkost, pro měření hmotnosti bude zakoupena váha a zvuk budeme měřit mikrofonom. Teplotní senzory se umístí jak vně, tak uvnitř úlu, a to pro možnost porovnání záznamů venkovních teplot společně s teplotou vnitřní, a také aktuální teplotou v přibližném středu chomáče včel. Vlhkost se bude také měřit jak venku,

tak uvnitř úlu, opět pro možnost porovnání hodnot. Poslední veličinou je hmotnost, a to celého úlu. Při počátku výzkumu je nutné nejdříve na váhu postavit samotný úl bez vnitřního osazení, abychom mohli poté odečítat hmotnost úlu od celkové hmotnosti úlu i s rojem včel a zaplněnými rámečky pylem nebo medem. Senzorů by bylo možné instalovat libovolný počet, omezením je pouze počet vstupních pinů na základní desce.

Takto tedy bude probíhat samotný záznam na základní desku, odkud se musí dostat na server. Pro tento přenos byl zvolen, díky poloze úlu v blízkosti univerzitní budovy, přenos pomocí optického kabelu rovnou na školní server.

Na serveru poběží script pro příjem dat, který bude data ukládat jak na diskovou jednotku, tak do zvolené databáze. Do databáze budou data ukládána až po aplikovaných analýzách, redukuje se tím objem dat ze “surové” podoby. U záznamu teploty, vlhkosti a hmotnosti je to jedno, protože tato data přijmeme jako reálná čísla a také je tak uložíme. Zvuk se bude ukládat jako *.wav soubor. To proto, aby bylo možné provést pozdější jiné analýzy. Data se tedy ukládají do standardního diskového úložiště.

Uložená data do databáze budou moci být pro lepší čitelnost zobrazena pomocí grafů.

1.5 Výběr základní desky

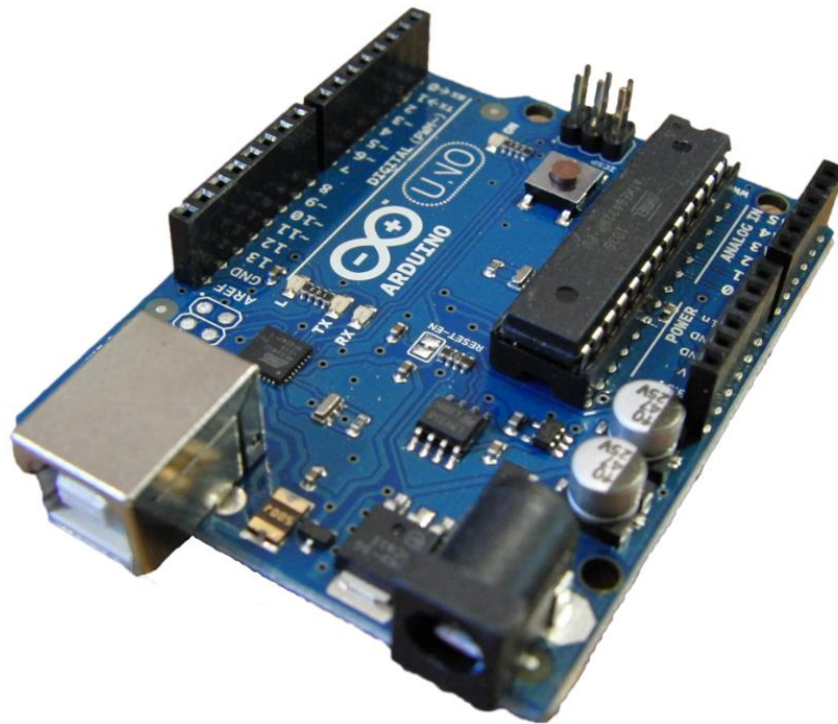
Základní deska, tedy primární funkční celek, byla vybírána z malých jednodeskových počítačů, které v posledních letech zažívají obrovský rozmach. Tyto malé počítače s jednou deskou plošných spojů byly vyvinuty pro podporu výuky. Jsou tedy dobré k seznámení studentů s principy řízení různých zařízení, od kuchyňských spotřebičů přes řízení dronů až po malé roboty.

Bylo vybíráno převážně z originálních produktů, tedy z desek Arduino a desek Raspberry Pi třetí generace.

1.5.1 Arduino

Projekt Arduino vznikl v Itálii a je pojmenovaný podle Arduina Ivrejského. Projekt je založený na filosofii open-source (otevřený-software), je tedy od základu dostupný pro všechny uživatele, kteří ho chtějí používat. Díky tomuto přístupu vzniklo také již plno klonů

originálního Arduina. Jedná se o desku založenou na mikrokontrolerech ATmega, které jsou vyráběné americkou firmou Atmel nyní Microchip.



Obrázek 2 Jednodeskový počítač Arduino UNO [4]

Hlavní mikrokontroler je uživatelsky programovatelný ve velmi podobném jazyce s C/C++. Programy tedy píšeme ve vývojovém prostředí na stolním PC a pomocí rozhraní USB poté výsledný program nahrajeme do mikrokontroleru.

Díky své architektuře mají malou spotřebu a lze je napájet pouze malou baterií. Arduino desky lze snadno rozšířit pomocí tzv. “shieldů”, tedy modulů, které se zapojí do standardních pinů Arduina. [3]

1.5.2 Raspberry Pi

Tyto malé jednodeskové počítače jsou již o trochu výkonnější než výše zmíněná Arduino. Dá se na ně již nahrát operační systém, a tím se stávají mnohem přístupnějšími. Ve své podstatě se jedná o velmi málo výkonný stolní počítač, obsahující jak HDMI výstup pro připojení monitoru, tak i USB vstupy, přes které můžeme připojit klávesnici a myš.

Raspberry Pi lze tedy použít jak pro interakci s moduly a senzory, tak také díky operačnímu systému pro samotný vývoj aplikací.



Obrázek 3 Raspberry Pi třetí generace verze B [5]

Pro Raspberry Pi byl vyvinut vlastní operační systém Raspbian, což je Linux založený na verzi Debian. Lze ale také nainstalovat některé verze Ubuntu nebo ArchLinuxu. Se svou verzí přispěl také Windows. Windows 10 IoT Core je nadstavbou základního Windows NT bez grafického uživatelského rozhraní. [5]

1.6 Komunikace

Vyřešení komunikace tohoto projektu bylo díky umístění včelnice v kampusu BC AV ČR, v.v.i. a JU jednoduché. Ke spojení včelnice a serveru bylo použito optické vlákno. Nicméně bych rád poukázal na jiné možnosti přenosu dat z inteligentního úlu i mimo dosah použití kabelu. Pokud budeme umisťovat úl do volné přírody musíme brát v úvahu, jak daleko bude od serveru nebo zařízení, který je schopné přijímat data ze senzorů inteligentního úlu.

Na základě této vzdálenosti bude vybrána nejvíce vyhovující technologie. Pokud je náš server, respektive zařízení přijímající data z inteligentního úlu v blízkosti včelnice, můžeme sáhnout po běžné wifi. Pokud se dostaneme ze vzdálenosti wifi, poslouží nám dobře síť ZigBee a z opravdu velkých vzdáleností musí být využita GSM síť.

1.6.1 Wifi

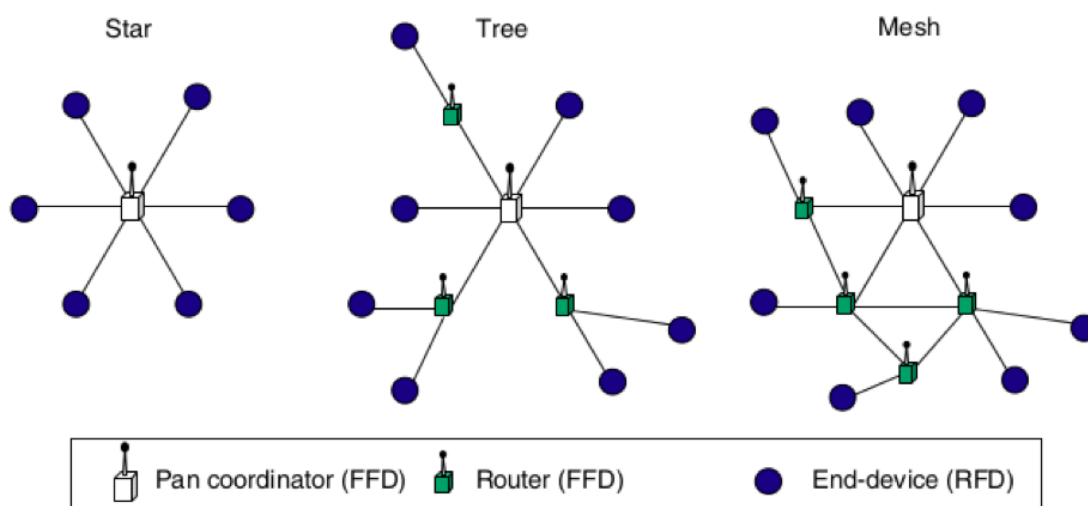
Wifi moduly jsou dostupné pro obě uvažované základní desky. Pro Raspberry Pi třetí generace je již wifi modul integrován přímo do základní desky. Pro starší modely lze wifi modul dokonce sehnat jako USB token. Instalace a používání jsou tedy opravdu jednoduché a postačí pouze doinstalovat správné ovladače a můžeme začít s wifi pracovat. K Arduinu je potřeba dokoupit samostatně prodáváný „shield“ a ten poté spojit s piny na základní desce Arduina. Poté moduly dokáží komunikovat na běžné wifi síti se standardem 802.11.x

1.6.2 ZigBee

Technologie ZigBee je vystavěná na standardu 802.15.4 a je určena pro spojení nízko výkonových zařízení v sítích při vzdálenostech do necelých osmdesáti metrů. Její primární účel tedy spadá do segmentu průmyslových a senzorových sítí. Standard je navržený tak, aby byl spolehlivý pro přenos malého objemu dat v rozsáhlejších bezdrátových sítích.

Dosah jednotlivých uzlů sítě (vysílač – přijímač) je standardně uváděn v rozpětí od 30 do 70 m, tento údaj, ale zpravidla platí pro komunikace ve volném prostoru bez překážek. Tato technologie je tedy nepoužitelná pro přenos naměřených dat z lesní mýtiny, ale dala by se použít při umístění úlu třeba na zahradě.

Z tohoto standardu vychází rodina modulů XBee. Tyto moduly jsou jednodušší a cenově dostupnější pro implementaci s deskami Arduino nebo Raspberry Pi. XBee zařízení mohou vystupovat ve třech módech, lišících se činností, jakou poté vykonávají.



Obrázek 4 Možné topologie sítě ZigBee [6]

Bílými body jsou na obrázku vyznačeny koordinátoři. Tento bod musí být v každé síti pouze jednou a musí být stále spuštěn, protože se stará o adresování v síti, umožňuje ostatním se do sítě připojovat a zajišťuje inicializaci sítě.

Zeleně jsou vyznačeny routery, těch může být od nuly až neomezeně. Podporují příjem a směrování zpráv. Routery také nemohou přejít do režimu spánku a musí být neustále bdělé, protože stejně jako koordinátoři dokáží uchovávat zprávy pro uspané koncové body.

Koncové body jsou vyznačené modrými tečkami a tyto body musí být vždy propojeny s koordinátorem buď přímo nebo přes router. Fungují jako sběrače dat, tedy pro odesílání a příjem zpráv. [6]

Architektura sítě ZigBee nám dovoluje díky přenosové rychlosti až 250 kbit/s využít i odesílání audio soborů.

1.6.3 GSM/GPRS

K řešení GSM/GPRS by se muselo přistoupit za předpokladu umístění úlu mimo obytné oblasti, to znamená někde na mýtinu u lesa nebo přímo do lesa. Dále je při tomto

řešení potřeba zakoupit sim-kartu od operátora, přes kterou budeme posílat SMS nebo využívat mobilní internet pro přenos dat na náš server.

Toto řešení vyžaduje pravidelnou investici v podobě placení paušálu operátorovi. Provozování takové sítě se tedy může v závislosti na zvoleném operátorovi dost prodražit.

Možným řešením jsou speciálně vyvíjené IoT sítě Sigfox nebo LoRa. Tato řešení jsou postavena na volném pásmu 868 MHz. SigFox využívá již existující pokrytí tuzemského operátora T-Mobile, kde pomocí signálu “ultra narrowband” doputují data ze senzoru až k nejbližší BTS operátora a poté do centrálního úložiště SigFox.

Zatímco SigFox využívá sítě operátorů, tak síť LoRa musíme vybudovat sami. Jinak sítě fungují shodně na volném pásmu 868 MHz. V obou případech těchto sítí platí malá omezení v komunikaci stanovené ČTÚ. [7]

Při použití tohoto řešení bychom museli počítat s maximálním objemem odesílaných dat do 12 bitů. Toto omezení by nám znemožnilo odesílat audio nahrávku. Někteří operátoři mají dokonce i denní limit pro maximální počet odeslaných zpráv.

1.7 Senzory

Senzory byly vybírány na základě důležitosti měření. Nejdůležitější jsou teplota, vlhkost, hmotnost úlu a také frekvence včelího bzučení.

1.7.1 Teplota, vlhkost

Na trhu jsou prodávány senzory DHT21/AM2301, což je digitální senzor teploty a vlhkosti. Můžeme tak jedním zařízením měřit dvě veličiny najednou a ušetřit si tím některé piny na základní desce.

Senzory DHT21/AM2301 mají ochranné plastové pouzdro. Toto plastové pouzdro má jemné žebrování, které by včely pravděpodobně během chvilky zalepily propolisem. Proto se senzor umístí do tzv. školkovací klíčky.



Obrázek 5 Senzor umístěný ve školovací kličce

Školovací klička se používá při “dochovu” včelích matek. Do kličky se umístí mateří kašička se včelí matkou. Poté se klička umístí do včelstva, aby se s ní včely dělnice měly možnost seznámit.

Školovací klička je na jedné straně opatřena jemným pletivem a na druhé plastovou mřížkou s malými kruhovými otvory, aby se včely nemohly dostat dovnitř. S velkou pravděpodobností včely propolisem nezalepí tu stranu opatřenou pletivem. Senzor se tedy umístí na tuto stranu a měření teploty a vlhkosti nebude tudíž zkresleno.

1.7.2 Hmotnost

Záznam o hmotnosti dává včelaři informace o včelstvu, a přitom neruší včely. Informace poslouží nejen pro pokyn ke sběru medových zásob, ale především jako dobrá informace o aktuální vitalitě včelstva.

Záznamy o hmotnosti však není jednoduché dešifrovat, protože jsou závislé na celé řadě dalších okolností. Průběžné sledování hmotnosti může včelaři poskytnout informace o tom, kolik včely za den přinesou nektaru a pylu. Záleží ovšem na tom, jak se včelstvo vyvíjí, a tedy kolik zásob je průběžně spotřebováváno na krmení mladých včel a kolik zásob zůstává.

Na hmotnosti se dá také pozorovat včelí aktivita, konkrétně kolik včel vylétá za potravou a jak vydatné naleziště včely navštěvují. Tyto údaje včelaři dobře poslouží při

přezimování včel, a to tak, že může v případě nedostatku zásob rychle zakročit a nedostačující zásoby doplnit cukernatým roztokem. Tento roztok si včely mohou zpracovat a tím zvětšit objem zásob pro přezimování. Dostatek zásob je důležitý pro dobrý rozvoj včelstva především z počátku sezóny.

1.7.3 Záznam zvuku

Po vyhodnocení možností, při kterých jsme museli brát v úvahu velikost mikrofonu i jeho zvukovou kvalitu, bylo rozhodnuto pro obyčejný klopový mikrofon. Ten můžeme snadno umístit uvnitř úlu a tím se dostat co nejbližší včelímu bzukotu, a také tím minimalizovat venkovní zvuky. Parametry vybíraného mikrofonu byly trochu upozaděny ve prospěch velikosti mikrofonu, ale mikrofon by měl mít dostatečnou citlivost na zvládnutí záznamu minimálně 100 Hz, protože včelí bzukot se bude pohybovat právě od 100 Hz až po jednotky kHz u druhu včel *A. mellifera*.



Obrázek 6 Klopový mikrofon Zalman ZM-MIC1 [8]

Záznam bude pořizován skrz USB zvukovou kartu, protože jak Arduino tak Raspberry Pi nemají vstup pro mikrofon. Záznam bude pořízen jako binární soubor, odeslán na server k dalšímu zpracování a poté bude finálně uložen jako *.wav soubor přímo do serverového úložiště.

1.8 Analýza včelího bzukotu

Pokus o monitorování včelího zvukového projevu je datován už v roce 1609, kdy Charles Butler vydal knihu *The Feminine Monarchie*, ve které bez pomoci nástrojů při popisu rojové nálady včelstva zaznamenává zvukovou charakteristiku notovým zápisem.

Zvukové projevy včelstva jsou slovně popisovány až do roku 1959, kdy po druhé světové válce došlo k rozvoji elektroniky. Roku 1959 vychází v časopise *Nature* přelomový článek “*Electronic Prediction of Swarming in Bees*” publikovaný E. F. Woodsem. Od té doby se ve včelařské komunitě začíná datovat pojem akustický monitoring.

Až Karl von Frisch, rakouský etolog a nositel Nobelovy ceny, zabývající se smyslovým vnímáním včel, zveřejnil roku 1965 dorozumívání včel pomocí důmyslných tanečků vyjadřujících jak polohu, tak také velikost vzdáleného zdroje potravy. [9]

Včely tedy vydávají v různých situacích různé zvuky. Ty slouží pro komunikaci v kolonii. Průměrná frekvenční hladina se pohybuje v rozmezí od 300 do 600 Hz. Včely si tak mohou předávat varovné signály nebo výzvy na sběr nektaru.

Nahráný zvuk včelího roje budeme na serveru analyzovat pomocí Python modulů Numpy a Scipy.io.

Níže se můžeme podívat na pozorované frekvence u včel.

Interpretace, vzorec chování	Typ signálu	Základní frekvence (Hz)	Přibližný tvar signálu
Zamezení líhnutí matek	“Troubení”	350 – 500	Sekvence pulzů
Informace o životaschopnosti další matky	“Kvákání”	200 – 350	Sekvence pulzů
Obrana / varování	“Syčení”	Od 3000	Širokopásmový puls
Rekrutování na donášku nektaru	“Pískání”	300 – 500	Sekvence pulzů
Vybídnutí roje k odletu	“Pískání”	100 – 2000	Sekvence pulzů
Upoutání pozornosti dělnic	“Pískání”	200 - 350	Puls

Tabulka 1 Tabulka včelích zvukových projevů [9]

1.9 Uchování naměřených hodnot

Naměřené hodnoty se průběžně ukládají do centrálního úložiště na serveru. Získané hodnoty z teplotních a vlhkostních senzorů a váhy přijdou na server již jako zformátovaný JSON, takto je tedy uložíme do textového souboru pojmenovaného jako datum pořízení záznamu. Tato data poté ještě odešleme do vybrané databáze pro možnost zobrazení. Zvukové soubory server přijímá jako bitový stream a ukládá je jako *.wav do úložiště. Takto je ukládáme pro možnost postprocessingu a tedy možnost použít jinou metodu získání frekvencí nebo pro zavedení jiného pokusu na těchto datech.

Uložená data budeme dále zobrazovat ve webovém rozhraní pomocí grafů. Pro možnost zobrazení se data budou ukládat ještě do jiného úložiště popsaného níže.

Pro ukládání dat bude využita dokumentová databáze místo klasické relační databáze, protože hlavním smyslem je uložená data indexovat pomocí časového údaje uložení, namísto jedinečného identifikátoru typického pro relační DB. Příklad ukládaného JSON dokumentu je v kapitole 3.6.

1.10 Zobrazení dat

Pro zobrazení dat je použito open source nástroje **Kibana**. Kibana vznikla jako grafická nadstavba pro dokumentovou databázi a fulltextový vyhledávač **ElasticSearch**.

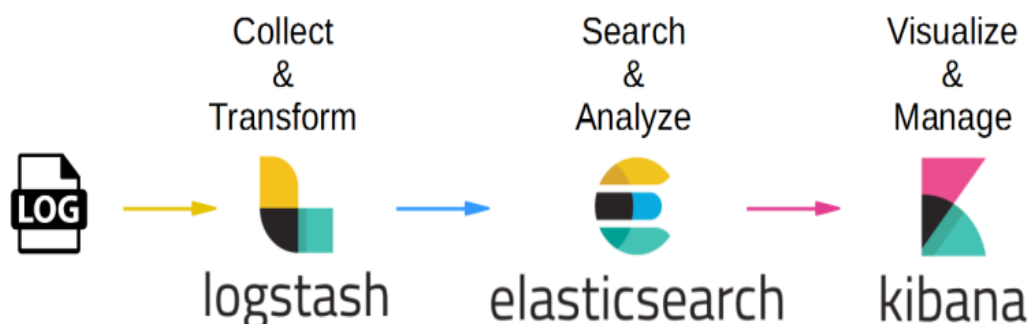
1.10.1 ElasticSearch

ElasticSearch je dokumentová databáze, do které se ukládají strukturované JSON dokumenty. Tyto dokumenty bývají při ukládání analyzovány pro potřebu vytvoření indexů k rychlému textovému vyhledávání. ElasticSearch je vyvíjen od roku 2010 jako vyhledávací engine postavený na knihovně Lucene, vyvíjenou společností Apache. ElasticSearch je vydáván pod open source licenci, funguje jako serverové řešení přístupné přes RESTfull rozhraní HTTP protokolu.[11]

Jeho hlavním úkolem mělo být vyhledávání v textových polích tzv. fulltextové vyhledávání. Pro svou škálovatelnost a rychlost vyhledávání téměř v reálném čase se uchytíl také jako nástroj pro zaznamenávání aplikačních a serverových logů.

Postupem času byl tedy k produktu ElasticSearch započat vývoj dalších produktů Logstash a Kibana. ElasticSearch slouží jako databáze logů. Tyto logy jsou sbírány pomocí produktu Logstash, který funguje jako vstupní bod pro různé systémové nebo aplikační logy. Na základě předepsaných pravidel umí příchozí logy zpracovat a vytvořit z nich dokumenty vhodné pro uložení v ElasticSeach. [10]

Ačkoliv jsou dokumenty snadno přístupné přes REST rozhraní, tak nejsou pro člověka moc dobře čitelné. Z tohoto důvodu byla vytvořena Kibana jako grafická nadstavba nad ElasticSearch. Společně se těmto produktům začalo říkat ELK stack.



Obrázek 7 ELK stack

Tyto produkty jsou tedy vhodné i pro ukládání strukturovaných dat jako jsou záznamy z teplotních senzorů.

1.11 Kibana

Jak je již zmíněno výše, Kibana je grafické rozhraní pro zobrazení dat uložených v ElasticSearch. Díky různorodosti grafů dokáže Kibana nejen veškerá data zobrazit v interaktivních grafech, ale hlavně nad nimi umožňuje další analýzy.

Přes Kibanu se dá díky grafické webové konzoli pohodlně ovládat nastavení úložiště ElasticSearch. Kibana také umožňuje díky integrované REST konzoli vytvářet dotazy přímo do ElasticSearch.

S každou novou verzí se produkt Kibana dostává o mílové kroky dále. S posledními verzemi byla zpřístupněna možnost tzv. “Machine Learning”. Tato funkčnost umí na základě analýzy, nad již uloženými daty, vyhodnotit nastalé anomálie, a tedy na ně upozornit uživatele. Poté dokáže do nějaké míry predikovat budoucí vývoj. Tato funkčnost je však bohužel jako jedna z mála funkčností zpoplatněná. [12]

Praktická část práce

Tato kapitola je věnována výběru samotného HW a jeho praktického použití. V kapitole jsou popsány kompletní funkční celky od komunikace se senzory až po vytvoření grafu na vzdáleném serveru.

1.12 Hardware

Ačkoliv je inteligentní úl umístěný pod zastřešenou pergolou v blízkosti akademického a univerzitního kampusu, jedná se o venkovní projekt a není tedy sám od sebe chráněn před rozmary počasí. Kompletní základní deska je umístěna do plastové elektroinstalační krabice a do ní jsou vyvrtány díry pro vodotěsné průchodky. Kvůli malým průměrům protahovaných kabelů jsou zvoleny co nejmenší průchodky, aby se nemusely utěšňovat více než je potřeba. Krabice je uchycena vruty na zadní straně nejspodnějšího nástavku inteligentního úlu.



Obrázek 8 Uchycení plastové krabice na nástavku úlu

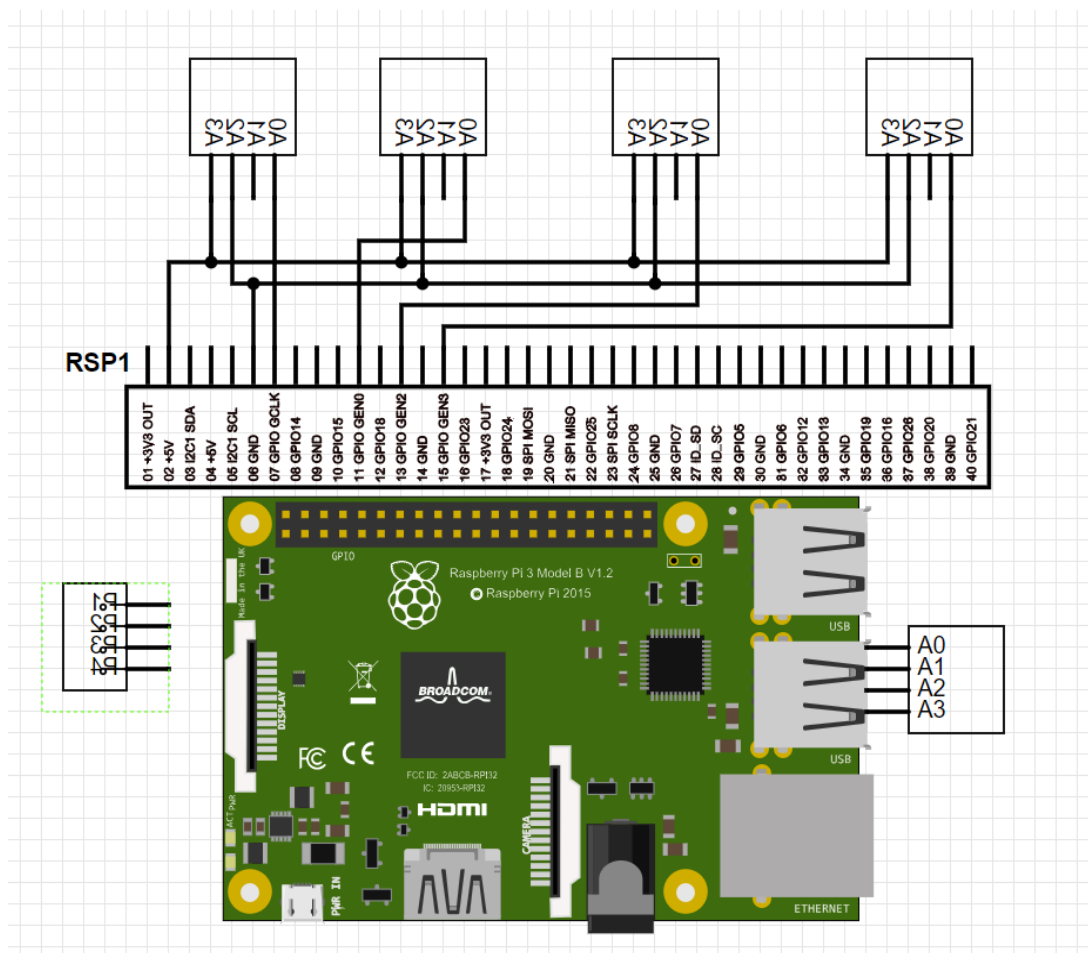
Jak můžeme vidět na obrázku, svou plastovou krabicí dostala i programová jednotka váhy. Na spodní straně krabic jsou viditelné přimontované průchodky, jimiž jsou vytaženy

teplotní a vlhkostní senzory umístěné v úle. Svou průchodku má také mikrofon pro nahrávání včelího bzučení. Na spodní stranu plastové krabice byl také umístěn senzor DHT21/AM2301, měřící venkovní teplotu a vlhkost.

Kvůli časté manipulaci a také usnadnění přístupu do úlu, byly krabice umístěny na zadní stranu nejnižšího nástavku úlu. Tento nástavek je základový a vůbec se s ním po čas roku nehýbe, předcházíme tím možnosti poničení elektroniky.

1.12.1 Základní deska

Srdcem celého projektu je základní deska, u které bylo vybíráno mezi produkty z řady Arduino a Raspberry Pi. Vzhledem ke svému přednostem bylo rozhodnuto pro základní desku Raspberry Pi. Tato deska je díky použití optimalizovaného operačního systému takřka plnohodnotným počítačem zvládajícím i náročnější výpočetní operace.



Obrázek 9 Schéma tištěného spoje základní desky

Základní deska Raspberry Pi usnadňuje práci hlavně při implementaci komunikace mezi inteligentním úlem a serverovým úložištěm. Raspberry Pi má již v základu zabudovaný ethernetový RJ45 konektor pro připojení datového kabelu. Přes tento konektor může být tedy základní deska jednoduše připojena do sítě.

Základní deska je umístěna na plošném spoji. Piny ze základní desky jsou pomocí čtyřiceti-žilového kabelu vyvedeny na plošný spoj. Na ten byly také umístěny čtyři svorkovnice pro připojení kabelů ze senzorů a posléze byla přidána ještě pátá svorkovnice pro připojení datového kabelu z váhy za účelem strojového čtení hmotnosti. Z této svorkovnice jsou data následně převedena přes adaptér na rozhraní USB. Na plošný spoj byl vyveden samostatný USB konektor pro napájení váhy.

Svorkovnice jsou umístěné na kraji plošného spoje. Se základní deskou jsou spojeny přes prepět'ové ochrany vytvořené z transilů a kondenzátorů. Pro správnou funkčnost senzorů je potřeba připojení opatřit také odpory, ty jsou připojené mezi napětí a signál.

Popis použití	Pořadí svorkovnice	Připojený GPIO pin
Venkovní teploměr umístěný na spodní straně ochranné krabice	Čtvrtá svorkovnice	GPIO 22
Teploměr s měřením vlhkosti umístěný v rohu uvnitř úlu	Třetí svorkovnice	GPIO 27
Teploměr uvnitř úlu co nejbliže středu chomáče včel	Druhá svorkovnice	GPIO 17
Prozatím nevyužitý	První svorkovnice	GPIO 4

Tabulka 2 Tabulky zapojení senzorů na piny základní desky

1.12.2 Hardwarové vybavení základní desky

V březnu roku 2018 byla vydána zatím nejnovější verze základní desky Raspberry Pi třetí generace s frekvencí procesoru 1.4 GHz. Na desce je integrovaná podpora wifi a bluetooth. Tato verze desky, po zakoupení přídatného modulu (PeE HAT), také podporuje

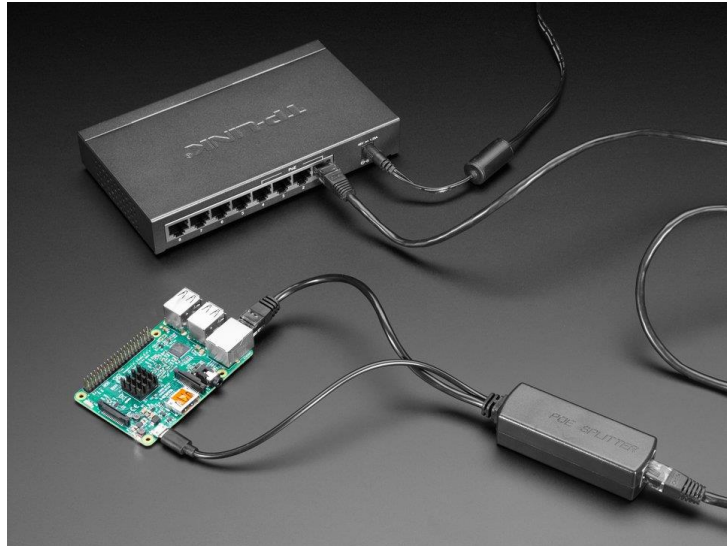
PoE (Power over Ethernet). Po přidání tohoto modulu může být deska napájena přes napětí vedené ethernetovým kabelem, v rozpětí 37 – 57V. PoE funguje na principu využití volných párů datového kabelu pro napájení. Základní deska má také čtyři USB vstupy a mimo jiné také zvukový výstup pro sluchátka nebo malé reproduktory. Na desce se nachází 17x GPIO pinů pro připojení vnějších periférií a výstupy pro 5V a 3V napájení. [5]

1.13 Komunikace

Jak je již popsáno výše, inteligentní úl je umístěn na včelnici v blízkosti akademických a univerzitních budov. To nám umožnilo použít pro komunikace mezi úlem a serverem optické vlákno a komunikace se tak díky tomu velmi zjednodušila.

Na včelnici, která je krytá přístřeškem, je upevněn rack obsahující router. Do routeru umístěném v racku je zavedeno optické vlákno vedené z budovy BC AV ČR. Uvnitř racku je také PoE switch, díky kterému bude umožněno spojení úlů s routerem. Protože je inteligentní úl umístěn také na pozemku školy, musí se řídit ČSN. Veškeré zásuvky a kabelové vedení musí být pro venkovní použití s odpovídajícím krytím. Nelze tedy jednoduše zapojit adaptér pro napájení do rozdvojky na včelnici a přivést tak energii k základní desce u inteligentního úlu. S ohledem na toto prostředí používáme PoE jako zdroj bezpečného malého napětí (48 V) pro napájení, které nevyžaduje revizní zprávu vystavenou revizním technikem.

Myšlenka je tedy taková, že od racku z PoE switche půjde ke každému úlu pouze jeden ethernetový kabel. Tím je tedy vyřešeno jak datové spojení, tak i napájení inteligentního úlu díky PoE. K Raspberry Pi lze zakoupit přídatný modul s možností připojení PoE rovnou k základní desce a umožnit tak napájení. Jak již bylo řečeno je tento modul velmi drahý, a proto bylo rozhodnuto o použití obyčejného PoE splitteru. Ten rozdělí energii a data z příchozího ethernetového kabelu do napájecího miniUSB a ethernetového kabelu pokračujícího do přípojky na základní desce.



Obrázek 10 Ilustrační foto zapojení Raspberry Pi pomocí PoE splitteru a PoE switche

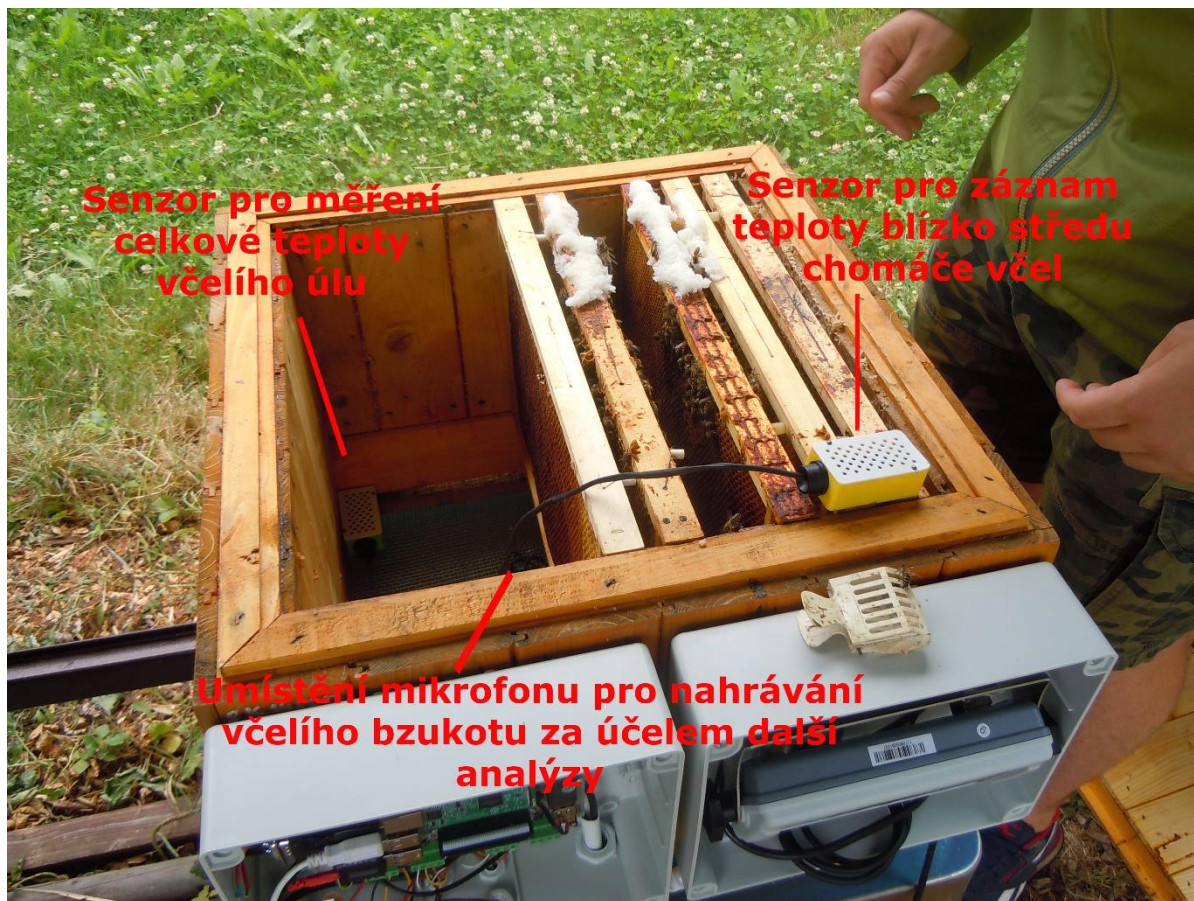
Naměřená data budeme poté odesílat pomocí Python modulu **requests**. Ten podporuje běžné HTTP metody jako jsou GET, POST, PUT, atd. V našem případě budeme pro odesílání dat na vzdálený server využívat metodu POST. Do hlavičky této metody musíme kvůli zpracování na serveru umístit parametr **content-type : application/json**, jinak bychom na serveru nepoznali, že se jedná o data ze senzorů. [14]

```
requests.post("http://160.217.213.136:9500", headers={'content-type' : 'application/json'},  
             data = json)
```

Blok kódu 1 Odeslání dat pomocí modulu requests

Ukázka odeslání dat na vzdálený server pomocí modulu **requests**. Popis scriptu měřící hodnoty ze senzorů bude popsán v kapitole Senzory.

1.14 Senzory



Obrázek 11 Fotka osazeného včelího úlu

Na obrázku můžeme vidět osazení inteligentního úlu senzory. Jak je vidět úl má dřevěnou konstrukci a aby mohly být senzory umístěny uvnitř, musela být do úlu vyvrtána díra, která byla po protažení kabelů utěsněna. Dále si na obrázku všimneme zelené klíčky umístěné v levém dolním rohu úlu, tedy senzoru pro měření teploty a vlhkosti bez „obležení“ včel. Uprostřed na dně je vidět položený mikrofón. Druhý vnitřní teplotní senzor v tuto chvíli ještě můžeme vidět položený na horních loučkách, ale můžeme si všimnout připravené mezery mezi rámkami, kam tento senzor ve žluté klíčce umístíme. Na obrázku je také vidět, že rámkami jsou již v tuto chvíli již obsazeny včelami.

1.14.1 Měření teploty a vlhkosti

Pro měření vlhkosti a teploty byly použity dva senzory DHT21/AM2301. První byl umístěn venku, na spodní stranu krabice se základní deskou, abychom mohli porovnat venkovní teplotu vůči teplotě uvnitř úlu a také teplotě uvnitř chomáče včel. Díky měření

vlhkosti můžeme také poznat, zda venku právě prší nebo ne. Tento údaj ale oceníme spíše za letních dnů, protože během podzimu senzor téměř stále ukazuje stoprocentní vlhkost.



Obrázek 12 DHT21/AM2301 Digitální senzor teploty a vlhkosti v pouzdru

Druhý senzor je umístěn uvnitř úlu, v rohu u zadní stěny. Umístěním senzoru co nejdále od včelího chomáče budeme moci měřit celkovou teplotu v úlu. Díky tomuto senzoru budeme moci dobře vidět, že včely nevytápějí celý prostor úlu, ale svými tělíčky zahřívají jenom tu oblast, kde se právě soustředí. V zimních měsících to bude s největší pravděpodobností v blízkosti včelí královny. Senzor umístěný uvnitř úlu měří stejně jako venkovní senzor také vlhkost.

Oba dva senzory jsou k desce připojené shodně, pomocí sběrnice One Wire. Přenos dat po této sběrnici však probíhá nestandardně. Jedná se o interní jednorúčelovou sběrnici předávající naměřená data senzoru ke svému nadřazenému obvodu. Pro čtení z těchto senzorů po sběrnici One Wire slouží knihovna `Adafruit_Python_DHT`. Po instalaci této knihovny lze využít její rozhraní takto: [15]

```
Humidity17, temperature17 = Adafruit_DHT_read_retry(Adafruit_DHT.AM2302, 17)
```

Blok kódu 2 Čtení ze senzoru DHT21/AM2301 pomocí knihovny Adafruit

Na příkladu můžeme vidět získání naměřených hodnot teploty a vlhkosti ze senzoru. Vstupními parametry jsou typ senzoru a GPIO pin na který je tento senzor připojen.

Nyní byly představeny teplotní senzory měřící zároveň také vlhkost ovzduší. Data z obou těchto senzorů jsou vyčítána pomocí knihovny Adafruit.

Projekt však obsahuje ještě další dva teplotní senzory DS18B20. Tyto senzory by měly být schopné měřit teplotu v rozmezí od -55 do + 128 stupňů Celsia. Jeden z nich je umístěn na sběrnici One Wire přímo u základní desky. Teploměr měří teplotu v bezprostřední blízkosti desky. Teplota je důležitá pro kontrolu a také pro vyhodnocení, zda se v budoucnu rozhodnout pro zavedení odvětrávání či naopak utěsnění ochranné krabice se základní deskou.

Poslední teplotní senzor měl být původně také typu DHT21/AM2301. Senzor měl být umístěn přímo uprostřed chomáče včel a měřit teplotu s vlhkostí. Tento senzor je však díky ochrannému pouzdru příliš velký, při zavedení do včelího chomáče působil nepřírozeně a mohl mít tak nepříznivý vliv na „pohodu“ včelstva. Proto byl nakonec zvolen teplotní senzor DS18B20.



Obrázek 13 Teplotní senzor DS18B20

Tento senzor je umístěn v nerezovém obalu na dva metry dlouhém kabelu a tím je plně vodotěsný. Jeho tělo je dlouhé asi 6 cm a tloušťka odpovídá obyčejné tužce. Hodí se tedy výborně pro snímání teploty přímo uvnitř chomáče a pro včely není příliš rušivý. Tento teploměr je také připojen na sběrnici One Wire.

One Wire je sběrnice, která se využívá pro přenos informací pouze jednoho vodiče, který může zároveň sloužit i pro napájení. Rychlost komunikace na sběrnici One Wire je 16 kbit/s, při zapojení overdrive může komunikovat až rychlostí 128 kbit/s.

Pokud chceme senzor zapojit na sběrnici One Wire, musíme nejdříve provést minimální nastavení v operačním systému desky. Do souboru **/boot/config.txt** zapíšeme **dtoverlay=w1-gpio gpioin=4**. Pokud bychom nspecifikovali, že chceme One Wire zapnout na gpiopinu čtyři, zapnula by se nám sběrnice One Wire pro všechny gpiopiny na desce. Abychom mohli sběrnici One Wire využívat po každém zapnutí Raspberry Pi, je nutné do spouštěcího scriptu (/etc/modules) přidat následující příkazy.

- modprobe w1-gpio
- modprobe w1-therm

Po zapsání příkazů je nutné desku restartovat.

Po restartu základní desky můžeme vidět zapojené senzory v adresáři **/sys/bus/w1/devices**. Zde jsou symlinky na adresáře s daty daných zařízení. Takto tedy můžeme jednoduše číst teplotní údaje přes sběrnici One Wire.

Pomocí příkazu **crontab -e** se dostaneme do editace záznamů tabulky pro pravidelně spouštěné scripty.

```
*/3 * * * * ~/scripts/data-reader.py
```

Blok kódu 3 Příklad zápisu do cron tabulky

Záznam teplot a vlhkostí provádí cronem spouštěný Python script data-reader.py. Ten načte záznamy jak pomocí knihovny Adafruit, tak i hodnoty ze senzorů připojených přes sběrnici One Wire. Načtení hodnot ze senzorů měřících teplotu a vlhkost je popsáno výše v **Bloku kódu 2**, za použití knihovny Adafruit_DHT.

Zajímavější je ovšem následující čtení teploty ze sběrnice One Wire popsané níže.

```
base_dir = '/sys/bus/w1/devices/10-0008031ef64a/w1_slave'  
def read_temp_raw():  
    f = open(base_dir, 'r')  
    lines = f.readlines()  
    f.close()  
    return lines
```

Blok kódu 4 Načtení řádků se záznamem teploty ze sběrnice One Wire

Výše můžeme vidět pomocnou funkci pro načtení kompletního záznamu s měřenou teplotou ze senzoru na sběrnici One Wire konkrétního zařízení s označením 10-0008031ef64a.

```
7f 01 4b 46 7f ff 0c 10 ba : crc=ba YES  
7f 01 4b 46 7f ff 0c 10 ba t=23937
```

Blok kódu 5 Ukázka načteného záznamu s teplotou ze sběrnice One Wire

Jak je vidět na ukázce, načtená data ze sběrnice One Wire neobsahují pouze teplotní údaj, ale kompletní výstup z teplotního senzoru. Tento výstup je nutné ještě zpracovat a získat pouze teplotní hodnotu. K tomuto nám pomůže následující funkce níže.

```

def read_temp():
    lines = read_temp_raw()
    # Kontrola zda jsou poslední 3 znaky 'YES'.
    while lines[0].strip()[-3:] != 'YES':
        time.sleep(0.2)
        lines = read_temp_raw()
    # Hledání počátku znaku 't='
    equals_pos = lines[1].find('t=')
    if equals_pos != -1:
        # Read the temperature .
        temp_string = lines[1][equals_pos + 2:]
        temp_c = float(temp_string) / 1000.0
        temp_f = temp_c * 9.0 / 5.0 + 32.0
        return temp_c, temp_f

```

Blok kódu 6 Vyseparování naměřené teploty z kompletního výstupu senzoru

Hned na prvním řádku funkce si můžeme všimnout volání předešlé funkce pro získání výstupu z teplotního senzoru. Po načtení sensorových dat zkontrolujeme poslední tři znaky na první řádce. Pokud měření proběhlo v pořádku bude řádka zakončená třemi znaky YES. Pokud ne, zkusíme měřit znovu. Po kladném změření hledáme počátek řetězce `t=`, který značí začátek teplotní hodnoty, tu musíme ještě vydělit tisícem, abychom dostali hodnotu teploty Celsia. Script po načtení teplotních a vlhkostních hodnot také načte hmotnost z připojené váhy, tento blok kódu je popsán dále.

1.14.2 Zvuk

Jak již bylo zmíněno v teoretické části, pro úsporu místa bylo zvoleno malého klopového mikrofону. Tento mikrofon se jednoduše vejde do úlu, bude moci být umístěn poblíž středu chomáče včel a svou velikostí nebude včely rušit. Jako prvotní investice byl zvolen levný Zalman ZM-MIC1. Jedná se o obyčejný klopový mikrofon se všesměrovou charakteristikou snímání zvuku, a proto je dobře uplatnitelný pro snímání včelího bzukotu.

Protože Raspberry Pi v základu neobsahuje vstupní jack pro připojení mikrofónu, musela být zakoupena Mini USB zvuková karta Axagon ADA-17 mini HQ. Je to externí

zvuková karta kompatibilní se všemi operačními systémy a snadno se zapojí přes USB vstup. Tato karta nepotřebuje žádné zvláštní nastavení.

Pro záznam zvuku byl zvolen Python modul **pyAudio**. Jako výchozí frekvence nahrávání záznamů bylo kvůli úspoře místa na serveru zvoleno rozmezí 5 minut s délkou nahrávání 10 sec. Výchozí hodnoty nahrávání jsou:

- počet záznamů za sekundu - 8000
- délka nahrávky - 10 sec
- počet nahrávajících kanálů - 1
- formát nahrávání zvuku - 16 bitů

Periodicitu spouštění scriptu pro nahrávání zajistí cron operačního systému. Na základě výše uvedených parametrů se pomocí modulu **pyAudio** vytvoří bitový stream, ze kterého se přečtou audio data. Tato data se poté uloží jako dočasný soubor, který se vzápětí znovu načte a odešle se jako bitový stream na server. Tato data se musí odesílat s parametrem hlavičky **content-type – application/octet-stream**. [16]

1.14.3 Váha

Při výběru váhy bylo upřednostněno komerční řešení před vlastním. Důvody tohoto rozhodnutí jsou:

1. Rychlost dodání funkčního zařízení schopného vážit.
2. Zakoupená váha je zkalibrována a na naměřené údaje se můžeme spolehnout.
3. Konstrukce váhy je z lakovaného svařeného železa a vážní plocha je nerezová. Dá se tedy bez problému umístit ve venkovním prostředí.
4. Zakoupená váha je zkonstruována pro dlouhodobé zatížení a nebude tedy po krátké době docházet k nepřesnostem v měření.

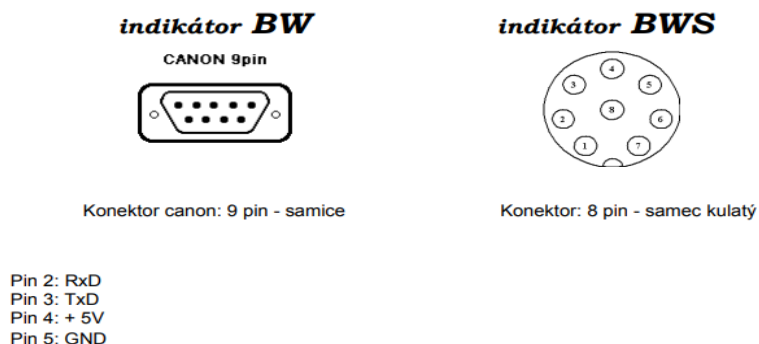


Obrázek 14 Včelařský vážící můstek [17]

Konkrétně byla pořízena průmyslová váha LESAK 1T6060LN-RWP, která je spolehlivá, přesně vážící a určená přímo pro včelaře. Váha je tedy ihned po rozbalení připravená vážit bez dodatečných kalibrací. Vzhledem k povaze váženého úlu byla zvolena váha s maximální vážnou hodnotou do 300 kg o rozměrech 60 x 60 cm. Součástí váhy je i A/D převodník s LCD displejem umístěný v plastovém pouzdru. Z váhy jde analogový

signál do A/D převodníku, který jej převede na digitální. Naměřená hodnota se poté zobrazí na LCD displeji.

Strojové vyčítání dat je umožněno přes sériový port RS232. Port je vyveden z A/D převodníku a umožňuje komunikaci s váhou. Tento sériový port využívá pouze 4 piny z 8, můžeme ho tedy snadno díky adaptéru dostat do USB portu.



Obrázek 15 Blokové schéma RS232 konektoru

Při snaze mít co nejmenší průchodky bylo rozhodnuto původní konektor odstranit a kabel připojit přímo k základní desce za použití redukce RS232 -> USB. Data z takto zapojené váhy můžeme číst jako sériové rozhraní, tedy Python modulem **serial**. Níže můžeme vidět popis spojení se sériovým portem pro čtení hmotnosti pořízené váhou.

```
serial.Serial(  
    port='/dev/ttyUSB0',  
    baudrate=9600,  
    parity=serial.PARITY_NONE,  
    stopbits=serial.STOPBITS_ONE,  
    bytesize=serial.EIGHTBITS,  
    timeout=1)
```

Blok kódu 7 Připojení sériového portu pomocí modulu serial

US,GS, 76.64kg

Blok kódu 8 Výsledek čtení sériového USB portu

Výsledkem čtení ze sériového portu je řetězec popsany v dokumentaci dodávané s váhou.

```

def read_weight():
    sb = []
    while 1:
        sb.append(ser.read())
        if ''.join(sb).endswith("kg"):
            return ''.join(sb[len(sb) - 8:-2])

```

Blok kódu 9 Čtení hmotnosti ve scriptu data-reader.py

Aby byl získaný řetězec parsovatelný jako reálné číslo, je potřeba ho oříznout, tedy z něj odstranit US,GS. Takto je naměřená hmotnost z váhy přidána do JSON a společně s teplotami a vlhkostmi odeslána na server. Hmotnost je měřena pomocí scriptu současně s teplotou a vlhkostí, aby mohly být tyto hodnoty odeslané společně na server.

Samotná váha funguje buď na integrovanou baterii a může se dobíjet pomocí adaptéru 12V/500mA anebo může být na tento adaptér stále připojená. Jelikož byl kladen důraz na to, aby k inteligentnímu úlu vedl pouze jeden kabel, a to ethernetový s podporou PoE, nemohl být adaptér použit. Místo něj byl zakoupen USB měnič, kterým se z USB na základní desce přivádí energie k váze.

1.15 Zpracování přijatých dat

Pro zpracování dat je použit server pro příjem GET a POST požadavků. Jedná se o Python script pojmenovaný vcely-server.py, který se spouští v nekonečné smyčce a může být přerušen pouze vstupem uživatele. Použité moduly na příjem HTTP požadavků jsou **HTTPServer** a **BaseHTTPRequestHandler**.

1.15.1 HTTPServer

Použitím tohoto modulu se vytvoří instance serveru, tedy HTTP socketu pro příjem requestů. Jedná se o potomka modulu TCPServer. Instanci serveru vytvoříme pomocí IP adresy, portu a třídy request handleru. Server se poté spustí ve smyčce příkazem uvedeným níže, který zobrazuje inicializaci instance serveru. Po spuštění je server ihned aktivní na zvolené adrese. [18]

```
httpd = server_class(server_address, handler_class)
http.server_forever()
```

Blok kódu 10 Spuštění HTTPServeru

1.15.2 BaseHTTPRequestHandler

Modul BaseHTTPRequestHandler pouze zpracovává příchozí requesty a odpovídá na ně. Nemůže sám vytvářet žádné HTTP požadavky jdoucí ze serveru. Příchozí zpráva se zpracuje buď jako GET, POST, atd. požadavek, podle příchozí HTTP metody requestu. Pro každou žádost typu GET, POST, atd. se musí vytvořit separátní metoda pro zpracování žádosti. Tyto metody mají prefix **do_** tedy **do_GET**.

```
def do_GET(self):
    logging.info("GET request,\nPath: %s\nHeaders:\n%s\n",
str(self.path), str(self.headers))
    self._set_response()
    self.wfile.write("Hello vcely-server is alive")
```

Blok kódu 11 Ukázka zápisu funkce pro zpracování GET request

Ukázka kódu pro zpracování GET requestu. Při zpracování se pomocí logovacího modulu Logging vypíše do terminálu URL cesta requestu a hlavičky odeslané s tímto requestem. V závěru další implementovanou funkcí nastaví do response hlavičky kód 200 (Standartní odpověď pro úspěšný HTTP požadavek) a parametry Content-type nastaví jako text/html. Posledním řádkem nastaví do těla response text, že včelí server pro zpracování dat běží. Jedná se tedy o „HEALTH check“ status serveru.

Data ze senzorů jsou všechna odesílána jako HTTP metoda POST, mají tedy vyplněné tělo zprávy. Musíme mít implementovanou metodu **do_POST**, kde se budou již zpracovávat obě příchozí zprávy. Tedy zpráva s daty o teplotě, vlhkosti a hmotnosti anebo zpráva se zvukovou stopou.

```

def do_POST(self):
    logging.info("POST request,\nHeaders:\n%s\n\n",
                 str(self.headers['content-type']))
    if str(self.headers['content-type']) ==
"application/octet-stream":
        self.do_octet()
    else:
        self.do_json()

```

Blok kódu 12 Zpracování POST požadavku na HTTP serveru

Tyto dvě různé zprávy od sebe dokážeme oddělit na základě hlavičky typu **Content-type**. Zpráva s daty ze senzorů bude mít **content-type: application/json** a zpráva se zvukovou stopou bude přijímána jako **content-type: application/octet-stream**. Na základě parametru hlavičky content-type metoda rozhodne, jak se požadavek zpracuje, protože jsou oba HTTP POST. [18]

1.15.3 Application/octet-stream

V HTTP světě zprávy obsahují binární data označována jako application/octet-stream. Po příjmu takovéto zprávy získáme nejdříve délku přijatých binárních dat, poté za pomoci této délky načteme všechna přijatá binární data z těla requestu. Binární data poté uložíme do nově vytvořeného souboru, jehož název bude mít formát aktuálního času přijetí zvukové nahrávky.


```

def do_octet(self):
    logging.info("Start process octet-stream")
    content_length = int(self.headers['Content-Length'])
    post_data = self.rfile.read(content_length)
    logging.info("data length = %s",
str(content_length))
    file_name =
"/home/pi/audio/{}.wav".format(datetime.now().strftime('%
y-%m-%d_%H:%M:%S'))
    with open(file_name, "wb") as f:
        f.write(post_data)
        f.close()

```

Blok kódu 13 Zpracování přijatých bitových dat a vytvoření audio souboru s datumovým názvem souboru

Takto ho poté uložíme do serverové složky. Uložený zvukový záznam bude nejen lépe dohledatelný, ale především budou moci být na jeho základě spouštěny další analýzy.

1.15.4 Application/json

Jak již samotný název napovídá, jedná se o hlavičku pro zprávy s obsahem typu JSON. Tělo této zprávy obsahuje naměřené hodnoty ze senzorů ve formátu JSON. Přijatý JSON vypadá takto:

```

{
"time" : "2018-12-11T01:30:09",
"zluty" : {
    "temp" : "21.25",
    "hum" : "0"
},
"venek" : {
    "temp" : "8.39999961853",
    "hum" : "99.9000015259"
},
"zeleny" : {
    "temp" : "11.6999998093",

```

```

        "hum" : "87.3000030518"
    },
    "deska" : {
        "temp" : "10.437",
        "hum" : "0"
    },
    "vaha" : {
        "kg" : " 37.08"
    }
}

```

Blok kódu 14 Přijatý JSON s daty ze senzorů

Popisky “zluty”, “venek”, “zeleny” a “deska” jsou poznávací názvy pro jednotlivé senzory zaznamenávající teplotu a vlhkost popsané v tabulce 2.

- “Zluty” je DS18B20, tento senzor měří pouze teplotu z prostředí přímo uprostřed včelího chomáče.
- Senzor označený “venek” je umístěn na spodní venkovní straně plastové ochranné krabice základní desky.
- “Zeleny” je senzor teploty a vlhkosti umístěny uvnitř úlu. Názvy zeleny a zluty vznikly díky barvám matečních klíček do kterých byly tyto senzory umístěny, aby nebyly zaneseny včelím propolisem.
- “Deska” je senzor teploty na základní desce, ten je také bez vlhkosti.
- “Vaha” znamená hmotnost včelího úlu udávaná v kilogramech.

U teplotních senzorů „zluty“ a „deska“ byl ponechán údaj o vlhkosti. Tato hodnota je nyní nulová, pokud však tyto hodnoty začneme měřit, nebudeme posléze muset zasahovat do scriptů, které zpracovávají tento JSON. Protože je hodnota nyní nulová, nepřenáší se do dokumentové databáze Elasticsearch.

```

logging.info("Start process json")
        content_length = int(self.headers['Content-
Length'])
        post_data = self.rfile.read(content_length)
        raw_json = post_data.decode('utf-8')
        filename =
'/mnt/storage/vcely/data/json/{}.txt'.format(datetime.now
()).strftime('%y-%m-%d')
        with open(filename, 'a') as file:
            file.write("{0}\n".format(raw_json))
            file.close()

```

Blok kódu 15 Zpracování příchozího JSON dokumentu a uložení do souboru

Senzorové údaje se ukládají do textového souboru vždy jako nový řádek. Textový soubor je pojmenován podle dne, ve kterém byla data přijata, pro lepší dohledání a možnosti dalších analýz. Zároveň je také odeslán do Elasticsearch uložení pro možnost zobrazení.

Zpracování dat tedy obslouží jednoduchý script napsaný v Pythonu (vcely-server.py), běžící po celou dobu na pozadí jako jednoduchý HTTP server.

1.16 Analýza zvuku

Nahrávání audia z úlu probíhá každých 5 min a nahrává se desetisekundový záznam. Tento záznam je poté odeslán na server, kde se uloží pod názvem konkrétního času vložení jako *.wav soubor.

Po uložení proběhne okamžitě analýza, ze které vzejdou hodnoty frekvencí a ty jsou následně uloženy do Elasticsearch, pro možnost zobrazení pomocí grafů. Analýza probíhá za pomoci scriptu napsaném v Pythonu a probíhá až po uložení zvukového souboru, aby nedošlo ke ztrátě zvukových dat při případné chybě v průběhu analýzy.

Při analýze je využito modulů knihovny Scipy.io, konkrétně Wavfile a Numpy. Knihovna Scipy je komunitou vytvořený open source ekosystém určený pro vědecké využití při použití jazyka Python. Modul Numpy patří do ekosystému Scipy a umožňuje operace s N-dimenzionálním polem a používání lineární algebry. Ta je společně s Fourierovou transformací podstatná pro zvukovou analýzu. [19]

Z modulu Wavfile použijeme metodu read pro načtení dat z uloženého wav souboru. Tímto vzniknou dvě proměnné, a sice vzorkovací frekvence nahrávky a načtený signál *numpy array* bitového pole. Signál je nyní třeba transformovat pomocí Fourierovy transformace.

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N}nk} \quad k = 0, \dots, N - 1$$

Obrázek 16 Vzorec diskrétní Fourierovy transformace

Protože diskrétní Fourierova transformace je velmi výkonnostně náročná bude použita její alternativa Fast Fourier transform – Rychlá Fourierova transformace (FFT). FFT tedy použijeme na signál načtený do bitového pole. Tím nám vznikne pole nové, jeho obsahem budou komplexní čísla:

```
[ (-45191.+0.j) , (-3225.9789213 +285.4298708j) ,
  (781.84450439+687.34751637j) ... ]
```

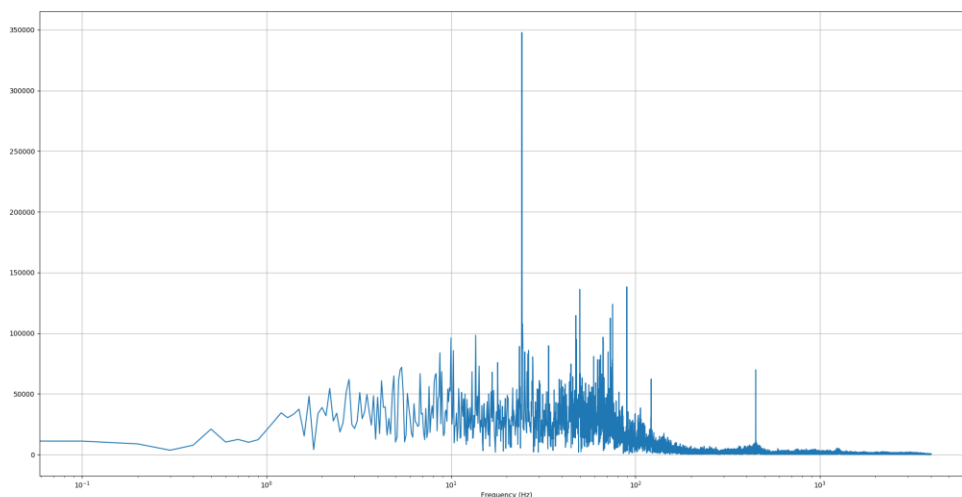
Blok kódu 16 Výstup numpy.array po použití Fourierovy transformace

Protože výsledkem Fourierovy transformace je pole komplexních čísel, tak abychom s těmito daty mohli dále pracovat, je nutné získat jejich absolutní hodnotu. Toho dosáhneme za pomoci metody Numpy.abs. [19]

```
[ 45191, 3238.5815123, 1041.02230394 ... ]
```

Blok kódu 17 Převedený výsledek FFT na jejich absolutní hodnoty

Takto dostaneme absolutní hodnoty amplitudy jednotlivých frekvencí. Abychom se dozvěděli, která frekvence má v dané nahrávce nevyšší amplitudu, je nutné vytvořit pole o stejné délce jako je délka pole po zpracování FFT. V tom nám pomůže metoda Numpy.rfftfreq. Nyní můžeme vytvořit graf pro kontrolu frekvenčního spektra dané nahrávky.

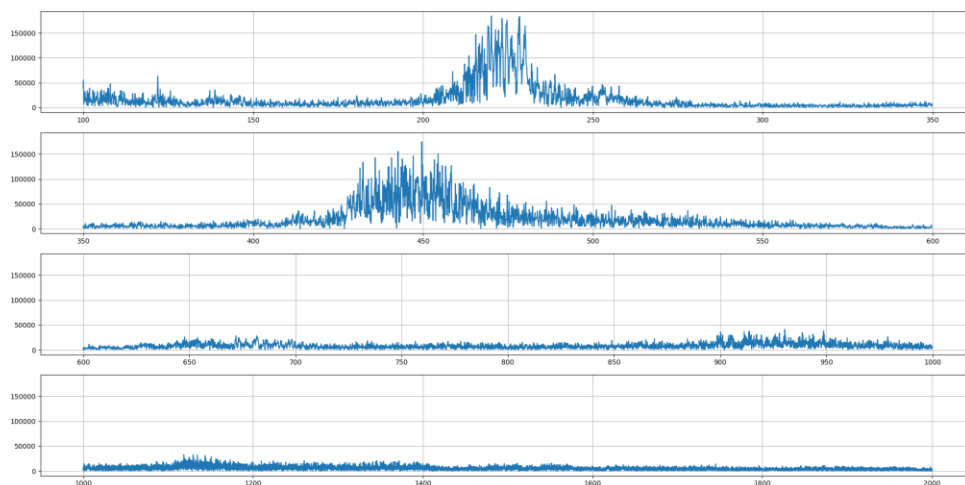


Obrázek 17 Celkový pohled na spektrum frekvencí v desetivteřinové nahrávce bzukotu, kde na ose Y vidíme amplitudu a na ose X hodnoty frekvence v Hz

Takto vypadá spektrum frekvencí pro desetivteřinovou nahrávku bzukotu. Z těchto dat ještě bude nutné odfiltrovat nízké frekvence, které označíme za šum. Protože cílem analýzy je získání frekvence s největší amplitudou, budeme výsledné spektrum ještě rozdělovat do následujících kategorií.

- 100 - 350 Hz
- 350 - 600 Hz
- 600 - 1000 Hz
- 1000 - 2000 Hz

Na základě sledování dat bylo zjištěno, že frekvence s největší amplitudou je nejčastěji okolo 450 Hz. V praxi bychom tedy dostávali z každé nahrávky téměř totožná výstupní data, a to hodnoty kolem 450 Hz, a proto bylo spektrum rozděleno do více kategorií. Pokud by nedošlo k rozdělení do kategorií, přišli bychom tak o ostatní zajímavé hodnoty frekvencí. Výsledný graf po přefiltrování do vybraných kategorií je vyobrazen níže:



Obrázek 18 Rozdělení frekvencí do kategorií, kde na ose Y je amplituda a na ose X jsou frekvenční hodnoty

Z desetivteřinového záznamu jsou tedy výsledkem tyto hodnoty frekvencí:

- 100 - 350: 121.8 Hz
- 350 - 600: 451.5 Hz
- 600 - 1000: 942.7 Hz
- 1000 - 2000: 1227.8 Hz

Takto vytvořená data následně zformátujeme do JSON a odešleme do Elasticsearch úložiště.

1.17 Ukládání naměřených dat

Pro ukládání dat byl po kladné zkušenosti řešitele vybrán produkt ElasticSearch. Jedná se o dokumentovou databázi, která je výborně zpracovaným nástrojem pro zobrazování uložených dat.

Protože je ElasticSearch vyvíjen jako open source, dá se jeho kompletní verze stáhnout zdarma z webu www.elastic.co. Po stažení, v našem případě linuxové distribuce, stačí balíček pouze rozbalit a není nutná instalace. ElasticSearch je kompletně napsán v programovacím jazyce Java a pro jeho správné fungování je tedy nutností, mít nainstalovanou Javu 8, popřípadě její starší verzi. Doporučené jsou stabilní verze Javy, tzv. LTS (Long Term Support).

Dále je několik důležitých upozornění pro správné provozování ElasticSearch.

- Vypnout swapování na disk - ElasticSearch běží jako služba spuštěna a kontrolována JVM.
- Zvětšit počet právě otevřených souborů na více jak 65536 - ElasticSearch může při práci využívat spoustu otevřených souborů při jejich zápisu nebo aktualizaci.
- Nastavení Virtuální paměti - ElasticSearch používá pro ukládání indexů defaultně MMap FS
- Možnost vytváření nových threadů - ElasticSearch potřebuje sám vytvářet thready pro své operace

Po dokončení těchto kroků může být ElasticSearch spuštěn z příkazového řádku - **./ElasticSearch-6.5.x/bin/elasticsearch**, následně bude server pro HTTP operace dostupný na adrese <http://127.0.0.1:9200>. [11]

Nyní je ElasticSearch schopný přijímat data. Pokud tedy odešleme z Python scriptu (vcely-server.py), zpracovávajícího příchozí data, ten samý JSON co přichází z Raspberry Pi, ElasticSearch ho dokáže sám zpracovat a uložit. Toto je základní rozdíl od relačních databází, ve kterých musí být nejdříve známo, jak budou ukládaná data vypadat a až poté se mohou ukládat. Data se ukládají do indexů, nad kterými se poté vyhledávají data.

Jak již bylo zmíněno výhodou dokumentových databází je možnost uložit dokument bez předchozího předpisu dat, nicméně díky předchozímu předpisu můžeme dosáhnout lepších výsledků jak v úspoře místa, tak také při rychlosti vyhledávání dat.

Takto vypadá JSON poslaný do ElasticSearch úložiště:

```
{
  "time": "2018-12-01T14:51:09-0000",
  "zluty_temp": "23.687",
  "zluty_hum": "0",
  "zeleny_temp": "4.5",
  "zeleny_hum": "99.9000015259",
  "venek_temp": "5.19999980927",
  "venek_hum": "95.5999984741",
  "deska_temp": "7.062",
  "vaha": "36.74"
}
```

Blok kódu 18 JSON dokument záznamu teplot, vlhkostí a váhy ukládaný do ElasticSearch

Takto naformátovaný JSON odesíláme, po přijetí a zpracování serverem (vcely-server.py), do ElasticSearch přes HTTP metodu POST. Tento JSON je tedy obsahem těla POST. Příklad odeslání z vcely-server.py pomocí modulu requests:

```
requests.post("http://127.0.0.1:9200/vcely/_doc",
headers={'content-type': 'application/json'}, data=y)
```

Blok kódu 19 Odeslání JSON dokumentu s naměřenými parametry do ElasticSearch

ElasticSearch se dá ovládat pomocí HTTP rozhraní přes příkazovou řádku pomocí cURL. Pohodlnější by však bylo mít možnost ovládání přímo z webového prohlížeče, proto použijeme další open source produkt z rodiny Elastic a tím je Kibana.

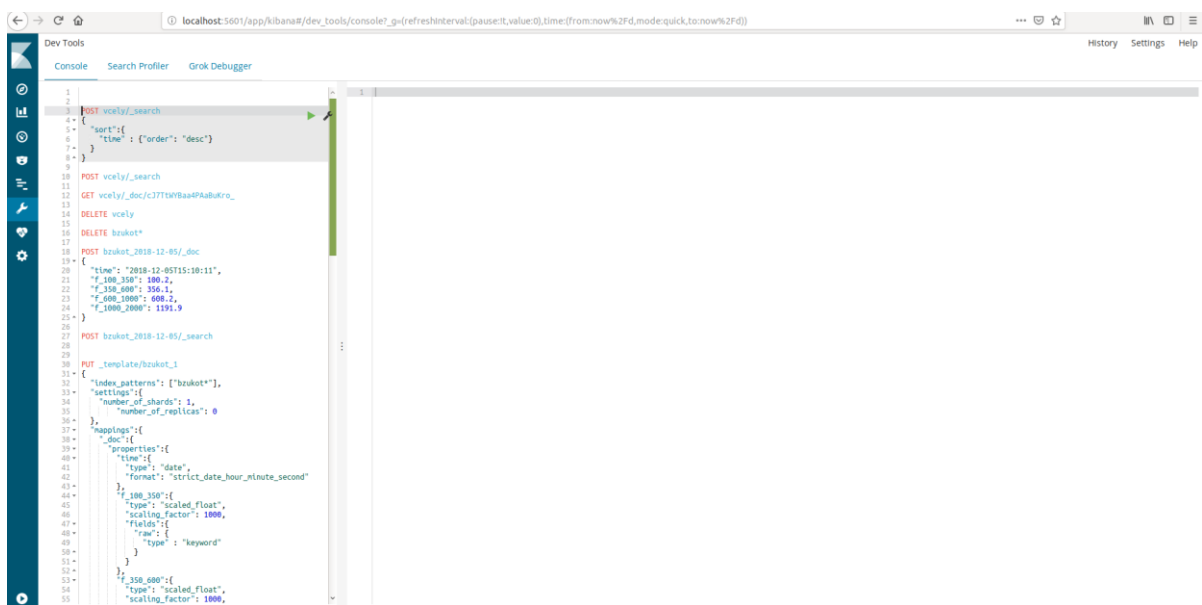
Kibana je designovaná pro souběh se serverem ElasticSearch s podporou stejných verzí. To znamená, že není podporováno mít Kibanu v 5.x a ElasticSearch v 2.x. Není také doporučeno mít vyšší verzi Kibany než ElasticSearch serveru, v praxi je ale verze Kibany 5.0.x a verze ElasticSearch 5.1.x. funkční.

Kibanu si také stáhneme z webu www.elastic.co, pro danou distribuci operačního systému a po stažení stačí pouze rozbalit. Před nastartováním serveru je potřeba podívat se do konfiguračního souboru kibana.yml. V tomto souboru se dá najít mnoho nastavení, nejdůležitější je však nastavení URL adresy spuštěného Elasticsearch serveru. Pokud je tato adresa stejná, můžeme Kibanu spustit.

Kibana již není vytvářena v programovacím jazyku Java, ale pomocí moderních JavaScript frameworků. Kibanu spustíme přes příkazovou řádku **./kibana-6.5.x-linux-x84/bin/kibana**. Během jejího spouštění můžeme v logu vidět, zda se podařilo připojit na zadaný Elasticsearch server nebo ne. Kibana od té doby běží na URL adrese 127.0.0.1:5601. [12]

Po zadání URL a vyčkání na načtení vidíme hlavní stránku, která v levé části obsahuje ovládací prvky. Přes tyto prvky se nejprve přemístíme do “Dev tools”.

DevTool slouží jako konzole pro zadávání příkazů pro Elasticsearch, lze jej tedy považovat za velmi dobrou náhradu klasické příkazové řádky.



Obrázek 19 Pohled na obrazovku kibany dev-tools

Konzole je rozdělená do dvou částí. V levé části zadáváme příkazy, které začínají vždy velkými písmeny napsanou operací POST, GET, DELETE nebo PUT a následuje cesta pro definici příkazu. Do konzole se již nemusí psát URL a PORT. Kompletní URL adresu si Kibana přebírá z konfiguračního souboru.

Přes konzoli tedy vytvoříme mapování pro náš index, abychom ho tím optimalizovali pro ukládání a vyhledávání.

```
PUT _mappings/vcely
"mappings":{
  "_doc":{
    "properties":{
      "time":{
        "type": "date",
        "format": "strict_date_hour_minute_second"
      },
      "zluty_temp":{
        "type": "half_float",
        "fields": {
          "raw": {
            "type": "keyword"
          }
        }
      },
      "zluty_hum":{
        "type": "half_float",
        "fields": {
          "raw": {
            "type": "keyword"
          }
        }
      }
    }
  }
}
```

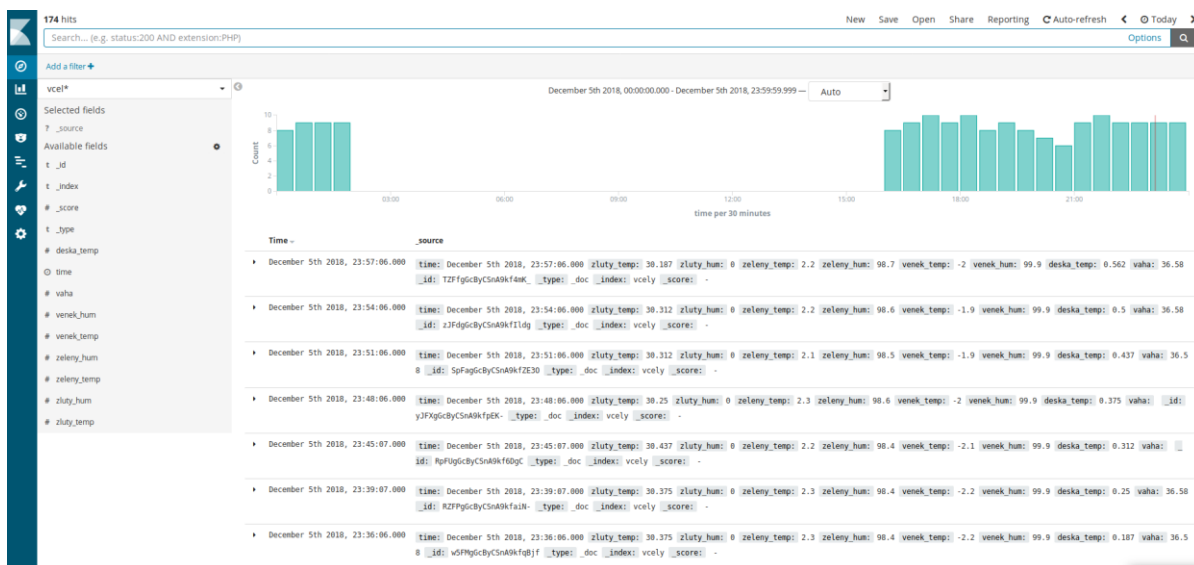
Blok kódu 20 Ukázka mapování indexu v úložišti Elasticsearch

Protože do tohoto indexu budeme ukládat pouze hodnoty teploty, vlhkosti a hmotnosti, použijeme datový typ `half_float`. Zároveň budeme také tyto hodnoty ukládat jako jednoduchý řetězec „keyword“. Tímto způsobem se tedy budou ukládat veškeré vložené JSON dokumenty. Takto je tedy vyřešeno ukládání dat a můžeme se přesunout k jejich samotnému zobrazení.

1.18 Zobrazení naměřených dat

Uložená data si zobrazíme pomocí Discover módu spuštěného z ovládacího panelu. Po načtení stránky uvidíme tabulkově zpracovaná data. Nahoře si všimneme pole pro vyhledávání. Přes toto pole můžeme data filtrovat pomocí “Kibana Query Language”

vycházející z nativního dotazovacího jazyku Lucene. Tímto si tedy filtrujeme tabulkově zobrazená data.



Obrázek 20 Pohled na Discovery obrazovku

Mód Discovery je nejvýhodnější pro rychlé vyhledávání a filtrování v uložených datech. V pravém horním rohu najdeme možnost data filtrovat podle vybraného času, můžeme tak zobrazit třeba jenom záznamy z posledních 15 minut nebo data mezi konkrétními hodinami v daném dni. Níže na stránce je vidět obrazovka rozdělená na dvě sekce. V levé části jsou všechny jednotlivé položky, pomocí kterých je možné filtrovat data nebo různě agregovat. Vpravo pak vidíme konkrétní vyfiltrovaná data. Jejich četnost za časové období je zobrazeno v histogramu, jak již bylo zmíněno v tabulkovém výpisu.

Některá data, především ta spojitá, je lepší zobrazovat jako lineární grafy. Grafy se skrývají pod položkou ovládacího panelu Vizualizace. Možností, jak uložená data nechat zobrazit, je mnoho. Níže jsou uvedené některé z nich

- Plošné grafy
- Sloupcový horizontální/vertikální graf
- Spojnicový graf
- Koláčový graf
- Datová tabulka

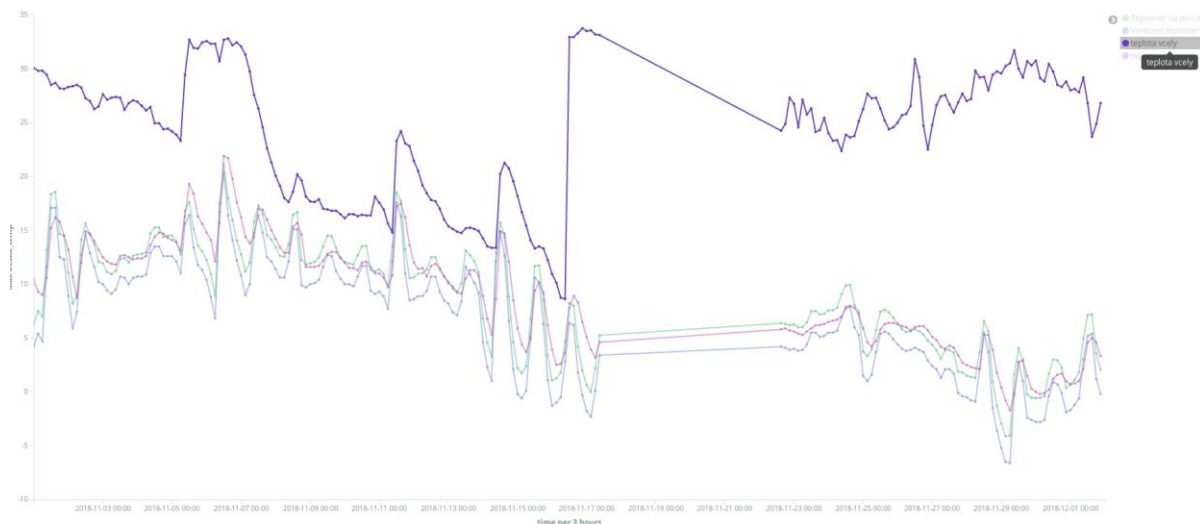
- Data zobrazí jako výsečový graf s aktuální hodnotou
- GEO data zobrazené body na mapě nebo jako celé regiony
- atd.

Pro zobrazení kontinuálního měření teploty nebo vlhkosti je nejvíce vyhovující spojnicový graf. Do tohoto grafu zaneseme data ze všech teploměrů, respektive vlhkostních senzorů a budeme moci pozorovat průběžné změny teplot, respektive vlhkostí.



Obrázek 21 Záznam měření teplot od 23.11. do 6.12.

Na obrázku výše je graf záznamu měření teplot vytvořený Kibanou. Fialová spojnice je teplota uvnitř chomáče včel. Na té můžeme vidět, že včely se neustále snaží zahřívát. Dolní trojice znázorňuje teploty z teploměrů umístěných venku, na desce a v rohu úlu.



Obrázek 22 Záznam pozorování přemístění včelího chomáče

Na tomto grafu je záznam teploty včelího chomáče (fialová linka), ze kterého je zřejmé, jak teplota v průběhu od 1. do 11. listopadu 2018 postupně klesá. Od 11. do 16. listopadu 2018 byl pozorován další konstantní propad až téměř na úroveň teplot z teploměru umístěného uvnitř úlu (růžová linka). Z důvodu poklesu byl 16. listopadu 2018 proveden zásah do úlu, při kterém bylo zjištěno, že se chomáč včel přestěhoval do přední části úlu. Z tohoto důvodu byl teploměr přemístěn také do přední části úlu, opět do středu chomáče. Můžeme tak pozorovat nejen okamžitý vzestup teploty, ale také její další mírný růst. Šikmá přímá čára značí bohužel nefunkčnost serveru. Následně je však opět zřetelné, že včely udržují teplotu v chomáči kolem třiceti stupňů Celsia.

Testování

Testování hardwaru probíhalo společně s prvotními pokusy o vytvoření softwaru pro záznam naměřených hodnot. Sensory použité v projektu byly od začátku zapojeny k základní desce. Software pro záznam naměřených hodnot byl vyvíjen na těchto senzorech, použitých následně při kompletaci celého inteligentního úlu.

Teplotní senzory byly v průběhu kompletace vystaveny pokusu, při kterém byly jejich naměřené hodnoty porovnávány vůči hodnotám z jiných teploměrů. Tyto teploměry jsme měli zapůjčené z laboratoří BC AV ČR. Teploměry byly jak digitální, tak také analogové.

Zkušební měření probíhalo v budově. Nejprve se porovnávala pokojová teplota naměřená senzory s teplotou naměřenou zapůjčenými teploměry. V tomto případě se hodnoty lišily pouze v rozmezích setin až desetin stupně Celsia. Poté byly teploměry umístěny do zapůjčené termo přepravky s ledem a průběžně se měřila klesající teplota. Při měření teploty „u ledu“ bylo zjištěno, že teploty v hodnotách nad nulou se liší pouze v desetinách stupně Celsia. Poté co bylo dosaženo teplot pod bodem mrazu, se měřené hodnoty začínaly lišit již výrazněji. Závěr je takový, že u hodnot pod bodem mrazu je odchylka až o jeden nebo dva stupně Celsia. Tato odchylka je pro prvotní implementaci našeho projektu přijatelná, a to z důvodu, že je pro nás nejdůležitější teplota uvnitř chomáče včel. Vzhledem k tomu, že teplota uvnitř chomáče nebude nikdy tak nízká, nemůže dojít k tak vysokým odchylkám. K teplotám pod nulou by došlo pouze v případě úhynu včelstva.

Po zkompletování celého inteligentního úlu proběhlo testovací měření při nepřítomnosti včelstva. Tento test proběhl bez problémů a mohli jsme přikročit k umístění včelstva do inteligentního úlu. Po umístění včelstva bylo opět provedeno testovací měření, které bylo taktéž úspěšné. Na tomto základě byl spuštěn proces průběžného měření. Správnost měření byla pravidelně kontrolována na uložených datech na serveru.

Na data bylo zprvu pohlíženo pouze pohledově, protože software na zobrazení a analýzu se vyvíjel až po dokončení vývoje softwaru pro záznam a uložení sensorových dat. Nyní jsou již data monitorována v zobrazovacím softwaru a také v logu spuštěného serverového skriptu pro příjem dat z včelího úlu.

Aktuální využití projektu

Díky tomuto projektu může být započat výzkum k objasnění příčin špatného zdravotního stavu u včely medonosné (*Apis mellifera*). Výzkum by měl být zahájen v prosinci 2018 a bude klást důraz na nepřírozené podněcování plodové aktivity a na nutriční skladbu potravy podávanou včelařem včelám jako příkrm v bezsnůškovém období, za pomoci sledování charakteristik včelstva v inteligentním úle, ve vztahu k výživě, populační dynamice a zdravotního stavu včelstva v inteligentním úlu.

Celosvětový pokles populace včely medonosné, v důsledku zvýšeného infekčního tlaku včelích patogenů, je vnímán s výrazným neklidem, a to především vezmeme-li v potaz, že včela medonosná je hlavním opylovačem většiny zemědělských plodin. Za tímto účelem výzkumu vznikají po celém světě výzkumné aktivity pro osvětlu příčiny úbytku včel. Hlavními viníky jsou vnímány ztráty přirozených ekosystémů, monotónnost včelí pastvy a nadměrné používání pesticidů v zemědělství. Významný faktor je také špatné příkrmování včelstva v průběhu podletí.

Výzkum se tedy zaměřuje na vyhodnocení vlivu intenzity plodování a síly včelstva na endokrinní regulaci a aktivitu telomerázy, jakožto markery biologického věku a posouzení, do jaké míry může pozdní příkrmování a nutriční skladba příkrmů vést k urychlenému stárnutí přezimující generace včel a kolapsům v zimním období.

V minulých letech bylo při experimentech zjištěno, že mikroskopická zelená řasa *Chlorella* sp. (produkt MBÚ AV ČR – Centrum ALGATECH, Třeboň) je pro včely velmi atraktivní potravou. Řasa je bohatým zdrojem bílkovin, tuků, škrobu, vitamínů, minerálů, antioxidantů a vlákniny. Podávání chlorelly vedlo u včel až k trojnásobnému nárůstu aktivity telomerázy.

Hlavním úkolem se tedy stane sledování parametrů včelstva v inteligentním úlu kde jsou včely krmené zelenou řasou *Chlorellou*, zpracování získaných dat, a to vhodnými data mining metodami za přispění umělé inteligence a neuronových sítí.

Metody sledování a analyzování se budou postupným monitoringem dále rozvíjet, konkrétně postupným zprovozněním dalších 2-3 inteligentních úlů.

Návrhy na zlepšení

Během vývoje inteligentního úlu bylo zjištěno, že některé části jak hardwarové, tak softwarové by mohly být ještě vylepšeny.

1.19 Senzory a mikrofon

Pro prvotní investici byly zakoupeny levné teplotní a vlhkostní senzory, a to především za účelem odzkoušení funkčnosti a zjištění přínosu těchto měření. Ještě přesnější výsledky bychom získali, pokud bychom zakoupili kvalitnější senzory a zároveň vyšší počet těchto senzorů. Mohli bychom je tak umístit na více místech v úle. Předěšlo by se tak ručním zásahům do úlu v situaci, kdy se včely v průběhu měsíce přestěhují ze zadní části úlu do přední. Tento jev si můžeme prohlédnout na obrázku 19. Tyto další senzory by se mohly rozmístit na vícero místech pouze za předpokladu, že umístění senzorů rapidně nezhorší možnost manipulace se včelstvem.

Další položkou by mohl být mikrofon, který ačkoli funguje výborně, je umístěn na dně úlu pod rámečky. Snímaný zvuk je nyní použitelný, ale pouze za předpokladu, že při letních měsících nebudou včely rozmístěné po celém úle, tedy na více jak jednom nástavku. Pokud by k tomu došlo, bylo by snímání zvuku velmi obtížné.

Zajímavé by mohlo být také umístění dalšího mikrofonu na vršek úlu a porovnávat včelí bzukot z horního nástavku oproti dolnímu nástavku. Včelaři totiž mohou díky matěří mřížce zamezit matce průchod do vrchních částí úlu. Dala by se tedy zkoumat povaha bzučení včel okolo matky, včel starajících se o plod a povaha bzučení včelích dělnic starajících se o donášku zásob.

V úvahu by mohly přijít i další typy senzorů, například senzor na měření plynů nebo senzor snímající aktivitu u vchodu do včelího úlu.

1.20 Ukládání dat

Současné ukládání dat do souborů jako fyzicky uložených kopií je dobré pro znovu-použitelnost uložených dat. S každým dnem však těchto dat přibývá, a proto by bylo vhodné mít možnost uložená data komprimovat a ušetřit tím místo pro další data. Jelikož jsou data

současně ukládána kvůli zobrazování ještě do Elasticsearch databáze, není potřeba mít k těmto datům přímý přístup.

Data ukládaná do Elasticsearch jsme již dostatečně ošetřili vypracováním mapovacího zápisu. Mapováním jsme zamezili vytváření přebytečných vnitřních tabulek například pro vyhledávání v textových polích. Nicméně pro budoucí použitelnost bychom mohli ukládaná data rozdělovat do indexů po dnech. Samotné vyhledávání by poté probíhalo napříč všemi indexy. Poté bude možné staré indexy buď pro úsporu místa a rychlost zobrazení dat smazat nebo pouze uzavřít, a tím tedy zamezit vyhledávání nad těmito daty uloženými v indexech.

Zjednodušeně řečeno, většinou potřebujeme vidět záznamy staré dva až tři týdny nazpět, proto není třeba vždy načítat a hledat v datech starších jak tento časový horizont. Zrychlíme tak zobrazení a vyhledávání dat. Pokud bychom chtěli zobrazit širší časový horizont, můžeme si potřebná data kdykoliv zase zpřístupnit.

1.21 Analýza zvuku

Momentálně provádíme analýzu zvuku na souboru obsahujícím desetivteřinový záznam vytvořený každých 5 minut, přičemž analýzu provádíme až po uložení na serveru. Jelikož je náš inteligentní úl umístěný na včelnici s možností připojení elektrické energie, nemusíme v tomto případě hledět na úsporu energie a můžeme základní desku Raspberry Pi využívat i více, a to tak, že budeme v postupných intervalech pořizovat kontinuální nahrávku včelího bzukotu.

Z důvodu, že na pořizování celodenních kontinuálních nahrávek není ani na serveru ani na samotném Raspberry Pi prostor, nahrávali bychom jí po částech. Po uložení nahrané části, bychom jí hned zanalyzovali a tím by vznikly pouze výsledné hodnoty frekvencí z dané nahrávky. Tyto frekvence bychom poté odeslali na server a zanalyzovanou nahrávku vymazali. Po vymazání poslední nahrávky se spustí znovu nahrávání další části. Nebyl by to tedy doslovný kontinuální záznam, ale i tak bychom dosáhli detailnějšího přehledu o včelím bzukotu. Předěšlé pořizování audio nahrávek pro možnost dalších analýz by mohlo probíhat jako doposud. Tyto uložené nahrávky by měly být také komprimovány pro efektivnější využití místa.

Závěr

Diplomová práce na téma „Sběr a analýza dat z inteligentního včelího úlu“ je rozdělena primárně na část teoretickou a praktickou. Cílem této práce bylo na základě realizovaného projektu Rozmanitost života a zachování ekosystému (ROZE) navrhnout a vytvořit funkční řešení, jak pro sběr a analýzu dat z včelího úlu, tak také pro jejich archivaci na vzdáleném serveru a následné zobrazení.

V teoretické části je čerpáno především ze zdrojů odborné literatury. Její součástí je přiblížit čtenáři téma včelařství jako takové, proto obsahuje stručně včelařskou historii a význam včelaření vůbec. Dále jsou zde popsány podmínky, které je nutné splnit, aby mohlo dojít ke splnění stanoveného cíle, od metodiky, přes výběr vhodného hardwaru, vytvoření softwaru až po zhodnocení výsledků.

Teoretická část práce se tedy zabývá možnostmi při vytváření architektury pro snímání senzorových hodnot včelího úlu. Jsou zde popsány hardwarové součástky a jejich možné využití. V první řadě bylo potřeba zvolit vhodnou základní desku. Ta byla vybírána z malých jednodeskových počítačů, konkrétně z originálních produktů Arduino a Raspberry Pi třetí generace. Ve finále byla zvolena právě Raspberry Pi, která je o něco silnější a také mnohem přístupnější než zmíněné Arduino. Další část popisuje vytvoření komunikačního rozhraní mezi inteligentním včelím úlem a serverem. Protože v našem případě je umístěna včelnice v kampusu akademie a univerzity, bylo ke spojení použito optické vlákno. Jsou zde však uvedené možnosti komunikace nejen z pohledu aktuální situace, tedy umístění úlu v blízkosti akademických budov a možnosti použít pevný spoj, ale také v případě umístění úlu mimo dosah obytných budov (Wi-Fi, ZigBee, GSM/GPRS). Dále byla práce věnována výběru senzorů na základě důležitosti měření a také opodstatnění pořizování dat. Například záznam o hmotnosti dává včelaři pokyn ke sběru medových zásob anebo informace o aktuální vitalitě včelstva. Kromě hmotnosti jsme se koncentrovali na teplotu a vlhkost uvnitř i vně úlu a na frekvenci včelího bzučení. V neposlední řadě teorie zahrnuje popis uchování naměřených dat na serveru, přičemž údaje o teplotě, vlhkosti a hmotnosti přijdou na server jako zformátovaný JSON a jsou do centrálního úložiště ukládány jako textové soubory. Zvuk server přijímá jako bitový stream a ukládá jako soubor * wav. Pro možné zobrazení těchto dat byla použita open source aplikace Kibana, která je grafickou nadstavbou pro dokumentovou databázi a fulltextový vyhledávač Elasticsearch. Kibana dokáže nejen

zobrazit veškerá data v interaktivních grafech, ale hlavně nad nimi umožňuje provádět další analýzy.

Praktická část je nejprve věnována popisu kompletace vybraného hardwaru s včelím úlem. Jak již bylo řečeno, pro náš projekt bylo využito včelího úlu umístěného na včelnici v blízkosti univerzitní budovy. Ačkoli je úl umístěn pod zastřešenou pergolou, jedná se o venkovní projekt, který není chráněn před různorodostí počasí. Základní desku bylo tedy potřeba umístit do plastové elektro-instalační krabice, do které byly vyvrtány díry pro vodotěsné průchodky. Do plastové krabice bylo potřeba umístit i váhu pro měření hmotnosti. Vzhledem k časté manipulaci a k usnadnění přístupnosti do úlu, byly krabice umístěny na zadní stranu nejspodnějšího nástavku úlu.

Dále bylo potřeba zapojit jednotlivé senzory k základní desce, která je srdcem celého projektu a následně senzory uvést do provozu. Pro měření teploty a vlhkosti byly vybrány celkem čtyři senzory, dva typu DHT21/AM2301 a dva typu DS18B20. Jeden senzor typu DHT21/AM2301 byl umístěn venku, druhý uvnitř úlu. Tento typ senzorů měří jak teplotu, tak i vlhkost. Další dva teplotní senzory jsou typu DS18B20, jeden je umístěný v blízkosti základní desky a druhý uvnitř chomáče včel. Všechny senzory jsou k desce připojeny pomocí sběrnice One Wire, přičemž ke čtení dat z těchto senzorů slouží knihovna `Adafruit_Python_DHT`. Ke měření hmotnosti úlu byla pro svou spolehlivost pořízena průmyslová váha `LESAK 1T6060LN-RWP`, která je určena přímo pro včelaře. Pro poslední ze sledovaných dat, tedy zvuk, byl zvolen malý klopový mikrofon `Zalman ZM-MIC1`, pro jehož zapojení musela být zakoupena externí Mini USB zvuková karta.

Ke komunikaci mezi včelím úlem a serverem bylo použito optické vlákno, zavedené z budovy AV ČR do routeru umístěným v racku. Uvnitř racku je PoE swich, který zajistí spojení úlu s routerem. Tím se vyřešilo jak datové spojení, tak také napájení inteligentního úlu. Naměřená data se pak odesílají pomocí scriptu v programovacím jazyce Python, který díky modulu `responses` podporuje běžné HTTP metody.

Programovací jazyk Python byl také použit pro vytvoření dalších programových částí inteligentního úlu. Tyto programové části byly vytvářeny souběžně s testováním. Dá se říci, že byl vždy sestaven funkční celek hardwaru, na kterém byl poté vyvíjen patřičný software a tím byl tento celek také rovnou testován. Před kompletací inteligentního úlu byl tedy tento software již řádně otestován. Testování samozřejmě proběhlo také ihned po zkompletování

úlu, ještě před umístěním včelstva. Při vývoji monitorovacího softwaru byl zároveň vyvíjen software pro příjem dat na vzdáleném serveru. Zde bylo nejdříve vytvořeno zpracování tepelných, vlhkostních a hmotnostních dat přicházejících HTTP POST. Po dokončení této části byla vytvořena část zpracovávající audio záznam včelího bzučení. Tímto bylo zabezpečeno získávání dat ze včelího úlu.

Poslední částí bylo zprovoznění dokumentové databáze Elasticsearch s grafickou nadstavbou pro vytváření grafů Kibana. Tyto dvě aplikace byly zprovozněny na vzdáleném serveru a následně mohl být vytvořen software analyzující data před uložením do této dokumentové databáze. Pro část s teplotou, vlhkostí a hmotností to představovalo pouze upravení stávajícího software na příjem dat z inteligentního úlu. Pro analýzu zvukových nahrávek musel být však vytvořen software nový.

Po zpracování již naměřených dat, uložených do Elasticsearch, byly vytvořeny grafy v Kibaně, díky kterým byl zprovozněn dohled nad funkčností inteligentního úlu. Na základě těchto pozorování jsme došli také k návrhům, na možné zlepšení. Například zakoupením kvalitnějších senzorů a jejich většího počtu bychom dosáhli ještě přesnějších výsledků nebo umístěním dalších mikrofónů by se dala zkoumat také různorodost bzučení v dalších částech úlu.



Obrázek 23 Celkový pohled na zastřešenou včelnici s umístěným inteligentním úlem (úplně vlevo)

Výsledkem kompletace hardwarových součástí a programové komunikace byl vytvořen komplex inteligentního úlu se sensorovým snímáním dat ze života včel a jejich následným uložením na vzdálený server s možnostmi zobrazení a dalších analýz. Průběžná měření a pozorování byla velice zajímavá a díky tomu se přišlo na to, jak se včely chovají v průběhu období roku, aniž by se muselo denně nahlížet do úlu a rušit tak včely v jejich pracovní aktivitě. Dle získaných dat může pak včelař zasáhnout a provést činnosti směřující k odstranění negativních výsledků v datech, např. včasný výběr medových zásob, příkrmování cukerným roztokem, atd.



Obrázek 24 Inteligentní úl

Tento projekt je důležitým podkladem pro další výzkumy zabývající se hledáním příčin mortality včely medonosné. Dalo by se tedy říci, že tento projekt je výchozím nástrojem pro záchranu včel.

Seznam použité literatury

- [1] TAUTZ Jürgen. *Fenomenální včely*. Překl. Olga Matyásková. 2009, Praha: Brázda. [cit. 2018-10-11]. ISBN 978-80-209-0376-1
- [2] VESELÝ Vladimír. *Včelařství 2013*, Praha: Brázda. [cit. 2018-22-11]. ISBN: 978-80-209-0399-0
- [3] About - Arduino. In: *Arduino* [online]., 2018 [cit. 2018-24-11]. Dostupné z: <https://www.arduino.cc/en/Main/AboutUs>
- [3] About - Arduino. In: *Arduino* [online]., 2018 [cit. 2018-24-11]. Dostupné z: <https://www.arduino.cc/en/Main/AboutUs>
- [4] List of Arduino boards and compatible systems. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2018 [cit. 2018-4-12]. Dostupné z: https://en.wikipedia.org/wiki/List_of_Arduino_boards_and_compatible_systems
- [5] Raspberry Pi info. In: *Raspberry Pi — Teach, Learn, and Make with Raspberry Pi* [online]. Raspberry Pi Foundation, 2018 [cit. 2018-24-11]. Dostupné z: <https://www.raspberrypi.org/>
- [6] A Hands-on Experience with Wireless Sensor Networks In: *P2PU / Learn* [online]. 2018 [cit. 2018-20-11]. Dostupné z: <https://courses.p2pu.org/en/courses/831/content/1739/>
- [7] Pojďme programovat elektroniku. In: *Živě.cz* [online]. CZECH NEWS CENTER a.s, 2018 [cit. 2018-24-11]. Dostupné z: <https://www.zive.cz/pojdme-programovat-elektroniku/sc-695/default.aspx>
- [8] Mikrofon Zalman ZM-MIC1 | Abacus Electric. In: *abacus.cz* [online]. 2018 [cit. 2018-4-12]. Dostupné z: https://www.abacus.cz/mikrofon-zalman-zm-mic1_d5033142.html
- [9] IVANSKÝ, Tomáš. Úvod do problematiky akustického monitoringu včelstva. *Moderní včelař*. Praha: PSNV-CZ. 2017, 1(01), [cit. 2018-27-11]. ISSN 1214-5793.
- [10] RESTful, Distributed Search & Analytics In: *Open Source Search & Analytic | Elastic* [online]. Elasticsearch B.V. 2018 [cit. 2018-26-11]. Dostupné z: <https://www.elastic.co/products/elasticsearch>

- [11] Elasticsearch. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2018 [cit. 2018-6-12]. Dostupné z: <https://cs.wikipedia.org/wiki/Elasticsearch>
- [12] Explore, Visualize, Discover data In: *Open Source Search & Analytic / Elastic* [online]. Elasticsearch B.V. 2018 [cit. 2018-26-11]. Dostupné z: <https://www.elastic.co/products/kibana>
- [13] Elastic Stack—A Brief Introduction In: *Medium – a place to read and write* [online]. 2018 [cit. 2018-26-11]. Dostupné z: <https://hackernoon.com/elastic-stack-a-brief-introduction-794bc7ff7d4f>
- [14] Requests: HTTP for Humans In: *Requests documentation* [online]. 2018 [cit. 2018-27-11]. Dostupné z: <http://docs.python-requests.org/en/master/#>
- [15] adafruit/Adafruit_Python_DHT: *GitHub* [online]. GitHub, Inc 2018 [cit. 2018-30-11]. Dostupné z: https://github.com/adafruit/Adafruit_Python_DHT
- [16] PyAudio v19 Python Bindings: *Csail edu* [online]. 2018 [cit. 2018-07-12]. Dostupné z: <http://people.csail.mit.edu/hubert/pyaudio/>
- [17] LESAK 1T4040LN-RWP | Včelařské váhy na úly: *elektro-garden.cz* [online]. 2018 [cit. 2018-05-12]. Dostupné z: <https://www.elektro-garden.cz/produkt/lesak-1t4040ln-rwp-150kg.html>
- [18] 3.7.1 Documentation: *Python org* [online]. Python Software Foundation 2018 [cit. 2018-02-12]. Dostupné z: <https://docs.python.org/3.7/>
- [19] SciPy – SciPy v1.1.0 Reference guid: *SciPy org* [online]. The SciPy community 2018 [cit. 2018-02-12]. Dostupné z: <https://docs.scipy.org/doc/scipy/reference/index.html>

Seznam obrázků

Obrázek 1 Návrh architektury	4
Obrázek 2 Jednodeskový počítač Arduino UNO [4].....	6
Obrázek 3 Raspberry Pi třetí generace verze B [5]	7
Obrázek 4 Možné topologie sítě ZigBee [6]	9
Obrázek 5 Senzor umístěný ve školkovací klícce	11
Obrázek 6 Klopový mikrofon Zalman ZM-MIC1 [8].....	12
Obrázek 7 ELK stack.....	16
Obrázek 8 Uchycení plastové krabice na nástavku úlu	17
Obrázek 9 Schéma tištěného spoje základní desky	18
Obrázek 10 Ilustrační foto zapojení Raspberry Pi pomocí PoE splitteru a PoE switche	21
Obrázek 11 Fotka osazeného včelího úlu	22
Obrázek 12 DHT21/AM2301 Digitální senzor teploty a vlhkosti v pouzdru	23
Obrázek 13 Teplotní senzor DS18B20.....	24
Obrázek 14 Včelařský vážící můstek [17].....	29
Obrázek 15 Blokové schéma RS232 konektoru	30
Obrázek 16 Vzorec diskrétní Fourierovy transformace	37
Obrázek 17 Celkový pohled na spektrum frekvencí v desetivteřinové nahrávce bzukotu, kde na ose Y vidíme amplitudu a na ose X hodnoty frekvence v Hz.....	38
Obrázek 18 Rozdělení frekvencí do kategorií, kde na ose Y je amplituda a na ose X jsou frekvenční hodnoty	39
Obrázek 19 Pohled na obrazovku kibany dev-tools	42
Obrázek 20 Pohled na Discovery obrazovku.....	44
Obrázek 21 Záznam měření teplot od 23.11. do 6.12.....	45
Obrázek 22 Záznam pozorování přemístění včelího chomáče	46
Obrázek 23 Celkový pohled na zastřešenou včelnicí s umístěným inteligentním úlem(úplně vlevo).....	53
Obrázek 24 Inteligentní úl	54

Seznam tabulek

Tabulka 1 Tabulka včelích zvukových projevů [9]	14
Tabulka 2 Tabulky zapojení senzorů na piny základní desky	19

Seznam bloků kódů

Blok kódu 1 Odeslání dat pomocí modulu requests	21
Blok kódu 2 Čtení ze senzoru DHT21/AM2301 pomocí knihovny Adafruit	24
Blok kódu 3 příklad zápisu do cron tabulky	25
Blok kódu 4 Načtení řádků se záznamem teploty ze sběrnice One Wire	26
Blok kódu 5 Ukázka načteného záznamu s teplotou ze sběrnice One Wire	26
Blok kódu 6 Vyseparování naměřené teploty z kompletního výstupu senzoru	27
Blok kódu 7 Připojení sériového portu pomocí modulu serial	30
Blok kódu 8 Výsledek čtení sériového USB portu	30
Blok kódu 9 čtení hmotnosti ve scriptu data-reader.py	31
Blok kódu 10 Spuštění HTTPServeru	32
Blok kódu 11 Ukázka zápisu funkce pro zpracování GET request	32
Blok kódu 12 Zpracování POST požadavku na HTTP serveru	33
Blok kódu 13 Zpracování přijatých bitových dat a vytvoření audio souboru s datumovým názvem souboru	34
Blok kódu 14 Přijatý JSON s daty ze senzorů	35
Blok kódu 15 Zpracování příchozího JSON dokumentu a uložení do souboru	36
Blok kódu 16 Výstup numpy.array po použití Fourierovy transformace	37
Blok kódu 17 Převedený výsledek FFT na jejich absolutní hodnoty	37
Blok kódu 18 JSON dokument záznamu teplot, vlhkostí a váhy ukládaný do ElasticSearch	41
Blok kódu 19 Odeslání JSON dokumentu s naměřenými parametry do ElasticSearch	41
Blok kódu 20 Ukázka mapování indexu v úložišti ElasticSearch	43