FAKULTA
INFORMAČNÍCH
TECHNOLOGIÍ
ČVUT V PRAZE

# ZADÁNÍ DIPLOMOVÉ PRÁCE

| | |
|---|---|
| **Název:** | Zvukový senzor pro Internet of Things, jeho zabezpečení a vzdálená správa |
| **Student:** | Bc. Ondřej Fuchs |
| **Vedoucí:** | Dr.-Ing. Martin Novotný |
| **Studijní program:** | Informatika |
| **Studijní obor:** | Počítačové systémy a sítě |
| **Katedra:** | Katedra počítačových systémů |
| **Platnost zadání:** | Do konce letního semestru 2018/19 |

## Pokyny pro vypracování

- Prostudujte dostupnou literaturu týkající se Internetu věcí (Internet of Things - IoT) zaměřenou na možnosti zabezpečení v rámci IoT senzorů a vzdálené správy, seznamte se s metodami zabezpečení.
- Navrhněte zařízení spadající do IoT pro nahrávání zvuku s ohledem na jeho zabezpečení. Zařízení primárně slouží pro nahrávání a streaming zvuků (například ve výrobní hale) do cloudu. Přenos dat do cloudu má být šifrovaný (zvolte nebo navrhněte vhodný způsob šifrování). Vyřešte vhodným způsobem odolnost zařízení vůči rozpouzdření (tzv. "tamper-resistance"); důležitá je ochrana firmware.
- Zařízení realizujte, otestuje a vyhodnoťte.
- Zhodnoťte výsledky práce a diskutujte případné pokračování nebo rozšíření práce.

## Seznam odborné literatury

Dodá vedoucí práce.

prof. Ing. Róbert Lórencz, CSc.
vedoucí katedry

doc. RNDr. Ing. Marcel Jiřina, Ph.D.
děkan

V Praze dne 12. února 2018

FAKULTA
INFORMAČNÍCH
TECHNOLOGIÍ
ČVUT V PRAZE

Diplomová práce

# Zvukový senzor pro Internet of Things, jeho zabezpečení a vzdálená správa

***Bc. Ondřej Fuchs***

Katedra počítačových systémů
Vedoucí práce: Dr.-Ing. Martin Novotný

22. června 2018

# Poděkování

Chtěl bych poděkovat panu Dr.-Ing. Martinovi Novotnému za odborné vedení a cenné rady, které mi pomohly při vytváření této práce. Dále bych rád poděkoval firmě Neuron soundware, která mi nabídla možnost spolupráce a rád bych poděkoval své rodině za podporu nejen při psaní diplomové práce, ale během celého studia.

# Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 46 odst. 6 tohoto zákona tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou, a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla, a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu), licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 22. června 2018 .....

### Odkaz na tuto práci

Fuchs, Ondřej. *Zvukový senzor pro Internet of Things, jeho zabezpečení a vzdálená správa*. Diplomová práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2018.

# Abstrakt

Diplomová práce se zabývá v dnešní době velmi zmiňovaným pojmem *Internet věcí (Internet of Things)* a zejména problematikou zabezpečení zařízení spadajících do této kategorie. Hlavním cílem práce bylo na základě podrobné analýzy bezpečnostních rizik a možných bezpečnostních incidentů aplikovat získané poznatky při vývoji zvukového senzoru, který bude sloužit k získávání zvukových dat pro následnou detekci zvukových anomálií firmou Neuron soundware. Zvukový senzor byl vytvořen podle předem definovaných pravidel a víceúrovňově zabezpečen proti odcizení citlivých dat. Využívá služby jako Amazon S3, Dataplicity, Elasticsearch, Logstash, Kibana a Filebeat.

**Klíčová slova** Zvukový senzor, Internet věcí, Zabezpečení, Vzdálená správa, Detekce manipulace, 3D tisk, Ansible, ELK Stack, Raspberry Zero.

# Abstract

Diploma thesis deals with nowadays very mentioned term Internet of Things and especially with the issue of security of devices from this category. The main objective of this thesis was to analyze security risks and possible security incidents and afterwards to apply the acquired knowledge in the development of the sound sensor, which will be used to obtain sound data for the detection

of sound anomalies by Neuron soundware company. The sound sensor was developed on the basis of predefined rules and it is multi-level security against the theft of sensitive data. Sensor uses services such as Amazon S3, Dataplicity, Elasticsearch, Logstash, Kibana and Filebeat.

**Keywords** Sound sensor, Internet of things, Security, Remote control, Detection of manipulation, 3D print, Ansible, ELK Stack, Raspberry Zero.

# Obsah

# Seznam obrázků

# Seznam tabulek

# Úvod

Diplomová práce se zabývá v dnešní době velmi zmiňovaným tématem *„Internet věcí“* a zejména problematikou zabezpečení zařízení spadajících do této kategorie. V práci jsou vyzdviženy největší hrozby a základní principy zabezpečení, které by měly být dodrženy při návrhu a realizaci nového zařízení spadajícího do *„Internetu věcí“*.

Poznatky týkající se základních nedostatků a možných řešení v oblasti zabezpečení zařízení byly následně aplikovány při návrhu a realizaci senzoru pro sběr zvukových nahrávek průmyslových strojů pro následnou detekci zvukových anomálií firmou Neuron soundware.

První kapitola popisuje výchozí podmínky pro sběr a zpracování zvukových dat, které slouží k detekci možných technických poruch průmyslových zařízení. Upozorňuje na široké možnosti dalšího použití metody v blízké budoucnosti v různých technických zařízeních a popisuje cíle práce a základní orientaci v problematice *„Internetu věcí“*.

Druhá kapitola pojednává o analýze a návrhu zvukového senzoru dle požadavků stanovených firmou Neuron soundware. Nejdříve jsou uvedeny možné hardwarové komponenty. Jelikož stále neexistují žádné oficiální standardy, které by popisovaly základní mechanismy zabezpečení těchto zařízení, autor předkládá různá bezpečnostní doporučení při návrhu zařízení z kategorie *„Internetu věcí“*. Další část kapitoly je věnována možnostem monitorování a vzdálené správy zařízení. V závěru druhé kapitoly je popsán návrh zvukového senzoru se zaměřením na detekci neoprávněné manipulace se zařízením, v anglické literatuře nebo na internetu často označované jako *„Tamper Mechanisms“*.

Třetí kapitola popisuje realizaci návrhu zvukového senzoru, zejména sestavení hardware, nastavení systému pomocí připravených skriptů a programu Ansible. Část kapitoly je věnována skriptu pro nahrávání zvukových záznamů a k přenosu záznamů do cloudu. Dále kapitola popisuje software pro detekci neoprávněné manipulace se zvukovým senzorem a vyvolání odpovídající obranné reakce zařízení.

Čtvrtá kapitola je věnována testování zvukového senzoru se zaměřením na hlavní funkce zařízení, tedy pořizování zvukových nahrávek, jejich přenos do cloudu a bezpečnostní mechanismy navržené pro ochranu firmware v zařízení.

V páté kapitole je nastíněno možné pokračování diplomové práce, zejména snaha dále snížit cenu a rozměry zařízení, a tím pádem rozšířit další možnosti použití zvukového senzoru.

Kapitola 1

# Cíl práce

Česká firma Neuron soundware se zabývá detekcí anomálií (anglicky „Anomaly detection“) průmyslových strojů pomocí zvuků. Na základě komplikovaných algoritmů spadajících do kategorie neuronových sítí vyhodnocuje zvukové nahrávky a upozorňuje na případný problém. Nabídka komerčně dostupných zařízení zaměřených na detekci anomálií průmyslových zařízení pomocí zvuku takřka neexistuje. Možnosti využití jsou přitom široké. Firma svůj prvotní vývoj cílí na výrobní linky v průmyslových komplexech z důvodu jednoduššího sběru dat a následného vyhodnocování.

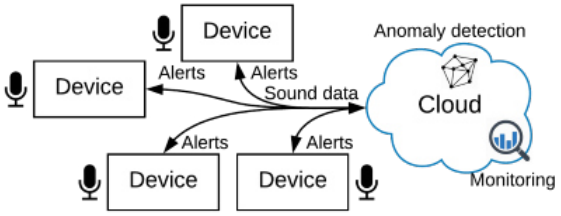
Budoucnost využití zařízení, které dokáže detekovat blížící se poruchu nebo nestandardní stav, je nepřeberné. Pokud dokážeme vytvořit dostatečně malé zařízení se spolehlivým algoritmem pro detekci anomálií, mohli bychom spolehlivě monitorovat stav např. automobilových motorů, spotřební elektroniky, průmyslových systémů, automatizovaných výrobních linek a dalších. Díky zajímavému nápadu se firmě Neuron soundware podařilo prosadit v prestižní soutěži Startup roku 2016 a získat zajímavé podmínky pro svůj další rozvoj.

Byl jsem osloven zakladatelem firmy panem Pavlem Konečným a požádán o spolupráci. V tuto chvíli je největším problémem nedostatek zvukových dat, pomocí kterých se algoritmy učí rozpoznávat nestandardní událost. Tuto situaci firma částečně řeší s využitím svého průmyslového zařízení s názvem NB.6 (viz tabulka 1.1, podrobněji v příloze F, Oficiální specifikace NB.6). Firma potřebovala vytvořit nové zařízení, které by díky svým menším rozměrům a nižším výrobním nákladům přineslo potřebná data pro další analýzy.

| Parametry | Hodnoty |
|---|---|
| Velikost | 165 x 145 x 55 mm |
| Váha | 1200 g |
| CPU | 4x ARM Cortex; 1,2 GHz |
| RAM | 1 GB |

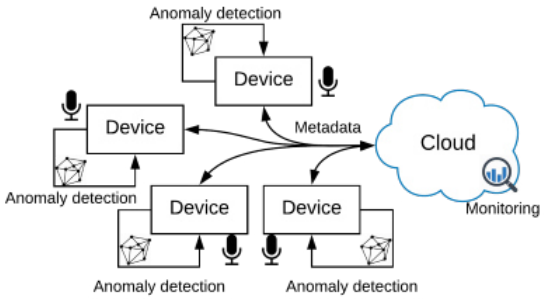
Tabulka 1.1: Specifikace NB.6[1].

Existují dva rozdílné pohledy na infrastrukturu a následné nasazení zařízení sloužících k detekci anomálií. Aktuálně firma vytvořila infrastrukturu zahrnující síť zařízení pro získání zvukových dat a síť zařízení sloužících k běhu algoritmů pro detekci anomálií (aktuálně v cloudu viz obrázek 1.1).




Obrázek 1.1: První pohled na infrastrukturu.

Do budoucna se však počítá i s možností vytvoření sítě „stand-alone“ zařízení, která by byla připojena k internetu, a kde by proudily jen informace o stavu zařízení. Hlavní algoritmy pro detekci anomálií by neběžely v cloudu, ale byly by provozovány v samotném zařízení (viz obrázek 1.2).




Obrázek 1.2: Druhý pohled na infrastrukturu.

Obě zmíněné infrastruktury počítají s provozem firmou vyvíjených algoritmů. V první popsané infrastruktuře probíhá na zařízení pouze sběr dat (vyhodnocování se provádí v cloudu). Ve druhé infrastruktuře „stand-alone“ typu se sběr dat i vyhodnocení provádí na samotném zařízení. Firma prozatím nevytvořila koncept zabezpečení firemních algoritmů proti krádeži. Na základě požadavků firmy byla vytvořena tato diplomová práce, která řeší problém chybějícího zařízení pro sběr dat ve firemním portfoliu a zároveň řeší problém možné krádeže „know how“ firmy.

[1]Převzato z: `http://neuronsw.com`

## 1.1 Úvod do problematiky

Problematice Internetu věcí byla věnována má předchozí bakalářská práce s názvem *„Internet of Things zařízení s podporou Bluetooth a CoAP“* [1]. Pro definici pojmu *„Internet věcí“* byla přejata část práce:

*„Internet věcí – obecně jde o označení nového trendu v oblasti informačních technologií využívajících různých zařízení s přístupem do sítě. Zařízení umožní sběr dat pomocí senzorů, jejich analýzu a zpracování. Data jsou následně využita nejčastěji pro zlepšení kvality lidského života.*

*Pro Internet věcí existuje mnoho různých definic podle různých pohledů na tuto problematiku.*

*‚Internet věcí můžeme chápat jako třívrstvý model, z čehož první vrstva je web (middleware). Druhá vrstva jsou jednotlivá zařízení (sensors) a třetí vrstva je sémantický model (knowledge). Pro úplnou využitelnost Internetu věcí musíme tyto tři modely propojit.‘ [2]*

*Jiná definice: ‚Internet věcí propojí objekty reálného světa s virtuálním světem, což umožní kdykoli a kdekoli komukoli se na cokoli připojit. Souvisí to se světem, kde fyzické objekty stejně jako virtuální data vzájemně spolu v čase interagují.‘ [3]*

*Další možná definice: ‚Je důležité pochopit zvláštní význam slova věci ve spojení Internet věcí. Věc může být reálná i virtuální. Tahle věc je vždy spojena s digitálním světem prostřednictvím bezdrátové komunikace. Jedna může najít ostatní kdekoli ve vesmíru.‘ [4]*

*Mezinárodní telekomunikační unie roku 2012 vydala dokument s názvem Overview of the Internet of things [5], v rámci kterého definuje několik pojmů včetně Internetu věcí takto: Jedná se o globální infrastrukturu pro IT společnosti, která umožní využití pokročilých služeb propojením (fyzických i virtuálních) věcí na základě stávajících a vyvíjejících informačních a komunikačních technologií.*

- *Poznámka 1 – prostřednictvím identifikace, sběru dat, zpracování a komunikačních schopností Internet věcí vytváří celé spektrum využití věcí a zároveň zajišťuje, aby byly splněny požadavky na bezpečnost a soukromí.*
- *Poznámka 2 – z obecnějšího pohledu může být Internet věcí vnímán jako vize se sociálními a technologickými dopady.*

*Dále pak popisuje rozdíl mezi zařízením a věcí. Zařízení musí splňovat schopnost komunikace s ostatními zařízeními a mělo by poskytovat aspoň jednu z možností snímání, sběr dat, ovládání či zpracování dat. Věc je v kontextu Internetu věcí předmět fyzického nebo informačního světa, u které je možné zajistit připojení a komunikaci s internetem.*

*Zařízení spadající do kategorie Internetu věcí můžeme rozdělit:*

- *Vysílač dat – zařízení je přímo připojeno na fyzickou ‚věc‘ a umožňuje bezdrátové připojení této ‚věci‘ do sítě.*

- *Přijímač dat – zařízení schopné čtení/zápisu dat. Umožňuje také interakci s fyzickou ‚věci'. Tato interakce může probíhat nepřímo s vysílačem dat nebo přímo s datovým nosičem.*
- *Senzor – zařízení detekuje informace z okolního prostředí a převádí je do digitální podoby.*
- *Obecné zařízení – dokáže komunikovat s internetem přes kabelové nebo bezdrátové spojení. Obecné zařízení může být také sada fyzických věcí.*

*Nové interakce přinášejí nové možnosti využití v mnoha oborech. Můžeme mluvit o chytrých domácnostech s interakcí s chytrými automobily. Vznik nositelných zařízení pro monitorování jednotlivých uživatelů. Interakce s mobilními telefony a mnoho dalších aplikací. Díky těmto zařízením dokážeme zlepšit kvalitu mnoha poskytovaných služeb, které budou moci být lépe zacíleny na konkrétního uživatele.“*

## 1.2 Zabezpečení a hrozby

Zabezpečení, a s tím spjaté bezpečnostní hrozby, je jedno z hlavních témat souvisejících právě s Internetem věcí. Všudypřítomná, vzájemně propojená a inteligentní zařízení spadající do kategorie Internetu věcí nabízejí poměrně vysokou výpočetní sílu. Shromažďování a zpracovávání soukromých informací z těchto zařízení činí ideální cíl pro kybernetické útoky.

Otázky bezpečnosti, jako je kontrola přístupu, bezpečná komunikace, soukromí a bezpečné ukládání dat, se stávají významnými problémy. Rychlý celosvětový nárůst počtu zařízení vedl k nasazení ne příliš dobře zabezpečených uzlů (často s výchozími přihlašovacími údaji). Tyto uzly mohou posloužit k nejrůznějším kybernetickým aktivitám jako např. DDoS útoky[2], spamming, man-in-the-middle útoky[3] a další.

Podle článku *„Internet of Things security and forensics: Challenges and opportunities“* [6] publikovaném v lednu 2018 v časopise Future Generation Computer Systems jsou hlavní bezpečnostní výzvy v této oblasti autentizace, autorizace, řízení přístupu, ochrana soukromí a zabezpečení infrastruktury. V rámci autentizace je vyzdviženo nasazení, správa a zabezpečení klíčů sloužících právě k autentizaci. V otázce ochrany soukromí autoři podotýkají skutečnost, že zařízení spadající do Internetu věcí primárně shromažďují soukromé informace většinou bez vědomí osob v okolí. Pravidla pro ochranu osobních údajů zahrnují utajení osobních údajů, jakož i možnost kontrolovat, co se s těmito informacemi stane (tato možnost může být mnohdy v heterogenní síti Internetu věcí problém).

[2] Útoky vedené z mnoha zařízení připojených k internetu za cílem znepřístupnit konkrétní služby pro ostatní uživatele.

[3] Útočník se stane prostředníkem v komunikaci dvou obětí. Veškerá komunikace je následně řízena útočníkem, bez povšimnutí zbylých účastníků.

Kapitola 2

# Analýza a návrh

## 2.1 Požadavky na systém

S vedením firmy Neuron soundware byly dohodnuty primární požadavky, které by měl zvukový senzor splňovat. Požadavky vycházely z jejich dřívější zkušenosti, stávající infrastruktury a byly přizpůsobeny budoucímu směru rozvoje firmy.

- Operační systém Linux.
- Nižší cena oproti modelu NB.6.
- Menší rozměry oproti modelu NB.6.
- Kompatibilní se zvukovými senzory.
- Rychlé připojení k internetu pomocí SIM karty pro odesílání dat a pro vzdálenou správu.
- Mechanické a elektronické zabezpečení firmware proti krádeži.

Na základě těchto primárních požadavků došlo k analýze možných řešení (viz níže) a návrhu finálního řešení.

## 2.2 Hardwarová platforma

Zvukový senzor pro sběr dat a odesílání dat do cloudu se skládá z několika základních komponent, a to z jednodeskového počítače (dále označován také jako základní modul), periferie pro nahrávání zvuku a periferie pro připojení k mobilnímu internetu pomocí SIM karty pro odesílání dat. Některé základní moduly obsahují již podporu Wi-Fi či kabelového internetu. Pro instalaci zvukového senzoru v průmyslových oblastech se často na tyto typy připojení nedá spolehnout z důvodu nedostupnosti kabelového internetu nebo Wi-Fi sítě.

### 2.2.1 Základní modul

Na trhu existuje mnoho jednodeskových počítačů, které se od sebe liší velikostí, cenou, spotřebou a výkonem. Bylo třeba zvážit možné použití, vyhovění primárním parametrům a dostupnost. Dále pak byl důležitý parametr podpora komunity, kompatibilita, množství rozšiřujících modulů a kvalitní dokumentace.

**Raspberry Pi 3 model B**

Jednodeskový počítač Raspberry Pi 3 model B [10] je nejnovější, nejvýkonnější (a také nejdražší) jednodeskový počítač vyvinutý nadací Raspberry Pi Foundantion. První verze tohoto počítače byla vytvořena pro podporu výuky informatiky v britských školách. Jednoduchost a nízké pořizovací náklady vedly k masovému rozšíření této desky, která splňuje všechny aspekty klasického stolního počítače. Jednotlivé generace se od sebe liší zejména konektivitou a výkonem. Ze tří generací klasických Raspberry Pi vybočuje pouze tzv. Raspberry Pi Zero nižší cenou a rozměry (viz níže).

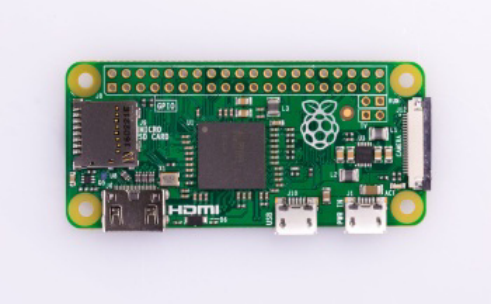
V roce 2016 byla představena třetí generace Raspberry Pi. Velikostně je srovnatelná s předešlou verzí. Došlo však k nárůstu výkonu především díky čtyřjádrovému, 64 bitovému procesoru ARMv8 s taktem 1,2 GHz. Velikost operační paměti 1 GB je stejná jako u předešlé verze. Deska se vyznačuje podporou Wi-Fi 802.11 b/g/n a Bluetooth 4.1 LE. Grafické jádro VideoCore pracuje na frekvenci 400 MHz. Výrobce udává nárůst celkového výkonu o 40% oproti druhé generaci. V důsledku toho se zvýšily požadavky na napájení a chlazení procesoru. Jednou z velkých nevýhod desky je absence rozhraní pro připojení pevného disku typu SATA. Disky lze připojit pouze pomocí USB. Deska není příliš vhodná pro případy použití, kde je zapotřebí velký přenos dat. Cena třetí generace ve verzi model B je aktuálně na Rpishop.cz 949,00 Kč.

Obrázek 2.1: Raspberry Pi 3 model B[4].

[4]Převzato z: `https://www.raspberrypi.org`

### Raspberry Pi Zero W

Již zmíněná verze Raspberry Pi Zero W [10] také od společnosti Raspberry Pi Foundantion se vyznačuje zejména menšími rozměry. Na úkor této vlastnosti došlo také ke snížení výkonu. Ubyl počet konektorů a snížila se i spotřeba. Zero má stejný procesor jako první generace Raspberry Pi, a to Broadcom BCM2835 s taktem 1 GHz (zvýšení o 300 MHz oproti první generaci Raspberry Pi). Operační paměť velikosti 512 MB je dostačující pro standardní úkony zařízení spadajících do Internetu věcí. Dále deska obsahuje konektor Mini-HDMI pro výstup zvuku a videa, jeden USB 2.0 konektor, čtečku micro SD karet a 40 GPIO pinů. Desku je nutné napájet pomocí micro USB. Znak W znamená doplnění podpory Wi-Fi a Bluetooth oproti původní verzi. Tato deska, díky svému poměru ceny a nabízeného výkonu s malými rozměry, se zasloužila o významný rozvoj Internetu věcí. Cena na Rpishop.cz je aktuálně 313,00 Kč.



Obrázek 2.2: Raspberry Pi Zero W[5].

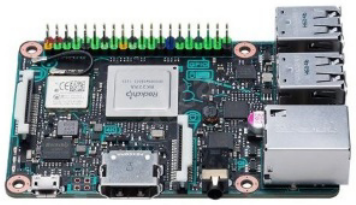
Výhodou všech desek od společnosti Raspberry Pi Foundantion je dobrá distribuční síť v České republice, velké množství přídavných modulů a rozšíření. Desky podporují velké množství linuxových distribucí. Nejznámější je operační systém Raspbian založený na Debianu.

### Asus Tinker Board

Deska od společnosti Asus nazvaná Tinker Board [11] je přímým konkurentem desky Raspberry Pi 3. Byla představena v první polovině roku 2017. Z obrázků 2.2 a 2.3 je patrné, že obě desky jsou si velmi podobné. Je to jeden z prvních faktů, které svědčí o určité kompatibilitě. Společnost Asus si uvědomuje, že desky od společnosti Raspberry Pi Foundantion tvoří velký ekosystém, proto společnost zvolila strategii vytvoření rozměrově stejné desky, která může konkurovat větším výkonem. Nevýhodou je vyšší spotřeba desky. Díky této vlastnosti je jasné, že deska cílí spíše na případy užití, kdy je potřeba vyšší výpočetní výkon, a ne na oblast trhu s dlouhodobou výdrží zařízení napájeno z baterie.

[5] Převzato z: `https://www.raspberrypi.org`

O výkon se stará procesor Rockchip RK3288 s jádrem Cortex-A17 a taktem 1,8 GHz. Operační paměť má velikost 2 GB, což je dvakrát více než konkurent Raspberry Pi 3. Asus pro svou desku vytvořil vlastní linuxový operační systém TinkerOS, který vychází z Debianu. Deska také podporuje např. Flint OS[6] či Android 6. Kompatibilita je podobná jako u desky Raspberry, jen obsahuje rychlejší 1 GB Ethernet. Cena na Alza.cz je aktuálně 1 499,00 Kč.



Obrázek 2.3: Asus Tinker Board[7].

**Orange Pi Zero Plus 2**

Deska s názvem Orange Pi Zero [12] pochází z rodiny jednodeskových počítačů Orange Pi od čínské společnosti Shenzhen Xunlong Software. Tato deska je přímým konkurentem stejnojmenné první generace Raspberry Pi Zero, jelikož deska Raspberry první generace neměla podporu Wi-Fi. Tuto výhodu však Orange Pi Zero ztratilo při představení již zmíněné verze Raspberry Pi Zero W.

O výkon se stará Quad core Cortex-A7 s taktem 1,2 GHz a grafický chipset Mali400MP2. Operační paměť je velká 512 MB. Zajímavostí je úložiště na desce eMMC[8] flash (8 G). Na desce se nachází Wi-Fi anténa (IEEE 802.11 b/g/n). Deska také podporuje Bluetooth ve verzi 4.2. Je zaručena kompatibilita s operačními systémy pro Raspberry Pi, dále pak Ubuntu, Debian a Android. Cena na Postavrobota.cz je aktuálně 849,00 Kč.

**NanoPi 2 Fire**

Deska s názvem NanoPi 2 Fire [13] je vyvinuta společností FriendlyARM. Svými rozměry je podobná Raspberry Pi Zero. O výkon se stará procesor Samsung Cortex-A9 Quad Core S5P4418, 1,4 GHz s operační pamětí 1 GB DDR3. Oproti Raspberry nabízí 1 GB Ethernet port, nepodporuje však Bluetooth ani Wi-Fi. Pokud bychom chtěli Bluetooth a Wi-Fi na úkor Ethernet portu, existuje verze NanoPi 2 s touto konfigurací.

[6]Free open-source operační systém od společnosti Flint vycházející z Chromium OS.
[7]Převzato z: `https://www.asus.com`
[8]Alternativní a méně složitý typ úložiště k SSD.
[9]Převzato z: `https://www.postavrobota.cz`

Obrázek 2.4: Orange Pi Zero Plus 2[9].

NanoPi 2 Fire obsahuje stejné GPIO konektory jako rodina Raspberry. Integrováno je rozhraní HDMI a LCD. Od výrobce je doporučeno chladit procesor desky. Pasivní chlazení procesoru bývá zahrnuto v základním balení. Napájení obsahuje podporu softwarového vypnutí, uspání a probuzení pro snížení okamžité spotřeby. Výrobcem je deklarovaná podpora operačního systému Android a Debian. Cena na portálu Neven.cz je aktuálně 1 525,00 Kč.

Obrázek 2.5: NanoPi 2 Fire[10].

**Nano Pi Neo Air**

Společnost FriendlyARM má ve svém portfoliu další desku s názvem Nano Pi Neo Air [13]. S rozměry 40 x 40 mm se jedná o nejmenší desku z již zmíněných. Je osazená procesorem Quad Core A7 s taktem 1,2 GHz. Dále nabízí 512 MB RAM, 8 GB eMMC a slot pro Micro SD kartu. Má Wi-Fi 802.11b/g/n a Bluetooth 4.0, 24 GPIO pinů a DVP rozhraní pro připojení externího modulu CAM500B (kamera).

Opět je výrobcem doporučeno chladit procesor desky pasivním chlazením dodávaným společně s Wi-Fi anténou v balení. Deska podporuje operační systém FriendlyCore, Debian a Ubuntu (od společnosti FriendlyARM je Ubuntu

[10]Převzato z: `http://www.neven.cz`

Core) z micro SD karty nebo z eMMC úložiště. Cena na portálu Neven.cz je aktuálně 1 210,00 Kč.

Obrázek 2.6: Nano Pi Neo Air[11].

### 2.2.2 Nahrávání zvuku

Firma Neuron soundware potřebuje pro svůj případ užití získat kvalitní zvukové nahrávky, ze kterých následně provádí detekci anomálií. Kvalitu výsledného zvukového záznamu z okolí ovlivňuje několik kritérií týkajících se samotného mikrofonu (směrová a frekvenční charakteristika, citlivost atd.) či zvukové karty (frekvence vzorkování, AD/DA převodník, počet kanálů atd.).

**Zvuková karta**
Jednodeskové počítače většinou nemají v základní konfiguraci podporu přímého nahrávání zvuku, jelikož ve většině případů desky neobsahují mikrofon nebo dokonce zvukovou kartu, a tak je třeba tento problém vyřešit pomocí připojení periferie. Periferie jsou primárně vyvíjeny pro konkrétní desku (většinou Raspberry Pi), ale většina zmíněných jednodeskových počítačů (viz podsekce 2.2.1, Základní modul) podporuje stejné přídavné periferie jako právě Raspberry Pi. K některým menším deskám je navíc potřeba dokoupit pro připojení redukci.

Pro zpracování analogového/digitálního zvuku je potřeba zvuková karta. Existuje mnoho typů těchto karet. Pro jednodeskové počítače se lze nejčastěji setkat se zvukovými kartami připojenými prostřednictvím USB. Jedna z nevýhod tohoto řešení je zaplnění jednoho USB portu, což může být významný problém pro menší desky.

Zástupcem této kategorie je zvuková karta AXAGON ADA-25 USB [14] (viz obrázek 2.7). Nabízí možnost vzorkovat signál až frekvencí 96 kHz. Má

[11] Převzato z: `http://www.friendlyarm.com`

LED indikaci „mute“ mikrofonu a tlačítka pro ovládání zapnutí vstupu a hlasitosti výstupu. Provedení této zvukové karty je jednoduché a její cena na portálu Rpishop.cz je aktuálně 359,00 Kč.

Obrázek 2.7: Axagon ADA-25 USB[12].

Existují i karty, které se připojují přes GPIO piny a poskytují zvukový vstup i výstup. Pro Raspberry Pi existuje seznam podporovaných zvukových karet připojovaných přes GPIO piny[13]. Je jich však mnohem méně než USB karet a jen některé poskytují nejenom výstup, ale i vstup pro připojení mikrofonu.

Příkladem zvukové karty připojované přes GPIO piny je karta Cirrus Logic Audio Card [15] (viz obrázek 2.8) od firmy Wolfson Microelectronics. Dokáže vzorkovat signál rychlostí až 192 kSa/s. Obsahuje dva DMIC mikrofony umožňující stereo záznam. Dále je možné pro vstup využít konektor 3,5 mm jack, a také S/PDIF[14] pro vstup/výstup. Je osazena výkonným zesilovačem. Všechny tyto vlastnosti souvisí s vyšší pořizovací cenou, která je na portálu Adafruit.com aktuálně 50 $ (přibližně 1050 Kč).

**Mikrofon**

Na trhu existuje mnoho různých mikrofonů, které se liší cenou, konstrukcí, frekvenční charakteristikou[16] či principem nahrávání zvuku. Zvukový senzor vyvíjený v rámci diplomové práce by měl být kompatibilní s mikrofony používanými firmou Neuron soundware. Tato firma vyvinula několik prototypů mikrofonů, které nepotřebují externí napájení a jsou používány i se zařízením NB.6 vyvinuté firmou. Mikrofony se připojují prostřednictvím RCA konektorů.

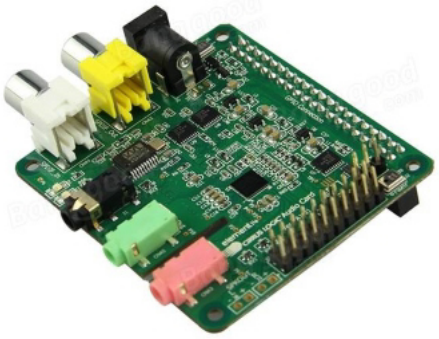
[12] Převzato z: `https://www.rpishop.cz`

[13] Více na: `https://elinux.org/RPi_VerifiedPeripherals`

[14] Hardwarové protokoly pro přenos digitálně kódovaného zvukového signálu mezi různými audio komponentami.

[15] Převzato z: `https://www.banggood.com`

[16] Závislost změn amplitudy a fázového posunutí na frekvenci signálu.



Obrázek 2.8: Cirrus Logic Audio Card[15].

**Formáty bezeztrátové komprese**
Aktuálně firma Neuron soundware počítá s přenosem zvukových záznamů pomocí internetového připojení do cloudu, kde dochází k jejich zpracování. V tomto případě by bylo vhodné uvažovat o možném zmenšení zvukových záznamů pomocí bezeztrátové komprese.

Zvukové formáty, které se vyznačují bezeztrátovou kompresí snižují velikost souborů bez ztráty informace. Díky bezeztrátové kompresi lze na rozdíl od komprese ztrátové rekonstruovat zvukový záznam do své originální podoby. Hlavní bezeztrátové formáty jsou FLAC, WMA, APE a ALAC. Kompresí pomocí FLAC formátu lze dosáhnout 60% velikosti původního souboru. Nejlepší kompresní poměr[17] má formát APE (takřka 50% původní velikosti). Komprese je však velmi pomalá.

### 2.2.3 Připojení k internetu

Firma Neuron soundware počítá s přenosem dat ze zařízení pomocí internetového připojení (viz sekce 2.1, Požadavky na systém). Některé jednodeskové počítače mají již v základu integrovanou Wi-Fi (pokud ne, lze rozšířit pomocí USB adaptéru) či Ethernet konektor pro připojení pomocí kabelu. Tyto možnosti jsou vítány (umožní zlevnit provoz zařízení). Ředitel firmy Pavel Konečný zdůraznil, že z předchozích zkušeností ve výrobních halách klientských společností je obtížně splnitelný požadavek dostupnosti kabelového nebo bezdrátového připojení. Jedná se o hlavní důvod, proč byl zvolen mobilní internet prostřednictvím SIM karty mobilního operátora.

Je nutné zmínit, že cena použití mobilního internetu souvisí s přenosovou rychlostí a silně se odvíjí od tarifu mobilního operátora v dané lokalitě instalace. Firma Neuron soundware počítá např. s instalací v Německu, Slovensku či Velké Británii a SIM karty od českého operátora nemusí mít dostatečné

[17]Podíl velikosti nekomprimovaných dat k velikosti komprimovaných.

pokrytí v dané lokalitě. Cena mobilního internetu má však tendenci mírně klesat.

Pro připojení k mobilnímu internetu je možno využít několik možností. První možností je využít speciální moduly, které rozšíří jednodeskový počítač o tuto funkcionalitu. Jeden z takových modulů je SparqEE GSM Cellular Board [16] (viz obrázek 2.9). Jedná se o přídavnou desku poskytující připojení k 2G a 3G síti pomocí SIM karty. V průměrném provozu má spotřebu okolo 500 mA. Pro snadné připojení k desce Raspberry Pi je doporučován SparqEE ShieldR B+/2/3 pro snadnější připojení. Velkou nevýhodou je cena tohoto sestavení, která se při spojení ShieldR + SparqEE GSM Cellular Board přenesla přes 3600 Kč.

Obrázek 2.9: SparqEE GSM Cellular Board[18].

Alternativní variantou je velké množství dostupných LTE modemů do USB s nativní podporou Linuxu. Jedním z těchto modemů je např. Huawei E3372h. Dle výrobce se k síti 4G připojí rychlostí až 150 Mb/s, k dalším sítím 3G-DC-HSPA+ až 42 Mb/s. Obsahuje potřebné ovladače a jeho cena je méně než poloviční k již zmíněné SparqEE GSM Cellular Board. Na portálu Alza.cz činí 1347,00 Kč.

### 2.2.4 Zdroj napájení

Existuje mnoho způsobů, jak napájet jednodeskové počítače či menší moduly spadající do Internetu věcí. První variantou je možnost napájet zařízení z elektrické sítě prostřednictvím napájecího zdroje. Jednodeskové počítače jsou většinou napájené prostřednictvím micro USB portu, který je standardizován. Tak lze použít jeden typ napájecího zdroje na všechny desky (např. oficiální micro USB napájecí zdroj pro Raspberry Pi stojí na Rpishop.cz aktuálně 229,00 Kč, jeho základní specifikace uvádí tabulka 2.1). Je třeba si však dát pozor na typ koncovky do zásuvky používaný v dané zemi.

Další možností napájení desek je pomocí baterie. V dnešní době existuje mnoho tzv. „Power bank“ (viz obrázek 2.10). Jsou to velkokapacitní akumulátory s možností opětovného nabití, které mají většinou výstup prostřednictvím

[18] Převzato z: `http://rpishop.cz`
[19] Převzato z: `http://rpishop.cz`

| | |
|---|---|
| Výstupní napětí | 5 V DC |
| Výstupní proud | 2.0 A |
| Jmenovitý výkon | 10 W |
| MTBF | 50000 h |

Tabulka 2.1: Specifikace oficiálního zdroje Raspberry Pi[19].

USB portu (je třeba redukci na micro USB). Hlavní specifikace těchto akumulátorů je jejich kapacita udávaná v jednotkách miliampérhodina (zkratka mAh). Jakmile dojde k vybití power banky, jednodeskový počítač se vypne a power banku je třeba nabít znovu prostřednictvím USB portu.


Obrázek 2.10: ADATA Power Banka 10000 mAh[20].

Společnost Adafruit vytvořila pro potřebu napájení desek modul s názvem Micro Lipo s micro USB konektorem [17] (viz obrázek 2.11). Nabízí podobnou možnost napájení jako připojení power banky prostřednictvím USB portu. Rozměry modulu jsou však výrazně menší. K modulu se následně připojí lithium-polymer nebo lithium-ion dobíjecí akumulátor (na obrázku 2.11 typ Li-Ion) prostřednictvím JST konektoru. Je na uživateli, s jak velkou kapacitou akumulátor zvolí. Cena tohoto modulu na Rpishop.cz je aktuálně 229,00 Kč.

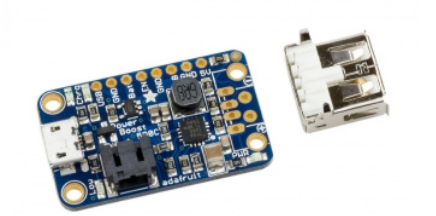
Nevýhoda napájení ze síťového napájecího zdroje i akumulátoru je taková, že jakmile se zdroj odpojí ze sítě nebo dojde k vybití akumulátoru, zařízení se vypne. Akumulátor je navíc třeba znovu externě nabít. Pokud však zkombinujeme oba tyto způsoby napájení, dostaneme zařízení souhrnně označené UPS (Uninterruptible Power Supply) napájeno ze zdroje. Jakmile se však zdroj odpojí, UPS plynule přepne napájení ze sítě na napájení z akumulátoru. Po znovupřipojení zdroje do elektrické sítě dobije akumulátory a zároveň poskytuje elektrický proud pro zařízení.

Zařízení UPS vhodných pro jednodeskové počítače na trnu není zdaleka tolik, jako např. již zmíněných power bank. Společnost Adafruit nabízí modul

[20]Převzato z: `https://www.compareraja.in`
[21]Převzato z: `https://blog.adafruit.com`


Obrázek 2.11: Adafruit Micro Lipo[21].

s názvem PowerBoost 500 Charger [17] (viz obrázek 2.12). Tento rozšířený modul obsahuje oproti modulu Micro Lipo s micro USB konektorem další USB konektor. K modulu je tedy v jednu chvíli připojena baterie, zdroj i modul, který je napájen. Modul PowerBoost však není k dostání v České republice. Jeho cena na Modmypi.com je 13,99 £ (aktuálně 402,93 Kč).



Obrázek 2.12: PowerBoost 500 Charge[22].

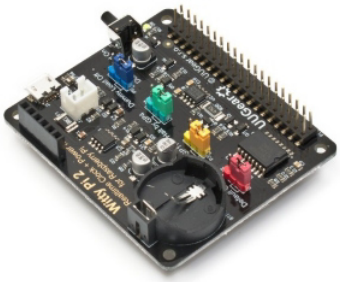
Alternativou k dostání na českém trhu je modul s názvem PiZ-UpTime (viz obrázek 2.13). Jedná se o UPS modul, který je svými rozměry přizpůsoben Raspberry Pi Zero. Tento modul je osazen držákem pro AAA baterii. Připojení k desce je možné pomocí GPIO pinů nebo pomocí extra výstupního micro USB portu. Stav baterie lze softwarově monitorovat a při kritickém stavu baterie dojde k vypnutí zařízení. Cena tohoto modulu na Rpishop.cz je aktuálně 699,00 Kč (bez baterie). Výrobcem je doporučeno použít AAA baterii 3,7 V s kapacitou 2800 mAh.

V rámci napájení může být zajímavý i přídavný správce napájení. Většina jednodeskových počítačů nemá ani zapínací tlačítko. Jakmile se přivede napájení na desku, zařízení se zapne automaticky. Je mnoho názorů, jak nejšetrněji zařízení vypnout, protože pokud zrovna zařízení zapisuje data na micro SD

[22]Převzato z: `https://www.modmypi.com`
[23]Převzato z: `http://rpishop.cz`

Obrázek 2.13: PiZ-UpTime[23].

kartu, může dojít k poškození této karty. Nejen šetrný management napájení, ale i jeho správu poskytuje přídavný modul Witty Pi 2 [18] (viz obrázek 2.14). Tento modul obsahuje tlačítko pro zapnutí, obvod reálného času (RTC) pro plánování zapnutí a vypnutí a externí baterii (ta však má primární účel napájet RTC obvod a není určena k napájení jednodeskového počítače). Cena modulu na Rpishop.cz je aktuálně 579,00 Kč.



Obrázek 2.14: Witty Pi 2[24].

Na závěr je třeba zmínit, že způsob napájení se odvíjí od celkové spotřeby sestaveného zařízení složeného z dílčích komponentů. Je dobré otestovat možnou dobu napájení zařízení na baterii a podle výsledků zvolit, zda je tento způsob napájení vhodný. Doba běhu zařízení na baterii souvisí i např. s aktuálním vytížením procesoru. Je doporučováno zařízení důkladně otestovat při plné zátěži, aby se potvrdila možná doba běhu. Při napájení pomocí UPS může tento způsob sestavení sloužit také jako bezpečnostní prvek, který zajistí běh zařízení po vypojení napájení z elektrické sítě.

[24]Převzato z: `http://rpishop.cz`

## 2.3 Možnosti zabezpečení

Při vývoji zařízení z kategorie Internetu věcí je bezpodmínečně nutné zajistit odpovídající bezpečnost zařízení. Tuto oblast nelze podceňovat. Z praxe jsou známé případy, kdy byla zařízení z kategorie Internetu věcí zneužita ke kybernetickým útokům. V práci je proto podrobně rozebrána problematika několika doporučení jako:

- IEEE standard P2413.
- Security in the Internet of Things: A Review.
- Internet of things security best practices.
- Security-in-depth strategie.

Znalost bezpečnostních doporučení může výrazně ovlivnit bezpečnost a spolehlivost výsledného zařízení z kategorie Internetu věcí. Dokud nebudou přijaty závazné normy, jsou tato doporučení klíčová.

### 2.3.1 IEEE standard P2413

Mezinárodní nezisková organizace Institute of Electrical and Electronics Engineers (zkratka IEEE), která se podílí na vývoji průmyslových standardů, vytvořila již několik dílčích podkladů pro vytvoření celistvého standardu pro vývoj a architekturu Internetu věcí. Tento standard s názvem *„P2413 – Standard for an Architectural Framework for the Internet of Things“* [19] byl iniciován organizací již v roce 2014. Nyní je označen jako tzv. „Aktivní projekt“, což znamená, že byl zahájen jeho vývoj a stále je možné se podílet na jeho dalším vývoji.

Původní odhad dokončení standardu byl stanoven na konec roku 2016. Jelikož se postupem času k vývoji přidávalo více a více firem a organizací (v roce 2016 celkem 27 firem jako např. Cisco Systems, Intel, Honeywell International, SIGFOX, Huawei Technologies, IoTecha, Kaspersky Lab, Qualcomm a Siemens), které měly k vývoji připomínky, došlo k prodloužení doby vývoje tohoto standardu. Poslední významná schůzka proběhla 1. prosince 2017 v Shenzhenu iniciovaná firmou Huawei. Společnost předložila návrh zahrnující správu zařízení a připojení k internetu, aktivaci aplikací v IoT platformě, a také se věnovala analýze dat. Vedení vývojové skupiny souhlasilo, že do standardu P2413 doplní text navržený společností Huawei.

Jak již bylo zmíněno, standard je prozatím ve vývoji. K hlavním dostupným dokumentům patří reporty a prezentace ze setkání vývojové skupiny. Jedním takovým zdrojem je dokument vydaný v září 2016, který byl vytvořen předsedou vývojové skupiny Olegem Logvinovem. Tento dokument popisuje hlavní rysy standardu jako:

- Popis různých domén v rámci Internetu věcí, jejich abstrakce a identifikace společných znaků mezi různými doménami.
- Podpora interakce mezi doménami v rámci Internetu věcí.
- Zajištění „čtveřice“ zahrnující ochranu, bezpečnost, soukromí a zabezpečení.
- Popis architektonického rámce zachycujícího společné prvky v různých oblastech poskytujícího základ pro konkrétní architekturu Internetu věcí.

Zmíněný architektonický rámec by měl sloužit jako referenční model, který bude definovat vztahy mezi různými odvětvími Internetu věcí. V současné době existuje mnoho standardů, které popisují specifické oblasti v rámci Internetu věcí, které předseda vývojové skupiny Oleg Logvinov popisuje jako „ostrovy“ nesouvislého vývoje. Přijetím jednotného přístupu k vývoji systémů Internetu věcí se sníží roztříštěnost odvětví a povede k vytvoření společné množiny činností s více zainteresovanými společnostmi po celém světě.

### 2.3.2 Základní bezpečnostní principy

K seznámení a k pochopení základních bezpečnostních principů spjatých s problematikou Internetu věcí byl vydán článek s názvem *„Security in the Internet of Things: A Review“* [20]. Ten dělí Internet věcí na čtyři základní vrstvy, podle kterých se následně odvíjejí doporučené bezpečnostní principy.

- Percepční vrstva – obsahuje senzory pro zachycení fyzického světa a převod informací do digitální podoby.
- Síťová vrstva – zodpovědná za spolehlivý přenos informací z percepční vrstvy. Zahrnuje např. komunikační protokoly, internetovou infrastrukturu atd.
- Vrstva podpory – vytváří platformu pro běh aplikační vrstvy a souvisí se síťovou vrstvou. Zahrnuje např. cloud computing.
- Aplikační vrstva – poskytuje personalizované informační služby podle potřeb uživatelů. Uživatelé mají přistup k zařízení prostřednictvím internetu a např. mobilního telefonu.

Z definic výše zmíněných vrstev vyplývají bezpečnostní požadavky. Percepční vrstva vyžaduje zejména ověření jednotlivých uzlů pro zamezení přístupu neoprávněného uzlu, zachování důvěrného přenosu dat mezi jednotlivými uzly a šifrování přenosu.

V rámci síťové vrstvy Internetu věcí jsou velkou hrozbou DDoS útoky. Je tedy potřeba vytvořit mechanismus k zabránění útokům (někdy označován

jako Anti-DDoS). Tato vrstva dále zahrnuje šifrovací mechanismy, mechanismy pro zabezpečení komunikace a klientskou autentizaci.

Vrstva podpory vyžaduje zaměření na zabezpečení platformy, na které bude běžet aplikační vrstva. Jedná se např. o zabezpečení cloud computingu či použití antivirů.

Pro zabezpečení aplikační vrstvy je třeba vyřešit autentizaci a správu klíčů napříč heterogenní sítí a dále vyřešit ochranu soukromí uživatele. Tato vrstva také zahrnuje vzdělávání samotných uživatelů (např. správa hesel). Důsledkem všech těchto požadavků je existence ověřených bezpečnostních mechanismů, a to zejména mechanismů šifrování a spjatými šifrovacími algoritmy, mechanismy zabezpečení komunikace a mechanismy zabezpečení senzorových dat (popsáno níže).

**Mechanismus šifrování**
V rámci internetu se můžeme setkat se dvěma různými způsoby šifrování podle toho, kdo daná data šifruje. První mechanismus je tzv. *„End-to-end“* šifrování. Již podle názvu je patrné, že komunikaci šifruje a dešifruje pouze odesílatel a příjemce.

Pokud není potřeba nebo se z nějakého důvodu End-to-end šifrování do konkrétního případu užití nehodí, lze zvolit druhý možný mechanismus, šifrování *„Hop-by-hop“*. V každém uzlu dochází k dešifrování a následnému zašifrování pro jeden další konkrétní uzel, což znamená, že v jednom okamžiku jsou informace na každém uzlu v tzv. *„plaintextu“*[25].

**Zabezpečení komunikace**
Existuje několik již ověřených mechanismů, které poskytují dosažení integrity, autenticity a důvěrné komunikace, a to TLS/SSL nebo IPSec. TLS/SSL je určen k šifrování spojení transportní vrstvy a protokol IPSec je navržen tak, aby poskytl zabezpečení síťové vrstvy.

Označení TLS [7] (Transport Layer Security) či SSL [8] (Secure Sockets Layer) popisuje kryptografické protokoly používané k zabezpečení komunikace v rámci internetových služeb (vytváří šifrované spojení). Protokol TLS vychází z protokolu SSL (novější verze) a obsahuje tři základní fáze komunikace.

1. Účastníci komunikace se domluví na použitých šifrovacích algoritmech např. RSA a AES. (jsou potřeba symetrické i asymetrické šifry).

2. Účastníci komunikace si vymění klíče pomocí asymetrického šifrování a autentizace.

3. Účastníci komunikace zahájí výměnu dat za pomoci symetrického šifrování.

[25]Nezašifrovaná informace často ve formě jednotlivých znaků (resp. číslic odpovídajících kódům použité znakové sady, např. ASCII) bez formátovacích informací.

Označení IPsec [9] je bezpečnostní rozšíření protokolu IP na síťové vrstvě (původně vyvinut pro verzi IPv6). Je tedy na úrovni operačního systémů a zajišťuje bezpečný přenos dat jakékoliv síťové aplikaci na vyšších vrstvách ISO/OSI modelu[26]. Podporuje velké množství šifrovacích algoritmů, které se v současné době běžně používají. Zaměřuje se na tyto problémy zabezpečení:

- Důvěrnost dat – šifrování obsahu datagramů.
- Původ dat – ověření odesílatele každého datagramu.
- Integrita dat – ověření konzistence datagramu mezi odesláním a přijetím.

**Zabezpečení senzorových dat odpovídající GDPR**
Úroveň zabezpečení senzorových dat se odvíjí zejména od typu dat. Pokud by se jednalo např. o audio záznamy konverzace osob v továrně pro výrobu automobilů, byla by zde možnost vyzrazení důležitého výrobního tajemství. Firma by jistě chtěla chránit své informace před krádeží. Obecně se však dá říci, že pokud má útočník možnost umístit vlastní zařízení pro sběr dat na stejné místo, nepotřebuje krást data cizí (v továrně by takovou možnost neměl).

Je důležité vyřešit otázku ochrany soukromí lidí. Ti často nevědí o senzorech v jejich okolí a ani o skutečnosti, že jsou nějakým způsobem monitorováni. Je třeba vytvořit předpisy, které by dokázaly zachovat soukromí těchto osob. Základem použitých principů by mohlo být:

- Uživatelé musí být informováni o tom, že jsou nahráváni senzorovým zařízením.
- Uživatelé si mohou vybrat, zda chtějí být nahráváni, či nikoliv.
- Uživatelé musí být schopni zůstat anonymní.

### 2.3.3 IEEE bezpečnostní opatření

V rámci organizace Institute of Electrical and Electronics Engineers bylo vydáno několik článků věnujících se problému zabezpečení Internetu věcí. Jeden z těchto článků nazvaný *„Internet of things security best practices“* [21] se věnuje bezpečnostním doporučením týkajících se Internetu věcí. Výrazně tak rozšiřuje článek *„Security in the Internet of Things: A Review“* [20] (již zmíněný v podsekci 2.3.2, Základní bezpečnostní principy) zabývající se základními bezpečnostními principy. Problematiku dělí na tři základní skupiny:

- Zabezpečení zařízení.
- Zabezpečení sítě.
- Zabezpečení infrastruktury.

[26] Model, který rozděluje internetovou komunikaci mezi počítači do sedmi souvisejících vrstev.

**Zabezpečení zařízení**
V první fázi se souhrn bezpečnostních opatření věnuje zabezpečení jednotlivých zařízení. Tato fáze zabezpečení je také významná z pohledu návrhu a realizace zvukového senzoru vytvářeného pro potřeby firmy Neuron soundware. Problematiku dělí na konkrétní doporučení:

1. Zajištění hardwaru proti otevření (označení jako *„Tamper resistant“*).
2. Poskytování aktualizací firmware.
3. Dynamické testování.
4. Upřesnění popisu ochrany dat při likvidaci zařízení.

První doporučení nabádá k vytvoření mechanismu, který detekuje fyzickou manipulaci se zařízením. Manipulace může být např. přenesení z místa „A“ do místa „B“, otevření krytu zařízení či odstranění části zařízení. Tyto informace mohou být součástí toku dat přenášených do cloudu a měly by sloužit k upozornění na neočekávané události.

Pro fyzickou ochranu koncových bodů jsou zmíněny např. softwarové zámky portů, které pomáhají předcházet nežádoucímu malware. Dále je kladen důraz na rozdělení zabezpečení do vrstev, což vyžaduje, aby útočníci čelili řadě překážek určených k ochraně zařízení. Na úrovni hardwaru a zaváděcího softwaru to mohou být např. silná přístupová hesla. Dalším zabezpečením může být vytvoření šifrovacího klíče pro šifrování úložiště na základě připojených periferií a jejich specifických identifikátorů. Při absenci některé periferie by nedošlo k dešifrování dat.

Druhé doporučení nabádá výrobce zařízení k aktualizacím firmwaru během životního cyklu zařízení. S tímto důležitým krokem již v dnešní době řada výrobců nepočítá (zvyšuje to náklady na vývoj a provoz). V důsledku toho znemožňují jakoukoli reakci na možné hrozby, čímž dochází k nasazení do provozu zařízení bez adekvátního a hlavně aktuálního zabezpečení.

Třetí doporučení se věnuje fázím testování. Výrobci často preferují tzv. *„Statické“* testování, které nevyžaduje běh software, a tak je možné již průběžné testování v raných fázích vývoje. Jakmile je dokončen spustitelný software, mělo by následovat tzv. *„Dynamické“* testování, které dokáže odhalit nejen chybu v softwaru, ale i chyby v propojení softwaru a hardwaru (či jen hardwaru), které se nedají odhalit při statickém testování.

Další doporučení pokračuje v problematice zastarávání zařízení. Pokud se uživatel rozhodne nahradit stávající zařízení novou verzí, měl by dohlédnout na správnou likvidaci zastaralého zařízení a zejména na likvidaci dat uložených v paměti zařízení. Při pouhém vyřazení zařízení z provozu by zařízení nemělo obsahovat soukromé údaje a jiná citlivá data, která by se takto mohla dostat do nesprávných rukou.

**Zabezpečení sítě**
Druhá fáze souhrnu bezpečnostních opatření organizace IEEE se věnuje zabezpečení propojení a komunikaci jednotlivých zařízení Internetu věcí v síti. Tato část obsahuje popis čtyř doporučení:

1. Použití silné autentizace.
2. Použití silného šifrování a bezpečných protokolů.
3. Omezení síťového provozu pouze na potřeby zařízení.
4. Rozdělení sítě do segmentů.

První doporučení mluví o hrozbě snadno „prolomitelných“ uživatelských jménech a heslech (např. „admin“ a „admin“). Jednotlivá zařízení by měla obsahovat ve výchozím stavu již od výrobce jedinečné uživatelské identifikační údaje a uživatel by měl mít možnosti jednoduše tyto výchozí hodnoty změnit. Velmi důležitá je také kontrola síly hesel, aby mohla odolat snaze o prolomení hlavně pomocí útoku hrubou silou[27].

Další úrovní bezpečnosti autentizace je použití tzv. *„dvoufaktorové“* autentizace (zkratka 2FA), která vyžaduje, aby uživatel použil jak heslo, tak další ověřovací informaci, která nesouvisí se znalostí uživatele. Příkladem může být náhodný řetězec znaků, který uživatel obdrží většinou prostřednictvím SMS zprávy.

V rámci Internetu věcí je také doporučovaná tzv. *„adaptivní“* autentizace. Využívá kontextové informace (např. behaviorální biometriku, která porovnává chování a interakci se zařízením a jeho periferiemi konkrétního jedince) a algoritmy pro strojové učení, které průběžně vyhodnocují riziko bez vědomí uživatele. Pokud je riziko vysoké, uživatel je požádán o dodatečnou autentizaci.

I když je zařízení chráněné heslem, stále je možné odposlouchávat komunikaci. Druhé doporučení proto nabádá k důkladnému zvážení použitého internetového protokolu a bezpečnostních principů založených na šifrování. Existuje mnoho protokolů, včetně Bluetooth, 6LoWPAN, Zigbee, Z-Wave, Wi-Fi, NFC, Sigfox a LoRaWAN. V závislosti na protokolu a dostupných výpočetních prostředcích může být zařízení schopné používat šifrování. Je vhodné analyzovat různé případy užití a na základě této analýzy se rozhodnout pro co nejsilnější šifrování. Nejlépe IPsec či TLS a SSL.

Třetí doporučení souvisí s možností útočníků využít zařízení spadající do Internetu věcí pro DDoS útoky. Většina zařízení jsou vyrobena z široce rozšířených komponent, které mohou mít značně předimenzované síťové schopnosti pro funkci, kterou mají zařízení vykonávat. Tento fakt potenciálně přispívá k masivnějším DDoS útokům na internetu (viz uvedený příklad).

[27] Systematické testovaní všech možných kombinací daného řetězce.

*Předpokládejme, že v budoucnu bude připojeno 50 miliard zařízení k internetu (některé analýzy přiřazují toto číslo roku 2023). Dále předpokládejme (na základě současných analýz), že 1,1% zařízení je zneužitelných pro DDoS útoky (550 milionů). Odhadněme, že z toho počtu 10% bude spadat do Internetu věcí, což je 55 milionů zařízení. Předpokládejme, že každé zařízení bude schopno generovat útok s lineární rychlostí ekvivalentní gigabitovému Ethernetu (od 81274 do 1488096 snímků za sekundu, podle velikosti snímků), např. ARM9 (System-on-a-chip[28]) nabízí rovnou dvě taková připojení a stojí méně než 5 dolarů. Útočník, který by dokázal zneužít všechna dostupná zařízení z Internetu věcí by mohly generovat mezi 4,47 a 81,8 biliony snímků za sekundu (řádově desítky petabitů za sekundu). Takové množství je mimo obrané schopností běžného poskytovatele internetových služeb.*

Pokud bychom snížili schopnosti generování počtu rámců za sekundu, výrazně se sníží i potencionální hrozba spjata s DDoS útoky. Zařízení by dále měla sloužit jako aktivní prvek v monitorování sítě a měla by obsahovat mechanismus, který by dokázal zařízení při detekci hrozby např. obnovit do továrního nastavení či alespoň restartovat a vymazat kód, který útočník nahrál do paměti.

Poslední doporučení ve fázi zabývající se zabezpečením sítě pojednává o rozdělení sítě do menších segmentů pomocí tzv. VLAN[29], rozsahu IP adres či jejich kombinací. Tyto metody umožňují organizacím vytvářet bezpečnostní zóny, které dokážou izolovat potencionálně nebezpečné zařízení od zařízení obsahující citlivá data.

**Zabezpečení infrastruktury**
Poslední fází bezpečnostních doporučení zařízení z Internetu věcí nahlíží na celkovou infrastrukturu. Obsahuje tři základní doporučení:

1. Ochrana citlivých informací.
2. Podpora etického hackingu a zveřejňování zranitelností.
3. Vytvoření institutu pro kontrolu bezpečnostních principů.

Prvního doporučení se týká ochrany citlivých informací o zařízení, které by mohl útočník využít k zacílení či šíření útoku jako je identifikační číslo zařízení nebo počet připojených sousedních zařízení. Jakýkoliv bezpečnostní mechanismus, který znemožní útočníkovy tzv. „scan“ síťové infrastruktury, zvýší možnost zacílení útoku na nejzranitelnější místo.

Druhé doporučení obsahuje tzv. *„Etický hacking“*. Aby bylo možné odstranit chyby v zabezpečení, musí výrobci hardwaru a vývojáři softwaru nejprve

[28] Integrovaný obvod, který zahrnuje všechny součásti počítače nebo jiného elektronického systému do jediného čipu.
[29] Logicky nezávislá síť v rámci jednoho nebo několika zařízení.

vědět, že tyto chyby v zabezpečení existují. Lidé, kteří objevují vážné bezpečnostní zranitelnosti a zodpovědně informuje výrobce či vývojáře, výrazně napomáhají k zlepšení současné situace.

Třetí doporučení pojednává o vytvoření zodpovědného institutu pro definování a následnou kontrolu bezpečnostních principů spjatých s vývojem zařízení z Internetu věcí. Institut by poskytoval profesionální certifikační program pro vývojáře nových technologií spjatých s Internetem věcí, kteří by se zavázali dodržovat osvědčené bezpečnostní postupy po celý životní cyklus zařízení. Institut by měl také pravomoc postihovat vývojáře, kteří by nechtěli dodržovat bezpečnostní standardy. Certifikační orgán by měl ověřit následující vlastnosti:

1. Dochází k ochraně dat při přenosu a zpracování.
2. Je použit protokol, který zabraňuje krádeži citlivých informací.
3. Pokud se objeví bezpečnostní problém, poskytovatel na tento problém bude okamžitě reagovat.
4. Je vyžadována silná autentizace.
5. Zařízení nejsou nedostatečně chráněna před okolními vlivy.
6. Zařízení obsahují identifikační štítek s webovým odkazem na stránku s ověřením informací o zařízení a jejich certifikační status.

### 2.3.4 Security-in-depth strategie

Společnost Microsoft si uvědomuje rizika spjatá se špatným zabezpečením Internetu věcí. Vydala dokument rozšiřující již zmíněné články zabývající se bezpečnostními principy v rámci Internetu věcí.

Společnost rozdělila problematiku na několik částí (viz níže). Souhrn těchto doporučení označuje jako tzv. *„Security-in-depth strategii“* [22]. Hlavními rysy je ochrana dat v cloudu, ochrana a integrita přenosu dat internetem a fyzická ochrana zařízení. Všechna tato opatření zvyšují zabezpečení dané instance infrastruktury. Jednotlivé části *„Security-in-depth strategie“* jsou:

- Hardwarový vývoj a integrace.
- Řešení vývoje.
- Řešení nasazení.
- Řešení řízení.

**Hardwarový vývoj a integrace**
Fáze hardwarový vývoj a integrace je první fáze *„Security-in-depth strategie“* a souhrnně označuje možné bezpečnostní opatření výrobců hardwaru Internetu věcí. Pokud se jedná o složitější zařízení, v dnešní době si tyto firmy často pomohou při vývoji např. integrací určitých částí hardwaru od jiného dodavatele. Společnost Microsoft pro výrobce hardwaru doporučuje čtyři bezpečnostní opatření:

1. Minimalizace hardware dle konkrétních požadavků.
2. Zajištění hardware proti otevření (označení jako *„Tamper resistant“*).
3. Využití podpůrných mechanismů pro zabezpečení hardware.
4. Aktualizace firmware.

První opatření nabádá výrobce, již při návrhu zařízení, ke snaze minimalizovat funkce hardwaru potřebné k provozu a zbytečně nenabízet rozšíření, která nejsou nezbytná. Jako příklad lze uvést např. zahrnutí portu USB pouze, pokud je nezbytně nutný pro provoz zařízení. Jakékoliv rozšíření zvyšují rizika zneužití zařízení.

Druhé opatření hovoří o zabezpečení hardwaru proti otevření jako článek *„Internet of things security best practices“* a jen to dokazuje, že hardwarová bezpečnost zařízení v rámci Internetu věcí je jeden z hlavních problémů, kterým je třeba čelit.

Třetí opatření je podmíněné firemní politice daného výrobce a hlavně tzv. *„Cost Of Goods Sold“*[30], jelikož jakékoliv bezpečnostní rozšíření zvyšuje cenu zařízení. Rozšířeními mohou být např. šifrování úložiště dat či funkce založené na tzv. *„Trusted Platform Module“*[31].

Poslední opatření v první fázi z *„Security-in-depth strategie“* dovysvětluje problematiku neaktuálního firmware na mnoha zařízeních z Internetu věcí. Hlavní příjmy výrobců zařízení pocházejí z prodeje zařízení a nikoliv z jejich údržby. Tito výrobci navíc nejsou právně zodpovědní za údržbu zařízení po celý jeho životní cyklus. K této situaci nenapomáhá ani konkurenční boj mezi výrobci, kteří se snaží představit svá zařízení jako první, a proto se věnují převážně vývoji.

**Řešení vývoje**
Další fází *„Security-in-depth strategie“* je řešení vývoje, která nabízí doporučení pro zabezpečení ze strany vývojářů softwaru. Tato fáze obsahuje opět několik bezpečnostních opatření:

[30] Přímé náklady související s výrobou zařízení určeného pro prodej.
[31] Specifikace šifrovaného procesoru sloužícího k uložení šifrovacích klíčů, poskytujících rozšíření pro generování pseudonáhodných čísel či vzdálené ověření.

1. Dodržení metodik vývoje softwaru s ohledem na bezpečnost.

2. Opatrnost při volbě softwaru s otevřeným zdrojovým kódem.

3. Důslednost při zajištění integrity.

První opatření nabádá k přemýšlení o bezpečnosti nového softwaru již od počátku projektu přes jeho implementaci, testování a nasazení. Velmi důležitý je výběr používané platformy, vhodných programovacích jazyků a vývojových nástrojů.

V pořadí druhé opatření více rozšiřuje opatření první se zaměřením na open-source software. Tento typ softwaru je často využitý u menších, začínajících firem, které si nemohou dovolit či z nějakého důvodu nechtějí používat placený software. Jedna z hlavních výhod použití open-source programů je možnost poměrně rychlého vývoje. Programátoři by před zvolením měli dobře zvážit např. podporu tohoto typu softwaru, jelikož aktivní komunita zajišťuje podporu softwaru a reaguje na nalezení problémů.

Poslední část se snaží upozornit na časté chyby v integracích mezi různými knihovnami a API[32]. Vývojáři mohou často nechat v implementaci funkci, která není pro současné řešení vyžadována (např. počítají s budoucím rozšířením). Tato funkce může být k dispozici prostřednictvím API rozhraní, což obvykle znamená bezpečnostní riziko. Řešením je důkladná kontrola a testování rozhraní všech součástí systému.

**Řešení nasazení**

Jakmile je dokončena fáze vývoje, začíná pro zařízení fáze nasazení. I této fázi se věnuje *„Security-in-depth strategie“*, která nabízí několik bezpečnostních opatření pro správce zařízení spadajících do Internetu věcí:

1. Bezpečné nasazení hardwaru.

2. Udržení v bezpečí autentizační klíče.

První část nazvaná „Bezpečné nasazení hardwaru“ popisuje různé možnosti umístění zařízení jako např. veřejné prostory, nesledované lokality (bez monitorovacích kamer), soukromé pozemky atd. Je třeba se ujistit, že umístění je oprávněné, nehrozí žádné vnější poškození (třeba zvážit i např. povětrnostní podmínky), a také se ujistit, že všechny konektory, ke kterým by se dalo připojit, jsou důkladně zakryty (např. USB). Tato část obsahuje i doporučení kontroly síťových portů (např. porty 22 či 23 pro SSH a Telnet).

Principy šifrování vyžadují klíče, kterými se šifruje a dešifruje daná informace (např. v internetové komunikaci nebo informace uložena na disku).

[32]Sada funkcí, tříd či protokolů určité knihovny, jiného programu nebo jádra operačního systému, kterou programátor využívá.

Podobné klíče jsou potřeba i k připojení zařízení ke cloudovým službám (společně s ID zařízení). Tyto klíče je třeba bezpečně spravovat a zamezit jejich odcizení.

**Řešení řízení**
Poslední fáze *„Security-in-depth strategie“* souvisí s řízením a monitorování stavu zařízení. Odpovědná osoba (správce) by měla mít aktuální přehled o stavu všech zařízení pro případnou reakci na možný problém a rychlé řešení tohoto problému.

1. Aktualizovaný systém.
2. Rozšiřující bezpečnostní opatření.
3. Pravidelný audit.

Spousta zařízení spadajících do Internetu věcí obsahují vlastní operační systém (např. Raspbian pro zařízení Raspberry Pi). Na takových zařízení je třeba počítat s aktualizací celého operačního systému či konkrétních ovladačů připojených periferií. Nové aktualizace často reagují na bezpečnostní problémy. Druhé doporučení souvisí s použitím rozšiřujících bezpečnostní opatření pro zařízení s operačním systémem jako je antivirus či antimalware a např. aktivace firewallu. Tyto podpůrné programy napomáhají zmenšit množinu vnějších hrozeb. Zejména v této části ochrany však platí pravidlo: „Útočník je vždy o krok napřed.“, jelikož obrana většinou pouze reaguje na zjištěné problémy.

Třetí, a velmi důležité doporučení, se týká pravidelného auditu zařízení. O zařízeních je třeba mít přehled. Je tedy vhodné sbírat a pravidelně analyzovat telemetrická data. Ta mohou často poskytnout informace o bezpečnostním incidentu i zpětně. Je třeba řešit jejich pravidelné zálohování. Je také doporučeno nasazení bezpečnostního mechanismu, který bude aktivně kontrolovat stav zařízení a v případě výskytu problému ihned informuje správce zařízení.

Pro zvýšení bezpečnosti zařízení je třeba zvážit, jaké informace o zařízení je třeba shromažďovat. Činnost logování může mít velký vliv na spotřebu a výkon zařízení. Je nutné zvolit optimální úroveň, která poslouží jako dostatečně obsáhlý informační kanál a zároveň tato činnost výrazným způsobem neovlivní chod zařízení. Jeden z hlavních problémů infrastruktury Internetu věcí je útok prostřednictvím fyzického přístupu. Je vhodné zvážit, zda např. nezaznamenávat GPS polohu zařízení, hodnoty z akcelerometru (náklon zařízení) či použití vstupních portů.

Zařízení spadající do Internetu věcí se mohou od sebe velmi lišit. Některá složitější mohou obsahovat operační systém. Jiná pouze jednoduchý firmware. Výše popsané principy a doporučení lze použít vždy v určité míře. Pokud jsou k dispozici další, není vhodné se omezovat pouze na tato opatření.

### 2.3.5 Ochrana proti fyzické manipulaci

Firma Neuron soundware pracovala řadu let na vývoji vlastního software. Vynaložila nemalé prostředky na vývoj a testování a považuje za rozhodující chránit firemní „know how“ před odcizením. Toto je to nejdůležitější, na čem firma staví svoji prosperitu. Zabezpečení proti odcizení bylo jedním z hlavních požadavků na konstrukci zvukového senzoru.

V minulosti byl pojem fyzická bezpečnost používán k označení ochrany majetku proti poškození např. ohněm, vodou nebo ochrany proti krádeži. Oblast počítačové bezpečnosti doplnila fyzickou bezpečnost o význam: *„Technologie používané k ochraně informací před fyzickým útokem“*. V tomto novém smyslu je fyzická bezpečnost jakousi bariérou umístěnou kolem výpočetní techniky pro zabránění neoprávněnému přístupu k samotnému zařízení. Pro splnění fyzické bezpečnosti je nutno dodržet několik kritérií:

- V případě útoku by měla být malá pravděpodobnost úspěchu a vysoká pravděpodobnost detekce, buď během útoku, nebo při kontrole zařízení.
- Pro ochranu citlivých dat je nutné vytvořit fyzické bezpečnostní systémy, které obsahují spouštěcí mechanismy k potlačení útoku podobně jako např. poplašný systém.
- Měly by být navrženy klasifikační systémy, které vyhodnocují výpočetní systémy podle kritérií, která klasifikují obtížnost provedení úspěšného útoku.

Mechanismy, souhrnně označené jako tzv. *„Tamper Mechanisms“* [23], zabraňující neoprávněnému přístupu k zařízení či datům, lze rozdělit do čtyř kategorií:

1. Resistance.
2. Evidence.
3. Detection.
4. Response.

První typ mechanismu nazvaný „Resistance“ spočívá v omezení fyzického přístupu k zařízení. Jedná se o nejjednodušší bezpečnostní opatření. Způsoby těchto opatření je několik. První souhrnný název je označován jako „Bank vault technology“. Popisuje vytvoření zařízení tak velkým a těžkým, že se nevyplatí útočníkům toto zařízení odcizit (jeden z těchto typů je např. bankomat). Vestavěné zařízení může obsahovat sebedestrukční mechanismy, které způsobí trvalé poškození při napadení. Další opatření může být např. „Tvrdá bariéra“. V tomto případě se jedná o vytvoření fyzické bariéry z oceli, kovu či pevných plastů k zabránění fyzického přístupu. Dále lze uvažovat o tzv.

„Kovovém stínění“, což znamená umístění zařízení do kovové klece či krabice pro ochranu před elektromagnetickým zářením z okolí.

Pro Tamper Resistance je možné použít opatření, které souvisí již s návrhem čipu zařízení. Určitým způsobem návrhu čipu je možné zajistit, aby vrstvy zajišťující funkčnost obklopovaly vrstvy s citlivými daty, které je třeba chránit. Tento princip zajistí, že chráněné oblasti nemohou být přečteny bez odstranění nebo poškození funkčních vrstev, které jsou nutné pro získání dat. Druhý typ mechanismu nazvaný „Evidence“ nechrání zařízení před útokem, ale je navržen tak, aby zajistil důkaz o útoku, jakmile k němu dojde. V tomto případě se často používají chemické či mechanické prostředky k prokázání útoků. Jelikož nedochází k oznámení útoku, je zapotřebí provádět pravidelný audit zařízení. K tomuto typu mechanismu se používají opatření jako např. umístění zařízení do křehkého obalu, který je při manipulaci poškozen a nemůže být opraven.

Dalším opatřením zajišťující Tamper Evidence je pošpinění pomocí barvy. Zařízení obsahuje balónky s barvou, které prasknou při manipulaci a útok je možné jednoznačně identifikovat. Další možností je např. holografická páska. Pokud dojde k poškození, opět je snadné identifikovat manipulaci.

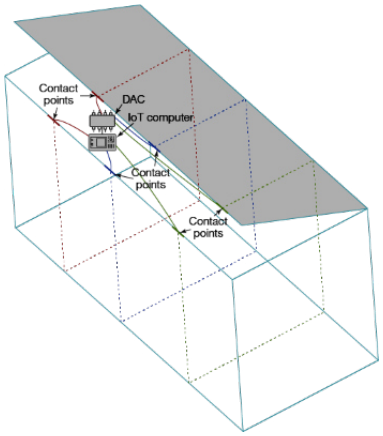
Další dva typy Tamper „Detection“ a Tamper „Response“ jsou často označovány souhrnně, protože Response musí obsahovat mechanismus Detection, což znamená, že pouze pokud je zařízení schopno detekovat útok, je možné vytvořit mechanismus reakce na tento útok. Detekci útoku lze provést instalací čidel a senzorů v okolí zařízení. Typ senzorů souvisí s tím, jaký typ manipulace je třeba detekovat (rozpouzdření, přemístění atd.). Lze dělit mechanismy detekce do kategorií:

- Spínač – zařízení, které detekuje mechanický pohyb.
- Senzor – zařízení, které detekuje změnu prostředí.
- Obvody – vodiče, které jsou omotané kolem zařízení.
- Elektronika – detekce a sledování změn frekvencí, hodinových impulzů nebo napětí vedoucí do nebo z čipu.

Při vyhodnocování možných metod implementace detekce neoprávněného zásahu je třeba vždy vzít při návrhu v úvahu několik aspektů, jako jsou náklady, proces montáže, nedestruktivní nebo destruktivní činnost, pasivní nebo aktivní detekce, mechanické nebo polovodičové technologie.

Nejjednodušší způsob detekce otevření či vniknutí do zařízení může být pás vodivého materiálu, který při zavření vytvoří elektrický obvod. Otevřením se obvod rozepne, což vede k nestandardní události, která může způsobit bezpečnostní opatření. Hlavní nevýhodou je koroze, která může postupem času tento mechanismus zcela poškodit a deaktivovat.

Další úroveň detekce otevření obalu, ve kterém je zařízení umístěno, může fungovat na bázi využití několika GPIO pinů a ADC převodníku. Pomocí spojení několika drátků, které vedou po obvodu krabice vytvoří obvod, ve kterém jsou generována náhodná data. Tento obvod je následně připojen k ADC převodníku a dochází ke kontrole, zda odeslaná a přijatá data se shodují (viz obrázek 2.15).




Obrázek 2.15: Mechanismus otevření obalu zařízení[33].

Další možností detekce otevření je pomocí tzv. magnetického spínače. Tento spínač používá dva magnety. Jeden je umístěn ve víku krabice a druhý v přilehavé stěně. Při otevření dochází k oslabení magnetického pole, ke kterému je připojený spínač, který vyvolá výjimku. Výhodou této konstrukce je možnost použití magnetů společně s nerezovou ocelí.

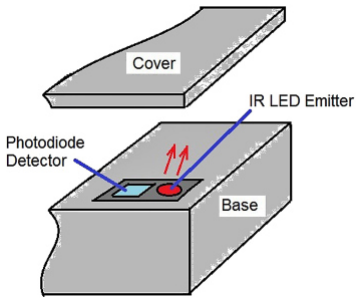
Další způsob detekce otevření může být pomocí Infračervené LED a fotodiody jako detektoru. Jakmile je kryt na svém místě, infračervený paprsek se odráží od víka do fotodiody (viz obrázek 2.16). Pokud fotodioda zjistí, že intenzita světla odražená od LED je nad určitou prahovou hodnotou, znamená to, že kryt je na svém místě. Po přemístění krytu se intenzita světla na fotodiodu dramaticky sníží, což senzoru umožní zaznamenat případnou neoprávněnou událost.

Příkladem tohoto senzoru může být tzv. Infra závora (viz obrázek 2.17). Tento senzor se primárně používá pro detekci překážky např. u pohybujících se robotů. Vzdálenost detekce je možné nastavit pomocí potenciometru. Od výrobce jsou udávány hodnoty 2 až 10 cm v závislosti na barvě a struktuře povrchu. Při otevření se signál přepne do digitální hodnoty 0, což vyvolá nestandardní událost (např. smazání důvěrných dat).

[33]Převzato z: `https://www.ibm.com/developerworks/library/iot-prevent-threats-iot-devices/index.html`

[34]Převzato z: `https://www.sensorsmag.com/components/how-to-implement-reliable-tamper-detection-a-standard-proximity-sensor-module`

[35]Převzato z: `http://www.hwpro.cz`




Obrázek 2.16: Detekce otevření pomocí infračerveného paprsku[34].

Obrázek 2.17: Infra závora[35].

Již zmíněná fotodioda, která je součástí modulu Infra závora, může být použita i samostatně. Tato elektronická součástka reaguje na změny okolního světla. Pokud by byla umístěna do tmavé krabičky, dokázala by detekovat rozpouzdření, jakmile by do krabičky vniklo okolní světlo. Detekci by bylo možno obejít rozpouzdřením za špatných světelných podmínek (což je však technicky náročnější). Tato varianta je však u mnoha elektrických zařízení nevhodná, protože plastové obaly jsou většinou opatřeny otvory, kterými proudí vzduch, zajišťující pasivní či aktivní chlazení zařízení.

Další možností detekce vniknutí je použití ultrazvuku, a to konkrétně detekce ultrazvukového echa. Jakmile se změní tlakové poměry uvnitř krabičky nebo odrazivost a absorpce materiálu, změní se i ultrazvuková ozvěna a senzor ultrazvukového echa umístěný v krabičce tuto změnu zaznamená.

K vytvoření úspěšného mechanismu detekce manipulace může často pomoci pouze pokus o úplnou demontáž cílového produktu. Je nutné počítat s tím, že útočník bude mít k dispozici více než jedno zařízení, což způsobí objevení bezpečnostních mechanismů a následně možnost mechanismy obejít.

Informace o detekci jsou následně využity pro vytvoření rekce na daný útok. Tato rekce může pramenit ze snahy zabránit získání citlivých dat z paměti (často typ RAM či ROM). Pokud jsou data v paměti RAM, je možné

vypnout napájení této paměti, což způsobí vymazání obsahu. Složitější způsob (platí i pro paměť ROM) je přepsání obsahu, protože vyžaduje určitý čas a např. spolehlivý zdroj energie.

**Fyzické útoky a mechanizmy obrany**
Firma Neuron soundware požadovala mechanické a elektronické zabezpečení proti odcizení firmware. Pro správné pochopení možných obraných mechanismů je vhodné seznámit se s útoky, kterým je nutno čelit pomocí „Tamper" mechanismů. Článek *„Physical Security Devices for Computer Subsystems: A Survey of Attacks and Defenses"* [24] popisuje fyzické útoky a mechanizmy obrany, které lze rozdělit od levných a snadno prolomitelných až po extrémně nákladné a obtížnější na prolomení. Jedná se o obsáhlý průvodce mnoha známými typy útoků a poskytuje seznam opatření, které by měli být implementovány pro zvýšení úspěšnosti ochrany. Vyšší bezpečnost je možné dosáhnout i kombinováním jednotlivých mechanismů.

Prvním druhem útoků je útok pomocí tzv. „Sondy". Jedná se o přímé připojení vodičů ke chráněným obvodům ve snaze získat nebo změnit informace. Sondy lze dělit na tzv. pasivní, aktivní a energetické sondy. Rozdíl mezi aktivními a pasivními sondami je schopnost změny dat v zařízení nebo pouze jejich odposlech. Energetické sondy mohou být elektronové paprsky, iontové paprsky nebo paprsky světla. V závislosti na napadené technologii mohou energetické sondy číst nebo zapisovat obsah polovodičového úložiště nebo měnit řídicí signály.

Dalším typem útoku je tzv. „Temperature Imprinting". Jakmile je snížena teplota paměti RAM, udrží svůj obsah bez napájení po dobu několika sekund až hodin. Tento efekt začíná těsně pod bodem mrazu.

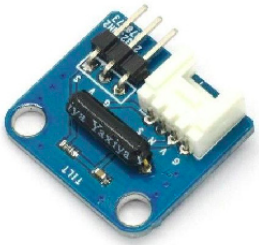
Dalšími druhy útoků jsou tzv. „Metody obrábění", kdy dochází k řezání či odstraňování krycího materiálu ve snaze získat přístup k citlivým obvodům. Jakmile je kryt odstraněn, může dojít k útoku pomocí sondy, jak je popsáno výše. Obrábění bývá provedeno mechanicky (laserem, vodou, ostrým předmětem či chemickou látkou).

I pouhá změna napětí může být použita ve snaze získat citlivé informace, a tedy je možné ji klasifikovat jako druh útoku. Při změně na abnormálně vysoké nebo nízké hodnoty může v mnohých obvodech vyvolat nestandardní chování. Toto chování může zahrnovat pokyny k nesprávné interpretaci procesoru, smazání nebo přepisování obvodů, selhání paměti atd. Pro ochranu před tímto typem útoku je třeba integrovat tzv. „Napěťový snímač". Tyto snímače jsou užitečné takřka u jakéhokoliv zařízení, které vyžaduje určitou energii k napájení.

Dalšími útoky mohou být prodloužení nebo zkrácení hodinového impulzu, poškození obvodů, použití elektronového paprsku pro čtení, či zápis jednotlivých bitů. Dále využití ultrazvuku či rentgenových paprsků pro zobrazení obsahu balení pro následnou identifikaci obsahu.

Existuje mnoho ochranných mechanismů. Jedny z takových mohou být např. snímače pohybu. Tyto snímače se obvykle používají pro snímání pohybu zařízení. Často je třeba tyto senzory používat v párech, protože každý typ může někdy způsobit falešnou detekci. Dalšími mohou být senzory náklonu či senzory teploty.

Jako konkrétní případ senzoru lze uvést senzor pro detekci náklonu zařízení (viz obrázek 2.18). Tento senzor se ustálí na určité hodnotě, což znamená umístění zařízení na dané místo. Jakmile pohybový senzor detekuje výraznější změny, vyvolá událost, která vede k určitému bezpečnostní opatření.



Obrázek 2.18: Senzor náklonu[36].

## 2.4 Logování činnosti

Proces logování (také žurnálování) souvisí s primárním požadavkem zabezpečení firmware zvukového senzoru proti krádeži, protože zvyšuje mírů informovanosti o stavu zařízení. Jde o souhrnné označení pro zaznamenávání informací o činnosti zařízení. Tato činnost probíhá nejčastěji ve formě textových záznamů s předem definovanou syntaxí. Proces je velmi důležitý z ohledem na všechny fáze vývoje, nasazení a údržby nového zařízení.

Tento proces se stal v dnešní době také důležitý díky velkému využití cloudové infrastruktury jako např. Amazon Web Services nebo Microsoft Azure. V těchto infrastrukturách je velmi obtížné dosáhnout izolace výkonu jednotlivých uzlů. Výkon může značně kolísat podle aktuální zátěže. Může docházet k výpadkům uzlů, a proto je kladen důraz na monitorování činnosti.

### 2.4.1 ELK Stack

Zkratka ELK Stack [25] je používána pro spojení tří různých open-source projektů, a to Elasticsearch, Kibana a Logstash (viz níže), vytvářející platformu pro analýzu dat vyvíjenou firmou Elastic.

[36]Převzato z: `http://robotstore.cz`

Elasticsearch je škálovatelný fulltextový vyhledávač, který umožňuje v reálném čase ukládat, vyhledávat a analyzovat velké množství dat. Využívá Representational State Transfer[37] architekturu a poskytuje RESTful API[38]. Existuje celá řada doplňků (viz níže), které rozšiřují základní funkcionalitu tohoto vyhledávače.

Data jsou uložena v tzv. „nerelační databázi“, což způsobí uložení dat do databázové struktury podle formátu ukládaných dat, a proto není potřeba databázovou strukturu předem definovat.

Jeden z hlavních Use cases[39] pro použití Elasticsearche je analýza logů díky své rychlosti vyhledávání v nestrukturovaných a polo strukturovaných databázích. Kibana je vizualizační nástroj vytvářející webové rozhraní pro zobrazení a analýzu dat z Elasticsearche v reálném čase. Umožňuje agregaci dotazů nad databází, filtrování dat, vytváření nejrůznějších grafů a dashboardů. Dále umožňuje generovat přehledy v PDF, za určitý časový interval podle specifického nastavení. Výhodou je rychle nastavení a jednoduchá správa.

Logstash je nástroj pro správu dat z více zdrojů, transformaci a přeposílání dat na specifický výstup. Data jsou často uložena různými programy v různých formátech. Logstash nad těmito daty z různých zdrojů provádí filtrování a transformaci (pomocí mnoho doplňků jako např. Grok). V ELK Stacku právě Logstash takto upravená data odesílá do Elasticsearche, který slouží jako úložiště. Je však možné tento výstup konfigurovat např. na Slack[40] či mailovou schránku.

**Zabezpečení ELK Stacku**

Firma Elastic primárně doporučuje k zabezpečení ELK Stacku vlastní doplněk zvaný X-pack. Toto rozšíření poskytuje v první řadě zabezpečení, dále pak monitorování, alerty[41], reportování a strojové učení v jednom balíčku. Tyto služby je možné podle potřeby libovolně nastavit. Uživatel může také použít pouze některou část jako např. monitorování.

Doplněk X-pack se vyskytuje ve čtyřech různých licencích, a to Basic, Gold, Platinum a Enterprise. Licence se odlišují počtem dostupných služeb. Je nutné upozornit, že základní (Basic) licence je pouze třicetidenní trial verze[42]. Existuje však mnoho alternativ, které nahrazují jednotlivé části X-packu (viz tabulka 2.2).

Pokud bychom se zaměřili pouze na zabezpečení, pak X-pack poskytuje šifrování komunikace, řízení přístupu založeného na rolích, autentizaci pomocí

[37] Architektura rozhraní pro jednoduché vytvoření, čtení, editaci a smazání informací pomocí HTTP volání.

[38] Webové služby odkazující se na implementaci REST architektury.

[39] Případ užití definující interakci mezi člověkem či externím systémem a jiným systémem.

[40] Sada nástrojů a služeb založených v cloudu pro podporu týmové komunikace.

[41] Signál či zpráva, která upozorňuje na útok nebo hrozící nebezpečí.

[42] Časově omezená plně funkční verze.

[43] Převzato z: `https://sematext.com/blog/x-pack-alternatives/`

| | |
|---|---|
| Zabezpečení | SearchGuard |
| Alerty | Elastalert |
| Monitorování | Sematext Cloud |
| Reportování | Skedler |
| Strojové učení | Knowi |

Tabulka 2.2: X-pack alternativy[43].

interních dat, LDAP[44], aktivního seznamu, dále pak záznamy auditů a šifrování v REST. Všechny tyto možnosti nejsou dostupné v základní bezplatné verzi.

**Beats**

Souhrnný název Beats [26] označuje další řadu open-source doplňků od společnosti Elastic. Jedná se o platformu pro jednorázové zasílání různých typů provozních dat. Fungují jako datově, paměťově i výkonnostně nenároční agenti, kteří mohou běžet na mnoha zařízeních a odesílat data do Logstashe nebo přímo do Elasticsearche. Jedná se o centralizující prvek toku dat v infrastruktuře.

Toto souhrnné označení v sobě obsahuje několik instancí, a to Filebeat, Metricbeat, Packetbeat, Winlogbeat, Auditbeat a Heartbeat. Jednoduchý přehled viz níže.

- Filebeat – Předávání a centralizace logů a souborů.
- Metricbeat – Sledování metrických údajů jako např. zatížení CPU.
- Packetbeat – Kontrola a centralizace síťového provozu.
- Winlogbeat – Centralizace logů ve Windows infrastruktuře.
- Auditbeat – Sběr rámcových dat pro audit ze systémů Linux.
- Heartbeat – Informování o dostupnosti internetových služeb.

Všechny části Beats jsou kompatibilní s doplňky Logstash a Elasticsearch, které by běžely v cloudu (např. Amazon Elasticsearch Service). Oficiální doplňky je možné rozšiřovat, a tak vzniklo mnoho dalších doplňků vytvořených vývojářskou komunitou jako např. Connbeat pro exportování metadat z TCP spojení nebo Dockbeat pro získávání a kolekci dat z Docker kontejnerů. Doplňky vytvořené vývojářskou komunitou však nebyly v práci použity.

[44]Protokol pro ukládání a pro přístup k datům na adresářovém serveru.

**Amazon Elasticsearch Service**
Společnosti Amazon rychle reagovala na stále se zvyšující oblibu ELK Stacku a množství svých cloudových služeb rozšířila o Amazon Elasticsearch Service [27]. Hlavním znakem nabízené služby je vynechání nástroje Logstash. Schopnosti Logstashe jsou nahrazeny vestavěnými funkcemi uvnitř nabízené služby. O transformaci a filtrování různých formátů logů se stará přímo Amazon Elasticsearch Service. Tímto krokem se zjednodušila infrastruktura při zachování stejné funkcionality. Liší se pouze přístup k možným rozšířením. Jelikož je Amazon ES řízená služba, nepodporuje přídavné moduly.

Pro vizualizaci dat z databáze Elasticsearch je opět několik možností. Lze použít již zmíněný doplněk Kibana, který bude běžet samostatně na klientském serveru a připojovat se k Amazon Elasticsearch Service. V rámci Amazon Elasticsearch Service je však předem nainstalován nástroj Kibana v základním nastavení dostupný přes Amazon ES domain, a tak celá služba může být v cloudu. Společnost Amazon dále poskytuje službu zvanou Amazon CloudWatch, kterou lze využít pro monitorování Elasticsearche.

Výhodou použití Amazon Elasticsearch Service je dostupnost všech ostatních cloudových služeb, co společnost Amazon nabízí. Pro zabezpečení lze využít např. Amazon Virtual Private Cloud, který vytvoří izolovanou soukromou síť. Šifrování dat v REST pomocí klíčů, které lze generovat pomocí AWS Key Management Service a také řízení přístupu ke službám pomocí AWS Identity and Access Management. Společnost se také stará o pravidelné aktualizace služby Amazon Elasticsearch Service.

Nevýhodou je cena cloudových služeb od společnosti Amazon. Pokud se některá společnost rozhodne využít Amazon Elasticsearch Service, v budoucnu bude obtížné migrovat na jiný cloudový systém jiné společnosti. Pokud však firma přenositelnost neřeší, Amazon Elasticsearch Service může být zajímavou volbou.

### 2.4.2 Alternativy k ELK Stacku

Již zmíněný ELK Stack je velmi oblíbený kvůli své dostupnosti, robustnosti, jednoduchému nastavení a podpoře komunity. Na trhu existují však alternativy, které nabízejí podobné služby nebo mají primárně podobné *Use cases*.

**Splunk**
Společnost Splunk [28], která vyvíjí stejnojmenný software, byla jedna z prvních, která se zaměřila na zpracování a analýzu velkého množství různorodých strojových dat v reálném čase. Tato firma se může pochlubit mnoha významnými referencemi jako jsou Adobe, ING či Tesco. Ke shromažďování dat také používá agenty zvané Splunk Universal Forwarder. Pro vizualizaci používá nástroj Search Head. Pro dotazování používá vlastní jazyk Splunk Porcessing Lanugage. Vyznačuje se velkým množstvím doplňků a hlavní rozdíl oproti ELK Stacku je pořizovací cena, jelikož Splunk není open-source. Cena se odvíjí od

množství indexovaných logů. Je ale plno dalších parametrů, které výslednou cenu ovlivní. Tento nástroj je také mnohem robustnější a hodí se spíše pro větší podnikové infrastruktury.

**Loggly**
Cloudová služba Loggly [29] spadá do třídy *„Software as a service“*[45], nelze tedy provozovat samostatně ve vlastní infrastruktuře. Pro vyzkoušení je možné využít měsíční trial verzi. Použití jako SaaS má za následek jednoduché, ale omezené nastavení. Pokud však nechceme strávit dlouhé hodiny nastavováním prostředí, Loggly může být správná volba. Loggly má však mnohem menší komunitu, než již dva zmínění, což vyplývá z případu užití.

**Graylog**
Open-source software Graylog [30] je velmi podobný ELK Stacku. Skládá se ze samostatné aplikace Graylog, která přijímá data od klientů, webového rozhraní pro vizualizaci (také v rámci Graylogu), databáze MongoDB[46] a Elasticsearche. Oproti ELK Stacku v základě umožňuje například konfiguraci uživatelů pro autentizaci. Na rozdíl od ELK Stacku neexistují žádné placené doplňky. Lze také ELK Stack upravit na ELG Stack nahrazením Kibany webovým prostředím Graylog, které má v základu správu uživatelů. Produkty od společnosti Graylog využívají firmy jako SAP, Lindedin či Cisco, ale prozatím nejsou tak rozšířeny jako ELK Stack.

## 2.5 Vzdálená správa

Vedení firmy Neuron soundware počítá se stovkami až tisíci zařízeními pro nahrávání zvuku a následnou detekci anomálií rozdistribuovanými po celém světě. Takové množství zařízení potřebuje možnost vzdáleného řízení a správy. Hlavní požadavky na software pro vzdálenou správu jsou:

- Jednoduchá instalace a použití.
- Možnost automatizace napříč všemi zařízeními.
- Přehled stavu zařízení.

Odpovědí na tyto požadavky může být několik různých technologií zmíněných níže.

[45] Model nasazení softwaru v cloudu, kde dochází k hostování aplikace.
[46] NoSQL databáze pro ukládání metadat a konfigurací.

### 2.5.1 Ansible

Open-source software Ansible [31] od společnosti Red Hat je orchestrační a konfigurační nástroj pro řízení, automatizaci a správu mnoha uzlů v infrastruktuře prostřednictví SSH nebo PowerShell[47]. Pro běh Ansible je pouze nutnost jazyka Python ve verzi 2.4 a vyšší (pro PowerShell ve verzi 3.0 a vyšší). Pro zápis a znovupoužitelnost příkazů se používá jazyk YAML. Hlavní výhody použití jsou minimalismus, konzistence, bezpečnost (pomocí OpenSSH), spolehlivost a jednoduchost. Ansible je kompatibilní s cloudovými službami jako jsou Microsoft Azure, Google Cloud či Amazon Web services.

Architektura Ansible rozlišuje dva základní uzly, a to řídící server, který řídí orchestraci a ostatní spravované uzly. Nejdůležitější pojmy týkající se Ansible jsou Playbook, Role, Module a Inventory.

Playbook zapsaný ve formátu YAML popisuje orchestraci a konfiguraci jednotlivých uzlů. Obsahuje jednu či více „plays“ a přiřazuje skupině uzlů skupinu Roles a další dodatečné úlohy. Tyto Roles a úkoly lze spouštět synchronně, či asynchronně na množinách či podmnožinách uzlů.

Roles jsou posloupnosti příkazů, které se mají provést na jednom konkrétním uzlu, zapsány pomocí jazyka YAML. Jeden konkrétní příkaz v Role je zvaný Ansible Module. Tyto Roles skládající se z jednotlivých Modules se následně spouštějí v Ansible Playbook na více uzlech současně, což znamená důležitou paralelizaci. Každá Role by měla odpovídat pouze jedné ucelené funkci např. instalaci serveru Apache.

Inventory je seznam uzlů, ke kterým může Ansible přistupovat. Tyto uzly jsou popsány buď IP adresou nebo jménem počítače. Lze vytvářet kolekce sdružených uzlů pro zjednodušení. Inventory lze vytvářet statické (popsané v souboru *../ansible/hosts*) nebo dynamicky generované např. cloudovými službami či libovolným informačním systémem.

### 2.5.2 Resin.io

Další možností vzdálené správy zařízení spadající do Internetu věcí je platforma *„Resin.io“* [32]. Na úvod je třeba zmínit, že se jedná o placený software, ale je možné vyzkoušet prvních 10 zařízení se všemi vlastnostmi placené služby (uživatel se musí registrovat). Hlavními výhodami použití platformy je její bezpečnost, automatizace přes všechna zařízení a velmi jednoduché použití.

Jakmile se uživatel (v tomto případě vývojář či správce) zaregistruje, dostane přístup do webového rozhraní. Tento úkon vyžaduje také nahrání veřejného SSH klíče pro vzdálenou komunikaci se zařízeními. Dále uživatel přidá zařízení (vybere z velkého množství podporovaných hardwarových základních modulů) a následně dojde ke stažení obrazu disku s operačním systémem

[47] Textový shell se skriptovacím jazykem od společnosti Microsoft.

ResinOS, který vychází ze standardního operačního systému zařízení (v případě Raspberry Pi se jedná o Raspbian). Pro nahrání operačního systému do paměti (v případě Raspberry Pi micro SD karta) je doporučeno použít program vyvinutý speciálně pro platformu Resin.io s názvem Etcher.

Po následném zapnutí zařízení trvá několik minut, než se vše nastaví. Následně se zařízení přidá do webového rozhraní, kde je uživatel přehledně informován o jeho stavu. Webové rozhraní umožňuje také připojení do zařízení pomocí SSH a uživatel má také informace o aktuálních stavových výpisech.

K zařízením je možné se připojit vzdáleně z lokálního počítače pomocí SSH připojení díky programu „Resin-cli“. Ten zprostředkovává bezpečné spojení do ResinOS. Pro nahrání aplikace do zařízení je třeba použít Git[48], který distribuuje aplikaci do všech zařízení a není potřeba aplikaci instalovat na všechny instance jednotlivě.

### 2.5.3 Dataplicity

Zajímavou odlehčenou alternativou Resin.io je Dataplicity. Aplikace zprostředkovává vzdálený přístup k zařízením pomocí VPN[49]. Instalace je velmi jednoduchá. Při registraci uživatele dojde k vygenerování specifického ID. Toto ID je následně použito při stažení potřebných souborů na zařízení. Po instalaci se zařízení automaticky přihlásí do webovém rozhraní Dataplicity.

Webové rozhraní umožňuje přístup k zařízení prostřednictvím SSH. Uživatel může odkudkoliv konfigurovat svá zařízení. Má také jednoduchý přehled, která zařízení jsou online, a která ne.

Velkou výhodou oproti Resin.io je možnost automatizace registrace zařízení. Operační systém zařízení lze doplnit o skript, který je spuštěn pro prvním startu a postará se o registraci zařízení do webového rozhraní Dataplicity. Podmínkou je připojení zařízení k internetu. Aby se zamezilo konfliktu názvů, zařízení jsou pojmenována pomocí jednoznačného sériového čísla. Není tedy nutné každé zařízení registrovat do webového rozhraní ručně, jako je to u Resin.io.

V poslední řadě je třeba zmínit, že Dataplicity je placený software. Pouze první zařízení je zdarma. Dále má uživatel na výběr mezi variantou „Standard“ nebo „Pro“. V první variantě je cena 3 $ za zařízení na měsíc a ve variantě „Pro“ je cena 4 $.

## 2.6 Návrh sestavení zvukového senzoru

Zvukový senzor, aby splňoval veškeré primární požadavky (viz sekce 2.1, Požadavky na systém) ze strany firmy Neuron soundware, byl navržen z několika

[48] Distribuovaný systém správy verzí vyvinutý primárně pro vývoj linuxového jádra.
[49] Propojení zařízení v rámci jediné uzavřené virtuální privátní sítě.

dílčích komponent (popsaných níže), které byly vybrány na základě existujících a dostupných řešení. Důraz byl kladen především na funkcionalitu, cenu a rozměry komponent. Výběr ovlivnily i další parametry jako kvalitní dokumentace, podpora komunity a také např. počet dostupných rozšíření.

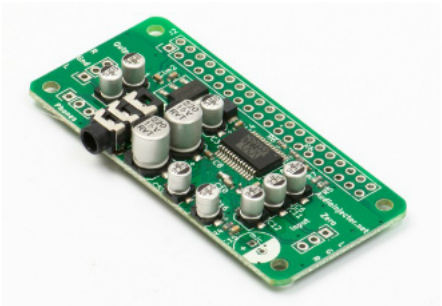
### 2.6.1 Raspberry Pi Zero W

Z analýzy základních modulů vyplynulo, že nejvhodnější deska pro vytvoření zvukového senzoru je deska Raspberry Pi Zero W. Tato deska má ze všech konkurentů nejnižší cenu. Také výkonnostně nezaostává za konkurencí. Jedná se spíše o rozměrově menší desku. Velká výhoda je kompatibilita s větším modelem od stejnojmenné společnosti, která se pyšní největší podporou ze strany komunity. Na tuto desku vzniklo již plno rozšiřujících modulů, čímž lze rozšířit základní funkcionalitu.

Na trnu existuje ještě levnější verze bez označení W. Cenový rozdíl je 156,00 Kč. Levnější varianta neposkytuje v základu podpory Wi-Fi a Bluetooth. Pro zvukový senzor však byla zvolena dražší verze, jelikož podpora Wi-Fi by mohla být do budoucna použita místo mobilního internetu. Tímto by výrazně zlevnila cenu zvukového senzoru.

### 2.6.2 Audio Injector Zero

Od volby základního modulu se odvíjely volby dalších komponent. Pro nahrávání zvuku byla zvolena zvuková karta s názvem *„Audio Injector Zero“* (viz obrázek 2.19). Zvuková karta je poměrně nová a v době analýzy ještě nebyla dostupná. Jedná se o projekt organizace Flatmax Studios, která na vývoj této desky vybrala peníze prostřednictvím serveru Kickstarter.com. V České republice se začala oficiálně prodávat v polovině února roku 2018 (prodává se jako menší stavebnice a vyžaduje pájení) a její cena je aktuálně 549,00 Kč (e-shop Rpishop.cz).



Obrázek 2.19: Audio Injector Zero[50].

[50] Převzato z: `http://rpishop.cz`

Svým tvarem je přizpůsobená velikosti Raspberry Pi Zero. Nabízí vstup i výstup pomocí JST kontaktu, obsahuje 3,5 mm jack, místo pro osazení mikrofonu a dále na pomocných deskách nabízí dvojitý RCA konektor (vstup a výstup) a ovládání hlasitosti. Výrobce deklaruje vzorkovací frekvenci 96 kHz.

### 2.6.3 Huawei E3372h

Pro připojení k mobilnímu internetu byl vybrán modem Huawei E3372h, který se připojí k Raspberry Zero pomocí USB konektoru. Detailněji popsaný byl již v podsekci 2.2.3, Připojení k internetu. Výrobce deklaruje kompatibilitu s operačním systémem Linux. Je výrazně levnější než SparqEE GSM Cellular Board. I přes to s cenou 1347,00 Kč se jedná o nejdražší položku celé sestavy.

### 2.6.4 PiZ-UpTime

Pro zvukový senzor byl vybrán způsob napájení pomocí UPS modulu PiZ-UpTime. Senzor bude primárně napájen z elektrické sítě prostřednictvím zdroje (předem domluveno s firmou Neuron soundware). UPS modul bude však plnit záložní funkci při výpadku proudu a bude mít také bezpečnostní funkci. Bylo obtížné vybrat mezi modulem PiZ-UpTime a modulem PowerBoost 500 Charge. Byla vybrána první varianta z důvodu použití typu baterií AAA. Výdrž zařízení na jednu AAA baterii bude testována. Pokud se zjistí, že výdrž není dostačující, bude alternativou modul PowerBoost 500 Charge, ke kterému je možné připojit baterii s větší kapacitou.

### 2.6.5 Doplňující komponenty

Aby byla zajištěna potřebná funkcionalita, je třeba sestavení doplnit o micro SD kartu s operačním systémem, zdroj napájení a baterii AAA pro zajištění běhu po přerušení napájení ze zdroje, SIM kartu pro připojení k mobilnímu internetu, externí mikrofon, který je připojen pomocí RCA konektoru a adaptér micro-B USB na USB A pro připojení modemu k Raspberry Zero.

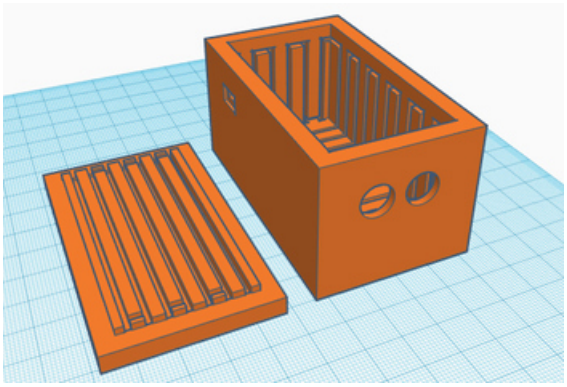
### 2.6.6 Mechanismus detekce manipulace

Návrh mechanismu detekce manipulace vychází z principů popsaných v podsekci 2.3.5, Ochrana proti fyzické manipulaci. Pro dosažení kvalitnějšího zabezpečení bylo zkombinováno více principů vhodných pro zvukový senzor. Mechanismus detekce se skládá ze tří elementárních principů pro zabezpečení zvukového senzoru:

- Detekce poškození krabičky, ve které je umístěn zvukový senzor.
- Detekce otevření krabičky.

- Detekce odpojení zvukového senzoru od zdroje napájení a zajištění běhu zařízení po odpojení.

Pro zjištění pokusu o násilné vniknutí do krabičky se zvukovým senzorem byl navržen způsob detekce pomocí natažení elektrického vodiče uvnitř stěn krabičky a vytvoření elektrického obvodu připojeného na zvukový senzor. Jakmile se útočník pokusí např. vyvrtat otvor do krabičky, přeruší elektrický obvod. Tato změna bude detekována a dojde k patřičným softwarovým reakcím (viz níže). Podobný typ krabičky není zcela standardní, a proto bylo rozhodnuto vyrobit tento prvek zabezpečení pomocí technologie 3D tisku. Byl vytvořen prvotní návrh krabičky v programu Tinkercad 3D (viz obrázek 2.20). Jedná se pouze o prvotní návrh a rozměry krabičky byly odvozeny pouze od rozměrů základního modulu Raspberry Zero.



Obrázek 2.20: Návrh krabičky pro zvukový senzor.

Návrh krabičky počítá s možností otevření např. servisním technikem pomocí použití konektorů (více v sekci 3.1, Sestavení hardwaru). Při odejmutí víka dojde k rozpojení konektorů, a nedochází tedy k trvalému poškození zařízení. Před otevřením je však potřeba softwarově vypnout detekci manipulace. Tímto se liší úkon servisního technika od neoprávněné manipulace.

Ke zvýšení úrovně zabezpečení krabičky byl následně vybrán princip detekce vniknutí okolního světla do krabičky pomocí fotodiody. Pro snadnější připojení ke zvukovému senzoru byl vybrán modul světelného čidla s fotodiodou. Návrh krabičky byl uzpůsoben požadavkům fotodiody, a je tedy navržena tak, aby do krabičky nepronikalo okolní světlo.

Pro princip detekce odpojení zvukového senzoru od zdroje napájení a zajištění běhu zařízení po odpojení byl vybrán modul PiZ-UpTime. Je třeba zdůraznit, že baterie slouží primárně jako bezpečnostní prvek a neslouží k dlouhodobému napájení zařízení. Souhrn všech částí zvukového senzoru s jejich cenou viz příloha D, Hardwarové části zvukového senzoru s cenou.

### 2.6.7 Mechanismus reakce na detekci manipulace

Jakmile dojde k detekci manipulace se zvukovým senzorem, je v první řadě nutné o této události informovat. Pro tuto činnost lze využít např. ELK Stack a doplňky z řady Beats (více viz sekce 2.4, Logování činnosti). Toto řešení nabídne výpis všech zaznamenaných nestandardních událostí (např. otevření krabičky zařízení). V grafickém rozhraní Kibana uživatel jednoduše vyfiltruje informace o konkrétním zařízení (např. typ události, čas, jméno zařízení).

Zvukový senzor by však neměl pouze informovat o nestandardní události, měl by i poskytovat určitou reakci na tuto událost. V podsekci 2.3.5, Ochrana proti fyzické manipulaci, je tento typ *„Tamper“* mechanismu označen jako tzv. *„Responce“*. Existuje několik základních principů, které lze aplikovat při reakci na detekci manipulace:

- Smazání citlivých informací.
- Smazání a přepsání paměti.
- Šifrování obsahu.
- Metody zvané *„obfuscating“*.

Pro doplnění k šifrování obsahu, pokud uživatel nemá potřebnou informaci (tzv. *„klíč“*) pro rozšifrování, nedokáže správně informace přečíst a jsou pro něj bezcenné. Existuje mnoho způsobů šifrování. Je možné např. šifrovat celé paměťové médium nebo pouze jeho část.

Operační systém Raspbian, po jeho instalaci na SD kartu, vytvoří dva diskové oddíly, a to tzv. *„boot“* a *„rootfs“*. První zmíněný slouží pro tzv. *„zavedení“* operačního systému. Tento oddíl není možné šifrovat, je možné jej však využít pro uložení *„klíče“*, který slouží pro dešifrování druhého oddílu, který by naopak šifrovaný byl.

Šifrování celé diskové jednotky a její následné dešifrování způsobí zpomalení celého zařízení. Pokud by zpomalení bylo tak výrazné, že by neumožňovalo standardní činnost zvukového senzoru, bylo by možné šifrovat pouze část jednotky *rootfs*. Při detekci manipulace by došlo k vymazání klíče potřebného k dešifrování. Při opětovném nahrání klíče by se zařízení vrátilo do původního stavu před detekcí.

Pokud bychom se zaměřili na ochranu softwaru před odcizením, lze při reakci na neoprávněnou detekci manipulace použít metody souhrnně popsány jako tzv. *„obfuscating“*, které doplní kód o znaky, díky kterým se stane hůře čitelný. Tyto metody dokážou pozměnit logiku programu (např. přidáním cyklů), aby byl analyzátor kódu ještě více zmaten. Záleží na použitém programovacím jazyku (metody např. nejsou vhodné pro jazyk Python, protože samotný jazyk je specifický svou striktní kontrolou zápisu a dalšími vlastnostmi jazyka, které jsou v přímém rozporu s metodami *„obfuscating“*).

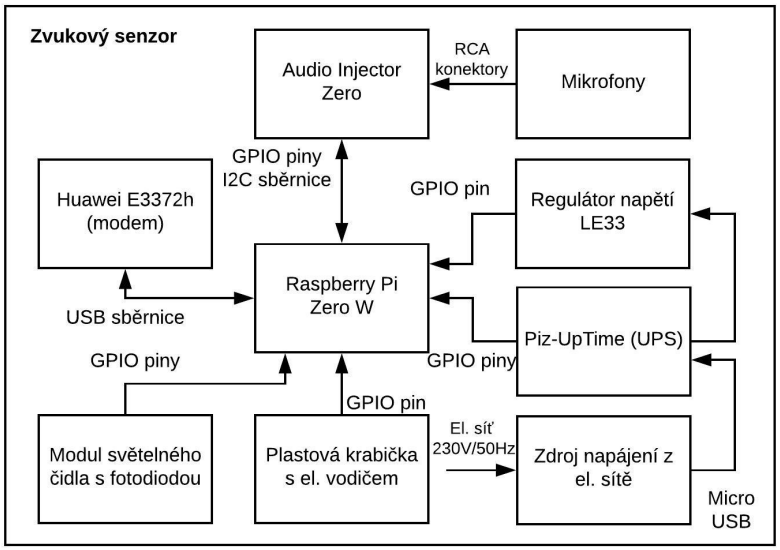
Kapitola 3

# Realizace

Kapitola pojednává o realizaci návrhu zvukového senzoru popsaného v kapitole 2, Analýza a návrh. Popisuje hardwarové sestavení jednotlivých komponent, software, který byl vytvořen pro zvukový senzor a zabezpečení zvukového senzoru realizováno pomocí kombinace několika mechanismů detekce neoprávněné manipulace a následné reakce na detekci manipulace se zařízením.

## 3.1 Sestavení hardwaru

Sekce popisuje sestavení hardwaru zvukového senzoru z dílčích komponent popsaných v oddílu 2.6, Návrh sestavení zvukového senzoru. Celkové sestavení viz obrázek 3.1.




Obrázek 3.1: Blokové schéma zvukového senzoru.

### 3.1.1 Připojení modulu Audio Injector Zero

Modul Audio Injector Zero není plug-and-play. Jeho zprovoznění však nebyl problém. Modul se připojí prostřednictvím GPIO pinů. Jelikož je modul relativně nový, výrobce prozatím nemá řádný oficiální popis postupu zprovoznění modulu. Jediná podpora je přes oficiální fórum[51] výrobce, kde lze nalézt několik tematických vláken zabývajících se prvním připojením. Výrobce také ne zcela přehledně rozlišuje všechny své produkty. Bylo nutné předpokládat, že jeden ovladač je použitelný na všechny zvukové karty výrobce.

Pro správnou funkčnost je třeba stáhnout a nainstalovat do zařízení již zmíněný ovladač (nazvaný *„audio.injector.scripts_0.1-1_all.deb.tar.gz“*). Po instalaci ovladače je nutné dále spustit konfigurační skript *„audioInjector-setup.sh“*, který je součástí archivu ovladače a po instalaci se nachází ve spustitelných skriptech. Dále je třeba zařízení restartovat. Pro ověření správného připojení lze nahlédnout do boot záznamu pomocí příkazu „dmesg“, kde by se měl objevit záznam: *„audioinjector-stereo soc:sound: wm8731-hifi <-> 20203000.i2s mapping ok“*.

Deska obsahuje dva možné vstupy/výstupy. První vstup je pomocí vestavěného mikrofonu a výstup pomocí 3,5 mm konektoru typu „jack“. Druhý pár vstupu a výstupu jsou RCA konektory na přídavných deskách. K přepínání mezi těmito dvojicemi lze použít výrobcem připravené soubory a příkaz:

```
$ alsactl --file /usr/share/doc/audioInjector/asound.
state.<MIC/RCA>.thru.test restore
```

Pro otestování správného zapojení a nastavení modulu výrobce poskytuje testovací skript s názvem *„audioInjector-test.sh“*, který je součástí balíčku s ovladačem. Tento skript přehrává pulzující tón 10 kHz po dobu pěti sekund, a také jej nahrává. Následně je generován spektrogram ve formě obrázku. V příloze C, Použité GPIO piny, je přehled využitých GPIO pinů desky Audio Injector Zero.

### 3.1.2 Připojení modulu PiZ-UpTime

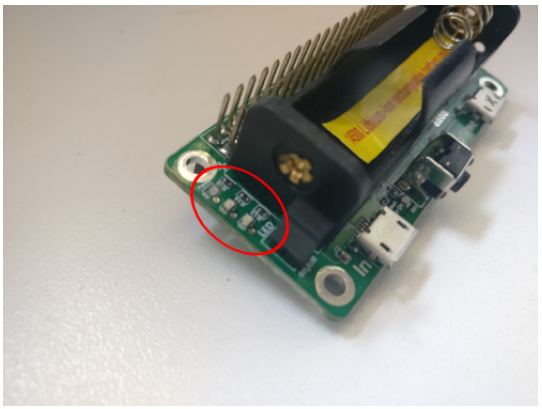
Modul PiZ-UpTime lze připojit k základnímu modulu Raspberry Zero dvěma způsoby.

1. Pomocí GPIO pinů.
2. Pomocí USB kabelu.

Obě tyto varianty byly testovány. Při zvolení připojení přes GPIO piny uživatel dostane možnost softwarově sledovat stav baterie na pinu 26. Tato funkce byla vytvořena výrobcem pro včasné vypnutí zařízení, pokud se baterie vybije pod určitou mezní hodnotu.

[51] Více na `http://forum.audioinjector.net/`

K modulu PiZ-UpTime neexistuje žádná oficiální dokumentace. Na webových stránkách[52] výrobce lze nalézt odpovědi na několik základních otázek týkajících se modulu. Modul např. obsahuje tři stavové diody (viz obrázek 3.2), které signalizují stav baterie (viz tabulka 3.1). Je zajímavé zmínit, že lze připojit k modulu Raspberry Zero více modulů PiZ-UpTime a dosáhnout tak delší doby provozu zařízení na baterii.



Obrázek 3.2: Stavové diody modulu PiZ-UpTime.

| **Stav** | **Připojen zdroj** | **Modrá** | **Žlutá** | **Zelená** |
|---|---|---|---|---|
| Vypnuto | Ne | OFF | OFF | OFF |
| Vybitá baterie | Ne | ON | OFF | OFF |
| Bez baterie, Standby mode nebo baterie špatně vložená | Ano | OFF | OFF | ON |
| Nabíjení | Ano | ON | OFF | ON |
| Nabito | Ano | OFF | ON | ON |
| Chyba | Ano | ON | ON | ON |

Tabulka 3.1: Stavové led modulu PiZ-UpTime.

Během komunikace s oficiálním výrobcem bylo potvrzeno, že modul nenabízí možnost detekovat odpojení napájení ze zdroje a přechod na napájení z baterie. Na modulu však nalezneme jumper[53], který slouží pro vypnutí stavových LED popsaných výše. Od výrobce bylo doporučeno sledování napětí na pinech jumperu, na kterých je přivedeno napětí 5 V a při odpojení zdroje nejsou stavové LED napájeny z baterie (z důvodu snížení spotřeby modulu).

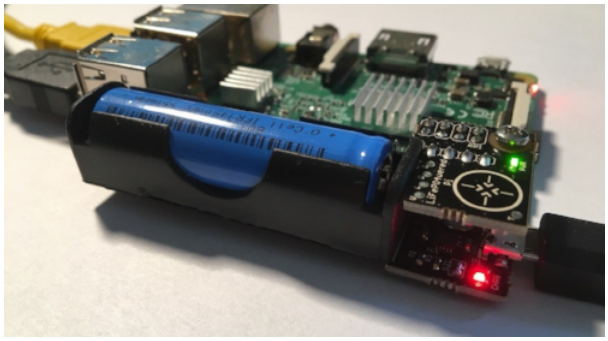
[52] Více na `http://alchemy-power.com/piz-uptime/`

[53] Mechanická spojka vodičů, se kterou se lze setkat např. v elektronice na tištěných spojích.

Způsob získání informace o odpojení zdroje napájení viz níže. Ve snaze minimalizovat světlo uvnitř zavřené krabičky byly stavové diody odpojeny.

Výrobce desek Raspberry Zero nedoporučuje připojovat na GPIO piny vyšší napětí, než je 3,3 V. Vyšší napětí by mohlo způsobit poškození desky. Při detekci změny napájení ze zdroje na baterii pomocí modulu PiZ-UpTime a pinu pro připojení stavových LED bylo potřeba regulovat napětí z původních 5 V z micro USB zdroje na 3,3 V, které je možné připojit na GPIO piny desky Raspberry Zero. Pro tento účel byl využit regulátor napětí LE33 (viz obrázek 3.1). Výsledné napětí je přivedeno na konkrétní GPIO pin 32, na kterém byl nastaven softwarově pull-down rezistor[54]. Detekce změny napájení ze zdroje na baterii je okamžitá. Záleží však na nastavení, jak často je potřeba kontrolovat zdroj napájení.

Nevýhodou tohoto řešení je nutnost dodatečné úpravy hardwaru všech modulů PiZ-UpTime. Výrobce slíbil, že potřebnou funkcionalitu přidá do další generace těchto UPS modulů. Úprava modulů není nijak složitá, ale pokud by to z hlediska masivní produkce zařízení bylo problém, existuje možnost využít konkurenční modul s názvem LiFePO4wered/Pi3 (viz obrázek 3.3), který dle výrobce obsahuje funkcionalitu detekce změny napájení ze zdroje na baterii. Modul nebyl testován.



Obrázek 3.3: Modul LiFePO4wered/Pi3[55].

### 3.1.3 Připojení modulu světelného čidla s fotodiodou

Dále bylo zapotřebí připojit modul světelného čidla s fotodiodou sloužící jako čidlo při otevření obalu zařízení. Tento modul vyžaduje napětí 3,3 V až 5 V. Obsahuje jeden digitální a jeden analogový výstup (v tomto případě využitý pouze digitální výstup). Modul byl připojen na napětí 3,3 V na GPIO pinech

[54]Prvek v logickém obvodu, díky kterému lze definovat logickou úroveň, konkrétně pull-up rezistor způsobí na vstupu logickou úroveň jedna a po připojení vstupu na nižší potenciál je na vstupu logická úroveň nula.

[55]Převzato z: `https://diyprojects.io/test-ups-lifepo4wered-raspberry-pi/`

desky Raspberry Zero a digitální výstup připojen na odpovídající pin desky, na kterém byl nastaven softwarově pull-up rezistor.

Modul fotodioda obsahuje dvě LED, které slouží k informování uživatele o stavu (napájení a detekce světla). Tyto LED, při vložení do krabičky zvukového senzoru, kde se předpokládá nízká úroveň světla, by mohly způsobit rušení při detekci. Od výrobce bylo potvrzeno, že tyto LED nejdou nijak vypnout a bylo doporučeno diody odpájet, což by nemělo mít vliv na funkčnost. LED byly pouze mechanicky ztmaveny, což je výrazně jednodušší.

### 3.1.4 Plastový obal s elektrickým vodičem ve stěnách

Výrazným prvkem hardwarového sestavení je plastový obal (také krabička), která byla navržena přímo pro potřeby zvukového senzoru (viz obrázek 3.4). Krabička vychází z původního návrhu (viz obrázek 2.20). Původní návrh byl upraven na konkrétní rozměry 80 x 140 x 60 mm. Tyto rozměry jsou zejména způsobeny LTE modemem do USB, který slouží pro bezdrátové připojení k internetu. Nabízela se varianta s vyvedeným USB konektorem vně krabičky a připojování modemu externě, což by výrazně zmenšilo velikost krabičky. Tato varianta však odporuje bezpečnostním doporučení popsaným v podsekci 2.3.4, Security-in-depth strategie. Konektor USB by snižoval úroveň zabezpečení a zvyšoval bezpečnostní riziko.

Další možností bylo vytvořit druhou variantu krabičky, která by byla použita pro případ užití zvukového senzoru bez LTE modemu. Může se vyskytnout možnost připojit senzor na Wi-Fi v místě použití. Firma Neuron soundware se takovému případu užití nebrání a již v minulosti se takový případ vyskytl. Tato možnost by způsobila výrazné zmenšení krabičky zvukového senzoru.

Krabička byla navržena pro funkci zapouzdření celkového sestavení (viz příloha E, Nákres plastové krabičky). Má však sloužit také jako bezpečnostní prvek. Návrh zvukového senzoru a jeho následná realizace byla vytvořena hlavně s ohledem na celkovou cenu zařízení. Z tohoto důvodu byl zvolen plastový obal zvukového senzoru místo kovového. Vždy je třeba zvolit bezpečnostní kompromis mezi úrovní zabezpečení a úsilím útočníka bezpečnostní opatření prolomit. Například není nutné zařízení vložit do kovového trezoru, pokud se na něm nenachází data, která by takovou ochranu vyžadovala či která by nebyla pro útočníka tak zajímavá, aby se rozhodl prolomit zabezpečení trezoru.

Plastový obal zvukového senzoru je specifický svými vnitřními drážkami. Jsou určeny pro zabudování elektrického vodiče tvořícího elektrický obvod. Při přerušení tohoto obvodu dojde k detekci manipulace a následné softwarové reakci. Krabička byla navržena s mřížkou drážek. V každé drážce mřížky lze vést vodič. Pro toto řešení je potřeba takřka 2,5 m vodiče. Pokud by v budoucnu způsobil takto dlouhý vodič problémy či toto řešení by bylo příliš nákladné, je možné natáhnout kratší vodič pouze do některých drážek (tímto je možné při sestavení zařízení zavést určitou náhodnost). Jelikož byl princip elektric-

kého vodiče zkombinován s modulem světelného čidla s fotodiodou pro detekci okolního světla, úroveň zabezpečení se zkrácením vodiče příliš nesníží.


Obrázek 3.4: Plastový obal zvukového senzoru.

Při návrhu plastové krabičky (drážkám pro vinutí vodiče) bylo zapotřebí vyvarovat se natažení vodiče v kruzích. Pokud by byl vodič kolem krabičky obtočen, vznikla by cívka, což by vyvolalo nežádoucí elektromagnetickou indukci. Tento jev by mohl výrazně ovlivnit zvukový senzor. Pro elegantnější uchopení vodiče ke stěnám krabičky byla dále navržena a vytištěna plastová „vložka“ (více příloha E, Nákres plastové krabičky). Tato vložka je tvořena čtyřmi stěnami (bez spodní a vrchní stěny), které přesně padnou do plastové krabičky na obrázku 3.4. Boční stěny krabičky jsou tedy dvojité, díky čemuž vodič ve stěnách krabičky snadno drží.

V jednom místě je možné elektrický vodič rozpojit (viz obrázek 3.5), což znamená, že lze nedestruktivně otevřít plastovou krabičku (např. pro potřebu výměny baterie zařízení). Před otevřením je však potřeba softwarově vypnout detekci manipulace. Fotografie celkového sestavení zvukového senzoru viz příloha A, Fotografie zvukového senzoru.

## 3.2 Nastavení OS před prvním startem

Modul Raspberry Zero potřebuje ke svému běhu micro SD kartu s operačním systémem. Pro potřeby diplomové práce byl nainstalován nejaktuálnější operační systém Raspbian Stretch Lite[56] duben 2018 s verzí jádra 4.14. Během vytváření diplomové práce byly testovány i starší verze operačního systému.

[56]Verze operačního systému na bázi Debian Stretch vyznačující se menší velikostí, jelikož obsahuje pouze nejzákladnější prvky potřebné pro běh.


Obrázek 3.5: Detail konektoru pro možnost otevření plastové krabičky.

U starších verzí se však objevoval problém s připojováním zvukové karty Audio Injector Zero.

Karta při instalaci, nebo také později při vývoji, nebyla detekována operačním systémem. Při důkladné studii internetového fóra[57], které slouží jako hlavní podpora pro karty Audio Injector, byl nalezen podobný problém u více uživatelů. Výrobci vydali pouze oficiální popis připojení a instalace karet, který však daný problém zcela nevyřešil. U nejnovější verze operačního systému se však tento problém během testování nevyskytl.

Pro instalaci systému na micro SD kartu lze využít několik nástrojů, podle toho, ze kterého operačního systému se nahrává obraz operačního systému na kartu. Pro operační systém Windows byla testována a je doporučena utilita[58] *„Win32DiskImager“*. V rámci systému Linux je možné využít utilitu v příkazovém řádku označenou *„dd“*. Firma Neuron soundware pro tyto potřeby doporučila program vytvořený pro nahrávání operačního systému ResinOS (viz podsekce 2.5.2, Resin.io) s názvem Etcher. Program poskytuje jednoduché uživatelské rozhraní a lze jej využít i pro nahrání systému Raspbian.

Pro základní nastavení operačního systému byl vytvořen skript s názvem *„firstStart.sh“*. Po instalaci systému na micro SD kartu má systém např. vypnutou Wi-Fi, nemá povolený přístup prostřednictvím SSH a mnoho dalších, a proto je třeba systém upravit ještě před prvním startem, aby se zařízení automaticky připojilo k internetové síti a aby jej bylo možné vzdáleně spravovat. Varianta, kdy by servisní technik musel u každého nového zařízení při prvotním startu nastavit vše potřebné ručně nebyla v tomto případě vhodná.

Chování systému Raspbian před prvním startem je možné upravit pomocí změn souborů nahraných během instalace systému na micro SD kartu. Pro

[57] Více na `http://forum.audioinjector.net`

[58] Pomocný program, který slouží k usnadnění činností spojených s používáním počítače.

nastavení připojení k Wi-Fi je pomocí skriptu *„firstStart.sh“* nahrán soubor s názvem *„wpa_supplicant.conf“* na micro SD kartu do *boot* oddílu. Soubor *„wpa_supplicant.conf“* obsahuje základní nastavení Wi-Fi sítě a danou lokalitu. Při prvním spuštění systém detekuje tento soubor a použije jej pro permanentní konfiguraci zařízení. Podobné chování je využito i pro povolení SSH připojení k zařízení. Stačí pouze umístit soubor s názvem *„ssh“* do *boot* oddílu a k zařízení bude možné vzdáleně přistupovat již po prvním startu prostřednictvím programu a zároveň komunikačního protokolu SSH.

## 3.3 Orchestrace zařízení pomocí Ansible

Program Ansible, popsaný v podsekci 2.5.1, Ansible, byl využit pro dodatečné nastavení operačního systému po prvním startu zařízení. V této fázi se objevily dvě možnosti nastavení:

1. Vygenerování vlastního operačního systému Raspbian pomocí speciálních nástrojů.
2. Použití výchozího operačního systému Raspbian a jeho následná úprava dle konkrétních požadavků.

Pro potřebu zvukového senzoru byla zvolena druhá varianta, protože se počítá s různým nastavením zařízení pro konkrétní použití. První varianta by tedy musela být kombinována s druhou (využití Ansible), protože při testování generování nového upraveného Raspbianu bylo zjištěno, že tento proces je výpočetně a časově náročný a při jakékoliv změně by tento proces musel být spouštěn znova. Byla tedy zvolena druhá varianta s programem Ansible a s výchozím operačním systémem.

Pro potřeby zvukového senzoru byly vytvořeny dva Ansible Playbooky s názvem *„default.yml“* a *„nsw.yml“*. Pro oba tyto Playbooky byly vytvořeny Inventory se stejným názvem. Ansible Playbook při připojení pomocí SSH k zařízení využije nastavení definované v jednom Inventory. Nelze vytvořit jeden Playbook, který se snaží změnit nastavení uživatele, pomocí kterého je připojen (uživatelské jméno a heslo) a následně pokračovat v činnosti. Znakem Ansible je také vlastnost, kdy dochází k přepisu proměnných v jednom Inventory definované pro jedno zařízení v různých Groups. Z těchto důvodů bylo potřeba vytvořit dva Inventory, protože jedním z úkonů je změna uživatele.

### 3.3.1 Playbook *„default.yml“*

První Playbook s názvem *„default.yml“* je využit pro základní uživatelské nastavení zařízení. V rámci tohoto Playbooku jsou spuštěny Role:

1. *change-default-pi*
2. *new-user*

Operační systém Raspbian ve výchozím nastavení obsahuje uživatele *„pi“* s heslem *„raspberry“*. První Role slouží ke změně výchozího hesla uživatele pro zvýšení zabezpečení zařízení. Při orchestraci zařízení programem Ansible bylo nutné přidělit uživateli stejná práva jako měl ve výchozím nastavení. Ansible totiž uživatele přepíše uživatelem novým, nastaveným v Role. Zejména je nutné přidělit jej do stejných *groups*. Při realizaci došlo ke špatnému nastavení a upravený uživatel neměl přístup k souborům potřebným pro běh zvukové karty. Nestandardní chování bylo nejdříve připisováno špatné instalaci zvukové karty, což komplikovalo vývoj.

Druhá Role vytváří nového uživatele *„nsw“*, který má stejná práva jako uživatel *„pi“*, je přidělen do stejných *groups* a je mu také vytvořena sekce */home/nsw*. Tento uživatel je následně použit pro další konfiguraci zařízení pomocí druhého Playbooku *„nsw.yml“*.

### 3.3.2 Playbook *„nsw.yml“*

Druhý Playbook byl vytvořený pro doplnění operačního systému Raspbian o potřebné balíčky či skripty pro běh zvukového senzoru. Obsahuje tyto Role:

1. *update*
2. *soundcard*
3. *ssh-keys*
4. *install-packages*
5. *turn-off-led*
6. *set-MIC / set-RCA*
7. *detection*
8. *filebeat*
9. *modem*
10. *crypto*
11. *recording*

První Role s názvem *„update“* provádí aktualizaci seznamu balíčků a následnou aktualizaci aplikací. Při použití staršího operačního systému Raspbian byla tato Role extrémně dlouhá, protože zde došlo k detekci zastaralého jádra operačního systému a následné aktualizaci. Je tedy výrazně rychlejší použití aktuální verze systému.

Druhá Role s názvem *„soundcard“* provádí instalaci ovladačů zvukové karty Audio Injector Zero. Ansible provádí kroky popsané již v sekci 3.1, Sestavení hardwaru. Zajímavostí je provedení restartu zařízení a následné čekání

na opětovné navázání spojení, aby mohly pokračovat další naplánované Role v Playbooku. Třetí Role vytváří SSH klíče použité pro autentizaci při připojení k zařízení.

Role s názvem *„install-packages“* slouží pro instalaci potřebných balíčků, které jsou využity v rámci zvukového senzoru a nejsou součástí výchozího operačního systému. Jedná se např. o balíček *„python-systemd“*, který je využit pro záznam činnosti nahrávacího softwaru do systémové *service „Journald“*[59].

Pátá Role s názvem *„turn-off-led“* slouží k vypnutí stavové LED desky Raspberry Zero. Tato zelená LED byla záměrně vypnuta, aby nedocházelo k ovlivnění fotodiody použité jako bezpečnostní prvek k detekci otevření plastového obalu zařízení.

Další dvě Role *„set-MIC / set-RCA“* navazují na nastavení zvukové karty. Pro nastavení vstupů zvukové karty je použit program *„alsamixer“*. Ten však po instalaci potřebných ovladačů pomocí Role *„soundcard“* není nastaven správně. Nastavení tohoto programu je možné uložit do souboru. Byly tedy vytvořeny dva soubory pro nastavení vstupů (vestavěný mikrofon či RCA konektory) a pomocí Role *„set-MIC / set-RCA“* je *„alsamixer“* přenastaven.

Role *„detection“* slouží k instalaci skriptů vytvořených pro detekci manipulace se zařízením (elektrický vodič, fotodioda, změna zdroje napájení). Tyto skripty jsou podrobněji popsány v sekci 3.5, Software pro detekci manipulace. Byl vytvořen podpůrný skript s názvem *„make_deb.sh“*, který z těchto skriptů vytváří tzv. *„Debianí balíčky“* pro snadnější instalaci pomocí programu Ansible.

Skript *„make_deb.sh“* byl vytvořen pro generování Debianích balíčků *„nsw-power.deb“*, *„nsw-diod.deb“* a *„nsw-box.deb“*, které obsahují konfiguraci softwaru vytvořeného pro detekci manipulace se zařízením. Skript *„make_deb.sh“* nejdříve vytvoří adresářovou strukturu podobnou struktuře desky. Následně je vytvořen soubor *„<servicename>.service“*, který obsahuje nastavení jednotlivých skriptů pouštěných jako tzv. *Systemd servise*.

*Systemd* [33] je démon pro správu systému Linux. Jde o proces spuštěný během zavádění operačního systému, který slouží k obsluze *services*, sleduje jejich stav, umí je restartovat, omezovat potřebné zdroje, logovat chyby atd. *Systemd* zajistí spuštění skriptů při každém startu. Dále poskytuje informace o stavu *services* a uživatel může jednoduše tyto *services* spravovat.

Podpůrný skript *„make_deb.sh“* následně vytvoří soubory, které musí obsahovat každý Debianí balíček. Zajímavý je soubor *„postinst“*, ve kterém se nacházejí příkazy, které se provedou po instalaci daného balíčků. Jsou zde příkazy pro povolení a start dané *service*. Dále je proveden kontrolní součet pomocí ulitily *„md5sum“* a balíček je vytvořen pomocí ulitily *„dpkg-deb“*. Celý podpůrný skript *„make_deb.sh“* je parametrizován pro určení, ze kterého skriptu je třeba vytvořit daný balíček.

[59] Systémová service pro shromažďování a ukládání dat (logů).

Jako poslední, v rámci Role *„detection“*, se kopíruje do zařízení skript s názvem *„detection.sh“*. Tento skript slouží pro zapnutí, či vypnutí všech *„<servicename>.service“*, tedy kompletní zapnutí respektive vypnutí mechanismů detekce manipulace. Správci umožní po instalaci zařízení na dané místo v průmyslové zóně zapnout mechanismus ochrany, aby nedocházelo z falešným poplachům např. během instalace technikem. Pro spuštění je možné využít Dataplicity pro vzdálenou správu. Role *„filebeat“* nastavuje shromažďování stavových informací o zařízení a je více popsána v podsekci 3.6.4, Filebeat.

Orchestrace programem Ansible pokračuje s nastavením modemu Huawei E3372h pomocí Role *„modem“*. Od firmy Neuron soundware byla zapůjčena SIM karta společnosti T-Mobile. Pro správnou konfiguraci a nastavení modemu byl využit nástroj *„wvdial“* a nastavení modemu se nachází v souboru *„/etc/wvdial.conf“*. Pokud není dostupná síť Wi-Fi, zařízení se automaticky připojí pomocí modemu k internetu. Pro kontrolu připojení lze využít stavovou diodu na modemu Huawei.

Role *„crypto“* nastavuje zabezpečení a připojení třetí diskové části se souborovým systémem XFS, která slouží k uložení zvukových nahrávek. Ty jsou dále odesílány do cloudu a průběžně na zařízení mazány. Vytvoření třetího diskového oddílu bylo z důvodu zvýšení zabezpečení zařízení, a také kvůli chybě v utilitě *„arecord“* používané pro záznam zvuku (více v sekci 3.4, Nahrávací software a vzdálená správa).

Pro zabezpečení zvukových nahrávek je třetí oddíl zašifrován skriptem *„firstStart.sh“* (již zmíněný v sekci 3.2, Nastavení operačního systému před prvním startem), který je spuštěn po instalaci operačního systému na micro SD kartu. Skript pomocí nástroje *„dm-crypt“* [34] a nadstavby *„LUKS“* [35] využívá infrastrukturu linuxového jádra zvanou *„Device mapper“*, díky které lze vytvořit virtuální vrstvu blokových zařízení. Dále nástroj využívá tzv. *„Kernel crypto API“*. Uživatel zašifruje diskový oddíl pomocí jedinečné šifry a klíče. Nástroj *„dm-crypt“* vytvoří novou virtuální vrstvu, která přistupuje k zašifrovanému oddílu. Změny virtuální vrstvy se promítnou do zašifrovaného oddílu.

Nejdříve je v rámci Role *„crypto“* na zařízení kopírován klíč, který slouží k dešifrování třetí diskové jednotky. Nadstavba jménem *„LUKS“* využívá tzv. *„dvouúrovňové“* šifrování. Disková jednotka je šifrována pomocí hlavního klíče, který je následně zašifrován uživatelským heslem. Toto šifrování nabízí možnost uživateli udržovat více hesel k dešifrování.

Dále jsou do zařízení kopírovány skripty *„openxfs.sh“* a *„closexfs.sh“*. Jak je již z názvu patrné, tyto skripty slouží k otevření respektive zavření virtuální vrstvy pomocí již zmíněného klíče a následného připojení ke složce *„/mnt/xfsdata“*.

V poslední fázi Role upraví skript *„rc.local“* ve složce *„/etc“*, který je spouštěn při každém startu. Úpravou je docíleno toho, že virtuální vrstva vždy otevře třetí diskovou jednotku a připojí se na zmíněnou složku *„/mnt/xfsdata“*.

Tohoto nastavení je možné docílit i pomocí úpravy souboru *„fstab“* a souboru *„crypttab“*. Oba se nachází ve složce *„/etc“*. Jakmile by se však nepodařilo odemknout daný oddíl (např. z důvodu smazání klíče), zařízení by požadovalo zadání klíče při startu a start zařízení by se zastavil. Dokud by nebyl zadán manuálně klíč, zařízení by nebylo možné vzdáleně spravovat. Toto chování by nebylo vhodné pro případy užití zvukového senzoru.

Poslední Role s názvem *„recording“* nastavuje potřebné skripty pro záznam zvuku z okolí a následného nahrávání do cloudu (skripty je více popsány v sekci 3.4, Nahrávací software a vzdálená správa). Hlavní skript napsaný v jazyce Python s názvem *„record.py“* je spouštěn vždy při startu jako tzv. *service* pomocí *Systemd* (viz výše). Zajímavostí skriptu je kontrola správného dešifrování a připojení třetího diskového oddílu se souborovým systémem XFS pro uložení nahrávek. Pokud se nepodaří dešifrovat a připojit třetí oddíl (např. z důvodu špatného či chybějícího klíče), skript nezačne nahrávat.

## 3.4 Nahrávací software a vzdálená správa

Sekce popisuje vytvořený software pro účely zvukového senzoru k nahrávání zvukových dat a k přenosu dat do cloudu. Dále pojednává o nastavení služby Dataplicity pro vzdálenou správu.

### 3.4.1 Záznam zvuku a přenos dat do cloudu

Pro záznam zvuku pomocí mikrofonů připojených ke zvukové kartě Audio Injector Zero bylo možné využít několik různých utilit, které jsou součástí linuxové distribuce. Nejznámější jsou tzv. *„arecord“* [36] a *„SoX“* [37]. První zmíněná pochází z open-source projektu Advanced Linux Sound Architecture (ALSA). Součástí tohoto projektu je např. utilita *„aplay“* a *„alsamixer“* pro přehrávání záznamů a nastavení audio vstupu a výstupu.

Druhá zmíněná utilita s názvem SoX (Sound eXchange), určena pro nahrávání, editaci a přehrávání zvuku, byla vytvořena prostřednictvím open-source platformy SourceForge. Hlavní vlastností SoX je čtení a zápis zvukových formátů Au, WAV, AIFF, MP3 (pomocí externího enkodéru MP3 LAME), FLAC atd. a úprava záznamů jako např. zřetězení, ořezání či nastavení hlasitosti. Při testování nahrávání pomocí obou utilit byla kvalita zvuku (při nastavení stejných parametrů) poslechově výrazně lepší pomocí utility *arecord*, která je také výrobcem zvukové karty doporučena.

**Skript *„record.py“* pro záznam zvuku**
Pro potřeby diplomové práce byl vytvořen skript *„record.py“* v jazyce Python, který využívá utilitu *arecord* pro záznam okolního zvuku z mikrofonu. Tento skript podporuje dva módy záznamu, a to nepřetržitý záznam zvuku či periodické nahrávání (např. 5 sekund co minutu). Druhý mód byl implemento-

ván na popud firmy Neuron soundware, která pro nahrávání zvuku v určitých případech užití potřebuje nahrávat pouze intervalově.

Dále skript umožňuje nastavit parametry nahrávání jako je formát (jako podporované formáty utility *arecord*), počet kanálů, frekvenci v hertzích, interval v sekundách a délka periody v módu periodického nahrávání v sekundách (výchozí hodnoty jsou S32_LE, 2, 44000, 5 a 15). Byl vybrán a nastaven zvukový formát Waveform Audio Format (zkratka WAV), u kterého nedochází ke ztrátě zvukové informace.

Skript pro každý nový záznam vytváří podproces, který spustí utilitu *arecord* s danými parametry. Rodičovský proces čeká na ukončení spuštěného podprocesu (čas daný parametrem interval). Zvukové záznamy jsou pojmenovány podle času vytvoření, názvu zařízení a náhodného řetězce pro zamezení vytvoření stejného názvu záznamu jiným zařízením. Činnost skriptu je zaznamenávaná pomocí *Systemd*, a to zejména informace o vytvoření nové nahrávky a nahrání zvukové nahrávky do cloudu.

**Diskový oddíl se souborovým systémem XFS**

Při implementaci módu pro nepřetržitý záznam zvuků byl zjištěn problém přetečení vyrovnávací paměti, která je k dispozici utilitou *arecord*. Ta není primárně určena pro tento druh záznamu. Podle oficiálního fóra Raspberry Pi tato chyba souvisí s rychlostí záznamu zvuků na SD kartu. Jednalo se o zásadní chybu, jelikož přetečení paměti způsobí zvukový výpadek a neúplnost dat. Řešení této chyby spočívalo ve vytvoření nové diskové jednotky (rozdělení a zmenšení původní jednotky na micro SD kartě), naformátování na souborový systém XFS [38] a ukládání zvukových záznamů na tuto diskovou jednotku. Při testování jiného souborového systému došlo pouze ke snížení počtu přetečení vyrovnávací paměti, ale chyba se zcela neeliminovala.

Souborový systém XFS s 64-bitovým adresováním byl původně vytvořen firmou Silicon Graphics pro high-end servery, vyznačuje se tedy rychlostí a výkonem (zejména při zápisu díky efektivním I/O operacím). XFS se vyznačuje stabilitou, není tak obvyklý, jako jiné souborové systémy.

Operačního systému Raspbian při instalaci na micro SD kartu obsahuje pouze dvě diskové jednotky, a to tzv. *„boot“* a *„rootfs“*. Při instalaci pomocí programu Etcher na kartu o velikosti 16 GB dosahuje oddíl *boot* 49,4 MB a oddíl *rootfs* velikosti 1862 MB. Po prvním startu dojde k rozšíření oddílu *rootfs* o zbývající volné místo na kartě, bez ohledu na velikost karty.

Pro úpravu stávajících oddílů systému a pro přidání nového oddílu se souborovým systémem XFS, pro potřeby nahrávání, byly do skriptu *„firstStart.sh“*, již zmíněného v sekci 3.2, Nastavení operačního systému před prvním startem, přidány příkazy pro konfiguraci jednotlivých oddílů. Nejdříve bylo nutné zablokovat automatické rozšíření oddílu *rootfs* na zbývající volné místo karty. Stačilo upravit soubor *„cmdline.txt“* na oddílu *boot*.

Pro úpravu diskových jednotek a následné automatizace byla využita linuxová utilita *„parted“* s rozhraním příkazové řádky. Existuje také varianta s GUI nazvaná Gparted (byla testována). Po instalaci systému je velikost oddílu *rootfs* nastavena na 1862 MB a je využita z 60%. Tato velikost by měla být dostatečná pro základní běh zvukového senzoru a nebylo třeba oddíl zvětšovat. Skript *„firstStart.sh“* však umožňuje oddíl zvětšit podle potřeby (limitem je velikost karty).

Skript *„firstStart.sh“* dále slouží k vytvoření nového oddílu se souborovým systémem XFS. Pro v pořadí třetí oddíl použije zbývající volné místo, které nebylo využito oddílem *boot* a *rootfs*. Variabilní velikost micro SD karet je vyřešena explicitním nastavením konce třetího oddílu.

**Upload zvukových dat do cloudu**

Hlavní skript pro záznam zvuku s názvem *„record.py“* slouží také pro upload nahrávek do cloudu. Firma Neuron soundware využívá několik poskytovatelů cloudových služeb. Pro potřeby diplomové práce byl poskytnut přístup do služby Amazon S3. Jedná se o placené datové úložiště, které se vyznačuje škálovatelností, trvanlivostí a rychlostí. Je vhodné pro dlouhodobé uložení dat. Hodí se i pro ukládání zvukových nahrávek pro následnou analýzu.

Aby bylo možné nahrát soubor do S3, je potřeba, aby zařízení obsahovalo tzv. *„Credentials“*. Jedná se o dvojici specifických řetězců s názvem *„Access key“* a *„Secret key“*. Pro úspěšné nahrání souborů je také potřeba specifikovat tzv. *„Region“*, kde se úložiště nachází. Společnost Amazon pro rychlejší odezvu poskytuje úložiště po celém světě. Pokud má některá firma převážnou část klientů v USA, je vhodnější zvolit úložiště tam. Nastavení *„Credentials“* probíhá pomocí Ansible role *„recording“* (viz sekce 3.3, Orchestrace zařízení pomocí Ansible).

Záznam zvukových nahrávek a jejich nahrávání do cloudu probíhá pomocí dvou vláken. První vlákno nahrává záznamy na diskový oddíl se souborovým systémem XFS. Druhé vlákno představuje frontu nahrávek čekajících na přenos do cloudu. Po pořízení zvukové stopy se přidá informace o novém souboru do fronty druhého vlákna. Druhé vlákno při detekci neprázdné fronty nahraje soubor do Amazon S3 a soubor na zařízení smaže, aby nedošlo k zaplnění paměti. Pro komunikaci s Amazon S3 je využita sada vývojových nástrojů s názvem *„Boto3“*.

Pro dokončení nastavení přenosu dat do cloudu bylo ještě zapotřebí specifikovat tzv. *„Bucket“* a *„Key“*. První zmíněný popisuje místo, kam budou data v rámci úložiště umístěny. *„Key“* dále popisuje cestu k danému souboru v *„Bucket“*. Aby byla data v úložišti přehledně umístěna, cesta ke konkrétní zvukové stopě byla nastavena na:

- *„<device_name>/<date>/<name_of_record>.wav“*.

S nahráváním zvukových nahrávek do cloudu souvisí bezpečnostní hrozba odcizení citlivých informací (v tomto případě zvukových nahrávek) během

přenosu ze zařízení do cloudu. Zvukový senzor pro ukládání nahrávek využívá službu Amazon S3. V budoucnu je možné, že firma Neuron soundware zvolí pro ukládání dat jinou cloudovou službu a aby se zjednodušila možná migrace, šifrování nahrávek probíhá v zařízení bez ohledu na použitou cloudovou službu. O způsob šifrování se stará skript *„record.py“*. Tento skript lze spustit s volitelným parametrem *„Crypt“*, který zašifruje zvukové záznamy pomocí *„GNU Privacy Guard“*[60].

### 3.4.2 Nastavení Dataplicity pro vzdálenou správu

Další významnou funkcí zařízení je možnost jeho vzdálené správy. Bash skript *„firstStart.sh“* upravuje oddíl *rootfs* pro automatickou registraci zařízení do Dataplicity (viz podsekce 2.5.3, Dataplicity). Tato platforma pro vzdálený přístup byla použita pro svou jednoduchost a automatickou registraci zařízení, což hlavní konkurenční služba Resin.io nenabízí. Dataplicity je možné zkombinovat i s programem Ansible.

Skript kopíruje soubor *„mass-install-dp“*, který obsahuje údaje k registraci zařízení, do složky *„/etc/network/if-up.d“* oddílu *rootfs*, ve které se nacházejí skripty spuštěné po připojení zařízení do sítě. Bylo pozorováno, že instalace a registrace zařízení do Dataplicity zpomalí první start o takřka čtyři minuty (v závislosti na rychlosti stahování potřebných souborů).

Při nastavování Dataplicity byly zjištěny zásadní chyby v oficiálním návodu k instalaci[61]. Soubor *„mass-install-dp“* ve složce *„/etc/network/if-up.d“* způsoboval chybu při stahování potřebných souborů z internetu. Zařízení se nemohlo připojit na server, jelikož v době startu konfigurace a následného stahování souborů ještě nedošlo k aktualizaci času zařízení. Čas, který byl nastavený špatně neumožnil připojení ke vzdálenému serveru. Bylo tedy nutné doplnit konfigurační soubor *„mass-install-dp“* o vynucení aktualizace času ještě před začátkem konfigurace Dataplicity.

Další změnu bylo třeba provést v *service* s názvem *„networking“*, která má na starost běh skriptů ve složce *„/etc/network/if-up.d“*. Konfigurace Dataplicity zpomalí start natolik, že *Systemd* automaticky vypínal *service „networking“* po uplynutí určité doby. Tuto dobu bylo tedy nutné prodloužit v nastavení samotné *service*, aby se konfigurace Dataplicity stihla dokončit. Tyto úpravy provádí skript *„firstStart.sh“*.

[60] Aplikace pro šifrování a dešifrování dat, které je součástí projektu GNU, spravující nadací Free Software Foundation.

[61] Více na `https://docs.dataplicity.com/docs/install-dataplicity-on-many-devices`

## 3.5 Software pro detekci manipulace a vyvolání následné reakce

Pro detekci manipulace jsou na zařízení instalovány skripty s názvem *„diod.py“*, *„power.py“* a *„box.py“* pomocí Ansible Role *„detection“* viz sekce 3.3, Orchestrace zařízení pomocí Ansible. Tyto skripty jsou instalovány jako Debianí balíčky a jsou nastaveny jako *Systemd service*, aby docházelo k jejich spuštění při každém startu zařízení.

Skripty jsou pojmenovány podle toho, jakou manipulaci se zařízením kontrolují:

- Skript *„diod.py“* – kontroluje otevření krabičky zařízení pomocí výstupů z modulu světelného čidla s fotodiodou.
- Skript *„power.py“* – kontroluje zdroj napájení zařízení (zdroj nebo baterie).
- Skript *„box.py“* – kontroluje uzavřenost krabičky a její stav pomocí elektrického vodiče vinutého po stranách plastového obalu

Všechny tři skripty běží konstantně na pozadí, čtou data z GPIO pinů, na kterých je detekce připojena a jakoukoli změnu zaznamenávají do souboru *„/var/log/<scriptname>.log“*, který je nastaven v agentu Filebeat pro sledování. Všechny zápisy do těchto souborů jsou poslány na server do ELK Stacku, kde jsou shromažďovány.

Tato část, popsána výše, tedy poskytuje tzv. *„Tamper Mechanisms“* (viz podsekce 2.3.5, Ochrana proti fyzické manipulaci. Jedná se zejména o tzv. *„Evidence“* a *„Detection“* díky záznamů v ELK Stacku a skriptů pro detekci manipulace spuštěných na zařízení.

### 3.5.1 Reakce na detekci neoprávněné manipulace se zařízením

Skripty pro detekci manipulace souvisí také s tzv. *„Response“*, čili s reakcí na nestandardní události. Je třeba zdůraznit, že hlavním cílem ochrany software nejsou samotné skripty pro detekci manipulace, ale především všechny citlivé informace, které jsou na zařízení, a to například tzv. *„AWS Credentials“*, díky kterým se zařízení ověřuje v AWS cloudu. Dále pak program na záznam zvukových nahrávek a samotné zvukové nahrávky uložené v zařízení.

Pro ochranu a zvýšení zabezpečení programu pro záznam zvukových nahrávek a nahrávání do cloudu bylo postupováno specificky. Zabezpečením programu se v této chvíli myslí zamezení zkopírování kódu programu. Postup aplikovaný na tento skript může sloužit jako příklad, jak by bylo možné zabezpečit další programy (např. pro detekci zvukových anomálií) spouštěných v budoucnu na zařízení.

Jedna z možností, jak ochránit kód, je použití tzv. *„obfuscating“* metod, které však pro použitý programovací jazyk Python nejsou vhodné (viz podsekce 2.6.7, Mechanismus reakce na detekci manipulace). Další možností je šíření programu v binární podobě. Ta je pro uživatele takřka nečitelná. Bohužel jazyk Python je znám tím, že za pomocí specifických nástrojů je celkem snadné obnovit původní kód z binárního programu.

Šíření binárního kódu není špatná myšlenka a do značné míry na ni staví nástroj *„Cython“* [39], který byl použit pro potřeby zvukového senzoru. Jedná se o optimalizovaný statický kompilátor, který umožňuje transformovat programy v jazyce Python do jazyka C. Program v jazyce C je následně kompilován do binární podoby. Tato transformace způsobí, že neexistuje takřka žádná šance, jak z binárního spustitelného programu obnovit původní program v jazyce Python. Omezení nástroje *„Cython“* je pouze v tom, že je nutné kompilovat binární soubory na dané platformě. Spustitelný soubor *„record“*, který je nahráván do zařízení pomocí Ansible a Role *„recording“* (viz sekce 3.3, Orchestrace zařízení pomocí Ansible) byl původně vytvořen ve zvukovém senzoru pomocí nástroje *„Cython“*.

Při detekci jakéhokoliv typu manipulace se zvukovým senzorem je spuštěna sekvence příkazů, které znamenají odpověď na danou reakci. Nejprve je vypnuto nahrávání. Následně je odpojena třetí oddílová jednotka se souborovým systémem XFS. Dále dochází k mazání binárního souboru *„record“*, souborů potřebných k připojení zařízení do cloudu a klíče pro odemčení již zmíněného třetího oddílu.

Pro bezpečné odstranění souborů se používá nástroj *„wipe“* [40], jelikož soubory smazané pomocí konvenčního nástroje *„rm“* lze obnovit. Nástroj *„wipe“* v podstatě přepíše soubory určené pro smazání v paměti. Záznam reakce je odeslán do ELK Stacku, kde je možné dohledat, jaký typ mechanismu pro detekci manipulace danou reakci spustil a v jaký čas.

Zvukové nahrávky se na zařízení nacházejí pouze dočasně mezi dobou jejich pořízení a nahrání do cloudu. Poté jsou smazány. Pokud by však nahrávání do cloudu z nějakého důvodu (např. pomalý internetový přenos) trvalo déle a došlo by k odcizení zařízení, mohly by v paměti zařízení některé zvukové nahrávky zůstat.

Tato na první pohled nevinná situace by však v některých případech vedla k velkým bezpečnostním problémům. Firma Neuron soundware již dříve chtěla pořizovat nahrávky v průmyslových oblastech se zvýšenou bezpečnostní kontrolou jako je např. továrna Škoda Auto. Firma má vysoké bezpečnostní nároky hlavně z důvodu ochrany výrobních technik a inovací. Nahrávka, na které by se nacházel, třeba jen omylem, záznam dvou dělníků hovořících o novém výrobním stroji by vedla k porušení bezpečnostních předpisů firmy. Je proto nutné zaměřit se i na ochranu pořízených záznamů.

Pro ochranu záznamů byla vytvořena speciální disková jednotka s XFS souborovým systémem a byl využit mechanismus šifrování pomocí nástroje *„dm-crypt“* a nadstavby *„LUKS“* (více v sekci 3.3, Orchestrace zařízení po-

mocí Ansible). Ansible Role *„crypto“* na zařízení kopíruje klíč pro dešifrování třetí oddílové jednotky, a také skript s názvem *„closexfs.sh“*. Jakmile dojde k detekci nestandardní události, v první řadě dojde k zastavení *Systemd service* pro nahrávání zvuků. Dále dojde k vyvolání skriptu *„closexfs.sh“*, který odpojí virtuální dešifrovanou vrstvu vytvořenou pomocí nástrojů *„dm-crypt“* a *„LUKS“* a tuto virtuální vrstvu uzavře. Dále dojde ke smazání klíče, který je potřebný pro otevření dané diskové jednotky. Následné zavolání skriptu *„openxfs.sh“* skončí s chybou, jelikož se nepodaří daný oddíl dešifrovat.

Tato reakce však není destruktivní z pohledu dalšího fungování zařízení. Jelikož dochází k dešifrování a připojení virtuální vrstvy až po provedení tzv. *„bootovací“* sekvence zařízení, neúspěšné připojení nemá na start zařízení vliv. Je tedy možné se vzdáleně na zařízení připojit a obnovit daný klíč. Zařízení do úspěšného dešifrování a připojení virtuální vrstvy nebude schopné pořizovat nahrávky.

## 3.6 Log Management

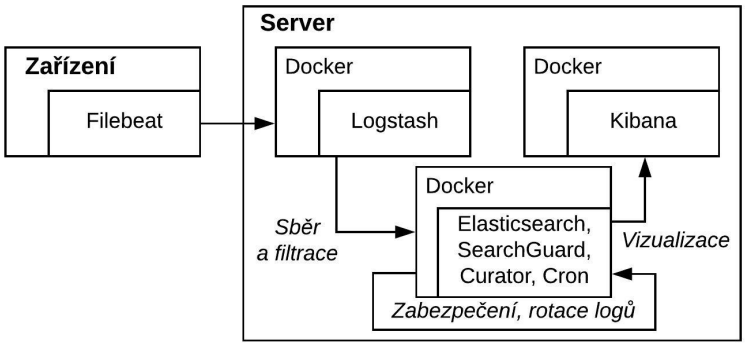
Firma Neuron soundware měla požadavek zlepšit monitoring všech zařízení. Počítá se s nasazením zařízení do průmyslových zón a jiných lokalit, kde může docházet k nestandardnímu chování zařízení vlivem např. špatného internetového připojení. Tyto lokality jsou k podobným zařízením často nevlídné a může docházet i k vnějšímu poškození. Je také vhodné zmínit, že zařízení jsou stále ve fázi vývoje či prvního nasazení do provozu a stále dochází k ladění funkcionality. Je třeba počítat i s tím, že v některých výrobních halách může vypadnout proud či být během směn úmyslně vypnut, což způsobí i vypnutí zařízení dodávaných firmou Neuron soundware. Vypnutí je třeba odlišit od jiných „nečekaných“ způsobených vlivem chyby.

Z analýzy možných řešení bylo vybráno spojení ELK Stacku (tzn. Elasticsearch, Logstash a Kibana) na straně serveru a na straně zařízení platforma Beats, a to zejména Filebeat pro monitorování a záznam celých souborů (popsáno níže). Schéma celkové infrastruktury viz obrázek 3.6.

### 3.6.1 Docker

Pro realizaci procesu logování a následný management je nejdříve potřeba vysvětlit tzv. „Docker“ [41]. Jedná se o software, který poskytuje virtualizované prostředí[62] pro běh aplikací. Pro zjednodušení se často princip Dockeru přirovnává k přepravím kontejnerům. Software poskytuje způsob vytvoření kontejneru podle specifických požadavků. V tomto kontejneru poté běží pouze jedna aplikace.

[62]Prostředí, které umožňuje přistupovat k dostupným zdrojům jiným způsobem, než jakým fyzicky existují.




Obrázek 3.6: Log management infrastruktura.

Struktura Dockeru využívá několik Docker Registry, což jsou repozitáře s Docker images. Tyto images jsou elementární jednotkou a slouží k vytváření Docker kontejnerů. Pokud by základní image nestačil a uživatel by musel každý kontejner rozšiřovat např. o specifické nastavení, lze v prostředí Dockeru také upravit samotný výchozí image.

Velkou výhodou Dockeru je přenositelnost a jednoduché použití. Jakmile si uživatel vytvoří Docker kontejner nebo upraví již existující, může tento kontejner spustit na jakémkoli podporovaném zařízení bez ohledu na hardwarovou specifikaci.

### 3.6.2 Docker-elk a SearchGuard

Důkladnější popsání software pro virtualizaci s názvem Docker byla prerekvizita pro pochopení následující části. Jak již bylo zmíněno, na straně serveru byla vybrána trojice nástrojů ELK, zejména pro svou rozšířenost. I když tyto nástroje poběží nepřetržitě na straně serveru, nově založené startupy často využívají cloudových služeb a k těmto službám mohou mít různé množství dotací ze strany investorů. Čas od času se může stát, že je nutné migrovat instance infrastruktury z jednoho cloudu do druhého. Jakmile se firmy rozrůstají, často investují do svého hardwaru, aby přestaly být závislé na službách třetích stran nebo je to finančně výhodnější. Firma Neuron soundware není výjimkou.

Tuto migraci může do jisté míry zjednodušit běh aplikací v již zmíněném Dockeru, který se postará o mezivrstvu pro odstínění hardwarových vlastností od aplikačních. Na jednom stroji může běžet více Docker kontejnerů, a to znamená více aplikací. V našem případě kontejnery obsahující aplikace:

- Elasticsearch.
- Logstash.
- Kibana.

Byl testován i kontejner, který zajišťoval běh všech aplikací. Tato možnost byla zprvu jednodušší na konfiguraci zejména kvůli tomu, že nebylo nutné nastavovat závislosti, protože aplikace běžely ve stejném prostředí. Tato možnost však odporuje standardům platformy Docker, a také tato možnost byla méně přehledná a při konfiguraci nebylo zcela zřetelné, kterou aplikaci se nepodařilo spustit.

Jednotlivé kontejnery vychází z oficiálních Docker images daných aplikací. Pro jednoduché spuštění všech aplikací jedním příkazem byl napsán tzv. *„docker-compose“*, který definuje závislosti jednotlivých kontejnerů. Zejména bylo třeba nastavit přesměrování portů využívaných aplikacemi mezi standardním a virtuálním prostředí, virtuální sít pro spojení všech kontejnerů, a také zálohu dat.

Běh aplikací v Dockeru je velmi specifický a jednou z vlastností je, že všechna data, která byla vytvořena či zpracována v Dockeru nejsou perzistentní při vypnutí Docker kontejneru. Tato vlastnost způsobí to, že jakmile dojde k zastavení Dockeru, dojde i ke ztrátě dat. Správce služeb se může rozhodnout vypnout či jen restartovat služby v Dockeru např. z důvodu již zmíněné migrace. Ztráta dat je nevyhnutelná. To je z pohledu dlouhodobého monitorování zařízení nepřijatelné.

Tuto vlastnost Dockeru je možné řešit několika způsoby. Jedním z nich je nastavení tzv. *„Docker volume“* uvnitř konfiguračního souboru *„docker-compose“*. Toto nastavení způsobí sdílení adresáře mezi serverovým prostředím a virtuálním prostředí uvnitř Dockeru. Všechny změny, které byly provedeny v adresáři Dockeru se provedou i vně virtuálního prostředí. Při opětovném spuštění Dockeru dojde k načtení dat z vnějšího prostředí do virtuálního, čímž nedojde ke ztrátě dat.

Pomocí *„Docker volume“* je doporučeno řešit i konfiguraci jednotlivých aplikací. Ty většinou obsahují uvnitř kontejneru několik konfiguračních souborů. Vlastnost neperzistence se projeví také u těchto souborů, pokud bychom je chtěli upravovat. Při každé změně by bylo třeba ukládat celý Docker kontejner, což není praktické. Sdílení konfiguračních souborů pomocí *„Docker volume“* a jejich lokální změny způsobí i změny uvnitř Dockeru (je třeba kontejner restartovat), což je výrazně elegantnější.

Další nastavení Dockeru se týkalo zabezpečení. V rámci *„docker-compose“* byly definovány cesty k certifikátům od certifikační autority, které využívají aplikace Elasticsearch a Kibana. Zabezpečení služeb v Dockeru byla věnována výraznější pozornost. Ani Elasticsearch, ani Kibana v základním nastavení neobsahují např. správu uživatelů pro zajištění procesu autorizace při přístupu ke službě. Zprvu bylo vyzkoušeno rozšíření s názvem X-pack, které je doporučeno ve spojení s používáním Elasticsearche. Konfigurace X-packu ve spojení s ELK Stackem je složitá, a proto je pro první spuštění doporučeno využít připravený oficiální Docker image poskytující základní konfiguraci ELK Stacku s X-packem.

Zajímavostí tohoto připraveného image je více módu spuštění. Od verze šest X-pack neobsahuje proprietární hesla základních uživatelů (jedná se o bezpečnostní prvek, kdy není možné použít rozšíření ELK Stacku s výchozími hesly). Při prvním spuštění je tedy nutné spustit Docker kontejner z Docker image v tzv. „Bootstrap“ módu, při kterém dojde k nastavení základních hesel pro uživatele „elastic“, „kibana“ a „logstash“, kteří jsou použiti při autorizaci komunikace mezi jednotlivými aplikacemi. Správce služeb při následném startu Docker kontejneru v normálním módu využije znalost těchto hesel k přihlášení do Elasticsearche a Kibany pro vytvoření dalších uživatelů.

Práce s doplňkem X-pack byla výrazně jednodušší, než jeho prvotní nastavení. Při konzultaci s firmou Neuron soundware však tato možnost nebyla přijata hlavně z důvodu placené verze. Bylo také definováno, že hlavní požadavek je zabezpečení ELK Stacku a není nutné řešit např. monitorování samotných aplikací na serveru (hlavní snaha je monitorovat zařízení). Proto byla zvolena bezplatná alternativa s názvem *„SearchGuard“* [42] uvedena v tabulce 2.2, X-pack alternativy.

Pro využití zabezpečení pomocí doplňku SearchGuard neexistuje žádný oficiální Docker image od společnosti Elastic (konkurence jako zjevný důvod). Existuje však celá řada manuálů, jak správně nainstalovat SearchGuard v prostředí Dockeru. Základním požadavkem na správnou funkčnost spojení ELK Stack a SearchGuardu je nutnost udržovat všechny aplikace ve stejných verzích.

Toto rozšíření poskytuje v základu několik zajímavých skriptů, které přijdou vhod při nastavení SearchGuardu. Doplněk načítá základní nastavení např. rolí, uživatelů a jejich hesel z konfiguračních souborů. Aby se v těchto souborech neocitly citlivé informace, je možné využít např. skript *„hash.sh“* pro hashování hesel. Dále je možné využít velké množství připravených skriptů pro generování certifikátů, které byly využity při spojení s Filebeat (viz podsekce 3.6.4, Filebeat).

### 3.6.3 Rotace logů

Bylo potřeba vyřešit množství dat, která budou přicházet od všech zařízení na server. Firma Neuron soundware v budoucnu počítá s nasazení stovek až tisíců zařízení. Takové číslo by mohlo přeplnit paměť serveru. ELK Stack ani doplněk SearchGuard neřeší žádnou agregaci dat. K mazání starších logů je doporučen doplněk s názvem *„Curator“* [43].

Doplněk je schopný mazat data z databáze na základě:

- Doby uložení dat v databázi.
- Celkové velikosti dat v databázi.

Jelikož doplněk pracuje pouze s Elasticsearchem, bylo potřeba jej doplnit jen do tohoto Docker kontejneru. Hlavní konfigurační soubory potřebné pro

běh Curatoru jsou *„curator.yml“* a *„delete_indices.yml“*. V prvním zmíněném jsou potřebné údaje pro připojení se k aplikaci Elasticsearch. Druhý zmíněný popisuje pravidla, na základě kterých bude docházet k mazání dat.

Po konzultaci s firmou Neuron soundware byla nastavena pravidla pro mazání starších dat na základě časového intervalu uložení v databázi a celkové velikosti dat. Hodnoty lze však jednoduše přenastavit. V budoucnu, jak bude množství dat přibývat, je to i dosti pravděpodobné. Celkové množství dat přicházejících na server se v tuto chvíli velmi špatně odvozuje, jelikož bude záležet na mnoha faktorech (např. úroveň logování či počtu zařízení).

Curator je nástroj, který ve svém základu neumožňuje nastavení pravidelného mazání. Jakmile se nástroj spustí, zkontroluje databázi, jestli v ní nejsou data, která by vyhovovala pravidlům pro mazání, a pokud ano, tak je smaže. Pro pravidelnost bylo zapotřebí využít utilitu s názvem *„Cron“* [44].

Tato utilita slouží jako plánovač úloh a podle nastavení (dle určitého času či stavu systému) dokáže pravidelně spouštět příkazy, krátké skripty nebo i rozsáhlé programy. Byl proto použit ve spojení s doplňkem Curator pro jeho každodenní spuštění. Tímto je docílena potřebná funkcionalita. Cron je také součástí Docker kontejneru s Elasticsearchem, a tak dojde k jeho startu automaticky při startu všech kontejnerů ELK Stacku.

### 3.6.4 Filebeat

Z open-source rodiny doplňků nazvaných *„Beats“* popsaných v podsekci 2.4.1, Beats, byl vybrán zástupce Filebeat pro předávání a centralizaci logů a souborů. Filebeat, jako agent, sleduje změny v předem daných souborech. Při přidání nového záznamu do souboru tento záznam odesílá do ELK Stacku. Tímto lze jednoduše centralizovat všechny informace o stavu mnoha zařízení do jednoho místa.

Instalace Filebeat souvisí s nastavením ELK Stacku, protože je zapotřebí, aby obě instance mezi sebou komunikovali. Při instalaci podle oficiální dokumentace bylo zapotřebí vyřešit jeden problém. Neexistuje přímo instalační balíček pro architekturu ARM (Raspberry Zero) a balíček určený obecně pro Linux nejde díky chybě použít. Tato chyba byla vyřešena díky vytvoření vlastního binárního souboru Filebeat upraveného pro architekturu ARM podle fóra podpory doplňků Beats.

Byl vytvořen skript *„filebeat_arm.sh“*, který pomocí virtuálního prostředí Dockeru generuje binární soubor s názvem *„filebeat“*. Tento binární soubor se následně použije při vytváření nového Debianího balíčku vhodného pro instalaci Filebeat na architektuře ARM.

Vytvoření Debianího balíčku je podobné jako skriptem *„make_deb.sh“*, který byl vytvořen speciálně pro potřeby zvukového senzoru a slouží k vytvoření balíčků se skripty pro detekci manipulace se zařízením, pro jejich snadnější instalaci. Skript pro vytvoření balíčku Filebeat je však o něco komplikovanější (vychází ze struktury oficiálního instalačního balíčku Filebeat ur-

čeného pro systém Linux), a tak byl pro tyto účely vytvořen nový s názvem *„filebeat_deb.sh“*.

Dále je třeba zmínit soubor *„filebeat.yml“*. V tomto souboru se nachází veškeré nastavení programu Filebeat, a to zejména seznam souborů, které má agent sledovat pro změny a nastavení spojení s ELK Stackem. Skript *„filebeat_deb.sh“* tento soubor vyžaduje pro správné vytvoření Debianího balíčku. Pro generování souboru *„filebeat.yml“* byl vytvořen skript s názvem *„filebeat_yml.sh“*. Správné pořadí spuštění skriptů tedy je:

1. *„filebeat_yml.sh“* – generování souboru s nastavením.
2. *„filebeat_arm.sh“* – vytvoření binárního souboru *„filebeat“*.
3. *„filebeat_deb.sh“* – vytvoření Debianího balíčku pro instalaci Filebeat na zvukovém senzoru.

Výsledkem těchto skriptů je Debianí balíček *„nsw-filebeat.deb“*. Pro instalaci tohoto balíčku byla vytvořena Ansible Role *„filebeat“*. Program Filebeat je následně spuštěn opět jako *Systemd service* vždy při startu zařízení.

V desce Raspberry Zero jsou základní stavové informace z jádra spravovány programem *„dmesg“* [45]. Jedná se o binární program, který při spuštění vypíše všechny zprávy jádra od zapnutí zařízení. V této sekvenci se mohou nacházet základní chyby (např. selhání některé části či periferie), které je třeba centralizovat do ELK Stacku. Filebeat však nedokáže získávat informace z binárního souboru.

Operační systém Raspbian nabízí několik způsobů, jak zařídit spuštění nějakého příkazu či celého skriptu vždy po startu zařízení. Jedním ze způsobů je editace souboru *„rc.local“*, který se nachází ve složce */etc* (dále možné použít např. program *„Cron“* zmíněný v podsekci 3.6.2, Docker-elk a SearchGuard). V tomto souboru byl přidán řádek: *„/bin/dmesg -f > /var/log/dmesg.log &“*. Parametr *-f* způsobí nepřetržité spuštění programu *„dmesg“* a zajistí tak předání nových zpráv jádra do souboru *„dmesg.log“*, se kterým dokáže Filebeat pracovat a jeho změny odesílat do ELK Stacku. Znak ampersand na konci řádku způsobí rozdvojení procesu, aby nedošlo k selhání startu zařízení.

### 3.6.5 Zabezpečení komunikace mezi Filebeat a ELK Stackem

Jelikož dochází ke komunikaci mezi zařízením a serverem, bylo potřeba tuto komunikaci určitým způsobem zabezpečit. Hlavně na straně serveru bylo třeba ověřit, že dané zařízení má povoleno odesílat data na server.

K zabezpečení komunikace mezi zařízením a serverem byly použity klíče generované nástrojem *„search-guard-ssl“*, který je součástí doplňku SearchGuard. Na straně serveru dochází ke generování jedinečných klíčů pro ověření komunikace mezi ELK Stackem a Filebeat. Tyto jednoznačné klíče jsou poté při instalaci nakopírovány do zařízení pomocí Ansible Role s názvem *„filebeat“* viz sekce 3.3, Orchestrace zařízení pomocí Ansible.

Všechny klíče zařízení vychází z certifikační autority uložené na serveru. Pokud by nastal problém s danými klíči (např. zařízení i s klíči by bylo odcizeno), stačilo by pouze vygenerovat novou certifikační autoritu a aktualizovat klíče zbylých zařízení (např. pomocí Ansible). Aby byl splněn tzv. *„Tamper Mechanisms“* a zejména *„Evidence“*, je nutné, aby zařízení po celou dobu běhu obsahovalo dané klíče, aby správně fungovalo informování o nestandardních událostech.

Kapitola 4

# Testování

Kapitola pojednává o testování zvukového senzoru. Popisuje, na které části byl kladen důraz a následně analyzuje výsledky testů. Je třeba zmínit, že zvukový senzor je první prototyp. Pro široké nasazení do provozu by bylo nutné připravit dlouhodobé testy, které by důsledně prověřily konstrukci zařízení.

## 4.1 Sestavení zařízení a nastavení

První fáze testování se zaměřila na samotné propojení hardwaru, zejména na:

- Propojení tří hlavních komponentů (Audio Injector Zero, PiZ-Uptime a Raspberry Pi Zero).
- Funkčnost modulu Audio Injector Zero.
- Připojení regulátoru napětí LE33 k modulu PiZ-Uptime.
- Připojení modemu Huawei E3372h a modulu světelného čidla s fotodiodou k Raspberry Pi Zero.
- Velikost plastové krabičky.
- Použití micro SD karet s různou kapacitou.
- Automatickou instalaci služby Dataplicity.
- Dobu potřebnou k nastavení jednoho zvukového senzoru.

I když jsou např. dvě hlavní části (Audio Injector Zero a Piz-UpTime) přímo konstruovány pro základní modul Raspberry Zero, nebyla zajištěna kompatibilita všech tří částí. Moduly jsou připojeny k Raspberry Zero pomocí GPIO pinů (viz příloha C, Použité GPIO piny). Pokud by nastal problém s kompatibilitou, bylo by možné připojit modul Piz-UpTime k Raspberry Zero pomocí USB kabelu. Obě tyto varianty byly testovány a chování zvukového

senzoru bylo stejné v obou případech. Je však výrazně elegantnější připojit modul přes GPIO piny. Proto byla zvolena tato varianta.

Dále je třeba zmínit, že modul Audio Injector Zero se prodává jako stavebnice a je zde nutné pájení jednotlivých částí. Funkčnost desky byla testována za provozu (viz sekce 4.2, Provoz zvukového senzoru), kdy byla zjištěna nefunkčnost mikrofonu umístěného na desce. Připojovaný mikrofon má jasně danou polaritu. Díky nesouměrnému umístění „nožiček“ dané součástky nebylo možné jednoduše mikrofon na desku připájet. Ve správném zapojení se totiž na desku nevešel. Bylo tedy nutné prodloužit „nožičky“ mikrofonu a umístit jej mimo desku. Mikrofon byl využit v první části testování nahrávání zvukového senzoru. Pro provoz zařízení se s ním však nepočítá, jelikož je umístěn uvnitř plastového obalu, kde jsou zvukové nahrávky okolí utlumené a zkreslené.

Dále došlo k testování připojení všech ostatních částí zvukového senzoru jako např. modul světelného čidla s fotodiodou nebo modem Huawei E3372h. Během testování byla nejdříve ověřena jejich funkčnost pouze s Raspberry Zero a následně se všemi ostatními komponentami. Zmíněné části obsahují stavovou LED, podle které je možné jednoznačně určit, zda jsou připojeny správně. Připojení modemu Huawei E3372h bylo následně softwarově kontrolováno pomocí linuxového příkazu *„lsusb“*.

Modul Piz-UpTime bylo potřeba, pro detekce odpojení zařízení od zdroje napájení, doplnit o regulátor napětí LE33 z 5 V na 3,3 V (popsáno v sekci 3.1, Sestavení hardwaru). Pokud by se nepodařilo správně zapojit regulátor napětí, připojení 5 V na GPIO piny by způsobilo zničení desky Raspberry Zero. Před připojením bylo tedy napětí ověřeno multimetrem. Části sloužící k detekci manipulace se zvukovým senzorem a zejména jejich správné připojení bylo dále testováno softwarově pomocí výpisů změn na daných GPIO pinech v příkazové řádce.

Velmi důležitá hardwarová část zvukového senzoru je plastová krabička, která byla navržena a vytištěna na 3D tiskárně. Bylo zapotřebí otestovat rozměry krabičky. Stačilo vložit hardwarové části do krabičky a tu uzavřít. Rozměry krabičky 80 x 140 x 60 mm (viz příloha E, Nákres plastové krabičky) by bylo možné ještě snížit na rozměry 80 x 125 x 60 mm. Tyto hodnoty jsou však pro použitý hardware hraniční. Pokud by v budoucnu došlo k nahrazení např. modemu Huawei E3372h menší variantou, rozměry krabičky by bylo možné ještě zmenšit.

Pro sestavení a následný provoz zvukového senzoru byly dále testovány různé micro SD karty s kapacitou 16 GB a 64 GB. Jelikož dochází k vytváření nových oddílových části paměti během instalace operačního systému, zejména skriptem *„firstStart.sh“* (viz sekce 3.2, Nastavení operačního systému před prvním startem), bylo nutné otestovat přizpůsobení oddílových částí k velikosti dané karty. Při startu skriptu *„firstStart.sh“* je možné velikost paměti přizpůsobit. Karty s větší kapacitou než 64 GB nebyly testovány.

Během testování nastavení zvukového senzoru pomocí programu Ansible byla zjištěna chyba v přidělených právech a skupin nového uživatele nazva-

ného *„nsw“*, pod kterým se spouštěl druhý Playbook s názvem *„nsw.yml“*. Zpočátku byl uživatel přidělen pouze do skupiny *„root“*. Jelikož však během Playbooku dochází k instalaci nového ovladače zvukové karty, tento uživatel neměl oprávnění k takovému úkonu. Uživatel *„nsw“* byl přidělen do skupin podle výchozího uživatele *„pi“* a problém se vyřešil.

Při testování služby Dataplicity pro vzdálenou správu zařízení a zejména schopnost automatické instalace služby v zařízení během prvního startu byla zjištěna chyba v oficiálním návodu instalace Dataplicity. Během instalace zařízení ještě nemělo aktualizovaný čas a nebylo možné se připojit na server Dataplicity pro stažení potřebných souborů. Instalace také prodloužila první start zařízení, který musel být tomuto přizpůsoben (vice v sekci 3.4, Nahrávací software a vzdálená správa). Následné testování zařízení neodhalilo další chyby.

První start zařízení a následná orchestrace programem Ansible byly části, které byly nejdéle testovány během vývoje zvukového senzoru proto, že docházelo k jejich průběžným změnám. Pro orchestraci programem Ansible byly testovány jednotlivé Role zvlášť a následně celý Playbook, jelikož je velmi důležité jejich pořadí. Není např. možné upravit nastavení hlasitosti připojených mikrofonů před samotnou instalací zvukové karty.

Pro testování byl k dispozici notebook s procesorem Intel Core 2 Duo a pamětí RAM 4 GB. Čas nahrávání operačního systému na micro SD kartu a následné nastavení pomocí skriptu *„firstStart.sh“* zabralo přibližně šest minut. Jedná se již o starší model notebooku. Nahrávání operačního systému by bylo možné zrychlit při použití novějšího typu s výkonnějšími komponentami. První start zařízení kvůli stahování potřebných souborů k instalaci Dataplicity trvá v průměru necelých pět minut.

První Playbook s názvem *„default.yml“* zabere asi patnáct sekund. Druhý Playbook s názvem *„nsw.yml“* je výrazně delší. Během testování bylo naměřeno v průměru dvacet osm minut. Po této době stačí zařízení kdekoliv připojit k napájení. Po startu začne automaticky sbírat potřebná data. Je důležité poznamenat, že fázi orchestrace pomocí Ansible je možné paralelizovat, jelikož je možné nastavovat více zařízení současně. Tuto fázi by bylo možné výrazně zkrátit pomocí připravení upraveného operačního systému Raspbian již se všemi soubory a skripty potřebnými pro běh zařízení (viz kapitola 5, Budoucí práce).

## 4.2 Provoz zvukového senzoru

Testování provozu zvukového senzoru bylo zaměřeno na jeho hlavní funkce:

- Nahrávání zvukových záznamů.
- Přenos záznamů do cloudu.
- Detekce neoprávněné manipulace se zařízením.

- Vyvolání reakce na neoprávněnou manipulaci se zařízením.
- Logování stavových informací.
- Vzdálená správa zařízení.

Testování nahrávání probíhalo se zapůjčeným mikrofonem firmy Neuron soundware. Mikrofon byl připojen k RCA konektoru a byl připevněn ke zdroji zvuků. Zařízení bylo zprvu připojeno k Wi-Fi, následně byla Wi-Fi vypnutá a senzor byl připojen pomocí modemu Huawei E3372h k mobilnímu internetu. Přenos nahrávek byl pomalejší vlivem pomalejšího připojení. Vliv na funkčnost zařízení to však nemělo. Nahrávky byly následně staženy z úložiště AWS S3 a poslechnuty. Kvalita zvuku a jeho hlasitost odpovídala zdroji zvuku. Následně byl soubor zvukového záznamu ve formátu WAV otevřen v programu Audacity. Zařízení automaticky nahrává na oba vstupní RCA porty. Při nahrávání na jeden vstupní port byl v Audacity druhý port zřetelně ztlumený. Program Audacity také potvrdil další nastavené parametry nahrávání jako např. rozsah frekvencí či délku zvukové nahrávky.

Testování detekce manipulace se zařízením se soustředilo na všechny tři implementované typy:

- Detekce otevření či porušení obalu pomocí vodiče ve stěnách krabičky.
- Detekce otevření krabičky pomocí modulu světelného čidla s fotodiodou.
- Detekce změny napájení z adaptéru na baterii.

Aby bylo testování detekce manipulace snadnější a efektivnější, byla vytvořena Ansible Role *„recovery“*, která vrátí zpět všechny změny v zařízení provedené reakcí na detekci manipulace se zvukovým senzorem. Reakce na detekci je spuštěna, i když dojde např. pouze k výpadku proudu. Obnovení zařízení do původního stavu po nestandardní událostí je tedy velmi důležité.

Testy detekce otevření či porušení plastového obalu pomocí vodiče ve stěnách krabičky probíhaly tak, že docházelo k pravidelnému rozpojení vodiče. Detekce rozpojení byla takřka okamžitá a ihned poté se spustila reakce na detekci manipulace. Vše proběhlo bez chyb.

Další testy byly zaměřené na detekci otevření krabičky pomocí modulu světelného čidla s fotodiodou. Během testů bylo několikrát otevřeno víko plastové krabičky. Bylo zjištěno, že detekce vniknutí světla do krabičky má určitou prodlevu. K detekci došlo vždy až po několika sekundách.

Testy zaměřené na detekci změny napájení z adaptéru na baterii byly podobné testům nataženého vodiče ve stěnách krabičky. Po zapnutí zařízení byl opakovaně odpojen napájecí zdroj a následně připojen. Detekce je velmi rychlá a reakce byla spuštěna hned po prvním odpojení zdroje.

Další klíčovou funkcí zařízení je logování stavových informací do ELK Stacku. Tato část byla testována pravidelným restartováním zařízení. Informace ze startu zařízení byly pomocí agenta Filebeat po startu nahrány do

Elasticsearche. Bylo možné je kontrolovat v Kibaně. Při detekci manipulace se zvukovým senzorem (viz výše) byly tyto události evidovány a bylo je možné kontrolovat opět v Kibaně.

Při testování logování stavových informací bylo zjištěno, že během skriptu pro pořizování zvukových nahrávek dochází k pravidelnému záznamu upozornění ve tvaru *„I2S SYNC error“* v utilitě *„dmesg“*. Podle oficiálního fóra[63] Raspberry se jedná o standardní hlášení ovladače *„bcm2835 i2s“*. Upozornění je způsobené připojením nové zvukové karty Audio Injector Zero a nainstalováním nových zvukových ovladačů. Hláška se týká synchronizace hodinového signálů a dochází k ní, jakmile je *„bcm2835 i2s“* konfigurován jako tzv. *„Clock slave“*. Na funkci zvukové karty hláška nemá vliv.

Další fáze testování byla zaměřená na rychlost připojení modemem Huawei E3372h a zda má vliv natažený vodič ve stěnách krabičky na výslednou rychlost připojení (zda nedochází ke stínění). Pro měření rychlosti byla použita utilita *„speedtest-cli“* spouštěná v příkazové řádce. Rychlost připojení k internetu byla testována dvanáctkrát. Během první poloviny testů bylo zařízení umístěno vně plastové krabičky (varianta „a“). V druhé části testů bylo zařízení umístěno uvnitř plastové krabičky s vodičem nataženým ve stěnách (varianta „b“). Výsledky testů viz tabulka 4.1.

| | **Čidlo vně (varianta „a“)** | | **Čidlo uvnitř (varianta „b“)** | |
|---|---|---|---|---|
| **Číslo testu** | **Download (Mbit/s)** | **Upload (Mbit/s)** | **Download (Mbit/s)** | **Upload (Mbit/s)** |
| 1. | 25,6 | 15,99 | 20,44 | 14,25 |
| 2. | 24,45 | 13,06 | 24,83 | 13,54 |
| 3. | 25,63 | 16,63 | 26,42 | 14,5 |
| 4. | 24,17 | 14,93 | 20,81 | 12,67 |
| 5. | 24,49 | 13,6 | 11,27 | 11,05 |
| 6. | 27,27 | 13,38 | 21,64 | 10,61 |
| | **25,27** | **14,6** | **20,9** | **12,77** |

Tabulka 4.1: Rychlost připojení k internetu modemem Huawei E3372h.

Testy rychlosti připojení prokázaly, že plastová krabička s vodičem ve stěnách vytváří stínění. Rychlost připojení k internetu je nižší. Konkrétně je stahování v průměru pomalejší o $4,37$ Mbit/s (to je 17%) a nahrávání o $1,83$ Mbit/s (to je 12,5%). Rychlost nahrávání je však pro zvukový senzor mnohem důležitější. Průměrná rychlost $12,77$ Mbit/s je pro provoz zvukového senzoru dostačující.

Dále byla testována doba běhu zařízení napájeného pouze AAA baterií 3,7 V s kapacitou 2800 mAh. Zařízení bylo primárně konstruováno pro napájení z elektrické sítě. Pokud by bylo možné napájet zařízení pouze z baterie po

[63] Více na `https://raspberrypi.stackexchange.com/questions/70614/audio-injector-codec-board-wm8731-bcm2835-i2s-20203000-i2s-i2s-sync-error-rasp`

dobu několika hodin, výrazně by se rozšířila možnost nasazení zvukového senzoru. Během testů byla zjištěna průměrná výdrž zvukového senzoru na baterii 84 minut. Tato hodnota potvrzuje předpoklady, že zařízení napájené pouze baterií nelze používat pro dlouhodobé nahrávání zvukových dat.

V poslední řadě byly testy zaměřené i na využití služby Dataplicity pro vzdálenou správu. Byla testována hlavně automatická registrace zařízení do služby Dataplicity. Dále byla služba testována při různých druzích připojení zařízení k internetu (Wi-Fi a mobilní internet). Během testů bylo zjištěno, že navázání spojení mezi webovým rozhraním služby Dataplicity a zařízením, při použití modemu Huawei E3372h, trvá déle, než když je zařízení připojeno k Wi-Fi. Jakmile je však spojení navázáno, konfigurace zařízení pomocí Dataplicity je dostatečně rychlá.

## 4.3 Doplnění testů

Testování v laboratorních podmínkách proběhlo na výbornou. Zařízení je plně funkční. Dokáže zaznamenávat zvuk a ten následně odesílat do cloudu. Poskytuje několik bezpečnostních mechanismů pro zabezpečení citlivých informací. Dalším krokem před průmyslovým nasazením by bylo testování v podmínkách stálého prostředí výrobních hal na strojích a zařízeních v reálné čase. Bude nutné připravit podrobnou metodiku testování se zaměřením na:

- Odolnost zařízení v reálném provozu.
- Zaplnění paměti zařízení.
- Teplotu zařízení v normálních i extrémních podmínkách.
- Konektivitu zařízení pro vzdálenou správu.
- Funkčnost reakce na bezpečnostní hrozbu.
- Kvalitu zvukových nahrávek.

Poslední fází před ostrým nasazením do provozu by bylo vypracování podrobných pracovních postupů nasazení zařízení v reálném provozu.

Kapitola 5

# Budoucí práce

Kapitola popisuje možné pokračování či doplnění tématu diplomové práce: *„Zvukový senzor pro Internet of Things, jeho zabezpečení a vzdálená správa“*. Téma Internet věcí je velmi rozsáhle a stále se objevují nové technologie, které téma rozšiřují. I když byl zvukový senzor vytvořen podle zadaných požadavků, bylo by možné jej dále hardwarově i softwarově rozšířit a modifikovat (viz níže).

## 5.1 Rozšíření či modifikace hardwaru

Zvukový senzor s rozměry 80 x 140 x 60 mm (viz příloha E, Nákres plastové krabičky) pro některé případy použití může mít ještě příliš velké rozměry. Tyto rozměry jsou dány zejména velikostí LTE modemu připojeného prostřednictvím USB. Pokud by se podařilo modem nahradit menší variantou, rozměry celého zvukového senzoru by se výrazně zmenšily. Pokud by se zvukový senzor obešel bez LTE modemu a byl by nasazován pouze s podmínkou dostupného kvalitního Wi-Fi signálu, výrazně by se rozměry senzoru přiblížily k možným limitům daným velikostí základního modulu.

Firma Neuron soundware by měla ráda ve svém portfoliu zvukový senzor, který by byl zcela napájený z baterie. Tato varianta by mohla být modifikací zvukového senzoru, který by obsahoval UPS a velkokapacitní baterii. Došlo by tedy k nahrazení modulu Piz-UpTime, který nedokáže napájet zařízení více jak hodinu a slouží spíše jako bezpečnostní prvek. Bylo by také možné určitým způsobem snížit spotřebu zařízení (např. různými režimy činnosti zařízení). Pokud by však v budoucnu běžely na zařízení výpočetní modely pro detekci anomálií, došlo by s velkou pravděpodobností ke zvýšení spotřeby zařízení. Nebylo by tedy fyzicky možné napájet zařízení více jak několik hodit.

Zvukový senzor je možné vzdáleně spravovat, pokud je připojeno k internetu. Správci zařízení však nedokáží rozpoznat stav, kdy dojde k výpadku sítě a kdy k selhání zařízení. Tento stav by dokázala rozlišit technologie Sig-

fox[64]. Technologii by bylo možné využít pro odesílání tzv. *„Heartbeat monitor“* zpráv. Zařízení by odesílalo malé zprávy na server, aby informovalo o svém stavu. Pokud by na server dorazila zpráva Sigfox a souběžně by zařízení nebylo dostupné, bylo by jasné, že selhalo připojení k internetu.

## 5.2 Rozšíření či modifikace softwaru

Deska Raspberry Zero umožňuje komunikovat i prostřednictvím technologie Bluetooth. Tato funkce desky v práci nebyla využita. Technologii Bluetooth by bylo možné využít při instalaci zařízení a ke komunikaci mezi zařízením a aplikací v mobilním telefonu. Technik by mohl komunikací ověřit, že se zařízení úspěšně zapnulo, připojilo k internetu, přihlásilo do programu pro vzdálenou správu (v diplomové práci využito Dataplicity) a začalo nahrávat. Tato funkce by zjednodušila proces instalace.

Při konfiguraci zvukového senzoru a zejména jeho operačního systému se vždy použije aktuální a hlavně výchozí verze. Systém je nutno doplnit o nastavení pro potřeby konkrétního použití. Tento krok by bylo do budoucna vhodné zjednodušit tím, že by se vytvořil upravený operační systém Raspbian, který by obsahoval již v základu všechny potřebné utility, skripty a balíčky. Proces konfigurace by se tak zkrátil na pouhou instalaci operačního systému na micro SD kartu.

Program použitý v diplomové práci pro vzdálenou správu jménem Dataplicity nepatří do kategorie bezplatného softwaru. Pokud by firma Neuron soundware v budoucích letech nasadila stovky či tisíce zařízení, paušál placený za využití Dataplicity by výrazně narostl. Použití služby by bylo zbytečně drahé. Do budoucna by tedy bylo vhodné nahradit program levnější variantou. Další možnost je vytvoření vlastního programu s podobnými funkcemi.

Nahrávání zvuků pomocí zvukového senzoru ve formátu WAV a odesílání dat pro následné zpracování do internetového úložiště znamená velký datový tok a vyžaduje velký úložný prostor. Velikost dat by bylo vhodné určitým způsobem snížit zvukovou kompresí popsanou v podsekci 2.2.2, Nahrávání zvuku, (např. použitím formátu FLAC) nebo předzpracováním zvuků již na zařízení. Tato úspora by vedla i ke snížení síťových požadavků (zejména rychlosti nahrávání).

Zvukový senzor je primárně určen pro záznam zvuků a pro následné nahrávání záznamů do cloudu. V budoucnu by se mohl vyskytnout případ použití, kde by nebyl dostupný žádný typ internetového připojení (např. důlní šachty). Doplněním skriptů o možnost vypnutí nahrávání do cloudu by bylo možné ukládat záznamy pouze do paměti zařízení (micro SD karta). Doba tohoto typu nahrávání by se odvíjela od velikosti použité micro SD karty. Bylo nutné doplnit mechanismus informování o stavu paměti zařízení.

[64]Bezdrátový systém pro přenos malého množství dat.

V budoucnu bude nutné se zaměřit na snížení celkové ceny zvukového senzoru, jeho rozměrů a doplnění funkcionality, čímž by se výrazně rozšířily možnosti jeho nasazení.

# Závěr

Hlavním cílem diplomové práce bylo popsat pojem *„Internet věcí“* a zejména problematiku zabezpečení zařízení z této kategorie. Prozatím neexistují oficiální standardy zabývající se hlavními principy zabezpečení zařízení z *„Internetu věcí“*. Diplomová práce obsahuje souhrn základních bezpečnostních doporučení pro zařízení z kategorie *„Internetu věcí“*.

Dále bylo cílem práce aplikovat popsané bezpečnostní mechanismy při návrhu a realizaci senzoru pro záznam zvuků průmyslových zařízení pro následnou detekci anomálií firmou Neuron soundware.

V rámci diplomové práce vznikl na platformě Raspberry Pi zvukový senzor splňující požadavky definované firmou Neuron soundware. Hlavní funkce senzoru je nahrávání zvukových záznamů pomocí dvou RCA vstupů a připojených mikrofonů. Nahrávky jsou následně přenášeny do cloudu. Zařízení je dále vybaveno mechanismy pro detekci neoprávněné manipulace, jako je odpojení zařízení ze zdroje napájení, detekce otevření krabičky pomocí modulu světelného čidla s fotodiodou a detekce poškození plastové krabičky pomocí elektrického vodiče ve stěnách krabičky. Zachycení nestandardní události vyvolá následnou reakci jako např. odpojení šifrované části disku a smazání přístupového klíče. Stav zvukového senzoru je monitorován a zařízení je možné vzdáleně spravovat.

Testování zvukového senzoru zaměřené na hlavní funkce neodhalilo žádné závažné problémy. Zvukový senzor je plně funkční zařízení, které při laboratorním testování splnilo všechna zadaná kritéria. Pro plné průmyslové nasazení bude potřeba vypracovat metodiku testování a testovat zařízení v reálném prostředí výrobních hal za plného provozu s vlivem všech nepříznivých faktorů, které se zde mohou vyskytovat.

S celkovou cenou 3552 Kč a s rozměry 80 x 140 x 60 mm je senzor výrazně levnější a menší, než dosavadní zařízení zvané NB.6, které používá firma Neuron soundware pro sběr dat. Je však důležité dodat, že zvukový senzor vytvořený během diplomové práce nemá sloužit jako konkurent původního zařízení, ale má spíše doplnit portfolio používaných zařízení firmou. Při realizaci zvu-

kového senzoru byly některé zde použité principy (zejména se jedná o způsob monitorování zařízení) přejaty i do zařízení NB.6 vyvinuté a používané firmou Neuron soundware.

Na závěr je třeba konstatovat, že technologie použité během zpracování diplomové práce se neustále vyvíjí. Tento fakt znamená, že zvukový senzor je možné dále upravovat a zdokonalovat ve snaze např. o snížení ceny a rozměrů zařízení. Všechny vytyčené cíle práce byly splněny. Navržené a zkonstruované zařízení – senzor pro záznam zvuků průmyslových zařízení pro následnou detekci anomálií je plně funkční.

# Literatura


[1] FUCHS, Ondřej. *Internet of Things zařízení s podporou Bluetooth a CoAP*. Brno, 2016. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Musil Petr.

[2] VENKATESH, K., S. CHANDRAKANTH, J UMA MAHESH a Dr. K.V. NAGANJANEYULU. Internet of things. *International Journal of Innovations & Advancement in Computer Science* [online]. 2014, extbf3(8), 5 [cit. 2016-02-07]. ISSN 2347–8616.

[3] NEISSE, Ricardo, Gary STERI, Igor Nai FOVINO a Gianmarco BALDINI. SecKit: A Model-based Security Toolkit for the Internet of Things. *Computers* [online]. 2015, extbf54, 60-76 [cit. 2016-02-07]. DOI: 10.1016/j.cose.2015.06.002. ISSN 01674048.

[4] NATALIIA, Lukianova a Fell ELENA. Internet of Things as a Symbolic Resource of Power. *Procedia - Social and Behavioral Sciences* [online]. 2015, extbf166, 521-525 [cit. 2016-02-07]. DOI: 10.1016/j.sbspro.2014.12.565. ISSN 18770428.

[5] Recommendation ITU-T Y.2060: Overview of the Internet of things. *International Telecommunication Union* [online]. International Telecommunication Union, 2012 [cit. 2016-02-25]. Dostupné z: `http://www.itu.int/rec/T-REC-Y.2060-201206-I`

[6] CONTI, Mauro, Ali DEHGHANTANHA, Katrin FRANKE a Steve WATSON. Internet of Things security and forensics: Challenges and opportunities. *Future Generation Computer Systems* [online]. 2018, **78**, 544-546 [cit. 2018-02-20]. DOI: 10.1016/j.future.2017.07.060. ISSN 0167739x. Dostupné z: `http://linkinghub.elsevier.com/retrieve/pii/S0167739X17316667`

[7] DIERKS, T. a E. RESCORLA. RFC 5246: The Transport Layer Security (TLS) Protocol Version 1.2. *Standards Track* [online]. 2008 [cit. 2018-02-22]. Dostupné z: https://tools.ietf.org/html/rfc5246

[8] FREIER, A., P. KARLTON a P. KOCHER. RFC 6101: The Secure Sockets Layer (SSL) Protocol Version 3.0. *Netscape Communications* [online]. 2011 [cit. 2018-02-22]. ISSN: 2070-1721. Dostupné z: https://tools.ietf.org/html/rfc6101

[9] KENT, S. a R. ATKINSON. RFC 2401: Security Architecture for the Internet Protocol. *The Internet Society* [online]. 1998, 1998 [cit. 2018-02-22]. Dostupné z: https://tools.ietf.org/html/rfc2401

[10] Oficiální stránky Raspberry Pi: Teach, Learn and Make with Raspberry Pi. *Raspberrypi.org* [online]. Cambridge, United Kingdom: Raspberry Pi Foundation, 2018 [cit. 2018-06-19]. Dostupné z: https://www.raspberrypi.org

[11] Oficiální stránky Asus: Tinker Board. *Asus.com/cz/* [online]. Na Pankráci 127, Praha 4: ASUSTeK Computer, 2018 [cit. 2018-06-19]. Dostupné z: https://www.asus.com/cz/Single-Board-Computer/Tinker-Board/

[12] Oficiální stránky Orange Pi. *Orangepi.org* [online]. Guangdong, China: Xunlong Software CO, 2016 [cit. 2018-06-19]. Dostupné z: http://www.orangepi.org/OrangePiZeroPlus2/

[13] Oficiální stránky Nano Pi: FriendlyARM. *Friendlyarm* [online]. Guangzhou, China: Friendlyarm, 2015 [cit. 2018-06-19]. Dostupné z: http://nanopi.io

[14] Oficiální stránky Axagon: Počítačové a mobilní příslušenství AXAGON. *Axagon.eu* [online]. Pisárecká 11, Brno: RealQ, 2018 [cit. 2018-06-19]. Dostupné z: http://www.axagon.eu/produkty

[15] Oficiální stránky Cirrus: Cirrus Logic, Inc. *Cirrus.com* [online]. Texas, United States: Cirrus Logic, 2018 [cit. 2018-06-19]. Dostupné z: https://www.cirrus.com/

[16] Oficiální stránky SparqEE: SparqEE CELL. *Sparqee.com* [online]. Fullerton, CA, United States: SparqEE, LLC., 2015 [cit. 2018-06-19]. Dostupné z: http://www.sparqee.com/portfolio/sparqee-cell/

[17] Adafruit Power: Adafruit Industries, Unique & Fun. *Adafruit.com* [online]. New York City, United States: Adafruit, 2018 [cit. 2018-02-26]. Dostupné z: https://www.adafruit.com/category/44

[18] Witty Pi 2: Realtime clock and power management for Raspberry Pi. *UUgear.com* [online]. Praha: UUGear, 2018 [cit. 2018-02-26]. Dostupné z: http://www.uugear.com/product/wittypi2/

[19] Standard for an Architectural Framework for the Internet of Things (IoT): IEEE P2413. *Grouper.ieee.org: 2413* [online]. IEEE, 2011, September 2016 [cit. 2018-03-19]. Dostupné z: http://grouper.ieee.org/groups/2413/Intro-to-IEEE-P2413.pdf

[20] SUO, Hui, Jiafu WAN, Caifeng ZOU a Jianqi LIU. Security in the Internet of Things: A Review. *2012 International Conference on Computer Science and Electronics Engineering* [online]. IEEE, 2012, 2012, 648-651 [cit. 2018-02-20]. DOI: 10.1109/ICCSEE.2012.373. ISBN 978-0-7695-4647-6. Dostupné z: http://ieeexplore.ieee.org/document/6188257/

[21] CORSER, George a Jared BIELBY. IoT Security Best Practices: Webinar. *Internetinitiative.ieee.org* [online]. 2017, 27. září 2017 [cit. 2018-02-25]. Dostupné z https://internetinitiative.ieee.org/events/webinars/iot-security-best-practices

[22] DIOGENES, Yuri a Dominic BETTS. Internet of Things security best practices. *Microsoft Azure* [online]. Microsoft Corporation, 2018 [cit. 2018-02-24]. Dostupné z: https://docs.microsoft.com/en-us/azure/iot-suite/iot-security-best-practices

[23] AARTS, Maurice. Hardware Attacks Tamper Resistance, Tamper Response and Tamper Evidence. *Date of retrieval* [online]. TU/e - Eindhoven University of Technology, Eindhoven, NL, 2016, 1-12 [cit. 2018-03-20]. Dostupné z: http://maurice.aarts.info/papers/tamper_resistance_evidence.pdf

[24] WEINGART, Steve H. Physical Security Devices for Computer Subsystems: A Survey of Attacks and Defenses. *Cryptographic Hardware and Embedded Systems - CHES 2000* [online]. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, 2000-1-29, , 302-317 [cit. 2018-03-20]. Lecture Notes in Computer Science. DOI: 10.1007/3-540-44499-8_24. ISBN 978-3-540-41455-1. Dostupné z: http://link.springer.com/10.1007/3-540-44499-8_24

[25] Elk stack: Elastic. *Elastic.co* [online]. San Francisco, CA, United States: Elasticsearch, 2018 [cit. 2018-02-22]. Dostupné z: https://www.elastic.co/elk-stack

[26] Beats: Elastic. *Elastic.co* [online]. San Francisco, CA, United States: Elasticsearch, 2018 [cit. 2018-02-22]. Dostupné z: https://www.elastic.co/products/beats

[27] Amazon elasticsearch service: AWS amazon. *Aws.amazon.com* [online]. Washington, United States: Amazon Web Services, 2018 [cit. 2018-02-22]. Dostupné z: https://aws.amazon.com/elasticsearch-service/

[28] Splunk. *Splunk.com* [online]. San Francisco, CA, United States: Splunk, 2018 [cit. 2018-02-22]. Dostupné z: https://www.splunk.com/

[29] Loggly: Log Management. *Loggly.com* [online].San Francisco, CA, United States: Loggly, 2017 [cit. 2018-02-22]. Dostupné z: https://www.loggly.com/

[30] Graylog: Enterprise Log Management. *Graylog.org* [online]. Houston, United States: Graylog, 2017 [cit. 2018-02-22]. Dostupné z: https://www.graylog.org/

[31] Ansible: Simple IT Automation. *Ansible.com* [online]. Durham: Red Hat, 2017 [cit. 2018-02-22]. Dostupné z: https://www.ansible.com/

[32] Resin.io Documentation. *Resin.io* [online]. Seattle, United States: Resin.io, 2018 [cit. 2018-04-01]. Dostupné z: https://docs.resin.io/learn/welcome/introduction/

[33] Systemd: System and Service Manager. *Freedesktop.org* [online]. Lennart Poettering, Kay Sievers, Harald Hoyer, 2018 [cit. 2018-06-18]. Dostupné z: https://freedesktop.org/wiki/Software/systemd/

[34] DMCrypt: Linux kernel device-mapper crypto target. *Gitlab.com* [online]. San Francisco, CA, United States: Gitlab, 2011 [cit. 2018-06-18]. Dostupné z: https://gitlab.com/cryptsetup/cryptsetup/wikis/DMCrypt

[35] Cryptsetup: Open-source disk encryption. *Gitlab.com* [online]. San Francisco, CA, United States: Gitlab, 2011 [cit. 2018-06-18]. Dostupné z: https://gitlab.com/cryptsetup/cryptsetup/

[36] Arecord: Command-line sound recorder and player for ALSA soundcard driver. *Linux man page* [online]. United States: Paul Winkler, 1996 [cit. 2018-06-18]. Dostupné z: https://linux.die.net/man/1/arecord

[37] SoX: Sound eXchange, the Swiss Army knife of audio manipulation. *Linux man page* [online]. United States: Chris Bagwell, 1996 [cit. 2018-06-18]. Dostupné z: https://linux.die.net/man/1/rec

[38] XFS Papers and Documentation: Primary XFS Documentation. *Xfs.org* [online]. California, United States: Dave Chinner, 2013, 15 October 2015 [cit. 2018-03-26]. Dostupné z: http://xfs.org/index.php/XFS_Papers_and_Documentation

[39] Cython: C-Extensions for Python. *Cython.org* [online]. Robert Bradshaw, Stefan Behnel, et al., 2010 [cit. 2018-06-18]. Dostupné z: http://cython.org/

[40] Wipe: Securely erase files from magnetic media. *Linux man page* [online]. United States: Berke Durak, 1996 [cit. 2018-06-18]. Dostupné z: https://linux.die.net/man/1/wipe

[41] Docker: Build, Ship and Run Any App, Anywhere. *Docker.com* [online]. San Francisco, CA, United States: Docker, 2018 [cit. 2018-06-18]. Dostupné z: https://www.docker.com/

[42] SearchGuard: Security for Elasticsearch. *Search-guard.com* [online]. Berlin, Germany: Floragunn, 2018 [cit. 2018-06-18]. Dostupné z: „https://search-guard.com/“

[43] Curator. *Elastic.co* [online]. San Francisco, CA, United States: Elasticsearch, 2018 [cit. 2018-06-18]. Dostupné z: https://www.elastic.co/guide/en/elasticsearch/client/curator/current/index.html

[44] Cron: Daemon to execute scheduled commands. *Man7.org: Linux man-pages project* [online]. Munich: Michael Kerrisk, 2008, 26. září 2013 [cit. 2018-06-18]. Dostupné z: http://man7.org/linux/man-pages/man8/cron.8.html

[45] Dmesg: Print or control the kernel ring buffer. *Man7.org: Linux man-pages project* [online]. Munich: Michael Kerrisk, 2008, Červenec 2012 [cit. 2018-06-18]. Dostupné z: http://man7.org/linux/man-pages/man1/dmesg.1.html

Příloha **A**

# Fotografie zvukového senzoru


Obrázek A.1: Zvukový senzor v plastovém obalu bez vodiče ve stěnách.


Obrázek A.2: Zavřený obal zvukového senzoru.


Obrázek A.3: Přední a boční strana zvukového senzoru.


Obrázek A.4: Hardware zvukového senzoru.


Obrázek A.5: Detail vodiče ve stěnách plastové krabičky.

Příloha B

# Parametry jednodeskových počítačů

| | |
|---|---|
| Rozměry | 85,60 mm x 56,5 mm x 17 mm |
| Hmotnost | 45 g |
| Procesor | 64-bit quad-core ARM Cortex-A53, 1.2 GHz |
| GPU | Broadcom VideoCore IV |
| Paměť | 1 GB, sdílená s GPU, SDRAM |
| USB 2.0 porty | 4 přes zabudovaný USB hub |
| Video vstup | 15-pinový MIPI konektor CSI |
| Zvukový vstup | I2C rozhraní |
| Zvukový výstup | Analogový pomocí 3,5 mm jack, digitální přes HDMI |
| Interní paměť | MicroSDHC, USB Boot Mode |
| Integrovaná síť | 10/100 Mbit/s Ethernet, Wi-Fi 802.11n a Bluetooth 4.1 |
| Jmenovitý výkon | 300 mA (1,5 W) v průměru |

Tabulka B.1: Parametry Raspberry Pi 3 model B[65].

[65]Převzato z: http://rpishop.cz/raspberry-pi-3b/283-raspberry-pi-3-model-b-64-bit.html

[66]Převzato z: http://rpishop.cz/raspberry-pi-zero/647-raspberry-pi-zero-w.html

[67]Převzato z: https://www.asus.com/cz/Single-Board-Computer/Tinker-Board/

[68]Převzato z: http://www.orangepi.org/OrangePiZeroPlus2/

[69]Převzato z: http://nanopi.io/nanopi2-fire.html

[70]Převzato z: http://nanopi.io/nanopi-neo-air.html

| | |
|---|---|
| Rozměry | 65 mm x 30 mm x 5 mm |
| Hmotnost | 9 g |
| Procesor | Single-core ARM1176JZF-S, 1 GHz |
| GPU | Broadcom VideoCore IV |
| Paměť | 512 MB, sdílená s GPU, SDRAM |
| USB 2.0 porty | micro-USB |
| Video vstup | MIPI kamerové rozhran |
| Zvukový vstup | I2C rozhraní |
| Zvukový výstup | Mini-HDMI, stereo audio skrz PWM na GPIO |
| Interní paměť | MicroSDHC |
| Integrovaná síť | 802.11n wireless, Bluetooth 4.1 |
| Jmenovitý výkon | 100 mA (0.5 W) v průměru |

Tabulka B.2: Parametry Raspberry Pi Zero W[66].

| | |
|---|---|
| Rozměry | 85,55 mm x 54 mm x 17 mm |
| Hmotnost | 55 g |
| Procesor | 32-bit Quad-Core RK3288, 1.8 GHz |
| GPU | Mali-T764 |
| Paměť | 2 GB, DDR3 |
| USB 2.0 porty | 4 přes zabudovaný USB hub |
| Video vstup | MIPI-CSI |
| Zvukový vstup | I2C rozhraní |
| Zvukový výstup | Analogový pomocí 3,5 mm jack |
| Interní paměť | MicroSDHC |
| Integrovaná síť | 1 GB Ethernet, Wi-Fi, Bluetooth V4.0, 802.11 b/g/n |
| Jmenovitý výkon | 2 W v průměru |

Tabulka B.3: Parametry Asus Tinker Board[67].

| | |
|---|---|
| Rozměry | 48 mm x 46 mm x 10 mm |
| Hmotnost | 20 g |
| Procesor | Quad core Cortex A7 processor, 1,2 GHz |
| GPU | Mali-400MP2 |
| Paměť | 512 MB (DDR3) |
| USB porty | SB 2.0 HOST, micro USB OTG port |
| Video vstup | MIPI-CSI |
| Zvukový výstup | HDMI |
| Interní paměť | MicroSDHC, 8 GB eMMC |
| Integrovaná síť | 802.11 b/g/n Wi-Fi, Bluetooth 4.0 LE |
| Jmenovitý výkon | 1 W v průměru |

Tabulka B.4: Parametry Orange Pi Zero Plus 2[68].

| | |
|---|---|
| Rozměry | 75 mm x 40 mm x 17 mm |
| Hmotnost | 22 g |
| Procesor | Samsung S5P 4418 Cortex-A9, 1.4 GHz |
| GPU | Mali-400MP2 |
| Paměť | 1 GB 32bit DDR3 |
| USB porty | USB 2.0 Type A |
| Video vstup | MIPI-CSI |
| Zvukový výstup | HDMI |
| Interní paměť | microSD |
| Integrovaná síť | 1 Gbps Ethernet |
| Jmenovitý výkon | 1,5 W v průměru |

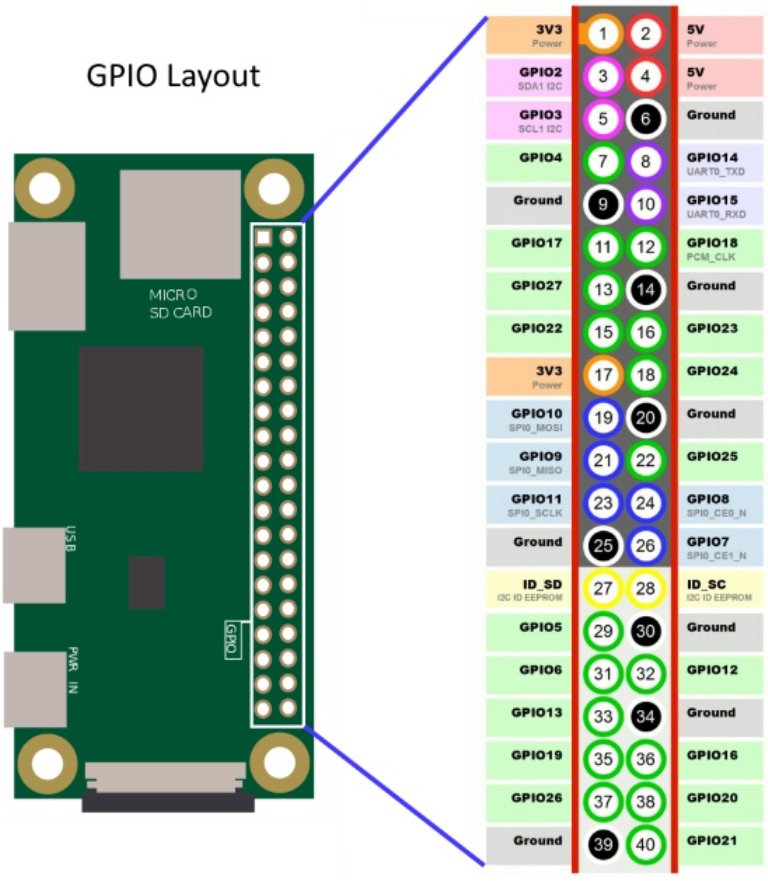
Tabulka B.5: Parametry NanoPi 2 Fire[69].

| | |
|---|---|
| Rozměry | 40 mm x 40 mm x 5 mm |
| Hmotnost | 10 g |
| Procesor | Quad-core Cortex-A7, 1.2 GHz |
| GPU | Mali-400MP2, 600 MHz |
| Paměť | 16-bit 512 MB DDR3 |
| USB porty | micro USB OTG port |
| Interní paměť | microSD, 8GB eMMC |
| Integrovaná síť | Wi-Fi 802.11b/g/n, Bluetooth 4.0 |
| Jmenovitý výkon | 1 W v průměru |

Tabulka B.6: Parametry Nano Pi Neo Air[70].

Příloha C

# Použité GPIO piny




Obrázek C.1: Raspberry Zero GPIO layout[71].

[71] Převzato z: `https://www.raspberrypi-spy.co.uk`

**Audio Injector Zero využité piny**[72]

- PIN 3 : I2C DATA
- PIN 5: I2C CLK
- PIN 12 : BIT CLOCK
- PIN 35: LR CLOCK
- PIN 38: DATA IN
- PIN 40: DATA OUT
- PIN 2 : 5 V
- PIN 4 : 5 V
- PIN 17 : 3,3 V

**PiZ-UpTime využité piny**[73]

- PIN 2 : 5 V
- PIN 4 : 5 V
- PIN 26 : LOW BATTERY

**Modul světelné čidlo s fotodiodou**

- PIN 17 : 3,3 V
- PIN 6 : GND
- PIN 31 : DATA IN

**Detekce změny zdroje napájení**

- PIN 30 : GND
- PIN 32 : DATA IN

**Vodič namotaný ve stěnách plastové krabičky**

- PIN 1 : 3,3 V
- PIN 36 : DATA IN

[72] Převzato z: `http://forum.audioinjector.net`
[73] Převzato z: `http://alchemy-power.com/piz-uptime/`

Příloha D

# Hardwarové části zvukového senzoru s cenou

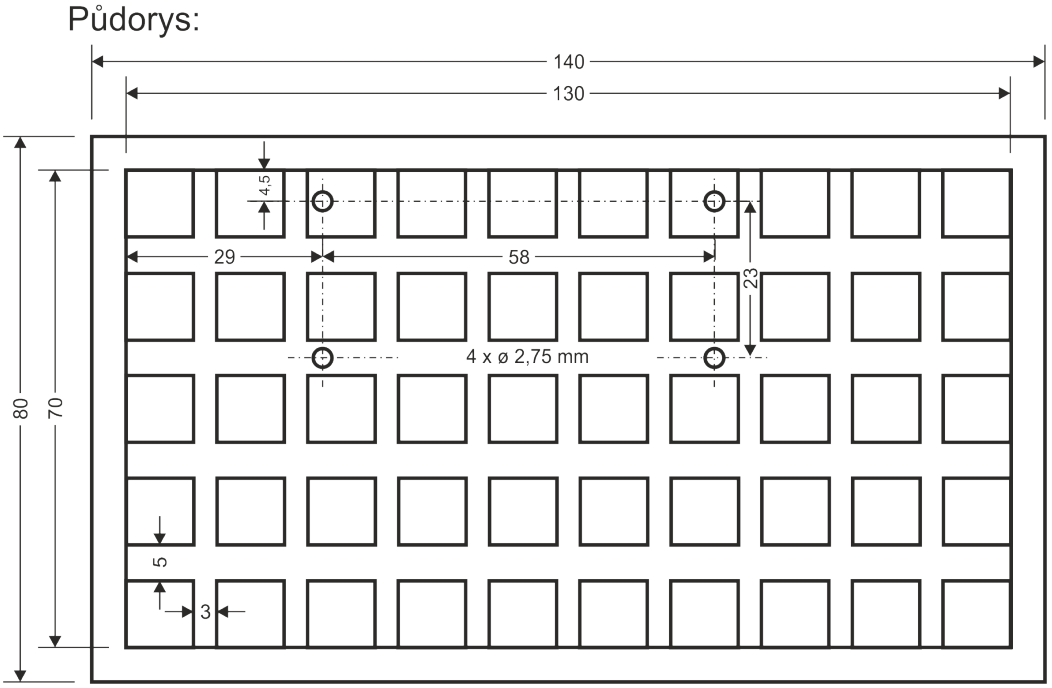
| Název hardwaru | Cena |
|---|---|
| Osazené Raspberry Pi Zero W | 437 Kč |
| Audio Injector Zero | 549 Kč |
| PiZ-UpTime | 699 Kč |
| Huawei E3372h | 1109 Kč |
| Kingston 16 GB microSDHC | 239 Kč |
| Baterie 2800 mAh | 80 Kč |
| Modul světelné čidla s fotodiodou | 79 Kč |
| USB Adapter | 70 Kč |
| Napájecí zdroj | 239 Kč |
| Plastová krabička | - |
| USB prodlužovací kabel | 51 Kč |
| SIM karta | - |
| Kabeláž a pájení | - |
| **Celkem** | **3552 Kč** |

Tabulka D.1: Cena jednotlivých částí zvukového senzoru a jejich suma.

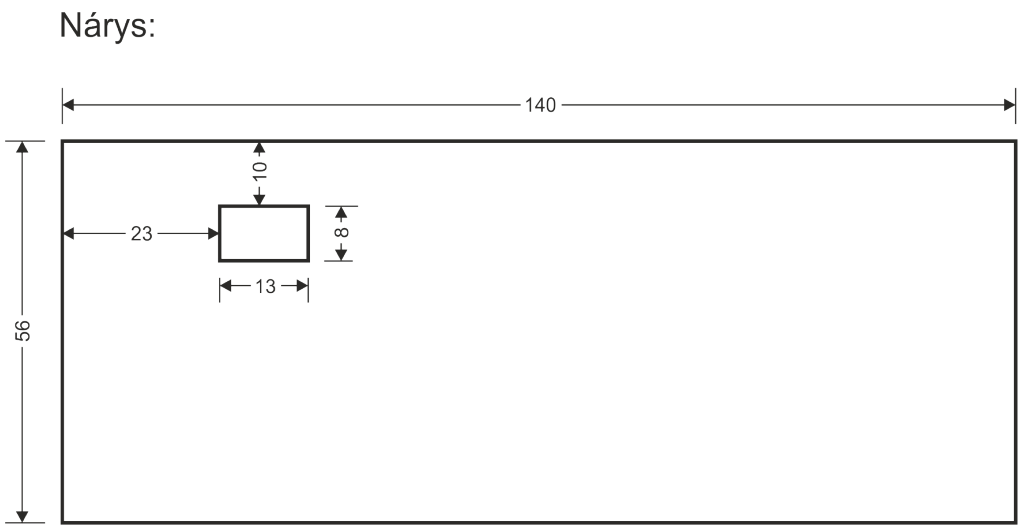
U položek s vynechanou částkou je velmi obtížné danou cenu jednoznačně stanovit. U 3D tisku záleží na zvoleném plastu, nastavené hrubosti jednoho kroku tisku, a tím pádem i čas tisku. Cena SIM karty se odvíjí od toho, pro jakou lokalitu bude daná karta pořízena (rozdílné ceny např. v České Republice a Slovensku). Pro celkové sestavení je také nutné pájení komponent (např. sestavení modulu Audio Injector Zero) a propojovací kabeláž (např. pro připojení modulu světelného čidla s fotodiodou k Raspberry Zero).

Příloha **E**

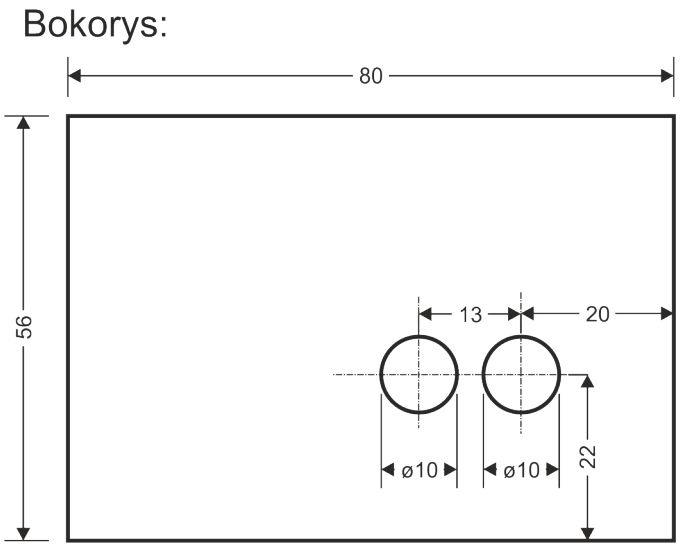
# Nákres plastové krabičky



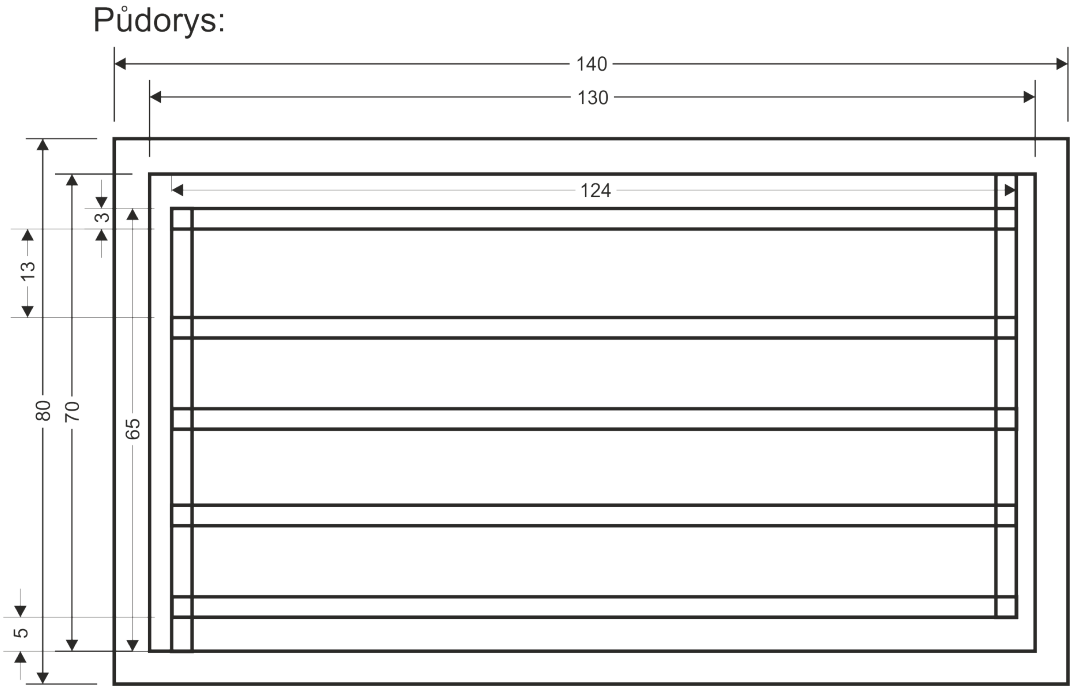

Obrázek E.1: Půdorys plastové krabičky.



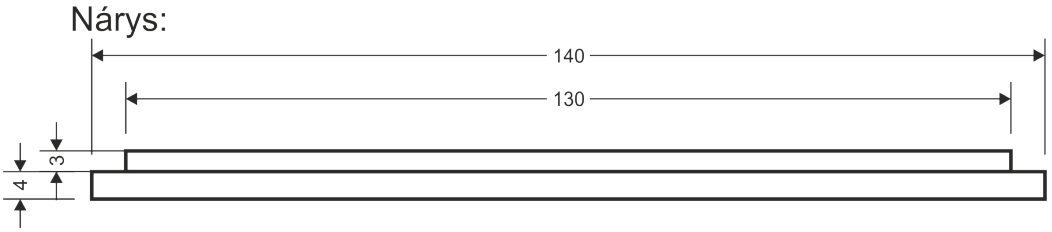

Obrázek E.2: Nárys plastové krabičky.



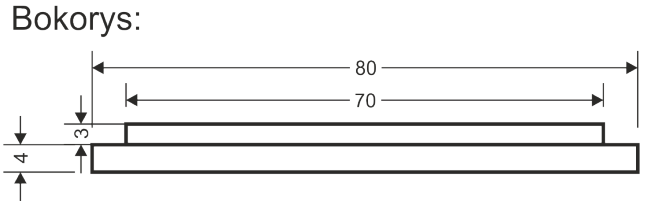

Obrázek E.3: Bokorys plastové krabičky.



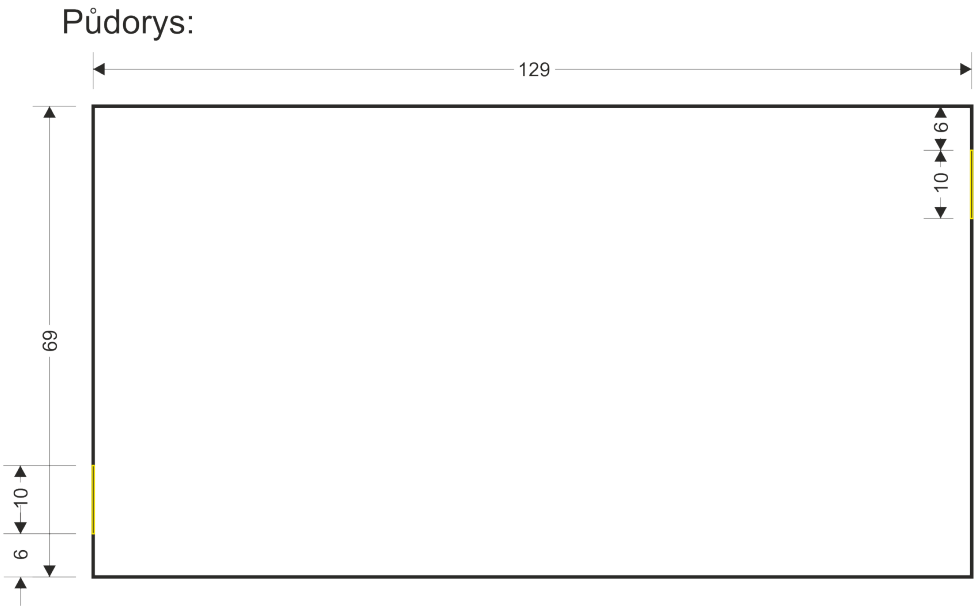

Obrázek E.4: Půdorys víka plastové krabičky.



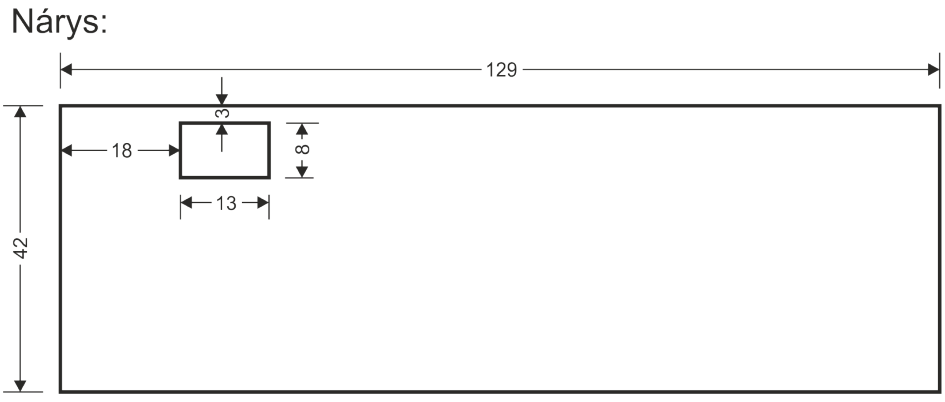

Obrázek E.5: Nárys víka plastové krabičky.




Obrázek E.6: Bokorys víka plastové krabičky.

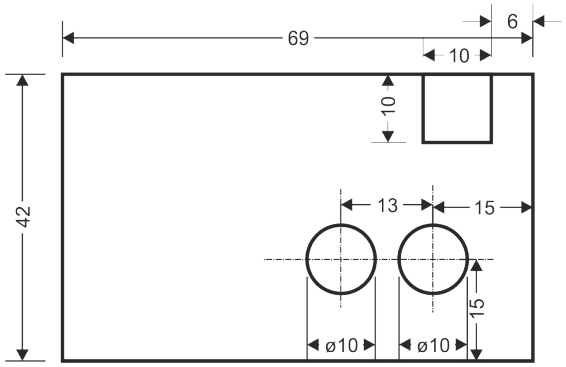



Obrázek E.7: Půdorys vložky plastové krabičky.




Obrázek E.8: Nárys vložky plastové krabičky.

Bokorys:




Obrázek E.9: Bokorys vložky plastové krabičky.

Příloha F

# Oficiální specifikace NB.6

# NeuronBox NB.6 IoT Device

Industrial grade edge computing device equipped with the audio digitalisation capabilities


## The Key Product Features:

- Standalone analytical IoT unit with variable acoustics sensors
- Up to 6 channel simultaneous and synchronous recording
- Edge computing software for the audio analysis based on neural networks
- Installation directly next to machine or into DIN ledge
- Records directly interpreted, transferred to cloud or store microSD on device
- Multiple audio output file types supported (WAV, OGG, FLAC)
- LAN/Wifi/LTE connectivity and Power supply (optional)

### Product Description:

| | |
|---|---|
| Sound converter: | Up to 6-channel ultrasonic analog to digital converter |
| Device design: | ALU chasi / stainless steel |
| Sensor type: | Broad range of sensors |
| Connectivity: | Wi-Fi , 3G/4G, Ethernet |
| Amplifier: | Digitally controlled gain in range from 0 to + 60dB |
| LED statuses: | LED signals status for each channel |

### Product specifications:

| | |
|---|---|
| Dimensions: | 165 x 145 x 55 mm (W x D x H) |
| Weight: | 1200g without power supply |
| CPU: | 4×ARM Cortex, 1.2GHz |
| RAM: | 1 GB |
| Storage: | 64GB - 256GB (microSD card) |
| USB output: | 2x USB output |
| Power supply: | 12-18V DC with 2.5A load |
| Power consumption: | min. 660mAh DC |
| Operating temperatures: | -25°C ... +70°C |

### Sensor types:

| | |
|---|---|
| Audio microphones: | All types including condenser (+48V phantom powered) |
| Piezo based microphones with amplifier: | Frequency range 7 Hz - 30kHz ( -20°C ... +70°C) |
| Ultrasonic sensors: | Frequency range up-to 100 kHz (-25°C ... +70°C) |
| MEMs accelerometers: | Dynamic range from ±1g to ±100g (-40°C ... + 130°C) |

## Contact us:

Jiří Čermák
+420 603 884 011
jiri.cermak@neuronsw.com

**NeuronSW LTD**
71-75 Shelton Street, Covent Garden, London
United Kingdom
www.neuronsw.com

Příloha G

# Obsah přiloženého CD

- `readme.txt` ..... Popis obsahu CD.
- `DP_Fuchs_Ondřej_2018.pdf` ..... Text práce ve formátu PDF.
- `PraceLatex.zip` ..... Soubory potřebné k vysázení diplomové práce.
- `Zdroje` ..... Zdroje dostupné ve formátu PDF.
- `Konfigurace` ..... Skripty napsané pro potřebu diplomové práce.
  - `crypto` .. Složka se skripty pro šifrování a dešifrování diskové jednotky.
  - `debPackage` ..... Složka se skripty pro vytvořené Debianích balíčků.
  - `filebeat` ..... Složka se skripty pro nastavení sběru dat.
  - `GPIO` ..... Složka se skripty pro detekci manipulace se zařízením.
  - `record` .. Složka se skriptem pro záznam zvuku a nahrávání do cloudu.
  - `setupOS` Složka se skripty pro nastavení zařízení před prvním startem.
- `ELK Stack` ..... Složka se soubory k nastavení ELK Stacku.
- `Ansible` ..... Soubory pro orchestraci zařízení pomocí Ansible.
  - `roles` ..... Složka obsahující všechny Ansible Role použité v práci.
  - `inventories` ..... Složka obsahující seznam konfigurovaných zařízení.
  - `default.yml` ..... První playbook pro konfiguraci zařízení.
  - `nsw.yml` ..... Druhý playbook pro konfiguraci zařízení.
- `Plastová krabička` .... Soubory potřebné pro 3D tisk krabičky zařízení.
  - `Zero.stl` ..... Návrh plastové krabičky ve formátu STL.
  - `Zero.zip` .... Návrh plastové krabičky ve formátu OBJ v archivu ZIP.