

Katedra informatiky  
Přírodovědecká fakulta  
Univerzita Palackého v Olomouci

# DIPLOMOVÁ PRÁCE

Měření průtoku proudu pomocí Raspberry Pi



2023

Vedoucí práce:  
RNDr. Eduard Bartl, PhD.

Bc. Martin Urbanec

Studijní program: Aplikovaná informatika,  
Specializace: Počítačové systémy a  
technologie

## **Bibliografické údaje**

Autor: Bc. Martin Urbanec  
Název práce: Měření průtoku proudu pomocí Raspberry Pi  
Typ práce: diplomová práce  
Pracoviště: Katedra informatiky, Přírodovědecká fakulta, Univerzita Palackého v Olomouci  
Rok obhajoby: 2023  
Studijní program: Aplikovaná informatika, Specializace: Počítačové systémy a technologie  
Vedoucí práce: RNDr. Eduard Bartl, PhD.  
Počet stran: 63  
Přílohy: 1 elektronická příloha  
Jazyk práce: český

## **Bibliographic info**

Author: Bc. Martin Urbanec  
Title: Current measurement using a Raspberry Pi  
Thesis type: master thesis  
Department: Department of Computer Science, Faculty of Science, Palacký University Olomouc  
Year of defense: 2023  
Study program: Applied Computer Science, Specialization: Computer Systems and Technologies  
Supervisor: RNDr. Eduard Bartl, PhD.  
Page count: 63  
Supplements: 1 electronic attachment  
Thesis language: Czech

## Anotace

*Cílem práce je sestavení zařízení na platformě Raspberry Pi pro firmu Edhouse. Zařízení bude umožňovat měření aktuálního průtoku proudu a současně bude nasbíraná data ukládat do databáze InfluxDB. Tato data si následně bude moci uživatel zobrazit pomocí webové aplikace Grafana. Práce se rovněž zabývá možnostmi fyzického zabezpečení datových center.*

## Synopsis

*The aim of the work is to build a Raspberry Pi based device for Edhouse. The device will allow the measurement of the current flow rate and at the same time will store the collected data in the InfluxDB database. This data can then be viewed by the user using the Grafana web application. The thesis also addresses the possibilities of physical security of data centres.*

**Klíčová slova:** měření proudu; Raspberry Pi; Grafana; InfluxDB; Docker; Python; serverovna; webová aplikace; webový server; databáze

**Keywords:** current measurement; Raspberry Pi; Grafana; InfluxDB; Docker; Python; server room; web application; web server; database

Mé poděkování patří RNDr. Eduardu Bartlovi, PhD. za odborné vedení této diplomové práce. Dále bych chtěl poděkovat Mgr. Pavlu Stržínkovi, Ing. Janu Rýcovi a Ing. Vladimíru Babicovi z firmy Edhouse za ochotnou pomoc a technické rady v rámci praktické části. V neposlední řadě děkuji i své přítelkyni a rodině za podporu při psaní této práce.

*Místopřísežně prohlašuji, že jsem celou práci včetně příloh vypracoval/a samostatně a za použití pouze zdrojů citovaných v textu práce a uvedených v seznamu literatury.*

datum odevzdání práce

podpis autora



# Obsah

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Úvod</b>                                  | <b>9</b>  |
| <b>2</b> | <b>Fyzické zabezpečení datových center</b>   | <b>10</b> |
| 2.1      | Standard Tier                                | 10        |
| 2.1.1    | Tier 1                                       | 10        |
| 2.1.2    | Tier 2                                       | 11        |
| 2.1.3    | Tier 3                                       | 11        |
| 2.1.4    | Tier 4                                       | 11        |
| 2.2      | Vzduchotechnika                              | 12        |
| 2.2.1    | System zdvojené podlahy                      | 12        |
| 2.2.2    | Stropní rozvod vzduchu                       | 13        |
| 2.2.3    | Horká a chladná ulička                       | 14        |
| 2.3      | Ochrana před neoprávněným vstupem            | 15        |
| 2.4      | Ochrana před požárem                         | 16        |
| 2.5      | Zálohování dat                               | 16        |
| 2.6      | Další možnosti fyzické bezpečnosti           | 17        |
| 2.7      | Projekt FENIX                                | 18        |
| <b>3</b> | <b>Zabezpečení datových center v praxi</b>   | <b>19</b> |
| 3.1      | Datové centrum společnosti Internext         | 19        |
| 3.1.1    | Energetické vedení                           | 19        |
| 3.1.2    | Vzduchotechnika                              | 20        |
| 3.1.3    | Požární ochrana                              | 21        |
| 3.1.4    | Ochrana                                      | 21        |
| 3.2      | Serverovna Edhouse                           | 21        |
| <b>4</b> | <b>Raspberry Pi</b>                          | <b>23</b> |
| <b>5</b> | <b>Fyzické sestavení a zapojení zařízení</b> | <b>25</b> |
| 5.1      | Princip fungování zařízení                   | 25        |
| 5.2      | Senzory                                      | 25        |
| 5.2.1    | Proudový senzor ACS712                       | 26        |
| 5.2.2    | ADC převodník ADS1115                        | 26        |
| 5.3      | Zapojení senzorů do Raspberry Pi             | 27        |
| 5.4      | Kalibrace senzorů                            | 28        |
| 5.5      | Připojení relé                               | 32        |
| 5.6      | LED diody                                    | 32        |
| 5.7      | Zapojení pro ověření funkčnosti              | 33        |
| <b>6</b> | <b>Softwarová část zařízení</b>              | <b>34</b> |
| 6.1      | Aktivace potřebných rozhraní na GPIO         | 34        |
| 6.2      | Hlavní skript pro zpracování dat             | 35        |
| 6.2.1    | Inicializace programu                        | 35        |

|          |  |           |
|----------|--|-----------|
| 6.2.2    | Měření proudu a sběr dat . . . . .               | 35        |
| 6.2.3    | Výpočet průměru . . . . .                        | 37        |
| 6.3      | Webový server Flask . . . . .                    | 38        |
| 6.3.1    | Obsah webové aplikace . . . . .                  | 39        |
| 6.3.2    | Použití lokální databáze . . . . .               | 40        |
| 6.3.3    | Odesílání e-mailových notifikací . . . . .       | 41        |
| 6.3.4    | Vzhled webového rozhraní . . . . .               | 43        |
| <b>7</b> | <b>Vyhodnocení dat</b>                           | <b>44</b> |
| 7.1      | Databáze InfluxDB . . . . .                      | 44        |
| 7.1.1    | Nastavení InfluxDB . . . . .                     | 44        |
| 7.2      | Webová aplikace Grafana . . . . .                | 45        |
| 7.2.1    | Zdroj dat . . . . .                              | 46        |
| 7.2.2    | Vytvoření grafů . . . . .                        | 46        |
|          | <b>Závěr</b>                                     | <b>48</b> |
|          | <b>Conclusions</b>                               | <b>49</b> |
|          | <b>A Fotografie z datového centra Internextu</b> | <b>50</b> |
|          | <b>B Fotografie ze serverovny Edhouse</b>        | <b>53</b> |
|          | <b>C Výsledné zařízení ECMD</b>                  | <b>56</b> |
|          | <b>D Použití s Docker Desktop pro Windows</b>    | <b>58</b> |
|          | D.1 Vytvoření síťového adaptéru . . . . .        | 58        |
|          | D.2 Stažení image . . . . .                      | 58        |
|          | D.3 Docker compose . . . . .                     | 58        |
|          | <b>E Obsah elektronické přílohy</b>              | <b>62</b> |
|          | <b>Literatura</b>                                | <b>63</b> |

## Seznam obrázků

|    |  |    |
|----|--|----|
| 1  | Systém zdvojené podlahy – převzato z článku HVAC Cooling Systems for Data Centers, str. 49 . . . . . | 13 |
| 2  | Stropní rozvod vzduchu – převzato z článku HVAC Cooling Systems for Data Centers, str. 50 . . . . .  | 14 |
| 3  | Horká/studená ulička – převzato z článku HVAC Cooling Systems for Data Centers, str. 53 . . . . .    | 15 |
| 4  | Psychrometrický diagram . . . . .  | 20 |
| 5  | Raspberry Pi verze 4 – převzato z Wikipedia EN; Raspberry Pi . . . . .                               | 24 |
| 6  | Princip fungování zařízení . . . . .   | 25 |
| 7  | Proudový senzor ACS712 – převzato z LáškaKit.cz . . . . .  | 26 |
| 8  | ADC převodník ADS1115 – převzato z LáškaKit.cz . . . . .   | 27 |
| 9  | Zapojení senzorů do Raspberry Pi . . . . .   | 28 |
| 10 | Kalibrace senzorů . . . . .  | 31 |
| 11 | Relé modul do 30 A – převzato z LáškaKit.cz . . . . .  | 32 |
| 12 | Zapojení komponent pro ověření funkčnosti . . . . .  | 33 |
| 13 | Detekce připojeného senzoru na I2C . . . . .   | 34 |
| 14 | Hlavní stránka webového dashboardu . . . . .   | 43 |
| 15 | Grafana – dashboard s grafy . . . . .  | 47 |
| 16 | Energetické vedení . . . . .   | 50 |
| 17 | Vstup do horké uličky . . . . .  | 51 |
| 18 | Hasicí tlaková nádoba . . . . .  | 52 |
| 19 | Servery pro provoz interních systémů a síťové prvky . . . . .  | 53 |
| 20 | Rozvod připojení k síti . . . . .  | 54 |
| 21 | Napájení serverů . . . . .   | 55 |
| 22 | Návrh krabičky – vnitřní část . . . . .  | 56 |
| 23 | Návrh krabičky – vnější část . . . . .   | 56 |
| 24 | Výsledné zařízení – vnější pohled . . . . .  | 57 |
| 25 | Výsledné zařízení – vnitřní pohled . . . . .   | 57 |
| 26 | Spuštění CLI přes Docker UI verze 4.8.x a nižší . . . . .  | 60 |
| 27 | Spuštění CLI přes Docker UI od verze 4.14.x . . . . .  | 60 |

## Seznam tabulek

|   |  |    |
|---|--|----|
| 1 | Kalibrace senzoru – naměřené hodnoty . . . . . | 29 |
| 2 | Kalibrace senzoru – porovnání hodnot . . . . . | 30 |

## Seznam zdrojových kódů

|   |  |    |
|---|--|----|
| 1 | Vytvoření vláken pro webový server a měření proudu . . . . . | 35 |
| 2 | Mezivýpočet – průměrná hodnota napětí . . . . .              | 36 |
| 3 | Výpočet finální hodnoty proudu . . . . .                     | 36 |

|    |   |    |
|----|---|----|
| 4  | Připojení na databázi a odeslání dat . . . . .                | 37 |
| 5  | Výpočet průměrné hodnoty proudu . . . . .                     | 37 |
| 6  | Spuštění webového serveru pomocí Geventu . . . . .            | 38 |
| 7  | Nastavení port forwardingu . . . . .                          | 39 |
| 8  | Ukázka JavaScriptové funkce pro obnovení průměrné hodnoty . . | 40 |
| 9  | Inicializace lokální SQLite databáze . . . . .                | 40 |
| 10 | Aktualizace údajů v SQLite databázi . . . . .                 | 41 |
| 11 | Nastavení SMTP serveru a odeslání zprávy . . . . .            | 42 |
| 12 | Připojení na SMTP server s ověřováním . . . . .               | 42 |
| 13 | Prvotní nastavení databáze InfluxDB . . . . .                 | 45 |
| 14 | Povolení autentizace v InfluxDB . . . . .                     | 45 |
| 15 | Nastavení síťového adaptéru v Dockeru . . . . .               | 58 |
| 16 | Stahování image pro Docker . . . . .                          | 58 |
| 17 | Spuštění CLI InfluxDB . . . . .                               | 59 |
| 18 | Zapnutí autentizace pro InfluxDB . . . . .                    | 59 |
| 19 | Konfigurační soubor pro docker compose ve formátu YAML . . .  | 61 |

# 1 Úvod

Cílem práce je vytvoření zařízení na platformě Raspberry Pi, které bude sloužit pro měření průtoku proudu v serverovně firmy Edhouse. Zařízení bude sledovat zátěž jednotlivých napájecích větví pomocí externího senzoru ACS712 pro měření proudu.

Na základě několika výpadků se rozhodlo, že je potřeba zajistit, aby se nepřetěžovaly jističe v rackových skříních. Nicméně v současné době serverovna nedisponuje žádným zařízením pro monitoring zátěže. Vytížení sítě lze tedy jen těžko predikovat, protože servery mají v různou dobu různý odběr odvíjející se od jejich zátěže. Toto zařízení umožní správcům IT nahlédnout do historie zátěže a z nasbíraných dat bude do jisté míry možné zátěž predikovat. Podle toho by se mohlo přistoupit k přepojení jednotlivých serverů do jiných jističů, čímž by se zátěž optimálně rozložila a nedocházelo by k případnému přetížení jističe a jeho následnému výpadku. Dále bude možné nastavit konkrétní hodnotu a pokud by byla tato hodnota překonána, byla by odeslána automatická e-mailová notifikace správcům IT.

Výhodou zařízení na platformě Raspberry Pi je především jeho nižší pořizovací cena a možnost implementace přímo na míru firmě Edhouse. Zařízení bude využívat výše uvedený senzor pracující na principu Hallova jevu. Signál bude naměřen v analogové formě a následně bude převeden pomocí 4-kanálového I2C ADC převodníku ADS1115 do digitální formy, která umožní širší rozsahy signálu.

Výsledné zařízení bude umožňovat následující funkce:

- měření aktuálního průtoku proudu,
- zaznamenání nasbíraných dat do databáze,
- vyobrazení nasbíraných dat v přehledných grafech s možností filtrování,
- vzdálené ovládání serveru (vypnutí, popřípadě zapnutí).

Původní návrh počítal s tzv. all-in-one řešením, nicméně z důvodu, že veškerá činnost Raspberry Pi by byla ukládána na SD kartu, čímž by při tak velkém množství zápisů byla podstatně zkrácena její životnost, jsem se rozhodl umístit InfluxDB a Grafanu na externí server. Na Raspberry Pi tak poběží pouze „light-weight“ Python webový server, který bude sloužit pro provoz webové stránky, tzv. dashboardu. Na tomto dashboardu budou zobrazeny různé informace o zařízení jako jeho stav, průměrná hodnota proudu, nastavení e-mailových notifikací a bude zde i možnost vypnout/zapnout měřené zařízení.

Jelikož se práce zabývá sestavením zařízení, které bude přispívat k fyzickému zabezpečení serverovny firmy Edhouse, rozhodl jsem se v následujících dvou kapitolách popsat fyzické zabezpečení v datových centrech v obecné rovině a poté nabízím i pohled na fyzickou bezpečnost v praxi.

## 2 Fyzické zabezpečení datových center

Zabezpečením datového centra máme na mysli fyzickou bezpečnost, nikoliv tedy ochrana dat a služeb před výpadkem například po DDoS<sup>1</sup> útoku, ale ochrana před fyzickým útokem, výpadkem elektrické energie, přírodním živlům nebo poruchám jednotlivých komponent.

### 2.1 Standard Tier

Jedná se o certifikaci (standard), která má čtyři úrovně. Každá z úrovní určuje, jaké nároky je potřeba splnit při provozu datového centra. Tato certifikace pak může být rozhodující pro zákazníka, který hledá vysokou spolehlivost a výkon. Rozhodujícími faktory pro udělení certifikace jsou:

- Záruky dostupnosti služeb.
- Úroveň redundance systémů, stav chladicí a energetické infrastruktury.
- Odborná způsobilost zaměstnanců.
- Zabezpečení datového centra.

*Níže uvedené požadavky jednotlivých úrovní jsou parafrázovány z článku Andreje Velimirovic „Data Center Tiers Explained“, který je uveden v sekci použité literatury.*

#### 2.1.1 Tier 1

Úrovní Tier 1 disponují datová centra, která mají pouze jedinou distribuční větev pro napájení a chlazení bez jakékoliv redundance. V případě výpadku nebo pravidelného servisního zásahu je nutné celý provoz centra vypnout. Úroveň Tier 1 definuje záruku dostupnosti 99,671 % ročně. Z toho vyplývá, že zákazníci datového centra úrovně Tier 1 mohou očekávat maximálně 28,8 hodin výpadku ročně.<sup>2</sup> Technické požadavky Tier 1 jsou:

- Zdroj napájení (UPS<sup>3</sup>) pro krátké výpadky a provoz ve špičce.
- Vyhrazený prostor pro IT systémy.
- Motor-generátor.
- Chlazení, které běží nepřetržitě i mimo pracovní dny.

<sup>1</sup>Denial of Service, tedy odepření služby, je útok, který má za cíl znepřístupnit danou službu ostatním uživatelům. Například útok na server, kdy na něj útočník zasílá obrovské množství požadavků v krátkém čase. Server takové množství požadavků nezvládne zpracovat a v nejhorším případě dojde k pádu celého systému. Následně je znemožněn přístup i ostatním uživatelům.

<sup>2</sup>Tato hodnota odpovídá 0,329 % z počtu hodin celého roku.

<sup>3</sup>Uninterruptible Power Supply/Source – Zdroj nepřerušovaného napájení

### 2.1.2 Tier 2

Vše co požaduje úroveň Tier 1 je obsaženo i v úrovni Tier 2. Tato úroveň přidává nutnost mít některé systémy redundantní. Avšak napájení a chlazení mají pořád jedinou distribuční větev a pravidelná servisní údržba nebo výpadek proudu může vyústit v úplnou nedostupnost služeb datového centra. Tier 2 definuje záruku dostupnosti 99,741 % ročně, což odpovídá výpadku až 22 hodin za rok. Požadavky Tier 2 jsou:

- Více motor-generátorů.
- Univerzální bateriová úložiště.
- Zdvojené podlahy.
- UPS moduly.
- Redundantní chladicí jednotky.

### 2.1.3 Tier 3

Úroveň Tier 3 již přináší změnu ohledně distribuční větve. V této úrovni je nutné mít minimálně jednu záložní distribuční větev. V případě výpadku jedné větve je přepnuto na větev záložní a nedochází k výpadkům služeb datového centra. A současně při plánované údržbě nedochází k úplnému odstavení datového centra. Tato úroveň vyžaduje veškerou infrastrukturu, kterou vyžadují předchozí úrovně. Zde je však nutné dbát na pravidlo  $N+1$ .  $N$  zde označuje nezbytnou kapacitu pro podporu plného zatížení serverů a  $+1$  nám udává, že každá komponenta musí být redundantní. Toto pravidlo tedy zajišťuje, že pokud dojde k poruše na primárním zařízení, je ihned uvedeno do provozu zařízení sekundární a nedochází k žádnému výpadku. V případě rozsáhlejšího výpadku elektrické energie je požadováno, aby bylo datové centrum v provozu nejméně 72 hodin po výpadku. Tier 3 definuje záruku dostupnosti služeb 99,982 %, což odpovídá 1,6 hodiny výpadku ročně.

V ČR splňuje tuto úroveň pouze několik datových center.<sup>4</sup>

### 2.1.4 Tier 4

Datová centra úrovně Tier 4 poskytují nevyšší spolehlivost a výkon a oproti předchozí úrovni přidávají postupy odolnosti proti chybám. Tato datová centra mají fyzicky oddělené systémy, které jako celky fungují jako redundantní komponenty a distribuční cesty. Tier 4 definuje záruku dostupnosti služeb 99,995 % ročně, což odpovídá pouhým 26,3 minutám ročně. Požadavky pro Tier 4 jsou:

- Všechny komponenty mají podporu dvou generátorů, dvou UPS a dvou chladících systémů.

---

<sup>4</sup>Jejich přehled je k dispozici na adrese <https://uptimeinstitute.com/tier-certification/tier-certification-list>.

- Každá distribuční cesta je nezávislá.
- Provoz pokračuje minimálně 96 hodin po rozsáhlejších výpadku proudu.
- Napájecí zdroj se nepřipojuje k žádnému externímu zdroji.

Datová centra na úrovni Tier 4 mají redundanci  $2N + 1$ , přičemž  $2N$  znamená  $N + N$ , tj. zařízení má zcela zrcadlený nezávislý systém v pohotovostním režimu a současně další záložní komponentu, pokud by došlo i k výpadku sekundárního systému.

## 2.2 Vzduchotechnika

Dobrá vzduchotechnika v datovém centru je jednou z hlavních věcí. Každý server je nutné při jeho provozu chladit, k tomu je vybaven svým vlastním chladičem. Ovšem teplo odcházející z tohoto chladiče se hromadí v místnosti a při vysokém počtu serverů se teplota může velmi rychle vyšplhat k hodnotám, kdy vlastní chlazení začíná být neúčinné a roste riziko poškození serveru nebo dokonce požáru. Proto je nutné teplo odvádět i z místnosti. Velmi důležité je také sledování vlhkosti, nižší vlhkost zvyšuje riziko vzniku výboje statické elektřiny. Udržování konkrétní hladiny vlhkosti nám může ušetřit finanční prostředky, protože nepotřebujeme příliš velký výkon na chlazení pro dosažení nižší teploty, ale udržujeme teploty vyšší, například kolem  $24\text{ }^{\circ}\text{C} - 25\text{ }^{\circ}\text{C}$ .

V menších datových centrech k tomu stačí běžná klimatizace, která bez problému zvládne udržet v místnosti přijatelnou teplotu. Ve velkých datových centrech se ale bez sofistikovanějšího řešení neobejdeme. V praxi se používají nejčastěji následující dvě řešení.

### 2.2.1 Systém zdvojené podlahy

Systém zdvojené podlahy je postaven na principu, že byla při výstavbě datového centra zvýšena původní podlaha. Vzniklý prostor poté může sloužit pro elektroinstalaci a přívody studeného vzduchu. Horký vzduch odcházející ze serverů je nasáván ventilací a poté proudí do horní části tzv. CRAC<sup>5</sup> jednotky. Zde dojde k jeho ochlazení a následně, díky statickému tlaku v přírodním potrubí, je tlačěn přes perforované podlahové dlaždice k serverům. Výhodou takového zařízení je, že studený vzduch je přiváděn primárně k serverům a celá místnost tak není ochlazována rovnoměrně. Uvedme pár výhod tohoto systému. *Dodávám, že následující blok je parafrázován z článku „HVAC Cooling Systems for Data Centers“, který je uveden v seznamu použité literatury.*

1. Jelikož je chladný vzduch přiváděn z malé vzdálenosti, nemusí být ochlazován na velmi nízkou teplotu. Na krátké vzdálenosti se jeho teplota příliš

---

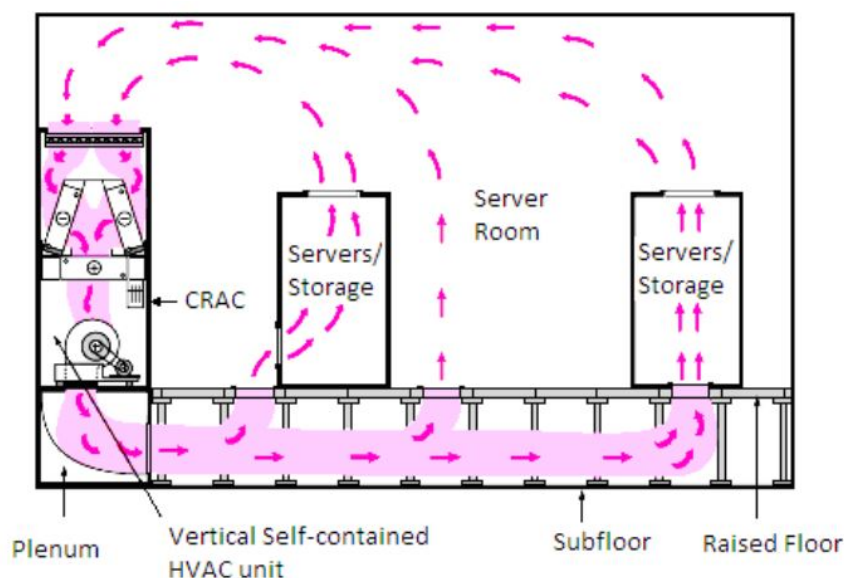
<sup>5</sup>Computer Room Air Conditioning Unit – je zařízení, které umožňuje monitorovat teplotu, rozvod vzduchu a vlhkost v datovém centru a současně ochlazuje horký vzduch odcházející ze serverů.



nezmění a navíc nepotřebujeme chladit celou místnost. Tím pádem máme menší nároky na výkon systému.

2. Vzduch může být přiváděn v prostoru zdvojené podlahy a není nutné jej přivádět potrubím. V praxi je však tento prostor využíván i pro elektroinstalaci a datové kabely a v tomto případě je výhodnější pro rozvod vzduchu potrubí využít.
3. Rozmístění serverů v serverovně lze kdykoliv změnit, protože podlaha se skládá z perforovaných dlaždic. Po přemístění serverů stačí dlaždice jen přeskládat a chlazení je opět funkční.
4. Nižší statický tlak zaručí nižší spotřebu energie a současně i nižší hladinu zvuku ventilátoru.

Celý systém ovšem musí být správně vytvořen. Špatná instalace, může zapříčinit nerovnoměrný rozvod vzduchu a vznik tzv. horkých míst.

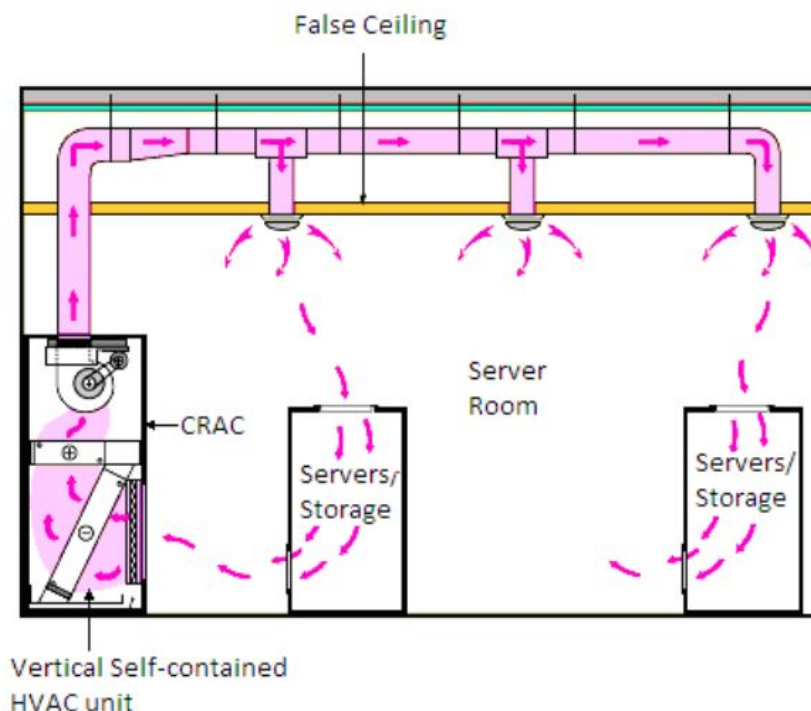


Obrázek 1: Systém zdvojené podlahy – převzato z článku HVAC Cooling Systems for Data Centers, str. 49

### 2.2.2 Stropní rozvod vzduchu

Stropní rozvod vzduchu využívá stropního potrubí pro rozvod chladného vzduchu a horký vzduch je poté nasáván CRAC zařízením. Celá místnost datového centra je ochlazována rovnoměrně a je zde udržována stálá teplota. Klesající studený vzduch je pak využit pro chlazení serverů. V tomto modelu ventilace vzduchu dochází k velkému promíchávání studeného a horkého vzduchu a tento model není energeticky účinný, což má za následek vyšší finální náročnost. Současně

je takový systém i náročnější na technickou údržbu. Tyto faktory se zdají jako rozhodující, a proto se v datových centrech setkáme především se systémem zdvojené podlahy.



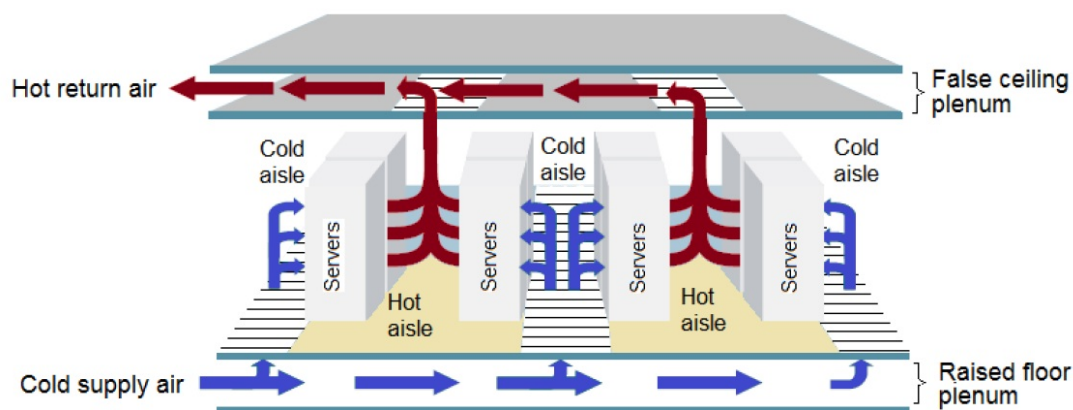
Obrázek 2: Stropní rozvod vzduchu – převzato z článku HVAC Cooling Systems for Data Centers, str. 50

### 2.2.3 Horká a chladná ulička

Pro snažší chlazení většího množství serverů v datovém centru existuje vhodné uspořádání, které vytvoří tzv. horké a chladné uličky. Tento návrh spočívá v seřazení serverových stojanů do střídavých řad s přívody studeného vzduchu na jedné straně stojanu a s odvodem horkého vzduchu na straně druhé. Přičemž stojany jsou k sobě otočeny čelem a prostor mezi nimi se nazývá chladná ulička. Na jejich zadní straně tomuto prostoru říkáme horká ulička. Teplota vzduchu v chladné uličce se může pohybovat okolo 20 °C – 25 °C (podle odpovídající vlhkosti). Naopak v horké uličce se teploty mohou pohybovat i nad 32 °C. Horký vzduch je následně ochlazován pomocí CRAC jednotky a studený vzduch je pak rozváděn podlahou opět k serverům. Cílem takové konfigurace je:

1. Zamezení smíchání horkého a studeného vzduchu, než horký vzduch projde ventilačním zařízením.
2. Přesun horkého vzduchu k jeho ochlazení a následnému opětovnému použití.

3. Šetření energie a nižší chladicí nároky díky dobrému řízení toku vzduchu.



Obrázek 3: Horká/studená ulička – převzato z článku HVAC Cooling Systems for Data Centers, str. 53

## 2.3 Ochrana před neoprávněným vstupem

Hovoříme-li o fyzické ochraně datového centra, nemůžeme pominout ani zabezpečení neoprávněného fyzického přístupu. V dnešní době může být totiž riziko takového útoku vyšší než v minulosti, protože datová centra mohou obsahovat velmi důležitá a citlivá data, především pokud se jedná o státní záležitosti.

Fyzická ochrana nezačíná přímo před samotným vstupem do datového centra. Mnohdy je udržován tzv. ochranný perimetr před celým objektem. Tzn. oplocení či zeď kolem objektu, stálá ozbrojená ostraha, atd. Ve venkovní části je ideální ponechat otevřený prostor. Pomocí kamerových systémů je jednoduché útočníka odhalit, jelikož se nemůže schovat za žádnou překážku. Kamerový systém může využívat umělou inteligenci pro detekci různých neobvyklých situací – například odhalení stojícího vozidla v místě, kde běžně vozidla neparkují, identifikace osob pohybujících se v objektu a jejich počet. Ve venkovním prostoru mohou být dále umístěna čidla na detekci pohybu, mikfony, které detekují podezřelý šum a moderní centra mají svůj perimetr zabezpečena i pomocí senzorů magnetických anomálií. Setkat se můžeme i s laserovými závorami nebo mikrovlnnými radiolokátory. Uvnitř serverovny by měly být zabezpečeny jak uličky, ve kterých jsou umístěny servery, ale i rackové skříně.[3]

Naopak zaměstnanci, jejichž náplň práce je v datovém centru, musejí mít přístup přímo do datového centra i k jednotlivým rackovým skříním. Jedná se o tzv. řízení přístupu osob, kdy se často využívá systému přístupových karet, které obdrží každý zaměstnanec. Softwarově je každé kartě nastaveno příslušné oprávnění pro přístup do jednotlivých částí objektu. Případně lze tímto způsobem monitorovat, v jaké části objektu se daný zaměstnanec nachází. Dnes se již

můžeme setkat i s moderními metodami ověřování, a sice biometrickým ověřením – použitím otisku prstu nebo skenování sítnice oka. Vyjíměčně se můžeme setkat i s rozpoznáváním tváře.

## 2.4 Ochrana před požárem

Hlavní zbraní proti vzniku požáru je dobrá vzduchotechnika. Ovšem ani tak nejsme schopni riziko jeho vzniku zcela vymýtit, jelikož může dojít k chybě na elektroinstalaci, případně k požáru způsobenému technickým problémem na některém ze serverů. Pro včasnou detekci požáru se používají detektory kouře nebo teploty, které jsou schopny včas uvědomit odpovědný personál nebo dokonce automaticky zaslat zprávu jednotkám IZS. Moderní datová centra využívají i automatické hasicí systémy. Hašení však neprobíhá vodou, protože by mohlo dojít k poškození techniky. V dnešní době se používají speciální plynová hasiva, nejčastěji FM-200, Novec 1320 nebo Inergen, která nesnižují hladinu kyslíku.[3] Hasivo je uchováváno ve speciálních tlakových nádobách a v případě požáru je veškerý obsah nádoby vyprázdněn během několika vteřin. Toto si vyžádá obrovský tlak, a proto i přírodní potrubí včetně kolen musí být velmi silné, aby takto silný tlak vydrželo.

## 2.5 Zálohování dat

Pokud se datové centrum stará i o uchovávání dat uživatelů, je nezbytně nutné tato data zálohovat. Díky velkému používání může docházet k poškození disků a jejich následnému znehodnocení. Uložená data je pak těžké, někdy i dokonce nemožné, obnovit. Proto je potřeba mít veškerá data neustále zálohovaná. V případě menších serveroven, kdy se data mění jen párkrát za den, je možné provádět zálohu například jednou denně a stačí data pouze překopírovat např. na zálohovací server. U velkých datových center by bylo takové řešení nedostačující, neboť uchovávají obrovské množství dat, která se rychle mění, a výpadek by mohl způsobit velké problémy. V praxi můžeme využít sjednocení několika disků do tzv. diskového pole. Tuto techniku nazýváme RAID<sup>6</sup> a umožňuje nám data rozložit na několik disků. V případě, že dojde k výpadku disku, můžeme jej bezpečně vyměnit za nový, aniž bychom přišli o část dat. Existuje několik typů pole RAID, níže uvádím ty nejznámější a nejčastěji používané.

- **RAID 0** – Tento způsob neslouží k záloze dat. Data jsou rozdělena na nejméně dva disky, kdy každý obsahuje část dat. V případě výpadku jednoho disku, přicházíme o veškerá data, protože část dat na nepoškozeném disku bude nečitelná. Tento způsob můžeme využít u méně důležitých systémů, kde chceme získat vyšší výkon a kapacitu úložiště.
- **RAID 1** – Zde se využívá opět nejméně dvou disků (jejich počet musí být vždy sudý), které jsou zrcadleny. Jinými slovy to znamená, že ta samá data,

---

<sup>6</sup>Redundant Array of Independent Disks neboli pole redundantních nezávislých disků.

kteřá jsou na prvním disku, jsou obsažena i na disku druhém. V případě výpadku jednoho z disků, nepřicházíme o žádná data, můžeme bezpečně provést výměnu poškozeného disku a následně přkopírovat data z funkčního disku na disk nový. Nevýhodou tohoto způsobu je využití dvojnásobné kapacity (pro uložení 500 GB dat, potřebujeme celkovou kapacitu nejméně 1 TB). Rovněž dochází ke zpomalení v případě zápisu, data jsou nejprve zapsána na jeden disk a poté probíhá zápis stejných dat na disk druhý. Naopak čtení může být v tomto případě rychlé, neboť můžeme data přčíst z disku, který je momentálně méně vytížen.

- **RAID 5** – Toto řešení je již sofistikovanější oproti předchozím dvěma řešením a vyžaduje nejméně tři disky, na která jsou data ukládána stejným způsobem, jako je tomu při **RAID 0**. Každý z disků uchovává i paritní informaci (samoopravný kód) o uložených datech. V případě výpadku jednoho z disků můžeme snadno zjistit, která data chybí, a po výměně za nový disk mohou být tato data obnovena. Výhodou je rychlejší čtení, ale díky výpočtu parity je zápis pomalejší.
- **RAID 6** – Jedná se o vylepšení předchozího řešení, kdy jsou namísto jedné parity použity parity dvě. Toto řešení si vyžadá minimálně čtyři disky a parita je opět uložena střídavě na všech discích. V tomto případě systém ustojí výpadek až dvou disků naráz. Nicméně díky výpočtu dvou paritních kódů je zápis ještě pomalejší než v případě předchozího řešení.
- **RAID 10** – Řešení kombinující **RAID 1** a **RAID 0** (odtud jeho název RAID 10). Veškerá data jsou uložena střídavě, čímž je umožněno například jejich rychlé čtení a zápis. Zároveň jsou data i zrcadlena. Pokud by tedy došlo k výpadku některého z disků, nedojde ke ztrátě dat. Tento způsob nám navíc umožňuje poměrně rychlé čtení i zápis.

## 2.6 Další možnosti fyzické bezpečnosti

Níže uvádím další možnosti, které mohou přispět k fyzickému zabezpečení datových center.

- **Záložní energetické vedení** – Je vhodné mít zajištěn přívod elektrické energie z více rozvodů a to i přesto, že je datové centrum vybaveno agregátem, který by zajistil přívod elektrické energie i při výpadku sítě.
- **Strategická poloha** – Datové centrum by mělo být postaveno v optimální časové dojezdnosti vozidel IZS. Může tak být zajištěn rychlý příjezd hasičských vozidel v případě vypuknutí požáru nebo rychlá reakce ozbrojených složek při útoku na datové centrum. Současně by měla být snaha vyhnout se například záplavovým oblastem, kdy by během nečekaných záplav mohla být poškozena veškerá technika v datovém centru.

## 2.7 Projekt FENIX

Jedná se o českou skupinu sdružující poskytovatele internetového připojení. Vznikl v roce 2013 po masivních DDoS útocích na Českou republiku, které byly směřovány na zpravodajské a mediální weby, banky, ale i mobilní operátory. Některé z těchto služeb byly v tu chvíli nedostupné – jednalo se například i o krátkodobou nedostupnost plateb kartou. Reakcí na tento útok byl právě vznik projektu FENIX. Tento projekt má za cíl v případě podobného útoku zajistit dostupnost služeb internetu pro české občany. ISP, který je součástí této skupiny může být považován za důvěryhodného poskytovatele a jedná se o jakousi „záruku kvality a bezpečnosti.“ Nicméně podmínky pro vstup jsou opravdu velmi přísné a v současné době má projekt FENIX pouze 28 členů<sup>7</sup>, mj. i společnost Internext 2000, s.r.o., jejíž datové centrum jsem měl možnost navštívit a je popsáno v [následující kapitole](#).

---

<sup>7</sup>Členy projektu FENIX jsou například Seznam.cz, CZ.NIC, O2, T-Mobile, CESNET, CETIN, České Radiokomunikace a další. Seznam všech členů je k dispozici na <https://fe.nix.cz/>.

## 3 Zabezpečení datových center v praxi

### 3.1 Datové centrum společnosti Internext

Pro účely této diplomové práce jsem měl možnost navštívit datové centrum společnosti Internext 2000, s.r.o. ve Zlíně. Za tuto možnost bych chtěl poděkovat panu Zbyňku Glacnerovi, který mě datovým centrem provedl a poskytl výklad ohledně fyzického zabezpečení.

Datové centrum zajišťuje stálou konektivitu klientů Internextu, nenalezme zde žádné datové úložiště. Centrum funguje zhruba již šestým rokem a poskytuje připojení nejen na Zlínsku, ale i na jižní Moravě, konkrétně až v Kyjově. Existuje zde i možnost pronájmu rackových skříní a umístění vlastních serverů. Zákazník pak zaplatí pouze za pronájem racku a spotřebovanou elektrickou energii. Toto datové centrum splňuje podmínky úrovně Tier 3.

Několik fotografií s krátkými popisky z datového centra společnosti Internext 2000, s.r.o. můžete najít v [příloze A](#).

#### 3.1.1 Energetické vedení

Tím, že datové centrum splňuje úroveň Tier 3, musí být připojeno ke dvěma nezávislým zdrojům napájení. Jelikož bylo vybudováno na poměrně strategické poloze ve Zlíně, bylo zajištění této podmínky jednoduché. O řízení energie se zde stará zařízení nazvané ATS<sup>8</sup>, tj. zařízení, které může automaticky přepínat mezi jednotlivými napájecími zdroji, v případě výpadku některého z nich. Standardně je elektrická energie odebírána z první (primární) trafostanice a pokud by došlo k jejímu výpadku, je ihned přepnuto na druhou (sekundární) trafostanici. Jelikož takové přepnutí není úplně triviální věcí, malou chvíli mezi odpojením primárního napájení a připojením sekundárního napájení jsou všechny servery napájeny pomocí UPS. Za této situace jsou baterie UPS velmi vytížené a je tedy v zájmu životnosti těchto baterií, aby se takové přepínání dělo co nejméně. Ovlivnit výpadek primárního napájení nelze, ale opětovné přepnutí ze sekundární větve na primární už ano. Poté, co znovu primární větev naskočí, může docházet k menším výpadekům, proto procedura zpětného přepnutí probíhá následovně – čeká se alespoň dvacet minut od posledního výpadku primární větve, poté se už dá napájení považovat za stabilní a může opět dojít k přepnutí ze sekundárního napájení na primární.

Datové centrum je zároveň vybaveno motor-generátorem, který přichází na řadu, když selže i sekundární napájení. Procedura zpětného přepnutí je zde stejná, jako v předchozím případě. U motor-generátoru je však potřeba dávat pozor, aby nedocházelo k většímu kolísání odběru, neboť by si s tímto motor-generátorem nemusel poradit. Při běžném provozu se permanentní odběr pohybuje okolo 50 kW.

Datové centrum disponuje i UPS a protože se v datovém centru nachází velké množství serverů, skládá se UPS z několika „velikých“ baterií, které musejí být

---

<sup>8</sup>Automatic Transfer Switches

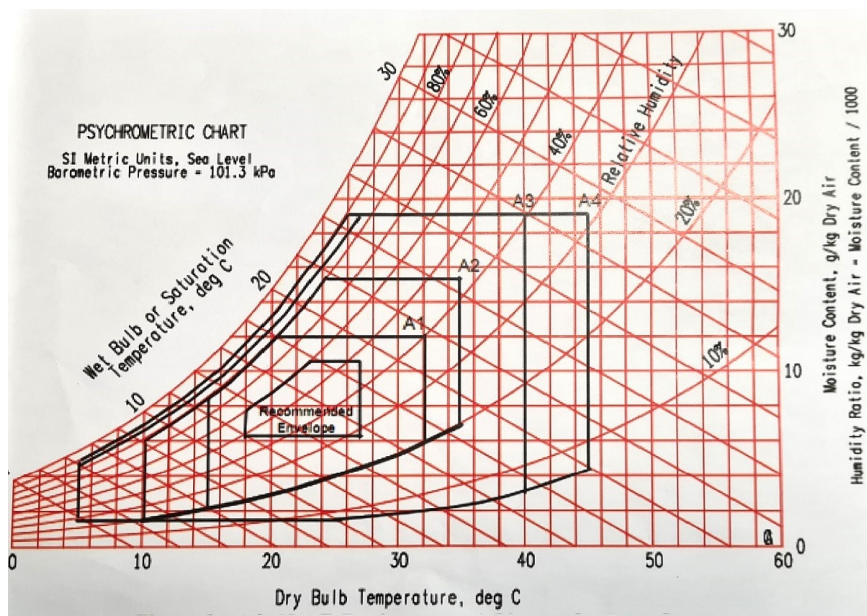


uloženy v samostatné skříni a jejich hmotnost přesahuje dvě tuny. V současné době je plánována jejich výměna, neboť některé z baterií jsou v provozu už šestým rokem, tedy od spuštění datového centra.

### 3.1.2 Vzduchotechnika

V datovém centru společnosti Internext využívají uspořádání do chladné a horké uličky. V chladné uličce je teplota příjemná, okolo 24 °C, naopak v horké uličce se může vyšplhat i nad 32 °C. Ventilace je postavena na systému zdvojené podlahy, vzduch však není přiváděn přímo k serverům, ale je ochlazována celá část chladné uličky, přičemž horká ulička je od ní fyzicky oddělena. Teplý vzduch je z horké uličky nasáván několika ventilačními zařízeními, které jsou vybaveny i speciálním filtrem částic. Dále je přiváděn potrubím přes zdvojenou podlahu k ochlazení a následně ventilován do chladné uličky. Zbylý prostor zdvojené podlahy pak slouží pro elektroinstalaci a k přivodu strukturované kabeláže. Opět aby byly splněny podmínky úrovně Tier 3, je datové centrum vybaveno záložním ventilačním zařízením, které je automaticky uvedeno do chodu, dojde-li k poruše na primárním zařízení.

V některých datových centrech se snaží teplotu v chladné uličce udržovat na 20 °C, avšak to si vyžádá vyšší spotřebu a tím pádem i vyšší finanční nároky. V Internextu se snaží, aby vzduchotechnika byla méně finančně náročná a současně se šetřilo životní prostředí. Díky správnému udržování vlhkosti stačí, aby byl vzduch ochlazen pouze na 24 °C. Vztah mezi vlhkostí a teplotou vzduchu je znázorněn na tzv. psychrometrickém diagramu.



Obrázek 4: Psychrometrický diagram



### 3.1.3 Požární ochrana

Ochranu před požárem řeší hasicí systém, který má rozvody po celé místnosti datového centra. V místnosti se nachází několik speciálních tlakových nádob se speciálním hasivem *NOVEC 1230*. Použití tohoto hasiva v datových centrech je dáno přímo zákonem,<sup>9</sup> protože je šetrné k životnímu prostředí (nezpůsobuje ozonovou vrstvu) a není tolik škodlivé pro lidské zdraví jako jiná hasiva. V případě vypuknutí požáru je celý obsah nádoby vyprázdněn během několika vteřin, což si vyžádá obrovský tlak. Veškeré potrubí, spoje a kolena musí tento tlak vydržet, proto mají opravdu velmi silnou vrstvu. V horké uličce se na stropní části nachází ventily, kterými by hasivo bylo dopraveno přímo k serverům.

Dále je datové centrum vybaveno požárními hlásiči a to jak uvnitř místnosti, tak před místností, aby byl personál o případném požáru včas informován. U vstupu se nachází i speciální protipožární dveře, které by požár neměly pustit dále do budovy. Tyto dveře disponují funkcí automatického uzamčení.

### 3.1.4 Ochrana

Celá místnost je zabezpečena kamerovým systémem, který snímá každou část prostoru. Jednotlivé rackové skříně jsou uzamčeny, aby nikdo nepovolaný nemohl manipulovat či sabotovat některý ze serverů. Přístup k nim mají buď zákazníci, kteří si daný rack pronajali, nebo přímo technici Internextu. Hlavní vstup je taktéž zabezpečen a ke vstupu je nutná bezpečnostní karta.

## 3.2 Serverovna Edhouse

Oproti datovému centru společnosti Internext je serverovna Edhouse velmi malá, proto nemá smysl bavit se zde o standardu Tier. Nicméně pro celou firmu pohodlně dostačuje a její kapacity jsou naddimenzované pro budoucí rozvoj. Pár fotografií ze serverovny Edhouse můžete najít v [příloze B](#). Oproti datovému centru Internextu, které především zajišťovalo konektivitu zákazníkům, slouží serverovna v Edhouse pro provoz serverů s interními systémy a k rozvodu strukturované kabeláže po celé firmě.

Jednotlivé body fyzického zabezpečení budou zmíněny ve stručnosti a tato kapitola je spíše pro porovnání, jak složité je fyzické zabezpečení v datovém centru, které poskytuje konektivitu pro tisíce lidí, oproti malé serverovně sloužící pro jedinou firmu.

- **Energetické vedení** – V serverovně firmy Edhouse není žádné redundantní napájení z důvodu, že nebylo technicky možné jej zajistit. Proto v případě výpadku elektrické energie přijdou na řadu UPS, které mají za úkol bezpečně vypnout všechny servery. Jejich baterie plně stačí na splnění tohoto úkolu, ovšem nemohly by sloužit na udržení chodu všech serverů. Servery se opět mohou zapnout až po skončení výpadku.

---

<sup>9</sup>Zákon taktéž umožňuje použití hasiva FM-200, od jeho používání se však postupně ustupuje.

- **Vzduchotechnika** – Jelikož se jedná o poměrně malou místnost, na její ochlazení stačí pouze klasická klimatizační jednotka, která se používá v kancelářích či domácnostech. Samotné servery negenerují příliš mnoho tepla, a proto je takové řešení zcela dostačující. Teplota udržovaná v serverovně se pohybuje okolo 23 °C – 24 °C.
- **Požární ochrana** – O ochranu před požárem se starají čidla detekující požár, v případě vypuknutí požáru by byl upozorněn odpovědný personál a hašení by probíhalo ručně, případně za pomoci hasičského záchranného sboru.
- **Ochrana** – Přístup do serverovny je umožněn pouze odpovědnému personálu. Serverovna je chráněna detektory pohybu a dveře mají magnetické zámky, které jsou schopny detekovat jejich otevření.

## 4 Raspberry Pi

Raspberry Pi je miniaturní počítač, který svou velikostí odpovídá velikosti platební karty. Byl vyvinut v roce 2012 ve Velké Británii a hlavním cílem jeho tvůrců byla propagace studia počítačových věd a současně zvýšení zájmu mladých lidí o programování a práci se samotným hardwarem. Na této malé desce najdeme následující rozhraní:

- HDMI (od verze 4 nahrazen dvěma mirco-HDMI porty),
- ethernet,
- čtyři USB porty pro připojení klávesnice a myši, případně dalších periférií (od verze 4 nahrazeny dva porty USB 2.0 za porty USB 3.0),
- micro USB port pro napájení,
- slot pro micro SD kartu (SD karta je použita pro uložení OS, který je z ní poté bootován),
- GPIO<sup>10</sup> porty, které slouží např. k připojení senzorů. Tyto porty je poté možné naprogramovat v jazyce Python. V této práci o nich bude ještě řeč, neboť senzory pro měření proudu budou připojeny právě přes GPIO porty,
- novější modely konkrétně Raspberry Pi verze 3 a 4 mají i moduly pro Wi-Fi (včetně 5GHz pásma) a Bluetooth 5.0 připojení.

Pro Raspberry Pi se primárně používá operační systém, který staví na linuxové distribuci Debian, tzv. *Raspbian GNU/LINUX*. Můžeme se ale setkat i s jinými linuxovými distribucemi. Dokonce lze na Raspberry Pi spustit taktéž systém Windows v „odhlečené“ verzi.

Přestože původní záměr tvůrců byl především pozvednutí zájmu o počítačové vědy, Raspberry Pi se velmi rychle uchytilo i v komerční sféře. Díky jeho levné pořizovací ceně se Raspberry Pi začalo používat jako např. výrobní terminály, na kterých zaměstnanci zadávají příkazy pro výrobu, docházkové terminály, rezervační terminály atd. Ve firmě Edhouse s.r.o. plní Raspberry Pi úlohu docházkových terminálů, případně jako rezervační terminál pro rezervaci zasedacích místností. Tento rezervační terminál je vybaven i senzorem pohybu a pokud detekuje pohyb, rozsvítí se jeho obrazovka a kolemjdoucí může ihned vidět, zdali je daná zasedací místnost obsazená, nebo jestli je tam naplánována nějaká rezervace. Pomocí dotykové obrazovky je možné si místnost ihned zarezervovat, přičemž lze zvolit i délku trvání mítinku. Navíc je každé zařízení vybaveno rozšířením pro podporu PoE<sup>11</sup> a tudíž není nutné připojovat zdroj napájení.

---

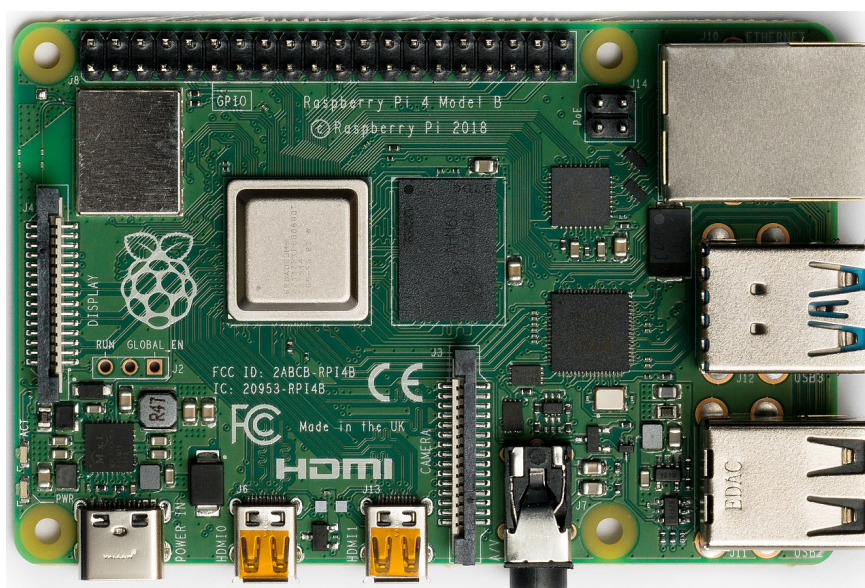
<sup>10</sup>General Purpouse Input/Output

<sup>11</sup>Power over Ethernet – napájení zařízení pomocí rozhraní Ethernet

V první verzi obsahoval Raspberry Pi procesor BCM2835 typu SoC<sup>12</sup> s kapacitou operační paměti 256 MB, jehož instrukční sada odpovídala architektuře ARM. Počet portů na GPIO byl 26, v dalších verzích se pak tento počet navýšil na 40. Následovaly další verze, které již nabízely podstatně vyšší kapacitu operační paměti – konkrétně Raspberry Pi verze 2 přichází s procesorem BCM2836/7 a až s 1 GB operační paměti.

Větší změnu přináší Raspberry Pi verze 3, které je vybaveno procesorem BCM2837B0 (operační paměť zůstává 1 GB jako v případě verze 2) a navíc obsahuje i gigabitový Ethernet a přichází s podporou Wi-Fi 2GHz a 5GHz.

Poslední verze Raspberry Pi, tedy verze 4, která byla uvedena na trh v roce 2019, obsahuje procesor BCM2711 a umožňuje kapacitu operační paměti až 8 GB.



Obrázek 5: Raspberry Pi verze 4 – převzato z Wikipedia EN; Raspberry Pi

Z důvodu přetrvávající čipové krize, která vypukla následkem pandemie nemoci COVID-19, je prakticky nemožné, pro běžného člověka, v současné době sehnat více kusů Raspberry Pi verze 3. Výrobci ve Velké Británii se je snaží dodávat především zákazníkům, kteří Raspberry Pi používají pro komerční účely.

---

<sup>12</sup>System on a Chip – integrovaný obvod, který odstraňuje hranici mezi mikroprocesorem a mikropočítačem.

## 5 Fyzické sestavení a zapojení zařízení

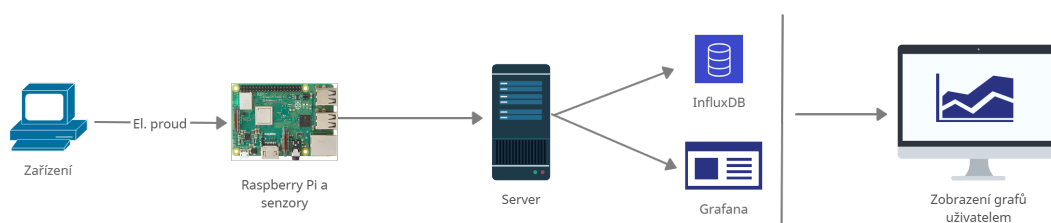
V této kapitole se zaměříme na samotné sestavení zařízení a popíšeme si jednotlivé komponenty, ze kterých bude výsledné zařízení sestávat.

### 5.1 Princip fungování zařízení

Následující blok shrnuje stručný přehled, na jakém principu by mělo výsledné zařízení fungovat.

1. Zařízení bude sbírat data o průtoku proudu pomocí externích senzorů.
2. Sesbíraná data budou ukládána do databáze InfluxDB, která poběží na externím serveru.
3. Na externím serveru bude fungovat webová aplikace Grafana, která bude přebírat data z databáze InfluxDB a zobrazí je v přehledných grafech s možností filtrování.
4. Uživatel přistoupí na webovou aplikaci Grafana a bude si moci procházet historii dat v grafu.
5. Uživatel si bude moci zobrazit webový dashboard přímo na Raspberry Pi, prohlédnout si aktuální informace o měření proudu, případně zaslat signál pro vypnutí/zapnutí serveru, a spravovat nastavení e-mailových notifikací.

Princip fungování je vyobrazen na následujícím grafu:



Obrázek 6: Princip fungování zařízení

### 5.2 Senzory

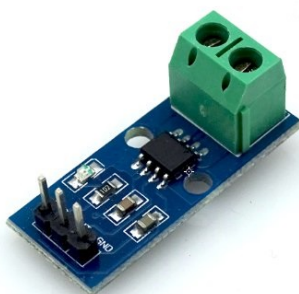
Abychom mohli senzory zapojit, je potřeba je připojit na porty GPIO, kterými je Raspberry Pi vybaveno. Jak již bylo uvedeno v kapitole o Raspberry Pi, máme k dispozici celkem 40 portů. Ne všechny porty však plní stejnou úlohu – je potřeba rozlišovat mezi porty, které slouží k uzemnění (označovány *GND*) a dále mezi porty označovanými jako *UART*, *I2C*, *SPI0*, *EEPROM*, *PWM* a *PCM*.<sup>13</sup>

<sup>13</sup>Pro naše potřeby budeme pracovat především s porty *GND* a *I2C*.

Třetí skupinou pinů jsou dvojice napájecích pinů (5V a 3,3V). Velmi dobrým pomocníkem může být webová stránka <https://pinout.xyz>, která nabízí pěkný přehled všech portů, včetně označení jejich funkce a krátkého popisku.

### 5.2.1 Proudový senzor ACS712

Proudový senzor s čipem ACS712 je jednou ze dvou hlavních komponent výsledného zařízení. Senzor pracuje na principu Hallova jevu<sup>14</sup> a umožňuje nám měření stejnosměrného i střídavého proudu. Senzor obsahuje tři piny – pro napájení senzoru (5 V), který je označen jako *VCC* a uzemnění, označen jako *GND* (oba budou připojeny na piny GPIO Raspberry Pi). Třetím pinem je analogový výstup, označen jako *OUT*, a připojen do ADC převodníku, který se postará o převedení analogového signálu do digitální podoby, což nám umožní přesnější datový výstup. Výstupem z tohoto senzoru jsou hodnoty napětí; hodnoty proudu budeme muset vypočítat podle vztahu, který je podrobněji popsán v podkapitole [Kalibrace senzorů](#).<sup>15</sup>



Obrázek 7: Proudový senzor ACS712 – převzato z LáskaKit.cz

### 5.2.2 ADC převodník ADS1115

ADC převodník s čipem ADS1115 obsahuje čtyři samostatné kanály a slouží k převedení analogového signálu do digitální podoby, přičemž nám poskytuje 16bitovou přesnost<sup>16</sup> při 860 vzorcích za sekundu přes sběrnici I2C. Je vybaven několika piny, mj. i pro uzemnění a napájení. Samotný senzor je připojen přímo na GPIO.<sup>17</sup> Použitý typ převodníku se mi jeví spíše jako méně používaný,

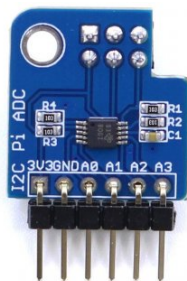
<sup>14</sup>Hallův jev se využívá nejen pro měření elektrického proudu, ale např. i pro bezkontaktní měření magnetického pole. Pro správné fungování je tedy nutné vyhnout se při měření silnému působení magnetického pole.

<sup>15</sup>Podrobná dokumentace k senzoru ACS712 je k dispozici zde: [https://www.laskakit.cz/user/related\\_files/acs712.pdf](https://www.laskakit.cz/user/related_files/acs712.pdf)

<sup>16</sup>Jinými slovy jsme schopni malý signál až 16× zesílit.

<sup>17</sup>Podrobná dokumentace k senzoru ADS1115 je k dispozici zde: <https://cdn-shop.adafruit.com/datasheets/ads1115.pdf>

případně jako nový model, neboť při hledání informací ohledně správného zapojení jsem nebyl schopen nalézt návod či video s použitím konkrétního typu převodníku.



Obrázek 8: ADC převodník ADS1115 – převzato z LáškaKit.cz

### 5.3 Zapojení senzorů do Raspberry Pi

Na [obrázku číslo 9](#) můžeme vidět, jakým způsobem budou senzory zapojeny ke sběrnici GPIO na Raspberry Pi. Použitý typ převodníku ADS1115 má připravené porty k připojení přímo na sběrnici, nicméně toto řešení by nám doslova „zablokovalo“ oba piny poskytující napájení 5 V, které je potřeba pro připojení proudového senzoru ACS712. Proto je převodník připojen samostatnými propojovacími vodiči přímo ke konkrétním pinům. Konkrétní zapojení vypadá následovně:

- 3,3V napájecí pin, který nám poskytne napájení převodníku.
- 2× 5V napájecí pin.
- Pin pro uzemnění.
- 2× GPIO standardní pin.

Senzor pro měření proudu vyžaduje napájení o velikosti 5 V. Jelikož ale máme oba 5V piny obsazené, naletujeme propojovací vodič na jeden z 5V pinů. Tím budeme mít napájení senzoru zajištěno. Prostřední pin na senzoru slouží jako analogový výstup a ten je pak připojen na pin převodníku s označením *A0*. Pro uzemnění máme na výběr dvě varianty – můžeme jej použít přímo z GPIO nebo využít pinu *GND* na převodníku.

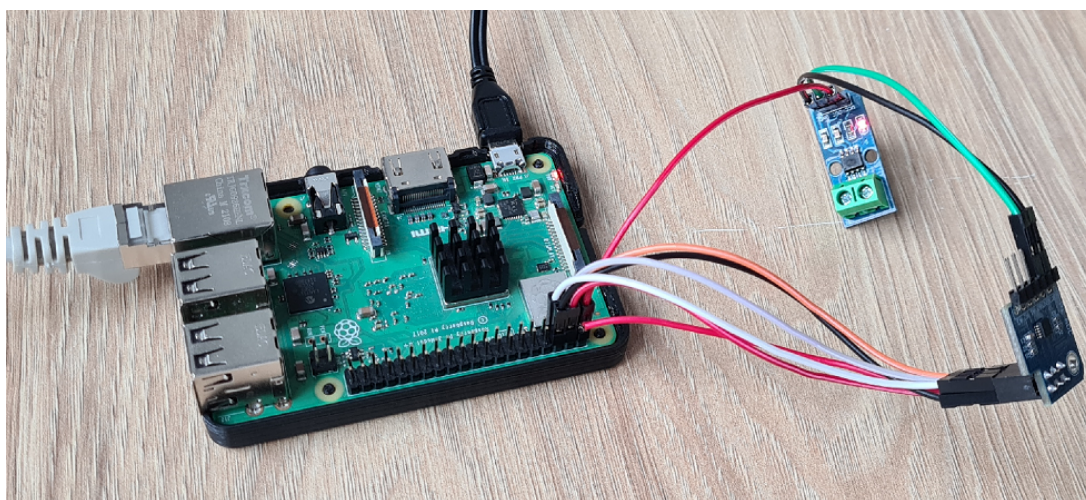
Pro lepší orientaci jsem zvolil stejnou barvu propojovacích vodičů pro stejný účel použití:

- Černá – uzemnění.
- Červená – napájení (5 V).
- Oranžová – napájení (3,3 V).



- Zelená – analogový výstup z proudového senzoru.
- Bílá – připojení převodníku na GPIO piny.

Pro vyšší bezpečnost v rámci testování a obhajoby této práce jsem se rozhodl, že bude prováděno měření slaboproudu. Z tohoto důvodu jsem použil proudový senzor s čipem do 5 A, aby byla zajištěna dostatečná citlivost měření i při nízkých hodnotách proudu. Záměna za senzor s čipem do 20 A pak bude pouze náhrada formou „kus za kus“ bez větších úprav.<sup>18</sup> Současně byly použity i tenčí vodiče. Poté budou ale nahrazeny silnějšími, které odpovídají použití pro 220 V.



Obrázek 9: Zapojení senzorů do Raspberry Pi

Pokud máme zapojení takto provedené, měla by se po zapnutí Raspberry Pi rozsvítit LED dioda umístěná na senzoru ACS712, která nám signalizuje, že senzor je napájen a obvod uzavřen.

## 5.4 Kalibrace senzorů

Senzor poskytuje hodnoty napětí, nicméně naším cílem je zjistit velikost proudu. Abychom dokázali hodnoty proudu odvodit z naměřeného napětí, musíme získat převodní vztah. K odvození vztahu budeme potřebovat následující hodnoty – napětí naměřené pomocí senzoru ACS712, označme ji jako  $U_{out}$ , napětí na senzoru ACS712 při nulovém proudu<sup>19</sup>, hodnotu označme jako  $VCC_0$ , a dále odchylku senzoru, označme jako *sensitivity*. Napětí při nulovém proudu a odchylka se dá vyčíst z dokumentace, tj.

$$VCC_0 = \frac{VCC}{2} \quad V$$

<sup>18</sup>V případě 20A senzoru bude pouze jiná hodnota odchylky.

<sup>19</sup>Vychází z napětí poskytovaného přes 5V GPIO pin. Toto záleží na použitém zdroji Raspberry Pi. V případě originálního zdroje k Raspberry Pi  $VCC \approx 5,214 \text{ V}$



$$sensitivity = 185 \text{ mV/A}$$

Pro odchylku doplním informaci, že ačkoliv je v dokumentaci udána hodnota uvedená výše, tak skutečná odchylka pro mnou použitý senzor je rovna:

$$sensitivity = 191 \text{ mV/A}$$

Jak uvádí i sám výrobce, hodnota odchylky uvedená v dokumentaci je pouze orientační. Mou hodnotu odchylky jsem stanovil výpočtem, kdy jsem si změřil hodnoty napětí ze senzoru ACS712 při 0 A a 3 A (tím jsem získal celý rozsah poskytovaný použitým laboratorním zdrojem) a použil následující vztah:

$$sensitivity = \frac{U_2 - U_1}{I_2 - I_1} \text{ [mV/A]}$$

Samotnou kalibraci jsem prováděl pomocí zmíněného laboratorního zdroje za použití řízené zátěže (na [obrázku číslo 10](#)). Nasbírané hodnoty napětí z ADC převodníku jsem porovnával s hodnotami napětí naměřeného klasickým multimetrem. Velikost procházejícího proudu byla zobrazena na displeji zátěže. Jednotlivé hodnoty jsou zobrazeny v následující tabulce:<sup>20</sup>

| Proud [A] | ACS712 [V] | Multimetr [V] |
|-----------|------------|---------------|
| 0         | 2,580      | 2,578         |
| 0,5       | 2,676      | 2,679         |
| 1         | 2,767      | 2,775         |
| 1,5       | 2,868      | 2,872         |
| 2         | 2,964      | 2,968         |
| 2,5       | 3,060      | 3,063         |
| 3         | 3,156      | 3,157         |

Tabulka 1: Kalibrace senzoru – naměřené hodnoty

Nyní, když známe hodnoty proudu a napětí, jsme schopni určit vztah pro výpočet proudu:

$$I = \frac{U_{out} - VCC_0}{sensitivity} \text{ [A]}$$

Pro ověření provedme výpočty z nasbíraných hodnot a porovnejme hodnoty z laboratorního zdroje s hodnotami vypočítanými podle zmíněného vztahu. Doplněná tabulka pak bude vypadat takto:<sup>21</sup>

<sup>20</sup>Proud – proud vyčtený z displeje zátěže; ACS712 – napětí vyčtené přes Raspberry Pi; Multimetr – napětí zjištěné klasickým multimetrem.

<sup>21</sup>Drobné odchylky hodnot proudu mohou být způsobeny nepřesností a chybami při měření.

| Proud [A] | Výpočet [A] | ACS712 [V] |
|-----------|-------------|------------|
| 0         | 0,005       | 2,580      |
| 0,5       | 0,50        | 2,676      |
| 1         | 1           | 2,767      |
| 1,5       | 1,474       | 2,868      |
| 2         | 2,01        | 2,964      |
| 2,5       | 2,51        | 3,060      |
| 3         | 3,01        | 3,156      |

Tabulka 2: Kalibrace senzoru – porovnání hodnot

Velmi důležitá je hodnota zdrojového napětí  $VCC$ , jejíž hodnota může podstatným způsobem změnit vypočítané hodnoty proudu. Senzor je kalibrován na použití s oficiálním zdrojem Raspberry Pi, který poskytuje napětí cca 5,1 V a tedy  $VCC_0 = 2,58$  V. Ovšem hodnota se může mírně měnit z důvodu kolísání napětí v různých energetických sítích. Při použití jiného než oficiálního zdroje k Raspberry Pi, není zaručena přesnost měření. Při testech s různými zdroji jsem zjistil, že navíc dochází k většímu kolísání hodnot při měření. Dále je doporučeno mít Raspberry Pi zapojeno přímo do elektrické zásuvky. V případě použití prodlužovacího kabelu, je nejlepší připojit jej přímo do první zásuvky. Bude tím zaručena nejpřesnější hodnota  $VCC_0$ .

Senzor ACS712 nám umožňuje jak měření stejnosměrného proudu (DC), tak i měření střídavého proudu (AC), z tohoto důvodu je jeho rozsah v rozmezí  $\pm 5$  A.<sup>22</sup> Uvedený vzoreček nám pak umožní stanovit hranice hodnot napětí:<sup>23</sup>

$$U_{out-min} \doteq 1,63 \text{ V}$$

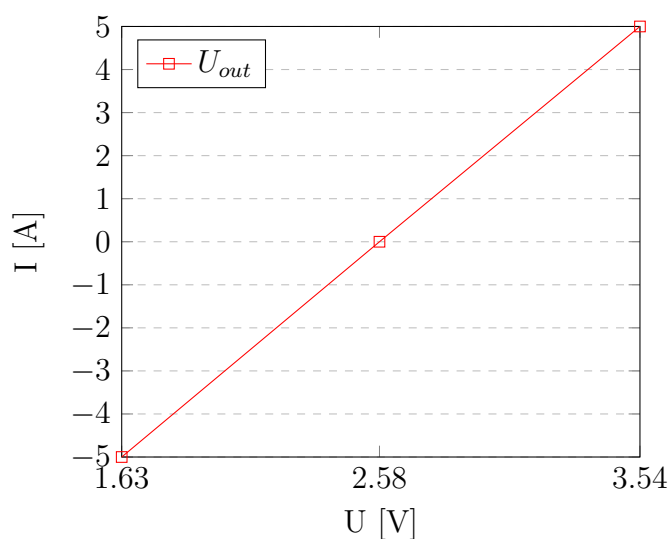
$$U_{out-max} \doteq 3,54 \text{ V}$$

<sup>22</sup>Při výpočtu proudu je počítána absolutní hodnota napětí, jelikož záporné či kladné hodnoty napětí nemají vliv na výslednou hodnotu proudu.

<sup>23</sup>Uvedené hodnoty jsou pro verzi 5 A. Pro 20 A budou hodnoty lehce odlišné, pro tuto verzi je  $sensitivity = 100$  mV/A.

Totéž graficky:

Graf mezních hodnot napětí 5A senzoru ACS712



Následující obrázek zachycuje kalibraci senzorů. Můžeme na něm vidět laboratorní zdroj, na jehož displeji svítí dvě hodnoty – napětí (9,509 V) a proud (1,003 A). Dalším zařízením je pak tzv. „zátěž“, na jejímž displeji můžeme vidět shodné hodnoty proudu<sup>24</sup>, ale lehce odlišné hodnoty napětí<sup>25</sup> (způsobeno ztrátami při přenosu).



Obrázek 10: Kalibrace senzorů

<sup>24</sup>Jedná se o třetí uvedenou hodnotu s označením  $C_u$ .

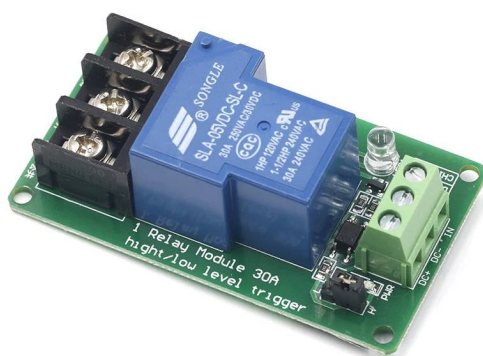
<sup>25</sup>Druhá zobrazená hodnota s označením  $V_{ol}$ .

## 5.5 Připojení relé

Jedním z cílů práce je umožnit vypnutí, popřípadě zapnutí serveru. Proto jsem se rozhodl přidat k sensorům ještě relé. Zvolil jsem relé pro proud do 30 A, aby toto použití bylo vhodné i do „ostrého“ provozu. Vypnutí a zapnutí serveru tak bude prováděno přímým odpojením měřeného zařízení od energie. Pokud bychom chtěli provést „soft reset“, tedy řízeně vypnout server, nejlepším řešením je použití UPS, která zmíněný postup umožňuje.<sup>26</sup>

Pokud zařízení, u kterého provádíme měření proudu, disponuje vlastní baterií (např. mobil, tablet nebo notebook), dojde pouze k zastavení nabíjení a zařízení zůstane v provozu. V praxi však servery baterií nedisponují a tudíž by došlo k jejich okamžitému vypnutí. Někdy je takový postup nevyhnutelný, např. pokud dojde k vysokému přehřátí serveru je nutné jej co nejrychleji odstavit i za cenu ztráty některých dat.

Relé obsahuje celkem tři porty – NC, NO, COM.<sup>27</sup> Pro naši potřebu využijeme porty NC a COM. Tj. v základním stavu bude obvod uzavřen a proud bude procházet, po sepnutí relé (na pokyn uživatele) dojde k otevření obvodu a proud protékat nebude. Měřené zařízení se tak vypne, popř. přejde na baterii (je-li k dispozici).



Obrázek 11: Relé modul do 30 A – převzato z LáškaKit.cz

## 5.6 LED diody

V neposlední řadě byly k zařízení přidány i tři LED diody (červená, zelená, žlutá), které vizuálně symbolizují jeho aktuální stav. Stav zařízení jsme tedy schopni zjišťovat jak přes webové rozhraní (v [příští kapitole](#)), tak i fyzicky.

Jednotlivé barvy představují následující stavy:

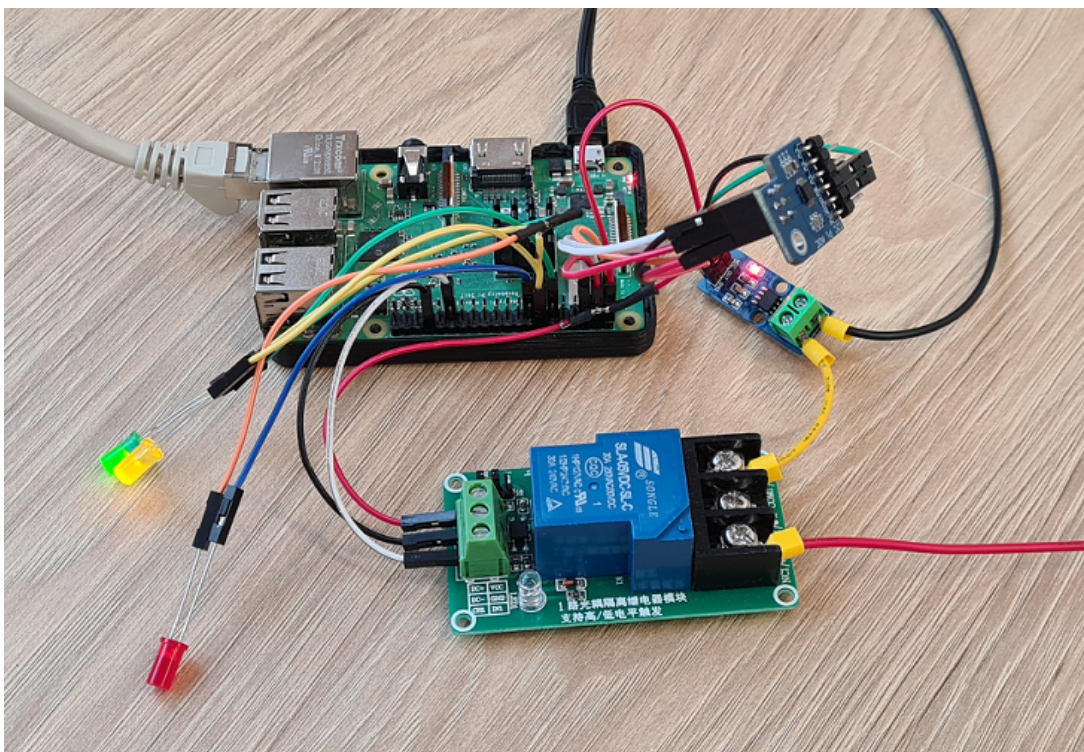
<sup>26</sup>Toto řešení je používáno i v praxi, lze se s ním setkat například v serverovně firmy Edhouse.

<sup>27</sup>NC – Normally Closed, NO – Normally Opened, COM – spojí obvod.

- Zelená – vše je v pořádku, měření probíhá.
- Zelená + žlutá – měření probíhá, ale vyžaduje pozornost. Detekována nízká hodnota proudu.
- Žlutá – měření vypnuto.
- Červená – měřené zařízení odpojeno, měření neprobíhá.

## 5.7 Zapojení pro ověření funkčnosti

Na fotografii je možné vidět zapojení všech jednotlivých komponent sestaveného zařízení, tedy senzorů ACS712 a ADS1115, relé a tří LED diod pro vizualizaci stavu zařízení, pro ověření funkčnosti.



Obrázek 12: Zapojení komponent pro ověření funkčnosti

Aby nedošlo při manipulaci k poškození nebo k odpojení kabelů, byla pomocí 3D tiskárny vyrobena krabička, do níž bylo celé zařízení vloženo. Návrh krabičky i fotografie jsou umístěny v [příloze](#).



## 6 Softwarová část zařízení

V této kapitole si popíšeme programovatelnou část zařízení, je zde popsán skript sloužící pro sběr dat a ovládání samotného měření, včetně konfigurace webového serveru *Flask*. Pro hlavní skript byl použit programovací jazyk Python, webový dashboard byl vytvořen pomocí HTML a JavaScriptu.

### 6.1 Aktivace potřebných rozhraní na GPIO

Jelikož budeme využívat rozhraní I2C, je nutné jeho použití nejdříve povolit. Toto lze snadno provést pomocí programu *raspi-config*. Připojíme se na Raspberry Pi (buď napřímo nebo přes SSH) a v terminálu zadáme příkaz:

```
sudo raspi-config
```

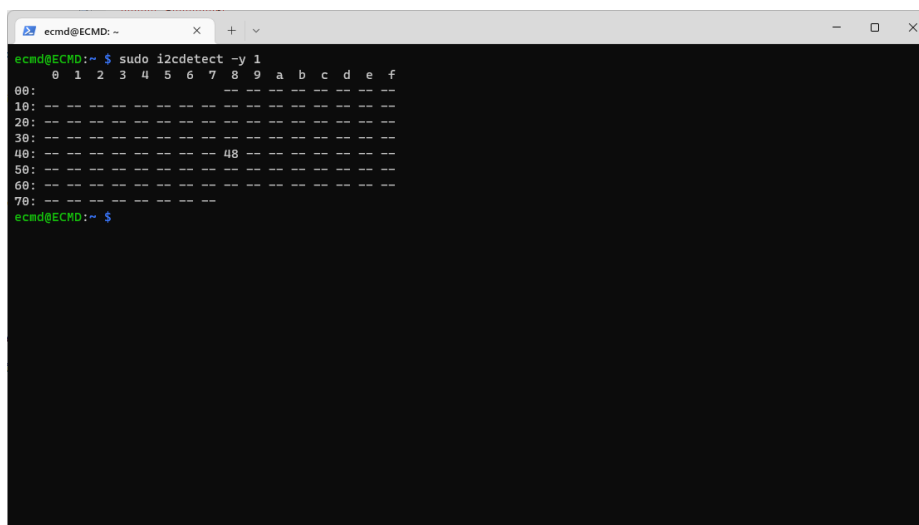
Otevře se nám stejnojmenný program a v úvodní nabídce zvolíme možnost *Interfacing options*, dále *I2C* a pouze potvrdíme zvolením možnosti *Yes*. Program *Raspi config* opustíme pomocí *Finish* a v případě potřeby provedeme restart zařízení. Tímto je rozhraní I2C aktivováno, ověřit pak můžeme pomocí příkazu:

```
sudo raspi-config nonint get_i2c
```

Pokud je výsledná hodnota 0, pak je I2C aktivní, pokud je výsledkem příkazu 1, pak I2C aktivní není. Ověřit detekci a úspěšné připojení senzoru můžeme zjistit příkazem:

```
sudo i2cdetect -y 1
```

Máme-li vše správně připojeno, měli bychom dostat následující výsledek:



```
ecmd@ECMD: ~ $ sudo i2cdetect -y 1
0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00: -- -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- 48 -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- -- -- -- -- -- -- --
ecmd@ECMD: ~ $
```

Obrázek 13: Detekce připojeného senzoru na I2C

## 6.2 Hlavní skript pro zpracování dat

Hlavním skriptem pro měření a odeslání dat do databáze včetně funkčnosti webového serveru je soubor *ecmd\_service.py*. Skript je automaticky spouštěn po spuštění Raspberry Pi díky službě *systemd*. Ve stejné složce se pak nachází pomocný konfigurační skript *ecmd\_conf.py*, který obsahuje hodnoty  $VCC_0$  a *sensitivity*, dále IP adresu (příp. DNS jméno) databáze InfluxDB, konfiguraci GPIO portů pro LED diody a relé a číslo verze webového dashboardu. Jedná se o hodnoty, u kterých by mohlo v budoucnu dojít ke změně (např. výměna senzoru ACS712 za 20A typ<sup>28</sup>, změna IP adresy InfluxDB atd.) a díky tomu bude tato změna snadno proveditelná.

### 6.2.1 Inicializace programu

Po spuštění skriptu dojde k inicializaci globálních proměnných (pro jednoduchý způsob komunikace mezi vlákny) a vytvoření dvou vláken (zajišťuje knihovna *Threading*) – jedno pro samotné měření a druhé pro integrovaný webový server. Vytvoření vláken v Pythonu je velice jednoduché, pokud máme naimportován modul *Thread* z knihovny *Threading*, stačí nám pro běh vlákna dva řádky kódu (v našem případě to budou čtyři řádky, neboť máme vlákna dvě).

```
1 thread_server = Thread(target=start_web_server)
2 thread_measurement = Thread(target=current_measurement)
3 thread_server.start()
4 thread_measurement.start()
```

Zdrojový kód 1: Vytvoření vláken pro webový server a měření proudu

### 6.2.2 Měření proudu a sběr dat

Tím, že nám měření proudu a webový server běží ve dvou samostatných vláknech, máme umožněno snadné vypnutí a zapnutí měření, které je uživateli zprostředkováno přes webový dashboard běžící na integrovaném webovém serveru.

V minulé kapitole jsem popisoval správnou kalibraci senzorů, nyní tedy použijeme obě konstanty potřebné pro výpočet proudu. Abychom ale výpočet ještě o něco zpřesnili, tak hodnota napětí ze senzoru ACS712 je počítána jako průměr sta nasbíraných hodnot. Díky tomu odbouráme drobné odchylky, které mohou nastat v důsledku působení magnetického pole. Takto získanou hodnotu napětí pak předáme funkci, která nám vypočítá finální hodnotu aktuálního proudu, kterou ještě zaokrouhlíme na tři desetinná místa. Tento proces se opakuje pro každou vypočítanou hodnotu proudu.

---

<sup>28</sup>Změna za 20A typ si vyžádá novou hodnotu *sensitivity*.

```

1 def collect_voltage_avg():
2     limit = 100
3     voltage = 0
4     for x in range(limit):
5         voltage = voltage + chan.voltage
6     voltage = voltage / limit
7     voltage = round(voltage, 3)
8     return voltage

```

Zdrojový kód 2: Mezivýpočet – průměrná hodnota napětí

```

1 def count_current(voltage_value):
2     current = (voltage_value - ecmd_conf.vcc_zero) / ecmd_conf.
3         sensitivity
4
5     if current < 0:
6         current = abs(current)
7
8     current = round(current, 3)
9     return current

```

Zdrojový kód 3: Výpočet finální hodnoty proudu

Vypočtenou hodnotu pak již stačí jen poslat do databáze InfluxDB. Na tu se musíme nejdříve připojit (proměnnou *dbclient* s konfigurací připojení lze používat v celém programu a v našem případě se jedná o globální proměnnou). Pokud je InfluxDB nedostupná, je implicitně nastaven timeout spojení na 20 vteřin. Poté dojde k vyvolání výjimky a zařízení přejde do stavu „Vyžaduje pozornost“, během této doby nedochází ke sběru dat.

Kalibrované hodnoty *vcc\_zero* a *sensitivity* jsou umístěny v souboru *ecmd\_conf.py* (proto i stejnojmenná přípona v kódu), aby byla jejich úprava snadná a rychlá, pokud by došlo k náhradě senzoru ACS712 za 20A verzi.

I přesto, že byly senzory kalibrovány a je použit „filtr“, který zprůměruje sto hodnot napětí, může docházet k nesrovnalostem v hodnotách proudu. Je to dáno působením vnějšího magnetického pole a výkyvy v elektrické síti.<sup>29</sup> Tyto nesrovnalosti se pak nejčastěji projevují při nižších hodnotách proudu, u vyšších hodnot by mělo dojít k částečné stabilizaci.

---

<sup>29</sup>Měření bylo testováno na několika různých místech a podle napětí v elektrické síti se výsledky mírně lišily.



Připojení k databázi InfluxDB a odeslání dat:

```
1 dbclient = InfluxDBClient(host=ecmd_conf.influxdb_addr, port=8086,
2   username="raspi", password="ecmddevice1+*")
3 data = []
4 data.append("{measurement} value={current}".format(measurement=
5   measurement_name, current=current_value))
6 try:
7   dbclient.write_points(data, database="ecmddb", time_precision=
8     "ms", batch_size=1, protocol="line")
9 except:
10  now = datetime.datetime.now()
11  print(now.strftime("[%Y-%m-%d %H:%M:%S]")+ " --- InfluxDB error
12    .")
```

Zdrojový kód 4: Připojení na databázi a odeslání dat

### 6.2.3 Výpočet průměru

Součástí funkce pro měření je i výpočet průměrné hodnoty proudu, kterou si může uživatel zobrazit pomocí webového dashboardu. Průměr je počítán z posledních 50 hodnot proudu. Pokud je průměr vynulován nebo je zařízení restartováno či nově zapnuto, chvíli trvá, než se aktuální hodnota zobrazí.

```
1 def count_average_current(current):
2   global avg_values_limit
3   global avg_current
4   global avg_value
5   if avg_values_limit < 50:
6     avg_lock.acquire()
7     avg_current_values.append(current)
8     avg_current += current
9     avg_values_limit += 1
10    avg_lock.release()
11  else:
12    avg_lock.acquire()
13    average = avg_current / (avg_values_limit + 1)
14    avg_value = round(average, 3)
15    avg_current -= avg_current_values[0]
16    avg_current_values.pop(0)
17    avg_current_values.append(current)
18    avg_current += current
19    avg_lock.release()
20  return
```

Zdrojový kód 5: Výpočet průměrné hodnoty proudu

Průměr je počítán v proměnné *avg\_current*, do které je vždy přičtena aktuální hodnota proudu. Abychom však věděli, kterou hodnotu musíme odečíst, aby bylo zajištěno, že je průměr počítán z posledních 50 hodnot proudu, máme k dispozici pole *avg\_current\_values*, které těchto 50 hodnot uchovává. „Nejstarší“ hodnota v poli je z průměru odečtena a vyřazena z pole. Naopak každá nová hodnota je přidána na konec pole a přičtena k průměru. Tímto máme ušetřen průchod celým polem, kdy bychom museli hodnoty sečíst a průměr vypočítat, a zároveň máme informaci, které hodnoty je třeba z průměru vyřadit, respektive odečíst.

Jelikož může uživatel přes webový dasbhoard průměr vynulovat (což provádí jiné vlákno), jsou veškeré hodnoty související s výpočtem průměru uzavřeny do kritické sekce, aby nedošlo k nesprávným hodnotám průměrného proudu.

### 6.3 Webový server Flask

Integrovaný webový server je postaven na microframeworku Flask. Jedná se o tzv. „python light-weight“ webový server. Flask nám mj. umožňuje vytvořit si vlastní routy pro definování adres pro každou stránku, případně routy pro zavolání specifických funkcí (použito v tomto případě). Jelikož ale Flask sám o sobě neumí zpracovat více požadavků najednou, je potřeba pro produkční nasazení zvolit některý z WSGI serverů.<sup>30</sup> V našem případě jsem zvolil *Gevent*, který zvládne zpracovat více požadavků najednou. Spuštění webového serveru je pak velmi jednoduché:

```
1 if __name__ == "__main__":
2     http_server = WSGIServer(('0.0.0.0', 5000), app)
3     http_server.serve_forever()
```

Zdrojový kód 6: Spuštění webového serveru pomocí Geventu

Webový server tímto běží na IP adrese, kterou má naše Raspberry Pi na portu 5000. Webové aplikace na HTTP ale běží na portu 80, případně HTTPS na portu 443. Spuštění našeho serveru na některém z těchto „well known“ portů by znamenalo, že aplikaci *ecmd\_service.py* musíme spustit s oprávněním správce. To by nemuselo být úplně žádoucí. Existuje však jednoduchý způsob, jak tento problém obejít – firewall forwarding. Do Raspberry Pi nainstalujeme balíček *firewalld*, díky tomu budeme mít k dispozici příkaz, známý z Red Hat distribucí Linuxu, *firewall-cmd* a pomocí následujících příkazů vytvoříme port forwarding z portu 80 na port 5000. Uživatel tak při přístupu na webový dashboard zadá pouze IP adresu (příp. DNS jméno) a nepotřebuje k tomu informaci, na kterém portu naše aplikace běží.

---

<sup>30</sup>Web Server Gateway Interface – univerzální rozhraní mezi webovým serverem a webovou aplikací.

```
1 firewall-cmd --add-port=5000/tcp
2 firewall-cmd --add-forward-port=port=80:proto=tcp:toport=5000
3 firewall-cmd --runtime-to-permanent
4 firewall-cmd --reload
```

Zdrojový kód 7: Nastavení port forwardingu

### 6.3.1 Obsah webové aplikace

Webová aplikace sestává ze dvou HTML souborů – hlavní soubor *index.html* a nápověda *help.html*. V nápovědě se uživatel může dozvědět o všech funkcích webové aplikace, včetně jejich použití a jak správně chápat stavy zařízení. Hlavní stránka pak nabízí uživateli možnost zjistit stav zařízení, průměrnou hodnotu proudu, nastavit e-mailovou adresu a limitní hodnotu pro zaslání e-mailových notifikací a vypnout/zapnout měření či přímo měřené zařízení.

Zařízení se může nacházet celkem ve čtyřech stavech:

- Vše v pořádku – označen ●.
- Vyžaduje pozornost – označen ●. Tento stav nastane, pokud je měření zapnuto, ale hodnoty proudu jsou menší než 0,05 A nebo je nedostupná databáze InfluxDB. Vypnutí a opětovné zapnutí měření tento stav resetuje do stavu OK.
- Měření vypnuto – označen ●. Vlákno s měřením bylo ukončeno, sběr dat neprobíhá. Lze opět zapnout.
- Zařízení vypnuto – označen ●. Vypnutí zařízení automaticky zastaví i měření. Po zapnutí zařízení je měření opět automaticky zapnuto.

Aktualizace stavu, včetně aktuálně nastavených hodnot limitu a e-mailové adresy je obnovována každé dvě vteřiny. O obnovení se stará funkce v JavaScriptu (s pomocí knihovny jQuery), která pošle požadavek na server, ten je zpracován a je zavolána funkce odpovídající pro danou routu. Obnovení průměru je nastaveno na každé tři vteřiny. Změna stavu měření a zařízení pak vyvolá okamžité obnovení hodnot, aby nedošlo k situaci, kdy například zařízení bylo vypnuto, ale na krátký čas se ještě webová stránka jeví, jako by zařízení zůstalo zapnuté. Rovněž obě funkce jsou implementovány v JavaScriptu a předávají požadavek na server. JavaScript byl využit z důvodu, aby při každé změně hodnot nedocházelo k obnovení celé stránky, ale jen ke změně určitých částí stránky. Nicméně i Flask umožňuje předání hodnot do šablony, aby ale došlo k jejich obnovení či změně, je potřeba obnovit celou stránku, což by pro uživatele nemuselo být přívětivé řešení.

Pro stylování byla použita knihovna Bootstrap verze 5.2, díky tomu je webová stránka optimalizována i pro mobilní zařízení.

```

1  setInterval(function () {
2      $.ajax({
3          type: "POST",
4          url: "/count_avg",
5          data: {},
6          success: function (response) {
7              $("#average").html(response)
8          },
9      });
10 }, 3000);

```

Zdrojový kód 8: Ukázka JavaScriptové funkce pro obnovení průměrné hodnoty

### 6.3.2 Použití lokální databáze

Kromě databáze InfluxDB, která je umístěna na serveru, je použita lokální databáze SQLite, která slouží pouze pro uchování konfigurace webového dashboardu. Existuje v ní pouze jediná tabulka, pomocí které se řídí e-mailové notifikace. Tabulka obsahuje atributy id,<sup>31</sup> limitní hodnotou, e-mailovou adresou a časovým razítkem posledního odeslání e-mailové notifikace. Při připojení na databázi je provedena i inicializace databáze (provede se pouze jednou díky klauzulím *NOT EXIST* a *OR IGNORE*).

```

1  def local_db_init():
2      try:
3          con = sqlite3.connect("ecmd_local.db", check_same_thread=False
4              )
5          cur = con.cursor()
6          cur.execute(
7              "CREATE TABLE IF NOT EXISTS notify_conf (id INTEGER PRIMARY
8                  KEY, notify_limit REAL DEFAULT 0, last_send INTEGER
9                  DEFAULT 0, email_address TEXT)")
10         cur.execute(
11             "INSERT OR IGNORE INTO notify_conf (id, notify_limit,
12                 last_send, email_address) VALUES (1, 5, 0, NULL)")
13         con.commit()
14     except:
15         now = datetime.datetime.now()
16         print(now.strftime("[%Y-%m-%d %H:%M:%S]")+ " --- Error during
17             local DB initialization.")
18         return -1
19     return con

```

Zdrojový kód 9: Inicializace lokální SQLite databáze

---

<sup>31</sup>Atribut id je v databázi jen jako funkce primárního klíče.

Při inicializaci databáze je vytvořena tabulka *notify\_conf* a vložen jeden řádek, kde limitní hodnota je 5 A, tj. maximální možná hodnota použitého typu senzoru ACS712. Uživatel si může nastavit i vyšší limit, ale je nutno brát na vědomí, že při vyšších hodnotách proudu dojde k poškození senzoru ACS712. E-mailová adresa zůstává *NULL*. Pro správné fungování musí uživatel zadat existující e-mailovou adresu. To lze provést přes webové rozhraní.

Při aktualizaci limitu nebo e-mailové adresy je brán vstup od uživatele. Aby nedošlo k útoku SQL injection použil jsem speciální formulaci pro SQLite, která by měla takovému útoku předejít. Zároveň je pomocí regulárního výrazu ověřováno, že uživatel zadal validní e-mailovou adresu. V opačném případě dojde k vyvolání výjimky a zadaná adresa není uložena.

```
1 @app.route("/set_email", methods=["POST"])
2     def change_email():
3         if request.method == "POST":
4             data = str(request.get_data(), "UTF-8")
5             try:
6                 regex = r'\b[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Z|a-z]{2,}\b'
7                 if re.fullmatch(regex, data):
8                     cur = local_db.cursor()
9                     cur.execute("UPDATE notify_conf SET email_address = ?
10                                WHERE id = 1", (data,))
11                     local_db.commit()
12                 else:
13                     raise ValueError
14             except:
15                 now = datetime.datetime.now()
16                 print(now.strftime("[%Y-%m-%d %H:%M:%S]")+ " --- Error.
17                        Can't save changes.")
18                 return {'result': -1}
19             return {'result': 0}
```

Zdrojový kód 10: Aktualizace údajů v SQLite databázi

Závěrem dodejme, že i když databázový soubor při spuštění programu neexistuje, je při inicializaci díky Python knihovně *sqlite3* automaticky vytvořen.

### 6.3.3 Odesílání e-mailových notifikací

E-mailová notifikace je zaslána na zadaný e-mail, pokud dojde k překročení nastaveného limitu. V případě, že je limit stále překročen, je e-mailová notifikace zaslána každé dvě minuty. Po každém odeslání je uložena aktuální hodnota časového razítka a uložena do databáze, poté je pouze kontrolováno, zdali je tato hodnota starší než dvě minuty. Pokud ano, je odeslána nová notifikace a celý proces se opakuje.

Knihovna Pythonu pro odesílání e-mailů se jmenuje *smtplib*. Abychom mohli zprávy odesílat je nutné nakonfigurovat SMTP server. Pro testovací účely a účely

obhajoby byl zvolen UPOL SMTP server, který ve vnitřní síti univerzity nevyžaduje ověřování.

S použitím knihovny *smtplib* tedy konfigurace vypadá následovně:

```
1     try:
2         msg = EmailMessage()
3         text = "ECMD varování \n\nZaznamenána nadlimitní hodnota
           proudu. Prosím zkontrolujte ihned své zařízení. \n\nNaměř
           ená hodnota proudů: {curr:} A. \n\nToto je automaticky
           generovaná zpráva, neodpovídejte na ni."
4         msg.set_content(text.format(curr=current_value))
5         msg["Subject"] = "ECMD varování"
6         msg["From"] = "ECMD <martin.urbanec01@upol.cz>"
7         msg["To"] = result[0]
8         server = smtplib.SMTP("smtp.upol.cz", 25, None, 3)
9         server.send_message(msg)
10        server.quit()
11        now = datetime.datetime.now()
12        print(now.strftime("[%Y-%m-%d %H:%M:%S]") + " --- Email
           notification sent.")
13        timestamp = time.time()
14        cur.execute("UPDATE notify_conf SET last_send = ? WHERE id = 1
           ", (timestamp,))
15        local_db.commit()
16    except:
17        now = datetime.datetime.now()
18        print(now.strftime("[%Y-%m-%d %H:%M:%S]") + " --- SMTP or
           local DB error.")
```

Zdrojový kód 11: Nastavení SMTP serveru a odeslání zprávy

Funkce nejdříve odešle e-mailovou zprávu a poté je uloženo časové razítko odeslání. Pokud by došlo k problému s připojením na SMTP server nebo k lokální databázi, dojde k vyvolání výjimky.

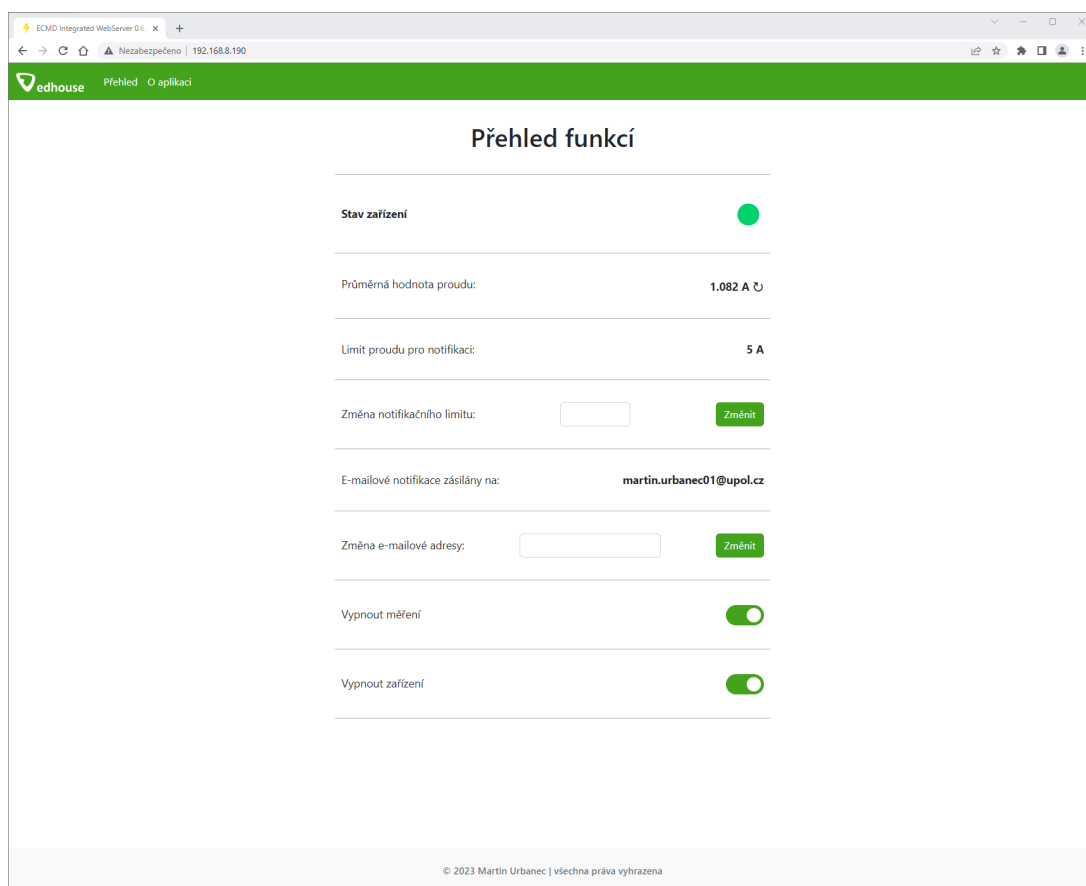
Pro použití SMTP serveru s ověřováním by změna vypadala takto:

```
1 server = smtplib.SMTP_SSL("smtp.example.com", 465)
2 server.login("username", "password")
```

Zdrojový kód 12: Připojení na SMTP server s ověřováním

### 6.3.4 Vzhled webového rozhraní

Následující snímek obrazovky zachycuje, jak vypadá webové rozhraní webového dashboardu.



Obrázek 14: Hlavní stránka webového dashboardu

Webový dashboard neposkytuje žádnou autentizaci uživatelů, jelikož předpokládané použití je v oddělené VLAN síti, do které mají přístup pouze IT správci firmy Edhouse. Její použití může být v budoucnu zavedeno, neboť Flask s takovým řešením počítá a nejednalo by se o velké změny v kódu.

Jako ikona adresáře webu byla použita ikona „Lightning“ z webu *Flaticon*. Její použití zdarma je podmíněno uvedením zdroje, který je uveden v seznamu literatury.



## 7 Vyhodnocení dat

Tato kapitola pojednává o vyhodnocení nasbíraných dat. Je zde popsáno nastavení databáze InfluxDB a webové aplikace Grafana, pomocí které si poté budeme moct zobrazit grafy z nasbíraných hodnot.

### 7.1 Databáze InfluxDB

InfluxDB je open-source databází, která je vhodná pro uložení časově označených dat a která poskytuje jazyk podobný SQL. Nejedná se však o relační databázi. Pro nastavení nám slouží CLI rozhraní (od verze 2 i rozhraní webové), které je nám zpřístupněno pomocí programu *influx*. Zde si můžeme zobrazit seznam databází nebo uživatelů a zvolit, kterou databázi chceme používat, případně vytvářet databáze a uživatele. Můžeme využít klasických příkazů jako *show databases*, *show users*, *use <db-name>* atd. Ovšem zde je potřeba brát v úvahu, že zmíněné příkazy se týkají verze 1.8 a nižší. Od verze 2 došlo k poměrně rozsáhlým úpravám v syntaxi a v době psaní této práce bylo poměrně obtížné najít dobře zdokumentované postupy týkající se verze 2.x, proto jsem použil verzi 1.8.10 (poslední verze InfluxDB 1.x), která disponuje dobrou dokumentací. Samotná databáze pak běží na portu 8086.

#### 7.1.1 Nastavení InfluxDB

Jednotlivé příkazy pro nastavení InfluxDB jsou uvedeny ve [zdrojovém kódu níže](#). Abychom mohli zaznamenávat data, je potřeba nejdříve vytvořit databázi (při použití s Dockerem je možné název databáze rovnou uvést do YAML souboru pro Docker Compose), vytvoření databáze ručně je možné prvním příkazem v kódu. Pro použití konkrétní databáze slouží druhý příkaz. Dále je potřeba vytvořit tzv. „měření“, též označované jako „measurement“<sup>32</sup>, ovšem měření nemůžeme vytvořit ručně, ale je vytvořeno automaticky s přidáním prvního záznamu ke konkrétnímu měření, tuto situaci popisuje třetí příkaz v kódu. Naopak smazání měření můžeme provést ručně a to čtvrtým uvedeným příkazem. Ve výchozím stavu nemáme v databázi žádnou autentizaci. Toto může být v některých případech nežádoucí, proto nejdříve vytvoříme uživatele (může jich být i více) a postupně jim nastavíme jednotlivá práva. Máme na výběr z několika možností:

- Nastavit uživateli administrátorská práva (má právo Read/Write na všechny databáze).
- Nastavit uživateli Read/Write práva na konkrétní databázi.
- Nastavit uživateli Read práva na konkrétní databázi.

Následně si můžeme vše ověřit zbylými příkazy pro výpis všech databází, měření, uživatelů a zobrazení výsledků.

---

<sup>32</sup>V relačních databázích může být „měření“ analogií k tabulkám.

```

1 CREATE DATABASE ecmdbb
2 USE ecmdbb
3 INSERT current value=0
4 DROP MEASUREMENT current
5 CREATE USER admin WITH PASSWORD <password> WITH ALL PRIVILEGES
6 CREATE USER user WITH PASSWORD <password>
7 GRANT [READ,WRITE,ALL] ON <database_name> TO <username>
8 REVOKE [READ,WRITE,ALL] ON <database_name> FROM <username>
9 SHOW DATABASES
10 SHOW MEASUREMENTS
11 SHOW USERS
12 SELECT * FROM current
13 SELECT * FROM current WHERE value > 0.050

```

Zdrojový kód 13: Prvotní nastavení databáze InfluxDB

Pro zapnutí autentizace je nutné ještě provést změny v konfiguračním souboru *influxdb.conf*, kde přidáme následující blok:

```

1 [http]
2   enabled = true
3   bind-address = ":8086"
4   auth-enabled = true
5   log-enabled = true
6   write-tracing = true

```

Zdrojový kód 14: Povolení autentizace v InfluxDB

Autentizaci v CLI *influx* vyvoláme příkazem *auth*.

## 7.2 Webová aplikace Grafana

Grafana je mutliplatformní aplikace pro vizualizaci dat a jejich analýzu. Umožňuje nám vizualizovat data do přehledných grafů, kde si můžeme nastavit velké množství parametrů (jako např. prahové hodnoty, barvy, popisky atd.) a zároveň můžeme zobrazit data v konkrétních časových úsecích.

Pro inicializaci aplikace Grafana pak stačí přejít na její webové rozhraní a pro přihlášení použít následující údaje:

- Přihlašovací jméno: admin
- Heslo: admin

Po přihlášení dojde k výzvě pro změnu hesla.

Jelikož uživatel *admin* má veškerá práva na úpravy, vytvořil jsem uživatele *guest*, který může grafy prohlížet, nikoliv však upravovat. Zobrazení grafů je dostupné až po přihlášení uživatele a není žádoucí, aby mohl kdokoliv nastavení

grafů upravovat. Uživatel *guest* si však může zvolit vlastní časový úsek zobrazovaných dat.

### 7.2.1 Zdroj dat

Abychom mohli do Grafany přijímat data, musíme nejprve nastavit zdroj dat. Grafanu můžeme propojit s různými typy databází, v našem případě zvolíme InfluxDB. Jazyk dotazů ponecháme *InfluxQL*<sup>33</sup> Dále nastavíme URL pro naši databázi a jelikož Grafana i InfluxDB běží v Dockeru ve stejném kontejneru, postačí pouze *http://influxdb:8086*.<sup>34</sup> Protože máme databázi zabezpečenou pomocí autentizace, doplníme ještě přihlašovací údaje.

Důležitou věcí pro připojení databáze je, aby v ní existovalo alespoň jedno měření. Z kapitoly výše víme, že pro vytvoření měření je potřeba alespoň jeden záznam pro dané měření. Po uložení Grafana otestuje připojení a je-li nalezeno alespoň jedno měření, je připojení úspěšně nastaveno.

### 7.2.2 Vytvoření grafů

Grafana nám nabízí širokou škálu grafů – např. klasický spojnicový a sloupcový graf, ale také graf podobný měřidlu. Dále to mohou být histogramy, statistiky nebo tzv. geomapa<sup>35</sup> atd. Pro naše potřeby jsem použil první tři zmíněné typy grafů.

Než ale přistoupíme k samotnému vytváření grafů, musíme si vytvořit vlastní dashboard, který bude grafy obsahovat. Ve vytvořeném dashboardu zvolíme možnost „Přidat panel“ a dále „Přidat nový panel“.

Nyní už můžeme začít s nastavováním grafu. Zvolíme, které hodnoty a z jakého měření z připojeného zdroje dat chceme vyobrazit. Data budeme seskupovat podle času, konkrétně pomocí parametru *\$\_interval*, což nám zajistí, že budeme mít vyobrazenu každou hodnotu z databáze (toto bude využito pro spojnicový graf). Pro druhý typ grafu ale použijeme seskupení podle hodinového časového úseku. Výsledkem budou průměrné hodnoty proudu z každé hodiny, kdy bylo měření zapnuté (zde jsem zvolil graf sloupcový). Pro poslední graf, tedy měřidlo, se bere maximální naměřená hodnota v daném časovém úseku (ten si zvolí uživatel sám) vůči maximální možné hodnotě (v našem případě 5 A).

Při nastavování grafů máme k dispozici velké množství nástrojů pro jejich vizualizaci (ať už textovou či grafickou). Jako velmi pěknou a užitečnou funkci hodnotím možnost nastavit si tzv. „prahové hodnoty“ a oddělit je barevně od ostatních hodnot. Jinými slovy pokud dojde k překročení prahové hodnoty, v grafu tyto překročené hodnoty uvidíme jinou barvou.

<sup>33</sup>InfluxQL je podobný SQL a pro naše potřeby zcela dostačující. Pro InfluxDB verze 1.8.x a 2.x existuje ještě jazyk Flux, který umožňuje pokročilejší dotazy a skriptování.

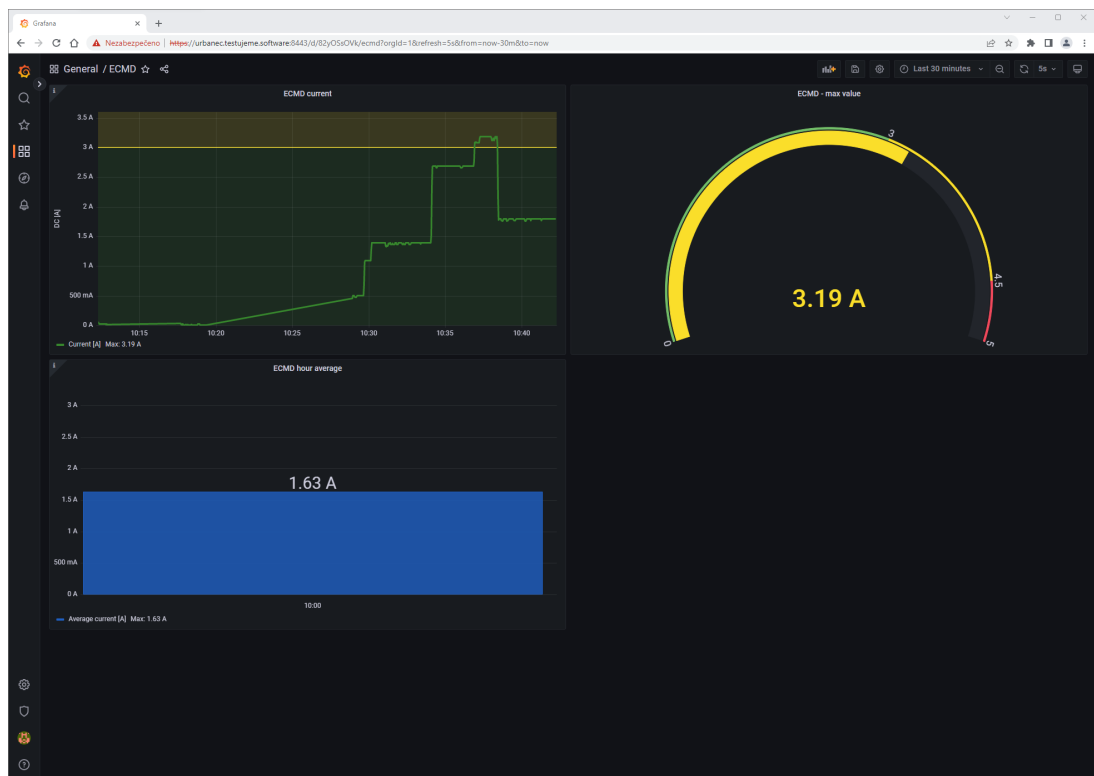
<sup>34</sup>Poznamenejme, že *influxdb* je název kontejneru v Dockeru.

<sup>35</sup>Pěkným příkladem využití tohoto typu grafu bylo v době pandemie COVID-19, kdy jsme byli schopni zaznamenat kolik případů v jaké zemi je a následně tyto údaje zobrazit do mapy světa.

Nastavíme tedy:

- Zelenou barvu jako výchozí.
- První prahovou hodnotu na 3 A a zvolíme žlutou barvu. Všechny hodnoty proudu od 3 A budou označeny žlutě.
- Druhou prahovou hodnotu na 4,5 A, kterou označíme červenou barvou. Všechny hodnoty nad 4,5 A budou označeny červeně.

Grafy s daty pak vypadají následovně:



Obrázek 15: Grafana – dashboard s grafy

## Závěr

Dozvěděli jsme se něco o problematice fyzického zabezpečení datových center, včetně uvedení konkrétních příkladů použití v praxi. Navíc jsme si také prošli a popsalí všechny úrovně standardu Tier.

Máme úspěšně sestavené plně funkční zařízení pro měření proudu na platformě Raspberry Pi. Interně jsem jej nazval jako *ECMD*.<sup>36</sup> Zařízení obsahuje webový dashboard na integrovaném webovém serveru, postaveném na mikroframeworku Flask, kde může uživatel provádět jeho základní obsluhu – zapnout/-vypnout měření, zapnout/vypnout měřené zařízení, spravovat nastavení zasílání e-mailových notifikací, tj. nastavení limitní hodnoty a také e-mailové adresy. Sbírané hodnoty proudu jsou odesílány do databáze InfluxDB a přes webovou aplikaci Grafana jsme schopni je zobrazit v přehledných grafech, které si navíc můžeme libovolně přizpůsobit podle našeho uvážení. Obě aplikace běží na externím serveru v Dockeru. Původní záměr, kdy by vše bylo integrováno pouze na Raspberry Pi, by nebyl šťastně zvolený. Raspberry Pi používá jako paměť SD kartu, jejíž životnost by byla z důvodu velkého počtu zápisů zkrácena.

Použitý senzor ACS712 pak není zcela vhodný pro přesné měření, neboť jeho odchylka je vyšší, než jsem očekával. Velmi záleží na poloze senzoru, jelikož může být ovlivňován okolním mangetickým polem a dále také na kolísání napětí v elektrické síti. Pro orientační měření proudu je ale takové řešení dostatečné.

---

<sup>36</sup>Electric Current Measurement Device.

## Conclusions

We learned about the problems of physical security in data centers, including concrete examples of how it is used in practice. In addition, we also went through and described all the tiers of the Tier standard.

We have successfully built a fully functional current measurement device on the Raspberry Pi platform. Internally, I have named it as *ECMD*.<sup>37</sup> The device contains a web-based dashboard on an integrated web server, built on the Flask microframework, where the user can perform its basic operation – enable/disable measurements, enable/disable the measured device, manage the settings for sending email notifications, i.e. setting the limit value and also the email addresses. Collected current values are sent to the InfluxDB database and via the Grafana web application we are able to display them in clear graphs, which we can also customize at our discretion. Both applications run on an external server in Docker. The original intention of having everything integrated only on the Raspberry Pi would not have been a happy choice. The Raspberry Pi uses an SD card as memory, the lifetime of which would be shortened due to the large number of writes.

The ACS712 sensor used is then not quite suitable for accurate measurement, as its deviation is higher than I expected. The position of the sensor is very important, as it can be affected by the surrounding magnetic field and also by voltage fluctuations in the electrical network. However, for indicative current measurements this solution is sufficient.

---

<sup>37</sup>Electric Current Measurement Device.

## A Fotografie z datového centra Internextu

Následující obrázek znázorňuje energetické vedení v datovém centru společnosti Internext. Lze na něm spatřit skříň s *ATS*<sup>38</sup> stanicí a dále skříňe s větvemi *A* a *B*. Na vnější části je pak zobrazeno schéma celého obvodu. Napravo těchto skříní je možné vidět ještě skříň s UPS, jejíž celková hmotnost, včetně baterií, dosahuje dvou tun.

Jako malý bonus je na této fotografii zachycen sloup s fotografiemi prostoru v době výstavby datového centra.



Obrázek 16: Energetické vedení

---

<sup>38</sup>Automatic Transfer Switches – v případě výpadku jedné napájecí větve, je ihned přepojeno na sekundární napájecí větev



Na dalším obrázku je vstup do tzv. horké uličky, kde se teplota vzduchu pohybuje okolo 32 °C.<sup>39</sup> Celý prostor horké uličky byl oddělen od prostoru chladné uličky a horký vzduch odsud byl nasáván vzduchoventilací a následně ochlazován.



Obrázek 17: Vstup do horké uličky

---

<sup>39</sup>Ve zbytku místnosti (chladná ulička) se teplota vzduchu pohybovala okolo 24 °C.

Poslední obrázek obsahuje tlakovou nádobu se speciálním hasivem *NOVEC 1230*. Za zmínku stojí i potrubí, kterým je hasivo dopravováno k serverům. Toto potrubí musí vydržet obrovský nápor tlaku, neboť v případě požáru je celá nádoba vyprázdněna během pár sekund.



Obrázek 18: Hasicí tlaková nádoba

## B Fotografie ze serverovny Edhouse

Na fotografii jsou zachyceny servery, na nichž jsou provozovány interní systémy firmy Edhouse virtualizované pomocí virtualizačních platforem oVirt a VMware ESXi. Některé ze zachycených serverů jsou dočasně mimo provoz a plánuje se jejich další využití. Současně je na obrázku zachycen hlavní router, který je spojen přímou linkou s datovým centrem Internext. Pod ním je možné vidět dva switche, které jsou v zapojení typu MLAG<sup>40</sup> a pro ostatní switche se tváří jako jedno zařízení. Jde o formu redundance v případě poruchy jednoho ze switchů. Úplně dole pod všemi servery se pak nachází UPS stanice, které zajistí bezpečné vypnutí serverů v případě výpadku elektřiny.



Obrázek 19: Servery pro provoz interních systémů a síťové prvky

---

<sup>40</sup>Mutli-chassis Link Aggregation Group



Další fotografie zachycuje zapojení ethernet kabelů vedoucích ze switchů, kterých je celkem osm,<sup>41</sup> do patch panelů, které následně rozvádějí konektivitu po celé firmě.

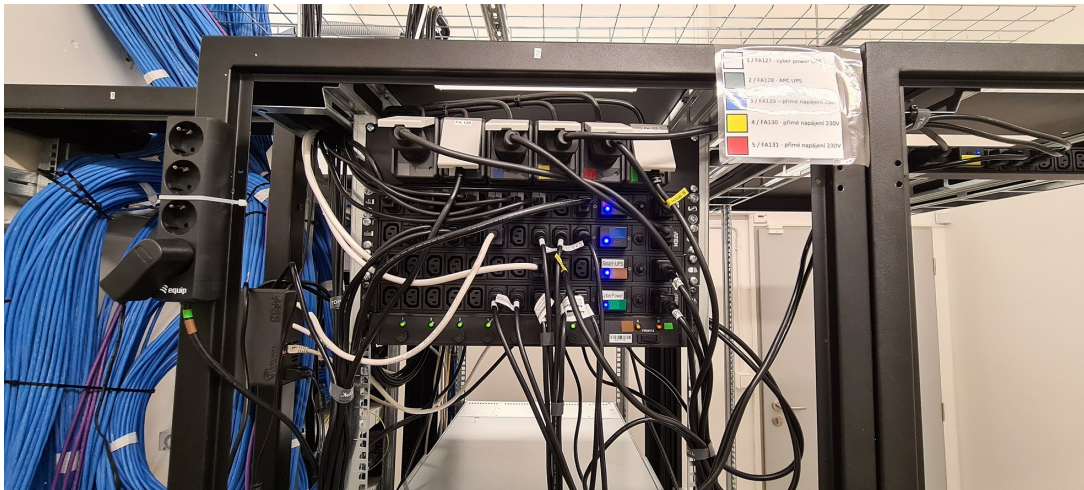


Obrázek 20: Rozvod připojení k síti

---

<sup>41</sup>Dva switche pak mají podporu PoE (Power over Ethernet) a napájí například docházkové terminály a přístupové body WiFi.

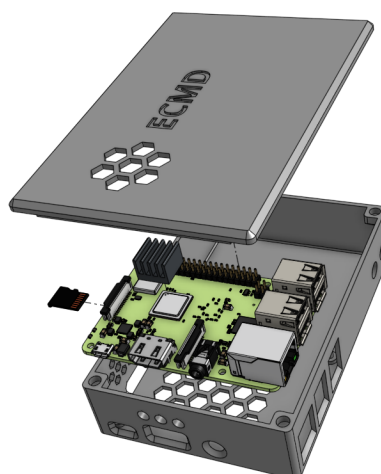
Následující obrázek pak ukazuje zapojení napájení. Každé zařízení má dvě napájecí větve – primární ze sítě a v případě výpadku přijde na řadu UPS, která zajistí bezpečné vypnutí všech serverů.



Obrázek 21: Napájení serverů

## C Výsledné zařízení ECMD

V této příloze je možné prohlédnout si obrázky výsledného zařízení. Raspberry Pi i všechny ostatní komponenty jsou umístěny v krabičce z nehořlavého materiálu, která byla vyrobena pomocí 3D tiskárny a za její výroby děkuji Mgr. Pavlu Stržínkovi. Příloha obsahuje jak návrh krabičky, tak i fotografie fyzického výsledného zařízení.



Obrázek 22: Návrh krabičky – vnitřní část



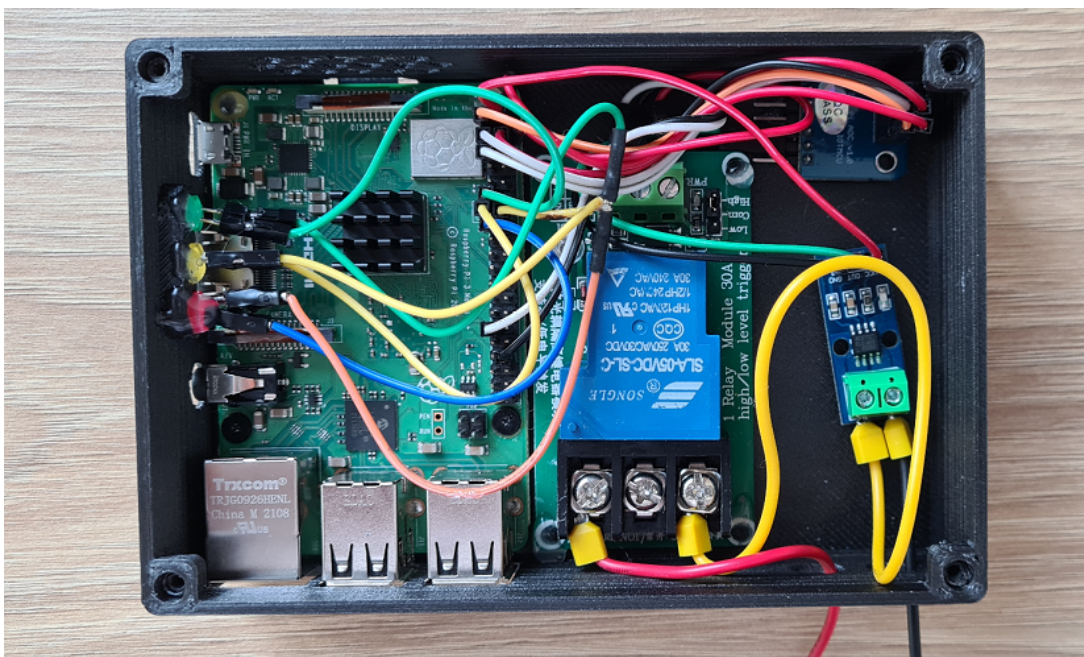
Obrázek 23: Návrh krabičky – vnější část

Návrh počítal s použitím neodymových magnetů pro přichycení víka. Takto silný magnet v blízkosti senzoru ACS712 by mohl zásadně ovlivnit měření proudu, a proto byly k přichycení víka použity 4 šroubky.





Obrázek 24: Výsledné zařízení – vnější pohled



Obrázek 25: Výsledné zařízení – vnitřní pohled



## D Použití s Docker Desktop pro Windows

Jedním z možných způsobů využití zařízení je s použitím dockerizace. V Docker kontejneru může běžet databáze InfluxDB i webová aplikace Grafana. Tato kapitola popisuje postup nastavení Dockeru pro spuštění obou kontejnerů a příjem dat z měření proudu. Uvedený postup se bude týkat desktopové verze Dockeru pro systém Windows.

### D.1 Vytvoření síťového adaptéru

Před samotným začátkem je nutné nastavit Docker síťový adaptér, který nám umožní komunikaci jak mezi kontejnery, které budou na stejné síti, tak komunikaci směrem ven (tedy z hosta do sítě). Pro tento účel využijeme adaptér, který bude plnit funkce síťového mostu („bridge“). Vytvoříme jej následujícím příkazem:

```
1 docker network create -d bridge bridge-network
```

Zdrojový kód 15: Nastavení síťového adaptéru v Dockeru

Název adaptéru můžeme zvolit libovolně, zde je použit název „bridge-network.“

### D.2 Stažení image

Pro spuštění kontejnerů je potřeba image, který bude použit pro daný kontejner. V případě webové aplikace Grafana můžeme použít nejnovější dostupnou verzi (stažení image bude s příznakem „latest“). Pro InfluxDB je však doporučena verze 1.8.10, což je poslední verzi verze 1. Od verze 2 se v InfluxDB zcela změnila syntaxe a dokumentace k této verzi byla v době psaní této práce nepříliš dobře vytvořená, což způsobovalo nemalé potíže při nastavování databáze.

```
1 docker pull grafana-enterprise:latest
2 docker pull influxdb:1.8.10
```

Zdrojový kód 16: Stažení image pro Docker

### D.3 Docker compose

Jelikož budeme používat dva kontejnery, je nejlepší možností využít nástroje „docker compose,“ který nám umožní vytvořit oba kontejnery v jednom kontejneru a současně nám umožní definovat určité nastavení aplikací. V konfiguračním souboru *docker-compose.yml*<sup>42</sup> pak mohou být definovány následující části:

<sup>42</sup>Název *docker-compose.yml* je nutné dodržet.

- Image, který bude použit pro Docker kontejner
- Název kontejneru
- Port, na kterém daný kontejner poběží
- Síť (zde je důležité mít stejnou síť pro oba kontejnery)
- Volitelně pak můžeme nastavit i virtuální disky, kde budou uložena data kontejnerů. Tyto disky budou automaticky vytvořeny při vytvoření kontejnerů.

Poté stačí v PowerShellu, případně v příkazovém řádku přejít do složky, která obsahuje zmíněný konfigurační soubor a zadat příkaz *docker compose up*. Pokud chceme zadat i název vytvořeného kontejneru, spustíme uvedený příkaz s parametrem *-p <container-name>*.

Po zadání následující adresy

*http://localhost:3000* <sup>43</sup>

se dostaneme na webové rozhraní aplikace Grafana.

Databázi InfluxDB máme vytvořenou a díky zadané konfiguraci v ní existuje i databáze *ecmddb*. InfluxDB neobsahuje webové rozhraní a přístup probíhá pouze přes CLI<sup>44</sup>. To můžeme spustit následujícím příkazem:

```
1 docker exec -it influxdb \bin\bash
```

Zdrojový kód 17: Spuštění CLI InfluxDB

Příložený kód pro *Docker compose* navíc obsahuje vlastní konfiguraci souboru *influxdb.conf* připojenou jako volume, ve které je zapnuta autentizace, která ve výchozím stavu není aktivní. Konfigurační soubor byl doplněn o následující řádky:

```
1 [http]
2 enabled = true
3 bind-address = ":8086"
4 auth-enabled = true
5 log-enabled = true
6 write-tracing = true
```

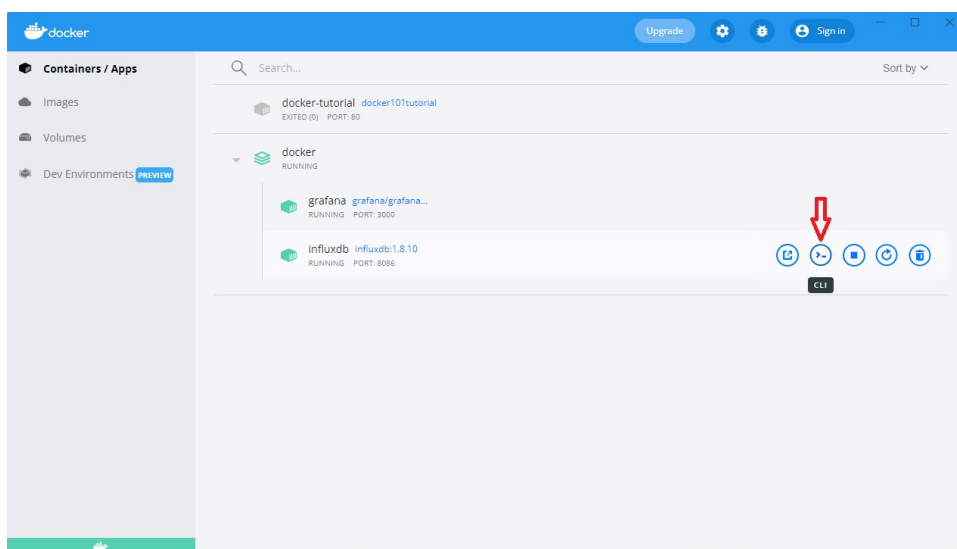
Zdrojový kód 18: Zapnutí autentizace pro InfluxDB

---

<sup>43</sup>Pro přístup z jiného zařízení je nutné znát IP adresu fyzického zařízení, na kterém běží Docker a tuto adresu poté zadat do adresního řádku v prohlížeči s daným číslem portu.

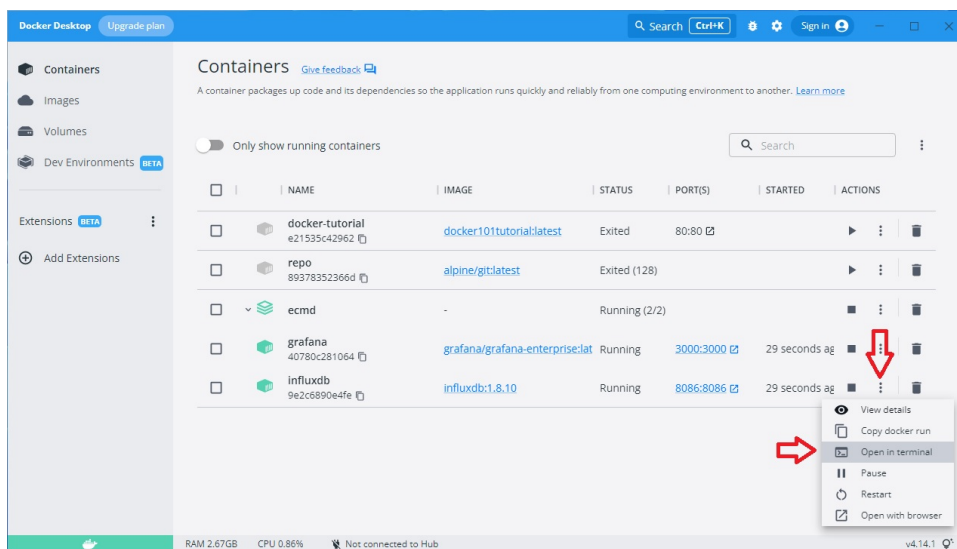
<sup>44</sup>Command Line Interface

CLI může být spuštěno rovněž i z desktopové aplikace:



Obrázek 26: Spuštění CLI přes Docker UI verze 4.8.x a nižší

Docker Desktop ve verzi 4.9.x prošel lehkou úpravou uživatelského rozhraní, co se týče tlačítka pro spuštění CLI, došlo pouze k jeho vizuální změně a rovněž byl změněn popisek – namísto „CLI“ nyní uživatel uvidí možnost „Open in terminal.“



Obrázek 27: Spuštění CLI přes Docker UI od verze 4.14.x

CLI lze spustit i pro webovou aplikaci Grafana. V tomto případě nám ale plně postačí její webové rozhraní, kde jsme schopni nastavit veškerou potřebnou funkcionalitu, např. odkud budeme data odebírat, jakým způsobem chceme nastavit zobrazení grafů atd.

Následující kód pak demonstruje podobu konfiguračního souboru, díky kterému vytvoříme a spustíme kontejnery s InfluxDB a Grafanou.

```
1 version: "3.9"
2 services:
3   grafana:
4     image: grafana/grafana-enterprise:9.3.1
5     container_name: grafana
6     restart: always
7     ports:
8       - 3000:3000
9     networks:
10      - ECMD-bridge
11     volumes:
12      - grafana_data:/var/lib/grafana
13     depends_on:
14      - influxdb
15
16
17   influxdb:
18     image: influxdb:1.8.10
19     container_name: influxdb
20     restart: always
21     ports:
22      - 8086:8086
23     networks:
24      - ECMD-bridge
25     volumes:
26      - influxdb_data:/var/lib/influxdb
27      - .\influxdb.conf:/etc/influxdb/influxdb.conf
28     environment:
29      - INFLUXDB_DB=ecmddb
30 networks:
31   ECMD-bridge:
32 volumes:
33   grafana_data:
34   influxdb_data:
```

Zdrojový kód 19: Konfigurační soubor pro docker compose ve formátu YAML

## E Obsah elektronické přílohy

Na samotném konci textu práce je uveden stručný popis obsahu elektronické přílohy, tj. její závazné adresářové struktury, důležitých souborů apod.

### **bin/**

Image operačního systému Raspbian nakonfigurovaný a uzpůsobený pro bezproblémový běh aplikace na měření proudu a integrovaného webového serveru. Z image lze pomocí SD karty vytvořit bootovatelné médium a použít pro spuštění na kterémkoliv Raspberry Pi 3B+.

### **doc/**

Text práce ve formátu PDF, vytvořený s použitím závazného stylu KI PřF UP v Olomouci pro závěrečné práce, včetně všech příloh, a všechny soubory potřebné pro bezproblémové vygenerování PDF dokumentu textu (v ZIP archivu), tj. zdrojový text textu, vložené obrázky, apod.

### **src/**

Kompletní zdrojové texty k zařízení ECMD se všemi potřebnými (příp. převzatými) zdrojovými texty, knihovnami a dalšími soubory potřebnými pro bezproblémové vytvoření spustitelných verzí programu.

### **readme.txt**

Instrukce pro instalaci a spuštění zařízení ECMD, včetně všech požadavků pro jeho bezproblémový provoz pro účel testování při tvorbě posudků práce a pro účel obhajoby práce.

Navíc elektronická příloha obsahuje:

### **data/**

Ukázková a testovací data použitá v práci a pro potřeby testování práce při tvorbě posudků a obhajoby práce.

### **install/**

Seznam Python knihoven, včetně instrukcí pro instalaci, pro provoz zařízení ECMD, které nejsou standardní součástí operačního systému určeného pro běh programu.

U veškerých cizích převzatých materiálů obsažených v elektronické příloze jejich zahrnutí dovolují podmínky pro jejich šíření nebo přiložený souhlas držitele copyrightu. Pro všechny použité (a citované) materiály, u kterých toto není splněno a nejsou tak obsaženy v elektronické příloze, je uveden jejich zdroj (např. webová adresa) v bibliografii nebo textu práce nebo v souboru `readme.txt`.

## Literatura

- [1] BHATIA, A. HVAC Cooling Systems for Data Centers. [online]. [cit. 2021-12-27]. Dostupné z: [https://www.cedengineering.com/userfiles/HVAC Cooling Systems for Data Centers R1.pdf](https://www.cedengineering.com/userfiles/HVAC%20Cooling%20Systems%20for%20Data%20Centers%20R1.pdf)
- [2] FLATICON, Lightning icons created by Freepik. [online]. [cit. 2022-12-04]. Dostupné z: [https://www.flaticon.com/free-icons/lightning"](https://www.flaticon.com/free-icons/lightning)
- [3] JIRÁSKO, Tomáš. Hospodářské noviny: Fyzická bezpečnost datových center. [online]. [2016-10-10]. [cit. 2021-12-27]. Dostupné z: [https://ictrevue.hn.cz/c3-65470890-0ICT00\\_d-65470890-fyzicka-bezpecnost-datovych-center](https://ictrevue.hn.cz/c3-65470890-0ICT00_d-65470890-fyzicka-bezpecnost-datovych-center)
- [4] KOČMAN, Rostislav. Projekt FENIX – klub ISP a provozovatelů, kteří si v oblasti bezpečnosti věří [online]. [2014-09-23]. [cit. 2022-01-19]. Dostupné z: <http://www.internetprovsechny.cz/projekt-fenix-klub-isp-kteri-si-v-oblasti-bezpecnosti-veri/>
- [5] MŮČKA, Jan. RAID disková pole: jaké jsou základní typy a v čem se liší? [online]. [2021-08-12]. [cit. 2022-01-14]. Dostupné z: <https://www.master.cz/blog/raid-diskova-pole-jake-jsou-zakladni-typy-a-v-cem-se-lisi/>
- [6] UPTON, Eben. HALFACREE, Gareth. Raspberry Pi Uživatelská příručka. [2012] [cit. 2022-01-25]. ISBN 978-80-251-4116-8.
- [7] VELIMIROVIC, Andreja. Data Center Tiers Explained [online]. [2021-11-25]. [cit. 2022-01-14]. Dostupné z: <https://phoenixnap.com/blog/data-center-tiers-classification>