

Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra informačního inženýrství



Diplomová práce

Chytrá serverová skříň řízená platformou Raspberry Pi

Bc. Jindřich Cvak

© 2024 ČZU v Praze

ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE

Provozně ekonomická fakulta

ZADÁNÍ DIPLOMOVÉ PRÁCE

Bc. Jindřich Cvak

Informatika

Název práce

Chytrá serverová skříň řízená platformou Raspberry Pi

Název anglicky

Smart server rack controlled by the Raspberry Pi platform

Cíle práce

Cílem práce je vytvoření zařízení pro serverovou skříň, které bude monitorovat stav skříně, včetně možnosti vzdáleného ovládání.

Cílem teoretické části práce je popis platformy Raspberry Pi, vysvětlení základních pojmů týkajících se používání Raspberry Pi, popis serverové skříně, provedení analýzy již existujících chytrých serverových skříní a výběr nejvhodnějších hardwarových komponent pro sestavení zařízení.

Cílem praktické části práce je sestavení a naprogramování vzorového zařízení, které po vmontování do serverové skříně umožní její vzdálené monitorování (například hlídání kouře, vytopení, měření vlhkosti a teplot v různých místech skříně) a ovládání z počítače.

Metodika

Metodika řešené problematiky diplomové práce je založena na studiu a analýze odborných zdrojů týkajících se serverových skříní, výběru vhodných dostupných hardwarových komponentů (čidel a ostatních komponentů pro sestavení zařízení) a popis a konfigurace platformy Raspberry Pi.

Na základě tohoto teoretického základu je vytvořen vlastní návrh zařízení pro serverovou skříň, které monitoruje stav skříně (hlídání kouře, vytopení, měření vlhkosti a teplot v různých místech skříně) a umožňuje její vzdálenou kontrolu a ovládání.

Doporučený rozsah práce

60–80 stran

Klíčová slova

Raspberry Pi, server rack, automatický systém, monitoring, Python

Doporučené zdroje informací

SUMMERFIELD, Mark. Programming in Python 3: a complete introduction to the Python language. 2nd ed., Fully rev. ed. Upper Saddle River, NJ: Addison-Wesley, c2010. Developer's library. ISBN 978-0-321-68056-3.

UPTON, Eben; HALFACREE, Gareth; GONER, Jakub. Raspberry Pi: uživatelská příručka. Brno: Computer Press, 2013. ISBN 978-80-251-4116-8.

Předběžný termín obhajoby

2023/24 LS – PEF

Vedoucí práce

Ing. Dana Vynikarová, Ph.D.

Garantující pracoviště

Katedra informačního inženýrství

Elektronicky schváleno dne 28. 11. 2023

Ing. Martin Pelikán, Ph.D.

Vedoucí katedry

Elektronicky schváleno dne 9. 2. 2024

doc. Ing. Tomáš Šubrt, Ph.D.

Děkan

V Praze dne 20. 02. 2024

Čestné prohlášení

Prohlašuji, že svou diplomovou práci "Chytrá serverová skříň řízená platformou Raspberry Pi" jsem vypracoval(a) samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu použitých zdrojů na konci práce. Jako autor uvedené diplomové práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne 31.03.2024

Poděkování

Rád bych touto cestou poděkoval Ing. Daně Vynikarové, Ph.D. za její velkou trpělivost a za odborné rady a připomínky, které mi pomohly při tvorbě této diplomové práce.

Chytrá serverová skříň řízená platformou Raspberry Pi

Abstrakt

Tato diplomová práce se zabývá návrhem a sestavením inteligentního zařízení určeného pro montáž do serverové skříně s cílem zajištění jejího stálého monitorování s možností vzdáleného ovládání. Hlavním cílem práce je vytvoření prototypu zařízení, které bude schopné snímat prostředí uvnitř serverové skříně, například teplotu, vlhkost, přítomnost kouře nebo známky vytopení, a odesílat tato data do počítače.

Teoretická část práce se zaměřuje na obeznámení čtenáře s problematikou serverových skříní. Dále na popis platformy Raspberry Pi včetně seznámení s obdobnými platformami a uvedením a popisem potřebného softwaru a nástrojů jako jsou například Thonny IDE, Raspberry Pi imager nebo SSH protokol. Dále práce obsahuje přehled existujících řešení chytrých serverových skříní a je proveden výběr nejvhodnějších hardwarových komponent pro sestavení zařízení.

Praktická část je věnována konstrukci a programování vzorového zařízení, které zahrnuje detailní popis sestavení, programování a testování funkčnosti v rámci serverové skříně.

Práce demonstruje, jak lze s využitím platformy Raspberry Pi a vhodně vybraných hardwarových komponentů sestavit jednoduché a ekonomicky šetrné řešení pro zvýšení bezpečnosti a efektivity provozu malých datových center.

Výsledky práce představují praktický návod na realizaci vlastního monitorovacího a ovládacího systému pro serverové skříně.

Klíčová slova: Raspberry Pi, monitoring, Python, serverová skříň, automatický systém, SSH

Smart server rack controlled by the Raspberry Pi platform

Abstract

This thesis focuses on the design and assembly of an intelligent device intended for installation in a server rack to ensure its continuous monitoring with the possibility of remote control. The primary goal of the work is to create a prototype device capable of sensing the environment inside the server rack, such as temperature, humidity, presence of smoke, or signs of flooding, and sending this data to a computer.

The theoretical part of the work is aimed at acquainting the reader with the issues of server racks. It also describes the Raspberry Pi platform, including an introduction to similar platforms, and outlines the necessary software and tools such as Thonny IDE, Raspberry Pi Imager, or the SSH protocol. Furthermore, the work includes an overview of existing solutions for smart server racks and selects the most suitable hardware components for assembling the device.

The practical part is devoted to the construction and programming of the prototype device, including a detailed description of assembly, programming, and testing of functionality within the server rack.

The work demonstrates how, by using the Raspberry Pi platform and appropriately selected hardware components, a simple and economically efficient solution can be assembled to enhance the security and efficiency of the operation of small data centers.

The results of the work provide a practical guide for implementing your own monitoring and control system for server racks.

Keywords: Raspberry Pi, monitoring, Python, server rack, automatic system, SSH

Obsah

1 Úvod.....	11
2 Cíl práce a metodika	12
2.1 Cíle práce	12
2.2 Metodika	12
3 Teoretická východiska	13
3.1 Serverová skříň.....	13
3.1.1 Vlastnosti serverových skříní.....	13
3.1.2 Server	14
3.2 Raspberry Pi	14
3.2.1 Historie Raspberry Pi.....	14
3.2.2 Zakladní rozdělení modelů Raspberry Pi.....	14
3.2.3 Obdobné platformy	16
3.2.3.1 Arduino.....	16
3.2.3.2 PICAXE.....	16
3.2.3.3 Feather	16
3.2.4 Výhody platformy Raspberry Pi	17
3.2.5 Raspberry Pi Imager	18
3.2.6 Programování Raspberry Pi.....	18
3.2.6.1 Thonny IDE	19
3.2.6.2 Spyder3.....	20
3.2.6.3 Arduino IDE	21
3.3 Python	22
3.3.1 Python a Raspberry	22
3.4 HTTP Web Server.....	23
3.5 Secure Shell (SSH).....	23
3.6 HTML	23
3.7 PuTTY.....	24
3.8 Analýza existujících řešení	25
3.8.1 Austin Hughes Intelligent Server Infra Rack.....	25
3.8.2 Canovate's Smart Server Rack	26
3.8.3 Raritan Smart Rack	27
3.8.4 Rosenberger Smart Server Rack System	28
3.9 Výběr hardwarevých komponentů	29
3.9.1 Raspberry Pi Zero 2 W	29
3.9.2 Serverová skříň	31

3.9.3	Teplotní senzor	32
3.9.4	Senzor vlhkosti	32
3.9.5	Senzor kouře	33
3.9.6	Senzor vytopení	33
3.9.7	Adresovatelný led pásek	34
3.9.8	Elektromagnetický zámek.....	35
3.9.9	Magnetický senzor	35
3.9.10	Napájecí zdroje	36
3.9.11	Vodiče	36
3.9.12	Tabulka cen hardwaru.....	36
4	Vlastní práce	37
4.1	Instalace Raspberry Pi OS.....	37
4.2	Zapojení.....	40
4.2.1	Elektrotechnické schéma zapojení.....	40
4.3	Instalace potřebných balíčků.....	42
4.4	Program Raspberry.....	43
4.4.1	Import knihoven.....	43
4.4.2	Konfigurace pro led pásek	43
4.4.3	Nastavení GPIO pinů	44
4.4.4	Funkce na zbarvení led pásku	45
4.4.5	Připravení csv souboru.....	45
4.4.6	Main program	46
4.4.6.1	Inicializační část	46
4.4.6.2	Hlavní cyklus.....	47
4.5	Webový server	50
4.5.1	Instalace potřebných balíčků pro WS	50
4.5.2	Import knihoven pro WS	50
4.5.3	Nastavení komunikace.....	51
4.5.4	Funkce pro generování grafů z csv souboru	52
4.5.5	Funkce pro získání teploty čipu	54
4.5.6	Funkce pro nastavení GPIO pinu.....	54
4.5.7	Vytvoření třídy MyServer.....	54
4.5.8	Main program	58
5	Výsledky a diskuse	59
5.1	Ukázka Grafů	59
5.2	Ukázka webového serveru	63
5.3	Komplikace a zjištění	63
6	Závěr.....	64

7	Seznam použitých zdrojů	65
8	Seznam obrázků, tabulek a grafů	68
8.1	Seznam obrázků	68
8.2	Seznam tabulek	69
8.3	Seznam grafů.....	69
9	Přílohy	70

1 Úvod

Rozvoj digitálních technologií a nárůst objemů dat, která podniky každodenně zpracovávají a ukládají, klade nové výzvy pro infrastrukturu datových center. V tomto duchu se chytré serverové skříně ukazují jako dobré řešení, které umožňuje organizacím nejen efektivněji spravovat svou IT infrastrukturu, ale také optimalizovat výkon a snížit energetickou spotřebu čímž by byly více ekologické. Díky integrovaným systémům pro monitorování a řízení mohou chytré racky předvídat například potřeby chlazení, což vede k výrazné úspoře energie a zvyšuje se tím celková efektivita datového centra.

Technologie využívané v chytrých serverových skříních poskytují správcům datových center detailní přehled o jejich stavu a hlídání poruch.

Chytré serverové skříně představují základní kámen pro rozvoj moderních technologií, jako jsou cloudové služby, edge-computing a big data. V éře digitalizace a internetu věci tak chytré serverové skříně nejen zefektivňují provoz datových center, ale poskytují i důležité a včasné informace správcům datových center, kteří s nimi mohou nadále pracovat.

Tento projekt je tvořen za účelem vytvoření jednoduchého, cenově příznivého a dostupného řešení pro amatéry či malé nenáročné firmy.

2 Cíl práce a metodika

2.1 Cíle práce

Cílem práce je vytvoření zařízení pro serverovou skříň, které bude monitorovat stav skříně, včetně možnosti vzdáleného ovládání.

Cílem teoretické části práce je popis platformy Raspberry Pi, vysvětlení základních pojmů týkajících používání Raspberry Pi, popis serverové skříně, provedení analýzy již existujících chytrých serverových skříní a výběr nejvhodnějších hardwarových komponent pro sestavení zařízení.

Cílem praktické části práce je sestavení a naprogramování vzorového zařízení, které po vmontování do serverové skříně umožní její vzdálené monitorování (například hlídání kouře, vytopení, měření vlhkosti a teplot v různých místech skříně) a ovládání z počítače.

2.2 Metodika

Metodika řešené problematiky diplomové práce je založena na studiu a analýze odborných zdrojů týkajících se serverových skříní, výběru vhodných dostupných hardwarových komponentů (čidel a ostatních komponentů pro sestavení zařízení) a popis a konfigurace platformy Raspberry Pi.

Na základě tohoto teoretického základu je vytvořen vlastní návrh zařízení pro serverovou skříň, které monitoruje stav skříně (hlídání kouře, vytopení, měření vlhkosti a teplot v různých místech skříně) a umožňuje její vzdálenou kontrolu a ovládání.

3 Teoretická východiska

Tato část práce se zaměřuje na obeznámení čtenáře s problematikou serverových skříní. Dále na popis platformy Raspberry Pi včetně seznámení s obdobnými platformami a uvedením a popisem potřebného softwaru a nástrojů jako jsou například Thonny IDE, Raspberry Pi imager nebo SSH protokol. Dále práce obsahuje přehled existujících řešení chytrých serverových skříní a je proveden výběr nejvhodnějších hardwarových komponent pro sestavení zařízení.

3.1 Serverová skříň

Serverové skříně jsou základním stavebním kamenem pro datacentra a IT infrastrukturu firem, poskytují nejen organizovaný prostor pro hardwarové komponenty, ale zároveň optimalizují proudění vzduchu čímž podporují efektivní chlazení zařízení instalovaných uvnitř. Serverová skříň je standardizovaná a navržena pro uspořádané uskladnění a instalaci různých typů síťových zařízení. Tyto skříně jsou obvykle vysoké a umožňují bezpečné a efektivní řazení hardwaru vertikálně, což šetří prostor a zjednodušuje správu kabelů a udržování systému. [1]

Vnitřní šířka standardního serverové skříně je 19 palců (48,26 cm), což je průmyslový standard označovaný jako EIA-310, vyvinutý organizací Electronics Industry Alliance. Tato šířka byla zvolena pro zajištění kompatibility s širokou škálou zařízení. Existují i širší varianty racků, jako jsou 23palcové a 24palcové modely, které mohou poskytovat více prostoru pro správu kabelů nebo akomodaci větších zařízení, ale 19palcové modely jsou nejběžnější. [1]

Hloubka serverového racku se liší v závislosti na konkrétním modelu a výrobci, ale nejběžnější hloubka se pohybuje mezi 36 a 42 palci (91,44 až 106,68 cm). Existují však také mělké i hlubší racky pro specifické potřeby. [1]

3.1.1 Vlastnosti serverových skříní

Jak již bylo zmíněno základními vlastnostmi serverových skříní jsou standardizovaná šířka 19 palců pro kompatibilitu s většinou zařízení a výška vyjádřena v jednotkách rack unit (U) pro určení kapacity skříně. Dále je důležité, aby měli dostatek místa na efektivní správu kabeláže, zabezpečení zařízení a optimalizaci proudění vzduchu

pro chlazení. Co však běžné skříně nemají za funkce jsou například světelná signalizace, monitorování přístupů, kontrola kouře či vytopení. [1]

3.1.2 Server

Server je systém navržený pro poskytování dat, služeb nebo aplikací ostatním počítačům nebo zařízením přes síť. Servery mohou být specializované na různé úkoly, jako je hostování webových stránek, ukládání dat a mnoho dalšího. Servery bývají nejčastěji uloženy ve skříních, tyto skříně poskytují, ochranu a organizaci pro servery a související hardware. Umožňují efektivní správu kabeláže, optimalizaci chlazení a zabezpečení zařízení vůči neoprávněné manipulaci či vniknutí. [2]

3.2 Raspberry Pi

Raspberry Pi jsou malé, cenově dostupné počítače malé velikosti, navržené pro vzdělávání v informatice a pro vývojáře k realizaci různých elektronických projektů, od domácích automatizací po robotiku. Fungují s různými operačními systémy, jako je například Raspbian, a lze je programovat v mnoha jazycích, včetně Pythonu a C++. Raspberry Pi se vyznačuje dobrým výkonem za nízkou cenu, ale vyžaduje doplňkové komponenty pro rozšíření funkcionalit, jako jsou například různé senzory a čidla, nebo popřípadě jiné periferie. [3]

3.2.1 Historie Raspberry Pi

Raspberry Pi začala v roce 2006 na Univerzitě v Cambridge, kde se skupina výzkumníků včetně Ebena Uptona a jeho kolegů snažila navrhnout cenově dostupný počítač, který by podporoval výuku programování ve školách. Cílem bylo vytvořit zařízení, které by studentům umožnilo lépe porozumět, jak počítače fungují, a zároveň zvrátit pokles zájmu o studium informatiky. Tento projekt vyústil ve vznik nadace Raspberry Pi a v únoru 2012 byl uveden na trh první model Raspberry Pi. [3]

3.2.2 Základní rozdělení modelů Raspberry Pi

Rozdělení desek Raspberry Pi se v průběhu let vyvíjelo a rozšiřovalo. Počínaje původním modelem B, který byl uveden na trh v roce 2012, se portfolio rozšířilo o několik hlavních řad a modelů, které se liší výkonem, velikostí, konektivitou a specializovanými funkcemi. [4]

Modely řady B:

1. Raspberry Pi 1 Model B: První vydání s procesorem ARM 700 MHz a 512 MB RAM. [4]
2. Raspberry Pi 2 Model B: Přináší čtyřjádrový procesor 900 MHz a 1 GB RAM. [3]
3. Raspberry Pi 3 Model B: Zavádí čtyřjádrový procesor 1,2 GHz, Wi-Fi a Bluetooth. [4]
4. Raspberry Pi 3 Model B+: Vylepšený model s čtyřjádrovým procesorem 1,4 GHz, lepším Wi-Fi a gigabitovým Ethernetem. [4]
5. Raspberry Pi 4 Model B: Nejnovější a nejvýkonnější model s možností 1, 2, 4, nebo 8 GB RAM, vylepšenou konektivitou a podporou pro dvojici 4K monitorů. [4]

Modely řady A:

1. Raspberry Pi 1 Model A: Zjednodušená a levnější varianta modelu B s 256 MB RAM a bez Ethernetu. [4]
2. Raspberry Pi 1 Model A+: Menší a energeticky efektivnější verze s 512 MB RAM. [4]
3. Raspberry Pi 3 Model A+: Poslední model řady A s čtyřjádrovým procesorem 1,4 GHz, 512 MB RAM, Wi-Fi a Bluetooth. [4]

Raspberry Pi Zero:

1. Raspberry Pi Zero: Extrémně kompaktní a cenově dostupná verze s jednojádrovým procesorem 1 GHz a 512 MB RAM. [4]
2. Raspberry Pi Zero W: Přidává Wi-Fi a Bluetooth konektivitu. [4]

Raspberry Pi 400:

1. Originální design, který dává výkon Raspberry Pi 4 Model B do kompaktního klávesnicového těla, poskytující tak uživatelsky přívětivý, all-in-one počítač. [4]

Compute Modules:

1. Compute Module 1, 3, 3+ a 4: Zaměřeny na průmyslové a komerční aplikace, tyto moduly nabízí možnost integrovat výpočetní sílu Raspberry Pi do vlastních hardwarových designů. Každý modul přináší různé výkonové specifikace a možnosti úložiště. [4]

3.2.3 Obdobné platformy

Raspberry Pi není jediná platforma na trhu, proto je vhodné mít povědomí i o jiných dostupných platformách jako například Arduino, PICAXE a Feather.

3.2.3.1 Arduino

Arduino se vyznačuje svou jednoduchostí a uživatelsky přívětivým prostředím, což přispělo k jeho rozšíření mezi širokou veřejností. Využívá mikrokontrolery od společnosti Atmel a umožňuje snadné připojení široké škály vstupních a výstupních zařízení, jako jsou senzory, čidla, servo motory a další. Programy pro Arduino jsou psány v jazyce Wiring, který je odvozen od C++. Rozšířenost Arduina má za následek vznik rozsáhlé komunitní podpory na různých fórech všude po světě. Zde uživatelé mohou nalézt nepřehledné množství informací a rad pro řešení různých problémů. Mezi nejoblíbenější modely desek Arduino patří Arduino Uno Rev3, Arduino Mega2560 Rev3 a Arduino Leonardo. [5]

3.2.3.2 PICAXE

Obdobně jako platformy Raspberry a Arduino byly mikroprocesory PICAXE navrženy pro vzdělávací účely. Tyto mikroprocesory používají mikročip PIC, který je již z výroby předprogramován pomocí PICKAXE bootstrapu a programují se v jazyce BASIC. Avšak pro potřeby tohoto projektu je tato platforma nedostačující. [11]

3.2.3.3 Feather

Feather je skupina mikrokontrolerových desek (Feathers) a dceřiných desek (Wings) od společnosti Adafruit, která nabízí flexibilitu a silné výpočetní možnosti pro různé projekty. Tato platforma není vázána na jednoho výrobce čipů, což umožňuje použití nejlepších čipsetů na trhu, včetně různých mikrokontrolérů a modulů pro bezdrátovou komunikaci. Feather podporuje široký rozsah funkčních rozšíření prostřednictvím Wings, které lze teoreticky přidávat, dokud nedojde k nedostatku výkonu, místa nebo konfliktů pinů. Veškerý kód pro desky a Wings je k dispozici v Arduino C/C++, a některé desky také podporují CircuitPython. [6]

3.2.4 Výhody platformy Raspberry Pi

Raspberry Pi nabízí několik výhod, které ji odlišují od jiných vývojových platforem mikropočítačů:

1. **Cena:** Ačkoli cena Raspberry Pi není v porovnání s ostatními platformami nejnižší, stále s ohledem na výkon Raspberry počítačů je velice přijatelná a dostupná pro širokou veřejnost.
2. **Komunita:** Raspberry Pi má velmi velké aktivní podpůrné online komunitní fórum a mnoho uživatelských skupin po celém světě. Tato komunita poskytuje nepřeborné množství návodů, projektů, a podpory, což usnadňuje řešení problémů a sdílení znalostí.
3. **Všestrannost:** Raspberry Pi lze použít pro širokou škálu aplikací, od učení programování a počítačové vědy, přes domácí automatizaci, až po pokročilé průmyslové projekty. Nabízí také rozsáhlou škálu příslušenství a rozšiřitelných modulů, které dále rozšiřují její využití.
4. **Portabilita:** Díky svým malým rozměrům je Raspberry Pi extrémně přenosné a lze je snadno integrovat do projektů, kde je potřeba kompaktní a mobilní výpočetní výkon.
5. **Operační systémy:** Raspberry Pi podporuje několik operačních systémů, včetně Raspberry Pi OS (dříve Raspbian), Ubuntu, a dokonce i Windows 10 IoT Core, což uživatelům umožňuje flexibilitu při výběru softwaru podle jejich potřeb.

3.2.5 Raspberry Pi Imager

Raspberry Pi Imager je oficiální nástroj vytvořený Raspberry Pi Foundation pro snadné zapisování obrazů operačních systémů na SD karty, určené pro použití v Raspberry Pi.

Umožňuje uživatelům vybrat a nainstalovat různé operační systémy speciálně optimalizované pro Raspberry Pi, včetně Raspberry Pi OS (dříve Raspbian), Ubuntu, a mnoha dalších. Imager poskytuje jednoduché grafické rozhraní, kde uživatelé mohou vybrat požadovaný obraz operačního systému z předem načteného seznamu, určit cílové zařízení a spustit proces zápisu. Nástroj také nabízí funkce, jako je možnost vymazání disku před zápisem nebo nastavení konfigurace sítě pro systémy, které to podporují, ještě před prvním spuštěním. Tím se usnadňuje nastavení nových zařízení Raspberry Pi a dělá ze zápisu operačních systémů rychlou, jednoduchou a bezproblémovou záležitost. [10]



Obrázek 1 Raspberry Pi Imager [10]

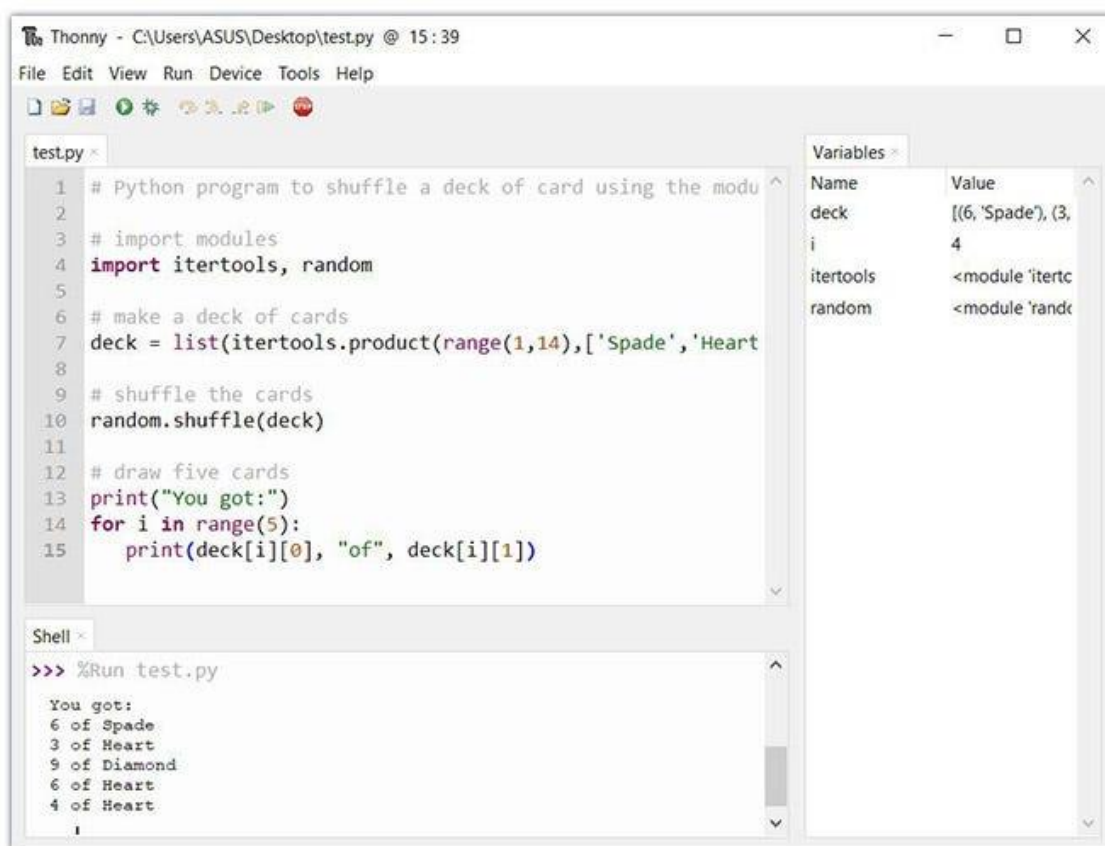
3.2.6 Programování Raspberry Pi

Raspberry není vázané pouze na jedno IDE (integrované vývojové prostředí), ale poskytuje programátorům při výběru nepřeberné množství možností, jako jsou například Thonny IDE, Spyder3 a překvapivě i Arduino IDE. [7, 8, 9]

3.2.6.1 Thonny IDE

Thonny je bezplatné vývojové prostředí pro Python, které je navrženo především pro začátečníky. Obsahuje vstavený debugger, což uživatelům usnadňuje a ladění kódu. Uživatelské rozhraní Thonny je navrženo tak, aby bylo pro nováčky co nejpřístupnější, zároveň však nabízí dostatečnou flexibilitu i pro pokročilé programátory. Thonny podporuje Mac, Windows a Linux. [7]

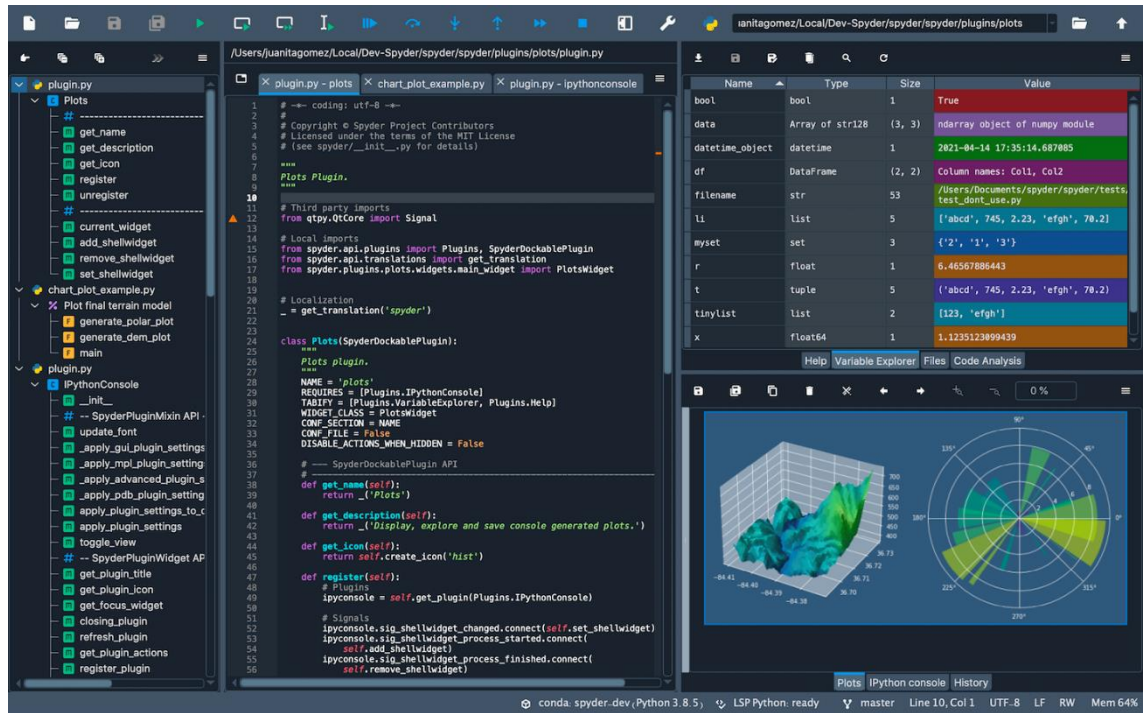
Pro tento projekt je to nejvhodnější vývojové prostředí a bude v něm dále pracováno.[7]



Obrázek 2 Thonny IDE [7]

3.2.6.2 Spyder3

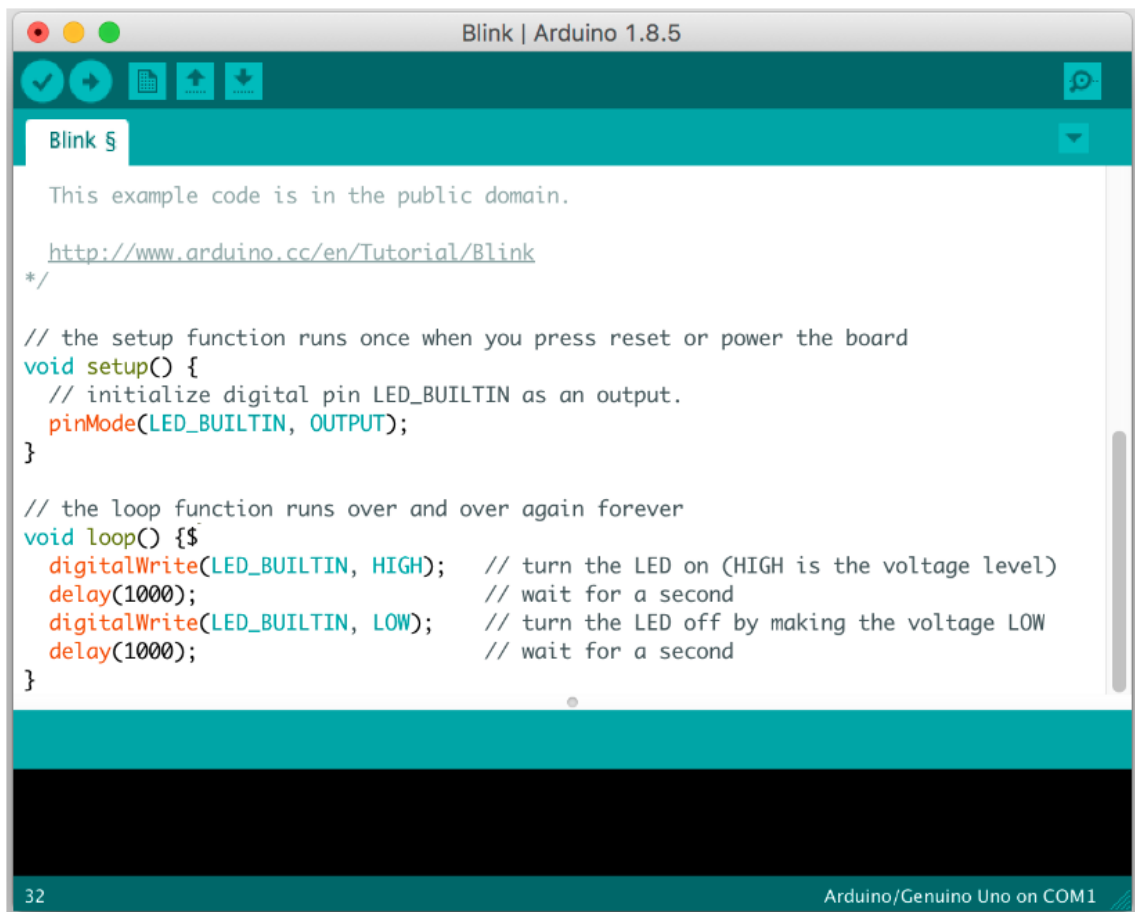
Spyder je open-source vývojové prostředí psané v Pythonu pro Python, navržené pro vědce, inženýry a analytiky dat. Nabízí kombinaci pokročilých funkcí pro editaci, analýzu a ladění společně s možnostmi pro průzkum dat, důkladnou inspekci a vizualizaci. [8]



Obrázek 3 Spyder3 IDE [8]

3.2.6.3 Arduino IDE

Arduino IDE je vývojové prostředí primárně určené pro platformu Arduino, fungující rovněž jako textový editor s přidávanými funkcemi a umožňuje vytváření nových projektů, kompilaci kódu a po připojení desky k počítači jeho nahrání do mikroprocesoru přes USB port. Součástí jsou předinstalované vzorové kódy pro práci s různými komponenty a integrované rozhraní pro sériovou komunikaci mezi deskou a počítačem. Díky tomu lze z počítače odesílat data do právě běžícího programu a přijímat z něj informace zpět. V prostředí najdeme také nástroje pro snadné přidávání uživatelských knihoven, což značně usnadňuje rozšíření možností vytvářených projektů. Dále toto vývojové prostředí. [12]

The image shows a screenshot of the Arduino IDE software. The window title is "Blink | Arduino 1.8.5". The interface has a teal header bar with icons for check, refresh, file, upload, and download. Below the header, there's a tab labeled "Blink §". The main area contains the following code:

```
This example code is in the public domain.

http://www.arduino.cc/en/Tutorial/Blink
*/

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {$
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}
```

At the bottom left, the number "32" is visible, and at the bottom right, it says "Arduino/Genuino Uno on COM1".

Obrázek 4 Arduino IDE [13]

3.3 Python

Python, vyvinutý Guidem van Rossumem v roce 1991, je vysokoúrovňový programovací jazyk. Je známý svou dynamickou kontrolu typů a podporuje různá programovací paradigmat, jako jsou například objektivě orientované, procedurální a funkcionální programování. Jeho popularita prudce vzrostla v roce 2018, kdy se stal jedním z nejpoblárnějších programovacích jazyků. Jako open-source projekt je Python zdarma dostupný pro většinu platforem a často je součástí základní instalace v mnoha Linuxových distribucích. Jméno Python je inspirováno komediální skupinou Monty Python, zatímco jeho logo je inspirováno krajtou (anglicky: python). [14]



Obrázek 5 Python logo [15]

3.3.1 Python a Raspberry

Python je v Raspberry Pi často využíván pro jeho jednoduchost a flexibilitu, což umožňuje uživatelům snadno vytvářet a testovat různé projekty od základních aplikací až po složitější systémy například pro automatizaci domácnosti, robotiku nebo sběr dat. Díky bohatému výběru knihoven může být Python použit pro čtení dat z čidel a ovládání nepřehledného množství periférií, které se dají k Raspberry připojit. [14,16]

3.4 HTTP Web Server

Web server funguje na principu požadavek-odpověď. Když uživatel zadá URL adresu webové stránky do svého prohlížeče, prohlížeč odešle HTTP požadavek na web server, na kterém je tato stránka hostována. Web server přijme tento požadavek, zpracuje ho a vrátí odpovídající obsah, například HTML stránku. Tento obsah se poté zobrazí v prohlížeči uživatele. [35]

Weby mohou být statické nebo dynamické:

Statický web jednoduše poskytuje obsah uložený na svém disku bez dalšího zpracování. Pokud uživatel požaduje konkrétní stránku, server najde tuto stránku ve svém úložišti a pošle ji zpět klientovi. [35]

Dynamický web pracuje s aplikacemi serverové strany, které mohou generovat a poskytovat personalizovaný obsah uživatelům na základě různých parametrů, jako jsou uživatelské přihlášení, vstupní data nebo interakce uživatele. Tyto servery často používají databáze pro ukládání dat a skriptovací jazyky jako PHP, Python nebo JavaScript pro generování dynamických webových stránek. [35]

3.5 Secure Shell (SSH)

Secure Shell je síťový protokol, který umožňuje bezpečný přístup a komunikaci s jiným počítačem přes nezabezpečenou síť. SSH poskytuje šifrovaný kanál pro přihlášení, přenos souborů a spouštění příkazů na vzdáleném počítači a zajišťuje ochranu proti odposlouchávání, manipulaci s daty a dalším bezpečnostním rizikům. [36]

3.6 HTML

HTML je standardní značkovací jazyk používaný pro tvorbu a design webových stránek a aplikací. Definiuje strukturu webového obsahu a je základní stavební jednotkou všech webových stránek a webových aplikací. [38]

Pomocí různých tagů umožňuje autorům obsahu určit, jak by měl být jejich obsah strukturován a zobrazován v webových prohlížečích. Tagy HTML obklopují obsah a definují jeho vlastnosti a vztahy s ostatními částmi webové stránky. [38]

Kromě strukturování obsahu může HTML spolupracovat s kaskádovými styly (CSS) a JavaScriptem, což umožňuje autorům vytvářet estetické a interaktivní webové stránky. CSS se používá k definování vizuálního stylu webové stránky, včetně layoutu,

barev a písem, zatímco JavaScript poskytuje metody pro přidání dynamického chování, jako jsou například animace nebo formulářové ověření. [38]

3.7 PuTTY

PuTTY je široce používaný volně dostupný klient pro SSH původně vyvinut Simonem Tathamem a udržován komunitou dobrovolníků. PuTTY podporuje širokou škálu síťových protokolů, jako jsou SCP, SSH, Telnet nebo rlogin. Umožňuje autentizaci pomocí veřejného klíče a jednotné přihlašování Kerberos. Kromě toho obsahuje implementace příkazové řádky pro SFTP a SCP. [39]

3.8 Analýza existujících řešení

3.8.1 Austin Hughes Intelligent Server Infra Rack

Klíčové vlastnosti:

1. Kontrola přístupu
2. Monitoring teploty
3. Osvětlení
4. Cena okolo 600 000 Kč

V rámci analýzy byl zkoumán produkt InfraRack od společnosti Austin Hughes, jehož klíčové funkce zahrnují chytré zámky na kartu pro bezpečnější přístup, LED osvětlení pro lepší viditelnost uvnitř racku a teplotní čidla pro monitorování teplot, což umožňuje sledování provozních podmínek a snížení rizika přehřívání zařízení. [17]

Tento rack je vyráběn ve dvou provedeních, buď Stanalone provedení, nebo Networked provedení. [17]



Obrázek 6 Austin Hughes Intelligent Server Infra Rack [17]

3.8.2 Canovate's Smart Server Rack

Klíčové vlastnosti:

1. Monitoring teploty a vlhkosti
2. Monitoring ventilátoru (tj. rychlosti v otáčkách za minutu, poruchy)
3. Kontrola přístupu (tj. otevřeno/zavřeno)
4. Upozornění
5. Cena okolo 750 000 Kč

Kontrola přístupu je řešena pomocí monitoringu otevřených a zavřených dveří. Na předních dveřích je zabudovaná jednotka pro monitorování prostředí s grafickou obrazovkou LCD, kam přicházejí informace od senzorů. Řídící jednotka je poháněna mikrokontrolérem Cortex-M3 pracujícím na frekvenci 100 MHz. Grafické zobrazení výstupů je přístupné také na příslušné webové stránce. Senzor teploty a vlhkosti je přímo připojený k řídicí jednotce. Informace ze všech třech ventilátorů lze číst prostřednictvím řídicí jednotky a upozornit na poruchu ventilátoru. Ventilátory mají výkon 625 m³/h pro chlazení. Pro upozornění podporuje SMTP (Simple Mail Transfer Protocol) pro poštovní upozornění a SNMP (Simple Network Management Protocol) trapy pro hlášení. [18]



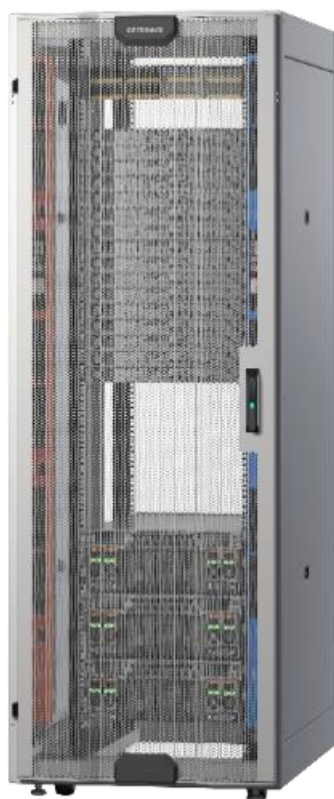
Obrázek 7 Canovate's Smart Server Rack [18]

3.8.3 Raritan Smart Rack

Klíčové vlastnosti:

1. Napájení
2. Monitoring prostředí
3. Kontrola přístupu
4. Upozornění
5. Sledování polohy IT aktiv
6. Cena okolo 450 000 Kč

Rozvod energie je řešen pomocí vzdáleného měření spotřeby energie v reálném čase. Pomocí monitoringu prostředí je možné lokalizovat přehřívaná místa, monitorovat proudění vzduchu a tlaku a optimalizovat chlazení. Přístup je řízen pomocí SMARTLOCK technologie, která zahrnuje síťové zamykání dveří a USB webovou kameru, která snímá, kdo přistupuje k racku. Zařízení je schopné posílat alerty a tvořit výstupy pomocí DCIM SW. Umožňuje také sledovat aktuální umístění IT aktiv. [19]



Obrázek 8 Raritan Smart Rack [19]

3.8.4 Rosenberger Smart Server Rack System

Klíčové vlastnosti:

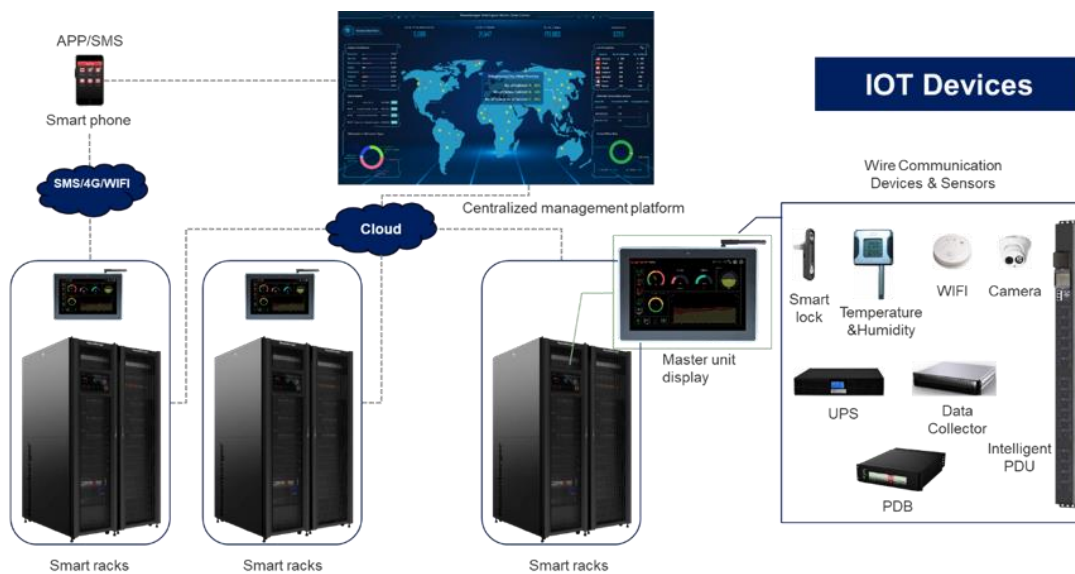
1. Monitoring teploty a vlhkosti
2. Monitoring úniku vody a proudění
3. Monitoring tlaku vzduchu
4. Napájení
5. Kontrola přístupu
6. Upozornění
7. Cena okolo 1 000 000 Kč

Tento rack monitoruje teplotu, vlhkost, únik a proudění vody a tlak vzduchu. Nepřetržitě přenáší informace v reálném čase a při zjištění hrozby odesílají upozornění. Hlavní zobrazovací jednotka je integrovaná s 10palcovým dotykovým displejem LCD. Obsahuje grafické rozhraní pro kontrolu a monitorování parametrů prostředí. Kontroluje přístup pomocí inteligentního zámku s biometrickým ověřováním. Inteligentní jednotky PDU poskytují distribuci energie a shromažďují informace o monitorování napájení v reálném čase. [20]



Obrázek 9 Rosenberger Smart Server Rack System 1 [20]

Prostřednictvím softwaru lze připojit externí zařízení IOT, jako jsou kamery a další senzory. Správa je centralizována pomocí Cloud technologie Smart Rack Cloud Central Monitoring, která umožňuje sledovat i více lokalit současně. [20]



Obrázek 10 Rosenberger Smart Server Rack System 2 [20]

3.9 Výběr hardwarevých komponentů

3.9.1 Raspberry Pi Zero 2 W

Raspberry Pi Zero 2 W je kompaktní a cenově dostupný mikropočítač, který byl navržen extrémně kompaktním formátu (65 mm × 30 mm). Raspberry Pi Zero 2 W představuje upgrade původního modelu Raspberry Pi Zero, nabízející zvýšený výkon. [21]

Procesor: Zařízení je vybaveno čtyřjádrovým 64bitovým procesorem ARM Cortex-A53 s taktem 1 GHz, což znamená značné zlepšení výkonu ve srovnání s původním Raspberry Pi Zero. [21]

Paměť: Obsahuje 512 MB SDRAM, což umožňuje efektivní běh aplikací a systémů. [21]

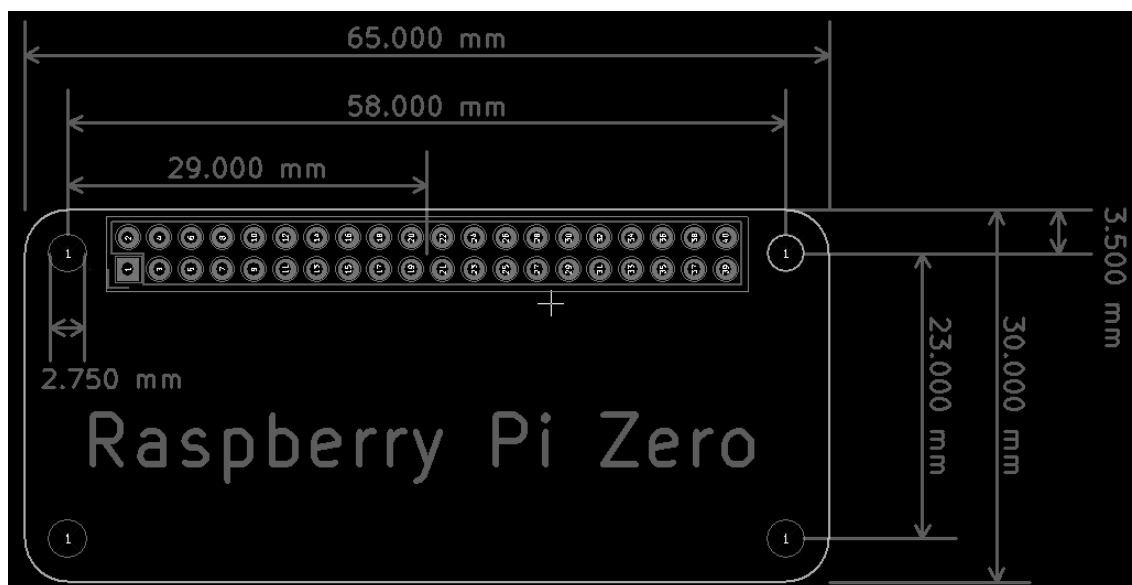
Bezdrátové připojení: Zařízení disponuje 2.4 GHz 802.11 b/g/n bezdrátovou LAN a Bluetooth 4.2, s vestavěnou anténou, což rozšiřuje jeho použitelnost pro projekty, které vyžadují bezdrátové spojení. [21]

Video a zvuk: Zařízení dále nabízí mini HDMI® port a mikro USB On-The-Go (OTG) port. Podporuje H.264 a MPEG-4 dekódování ve 1080p30 a H.264 kódování ve 1080p30, spolu s OpenGL ES 1.1, 2.0 grafikou. [21]

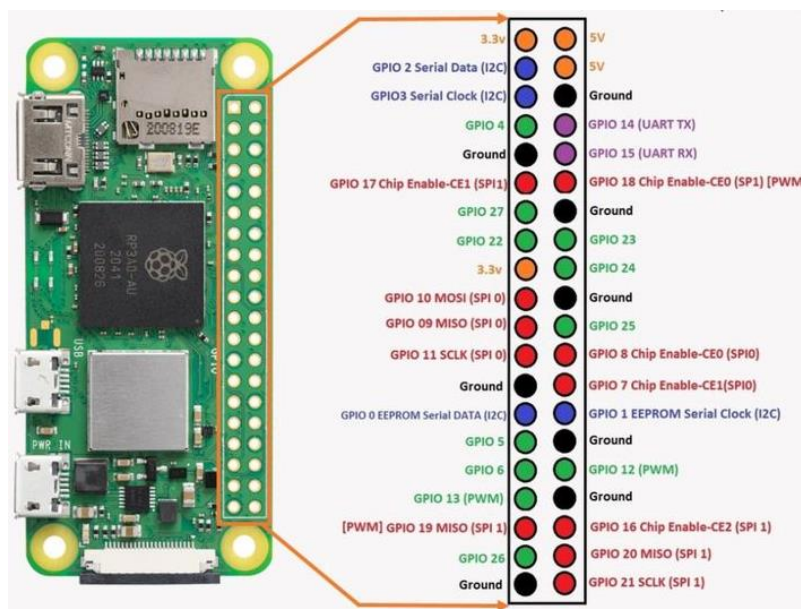
Úložiště a rozšiřitelnost: Zařizování je vybaveno slotem pro microSD kartu pro operační systém a úložiště dat, dále obsahuje 40pinový GPIO header pro připojení dalších různých vhodných zařízení a komponent, CSI-2 konektor pro kameru a port USB. [21]

Napájení: Napájení je řešeno jedním micro USB portem (5 V DC 2.5 A) a provozní teploty jsou od -20 °C do +70 °C. [21]

Tato platforma zůstane ve výrobě alespoň do ledna 2028. [21]



Obrázek 11 Rozměry Raspberry Pi Zero2 W [22]



Obrázek 12 Popis GPIO pinů desky Raspberry Pi Zero 2 W [23]

3.9.2 Serverová skříň

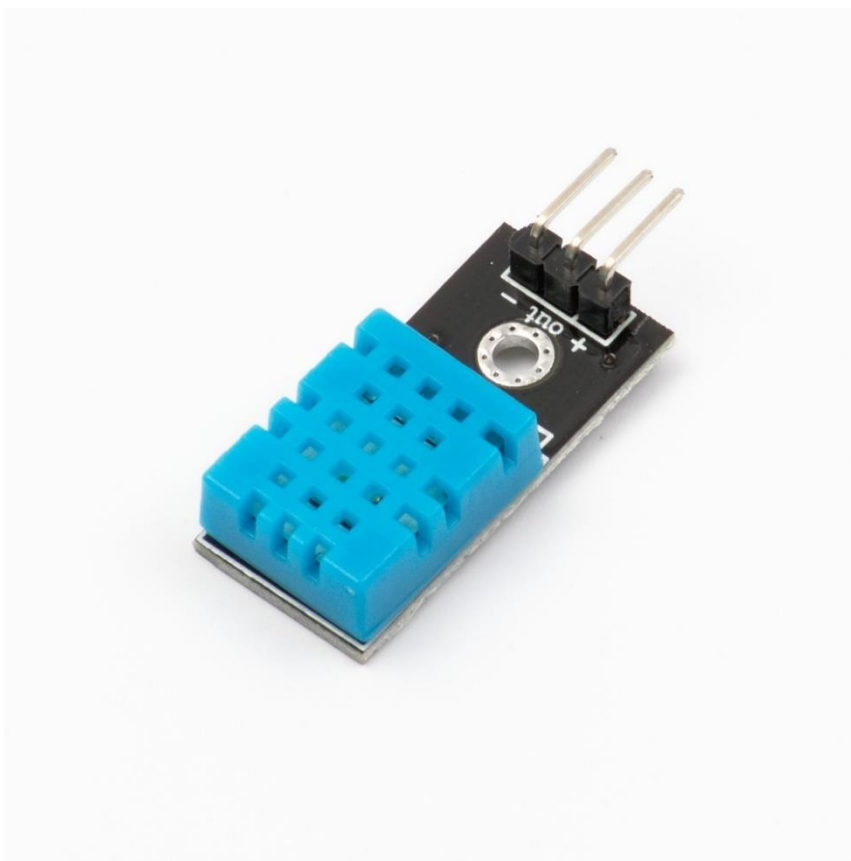
Pro projekt byla vybrána serverová skříň Legrand EvoLine 19“. Jedná se o standardní 19palcovou skříň vyrobenou z oceli, což zajišťuje její bytelnost s prosklenými dveřmi. Prosklené dveře jsou důležitým aspektem, jelikož skříň bude vybavena světelnou signalizací pro zamčení, otevření a signalizaci chyb. Skříň má výšku 47 U s nosností 1000 kg, a vnější rozměry 2380×600×600 mm. [24]



Obrázek 13 Serverová skříň Legrand EvoLine 19" [24]

3.9.3 Teplotní senzor

Jako vhodný teplotní senzor byl vybrán senzor DHT11. DHT11 je jednoduchý senzor na měření teploty vzduchu. Modul má v sobě integrovaný pullup rezistor i kondenzátor, tudíž nejsou zapotřebí žádné další součástky k propojení čidla s deskou. Napájecí napětí se pohybuje v rozmezí 3.5-5 V. Rozsah měření se pohybuje v rozmezí 0-60 °C s přesností ± 2 °C. Toto čidlo je i přes velikosti odchylek pro projekt dostačující. [25]



Obrázek 14 Teplotní senzor DHT11 [25]

3.9.4 Senzor vlhkosti

Jako senzor vlhkosti bude využit již dříve zmíněný teplotní senzor DHT11, který kromě teploty dokáže snímat i vlhkost. Vlhkost je možné snímat v rozmezí od 20 % RH až do 95 % RH s přesností ± 2 % RH. Ten bude sloužit pro detekci příliš vysoké úrovně vlhkosti. [25]

3.9.5 Senzor kouře

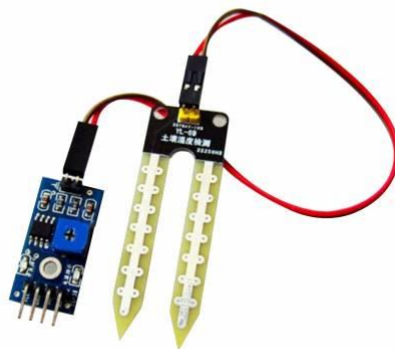
Jako vhodný senzor kouře byl vybrán senzor MQ - 135, který je schopný snímat NH₃, NO_x, alkoholy, benzeny, a pro projekt důležitý kouř a CO₂. Tento senzor pro svoje využití potřebuje napájení poskytující napětí 5 V a proud okolo 150 mA. Senzor lze připojit k Raspberry Pi prostřednictvím digitálního či analogového rozhraní. Lze jej využít v pracovním prostředí, ve kterém se teplota pohybuje mezi -20 až 50 °C a je schopný měřit v rozsahu až 1000 ppm. [26]



Obrázek 15 Senzor kouře MQ – 135 [26]

3.9.6 Senzor vytopení

Jako vhodný senzor vytopení byl vybrán senzor k detekci vlhkosti půdy, jde o jednoduchý senzor skládající se ze dvou elektrod, které se vzájemně nedotýkají, dojde-li však k vytopení, voda zapříčiní propojení obvodu, což je pak dále vyhodnoceno jako sepnutí. Pro zjednodušení vyhodnocování je součástí i komparátor LM393 který převádí analogový signál na digitální. [27]



Obrázek 16 Senzor na měření vlhkosti [27]

3.9.7 Adresovatelný led pásek

Jako vhodný led pásek byl vybrán Adresovatelný LED pásek se 30 LED/m. Tento pásek je ideální pro vytváření dynamických světelných efektů a je kompatibilní s Raspberry Pi, což umožňuje snadné programování a ovládání jednotlivých LED. Lze jej využít v pracovním prostředí, ve kterém se teplota pohybuje mezi -20 až 50 °C pracuje s napětím 5 V a výkonem 1,5 W. [28]



Obrázek 17 Adresovatelný led pásek [28]

3.9.8 Elektromagnetický zámek

Pro další úroveň zabezpečení bude do racku přidán bezpečnostní elektromagnetický zámek, který bude ovládán skrze Raspberry Pi. Díky tomu bude možné zajistit dveře proti nežádoucímu přístupu. Pracuje s napětím 12 VDC a proudem 2 A. [29]

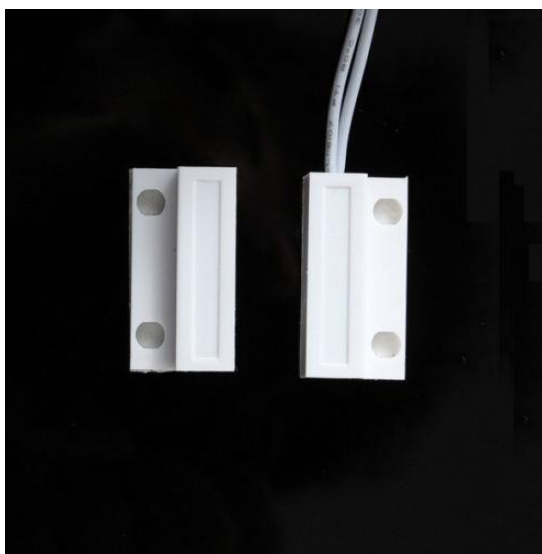


Obrázek 18 Elektromagnetický zámek [29]

3.9.9 Magnetický senzor

Jako vhodný senzor na kontrolu, zda jsou dvířka od skříně otevřena či zavřena byl vybrán jazýčkový magnetický kontakt MC-38. [30]

Funguje jako jednoduchý spínač, který se uzavře při přiblížení magnetu. Podporuje napětí až 30 V DC s maximálním proudem 500 mA a maximálním výkonem 1 W. Pro potřeby projektu je naprosto dostatečný. [30]



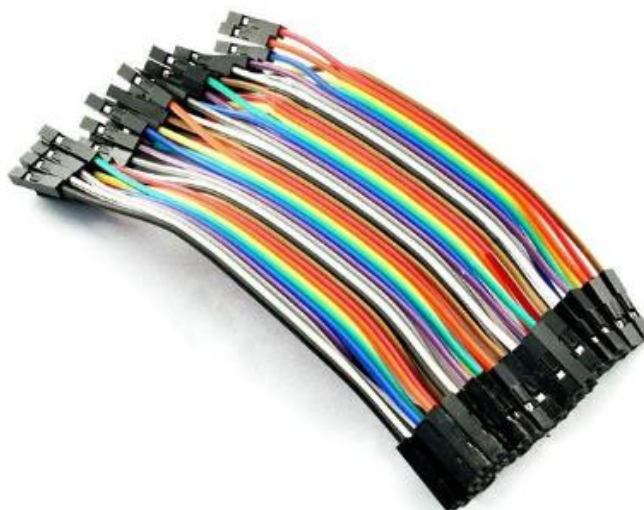
Obrázek 19 Magnetický senzor [30]

3.9.10 Napájecí zdroje

Pro potřeby napájení Raspberry Pi, adresovatelného led pásku a elektromagnetického zámku jsou zapotřebí externí zdroje. 1x microUSB 5 V určený pro Raspberry Pi Zero, 1x 5 V s možností napojení vodičů pro adresovatelný led pásek a 1x 12 V pro elektromagnetický zámek. [32,33,34]

3.9.11 Vodiče

K propojení jednotlivých komponent byly vybrány jednoduché snadno propojitelné vodiče bez potřeby pájení. [31]



Obrázek 20 Vodiče [31]

3.9.12 Tabulka cen hardwaru

Zde je uvedena tabulka s celkovými náklady na hardware, potřebnými k sestavení a realizaci projektu. [21,24,25,26,27,28,29,30,31,32,33,34]

Název komponentu	Cena
Raspberry Pi Zero 2 W	629 Kč
Serverová skříň	12 988 Kč
Senzor teploty a vlhkosti DHT11 (3x)	171 Kč
Senzor kouře MQ - 135	57 Kč
Senzor vytopení s komparátorem LM393	25 Kč
Adresovatelný led pásek	839 Kč
Elektromagnetický zámek	208 Kč
Magnetický senzor	23 Kč
Napájecí zdroje	577 Kč
Vodiče	299 Kč
Celkem	15 816 Kč

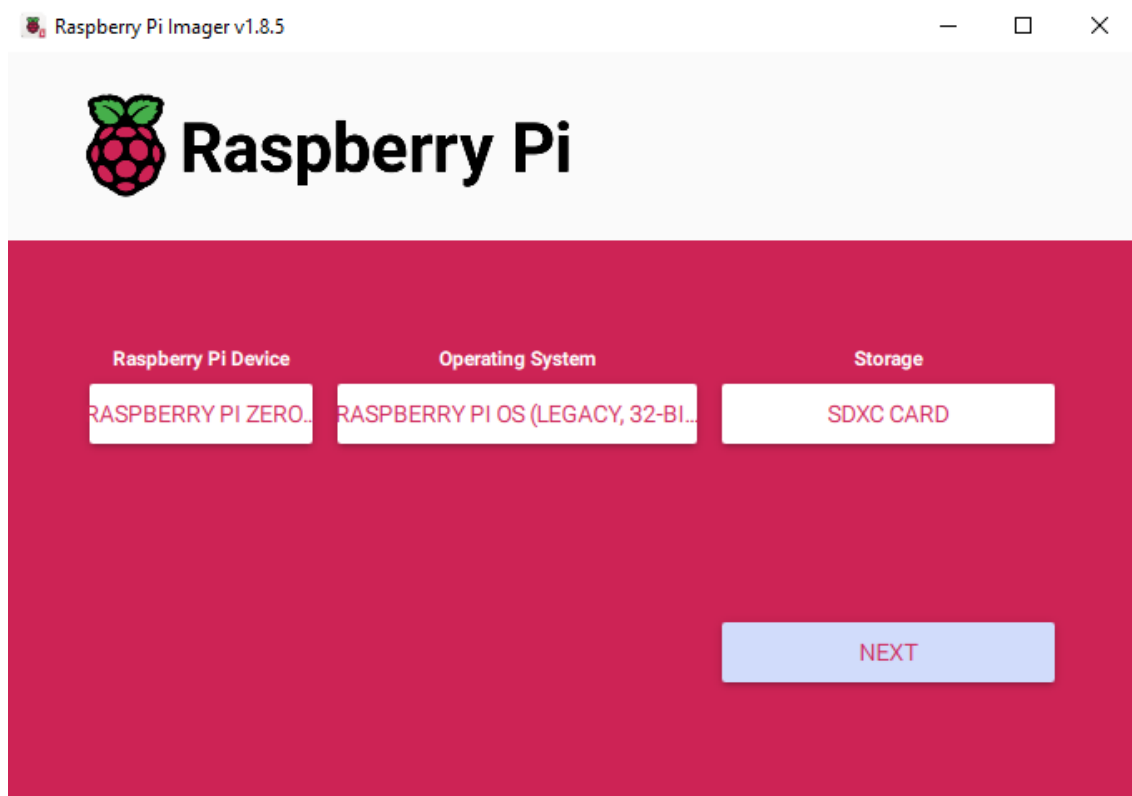
Tabulka 1 Tabulka cen hardwaru

4 Vlastní práce

Tato část práce je věnována konstrukci a programování vzorového zařízení, která zahrnuje detailní popis instalace, sestavení, naprogramování a testování funkčnosti v rámci serverové skříně. Zařízení má za úkol monitorovat teplotu a vlhkost uvnitř serverové skříně. Pomocí adresovatelného led pásku pak indikovat stav skříně (odemčeno, zamčeno, otevřeno a porucha). Dále kontrolovat přítomnost kouře a detekovat možnost vytopení skříně a všechny tyto informace odesílat do počítače s možností vzdáleného ovládání.

4.1 Instalace Raspberry Pi OS

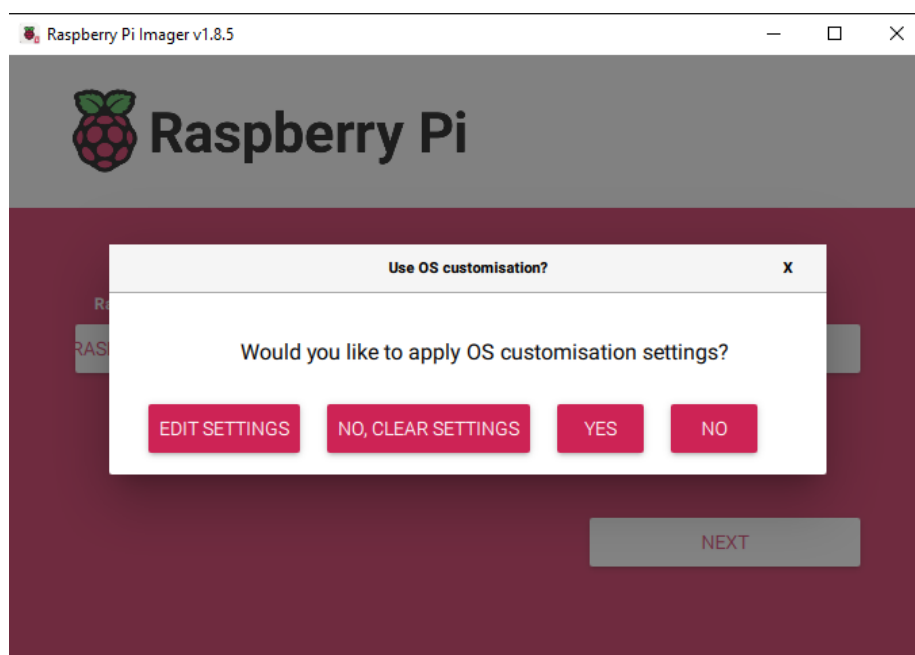
Prvním krokem je instalace samotného operačního systému Raspberry Pi OS. K instalaci je nutná microSD karta o kapacitě nejméně 8 GB, čtečka microSD karet a pro usnadnění procesu instalace také program Raspberry Pi Imager.



Obrázek 21 Raspberry Pi Imager

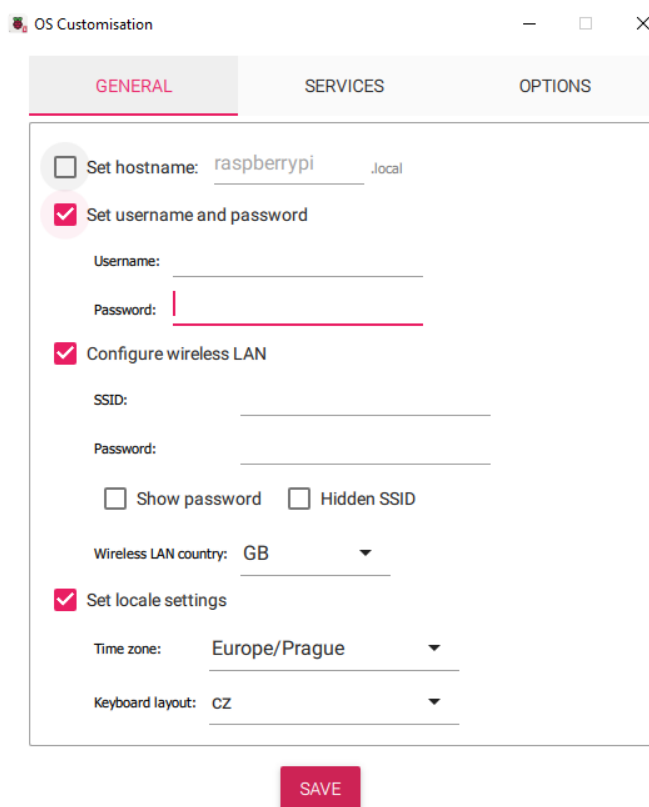
Po instalaci tohoto softwaru se zobrazí dialogové okno, které slouží k výběru operačního systému a místa pro instalaci. Z nabídky operačních systémů je třeba vybrat Raspberry Pi OS (32-bit) a jako uložení vybrat připojenou microSD kartu. Dále je nutné

stisknout tlačítko "next". Pote se zobrazí dialogové okno, kde je potřeba stisknout nejprve tlačítko "edit settings" pro otevření možností dalšího nastavení.

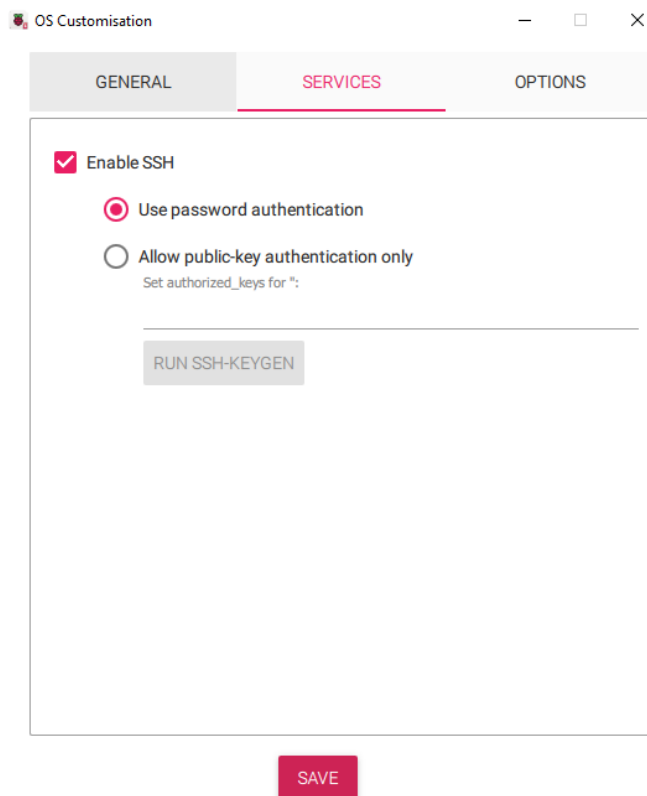


Obrázek 22 Raspberry Pi Imager Edit settings

V rámci tohoto nastavení je důležité aktivovat SSH přístup s autentizací pomocí hesla a také přidat konfiguraci WiFi sítě.



Obrázek 23 Raspberry Pi Imager nastavení WIFI



Obrázek 24 Raspberry Pi Imager nastavení SSH

Po provedení těchto změn je nutné uložit nastavení pomocí tlačítka "save" a následně v původním dialogovém okně zahájit proces zápisu na kartu stiskem tlačítka "yes". Po úspěšném dokončení zápisu lze microSD kartu bezpečně odpojit a vložit do Raspberry Pi.

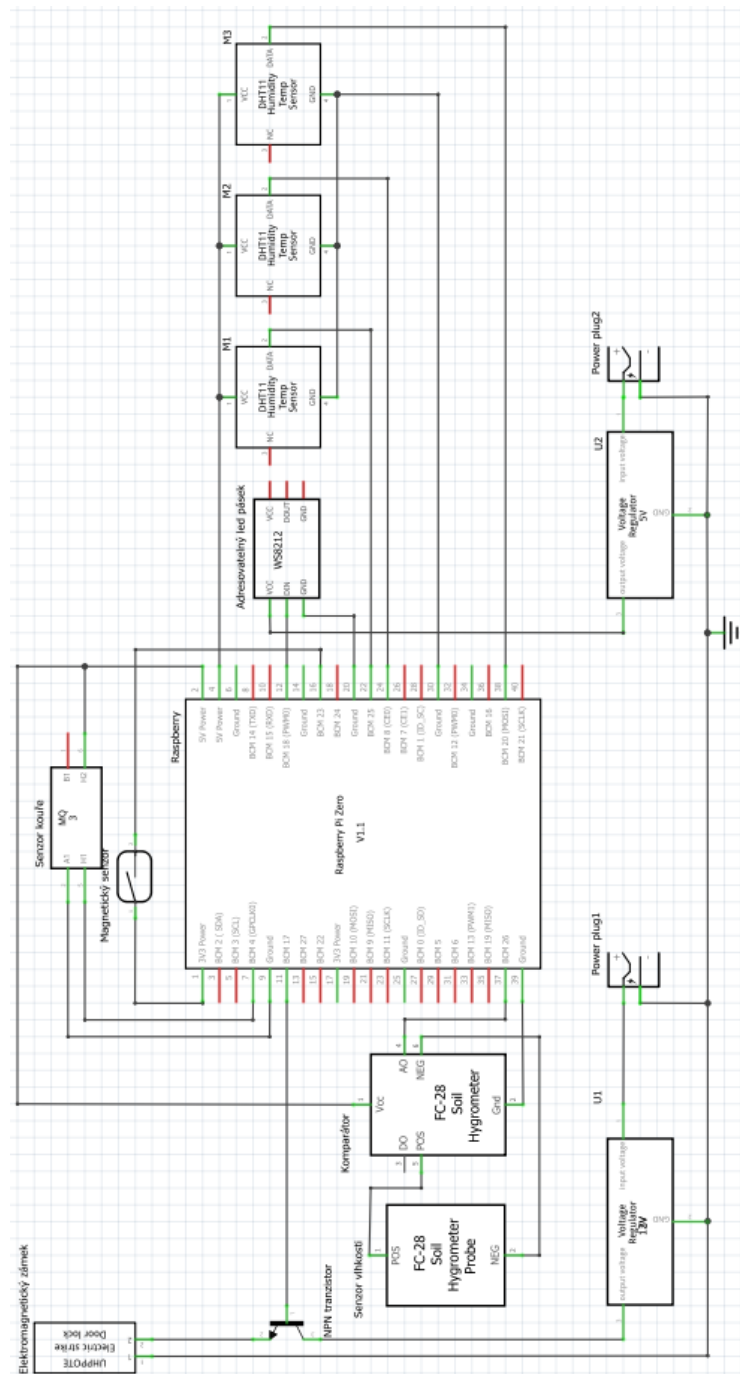
Poté již stačí samotné Raspberry Pi připojit k monitoru, klávesnici, myši a zdroji. Raspberry Pi Zero 2 W nemá žádné startovací tlačítko, spouští se automaticky připojením ke zdroji. Pokud známe IP adresu, na které je Raspberry Pi připojeno lze se na něj připojit pomocí PuTTY a pracovat v režimu příkazové řádky.

4.2 Zapojení

Tato část práce se zabývá praktickým zapojením projektu, kde je navrženo elektrotechnické.

4.2.1 Elektrotechnické schéma zapojení

Elektrotechnické schéma zapojení slouží ke znázornění propojení všech použitých komponent, které jsou reprezentovány normovanými značkami.



Obrázek 25 Elektrotechnické schéma zapojení

Zapojení senzorů teploty

Senzory teploty jsou zapojeny konektory VCC na zdroj 5 V, konektory GND na ground a DATA konektory na GPIO (General Purpose Input/Output) piny desky Raspberry PI. Přesněji na piny GPIO 24, 8 a 20, které budou v programu nastaveny jako Input.

Zapojení senzoru kouře

Senzor kouře je zapojen konektorem VCC na zdroj 5 V, konektorem GND na ground a DATA konektorem na GPIO pin 4 který bude v programu nastaven jako Input.

Zapojení senzoru vytopení

Senzor vytopení je připojen konektorem VCC na zdroj 5 V, konektorem GND na ground a DATA konektorem na GPIO pin 26 který bude v programu nastaven jako Input.

Zapojení adresovatelného led pásku

Adresovatelný led pásek je zapojen napájecím konektorem k výkonnějšímu 5 V zdroji z důvodu, že deska Raspberry Pi by nedokázala rozsvítit celý tento pásek. Konektory GND jsou uzemněny a datový kabel je připojen na pin GPIO 18

Zapojení Elektromagnetického zámku

Elektromagnetický zámek funguje tak, že když je po proudem je ve stavu odemčeno, v opačném případě je ve stavu zamčeno, a to z důvodu, aby při výpadku elektrické energie nedošlo k jeho odemčení. Zámek je připojen ke zdroji 12 V přes spínací tranzistor připojen na GPIO pin 17 který je v programu nastaven jako Output. GND kabel je opět připojen na ground.

Zapojení Magnetického senzoru

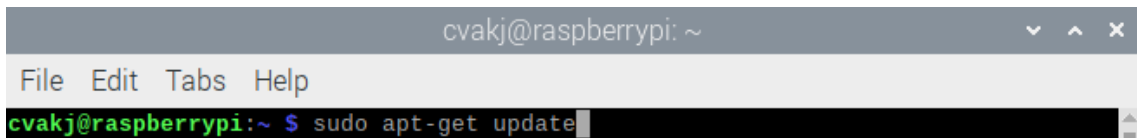
Magnetický senzor je připevněn vedle dveří skříně a je připojen na zdroj 3,3 V. Druhý kabel je připojen na GPIO pin 23, který je nastaven jako Input.

4.3 Instalace potřebných balíčků

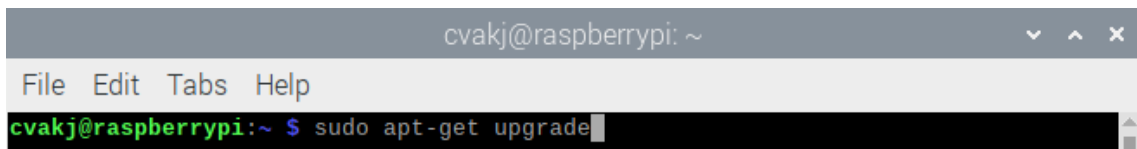
Po spuštění a zapojení Raspberry Pi a všech potřebných komponent je potřeba před samotným programováním nainstalovat nejnovější aktualizace. To je provedeno příkazy v konzoly.

```
sudo apt-get update
```

```
sudo apt-get upgrade
```

A screenshot of a terminal window titled 'cvakj@raspberrypi: ~'. The window has a menu bar with 'File', 'Edit', 'Tabs', and 'Help'. The terminal prompt is 'cvakj@raspberrypi:~ \$' and the command 'sudo apt-get update' is being entered.

Obrázek 26 sudo update

A screenshot of a terminal window titled 'cvakj@raspberrypi: ~'. The window has a menu bar with 'File', 'Edit', 'Tabs', and 'Help'. The terminal prompt is 'cvakj@raspberrypi:~ \$' and the command 'sudo apt-get upgrade' is being entered.

Obrázek 27 sudo upgrade

Dále je potřeba nainstalovat balíčky pro snadnou komunikaci s připojeným hardwarem.

Pro adresovatelný LED pásek příkazy:

```
sudo pip3 install rpi_ws281x
```

```
sudo pip3 install adafruit-circuitpython-neopixel
```

```
sudo python3 -m pip install --force-reinstall adafruit-blinka
```

A pro senzory teploty a vlhkosti

```
sudo pip3 install Adafruit_DHT
```

A screenshot of a terminal window titled 'cvakj@raspberrypi: ~'. The window has a menu bar with 'File', 'Edit', 'Tabs', and 'Help'. The terminal shows the command 'sudo pip3 install rpi_ws281x' and its output: 'Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple', 'Collecting rpi_ws281x', 'Downloading https://www.piwheels.org/simple/rpi-ws281x/rpi_ws281x-5.0.0-cp39-cp39-linux_armv6l.whl (119 kB)', a progress bar, and 'Installing collected packages: rpi_ws281x', 'Successfully installed rpi_ws281x-5.0.0'. Then the command 'sudo pip3 install adafruit-circuitpython-neopixel' is entered, and the output shows 'Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple' and 'Collecting adafruit-circuitpython-neopixel'.

Obrázek 28 instalace balíčků

4.4 Program Raspberry

Na desce Raspberry Pi je spuštěn samostatný program "RP_dp" psaný v pythonu, který má za úlohu měřit teploty, kontrolovat čidlo kouře a vytopení, dále sledovat odemčení a otevření dveří a tyto informace každých 10 vteřin zapisovat do souboru mereni.csv se kterým nadále pracuje webový server který je taktéž spuštěn na desce Raspberry Pi.

4.4.1 Import knihoven

Nejprve je zapotřebí nainportovat knihovny která budou dále v programu používány. Pro práci s časem je potřeba knihovna time, pro snadné ovládaní led pásku knihovny rpi_ws281x a argparse. Pro kontrolu GPIO pinů knihovna RPi.GPIO. A na závěr knihovny Adafruit_DHT a csv pro práci s DHT senzory a správu csv souborů.

```
3 # Importy knihoven
4 import time
5 from rpi_ws281x import *
6 import argparse
7 import RPi.GPIO as GPIO
8 import Adafruit_DHT
9 import csv
```

Obrázek 29 Import knihoven

4.4.2 Konfigurace pro led pásek

Dále je potřeba nakonfigurovat si proměnné díky kterým bude led pásek správně nastaven. Proměnné jsou popsány v programu a jsou zobrazeny níže na obrázku.

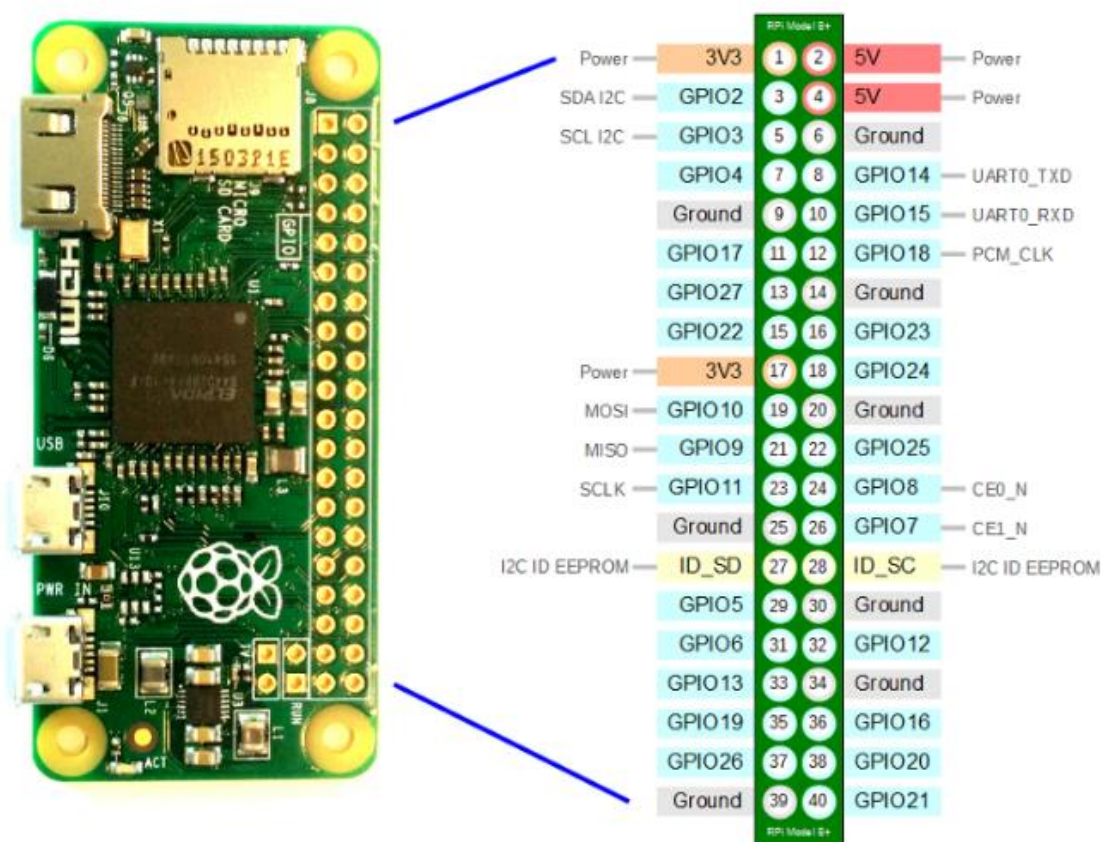
```
10
11
12 # konfigurace led pasku:
13 LED_COUNT      = 150      # Pocet ledek
14 LED_PIN        = 18       # GPIO pin kde je pripojen
15 LED_FREQ_HZ    = 800000   # Frekvence signalu
16 LED_DMA        = 10       # DMA kanál pro generování signálu
17 LED_BRIGHTNESS = 65       # Nastavení jasu
18 LED_INVERT     = False    # Invertovaný režim
19 LED_CHANNEL    = 0        # speciální nastavení pro vybrane GPIO piny
20
```

Obrázek 30 Nastavení konfigurace led pásku

4.4.3 Nastavení GPIO pinů

Dalším krokem je nastavení GPIO pinů.

Příkaz `GPIO.setmode(GPIO.BCM)` byl použit s knihovnou `RPi.GPIO` k nastavení schématu číslování pinů na BCM (Broadcom) mode. Raspberry Pi podporuje dva způsoby číslování GPIO pinů, BCM a BOARD a bylo potřeba jeden z těchto módů zvolit. BCM používá čísla pinů, jak jsou definována v dokumentaci Broadcom čipu, zatímco BOARD režim používá fyzická čísla pinů na konektoru.



Obrázek 31 Číslování GPIO pinů [36]

`GPIO.setwarnings(False)` slouží k vypnutí varovných hlášení, které knihovna `RPi.GPIO` generuje.

Piny 23, 4, 26 byly příkazem `GPIO.setup(X, GPIO.IN, pull_up_down = GPIO.PUD_DOWN)` nastaveny pro čtení dat a budou dále použity pro sběr dat z magnetického senzoru, senzoru kouře a senzoru vytopení, zatímco pin 17 byl nastaven jako výstup příkazem `GPIO.setup(17, GPIO.OUT)` a bude použit pro spínání elektromagnetického zámku.

Pokud je na GPIO přiveden signál 3,3 V vrací hodnotu 1, pokud jsou připojen na ground vrací hodnotu 0, pokud se ale pokusíme číst nepřipojený pin bude vracet náhodnou hodnotu 1 nebo 0. Proto je při definici input pinů použito: `pull_up_down = GPIO.PUD_DOWN`, což má za následek, pokud na input pinu není přečtena 1 automaticky je tento vstup brán jako 0.

```
21 GPIO.setmode(GPIO.BCM)
22 GPIO.setwarnings(False)
23 GPIO.setup(23, GPIO.IN, pull_up_down = GPIO.PUD_DOWN) # Vstup magnetickeho senzoru
24 GPIO.setup(4, GPIO.IN, pull_up_down = GPIO.PUD_DOWN) # Vstup senzoru koure
25 GPIO.setup(26, GPIO.IN, pull_up_down = GPIO.PUD_DOWN) # Vstup senzoru vytopení
26 GPIO.setup(17, GPIO.OUT) # Výstup na ovládání zámku
27
```

Obrázek 32 Nastavení GPIO pinů

4.4.4 Funkce na zbarvení led pásku

Jedná se o jednoduchou funkci, která má na starost zbarvení led pásku do určité barvy. Funkce má 2 vstupní proměnné, `strip`, což je již definovaná konfigurace led pásky a `color`, který představuje barvu a je zapsána jako tři čísla 0-255 oddělena čárkou.

Funkce funguje na principu for cyklu, kdy postupně na každou diodu zapíše požadovanou barvu.

```
29 def colorStripe(strip, color):
30     for i in range(strip.numPixels()):
31         strip.setPixelColor(i, color)
32     strip.show()
```

Obrázek 33 Funkce na zbarvení led pásku

4.4.5 Přípravení csv souboru

Program má získaná data zapisovat do csv souboru který bude dále používán, nejprve je ale potřeba takovýto soubor otevřít a pokud neexistuje tak vytvořit. To má na starost příkaz "with open". Ten vytvoří nebo otevře soubor `mereni.csv` a zápis bude do tohoto souboru pokračovat přidáváním nových hodnot na konec. To definuje `mode='a'`. Pokud by bylo potřeba, aby při každém spuštění byl soubor `mereni.csv` přepsán, postačí změnit `mode='a'` na `mode='w'`.

```
33 # Inicializace CSV souboru
34 with open('mereni.csv', mode='a', newline='') as file:
35     writer = csv.writer(file)
36     #writer.writerow(["Čas", "Teplota1", "Vlhkost1", "Teplota2",
37
```

Obrázek 34 Přípravení csv souboru

4.4.6 Main program

Main program se dělí na 2 části. Inicializační část, kde jsou definovány a nastaveny proměnné a část s hlavním cyklem, kde je napsán hlavní kód programu.

4.4.6.1 Inicializační část

Díky knihovnám jsou zde jsou definovány proměnné `strip`, ten si bere data z konfigurace led pásku, která proběhla na začátku programu a `sensorDHT` jenž je nastaven jako sensor DHT11 a nadále je s ním tak pracováno.

Dále jsou zde nastaveny čísla GPIO pinů pro DHT senzory a stavové proměnné do kterých jsou ukládána měřená a monitorovaná data pro následní ukládání do csv souboru.

Nakonec je zde definována i proměnná `counter` sloužící k uchování počtu iterací `while` cyklu, podle které je pak voláno zapisování do csv souboru.

```
37
38 # Main program:
39 if __name__ == '__main__':
40 |
41     # Nastavení led pásku.
42     strip = Adafruit_NeoPixel(LED_COUNT, LED_PIN, LED_FREQ_HZ, LED_DMA, LED_INVERT, LED_BRIGHTNESS, LED_CHANNEL)
43     strip.begin()
44     sensorDHT = Adafruit_DHT.DHT11 # Definování senzorů teploty
45     # Nastavení pinů pro senzory teplot.
46     pinDHT1 = 25
47     pinDHT2 = 8
48     pinDHT3 = 20
49     # Nastavení proměnných.
50     hum1w = 24 # Proměnné pro ukládání do csv
51     temp1w = 24
52     hum2w = 24
53     temp2w = 24
54     hum3w = 24
55     temp3w = 24
56     smokew = 0
57     floodw = 0
58     doorw = 0
59     lockw = 0
60
61     counter = 0 # Počítadlo
62
```

Obrázek 35 Část main programu

4.4.6.2 Hlavní cyklus

Tato část programu je spuštěna v nekonečné smyčce `while True:`

```
63 while True:
64     hum1, temp1 = Adafruit_DHT.read_retry(sensorDHT, pinDHT1) # Měření teplot
65     if hum1 > 0 and hum1 < 100 and temp1 > 0 and temp1 < 70: # Zápís teplot pokud jsou validní
66         temp1w=temp1
67         hum1w=hum1
68     hum2, temp2 = Adafruit_DHT.read_retry(sensorDHT, pinDHT2) # Měření teplot
69     if hum2 > 0 and hum2 < 100 and temp2 > 0 and temp2 < 70: # Zápís teplot pokud jsou validní
70         temp2w=temp2
71         hum2w=hum2
72     hum3, temp3 = Adafruit_DHT.read_retry(sensorDHT, pinDHT3) # Měření teplot
73     if hum3 > 0 and hum3 < 100 and temp3 > 0 and temp3 < 70: # Zápís teplot pokud jsou validní
74         temp3w=temp3
75         hum3w=hum3
76     #print('Temp={0:0.1f}*C Humidity={1:0.1f}%'.format(temp1w, hum1w))
77     #print('Temp={0:0.1f}*C Humidity={1:0.1f}%'.format(temp2w, hum2w))
78     #print('Temp={0:0.1f}*C Humidity={1:0.1f}%'.format(temp3w, hum3w))
79     if GPIO.input(23) and GPIO.input(17): # Zavreno a odemceno
80         lockw = 1 # Zapsání odemčení
81     if not (GPIO.input(23)): # Otevreno
82         doorw = 1 # Zapsání otevření
83     if not (GPIO.input(26)): # Vytopení
84         floodw = 1 # Zapsání Vytopení
85     if not (GPIO.input(4)): # Kour
86         colorStripe(strip, Color(255, 0, 0)) # Cervena
87         smokew = 1 # Zapsání Kore
88     elif not (GPIO.input(26)): # Vytopení
89         colorStripe(strip, Color(255, 145, 0)) # Oranzova
90     elif (GPIO.input(23)) and not GPIO.input(17): # Zvreno a zamceno
91         colorStripe(strip, Color(0, 0, 255)) # Modra
92     elif GPIO.input(23) and GPIO.input(17): # Zavreno a odemceno
93         colorStripe(strip, Color(0, 255, 0)) # Zelena
94     else: # Otevreno
95         GPIO.output(17, GPIO.LOW) # Vypnutí proudu do zámku jelikož už jsou dveře otevřené z důvodu ochrany zámku
96         colorStripe(strip, Color(255, 255, 255)) # Bila
97         time.sleep(1)
98     if counter >= 10: # Podmínka pro zápís do csv souboru
99         data = [time.strftime("%Y-%m-%d %H:%M:%S"), temp1w, hum1w, temp2w, hum2w, temp3w, hum3w, smokew, floodw, lockw, doorw]
100        counter = 0 # Nulování counteru a stavových proměnných
101        smokew = 0 # Nulování stavových proměnných
102        floodw = 0
103        doorw = 0
104        lockw = 0
105        with open('mereni.csv', mode='a', newline='') as file: # Zápís do csv souboru
106            writer = csv.writer(file)
107            writer.writerow(data)
108    else:
109        counter += 1
```

Obrázek 36 Hlavní While cyklus

Nejprve se načítají data ze senzorů teploty a vlhkosti, následně jsou tato data porovnána, zda jsou v logických mezích pro eliminaci chybných čtení (vlhkost v rozmezí 0-100 % a teplota 0-70 °C) a následně jsou zapsána do proměnných určených pro ukládání do csv souboru.

```
64     hum1, temp1 = Adafruit_DHT.read_retry(sensorDHT, pinDHT1) # Měření teplot
65     if hum1 > 0 and hum1 < 100 and temp1 > 0 and temp1 < 70: # Zápís teplot pokud jsou validní
66         temp1w=temp1
67         hum1w=hum1
68     hum2, temp2 = Adafruit_DHT.read_retry(sensorDHT, pinDHT2) # Měření teplot
69     if hum2 > 0 and hum2 < 100 and temp2 > 0 and temp2 < 70: # Zápís teplot pokud jsou validní
70         temp2w=temp2
71         hum2w=hum2
72     hum3, temp3 = Adafruit_DHT.read_retry(sensorDHT, pinDHT3) # Měření teplot
73     if hum3 > 0 and hum3 < 100 and temp3 > 0 and temp3 < 70: # Zápís teplot pokud jsou validní
74         temp3w=temp3
75         hum3w=hum3
```

Obrázek 37 Čtení a kontrola teploty a vlhkosti

Dále jsou testovány input piny, dle kterých jsou nastavovány stavové proměnné.

Podmínka `if GPIO.input(23) and GPIO.input(17):` značí, pokud je hodnota 1 na pinu 23 a hodnota 1 na pinu 17, znamená to, že dveře jsou zavřeny ale odemčeny, následně bude do stavové proměnné "lockw" zapsána hodnota 1 značící odemčení dveří.

```
79     if GPIO.input(23) and GPIO.input(17): # Zavřeno a odemčeno
80         lockw = 1 # Zapsání odemčení
```

Obrázek 38 Kontrola odemčení

Obdobně fungují i další podmínky.

Podmínka pro čtení otevření dveří. Pokud na pinu 23 bude hodnota 0 znamená to, že magnetické čidlo instalované na roh dveří serverové skříně bylo rozpojeno, což značí otevření dveří. Následně je do proměnné "doorw" zapsána hodnota 1.

```
81     if not (GPIO.input(23)): # Otevřeno
82         doorw = 1 # Zapsání otevření
```

Obrázek 39 Kontrola otevření

Podmínka pro kontrolu vytopení. Pokud na pinu 26 bude hodnota 0 znamená to, že byl senzor vytopení instalovaný v dolní části skříně sepnut, což značí vytopení. Následně je do proměnné "floodw" zapsána hodnota 1.

```
83     if not (GPIO.input(26)): # Vytopení
84         floodw = 1 # Zapsání Vytopení
```

Obrázek 40 Podmínka vytopení

Podmínka pro kontrolu kouře uvnitř skříně. Pokud na pinu 4 bude hodnota 0 znamená to, že byl senzor kouře aktivován. Následně je na indikačním led pásku nastavena červená barva značící kouř a do proměnné "smokew" zapsána hodnota 1.

```
85     if not (GPIO.input(4)): # Kour
86         colorStripe(strip, Color(255, 0, 0)) # Cervena
87         smokew = 1 # Zapsání Kouře
```

Obrázek 41 Podmínka kouře

Dále navazují identické podmínky, tentokrát ale v elif struktuře a zde slouží k nastavování světelné signalizace adresovatelného led pásku. Jak již bylo zmíněno červená barva pro kouř, oranžová pro vytopení, modrá pro stav zavřeno a zamčeno, zelená pro zavřeno a odemčeno. Bílá barva se rozsvítí v moment, kdy ani jedna z podmínek výše není splněna, tato situace nastane jedině v moment, kdy skříň bude otevřena. Následně bude vypnuto napájení pinu 17, který odemykal elektromagnetický zámek skříně. Tím je docíleno, že zavřením dveří dojde k jejich zamčení a cívka uvnitř zámku nebude zbytečně přetěžována. Bílá barva byla zvolena, z důvodu poskytování nejintenzivnějšího světla, jelikož se počítá s tím, že skříň bude otevřena ve výjimečných případech, a to většinou z důvodu servisu nebo údržby, kdy je světlo vyžadováno.

Nakonec je vyvolán příkaz `time.sleep(1)`, který pozastaví běh programu na 1 sekundu z důvodu, že není potřeba provádět měření a testování častěji.

```

88     elif not (GPIO.input(26)): # Vytopeni
89         colorStripe(strip, Color(255, 145, 0)) # Oranzova
90     elif (GPIO.input(23) and not GPIO.input(17)): # Zvreno a zamceno
91         colorStripe(strip, Color(0, 0, 255)) # Modra
92     elif GPIO.input(23) and GPIO.input(17): # Zavreno a odemceno
93         colorStripe(strip, Color(0, 255, 0)) # Zelena
94     else: # Otevreno
95         GPIO.output(17, GPIO.LOW) # Vypnutí proudu do zámku jelikož už jsou dveře otevřené z důvodu ochrany zámku
96         colorStripe(strip, Color(255, 255, 255)) # Bíla
97         time.sleep(1)

```

Obrázek 42 Zbarvení led pásku dle podmínek

Poslední část programu v nekonečném cyklu `while` má za úlohu počítat jeho iterace.

Dosáhne-li program 10 iterací dojde k přípravě dat, vynulování stavových proměných a následnému zapsání hodnot do souboru `mereni.csv`

Nedosáhne-li počítadlo iterací čísla ≥ 10 dojde k jeho navýšení o 1 a celý cyklus `while` běží od znovu.

Stavové proměnné se nulují vždy až po 10 iteracích, a to z důvodu rizika zamlčení odemčení a otevření skříně. Pokud by se nulovala po každém cyklu existovala by možnost odemknout otevřít a opět zavřít skříň v 10 sekundách mezi zapisováním dat, která by ale nebyla zaznamenána.

```

98     if counter >= 10: # Podmínka pro zápis do csv souboru
99         data = [time.strftime("%Y-%m-%d %H:%M:%S"), temp1w, hum1w, temp2w, hum2w, temp3w, hum3w, smokew, floodw, lockw, doorw]
100        counter = 0 # Nulování counteru a stavových proměných
101        smokew = 0 # Nulování stavových proměných
102        floodw = 0
103        doorw = 0
104        lockw = 0
105        with open('mereni.csv', mode='a', newline='') as file: # Zápis do csv souboru
106            writer = csv.writer(file)
107            writer.writerow(data)
108    else:
109        counter += 1

```

Obrázek 43 Zápis do csv souboru

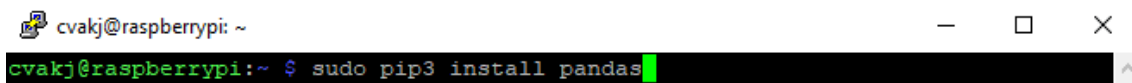
4.5 Webový server

Z důvodů jednoduchosti bylo zvolena možnost vytvořit webový server přímo na Raspberry Pi, Hlavním důvodem byla zjednodušená možnost přidat na server ovládání odemčení dveří pomocí tlačítka. Dále by měl server číst data z vygenerovaného csv souboru a zobrazovat je ve formě grafů.

4.5.1 Instalace potřebných balíčků pro WS

Nejprve je zapotřebí nainstalovat nezbytné balíčky se kterými budeme dále pracovat. Balíčky nainstalujeme vepsáním následujících příkazů do terminálu

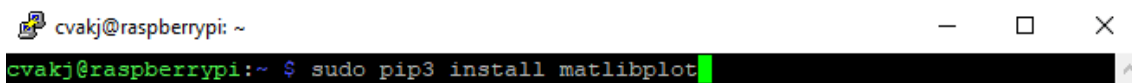
```
sudo pip3 install pandas
```



```
cvakj@raspberrypi: ~  
cvakj@raspberrypi:~ $ sudo pip3 install pandas
```

Obrázek 44 Instalace pandas

```
sudo pip3 install matplotlib
```



```
cvakj@raspberrypi: ~  
cvakj@raspberrypi:~ $ sudo pip3 install matplotlib
```

Obrázek 45 Instalace matplotlib

Balíček pandas je používán pro práci s csv soubory. Zatímco balíček matplotlib slouží jako nástroj pro generování a ukládání grafů.

4.5.2 Import knihoven pro WS

Na začátku programu je opět nutné naimportovat si knihovny se kterými bude dále pracováno. Knihovna RPi.GPIO slouží k ovládání GPIO pinů. Knihovna os poskytuje funkce pro interakci s operačním systémem a je dále použita za účelem čtení teploty procesoru. Dále je zde naimportována knihovna pandas sloužící na práci s csv soubory a knihovna matplotlib, která umožňuje jednoduchou tvorbu grafů. Na závěr jsou naimportovány knihovny na BaseHTTPRequestHandler a HTTPServer, které se používají pro vytvoření základního HTTP serveru v Pythonu a urlparse obsahující funkce pro práci s URL.

```
3 import RPi.GPIO as GPIO  
4 import os  
5 import pandas as pd  
6 import matplotlib.pyplot as plt  
7 from http.server import BaseHTTPRequestHandler, HTTPServer  
8 from urllib.parse import urlparse  
9
```

Obrázek 46 Import knihoven pro WS

4.5.3 Nastavení komunikace

Pro spuštění webového serveru na Raspberry Pi potřebujeme znát jeho IP adresu v síti kde je připojeno.

IP adresu získáme jednoduchým příkazem `ifconfig`, po vložení do terminálu. Jelikož je Raspberry Pi připojeno na síti pomocí wifi nalezneme jeho IP adresu pod `wlan0`:

```
cvakj@raspberrypi:~ $ ifconfig
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 26 bytes 2744 (2.6 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 26 bytes 2744 (2.6 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.50.13 netmask 255.255.255.0 broadcast 192.168.50.255
    inet6 fe80::c8c2:857a:da4f:1102 prefixlen 64 scopeid 0x20<link>
    ether b8:27:eb:b0:2a:99 txqueuelen 1000 (Ethernet)
    RX packets 22643 bytes 4104532 (3.9 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 2475 bytes 891699 (870.7 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

cvakj@raspberrypi:~ $ █
```

Obrázek 47 `ifconfig`

Tuto hodnotu si zapíšeme. Jedná se o IP adresu zařízení, na kterém běží webový server. Toto nastavení určuje, na jaké síťové adrese bude server naslouchat příchozím požadavkům.

Dále je nastaveno číslo portu, na kterém bude server naslouchat. Nejčastěji bývá používán port 8000 jelikož je brán jako bezpečný.

```
10 host_name = '192.168.50.13' # IP adresa Raspberry Pi
11 host_port = 8000
12
```

Obrázek 48 Nastavení komunikace

4.5.4 Funkce pro generování grafů z csv souboru

Pro tvorbu a ukládání grafů je vytvořena funkce `generate_and_save_graphs()`. Funkce nejprve načte pomocí knihovny `pandas` požadovaný csv soubor, který si uloží do proměnné `df`.

Dále je proces samotného tvoření a uložení grafu:

1. Nejprve je nastavena velikost grafu.
2. Poté je vytvořen a pojmenován graf z nadefinovaných sloupců s příslušným názvem
3. Dále je vytvořen název celého grafu, popsány osy X a Y a vytvořena legenda
4. Z důvodu rizika nepřehlednosti informací na ose X, jsou tyto informace pootočený o 90° a je jim nastaveno automatické upravení velikosti a pozic tak, aby se minimalizovalo překrývání mezi nimi, čímž je zajištěna jejich dobrá čitelnost
5. Na konec je soubor pojmenován a uložen do určeného adresáře.

```
14 def generate_and_save_graphs():
15     df = pd.read_csv('/home/cvakj/mereni.csv')
16     # Vykreslení grafu Teploty1
17     plt.figure(figsize=(10, 5))
18     plt.plot(df['Cas'], df['Teplota1'], label='TeplotaUP') # Vyber sloupcu pro graf
19     plt.title('Teplota v case') # Nazev grafu
20     plt.xlabel('Cas') # Popis osy X
21     plt.ylabel('Teplota (*C)') # Popis osy Y
22     plt.legend()
23     plt.xticks(rotation=90) # Otoceni textu na ose x o 90 stupnu
24     plt.tight_layout() # Zajisteni ze data na ose x budou citelna
25     plt.savefig('/home/cvakj/temperature_graphUP.png') # Uklozeni obrazku
26     plt.close()
```

Obrázek 49 Tvorba grafu *TeplotyUP*

Takto jsou vytvořeny grafy pro všechny monitorované veličiny (*TeplotaUP*, *TeplotaMID*, *TeplotaDOWN*, *Indikace Koure*, *Indikace Vytopeni*, *Odemceni* a *Otevreni*)

```

14 def generate_and_save_graphs():
15     df = pd.read_csv('/home/cvakj/mereni.csv')
16     # Vykreslení grafu Teploty1
17     plt.figure(figsize=(10, 5))
18     plt.plot(df['Cas'], df['Teplota1'], label='TeplotaUP') # Vyber sloupce pro graf
19     plt.title('Teplota v case') # Nazev grafu
20     plt.xlabel('Cas') # Popis osy X
21     plt.ylabel('Teplota (°C)') # Popis osy Y
22     plt.legend()
23     plt.xticks(rotation=90) # Otoceni textu na ose x o 90 stupnu
24     plt.tight_layout() # Zajisteni ze data na ose x budou citelna
25     plt.savefig('/home/cvakj/temperature_graphUP.png') # Uklozeni obrazku
26     plt.close()
27     # Vykreslení grafu Teploty2
28     plt.figure(figsize=(10, 5))
29     plt.plot(df['Cas'], df['Teplota2'], label='TeplotaMID')
30     plt.title('Teplota v case')
31     plt.xlabel('Cas')
32     plt.ylabel('Teplota (°C)')
33     plt.legend()
34     plt.xticks(rotation=90)
35     plt.tight_layout()
36     plt.savefig('/home/cvakj/temperature_graphMID.png')
37     plt.close()
38     # Vykreslení grafu Teploty3
39     plt.figure(figsize=(10, 5))
40     plt.plot(df['Cas'], df['Teplota3'], label='TeplotaDOWN')
41     plt.title('Teplota v case')
42     plt.xlabel('Cas')
43     plt.ylabel('Teplota (°C)')
44     plt.legend()
45     plt.xticks(rotation=90)
46     plt.tight_layout()
47     plt.savefig('/home/cvakj/temperature_graphDOWN.png')
48     plt.close()
49     # Vykreslení grafu Indikace Koure
50     plt.figure(figsize=(10, 5))
51     plt.plot(df['Cas'], df['Kour'], label='Indikace Koure')
52     plt.title('Indikace Koure')
53     plt.xlabel('Cas')
54     plt.ylabel('Indikace Koure')
55     plt.legend()
56     plt.xticks(rotation=90)
57     plt.tight_layout()
58     plt.savefig('/home/cvakj/Smoke_graph.png')
59     plt.close()
60     # Vykreslení grafu Indikace Vytopeni
61     plt.figure(figsize=(10, 5))
62     plt.plot(df['Cas'], df['Vytopeni'], label='Indikace Vytopeni')
63     plt.title('Indikace Vytopeni')
64     plt.xlabel('Cas')
65     plt.ylabel('Indikace Vytopeni')
66     plt.legend()
67     plt.xticks(rotation=90)
68     plt.tight_layout()
69     plt.savefig('/home/cvakj/Flood_graph.png')
70     plt.close()
71     # Vykreslení grafu Odemceni
72     plt.figure(figsize=(10, 5))
73     plt.plot(df['Cas'], df['Zarek'], label='Odemceni')
74     plt.title('Indikace Odemceni')
75     plt.xlabel('Cas')
76     plt.ylabel('Indikace Odemceni')
77     plt.legend()
78     plt.xticks(rotation=90)
79     plt.tight_layout()
80     plt.savefig('/home/cvakj/Lock_graph.png')
81     plt.close()
82     plt.xticks(rotation=90)
83     plt.tight_layout()
84     # Vykreslení grafu Otevreni
85     plt.figure(figsize=(10, 5))
86     plt.plot(df['Cas'], df['Dvere'], label='Otevreni')
87     plt.title('Indikace Otevreni')
88     plt.xlabel('Cas')
89     plt.ylabel('Indikace Otevreni')
90     plt.legend()
91     plt.xticks(rotation=90)
92     plt.tight_layout()
93     plt.savefig('/home/cvakj/Door_graph.png')
94     plt.close()
95

```

Obrázek 50 Tvorba všech grafů z csv souboru

4.5.5 Funkce pro získání teploty čipu

Informace o teplotě čipu mohou být zdrojem informací v jakém stavu se nachází samotné Raspberry Pi, díky nimž může být předejito neočekávaným problémům.

Samotná informace o teplotě je snadno získatelná pomocí knihovny `os`, kdy se vrátí hodnota po provedení příkazu `os.popen("/usr/bin/vcgenclmd measure_temp").read()`. Tato hodnota je uložena do proměnné `tmp` a funkce ji vrací jako návratovou hodnotu.

```
96 # Ziskani teploty cipu
97 def getTemperature():
98     temp = os.popen("/usr/bin/vcgenclmd measure_temp").read()
99     return temp
```

Obrázek 51 Funkce na čtení teploty čipu

4.5.6 Funkce pro nastavení GPIO pinu

Příkaz `GPIO.setmode(GPIO.BCM)` je použit s knihovnou `RPi.GPIO` k nastavení schématu číslování pinů na BCM (Broadcom) mode.

Dále byly vypnuty varovná hlášení, která knihovna `RPi.GPIO` generuje. GPIO pin 17 byl nastaven jako výstupní pin.

```
101 # Priprava GPIO
102 def setupGPIO():
103     GPIO.setmode(GPIO.BCM)
104     GPIO.setwarnings(False)
105     GPIO.setup(17, GPIO.OUT)
106
```

Obrázek 52 Nastavení GPIO pinů pro zámeček

4.5.7 Vytvoření třídy MyServer

Třída `MyServer` obsahuje metody pro zpracování různých typů HTTP požadavků (GET a POST), posílání souborů (obrázků), posílání HTML obsahu, nastavení HTTP hlaviček, a přesměrování. Slouží jako základ webového serveru na Raspberry Pi

```
107 class MyServer(BaseHTTPRequestHandler):
```

Obrázek 53 Třída `MyServer`

Metoda do_GET(self)

Jedná se o metodu pro ošetření GET požadavků. Nejdříve analyzuje požadovanou cestu URL a sestaví cestu k požadovanému souboru. Dále je kontrolováno, zda je soubor obrázek, pokud ano je zavolána metoda `send_image` v opačném případě je použita metoda `send_html`.

```
108     def do_GET(self):
109         parsed_path = urlparse(self.path)
110         file_path = '/home/cvakj' + parsed_path.path # Pridani cesty
111
112         if parsed_path.path.endswith(".png"):
113             self.send_image(file_path) # Predani cesty k souboru jako argument
114         else:
115             self.send_html()
116
```

Obrázek 54 Metoda do_GET

Metoda send_image(self)

Tato metoda se pokusí otevřít požadovaný obrázek pro čtení v binárním režimu.

Uspěje-li odešle HTTP odpověď 200 což znamená úspěch. Dále nastaví hlavičku pro typ obsahu jako `image/png` a ukončí hlavičky HTTP odpovědi. Nakonec zapíše obsah souboru do těla HTTP odpovědi.

Pokud se obrázek nepodařilo načíst nebo neexistuje odešle se HTTP chyba 404 se zprávou `File Not Found`.

```
117     def send_image(self, file_path): # Pridani argumentu file_path
118         try:
119             with open(file_path, 'rb') as file:
120                 self.send_response(200)
121                 self.send_header('Content-type', 'image/png')
122                 self.end_headers()
123                 self.wfile.write(file.read())
124         except FileNotFoundError:
125             self.send_error(404, 'File Not Found')
126
```

Obrázek 55 Metoda send_image

Metoda `send_html(self)`

Tato metoda po jejím zavolání nejdříve vygeneruje a uloží požadované grafy a sestaví HTML obsah stránky jako dlouhý textový řetězec který následně uloží do proměnné

HTML obsahuje základní strukturu s hlavičkou a tělem, nadpis stránky, aktuální teplotu čipu Raspberry Pi, formulář s metodou POST obsahující 2 tlačítka (Zamknout, Odemknout) a sérii obrázků které byly předtím vygenerovány a uloženy.

Poté získá aktuální teplot čipu, odešle http hlavičky a zapíše formátovaný HTML obsah zpět do těla HTTP odpovědi , která se odesílá zpět klientovi.

```
160     def send_html(self):
161
162         generate_and_save_graphs()
163         html_content = '''
164         <html>
165         <body style="width:960px; margin: 20px auto;">
166         <h1>Chytra serverova skrin</h1>
167         <p>Aktualni teplota cipu Raspberry Pi: {}</p>
168         <form action="/" method="POST">
169             Zamek:
170             <input type="submit" name="submit" value="Odemknout">
171             <input type="submit" name="submit" value="Zamknout">
172         </form>
173         
174         
175         
176         
177         
178         
179         
180         
181         
182         
183         </body>
184         </html>
185         ...
186         temperature = getTemperature()
187         self.do_HEAD()
188         self.wfile.write(html_content.format(temperature[5:]).encode("utf-8"))
```

Obrázek 56 Metoda `send_html`

Metoda `do_HEAD(self)`

Metoda pro nastavení HTTP hlaviček odpovědi.

```
153     def do_HEAD(self):
154         self.send_response(200)
155         self.send_header('Content-type', 'text/html')
156         self.end_headers()
157
```

Obrázek 57 Metoda `do_HEAD`

Metoda `_redirect(self, path)`

Metoda na přesměrování klienta.

```
158     def _redirect(self, path):
159         self.send_response(303)
160         self.send_header('Content-type', 'text/html')
161         self.send_header('Location', path)
162         self.end_headers()
163
```

Obrázek 58 Metoda `_redirect`

Metoda `do_POST(self)`

Metoda nejdříve získá délku obsahu požadavku, poté přečte data z těla požadavku a rozdělí je na klíč a hodnotu.

Dále se nastaví GPIO a podle hodnoty v `post_data` se buď zapne nebo vypne GPIO pin 17 ovládající zámek serverové skříně.

Nakonec je zavoláno přesměrování což způsobí znovunačtení stránky.

```
164     def do_POST(self):
165         content_length = int(self.headers['Content-Length'])
166         post_data = self.rfile.read(content_length).decode("utf-8")
167         post_data = post_data.split("=")[1]
168
169         setupGPIO()
170
171         if post_data == 'Odemknout':
172             GPIO.output(17, GPIO.HIGH)
173         else:
174             GPIO.output(17, GPIO.LOW)
175         self._redirect('/') # Presmerovani zpet na zakladni URL
176
```

Obrázek 59 Metoda `do_POST`

4.5.8 Main program

V main programu je spuštěn webový server za pomoci importovaných knihoven a informací, které byly dříve zjištěny a uloženy. Server o svém spuštění informuje vypsáním do příkazové řádky. A je spuštěn do doby, dokud není uživatelsky přerušeno.

```
177 if __name__ == '__main__':
178     http_server = HTTPServer((host_name, host_port), MyServer)
179     print("Server spusten - %s:%s" % (host_name, host_port))
180
181     try:
182         http_server.serve_forever()
183     except KeyboardInterrupt:
184         http_server.server_close()
```

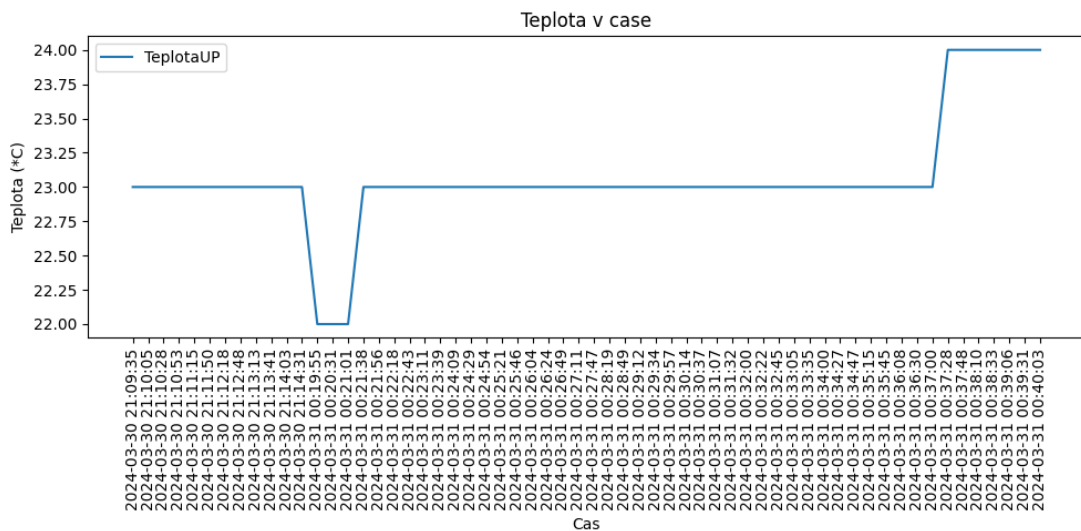
Obrázek 60 Main program

5 Výsledky a diskuse

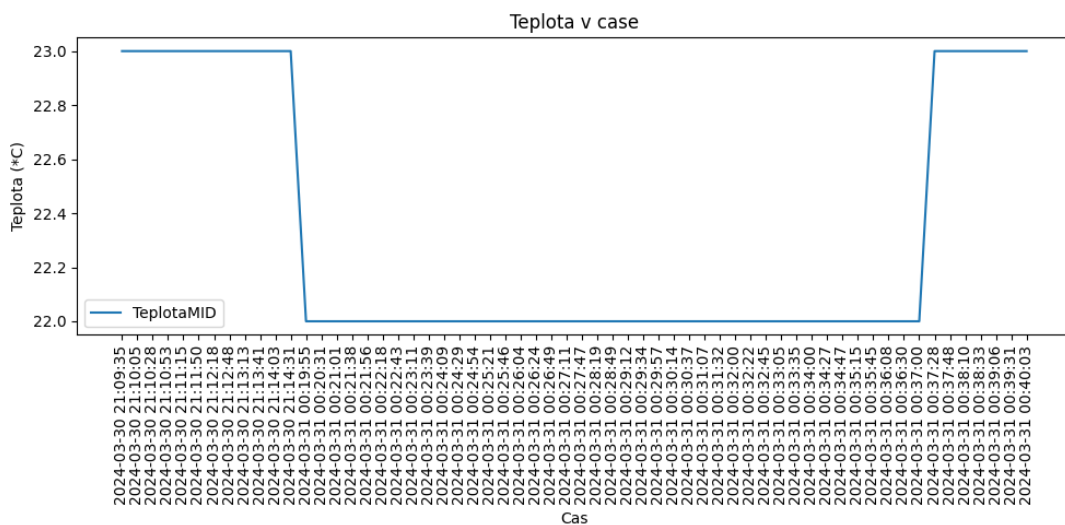
Výsledkem práce je sestavené, funkční a otestované zařízení. Raspberry Pi čte data ze senzorů na které reaguje svými výstupy, respektive světelnou signalizací na adresovatelném led pásku a taktéž tato data ukládá do csv souboru se kterým dále pracuje webový server. Webový server je stejně jako hlavní program spuštěn na Raspberry Pi a jeho úlohou je číst data z csv souboru, generovat z těchto dat grafy a následně je zobrazovat uživateli. Dále server zobrazuje aktuální stav teploty čipu a implementuje funkci odemykání/zamykání serverové skříně pomocí tlačítka.

5.1 Ukázka Grafů

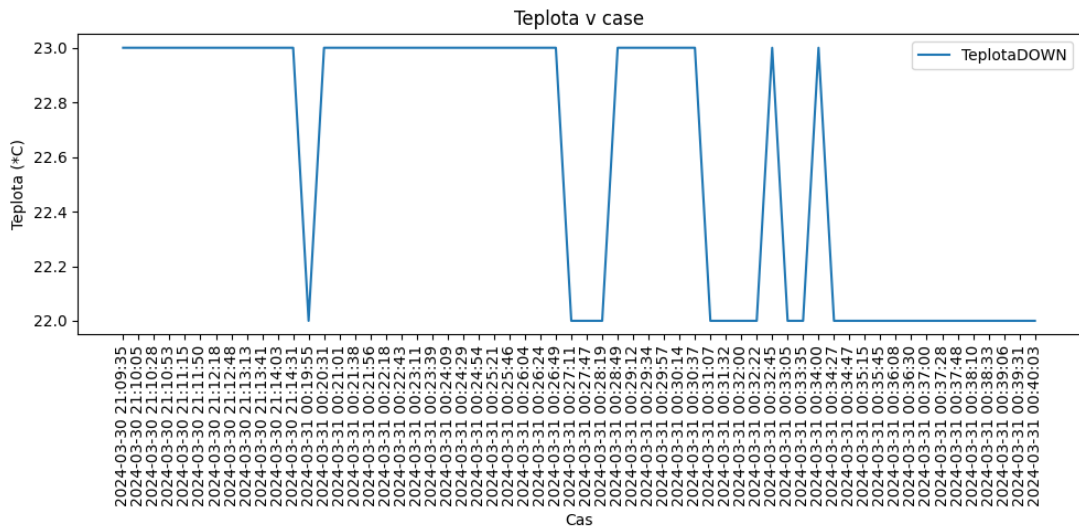
Ukázka grafů, které generuje, ukládá a zobrazuje webový server



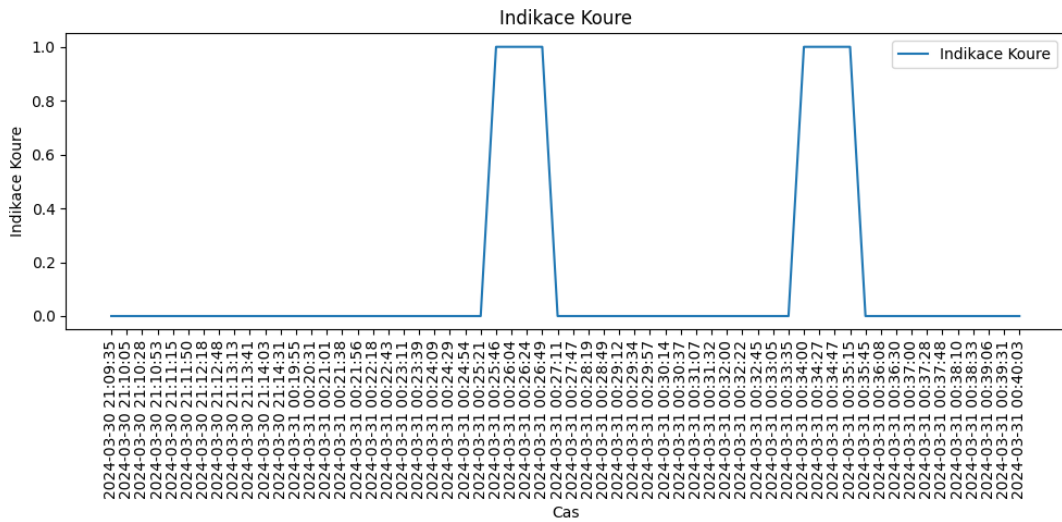
Graf 1 TeplotaUP v čase



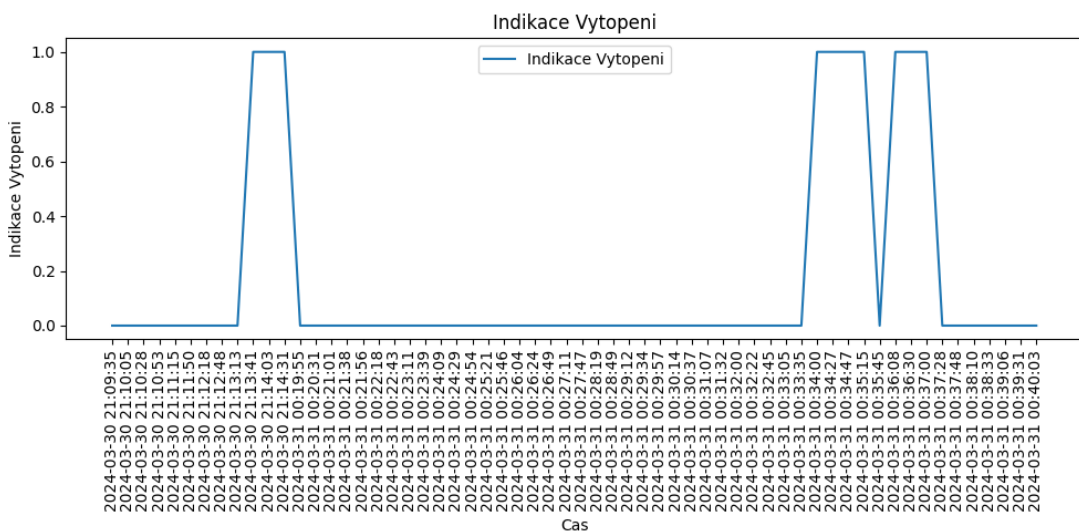
Graf 2 TeplotaMID v čase



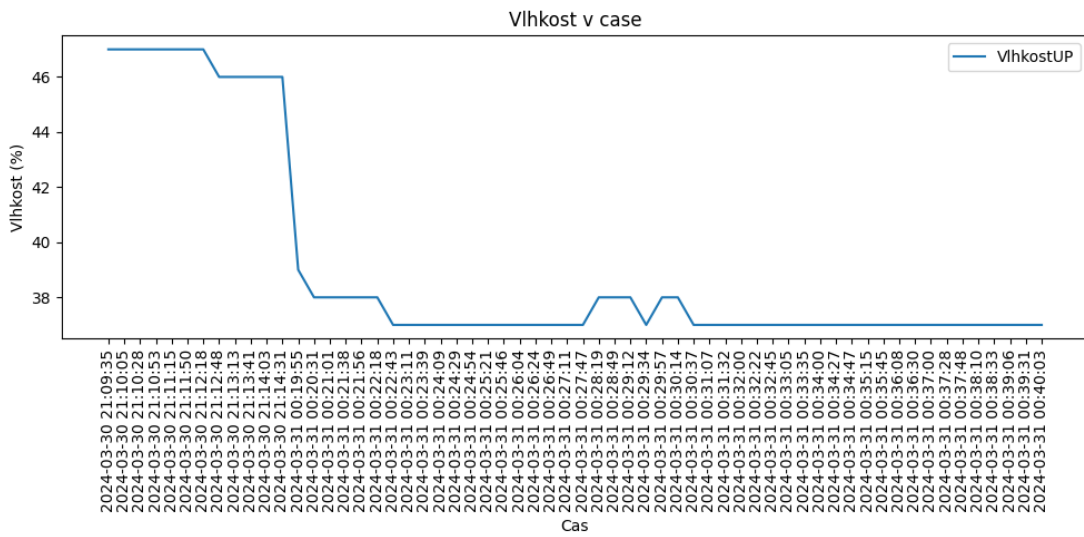
Graf 3 TeplotaDOWN v čase



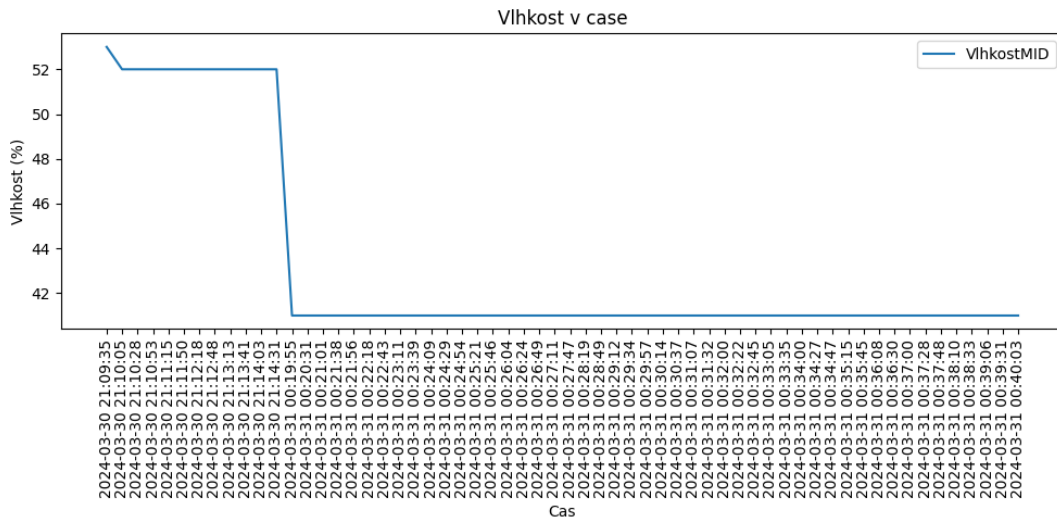
Graf 4 Indikace Kouře



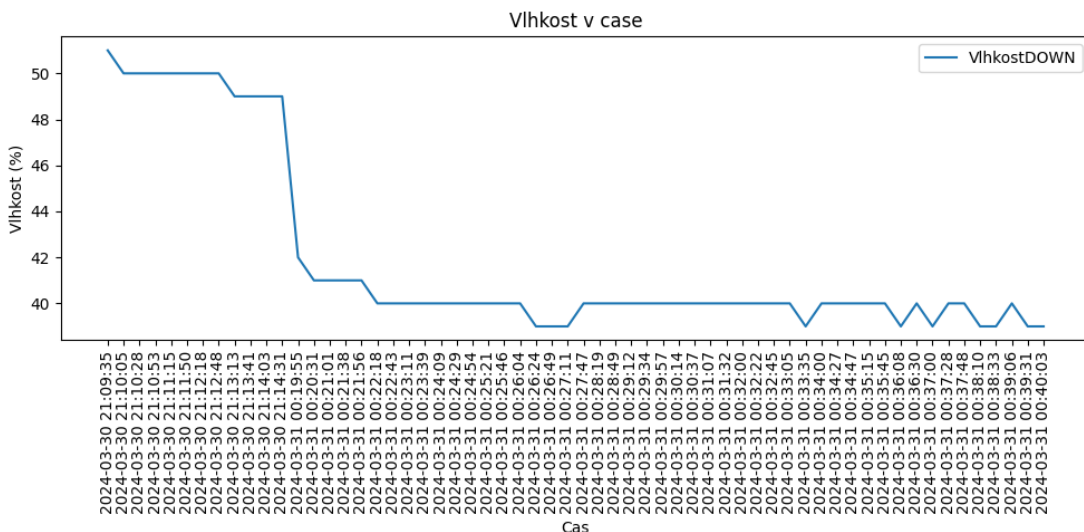
Graf 5 Indikace Vytopeni



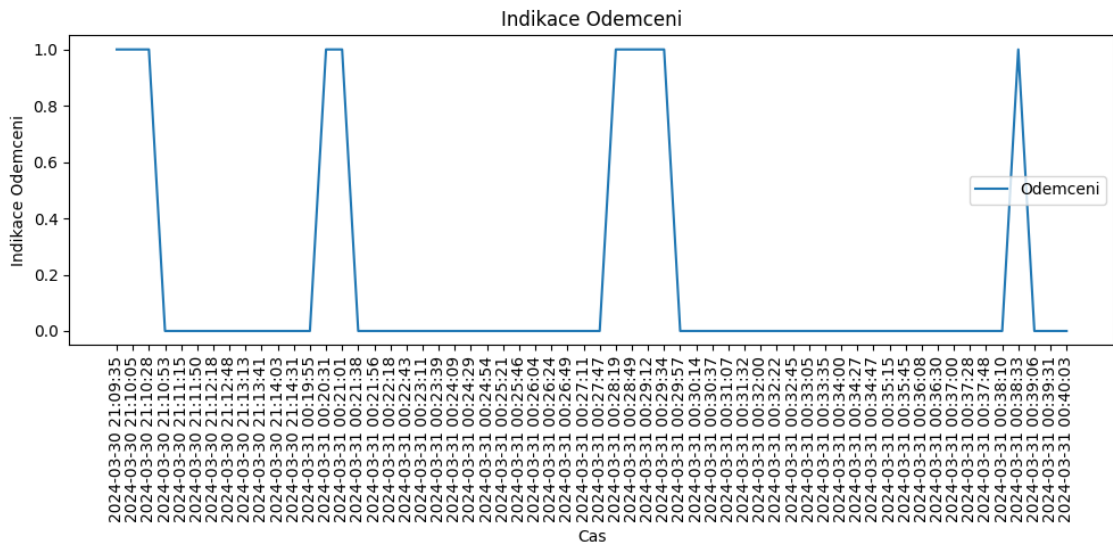
Graf 6 VlhkostUP v čase



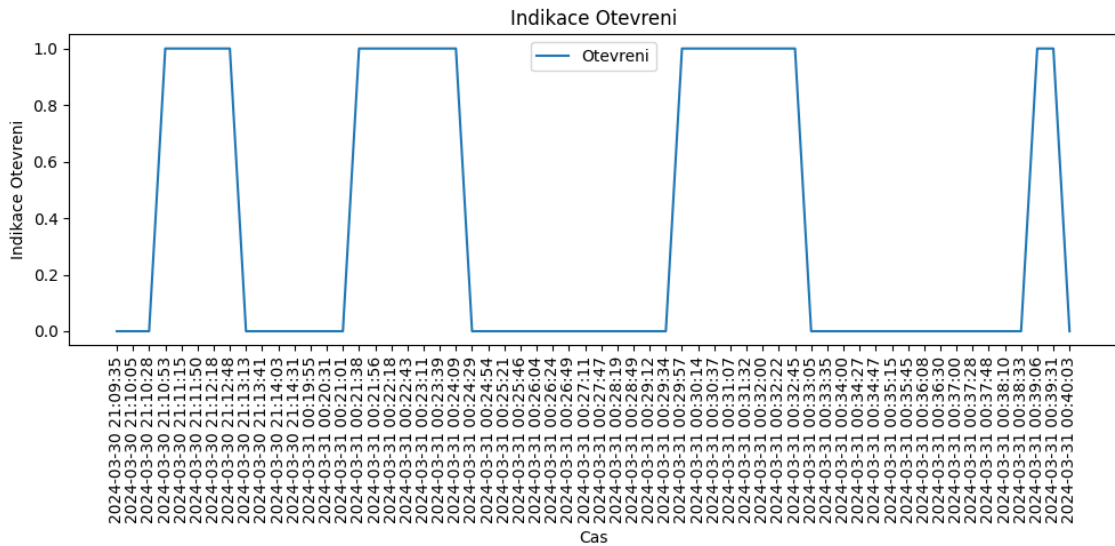
Graf 7 VlhkostMID v čase



Graf 8 VlhkostDOWN v čase



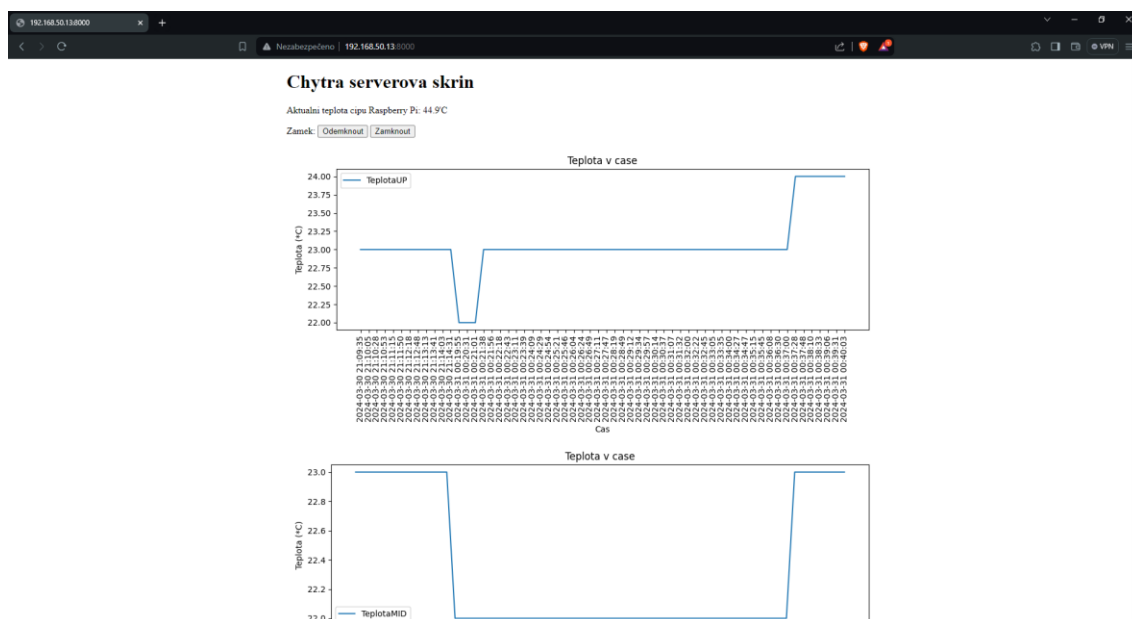
Graf 9 Indikace odemceni



Graf 10 Indikace Otevreni

5.2 Ukázka webového serveru

Cílem práce bylo vypracovat funkční řešení, bez ohledu na grafický design.



Obrázek 61 Ukázka webového serveru

5.3 Komplikace a zjištění

Bohužel se proces tvoření projektu neobešel bez komplikací.

Při spuštění programů přímo z vývojového prostředí Thonny docházelo ke kolizi práv, tento problém se dal však snadno obejít spuštěním programů z příkazové řádky.

Při testování bylo zjištěno, že není vhodné držet elektromagnetický zámek v poloze odemčeno příliš dlouho, z důvodu přehřívání elektromagnetické cívky uvnitř zámku, což může mít za následek její spálení a tím zničení celého zámku.

Dále bylo zjištěno že senzor primárně určený na měření vlhkostí půdy je vhodný i jako senzor vytopení po správném nastavení, jediným nedostatkem může být riziko falešných signálů způsobených vodivými nečistotami které se na nebo pod senzor umístěný na spodní části skříně mohou dostat. Tento problém by se pak dal řešit zapouzdřením senzoru.

Řešení však dává uživateli možnost snadno upravovat a přidávat další funkcionality a periferie. Je možné například přidat chlazení řízené dle měřených teplot a další různá vylepšení, která nebyla součástí projektu.

6 Závěr

Cílem této závěrečné práce bylo vytvoření zařízení určeného pro montáž do serverové skříně s cílem zajištění jejího stálého monitorování teploty, vlhkosti, přítomnosti kouře nebo známek vytopení a odesílat tato data do počítače. Dále také monitorovat stav dveří, který je následně indikován adresovatelným led páskem uvnitř skříně.

V teoretické části byly vysvětleny základní pojmy a byla provedena analýza existujících řešení. Dále byly popsány použité technologie, které byly při realizaci projektu použity a byl proveden výběr vhodných hardwarových komponentů.

V praktické části bylo navrženo řešení prototypu chytré serverové skříně včetně popisu instalace, elektrotechnického zapojení a samotného zprovoznění Raspberry Pi. Dále byly vypracovány skripty psané v jazyce Python, které byly podrobně rozepsány. První sloužící k měření a ukládání dat ze senzorů podle kterých je dále ovládána světelná signalizace a druhý, který je webový server spuštěný přímo na desce Raspberry Pi, jehož úkolem je zobrazovat aktuální stav teploty čipu, implementace funkce odemykání/zamykání serverové skříně pomocí tlačítka a zobrazování uložených dat ve formě grafů.

Celkově tedy praktická realizace projektu demonstrovala efektivní využití Raspberry Pi jakožto kompaktního, ale výkonného nástroje pro sběr dat, vizualizaci a ovládání v reálném čase. Navíc oproti s existujícími řešeními je vytvořené řešení mnohonásobně levnější a dává uživateli možnost snadno upravovat a přidávat další funkcionality dle specifických potřeb.

7 Seznam použitých zdrojů

1. Server Racks: Everything you need to know | Eaton. Eaton Website, [online] 2024 [cit. 2024-26-03]. Dostupné z: <https://tripplite.eaton.com/products/server-racks-explained>.
2. ŠTRÁFELDA, Jan. Co je server. Jan Štráfelda [online] 2024 [cit. 2024-26-03]. Dostupné z: <https://www.strafelda.cz/server>.
3. UPTON, Eben; HALFACREE, Gareth; GONER, Jakub. Raspberry Pi: uživatelská příručka. Brno: Computer Press, 2013. ISBN 978-80-251-4116-8.
4. Raspberry Pi. Buy a Raspberry Pi – Raspberry Pi [online] 2024 [cit. 2024-26-03]. Dostupné z: <https://www.raspberrypi.com/products/>.
5. Co je to Arduino?: Bastlírna HWKITCHEN, [online] 2017 [cit. 2024-26-03]. Dostupné z: <https://bastlirna.hwkitchen.cz/co-je-to-arduino/>.
6. Introducing Adafruit Feather. Adafruit Learning System, [online] 2017 [cit. 2024-26-03]. Dostupné z: <https://learn.adafruit.com/adafruit-feather/overview>.
7. Python, Real: Thonny: the Beginner-Friendly Python editor, [online] 2021 [cit. 2024-26-03]. Dostupné z: <https://realpython.com/python-thonny/>.
8. Team, Spyder: Home — Spyder IDE, Spyder Website Contributors, [online] 2023 [cit. 2024-26-03]. Dostupné z: <https://www.spyder-ide.org/>.
9. Raspberry Pi Pico a Arduino IDE. Jak na to? Chiptron [online] 2021 [cit. 2024-26-03]. Dostupné z: <https://chiptron.cz/news.php?readmore=1231>.
10. HOLLINGWORTH, Gordon: Introducing Raspberry Pi Imager, our new imaging utility, Raspberry Pi, [online] 2021 [cit. 2024-26-03]. Dostupné z: <https://www.raspberrypi.com/news/raspberry-pi-imager-imaging-utility/>.
11. What Is PICAXE? - What is PICAXE – PICAXE [online] 2017 Dostupné z: <https://picaxe.com/what-is-picaxe/>.
12. Arduino IDE – Bastlírna HWKITCHEN: Bastlírna HWKITCHEN, [online] 2017 [cit. 2024-26-03]. Dostupné z: <https://bastlirna.hwkitchen.cz/arduino-ide/>.
13. Wikipedia contributors: File: Arduino IDE - Blink.png - Wikipedia, [online] 2017 [cit. 2024-26-03] Dostupné z: https://en.m.wikipedia.org/wiki/File:Arduino_IDE_-_Blink.png.
14. SUMMERFIELD, Mark. Programming in Python 3: a complete introduction to the Python language. 2nd ed., Fully rev. ed. Upper Saddle River, NJ: Addison-Wesley, c2010. Developer's library. ISBN 978-0-321-68056-3.
15. Wikipedia contributors: Python (programming language), Wikipedia, [online] 2024 [cit. 2024-26-03] Dostupné z: [https://en.wikipedia.org/wiki/Python_\(programming_language\)](https://en.wikipedia.org/wiki/Python_(programming_language)).
16. Python, Real: Build physical projects with Python on the Raspberry Pi, [online] 2023 [cit. 2024-26-03] Dostupné z: <https://realpython.com/python-raspberry-pi/>.
17. Intelligent Server Rack Archives: Austin Hughes, [online] 2023 [cit. 2024-26-03] Dostupné z: <https://www.austin-hughes.com/solutions/server-racks/infrarack/>.
18. CANOVATE – Smart Server Rack: Pulsar trading, [online] 2023 [cit. 2024-26-03] Dostupné z: <https://pulsartrading.co.uk/urun/canovate-smart-server-rack/>.

19. Smart Rack Solutions: Raritan [online] 2022 [cit. 2024-26-03]. Dostupné z: <https://www.raritan.com/landing/smart-rack-solutions>.
20. Rosenberger Smart Server Rack; SystemRosenberger Asia Pacific – A leading manufacturer of telecom & high-speed interconnect solutions provider [online] 2020 [cit. 2024-26-03] Dostupné z: <https://www.rosenbergerap.com/newsDetail.html?id=80>.
21. Raspberry Pi. Buy a Raspberry Pi – Raspberry Pi [online] 2024
22. [cit. 2024-26-03]. Dostupné z: <https://www.raspberrypi.com/products/raspberry-pi-zero-2-w/>.
23. Jak postavit videopřehrávač za tři stovky? Experimentujeme s Raspberry Pi Zero W | Diit.cz (n.d.): [online] 2018 [cit. 2024-26-03]. Dostupné z: <https://diit.cz/clanek/jak-postavit-videoprehravac-za-tri-stovky-experimentujeme-s-raspberry-pi-zero-w>.
24. Rozvaděč Legrand EvoLine 19" stojanový rozvaděč 47U, 600x600mm, 1000 kg, jednokřídlé skleněné dveře, Alza.cz, [online] 2024 [cit. 2024-26-03]. Dostupné z: <https://www.alza.cz/legrand-evoline-19-stojanovy-rozvadec-47u-600x600mm-1000kg-jednokridle-sklenene-dvere-d5692530.htm>.
25. Teploměr a vlhkoměr DHT11 - RPishop.cz, RPishop.cz, [online] 2024 [cit. 2024-26-03]. Dostupné z: <https://rpishop.cz/teplota-vlhkost-a-tlak/1325-teplomer-a-vlhkomer.html>.
26. Modul senzoru plynu MQ-135 - RPishop.cz, RPishop.cz, [online] 2024 [cit. 2024-26-03]. Dostupné z: <https://rpishop.cz/plyny-a-prasnost/2445-modul-senzoru-plynu-mq-135.html>.
27. CzechProject Spol. S: Půdní Vlhkoměr Modul pro Arduino | dratek.cz, dratek.cz, [online] 2024 [cit. 2024-26-03]. Dostupné z: <https://dratek.cz/arduino/1399-eses-pudni-vlhkomer-pro-jednodeskove-pocitace.html>.
28. Adresovatelný LED pásek 30 LED/m, voděodolný – 5 m, černá - RPishop.cz, RPishop.cz, [online] 2024 [cit. 2024-26-03]. Dostupné z: <https://rpishop.cz/led/5402-adresovatelný-led-pasek-30-ledm-vodeodolny-5m-cerna.html>.
29. LaskaKit: Elektromagnetický zámek 12VDC, 2A | LaskaKit, laskakit.cz, [online] 2021 [cit. 2024-26-03]. Dostupné z: https://www.laskakit.cz/elektromagneticky-zamek-12vdc--2a/?gad_source=1.
30. CzechProject Spol. S: Magnetický kontakt MC-38 | dratek.cz, dratek.cz, [online] 2024 [cit. 2024-26-03]. Dostupné z: <https://dratek.cz/arduino/7700-jazyckovy-magneticky-kontakt.html>.
31. CzechProject Spol. S: 10cm DuPont Kably Barevné 40 žil samice samice | dratek.cz, dratek.cz, [online] 2024 [cit. 2024-26-03]. Dostupné z: https://dratek.cz/arduino/1561-10cm-dupont-f-f-kably-barevne-40-zil-samice-samice.html?gad_source=1.
32. CzechProject Spol. S: Napájecí adaptér síťový 5V 1000mA 5,5/2,1 mm | dratek.cz, dratek.cz, [online] 2024 [cit. 2024-26-03]. Dostupné z: <https://dratek.cz/arduino/1440-napajeci-adapter-sitovy-5v-1000ma-5-5-2-1-mm-vigan.html>.
33. RPishop.cz: Raspberry Pi 12,5W Micro USB 1,5m napájecí zdroj, EU, bílá - RPishop.cz, RPishop.cz, [online] 2024 [cit. 2024-26-03]. Dostupné z: <https://rpishop.cz/zdroje-s-microusb-kabelem/4313-raspberry-pi-125w-micro-usb-15m-napajeci-zdroj-eu-bila.html>.

34. Alza: Napájecí zdroj Virtuos 12V pro zákaznické displeje, Alza.cz, [online] 2024 [cit. 2024-26-03]. Dostupné z: <https://www.alza.cz/virtuos-12v-pro-zakaznicke-displeje-d4124829.htm>.
35. What is a web server? - Learn web development | MDN: MDN Web Docs, [online] 2023 [cit. 2024-26-03]. Dostupné z: https://developer.mozilla.org/en-US/docs/Learn/Common_questions/Web_mechanics/What_is_a_web_server.
36. OpenBSD manual pages: [online] 2017 [cit. 2024-26-03]. Dostupné z: <https://man.openbsd.org/ssh.1>.
37. freeCodeCamp.org: Controlling an External LED using a Raspberry Pi and GPIO pins, freeCodeCamp.org, [online] 2018 [cit. 2024-26-03]. Dostupné z: <https://www.freecodecamp.org/news/hello-gpio-blinking-led-using-raspberry-pi-zero-wh-65af81718c14/>.
38. HTML: HyperText Markup Language | MDN (2024): MDN Web Docs, [online] 2024 [cit. 2024-26-03]. Dostupné z: <https://developer.mozilla.org/en-US/docs/Web/HTML>.
39. PuTTY: a free SSH and Telnet client (n.d.): [online] 2024 [cit. 2024-26-03]. Dostupné z: <https://www.chiark.greenend.org.uk/~sgtatham/putty/>.

8 Seznam obrázků, tabulek a grafů

8.1 Seznam obrázků

Obrázek 1 Raspberry Pi Imager.....	18
Obrázek 2 Thonny IDE.....	19
Obrázek 3 Spyder3 IDE.....	20
Obrázek 4 Arduino IDE.....	21
Obrázek 5 Python logo.....	22
Obrázek 6 Austin Hughes Intelligent Server Infra Rack.....	25
Obrázek 7 Canovate's Smart Server Rack.....	26
Obrázek 8 Raritan Smart Rack.....	27
Obrázek 9 Rosenberger Smart Server Rack System 1.....	28
Obrázek 10 Rosenberger Smart Server Rack System 2.....	29
Obrázek 11 Rozměry Raspberry Pi Zero2 W.....	30
Obrázek 12 Popis GPIO pinů desky Raspberry Pi Zero 2 W.....	30
Obrázek 13 Serverová skříň Legrand EvoLine 19".....	31
Obrázek 14 Teplotní senzor DHT11.....	32
Obrázek 15 Senzor kouře MQ – 135.....	33
Obrázek 16 Senzor na měření vlhkosti.....	33
Obrázek 17 Adresovatelný led pásek.....	34
Obrázek 18 Elektromagnetický zámek.....	35
Obrázek 19 Magnetický senzor.....	35
Obrázek 20 Vodiče.....	36
Obrázek 21 Raspberry Pi Imager.....	37
Obrázek 22 Raspberry Pi Imager Edit settings.....	38
Obrázek 23 Raspberry Pi Imager nastavení WIFI.....	38
Obrázek 24 Raspberry Pi Imager nastavení SSH.....	39
Obrázek 25 Elektrotechnické schéma zapojení.....	40
Obrázek 26 sudo update.....	42
Obrázek 27 sudo upgrade.....	42
Obrázek 28 instalace balíčků.....	42
Obrázek 29 Import knihoven.....	43
Obrázek 30 Nastavení konfigurace led pásku.....	43
Obrázek 31 Číslování GPIO pinů.....	44
Obrázek 32 Nastavení GPIO pinů.....	45
Obrázek 33 Funkce na zbarvení led pásku.....	45
Obrázek 34 Přípravení csv souboru.....	45
Obrázek 35 Část main programu.....	46
Obrázek 36 Hlavní While cyklus.....	47
Obrázek 37 Čtení a kontrola teploty a vlhkosti.....	47
Obrázek 38 Kontrola odemčení.....	48
Obrázek 39 Kontrola otevření.....	48
Obrázek 40 Podmínka vytopení.....	48
Obrázek 41 Podmínka kouře.....	48
Obrázek 42 Zbarvení led pásku dle podmínek.....	49
Obrázek 43 Zápis do csv souboru.....	49
Obrázek 44 Instalace pandas.....	50

Obrázek 45 Instalace matplotlib	50
Obrázek 46 Import knihoven pro WS	50
Obrázek 47 ifconfig	51
Obrázek 48 Nastavení komunikace	51
Obrázek 49 Tvorba grafu TeplotyUP	52
Obrázek 50 Tvorba všech grafů z csv souboru	53
Obrázek 51 Funkce na čtení teploty čipu.....	54
Obrázek 52 Nastavení GPIO pinů pro zámek.....	54
Obrázek 53 Třída MyServer.....	54
Obrázek 54 Metoda do_GET	55
Obrázek 55 Metoda send_image.....	55
Obrázek 56 Metoda send_html	56
Obrázek 57 Metoda do_HEAD.....	56
Obrázek 58 Metoda _redirect	57
Obrázek 59 Metoda do_POST	57
Obrázek 60 Main program.....	58
Obrázek 61 Ukázka webového serveru	63

8.2 Seznam tabulek

Tabulka 1 Tabulka cen hardwaru.....	36
-------------------------------------	----

8.3 Seznam grafů

Graf 1 TeplotaUP v čase	59
Graf 2 TeplotaMID v čase	59
Graf 3 TeplotaDOWN v čase.....	60
Graf 4 Indikace Kouře	60
Graf 5 Indikace Vytopení	60
Graf 6 VlhkostUP v čase	61
Graf 7 VlhkostMID v čase.....	61
Graf 8 VlhkostDOWN v čase	61
Graf 9 Indikace odemčení.....	62
Graf 10 Indikace Otevření	62

9 Přílohy

Příloha 1: CD se zdrojovými kódy