



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**

FACULTY OF INFORMATION TECHNOLOGY

**ÚSTAV INFORMAČNÍCH SYSTÉMŮ**

DEPARTMENT OF INFORMATION SYSTEMS

**DASHBOARD PRO PŘEHLED STAVU CHYTRÉ  
DOMÁCNOSTI ZAMĚŘENÝ NA ZAŘÍZENÍ  
S NÍZKOU SPOTŘEBOU**

DASHBOARD FOR SMART HOME STATUS FOCUSED ON LOW CONSUMPTION  
INSTALLATIONS

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

**VEDOUCÍ PRÁCE**

SUPERVISOR

**TOMÁŠ FIŠER**

**Ing. PETR JOHN**

BRNO 2024

## Zadání bakalářské práce



154664

Ústav: Ústav informačních systémů (UIFS)  
Student: **Fišer Tomáš**  
Program: Informační technologie  
Název: **Dashboard pro přehled stavu chytré domácnosti zaměřený na zařízení s nízkou spotřebou**  
Kategorie: Vestavěné systémy  
Akademický rok: 2023/24

### Zadání:

1. Prostudujte oblast internetu věcí (Internet of Things, IoT) a koncept chytrých měst (Smart City), často používané mikrokontrolery a mikropočítače (Arduino, Raspberry Pi).
2. Prostudujte principy tvorby přehledových obrazovek typu dashboard a existující typy vizualizací vhodných pro jejich návrh a problematiku různých způsobů zobrazení informací uživateli (zaměřte se na způsoby s nízkou spotřebou – např. e-ink).
3. Prostudujte způsoby a existující knihovny pro vykreslování vizualizací na různých typech displejů. Zaměřte se na způsoby zobrazení aplikovatelné na typické zařízení chytré domácnosti (např. čidla teploty, vlhkosti, kvality vzduchu).
4. Proveďte analýzu požadavků na vizualizaci dat získaných ze senzorů chytrých zařízení. Zaměřte se na problematiku vizualizaci jejich stavu s ohledem na spotřebu. Dále proveďte analýzu existujícího API firmy Logimic.
5. Navrhněte architekturu řešení, způsob zapojení koncového zařízení a způsob vizualizace stavu zařízení.
6. Po konzultaci s vedoucím navržené řešení implementujte.
7. Výsledné řešení otestujte z hlediska použitelnosti, porovnejte spotřebu oproti použití tradičního displeje.

### Literatura:

- Greengard, S.: The Internet of Things. MIT Press, 2015, ISBN 978-026-2527-736.
- Few, S.: Information Dashboard Design: The Effective Visual Communication of Data. Sebastopol, USA: O'Reilly, 2006, ISBN 978-059-6100-162.
- Interní dokumentace firmy Logimic

Při obhajobě semestrální části projektu je požadováno:  
Body 1 až 5.

Podrobné závazné pokyny pro vypracování práce viz <https://www.fit.vut.cz/study/theses/>

Vedoucí práce: **John Petr, Ing.**  
Vedoucí ústavu: Kolář Dušan, doc. Dr. Ing.  
Datum zadání: 1.11.2023  
Termín pro odevzdání: 9.5.2024  
Datum schválení: 30.10.2023

## Abstrakt

V rychle rozvíjejícím se konceptu IoT, neboli síti tvořené fyzickými zařízeními s elektronickou a softwarovou vybaveností pro vzájemné propojení a spolupráci pro vykonávání autonomních akcí bez zásahu člověka, jsou pro uživatele důležitým prvkem přehledové obrazovky neboli dashboardy, které poskytují vizualizace dat a ovládací prostředky nad daty. Tato práce se zabývá tvorbou dashboardu, který má za hlavní cíle nízkou energetickou náročnost oproti tradičním řešením, jednoduché ovládací rozhraní a přehlednou vizualizaci dat. Parametr nízké spotřeby se snaží splnit užitím energeticky efektivního výpočetního hardwaru, mikrokontroler ESP32 a užitím E-ink zobrazovacího panelu, který opět disponuje nízkou energetickou náročností. Výsledný dashboard by měl být schopen běhu z baterií po dlouhou časovou periodu bez nutnosti výměny nebo dobití baterií.

## Abstract

In the rapidly evolving concept of IoT, or a network consisting of physical devices with electronic and software capabilities to interconnect and collaborate to perform autonomous actions without human intervention, dashboards are an important element for users, providing data visualizations and controls over data. This work deals with the creation of a dashboard that has as main goals low energy consumption compared to traditional solutions, simple control interface and clear data visualization. It tries to meet the parameter of low power consumption by using energy efficient computing hardware, ESP32 microcontroller and by using E-ink display panel which again has low power consumption. The resulting dashboard should be able to run on batteries for long periods of time without the need to replace or recharge the batteries.

## Klíčová slova

IoT, Smart City, mikrokontroler, mikropočítač, API, E-ink, Arduino, ESP32, Wi-Fi, vizualizace, přehledová obrazovka

## Keywords

IoT, Smart City, microcontroller, microcomputer, API, E-ink, Arduino, ESP32, Wi-Fi, visualization, dashboard

## Citace

FIŠER, Tomáš. *Dashboard pro přehled stavu chytré domácnosti zaměřený na zařízení s nízkou spotřebou*. Brno, 2024. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Petr John

# Dashboard pro přehled stavu chytré domácnosti zaměřený na zařízení s nízkou spotřebou

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Petra Johna. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....

Tomáš Fišer  
4. května 2024

## Poděkování

Rád bych poděkoval svému vedoucímu Ing. Petru Johnovi za jeho odborné vedení a trpělivost během celého procesu psaní této bakalářské práce. Jeho vedení, podněty a konstruktivní zpětná vazba mi byly velkou oporou při mé práci. Díky jeho vstřícnému přístupu jsem nejen rozšířil své znalosti, ale také získal cenné dovednosti. Oceňuji jeho ochotu naslouchat a věnovat mi svůj čas, abych mohl dosáhnout svých cílů.



# Obsah

<b>1</b>	<b>Úvod</b>	<b>3</b>
<b>2</b>	<b>Internet věcí a chytrá města</b>	<b>5</b>
2.1	Chytrá města . . . . .	6
2.2	Mikrokontrolery a mikropočítače . . . . .	8
<b>3</b>	<b>Typy dashboard vizualizací</b>	<b>11</b>
3.1	Existující typy vizualizací pro dashboardy . . . . .	12
3.2	Problematika různých způsobů zobrazení informací . . . . .	15
<b>4</b>	<b>Způsoby a existující knihovny pro vykreslování vizualizací</b>	<b>18</b>
4.1	Zobrazovací způsoby aplikovatelné v chytrých domácnostech . . . . .	20
<b>5</b>	<b>Analýza požadavků a současného stavu</b>	<b>22</b>
5.1	Analýza firmy Logimic . . . . .	23
5.2	Současný stav nabízený firmou Logimic . . . . .	25
<b>6</b>	<b>Návrh architektury řešení</b>	<b>27</b>
6.1	Návrhy vizualizací . . . . .	29
<b>7</b>	<b>Implementace architektury řešení</b>	<b>31</b>
7.1	Hardwarová architektura implementace . . . . .	31
7.2	Programová architektura implementace . . . . .	33
7.3	Implementované vizualizace . . . . .	36
<b>8</b>	<b>Testování, spotřeba a použitelnost</b>	<b>40</b>
8.1	Testování použitelnosti . . . . .	40
8.2	Testování spotřeby energie . . . . .	41
<b>9</b>	<b>Závěr</b>	<b>43</b>
	<b>Literatura</b>	<b>44</b>
<b>A</b>	<b>Obsah přiloženého paměťového média</b>	<b>50</b>

# Seznam obrázků

3.1	Sloupcový graf zobrazující průměrné teploty . . . . .	13
3.2	Řádkový graf zobrazující průměrné teploty . . . . .	14
3.3	Rozptylový graf zobrazující teploty . . . . .	15
5.1	Architektura komunikace . . . . .	24
5.2	Postup získání hodnoty parametru v platformě ACADA . . . . .	26
6.1	Návrh vývojového diagramu architektury . . . . .	29
6.2	Příklady vizualizací . . . . .	30
6.3	Další příklady vizualizací . . . . .	30
7.1	Zadní strana hardwarového řešení . . . . .	32
7.2	Schéma zapojení - zadní strana . . . . .	32
7.3	Zapojení mikrokontroleru . . . . .	33
7.4	Menu pro výběr parametru . . . . .	34
7.5	Webový konfigurátor . . . . .	35
7.6	Vývojový diagram implementované architektury . . . . .	36
7.7	Implementovaná informační vizualizace . . . . .	37
7.8	Implementovaný řádkový graf . . . . .	38
7.9	Implementovaný rozptylový graf . . . . .	38
7.10	Implementovaná tabulka . . . . .	39

# Kapitola 1

## Úvod

Internet věcí (IoT) je síť fyzických zařízení vybavených elektronikou, softwarem, senzory a sítovou konektivitou, která umožňuje těmto zařízením vzájemné propojení, výměnu dat a spolupráci. Zjednodušeně lze internet věcí popsat jako síť chytrých zařízení, která jsou schopna sbírat, odesílat a zpracovávat data o sledovaných veličinách a zároveň na jejich základě provádět určité autonomní akce bez zásahu člověka. Využití IoT se nachází v průmyslu pro monitorování výrobních procesů, preventivní údržbě nebo bezpečnosti zaměstnanců, dále ve zdravotnictví, dopravě, službách a domácnostech. Tento koncept dal možnost vzniku několika dalším navázaným konceptům, mezi které patří chytrá města přinášející lepší podmínky pro život svým obyvatelům, Průmysl 4.0 nahrazující jednoduché opakované lidské činnosti prací strojů nebo ekosystémy chytrých domácností zlepšující pohodlí, bezpečnost, energetickou efektivitu, zdraví a pohodu členům domácnosti.

Aby koncept IoT mohl fungovat, je nutné vytvořit dostatečné množství malých, výkonných, energeticky úsporných, snadno programovatelných a cenově dostupných zařízení schopných vykonávat daný úkol v rámci IoT. Tuto pozici zaujaly mikropočítače a mikrokontrolery, bez kterých by koncept IoT bylo složité realizovat. Mikropočítače zastupují výkonnějšího zástupce než mikrokontrolery, s tím však přicházejí vyšší pořizovací náklady, rozměry a vyšší energetická náročnost, proto nacházejí využití hlavně v chytrých zařízeních pro výpočetně komplexnější aplikace v průmyslu nebo dopravě, mikrokontrolery naopak svými nízkými pořizovacími náklady, malými rozměry a nízkou energetickou náročností nacházejí uplatnění v běžnějších aplikacích, například chytré domácnosti nebo zdravotnictví. Mezi hlavní zástupce mikropočítačů patří mikropočítače z rodiny Raspberry Pi, ze skupiny mikrokontrolerů se jedná převážně o platformu Arduino.

Data získaná ze zařízení je nutné uživateli vhodně vizualizovat. Při návrhu dané vizualizace je nutné dbát na několik klíčových aspektů, zejména srozumitelnost, výkonnost a přehlednost, také je nutné brát zřetel na fyzické parametry a schopnosti zařízení, které bude danou vizualizaci zobrazovat a také na podmínky, ve kterých bude vizualizovaná.

Tato práce si bere za cíl vytvoření zobrazovacího zařízení poskytující uživateli přehled o klíčových informacích v rámci jednoho grafického rozhraní neboli dashboard, zaměřené na užití v rámci chytré domácnosti. Jeho hlavním rozdílem oproti tradičním řešením bude velmi nízká energetická náročnost optimalizovaná pro dlouhodobé napájení z baterií díky užití mikrokontroleru ve spojení se zobrazovacím panelem typu E-ink, poskytující nejen výbornou energetickou efektivitu, ale zároveň skvělé kontrastní vlastnosti a čitelnost v oblastech s intenzivním zdrojem světla. Dashboard bude také disponovat bezdrátovým připojením pro stahování dat z cloudového prostředí firmy Logimic a webovým konfiguratorem

zprostředkovávající jednoduché a přehledné uživatelské rozhraní pro nastavení parametrů dashboardu.

Práce je dělená do několika kapitol, první obsahová kapitola 2 pojednává o problematice konceptu internetu věcí, chytrých měst a rozdílech mezi mikrokontrolery a mikropočítači. Následující kapitola 3 navazuje na předchozí a zabývá se problematikou vizualizací, přehledovými obrazovkami (dashboardsy) a technologiemi zobrazovacích panelů využívajících se při tvorbě přehledových obrazovek. V kapitole 4 práce hlouběji prozkoumává oblast knihoven pro zjednodušení tvorby vizualizací a zobrazovací způsoby aplikovatelné v rámci konceptu chytré domácnosti. Kapitola 5 zkoumá data získávaná ze senzorů, dále pojednává o firmě Logimic, aplikačním programovém rozhraní firmy, analýze požadavků a současného stavu. Na základě analýzy z předchozí kapitoly je v kapitole 6 vytvořen návrh možné architektury hardwarového, softwarového i programového řešení společně s návrhy vizualizací. V předposlední kapitole 7 je na základě návrhu provedena implementace hardwarového i softwarového řešení společně s implementací navržených vizualizací. V poslední kapitole 8 je provedeno testování implementovaného řešení, zkoumání spotřeby elektrické energie a stability implementace společně s použitelností implementace v praxi.

## Kapitola 2

# Internet věcí a chytrá města

Internet věcí neboli Internet of Things, dále jen IoT, je konceptem označujícím propojení fyzických zařízení a objektů v rámci globální sítě umožňující sběr, výměnu a analýzu dat. Tyto zařízení jsou vždy vybaveny softwarem a rozhraním umožňující komunikaci po internetu, dále také velkou škálou senzorů pro sběr dat a informací. V dnešním světě, který je stále více urbanizovaný<sup>1</sup> a zároveň technicky vyspělí, nachází tento koncept uplatnění v mnoha odvětvích, včetně průmyslu, zdravotnictví, energetiky a také v městském plánování [11, 51].

Historie IoT se začíná psát v první polovině 20. století, kdy se začaly objevovat první nápady spojené hlavně s rozvojem prvních počítačů, nicméně se jednalo hlavně o teorii, která díky dostupným technologiím té doby implementaci zatím neumožňovala. Významným milníkem bylo období 50. a 60. let, kdy se začaly objevovat v průmyslu první senzory založené na radiofrekvenční identifikaci (RFID) [19]. V 70. letech 20. století se začínají objevovat první experimentální projekty fyzických zařízení komunikujících po počítačové síti, nicméně vznik samotného termínu Internet of Things neboli IoT, vzniká až v roce 1999, kdy byl definován britským vizionářem Kevinem Ashtonem, což výrazně přispělo k popularitě tohoto konceptu a rozvoji až do dnešní podoby.

Funkcí, které IoT plní je několik, za nejzákladnější, na které staví ostatní, můžeme označit sběr dat. Zařízení připojená k IoT jsou vybavena různými senzory, například teploměry, senzory pohybu, vlhkosti, kamery a mnoho dalších. Pomocí těchto senzorů sbírá systém z okolního prostředí data. Příkladem mohou být třeba senzory vibrací<sup>2</sup> na výrobní lince, pomocí kterých lze poznat, zda stroj pracuje správně a nevznikají při jeho práci znaky mechanického problému, ze senzorů pro běžného uživatele se může například jednat o chytrý termostat od firmy Netatmo<sup>3</sup> nebo chytrou bezpečnostní kameru Mi 360° od firmy Xiaomi.<sup>4</sup>

Získaná data je třeba odeslat ke zpracování, tedy další funkcí je IoT je komunikace. Získaná data jsou prostřednictvím internetu nebo jiného druhu komunikační sítě přenášena do dalších zařízení, které je buď zpracují nebo odešlou dále. Technologie pro přenos je libovolná, musí však splnit požadavky na propustnosti sítě a také na dosah [21]. Dnes nejčastěji používanější drátovou technologií pro přenos je Ethernet, mezi bezdrátovými technologiemi se jedná hlavně o technologii Wi-Fi [61], která je vysoce rozšířená a nabízí vysokou rychlost přenosu, dále Bluetooth, který však na rozdíl od Wi-Fi slouží pouze pro komunikaci na krátké vzdálenosti, tedy nejčastěji pro zařízení pracující v rámci PAN (Personal Area

<sup>1</sup>Urbanizace v dnešním světě: <https://ourworldindata.org/urbanization>

<sup>2</sup>Analýzou a diagnostikou vibrací se zabývá česká firma SKF: <https://www.skf.com/cz/services/condition-based-maintenance/vibration-analysis-and-diagnostics>

<sup>3</sup>Webové stránky produktu: <https://www.netatmo.com/cs-cz/smart-thermostat>

<sup>4</sup>Webové stránky produktu: <https://www.mi.com/global/camera-360/>

Network), jedná se tedy převážně o osobní elektroniku (chytré hodinky, chytré boty a další). Existuje i několik dalších běžně užívaných přenosových technologií v IoT, které však nejsou tak obecně známé, mezi ně se řadí Zigbee [3] (energeticky efektivní bezdrátová síť s malým výkonem pro malá zařízení v chytré domácnosti), Z-Wave [3] (velmi podobná technologie se Zigbee s rozdílem, že využívá proprietární standart, takže existuje jen omezené množství výrobců s možností výroby Z-Wave zařízení) nebo technologie Sigfox [3] disponující globální komunikační sítí speciálně navrženou pro IoT, jejími hlavními parametry je dlouhý dosah, nízká spotřeba energie, která je však vykoupena nízkou přenosovou rychlostí v řádu desítek bajtů až po jednotky kilobajtů za sekundu. Některé z technologií, například technologie Wi-Fi, umožňuje zprávu a ovládání v reálném čase, můžeme si tedy například v reálném čase nastavovat pomocí mobilního zařízení teplotu v naší chytré domácnosti i bez naší fyzické přítomnosti v ní [58].

Data, která získáváme a odesíláme, je následně třeba zpracovat a analyzovat. Postupy analýzy se liší v závislosti na technickém řešení, nicméně se může jednat o jednoduché filtry nebo pravidla, ale také může jít o komplexní algoritmy, například strojového učení. Cílem této analýzy je nalézt určité vzory v datech, ze kterých můžeme například odvodit nestandardní chování daného zařízení [2].

Na základě výsledků poskytnutých analýzou je třeba provést příslušné akce a reakce. To může zahrnovat například spuštění konkrétní obslužné rutiny na základě nastavených pravidel a algoritmů. Pro demonstraci využijeme situaci uvedenou výše, kdy máme senzor snímající vibrace stroje. Při analýze dat je zjištěna nadměrná úroveň vibrací značící nestandardní chování a na základě pravidel je vybrána a aplikována reakce na dané chování, například zahájení údržby stroje.

Poslední z hlavních funkcí IoT je ukládání dat, která jsou nezbytná pro jejich dlouhodobou dostupnost a v případě potřeby i zpětnou analýzu. Data mohou být uchována vzdáleně, datová centra, cloudové služby, ale i lokálně. Důležitým faktorem při ukládání dat je vysoká míra zabezpečení, protože mohou obsahovat soukromé nebo citlivé informace.

Koncept IoT nám tedy přináší řadu benefitů, zvýšení automatizace, vzdálená správa, efektivnější využití zdrojů a další. Princip celého konceptu bez ohledu znalosti konkrétní realizace je založen na sběru dat, jejich odeslání na příslušné zařízení, kde jsou data zpracovávána a analyzována, na základě analýzy jsou provedeny konkrétní akce a reakce, závěrem jsou data uložena pro pozdější využití. Problematiku posledních tří odstavců řeší autor v publikaci [48].

## 2.1 Chytrá města

Chytrá města, také známá jako smart cities, je koncept komplexního a transformačního přístupu k urbanizaci a rozvoji měst zaměřený hlavně na zlepšení kvality života obyvatel, do kterého patří zlepšení infrastruktury, životního prostředí, služeb, ale i bezpečnosti [39]. Klíčovými funkcemi je sběr a analýza informací, která využívají pro efektivní řízení a rozhodování v reálném čase. Velký rozvoj tohoto konceptu začal počátkem tohoto tisíciletí v důsledku velkého rozvoje konceptu IoT [40].

Celý koncept chytrých měst je založen na několika důležitých technologiích, ale také na infrastruktuře, na kterou by se nemělo zapomínat. Základním kamenem téměř celého konceptu chytrých měst je koncept IoT, výjimkou je oblast smart governance, která je primárně založena na plošné digitalizaci a datové otevřenosti, i když prvky konceptu IoT v ní nalézt můžeme také. Tento koncept umožňuje společnou komunikaci velkého množství zařízení a sdílení dat prostřednictvím sítí příslušné technologie. Rozmanitost zařízení spadajících



do tohoto konceptu nám umožňuje monitorovat a řídit velké množství aspektů městského života, příkladem může být chytrý semafor<sup>5</sup>. Na základě času a dopravní situace může sám upravovat časové intervaly pro ideální plynulost provozu v danou chvíli.

Získaná data, tzv. Big Data [4], ze senzorů je třeba také pokročile analyzovat. Jejich analýzou získají města hlubší porozumění městských vzorců a potřeb, které se dají využít například ke zlepšení plánování oprav silnic, kdy na základě dat o provozu může být oprava naplánována nejen na ideální čas, ale i roční období, aby byl dopad opravy na okolí co nejmenší. Nejrozšířenější technologií pro zpracování Big Data je open-source framework pro distribuované zpracování velkého množství dat Hadoop [6]. Jeho hlavní funkcionalitou je využití modelu MapReduce, který umožňuje paralelní zpracování dat na clusteru serverů, čímž zefektivňuje analýzu velkých datových souborů. Dalšími příklady technologií mohou být například distribuovaný framework Apache Spark, vynikající svou rychlostí a jednoduchostí použití nebo SQL databáze a také frameworky založené na strojovém učení. Data jsou následně ukládána na vzdálených serverech umožňující centralizovaný přístup k datům a službám. Tento přístup přináší jednoduché sdílení informací, spolupráci a komunikaci v rámci institucí [4].

Možnost aplikace tohoto konceptu je v městech nezměrná [5], od dopravy, jak už bylo uvedeno například v podobě chytrých semaforů, dalšími oblastmi je energetika, tedy umožnění lepší výroby a distribuce elektrické energie s vyšší účinností a kladením důrazu na využití obnovitelných zdrojů výroby. Energetické systémy chytrých budov umožňují efektivní snížení spotřeby energie a tím i snížení provozních nákladů svým obyvatelům. Koncept také přináší zlepšení bezpečnosti ve městech, kamerové systémy a senzory pro monitorování veřejných prostorů, ty přináší rychlé řešení mimořádných událostí, ale i díky automatické analýze obrazu možnosti předcházení vzniku nestandardních událostí. S využitím se také lze setkat ve zdravotnictví, elektronické zdravotní záznamy a telemedicina (dálkový přenos lékařských informací<sup>6</sup>) zrychlují a zjednodušují přístup ke zdravotnickým službám, dále ve vzdělávání (studium na dálku, elektronické učebnice, výukové aplikace), ochraně životního prostředí (monitorování kvality ovzduší, podpora ekologických aktivity), správě odpadu (optimalizace sběru a recyklace, senzory zaplnění kontejnerů umožňující efektivní plánování svozů) a v neposlední řadě jednoduché elektronické zapojení samotných občanů do rozhodování o důležitých městských záležitostech ovlivňující chod města.

Výhod má koncept chytrých měst mnoho, nicméně má i svá omezení a výzvy, kterým je při realizaci třeba čelit. Největší otázkou je ochrana osobních údajů [20] a soukromí samotných obyvatel měst. Dochází k velkému sběru často i osobních dat, které by mohli být proti obyvatelům v případě jejich úniku zneužity, například hackerské útoky. Nezanedbatelným problémem je také vysoká technologická a infrastrukturní náročnost konceptu, kdy města nejsou často na takový razantní přechod připravena a úpravy jsou velmi finančně náročné. Na toto se také váže digitální propast, některých městských oblastí oproti jiným, které jsou často technologicky vyspělejší, mělo by se tedy také pamatovat, aby byla digitalizace spravedlivá a zahrnovala všechny obyvatele měst stejně.

Ve světě se již s několika chytrými městy můžeme setkat, za nejvyspělejší chytré město je považován městský stát Singapur. Koncept využívá téměř ve všech oblastech, projekt nazývá Smart Nation a zahrnuje prostředky pro zlepšení dopravy (chytré semaforey), bezpeč-

---

<sup>5</sup>Článek popisující technologii chytrých semaforů: [https://www.lta.gov.sg/content/ltagov/en/getting\\_around/driving\\_in\\_singapore/intelligent\\_transport\\_systems/green\\_link\\_determining\\_system.html](https://www.lta.gov.sg/content/ltagov/en/getting_around/driving_in_singapore/intelligent_transport_systems/green_link_determining_system.html)

<sup>6</sup>Definice významu: <https://euc.cz/clanky-a-novinky/clanky/kam-kraci-ceska-telemedicina-lekarske-vysetreni-na-dalku-chrani-pacienty-pred-nebezpecnou-samodiagnostikou/>

nosti (digitální identifikace občanů), výzkumu, životního prostředí (senzory kvality ovzduší), zdravotnictví, plánování, stavba chytrých domů a další odvětví [42]. Ve městech Barcelona, Kodaň, Tokio a New York City se může taky setkat s různými stupni aplikace konceptu chytrého města, nejčastěji se jedná aplikaci v oblastech dopravy a bezpečnosti [39].

## 2.2 Mikrokontrolery a mikropočítače

Mikrokontrolery a mikropočítače, tyto dva pojmy jsou často označovány za totožné, nicméně i přes to, že jsou oba termíny zaměňovány, popisují dvě rozdílné kategorie elektronických zařízení s různými vlastnostmi, schopnostmi a hlavně aplikací, ve kterých může být často aplikována pouze jedna z těchto kategorií [26].

Mikrokontrolery jsou zpravidla malé integrované obvody obsahující procesor, malou operační RAM paměť, flash paměť a vstupně-výstupní periferie. Mezi hlavní znaky mikrokontrolerů patří vysoká míra integrace, kdy jsou všechny důležité části integrovány do jednoho čipu. S tímto je spojený důraz na energetickou efektivitu. Často jsou využity v bateriových zařízeních, která musí na jedno nabití mít určitou životnost s ohledem na požadavky. Dalším znakem je schopnost real-time zpracování výpočetně jednodušších úkolů s důrazem na rychlou reakci na událost, mezi které se řadí interakce s periferiemi, jako jsou tlačítka, senzory, LCD displeje, LED osvětlení, krokové motorčky a mnohé další, proto nacházejí často využití ve vestavěných systémech, počítačový systém navržený pro vykonávání konkrétního úkolu nebo funkce v rámci určitého zařízení, stroje nebo systému. Zde jsou implementovány za účelem využití pro konkrétní úkol s ohledem na jejich omezené výpočetní možnosti [26, 29].

Mikropočítače jsou naopak samostatné počítačové systémy s vyšším výpočetním výkonem. S tím přichází větší rozměry a menší míra integrace, kdy všechny důležité části nejsou integrovány do jednoho čipu. Jako datové úložiště se již nevyužívá malá integrovaná flash paměť, ale může být užito SSD nebo HDD disků. Mikropočítače jsou schopny běhu plnohodnotného operačního systému, což umožňuje použití různých aplikací a programů. Jejich využití není tedy tak jednoúčelové, proto se s nimi lze setkat i v oblastech realizace herních konzolí, stanic pro softwarový vývoj, výpočetní technice, ale i v roli plnohodnotného osobního počítače [26, 66].

Z vlastností mikrokontrolerů a mikropočítačů je patrné, že jejich oblast využití se tedy výrazně liší, mikrokontrolery jsou vhodné hlavně pro jednoduché jednoúčelové výpočetní úkony s důrazem na malé rozměry, energetickou efektivitu a nízkou cenu, naopak mikropočítače nacházejí uplatnění v případech nutnosti vysokého výkonu a potřeby plnohodnotného a samostatně pracujícího zařízení.

Příkladů mikrokontrolerů lze na trhu nalézt několik desítek, kdy výběr závisí na konkrétní implementaci. Mezi nejběžnější zástupce patří ATmega328, mikrokontroler od společnosti Microchip, dřívě Atmel, široce využívaný v platformách Arduino. Disponuje jedním jádrem AVR architektury (RISC instrukční sada a využití Harvardské architektury umožňuje snadné programování a zvyšuje výkon a efektivitu) s možností programování v jazyce C nebo assembleru s množstvím dostupných knihoven pro usnadnění vývoje. Je tedy vhodný jak pro začátečníky, tak i pro pokročilé vývojáře s vysokou škálou užití, od jednoduchých elektronických hraček až po sofistikovanější robotiku [31]. ESP8266 je další ze skupiny nejběžněji užívaných mikrokontrolerů. Vyvinut byl společností Espressif Systems. Jeho klíčovou vlastností je vestavěný Wi-Fi čip s možností komunikace přes bezdrátovou síť, tato skutečnost je ideální pro využití v projektech z oblasti IoT, kde je bezdrátová komunikace



přes internet jednou z klíčových vlastností. Díky své hardwarové vybavenosti a nízké ceně je často užíván i pro hobby projekty [36].

Výše zmínění zástupci mikrokontrolerů se vyznačovali svou jednoduchostí, nižším výkonem, energetickou efektivitou a nízkou cenou se zaměřením na jednoduché projekty i z oblasti hobby. Pro složitější a průmyslové implementace je tedy nutné zvolit jiného zástupce, příkladem může být STM32F4. Mikrokontroler společnosti STMicroelectronics patří do širší řady STM32. Klíčovými vlastnostmi je užití výkonného jádra ARM Cortex-M4 s obsahem instrukcí pro digitální signálový prostor (DSP) a možností práce s pohyblivou řádovou čárkou (FPU). Tyto vlastnosti mu otevírají dveře do implementací založených na zpracování signálů a výpočetně náročných úloh. Dalšími benefity toho mikrokontroleru je široká škála možností periferních zařízení včetně komunikačních rozhraní SPI, I2C, UART, A/D převodníků, časovačů a mnoho dalšího. STM32F4 je dostupný v několika variantách lišících se v paměťových kapacitách, rychlostech procesoru, spotřebě energie a možnostech periferních zařízení, tak aby si vývojář mohl zvolit ideální zařízení pro své potřeby [7].

Mezi zástupci z řad mikropočítačů je opět z čeho vybírat. K nejběžnějším patří open-source platforma Arduino. Díky svým nízkým pořizovacím nákladům našla oblibu nejen mezi profesionály, ale i mezi amatérskou veřejností a studenty. Existuje několik verzí toho mikropočítače. Nejvýkonnější z této rodiny je Arduino Due postavené na mikrokontroleru ARM Cortex-M3 o frekvenci 84 MHz, dále Arduino Zero, Arduino Mega a nejméně výkonné z této rodiny, ale zároveň nejrozšířenější Arduino Uno disponující již zmíněným mikrokontrolerem ATmega328 o frekvenci 16 MHz. Všichni zástupci této platformy umožňují komunikaci v rámci rozhraní USB, UART, I2C a SPI. Důležité taky nezapomenout na sérii desek Arduino MKR, které byly vytvořeny za účelem užití v konceptu IoT, tedy splňují vlastnosti kompaktnosti, nízké spotřeby energie a umožňují spolehlivou bezdrátovou konektivitu. Mimo klasických komunikačních rozhraní disponují třeba technologií Wi-Fi<sup>7</sup>, GSM neboli komunikace po síti mobilního operátora, NB-IoT<sup>8</sup>, opět komunikace s využitím mobilní sítě a technologií LoRa<sup>9</sup>. Mikropočítače ze série Arduino MKR staví na různých mikrokontrolerech z rodiny ARM Cortex-M0+. Platforma Arduino disponuje i vlastním vývojovým prostředím Arduino IDE [22] s možností vytváření, ladění a nahrávání programů přímo do desek. Díky popularitě platformy existuje i bohatý ekosystém knihoven usnadňující vývoj na této platformě [57].

Další rozsáhlou platformou, někdy také označovanou za platformu miniaturních jednodeskových počítačů, je skupina Raspberry Pi. Ve srovnání s platformou Arduino jsou mikropočítače Raspberry Pi výrazně výkonnější a umožňují plnohodnotný běh operačního systému, tento benefit samozřejmě přináší i svá negativa, mikropočítače Raspberry Pi mají výrazně vyšší energetické nároky, větší rozměry a také pořizovací cenu. Nejvýkonnějším modelem z této rodiny je aktuálně Raspberry Pi 5 s čtyřjádrovým procesorem Arm Cortex-A76, možností až 8 GB operační paměti a podporou až dvou zobrazovacích monitorů. Platforma však nabízí i kompaktnější a méně výkonný model Raspberry Pi Zero (existuje i verze Raspberry Pi Zero W s integrovaným Wi-Fi a Bluetooth modulem) vybaven jednojádrovým procesorem Broadcom BCM2835 s frekvencí 1 GHz a operační pamětí 512 MB. Platforma nachází stejně jako platforma Arduino využití v oblasti IoT zásluhou hardwarové vybavenosti umožňující spolehlivou bezdrátovou konektivitu. Mikropočítače z této rodiny nalezneme v konceptu IoT zejména v místech potřeby běhu operačního systému (například

<sup>7</sup>Seznam desek s podporou Wi-Fi: <https://docs.arduino.cc/arduino-cloud/hardware/wifi/>

<sup>8</sup>Seznam desek s podporou GSM a NB-IoT: <https://docs.arduino.cc/arduino-cloud/hardware/cellular/>

<sup>9</sup>Seznam desek s podporou LoRa: <https://docs.arduino.cc/arduino-cloud/hardware/lora/>

projekty s webovým serverem), zpracování obsahu vyžadující vysoký výkon (zpracování obrazu, videa, streaming, komplexní výpočetní aplikace) nebo v případech potřeby multijazyčného programování, Raspberry Pi podporuje programování nejen v C++, ale i ve velmi rozšířeném programovacím jazyce pro IoT Pythonu [66], dalšími méně rozšířenými alternativy mohou být platformy mikropočítačů BeagleBone, Banana Pi nebo VisionFive, která však nepoužívá ARM architekturu procesorů.

Parametry a vlastnosti mikropočítačů i mikrokontrolerů nabízejí tedy ideální rozhraní pro implementaci v rámci konceptu IoT, potažmo v konceptu Smart cities, kde je využito hlavně jejich nízkých pořizovacích nákladů, protože v rámci rozsáhlých implementací v IoT se může jednat klidně i o tisíce zařízení. Dále taky díky malým rozměrům umožňujících jednoduchou integraci do různých zařízení a senzorů, nízké spotřebě energie, možnosti komunikace v rámci různých komunikačních rozhraní (Wi-Fi, Bluetooth, Zigbee, Lora a další [3]), přizpůsobitelnost díky podpoře širokého spektra programovacích jazyků nabízejících flexibilitu a implementaci různorodých funkcionalit pro přizpůsobení zařízení specifickým funkcionalitám. V neposlední řadě by se taky nemělo zapomínat na rozsáhlou podporu a širokou komunitu vývojářů pracujících s mikrokontrolery a mikropočítači, kteří se podílejí na vytváření knihoven, kódů a nástrojů, například již zmíněné Arduino IDE [22] nebo nástroj Ardestan [43] (vizuální programovací jazyk pro Arduino) usnadňující vývoj aplikací pro IoT.

Příkladem konkrétní realizace může být projekt Smart Home with Raspberry Pi and Home Assistant<sup>10</sup>, který staví na platformě Home Assistant a mikropočítači Raspberry Pi 4, na který je platforma nainstalována. Jedná se o projekt pro domácí automatizaci umožňující nejen fyzické, ale i vzdálené řízení osvětlení, teploty, vlhkosti, ovládání spotřebičů na dálku (spínače a zásuvky), řízení bezpečnosti a mnoho dalších chytrých zařízení, které lze připojit k centrálnímu řídicímu mikropočítači v podobě Raspberry Pi.

Příkladem projektu na méně výkonné platformě mikropočítačů Arduino může být projekt Air Quality Monitor<sup>11</sup>, zaměřující se na monitorování kvality ovzduší v různých prostředích, od domácnosti až po průmyslové zóny. Základem je tedy mikropočítač z rodiny Arduino, konkrétně Arduino Uno, který komunikuje s NOx (koncentrace oxidů dusíku), CO (úroveň oxidu uhelnatého) a PM2.5 (množství jemných částic ve vzduch) senzory. Díky přítomnosti Wi-Fi modulu může se senzory komunikovat bezdrátově a získaná data zasílat na vzdálený server, pro zobrazení dat například ve webovém rozhraní.

---

<sup>10</sup>Odkaz na projekt implementující Raspberry Pi: <https://www.home-assistant.io/>

<sup>11</sup>Odkaz na Arduino projekt: [https://projecthub.arduino.cc/abid\\_hossain/air-quality-monitor-14f9b4](https://projecthub.arduino.cc/abid_hossain/air-quality-monitor-14f9b4)

## Kapitola 3

# Typy dashboard vizualizací

Dashboard je moderní nástroj poskytující uživatelům přehled o klíčových informacích pomocí jednoho grafického rozhraní. Lze jej tedy také definovat jako GUI neboli grafické uživatelské rozhraní, nebo také jako nástroj pro vizualizaci klíčových informací v kompaktní formě. Cílem přehledových obrazovek typu dashboard je poskytnutí okamžitého přehledu o důležitých aspektech daného systému a usnadnit tak rychlé informované rozhodování. [49] Hlavní funkcí je tedy vizualizace dat, která musí být ve srozumitelné formě, aby i složitější informace bylo možné jednoduchou a srozumitelnou formou prezentovat ve formě podporující rychlé a efektivní vnímání dat (využití grafů, tabulek a dalších vizuálních prvků). Dashboard je zároveň také centralizované místo sběru informací z různých zdrojů, což usnadňuje přístup k informacím bez nutnosti přepínání mezi různými aplikacemi nebo systémy. Tento fakt umožní nejen zrychlený přístup k informacím z mnoha zdrojů na jednom místě, ale také eliminuje fragmentaci dat (vznik důsledkem používání různých aplikací a systémů), zlepšuje konzistenci a jednotnost dat, protože pro sběr dat mohlo být použito různých nástrojů a platforem. Uživatelé také umožňují mít kontrolu a reagovat na události ve větším celku v závislosti na počtu zdrojů dat a informací. Tyto získané informace poskytují kontrolu nejen nad aktuálním stavem, ale při pohledu z širší perspektivy poskytují informace o celkové výkonnosti organizace nebo projektu. Tím vzniká další funkce zobrazující klíčové ukazatele výkonnosti (KPI) [49, 25].

Jak již bylo zmíněno, KPI [25] poskytují uživatelé komplexní a snadno srozumitelný pohled na klíčové aspekty výkonnosti organizace, jsou tedy důležitým obohacujícím prvkem dashboardů a klíčovým nástrojem pro efektivní řízení a rozhodování. Pro větší pochopení jejich důležitosti je nutné si osvětlit některé ze základních funkcionalit, které poskytují [47, 25].

KPI umožňují definování a měření měřitelných cílů organizace, tedy po integraci těchto cílů do dashboardu je možné sledovat, zda organizace dosahuje svých strategických plánů. Cíle musejí být jasně definovány a specifikovány bez obecných formulací, tak aby obsahovali konkrétní měřitelný popis. Cíle musejí být zároveň vyhodnotitelné pomocí numerických nebo kvantifikovatelných ukazatelů, pro přesné hodnocení úrovně dosažení. Důležité je také stanovení časového horizontu dosažení cíle. Všechny tyto vlastnosti by měli být také definovány s ohledem na realističnost a dosažitelnost, protože sledování cílů, které není možné splnit je nic nevypovídající a může vést ke frustraci a ztrátě motivace mezi zaměstnanci [47].

Na tuto funkcionalitu navazuje sledování výkonnosti v reálném čase umožňující sledování aktuální situace, reakce na okamžité události a optimalizace strategie v reálném čase. S touto funkcionalitou je spojeno několik klíčových aspektů pro její správné fungování, je vyžadován rychlý přístup k aktuálním datům s minimem zpoždění, automatizace sběru

dat, notifikace a upozornění na specifické události nebo prahové hodnoty a podpora pro okamžitou reakci na změny na základě aktuálního stavu [47].

Dalšími funkcionalitami je správná vizualizace a přehlednost KPI pro rychlé porozumění klíčových údajů, orientace na prioritní aspekty výkonnosti, poskytnutí informací o efektivitě implementovaných strategií organizace pro pružnou optimalizaci strategií a taktik, identifikace dlouhodobých trendů a odhalení potencionálních problémů. Všechny tyto faktory ukazují, že KPI je důležitou součástí dashboardů [47].

Budoucnost oblasti dashboardů je poměrně pozitivní. Neustálý rozvoj technologií společně s narůstajícím objemem dat, tzv. Big Data, a rozvojem umělé inteligence společně se strojovým učením, mohou z dashboardů udělat jeden z klíčových nástrojů budoucího digitálního věku, který poskytne lepší predikce, automatizaci procesu rozhodování, rychlé získávání a zpracování informací a dat, což z nich učiní nenahraditelný prvek moderního řízení a analýzy dat [34].

### 3.1 Existující typy vizualizací pro dashboardy

Vizualizace hraje klíčovou roli při návrhu dashboardů. Správně zvolený návrh vizualizace přispívá k vytvoření přehledných a přitažlivých dashboardů podporujících uživatelskou interakci a zapojení, rychlé a intuitivní porozumění informacím, které zobrazují, podporu srozumitelnosti komplexních dat, která by pouze v textovém formátu mohla být nepochopitelná. Zároveň musí uživateli poskytnout přehled o klíčových informacích na jednom místě, pohled by měl zprostředkovat kompaktní pohled na data sledující vývoj a identifikovat důležité údaje. Data by měla být také vizualizována v čase, pro přesnější identifikování změn a hledání vzorů. Vizualizace by měla umožňovat užití barevného schéma pro zvýraznění klíčových informací, tento faktor je samozřejmě závislý na technologii zobrazovacího zařízení užívaného pro zobrazení dashboardu, v případě že je užití monochromatického zobrazovacího panelu je vhodné užití stupňů šedi pro zvýraznění důležitých prvků. Zobrazovaná data by také měla vizualizace umožňovat zobrazit v různých úrovních, tedy od přehledového pohledu až po úroveň detailů. S tímto je spojena možnost uživatele interagovat s dashboardem od jednoduchých změn pohledu až po aktivní manipulaci s daty. Celá vizualizace by měla být navržena s ohledem na optimalizaci na různá zařízení, na kterých může být zobrazována mimo dashboard a s vědomím, že ne každý uživatel bude odborník na danou oblast problematiky [8].

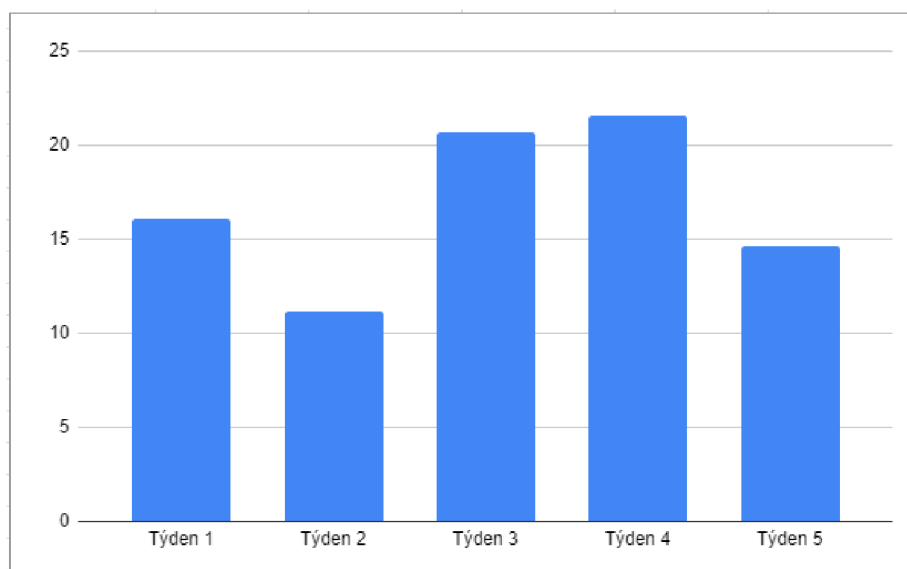
Způsobů vizualizací pro dashboardy je nepřehledné množství. Mezi ty nejjednodušší patří tabulka, jedná se o jednoduchou strukturu skládající se z řádků a sloupců, kde každý řádek odpovídá jednomu záznamu a každý sloupec jedné proměnné nebo atributu. Výhodou tohoto typu vizualizace je univerzálnost, prezentace dat prostřednictvím tabulek nachází uplatnění v mnoha odvětvích a je obecně známá, takže umožňuje čtení velkého spektra uživatelů. V případě užití pro malé datové sady může být výhodou i přehlednost a jednoduchá možnost porovnávání hodnot. V případě užití tabulek na velké datové sady se však z této výhody stává její hlavní nevýhoda, protože se tabulka stává obtížně čitelnou a hlavně nepřehlednou. Obecnou nevýhodou užití tabulek na dashboardech je jejich omezená schopnost vizualizace vztahů a vzorů mezi daty, což ovlivní schopnost pochopení celkového kontextu, který je tabulkou zobrazen. Jejich užití je tedy ideální do oblastí, která jejich prostřednictvím zobrazují jednoduchá data s cílem zkoumání přesných hodnot pro konkrétní body, porovnávání hodnot mezi různými kategoriemi, obdobími a pro data, která jsou dobře organizovaná do kategorií a podkategorií. Příkladová tabulka 3.1 níže zobrazuje hodnoty teplot v jednotlivé dny ve sledované týdny.



	Pondělí	Úterý	Středa	Čtvrtek	Pátek	Sobota	Neděle
<b>Týden 1</b>	10,1	12,1	14,1	16,1	18,1	20,1	22,1
<b>Týden 2</b>	12,1	11,8	11,5	11,2	10,9	10,6	10,3
<b>Týden 3</b>	20,1	20,3	20,5	20,7	20,9	21,1	21,3
<b>Týden 4</b>	23,1	22,6	22,1	21,6	21,1	20,6	20,1
<b>Týden 5</b>	18,2	17	15,8	14,6	13,4	12,2	11

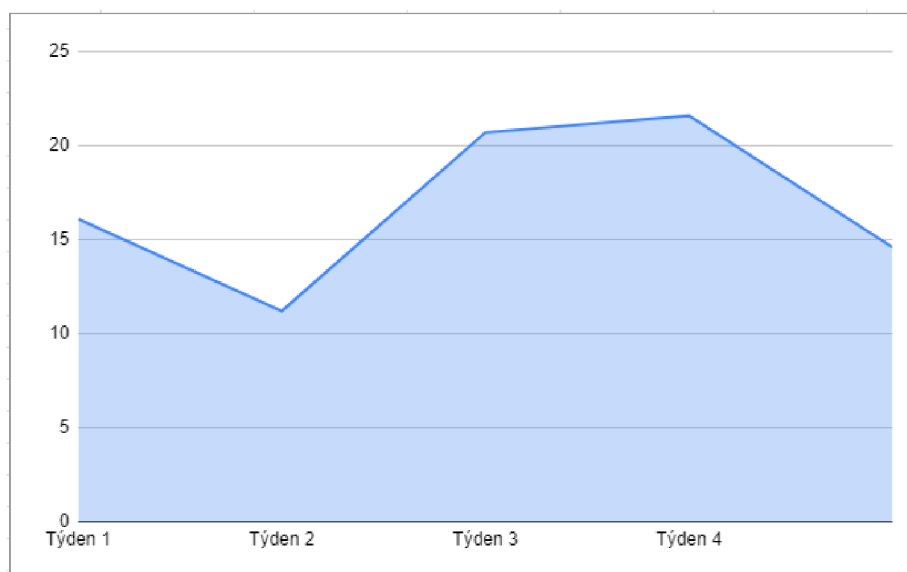
Tabulka 3.1: Tabulka týdenních hodnot

Dalším již mírně pokročilejší formou vizualizací jsou Sloupcové grafy. Jejich uplatnění je ideální pro zobrazování kvantitativních dat a porovnání mezi různými kategoriemi. Graf se skládá z více sloupců, kde každý sloupec reprezentuje hodnotu jednotlivých kategorií. Sloupce jsou nanášeny na osu X (horizontální osa), která je obvykle složena z jednotlivých identifikátorů dané kategorie, aby bylo možné sloupce přiřadit. Osa Y (vertikální osa) reprezentuje kvantitativní hodnoty, které jsou vizualizovány. Výška sloupce na ose Y znázorňuje hodnotu, které daná kategorie dosahuje. Při realizaci je vhodné užití barevného schéma, v případě monochromatických zobrazovacích panelů stupňů šedi pro zlepšení vizuální čitelnosti. Výhodou jejich užití je jednoduchá interpretace, protože jejich forma vizualizace dat je z pravidla snadno čitelná i pro osoby bez školení v oblastech statistiky nebo vizualizace dat, taktéž jednoduché a rychlé porovnávání mezi různými kategoriemi nebo obdobími, které jsou v grafu každý interpretován jedním sloupcem. Nevýhodou sloupcových grafů je jejich nepřesnost pro spojitá data neboli data, která mohou nabývat libovolné hodnoty v určitém rozsahu, kategorická data jsou omezena na konkrétní hodnoty nebo kategorie. Sloupcové grafy jsou primárně optimalizovány pro vizualizaci diskrétních hodnot nebo kategorií dat [1]. Příkladový sloupcový graf níže zobrazuje průměrné teploty v jednotlivé týdny, hodnota průměrné teploty je dána výškou sloupce pro daný týden, na ose Y lze následně z výšky vyčíst úroveň, které teplota dosahuje.



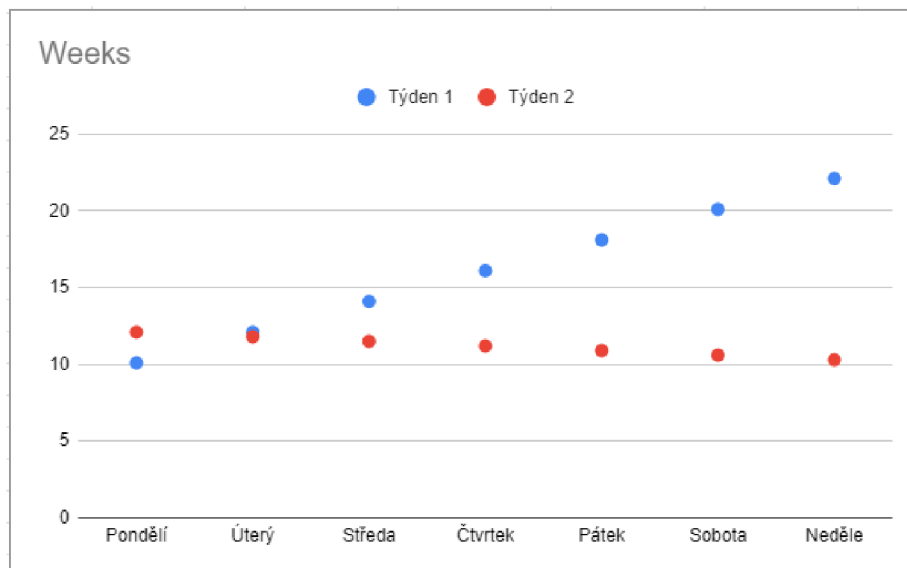
Obrázek 3.1: Sloupcový graf zobrazující průměrné teploty

Graf řádků s užitím lineární časové osy je dalším vizualizačním prvkem. Zobrazuje časové posloupnosti událostí nebo hodnot na rovnoměrné časové škále. Graf je tvořen dvourozměrnou osou, osa X a osa Y. Osa X prezentuje časovou osu chronologicky uspořádanou zleva doprava. Osa Y vyjadřuje hodnoty nebo události spojené s daným časovým okamžikem. Z definice je tedy patrné, že hlavním místem využití je zobrazení časového vývoje, tedy vývoje dat nebo událostí v čase. Lineární časová osa je nezbytná pro analýzu časových řad, mezi které patří finanční data, teplotní a vlhkostní trendy nebo demografické změny. Tento typ grafu je navržen pro vizualizaci spojitých dat, není tedy vhodný pro vizualizaci kategorií nebo diskrétních dat [42]. Na příkladu řádkového grafu níže jsou zobrazeny průměrné hodnoty teplot v dané týdny. Osa X reprezentuje jednotlivé týdny, osa Y značí hodnotu teploty, zní lze následně vyvodit výši dosahované průměrné teploty.



Obrázek 3.2: Řádkový graf zobrazující průměrné teploty

Posledním příkladem vizualizace, vhodný pro užití v této práci je scatter plot neboli rozptylový graf. Jedná se o jeden z nejznámějších vizualizačních nástrojů pro zobrazení dvou spojených kvantitativních hodnot na dvourozměrné ose. Každý bod v grafu vyjadřuje na jedné z os (osa X nebo Y, ze kterých se graf skládá) hodnotu z dané dvojice. Hlavním užitím je vizualizace vztahu mezi dvěma proměnnými, identifikace trendů nebo odhalení vzorců. Jedná se tedy o oblasti statistiky, vědeckého výzkumu, ekonomie a dalších disciplín vyžadující rychlé porozumění datům a snadnou identifikaci vzorců. Výhodou vizualizace touto formou je zjednodušení identifikace anomálií nebo odlehlých hodnot v datech, zachování přehlednosti jak pro velké, tak i malé sady dat nebo efektivita vizualizace vztahů mezi dvěma proměnnými. Mezi nevýhody patří snížená schopnost identifikace trendů v datech a nemožnost zobrazování více než dvou proměnných v rámci jedné vizualizace, což je nevýhodné při práci s více proměnnými [1, 59]. Příkladný rozptylový graf níže zobrazuje hodnoty teploty v jednotlivé dny ve sledované dva týdny. Hodnoty teplot daného týdne mají specifickou barvu, aby je bylo možné od sebe vzájemně odlišit a případně přiřadit



Obrázek 3.3: Rozptylový graf zobrazující teploty

## 3.2 Problematika různých způsobů zobrazení informací

Při výběru vizualizace informací je nutné dbát na několik klíčových aspektů, zejména na srozumitelnost, přehlednost a výkonnost. Existuje mnoho typů vizualizací, některé jsou založeny na zobrazení dat v textové formě a některé naopak v grafické podobě [46]. V ohledu srozumitelnosti a přehlednosti v případě užití grafické vizualizace je výhodou rychlá a intuitivní interpretace dat, ve které lze na první pohled identifikovat trendy, vzory a odchylky i bez hlubšího školení z oblastí statistiky nebo vizualizací dat, jsou tedy zejména vhodné pro prezentaci komplexní dat a informací. S tímto přichází ve většině případů i nevýhoda v podobě ztráty detailu nebo specifické hodnoty, kterou je v grafické vizualizaci, tedy v nějaké formě grafu obtížné identifikovat. Tento problém řeší vizualizace v textové podobě, která umožňuje detailní zkoumání jednotlivých specifických hodnot, s tím ale přichází nárůst časové náročnosti, obzvláště v případě užití pro velké sady dat a samozřejmě ztráta přehlednosti a očividných vztahů a vzorů mezi daty, které poskytuje grafická vizualizace. S ohledem na parametr srozumitelnosti je užití grafické vizualizace tedy zejména vhodné pro velké datové sady, ve kterých je cílem identifikace trendů, vzorů a odchylek, bez nutnosti zkoumání specifických hodnot s důrazem na rychlou vizuální analýzu dat. Textové vizualizace jsou naopak vhodné pro užití v místech potřeby detailního popisu, technických informací a přesných specifikací dat [68, 64].

Výkonnostní aspekty hrají také velmi důležitý parametr při výběru druhu vizualizace. V případě užití grafické vizualizace je nutné pamatovat na složitost implementace daného řešení a případné hardwarové nároky na zobrazení dané vizualizace, v případě textových vizualizací je implementace z pravidla jednodušší a méně časově náročná, nicméně zobrazená data vyžadují vyšší časovou náročnost pro jejich přečtení a interpretaci zobrazovaných informací. V obou případech je také třeba při implementaci pamatovat na uživatele, velké množství informací v textové podobě nebo grafická vizualizace v podobě složitých a komplexních grafů může vést k informačnímu přetížení uživatele a s tím navazujících důsledků [38].

Výslednou vizualizaci je také vhodné optimalizovat pro daný zobrazovací panel dashboardu, který může být implementován technologií, která může přinášet specifická omezení, na které je při návrhu také nutné dbát, například rychlost obnovovací frekvence, spektrum zobrazovaných barev, životnost panelu nebo energetická náročnost vzhledem k zadání realizace, pro kterou je dashboard navržen.

Nejběžnější technologií pro realizaci zobrazovacích panelů je technologie LCD, pro zobrazení obsahu využívá tekutých krystalů, organické sloučeniny v tekutém stavu, které je možné měnit pomocí elektrického pole. LCD panely jsou ve většině případů barevné, existují však i monochromatické varianty. Barvy jsou zobrazeny pomocí zabudovaných filtrů, které kontrolují průchod světla a umožňují zobrazení barev. Podsvícení je realizováno pomocí zářivky se studenou katodou (CCFL) [67] u starších LCD panelů nebo pomocí LED diod [71] u novějších panelů. Všechny pixely na panelu jsou ovládány najednou. Výhodou užití tohoto zobrazovacího panelu je široká dostupnost a tedy i nízká pořizovací cena, panely jsou navíc dostupné v několika kategoriích dle energetické náročnosti. Nevýhodou této technologie je nižší kontrast a problém se zobrazením hluboké černé, takže nemožnost zobrazit určité hodnoty detailu oproti jiným technologiím [24].

Další volbou, kterou lze při realizaci užít je technologie OLED, kterou lze považovat za nástupce LCD. Hlavním technologickým posunem oproti LCD je možnost ovládnutí každého pixelu zvlášť, kdy každý pixel má vlastní podsvícení. Tímto lze přesně určit, které pixely mají být aktivní a které mají být vypnuté. Tento fakt umožňuje zobrazení hluboké černé, vysoký kontrast a nižší energetické náročnosti, protože ne všechny pixely musejí být zapnuté. Dalším benefitem je vysoká odezva, tenký profil a flexibilita. OLED panely lze vyrobit jako flexibilní nebo dokonce zakřivené. Technologický pokrok oproti LCD je však vykoupen vysokou pořizovací cenou a nižší životností, kdy pixely jsou náchylné na degradaci a vypalování [33].

Zajímavou alternativou zobrazovacího panelu ke dvěma již zmíněným technologiím může být technologie E-ink. Na rozdíl od tradičních zobrazovacích technologií využívá k zobrazení obsahu mikroskopické černobílé částice [18] mezi dvěma vrstvami pružného materiálu, částice jsou následně vlivem elektrického pole přesouvány a tvoří černobílý obraz. Existuje i barevné provedení E-ink zobrazovacího panelu, ten na rozdíl od černobílého panelu disponujícího černými a bílými částicemi inkoustu využívá částice tří základních barev, červené, modré a zelené [28]. Hlavní výhodou technologie E-ink je využití okolního světla pro zobrazení obsahu místo emitovaného světla z pozadí u tradičních zobrazovacích panelů, z toho vyplývá, že vyšší intenzita okolního světla oproti panelům jiných technologií jeho čitelnost zvyšuje, tedy ideální například pro čtení na přímém slunci. E-ink zobrazovací panely nevyzařují žádné světlo, takže jsou šetrné k očím při dlouhém čtení, tento fakt zároveň znamená, že bez existence okolního světla není zobrazovací panel čitelný. Jejich zobrazovací schopnost je velmi podobná tištěnému papíru [56], takže reprodukuje obraz s vysokým kontrastem. Zároveň jsou zobrazovací panely technologie E-ink velmi energeticky úsporné, jakmile je jednou daný obraz na panel vykreslen, je udržován bez potřeby elektrické energie až do doby, než je překreslen jiným obrazem. Zásadními nevýhodami E-ink zobrazovacích panelů je již zmíněná neschopnost čitelnosti panelu bez existence okolního zdroje světla a omezená obnovovací frekvence, kterou technologie disponuje. Tento omezující faktor plyne ze samotné konstrukce a technologie vykreslování, kdy jsou obrazy na panelu zobrazovány pomocí přesunu mikročástic, který sám o sobě vyžaduje nějaký čas. Moderní E-ink panely dosahují obnovovací frekvence až 5 Hz, tato hodnota však stále není dostatečná pro zobrazení videí nebo plynulých animací, u barevných E-ink panelů je tato hodnota nižší. E-ink panely jsou navrženy primárně pro zobrazování statického obsahu, kde nízká obno-



vací frekvence není problém, nejčastější oblastí užití na trhu jsou elektronické čtečky knih, u kterých nízká obnovovací frekvence nevadí, protože zobrazují statický obsah v podobě textových stránek, naopak mají prospěch z velmi nízkých energetických nároků, které panely typu E-ink k zobrazování obsahu potřebují a přinášejí tedy výbornou výdrž na baterii. Při porovnání technologie E-ink s technologií LCD nebo OLED v oblasti energetické náročnosti technologie E-ink jasně dominuje, konkrétní porovnání je komplikované nalézt, protože záleží vždy na rozlišení panelů, jasů a dalších faktorech, ale u technologie E-ink je spotřeba vždy v rámci jednotek mW během normálního čtení, u LCD nebo OLED technologie vždy v rozmezí několika set mW až po několik W nebo desítek W [55, 13, 41].

Pro případy postavené na aspektu nízké energetické náročnosti je tedy užití technologie E-ink ideální volba. Díky vlastnosti permanentního zobrazení již vykresleného obrazu bez nutnosti další elektrické energie je velmi energeticky efektivní. Před užitím v dané realitaci je však nutné pamatovat na omezení této technologie, které spočívají hlavně v užití okolního světla pro zobrazení místo tradičně emitovaného světla z pozadí, nízké obnovovací frekvence neumožňující zobrazování plynulých animací nebo videí a další omezení v případě zobrazování barevného obsahu.

## Kapitola 4

# Způsoby a existující knihovny pro vykreslování vizualizací

Vizualizace lze vytvářet v několika úrovních. V nejnižší úrovni se jedná o ruční vytváření, kdy programátor vytvoří vizualizaci celou sám od základu bez užití kódu třetí strany. Častou oblastí výskytu těchto vizualizací stavěných od základu programátorem je oblast webových aplikací. Mezi tyto webové prvky patří prvek Canvas [32] a formát SVG [35]. Canvas je prvek vykreslující grafické elementy webové aplikace pomocí skriptů JavaScriptu. Oblast ohraničenou tímto prvkem v kódu lze programově ovládat a kreslit na ni, využívá metodu bitmapové (rastrové) kresby, takže pracuje s jednotlivými pixely. Scalable Vector Graphics neboli SVG je formát XML pro popis vektorových dvourozměrných grafik, které jsou vytvářeny pomocí matematicky definovaných tvarů a objektů. Díky popisu každého prvku vizualizace pomocí matematické funkce lze prvkům měnit velikost bez ztráty kvality, pracuje tedy s vektorovou kresbou. V případě, že programátor vizualizace nechce stavět od základu, může se přesunout o úroveň výše a využít vizualizačních knihoven pro vytvoření dané vizualizace.

V programování termín knihovna odkazuje na kolekci již napsaného kódu, který je možné užít ve vlastních projektech. Většinou obsahují sady funkcí, tříd nebo modulů s určitou funkcí pro zjednodušení vývoje softwaru. Každá knihovna má určený účel (matematické, vizualizační, síťové a mnohé další knihovny) a platformu neboli programovací jazyk, pro který je určená, některé knihovny mohou být i multiplatformní. Knihovny specializující se na vykreslování vizualizací přináší vývojáři sadu funkcí a nástrojů pro vytváření grafických reprezentací dat. Mezi klíčové funkce většinou patří vytváření grafů a grafických vizualizací, interaktivita (poskytnutí uživateli interaktivních prvků, pro manipulaci s vizualizací, přiblížení, přesouvání, označení dat a další), podpora různých formátů výstupů, které mohou být využity k exportu, customizace a stylování, takže programátor může jednoduše změnit vzhled vizualizace dle potřeby. Mezi další funkce, které některé knihovny pro vykreslování vizualizací přináší jsou nástroje s optimalizací pro práci s velkým množstvím dat, tzv. Big Data, zahrnující techniky pro agregaci, vzorkování nebo dynamické načítání dat. Také se může jednat o nástroje integrující funkce pro analýzu dat, které umožňují snadnější propojení analýzy s vizualizací [17].

D3.js [10] je open-source knihovna umožňující vytváření dynamických a interaktivních vizualizací pro webové aplikace napsaná ve skriptovacím jazyce JavaScript. Popularitu a oblibenost získala zejména díky funkcím a velké komunitě, která ji aktivně udržuje. Mezi klíčové funkce patří již zmíněná interaktivita, tedy možnost přidávat události, animace a roz-

hraní pro uživatele pro zvýšení angažovanosti při prohlížení vizualizací. Dále škálovatelnost a efektivní práce s velkým množstvím dat, plynulé aktualizace vizualizací v reálném čase, podpora širokého množství funkcí pro tvorbu různých typů grafů nebo vizualizací a také selekce, vytváření a manipulace s elementy v DOM stránky [63] (reprezentace struktury dokumentu webové stránky ve formátu stromu, kde každý uzel představuje část dokumentu, navíc umožňuje dynamicky měnit obsah, strukturu a styl webové stránky). Tato vlastnost usnadňuje propojení vizualizace s aktuálními daty.

V případě odhlédnutí od webových aplikací je oblíbenou knihovnou pro vykreslování vizualizací knihovna Matplotlib [69]. Jedná se o knihovnu psanou v jazyce Python široce využívanou ve vědeckých výzkumech, analýze dat a různých oblastech datové vědy. Silnou stránkou jsou grafové vizualizace, od základních sloupcových grafů přes histogramy až po komplexnější 3D grafy, všechny grafové vizualizace jsou zároveň interaktivní. Podporuje sázení matematických vzorců pomocí syntaxe LaTeX, což je při komplexních řešeních velký benefit. Vzhledem k zaměření hlavně do výpočetní sféry je jednou z dalších výhod jednoduchá integrace s knihovnou NumPy [45], knihovna pro jazyk Python podporující práci s multidimenzionálními poli a matematickými funkcemi pro operace nad danými poli. Jako v případě D3.js knihovny i knihovna Matplotlib disponuje širokou komunitou s rozsáhlou podporou. Nevýhodami jsou složitosti implementace komplexních vizualizací, kdy je mnohdy nutné užití dalších knihoven. Kód knihovny zároveň není vždy úplně aktuální podle nejmodernějších trendů a působí zastarale. Na této knihovně staví další modernější knihovna Seaborn [54], která přidává funkce pro zjednodušení tvorby esteticky příjemných a informativních vizualizací grafů. Omezením této knihovny je zaměření hlavně na statistické analýzy ve dvou dimenzích, nemá tedy podporu pro komplexní vizualizace hlavně z oblasti 3D, v případě nutnosti užití vizualizace z této oblasti je tedy nutný návrat ke starší knihovně Matplotlib, což naštěstí díky úzkému spojení těchto dvou knihoven není zásadním problémem.

Knihovnami pro vykreslování vizualizací disponují i netradiční zobrazovací platformy, příkladem může být technologie E-ink. Ve spojení sní lze vybrat například open-source knihovnu U8g2 Library [44] (Universal Graphics Library for 8-bit and 32-bit Embedded Systems) pro programovací jazyk C. Tato knihovna není sice specificky určená jen pro zobrazovací panely technologie E-ink, nýbrž i pro panely typu LCD nebo OLED, nabízí však vysoký počet nástrojů právě i pro E-ink panely. Knihovna je navržena pro vývoj embedded systémů postavených na 8 až 32 bitových mikrokontrolerech, speciálně pro mikrokontrolery z rodiny Arduino. Mezi základní funkce, kterými knihovna mimo podpory různých typů zobrazovacích panelů disponuje patří podpora pro scrolling neboli posouvání obsahu po zobrazovacím panelu, různé fonty pro zobrazený text, podpora různorodosti rozlišení a fyzických rozměrů panelů, optimalizace pro efektivní a rychlé vykreslování obsahu a vykreslování různých grafických prvků, například čáry, kruhy, obdélníky, trojúhelníky a další. V neposlední řadě taky disponuje širokou aktivní komunitou, která přináší obsáhlou dokumentaci, širokou škálu zdrojů a podporu od ostatních vývojářů.

Mezi nejobvyklejší řešení vizualizací pro dashboardy se řadí webové technologie, například knihovna D3.js díky rozsáhlým možnostem a funkcím, kterými disponuje. Pokud však vývojář vytváří dashboard na základě nízkourovňového zařízení ve spojení s netradičním zobrazovacím panelem například typu E-ink, jsou jeho možnosti výrazně omezenější. Zařízení často nemá podporu nebo dostatečný výkon pro renderování složitých webových vizualizací, zároveň chybí podpora pro samotné zobrazovací panely. Dostupné knihovny pro E-ink panely jsou nízkourovňové a disponují vestavěnými vizualizacemi pouze pro základní geometrické tvary s pár základními textovými sadami v omezených možnostech nastavení

velikosti fontu. Vývojář je tedy nucen v případě tvorby složitější vizualizace, například v případě vykreslování grafů, celou grafovou knihovnu napsat sám od základu jen s užitím základních vestavěných geometrických tvarů, navíc renderování probíhá manuálně na základě definování souřadnic, ve kterých se mají jednotlivé prvky (čáry, osy, tečky a další), ze kterých se skládá vizualizace, vykreslit. Vytvořené vizualizace navíc nejsou obvykle mezi E-ink panely různých výrobců přenosné, protože každý panel má jinou implementaci zobrazování.

Na trhu jsou nyní dostupné firmy, které se zabývají vytvářením jednoduchých drag & drop editorů pro návrhy a realizace vizualizací, například firma DWIN<sup>1</sup> nebo Game Ever s jejich produktem SquareLine Studio<sup>2</sup>, v obou případech však stále chybí podpora pro zobrazovací panely typu E-ink.

## 4.1 Zobrazovací způsoby aplikovatelné v chytrých domácnostech

Data získaná prvky chytré domácnosti je nutné vhodným způsobem uživateli prezentovat. Při volbě daného způsobu je nutné brát ohled na dostupnost, odkud je uživateli umožněn přístup k vizualizaci, velikost a formát, hardwarové nároky dané vizualizace, energetická náročnost provozu daného způsobu, cena a případná integrace s jinými zařízeními nebo sdílení mezi uživateli.

Základním zobrazovacím prvkem chytré domácnosti jsou dashboardy. Ty lze realizovat pomocí mnoha technologií, nejzákladnějším způsobem realizace jsou segmentové displeje [60]. Segmentové displeje jsou tvořeny z mnoha samostatných segmentů. Segmenty lze realizovat pomocí několika technologií, mezi nejběžnější patří technologie LCD [65] (segment je tvořen z tekutých krystalů), LED [65] (segmenty používají LED diody jako zdroj světla, v každém segmentu je jedna nebo více diod, vyšší kontrast než segmenty tvořené technologií LCD), OLED [27] nebo technologie VFD [70] neboli vakuový fluorescenční displej tvořený z fluorescenčních trubic umístěných ve vakuu. VFD disponuje vysokým kontrastem a širokými pozorovacími úhly. Obecně lze u všech segmentových displejů nezávisle jednotlivé segmenty aktivovat nebo deaktivovat, aby tvořily jednoduché symboly, písmena nebo čísla. Segmentové dashboardy nejsou vhodné pro zobrazování složitých grafických informací nebo barev, jsou efektivní pro jednoduché úkoly, zejména konkrétní čísla nebo stavové informace. V případě potřeby zobrazování komplexnějších informací je nutné zvolit jinou technologii pro dashboard než segmentový displej, jedná se zejména o technologie LCD, LED, OLED nebo E-ink. Problematiku výběru mezi danými technologiemi pro dashboard řeší již kapitola 3.2 výše.

Variantou pro prezentaci dat chytré domácnosti ke klasickým dashboardům mohou být mobilní aplikace [23, 50]. Ty nabízejí uživateli monitorování a změny stavu chytré domácnosti pomocí svého mobilního zařízení. Aplikace by měly být multiplatformní (alespoň Android a iOS), aby pokryly co možná největší spektrum uživatelů. V případě, že chytrá domácnost data odesílá na vzdálený cloud nebo server, je uživateli s přístupem na internet umožněn přístup do ekosystému dané chytré domácnosti téměř odkudkoliv. Mobilní aplikace mohou s prvky chytré domácnosti komunikovat i na přímo, například pomocí technologie Wi-Fi nebo Bluetooth. V takovém případě je přístup k datům a správě daných prvků chytré domácnosti omezen na základě vlastností užitých komunikačních technologií, podrobněji jsou

<sup>1</sup>Firma DWIN: <https://www.dwin-global.com/>

<sup>2</sup>Drag & drop SquareLine Studio editor: <https://squareline.io/>



jednotlivé komunikační technologie rozebrány v kapitole 2. Součástí mobilních aplikací bývají často hlasoví asistenti [53] umožňující uživateli komunikovat se systémem jen pomocí hlasových příkazů a posouvají tak možnosti ovládání na novou úroveň. Nevýhodou mobilních aplikací bývá zaměřenost na jeden konkrétní ekosystém chytré domácnosti daného výrobce, proto v případě kombinace více ekosystémů, například Apple HomeKit<sup>3</sup>, Google Home<sup>4</sup> nebo Xiaomi Home<sup>5</sup>, je nutné mít ke každému vlastní mobilní aplikaci. Výhodami oproti dashboardům jsou rozsáhlejší možnosti práce s chytrou domácností. Nejedná se ve většině případů jen o vizualizaci dat, mezi rozšíření, které nabízí, patří již zmíněný přístup a správa chytré domácnosti téměř odkudkoliv. V případě, že daná chytrá domácnost komunikuje s uživatelem skrz cloud nebo vzdálený server, dále také možnosti změny nastavení nebo aktualizace firmwaru jednotlivých prvků chytré domácnosti. V neposlední řadě je mezi výhody důležité zařadit schopnost zasílání notifikací nebo upozornění o stavu chytré domácnosti na dané mobilní zařízení, kdy je uživatel automaticky upozorněn na změnu stavu dané sledované události a může pohotově reagovat. V dnešní době jsou mobilní aplikace u komerčních ekosystémů chytrých domácností považovány za samozřejmost, která uživateli přináší přehledný a pohodlný nástroj pro monitorování a správu celého systému.

V případě nutnosti vzdáleného monitorování a správy chytré domácnosti bez užití mobilního zařízení se nabízí alternativa webového portálu [30, 15]. Jedná se o webovou aplikaci získávající data z cloudu nebo vzdáleného serveru, který komunikuje s chytrou domácností. Hlavní výhodou oproti mobilní aplikaci je platformní nezávislost, stačí tedy, aby zařízení bylo připojeno k internetu a disponovalo webovým prohlížečem. Vzhledem k požadavkům jej lze taktéž zobrazit pomocí mobilních zařízení, navíc však i pomocí zařízení, jako jsou klasické počítače, notebooku, mikropočítače nebo chytré televize, které také disponují přístupem k internetu a webovým prohlížečem. Stejně jako v případě mobilních aplikací mají často k dispozici funkce pro správu, konfiguraci a monitorování jednotlivých prvků dané chytré domácnosti. Nevýhodami webových portálů je optimalizace uživatelského rozhraní, spektrum zařízení a formátů zobrazovacích panelů, které mohou obsah webového portálu zobrazovat je tak rozsáhlý, že není možné zajistit, aby byl obsah vždy intuitivně a přehledně zobrazen jako u mobilní aplikace. Dále nedisponují offline přístupem, mobilní aplikace na rozdíl od webových portálů mohou mít funkce k procházení historie uložených záznamů nebo plánování události, které se aplikují v momentě, kdy se zařízení opět připojí k internetu. Mobilní aplikace zároveň často nabízejí lepší schopnosti využití hardwaru daného mobilního zařízení, senzory, fotoaparát nebo mikrofony pro již zmíněné hlasové asistenty.

Při reálném užití jsou většinou zmíněné zobrazovací způsoby mezi sebou kombinovány vždy s ohledem na kontextu použití, v praxi se lze nejčastěji setkat s kombinacemi mobilní aplikace a webového portálu nebo dashboard panelu společně s webovým portálem.

---

<sup>3</sup>Odkaz na platformu Apple HomeKit: <https://developer.apple.com/design/human-interface-guidelines/homekit>

<sup>4</sup>Odkaz na platformu Google Home: <https://home.google.com/what-is-google-home/>

<sup>5</sup>Odkaz na platformu Xiaomi Home: <https://trust.mi.com/docs/iot-privacy-white-paper-global/>  
3/6

## Kapitola 5

# Analýza požadavků a současného stavu

Návrh vizualizace dat ze senzorů chytré domácnosti je nutné vždy promyslet a přizpůsobit mnoha faktorům, základním bodem při návrhu je analýza uživatelů potenciaálně užívající dané řešení. Uživatelé nejsou vždy na stejné technické úrovni, tedy všichni nemají stejné schopnosti chápání komplexních vizualizací se správným porozuměním, důležitým faktorem je i věk uživatelů viz zdroj [16], s tímto je spojené také rozdělení uživatelů do kategorií na správce a uživatele bez větších pravomocí. V případě běžných uživatelů se jedná tedy primárně o odebírání obsahu bez přístupu ke konfiguraci a nastavení dané chytré domácnosti, tyto možnosti by naopak měly být zpřístupněny správcům. Pro správce tedy bývá vhodným řešením vizualizací komplexnější aplikační řešení s možnostmi konfigurace, většinou se jedná o mobilní aplikace nebo webové portály. Pro ostatní uživatele bývá vhodným řešením pro prezentaci dat přehledový dashboard, který obvykle umožňuje jen vizualizaci obsahu bez dalších komplexnějších funkcí a zároveň uživateli usnadní a urychlí přístup k informacím, protože není nucen navštěvovat složitý aplikační systém. Dashboard může v chytré domácnosti nalézt také uplatnění v místech, kde by byl komplikovaný přístup do složitějšího aplikačního systému pro všechny uživatele bez rozdílu, například místa bez přístupu k elektrické energii (skleník, bazén a další) - tato realizace je samozřejmě možná jen pro dashboard s implementovaným akumulátorem bez nutnosti napájení ze sítě. Dalším bodem, který je při návrhu nutné zohlednit jsou typy použitých senzorů v rámci domácnosti.

Oblast senzorů je velmi rozsáhlá sféra tvořena mnoha zástupci a typy, každý měří různá data dle sledované skutečnosti nebo veličiny, proto nelze obecně generalizovat výstup ze všech senzorů a je nutné přizpůsobit zpracování i následnou vizualizaci každému typu senzoru zvlášť. Důležité je pamatovat na skutečnost, že surová data, která senzory získávají jsou nějaké informace z fyzikálního světa reprezentovaná v měřitelné a zpracovatelné signálové podobě (může se například jednat o hodnoty teploty reprezentované jako napětí [9]), senzory tedy většinou disponují také vybavením, které tyto data v surové podobě (signály, napětí, atd.) zpracuje do srozumitelného významu v podobě konkrétní veličiny.

Nejběžnějšími senzory jsou senzory pro sledování teploty, vlhkosti, senzory pohybu (tyto senzory obvykle fungují na principu poskytování dat o lineární vzdálenosti nebo uhlových stupních), energetické spotřeby nebo světelnosti, datovým výstupem senzoru je v tomto případě obvykle hodnota luxů udávající intenzitu světla. Mezi senzory s primárně průmyslovým zaměřením lze zařadit senzory kvality vzduchu (data o aktuální úrovni CO<sub>2</sub> nebo počtu prachových částic), hluku nebo senzory pro sledování vibrací, které poskytují dato-

vou informaci o jejich úrovni [52, 62]. Sensory se od sebe nejen liší výstupními daty, mohou se lišit i v samotném formátu, ve kterém data poskytují, tato vlastnost je nicméně také ovlivněna platformou, pro kterou je daný senzor vytvořen.

Jak bylo již zmíněno, každému typu senzorů je nutné nejen přizpůsobit zpracování poskytovaných dat, ale i jejich následnou vizualizaci. Mezi základní faktory pro výběr nejvhodnější vizualizace výstupu z daného senzoru patří samotný typ a množství získaných dat, které je nutné vizualizovat pro zachování kontextu, například teplotní nebo vlhkostní senzory ve většině případů poskytují data závislé v čase, tedy hodnotu a k ní časové razítko, kdy byla zaznamenána. Z těchto důvodů je vhodné zvolit vizualizaci, která bude také data vizualizovat v čase. Řádkové grafy jsou častým prostředkem vizualizace časových posloupností událostí, mezi které výstupy z těchto senzorů obvykle spadají. V případě potřeby vizualizace více senzorů s časově závislým výstupem v rámci jedné vizualizace jsou vhodnou a přehlednou alternativou rozptylové grafy. Příklady zmíněných grafů se detailněji zabývá kapitola 3.1. Existují však i senzory poskytující komplexnější sady dat, například akcelerometry, které poskytují data o zrychlení ve třech osách, X, Y a Z. V těchto případech už nelze v rámci zachování kontextu a srozumitelnosti vizualizace využít jen dvouosé grafy, nýbrž komplexnější 3D vizualizace [12]. U těchto typů vizualizací je navíc nutné počítat s vyššími požadavky na hardware a komplikovanější implementací pro danou vizualizace.

Výběr vizualizace pro daný senzor není vždy na první pohled jednoznačný, je nutné jej vždy přizpůsobit charakteristikám poskytovaných dat, nejen jejich typu, ale i časovým rozmezím, ze kterého pocházejí. Pro vizualizace typu řádkového, sloupcového nebo rozptylového grafu je nutné užití dat v numerické podobě, ideálně se stálým časovým odstupem pro zachování srozumitelnosti kontextu. Dále hardwarovým schopnostem zařízení vykreslující vizualizaci, ale i potřebám uživatele nebo aplikace, které může být vizualizace součástí. Uživatel může například požadovat daná data zobrazovat pomocí různých typů vizualizací dle jeho potřeby a situace. Vhodnou součástí metodiky návrhu a tvorby vizualizace je tedy experimentování s různými vizualizačními technikami pro výběr nejefektivnější formy prezentace naměřených dat z daného senzoru a zároveň splnění potřeb uživatele.

## 5.1 Analýza firmy Logimic

Česká firma založená v roce 2017<sup>1</sup> zabývající se vývojem a implementací IoT se zaměřením na průmysl a veřejnou správu. Mezi produkty, které firma nabízí, patří řízení a monitorování venkovního osvětlení za účelem úspory elektrické energie, programování světelných profilů a notifikace o poruchách společně se statistiky a reporty, dalšími příklady jsou inteligentní řízení topných těles v průmyslových budovách ve spolupráci s firmou Lersen CZ, monitorování zejména teplotních a vlhkostních stavů budov (bezdrátové řešení iTemp) nebo měření kvality ovzduší a mnohé další.

Firma tedy nabízí a poskytuje širokou škálu produktů a služeb, jeden z nich svou významností pro firmu však stojí nad ostatními, jedná se o platformu ACADA (Asset Control and Data Acquisition)<sup>2</sup>, cloudová platforma umožňující zákazníkům shromažďovat, analyzovat a vizualizovat data z IoT zařízení. Je hostovaná na architektuře AWS a část na AWS Lambdách (bezobslužná výpočetní platforma umožňující spouštění kódu bez správy serverů) zejména za účelem automatizace obchodních procesů. Je založena na otevřených standardech, mezi které patří REST-API, HTTP, LoRa a další. Platforma nabízí i On-

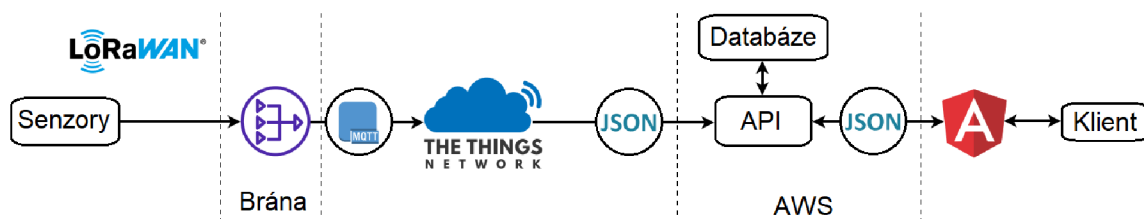
<sup>1</sup>Oficiální webové stránky a zdroj informací o firmě Logimic: <https://www.logimic.com/cs/>

<sup>2</sup>Zdroj informací o platformě ACADA: <https://www.logimic.com/cs/platforma/>

premise řešení, tedy možnost umístění veškerého softwaru a hardwaru v interní infrastruktuře firmy zákazníka, zároveň se také umí integrovat s technologiemi třetích stran (ERP systémy - Enterprise Resource Planning, CRM systémy - Customer Relationship Management, E-commerce platformy a další). Běh platformy je současně hardwarově nenáročný schopný běhu i na mikropočítačích typu Raspberry Pi. Poskytování, sběr a sdílení dat řeší pomocí aplikačního programového rozhraní.

Aplikační programové rozhraní, dále jen API [14], je rozhraní umožňující komunikaci jedné aplikace s jinou. Umožňuje programátorům integrovat a využívat funkce jiných programů nebo služeb bez nutnosti znalosti vnitřní implementace. API se dělí na několik typů, API operačního systému umožňující komunikaci s jádrem operačního systému za účelem poskytnutí rozhraní pro práci s funkcemi souborového systému, pamětí, procesy a dalšími systémovými zdroji. Dalším příkladem jsou knihovní API poskytující sady procedur a funkcí zabudovaných v knihovnách nebo frameworkích pro vývoj aplikací, tyto API slouží pro interakci s funkcionalitami dané knihovny nebo frameworku. V neposlední řadě je nutné zmínit ještě kategorii webových API, která slouží pro komunikaci mezi různými webovými službami pomocí protokolu HTTP.

Architektura sběru dat u firmy Logimic je založena na získávání dat ze senzorů a jiných koncových zařízení, které jsou obvykle přeneseny pomocí technologie LoRaWan k nejbližší dostupné bráně. Z ní data putují ve formátu zpráv protokolu MQTT (jedná se o lehký, otevřený a spolehlivý protokol pro přenos zpráv mezi zařízeními v IoT) až na TTN (The Things Network - decentralizovaná infrastruktura pro internet věcí), kde proběhne dekodování zprávy a uložení odpovídající reprezentace ve formátu JSON. Před samotným uložením do databázových systémů je nutné nad daty provést další úpravy (převod do určitého formátu, výpočet odvozených hodnot), které samotné senzory neposkytují. O tyto úpravy a zpracování se stará vrstva složená z AWS Lambda funkcí. Následně získávání již zpracovaných a uložených dat z databázových systémů probíhá obdobným způsobem, jako jejich zápis na databázový systém, dochází ke kontaktu odpovídající AWS Lambdy prostřednictvím API, která provede autentifikaci a autorizaci uživatele a pokud je k požadované operaci oprávněn, vyhodnotí dotaz a zašle výsledky ve formátu JSON. Na klientské frontendové části jsou výsledky zobrazeny aplikací založené na aplikačním rámci Angular (open-source framework pro vývoj webových aplikací). Celou architekturu graficky popisuje obrázek 5.1.



Obrázek 5.1: Architektura komunikace

Pro komunikaci a získávání dat od Logimicu je nejdůležitější oblast webových API, firma Logimic má standardizovaný popis pro své API - sada specifikací, které popisují, jak s daným API komunikovat, například definice dostupných endpointů v API spolu s metodami (GET, POST a další), které na nich lze použít, dále informace o možných parametrech předaných do API, definice vstupních a výstupních datových typů API společně s jejich strukturou a možnými hodnotami, kterých mohou nabývat. Součástí standardizovaného popisu je také



bezpečnostní schéma (popis autentizace a autorizace přístupu), metadata a popis různých typů odpovědí, které může API vrátit.

Logimic pro přístup ke svému API využívá návrhový vzor REST API, podrobně o tomto návrhovém vzoru pojednává kniha [37]. Jeho užití poskytuje přístup ke CRUD operacím (vytváření, čtení, aktualizace, mazání informací se serveru) prostřednictvím HTTP přístupu. Komunikace prostřednictvím webových API funguje na základě několik kroků, nejprve je zaslána žádost (request), nejčastěji se jedná o HTTP požadavek nesoucí informace o typu operace, která bude provedena na zdroji (GET - získání dat, POST - odeslání dat, PUT - aktualizace dat, DELETE - smazání dat), adresu zdroje (URL - Uniform Resource Locator), hlavičky s dodatečnými informacemi o žádosti (typ obsahu, požadovaný formát odpovědi, autorizaci a další) a na závěr samotné tělo požadavku nesoucí data, obvykle pro požadavky typu PUT a GET. Serverová strana zpracuje žádost podle definovaných pravidel a informací a vytvoří odpověď pro klienta. Odpovědi může být výsledek operace nebo požadovaná data obvykle ve strukturované formě dat, u firmy Logimic ve formátu JSON. Následné zpracování odpovědi závisí jen na klientských potřebách.

Pro integraci platformy programátora s platformou firmy Logimic je nutné užití webových API firmy, které je dostupné pouze pro vývojáře s uděleným přístupem. Pověřený uživatel následně získá přístup k dokumentaci popisující funkcionality API a také unikátní API klíč potřebný pro provádění požadavků na API. Programátorovi je k dispozici sada endpointů pro získávání dat, v prvním kroku si musí uživatel vygenerovat na základě svých přidělených přístupových údajů přístupový JWT access token s určitou dobou platnosti. Tento token je nezbytný pro ověřování a je tedy nutným parametrem každého dotazu.

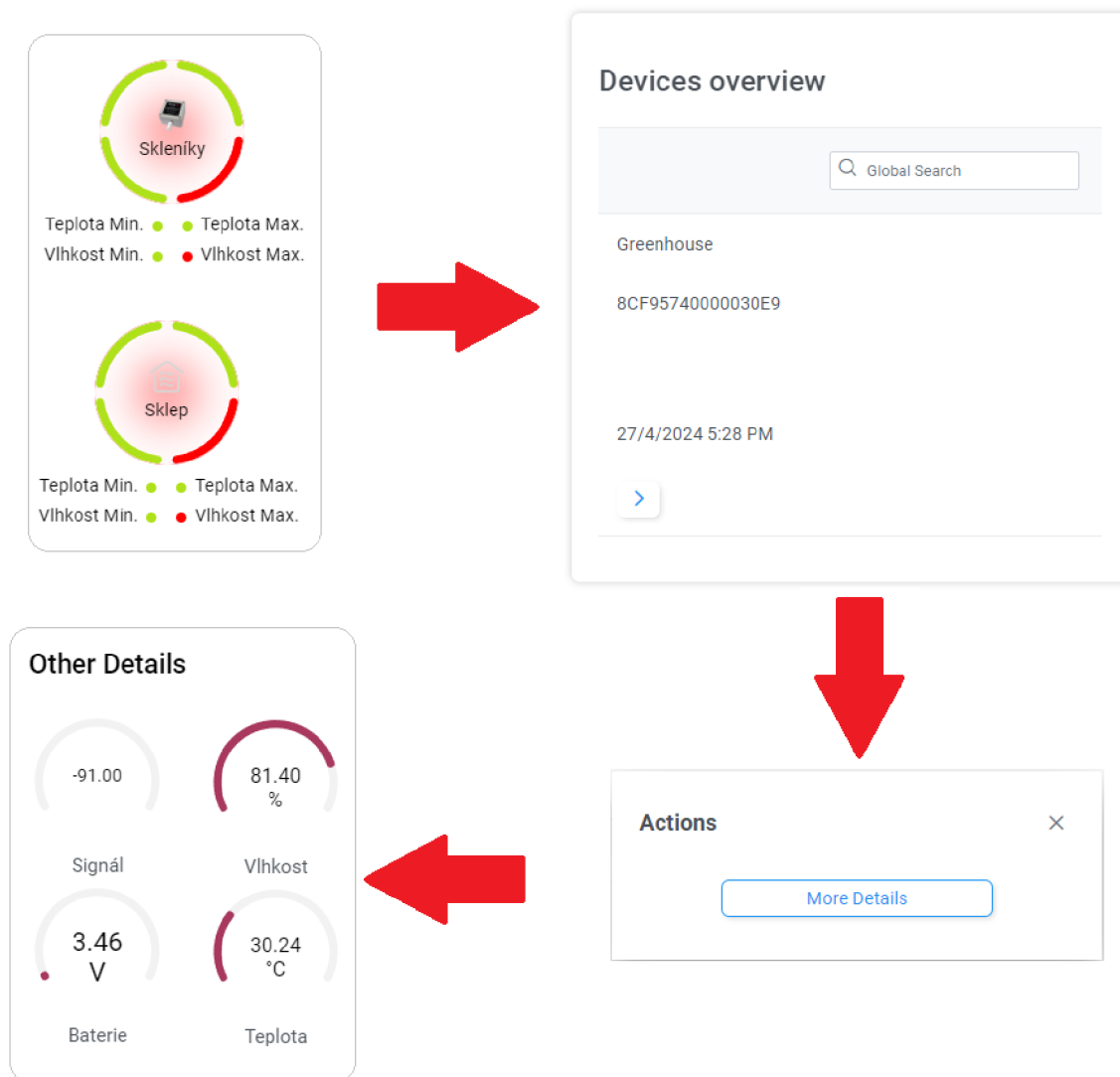
Pro další dotazy na ostatní dostupné endpointy musí uživatel znát také uživatelský token (trvale svázan k uživateli) nebo ID systému, ke kterému má přístup, a z kterého bude data stahovat. Na získání ID systému je nutné znát uživatelský token, bohužel na jeho získání systém neobsahuje příslušný endpoint, proto je nutné ho odposlechnout z komunikace s webovým rozhraním firmy nebo získat v aplikaci ACADA. Po získání všech nezbytných tokenů a stažení ID systému jsou uživateli zpřístupněny endpointy na získání počtu skupin v systému společně s jejich výpisem. Z výpisu skupin si uživatel může vyparsovat ID identifikující jednotlivé skupiny. S daným ID skupiny může zavolat další dostupný endpoint pro detailní výpis zařízení v dané skupině společně s parametry každého zařízení a jednoznačným identifikátorem zařízení - UID.

Na základě UID zařízení může uživatel zavolat statistický endpoint (získání historických hodnot) pro každé zařízení. Při volání tohoto endpointu je nutné specifikovat parametry pro časové období, ze kterého má být statistika hodnot pro dané zařízení zobrazena, sledované veličiny na zařízení (teplota, vlhkost, napětí baterie a další) a časovou agregaci, tedy statické časové odstupňování, které má být mezi jednotlivými hodnotami navracené statistiky. Všechny zařízení v systému nemusí mít statistiku povolenou nebo ji také nemusí podporovat, lze z nich tedy získat jen poslední hodnotu. Tato hodnota je součástí výstupu endpointu na výpis zařízení ve skupině.

## 5.2 Současný stav nabízený firmou Logimic

Uživatel může nyní data ze senzorů jednotlivých zařízení své chytré domácnosti sledovat jediným dostupným způsobem, tím je neveřejná webová platforma ACADA přístupná pouze pro autorizované uživatele s validními přístupovými údaji. Schopnosti a možnosti, které poskytuje popisuje již druhý odstavec textu 5.1. Její informační obsáhlost je velmi rozsáhlá, nicméně pro méně zdatného uživatele může být velmi nepřehledná a komplikovaná.

V případě potřeby sledování jedno údaje, například teploty, kterou poskytuje daný senzor daného zařízení je uživatel po vstupu na platformu nucen nejprve vybrat skupinu, ve které se nachází dané zařízení, následně z množiny zařízení vybrat dané jedno konkrétní a až následně získá hledanou hodnotu teploty. Celý postup získání hodnoty graficky popisuje obrázek 5.2.



Obrázek 5.2: Postup získání hodnoty parametru v platformě ACADA

Řešením pro méně zdatné uživatele by mohlo být vytvoření samostatného zařízení v podobě dashboardu s jednoduchým ovládním a možností sledování konkrétní veličiny bez nutnosti interakce s uživatelem. Kvůli rozsáhlosti a komplikovanosti systému by bylo určitě vyžadováno prvotní nastavení zařízení uživatelem s mírně pokročilými schopnostmi, nicméně následný běh zařízení by měl být uživatelsky nenáročný. Zařízení by také mělo disponovat různými datovými vizualizačními prostředky pro uspokojení co největšího spektra uživatelů a aplikační vybaveností pro zpracování různých typů datových vstupů, aby bylo schopné pokrýt celý systém firmy Logimic.

## Kapitola 6

# Návrh architektury řešení

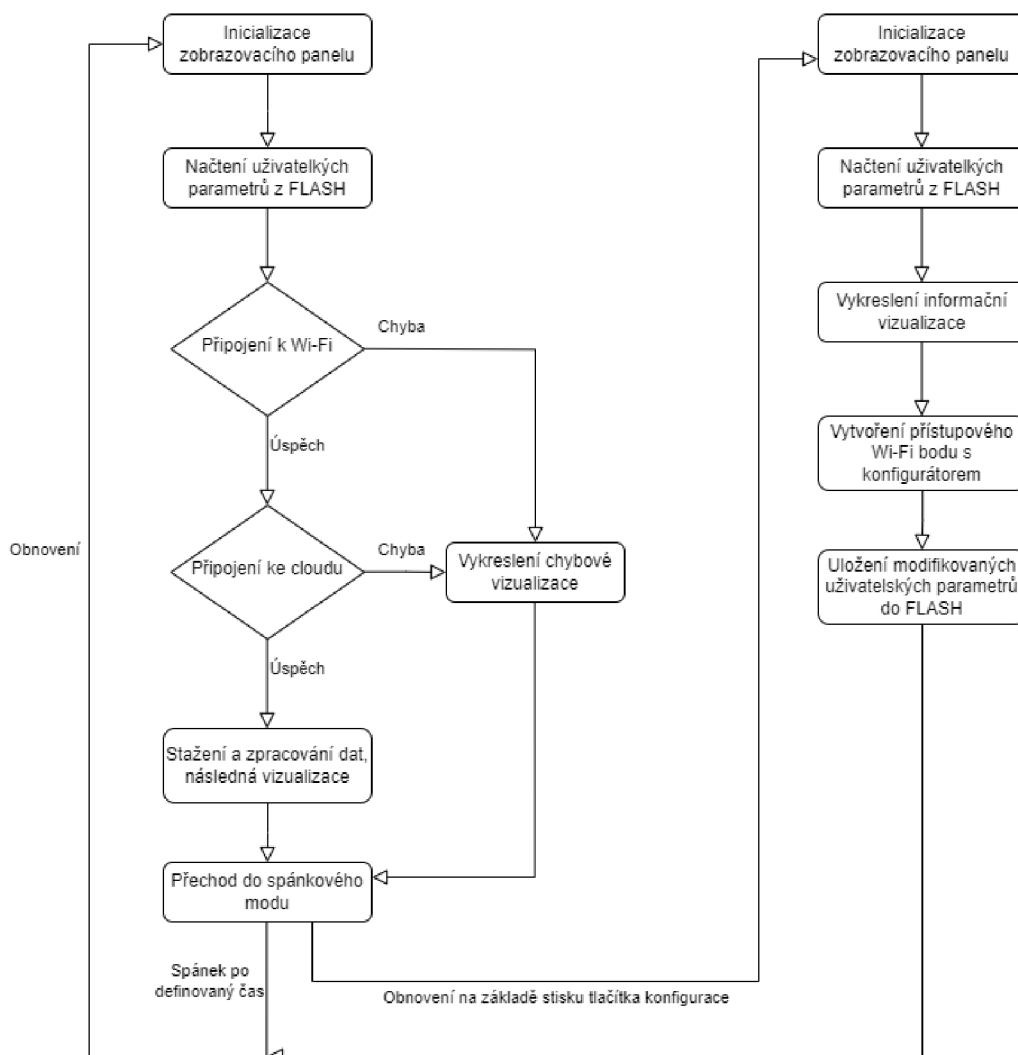
Hlavními parametry, které návrh architektury realizovaného dashboardu musí splňovat, je nízká energetická náročnost běhu s možností provozu z bateriového napájení, přehledná vizualizace dat a schopnost vzdálené komunikace pro příjem dat ze senzorů chytré domácnosti. Zařízení by také mělo být navrženo přehledně a disponovat jednoduchým ovládáním a nastavením, kterému zvládne porozumět i méně zdatný uživatel. Zařízení by také mělo být samostatného běhu podle předem nastavených parametrů bez zásahu uživatele.

Prvním bodem návrhu celé architektury je výběr správného hardwaru pro realizaci. Ten musí splňovat parametry energetické nenáročnosti běhu a možností vzdálené komunikace. Dobrým kandidátem pro realizaci výpočetní stránky architektury může být mikrokontroler ESP32, který dosahuje vysoké energetické efektivity a zároveň disponuje podporou bezdrátové komunikace prostřednictvím technologie Wi-Fi nebo Bluetooth, výhodnou vlastností je provoz v tzv. deepsleep modu, kdy mikrokontroler je uveden do stavu spánku a jeho spotřeba je v tomto módu v řádu  $\mu A$ , následně je možné mikrokontrolér z tohoto stavu probudit definovaným časovým intervalem nebo signálem na příslušném GPIO portu. Ke stahování dat z cloudu firmy Logimic prostřednictvím API je nutná internetová konektivita, proto je technologie Wi-Fi ideálním a zároveň nejrozšířenějším dostupným řešením pro bezdrátovou komunikaci. Další alternativou výběru může být také mikrokontroler ESP8266, taktéž disponuje bezdrátovou konektivitou, nicméně s omezením jen na technologii Wi-Fi. Jedná se avšak o starší architekturu než v případě ESP32, proto nedosahuje takového výkonu ani energetické efektivity jako jeho novější model. Dostupný výpočetní výkon, kterého dosahují mikrokontrolery typu ESP32 bude schopen pokrýt veškeré výkonnostní nároky, které bude implementace vyžadovat, proto zvažování výkonnějších alternativ již z oblasti mikropočítačů popsané kapitolou 2.2 není nutné. Technologií pro zobrazovací panel vyhovující požadavkům implementace, tedy energetické nenáročnosti, splňují panely technologie typu E-ink, jejichž hlavní výhodou je vlastnost zachování obrazu po jeho vykreslení bez potřeby elektrické energie až do doby, než je překreslen jiným obrazem, detailněji technologii a výhody zobrazovacích panelů typu E-ink popisuje šestý odstavec kapitoly 3.2. Alternativním řešením technologie pro zobrazovací panel by mohla být i technologie LCD, opět popsána v kapitole 3.2. Ta však pro zobrazení obsahu musí být připojena permanentně na zdroji elektrické energie a není tedy tak efektivní a energeticky nenáročná. Tento problém by mohl být řešen například pohybovým senzorem, který by zobrazovací panel spustil jen v případě detekce pohybu, nicméně ošetření mylných podnětů, například pohyb domácího mazlíčka po domácnosti by bylo komplikované ošetřit. Spínat zapnutí zobrazovacího panelu by bylo možné alternativně i fyzickým tlačítkem, toto řešení by řešilo problémy pohybového senzoru, přineslo by však potencionálnímu uživateli značný diskomfort.

Při výběru komponent pro tvorbu hardwarové realizace by mělo být pamatováno na případ užití s napájením v podobě baterií, tedy všechny komponenty by této alternativě měly být přizpůsobeny.

Softwarové řešení architektury musí pokrýt požadavky jednoduchosti ovládání a nastavení, zároveň musí být navrženo s ohledem na užitý hardware, aby byl splněn požadavek energetické efektivity a zároveň bylo maximalizováno užití dostupných zdrojů. Výsledné řešení by také mělo být schopno uspokojení co možná největšího spektra uživatelů, měla by tedy být pro uživatele implementovaná dostatečná sada funkcí a datových vizualizací, aby se dalo zařízení personalizovat dle potřeb uživatele. Architektura zpracování a zobrazení dat, která bude zařízení z cloudu firmy Logimic získávat, musí být dostatečně univerzální, aby byla schopna pokrýt různorodé potřeby jednotlivých částí systému - struktura, množství a rozsahy dat.

Obrázek 6.1 popisuje návrh architektury systému. V prvním kroku po spuštění zařízení je inicializován zobrazovací panel zařízení, následně dochází k načtení dat z nevolatilní flash paměti, která uchovává uživatelské nastavení a parametry (přístupové údaje k Wi-Fi, cloudu, časový interval spánku a další). Na základě dat z paměti se pokusí zařízení připojit k Wi-Fi, pokud se připojení nezdaří, je uživatel informován chybovou vizualizací, pokud se zdaří, následuje pokus o připojení ke cloudu. Zde je opět v případě neúspěchu uživatel informován, jinak dochází ke stažení a zpracování dat z cloudu a jejich následná vizualizace dle specifik uživatele. Po úspěšném vykreslení vizualizace přechází zařízení automaticky do režimu spánku pro minimalizaci energetické spotřeby. V tomto režimu setrvává po uživatelem definovaný čas. Po jehož uplynutí je zařízení automaticky probuzeno a již zmíněný proces se opakuje. V průběhu spánku může být také zařízení probuzeno uživatelem formou stisknutí aktivačního tlačítka. Po jeho stisknutí přechází zařízení do stavu přístupového Wi-Fi bodu, na který se uživatel dle pokynů zobrazených na zobrazovacím panelu může připojit. Po připojení je uživateli nabídnut webový konfigurator celého zařízení, zde si může uživatel definovat parametry přístupu k Wi-Fi, cloudu, časový interval spánku a zároveň data, která mají být na zobrazovacím panelu vizualizována a v jaké podobě. Po potvrzení webového konfiguratoru uživatelem dochází k přepsání parametrů v nevolatilní flash paměti novými hodnotami a zařízení přechází zpět do normálního operačního cyklu popsaneho na začátku tohoto odstavce.

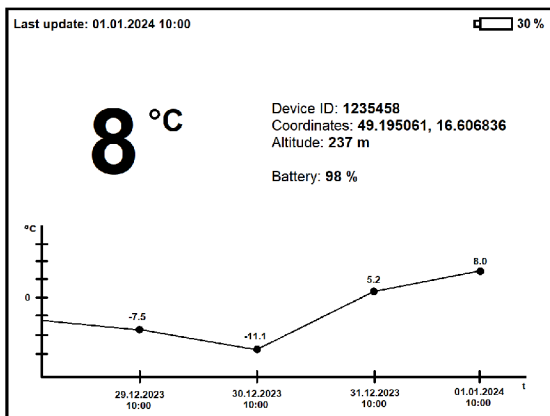


Obrázek 6.1: Návrh vývojového diagramu architektury

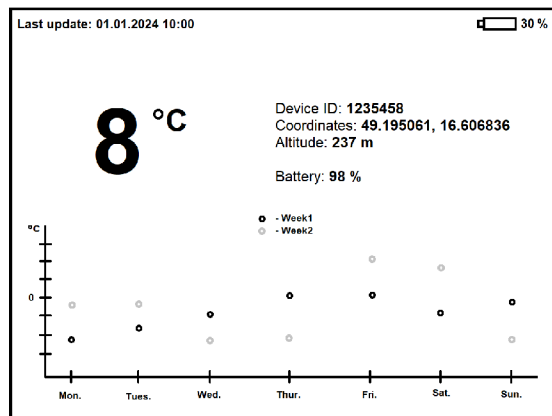
## 6.1 Návrhy vizualizací

Z návrhu architektury je patrné, že zařízení bude zaměřeno primárně na statické vizualizace, které se budou měnit/aktualizovat jednou za definovaný čas, proto je při návrhu nutné s touto vlastností počítat.

První sada návrhů vizualizací viz 6.2 zobrazuje v horní části nejzákladnější informace, čas poslední aktualizace zařízení a zbývající kapacita baterie, kterou zařízení ještě disponuje. Centrální část obou vizualizací je shodná, v levé části centrální sekce je zobrazena poslední hodnota sledované veličiny, kterou v době poslední aktualizace zařízení z cloudu získalo, v pravé části centrální sekce je výpis informací o zařízení, ze kterého data pocházejí. Spodní část vizualizace je u obou návrhů rozdílná, v případě levého návrhu jsou data jedné sledované veličiny daného zařízení vizualizována ve formě řádkového grafu. Pravý návrh ukazuje možnost vizualizace dat z daného zařízení pomocí rozptylového grafu, v tomto případě však není sledována pouze jedna veličina daného zařízení, ale rovnou dvě. Vlastnosti a schopnosti obou typů datových vizualizací jsou popsány více v kapitole 3.1.



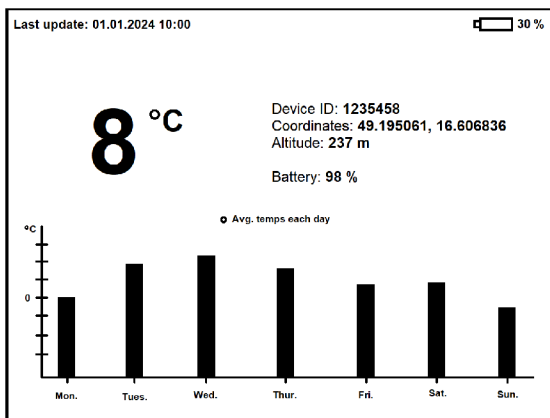
(a) Využití řádkového grafu



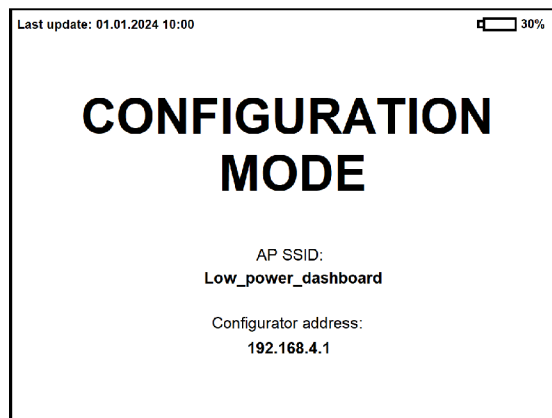
(b) Využití rozptylového grafu

Obrázek 6.2: Příklady vizualizací

V druhé sadě návrhů vizualizací viz 6.3 jsou dva návrhy s odlišným účelem, levý návrh sdílí vrchní a středovou část s návrhy v sadě 6.2 a jedná se o další návrh datové vizualizace, která v tomto případě pro vizualizaci dat jedné veličiny z daného zařízení využívá sloupcového grafu, vlastnostmi a schopnostmi této datové vizualizace se zabývá kapitola 3.1. Pravý návrh ukazuje vizualizaci informační obrazovky, v tomto konkrétním případě se jedná o návrh informační vizualizace pro Wi-Fi přístupový bod, vizualizace se snaží poskytnout informace uživateli o vytvořeném přístupovém bodu.



(a) Využití sloupcového grafu



(b) Informační obrazovka

Obrázek 6.3: Další příklady vizualizací



## Kapitola 7

# Implementace architektury řešení

Na základě návrhu architektury řešení došlo pro realizaci výpočetní a výkonnostní stránky k volbě mikrokontroleru ESP32-WROOM-32E<sup>1</sup>, dále jen ESP32, od společnosti ESPRESS-SIF s dvoujádrovým 32bitovým procesorem Xtensa LX6, integrovaným Wi-Fi a Bluetooth modulem pro bezdrátové připojení a řadou periférií mezi které spadá I2C, SPI nebo UART. Hlavním důvodem této volby je design mikrokontroleru pro energetickou účinnost a nízkou spotřebu vhodnou i pro napájení z baterií.

Jako zobrazovací panel pro zobrazování vizualizací byl zvolen 4,3 palcový černobílý UART E-ink zobrazovací panel<sup>2</sup> od společnosti Waveshare. Panel disponuje vysokým rozlišením 800x600 bodů, vestavěným kontrolérem, UART rozhraním pro komunikaci s mikrokontrolerem nebo mikropočítačem. Společnost Waveshare k panelu poskytuje vykreslovací knihovnu EPD<sup>3</sup>, která mimo vykreslování základních geometrických tvarů v souřadnicích umožňuje i správu panelu, tedy jeho inicializaci, nastavení čtyř stupňů šedi a uvedení panelu do režimu spánku, konkrétně kontroléru, který panel ovládá. Panel nedisponuje možností částečného překreslení, je tedy nutné jej překreslit vždy celý, překreslení celého panelu trvá přibližně 2 sekundy.

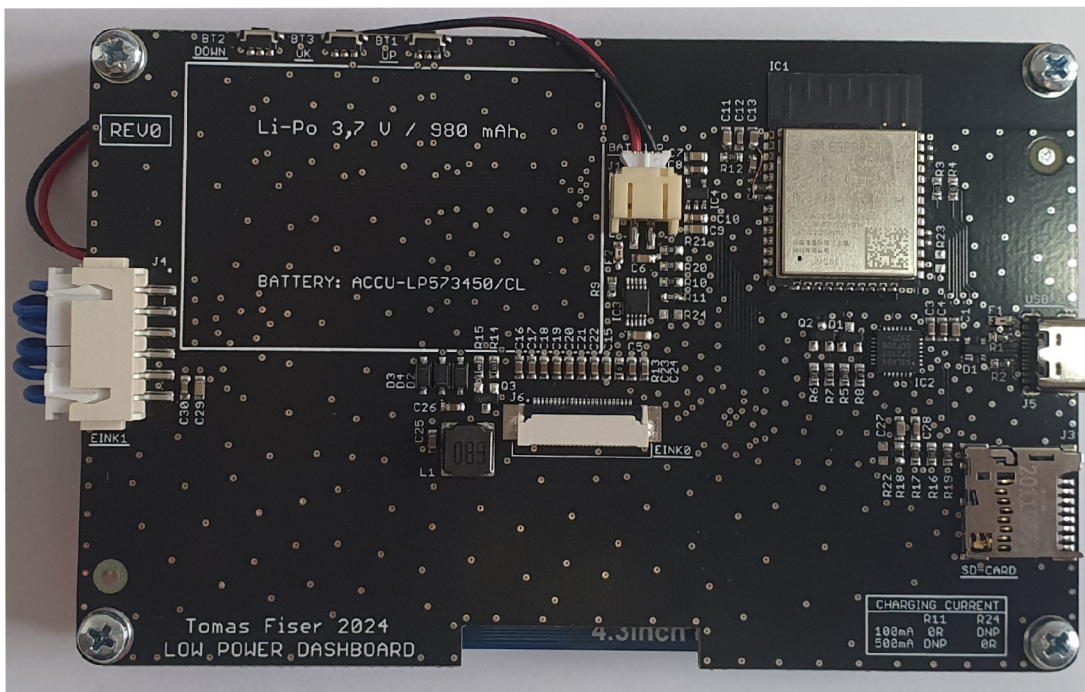
### 7.1 Hardwarová architektura implementace

Na základě zvoleného hardwaru proběhl návrh a výroba desky viz obrázek 7.1. Deska je osazena zmíněným mikrokontrolerem ESP32, akumulátorem pro napájení zařízení, USB-C konektorem pro nabíjení akumulátoru a nahrávání firmwaru zařízení. Na desce je také implementován konektor pro připojení E-ink panelu (bílý konektor na levé hraně desky), zároveň deska disponuje i druhým konektorem pro připojení E-ink panelu (bílý konektor ve středu desky), tento konektor však slouží pro užití E-ink panelu bez vlastního kontroléru, který v této implementaci není využit. V levé části horního okraje jsou implementovány tři tlačítka pro ovládání zařízení. Prostřední tlačítko disponuje funkcí probuzení zařízení z tzv. deepsleep modu. Na desce je pro budoucí rozšíření také implementován slot na microSD kartu, například pro potřeby logování chyb nebo ukládání dat offline. Deska prošla za dobu implementace také vlastním vývojem, první navržená a vyrobená verze obsahoval několik chyb, bylo nutné ji tedy nechat drobně upravit do aktuální podoby druhé verze, kterou lze již považovat za plně funkční prototyp. Detailní strukturu desky zobrazuje obrázek 7.2.

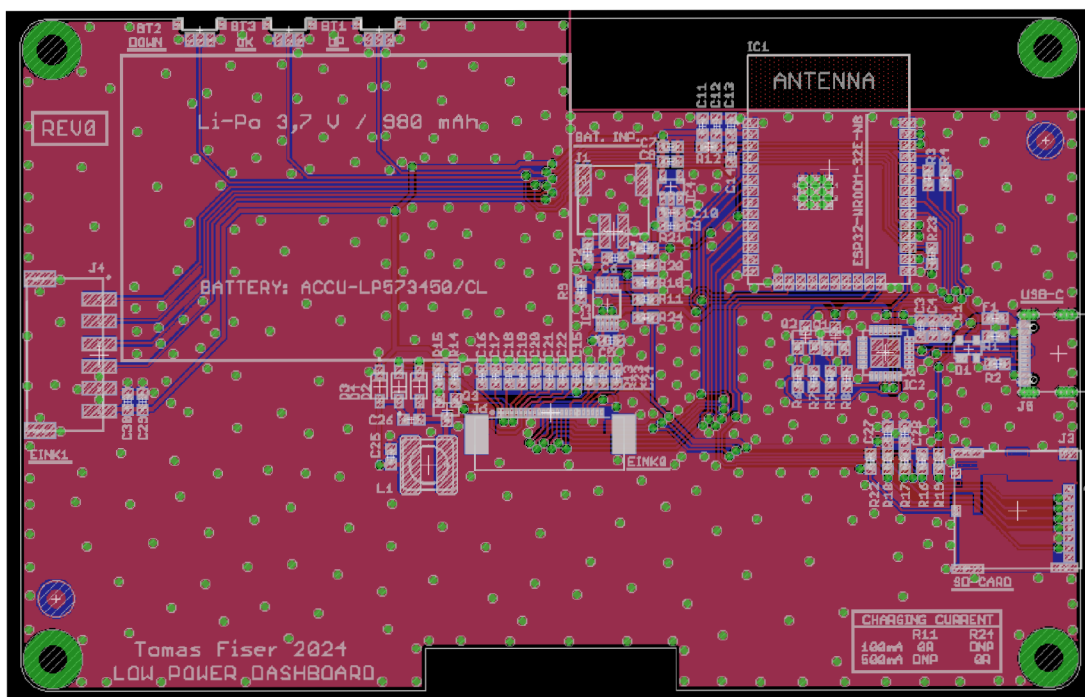
<sup>1</sup>Dokumentace mikrokontroleru: [https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32e\\_esp32-wroom-32ue\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32e_esp32-wroom-32ue_datasheet_en.pdf)

<sup>2</sup>Zvolený panel: [https://www.waveshare.com/wiki/4.3inch\\_e-Paper\\_UART\\_Module](https://www.waveshare.com/wiki/4.3inch_e-Paper_UART_Module)

<sup>3</sup>Vykreslovací knihovna EPD: <https://github.com/soonuse/epd-library-arduino>



Obrázek 7.1: Zadní strana hardwarového řešení

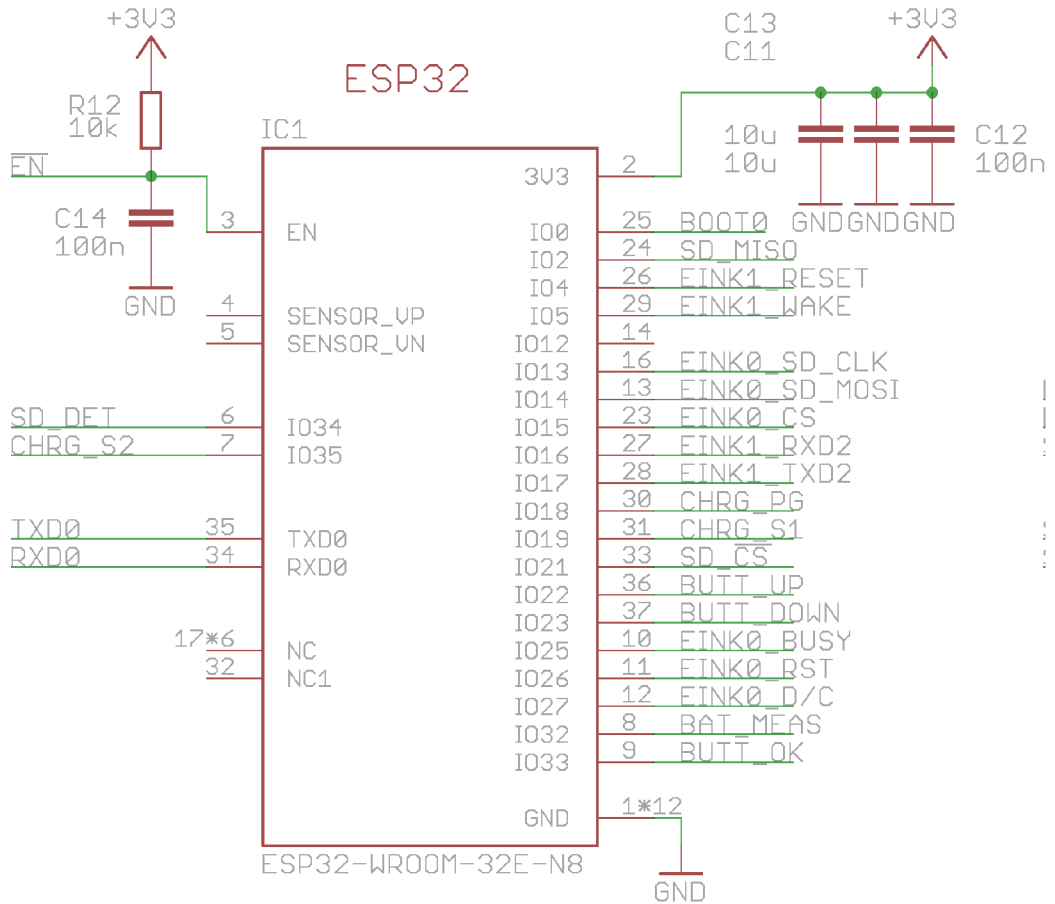


Obrázek 7.2: Schéma zapojení - zadní strana

Na obrázku 7.3 je detail zapojení GPIO pinů mikrokontroleru na desce. GPIO piny 22, 23 řeší obsluhu krajních tlačítek, pin 33 zajišťuje obsluhu prostředního "OK" tlačítka se schopností buzení zařízení z tzv. deepsleep modu. Piny 4, 5, 16 a 17 se starají o obsluhu



konektoru pro připojení E-ink panelu, který je v této implementaci využit, tedy panel s vlastním kontrolérem.



Obrázek 7.3: Zapojení mikrokontroleru

## 7.2 Programová architektura implementace

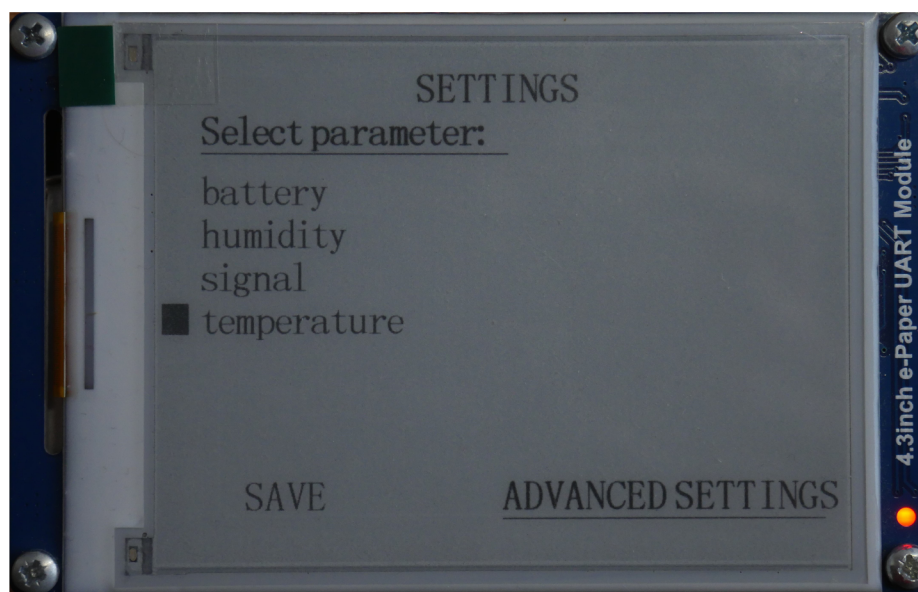
Softwarové řešení projektu je implementováno pomocí programovacího jazyka C. Dashboard je navržen pro sledování jednoho zařízení ze systému firmy Logimic 5.1, které si uživatel zvolí.

Po spuštění dashboardu začíná běh v souboru `low-power-dashboard.c`, kde dochází k inicializaci GPIO pinů společně s inicializací E-ink panelu. Následuje inicializace C struktury uložené v nevolatilní paměti mikrokontroleru (NV storage), tato struktura uchovává data o konfiguraci zařízení a hodnoty, které je pro synchronizaci mezi jednotlivými spacími cykly nutné uchovat, skládá se z přístupových údajů k Wi-fi a cloudu, času spánku, ID systému, se kterým zařízení na cloudové straně pracuje, dále z informací o sledovaném zařízení, zvolené vizualizaci a časové periody, ze které mají být data vizualizována. V neposlední řadě struktura ukládá i čas poslední aktualizace dashboardu. Celá struktura je definována v hla-

vičkovém souboru `NV_storage.h`. Inicializaci, čtení, zápis a definování výchozích hodnot v případě výskytu chyby zajišťují funkce implementované v souboru `NV_storage.c`. Po inicializaci paměti přechází program do souboru `sleep.c`, ve kterém na základě formy probuzení (uplynutí časového intervalu spánku nebo stisknutím tlačítka) je rozhodnuto o dalším postupu.

V případě, že se jedná o automatické zapnutí zařízení vlivem uplynutí uživatelem nastaveného času spánku, dochází k zapnutí Wi-Fi v modu station, tento mod realizuje soubor `wifi_sta.c`. Zařízení se pokusí připojit k internetu, pokud neuspěje, je zobrazena chybová vizualizace a zařízení přechází do režimu spánku. Pokud je však výsledkem úspěch, zařízení stáhne aktuální čas z internetu (hardwarová implementace neobsahuje RTC modul) pomocí SNTP (Simple Network Time Protocol) implementovaného v souboru `get_sntp.c`, následuje pokus o připojení ke cloudu firmy Logimic, neúspěch je doprovázen identickým chováním jako v případě station modu Wi-Fi. Po úspěšném připojení ke cloudu stáhne dashboard příslušná data dle specifikace, veškerá data stahuje pomocí analytického endpointu firmy, detailně je endpoint popsán v kapitole 5.1. Data jsou následně předána parseru implementovaného v souboru `data_processing.c`, kde jsou vyparsované hodnoty uloženy do dynamicky alokované struktury. V posledním kroku je zavolána příslušná vizualizační funkce ze souboru `visual.c`, kde jsou data ze struktur zkontrolována a zpracována do polí, která vizualizační funkce očekává, následuje vykreslení vizualizace a dashboard přechází do modu spánku na uživatelem specifikovaný čas.

Druhou možností probuzení je případ interakce uživatele s dashboardem, tedy stisknutím prostředního tlačítka uživatelem. Uživateli je následně na panelu vykresleno dynamické menu 7.4, ve kterém si může vybrat parametr nebo parametry, které mají být ze sledovaného zařízení vizualizovány, implementované vizualizace jsou navrženy na zobrazování jednoho (řádkový graf) nebo dvou (rozptylový graf) parametrů (teplota, vlhkost, napětí baterie a další).



Obrázek 7.4: Menu pro výběr parametru

V menu, které je implementováno v souboru `menu.c`, se uživatel pohybuje pomocí zbylých dvou tlačítek na stranách, prostřední tlačítko slouží k potvrzení výběru. Z menu uží-

vatel může odejít dvěma způsoby, prvním je výběr parametru a následné uložení výběru, v tomto případě dochází k restartu dashboardu, který je následně zapnut a chová se jako v případě probuzení po uplynutí časovače. Druhou možností je přechod do pokročilého konfiguratoru - Advanced settings. Po zvolení této možnosti je dashboard přepnut do Wi-Fi station modu a pokouší se připojit k internetu, pokud se mu to zdaří, následuje pokus o připojení ke cloudu. Dashboard tentokrát nepřistupuje k analytickému endpointu, ale k endpointu pro získání všech skupin v systému firmy Logimic a zároveň všech zařízení v nich, oba potřebné endpointy popisuje kapitola 5.1. Uživatel je o chování dashboardu informován prostřednictvím informační vizualizace. Po stažení všech dat jsou data předána parseru, který je uloží do dynamicky generovaných struktur. Následně je ukončen Wi-Fi station mod a dashboard se přepíná do modu přístupového Wi-Fi bodu. O vytvoření přístupového bodu a vygenerování dynamicky generované webové stránky se stará implementace ze souboru `wifi_ap.c`. Uživatel je o úspěšném vytvoření přístupového bodu informován vizualizací na panelu s pokyny pro připojení. Po připojení může uživatel přejít do dynamicky generovaného webového konfiguratoru viz obrázek 7.5.

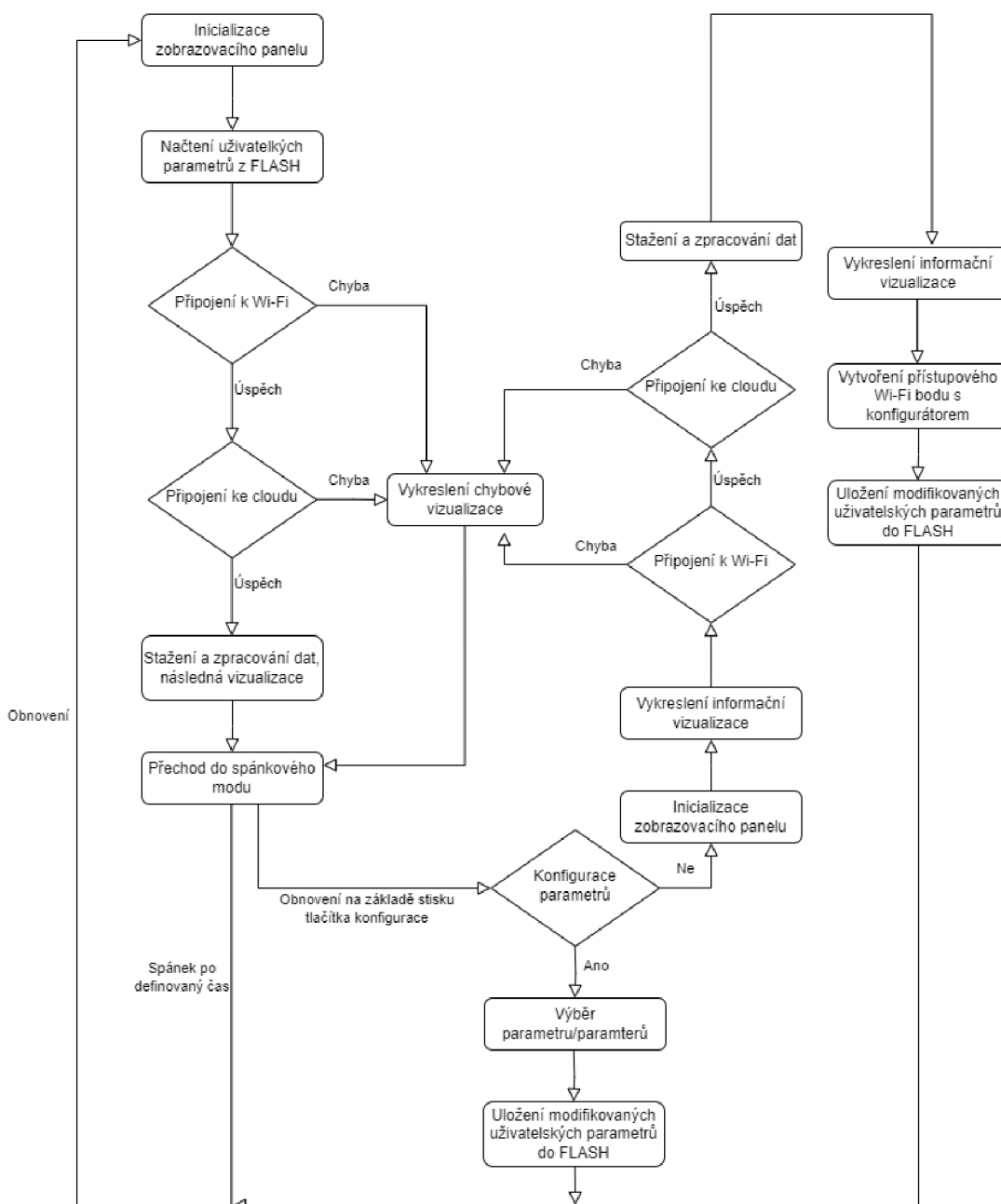
The image shows a web browser window with the address bar displaying '192.168.4.1'. The main content area has a dark purple header with the text 'LOW POWER DASHBOARD CONFIGURATION WEB PAGE'. Below the header, there are several white panels with rounded corners, each containing a different category of settings:

- WIFI SETTINGS:** Includes fields for 'Network SSID' (value: COM-FI-n), 'Network password' (value: thomson), and 'Grant type' (value: client\_cred).
- CLOUD SETTINGS:** Includes fields for 'Client ID' (value: 7q7e084r), 'Client password' (value: qs77i0qs), 'Auth scope' (value: scope), 'Auth URL' (value: uri), 'Cloud URL' (value: clouUri), and 'System ID' (value: 129).
- DISPLAY SETTINGS:** Includes a 'Refresh time' field with the value 1800.
- VISUALISATION SETTINGS:** Includes a 'Choose visual' dropdown menu (value: Table) and a 'Time period' dropdown menu (value: Day).
- CHOOSE SENSORS:** Lists three groups of sensors:
  - Group Skleniky:** Device Greenhouse (radio button).
  - Group Sklep:** Device Cellar (radio button).
  - Group Goodwe:** Device Goodwe (radio button).

Obrázek 7.5: Webový konfigurator

Zde si může nastavit veškeré parametry potřebné pro běh dashboardu, přístupové údaje k Wi-Fi a cloudu, čas spánku, výběr druhu vizualizace společně časovým oknem, ze kterého mají vizualizovaná data pocházet. V závěrečné části konfiguratoru si uživatel může vybrat zařízení, které má dashboard sledovat. Po potvrzení formuláře na webu jsou zadané údaje zkontrolovány a uloženy do nevolatilní paměti, následuje restart dashboardu. Po restartu se dashboard chová jako v případě probuzení po uplynutí časovače.

Celou programovou implementací graficky shrnuje obrázek 7.6.



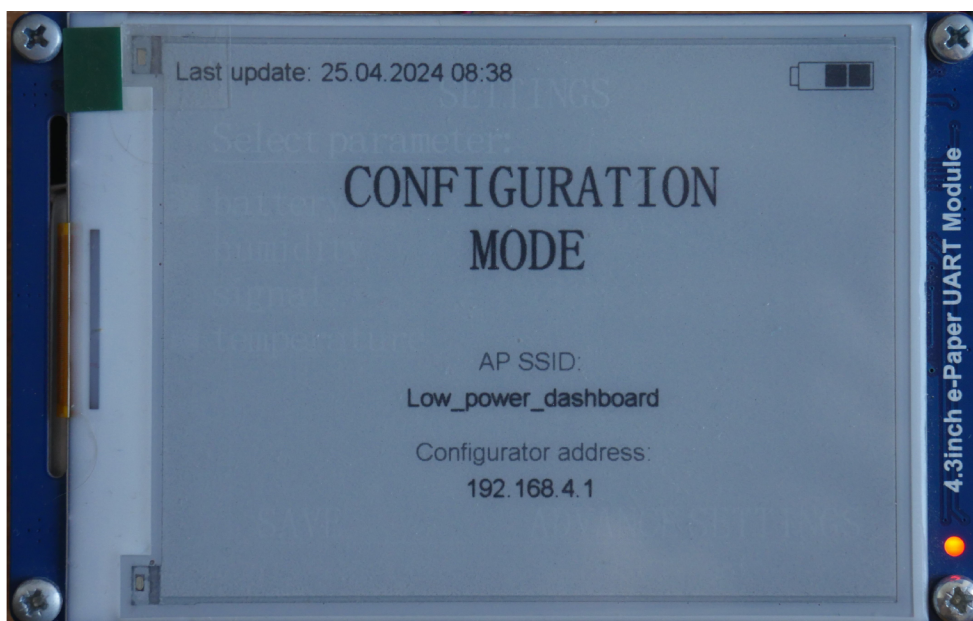
Obrázek 7.6: Vývojový diagram implementované architektury

### 7.3 Implementované vizualizace

Všechny vizualizace v implementaci jsou vytvořeny prostřednictvím knihovny EPD poskytované výrobcem E-ink panelu. Knihovna, jak bylo již zmíněno, poskytuje vývojáři jen vykreslení základních geometrických tvarů v souřadnicích a dvě znakové sady ve třech velikostech fontu. Vykreslování je založeno na principu ukládání jednotlivých částí vizualizace do vyrovnávací paměti, která je pak vestavěnou funkcí na panel vykreslena. Díky nízkourovňovému designu knihovny EPD bylo nutné vytvořit vlastní nadstavbovou knihovnu

grafových vizualizací. Všechny vytvořené vizualizace jsou na základě datového vstupu dynamicky generované, tedy mají schopnost automatického přizpůsobení. Vizualizace lze také jednoduše přizpůsobit rozměrově jinému panelu od firmy Waveshare s podporou knihovny EPD. Stačí do dashboardu nainstalovat ovladač pro daný panel a vizualizacím nastavit nové rozlišení panelu. Vytvořená vizualizační knihovna postavená na knihovně EPD je implementována v souboru `visual.c`.

Prvním příkladem implementované vizualizace, obrázek 7.7, je statická informační vizualizace informující uživatele o přístupových parametrech pro připojení na vytvořený přístupový Wi-Fi bod, který obsahuje webový konfigurátor. Vizualizace vychází z návrhu vizualizací viz 6.3.

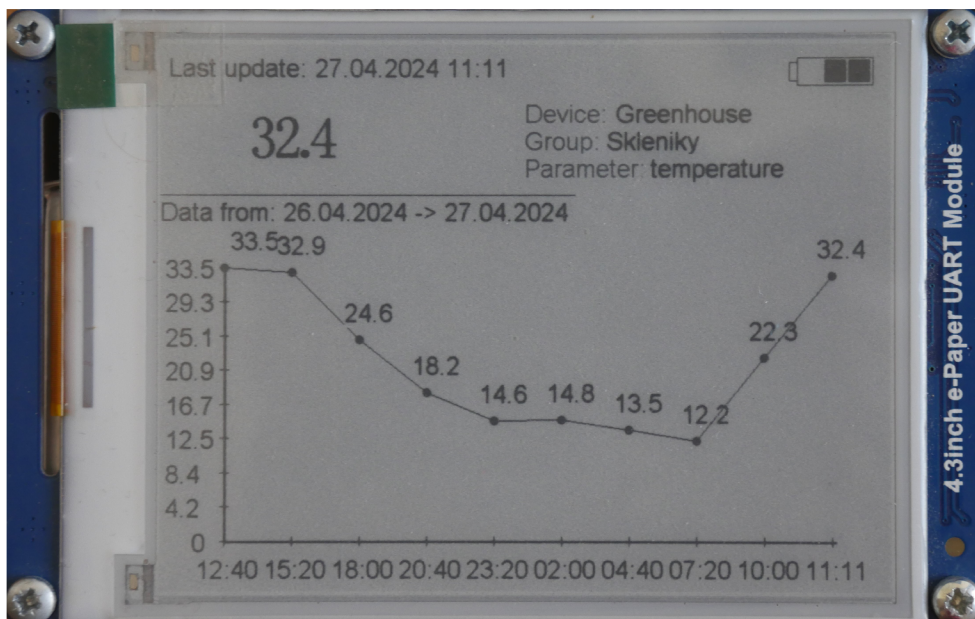


Obrázek 7.7: Implementovaná informační vizualizace

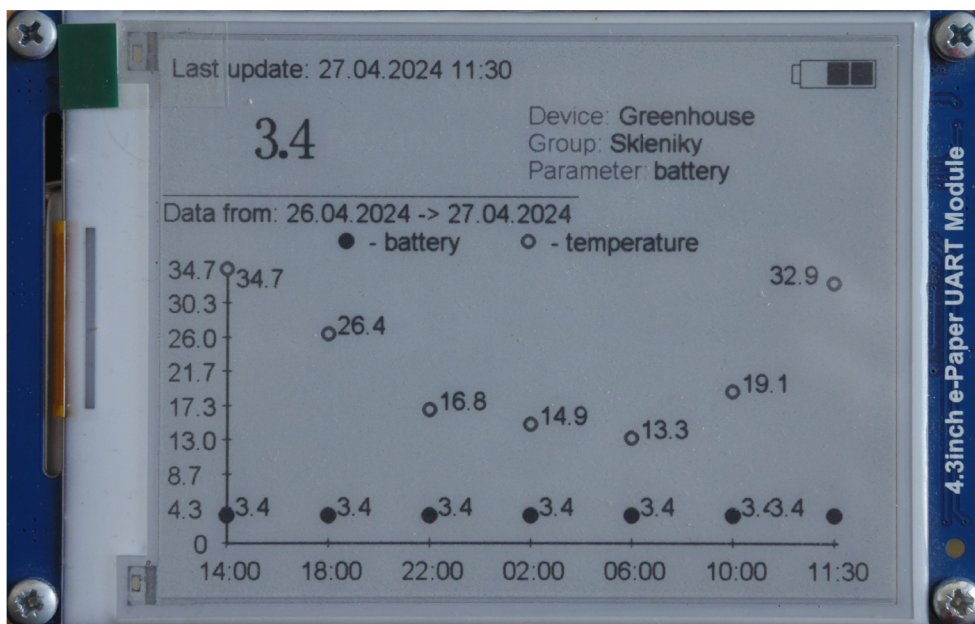
Vizualizace viz obrázky 7.8 a 7.9 ukazují již komplexnější, dynamicky generované vizualizace. Obě implementace vycházejí z prvotních návrhů viz 6.2 s rozdílem zobrazování stavu baterie pomocí piktogramu namísto procent. Na horní hraně vizualizací se nachází údaje informující uživatele o čase poslední aktualizace dashboardu, v pravém rohu je následně zobrazen piktogram s informací o zbývajícím množství energie v baterii dashboardu. Ve středové části vizualizace jsou zobrazeny informace o sledovaném zařízení, na levé straně je poslední hodnota sledovaného parametru daného zařízení, na pravé straně jsou zobrazeny informace, o jaké zařízení se jedná, do které skupiny zařízení patří a název parametru daného zařízení, který je sledován. Ve spodní části obou implementací se nachází samotné dynamicky generované grafy. Na základě vstupních dat si grafové vizualizace automaticky spočítají stupnice na ose Y, do vyrovnávací paměti se následně obě osy (X i Y) společně s příslušnými hodnotami zapíší. Výpočet stupnice je založen na principu hledání nejmenší a největší hodnoty v datech a definované ploše, kterou graf může využít. V posledním kroku vykreslení jsou procházena veškerá data a na základě hodnoty a velikosti stupnice jsou vykreslena na příslušné místo, v případě řádkového grafu na obrázku 7.8 jsou navíc ještě spojeny s předchozí hodnotou. V návrhu vizualizací byl vytvořen i návrh sloupcového grafu viz 6.3. Tento typ datové vizualizace byl i implementován, nicméně vstupní data od firmy



Logimic neobsahuje vhodný pattern pro účinné seskupování dat, proto tento typ datové vizualizace ve finálním řešení nebyl využit.



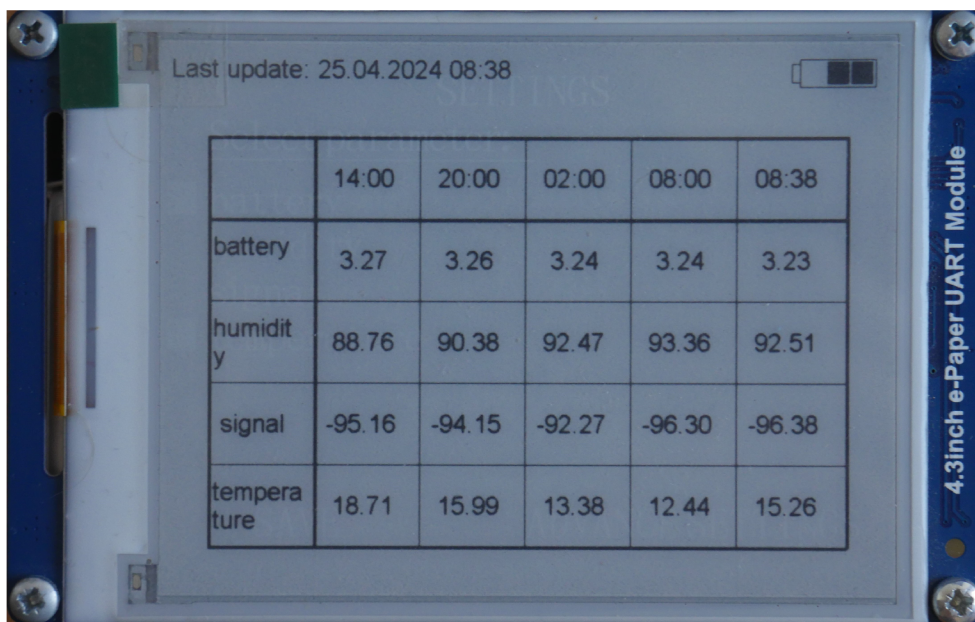
Obrázek 7.8: Implementovaný řádkový graf



Obrázek 7.9: Implementovaný rozptylový graf

Posledním příkladem implementované datové vizualizace je vizualizace prostřednictvím tabulky viz obrázek 7.10. Tabulka zobrazuje veškeré dostupné parametry daného zařízení v daném časovém okně. Celá tabulka je na základě vstupních dat dynamicky generována, zároveň je implementováno hlídání přetékání buněk, pokud se řetězec v poli tabulky nevejde na jeden řádek, je automaticky zalomen na další řádek, pokud to výška buňky dovoluje.

V případě, že se další řádek do buňky již nevejde, je řetězec automaticky o zbylé znaky oříznut.



	14:00	20:00	02:00	08:00	08:38
battery	3.27	3.26	3.24	3.24	3.23
humidity	88.76	90.38	92.47	93.36	92.51
signal	-95.16	-94.15	-92.27	-96.30	-96.38
temperature	18.71	15.99	13.38	12.44	15.26

Obrázek 7.10: Implementovaná tabulka



## Kapitola 8

# Testování, spotřeba a použitelnost

Dashboardu vznikly po dokončení implementace dva funkční prototypy. V obou případech proběhlo testování s více uživateli v řádu několika dní. Oba prototypy po celou dobu testování fungovaly stabilně bez větších poruch, jediným zásadnějším problémem, který se objevil byla špatná implementace SNTP klienta stahující aktuální čas z internetu. V dlouhodobějším testování začalo docházet k chybám synchronizace a získávaný čas byl chybný. Znalost aktuálního času je nezbytný parametr pro stahování dat z cloudu firmy Logimic, proto bylo nezbytné provést úpravu implementace klienta a problém odstranit. Dalším úskalím, které testování odhalilo je množství dostupné operační paměti, kterou hardwarová implementace poskytuje. Pro systém firmy Logimic, pro který je primárně implementace vytvořena, je dostatečná, nicméně v případě využití dashboardu pro pokrytí systému většího rozsahu už by dostupné zdroje nemusely být dostačující, proto by hardwarová implementace v další verzi měla disponovat větším množstvím operační paměti než současná. V rámci testování se mimo stabilitu běhu testovala také spotřeba dashboardu a použitelnost.

### 8.1 Testování použitelnosti

Testování použitelnosti bylo provedeno s třemi uživateli ve věkovém rozmezí 50 až 70 let. Uživatelé byli na různé úrovni technické zdatnosti pro validaci parametru jednoduchosti ovládání. Všichni testovací uživatelé byli schopni zařízení bez větších obtíží používat, mezi klíčové prvky jejich zpětné vazby patří dobrá čitelnost zařízení - čitelnost textu a dobré rozložení, jednoduché a přehledné ovládání společně s konfigurací zařízení, kterou byli všichni s jednoduchým návodem schopni provést. Částí zpětné vazby byly i nedostatky a případné návrhy na úpravy, které by uživatelé využili. Největším nedostatkem, který uživatelé shledali je slabý výkon a tedy i dosah bezdrátového připojení. Tyto limity vestavěné Wi-Fi antény užitého mikrokontroleru limitovali uživatele v umístění zařízení v rámci domácnosti s ohledem na umístění jejich Wi-Fi přístupového bodu. V další verzi hardwarové implementace bude nutné tento nedostatek odstranit přidáním externí Wi-Fi antény. Mezi úpravy a rozšíření, které uživatelé navrhli patří hlavně sledování více zařízení najednou (aktuálně je podpora jednoho zařízení) a zobrazení více datových vizualizací najednou - obrazovka rozdělená do bloků z více jednodušších vizualizací, které si může uživatel sám vybrat.

Výsledné zařízení bylo také předvedeno v rámci testování použitelnosti zástupci, Ing. Michalu Valnému, Ph.D., z firmy Logimic. Jeho zpětnou vazbou byla pochvala implementace, jak softwarové tak i hardwarové, zejména dobrý výběr typu zobrazovacího panelu do domácnosti, panel nemá vlastní podsvícení, takže i když je zapnut přes noc, tak nepůsobí

jako rušivým elementem. Zároveň řešení ohodnotil jako smysluplné, funkční a s budoucím potenciálem.

## 8.2 Testování spotřeby energie

Následně byla se zařízením provedena sada měření spotřeby elektrické energie pro porovnání s ostatními dostupnými řešeními na trhu a také za účelem zjištění přesných hodnot výdrže dashboardu na jedno nabití baterie. V následující tabulce 8.1 je přehled spotřeby dashboardu v režimu spánku a v zapnutém režimu při napájecím napětí 3,7V.

Režim	s E-ink panelem [mA]	bez E-ink panelu [mA]
Spánek	3,90	0,16
Běh	6,25	2,79

Tabulka 8.1: Spotřeba v jednotlivých režimech

Z tabulky je patrné, že zobrazovací panel je dominantním spotřebičem implementace. I přes uvedení panelu, tedy jeho kontroléru, do stavu spánku je jeho spotřeba mnohonásobně vyšší než samotného ESP32. Tento fakt výrazně omezuje provoz zařízení při napájení čistě z akumulátoru. Řešením této přebytkové spotřeby zobrazovacího panelu v momentě, kdy zařízení spí a není tedy třeba, by bylo užití panelu bez vlastního kontroléru nebo úpravou hardwaru a přidáním elektrického spínače pro napájení panelu. Následující tabulka 8.2 ukazuje, jak vysoká spotřeba kontroléru panelu ve spacím režimu ovlivňuje jeho výdrž na akumulátor. Pozn.: V původním návrhu je dashboard napájen baterií o kapacitě 980 mAh, testovací prototyp byl nakonec osazen akumulátorem o kapacitě 4000 mAh, proto tabulka obsahuje výsledky pro obě kapacity.

Obnovení	E-ink bez spínače [dny]		E-ink se spínačem [dny]	
	980 mAh	4000 mAh	980 mAh	4000mAh
1x za 10 minut	6,41	26,15	14,97	61,10
1x za hodinu	9,47	38,65	69,86	285,13
1x za 24h	10,42	42,55	235,02	959,29

Tabulka 8.2: Výdrž dashboardu na baterii

Výsledky v tabulce ukazují, že dashboard je v aktuální implementaci s menším akumulátorem při obnově jedenkrát za hodinu schopen běhu necelých deset dní, s větším akumulátorem je hodnota výdrže téměř čtyřnásobná, tedy necelých čtyřicet dní. V případě modifikace užívaného hardwaru a přidání spínače pro E-ink panel se výdrž rapidně zvyšuje, v případě menší baterie na necelých sedmdesát dní, v případě větší baterie na více než tři čtvrtě roku bez potřeby nabíjení.

Při porovnání výsledků spotřeby energie implementace s E-ink panelem například s implementací, ve které bude panel realizován technologií LCD, jsou výsledky i bez spínače pro

E-ink dobré, například panel RVT70HSTNWC00<sup>1</sup> technologie TFT LCD dle specifikace výrobce má v režimu spánku spotřebu 10 mA a v aktivním režimu až 90 mA, tedy mnohonásobně více než aktuální implementace i s mikrokontrolerem ESP32 dohromady. Výsledek tohoto porovnání také potvrzuje fakt, že všechny běžně dostupné dashboardy na českém trhu jsou designovány pro trvalé napájení ze sítě viz nabídka zařízení na Alza.cz<sup>2</sup>.

---

<sup>1</sup>Odkaz na LCD panel: <https://riverdi.com/product/high-brightness-ips-display-rvt70hstnwc00-7-inch-projected-capacitive-touch-panel-air-bonding-uxtouch-rgb>

<sup>2</sup>Nabídka dashboardů pro chytrou domácnost: <https://www.alza.cz/centralni-jednotky-pro-chytrou-domacnost/18860123.htm>

## Kapitola 9

# Závěr

Cílem práce byl průzkum rozsáhlé oblasti konceptu IoT, konceptů navázaných (chytrá města, chytré domácnosti), důležitost a význam mikrokontrolerů a mikropočítačů pro IoT, principy a omezení při tvorbě datových vizualizací. Na základě těchto průzkumů a analýz byl vyhotoven přehledový panel neboli dashboard, při jehož návrhu byla hlavním cílem nízká energetická náročnost a energetická efektivita, jednoduché ovládací rozhraní a přehlednost datových vizualizací. Těchto cílů je v práci dosaženo navržením architektury řešení využívající jako výpočetní základ mikrokontroler z platformy Arduino, ESP32-WROOM-32E, a zobrazovací panel typu E-ink. Po úspěšné implementaci návrhu se podařilo vytvořit dashboard schopný stabilního běhu, splňující požadavky energetické efektivity a nenáročnosti při porovnání s tradičními řešeními. Zároveň implementace také splňuje požadavky jednoduchého ovládání a přehledných vizualizací.

Na základě uživatelského testování vzniklo několik návrhů a nápadů, jak zařízení vylepšit a přiblížit z prototypu k produkčnímu řešení. Mezi vylepšení patří větší možnosti uživatelské personalizace, přidání několika důležitých indikátorů, například signál Wi-Fi, který aktuální implementace postrádá, schopnosti sledování více zařízení najednou a porovnávání parametrů z více zařízení. Důležitým vylepšením pro další verzi je také přidání spínače pro E-ink zobrazovací panel pro snížení energetické náročnosti, dále implementace externí Wi-Fi antény a zvýšení množství operační paměti, aby dashboard v případě užití mimo systém Logimic nenarazil na hardwarový strop.

# Literatura

- [1] ADITYA, T., SRINIVAS, T. A., SRAVANTHI, Y., KUMAR, Y. a SRIHITH, I. From Charts to Dashboards: An Overview of Data Visualization Methods. *Advancement of Computer Technology and its Applications*. Listopad 2023, sv. 7, č. 1, s. 1–14. DOI: 10.5281/zenodo.10060647.
- [2] AHMAD, A., SAYEED, M. S., TAN, C., TAN, K., BARI, M. A. et al. A Review on IoT with Big Data Analytics. In: IEEE. *2021 9th International Conference on Information and Communication Technology (ICoICT)*. Srpen 2021, s. 160–164. DOI: 10.1109/ICoICT52021.2021.9527503. ISBN 978-1-6654-0447-1.
- [3] AL SARAWI, S., ANBAR, M., ALIEYAN, K. a ALZUBAIDI, M. Internet of Things (IoT) communication protocols: Review. In: IEEE. *2017 8th International Conference on Information Technology (ICIT)*. 2017, s. 685–690. DOI: 10.1109/ICITECH.2017.8079928. ISBN 978-1-5090-6332-1.
- [4] ALEXANDRU, A., ALEXANDRU, C., COARDOS, D. a TUDORA, E. Big data: Concepts, Technologies and Applications in the Public Sector. *International Journal of Computer, Electrical, Automation, Control and Information Engineering*. Říjen 2016, sv. 10, č. 10, s. 1629–1635.
- [5] AMINE, R. Smart Cities: Development and Benefits. In: *Smart Cities: Social and Environmental Challenges and Opportunities for Local Authorities*. Springer International Publishing Cham, říjen 2023, s. 45–53. DOI: 10.1007/978-3-031-35664-3\_4. ISBN 978-3-031-35663-6.
- [6] AWASTHY, N. a VALIVARTHI, N. Evolution of Hadoop and Big Data Trends in Smart World. In: *Sustainable Computing: Transforming Industry 4.0 to Society 5.0*. Springer, Leden 2023, s. 99–127. DOI: 10.1007/978-3-031-13577-4\_6. ISBN 9783031135774.
- [7] BABALOLA, A. a UBOCHI. The Performance of the STM32 Microcontroller and MAX30102 for Remote Health Monitoring Device Design. *Journal of Digital Innovations & Contemporary Research in Science, Engineering & Technology*. Zář 2022, sv. 10, č. 3, s. 51–55. DOI: 10.22624/AIMS/DIGITAL/V10N1P4.
- [8] BACH, B., FREEMAN, E., ABDUL RAHMAN, A., TURKAY, C., KHAN, S. et al. Dashboard Design Patterns. *IEEE Transactions on Visualization and Computer Graphics*. Zář 2022, sv. 29, č. 1, s. 342–352. DOI: 10.1109/TVCG.2022.3209448.
- [9] BAKKER, A. a HUIJSING, J. H. *High-accuracy CMOS smart temperature sensors*. 1. vyd. Springer Science & Business Media, 2000. ISBN 9781475731903.



- [10] BAO, F. a CHEN, J. Visual framework for big data in d3.js. In: IEEE. *2014 Ieee Workshop on Electronics, Computer and Applications*. Květen 2014, s. 47–50. DOI: 10.1109/IWECA.2014.6845553. ISBN 978-1-4799-4565-8.
- [11] BERTE, D.-R. Defining the IoT. *Proceedings of the International Conference on Business Excellence*. Květen 2018, sv. 12, č. 1, s. 118–128. DOI: 10.2478/picbe-2018-0013.
- [12] BESANÇON, L. *An interaction Continuum for 3D Dataset Visualization*. 2017. Disertační práce. Université Paris Saclay (COmUE).
- [13] BHATTA, G. E-INK (ELECTRONIC INK). *International Journal of Computer Science and Engineering*. 2017, sv. 6, č. 1, s. 55–66.
- [14] BIEHL, M. *API Architecture*. API-University Press, 2015. ISBN 9781508676645.
- [15] BOHARA, B., MAHARJAN, S. a SHRESTHA, B. R. IoT based smart home using Blynk framework. *ArXiv preprint arXiv:2007.13714*. 2020.
- [16] CHANG, F. a ÖSTLUND, B. Perspectives of older adults and informal caregivers on information visualization for smart home monitoring systems: A critical review. In: Springer. *Proceedings of the 20th Congress of the International Ergonomics Association (IEA 2018) Volume V: Human Simulation and Virtual Environments, Work With Computing Systems (WWCS), Process Control 20*. 2019, s. 681–690. DOI: 10.1007/978-3-319-96077-7\_74. ISBN 978-3-319-96076-0.
- [17] CHEN, C. a XING, Z. SimilarTech: automatically recommend analogical libraries across different programming languages. In: *2016 31st IEEE/ACM International Conference on Automated Software Engineering (ASE)*. Srpen 2016, s. 834–839. DOI: 10.1145/2970276.2970290. ISBN 978-1-4503-3845-5.
- [18] CHEN, F., ZHANG, X., KUANG, T., LI, M., HU, J. et al. Synthetic Melanin E-Ink. *ACS Applied Materials & Interfaces*. ACS Publications. Duben 2017, sv. 9, č. 19, s. 16553–16560. DOI: 10.1021/acsami.7b03890.
- [19] CREPALDI, P. a PIMENTA, T. Introductory Chapter: RFID: A Successful History. In: *Radio Frequency Identification*. IntechOpen, Listopad 2017. DOI: 10.5772/intechopen.69602. ISBN 978-953-51-3629-3.
- [20] CUI, L., XIE, g., QU, Y., GAO, L. a YANG, y. Security and Privacy in Smart Cities: Challenges and Opportunities. *IEEE Access*. Červenec 2018, PP, s. 1. DOI: 10.1109/ACCESS.2018.2853985.
- [21] DING, J., NEMATİ, M., RANAWEERA, C. a CHOI, J. IoT Connectivity Technologies and Applications: A Survey. *IEEE Access*. Duben 2020, PP, s. 1–7. DOI: 10.1109/ACCESS.2020.2985932.
- [22] FEZARI, M. a AL DAHOUD, A. Integrated Development Environment IDE For Arduino. *WSN applications*. ResearchGate. Říjen 2018, sv. 11, s. 1–2.
- [23] GOVINDRAJ, V., SATHIYANARAYANAN, M. a ABUBAKAR, B. Customary homes to smart homes using Internet of Things (IoT) and mobile application. In: IEEE. *2017 International Conference On Smart Technologies For Smart Nation (SmartTechCon)*. 2017, s. 1059–1063. ISBN 978-1-5386-0569-1.

- [24] GURSKI, J. a QUACH, L. M. Display technology overview. *Lytica White Paper*. Citeseer. 2005, sv. 37.
- [25] GUSNADI, Y. a HERMAWAN, A. Designing Employee Performance Monitoring Dashboard Using Key Performance Indicator (KPI). *Bit-Tech*. Leden 2020, sv. 2, s. 19–26. DOI: 10.32877/bt.v2i2.107.
- [26] GÜVEN, Y., COŞGUN, E., KOCAOĞLU, S., GEZICI, H. a YILMAZLAR, E. Understanding the Concept of Microcontroller Based Systems To Choose The Best Hardware For Applications. *Research Inveny: International Journal of Engineering And Science*. Prosinec 2017, sv. 7, s. 38.
- [27] HE, Y., GONG, S., HATTORI, R. a KANICKI, J. 35.3: OLED Based Display for Automotive Applications. *Polymer*. sv. 3, č. 2, s. 1.
- [28] HUITEMA, E. a FRENCH, I. E Ink’s Technicolor Moment: The Road to Color E-Paper Took Two Decades. *IEEE Spectrum*. Únor 2022, sv. 59, s. 30–35. DOI: 10.1109/MSPEC.2022.9706404.
- [29] HUSSAIN, A., HAMAD, M., HAFEEZ, K. a ZAIBNAB, T. Programming a Microcontroller. *International Journal of Computer Applications*. Prosinec 2016, sv. 155, s. 21–26. DOI: 10.5120/ijca2016912310.
- [30] IQBAL, A., ULLAH, F., ANWAR, H., KWAK, K. S., IMRAN, M. et al. Interoperable Internet-of-Things platform for smart home system using Web-of-Objects and cloud. *Sustainable Cities and Society*. Elsevier. 2018, sv. 38, s. 636–646.
- [31] JAYANTH, S. N., TEJA, N., BHAGYASRI, Y. a SHAHAPUR, K. RF based Remote Stage Lighting Controller. Květen 2016, s. 19–30. DOI: 10.13140/RG.2.1.4553.8165.
- [32] JOSEPH, R. J. Single page application and canvas drawing. *ArXiv preprint arXiv:1502.03530*. 2015.
- [33] KHAZANCHI, A., KANWAR, A., SALUJA, L., DAMARA, A. a DAMARA, V. OLED: a new display technology. *International Journal Of Engineering And Computer Science*. 2012, sv. 1, č. 2, s. 75–84.
- [34] KIRON, D. a SCHRAGE, M. Strategy For and With AI. *MIT Sloan Management Review*. Prosinec 2019, s. 30–35.
- [35] LARSEN, R. *Mastering SVG: Ace web animations, visualizations, and vector graphics with HTML, CSS, and JavaScript*. Packt Publishing Ltd, 2018. ISBN 9781788621984.
- [36] MAHMOOD, S., ALANI, S., HASAN, F. a MUSTAFA, M. ESP 8266 Node MCU Based Weather Monitoring System. In: *Proceedings of the 1st International Multi-Disciplinary Conference Theme: Sustainable Development and Smart Planning, IMDC-SDSP 2020, Cyberspace, 28-30 June 2020*. EAI, Leden 2020, s. 1–3. DOI: 10.4108/eai.28-6-2020.2298609.
- [37] MASSE, M. *REST API design rulebook: designing consistent RESTful web service interfaces*. O’Reilly Media, Inc., 2011. ISBN 9781449310509.

- [38] MEYER, J., SHINAR, D. a LEISER, D. Multiple Factors that Determine Performance with Tables and Graphs. *Human Factors*. Červen 1997, sv. 39, s. 268–286. DOI: 10.1518/001872097778543921.
- [39] MOHANTY, S. Everything You Wanted to Know About Smart Cities. *IEEE Consumer Electronics Magazine*. Červenec 2016, sv. 5, s. 60–70. DOI: 10.1109/MCE.2016.2556879.
- [40] MONTES, J. A Historical View of Smart Cities: Definitions, Features and Tipping Points. *SSRN Electronic Journal*. Leden 2020, s. 1–8. DOI: 10.2139/ssrn.3637617.
- [41] MORRISON, I. E Ink Displays. In: Society of Imaging Science and Technology. *NIP & Digital Fabrication Conference*. 2001, sv. 17, s. 523–525.
- [42] NATION, S. a OFFICE, D. G. *Smart Nation: The Way Forward Executive Summary*. Listopad 2018. Dostupné z: <https://www.smartnation.gov.sg/files/publications/smart-nation-strategy-nov2018.pdf>.
- [43] NISHINO, H. Ardestan: A Visual Programming Language for Arduino. In: *Adjunct Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*. říjen 2019, s. 93–95. DOI: 10.1145/3332167.3357126. ISBN 978-1-4503-6817-9.
- [44] OLIKRAUS. *Universal 8bit Graphics Library*. GitHub, 2016. Dostupné z: <https://github.com/olikraus/u8g2/wiki>.
- [45] OLIPHANT, T. E. *Guide to numpy*. 2nd. CreateSpace Independent Publishing Platform, 2015. ISBN 978-1-5173-0007-4.
- [46] PAPPAS, L. a WHITMAN, L. Riding the Technology Wave: Effective Dashboard Data Visualization. In: Springer. *Human Interface and the Management of Information. Interacting with Information: Symposium on Human Interface 2011, Held as Part of HCI International 2011, Orlando, FL, USA, July 9-14, 2011, Proceedings, Part I*. červenec 2011, s. 249–258. DOI: 10.1007/978-3-642-21793-7\_29. ISBN 978-3-642-21792-0.
- [47] PARMENTER, D. *Key Performance Indicators (KPI): Developing, Implementing, and Using Winning KPIs*. John Wiley & Sons, 2010. ISBN 9780470545157. Dostupné z: <https://books.google.cz/books?id=ZqmaJpgz4LsC>.
- [48] PATEL, K., PATEL, S., SCHOLAR, P. a SALAZAR, C. Internet of Things-IOT: Definition, Characteristics, Architecture, Enabling Technologies, Application & Future Challenges. Květen 2016.
- [49] PAUWELS, K., AMBLER, T., CLARK, B., LAPOINTE, P., REIBSTEIN, D. et al. Dashboards as a Service : Why, What, How, and What Research Is Needed? *Journal of Service Research*. Říjen 2009, sv. 12, s. 175–189. DOI: 10.1177/1094670509344213.
- [50] RAJABZADEH, A., MANASHTY, A. R. a JAHROMI, Z. F. A mobile application for smart house remote control system. *ArXiv preprint arXiv:1009.5557*. 2010.

- [51] RAMASAMY, L. a KADRY, S. Internet of things (IoT). In: *Blockchain in the Industrial Internet of Things*. IOP Publishing, Květen 2021, s. 1–16. 2053-2563. DOI: 10.1088/978-0-7503-3663-5ch1. ISBN 978-0-7503-3663-5.
- [52] SEHRAWAT, D. a GILL, N. S. Smart sensors: Analysis of different types of IoT sensors. In: IEEE. *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)*. 2019, s. 523–528. ISBN 978-1-5386-9439-8.
- [53] SHARIF, K. a TENBERGEN, B. Smart home voice assistants: a literature survey of user privacy and security vulnerabilities. *Complex Systems Informatics and Modeling Quarterly*. 2020, č. 24, s. 15–30.
- [54] SIAL, A. H., RASHDI, S. Y. S. a KHAN, A. H. Comparative analysis of data visualization libraries Matplotlib and Seaborn in Python. *International Journal*. 2021, sv. 10, č. 1.
- [55] SIEGENTHALER, E., SCHMID, L., WYSS, M. a WURTZ, P. LCD vs. E-ink: An Analysis of the Reading Behavior. *Journal of Eye Movement Research*. 2012, sv. 5, č. 3.
- [56] SIEGENTHALER, E., WURTZ, P., BERGAMIN, P. a GRONER, R. Comparing reading processes on e-ink displays and print. *Displays*. Prosinec 2011, sv. 32, s. 268–273. DOI: 10.1016/j.displa.2011.05.005.
- [57] SINGH, D., SANDHU, A., THAKUR, A. a PRIYANK, N. An Overview of IoT Hardware Development Platforms. *Int. J. Emerg. Technol.* Srpen 2020, sv. 11, s. 155–163.
- [58] SRINIVASAN, C., BODDUNA, R., SAIKALYAN, P., PREMSAGAR, K. a YADAV, E. S. A review on the different types of internet of things (IoT). *Journal of Advanced Research in Dynamical and Control Systems*. Leden 2019, sv. 11, s. 154–158.
- [59] SRIVASTAVA, D. An Introduction to Data Visualization Tools and Techniques in Various Domains. *International Journal of Computer Trends and Technology*. Duben 2023, sv. 71, s. 125–130. DOI: 10.14445/22312803/IJCTT-V71I4P116.
- [60] THIMBLEBY, H. Reasons to question seven segment displays. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. Duben 2013, s. 1431–1440. DOI: 10.1145/2470654.2466190. ISBN 978-1-4503-1899-0.
- [61] VARGA, P. a ILLÉSI, Z. Wi-Fi 6 Application in Iot Environment. *Interdisciplinary Description of Complex Systems*. Červen 2022, sv. 20, s. 277–283. DOI: 10.7906/indecs.20.3.7.
- [62] VETELINO, J. a REGHU, A. *Introduction to sensors*. CRC press, 2017. ISBN 9781315218274.
- [63] VINEEL, G. Web page DOM node characterization and its application to page segmentation. In: *2009 IEEE International Conference on Internet Multimedia Services Architecture and Applications (IMSAA)*. IEEE, 2009, s. 1–6. DOI: 10.1109/IMSAA.2009.5439444. ISBN 978-1-4244-4792-3.
- [64] WANG, Z., SUNDIN, L., MURRAY RUST, D. a BACH, B. Cheat Sheets for Data Visualization Techniques. In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. Leden 2020, s. 1–15. DOI: 10.1145/3313831.3376271. ISBN 978-1-4503-6708-0/20/04.

- [65] WILSON, D. Liquid Crystal Display (LCD) Inspection System. *National Instruments Case Study, available May*. 2009, sv. 10, s. 2.
- [66] WÓJCIK, P. a NEUMANN, T. A Personal Microcomputer as an Access Control Management Platform in Road Transport. *Applied Sciences*. Srpen 2023, sv. 13, s. 9770. DOI: 10.3390/app13179770.
- [67] YAMAGUCHI, K. CCFL Backlights. In: *LCD Backlights*. John Wiley & Sons, Ltd, 2009, kap. 7, s. 89–101. DOI: <https://doi.org/10.1002/9780470744826.ch7>. ISBN 9780470744826. Dostupné z: <https://onlinelibrary.wiley.com/doi/abs/10.1002/9780470744826.ch7>.
- [68] YANG, H., LI, Y. a ZHOU, M. Understand Users' Comprehension and Preferences for Composing Information Visualizations. *ACM Transactions on Computer-Human Interaction (TOCHI)*. Únor 2014, sv. 21, s. 1–25. DOI: 10.1145/2541288.
- [69] YIM, A., CHUNG, C. a YU, A. *Matplotlib for Python Developers: Effective techniques for data visualization with Python*. 2. vyd. Packt Publishing Ltd, 2018. ISBN 9781788628136.
- [70] YOSHIDA, Y., ISHIZUKA, A. a MAKISHIMA, H. Present and future of vacuum fluorescent display and field emission display. *Materials chemistry and physics*. Elsevier. 1995, sv. 40, č. 4, s. 267–272.
- [71] ZEILER, M. a HÜTTNER, J. LED Backlights. In: *LCD Backlights*. John Wiley & Sons, Ltd, 2009, kap. 14, s. 181–194. DOI: <https://doi.org/10.1002/9780470744826.ch14>. ISBN 9780470744826. Dostupné z: <https://onlinelibrary.wiley.com/doi/abs/10.1002/9780470744826.ch14>.



# Příloha A

## Obsah přiloženého paměťového média

- `./xfiser16.pdf` (finální text práce)
- `./xfiser16.zip` (zdrojové kódy v LaTeXu)
- `./hw_implementation/` (zdrojové soubory HW implementace - lze otevřít v programu EAGLE)
  - `./hw_scheme.sch` (schéma HW implementace)
  - `./hw_final_board.brd` (design HW implementace)
- `./sw_implementation/` (zdrojové soubory SW implementace)
  - `./README.md` (návod na sestavení implementace a ovládání zařízení)
  - `./CMakeLists.txt` (konfigurace a řízení procesu sestavování)
  - `./sdkconfig` (konfigurační nastavení pro projekt)
  - `./main/` (zdrojové soubory)
    - \* `./CMakeLists.txt` (konfigurace a řízení procesu sestavování)
    - \* `./*.c` (zdrojové C soubory)
    - \* `./*.h` (hlavičkové soubory ke zdrojovým C souborům)
- `./visualization_designs/` (návrhy vizualizací, ze kterých vychází finální implementace)
  - `./*.png` (návrhy designů)