

České vysoké učení technické v Praze

Fakulta elektrotechnická

Katedra radioelektroniky



Bakalářská práce

System pro automatizaci domácnosti

System for Home Automation

Autor: Václav Bernášek

Vedoucí práce: Ing. Stanislav Vítek, Ph.D.

Studijní program: Elektronika a komunikace

Praha 2023

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Bernášek** Jméno: **Václav** Osobní číslo: **498999**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávající katedra/ústav: **Katedra radioelektroniky**
Studijní program: **Elektronika a komunikace**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Systém pro automatizaci domácnosti

Název bakalářské práce anglicky:

System for Home Automation

Pokyny pro vypracování:

Cílem práce je vytvoření systému pro domácí automatizaci s využitím mikrokontroléru ESP32 a bezdrátové komunikační technologie ZigBee. Při vypracování se řiďte následujícími pokyny:

- 1) vypracujte přehled dostupných systémů pro domácí automatizaci
- 2) navrhnete vlastní řešení systému domácí automatizace, specializovaného na dosažení tepelného komfortu uživatele
- 3) navržený systém implementujte tak, aby bylo možné monitorovat klíčové parametry systému a všechny měřené veličiny
- 4) vlastní implementaci porovnejte s dostupnými systémy

Seznam doporučené literatury:

- [1] RAZA, Ali, et al. A review of low cost and power efficient development boards for IoT applications. In: 2016 Future Technologies Conference (FTC). IEEE, 2016. p. 786-790.
[2] BABIUCH, Marek; POSTULKA, Jiri. Smart Home Monitoring System Using ESP32 Microcontrollers. IntechOpen, 2021.

Jméno a pracoviště vedoucí(ho) bakalářské práce:

doc. Ing. Stanislav Vítek, Ph.D. katedra radioelektroniky FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **13.02.2023**

Termín odevzdání bakalářské práce: **26.05.2023**

Platnost zadání bakalářské práce: **22.09.2024**

doc. Ing. Stanislav Vítek, Ph.D.
podpis vedoucí(ho) práce

doc. Ing. Stanislav Vítek, Ph.D.
podpis vedoucí(ho) ústavu/katedry

prof. Mgr. Petr Páta, Ph.D.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

Datum převzetí zadání

Podpis studenta

Poděkování

Děkuji vedoucímu práce, doc. Ing. Stanislavu Vítkovi, Ph.D. za veškerou pomoc při tvorbě a za poskytnuté zázemí. Dále bych chtěl poděkovat Ing. Vladimíru Janíčkovi, Ph.D. za jeho čas při konzultacích a za doporučení hardwaru.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne 22.5.2023

.....

Abstrakt

Cílem této bakalářské práce je vytvoření systému pro snímání, ukládání a následné zobrazení naměřených dat. Uživatel je schopen nejenom data zobrazit, ale i změnit parametry měřené místnosti, a to konkrétně ovládání světla. Samotný sběr dat probíhá přes platformu ESP32 a přes bezdrátovou síť ZigBee. Data jsou zálohovaná v cloudu InfluxDB a zároveň jsou přístupná uživateli kdekoliv na světě prostřednictvím mobilní aplikace. Díky využití open source platform jako je ESP32, ZigBee2MQTT či Node RED je velký prostor pro případné budoucí přizpůsobení nelimitované samotnou platformou, na které systém funguje. Finální systém bohužel celkově nebyl konkurenčně schopný oproti jiným komerčním možnostem v oblasti automatizace domácnosti. Je zřejmé, že úspěšné vytvoření vlastního systému bez spolehnutí se na platformy jako např. Home Assistant vyžaduje mnohonásobně vyšší úsilí a lepší technické znalosti, než kterými v tuto chvíli disponuji.

Klíčová slova: automatizace domácnosti, mikrokontroler, cloud, databáze, sensor, zigbee, ESP32, zigbee2mqtt, node RED, InfluxDB

Abstract

The goal of this bachelor's thesis is to create a system for capturing, storing and then displaying measured data. The user is able not only to display the data, but also to change the parameters of the measured room, specifically the light control. The data collection itself takes place via the ESP32 platform and via the ZigBee wireless network. The data is backed up in the InfluxDB cloud and at the same time is accessible to users anywhere in the world via a mobile application. Thanks to the use of open-source platforms such as ESP32, ZigBee2MQTT or Node RED, there is a lot of room for possible future customization, not limited by the platform itself, on which the system works. Unfortunately, the final system overall failed to be competitive against other commercial options in the field of home automation. Successfully creating your own system without relying on platforms such as Home Assistant requires many times more effort and better technical knowledge than I have at disposal.

Keywords: home automation, microcontroller, cloud, database, sensor, zigbee, ESP32, zigbee2mqtt, node RED, InfluxDB

Seznam použitých zkratek

AES	Advanced encryption standard
API	Application programming interface
BLE	Bluetooth low energy
BPSK	Binary-phase shift keying
CLI	Command line interface
CSI	Camera serial interface
DPSK	Differential phase shift keying
DSI	Display serial interface
DSSS	Direct-sequence spread spectrum
DQPSK	Differential quadrature phase shift keying
DI	Discomfort index
DIY	Do it yourself
ESP	Embedded systems programming
FFD	Full function device
FHSS	Frequency hopping spread spectrum
FBD	Flow-based programming
GFSK	Gaussian frequency-shift keying
GPIO	General-purpose input/output
GND	Ground (v elektrických obvodech)
HDMI	High definition multimedia interface
HTTP	Hypertext transfer protocol
I2C	Inter integrated controller
ICSP	In-circuit serial programming
IDE	Integrated development environment
IoT	Internet of things
LPDDR	Low-power double data rate
MQTT	MQ telemetry transport
NFC	Near-field communication
OFDM	Orthogonal frequency division multiplexing
OS	Operating system
OQPSK	Offset quadrature phase-shift keying
P2P	Peer to peer

PMV	Predicted mean vote
QAM	Quadrature amplitude modulation
QPSK	Quadrature phase-shift keying
RFD	Reduced function device
SD	Secure digital
SDRAM	Synchronous dynamic random access memory
SPI	Serial peripheral interface
SoC	System on chip
SSH	Secure Shell
UART	Universal asynchronous receiver-transmitter
UI	User interface
URL	Uniform resource locator (web)
USB	Universal serial bus
WEP	Wired equivalent privacy
WiFi HaLow	Low-power WiFi
WPA	WiFi protected access

Seznam obrázků

Obrázek 1: Vizuální podoba vybraných mikrokontrolerů [4], [5], [7], [11], [12]	6
Obrázek 2: ZigBee2MQTT architektura [20]	14
Obrázek 3: Příklad záznamu InfluxDB databáze.....	15
Obrázek 4: Příklad Flow kódu v Node RED se zobrazením dat z cloudu na Dashboard	16
Obrázek 5: Faktory ovlivňující tepelnou pohodu [24].....	17
Obrázek 6: Stupnice pro PMV [24].....	17
Obrázek 7: Foto hardwarového jádra	20
Obrázek 8: Bezdrátové periferie použité v realizaci	21
Obrázek 9: Fyzická podoba implementace s ESP32	23
Obrázek 10: Graf závislosti parametru DI na čase (15.5 – 21.5.)	29
Obrázek 11: Schéma obvodu s ESP32	36
Obrázek 12: Diagram propojení systému	37
Obrázek 13: Schéma kódu v Node RED.....	38
Obrázek 14: Uživatelské rozhraní (UI) na počítači/telefonu	39

Seznam Tabulek

Tab. 1: Srovnání vybraných kontrolerů [4], [5], [7], [11], [12]	6
Tab. 2: Typické zařízení IoT a jejich způsob komunikace [14].....	8
Tab. 3: Srovnání různých bezdrátových technologií komunikace [3]	11
Tab. 4: Klasifikace tepelného vnímání člověka dle parametru DI [26].....	18

Obsah

1. Úvod.....	1
2. Teoretická část.....	3
2.1. Mikrokontroléry.....	4
2.1.1. ESP32 WROOM 32D	4
2.1.2. Raspberry Pi Pico WH	4
2.1.3. Arduino UNO.....	4
2.1.4. BeagleBone Black	5
2.1.5. Raspberry Pi 3 Model A+.....	5
2.1.6. Srovnání jednotlivých mikrokontrolérů	6
2.2. Typy bezdrátové komunikace v IoT	8
2.2.1. Bluetooth	9
2.2.2. WiFi	9
2.2.3. ZigBee	10
2.2.4. Srovnání jednotlivých bezdrátových sítí	11
2.3. Systémy pro automatizaci domácnosti.....	13
2.3.1. Google Nest (Google home)	13
2.3.2. SmartThings	13
2.3.3. Apple HomeKit	13
2.3.4. Amazon Alexa (Alexa).....	13
2.4. ZigBee2MQTT	14
2.5. InfluxDB	15
2.6. Node RED.....	16
2.7. Subjektivní pohodlí člověka.....	17
3. Praktická část	19
3.1. Hardware zpracovávající data	20
3.2. Výběr připojovaných zařízení	21
3.3. Popis celkové konstrukce systému.....	22
3.4. Implementace ESP32	23
3.4.1. Popis obvodu.....	23
3.4.2. Kód ESP32	24
3.5. Node RED v implementaci	25
3.5.1. Popis způsobu práce v Node RED	25
3.5.2. Popis kódu v Node RED	25

3.5.3. Popis uživatelského rozhraní (UI).....	26
4. Diskuse nad výsledky	27
4.1. Srovnání s komerčními platformami.....	27
4.1.1. Výhody.....	27
4.1.2. Nevýhody	27
4.1.3. Srovnání	28
4.2. Zhodnocení hodnot změřené senzory.....	29
5. Závěr.....	30
6. Bibliografie.....	31
Přílohy	36

1. Úvod

Automatizace domácnosti nabízí velký potenciál pro obyvatele domů takové systémy obsahující. Zlepšení můžeme pozorovat ve zvýšení pohodlí, v automatizaci některých úloh, v šetření energií chytrým vypínáním spotřebičů, či ve zvýšení bezpečnosti přidáním kamer či chytrých zámek. Propojením těchto výhod do jednoho systému celkově pak nakonec šetříme čas i peníze. Poptávka o tyto systémy je skutečně velká, což je číselně dokazatelné velikostí trhu. Ten měl v roce 2018 hodnotu 59,5 miliard dolarů či v přepočtu 1,3 bilionu korun českých [1]. A nemusíme se spoléhat pouze na jedno číslo. Celý IoT průmysl v této době prochází masivním růstem, kdy např. v roce 2020 velikost trhu IoT vzrostla o 38,3 % oproti roku 2019 [2]. Není pak náhoda, že mnoho značek jako např. Google, Samsung, Apple či Amazon bojuje o svůj tržní podíl. Produkty všech výše zmíněných značek bohužel v menší, či větší míře bývají soustředěné na svůj vlastní ekosystém, který pak odrazuje zákazníky před zakoupením zajímavých produktů, které však nejsou slučitelné s jejich dosavadním systémem.

Možná i proto se v této době začínají objevovat projekty používající systémy, které nejsou svojí platformou omezovány, jako např. Home Assistant (viz [2]). Home Assistant je právě jeden ze systémů bojující proti exkluzivitě poskytováním platformy, která je open source a která má širokou škálu podporovaných zařízení a softwarů napříč značkami.

Cílem této bakalářské práce bylo prozkoumat možnosti pro automatizaci domácnosti a pak na základě dat ze zkoumání převzaté vytvořit systém vlastní. Ten by podobně jako Home Assistant byl open source, vycházel by levněji než komerční řešení, poskytoval by zajímavé funkcionality a měl by možnosti budoucího rozšíření.

2. Teoretická část

V této kapitole je rozebrán hardware a software často využívaný v IoT odvětví. Nejdříve je v kapitole 2.1 představeno několik mikrokontrolerů často využívaných v projektech s IoT, které jsou následně mezi sebou porovnány. Poté je probráno několik bezdrátových technologií (kapitola 2.2), které jsou zhodnoceny v jejich vhodnosti pro použití v chytré domácnosti. Pak je v kapitole 2.3 jen velice krátce ukázáno několik platforem nabízející komerční produkty automatizace domácnosti, které později pak ve zhodnocení (kapitola 4.1) jsou porovnány s praktickou částí této práce. Dále kapitoly 2.4, 2.5 a 2.6 se věnují službám, využívaných v praktické části. Konkrétně se jedná o databázovou službu InfluxDB, open source projekt ZigBee2MQTT a programovací nástroj Node RED. Poslední část této kapitoly (kapitola 2.7) se věnuje představení možného měřítka, kterým je možné měřit tepelnou komfortnost člověka.

2.1. Mikrokontroléry

2.1.1. ESP32 WROOM 32D

ESP32 je skupina mikrokontrolérů zaměřená na širokou škálu aplikací, a to až už jde o sbírání dat, kontrolu různých periférií či zpracování a bezdrátové posílání dat. Díky své nízké ceně a vysoké všestrannosti jsou velice oblíbené. Konkrétní specifikace pro verzi WROOM 32D jsou pak v Tab. 1. Přednostmi ESP32 je například schopnost bezdrátové komunikace (WiFi, Bluetooth, BLE), nízká spotřeba, integrovaný senzor Hall-ova efektu, možnost režimu spánku či schopnost náročných aplikací jako MP3 dekodování. Pro programování desky je možno využít velké řady programovacích jazyků včetně JavaScript, C++ či Mikro Python.

Pro propojení s perifériemi má deska 34 GPIO pinů, 3x UART, 4x SPI bus, 2x I2C bus, 1x 5 V napájení, 3x 3.3 V napájení, 4x GND [3]. Kromě toho má i 10 GPIO pinů, které jsou schopné pracovat s kapacitním displejem.

2.1.2. Raspberry Pi Pico WH

Raspberry Pi Pico je nejnovější z mikrokontrolerů mého výběru. Obsahuje první čip vyvinutý společností Raspberry Pi. Přesné specifikace jsou pak v Tab. 1. Mezi přednosti desky patří uživatelská přívětivost, nízká energetická náročnost a nízká cena. Pro programování je v základní konfiguraci možno použít C, C++ či Mikro Python.

Pro propojení s perifériemi je na desce 26 GPIO pinů, 2x UART, 2x SPI bus, 2x I2C bus, 1x 3.3 V napájení, 8x GND [4].

2.1.3. Arduino UNO

Arduino UNO je jedna z nejznámějších platforem s mikrokontroléry vůbec. I když byla vydaná už v roce 2010, tak stále dostává pravidelné úpravy a zlepšení. Kromě programování přes USB je možné nahrát program i napřímo přes ICSP piny. Konkrétní specifikace jsou pak v Tab. 1. V základní konfiguraci je programovacím jazykem C či C++.

Deska Arduino UNO má 14 GPIO pinů, 6 analogových vstupů, 1x 5 V napájení, 1x 3.3 V napájení, 2x GND. [5]

Zajímavostí této desky od ostatních je vyměnitelný čip mikrokontroleru. Pokud tedy dojde k poškození uživatelem, tak není potřeba vyhodit desku celou, ale jen vyměnit čip za 200 Kč [6].

2.1.4. BeagleBone Black

BeagleBone Black je nejdražší z vypsanych mikrokontrolerů. Přesné specifikace lze pak najít v Tab. 1. Mezi jeho přednosti patří NEON Akcelerometr, 3D grafický akcelerátor či interní flash paměť se 4 GB. Případný OS je pak nahráván přes externí mikro SD kartu. Poté je možno ovládat mikrokontroler napřímo přes OS či využitím mnoha podporovaných programovacích jazyků včetně Pythonu, Java či C++.

Velká přednost této desky je její množství připojitelných konektorů. Je možno připojit až 65 GPIO pinů, 4x UART (s omezením), 2x I2C bus, 2x SPI bus, 2x 5V napájení, 2x 3.3 V napájení a 2x GND. Důležité je také zmínit, že některé piny mají až 8 různých nastavení, včetně možnosti připojení dalšího HDMI. K tomu navíc samotná deska má vestavěné 2 malé 32bitové 200 MHz mikrokontroléry, které mohou být nastaveny pro dedikované zvládání podružných úkonů [7]. S ohledem na IoT je tato deska vhodná pro projekty vyžadující nadměrné množství pinů.

2.1.5. Raspberry Pi 3 Model A+

Raspberry Pi 3 Model A+ je jeden z novějších mikrokontrolerů vyvinutý společností Raspberry Pi. Díky několika iteracím od prvního modelu je tato verze dobře optimalizovaná a velmi výkonná ve své cenové kategorii [8]. Konkrétní specifikace jsou pak vypsány v Tab. 1. Kvůli neexistenci vestavěného úložiště je potřeba připojení mikro SD karty, na které se nahrávají všechny potřebné soubory včetně případného OS (většinou na bázi Linuxu). Pro ovládání je sice možno využít periférií jako HDMI a USB, ale pro potřeby IoT je mnohem lepší vzdálený přístup přes WiFi, protože pak není potřeba mít fyzický přístup k samotné desce a jsme ji schopni ovládat i z druhé strany světa (při správné konfiguraci) [9]. Pro práci s Raspberry se obvykle nahraje operační systém a pak se mikrokontroler podobně jako BeagleBone Black kontroluje napřímo přes příkazový řádek či přes programovací jazyky.

Pro připojení k externím perifériím je možno využít buď již připravených konektorů (USB, HDMI, atd.), či GPIO pinů. Z nich 24 jsou klasické GPIO, 1x mini UART, 2x SPI bus, 1x I2C bus, 2x 5V napájení, 2x 3.3V napájení, 8x GND [10].

2.1.6. Srovnání jednotlivých mikrokontrolérů

Tab. 1: Srovnání vybraných kontrolérů [4], [5], [7], [11], [12]

Název	Rozměry [mm]	Cena bez periferií	Procesor	Paměť	Bezdrátové připojení	Kabelové připojení
ESP32 WROOM 32D	55 x 27	240 Kč	ESP32-D0WDQ6	SRAM 520 kB	WiFi, Bluetooth, BLE	Mikro USB, 25 GPIO pin header
Raspberry Pi Pico WH	51 x 21	210 Kč	Dual-core Arm Cortex M0+	264KB SRAM	WiFi, Bluetooth	Mikro USB, 26 GPIO pin header
Arduino UNO	53 x 69	800 Kč	ATmega328P	SRAM 2 kB	Nemá	USB konektor, Jack k napájení, 14 GPIO pin header, 6 analog vstupů
BeagleBone Black	86 x 53	1 400 Kč	AM335x 1GHz ARM® Cortex-A8	512MB DDR3 RAM	nemá (existují však verze s WiFi)	2x 48 GPIO pin header, Ethernet 10/100, 1x USB 2.0, Mikro HDMI, napájecí konektor 5.5 mm
Raspberry Pi 3 Model A+	65 x 56	800 Kč	Broadcom BCM2837B0	512MB LPDR2 SDRAM	WiFi, Bluetooth, BLE	40 GPIO pin header, HDMI, DSI displej port, CSI displej port, 3.5mm audio Jack s možností kompozitního video signálu, USB 2.0 port, Mikro USB pro napájení



ESP32



Raspberry Pi Pico



Arduino UNO



BeagleBone Black



Raspberry Pi 3

Obrázek 1: Vizuální podoba vybraných mikrokontrolérů [4], [5], [7], [11], [12]

Každý z mikrokontrolérů v Tab. 1 se hodí pro něco jiného. Pro snadnější porovnání je můžeme rozdělit do dvou kategorií, a to ty s vlastním operačním systémem a ty bez něj. To nám pak umožňuje odvodit jejich potenciální využití.

První tři kontroléry v tabulce (bez operačního systému) jsou ideální pro specializované úlohy, kde stačí desku naprogramovat a pak ona sama dlouhodobě vykonává vyžadovanou úlohu s velkou úsporou energie (díky energeticky nenáročným procesorům). Konkrétně ESP32 je velice všestranné, Raspberry Pi Pico je jednoduché k použití a Arduino UNO funguje dobře s Arduino ekosystémem.

Naopak poslední dva kontroléry jsou vhodné pro úlohy, které operační systém vyžadují, jako je např. používání různého softwaru. Tato schopnost kombinovaná s nízkou spotřebou oproti normálním počítačům je perfektní pro dlouhodobý běh cloudu či obecně serveru. BeagleBone Black má výhodu většího vestavěného paměťového modulu a většího počtu periférií. Naopak Raspberry Pi 3 model A+ má vyšší výpočetní výkon a širokou škálu kompatibilních modulů díky ekosystému Raspberry Pi.

2.2. Typy bezdrátové komunikace v IoT

Pro správné zvolení bezdrátové komunikace je potřeba si jednotlivé možnosti rozebrat. Avšak kvůli jejich velkému množství zde bude popsán jen určitý výběr týkající se IoT. Pak dojde k srovnání s ohledem na použití v automatizaci domácnosti.

Při výběru typu komunikace pro konkrétní aplikaci je třeba zohlednit následující parametry:

- energetické náročnost,
- cena,
- bezpečnost,
- maximální vzdálenost,
- dostupné topologie,
- možná hustota sítě (rušení),
- průtok dat,
- frekvence, na které technologie funguje.

Informace o jednotlivých sítích byli částečně převzaty z [13] a [14].

Tab. 2: Typické zařízení IoT a jejich způsob komunikace [14]

Typ zařízení	Chipset	CPU frekvence	RAM	ROM/Flash	Způsob napájení	Technologie komunikace
iPhone A7x Quad-core Processor	Quad-core Processor	1.7 GHz	2 Gb	128 Gb	Baterie	WiFi, Bluetooth, NFC
Samsung Smart TV	Exonys SoC	1.3 GHz	1Gb	-	Zásuvka	WiFi
Nest Learning Thermostat	ARM Cortex-A8	800 MHz	512 Mb	2 Gb	Baterie	WiFi
Nest Smoke Detector	ARM Cortex-M0	48 MHz	16 Kb	128 Kb	Baterie	WiFi
Fitbit Smart Wrist Band	ARM Cortex-M3	32MHz	16 Kb	128 Kb	Baterie	Bluetooth Low Energy
Philips Hue Light bulb	TI CC2530 SoC	32MHz	8 Kb	up to 256 Kb	Baterie	ZigBee
Sensor Devices Microcontroller	4-32 MHz	4-16 Kb	4-16 Kb	16-128 Kb	baterie, zásuvka	ZigBee, WiFi, Bluetooth

2.2.1. Bluetooth

Standard Bluetooth byl původně vyvinut jako způsob bezdrátové komunikace mezi mobilními zařízeními na krátkou vzdálenost. Mezi jeho klíčové vlastnosti patří nízká spotřeba energie, cenová dostupnost a malé rozměry čipu. Díky těmto vlastnostem je široce využíván v širokém spektru odvětví.

Komunikace probíhá na 2.4 GHz, která je bohužel často využívaná sítěmi jako WiFi či ZigBee, a proto se často můžeme setkat s rušením. Rychlost je obvykle do 3 Mb/s. Frekvenční pásmo je od 2402 MHz do 2480 MHz a je rozděleno do 79 kanálů (tedy 1 MHz na kanál).

Ve spojitosti s IoT je vhodné zmínit ještě BLE, což je typ Bluetooth specializovaný právě pro malé zařízení, vyžadující vysokou úsporu energie. Na rozdíl od normálního Bluetooth může mít i stonásobně nižší spotřebu energie (liší se produkt od produktu) [15]. To je umožněno dávkováním dat, kdy zařízení posílá informace jen v určitých časových intervalech, ale jinak je v režimu spánku. Dále množství posílaných dat je poloviční (záleží na konkrétním zařízení), což dále snižuje spotřebu energie. BLE navíc podporuje tvorbu sítí s topologií mesh, které jsou velmi vhodné pro IoT.

Srovnání jak Bluetooth, tak BLE, je pak popsáno níže v Tab. 3.

2.2.2. WiFi

WiFi je skupina protokolů bezdrátové komunikace užívaná pro připojení zařízení vyžadující vysoký tok dat. Rychlost v ideálních podmínkách může být i několik Gb/s. Díky tomu je velmi rozšířená a je preferovaná jako bezdrátový způsob komunikace zařízení s internetem. V reálném světě však podléhá značnému rušení z okolí (ze stejného důvodu jako Bluetooth), a tak teoretická rychlost je obvykle těžko dosažitelná.

Komunikace funguje na 2,4 GHz či na 5 GHz. Vysoká rychlost přenosu dat je dosažena pomocí širokých kanálů s šířkou až 22 MHz.

Protokol WiFi se využívá i v IoT aplikacích. Přináší to ale určité nevýhody, jako je například vysoká spotřeba energie. Na druhou stranu však má vyšší bezpečnost než Bluetooth či ZigBee, protože můžeme komunikovat přes bezpečnostní protokoly jako WEP, WPA, WPA2 či WPA3.

Pokud bychom i tak chtěli WiFi pro IoT využít, tak WiFi Aliance vytvořila protokol WiFi HaLow. Ten byl vytvořen právě v reakci na zvětšující se poptávku o IoT komunikaci. Nese si sebou výhody WiFi, ale přitom mění určité své parametry pro IoT aplikace. Zvýšen byl například dosah, který je pro WiFi HaLow i vyšší stovky metrů. Dále byla snížena energetická náročnost, i když ta je u WiFi HaLow obecně stále vyšší než u ZigBee či BLE [14]. Pozor ale na to, že WiFi a WiFi HaLow nejsou spolu kompatibilní, takže není možné využít stejný hardware (jiná pracovní frekvence). Podrobné specifikace pak v Tab. 3.

2.2.3. ZigBee

ZigBee je komunikačním standardem, který se specializuje na spolehlivý přenos dat na relativně krátkou vzdálenost, a to při zachování nízké energetické náročnosti a nízkých nákladů na konstrukci [14]. Maximální rychlost je jen do 250 kbit/s, ale to je v případě jeho aplikací dostačující, protože většinou pracujeme jen s daty senzorů, žárovek a tak podobně.

ZigBee může operovat na třech různých frekvencích: 2.4 GHz, 868 MHz nebo 915 MHz. Používá DSSS modulaci, která přispívá k vyšší spolehlivosti a odolnosti proti silnému rušení [16]. Při komunikaci se používají tři základní profily: ZigBee, ZigBee PRO a ZigBee IP. ZigBee PRO je optimalizováno pro sítě až s tisíci zařízeními, zatímco profil ZigBee je vhodný pro komunikaci menších sítí se stovkami zařízeními. Zařízení dokážou navzájem komunikovat, aniž by záviselo na použitém profilu. Výjimkou je ZigBee IP, který navíc umožňuje připojení k internetu.

ZigBee PRO má 3 prvky, které výsledně tvoří celou síť. Je to jmenovitě:

- ZigBee Koordinátor (FFD) – pouze jeden pro celou komunikační síť,
- ZigBee Router (FFD),
- ZigBee koncové zařízení (RFD/FFD).

Základem celé sítě je ZigBee Koordinátor, který plní klíčovou roli v procesu vytváření nové sítě, v připojování nových zařízení, v udržování funkčnosti sítě, v ukládání informací o síti a v zajištění bezpečnosti sítě. Routery pak slouží jako rozšíření sítí, neboť koordinátor má omezený počet připojitelných zařízení. Zařízení, která nejsou ani koordinátorem, ani routerem, jsou označována jako koncová zařízení.

Ty se dělí do dvou typů: RFD a FFD. RFD zařízení není schopno komunikovat přímo s jinými zařízeními, mimo router nebo koordinátor. Navíc tato zařízení nelze překonfigurovat na routery nebo koordinátory, a tak zůstávají trvale koncovými zařízeními. Na druhou stranu, RFD zařízení je možné překonfigurovat. Pokud tedy máme tři RFD zařízení, tak můžeme například nastavit jedno jako koordinátor, jedno jako router a jedno jako koncové zařízení.

2.2.4. Srovnání jednotlivých bezdrátových sítí

Tab. 3: Srovnání různých bezdrátových technologií komunikace [3]

		ZigBee	WiFi HaLow	Bluetooth	BLE
Standart		IEEE 802.15.4	IEEE 802.11ah	IEEE 802.15.1	IEEE 802.15.1
Frekvence		2.4 GHz, 868, 915 MHz	900 MHz	2.4 GHz	2.4 GHz
Dosah [m]	uvnitř	10-100	< 700	1, 10, 100	50
	venku		< 1000		
Rychlost dat (teoreticky)		20, 40, 250 Kb/s	150-400, 650-780 Kb/s	1, 2, 3 Mb/s	1 Mb/s
Rychlost dat (prakticky)		10-115.2 Kb/s	> 100 Kb/s	0.7-2.1 Mb/s	305 Kb/s
Míra odběru energie [mA]		< 40	-	< 30	< 12.5
Tx výstupní výkon [dBm]	z	-3	10	-6	< 19
	do	10	30	20	
Multiplexování		DSSS	OFDM	FHSS	FHSS
Modulace		DSSS, OQPSK, BPSK	BPSK, QPSK, 16-QAM, 64-QAM, 256-QAM	GFSK, $\pi/4$ - DQPSK, 8DPSK	GFSK
Algoritmus Bezpečnosti		AES-128	WEP, WPA, WPA2	E0, E1, E21, E22, E3, 56-128 bit	AES-128
Topologie		star, tree, mesh	one-hop	p2p, scatternet	p2p, star, mesh

Jak můžeme vidět, tak technologie BLE spotřebovává nejméně energie. To však musíme brát s rezervou, protože spotřeba více souvisí s konkrétní aplikací než plošně pro všechna užití. Dále největší dosah a bezpečnost má WiFi HaLow a nejrychlejší přesun dat má Bluetooth. ZigBee, podobně jako BLE, podporuje mesh sítě, ale kvůli optimalizaci ZigBee pak vyniká nad BLE při práci s větším množstvím zařízení.

Z dat vypsanych v Tab. 3 jsem vybral dva druhy bezdrátové komunikace, a to BLE a ZigBee. Ze všech možností výše zmíněných mají největší potenciál pro tvorbu sítí na automatizaci domácnosti.

Výhody BLE jsou:

- vyšší rychlost,
- větší propojitelnost se zařízeními jako mobilní telefony.

Výhody ZigBee jsou:

- lepší škálovatelnost,
- větší dosah (pouze oproti konvenčním typům zařízení BLE),
- menší rušení z okolí (může pracovat na různých frekvencích, je více optimalizovaný pro potlačení rušení).

2.3. Systémy pro automatizaci domácnosti

Na trhu existuje spousta komerčních voleb pro tvorbu chytré domácnosti. Níže jsou vypsané některé z prominentních značek takové řešení nabízející. Informace byly převzaty z [17] a jsou z roku 2018.

2.3.1. Google Nest (Google home)

Společnost Google Nest nabízí širokou řadu produktů přizpůsobených pro chytrou domácnost. Filozofie těchto produktů spočívá v propojení chytrých zařízení s umělou inteligencí. Jejich produktová řada zahrnuje senzory, chytré reproduktory a zabezpečovací systémy, včetně kamer a zámků. Hlavní výhodou je příjemné uživatelské rozhraní a pokročilé hlasové ovládání. Je však třeba zdůraznit, že tyto produkty mají omezenou podporu pro jiné značky než od Google.

2.3.2. SmartThings

Společnost SmartThings nabízí řešení pro domácí automatizaci s širokým výběrem zařízení. Podporuje hlasové ovládání pomocí asistentů jako je Alexa nebo Google Assistant a zvládá různé typy bezdrátové komunikace, včetně WiFi, ZigBee a Z-Wave. Vše je přehledně shrnuto do jedné intuitivní mobilní aplikace, která umožňuje automatizovat akce jednotlivých zařízení.

2.3.3. Apple HomeKit

Apple HomeKit je systém vyvinutý s cílem usnadnit úkony v domácnosti. Jeho specializací je propojení různých Apple zařízení do chytré domácnosti, které se pak dají ovládat ve velmi propracované aplikaci. Je však třeba poznamenat, že ekosystém Apple HomeKit nemá velkou podporu produktů třetích stran.

2.3.4. Amazon Alexa (Alexa)

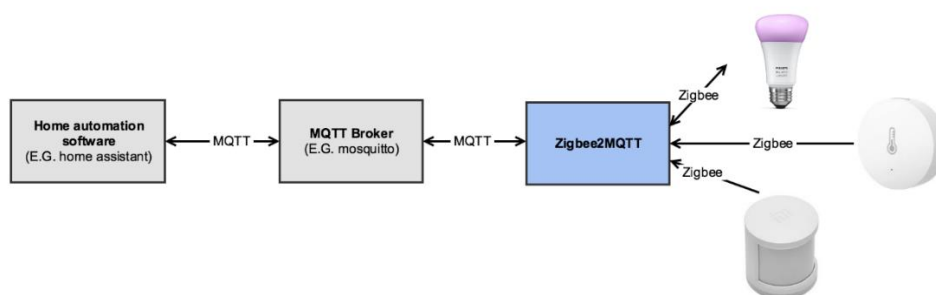
Amazon Alexa je systém automatizace specializující se na ovládání domácnosti hlasem. Podporuje docela slušné množství zařízení a bezdrátově umí komunikovat přes WiFi a ZigBee. Není možná ale vždy nejspolehlivější a některé části mobilní aplikace mohou být až trochu složité.

2.4. ZigBee2MQTT

ZigBee2MQTT je o open source projekt vyvíjený Koenem Kantersem. V jeho jádru má podobné funkcionality jako MQTT protokol, avšak je obohacen o schopnost interpretovat ZigBee zprávy a ty následně přes MQTT je schopen posílat dále. Právě díky tomuto projektu je pak umožněno použít zařízení od různých výrobců (výpis podporovaných zařízení viz [18]). ZigBee2MQTT také podporuje systémy automatizace domácnosti jako Home Assistant, Domoticz či openHAB [2].

Menším problémem je jen možná právě skutečnost, že projekt ZigBee2MQTT není zaštitěný určitou organizací. Informace o nastavení a postavení systému se tak musí čerpat z webových stránek autora na téma ZigBee2MQTT [19] a z jeho GitHub depozitáře [20]. Z těchto zdrojů lze však najít skoro vše, co je pro řádné sestavení systému potřebné. A to od firmware updatu ZigBee koordinátora, propojení ZigBee2MQTT brány s různými operačními systémy, párování nových zařízení do sítě až do integrace s jednotlivými systémy automatizace domácností zmíněné výše v této kapitole.

ZigBee2MQTT se pak dělí do 3 modulů. První z nich, zigbee-herdsman, je open source řešení pro ZigBee bránu. Druhý modul, zigbee-herdsman-converters, se stará o mapování ZigBee zařízení do patřičných skupin. Tyto skupiny pak definují, jak ZigBee síť funguje. Z pohledu ZigBee zařízení se tyto dva moduly dohromady jeví jako ZigBee koordinátor, čímž zprostředkovávají veškerou ZigBee komunikaci (viz kapitola 2.2.3). Poslední modul, ZigBee2MQTT, používá předchozí dva moduly a překládá ZigBee zprávy do MQTT zpráv. Také dělá zálohu každému zařízení o jeho nejnovějším stavu ve složce „database.db“. Status je uložen ve formátu JSON databáze a obsahuje všechna důležitá data o každém spárovaném zařízení [2].



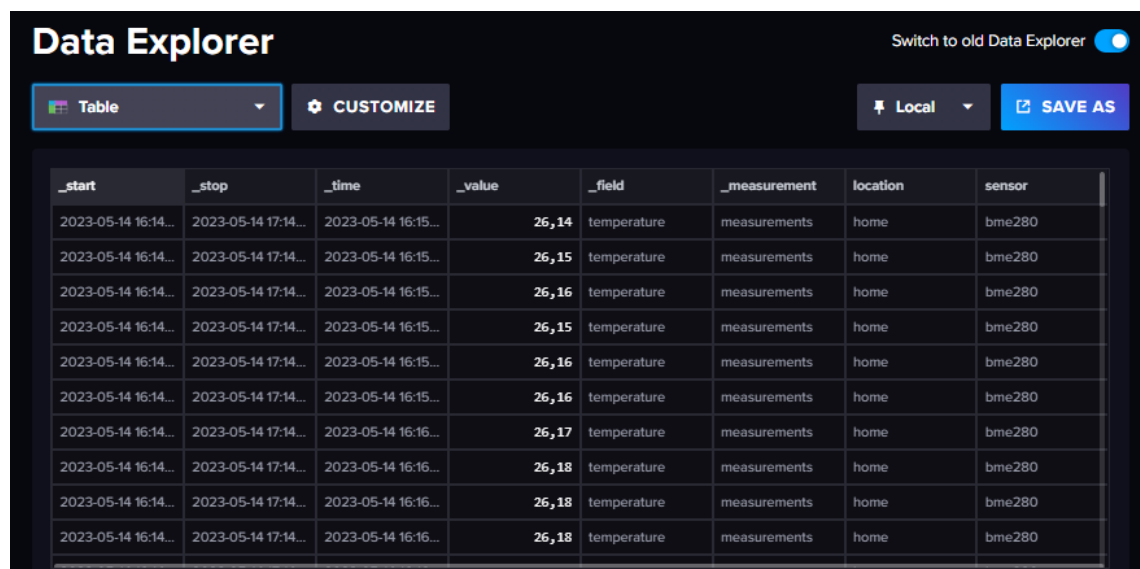
Obrázek 2: ZigBee2MQTT architektura [20]

2.5. InfluxDB

InfluxDB je o databáze časových řad postavená zvládat vysoké nároky na vkládání, úpravu a stahování dat. Je vyvíjena firmou InfluxData. Postavená je pro optimalizaci operací jako je sběr, monitorování a analýza dat v reálném čase [21].

Data jsou ukládány ve formě datových bodů, ve kterých je zvykem popsat alespoň čas a hodnotu změřené veličiny. K tomu navíc můžou být pro přehled přidány štítky pro lepší orientaci mezi údaji. Příklad takového záznamu je například obrázek 3.

Komunikace s databází pak probíhá na základě posílání dat (write) a dotazování na se ně (query). To lze dělat mnoha způsoby. Mezi ně patří například software influx CLI, Chronograf či Grafana. Další možností jsou programovací jazyky s knihovnou InfluxDB. V době, kdy tato práce byla psána, tak bylo podporováno až 14 programovacích jazyků včetně Pythonu, Java, JavaScript či C++. Funkce těchto knihoven je v podstatě jen více uživatelsky přívětivé rozhraní, které posílá a dotazuje se na datové body. Ty jsou pak interpretovány na informace akceptovatelné InfluxDB v2 API.



The screenshot shows the InfluxDB Data Explorer interface. At the top, there is a header with 'Data Explorer' on the left and 'Switch to old Data Explorer' with a toggle switch on the right. Below the header, there are several controls: a 'Table' view selector, a 'CUSTOMIZE' button, a 'Local' dropdown menu, and a 'SAVE AS' button. The main area displays a table with the following columns: `_start`, `_stop`, `_time`, `_value`, `_field`, `_measurement`, `location`, and `sensor`. The table contains 11 rows of data, all representing temperature measurements from a sensor named 'bme280' at a location named 'home'. The values range from 26,14 to 26,18.

<code>_start</code>	<code>_stop</code>	<code>_time</code>	<code>_value</code>	<code>_field</code>	<code>_measurement</code>	<code>location</code>	<code>sensor</code>
2023-05-14 16:14...	2023-05-14 17:14...	2023-05-14 16:15...	26,14	temperature	measurements	home	bme280
2023-05-14 16:14...	2023-05-14 17:14...	2023-05-14 16:15...	26,15	temperature	measurements	home	bme280
2023-05-14 16:14...	2023-05-14 17:14...	2023-05-14 16:15...	26,16	temperature	measurements	home	bme280
2023-05-14 16:14...	2023-05-14 17:14...	2023-05-14 16:15...	26,15	temperature	measurements	home	bme280
2023-05-14 16:14...	2023-05-14 17:14...	2023-05-14 16:15...	26,16	temperature	measurements	home	bme280
2023-05-14 16:14...	2023-05-14 17:14...	2023-05-14 16:15...	26,16	temperature	measurements	home	bme280
2023-05-14 16:14...	2023-05-14 17:14...	2023-05-14 16:16...	26,17	temperature	measurements	home	bme280
2023-05-14 16:14...	2023-05-14 17:14...	2023-05-14 16:16...	26,18	temperature	measurements	home	bme280
2023-05-14 16:14...	2023-05-14 17:14...	2023-05-14 16:16...	26,18	temperature	measurements	home	bme280
2023-05-14 16:14...	2023-05-14 17:14...	2023-05-14 16:16...	26,18	temperature	measurements	home	bme280

Obrázek 3: Příklad záznamu InfluxDB databáze

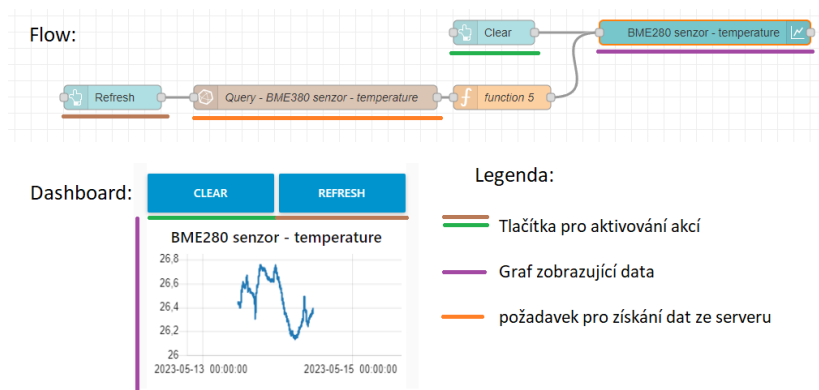
2.6. Node RED

Node RED je programovací nástroj fungující na bázi FBD [22]. Jeho vzhled můžeme připodobnit stavovému diagramu, kde každá z buněk má svoji funkčnost. Samotné programování pak funguje propojením několika buněk, kde každá buňka je již předdefinovaná (podmínka, smyčka, zobrazení informací na displeji, přijímání či posílání informací) a samotný uživatel jen pak upravuje parametry v buňkách připravené. Pokud je toho třeba, tak je možné vytvořit i buňku vlastní s vlastnostmi pak naprogramovanými v jazyce JavaScript. Další možností je vytvoření shluku buněk, které pak předefinujeme jako buňku jednu (velice podobné jako tvoření funkcí v klasickém programování). Jinak existuje mnoho palet (knihoven) rozšiřující základní buňky o všemožné funkcionality jako:

- komunikace se sociálními sítěmi,
- práce s cloudem (např. InfluxDB),
- posílání a přijímání informací přes různý software (např. MQTT, InfluxDB).

Průběh procesu naprogramovaný přes Node RED vypadá trochu odlišně než přes Python či C++. Podrobně je to popsáno literatuře [22], ale velice zkrátka kód každé buňky zpracovává informace mezi buňkami poslané. Tento kód běží pouze dokud nezpracuje celou příchozí informaci. Tato informace se pak nazývá zpráva (msg). Z pohledu jazyka JavaScript je zpráva objekt, obvykle obsahující set datových typů s určitými informacemi. Pro Node RED to obvykle je:

- Timestamp – časový údaj zprávy, často užíván při sběru naměřených dat,
- ID – kódové označení, bývá unikátní pro každou zprávu,
- Payload – samotná přenášená informace.



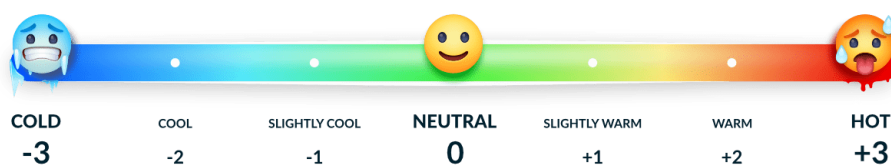
Obrázek 4: Příklad Flow kódu v Node RED se zobrazením dat z cloudu na Dashboard

2.7. Subjektivní pohodlí člověka

Člověk ze své podstaty má velmi subjektivní názor na to, jaké prostředí je pro něj komfortní. Proto je pohodlnost prostředí z technického hlediska obtížné kvantifikovat. I tak však byli pokusy takové měřítko vytvořit. Jeden z takových pokusů se později stal standardem EN ISO 7730:2005. Tento standart dělí komfortnost na objektivní a subjektivní proměnné¹. Celková komfortnost je pak shrnuta do jedné proměnné, a to konkrétně parametr PMV [-]. Ten bere v ohledu rovnováhu mezi aktuálním tepelným prouděním v lidském těle v daném prostředí s odpovídající úrovní komfortu ve vztahu k určité činnosti. Tento ukazatel je pak třeba brán v potaz při budování energeticky efektivních budov [23].



Obrázek 5: Faktory ovlivňující tepelnou pohodu [24]



Obrázek 6: Stupnice pro PMV [24]

¹ Tyto dvě kategorie obsahují: objektivní proměnné – teplota vzduchu, relativní vlhkost, rychlost vzduchu, střední radiační teplota subjektivní proměnné – metabolická aktivita, oblečení

Pro stanovení PMV je třeba relativně složitěho výpočtu (viz [25]). Principiálně je však rovnost docela jednoduchá. A to tedy že:

$$PMV = f(t_a, t_r, v_{ar}, p, Rh, M, W, I_d).$$

Rovnice 1: Symbolické znázornění veličin ovlivňujících PMV [25]

Veličiny ovlivňující PMV z Rovnice 1 jsou:

t_a – teplota vzduchu [°C] – lineární závislost s přímou úměrou,

t_r – povrchová teplota [°C],

v_{ar} – relativní rychlost vzduchu [m/s],

Rh – relativní vlhkost vzduchu [%] – lineární závislost s přímou úměrou,

p – tlak vzduchu [Pa] – lineární závislost a nepřímou úměrou,

M – intenzita lidského metabolismu [W/m²],

W – Intenzita mechanického záření [W/m²,

I_d – Míra izolace oblečením [m² · K/W].

Parametr PMV je standardem právě proto, protože je brán jako relativně dobré měřítko pro tepelnou komfortnost. Kromě něho však existují i jiná měřítka, jako například parametr DI (Diskomfort Index) [26]. Ten není sice tak přesný jako PMV, ale k jeho zjištění stačí pouze znalost teploty vzduchu – t_a [°C] a relativní vlhkosti vzduchu – Rh [%]. Bohužel DI lze aplikovat pouze pro teploty od 24 °C výše. Informace o procentu lidí z populace pociťující diskomfort pak vychází z Tab. 4. DI se počítá pak jako:

$$DI = t_a - 0.55(1 - 0.01Rh)(t_a - 14.5).$$

Rovnice 2: Výpočet parametru DI [26]

Tab. 4: Klasifikace tepelného vnímání člověka dle parametru DI [26]

Míra populace vnímající diskomfort [%]	Meze parametru DI [°C]
0	$18 \leq DI \leq 21$
10	$21 \leq DI \leq 24$
50	$24 \leq DI \leq 26$
100	$26 \leq DI \leq 26.7$
mimo všechny meze	$26.7 \leq DI$

3. Praktická část

Tato kapitola se věnuje přípravě a samotné implementaci bakalářské práce. Velký důraz je kladen na vysvětlení celkového propojení systému a na popsání jeho limitů. Celkové hodnocení je však až později v kapitole 4.

Kapitola 3.1 a 3.2 se věnují přípravě. Konkrétně přípravou se myslí výběr použité platformy (seznam možných platforem v kapitole 2.1), na které celý systém běží a výběru periférií, které se systémem budou komunikovat.

Kapitola 3.3 se soustředí na popsání propojení všech komponent (viz obrázek 12). Zde je vysvětleno, jaké programy byli využity při implementaci a jak se všechny data dostanou ze senzoru až k uživateli do mobilní aplikace.

Nakonec pak v kapitole 3.4 a 3.5 je vysvětleno použití ESP32 a Node RED. Node RED je obzvlášť důležité probrat, protože se v něm děje většina zpracování, zálohují se přes něj data na cloud InfluxDB a vytváří se komunikační tunel mezi uživatelem a uživatelským rozhraním (UI). V kapitole o ESP32 je popsán obvod propojení s jeho moduly (viz obrázek 11) a pak jen zkráceně průběh vnitřního programu.

3.1. Hardware zpracovávající data

Nejdříve je potřeba rozhodnout přes jakou platformu se sbíraná data budou zpracovávat. To pak následně vymezení senzory, které můžeme pro sběr dat použít a dále posílat. Kritéria pro výběr jsou:

- cena,
- interkompatibilita se senzory,
- možnost mnoha vstupů a výstupů pro možnou budoucí expanzi,
- schopnost připojení internetu pro posílání a stahování dat na cloud,
- možnost ovládání na dálku, tedy přes síť,
- určitá oblíbenost hardwaru, a tedy existence informací pro hledání problémů při implementaci.

S ohledem na výše zmíněná kritéria a na základě parametrů kontrolérů popsaných v Tab. 1 jsem zvolil ESP32. K tomu namísto dalšího mikrokontroleru byl použit standardní počítač připojený se ZigBee koordinátorem Sonoff 3.0 USB dongle plus. Důvod, proč není jádro systému pouze jedno je, že se mi nepodařilo implementovat všechny funkcionality do jednoho zařízení.

Z praktických důvodů jsem použil počítač značky Lenovo, který jsem měl k dispozici. Pro zlevnění systému by bylo možné použít i Raspberry Pi 3 model A+ (viz Tab. 1). Díky tomu, že protože programy ZigBee2MQTT a Node RED existují i na Linuxu, tak by byl přesun funkcionalit velice jednoduchý. Naštěstí však až na vysokou cenu jsou ostatní požadavky u počítače v rámci mezí. Část systému s ESP32 na druhou stranu požadavky splňuje skoro všechny perfektně (viz Tab. 1).



Obrázek 7: Foto hardwarového jádra

3.2. Výběr připojovaných zařízení

Aby systém měl možnost pracovat s prostředím kolem něj, tak se samozřejmě neobejde bez periférií. V mém případě je využíváno bezdrátových senzorů pohybu Sonoff ZigBee Motion Sensor – SNZB-03 a WOOX ZigBee senzor teploty a vlhkosti vzduchu – R7048. Dále jako dvojitá kontrola bylo připojeno k ESP32 přes I2C sběrnici kabelový senzor BME280 (snímá teplotu, vlhkost a tlak). K tomu navíc jsou k systému připojeny 3 žárovky IKEA ZigBee TRADFRI LED bulb E26/27 1000 lumen. Diagram celkového propojení je ukázán později na 0.

Jak je z výpisu produktů výše poznat, tak tento systém je schopný spolupracovat s bezdrátovými senzory různých značek. Důvodem právě využití ZigBee2MQTT viz kapitola 2.4.

Pokud by bylo třeba v budoucnu rozšíření o senzory další, tak technologie ZigBee je k tomu nejvhodnější, protože podporuje i tisíce zařízení pro profil ZigBee PRO (viz kapitola 2.2.3). A co se týče kabelových senzorů u ESP32, tak díky technologiím I2C bus, SPI bus a UART (viz kapitola 2.1.1), není limitem ani tak počet senzorů, ale schopnost mikrokontroleru se všemi perifériemi pracovat.



Obrázek 8: Bezdrátové periferie použité v realizaci

3.3. Popis celkové konstrukce systému

Celý systém je názorně popsán ve zjednodušeném diagramu na obrázku 12. Z legendy můžeme vyčíst, že systém má 3 ovládající zařízení, tedy zařízení zpracovávající, zobrazující, či měnící data.

Nejdříve něco k mikrokontroleru ESP32. Podrobné schéma připojení k jednotlivým pinům je pak na obrázku 11. Principiálně se jedná o jednoduché schéma. Existuje senzor BME 280 přes I2C bus připojený k mikrokontroleru. Stejně je připojen i IIC I2C OLED displej s rozměry 128 x 64. Data ze senzoru jsou pak zobrazeny na displeji a zároveň jsou posílány pře WiFi do InfluxDB cloudu.

Druhým zařízením je počítač značky Lenovo, který je připojen k Sonoff 3.0 USB dongle plus, na který je nahrán firmware pro ZigBee2MQTT. Samotný počítač pak má spuštěné 2 programy v pozadí (ZigBee2MQTT a Node RED). ZigBee2MQTT bere data ze senzorů (R7048 a SNZB-03) a ovládá stav IKEA žárovek z řady TRADFRI. Díky protokolu MQTT se pak stavy jednotlivých zařízení zveřejňují (publish) na lokální síti schované za identifikátorem daných zařízení. Node RED se pak stará o zpracování naměřených dat, jejich zálohování přes cloud a poskytuje rozhraní (přes Remote RED), kterým uživatel data může sledovat a provádět změny.

Poslední částí je mobilní zařízení, na kterém je přes Google play stáhnutá a nainstalovaná aplikace Remote RED. Přes tuto aplikaci je možné se připojit vzdáleně k UI Node RED (více v kapitole 3.5). Velice zjednodušeně celá komunikace funguje na principu SSH tunelu, který Node RED propojí se vzdáleným serverem, na který se může připojit mobilní zařízení (viz [27]). Velkou výhodou této konfigurace pak je, že je možno poslat pokyn i z druhé strany světa. Bohužel je zde ale i nevýhoda závislosti na serveru, který je bohužel zpoplatněn (viz [28]).

3.4. Implementace ESP32

3.4.1. Popis obvodu

Jak je ze schématu na obrázku 11 vidět, tak ESP32 má ještě spoustu dalšího prostoru pro připojení dalších modulů v budoucnu. Moduly, v tuto chvíli k ESP32 připojené, jsou tři, a to:

- OLED displej,
- BME 280,
- Reprodukční s LM386 zesilovačem.

OLED displej a BME 280 využívají I2C bus. Jelikož je sběrnice sdílená, tak se pro komunikaci každý modul volá přes svojí identifikační adresu. Pak se jen pomocí řady knihoven data dekodují na čitelné informace či se kódují pro následné posílání sběrnici. Jak senzor, tak displej mají velice malý odběr proudu (viz [29], [30]), což snižuje energetickou náročnost.

Poslední z bloků je možná trochu nadbytečný, ale na druhou stranu ukazuje na limity a problémy mikrokontroleru. Konkrétně tím myslím reproduktor s LM386 se zesilovačem. Tento blok v obvodu funguje jako informátor o tom, že systém správně funguje zahráním krátké znělky při ukončení inicializace. Tento modul by však mohl zastávat mnohem zajímavější funkce, jako je např. jednosměrná hlasová komunikace. Bohužel však kvůli malé paměti ESP32 není možno zahrát cokoliv komplexního. Zároveň kvůli periodickému posílání dat do cloudu přes WiFi (každých 10 sekund) dochází k velkému rušení způsobené změnou odběru proudu. Oba problémy by byli samozřejmě řešitelné úpravou obvodu, a to konkrétně například přidáním odděleného napájení k samotnému zesilovači a přidáním externí paměti ke kontroléru (viz [31]). Stále však to je jen částečné řešení přidávající možná až zbytečně komplexitu do jinak docela jednoduchého zapojení.



Obrázek 9: Fyzická podoba implementace s ESP32

3.4.2. Kód ESP32

Samotný kód lze najít v elektronické příloze s názvem Final_code_esp32.txt. Pro jeho tvorbu jsem se inspiroval příklady kódů nahrané společně s knihovny pro komunikaci s jednotlivými periferiemi a z [32], [33]. Užitý programovací jazyk je Arduino, což je v podstatě C++ s určitými změnami. Funkce kódu nahranou v ESP32 lze rozdělit do tří oddílů a to příprava, inicializace a smyčka.

V přípravě jsou vypsány všechny potřebné knihovny, nastavená znělka pro reproduktor, příprava struktur pro práci s BME280, OLED displejem, zesilovačem LM386 a cloudem InfluxDB.

Inicializace se stará o správné spojení se všemi periferiemi. V případě problému s připojením k WiFi se program zastaví do té doby, než se spojení naváže. Pokud dojde k problému s připojením na I2C sběrnici, či pokud nedojde k připojení ke cloudu, tak se program zastaví. Pokud vše proběhlo, jak má, tak se spustí znělka hrající přes reproduktor.

Smyčka je pak jen opakování 3 činností. A to sběr dat z BME280, zobrazení dat z BME280 na OLED displeji a posílání dat do InfluxDB cloudu.

3.5. Node RED v implementaci

3.5.1. Popis způsobu práce v Node RED

Node RED má v mé implementaci možná i největší roli, protože propojuje ostatní systémy a dává možnost interaktivní komunikace. Jak je z diagramu na obrázku 12 vidět, tak Node RED funguje jako program spuštěný na pozadí počítače. Pro práci s ním se využívá internetového prohlížeče, kde probíhá veškeré programování. Práce s ním je trochu netradiční, protože využívá FBD architektury (viz 2.6). Místo nahrávání knihoven se nahrávají takzvané palety, přes které se získávají potřebné buňky, a ze kterých se pak tvoří celý kód. Další nezvyklostí je, že kód nefunguje na základě pořadí řádků, tak jak tomu je v pythonu, c++ a tak podobně, ale na základě pořadí buněk posílající a přijímající zprávy. Pro velkou řadu lidí může být vizualizace s FBD příjemnější, než normální programovací IDE což může způsobit rychlejší a přehlednější práci s kódem.

3.5.2. Popis kódu v Node RED

Samotný kód lze najít v elektronické příloze v souboru Node_RED_flow.txt. Pro popis kódu pro node RED bych upozornil, že posílání zpráv mezi buňkami probíhá na více místech zároveň, a proto neexistuje v pravém slova smyslu začátek či konec kódu.

První část je věnovaná sběru dat ze senzorů pohybu Sonoff a senzoru měření teploty/vlhkosti WOOX. Pro každý senzor je daný určitý topic určený právě přes ZigBee2MQTT, který musí být specifikován v nastavení buňky informaci sbírající. Tyto nasbírané informace musí být upraveny do lepšího formátu pro další práci s nimi. Funguje to poté tedy tak, že každá zpráva je ucelený balíček informací o jednom měření z jednoho senzoru. Data jsou pak nakonec nahrána do uživatelského rozhraní, kde jsou ukázány pro uživatele k okamžitému nahlédnutí.

Kvůli mému výběru senzorů není možné speciálně vyžádat informace dotazem, ale spíše se pasivně čeká, než se data za určitý časový interval přijmou. Pro určitou ilustraci však je v kódu implementován prvek, který takovou zpětnou vazbu podporuje, a to je TRADFRI LED žárovka z IKEA. Ta na rozdíl od dvou výše zmíněných senzorů je schopná stát se routerem v ZigBee síti a je možno na základě požadavku ji rozsvítit/zhasnout či zvýšit/snížit jas. Proto je v UI několik tlačítek takovou změnu umožňující. Informace o požadavku ze strany uživatele jsou pak posílány do buňky s MQTT,

kde zpráva je interpretovaná přes ZigBee2MQTT do LED žárovky. Konkrétní formulace takového požadavku lze najít na stránkách pro ZigBee2MQTT [19] (konkrétně v záložce pro žárovky TRADFRI led).

Jak už jsem více v této kapitole uvadl, tak je možné naměřená data sledovat přes uživatelské UI. Aby se však data zálohovali v případě restartu programu Node RED, tak jsou zároveň také posílána na InfluxDB cloud. Data uložená na cloudu si pak uživatel může vyžádat a prohlédnout ve stejném UI v grafech.

Poslední částí kódu v Node RED je funkcionalita sdílení UI tak, že je možno se k němu vzdáleně připojit přes mobilní zařízení. Z hlediska programování to není složité, protože stačí jen volně do prostředí v Node RED vložit buňku s Remote RED. V této buňce se jen nastaví, co se má sdílet a Node RED vše ostatní už si připraví sám. Na pozadí dojde k započatí SSH komunikace a pak se přes určitý port informace posílají do mobilního zařízení a zpět.

3.5.3. Popis uživatelského rozhraní (UI)

Uživatelské rozhraní je v mém řešení velmi minimalistické. V podstatě máme rozdělené UI do 4 sekcí, kde každá sekce má svojí specializaci. Na počítači můžeme vidět všechny sekce najednou, zatímco na mobilních telefonech se musí posouvat nahoru a dolů pro přesunutí se ze sekce do sekce.

V první sekci máme možnost ovládat zapnutí a vypnutí světla a je možno měnit jas žárovek. V druhé sekci máme zobrazení důležitých dat zálohované přes cloud InfluxDB. Nakonec sekce 3 a 4 jsou pro doplňující informace o aktuálním stavu bezdrátových senzorů. Z těchto sekcí pouze sekce 2 má schopnost i po restartu Node RED si stáhnout data o předchozích měření. Vzhled uživatelského rozhraní lze vidět na obrázku 14.

4. Diskuse nad výsledky

4.1. Srovnání s komerčními platformami

V této kapitole dojde k zamyšlení nad úspěšností této práce vůči komerčním platformám pro automatizaci domácnosti (porovnávané komerční platformy viz kapitola 2.3). Poukáže se na benefity ale i na slabiny mého systému. Po celkovém srovnání bude doporučen ten systém, který přináší nejvíce výhod.

4.1.1. Výhody

Největší výhodou mého systému je určitě volnost přizpůsobení požadavkům uživatele. Je např. velká volnost v typu periferií, které se dají použít. Je pak možno použít ve-lice specializované moduly neslučitelné s komerčním systémem. V případě mé realizace do toho spadá např. OLED displej či senzor BME280 připojený k ESP32.

Další výhodou je široký výběr použitelných ZigBee zařízení díky použití ZigBee2MQTT. Více k tomuto softwaru v kapitole 2.4.

Také určitě je důležité zmínit cenovou stránku věci, kde v určitých případech bude ručně vytvořené řešení levnější² než komerční, pokud se nezapočítá lidská práce. Cena komerčního řešení v sobě zahrnuje např. investice společností do vývoje, stálé výdaje na zlepšování produktu či čistě jen prémiovost značky (například Apple produkty).

Poslední výhodou mého řešení je možnost úprav UI. Díky tomu, že se v podstatě jedná o html stránku vytvořenou přes Node RED, tak je možné si ji upravit zcela dle uživatelských preferencí.

4.1.2. Nevýhody

Hlavní nevýhodou je určitě množství stráveného času a potřebná technická zdatnost. Tím myslím množství času stráveného samotným programováním, či jen vůbec znalost softwaru (ZigBee2MQTT, ZigBee, MQTT, Node RED, práce s firmware, programovací jazyky) a hardwaru (ESP32, typy sběrnic, základy obvodů).

² Slovo levnější je při přirovnání této práce s komerčním řešením diskutabilní, protože součástí mého systému je počítač Lenovo s cenou přes 20 000 Kč. Pokud by byl však nahrazen Raspberry Pi 3, tak věřím, že cena by byla opravdu nižší.

Dále je potřeba mít silnou vůli učit se novým věcem a nevzdávat se při setkání se s problémy. Každý z komerčních systémů v kapitole 2.3 je v menší či větší míře přivítivým na práci s ním, protože byl pro to již dlouho optimalizován. Oproti tomu v mém případě bylo potřeba od základu vymyslet, jak se data budou sbírat, zálohovat, zpracovávat a zobrazovat.

Navíc dost času je nevyhnutelně věnováno věcem, které se nakonec vůbec se nedostali do finálního řešení. V případě mé realizace konkrétní příklady neúspěšných systémů bylo např. využití virtuálního OS pro samotné jádro systému nebo použití ZigBee komunikace přes ESP32 (přes externí modul XBee).

4.1.3. Srovnání

Pokud bych vzal v úvahu veškeré funkcionality mého systému a porovnal je i s kteroukoliv komerční platformou, tak můj systém je objektivně prostě horší. Můj systém nebyl vyzkoušen miliony uživatelů, neposkytuje hlasovou komunikaci a celkově není uživatelsky tolik přívětivý.

Pokud jediné, co uživatel od automatizaci domácnosti chce, je funkční a celkově dobrý systém, tak bych doporučil platformu SmartThings. Podporuje velkou škálu zařízení, komunikace funguje přes WiFi, ZigBee či Z-wave, má skvělé ovládání hlasem a má dobře propracovanou aplikaci pro ovládání [17].

Zvolení mého řešení dává smysl teprve v případě, že se spíše jedná o koníček, kterým se člověk věnuje než výběr plnohodnotného systému. To anebo pokud v automatizaci domácnosti je potřeba velice speciální funkce, které by jen těžko byla uskutečnitelná v uzavřených systémech komerčních řešení.

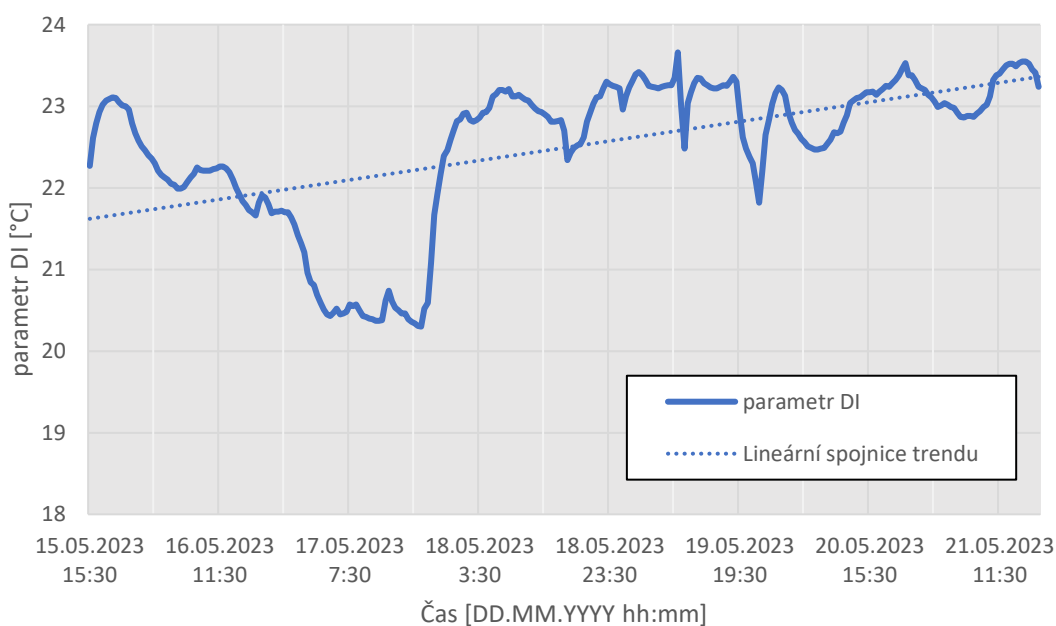
V případě tvorby budoucích prací s podobným cílem by dle mého názoru bylo moudré použít již předpřipravenou platformu jako je například Home Assistant (viz např. tento projekt [2]). Ten spojuje open source platformu s širokou základnou uživatelů, které už platformu využívají, a tak dopomáhají k jejímu zlepšení.

4.2. Zhodnocení hodnot změřené senzory

Cílem každého systému pro automatizaci domácnosti je zvýšení pohodlí. V případě této bakalářské práce je toho dosahováno dlouhodobým monitorováním prostředí kde jedinec tráví velkou část svého života neboli jeho pokoj. Pro objektivní posouzení komfortnosti se pak může využívat parametru PMV či DI (viz kapitola 2.7).

Z veličin ovlivňující tepelný komfort byli v místnosti měřené tři, a to teplota, tlak a vlhkost vzduchu. Z těchto údajů lze zjistit parametr DI, který je pak zobrazen grafem pro období sedmi dnů (viz obrázek 10).

Z datových bodů tohoto grafu a zároveň z informací z Tab. 4. můžeme vyvodit, že kromě 17.5. by v ostatní dny 10 % z populace považovalo prostředí za nekomfortní. Větší nekomfortnost potvrzují i z mého subjektivního pozorování. Důvodem zhoršení podmínek pro komfort může být příchod léta, ale také možná špatná cirkulace vzduchu. Jako řešení se nabízí přidání zařízení pro cirkulaci vzduchu (ventilátor), avšak to přidává značný hluk a odběr energie. Druhou možností by mohlo být chytré otevírání okna (viz [34]). V případě ovládání okna by bylo možné využít již existující infrastruktury v podobě mikrokontroleru ESP32.



Obrázek 10: Graf závislosti parametru DI na čase (15.5 – 21.5.)

5. Závěr

Cílem této práce bylo prozkoumání možností automatizace domácnosti a na základě toho sestavit platformu, která by mohla alespoň z části konkurovat platformám komerčním.

V teoretické části jsem se nejdříve věnoval průzkumu trhu studiem různých mikrokontrolerů pro IoT a bezdrátových technologií používaných v IoT. Soustředil jsem se na vysvětlení výhod každé z platform, abych později mohl se správně pro jednu rozhodnout. Prozkoumal jsem možnosti platform nabízené komerčními společnostmi, abych je mohl později porovnat s řešením v praktické části. Také jsem v krátkosti zmínil několik softwarů, které byli v praktické části využity. Nakonec jsem poukázal na metody pro definování tepelného komfortu.

V Praktické části jsem se nejprve věnoval výběrem správné platformy, použité k realizaci sběru a zpracování dat, a výběru periférií pro ně. Soustředil jsem se na vysvětlení mých důvodů, proč bylo zvoleno zrovna toto řešení. Pak jsem představil, jak celý systém spolu komunikuje. Důraz byl kladen na vysvětlení částí systému, které byli kritické pro jeho běh. Tyto části jsem pak ještě více rozvedl v posledních dvou kapitolách praktické části.

Nakonec v kapitole pro diskusi, jsem srovnal řešení systému této bakalářské práce s možnostmi nabízené komerčními platformami. Zmínil jsem se o benefitech, který můj systém přináší včetně: lepšího přizpůsobení uživatelským požadavkům či nižších nákladů na určité moduly. S tím jsem však zmínil i nedostatky mého systému, a nakonec jsem zhodnotil, že je pořád lepší spolehnout se na určitou platformu jako je například SmartThings či Home Assistant, než se pokoušet o systém vlastní. Na závěr jsem prozkoumal výsledky z měřené místnosti. Porovnáním těchto výsledků s mým subjektivním pocitem vyhodnocuji, že se můj subjektivní pocit komfortu neshoduje s 90% populace. Vyhodnotil jsem, že pro zvýšení komfortu by bylo vhodné zvýšit cirkulaci vzduchu přidáním ventilátoru či chytrým ovládním okna přes ESP32.

6. Bibliografie

- [1] *Global Smart Home Automation Market Size* [online]. In: . [cit. 2023-05-18]. Dostupné z: <https://www.researchandmarkets.com/reports/5645777/global-smart-home-automation-market-size-share>
- [2] PILÍK, Martin. *Centrální domácí hub*. Praha, 2020. Dostupné také z: <http://hdl.handle.net/10467/90115>. Diplomová práce. ČVUT, Fakulta Elektrotechnická, obor Elektroniky a komunikace. Vedoucí práce Janíček Vladimír.
- [3] Use the GPIO pins of the ESP32. In: *UPesy* [online]. 2020 [cit. 2022-12-08]. Dostupné z: <https://www.upesy.com/blogs/tutorials/esp32-pinout-reference-gpio-pins-ultimate-guide>
- [4] *Raspberry Pi Pico board WH* [online]. In: . [cit. 2023-05-13]. Dostupné z: <https://deskpi.com/products/raspberry-pi-pico-w-board>
- [5] *Arduino UNO* [online]. In: . [cit. 2023-05-18]. Dostupné z: https://rpishop.cz/arduino-uno/1356-arduino-uno-r3-7630049200050.html?gclid=Cj0KCQjwmZejBhC_ARIsAGhCqnfVsCGQMT EPPmkfAlURnZvrOZFcsvA_IWUdd5MD_c3qzuthCkQ1CiEaAmnAEALw_wcB
- [6] *ATMega328P* [online]. In: . [cit. 2023-05-18]. Dostupné z: https://www.laskakit.cz/atmel-atmega328p-pu--dip-28/?gclid=Cj0KCQjwmZejBhC_ARIsAGhCqndwAHC0-NK09InbYgpIVFyjLQhCq8CLY68TdMbsQqKtt2FwBwRnBVQaAhv2EALw_wcB#relatedFile
- [7] *BeagleBone Black* [online]. In: . [cit. 2023-05-18]. Dostupné z: <https://cz.farnell.com/seeed-studio/102110420/beaglebone->

black/dp/3520081?gclid=Cj0KCQjwmZejBhC_ARIsAGhCqnfzm308lXaVY05dOxGW97mhndjkseg50mTCnSUI5KUtRSJnyvTkdcAahmkEALw_wcB&mckv=s_dc|pcrid|621552844967|plid||keyword||match||slid||product|3520081|pgri d|138503362942|ptaid|pla-890193361835|&CMP=KNC-GCZ-GEN-SHOPPING-PLA&gross_price=true&gad=1

- [8] *Raspberry Pi 3 Model A+ Review* [online]. In: . [cit. 2023-05-22].
Dostupné z: <https://www.pcmag.com/reviews/raspberry-pi-3-model-a-plus>
- [9] KL, Arun. *Five Easiest Ways to Connect Raspberry Pi Remotely* [online].
In: . [cit. 2023-05-20]. Dostupné z: <https://thesecmaster.com/five-easiest-ways-to-connect-raspberry-pi-remotely-in-2021/>
- [10] WARWICK, Terry, Matt WOJCIAKOWSKI, Sara CLAY a Riddhi AMESER. *Raspberry Pi 2 & 3 Pin Mappings* [online]. In: . [cit. 2023-05-12].
Dostupné z: <https://learn.microsoft.com/en-us/windows/iot-core/learn-about-hardware/pinmappings/pinmappingsrpi>
- [11] ESP32-DevKitC, vývojová deska s modulem ESP-WROOM-32D. In:
Hadex [online]. [cit. 2022-12-08]. Dostupné z:
<https://www.hadex.cz/img/zbozi/m432n.jpg>
- [12] *Raspberry Pi 3 Model A+* [online]. In: . [cit. 2023-05-12]. Dostupné z:
https://rpishop.cz/4280-large_default/raspberry-pi-3-model-a.jpg
- [13] KARIMI, Khaoula a Salah KRIT. *Systems and technologies for Smart Homes/Smart Phones: A study and comparison* [online]. 7 [cit. 2023-05-10].
Dostupné z: doi:<https://dl.acm.org/doi/10.1145/3234698.3234706>
- [14] HORYACHYY, Oleh. *Comparison of Wireless Communication Technologies used in a Smart Home* [online]. Švédsko - Karlskrona, 2017, 73 s. [cit. 2023-05-10]. Dostupné z: <http://www.diva-portal.org/smash/get/diva2:1118965/FULLTEXT02.pdf>. Diplomová práce.

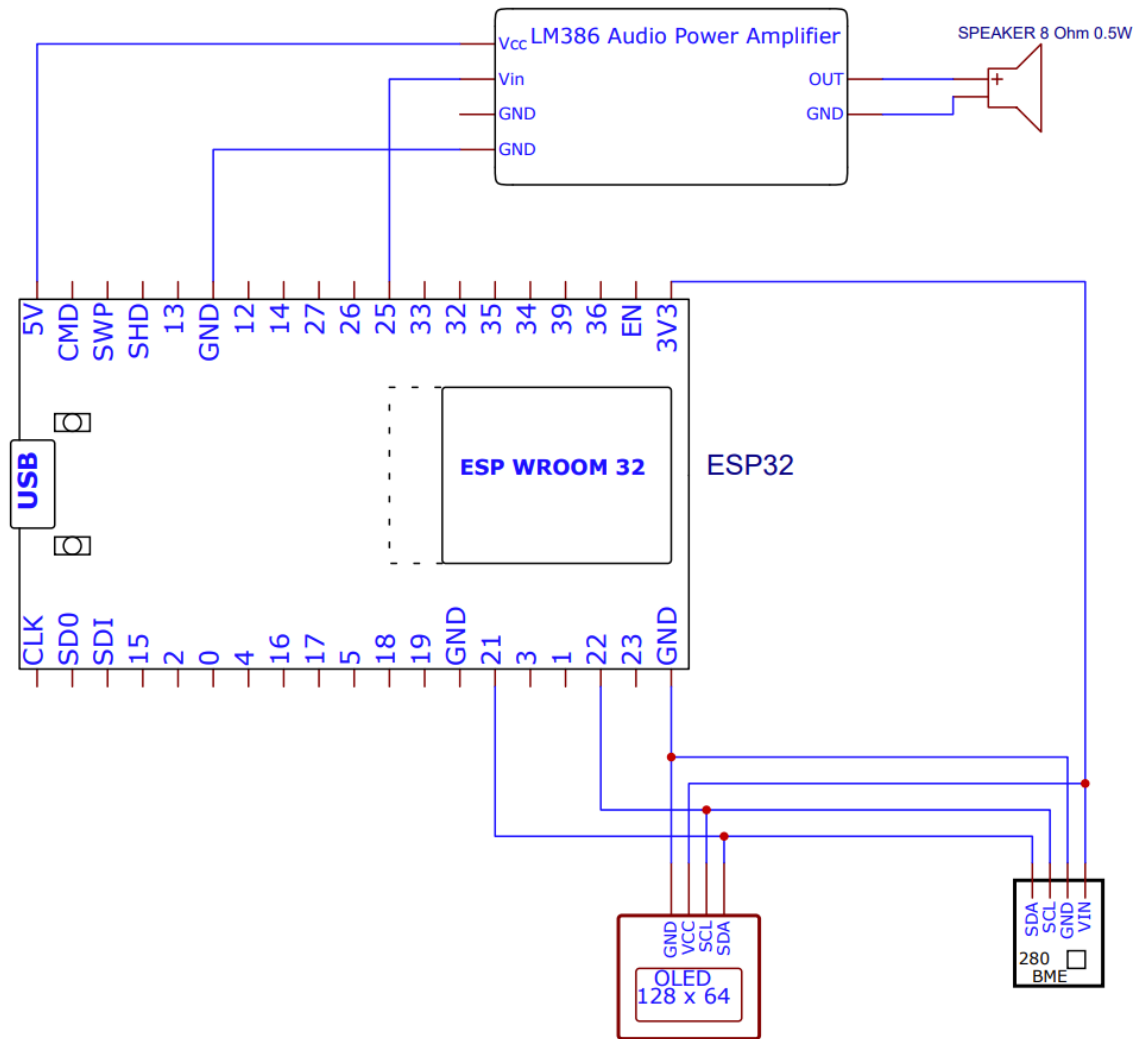
Faculty of Computing Blekinge Institute of Technology - Karlskrona Sweden.
Vedoucí práce Prof. Kurt Tutschku.

- [15] *BLE description* [online]. In: . [cit. 2023-05-19]. Dostupné z:
<https://www.makeuseof.com/what-is-ble-bluetooth-low-energy/>
- [16] SHUAIB, Khaled, Mariam ALNUAIMI, Mohammed BOULMALF, Imad JAWHAR, Farag SALLABI a Abderrahmane LAKAS. *Performance Evaluation of IEEE 802.15.4: Experimental and Simulation Results* [online]. 10 [cit. 2023-05-19]. Dostupné z: doi:10.4304/jcm.2.4.29-37
- [17] DIAZ, Maria. *The best home automations systems: Put your home on auto-pilot* [online]. In: . [cit. 2023-05-19]. Dostupné z:
<https://www.zdnet.com/home-and-office/smart-home/best-home-automation-system/>
- [18] KANTERS, Koen. Zigbee2MQTT list of supported devices. In: *Zigbee2MQTT* [online]. [cit. 2023-04-29]. Dostupné z:
<https://www.zigbee2mqtt.io/supported-devices/>
- [19] KANTERS, Koen. *ZigBee2MQTT* [online]. In: . [cit. 2023-05-14]. Dostupné z: <https://www.zigbee2mqtt.io/>
- [20] KANTERS, Koen. ZigBee2MQTT. In: *GitHub* [online]. [cit. 2023-05-14]. Dostupné z: <https://github.com/Koenkk/zigbee2mqtt>
- [21] InfluxDB. In: *Wikipedia* [online]. 2001 [cit. 2023-05-14]. Dostupné z: <https://en.wikipedia.org/wiki/InfluxDB>
- [22] HAGINO, Taiji. *Practical Node-RED Programming: Learn Powerful Visual Programming Techniques and Best Practices for the Web and IoT* [online]. 1. Birmingham UK: Packt Publishing, Limited, 2021 [cit. 2023-05-12]. ISBN 9781800207660. Dostupné z: <https://ebookcentral.proquest.com/lib/cvut/detail.action?docID=6525477>

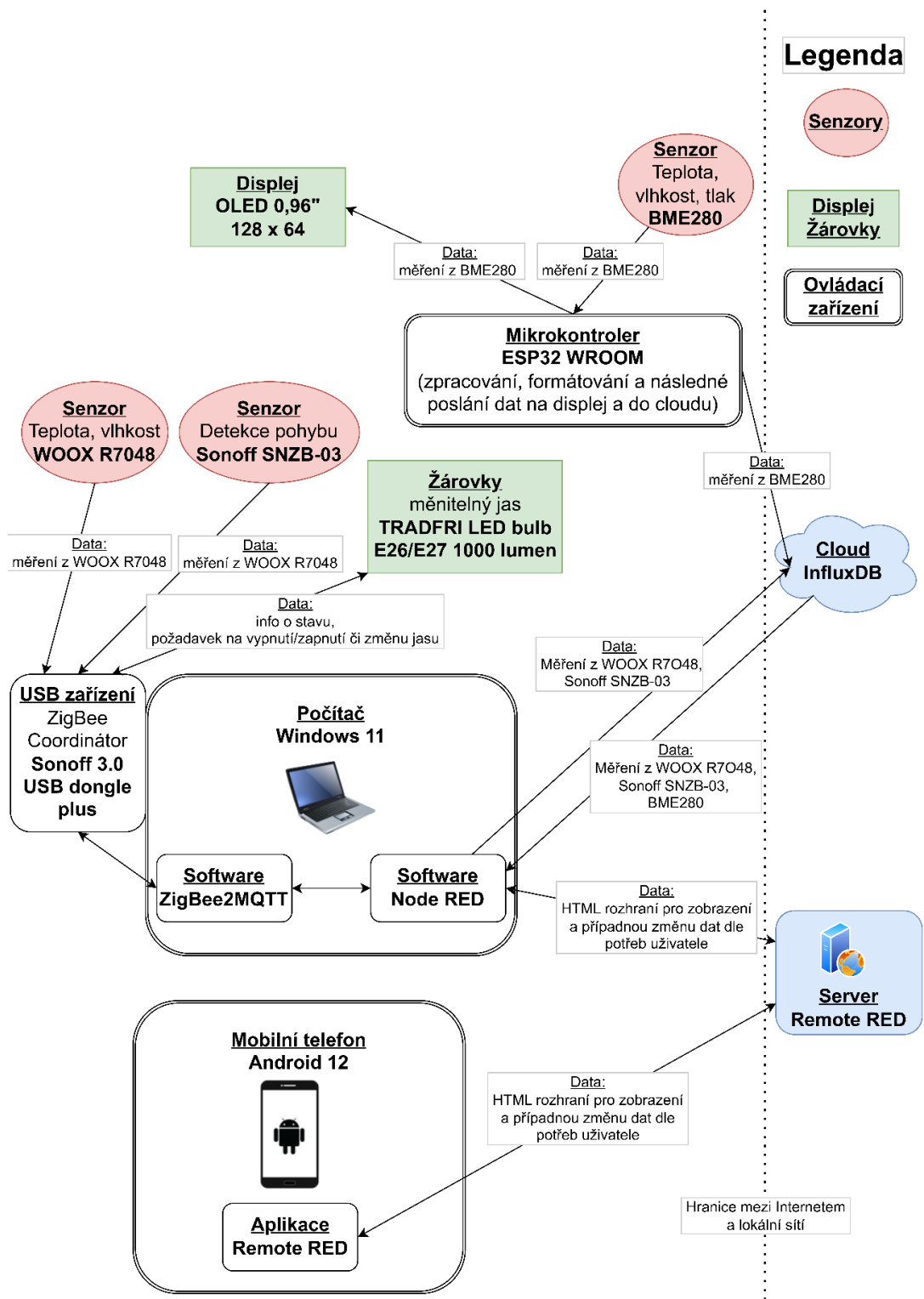
- [23] SALAMONE, Francesco, Lorenzo BELUSSI, Cristian CURRò et al. Integrated Method for Personal Thermal Comfort Assessment and Optimization through Users' Feedback, IoT and Machine Learning: A Case Study †. *Sensors* [online]. 2018, **18**(5), 18 [cit. 2022-12-07]. ISSN 1424-8220. Dostupné z: doi:10.3390/s18051602
- [24] KUMAR, Praveen. Role of CFD in Evaluating Occupant Thermal Comfort. In: *Linked in* [online]. 2003 [cit. 2022-12-08]. Dostupné z: <https://www.linkedin.com/pulse/role-cfd-evaluating-occupant-thermal-comfort-sandip-jadhav/>
- [25] WU, Zeqing, Yunfei MU, Shuai DENG et al. Towards comfortable and cost-effective indoor temperature management in smart homes: A deep reinforcement learning method combined with future information. *Energy and Buildings* [online]. 2022, **275**, 16 [cit. 2022-12-08]. ISSN 03787788. Dostupné z: doi:10.1016/j.enbuild.2022.112491
- [26] Evaluating thermal comfort conditions and health responses during an extremely hot summer in Athens. *Building and Environment* [online]. 2011, **46**(2), 6 [cit. 2023-05-22]. Dostupné z: doi:10.1016/j.buildenv.2010.07.026
- [27] *Terms of use of Remote-RED* [online]. In: . [cit. 2023-05-22]. Dostupné z: <https://www.remote-red.com/en/terms/>
- [28] *Pricing of Remote RED* [online]. In: . [cit. 2023-05-22]. Dostupné z: <https://www.remote-red.com/en/pricing/>
- [29] *BME280 datasheet* [online]. In: . [cit. 2023-05-22]. Dostupné z: https://www.laskakit.cz/user/related_files/bst-bme280.pdf
- [30] *IIC I2C OLED displej 0,96" - 128 x 64* [online]. In: . [cit. 2023-05-22]. Dostupné z: <https://dratek.cz/docs/produkty/0/114/1487765029.pdf>

- [31] SANTOS, Rui. ESP32: Guide for MicroSD Card Module using Arduino IDE. In: *Random Nerd Tutorials* [online]. 2013 [cit. 2022-12-08]. Dostupné z: <https://randomnerdtutorials.com/esp32-microsd-card-arduino/>
- [32] CHOUDHARY, Ashish. *DIY ESP32 Based Audio Player* [online]. In: . [cit. 2023-05-23]. Dostupné z: <https://circuitdigest.com/microcontroller-projects/esp32-based-audio-player>
- [33] *ESP32 communication with InfluxDB* [online]. In: . [cit. 2023-05-23]. Dostupné z: <https://randomnerdtutorials.com/esp32-influxdb/>
- [34] Intelligent Opening and Closing System of Doors and Windows. *IOP Conference Series: Earth and Environmental Science* [online]. 2019, **252**, 8 [cit. 2023-05-21]. ISSN 1755-1315. Dostupné z: doi:10.1088/1755-1315/252/2/022010

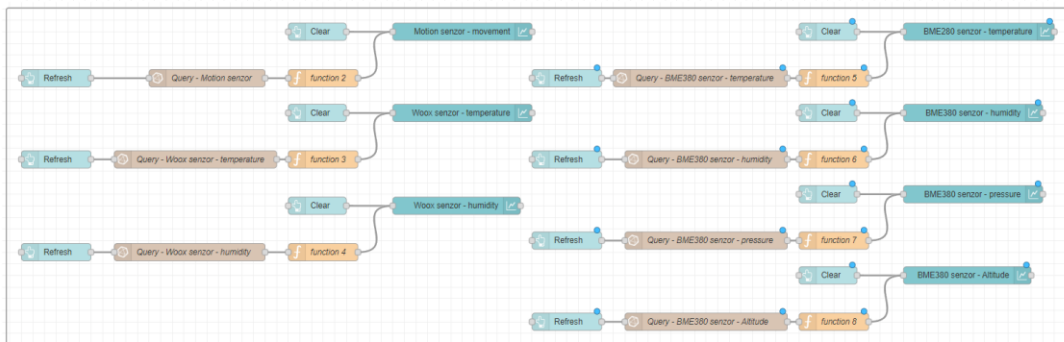
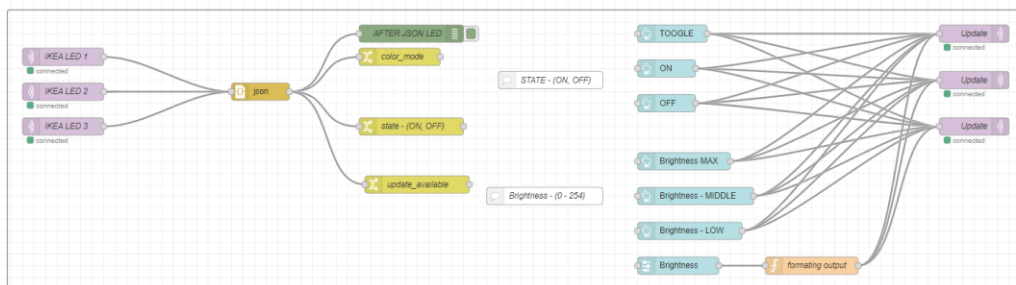
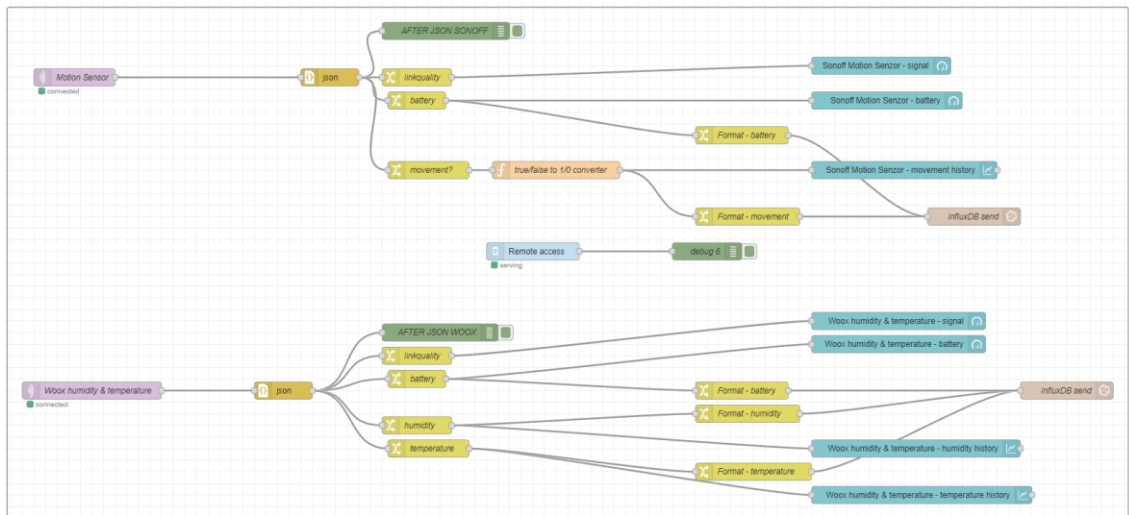
Přílohy



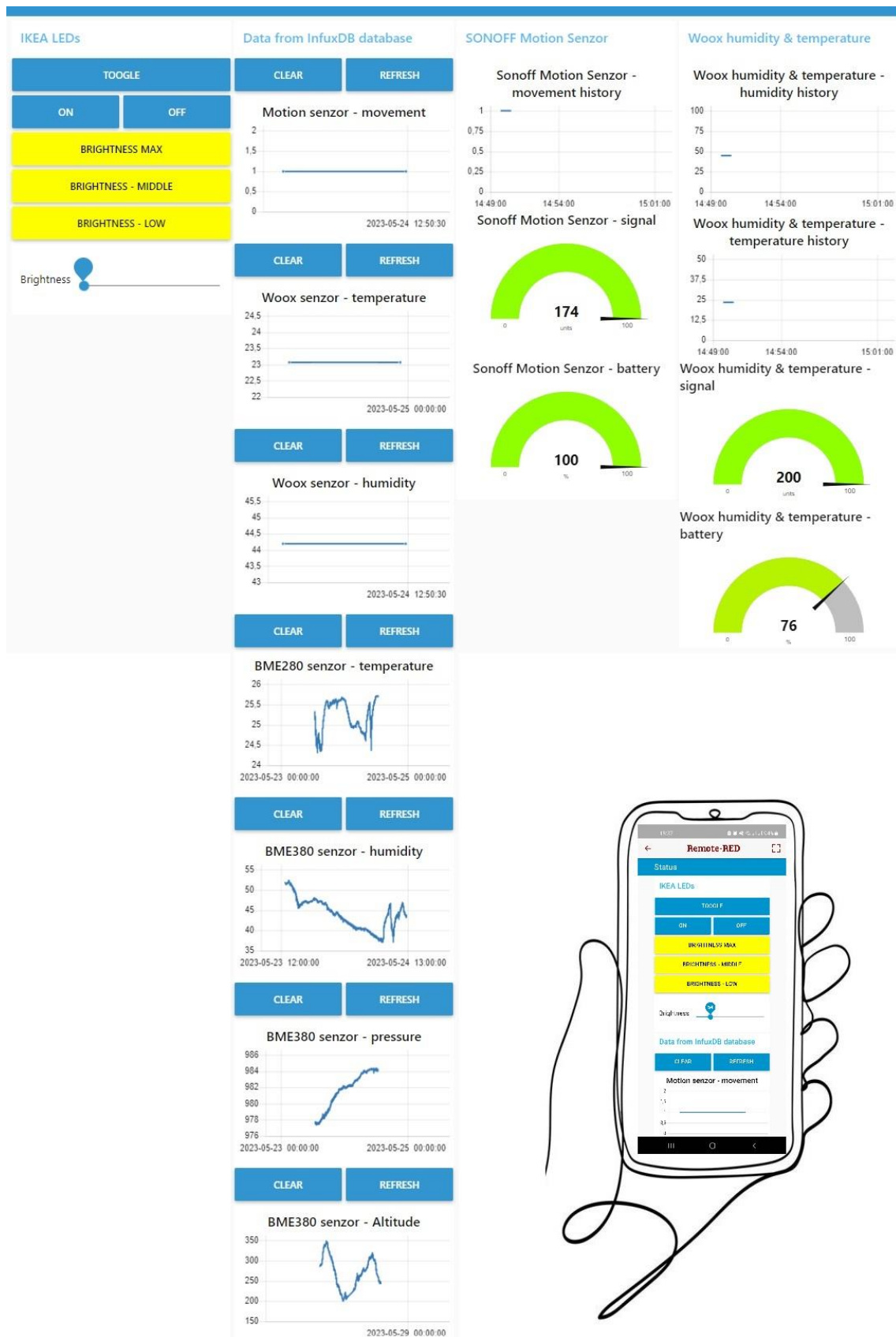
Obrázek 11: Schéma obvodu s ESP32



Obrázek 12: Diagram propojení systému



Obrázek 13: Schéma kódu v Node RED



Obrázek 14: Uživatelské rozhraní (UI) na počítači/telefonu