

Bakalárska práca



České
vysoké
učení technické
v Praze

F3

Fakulta elektrotechnická

Senzor kvality vzduchu v domácnosti

Jakub Zelinka

Školitel: Ing. Vladimír Janíček, Ph.D.

Odbor: Kybernetika a robotika

Máj 2024

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Zelinka**

Jméno: **Jakub**

Osobní číslo: **492377**

Fakulta/ústav: **Fakulta elektrotechnická**

Zadávací katedra/ústav: **Katedra měření**

Studijní program: **Kybernetika a robotika**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Senzor kvality vzduchu v domácnosti

Název bakalářské práce anglicky:

Home air quality sensor

Pokyny pro vypracování:

- 1) Vytvořte rešerši věnovanou principům měření kvality vzduchu, popište sledované veličiny a princip jejich měření.
- 2) Navrhněte zařízení pro měření kvality vzduchu v domácích podmínkách sestávající z centrální jednotky a podružných sensorových jednotek. Zařízení vybavte potřebnými sensory, uvažujte případně mobilní provedení. Vytvořte potřebné SW vybavení. Využijte ESP32 platformu. Propojte s komunitními projekty typu HA nebo OpenHab.
- 3) Zařízení sestavte a ověřte jeho parametry.
- 4) Porovnejte dosažené parametry s komerčními řešeními.

Seznam doporučené literatury:

- 1) Measurement and Sensor Systems: A Comprehensive Guide to Principles, Practical Issues and Applications, Koch A., ISBN 9783031158698.
- 2) ESP32-C3 Wireless Adventure: A Comprehensive Guide to IoT, Espressif, online.

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. Vladimír Janíček, Ph.D. katedra mikroelektroniky FEL

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

Datum zadání bakalářské práce: **15.02.2024**

Termín odevzdání bakalářské práce: **24.05.2024**

Platnost zadání bakalářské práce:

do konce letního semestru 2024/2025

Ing. Vladimír Janíček, Ph.D.
podpis vedoucí(ho) práce

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

Podakovanie

Týmto by som sa chcel poďakovať môjmu vedúcemu práce Ing. Vladimírovi Janičkovi, Ph.D. z katedry mikroelektroniky za jeho odbornú pomoc, ktorú mi v priebehu vypracovania tejto bakalarskej práce poskytol, a tiež sa mu chcem poďakovať za jeho vytrvalý prístup.

Prehlásenie

Prehlasujem, že som predloženú prácu vypracoval samostatne, na základe získaných poznatkov, a že som uviedol všetkú použitú literatúru.

V Prahe dňa 19.5.2024

.....

Abstrakt

Zdravie je u väčšiny ľudí na prvom mieste, a vzduch, ktorý dýchame, ovplyvňuje naše zdravie. Aby sme monitorovali zložky vzduchu, v ktorom sa nachádzame, rozhodol som sa zostrojiť zariadenie na monitorovanie kvality ovzdušia v domácnosti. V tejto práci som sa zameril na meranie veličín teploty, vlhkosti, CO₂ a TVOC. Obzvlášť zvýšená hodnota CO₂ je veľmi škodlivá a preto na ňu budem upozorňovať.

Kľúčové slová: senzor, ovzdušie, zdravie, ESP32, WiFi, aplikácia

Školiteľ: Ing. Vladimír Janíček, Ph.D.
Praha, ČVUT, fakulta: elektrotechnická,
katedra: mikroelektroniky, Technická
1902/2

Abstract

Health is a top priority for most people. The air we breathe affects our health. To monitor the components of the air around us, I decided to build a device for monitoring indoor air quality. In this project, I focused on measuring temperature, humidity, CO₂, and TVOC levels. Particularly, an increased level of CO₂ is very harmful, and therefore, I will emphasize it.

Keywords: Sensor, air, health, ESP32, WiFi, application

Title translation: Home air quality monitor

Obsah

1 Úvod	1		
1.1 Motivácia	1		
1.2 Formulácia problému	1		
1.3 Cieľ práce	1		
2 Prieskum trhu	3		
2.1 Aqara TVOC Air Quality Monitor (AAQS-S01)	3		
2.2 Detektor Honeywell Home Resideo R200C2-E	4		
2.3 iQtech SmartLife WiFi AirBox01	5		
2.4 AirThings Wave plus	6		
2.5 Záver prieskumu trhu	7		
3 Návrh senzora - HARDWARE	9		
3.1 Rozbor zariadenia	9		
3.2 Schéma zariadenia	9		
3.3 Výber komponent	10		
3.3.1 Mikrokontrolér	10		
3.3.2 Sensory	11		
3.3.3 Displej	13		
3.3.4 Bzučiak	13		
3.3.5 Akumulátor	14		
3.3.6 Záver výberu komponent	15		
3.4 Komunikácia medzi komponentami	16		
3.4.1 Porovnanie komunikačných periférií	16		
3.4.2 Komunikačné rozhranie SPI	16		
3.4.3 Komunikačné rozhranie I2C	17		
3.5 Schématické zapojenie zariadenia	18		
3.5.1 Napájanie cez USB-C 2.0	18		
3.5.2 Nabíjanie LiFePO4 akumulátoru	19		
3.5.3 Vedenia napätia z batérie alebo zdroja	20		
3.5.4 Zapojenie modulov senzorov	21		
3.5.5 Zapojenie bzučiaka	22		
3.5.6 Zapojenie displeja	23		
3.6 Výroba osadeného plošného spoju	23		
4 Návrh senzora - SOFTWARE	25		
4.1 Vlastnosti a spôsob prevedenia	25		
4.1.1 Meranie veličín zo senzorov	26		
4.1.2 Komunikácia s Cloudom, zobrazovanie dát online a posielanie upozornení	26		
4.1.3 Grafické zobrazovanie dát na displeji	27		
4.1.4 Zvuková signalizácia, aktualizácia času, stavu batérie a optimalizácia spotreby	28		
4.2 Vývojové diagramy	30		
5 Testovanie	33		
6 Záver	37		
6.1 Postup zostrojenia	37		
6.2 Obmedzenia práce	38		
6.3 Zhodnotenie cieľov a prínos práce	38		
6.4 Odporúčanie na rozšírenie	39		
6.5 Osobné zhodnotenie	39		
Literatúra	41		
Použité skratky	43		

Obrázky

2.1 Dizajn zariadenia AAQS-S01 v bielom prevedení[1]	4	3.19 Vizualizácia osadeného plošného spoju od spoločnosti JLCPCB	24
2.2 Dizajn zariadenia R200C2-E v bielom prevedení[2]	4	3.20 Plošný spoj obsahujúci všetky komponenty	24
2.3 Dizajn zariadenia AirBox01 v bielošedom prevedení[3]	5	4.1 Príklad zobrazovanie dát na platforme Blynk	26
2.4 Dizajn zariadenia Wave plus v bielom prevedení a dizajn mobilnej aplikácie[4]	6	4.2 Statická obrazovka po inicializácii displeja	27
3.1 Schéma nášho zariadenia na monitorovanie kvality vzduchu v domácnosti	9	4.3 Tabuľka indexu CO2[22]	28
3.2 Modul mikrokontroléru ESP32 DEVKIT V1 - DOIT, ktorý obsahuje čip ESP32-WROOM[6]	11	4.4 Hlavný vývojový diagram programu	30
3.3 Senzor na monitorovanie teploty, vlhkosti a tlaku[7]	12	4.5 Vývojový diagram funkcie pre čítanie zo senzorov	31
3.4 Senzor na monitorovanie CO2 a TVOC[8]	12	4.6 Vývojový diagram funkcie na vypísanie hodnôt na displej	31
3.5 Displej od firmy Waveshare nami zvolený pre zobrazovanie údajov zo senzorov[9]	13	4.7 Vývojový diagram spustenia alarmu a následného odoslania upozornenia	32
3.6 Piezoelektrický bzučiak Ø10 [mm], 85 [dB], 3-12 [V][11]	14	4.8 Vývojový diagram posielania dát na Blynk cloud	32
3.7 Akumulátor LiFePO4 s kapacitou 2000 [mAh][13]	15	5.1 Vzhľad nášho zariadenia hneď po zapnutí	33
3.8 Vizualné zobrazenie SPI komunikácie Master/Slave[15]	17	5.2 Vzhľad nášho zariadenia po jednom meraní	34
3.9 Vizualné zobrazenie I2C komunikácie Master/Slave[16]	17	5.3 Grafické zobrazenie teploty a vlhkosti na cloude na platforme Blynk	34
3.10 Schématické zapojenie všetkých komponentov	18	5.4 Grafické zobrazenie CO2 a TVOC cloude na platforme Blynk	35
3.11 Schéma zapojenia napájania zo zdroja	19	5.5 Grafické zobrazenie veličín v mobilnej aplikácii Blynk	35
3.12 Schéma zapojenia nabíjacieho čipu	19	5.6 Upozornenie poslané na mobil pri nebezpečnej hodnote CO2	36
3.13 Vedenie napájania cez PMOS ..	20		
3.14 Menič napätia od spoločnosti dratek.cz[19]	20		
3.15 Zapojenie modulov senzorov na plošnom spoji, pripojenie k ESP32 ..	21		
3.16 Schéma modulov senzorov BME280 a CCS811	22		
3.17 Zapojenie bzučiaka k ESP32 ..	22		
3.18 Zapojenie pinov displeja k ESP32	23		

Tabuľky

3.1 Vlastnosti vybraných mikrokontrolérov[5]	10
3.2 Základné špecifikácie displeja Waveshare 2.9-inch e-Paper HAT[9]	13
3.3 Technické špecifikácie piezoelektrického bzučiaku[11]	14
3.4 Základne vlastnosti akumulátor, zohľadnené pri rozhodovaní	14
3.5 Parametry LiFePO4 akumulátoru s kapacitou 2 [Ah][13]	15
3.6 Vlastnosti komunikačných periférií + výhody/nevýhody	16
3.7 Pripojenie pinov senzorov k ESP32	21
3.8 Zapojenie pinov displeja pre SPI komunikáciu	23
4.1 Funkcie a ich spotreba v rôznych režimoch ESP32[23]	29
6.1 Odhadovaná cena zariadenia bez zohľadnenia nákladov na prácu ...	38

Kapitola 1

Úvod

1.1 Motivácia

Prácu na tému kvality vzduchu v domácnosti som si vybral pretože má trápi situácia znečistenia ovzdušia v mojom obydli, ktorá sa neustále zhoršuje. Tiež som si túto tému vybral, aby som nazrel do sveta mikroelektroniky a naučil sa zostrojiť zariadenie od základov až po naprogramovanie mnou zostrojeného zariadenia.

1.2 Formulácia problému

Vo vzduchu sa môžu nachádzať aj látky, ktoré by mohli ohrozovať život ľudí (napríklad oxid uhličitý). Okrem toho, aby sme vedeli efektívne fungovať, bolo by veľmi vhodné, ak by bolo ovzdušie okolo nás ideálnej teploty, vlhkosti a tak ďalej. Ako teda zistíme kvalitu ovzdušia v našom okolí? Aby sme vedeli aký vzduch dýchame, je potrebné ho najskôr monitorovať. A to je presne problematika tejto práce, **monitorovanie ovzdušia**.

1.3 Cieľ práce

Cieľom práce bude navrhnuť a zostrojiť zariadenie na monitorovanie ovzdušia v uzavretom priestore. Zostrojím prístroj, ktorý bude merať nasledujúce veličiny:

- Teplota
- Vlhkosť
- CO₂
- TVOC

Toto zariadenie bude obsahovať displej **e-ink**, ktorý bude namerané hodnoty graficky zobrazovať. Okrem toho ešte do zariadenia pridám **bzučiak**, ktorého úlohou bude zvukovo signalizovať prekročené hodnoty meranej veličiny, ktorá by potenciálne mohla ohrozovať život. Toto zariadenie bude merané veličiny

nielen zobrazovať na svojom displeji, ale tiež ich bude posielať na internetový **cloud** a do **mobilnej aplikácie** cez **WiFi**. Tiež chcem, aby zariadenie bolo mobilné, a preto bude napájané z dobíjateľnej batérie.

Kapitola 2

Prieskum trhu

Samozrejme, že niesme prvý kto sa zabýva kvalitou a monitorovaním ovzdušia. V tejto kapitole sa pozrieme na rôzne zariadenia od rôznych výrobcov a rozoberieme si ich plusy a mínusy.

2.1 Aqara TVOC Air Quality Monitor (AAQS-S01)

Aqara AAQS-S01 dokáže detekovať koncentráciu a hladinu TVOC vo vzduchu, rovnako ako teplotu a vlhkosť. Potom, čo je prepojený s Hub-om, môžete používať koncentráciu alebo hladinu TVOC, teplotu a vlhkosť ako podmienky automatizácie, dokáže pracovať s inými zariadeniami pre inteligentné domácnosti k implementácii rôznych inteligentných scenárov a odosielať oznámenia na váš mobilný telefón.[2]

Vlastnosti:

1. **Detekcia TVOC**
2. **Detekcia teploty**
3. **Detekcia vlhkosti**
4. **Displej:** Je vybavený ihličkovou E-Ink obrazovkou.
5. **Inteligentná automatizácia:** S protokolom Zigbee 3.0 môže pracovať s inými zariadeniami prostredníctvom hub-u pre cloudovú a miestnu automatizáciu.



Obrázok 2.1: Dizajn zariadenia AAQS-S01 v bielom prevedení[1]

Kúpna cena: 1022 CZK

Porovnanie s naším zariadením: Vo veľa veciach je to veľmi podobné zariadenie, s podobnými vlastnosťami aké bude obsahovať náš monitor kvality vzduchu. Rozdiel je v tom, že prístroj AAQS-S01 komunikuje cez platformu ZigBee a naše zariadenie cez WiFi. Naše zariadenie bude merať okrem tých istých meraných veličín ešte aj oxid uhličitý a zvukovou signalizáciou hlásiť jeho zvýšenú koncentráciu.

2.2 Detektor Honeywell Home Resideo R200C2-E

Zariadenie R200C2-E primárne detekuje koncentráciu oxidu uhličitého v uzavretom priestore. Pri jeho zvýšenej koncentrácii Vás na to zvukovo upozorní.[2] Vlastnosti:

1. **Detekcia CO₂**
2. **Detekcia teploty**
3. **Detekcia vlhkosti**
4. **LED displej:** Obsahuje LED displej s nastaviteľným jasom.
5. **Hlasové upozornenie:** Pri zvýšených hodnotách meranej veličín vás hlasovo upozorní.



Obrázok 2.2: Dizajn zariadenia R200C2-E v bielom prevedení[2]

Kúpna cena: 982 CZK

Porovnanie s naším zariadením: Všetky tieto vlastnosti čo obsahuje R200C2-E bude obsahovať aj naše zariadenie (len okrem LED displeja budeme mať e-ink). Okrem toho budeme merať TVOC a informácie posielať v reálnom čase na internet, takže budeme mať prehľad o hodnotách z hocikákeho miesta na Zemi.

2.3 iQtech SmartLife WiFi AirBox01

Inteligentný senzor iQtech SmartLife WiFi AirBox01 svojim užívateľom ponúka prehľad o veličinách ako je teplota, vlhkosť, oxid uhličitý a TVOC. Hoci tento senzor neobsahuje displej, je možné ho pripojiť cez WiFi k mobilu a merané veličiny sledovať cez svoj smartfón v aplikácii.

Vlastnosti:

1. **Detekcia CO₂**
2. **Detekcia teploty**
3. **Detekcia vlhkosti**
4. **Detekcia TVOC**
5. **Pripojenie k WiFi:** Senzor pripojíte cez Wi-Fi k mobilnej aplikácii a zmeny sledujete v nej.
6. **Možnosť rozšírenia:** Môžete využiť aj funkciu automatizácie s ďalšími zariadeniami z rodiny iQtech SmartLife.



Obrázok 2.3: Dizajn zariadenia AirBox01 v bielošedom prevedení[3]

Kúpna cena: 982 CZK

Porovnanie s naším zariadením: Tento inteligentný senzor postráda displej, na ktorom by zobrazoval merané hodnoty a oproti nášmu senzoru mu ešte chýba zvuková signalizácia pri prekročení hodnoty meranej veličiny. Okrem týchto troch vlastností je nášmu projektu veľmi podobný.

2.4 AirThings Wave plus

Ako na posledné zariadenie sa pozrieme na senzor AirThings Wave plus, ktorý je ale cenou násobne drahší než predchádzajúce senzory kvality vzduchu, tomu odpovedá aj spektrum jeho funkcií. Senzor pre interiér, 6 senzorov pro radón, CO₂, vlhkosť, teplotu, TVOC a tlak. Merané veličiny síce nezobrazuje na svojom displeji, ale posiela ich do mobilnej aplikácie, kde vidíme ich aktuálne hodnoty.[4]

Vlastnosti:

1. **Detekcia Teploty**
2. **Detekcia Vlhkoti**
3. **Detekcia Tlaku**
4. **Detekcia CO₂**
5. **Detekcia TVOC**
6. **Detekcia Radónu**
7. **Pripojenie cez bluetooth:** K zariadeniu sa vieme pripojiť cez bluetooth a merané veličiny sledovať v mobilnej aplikácii.
8. **Vizuálna indikácia:** Zariadenie obsahuje LED pásik s tromi možnými farbami, ktoré indikujú aké ovzdušie sa okolo nás nachádza (červená, žltá a zelená).



Obrázok 2.4: Dizajn zariadenia Wave plus v bielom prevedení a dizajn mobilnej aplikácie[4]

Kúpna cena: 5304 CZK

Porovnanie s naším zariadením: Toto zariadenie obsahuje o niečo viac vlastností ako naše (my nebudeme merať radón a tlak). Rozdiel tu je v tom, že komunikačný protokol senzoru Wave plus je bluetooth a u nás bude WiFi (tj. merané hodnoty nášho senzoru budú distribuované po internete na väčšie vzdialenosti ako keby to bolo cez bluetooth). Absenciu zvukového hlásenia pri nebezpečne vysokej hodnote meranej veličiny nahrádza u tohto zariadenia

svetelná signalizácia. Ale pre zistenie hodnôt je treba mať mobil a aplikáciu, keďže Wave plus postráda displej, kde by zobrazoval hodnoty veličín.

■ 2.5 Záver prieskumu trhu

Po prieskume tých typicky často predávaných zariadení sme zistili, že väčšina zo skúmaných senzorov postráda nejakú z vlastností nášho návrhu. Tým pádom konštatujeme, že naše zariadenie bude o niečo málo komplexnejšie ako sú bežne dostupné riešenia, ktoré sú dostupné v predajniach. Môžeme si všimnúť, že zariadenia, ktoré obsahujú displej tak merané veličiny zväčša iba zobrazujú a neposielajú ich do inej užívateľskej aplikácie. A naopak, tie čo posielajú údaje do aplikácie, väčšinou neobsahujú displej, kde sú zobrazené všetky merané veličiny. To sa budeme v návrhu nášho senzoru snažiť urobiť inak a merané veličiny jednak zobrazovať na displeji a aj posielat' na online platformu.

Kapitola 3

Návrh senzora - HARDWARE

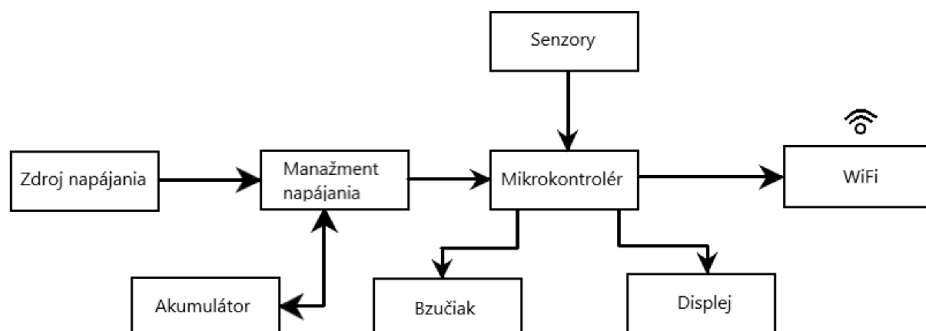
V tejto kapitole sa budeme venovať konceptu nášho zariadenia na monitorovanie kvality ovzdušia. Ďalej sa pozrieme schématicky na návrh senzora a tiež si uvedieme a odôvodnime vybrané komponenty pre jednotlivé časti zariadenia.

3.1 Rozbor zariadenia

Ako sme spomínali v kapitole 1, naším cieľom je zostrojiť zariadenie, ktoré bude monitorovať, zobrazovať a odosielať hodnoty meraných veličín. Aby sme týchto úkonov boli schopný, budeme potrebovať centrálnu riadiacu jednotku - **mikrokontrolér**. Do mikrokontroléru budeme posielat hodnoty zo **senzorov**, ktoré budu merať veličiny spomenuté v sekcii 1.3. Z mikrokontroléru budeme ovládať **displej** pre zobrazovanie údajov, **bzučiak** na signalizovanie nebezpečnej vysokej hodnoty veličiny CO₂ a tiež riadiť pripojenie k internetu cez WiFi. Ak požadujeme aby naše zariadenie bolo prenosné a mobilné, budeme taktiež potrebovať dobíjateľnú batériu - **akumulátor**.

3.2 Schéma zariadenia

Prvky do nášho senzora kvality vzduchu implementujeme podľa nasledujúcej schémy:



Obrázok 3.1: Schéma nášho zariadenia na monitorovanie kvality vzduchu v domácnosti

3.3 Výber komponent

Podľa schémy vyberieme hardvérové komponenty, pomocou ktorých naše zariadenie zostrojíme. V nasledujúcich podkapitolách sa pozrieme na každý jeden komponent a odôvodníme si jeho výber.

3.3.1 Mikrokontrolér

Mikrokontrolér je stavebnou komponentou celého zariadenie. Všetky informácie prijíma, spracúva a ďalej posiela. Pozrieme sa na 3 veľmi často používané mikrokontroléry **ESP32**, **Raspberry Pi 2040** a **STM32** a odôvodníme si výber toho, ktorý sa najviac hodí pre našu prácu.

Vlastnosť	ESP32	RP2040	STM32
Procesor	Xtensa LX7 32 bit Dual-core 240 MHz	ARM Cortex-M0 Dual-core 133 MHz	32-bit ARM cortex M3 72 MHz
Počet jadier	2	1	1 a viac
RAM	520 [kB]	264 [kB]	20 [kB]
WiFi	Áno	Nie	Nie (iba s modulom)
Bluetooth	Áno	Nie	Nie (iba s modulom)
I2C	Áno	Áno	Áno
Spotreba v spánkovom režime	10 [μA]	39 [μA]	20 [μA]

Tabuľka 3.1: Vlastnosti vybraných mikrokontrolérov[5]

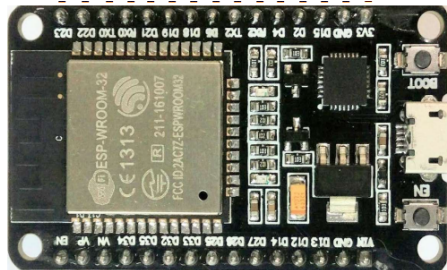
Vzhľadom na tabuľku (3.1) si kvôli integrácii komunikačného rozhrania WiFi a nízkej spotrebe energie volíme pre náš projekt mikrokontrolér typu **ESP32**, konkrétne vývojový modul **ESP32 DEVKIT V1 - DOIT**, kvôli širokej podpore komunity a veľkom množstve materiálov pre prácu s ním. Tento modul obsahuje čip ESP32-WROOM32.

ESP32-WROOM32

Vybraný mikrokontrolér má nasledujúce vlastnosti[6]:

- **Výkon:** Obsahuje procesor Xtensa Dual-Core 32-bit LX6 s taktom 160MHz, ktorý je určený pre aplikácie s **nízkou spotrebou energie**.
- **Pamäť:** 520 [kB] SRAM a 448 [kB] ROM.
- **Bezdrôtové funkcie:** Podporuje WiFi pripojenie s rýchlosťou až **150.0 [Mbps]**.

- **Periférie:** Obsahuje periférie I2C a SPI, ktoré použijeme na pripojenie senzorov a displeja.



Obrázok 3.2: Modul mikrokontroléru ESP32 DEVKIT V1 - DOIT, ktorý obsahuje čip ESP32-WROOM[6]

■ 3.3.2 Sensory

Keď sme už vybrali mikrokontrolér, ďalšou komponentou sú senzory. Budeme potrebovať dva moduly senzorov. Jeden bude merať **teplotu a vlhkosť** a druhý bude slúžiť na meranie **CO2 a TVOC**.

■ BME280 - teplota, vlhkosť

Pre jednoduchú implementáciu, kompaktnosť a presnosť sme si zvolili senzor BME280. Je to kombinovaný senzor a používa sa na meranie teploty, vlhkosti a tlaku okolia.

Technické parametre senzoru[7]:

1. Teplota:
 - **Rozsah meranie:** -40 [°C] až 80 [°C]
 - **Presnosť:** ±1 [°C]
2. Vlhkosť:
 - **Rozsah meranie:** 0 [%] až 100 [%]
 - **Presnosť:** ±3 [%]
 - **Rezolúcia:** ± 0.01 [%]

3. Atmosférický tlak:

- **Rozsah merania:** 300 [hPa] až 1100 [hPa]
- **Presnosť:** ± 1 [%]
- **Rezolúcia:** 0.18 [hPa]



Obrázok 3.3: Senzor na monitorovanie teploty, vlhkosti a tlaku[7]

■ **CCS811 - Oxid uhličitý, TVOC**

Senzor CCS811 je opäť kombinovaný senzor od spoločnosti Adafruit. Používa sa na monitorovanie veličín CO₂ a TVOC.

Technické parametre senzoru:

1. Celkove organické tekavé zlúčeniny (TVOC):

- **Rozsah merania:** 0 [ppb] až 1187 [ppb]
- **Presnosť:** ± 15 [%] pri 400 [ppb]
- **Rezolúcia:** 1 [ppb]

2. Oxid uhličitý (CO₂):

- **Rozsah merania:** 400 [ppm] až 8192 [ppm]
- **Presnosť:** ± 50 [ppm] + 3 [%] relatívna presnosť
- **Rezolúcia:** 1 [ppm]



Obrázok 3.4: Senzor na monitorovanie CO₂ a TVOC[8]

3.3.3 Displej

Pri výbere displeja sme sa hlavne zamerali na to, aby mal nízku spotrebu energie. Pri tomto požiadavku sme sa rozhodli pre **elektronický inkous**. Displej typu e-ink nieje náročný na spotrebu energie a je dostatočne kvalitný na to, aby boli merané údaje zobrazené čitateľne. Kvôli úspore energie sme sa rozhodli pre **čiernobiely displej**, ktorý je plne dostačujúci na grafiku zobrazovaných údajov. Rozhodli sme sa pre displej od spoločnosti Waveshare, konkrétne model **Waveshare 2.9-inch e-Paper HAT**. Špecifikácie tohoto displeja sú uvedené v nasledujúcej tabuľke.

Veľkosť	Rozlíšenie	Rozhranie	Napájanie
2.9 palca (približne 7.4 cm)	296 x 128 pixelov	SPI	3.3 [V]

Tabuľka 3.2: Základné špecifikácie displeja Waveshare 2.9-inch e-Paper HAT[9]



Obrázok 3.5: Displej od firmy Waveshare nami zvolený pre zobrazovanie údajov zo senzorov[9]

3.3.4 Bzučiak

Pri výbere bzučiaku, sa zameriame na nízku spotrebu energie, dostatočnú frekvenciu, jednoduchú ovládateľnosť a programovateľnosť. Pri výbere komponenty pre zvukovú signalizáciu sme sa rozhodovali medzi použitím reproduktora a piezoelektrického bzučiaka. Rozhodli sme sa pre použitie **piezoelektrického bzučiaka**.

Vlastnosti bzučiaka:

- **Princíp:** Piezoelektrický bzučiak využíva piezoelektrický jav na generovanie zvuku. Keď sa k nemu privedie elektrický signál, piezoelektrický materiál sa deformuje a vytvára mechanické vibrácie, ktoré produkujú zvukové vlny[10].
- **Charakteristika:** Piezoelektrické bzučiaky majú obvykle jednoduchý dizajn, sú kompaktné a majú nízku spotrebu energie. Vytvárajú často ostrý, výrazný zvuk, ktorý je vhodný na generovanie jednoduchých zvukových signálov alebo signalizáciu.

Šírka	Výška	Napájanie	Frekvencia	Hlasitosť
10 [mm]	5 [mm]	3-12 [V]	± 3000 [Hz]	85 [dB]

Tabuľka 3.3: Technické špecifikácie piezoelektrického bzučiaku[11]

Pretože na našu signalizáciu nebezpečne vysokej hodnoty meranej veličiny stačí rezonantný zvuk, je jednoduchšou a energiu šetriacou voľbou, aj pre svoje kompaktné rozmery, piezoelektrický bzučiak.



Obrázok 3.6: Piezoelektrický bzučiak Ø10 [mm], 85 [dB], 3-12 [V][11]

3.3.5 Akumulátor

Výber vhodného akumulátora a celkovo vhodné riešenie napájania je pre náš projekt dôležitou oblasťou. Potrebujeme, aby náš napájaci zdroj mal dostatočnú hustotu napájania, tiež aby bol bezpečný a ideálne nie príliš veľkej hmotnosti. Rozhodovali sme sa medzi tromi typmi akumulátorov: Lítiový polymerový (**LiPo**) akumulátor, Lítiový železofosfátový (**LiFePO4**) akumulátor a Nikl-metalhydridový (**NiMH**) akumulátor. V nasledujúcej tabuľke sú uvedené základné vlastnosti týchto troch typov akumulátorov.

Vlastnosť	LiPo	LiFePO4	NiMH
Hustota energie [Wh/kg]	Vysoká	Stredná až vysoká	Stredná
Životnosť	Nízka	Vysoká	Stredná až vysoká
Bezpečnosť	Stredná	Vysoká	Stredná
Nabíjacie rýchlosti	Vysoká	Stredná	Nízka
Cena	Stredná až vysoká	Vysoká	Nízka až stredná

Tabuľka 3.4: Základne vlastnosti akumulátor, zohľadnené pri rozhodovaní

Vzhľadom na predchádzajúcu tabuľku sme uznali za vhodné aby naše zariadenie obsahovalo akumulátor s vysokou životnosťou a bezpečnosťou. Rozhodli sme sa pre akumulátor lítium železofosfátový (**LiFePO4**).

LiFePO4

LiFePO4 akumulátor funguje na základe elektrochemických reakcií medzi lithiovými iónmi v elektrolyte a elektrodami batérie. Počas nabíjania sa

lithiové ióny presúvajú z anódy na katódu, čím sa batéria nabíja. Počas vybíjania sa lithiové ióny presúvajú z katódy späť na anódu, čím sa uvoľňuje energia a batéria sa vybíja[12]. Veľkosť akumulátora sme si zvolili **2 000 [mAh]**. Batéria s touto kapacitou je typu IFR18650.



Obrázok 3.7: Akumulátor LiFePO4 s kapacitou 2 000 [mAh][13]

V nasledujúcej tabuľke si uvedieme parametry použitého akumulátora.

Hmotnosť	± 44 [g]
Energia	6.4 [kWh]
Menovitá kapacita	2 [Ah]
Menovité napätie	3.2 [V]
Maximálne nabíjacie napätie	3.65 [V]

Tabuľka 3.5: Parametry LiFePO4 akumulátora s kapacitou 2 [Ah][13]

3.3.6 Záver výberu komponent

Začali sme výberom mikrokontroléru, pre svoju nízku spotrebu energie a komunikáciu cez WiFi sme zvolili mikrokontrolér **ESP32** konkrétne vývojový modul typu DOIT. Ako ďalšie sme vybrali senzory, ktoré budú do mikrokontroléra posielat svoje namerané dáta (teplota, vlhkosť,...). Vybrali sme dva kombinované senzory **BME280** (teplota, vlhkosť) a **CSS811** (CO₂, TVOC). Dáta z týchto senzorov budú spracované v riadiacej jednotke a tá ich ďalej pošle jednak na svoj displej a druhak na internetový cloud. Pre úsporu energie, ale stále dostatočné rozlíšenie sme si zvolili displej typu e-ink, konkrétne **Waveshare 2.9-inch e-Paper HAT** v čiernobielej farbe zobrazovania údajov. Na zvukovú signalizáciu použijeme **piezoelektrický bzučiak** s priemerom 10 [mm], ktorý vydáva rezonantný zvuk pri zapnutí a tiež je šetrný k spotrebe. V neposlednom rade sme vybrali akumulátor. Ako akumulátor sme zvolili typ **LiFePO4**, ktorý je síce o niečo drahší, ale zato má vysokú životnosť a bezpečnosť voči prebitiu. Nakoniec ešte chcem spomenúť, že všetky komponenty pracujú pri napätí **3.3 [V]**.

3.4 Komunikácia medzi komponentami

V predchádzajúcej sekcii sme vybrali komponenty, ktoré použijeme pri zostrojení nášho senzoru na sledovanie kvality ovzdušia. V tejto sekcii sa pozrieme na to, ako medzi sebou budú senzory, bzučiak, displej a mikrokontrolér **komunikovať**. Porovnáme rôzne **komunikačné periférie**.

3.4.1 Porovnanie komunikačných periférií

Porovnali sme tieto periférie na komunikáciu medzi zariadeniami: **UART**, **I2C** a **SPI**.

Vlastnosť	UART	I2C	SPI
Typ	Asynchronný	Synchronný	Synchronný
Topológia	Bod-bod	Viac-masterový	Bod-bod
Počet pinov (pre 2 zariadenia)	2 (Tx, Rx)	2 (SDA, SCL)	4 (MOSI, MISO, SCK, SS/CS)
Rýchlosť prenosu	Od pár bitov za sekundu po 10 [mb/s]	100 [kHz] až 3.4 [MHz]	Závisí na frekvencii SCK
Protokol prenosu	-	Start/stop bit, ACK/NACK	Plne-duplexný, polo-duplexný
Výhody	Jednoduché pripojenie, rozšírenie v rôznych aplikáciách	Jednoduché adresovanie viacero zariadení na zbernicu	Vysoká rýchlosť prenosu, vyššia spoľahlivosť
Nevýhody	Menej flexibilný, len bod-bod	Obmedzená rýchlosť, a počet zariadení	Viac pinov, väčšia komplexita

Tabuľka 3.6: Vlastnosti komunikačných periférií + výhody/nevýhody

Nakoľko **UART** podporuje komunikáciu len medzi dvomi zariadeniami, tak nebude použitý pri komunikácii medzi komponentami. Zameriame sa na rozhrania **I2C** a **SPI**.

3.4.2 Komunikačné rozhranie SPI

Komunikačné rozhranie SPI (Serial Peripheral Interface) je sériový, synchronný protokol, ktorý použijeme na komunikáciu medzi displejom a mikrokontrolérom.

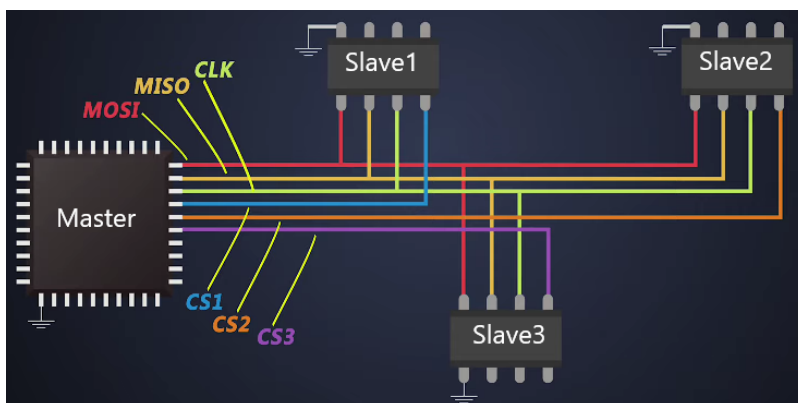
Rozdelenie zariadenia na sériovej SPI zbernici[14]:

Master:

- riadi komunikáciu pomocou hodinového signálu.
- určuje, s ktorým zariadením na zbernici sa bude komunikovať SS-slave select (alebo CS-chip select).

Slave:

- vysiela podľa hodinového signálu, pokiaľ je aktivovaný pomocou SS/CS.



Obrázok 3.8: Vizuálne zobrazenie SPI komunikácie Master/Slave[15]

3.4.3 Komunikačné rozhranie I2C

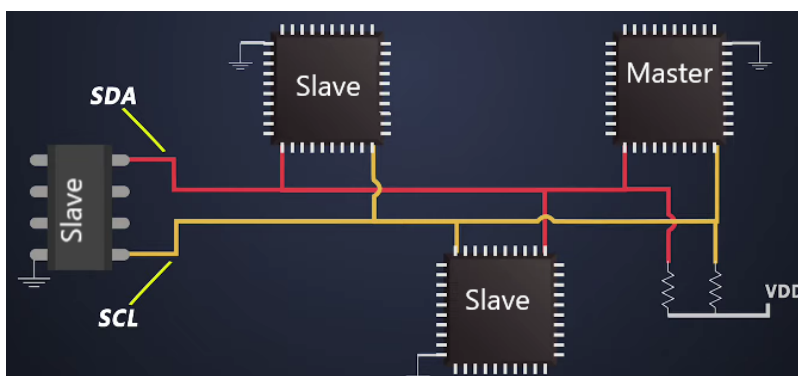
Princíp I2C periférie je založený na dvoch signáloch: hodinovom signále (SCL) a dátovom signále (SDA). Tieto dva signály umožňujú zariadeniam komunikovať pomocou jedného spoločného zbernicového vedenia. I2C periférie použijeme na komunikáciu medzi senzormi a mikrokontrolérom. Rozdelenie zariadenia na I2C zbernici:

Master:

- Zariadenie Master je to, ktoré generuje hodinový signál.

Slave:

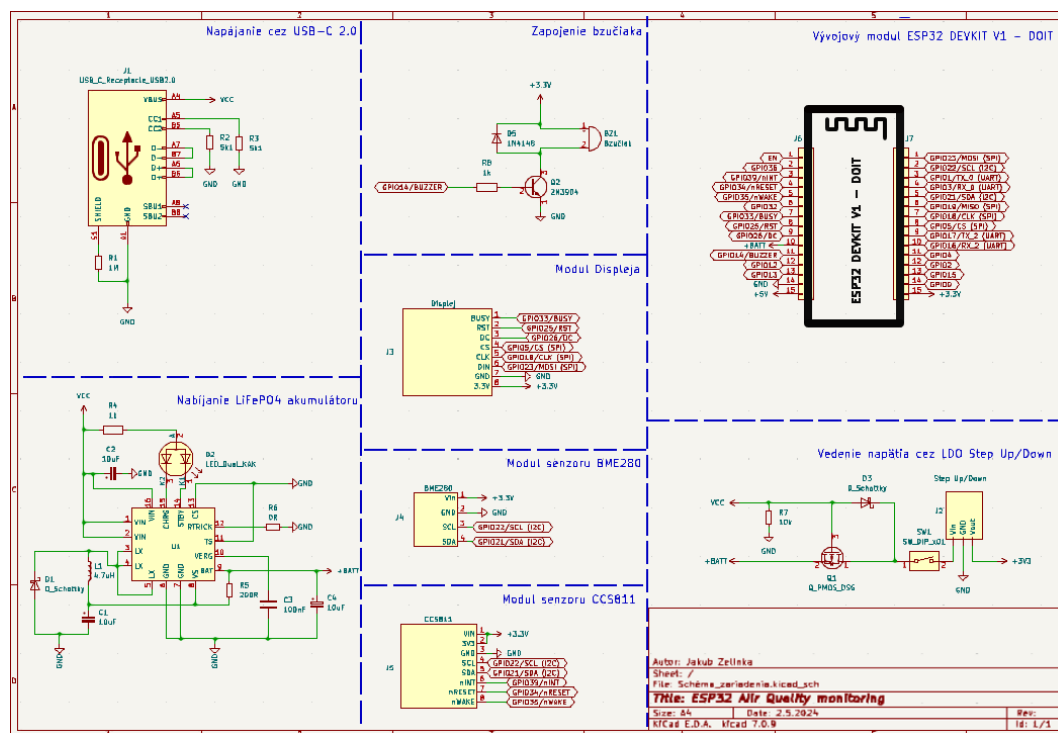
- Slave je zariadenie, ktoré prijíma rozkazy od Mastera reaguje na ne.



Obrázok 3.9: Vizuálne zobrazenie I2C komunikácie Master/Slave[16]

3.5 Schématické zapojenie zariadenia

Keďže už sme vybrali prvky, ktoré bude naše zariadenie obsahovať a tiež sme spracovali komunikáciu medzi nimi, je na rade schématicky navrhnuť prepojenie komponentov so správou batérie. Schéma hardvéru nášho zariadenia je zobrazená na nasledujúcom obrázku (viz. 3.10.).

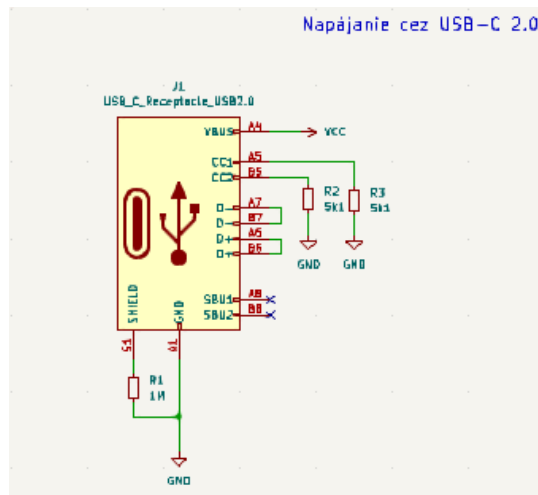


Obrázok 3.10: Schématické zapojenie všetkých komponentov

V nasledujúcich podkapitolách si okomentujeme zapojenie jednotlivých komponentov.

3.5.1 Napájanie cez USB-C 2.0

Pre napájanie sme sa rozhodli použiť konektor **USB typu C**, konkrétne model 2.0, ktorý je v dnešnej dobe veľmi používaný a je možné ho zapojiť obojstranne.

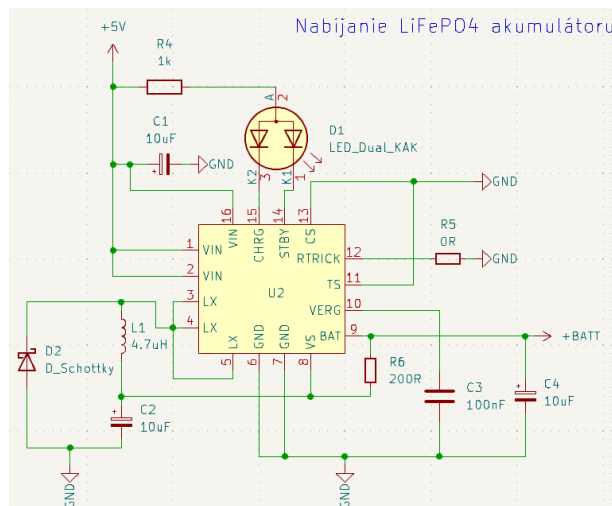


Obrázok 3.11: Schéma zapojenia napájania zo zdroja

Napájacie napätie je továrensky nastavené na 5 [V]. Prúd do obvodu takto zapojený konektor dodáva o veľkosti 1.5 [A] (nastaviteľný pinmi CC1 a CC2, ktoré sme privedli na zem cez 5.1 [kΩ] rezistory)[17].

3.5.2 Nabíjanie LiFePO4 akumulátoru

Naše zariadenie bude obsahovať LiFePO4 akumulátor s kapacitou **2 000 [mAh]**. Pre nabíjanie tohto akumulátoru sme si vybrali nabíjaci čip **TP5000**, ktorý disponuje možnosťou nastavenia nabíjacieho prúdu a tiež ochranou proti prebitiu. V neposlednom rade obsahuje aj tepelnú ochranu batérie.

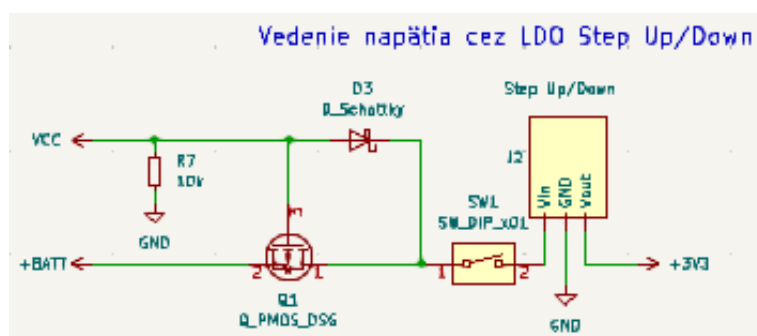


Obrázok 3.12: Schéma zapojenia nabíjacieho čipu

Nabíjací prúd sa nastavuje veľkosťou rezistoru "R6". My sme ho nastavili na 200 [Ω], takže nabíjací prúd je 500 [mA]. Okrem týchto nabíjacích vlastností takto zapojený obvod obsahuje dvojfarebnú **indikačnú LED** so spoločnou anódou na signalizáciu stavu nabíjania (nabité - zelená, nabíja sa - červená)[18].

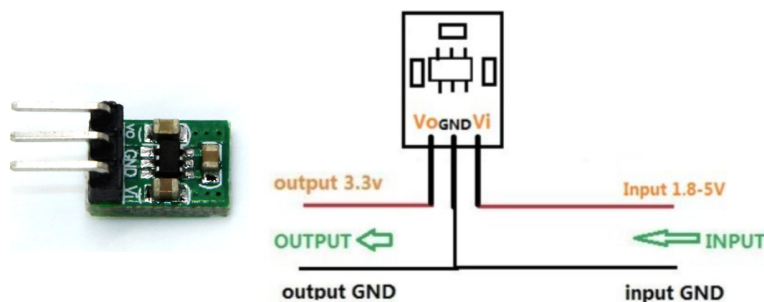
3.5.3 Vedenia napätia z batérie alebo zdroja

Chceme, aby sme naše zariadenie mohli používať, aj keď budeme nabíjať batériu. Následujúca časť obsahujúca **tranzistor PMOS** túto vlastnosť zaisťuje.



Obrázok 3.13: Vedenie napájania cez PMOS

Ak je zariadenie zapojené v zdroji, tak na "+BATT" je nulové napätie, pretože sa akumulátor nabíja a v "VCC" (výstup z USB-C konektoru) je napätie 5 [V], ktoré cez Schotkyho diodu ide ďalej do obvodu. Ak zariadenie nie je zapojené do zdroja, tak na "VCC" je nulové napätie a "+BATT" dodáva energiu batérie do obvodu. Po meraní sme zistili, že napätie ktoré dodáva batéria do obvodu, ak je zdroj napájania odpojený, je 2.58 [V] na výstupe z tranzistoru PMOS. Tento úbytok je spôsobený vysokým saturačným napätím tranzistoru. Aby sme napätie regulovali a do mikrokontroléru ESP32, senzorov a displeju dostali 3.3 [V], tak sme pridali modul na menič napätia **Step Up/Down**[18]. Tento modul mení napätie z rozsahu na vstupe 1.8 - 5 [V] na výstupné napätie 3.3 [V]. V obvode je ešte umiestnený **jednopolový manuálny spínač**, pomocou ktorého naše zariadenie odpojíme od energie čím ho môžeme vypnúť, alebo zapnúť.



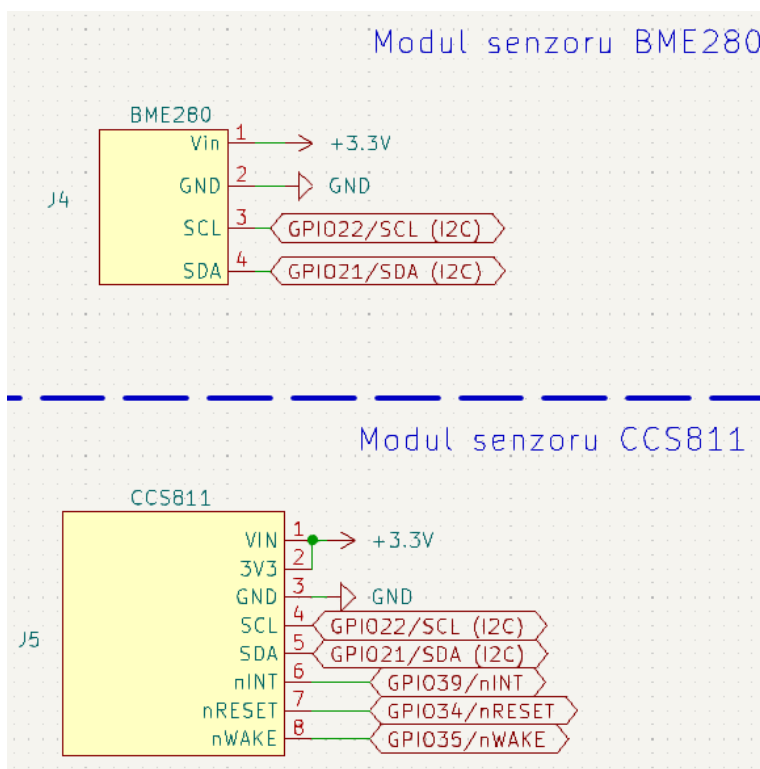
Obrázok 3.14: Menič napätia od spoločnosti dratek.cz[19]

3.5.4 Zapojenie modulov senzorov

Sensory sme pripojili k ESP32 cez komunikačné rozhranie I2C. Pri pripojení senzorov si bolo treba dať pozor na správne zapojenie pinov "SDA", "SCL".

Pin Senzoru	Pin ESP32
VIN	3.3
GND	GND
SDA	GPIO21
SCL	GPIO22

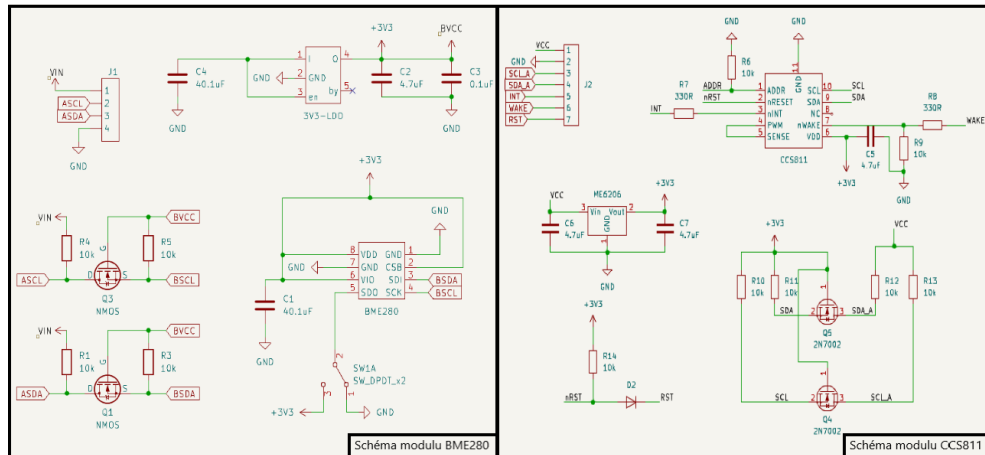
Tabuľka 3.7: Pripojenie pinov senzorov k ESP32



Obrázok 3.15: Zapojenie modulov senzorov na plošnom spoji, pripojenie k ESP32

Piny senzoru CCS811 ako sú "nINT", "nRESET", "nWAKE" sme nepoužili a preto niesú uvedené ani v tabuľke vyššie (no tieto piny môžu byť pripojené k hociktorému GPIO pinu na ESP32).

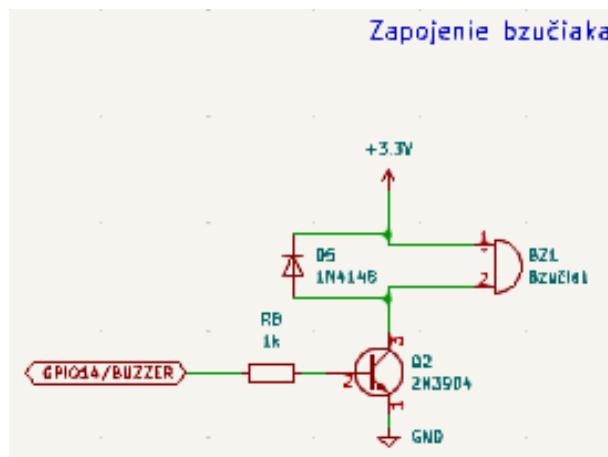
Nakoľko sme použili moduly senzorov, ktoré sú už osadené na plošnom spoji a ich periférie rovno pripravené na komunikáciu cez I2C, ukážeme si schémy týchto modulov.



Obrázok 3.16: Schéma modulov senzorov BME280 a CCS811

3.5.5 Zapojenie bzučiaka

Aby sme z bzučiaka dostali zvuk, ktorý považujeme za dostatočne hlučný, pripojili sme jeho kladný pól k napätiu 3.3 [V] a záporný k zemi. Ak je takto zapojený, vydáva veľmi prenikavý tón ideálny pre signalizovanie, ktoré chceme použiť. Avšak keď ho chceme ovladať, tak sme medzi záporný pól a zem umiestnili NPN tranzistor, ktorý je otváratelny a zatváratelny cez GPIO pin mikrokontroléru ESP32,.



Obrázok 3.17: Zapojenie bzučiaka k ESP32

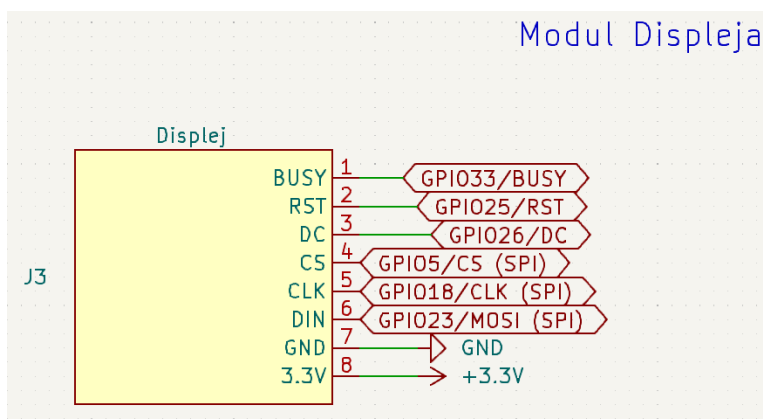
3.5.6 Zapojenie displeja

Displej pripojíme k ESP32 tak, aby prenášal dáta cez komunikačné periférie SPI.

Pin Displaja	Pin ESP32
VIN	3.3 [V]
GND	GND
DIN	GPIO23 (MOSI)
CLK	GPIO18 (CLK)
CS	GPIO5 (CS)

Tabuľka 3.8: Zapojenie pinov displeja pre SPI komunikáciu

Rovnako ako pri senzore CCS811 sme niektoré piny nepoužili ("BUSY", "RST", "DC") tak sme ich neuviedli do tabuľky. Zapojenie je zobrazené na nasledujúcom obrázku.

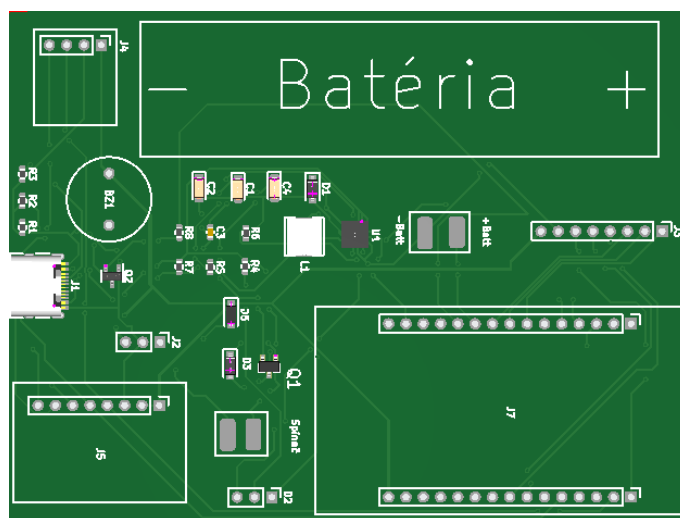


Obrázok 3.18: Zapojenie pinov displeja k ESP32

3.6 Výroba osadeného plošného spoju

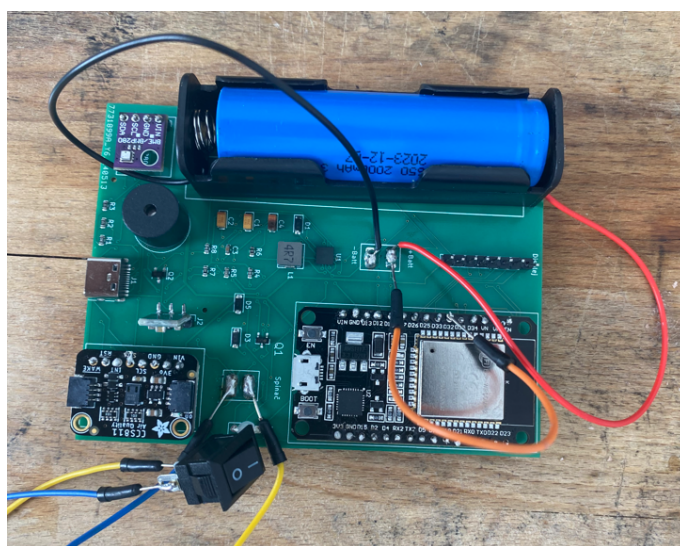
Schému sme navrhli v softwérovom programe **KiCad**. Jakmile bola schéma nakreslená, zvolili sme do nej konkrétne komponenty, ktoré boli dostupné v spoločnosti **JLCPCB**[20]. Táto firma sa špecializuje na výrobu a osadenie plošných spojov (nakoľko sme použili veľa SMD súčiastiek, ktorých pájkovanie by bolo kvôli našim obmedzeným skúsenostiam náročné a riskantné, rozhodli sme sa využiť osadenie súčiastok cez JLCPCB). Tejto spoločnosti sme dodali **Gerber** súbor (prepojenie komponentov v PCB editore), súbor s použitými komponentami ktoré chceme aby osadili a súbor s ich pozíciami na plošnom spoji.

Výroba osadeného plošného spoju trvala približne 48 hodín a dovoz z fabriky, ktorá sa nachádza v Číne tiež asi 48 hodín.



Obrázok 3.19: Vizualizácia osadeného plošného spoju od spoločnosti JLCPCB

Komponenty ako sú vývojový modul ESP32, displej, moduly pre BME280 a CCS811 a ostatné, sme napájkovali pomocou trafopájky. Na nasledujúcom obrázku je zobrazený spoj osadený všetkými komponentami .



Obrázok 3.20: Plošný spoj obsahujúci všetky komponenty

Museli sme k ESP32 pridať ešte jeden spoj (oranžový kábel). Je to kvôli čítaniu napätia z batérie, aby sme mohli sledovať stav batérie. Piny označené ako ADC2 nefungujú spoľahlivo ak používame WiFi. Na to sme prišli, až keď sme skúšali softvér, a ak sme chceli čítať napätie z batérie z pinu GPIO27 (ADC2), tak to nefungovalo, a preto to museli zmeniť na pin GPIO32 (ADC1)[21].

Kapitola 4

Návrh senzora - SOFTWARE

4.1 Vlastnosti a spôsob prevedenia

Hlavnou vlastnosťou softvéru nášho monitorovacieho zariadenia je **merať a ukladať požadované veličiny**. Tieto údaje chceme mať jednoducho dostupné z hocijakého miesta na Zemi. To znamená, že musia byť uložené na internete. Za týmto účelom bude naše zariadenie dáta zbierať a ukladať na **cloud** a **online zobrazovať** na určitej internetovej platforme. Ďalej chceme dáta **zobrazovať na displeji** v reálnom čase. K tomu použijeme už vyššie zmienený displej typu e-ink. Vlastnosť ktorá je nevyhnutná pre naše zariadenie je **hlásenie nebezpečia**. Pri nebezpečne vysokej hladine CO₂ pošleme upozornenie o prekročení danej hranice na mobil a tiež implementovaný bzučiak vydá zvukovú signalizáciu. Okrem týchto vlastností ešte optimalizujeme **spotrebu energie**, budeme zobrazovať **aktuálny čas** a stav **batérie**. Keď si bodovo zhrnieme, čo musíme softvérovo implementovať, sú to nasledujúce úlohy:

1. Meranie veličín zo senzorov.
2. Komunikácia s Cloudom, zobrazovanie dát online a posielanie upozornení.
3. Grafické zobrazovanie dát na displeji.
4. Optimalizácia spotreby, aktualizácia času, stavu batérie a zvukova signalizácia.

Ako programovaciu platformu sme si zvolili ArduinoIDE, v ktorej je vďaka množstvu knižníc a jednoduchému užívateľskému rozhraniu možné implementovať nami zvolené vlastnosti zariadenia. Túto platformu sme si vybrali kvôli širokej podpore komunity s veľkým množstvom materiálov. Nasledujúce podkapitoly sa budú venovať spôsobu implementácie vyššie zmienených funkcií.

4.1.1 Meranie veličín zo senzorov

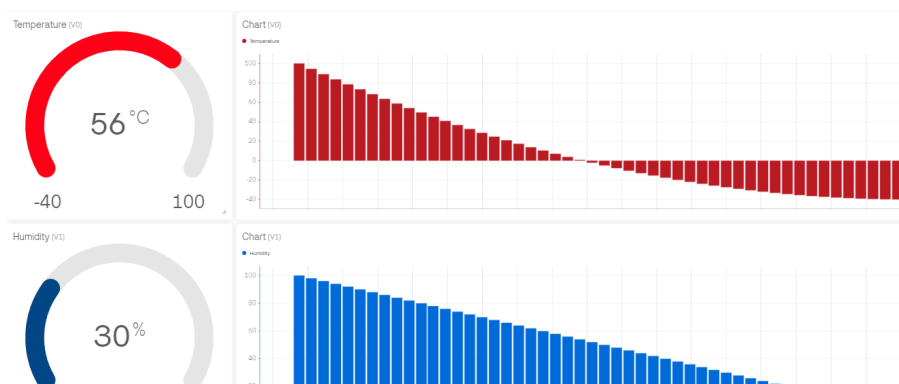
Ako už bolo spomenuté veličiny teplotu, vlhkosť, CO2 a TVOC meriame pomocou senzorov BME280 a CCS811. Na obidva senzory má **adafruit** knižnice, ktoré nám uľahčili prácu s nimi. Sensory nám len treba inicializovať a potom si pomocou funkcií na čítanie zapíšeme údaje zo senzorov do premenných. Dáta zo senzorov meriame raz za 10 sekúnd a do premenných ukládame priemernú hodnotu zo šiestich meraní, čiže priemernú hodnotu za minútu. Tú zobrazujeme na displej.

Knižnice použité pre prácu so senzormi:

- **Adafruit_Sensor.h** //obsahuje senzory od Adafruit
- **Adafruit_BME280.h** //pre prácu so senzorom BME280
- **Adafruit_CCS811.h** //pre prácu so senzorom CCS811
- **Wire.h** //pripojenie periférií (I2C pre senzory)

4.1.2 Komunikácia s Cloudom, zobrazovanie dát online a posielanie upozornení

Pre ukladanie dát na cloud použijeme IoT platformu **Blynk**. Je to platforma, ktorá je programátorsky aj užívateľsky veľmi prívetivá. Obsahuje svoj vlastný komunikačný protokol a po pripojení k Blynku môžeme dáta posilať, prijímať a zobrazovať v reálnom čase. V bezplatnej verzii tejto platformy sa dáta na cloude uschovávajú 1 týždeň a môžeme graficky zobrazovať až 5 meraných veličín. Dá sa požívať jak vo webovom prehliadači, tak v mobilnej aplikácii. Táto platforma obsahuje **grafanu** a teda merané veličiny sú v nej zobrazované graficky atraktívne. Na blynk cloud sa pripájame príkazom podobnému pripojeniu na WiFi. Po zadaní príkazu na pripojenie ku Blynk cloudu (Blynk.begin(auth, ssid, password)) sa najprv ESP32 pripojí na WiFi a následne za pomoci authorizačného tokenu sa pripojíme na cloud, kde si nastavíme grafiku zobrazovaných údajov.



Obrázok 4.1: Príklad zobrazovanie dát na platforme Blynk

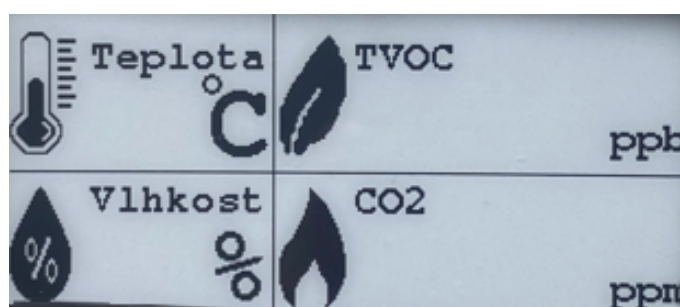
Ďalšou výhodou platformy Blynk je možnosť **posielanie upozornení** (eventov), pomocou funkcie (Blynk.logEvent()) dostupnej v knižnici pre Blynk. Treba len nastaviť, čo sa má stať za udalosť (napríklad poslať upozornenie do mobilu), stiahnuť mobilnú aplikáciu a potom už sa len v programe odkázať na túto udalosť v kóde. Okrem upozornení na mobil je možné nastaviť, aby Blynk poslal automaticky e-mail alebo SMS správu. Nevýhodou tejto platformy je, že ak zafažujeme cloud požiadavkami, tak nás blynk od portálu odpojí. No nakoľko sa budeme pripájať a posielat dáta na blynk len raz za 5 minút, táto nevýhoda nás neovplyvní.

Uvedieme si ešte použité knižnice pri komunikácii s Blynkom:

- **BlynkSimpleEsp32.h** //funkcie ktoré ponúka Blynk (posielanie, prijímanie dát, posielanie upozornení,...)
- **WiFi.h** //pripojenie ESP32 k WiFi
- **WiFiClient.h** //komunikácia cez WiFi

4.1.3 Grafické zobrazovanie dát na displeji

Už vieme ako dáta čítať zo senzorov, posielat na cloud a zobrazovať ich online. Čo ak ich ale chceme zobrazovať aj **offline**, bez pripojenia k internetu? K tomu sme pridali dvojfarebný displej e-ink veľkosti 2.9 palcov. Naše zariadenie je napájané z batérie a preto chceme, aby komponenty mali nízku spotrebu energie. A tak sme sa rozhodli meniť hodnoty na displeji len jedenkrát za minútu. A aj to neprekresľujeme celý displej, ale len niektoré **parciálne časti**. Meníme len hodnoty meraných veličín, času a stavu batérie. Ak displej neprekresľujeme spotreba je prakticky nulová. Na vytvorenie grafických symbolov sme obrázky formátovali na **bitovú mapu**. Symboly majú rozmery len 29x45 pixelov. Nakreslili sme ho pixel po pixeli v aplikácii maľovanie a uložili ako bitovú mapu. Na začiatku sme inicializovali displej, ktorý sa celý už v priebehu programu neprekresľuje, nemenná časť displeja vyzerá následovne (viz obr. 4.2).



Obrázok 4.2: Statická obrazovka po inicializácii displeja

Použité knižnice:

- **Wire.h** //pripojenie periférie (SPI pre displej)

- **GxEPD2.h** //grafika na rôzne veľkosti a druhy displejov
- **Fonts/FreeMonoBold24pt7b.h** //font hrubého písma veľkosti 24
- **Fonts/FreeMonoBold9pt7b.h** //font hrubého písma veľkosti 9
- **Fonts/FreeMono9pt7b.h** //font písma veľkosti 9

4.1.4 Zvuková signalizácia, aktualizácia času, stavu batérie a optimalizácia spotreby

Zvuková signalizácia

Teraz chceme zariadenie rozšíriť o ďalšie funkcionality. Prvou z nich je zvuková signalizácia pri prekročení život ohrozujúcej hodnoty CO₂.

CO ₂ [ppm]	Air Quality
2100	BAD Heavily contaminated indoor air Ventilation required
2000	
1900	
1800	
1700	
1600	MEDIOCRE Contaminated indoor air Ventilation recommended
1500	
1400	
1300	
1200	
1100	FAIR
1000	
900	
800	GOOD
700	
600	EXCELLENT
500	
400	

Obrázok 4.3: Tabuľka indexu CO₂[22]

Zvukovú signalizáciu chceme použiť len v nevyhnutných prípadoch. Preto sme ju pevne nastavili na hodnotu 2000 [ppm]. Pri prekročení tejto hranice môžu nastať **zdravotné komplikácie**. Naše zariadenie je vyrobené na monitorovanie kvality vzduchu **v domácnosti**, kde sa táto hodnota môže vyskytovať. Pri tejto hodnote bzučiak trikrát zapípa a ak do minúty neklesne hladina pod limit, tak nás informuje znovu.

Aktualizácia času

Čas budeme aktualizovať v pravom hornom rohu displeju (viz kapitola testovanie). Nechceme však byť stále pripojený k internetu a sledovať aktuálny čas. A preto čas budeme zaznamenávať z internetu len vtedy, keď budeme posielat dáta na cloud Blynk (raz za 5 minút) a mimo to budeme inkrementovať premennú minút a hodín každých šesť meraní (teda každú minútu). Aktuálny čas z internetu získavame pomocou NTP protokolu tým, že sme si stiahli dáta z NTP serveru.

Použité knižnice:

- **time.h** //definuje premenné pre manipuláciu s časom a dátumom
- **WiFi.h** //pripojenie k internetu

■ Stav batérie

Aby sme vedeli, aká je aktuálna kapacita batérie v našom zariadení, budeme sledovať napätie batérie, ktoré je pri plnom nabití niečo cez 3.3 [V] a čím je viac vybitá, tak tým menšie napätie má. Napätie nemusíme nijak upravovať na deliči, pretože ADC piny 3.3 [V] na vstupe zvládnu prečítať. Napätie sledujeme na analogovom pine číslo 34 a pomocou funkcie `analogRead()` ho prevedieme na digitálnu hodnotu. Hodnotu graficky signalizujeme v pravom hornom okraji displeja vedľa ukazateľa času.

■ Optimalizácia spotreby

Vlastnosť, ktorá v našom zariadení spotrebúva najviac energie je WiFi pripojenie a komunikácia cez internet s cloudom. Ak chceme predĺžiť výdrž batérie musíme byť pripojený na WiFi čo najkratšiu dobu. ESP32 má rôzne režimy chodu:

Režim	Funkcie	Spotreba
Aktívny	WiFi, Bluetooth zapnuté	90 - 240 [mA]
Modem-sleep	CPU zapnuté	5-10 [mA]
Light-sleep	CPU vyčkáva	0.8 [mA]
Deep-sleep	ULP zapnuté len RTC pamäť	0.15 [mA] 10 [μ A]
Hibernácia	RTC časovač	5 [μ A]

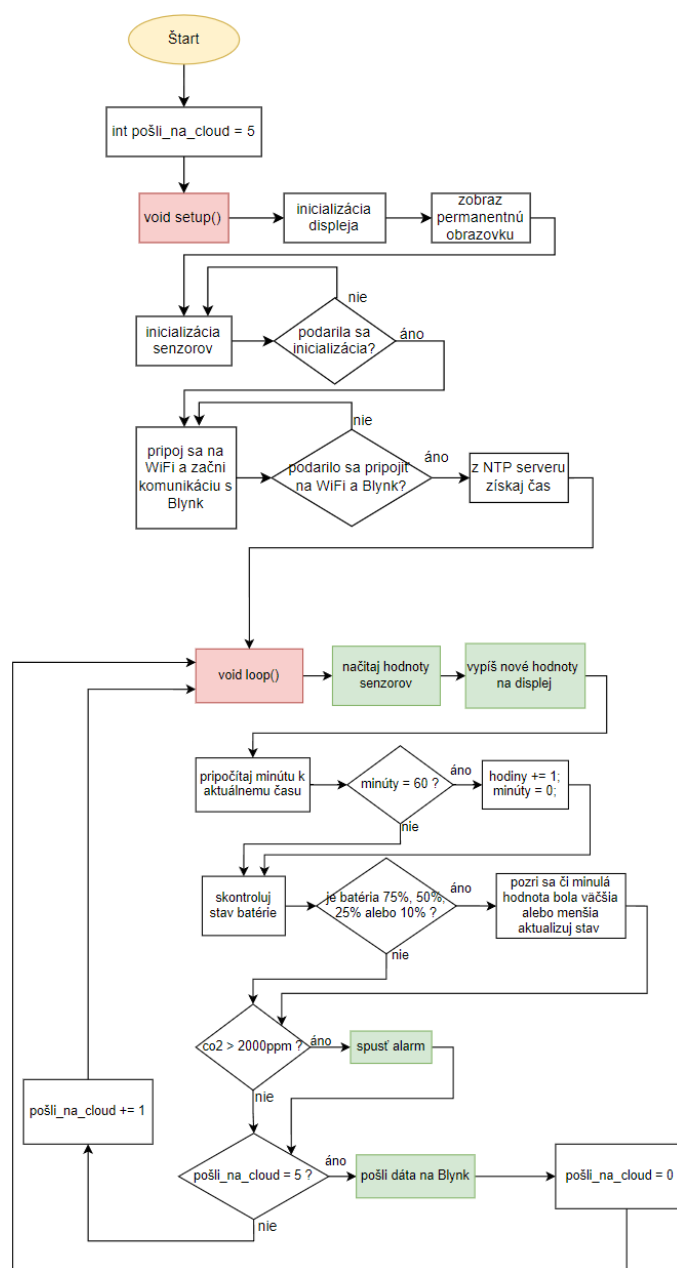
Tabuľka 4.1: Funkcie a ich spotreba v rôznych režimoch ESP32[23]

V našom programe budeme používať dva režimy, **aktívny režim** a režim **modem-sleep**. Modem-sleep je v podstate aktívny režim, ale s vypnutými WiFi a bluetooth. Znížením frekvencie CPU môžeme ušetriť ďalšiu energiu. Režim modem-sleep použijeme tak, že akonáhle sa raz za 5 minút pripojíme na Blynk a pošleme tam dáta, tak hneď po tom čo sa táto akcia vykoná sa odpojíme od Blynku a vypneme WiFi na 5 minút. Časový úsek odkedy sa WiFi zapne, pošlú sa dáta na Blynk až dokedy sa WiFi vypne trvá približne 0.5 [s]. Tým že sa od WiFi odpájame znižujeme spotrebu približne 10-násobne.

Okrem režimu v ktorom ESP32 používame tiež optimalizujeme spotrebu tým, že pri prekresľovaní hodnôt na displeji ho neprekresľujeme celý, ale len **parciálne** obdĺžniky veľkosti zobrazenia hodnoty, ktorú chceme prekresliť.

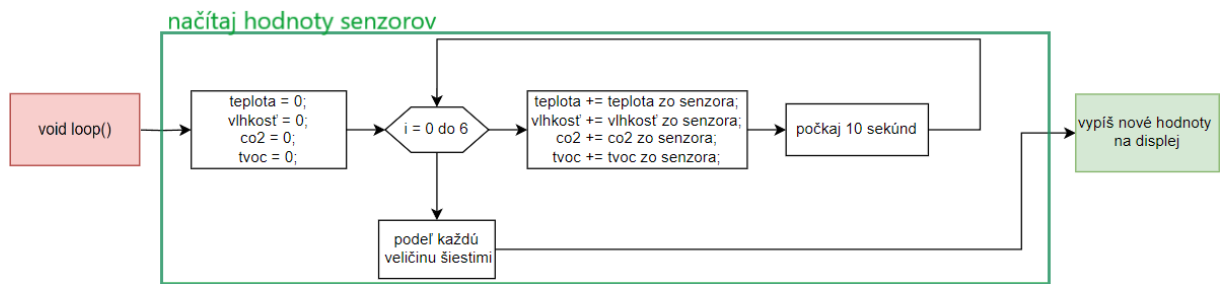
Nakoľko sme zapojenie modelovali tak aby bolo zariadenie **používateľné aj počas nabíjanie**, tak nie je nevyhnutné viac predĺžovať výdrž batérie.

4.2 Vývojové diagramy

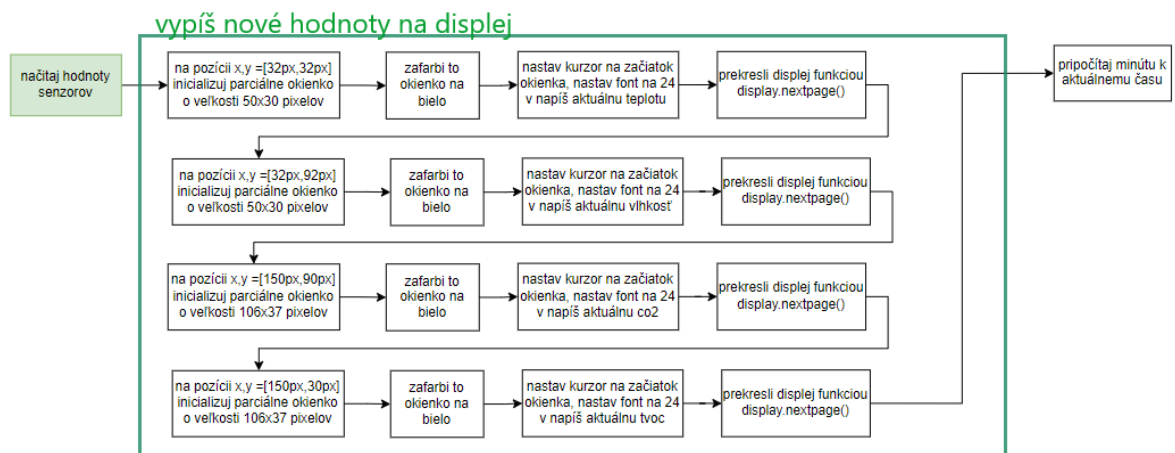


Obrázok 4.4: Hlavný vývojový diagram programu

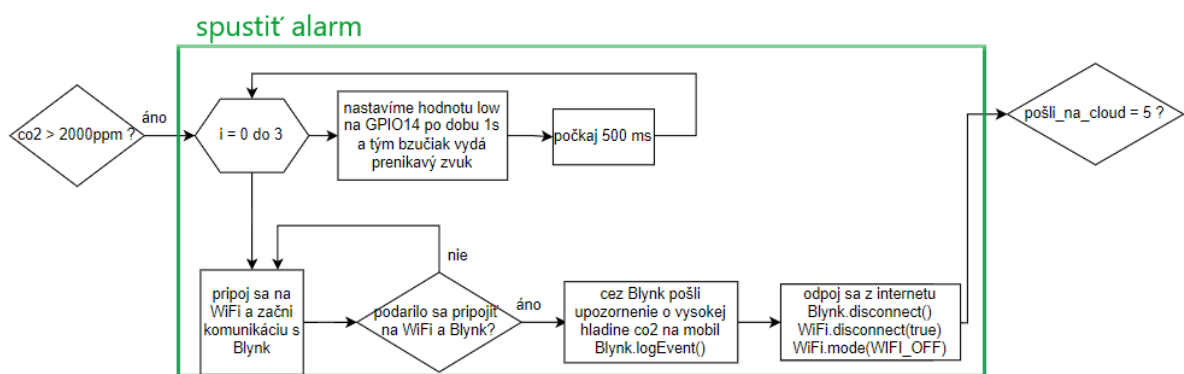
Bloky zobrazené na diagrame zelenou farbou sú trochu komplexnejšie a pre prehľadnosť princíp ich fungovania popíšeme na nasledujúcich stranách.



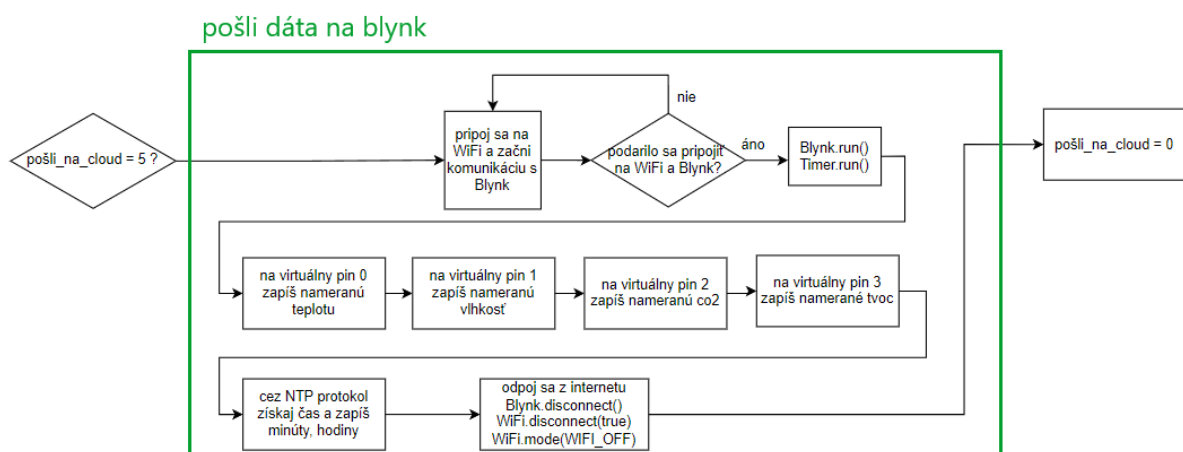
Obrázok 4.5: Vývojový diagram funkcie pre čítanie zo senzorov



Obrázok 4.6: Vývojový diagram funkcie na vypísanie hodnôt na displej



Obrázok 4.7: Vývojový diagram spustenia alarmu a následného odoslania upozornenia



Obrázok 4.8: Vývojový diagram posielania dát na Blynk cloud

*POZN.: "**pošli_na_cloud**" - je premenná, ktorá sa ikrementuje každú minútu o jedna a ak je rovná piatim znamená že ubehlo 5 minút od posledného poslania na cloud a teda od pripojenia na internet (dáta posielame na cloud raz za 5 minút).

Kapitola 5

Testovanie

Zariadenie na monitorovanie kvality ovzdušia v domácnosti máme navrhnuté a zostrojené. V tejto kapitole vyskúšame funkčnosť vlastností zariadenia. Pozrieme sa na grafické rozhranie displeja a ukážeme si zobrazené hodnoty vo webovom prehliadači na platforme Blynk, tiež v mobilnej aplikácii. V neposlednom rade vyskúšame, či nám Blynk naozaj posiela upozornenia na mobil pri prekročení hodnoty CO₂.

■ Vzhľad zariadenia po zapnutí

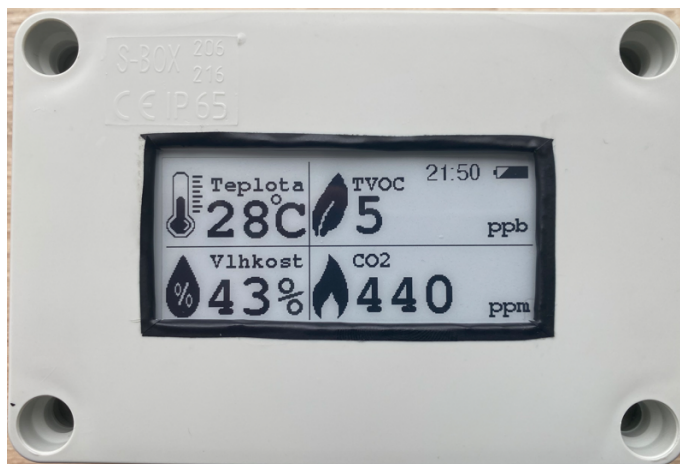
Keď zariadenie zapneme, tak sa nám zobrazí inicializačná obrazovka, na ktorej ešte niesú namerané veličiny. Je tam zobrazené to, čo sa už počas celého chodu zariadenia nemení, ani neprekresľuje. Všetky komponenty sme umiestnili do plastovej priemyslovej krabičky, do ktorej sme vyrezali otvory pre displej, napájanie a spínač.



Obrázok 5.1: Vzhľad nášho zariadenia hneď po zapnutí

■ Zobrazené hodnoty na displeji

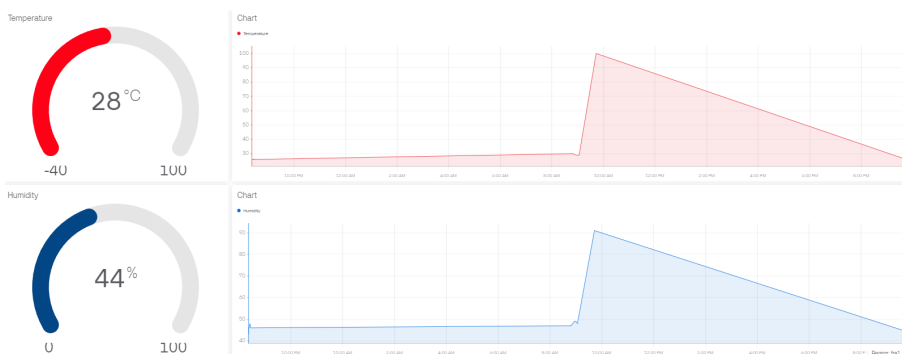
Keď zariadenie zapneme, tak sa minútu merajú hodnoty, z ktorých sa vyratá priemerná hodnota pre väčšiu presnosť. Po minúte sa nám začnú meniť hodnoty na displeji v závislosti na meraných veličinách. Tiež začneme merať čas a stav nabitia batérie.



Obrázok 5.2: Vzhľad nášho zariadenia po jednom meraní

■ Grafické zobrazenie meraných veličín na cloude Blynk v prehliadači

Offline sa nám dáta na displeji zobrazujú podľa očakávaní. Teraz sa pozrieme ako sa dáta zobrazujú v reálnom čase vo webovom prehliadači na IoT platforme Blynk.



Obrázok 5.3: Grafické zobrazenie teploty a vlhkosti na cloude na platforme Blynk

V Blynku je možné vybrať si rôzne možnosti zobrazenia, ako sú grafy, kruhové diagramy, digitálne zobrazenie dát a ďalšie. V platenej verzii je možností zobrazovania oveľa viac. My sme si vybrali zobrazovanie takýmto kruhovým widgetom a tiež grafom, aby sme sa vedeli pozrieť do histórie, ako sa hodnoty veličín menili v čase (viz. obr. 5.3). Graf má mierku aktuálnu, 30-minutovú, hodinovú, dennú a týždennú.

To boli zobrazované veličiny teploty a vlhkosti, ešte otestujeme grafiku CO₂ a TVOC.

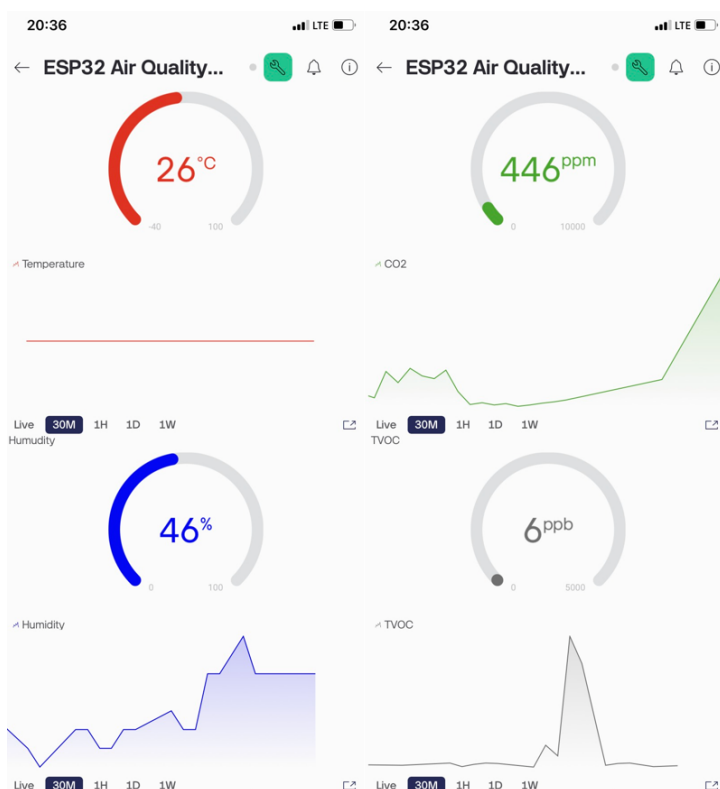


Obrázok 5.4: Grafické zobrazenie CO₂ a TVOC cloudu na platforme Blynk

Hranice zobrazenia je možné ľubovoľne meniť a na základe toho sa mení veľkosť grafu a zobrazenie dát.

■ Grafické zobrazenie meraných veličín na cloudu Blynk v mobilnej aplikácii

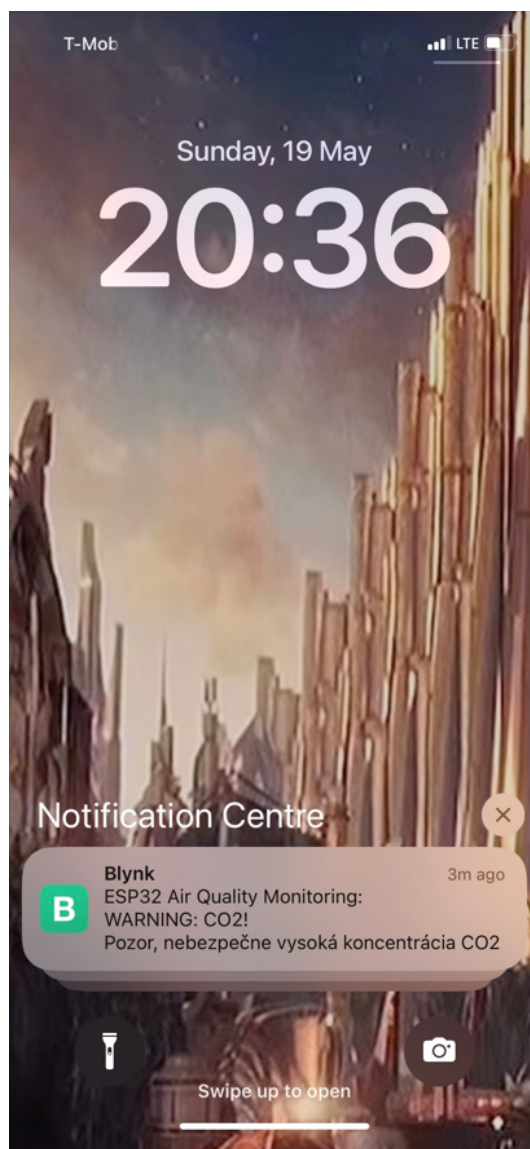
Okrem zobrazovania meraných veličín v prehliadači si ich zobrazíme aj v mobilnej aplikácii Blynk.



Obrázok 5.5: Grafické zobrazenie veličín v mobilnej aplikácii Blynk

■ Upozornenie v podobe hlásenia na mobil

V neposlednej rade sme otestovali funkciu alarmu, ktorá signalizuje prekročenie hodnoty oxidu uhličitého cez 2000 [ppm]. Po prekročení tejto hranice môžu nastať vážne zdravotné komplikácie a preto sme sa rozhodli implementovať funkciu, ktorá upozorní užívateľa na prekročenie tejto hodnoty. Máme implementovaný bzučiak, ktorý ak dosiahneme túto hranicu trikrát prenikavo zapípa. Ak do minúty hodnota oxidu uhličitého neklesne pod 2000 [ppm] znovu trikrát zapípa, a takto dookola, kým hladina neklesne, alebo zariadenie nevypneme. Okrem bzučiaka posielame varovné upozornenie na mobil prostredníctvom platformy Blynk, ktorá má túto funkciu implementovanú v knižnici pre ESP32 (Blynk.event()).



Obrázok 5.6: Upozornenie poslané na mobil pri nebezpečnej hodnote CO2

Kapitola 6

Záver

Zariadenie je zostrojené a otestované. V poslednej kapitole zhrniem a zhodnotím, čo som vykonal, a či sa mi podarilo všetko čo som si dal za cieľ. Prípadne ako by som zariadenie mohol ďalej rozširovať a vyvíjať. Nakoniec pridám svoje dojmy a pocity z tejto práce.

6.1 Postup zostrojenia

Po prieskume trhu zariadení na monitorovanie ovzdušia som sa zamerlal na to, ako by som mohol zostrojiť **komplexnejšie** mobilné zariadenie na monitorovanie ovzdušia v domácnosti. Rozhodol som sa zostrojiť zariadenie, ktoré bude monitorovať teplotu, vlhkosť, CO₂ a TVOC. Tieto veličiny **ukladám online na cloud** a tiež zobrazujem na **displeji**. Pri nebezpečne veľkej koncentrácii CO₂ užívateľa jasne **upozorní** na túto prekročenú hladinu.

Pri zostrojení som najskôr **vybral komponenty**, ktoré chcem použiť, ako senzory, mikrokontrolér, displej, atď. Bližšie som sa pozrel na **komunikáciu** medzi komponentami. Keďže je napájané z batérie, tak som navrhol **manažment napájania** LiFePO₄ batérie. V prostredí KiCad som zapojil napájanie, periférie a ďalšie prvky obvodu k vývojovému modulu ESP32 DEVKIT V1 - DOIT.

K vyrobenému a osadenému plošnému spoju som dopájkoval použité moduly a išiel som na programovanie softvéru. V prostredí **Arduino IDE** za použitia určitého počtu knižníc som najskôr spracoval funkciu na čítanie meraných veličín **zo senzorov**. V ďalšom kroku som tieto hodnoty vypísal na **displej** a následne som dáta poslal na **cloud Blynk**, kde sú graficky zobraziteľné. Po týchto krokoch som sa pustil do programovania **upozornenia prekročenia CO₂**. Nakoniec som ešte pridal meranie a zobrazovanie času a stavu batérie.

Po naprogramovaní som všetky komponenty umiestnil do mnou pripraveného úložného boxu a na produkte som **otestoval** všetky požadované vlastnosti.

6.2 Obmedzenia práce

Pri zhotovovaní zariadenia som narazil na niekoľko **prekážok**, s ktorými som sa v priebehu práce musel vysporiadať. Pomerne dosť som sa trápil s obmedzenými skúsenosťami s **mikroelektronikou** a **hardvérom** celkovo, nakoľko sa na programe kybernetika a robotika sa s hardvérom stretávam minimálne. Naučil som sa pracovať v prostredí pre **vytváranie plošných spojov** a ako plošný spoj **navrhnuť**. Spoj som chcel osadiť sám, ale kvôli minimálnym skúsenostiam s pájkovaním som si ho nechal osadiť v spoločnosti JLCPCB.

6.3 Zhodnotenie cieľov a prínos práce

Po prekonaní prekážok, ktoré má pri zostrojovaní postretli sa mi nakoniec podarilo zariadenie zostrojiť a spojazdniť. Ciele, ktoré som si stanovil v úvode:

- Meranie požadovaných veličín
- Zobrazovanie na displej
- Ukladanie a zobrazovanie na Cloude
- Varovná signalizácia pri nebezpečí
- Napájanie z batérie v chode nepretržite 24 hodín

sa mi úspešne podarilo do zariadenia implementovať. K týmto funkciám som ešte pridal zobrazovanie aktuálneho času a stavu batérie ako rozšírenie zariadenia.

Ako už bolo spomenuté v úvode, táto práca slúži k tomu, aby užívateľ nebol vystavený nebezpečnému prostrediu. Vďaka tomuto zariadeniu sme schopný odhadnúť prípadné nebezpečné zloženie vzduchu a máme dostatok času na preventívne opatrenia (odísť z miestnosti, vyvetrať). Zariadenie má tiež priaznivý dopad na optimalizáciu pracovných podmienok, v ktorých ľudia trávajú najviac času.

Tu je cenový odhad zariadenia:

Komponent	Cena [czk]
Senzory	1000
Displej	500
Mikrokontrolér	250
Ostatné	250
Spolu	2 000

Tabuľka 6.1: Odhadovaná cena zariadenia bez zohľadnenia nákladov na prácu

6.4 Odporúčanie na rozšírenie

Táto práca ponúka široku paletu rozšírení nakoľko je to komplexné mobilné zariadenie pripojené k internetu. Uvedieme si pár možných rozšírení, ktoré mi napadli, že by mohli byť užitočné:

- Ako možnosť sa naskytá zostrojiť viac totožných zariadení, ktoré budú súčasne posielať dáta na spoločnú platformu a tým získame prehľad o kvalite ovzdušia vo viacerých miestnostiach.
- Pridaním senzorov meriacich aj iné veličiny je možné zariadenie rozšíriť o ďalšie funkcionality.
- Ďalšou myšlienkou je pridanie chytrého spínača na ventiláciu v miestnosti a tú spustiť pri prekročení daných hodnôt.

To je len pár nápadov, ktoré by bolo možné implementovať a tým ešte rozšíriť užívateľské rozhranie zariadenia. Tieto návrhy som nechal v štádiu odporúčaní na rozšírenie, pretože by sa tým výrazne zvýšila cena zariadenia.

6.5 Osobné zhodnotenie

Tento projekt mi prišiel veľmi zaujímavý, pretože som získal prehľad o tom ako to funguje pri vytváraní zariadenia od základov až po otestovanie. Videl som aký je svet mikroelektroniky pestrý a tiež zložitý. Na programe kybernetika a robotika, ktorý študujem, sa s hardvérom stretávam minimálne a preto mi táto práca priniesla skúsenosti s návrhom hardvéru. Som vďačný za túto možnosť prepojiť svety softvéru s hardvérom.



Literatúra

- [1] Datacomp.sk. *AQARA TVOC Air Quality Monitor*. [Online]. 2024. https://datacomp.sk/aqara-tvoc-air-quality-monitor-aaqs-s01-zigbee-senzor-kvality-vzduchu_d427080.html
- [2] Mironet.cz. *Honeywell/Resideo R200C2-E*. [Online]. 2024. <https://www.mironet.cz/honeywell-home-by-resideo-r200c2e-detektor-monitor-co2-cz-sk-hu-zvuk+dp549795/>
- [3] Mironet.cz. *AirBox01*. [Online]. 2024. <https://www.mironet.cz/iqtech-smartlife-wifi-airbox01-wifi-senzor-kvality-vzduchu-teplota-vlhkost-co2-voc-hco+dp601198/>
- [4] Airthings.com. *Wave Plus*. [Online]. 2024. <https://www.airthings.com/en/wave-plus>
- [5] Socialcompare.com. *RP2040 vs STM32 vs ESP32*. [Online]. 2024. <https://socialcompare.com/en/comparison/rp2040-vs-stm32-vs-esp32-vs-esp8266>
- [6] Randomnerdtutorials.com. *ESP32 Development Board*. [Online]. 2024. <https://randomnerdtutorials.com/getting-started-with-esp32/>
- [7] Dratek.cz *BME280*. [Online]. 2024. <https://dratek.cz/arduino/1361-bme280-modul-mereni-teploty-vlhkosti-a-barometrickeho-tlaku-precizni.html>
- [8] Adafruit.cz. *CCS811*, [Online]. 2024. <https://www.adafruit.com/product/3566>
- [9] Waveshare.com. *296x128, 2.9inch E-Ink display module*. [Online]. 2024. <https://www.waveshare.com/2.9inch-e-paper-module.htm>
- [10] Manorshi.com. *Analýza piezo bzučáků*. [Online]. 2024. <https://www.manorshi.com/cs/Analysis-of-Piezo-Buzzers-An-Overview-id3165733.html>
- [11] Onpira.cz. *Piezo bzučák Ø 23mm*. [Online]. 2024. <https://www.onpira.cz/zbozi/piezo-bzucak-3-24v-nepretrzity-ton/>

- [12] Wikipedia.org. *Lithium-železo-fosfátový akumulátor*. [Online]. 2024.
https://cs.wikipedia.org/wiki/Lithium-železo-fosfátový_akumulátor
- [13] Akkuteile.de. *3.3V 2000mAh LiFePO4*. [Online]. 2024.
https://www.akkuteile.de/en/ifr22650-3-2v-3-3v-2000mah-lifepo4-lithium-iron-phosphate-battery_100728_2460
- [14] Wikipedia.org. *Serial Peripheral Interface*. [Online]. 2024.
https://cs.wikipedia.org/wiki/Serial_Peripheral_Interface
- [15] Foolish Engineer. *Basics of SPI communication*. [Online]. 2024.
<https://www.youtube.com/watch?v=xogsRnnhK44>
- [16] Foolish Engineer. *Basics of I2C communication*. [Online]. 2024.
<https://www.youtube.com/watch?v=pbqk5yqbfuw>
- [17] Hackday.com. *ALL ABOUT USB-C: RESISTORS AND EMARKERS*. [Online]. 2024.
- [18] Components101.com. *TP5000 datasheet*. [Online]. 2024.
https://components101.com/sites/default/files/component_datasheet/TP5000-Datasheet.pdf
- [19] Dratek.cz. *DC-DC 1,8V-5V na 3.3V*. [Online]. 2024.
<https://dratek.cz/arduino/2032-step-down-up-menic-dc-dc-1.8v-5v-na-3.3v.html>
- [20] Jlcpcb.com. *Order PCB now*. [Online]. 2024.
<https://cart.jlcpcb.com/quote>
- [21] Espressif.com. *Analog to Digital Converter*. [Online]. 2024.
<https://docs.espressif.com/projects/esp-idf/en/v4.2/esp32/api-reference/peripherals/adc.html>
- [22] Iqhome.org. *Indoor Air Quality*. [Online]. 2024.
https://www.iqhome.org/index.php?route=extension/d_blog_module/post&post_id=17
- [23] Randomnerdtutorials.com. *ESP32 Deep Sleep*. [Online]. 2024.
<https://randomnerdtutorials.com/esp32-deep-sleep-arduino-ide-wake-up-sources/>



Použité skratky

- **CO₂** - carbon dioxide (oxid uhličitý)
- **TVOC** - Total Volatile Organic Compounds (těkavé organické látky)
- **e-ink** - Elektornický inkoust
- **RAM** - Random Access Memory
- **kB** - Kilobajt
- **MHz** - Megahertz
- **I²C** - Inter-Integrated Circuit
- **μA** - Mikroampér
- **SRAM** - Static Random Access Memory
- **ROM** - Read-Only Memory
- **Mbps** - Megabits per second
- **SPI** - Serial Peripheral Interface
- **hPa** - Hektopascal
- **ppb** - parts per billion
- **ppm** - parts per million
- **LiPo** - Lítium polymer
- **LiFePO₄** - Lítium železofosfát
- **NiMH** - Nikel-metalhydrid
- **Wh** - Watthodina
- **mAh** - Miliampérodina
- **UART** - Universal asynchronous receiver-transmitter

- **V** - Volt
- Ω - Ohm
- **PMOS** - P-type metal-oxide-semiconductor
- **ADC** - Analog to digital converter
- **GPIO** - General-purpose input/output
- **IoT** - Internet of Things
- **NTP** - Network Time Protocol
- **dB** - Decibel