

Bakalářská práce



České
vysoké
učení technické
v Praze

F3

Fakulta elektrotechnická
Katedra měření

System simulující chytrou domácnost

Kamil Herman

Vedoucí: Ing. Jakub Svatoš, Ph.D.
Studijní program: Kybernetika a robotika
Květen 2023

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Herman** Jméno: **Kamil** Osobní číslo: **492364**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávací katedra/ústav: **Katedra měření**
Studijní program: **Kybernetika a robotika**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Systému simulující inteligentní domácnost

Název bakalářské práce anglicky:

A System Simulating a Smart Home

Pokyny pro vypracování:

Prostudujte možnosti návrhu systému simulujícího inteligentní domácnost.
Vytipujte vhodné senzory a prvky používané v chytré domácnosti včetně možností komunikace s nadřazenou jednotkou.
Na základě nabytých znalostí navrhnete a realizujete systém, který bude simulovat monitoring a ovládání chytré domácnosti.
Součástí práce bude i jednoduchá aplikace pro zobrazení a ovládání jednotlivých prvků chytré domácnosti.

Seznam doporučené literatury:

[1] Li Jiang, Da-You Liu and Bo Yang, "Smart home research," Proceedings of 2004 International Conference on Machine Learning and Cybernetics (IEEE Cat. No.04EX826), 2004, pp. 659-663 vol.2, doi: 10.1109/ICMLC.2004.1382266.
[2] Marikyan, D., Papagiannidis, S., & Alamanos, E. (2019). A systematic review of the smart home literature: A user perspective. Technological Forecasting and Social Change, 138, 139-154.
<https://doi.org/10.1016/j.techfore.2018.08.015>

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. Jakub Svatoš, Ph.D. katedra měření FEL

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

Datum zadání bakalářské práce: **30.01.2023**

Termín odevzdání bakalářské práce: _____

Platnost zadání bakalářské práce:

do konce letního semestru 2023/2024

Ing. Jakub Svatoš, 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

Poděkování

V první řadě bych rád poděkoval vedoucímu práce panu Ing. Jakubu Svatošovi Ph.D. za vedení po celou dobu vypracovávání této bakalářské práce, za jeho rady, přátelský přístup, jeho celkový strávený čas a skvělý přístup ke všem problémům. Dále bych chtěl poděkovat panu doc. Ing. Janu Fischerovi, CSc. za příležitostnou výpomoc při vzniklých problémech. V neposlední řadě bych chtěl poděkovat rodině, přátelům a přítelkyni za veškerou podporu v průběhu mého celého studia.

Prohlášení

Prohlašuji, že jsem zadanou bakalářskou prací vypracoval sám, pouze za pomoci vedoucího práce a odborné literatury, která je uvedena níže. Souhlasím se zapůjčováním a zveřejňováním práce.

V Praze dne

Abstrakt

Cílem této bakalářské práce je realizace simulace inteligentní domácnosti spolu s návrhem komunikace pro vizualizaci dat. Práce pojednává o návrhu právě takového systému, o jeho doplnění o vhodné senzory, ovládací prvky, centrální jednotku a komunikaci. V teoretické části jsou vysvětleny základní principy fungování a sběrnice použitých prvků. Dále je zde také uveden 3D model pro krabičku, kde bude celý systém uchován. Následuje část o použitém softwaru spolu s ukázkami jednotlivých programů doplněných o jejich vysvětlení. V této části je také uvedena ukázka vizualizace naměřených dat. Poslední částí práce je praktické proměření funkčnosti celého návrhu a následné shrnutí s doplněním budoucího rozvoje a vylepšení práce.

Klíčová slova: Chytrá domácnost, PWM, I_2C , WIFI, senzory, ovládací prvky

Abstract

The main goal of this bachelor's thesis is to implement the simulation of a smart home, together with communication and the design of data visualization. After the introduction, the work deals with the design of just such a system, choosing the suitable sensors, control elements, central unit, and communication. The theoretical part explains the basic principles of operation and communication over the bus of used sensors and central unit. There is also a model of the cover box where the entire system will be kept. The following is a section about the used software, together with examples of individual programs supplemented with comments. This part also shows an example of the visualization with the measured data. The last section of the work is a practical functionality measurement of the entire realization and a subsequent summary with the possible future development.

Keywords: Smart home, PWM, I_2C , WIFI, sensors, control elements

Obsah

1 Úvod	1	5 Simulace systému	33
1.1 Motivace	1	5.1 Teplota	33
1.2 Obecný úvod	1	5.2 CO_2	34
1.3 Kapitoly	2	5.3 Vlhkost	34
		5.4 Tlak	35
2 Návrh systému	3	6 Závěr	37
2.1 Postup při návrhu	3	7 Přílohy	39
2.2 Měřené veličiny	4	8 Popis příloh	41
2.2.1 Měření CO_2	4	Literatura	43
2.2.2 Měření teploty	5		
2.2.3 Měření vlhkosti	5		
2.2.4 Měření tlaku	5		
2.2.5 Simulace osvětlení	6		
2.2.6 Simulace větrání	6		
2.3 Teoretický rozbor	6		
2.3.1 Senzory s digitálním výstupem	6		
2.3.2 Centralizovaný systém	6		
3 Hardware	7		
3.1 Vývojová deska	8		
3.1.1 NUCLEO-L053R8	8		
3.2 Senzory	9		
3.2.1 SCD41	9		
3.2.2 BMP180	11		
3.3 Ovládací prvky	13		
3.3.1 Keyes KY-016	13		
3.3.2 D4184	14		
3.4 Komunikace	16		
3.4.1 ESP8266	16		
3.5 3D návrh modelu krabičky	18		
3.5.1 Software	18		
3.5.2 3D model	18		
4 Software	19		
4.1 Použitý software	19		
4.1.1 Software pro programování prvků měřicího systému	19		
4.1.2 Software pro vizualizaci dat	19		
4.2 Firmware	20		
4.2.1 Nastavení projektu	20		
4.2.2 Hlavní program	21		
4.2.3 Senzor SCD41	23		
4.2.4 Senzor BMP180	25		
4.2.5 LED modul	26		
4.2.6 MOSFET modul	27		
4.2.7 WIFI modul	29		
4.3 Vizualizace dat	31		

Obrázky

2.1	Obecné schéma návrhu systému	3	5.4	Graf tlaku	35
3.1	Hardwarové blokové schéma	7	7.1	Vývojový diagram fungování firmwaru	40
3.2	Nucleo deska L053R8	8			
3.3	Schéma zapojení sběrnice I_2C [14]	9			
3.4	Senzor SCD41	10			
3.5	Princip NDIR měření [16]	11			
3.6	Senzor BMP180	12			
3.7	Piezo-odporové měření tlaku [20]	13			
3.8	LED modul Keyes KY-016	14			
3.9	MOSFET modul D4184	15			
3.10	Elektronické schéma D4184 modulu	15			
3.11	Struktura MOSFET tranzistoru [21]	16			
3.12	TCP/IP WIFI modul	17			
3.13	Usměrňovač napětí LD1117	17			
3.14	3D model krabičky	18			
4.1	Zapojení pinů na vývojové desce	21			
4.2	Nastavení hodinového signálu	21			
4.3	Začátek programu	22			
4.4	Nekonečná smyčka programu	22			
4.5	Definování adres a příkazů	23			
4.6	Počáteční nastavení senzoru	24			
4.7	Počítání jednotlivých veličin	25			
4.8	Kalibrace senzoru BMP180	25			
4.9	Vyčítání surových dat pro výpočet tlaku	26			
4.10	Počáteční spuštění a nastavení PWM pro LED modul	26			
4.11	Nastavení diod podle hodnoty CO_2	27			
4.12	Počáteční spuštění a nastavení PWM pro MOSFET modul	28			
4.13	Nastavení větráku podle hodnoty CO_2	28			
4.14	Připojení k WIFI síti	29			
4.15	Posílání dat na platformu Thingspeak	30			
4.16	Vizualizace: aktuální hodnoty	31			
4.17	Vizualizace: signalizace	31			
4.18	Vizualizace: grafy	32			
5.1	Graf teploty	34			
5.2	Graf CO_2	34			
5.3	Graf vlhkosti	35			

Tabulky

2.1 Tabulka rizikových hodnot CO_2 [15]	5
--	---

Kapitola 1

Úvod

1.1 Motivace

V dnešní době můžeme čím dál více vidět potřebu chytrých technologií v naší domácnosti pro usnadnění každodenního života. Se zvyšující životní úrovní zároveň stoupá potřeba lidí ovládat svou domácnost pouze za použití nějakého zařízení, například ovládat osvětlení pomocí stisku jednoho tlačítka na mobilu. Zároveň lidé chtějí vědět, jestli doma mají optimální teplotu, kvalitu ovzduší a celkově ideální podmínky pro své zdraví. Všechny tyto potřeby dokáže vyřešit inteligentní domácnost [1], která se stává čím dál více cenově dostupnou i pro běžné uživatele.

1.2 Obecný úvod

Tato bakalářská práce se zabývá návrhem právě takové chytré domácnosti. Cílem práce je nejdříve navrhnout vhodný systém, ten doplnit o příslušné periferie, které budou pokrývat simulaci základních potřeb běžné domácnosti. Následně všechny periferie oživit, naprogramovat komunikaci systému s chytrým zařízením, kterým půjdou ovládat a vyčítat z nich potřebné veličiny a zároveň navrhnout uživatelské rozhraní, přes které vše půjde řídit. Práce může pomoci při návrhu podobného systému, avšak není primárně určena jakožto návod k realizaci.

Chytré domácnosti budou nevyhnutelnou součástí našich životů v budoucnu i ze zdravotních důvodů. Dle světové zdravotní organizace 650 milionů lidí žije s nějakým zdravotním omezením, jako je například cukrovka, rakovina nebo HIV/AIDS [3]. Není možné, aby všichni tito lidé trvale pobývali ve zdravotnických zařízeních. Odpovědí na danou problematiku mohou být právě chytré domácnosti, spolu s domácí zdravotní péčí, které už nyní častokrát dokáží měřit velmi detailní a přesné hodnoty, důležité právě pro zdraví člověka. Mohou tedy posloužit jako jistá výpomoc a ulehčení ve vytíženém zdravotnictví. I z toho důvodu se dá předpokládat, že chytrá zařízení brzy budou v téměř každé domácnosti. Tento projekt by konkrétně měl sloužit jako ukázka vyčítání dat a komunikace na dnech otevřených dveří a podobných akcích fakulty.

■ 1.3 Kapitoly

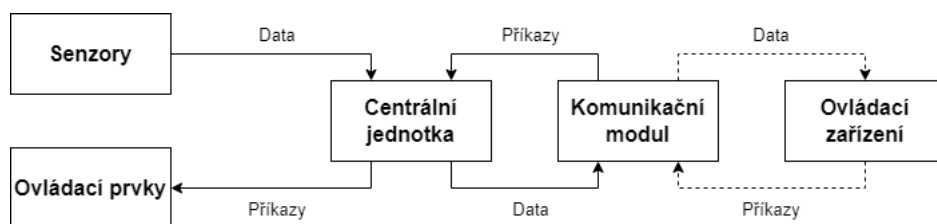
V první části se práce zaměřuje na samotný návrh systému [2], výběr vhodných periférií k jeho realizaci a potřebnou teorii k porozumění návrhu. Ve druhé kapitole pak pojednává o použitém hardwaru, o principech a vlastnostech jednotlivých prvků spolu s návrhem 3D modelu pro úschovu celého systému. Dále se probírá návržení firmwaru pro jednotlivé senzory, ovládací prvky a naprogramování komunikace přes WIFI modul. Zároveň je zde ukázána vizualizace dat ve webové platformě Thingspeak. V následující kapitole je uvedena funkčnost návrhu, proměření hodnot v běžném prostředí a porovnání s reálnými hodnotami. Poslední částí je celkové zhodnocení výsledků a návržení budoucího rozvoje celého projektu.

Kapitola 2

Návrh systému

V této kapitole bude detailně probrán návrh systému a jak při takovém návrhu postupovat. Dále se zde budou hlouběji řešit vybrané systémové metody, tedy jejich základní vlastnosti, výhody a nevýhody.

Obecné schéma fungování návrhu systému lze vidět níže na obrázku 2.1. Pro klasickou fyzickou komunikaci přes kabel je použita spojitá čára. Pro komunikaci skrz bezdrátovou technologii je naopak použita čára přerušovaná.



Obrázek 2.1: Obecné schéma návrhu systému

Pro začátek je dobré si říct, jak obecně postupovat při návrhu právě takového systému a co vše bylo vybráno v tomto provedení.

2.1 Postup při návrhu

Měřené veličiny a simulace

Jako první je třeba se zamyslet nad tím, co všechno bude dobré v daném provedení měřit a ovládat. V tomto konkrétním případě bylo vybráno měření CO_2 , vlhkosti, teploty a tlaku. Tato volba pokrývá potřeby v dnešní běžné domácnosti, jelikož se jedná o základní veličiny simulace moderního domácího prostředí. Například dříve hladina CO_2 nebyla pro běžného uživatele nijak důležitá, ale dnes je již tato veličina velmi aktuálním tématem. Pro ovládací prvky byla zvolena simulace osvětlení a zavírání, otevírání dveří nebo oken pro možné větrání.

	Koncentrace CO_2 ve vzduchu (ppm)	Doba vystavení plynu
<i>Negativní účinky při krátkodobém vystavení CO_2</i>		
Únava a snížená pozornost	> 1500	< 4hodiny
Ospalost, letargie, bolest hlavy	> 2500	< 4hodiny
Nevolnost a zvýšený tep	> 5000	≈ 1 – 6hodin
Dýchací potíže	> 15000	≈ 1 – 6hodin
Možná ztráta vědomí	> 40000	≈ 1 – 6hodin
<i>Negativní účinky při dlouhodobém vystavení CO_2</i>		
Chronický systematický zánět	≈ 3000	13dní
Demineralizace kostí a kalcifikace ledvin	≈ 2000 – 3000	60 – 90dní
Změny chování a psychický stres	700 – 3000	13 – 15dní

Tabulka 2.1: Tabulka rizikových hodnot CO_2 [15]

■ 2.2.2 Měření teploty

Měření teploty je pravděpodobně jedna z nejdéle měřených a nejzákladnějších veličin v domácím prostředí. Pro svoje pohodlí každý člověk potřebuje mírně odlišnou teplotu, nelze tedy přesně určit, jaká je ideální hodnota. Právě kvůli tomu je nutné tuto veličinu měřit a regulovat dle vlastní potřeby. Volbu teploty je nutné přizpůsobit ale také tomu, o jakou místnost se jedná. Například v koupelně by teplota mohla dosahovat až $25^{\circ}C$, ale v obývacím pokoji by neměla přesahovat $21^{\circ}C$.

■ 2.2.3 Měření vlhkosti

Další velmi důležitou veličinou pro naše zdraví ale i domácnost obecně je hladina vlhkosti. Příliš nízká hladina negativně ovlivňuje převážně zdraví člověka. Může způsobovat vysušování sliznic, očí nebo pokožky. Má ale negativní vliv i na dřevěný nábytek. Naopak vysoká hladina škodí převážně naší domácnosti. Způsobuje vznik plísní a roztočů, má negativní vliv také na potraviny a léky. Zároveň si v takovém prostředí jen těžko usušíte své oblečení. I zde je ale možné najít příklad vlivu na naše zdraví, jelikož výskyt plísní a roztočů může podporovat vznik alergií. Ideální hladina vlhkosti pro běžnou domácnost se uvádí v rozmezí 40% – 60%.

■ 2.2.4 Měření tlaku

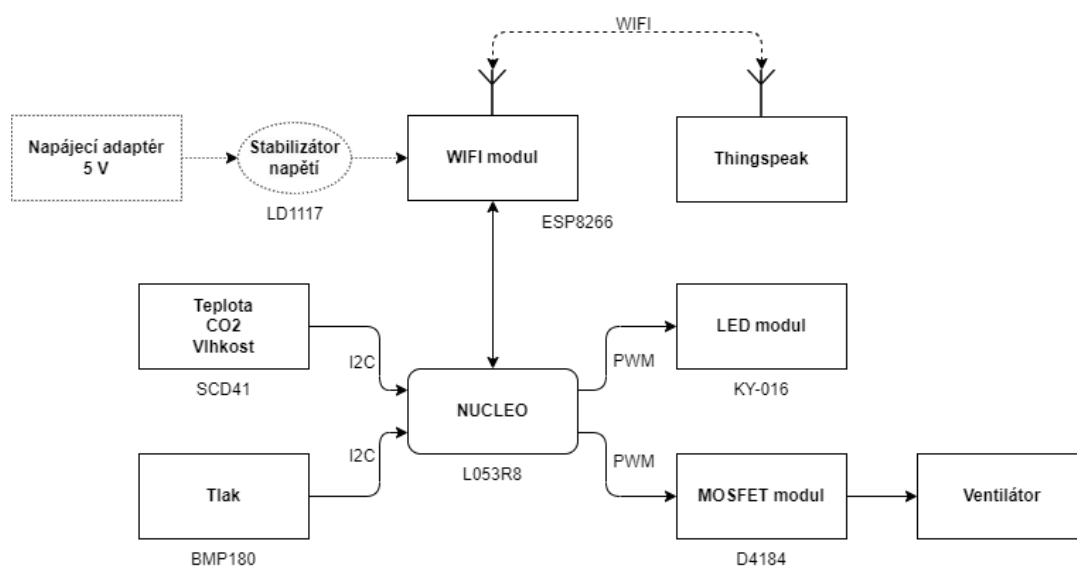
Vysoká hladina tlaku může také negativně ovlivnit lidské zdraví. Způsobuje celkově vyšší zatížení lidského organismu, což může způsobovat například závratě nebo motání hlavy. Nebezpečí hrozí hlavně kardiakům a lidem s dýchacími potížemi. Ačkoliv s touto hodnotou člověk doma jen těžko něco udělá, je dobré alespoň o možném nebezpečí a rizicích vědět. Běžná hladina tlaku v České republice se pohybuje okolo $1013hPa$.

Kapitola 3

Hardware

Kapitola pojednává o celkovém použitém hardwaru, o jeho vlastnostech a základních principech. Dále je zde uvedena základní teorie fungování daných senzorů a ovládacích prvků, pro jejich pochopení. Kapitola také obsahuje ukázkou veškerého použitého hardwaru. Jako poslední se zde nachází ukázkou navržené krabičky, ve které bude vše uchováno.

Z kapitoly 2 o návrhu systému vyplývají požadavky na měřicí systém. Konkrétní blokové schéma dle daných požadavků je zobrazeno na obrázku 3.1.



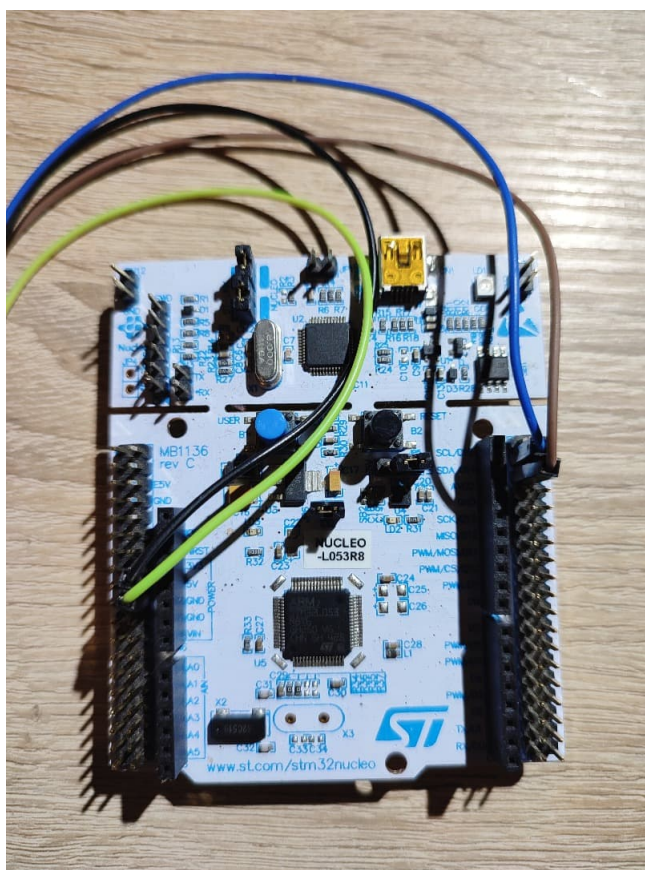
Obrázek 3.1: Hardwarové blokové schéma

3.1 Vývojová deska

Zvolená centrální jednotka celého měřicího systému:

3.1.1 NUCLEO-L053R8

Mikrokontrolér od společnosti ST. Nucleo bylo vybráno díky již získaným zkušenostem z předmětu laboratoří aplikované elektroniky a řízení, nebo programování v C. Jedná se o 64 pinovou verzi [12] s flash pamětí 64Kbytes. Tato paměť by měla být pro toto provedení dostačující. Deska má dále dva I_2C porty a šest PWM portů. Porty tedy dostatečně pokrývají potřeby návrhu. Nucleo poskytuje napájení na 3.3 V, 5 V nebo 6 – 12 V. To opět ideálně pokrývá potřeby vybraných senzorů a ovládacích prvků.

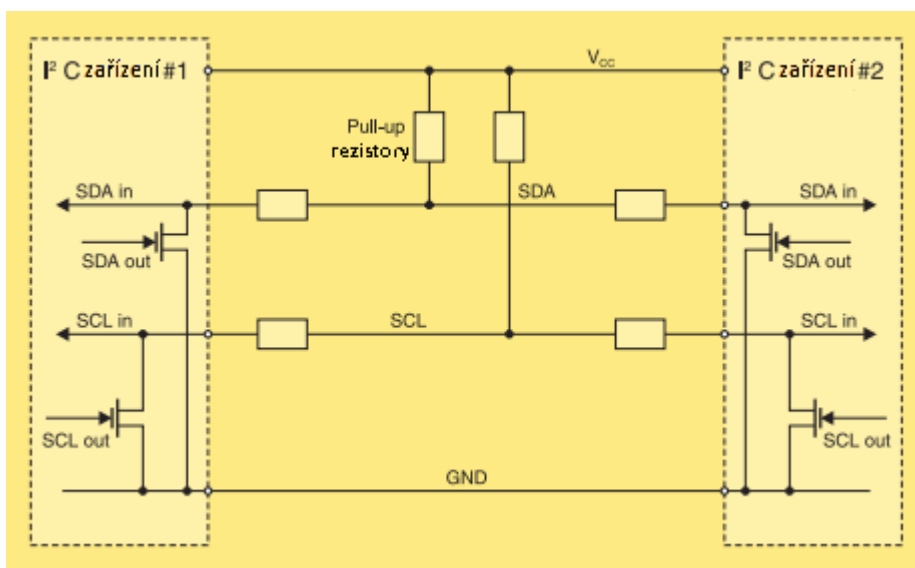


Obrázek 3.2: Nucleo deska L053R8

I_2C

Počítačová, dvou vodičová, sériová sběrnice pro přenos dat [14]. Přenos dat probíhá pomocí dvou kanálů, kanálu SCL, který se používá pro hodinový signál a kanálu SDA, který je použit pro přenos dat. Na začátku každé

komunikace je zaslán takzvaný start bit a na konci zase stop bit. Po start bitu následuje sedm bitů adresy senzoru a až následně jsou posílána samotná data. Výhodou sběrnice je komunikace pouze přes dva dráty a možnost mít vícero zařízení ve funkci master. Nevýhodou je například pouze poloduplexní režim nebo nižší rychlost v porovnání s ostatními sběrnicemi. Řízení přístupu funguje metodou master-slave. Jedná se o deterministickou metodu, tedy nemohou nastat kolize. Máme dva druhy uzlů - master a slave. Master řídí celou komunikaci, dotazuje se uzlů typu slave, které nesmí vysílat samovolně. Uzly slave jsou jen "hloupá" zařízení, nesmí komunikovat mezi sebou, jen s uzlem typu master. Výhodou tohoto přístupu je jednoduchost provedení. Nevýhodou je závislost na master uzlu. Kdyby se tento uzel poškodil, nemohla by komunikace proběhnout.



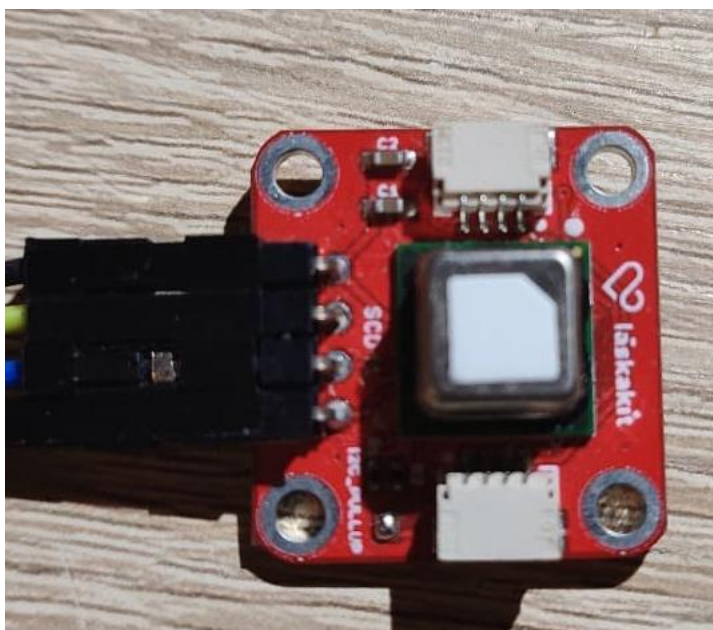
Obrázek 3.3: Schéma zapojení sběrnice I_2C [14]

3.2 Senzory

Měřicí systém obsahuje následující dva senzory:

3.2.1 SCD41

Senzor je použit pro měření vlhkosti, teploty a CO_2 [8]. Byl vybrán jako primární senzor, jelikož pokrývá většinu potřebných veličin. Má digitální výstup a komunikace probíhá po digitální sběrnici I_2C . Napájí se přes nucleo na 5 V.

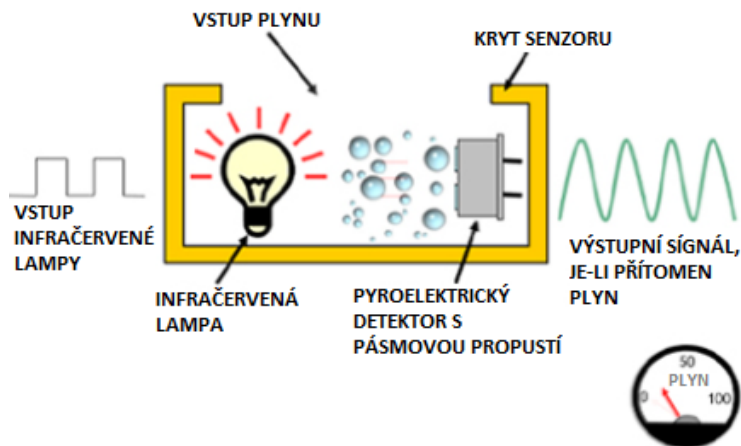


Obrázek 3.4: Senzor SCD41

Následující tři způsoby a metody se používají senzorem SCD41 pro měření CO_2 , teploty a vlhkosti v uvedeném pořadí.

■ NDIR měření

Detekce oxidu uhličitého probíhá pomocí absorbování infračerveného světla na dané vlnové délce [15]. Pokud je infračervené světlo absorbováno, je detekována přítomnost CO_2 . Dle úrovně absorpce je určeno, kolik je ve vzduchu plynu [16]. V NDIR provedení je potřeba mít zdroj infračerveného světla, pásmovou propust, světelnou trubici a přijímač. Nejčastěji používaná vlnová délka je okolo 4.28 μm kvůli tomu, že na ni příliš nereagují ostatní plyny, takže lze oxid uhličitý rozeznat. Výhodou NDIR senzorů je vysoká přesnost a nízká spotřeba energie. Na obrázku 3.5 můžeme vidět obrázek s principem detekce přítomnosti daného plynu.



Obrázek 3.5: Princip NDIR měření [16]

■ Akustická termometrie

Zjednodušeně by se metoda dala popsat jako měření doby průchodu zvuku daným prostředím. Z doby průchodu lze dopočítat potřebnou teplotu, jelikož se zvuk šíří rychleji v teplejším prostředí nežli v prostředí chladnějším [22]. Prostředí může být například tekutina, plyn nebo pevný předmět. Přes dané prostředí se pomocí sonického nebo zvukového vysílače posílají signály o přesné vlnové délce. Na druhé straně je přijímač, který zaznamenává čas přijetí zvuku. Jelikož známe vzdálenost mezi vysílačem a přijímačem, můžeme z času dopočítat rychlost šíření a následně tedy i teplotu. Výhodou tohoto měření je velmi rychlá odezva a také odolnost proti extrémním podmínkám jako je například vysoká teplota. Díky těmto výhodám se mohou tyto senzory používat například i v jaderných reaktorech [23].

■ Kapacitní hygrometry

Hygrometry [18] lze v základu dělit na kapacitní nebo odporové dle použitého materiálu. Podle datasheetu senzoru nešlo přesně dohledat, jakým z těchto dvou způsobů je vlhkost měřena. Bylo ale dle daných vlastností odhadnuto, že se bude jednat o hygrometr kapacitní [17]. Klíčový je zde materiál mezi elektrodami, který musí mít rychlou absorpci vody. Takový materiál je například polymer. Princip funkčnosti spočívá v tom, že se do daného materiálu dostane voda pomocí propustnosti jedné z elektrod. To mění vlastnosti obvodu a tedy i danou kapacitu. Ze změny její hodnoty se následně určuje úroveň vlhkosti.

■ 3.2.2 BMP180

Senzor je použit pro měření tlaku [9]. Je s ním možno měřit také teplotu, ta je v daném provedení měřena ale již senzorem SCD41, tedy tato funkce není využita. Senzor byl vybrán pro měření zbylých veličin, které neměří

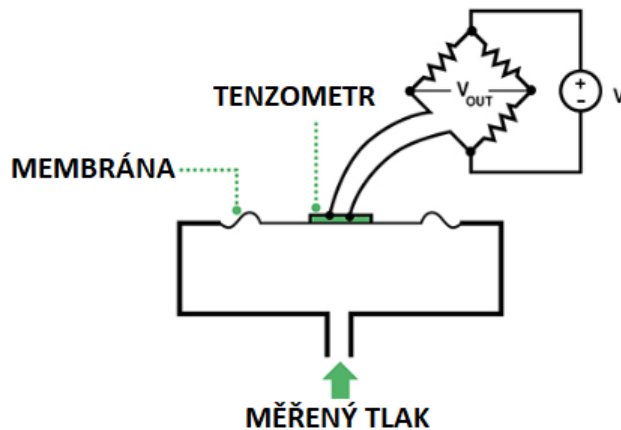
SCD41. Má opět digitální výstup a komunikuje po sběrnici I_2C . Napájí se opět samostatně skrz nucleo na 5 V.



Obrázek 3.6: Senzor BMP180

■ Piezo-odporové měření

Jedná se o jedny z nejpoužívanějších senzorů tlaku, díky jejich robustnosti a přesnosti. Princip staví na změně elektrického odporu při mechanickém namáhání daného materiálu [20]. Základem měření je použití tenzometru z vodivého materiálu, který je umístěn na membráně. Tato membrána přenáší právě změnu odporu. Ta je následně převedena na potřebný signál, ze kterého je pak dopočítán samotný tlak. Odpor vodiče je určen jeho délkou a průřezem. Právě tohoto se při měření využívá. Na obrázku 3.7 níže lze vidět princip měření tenzometru pomocí zapojení s Wheatstoneovým můstkem.



Obrázek 3.7: Piezo-odporové měření tlaku [20]

3.3 Ovládací prvky

Následují dva ovládací prvky simulující osvětlení a větrání v místnosti:

3.3.1 Keyes KY-016

RGB LED modul [11]. Skládá se ze tří samostatných diod, ovládaných přes PWM. Diody reprezentují každá právě jednu z trojice barev - zelené, modré a červené. Pomocí hodnot PWM lze korigovat intenzitu dané diody, každá z diod má vlastní PWM vstup. LED modul je v návrhu využit pro identifikaci hladiny CO_2 . Pokud je hladina příliš vysoká, tak modul svítí červeně, pokud je však dostatečně nízká, tak svítí zeleně.

PWM

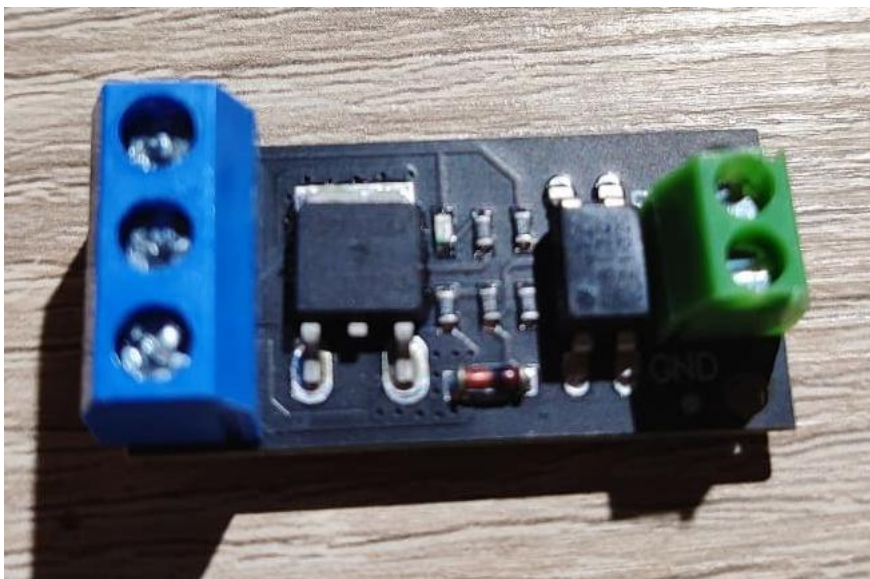
Neboli pulzně šířková modulace [7]. Jedná se o diskretní modulaci analogového signálu. Tvoří čtvercové vlny, které mají dvě logické úrovně log. 1 a log. 0. To může být simulováno například pomocí napětí 5 V a 0 V , nebo pomocí proudu. Přenášenou informaci lze vždy převést do intervalu $0\% - 100\%$. Hodnota informace je dána poměrem logických úrovní, neboli střídou. Hodnotu lze také vypočítat přes neurčitý integrál podělený danou periodou signálu. U PWM modulace musí být také dodržen vzorkovací teorém, který říká, že vzorkovací frekvence musí být větší než dvojnásobek maximální frekvence. Výhodou PWM je levné provedení, malé napěťové ztráty a jednoduchost použití. Nevýhody jsou komplexita metody a vznik napěťových špiček.



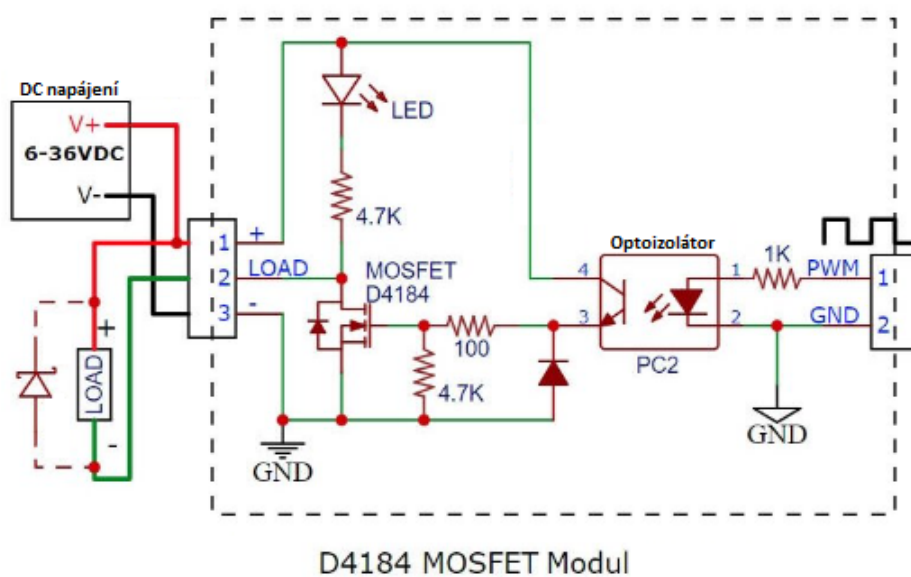
Obrázek 3.8: LED modul Keyes KY-016

■ 3.3.2 D4184

PWM MOSFET modul [10]. Nastavením hodnot PWM lze korigovat výkon na výstupu. V této konfiguraci simuluje zavírání a otevírání dveří nebo oken. Napájení opět přes PWM. Konkrétně na výstup modulu je připojen ventilátor, jehož rychlost otáčení lze regulovat právě pomocí MOSFETU. Ventilátor simuluje větrání v místnosti, které je zároveň potřeba pro redukci hladiny CO_2 , teploty nebo vlhkosti. Na obrázku 3.10 lze vidět elektronické schéma samotného modelu. Oproti původní verzi byl zkratován odpor na čtvrtém výstupu z optočlenu, který dělil vstupní napětí na půlku, tedy na přibližně 2.5 V. Půlené napětí způsobovalo nedostatečné spínání MOSFET tranzistoru a následně i nedostatečné napětí pro roztočení ventilátoru. Bylo tedy nutné tento odpor odebrat.



Obrázek 3.9: MOSFET modul D4184

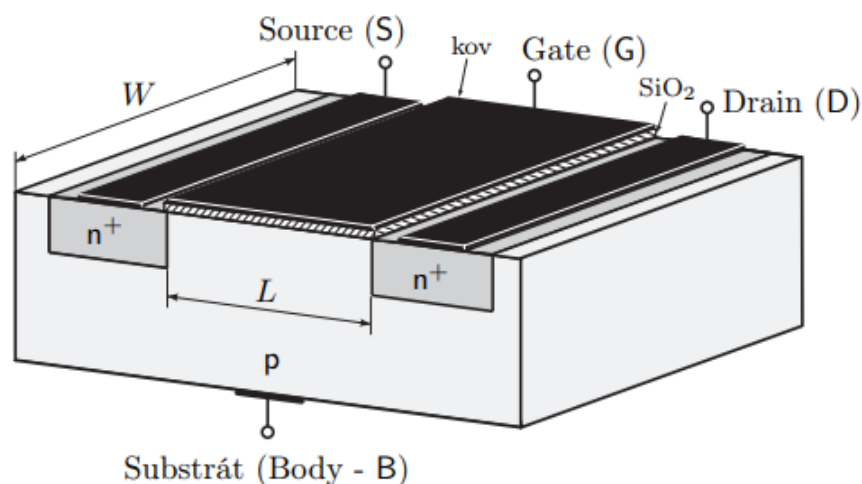


Obrázek 3.10: Elektronické schéma D4184 modulu

■ MOSFET

Elektrickým polem řízený tranzistor, jehož řídicí elektroda je hliníková a je izolována tenkou vrstvičkou oxidu křemičitého [21]. Rozlišujeme dva druhy tohoto tranzistoru, s kanálem typu N a s kanálem typu P. Je také nutné rozlišovat tři kovové elektrody nazvané gate, source a drain. Zde je použit

MOSFET s N kanálem, dále tedy bude popsána funkčnost právě tohoto typu tranzistoru. Aby se vytvořil vodivý kanál mezi drainem a sourcem, je třeba přivést dostatečně velké napětí, které přesáhne prahové napětí, na řídicí elektrodu gate. Následně pokud přivedeme mezi drain a source napětí, začne kanálem téct proud. Níže je struktura právě takového MOSFET tranzistoru i s příslušnými elektrodami.



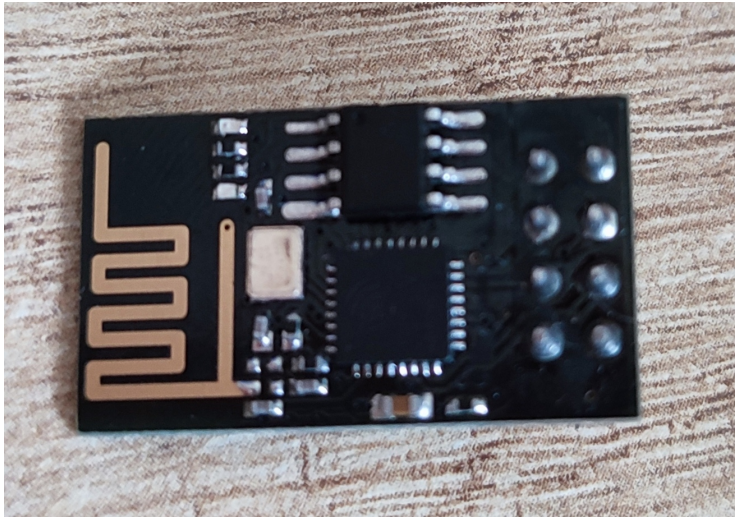
Obrázek 3.11: Struktura MOSFET tranzistoru [21]

3.4 Komunikace

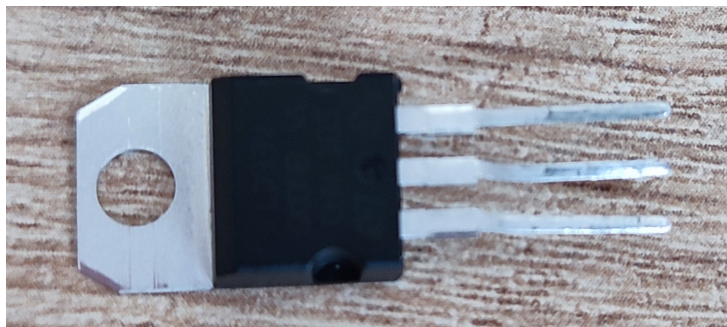
3.4.1 ESP8266

Jedná se o ESP-01 TCP/IP WIFI modul [13]. Poskytuje kompletní a soběstačné řešení WIFI sítě. Dokáže pracovat v nízko-napěťovém režimu pro co nejmenší spotřebu energie. Umí také rychle přepínat mezi aktivním a úsporným režimem pro další šetření spotřeby. Obsahuje 2.4 GHz přijímač a 2.4 GHz vysílač. Mezi další vlastnosti patří TCP/IP protokol, 802.11 b/g/n protokol a výstupní SPI, UART port.

Pro napájení WIFI modulu je použit usměrňovač napětí LD1117 [24]. Na vstup usměrňovače je přivedeno napětí 5V, které je převedeno na výstupní napětí 3.3 V. LD1117 musel být použit z důvodu nestabilního 3.3 V zdroje z nuclea, které není pro napájení vhodné. Místo toho bylo napájení na 5 V přivedeno přes síťový napájecí adaptér, aby modul fungoval stabilně a nedocházelo k výpadkům komunikace.



Obrázek 3.12: TCP/IP WIFI modul



Obrázek 3.13: Usměrňovač napětí LD1117

■ WIFI

V technické praxi označovaná spíše názvem 802.11. Jedná se o bezdrátovou technologii určenou pro přenos dat na střední vzdálenosti. Přenos dat zde probíhá přes rádiové vlny o frekvenci od 2.4 GHz do 5 GHz . WIFI lze brát jako nastavbu klasického ethernetu, do kterého přidává novou fyzickou a spojovou vrstvu. Protokol 802.11 lze dále rozdělit do několika podskupin podle rozdílného vývoje a vlastností [4]. Výhodou WIFI oproti bluetooth je například rychlejší přenos dat, delší vzdálenost přenosu a přijatelnější připojení vícero uživatelů najednou. Nevýhodou je větší spotřeba energie, což zde lze opět zanedbat, jelikož systém spotřebu bez problému zvládne. Díky převažujícím výhodám byl vybrán právě protokol 802.11. Další potřebné parametry a vlastnosti lze dohledat v příslušných citovaných datasheetech.

3.5 3D návrh modelu krabičky

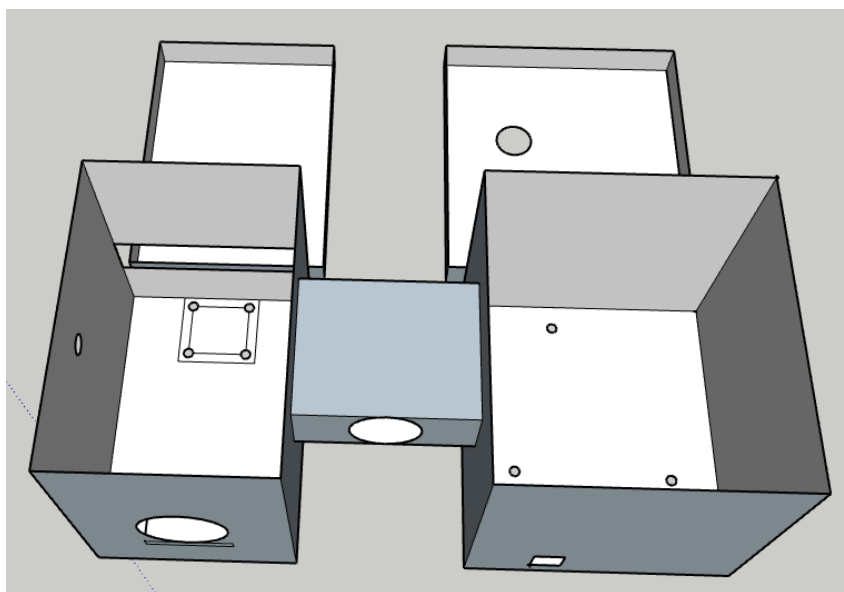
Pro uzavření a ochranu veškerého navrženého hardwaru byl navržen model dvou propojených krabic i s příslušnými víky.

3.5.1 Software

Pro celý návrh modelu krabičky bylo použito prostředí SketchUp. Jedná se o freeware internetový software, který je založen na principu přímého modelování. K jeho použití není tedy potřeba žádné znalosti CADu.

3.5.2 3D model

Níže na obrázku 3.14 je možné vidět tento kompletní 3D návrh. Ve větší ze dvou krabiček bude následně uzavřeno nucleo. Je zde tedy udělán boční otvor pro přívodní napájení a zároveň horní otvor ve víku pro přístup k RESET tlačítku. Jsou zde také udělány otvory pro izolační sloupky, přes které bude nucleo uchyceno. V druhé části je vymodelován výstupek, kde bude umístěn ventilátor tak, aby směřoval na všechny ostatní senzory. Za ventilátorem a naproti němu je pak v krabičce okénko, aby bylo možné simulovat větrání a následně i snižování teploty, vlhkosti a CO_2 . Na druhé straně jsou vymodelovány další otvory pro izolační sloupky, tentokrát pro uchycení senzoru SCD41. Na boku této části je okénko pro LED modul, který bude ke straně přilepen tavnou pistolí. Okénko je uděláno tak, aby dioda přesahovala ven a bylo možné sledovat její barvu. Stejným způsobem budou přilepeny ostatní prvky návrhu ke dnu, aby se zamezilo jejich případnému posuvu. Mezi boxy je uzavřená část, kde budou vedeny všechny kabely. V této části je opět udělán otvor, v tomto případě pro přívod napájecího adaptéru.



Obrázek 3.14: 3D model krabičky

Kapitola 4

Software

Tato kapitola pojednává o veškerém použitém softwaru a navrhovaném firmwaru v systému, včetně aplikace pro vizualizaci dat.

4.1 Použitý software

V této části jsou uvedeny všechny programové prostředky, které byly použity při programování a zprovoznění obou senzorů i ovládacích prvků. Zároveň jsou zde uvedeny programy použité při návrhu vizualizace dat.

4.1.1 Software pro programování prvků měřicího systému

STM32CubeIDE

Programovací prostředí od firmy STM. Tento software byl zvolen kvůli největšímu přístupu k vývojové desce Nucleo a zkušenostem práce s ním z laboratoří aplikované elektroniky. V tomto vývojovém prostředí byly následně naprogramovány a odladěny všechny programy daných prvků.

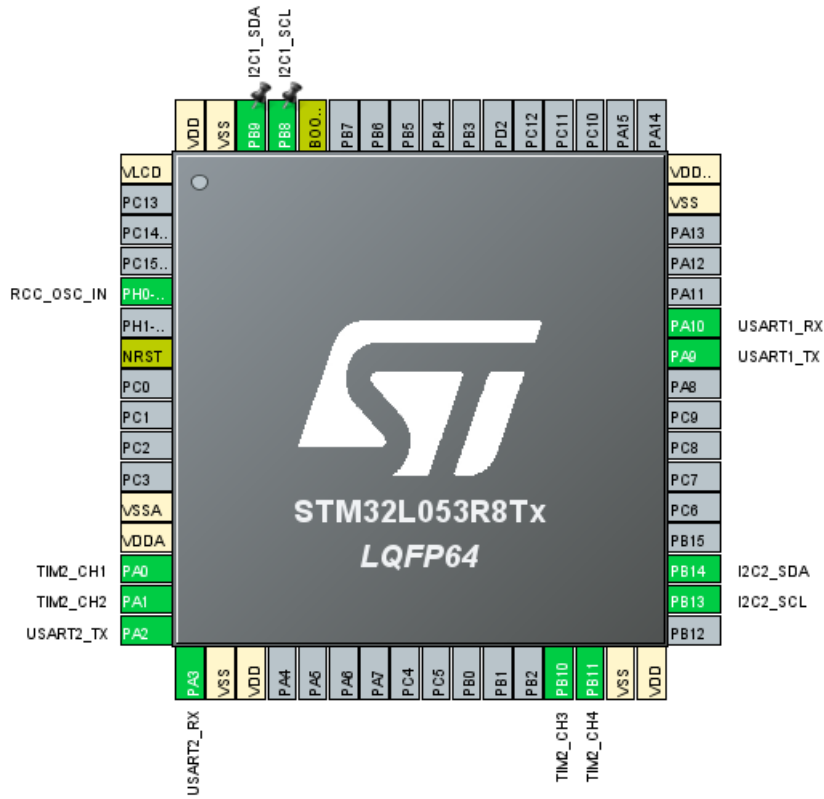
Hterm

Terminál určený pro sériovou komunikaci desky se senzory. Byl použit pro počáteční zprovoznění a odladění komunikace se senzory před použitím WIFI modulu. Přes druhý UART lze v terminálu sledovat průběh komunikace a případné chybové hlášky.

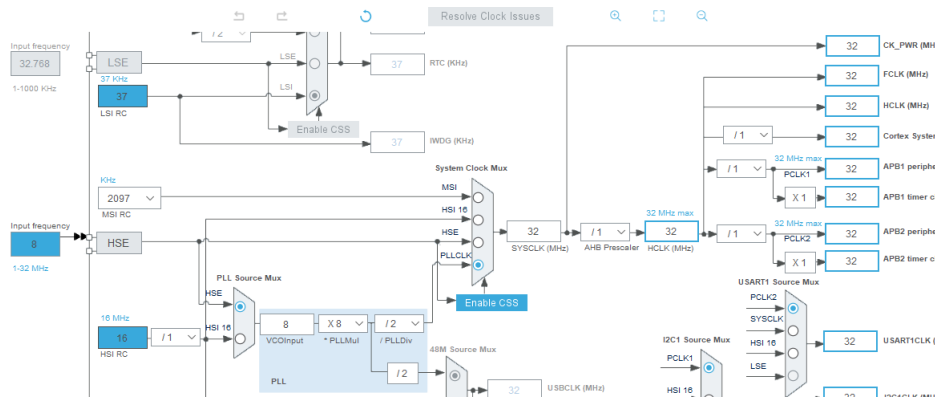
4.1.2 Software pro vizualizaci dat

Thingspeak

Jedná se o cloud platformu internetu věcí, určenou pro vizualizaci a analýzu dat. Platforma je přímo propojena s matlabem, takže lze data ihned přeposílat a využívat dle potřeby. Vizualizaci lze provádět pomocí grafů, kde se může zobrazovat téměř libovolný počet měření, takže se dají sledovat i v dlouhodobém časovém období. V grafu lze navíc provádět výpočet průměrné



Obrázek 4.1: Zapojení pinů na vývojové desce



Obrázek 4.2: Nastavení hodinového signálu

4.2.2 Hlavní program

Nejdříve je potřebné si ukázat hlavní program celého firmwaru. Zde se volají všechny funkce prvků, které budou popsány níže u jednotlivých periférií.

■ Inicializace periférií

Pro spuštění všech prvků je při zapnutí programu vloženo vteřinové zpoždění. Pro přehlednost sledování průběhu programu jsou zde poslány dvě zprávy o začátku a konci inicializace. Mezi jednotlivými zprávami jsou postupně spuštěny a nastaveny všechny periferie včetně připojení k internetu. K připojení na WIFI je potřebné zadat správné jméno a heslo internetu, na kterém je notebook připojený.

```

121 //Delay for starting all peripherals
122 HAL_Delay(1000);
123
124 //Initing all peripherals
125 HAL_UART_Transmit(&huart2, "InitStart ", 10, HAL_MAX_DELAY);
126 ESP_Init(SSID,PASS);
127 //BMP180_Start();
128 SCD41_Init_Start_Measurement();
129 LED_Init();
130 MOSFET_Init();
131 HAL_UART_Transmit(&huart2, "InitDone ", 9, HAL_MAX_DELAY);

```

Obrázek 4.3: Začátek programu

■ Vyčítání a posílání dat

Po kompletním spuštění je možné začít v nekonečné smyčce vyčítat hodnoty z obou senzorů. Dále se naměřená hodnota CO_2 využívá k nastavení svitu diod a otáčení ventilátoru. Jako poslední se hodnoty posílají pro vizualizaci do platformy Thingspeak. Následuje zpoždění pěti vteřin, jelikož senzor SCD41 je v periodickém režimu schopen měřit nová data právě po pěti vteřinách.

```

143 //Reading measurement
144 SCD41_Read_Measurement(&co2, &temperature, &humidity);
145 pressure=BMP180_GetPress(0);
146
147 //Setting LED and ventilator
148 LED_CO2(co2);
149 MOSFET_CO2(co2);
150
151 //Sending data to Thingspeak
152 Buffer[0]=temperature;
153 Buffer[1]=co2;
154 Buffer[2]=humidity;
155 Buffer[3]=pressure;
156 ESP_Send_Multi(API_KEY, 4, Buffer);
157 HAL_UART_Transmit(&huart2, "DataSent ", 9, HAL_MAX_DELAY);
158 HAL_Delay(5000);

```

Obrázek 4.4: Nekonečná smyčka programu

4.2.3 Senzor SCD41

Hlavním senzorem návrhu je SCD41, tato část je tedy zaměřena na ukázkou implementace firmwaru právě tohoto senzoru.

Adresy a příkazy

Jako první je nutné si definovat všechny potřebné adresy a příkazy, kterými je následně senzor ovládán. To lze vidět v ukázce na obrázku 4.5. V programu jsou použity příkazy pro zahájení a ukončení periodického měření a faktory reset při počátečním nastavení senzoru. Následuje příkaz pro čtení, který slouží k vyčítání všech surových dat potřebných pro výpočet veličin. Testovací příkaz lze použít pro otestování funkčnosti senzoru, není však v základním programu využit.

```

6 //Definition of all necessary address and commands
7 #define SCD41_ADDRESS (0x62 << 1)
8 #define SCD41_CMD_START_MEASUREMENT 0x21B1
9 #define SCD41_CMD_READ_MEASUREMENT 0xEC05
10 #define SCD41_STOPPERIODICMEASUREMENT 0x3F86
11 #define SCD41_SELFTEST 0x3639
12 #define SCD41_CMD_SOFT_RESET 0x3632

```

Obrázek 4.5: Definování adres a příkazů

Inicializace senzoru

Dále je možné vidět ukázkou inicializace senzoru na obrázku 4.6. Před vysláním daných příkazů je nejprve potřebné, převést definované povely do správného endianu a seřadit podle nejvýznamnějšího bitu. Následně lze data poslat po správné I_2C na adresu senzoru. V programu je vidět i ošetření chybného přenosu dat, které se řeší posláním chybové hlášky pomocí sériové komunikace do terminálu přes určený UART, který je propojen s počítačem. V inicializaci musí být vždy první zastavení měření, jelikož jinak při resetování desky přes RESET tlačítko dojde k chybnému odeslání následujících dat. Faktory reset je používán pro vymazání všech předešlých měření. Musí být zároveň dodržena i všechna zpoždění, jelikož senzor potřebuje čas pro zahájení periodického měření. Pokud by bylo zpoždění kratší, tak by docházelo k chybě při prvním vyčítání dat.

```

20 //Stop periodic measurement
21 //Shifting and sorting by most significant bit
22 uint8_t tx_data[2] ={SCD41_STOPPERIODICMEASUREMENT >> 8,
23   SCD41_STOPPERIODICMEASUREMENT & 0xFF};
24 //Transmitting modified command to direct address
25 status = HAL_I2C_Master_Transmit(&hi2c1, SCD41_ADDRESS,
26   tx_data, 2, HAL_MAX_DELAY);
27 //Error handling
28 if (status != HAL_OK) {
29   sprintf(text,"ErrorS0: %d\n", status);
30   HAL_UART_Transmit(&huart2, text, 10, HAL_MAX_DELAY);
31 }
32 HAL_Delay(1000);
33
34 //Factory reset of sensor
35 uint8_t tx_data1[2] ={SCD41_CMD_SOFT_RESET >> 8,
36   SCD41_CMD_SOFT_RESET & 0xFF};
37 status = HAL_I2C_Master_Transmit(&hi2c1, SCD41_ADDRESS,
38   tx_data1, 2, HAL_MAX_DELAY);
39 //Error handling
40 if (status != HAL_OK) {
41   sprintf(text,"ErrorS1: %d\n", status);
42   HAL_UART_Transmit(&huart2, text, 10, HAL_MAX_DELAY);
43 }
44 HAL_Delay(1000);
45
46 //Start periodic measurement
47 uint8_t tx_data2[2] ={SCD41_CMD_START_MEASUREMENT >> 8,
48   SCD41_CMD_START_MEASUREMENT & 0xFF};
49 status = HAL_I2C_Master_Transmit(&hi2c1, SCD41_ADDRESS,
50   tx_data2, 2, HAL_MAX_DELAY);
51 //Error handling
52 if (status != HAL_OK) {
53   sprintf(text,"ErrorS2: %d\n", status);
54   HAL_UART_Transmit(&huart2, text, 10, HAL_MAX_DELAY);
55 }
56 HAL_Delay(5000);

```

Obrázek 4.6: Počáteční nastavení senzoru

■ Vyčítání veličin

V ukázce kódu na obrázku 4.7 lze vidět již konkrétní počítání veličin z příchozích dat. Pro přečtení dat je nejdříve třeba vyslat příkaz pro čtení, stejným způsobem jako lze vidět výše. Následně je možné přijmout potřebná data a načíst je do daného pole délky devět. Vždy dva bajty slouží pro vyčtení dané veličiny jako na obrázku a třetí bajt slouží jako CRC pro kontrolu správnosti dat. CRC v tomto provedení není počítáno, jelikož se nepředpokládá větší rušení dat. Nejdříve je vždy třeba první bajt posunout pro převedení endianu a následně je spojen s druhým bajtem. Tímto získáme surová data, která u CO_2 stačí již pro přečtení dané hodnoty v ppm. U zbylých

dvou veličin je třeba surová data přepočítat dle ukázky, aby se dostala teplota ve stupních Celsia a vlhkost v procentech.

```

77 // Parse CO2 value
78 *co2_ppm = (float)((((uint16_t)buf[0] << 8) | buf[1]));
79
80 // Parse temperature value
81 *temperature = (float)(-45.0f+175.0f*
82     (((uint16_t)buf[3] << 8) | buf[4]) / 65535.0f);
83
84 // Parse humidity value
85 *humidity = (float)(100.0f*
86     (((uint16_t)buf[6] << 8) | buf[7])/ 65535.0f);

```

Obrázek 4.7: Počítání jednotlivých veličin

4.2.4 Senzor BMP180

Druhým senzorem je BMP180, využitý pro měření tlaku. Většina tohoto firmwaru je ale poměrně složitá, jelikož využívá k výpočtu větší množství kalibračních konstant. Jsou zde tedy naznačeny pouze krátké ukázky pro pochopení funkčnosti. Navržený firmware pro tento senzor je inspirován článkem [26].

Kalibrace senzoru

Při spuštění senzoru se nejdříve vypočítají kalibrační konstanty. Stejně jako u předešlého senzoru se na definovanou adresu 0xEE pošle příkaz pro kalibraci 0xAA. Následně je přijato 22 bajtů kalibračních dat, ze kterých je vypočteno jedenáct kalibračních konstant. Zde je výpočet opět stejný jako u předešlého senzoru, první byt je posunut doleva a následně je sečten s bytem druhým. Jediný rozdíl je, že zde není posílán třetí byt s CRC.

```

45 //Reading calibration parameters with command 0xAA
46 uint8_t Callib_Data[22] = {0};
47 uint16_t Callib_Start = 0xAA;
48 HAL_I2C_Mem_Read(BMP180_I2C, BMP180_ADDRESS, Callib_Start,
49     1, Callib_Data,22, HAL_MAX_DELAY);
50
51 //Shifting first byte of data and add to the second one
52 AC1 = ((Callib_Data[0] << 8) | Callib_Data[1]);

```

Obrázek 4.8: Kalibrace senzoru BMP180

Vyčítání tlaku

Na obrázku 4.9 je ukázka vyčítání surových dat tlaku. Následně je pomocí těchto dat a kalibračních konstant dopočítán tlak klasický v Pascalech.

Nejdříve je příkazem 0xF4 požádáno o data tlaku a následně jsou přečteny tři bajty příkazem 0xF6. Mezi požadavkem a přečtením je drobné zpoždění, aby senzor stihl tlak naměřit. Nakonec jsou bajty opět posunuty do správného formátu a jsou sečteny.

```

90 //oss is oversampling setting
91 uint8_t datatowrite = 0x34+(oss<<6);
92 uint8_t Press_RAW[3] = {0};
93 //Transmitting and reading 3 bytes of
94 //raw pressure data with command 0xF4 and 0xF6
95 HAL_I2C_Mem_Write(BMP180_I2C, BMP180_ADDRESS,
96                   0xF4, 1, &datatowrite, 1, 1000);
97 //Delay for getting pressure data
98 HAL_Delay (26);
99 HAL_I2C_Mem_Read(BMP180_I2C, BMP180_ADDRESS,
100                 0xF6, 1, Press_RAW, 3, 1000);
101 //Shifting and adding together 3 bytes of data
102 return (((Press_RAW[0]<<16)+(Press_RAW[1]<<8)
103         +Press_RAW[2]) >> (8-oss));

```

Obrázek 4.9: Vyčítání surových dat pro výpočet tlaku

4.2.5 LED modul

Nyní následují dva PWM prvky. Prvním je LED modul, který se skládá ze tří samostatných diod. Každá je tedy ovládána separátním časovačem.

Inicializace modulu

Při inicializaci je nejprve nutné zapnout všechny tři PWM časovače a nastavit je tak, aby žádná dioda na počátku nesvítila. Po tomto kroku lze již libovolně nastavovat hodnoty PWM v daném intervalu periody.

```

9 //Starting PWM
10 HAL_TIM_PWM_Start(&htim2, TIM_CHANNEL_1);
11 HAL_TIM_PWM_Start(&htim2, TIM_CHANNEL_2);
12 HAL_TIM_PWM_Start(&htim2, TIM_CHANNEL_3);
13
14 //Setting PWM to 0
15 TIM2->CCR1 = 0;
16 TIM2->CCR2 = 0;
17 TIM2->CCR3 = 0;

```

Obrázek 4.10: Počáteční spuštění a nastavení PWM pro LED modul

Nastavení LED PWM

Vstupem této funkce je naměřená hodnota CO_2 . Podle tabulky 2.1 v kapitole o návrhu je následně rozhodnuto, jak budou dané diody svítit. Pokud je

hodnota pod 1000 *ppm*, tak svítí pouze dioda zelená, jelikož úroveň oxidu uhličitého je v doporučených mezích. Naopak pokud je hodnota nad 2000 *ppm*, tak svítí dioda červená, protože je hladina příliš vysoká a je třeba ji urychleně snížit. Pokud je úroveň mezi těmito hodnotami, tak pomalu přechází od zelené k červené úměrně k velikosti CO_2 .

```

24 //When co2 is low, green LED is on and red off
25 if (co2<=1000){
26     TIM2->CCR1 = 0;
27     TIM2->CCR2 = 1000;
28 }
29 //When co2 is too high, red LED is on and green off
30 else if(co2>=2000){
31     TIM2->CCR1 = 1000;
32     TIM2->CCR2 = 0;
33 }
34 //Otherwise red and green luminances are calculated
35 //according to co2 level
36 else{
37     start = co2-1000;
38     TIM2->CCR1 = 0+start;
39     TIM2->CCR2 = 1000-start;
40 }
41 }

```

Obrázek 4.11: Nastavení diod podle hodnoty CO_2

■ Další funkce LED

Tyto funkce nejsou použity v hlavním firmwaru, jsou ale součástí knihovny LED.h.

Funkce LED_ON_OFF má 3 vstupy, každý pro jednu LED. Vstupy reprezentují logické úrovně. Pokud je hodnota 0, tak se daná dioda vypne, pokud 1, tak zase naopak zapne na plný výkon.

Funkce LED_SET má také tři vstupy pro každou diodu, tentokrát ale udanou v %, tedy 1-100. Funkce nastaví hodnoty PWM jednotlivým LED. Je možné tedy regulovat svítivost daných diod.

■ 4.2.6 MOSFET modul

Všechny funkce MOSFET modulu jsou velmi podobné těm předchozím u LED, jelikož se jedná o stejný princip nastavení hodnot časovače.

■ Inicializace modulu

Opět stačí pouze zapnout daný kanál časovače a nastavit počáteční hodnotu na nulu.

```

9      //Starting PWM
10     HAL_TIM_PWM_Start(&htim2, TIM_CHANNEL_4);
11     //Setting PWM to 0
12     TIM2->CCR4 = 0;

```

Obrázek 4.12: Počáteční spuštění a nastavení PWM pro MOSFET modul

■ Nastavení MOSFET PWM

Stejně jako u LED, je vstupem hodnota CO_2 , která nastavuje rychlost otáčení ventilátoru. První změna přichází v mezích hodnot oxidu uhličitého. V tomto případě je ventilátor vypnut, pokud jsou hodnoty nižší než 1250 *ppm*. Horní hranice je 1750 *ppm*, nad kterou naopak ventilátor funguje na plný výkon. Mezi těmito hranicemi se opět otáčení ventilátoru reguluje podle úrovně oxidu uhličitého. Druhou důležitou změnou je, že regulace otáčení začíná až na hodnotě poloviny PWM. Tato hodnota je zde kvůli tomu, aby se ventilátor ihned roztočil. Kdyby byla regulace od nuly, tak by výkon nestačil pro jeho počáteční roztočení.

```

19     //When cO2 is low, ventilator is off
20     if (co2<=1250){
21         TIM2->CCR4 = 0;
22     }
23     //When cO2 is too high, ventilator is fully on
24     else if(co2>=1750){
25         TIM2->CCR4 = 1000;
26     }
27     //Otherwise ventilator revolutions are according to co2 level
28     else{
29         start = co2-1250;
30         TIM2->CCR4 = 500+start;
31     }

```

Obrázek 4.13: Nastavení větráku podle hodnoty CO_2

■ Další funkce MOSFET

Tyto funkce nejsou použity v hlavním firmwaru, jsou ale součástí knihovny MOSFET.h.

Funkce MOSFET_ON_OFF má jeden vstup s logickou úrovní 1 nebo 0, stejně jako u LED. Pokud je vstup 0, tak se ventilátor vypne. Když je vstup 1, tak se naopak zapne na plný výkon.

Funkce MOSFET_SET má také pouze jeden vstup, stejně jako u LED v %. Tato funkce nastaví PWM podle vstupní hodnoty. Lze s ní tedy libovolně nastavovat rychlost otáčení ventilátoru. Je ale nutné upozornit, že pro hodnoty nižší než 50 % se ventilátor pravděpodobně neroztočí.

4.2.7 WIFI modul

Posledním dílčím firmwarem je ten pro komunikaci přes WIFI modul. Ten je řízený přes AT příkazy, které mu jsou posílány jako řetězce. Úplný seznam všech příkazů i s vysvětlením lze nalézt na [27]. Navržený firmware pro WIFI modul je inspirován článkem [25].

Inicializace modulu

Jak bylo vysvětleno u hlavního programu, vstupem této funkce je jméno a heslo WIFI sítě, na kterou se uživatel chce připojit. První je modulu poslán příkaz na ověření komunikace spolu s pokynem pro jeho restart. Následuje opět ověření správnosti komunikace. Pak je nastaven režim, kde se modul bude připojovat na již existující síť a nebude tedy fungovat jako zdroj sítě. V předposledním kroku se už ESP8266 připojí na zadanou WIFI. Posledním krokem je nastavení, že se jedná o připojení skrz jedno zařízení. Tímto je modul připojen k internetu a je možné data přeposílat na zadané adresy.

```

22     //Checking connection to module and resetting it
23     Uart_sendstring("AT+RST\r\n");
24     HAL_Delay(1000);
25
26     Uart_flush();
27
28     //Checking connection
29     Uart_sendstring("AT\r\n");
30     while(!(Wait_for("OK\r\n")));
31
32     Uart_flush();
33
34
35     //Setting module to station mode
36     Uart_sendstring("AT+CWMODE=1\r\n");
37     while (!(Wait_for("OK\r\n")));
38
39     Uart_flush();
40
41
42     //Connecting to WiFi with SSID AND PASSWORD
43     sprintf (data, "AT+CWJAP=\"%s\", \"%s\"\r\n", SSID, PASSWD);
44     Uart_sendstring(data);
45     while (!(Wait_for("GOT IP\r\n")));
46
47     Uart_flush();
48
49     //Setting module to single connection mode
50     Uart_sendstring("AT+CIPMUX=0\r\n");
51     while (!(Wait_for("OK\r\n")));
--

```

Obrázek 4.14: Připojení k WIFI síti

■ Posílání dat

Funkce má tři vstupy, prvním je konkrétní klíč pro připojení k Thingspeak projektu, druhým je počet posílaných hodnot a posledním pole právě těchto hodnot. Nejprve proběhne připojení na adresu Thingspeaku a následuje posílání klíče pro identifikaci přesného projektu, kde se budou data zobrazovat. Dále už je poslána délka pole a samotné pole dat. Po tomto kroku Thingspeak již obdrží poslaná data s hodnotami naměřených veličin, se kterými je možné dále v tomto prostředí manipulovat.

```

94     //Connecting to Thingspeak
95     Uart_sendstring("AT+CIPSTART=\"TCP\", \"184.106.153.149\", 80\r\n");
96     while (!(Wait_for("OK\r\n")));
97
98     //Sending individual APIkey for connection to Thingspeak project
99     sprintf (data, "GET /update?api_key=%s", APIkey);
100    for (int i=0; i<numberoffileds; i++)
101    {
102        sprintf(data2, "&field%d=%.2f", i+1, value[i]);
103        strcat (data, data2);
104    }
105
106    strcat(data, "\r\n");
107    int len = strlen (data);
108
109    //Sending length of data array
110    sprintf (data1, "AT+CIPSEND=%d\r\n", len);
111    Uart_sendstring(data1);
112    while (!(Wait_for(">")));
113
114    //Sending data array with multiply values
115    Uart_sendstring (data);
116    while (!(Wait_for("SEND OK\r\n")));
117
118    while (!(Wait_for("CLOSED")));
119    ...

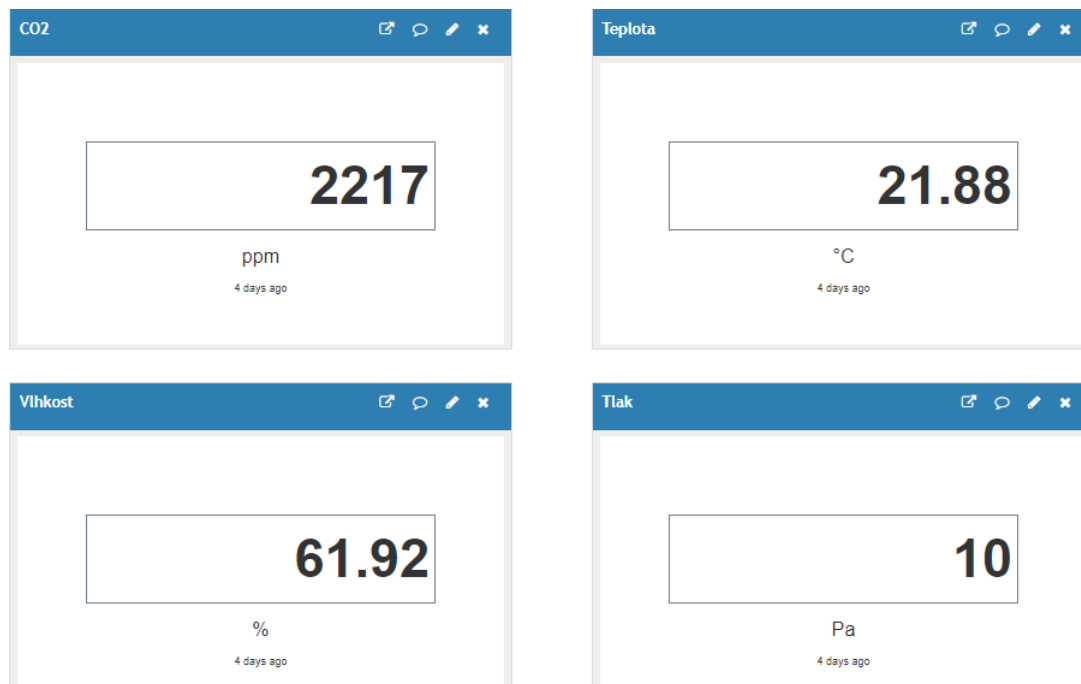
```

Obrázek 4.15: Posílání dat na platformu Thingspeak

4.3 Vizualizace dat

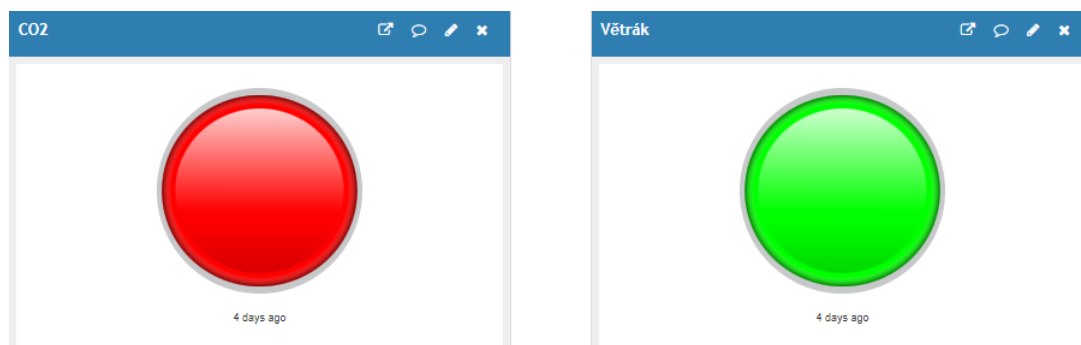
Vizualizaci naměřených dat přes Thingspeak lze rozdělit do tří částí.

V té první je možné najít aktuální hodnotu CO_2 v částicích na milion, teplotu ve stupních Celsia, vlhkost v procentech a tlak v Paskalech, v tomto pořadí. Tuto část lze vidět níže na obrázku 4.16.



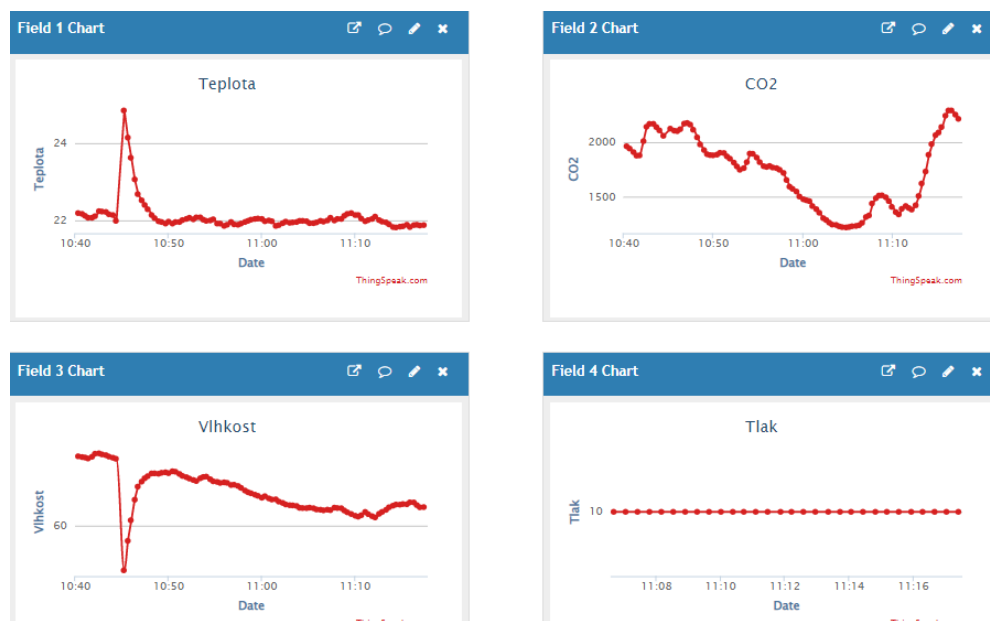
Obrázek 4.16: Vizualizace: aktuální hodnoty

Ve druhé části se nachází signalizace světelnými kontrolkami. První z nich signalizuje příliš vysokou hladinu oxidu uhličitého ve vzduchu. Druhá zase svítí, pokud je zapnutý ventilátor, tedy pokud je CO_2 vyšší než 1250 *ppm*. Obě signalizace je vidět na obrázku 4.17.



Obrázek 4.17: Vizualizace: signalizace

Poslední částí celé vizualizace jsou grafy jednotlivých veličin ve stejném pořadí jako v první části. U bodů v grafu lze vidět přesnou hodnotu a čas, kdy daná veličina byla přijata. Je tedy možné následně analyzovat časový vývoj těchto dat.



Obrázek 4.18: Vizualizace: grafy

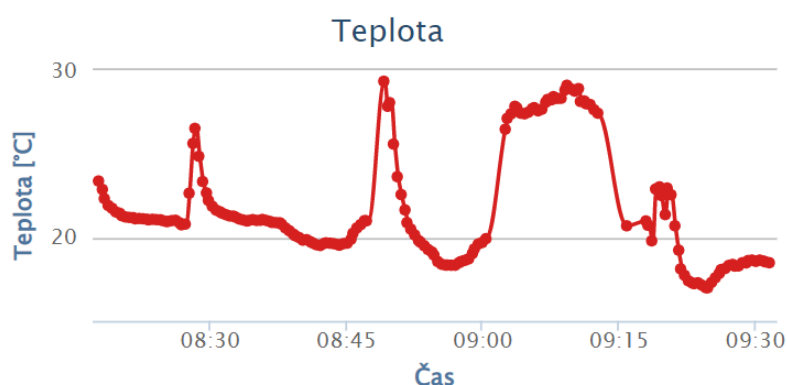
Kapitola 5

Simulace systému

Pro odzkoušení správnosti celého návrhu a naprogramovaného firmwaru byla provedena simulace, kde byl po dobu jedné hodiny celý systém vystaven odlišným podmínkám. Veškerá měření byla následně monitorována a poslána do Thingspeaku, kde byla provedena vizualizace v podobě grafů. Jednotlivé grafy veličin i s vysvětlením průběhu jsou postupně uvedeny níže. Nejdříve byl systém po dobu dvaceti minut vystaven domácímu prostředí bez jakéhokoliv pasivního větrání. Následně bylo ve vzdálenosti jednoho metru od systému otevřeno okno a měření pokračovalo po dobu dalších dvaceti minut s pasivním větráním. V poslední části byl systém přesunut do otevřeného okna, kde proběhl zbytek měření. V půlce každého časového intervalu byl senzor zadýchán tak, aby došlo k výraznému navýšení hodnot a následnému snížení pomocí vyvětrání. Během celého měření bylo zapnuto větrání aktivní pomocí zapojeného ventilátoru. Ten se zapnul vždy podle podmínek popsanych v předešlé kapitole o softwaru. V poslední části, při venkovním měření, byl senzor položen tak, že na něj dopadalo sluneční světlo a po zadýchání byl přemístěn do stínu. Celkově bylo naměřeno a zaznamenáno do grafu 180 hodnot.

5.1 Teplota

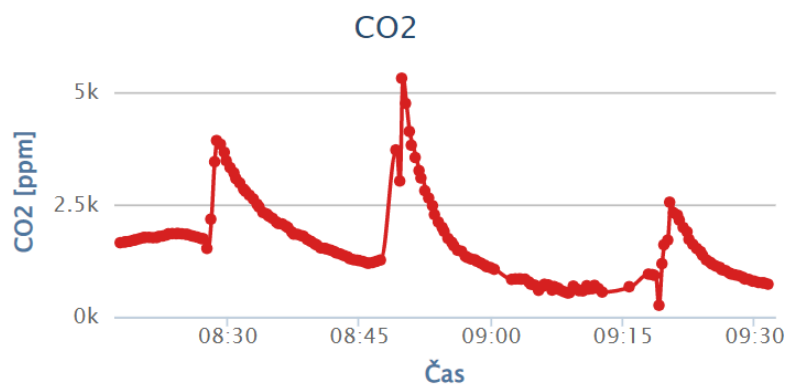
U teploty lze pozorovat, že po zapnutí senzoru nějakou dobu trvá, než se hodnota ustálí a je měřena správně. V časech 8:30, 8:50 a 9:20 je možné vidět výrazné navýšení teploty kvůli zadýchání senzoru a následné snížení vlivem vyvětrání okolního prostředí. Na konci vnitřního měření můžeme již pozorovat výrazně nižší teplotu než na začátku. Vnitřní teplota metr od okna klesla za čtyřicet minut o přibližně $2.5\text{ }^{\circ}\text{C}$. Po přesunu systému ven je vidět výrazné navýšení teploty bez zadýchání. Tento jev je způsoben tím, že na senzor svítilo v první polovině Slunce. Naopak v druhé půlce teplota klesla níže, než při měření uvnitř. Senzor byl ve stínu přesunut také blíže k vnitřnímu prostředí, takže teplota byla i tak o dost vyšší než teplota venkovní, která v době měření byla okolo $15\text{ }^{\circ}\text{C}$.



Obrázek 5.1: Graf teploty

5.2 CO_2

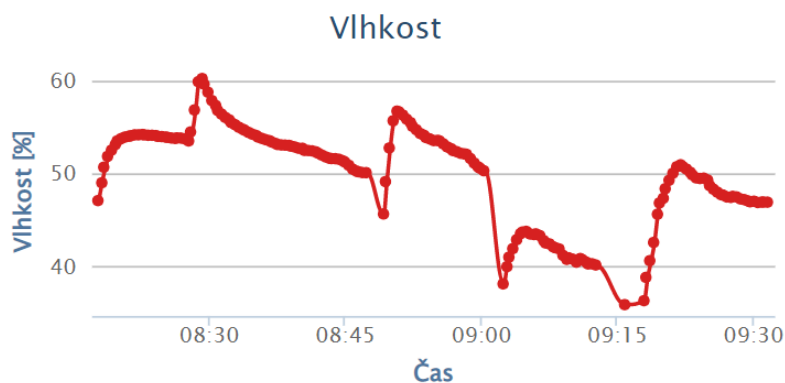
Měření oxidu uhličitého probíhá bez většího počátečního ustalování na správnou hodnotu. Prudké navýšení hodnot je vidět ve stejných časech jako bylo zmíněno výše. Na začátku, v nevyvětrané místnosti, byla hodnota CO_2 okolo 1750 *ppm*, tedy byla příliš vysoká. Na druhou stranu, po dvacetiminutovém vyvětrávání klesla tato hodnota na 1100 *ppm*, což už je vhodná hladina pro pobyt v místnosti. Po přesunu na okno se hodnota přiblížila k venkovním hodnotám na 700 *ppm*. Zde je nutné dodat, že měření bylo prováděno v Praze, kde je průměrná venkovní úroveň CO_2 okolo 600 – 650 *ppm*.

Obrázek 5.2: Graf CO_2

5.3 Vlhkost

Pro vlhkost platí stejné časové intervaly jako u předešlých dvou veličin, také je na grafu vidět tři prudká navýšení způsobená zadýcháním a následné vyvětrání. Při spuštění měření byla vlhkost v místnosti asi 55 %. Následně vlivem vyvětrání klesla úroveň o přibližně 4 %. Všechny tyto naměřené hodnoty ještě spadají do intervalu ideální humidity v domácnosti. V době

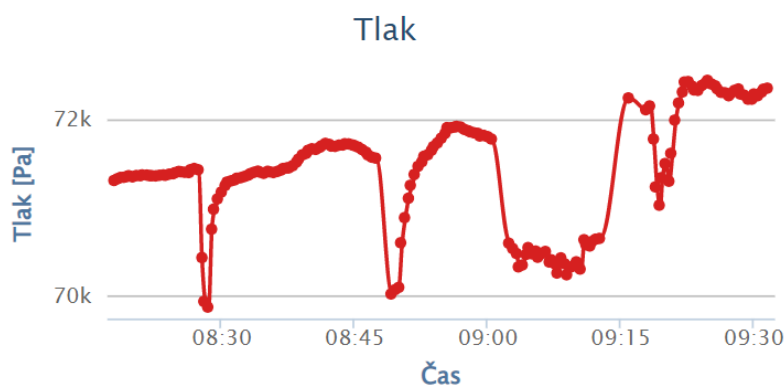
měření byla relativní vlhkost vzduchu v Praze okolo 60 %. Na počátku trvalo ustálení na správnou hodnotu kolem 4 *min.*



Obrázek 5.3: Graf vlhkosti

5.4 Tlak

Ačkoliv na grafu vypadá, že úroveň tlaku se v průběhu měření dosti mění, tak to tak ve skutečnosti není, jelikož se hodnoty pohybují v rozmezí 2000 *Pa*. To je k celkové hodnotě přes 70000 *Pa* zanedbatelná odchylka. Hladina tlaku se primárně mění se změnou nadmořské výšky, nicméně lze pozorovat mírné snížení tlaku zhruba o již zmíněných 2000 *Pa*, pokud je senzor zadýchán, nebo pokud se v blízkosti senzoru otevře okno. Pro větší vychýlení tlaku by bylo potřeba senzor přesunout do rozdílné nadmořské výšky, což v tomto měření nebylo z praktických důvodů realizováno. Z naměřených hodnot je patrné, že i přes kalibraci senzoru je zde nutné přidat offset zhruba 30000 *Pa*, jelikož referenční hodnota tlaku vzduchu pro Prahu je okolo 1000 *hPa*.



Obrázek 5.4: Graf tlaku

Kapitola 6

Závěr

Cílem této bakalářské práce bylo navrhnout kompletní systém pro simulaci inteligentní domácnosti spolu s aplikací pro vizualizaci a kontrolu všech dat. Práce by měla v budoucnu sloužit jako ukázka komunikace a vyčítání ze senzorů na dnech otevřených dveří a podobných akcích fakulty.

V části zaměřené na návrh systému bylo rozhodnuto, že systém bude měřit čtyři základní veličiny domácnosti a to teplotu, vlhkost, CO_2 a tlak. Pro simulaci bylo zvoleno osvětlení spolu s řízeným větráním pomocí ventilátoru. Pro komunikaci s aplikací byla zvolena WIFI síť.

V další části byly vybrány dva senzory SCD41 a BMP180, které pokrývají měření všech čtyř veličin. Dalším prvkem je MOSFET modul, u kterého bylo v průběhu práce rozhodnuto, že bude regulovat rychlost otáček ventilátoru podle hodnoty CO_2 . Druhým ovládacím prvkem hardwaru je LED modul, u kterého byla jeho funkcionalita taktéž v průběhu práce pozměněna a ve výsledku slouží jako signalizace hladiny oxidu uhličitého. Další je TCP/IP WIFI modul, který umožňuje komunikaci přes internet. Centrální jednotkou bylo zvoleno nucleo L053R8, pro jeho dostatečné pokrytí všech požadavků. V této kapitole je také ukázán 3D návrh krabičky, kde bude celý systém uchován.

Následně práce pojednává o veškerém použitém softwaru, spolu s ukázkami dílčích programů jednotlivých prvků návrhu, doplněné o jejich vysvětlení. K vizualizaci naměřených dat bylo využito platformy Thingspeak, kam jsou tato data posílána přes WIFI síť.

V závěru práce je ověřena funkcionalita celého systému v odlišných podmínkách. Zde jsou všechny měřené veličiny monitorovány po dobu jedné hodiny s postupně odlišnými způsoby větrání prostředí. Měřením byla ověřena správná funkčnost všech senzorů s dostatečnou citlivostí a zároveň regulace ovládacích prvků dle hladiny CO_2 .

Tato práce bude sloužit v případě úspěšného obhájení jako základ pro budoucí pokračování v rámci práce diplomové. Bylo by vhodné přidat například regulaci teploty pomocí možnosti přeposílání dat do Matlabu přes Thingspeak. Další bodem budoucího rozvoje by měla být vlastní aplikace s možností kompletního ovládní všech prvků pomocí již navržených funkcí, které v

tomto provedení ještě nebyly použity. Systém by bylo možné také doplnit o další periférie, například o modul s mikrofonom, který by umožňoval ovládání systému pomocí hlasu. Pro systém by v tomto případě bylo pravděpodobně nutné navrhnout jiný zdroj napájení, jelikož USB již neposkytuje dostatečně velký proudový odběr.



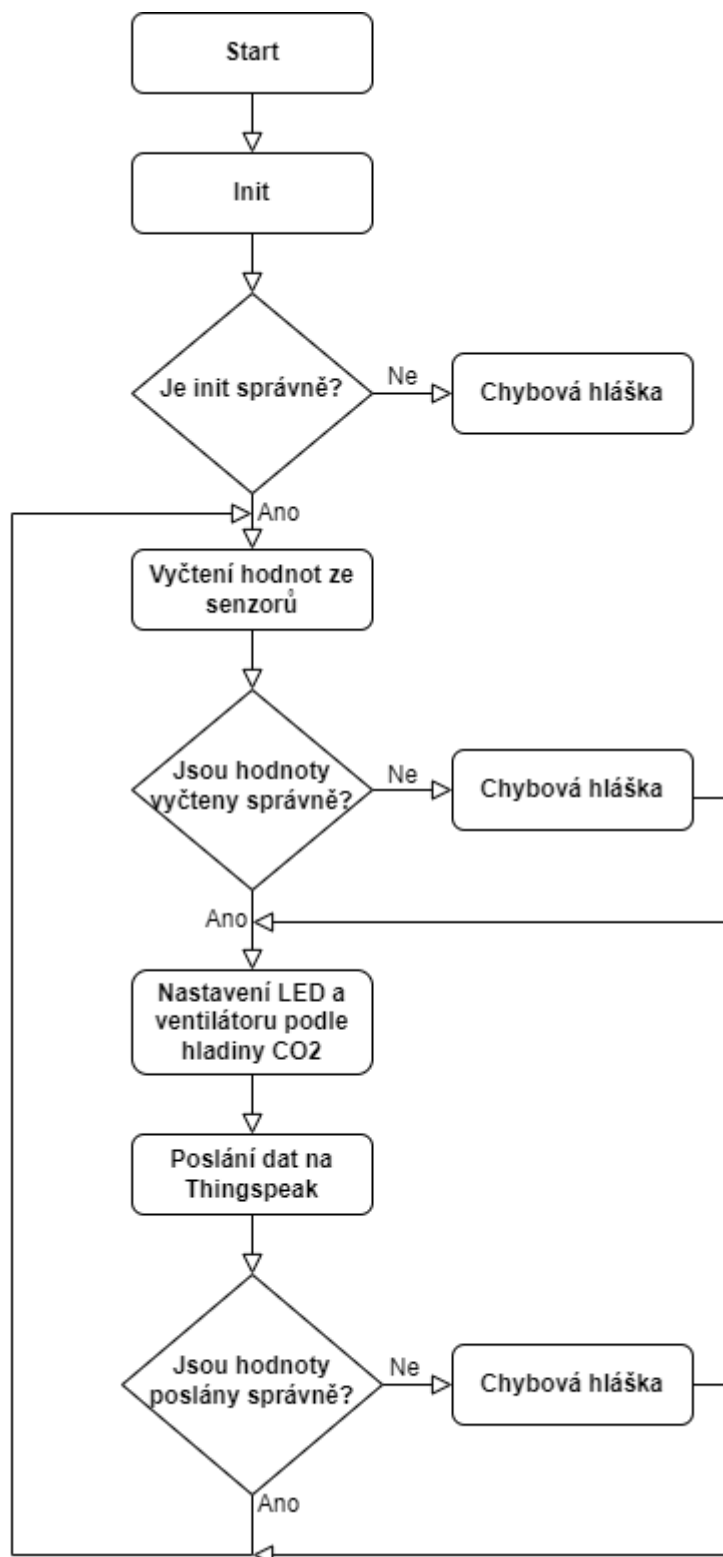
Kapitola 7

Přílohy

1. Strom programu

- Finalprogram
 - Core
 - Inc
 - BMP180.h
 - ESPDataLogger.h
 - LED.h
 - main.h
 - MOSFET.h
 - SCD41.h
 - stm3210xx_it.h
 - UartRingbuffer.h
 - Src
 - BMP180.c
 - ESPDataLogger.c
 - LED.c
 - main.c
 - MOSFET.c
 - SCD41.c
 - stm3210xx_it.h
 - UartRingbuffer.h
 - Finalprogram.ioc

2. Vývojový diagram



Obrázek 7.1: Vývojový diagram fungování firmwaru

Kapitola 8

Popis příloh

1. Strom programu

V této příloze lze vidět strukturu celého navrženého firmwaru. V částech, které jsou pojmenovány po prvcích návrhu, jsou implementace jejich jednotlivých funkcí. ESPDataLogger a UartRingbuffer jsou použity pro komunikaci skrz modul ESP8266. Main je hlavní část programu, kde jsou volány všechny ostatní podprogramy. Stm3210xx_it je použit pro ovládání přerušení. Je zde nutné volat inicializaci UARTU, který je využíván ke komunikaci přes WIFI. Finalprogram.ioc slouží k definování pinů a nastavení hodinového signálu. V kořeni stromu jsou i další soubory, které nejsou vyjmenovány výše. Je to kvůli tomu, že v nich nebyly prováděny žádné změny při sestavování firmwaru a tudíž nejsou podstatné pro jeho pochopení.

2. Vývojový diagram

V druhé příloze je ukázán vývojový diagram znázorňující chod celého programu i s příslušnými chybovými stavy.



Literatura

- [1] Li Jiang, Da-You Liu and Bo Yang, "Smart home research," Proceedings of 2004 International Conference on Machine Learning and Cybernetics (IEEE Cat. No.04EX826), pp. 659-663 vol.2, 2004, doi: 10.1109/ICMLC.2004.1382266.
- [2] D. Marikyan, S. Papagiannidis, and E. Alamanos, "A systematic review of the Smart Home Literature: A user perspective," Technological Forecasting and Social Change, vol. 138, pp. 139–154, 2019, doi: 10.1016/j.techfore.2018.08.015.
- [3] M. R. Alam, M. B. I. Reaz and M. A. M. Ali, "A Review of Smart Homes—Past, Present, and Future," in IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 42, no. 6, pp. 1190-1203, Nov. 2012, doi: 10.1109/TSMCC.2012.2189204.
- [4] S. Kapp, "802.11: leaving the wire behind," in IEEE Internet Computing, vol. 6, no. 1, pp. 82-85, Jan.-Feb. 2002, doi: 10.1109/4236.989005
- [5] Smillie, G. "Analogue and digital communication techniques." Oxford: Elsevier Science and Technology, 1999, ISBN: 0340731257;9780340731253
- [6] P. Chen and Z. Lu, "A Web-Based Indoor Environment Monitoring System Using Wireless Sensor Networks," 2013 International Conference on Computational and Information Sciences, Shiyang, China, 2013, pp. 2007-2010, doi: 10.1109/ICCIS.2013.529
- [7] Christ R. and Robert L. "Rov Manual: A user guide for observation class remotely operated vehicles". Elsevier Science and Technology, pp. 161-163, 2013.
- [8] "SCD4x Datasheet", 2023. [Online]. Available: https://sensirion.com/media/documents/E0F04247/631EF271/CD_DS_SCD40_SCD41_Datasheet_D1.pdf. Accessed on: July 20, 2022.
- [9] "BMP180 Datasheet", 2023. [Online]. Available: <https://cdn-shop.adafruit.com/datasheets/BST-BMP180-DS000-09.pdf>. Accessed on: July 20, 2022.

- [10] “AOD4184A Datasheet“, 2023. [Online]. Available: <https://aosmd.com/products/mosfets/medium-voltage-mosfets-40v-400v/aod4184a>. Accessed on: July 20, 2022.
- [11] “KY-016 Datasheet“, 2023. [Online]. Available: <https://joy-it.net/files/files/Produkte/COM-KY016-RGBLED/COM-KY016RGBLED-Datasheet.pdf>. Accessed on: July 20, 2022.
- [12] “NUCLEO-L053R8“, 2023. [Online]. Available: <https://os.mbed.com/platforms/ST-Nucleo-L053R8/>. Accessed on: July 20, 2022.
- [13] “ESP8266 Datasheet“, 2023. [Online]. Available: https://nurdspace.nl/images/e/e0/ESP8266_Specifications_English.pdf. Accessed on: July 20, 2022.
- [14] F. Leens, "An introduction to I2C and SPI protocols," in IEEE Instrumentation & Measurement Magazine, vol. 12, no. 1, pp. 8-13, February 2009, doi: 10.1109/MIM.2009.4762946.
- [15] Technologies D. (2022, March 27). "What is a CO2 sensor and how does it work?" Sensors; IoT Infrastructure. Retrieved March 1, 2023, from <https://www.disruptive-technologies.com/blog/what-is-a-co2-sensor-and-how-does-it-work>
- [16] “How Do CO2 Sensors Work“. PROCESS SENSING TECHNOLOGIES. (n.d.). Retrieved March 5, 2023, from <https://www.processsensing.com/en-us/blog/how-CO2-sensors-work.htm>
- [17] Topinfo s.r.o. (2023, March 1). "Vlhkost vzduchu a její měření." TZB. Retrieved March 1, 2023, from <https://vetrani.tzb-info.cz/teorie-a-vypocty-vetrani-klimatizace/3137-vlhkost-vzduchu-a-jeji-mereni>
- [18] “What is hygrometer?“ - definition, resistive, capacitive, & crystal hygrometer. Circuit Globe. (2018, January 15). Retrieved March 5, 2023, from <https://circuitglobe.com/hygrometer.html>
- [19] ĎAĎO, Stanislav. "Senzory a měřicí obvody". Vyd. 2. Praha: ČVUT, 1999. 315 s. ISBN 80-01-02057-6.
- [20] “Piezoresistive pressure sensors“. Avnet. (n.d.). Retrieved March 5, 2023, from <https://www.avnet.com/wps/portal/abacus/solutions/technologies/sensors/pressure-sensors/core-technologies/piezoresistive-strain-gauge/>
- [21] doc. Dr. Ing. Hospodka, J. (n.d.). “Základní elektronické prvky a jejich modely“. 13. pp. 22-27. Retrieved March 7, 2023, from <https://hippo.feld.cvut.cz/vyuka/soubory/modely.pdf>.
- [22] “Acoustic thermometry“. Merriam-Webster.com Dictionary, Merriam-Webster, <https://www.merriam-webster.com/dictionary/acoustic%20thermometry>. Accessed 7 Mar. 2023.

- [23] Williams, J., & Sanchez-Felipe, O. (2011, February). An introduction to acoustic thermometry. Linear Technology. Retrieved March 7, 2023, from <https://www.analog.com/media/en/technical-documentation/application-notes/an131f.pdf>
- [24] "LD1117 Datasheet", 2023. [Online]. Available: <https://www.st.com/resource/en/datasheet/ld1117.pdf>. Accessed on: April 6, 2023.
- [25] Admin. (2023, February 10). "How to log data into Thingspeak using STM32 and ESP8266". ControllersTech. Retrieved May 6, 2023, from <https://controllerstech.com/data-logger-using-stm32-and-esp8266/>
- [26] Admin. (2023, February 10). "Interface BMP180 with STM32" controllerstech. Retrieved May 6, 2023, from <https://controllerstech.com/interface-bmp180-with-stm32/>
- [27] Bhimsen. (2021, December 29). "ESP8266 at commands list and working explained". Electronics fun. Retrieved May 7, 2023, from <https://electronics-fun.com/esp8266-at-commands/>