



F3

**Fakulta elektrotechnická
Katedra elektrických pohonů**

Bakalářská práce

Model chytré domácnosti

Josef Šimek

Elektrotechnika, energetika a management

Únor 2023

Vedoucí práce: Ing. Jakub Svatoš, Ph.D.

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Šimek** Jméno: **Josef** Osobní číslo: **495647**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávající katedra/ústav: **Katedra elektrických pohonů a trakce**
Studijní program: **Elektrotechnika, energetika a management**
Specializace: **Aplikovaná elektrotechnika**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Model chytré domácnosti

Název bakalářské práce anglicky:

Smar Home Model

Pokyny pro vypracování:

- 1) Navrhněte a realizujte model chytré domácnosti. Řídicím prvkem celého systému bude mikroprocesor s možností přístupu přes mobilní aplikaci.
- 2) Navrhněte systém, který umožní monitorovat základní parametry týkající se standardního komfortu domácnosti jako je např. teplota, CO₂, vlhkost a pod.
- 3) Vytvořte uživatelské prostředí, se zobrazením měřených veličin a prvky na jednoduché ovládání.

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: **01.12.2022**

Termín odevzdání bakalářské práce: _____

Platnost zadání bakalářské práce: **24.09.2024**

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 / Prohlášení

Děkuji vedoucímu práce Ing. Jakobovi Svatošovi, Ph.D. za projevenou vstřícnost a trpělivost v průběhu plnění práce.

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

V Praze dne 26. 5. 2023

.....

Abstrakt / Abstract

Práce se věnuje návrhu systému, pro sledování a ovládání základních parametrů a prvků standardního komfortu domácnosti. Teoretická část se věnuje technologiím a jejich principům, které by bylo možné aplikovat. Rozbor řešení je zaměřen na výběr součástí a metod. Realizace obsahuje postup zprovoznění dílčích prvků skládající funkční systém. Závěrečná část nazvaná testování ověřuje spolehlivost ovládání a monitorování základních parametrů.

Klíčová slova: Chytrá domácnost, mikroprocesor, komunikace, měření, senzor, teplota, vlhkost, CO_2

The thesis deals with the design of a system for monitoring and controlling basic parameters and elements of standard home comfort. The theoretical part is devoted to technologies and their principles that could be applied. The analysis of the solution is focused on the selection of components and methods. The implementation includes the procedure of commissioning the sub-elements composing the functional system. The final part, called testing, verifies the reliability of the control and monitoring of the basic parameters.

Keywords: Smart home, microprocessor, communication, measurement, sensor, temperature, humidity, CO_2

Obsah /

| | | | |
|---|-----------|--|-----------|
| Úvod | 1 | 3.5 Příprava webové aplikace | 26 |
| 1 Teoretická část | 2 | 4 Testování | 28 |
| 1.1 Mikropočítač | 2 | 4.1 Provoz komunikace s clou- dovou službou | 28 |
| 1.2 Senzory | 2 | 4.1.1 Naměřené hodnoty | 28 |
| 1.2.1 Měření teploty | 2 | 4.1.2 Ovládací prvky | 28 |
| 1.2.2 Měření relativní vlh- kosti vzduchu | 5 | 4.2 Naměřené hodnoty a jejich průběh | 29 |
| 1.2.3 Měření kvality vzduchu | 6 | Závěr | 32 |
| 1.3 Cloudová služba a webová aplikace | 7 | zdroje | 33 |
| 1.3.1 Cloudová služba | 7 | | |
| 1.3.2 Webová aplikace | 7 | | |
| 1.4 Lokální síť a komunikační protokol MQTT | 7 | | |
| 1.4.1 LAN | 7 | | |
| 1.4.2 Komunikační protokol MQTT | 8 | | |
| 2 Rozbor řešení | 9 | | |
| 2.1 Blokové schéma | 9 | | |
| 2.1.1 I2C | 9 | | |
| 2.1.2 1-Wire | 10 | | |
| 2.1.3 PWM | 10 | | |
| 2.2 Výběr komponentů | 11 | | |
| 2.2.1 Senzor pro měření tep- loty a vlhkosti | 11 | | |
| 2.2.2 Senzor pro měření kva- lity vzduchu | 12 | | |
| 2.2.3 Světelné zdroje | 14 | | |
| 2.2.4 Mikropočítač | 14 | | |
| 2.3 Výběr cloudové služby pro provoz webové aplikace | 15 | | |
| 2.4 Dispozice pozorovaného prostředí | 16 | | |
| 3 Realizace | 17 | | |
| 3.1 Zprovoznění mikropočítače | 17 | | |
| 3.1.1 Vzdálený přístup | 17 | | |
| 3.2 Programování a funkce senzorů | 19 | | |
| 3.2.1 Senzor DHT22 | 20 | | |
| 3.2.2 Senzor SCD41 | 21 | | |
| 3.3 Světelná signalizace a ovlá- dání osvětlení | 23 | | |
| 3.4 Obslužný program | 24 | | |
| 3.4.1 Spojení s webovou aplikací | 24 | | |
| 3.4.2 Základní funkce | 25 | | |
| 3.4.3 Odesílání hodnot | 26 | | |

Tabulky / Obrázky

| | | |
|-------------|--|----|
| 1.1 | Vliv oxidu uhličitého na lidský organismus | 6 |
| 2.1 | Parametry senzoru DHT22 | 12 |
| 2.2 | Parametry senzoru SCD41..... | 13 |
| 1.1 | Odporový kovový teploměr | 3 |
| 1.2 | Tolerance platinového senzoru ... | 3 |
| 1.3 | Tolerance niklových měřících odporů | 4 |
| 1.4 | Charakteristika termistorů | 4 |
| 1.5 | Struktura a charakteristika negastoru | 5 |
| 1.6 | Senzor vlhkosti a jeho charakteristika | 5 |
| 1.7 | Princip měření plynu oxidu uhličitého | 7 |
| 1.8 | Architektura MQTT protokolu .. | 8 |
| 2.1 | Blokové schéma | 9 |
| 2.2 | Průběh signálů protokolu I2C . | 10 |
| 2.3 | Interpretace průběhu signálu PWM | 11 |
| 2.4 | Senzor DHT22..... | 12 |
| 2.5 | Senzor LaskaKit SCD41 | 13 |
| 2.6 | Mikropočítač Raspberry Pi 4 .. | 14 |
| 2.7 | Cloudová služba Adafruit IO .. | 15 |
| 2.8 | Funkční bloky webové aplikace | 16 |
| 3.1 | Příkaz pro spuštění konfiguračního nástroje | 17 |
| 3.2 | Volba možnosti konfigurace rozhraní..... | 18 |
| 3.3 | Nabídka nastavení komunikačního rozhraní | 18 |
| 3.4 | Zobrazení IP adresy..... | 18 |
| 3.5 | Povolení rozhraní pro I2C a 1-Wire | 20 |
| 3.6 | Nahrání knihovny pro sensor DHT22..... | 20 |
| 3.7 | Zahájení komunikace se senzorem DHT22 | 20 |
| 3.8 | Zakódované hodnoty v signálu . | 21 |
| 3.9 | Proces měření teploty a vlhkosti..... | 21 |
| 3.10 | Detekce adresy I2C sběrnice ... | 22 |
| 3.11 | Nahrání knihovny pro SCD41 . | 22 |
| 3.12 | Změna hodnoty nadmořské výšky senzoru LaskaKit SCD41 | 22 |
| 3.13 | První příkaz i2c | 23 |
| 3.14 | Čti i2c | 23 |

| | | |
|-------------|--|----|
| 3.15 | Obsluha senzoru LaskaKit SCD41 | 23 |
| 3.16 | Inicializace výstupních pinů ... | 24 |
| 3.17 | Signalizace dosažení úrovně CO ₂ | 24 |
| 3.18 | Intenzita světla | 24 |
| 3.19 | Získání klíče pro komunikaci s cloudovou službou | 25 |
| 3.20 | Zadání přihlašovacích para- metrů | 25 |
| 3.21 | Implementace přihlašovaích údajů | 25 |
| 3.22 | Definování funkcí programu ... | 26 |
| 3.23 | Odesílání hodnot | 26 |
| 3.24 | Seznam feedů | 26 |
| 3.25 | Nastavení bloku | 27 |
| 3.26 | Uživatelské prostředí | 27 |
| 4.1 | Konzole s výpisem naměře- ných hodnot | 28 |
| 4.2 | Konzole s výpisem nastave- ných hodnot nebo stavu | 28 |
| 4.3 | Interpretace naměřených hodnot | 29 |
| 4.4 | Testování systému | 29 |
| 4.5 | Průběh naměřených hodnot CO ₂ | 30 |
| 4.6 | Průběh naměřených hodnot teploty | 30 |
| 4.7 | Průběh naměřených hodnot teploty | 31 |

Úvod

Koncept chytré domácnosti spočívá v technickém řešení pomocí automatizace, tedy možnosti regulace fyzikálních veličin, uvnitř budovy. Díky technickým inovacím, které umí monitorovat stav domácnosti, se může zvýšit životní úroveň uživatele. Monitorování může být zpotřebkováno chytrým zařízením, využívající internet, které je v angličtině označováno jako IoT (Internet of Things). [1]

Jelikož existuje spousta možností, jak toto téma obsáhnout, je vhodné si v úvodní části vymežit, čím se bude práce zabývat. Práce se bude zabývat měřením relativní vlhkosti vzduchu, teploty a kvality vzduchu v obývaných prostorách a jejich analýzou včetně historického vývoje. Pozorované veličiny se budou měřit pomocí senzorů a zpracovávat mikropočítačem obsahující mikroprocesor s následným odesláním na cloudovou službu. Na ní bude v provozu webová aplikace, kterou bude možné navštívit chytrým telefonem nebo tabletem.

Vlhkost má vliv nejen na stav obývaných prostor, ale také na zdraví členů, kteří je obývají. Nízká relativní vlhkost by mohla mít za následek podráždění sliznic a očí, vyšší náchylnost k infekcím nebo v extrémním případě narušení imunitního systému. Vysoká relativní vlhkost může napomáhat k růstu plísní na chladnějších plochách a k násobnému rozmnožování různých mikroorganismů. [2]

Teplota má vliv na tepelný komfort. Avšak k dosažení její optimální hodnoty je nutné prostory vybavit zdrojem tepla nebo chladu. Aby bylo možné je automatizovaně ovládat, je nutné monitorovat aktuální hodnotu teploty. Díky tomu lze včas nastavit parametry výkonu zdroje tak, aby se co nejvíce zefektivnil čas provozu. [3]

Měřením kvality vzduchu je míněno sledování koncentrace plynu oxidu uhličitého CO_2 , který má vliv na stav lidského organismu. Oxid uhličitý je produktem procesu dýchání a jeho vysoký podíl v inhalovaném vzduchu může způsobit zdravotní problémy. Tento parametr je tudíž nutné sledovat v uzavřených prostorech, aby mohla být zajištěna dostatečná výměna vzduchu pro snížení podílu CO_2 , který by mohl způsobit zdravotní dopady. [4]

Obecně je známo, že by se měla dodržovat perioda výměny vzduchu větráním okny každé dvě hodiny minimálně po dobu pěti minut. Potíž nastává v průběhu spánku, kdy lidé nemohou zajistit obměnu vydýchaného vzduchu se zvýšeným obsahem plynu CO_2 . [5] Z toho plyne, že při průměrném spánku osmi hodin denně jsou zanedbána čtyři větrání.

Cílem práce je návrh systému včetně realizace pro modelovou situaci a posouzení přínosů, které by bylo možné aplikovat v praxi. Především bude zaměřena na monitorování naměřených hodnot pomocí vzdáleného přístupu skrze mobilní zařízení.

Kapitola 1

Teoretická část

V rámci této části práce budou uvedeny různé metody a techniky, pomocí kterých lze najít řešení pro danou problematiku. Budou zde prezentovány především principy, na kterých se mohou stavět návrhy řešení.

1.1 Mikropočítač

Je elektronické zařízení tvořené několika základními elektrotechnickými komponenty. Hlavní částí je centrální procesorová jednotka (CPU), někdy označována jako micro-procesor. Ta je doplněna o nezbytné rozhraní, která umožňují interagovat se vstupními a výstupními periferiemi. Na rozdíl od microcontrolleru obsahuje operační systém s uživatelským prostředím. Tyto termíny bývají často zaměňovány. Díky miniaturizaci výpočetní techniky v posledním dvacetiletí jsou v současné době k dostání mikropočítače přibližně o velikosti platební karty.

1.2 Senzory

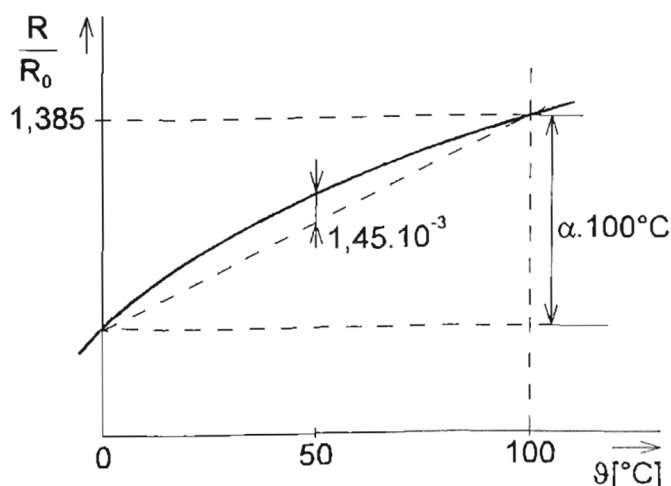
Získané informace o stavu nebo hodnotě sledovaných veličin lze převést na elektrický signál. Právě tuto funkci zastávají senzory, někdy také nazývané jako čidla nebo snímače. Je potřeba zvážit jaká bude jejich aplikace a jaké veličiny nebo stavy se budou pozorovat, protože se od sebe principiálně liší. [6]

1.2.1 Měření teploty

Měření teploty lze rozdělit do dvou základních druhů a to na bezdotykové a dotykové. [6] Hlavním rozdílem je místo měření. Dotykové umí měřit pouze v místě nebo v blízkém okolí, kde je umístěna měřící technika. Bezdotykové neměří v oblasti umístění, nýbrž v určité vzdálenosti od pozorovaného objektu. [7] Tento způsob měření by se tak mohl uplatnit případech, u kterých není možné se s měřící technikou přiblížit na dané místo nebo by přiblížení zvýšilo riziko negativních následků na zdraví pozorovatele. Bezdotykové měření bylo možné pozorovat v době, kdy probíhal vrchol pandemie COVID-19. Z toho lze vyvodit, že zmíněný typ měření bude vhodnější především aplikovat na měření teploty povrchu konkrétních fyzických těles. V našem případě nebudeme měřit teplotu konkrétního předmětu, budeme měřit teplotu vzduchu, který bude tvořit pozorované prostředí. Zaměříme se tedy na dotykové senzory teploty.

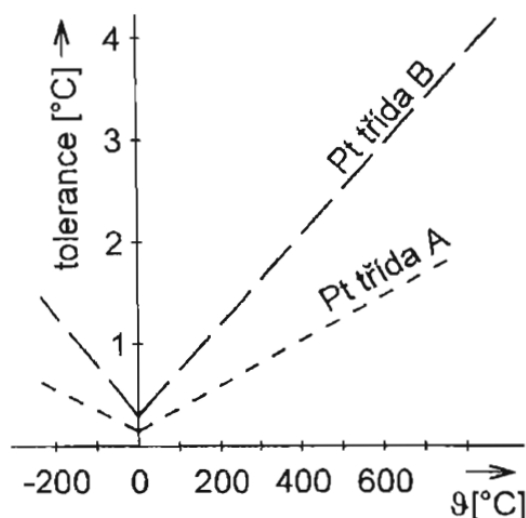
Mezi ně patří například odporové teploměry na bázi kovů.

Ty jsou nejčastěji vytvořeny z materiálu platina, ta má vyšší teplotní stabilitu nežli polovodičové senzory. Nejvíce obecně známý teplotní senzor z platiny je Pt100. Z jeho názvu lze odvodit, že nominální teplota odporu při 0° je 100 Ω. [6]



Obrázek 1.1. Závislost velikosti relativního odporu platiny na teplotě. [6]

Při pohledu na obrázek 1.1 je patrné, že se hodnota odporu s rostoucí teplotou bude zvyšovat. V zobrazeném rozsahu od 0 do 100 °C se hodnota odporu zvyšuje téměř lineárně, čímž lze senzor použít pro takto velký rozsah teplot.

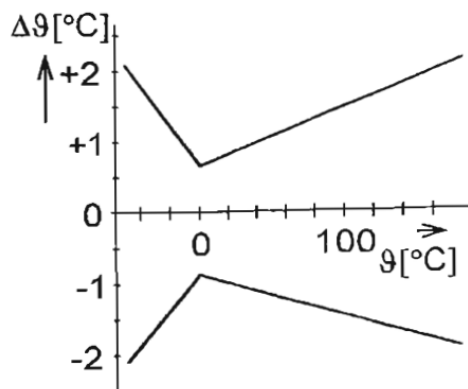


Obrázek 1.2. Tolerance standardních měřících platinových odporů dle IEC. [6]

Z uvedeného obrázku 1.2 je patrné, že podle International Electrotechnical Commission (IEC) se člení na dvě třídy. Každá z tříd se od sebe odlišuje rozsahem teplot a tolerancí teploty. Zaměříme-li se na toleranci do 100 °C, což je nejčastější rozsah teploty v obývaných prostrátech. Třída A má toleranci při 0 °C $\pm 0,125$ °C, tomu odpovídá $\pm 0,06$ Ω což může být označeno jako přesnější měření a vhodnější pro naše použití. Za nepravděpodobné lze označit potřebu měřit teploty mezi 650 °C a 850 °C, kde třída A již nesplňuje rozsah, tudíž bylo nutné zvolit třídu B za cenu vyšší tolerance.

Dalšími odporovými kovovými teploměry, které je vhodné zmínit, jsou niklové.

Nevýhody, kterou niklové teploměry disponují v poměru s platinovým jsou: omezený teplotní rozsah a značná nelinearita. [6] Z přiloženého obrázku 1.3 v uvedeném rozsahu, lze vyvodit, že oproti platinovému disponuje i vyšší tolerancí.



Obrázek 1.3. Tolerance niklových měřících odporů dle Deutsches Institut für Normung. [6]

Pro kovové teploměry v rozmezí teplot 0 až 100 °C platí:

$$R = R_0 \cdot (1 + \alpha\vartheta) \quad (1)$$

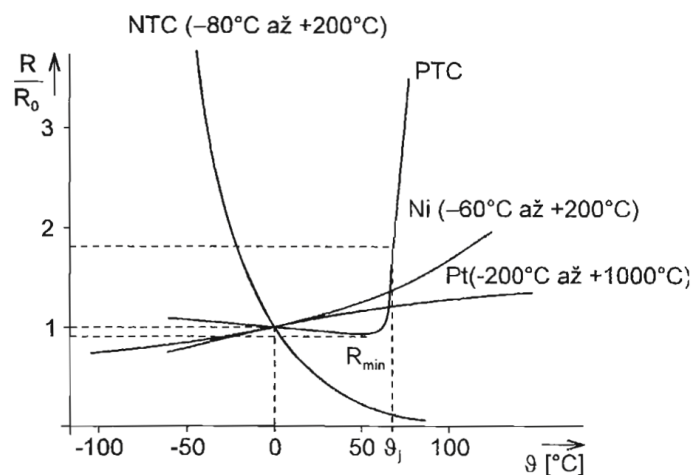
kde $\alpha[K]$ je teplotní součinitel odporu, $R_0[\Omega]$ velikost odporu při teplotě 0 °C, ϑ [°C] teplota

Teplotní součinitel lze určit i empiricky a to pomocí vztahu:

$$\alpha = \frac{R_{100} - R_0}{100 \cdot R_0} \quad (2)$$

Další alternativa pro dotykové měření teploty je proměnlivý polovodičový odpor neboli termistor.

Ten v závislosti na teplotě mění svoji hodnotu odporu. Existují dva typy a to posistor (PTC) a negastor (NTC). [6] Rozdíl mezi těmito dvěma typy je, že PTC zvyšuje svoji hodnotu odporu při zvyšování teploty zatímco NTC snižuje svoji velikost odporu. Lze tak sledovat pomocí převodníku odpor/napětí změnu úbytek napětí a v kombinaci s analog/digital převodníkem tak nepřímou získat digitální hodnoty teploty v okolí místa měření.



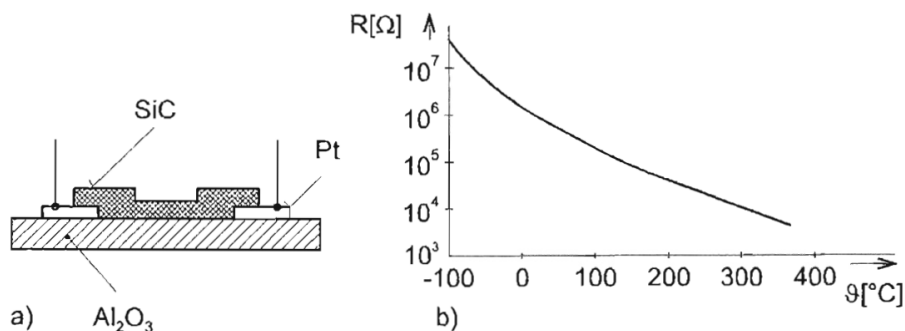
Obrázek 1.4. Závislost velikosti relativního odporu termistoru na teplotě. [6]

Závislost tak představuje potenciální komplikaci při uplatnění tohoto typu senzoru v regulaci, řešením může být linearizace.

Velikost odporu pro NTC je dána závislostí:

$$R_1 = R_0 \cdot e^{B\left(\frac{1}{T_1} - \frac{1}{T_0}\right)} \quad (3)$$

kde $R_1[\Omega]$ je odpor termistoru při teplotě T_1 , $R_r[\Omega]$ odpor termistoru při referenční teplotě T_r , $B[K]$ teplotní konstanta daného materiálu

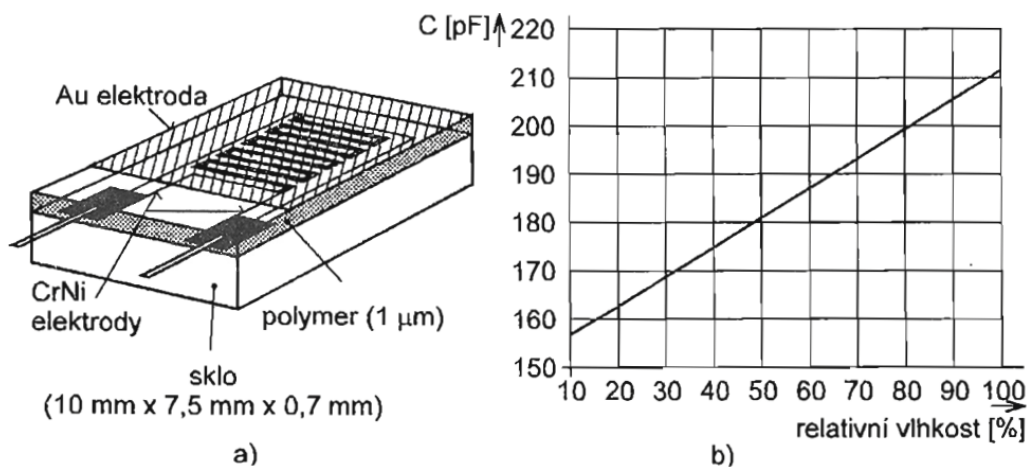


Obrázek 1.5. Negastor typu SiC (a - struktura, b - charakteristika). [6]

Je potřeba také neopomenout vhodné místo k měření teploty. V případě, že je senzor umístěn co nejbližší k podlaze, naměří se nižší teplota nežli u stropu, protože molekuly teplého vzduchu se pohybují rychleji než molekuly chladného vzduchu, čímž vznikne rozdíl tlaků a pomocí nich se teplejší vzduch vytlačí výše.

1.2.2 Měření relativní vlhkosti vzduchu

Procentuální relativní vlhkost je možné měřit na základě změny kapacit mezi elektrodami. Vrstva izolantu, která bývá tvořená polymerem, reaguje na molekuly vody. [6] Senzor tak zvyšuje svoji kapacitu, jejímž důsledkem je snižující se impedance, která způsobí vyšší průtok proudu, je-li připojeno napětí. Prostřednictvím převodníku proud/napětí a analog/digital je možné získat hodnotu relativní vlhkosti vzduchu na digitální hodnotu a tu následně zpracovat.



Obrázek 1.6. Kapacitní senzor vlhkosti (a - struktura, b - typická charakteristika). [6]

Velikost impedance sériového zapojení kapacitního charakteru:

$$Z = \sqrt{R^2 + \frac{1}{(\omega \cdot C)^2}} \quad (4)$$

kde $Z[\Omega]$ je výsledná impedance, $R[\Omega]$ velikost sériového odporu, $\omega[Hz]$ úhlová frekvence, $C[F]$ kapacita kapacitoru

K tomuto typu řešení existuje alternativa v podobě polovodičové součástky a to MOS tranzistoru. Tranzistor obsahuje polyetylenoxid, který má hydrofilní vlastnosti. Důsledkem setkání s vlhkostí je tak i změna elektrických vlastností ovlivňující průtok proudu zapříčiňující úbytek napětí. [6] Tímto způsobem se mění vodivost neboli odpor, který lze změřit a digitalizovat jej pro další zpracování.

1.2.3 Měření kvality vzduchu

Tato tematika se zabývá měřením koncentrace plynu oxidu uhličitého, který je produkován člověkem v uzavřených prostorách. Především je nevědomě vytvářen při vydechování a v situaci, kdy není dostatečný přísun čerstvého vzduchu. Důsledkem může být negativní dopad na kognitivní funkce člověka. Měří se v absolutních hodnotách a to v jednotce ppm, což je anglická zkratka pro parts per million. S přibývajícím zastoupením CO_2 v nedostatečně větraném prostředí to může mít za následek významný negativní dopad na lidský organismus a to tak, že v extrémních případech může dojít ke ztrátě vědomí. [4]

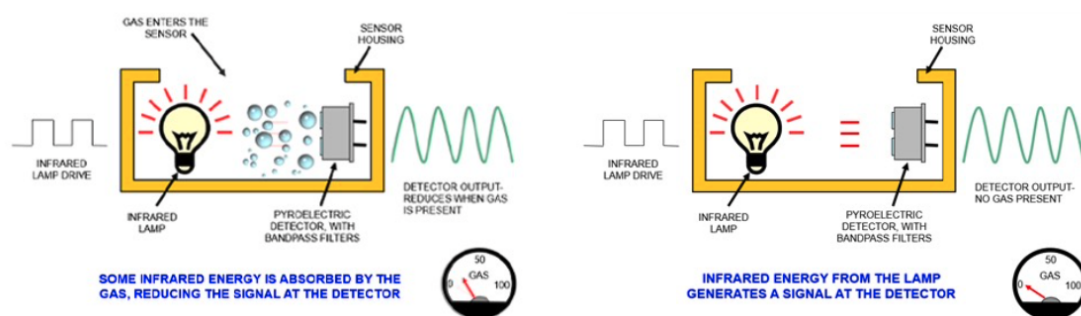
Tímto tématem se zabývají především v problematice pasivních domů, v literatuře také označovány jako nulově energetické domy. Ty mají téměř dokonale utěsněné prostory proto, aby nevnikaly oblasti odkud by unikalo teplo či chlad. Odpovědí je technické řešení realizované pomocí nuceného větrání. Toto samotné opatření pro udržení vytvořeného mikroklimatu nestačí, proto tento systém obsahuje i rekuperační jednotku. Ta disponuje dvěma kanály a to pro nasávaný a odsávaný vzduch. V průběhu výměny vzduchu dochází k distribuci tepelné energie mezi nasávaným a odsávaným vzduchem. Za nejkvalitnější rekuperační jednotky se označují takové, které mají účinnost přibližně 90%. [5]

| ppm | koncentrace a její vliv |
|-----------|---|
| 350-400 | venkovní prostředí |
| -1000 | doporučená úroveň v uzavřených prostorách |
| 1200-1500 | maximální doporučená úroveň v uzavřených prostorách |
| 1000-2000 | počátky příznaků únavy a snižování koncentrace |
| 2000-5000 | možné bolesti hlavy |
| 5000 | nevolnost a zvýšený tep |
| 15000 | dýchací potíže |
| 40000 | možná ztráta vědomí |

Tabulka 1.1. Vliv koncentrace oxidu uhličitého na lidský organismus.

Jedna z možností, jak měřit tento plyn je nedisperzní infračervené záření, v anglické literatuře označovány jako NDIR (nondispersive infrared). Na molekuly plynu působí infračervené záření o vlnové délce $4,3 \mu m$, protože plyn CO_2 má pro tuto hodnotu nejvyšší absorbitivitu. Na této vlastnosti je postaven princip měření. V situaci, kdy je zvýšený výskyt plynu, je záření absorbováno a tím se sníží jeho intenzita. Následně

intenzita je monitorována pyroelektrickým detektorem, který jej zaznamenává. Oproti tomu, při absenci výskytu plynu, není záření absorbováno a veškerá intenzita zdroje je zaznamenána. Výsledkem je elektrický signál pro další vyhodnocení. [8]



Obrázek 1.7. Stav měření bez molekul plynu typu NDIR. [8]

Alternativou k NDIR je měření pomocí fotoakustická spektroskopie. Tato metoda měření je založena na mechanických vlastnostech molekul plynu CO_2 , které jsou zdrojem elektromagnetického záření, opět v podobné vlnové délce, rozkmitány. Mikrofonem, který zaznamenává v reálném čase akustické vlny tvořené molekulami, se převádí množství plynu na elektrický signál. Ten je následně zpracován pomocí elektronického chipu pro digitalizaci. [9].

1.3 Cloudová služba a webová aplikace

1.3.1 Cloudová služba

Je specifická internetová služba, ke které je možné odkudkoli přistupovat. Lze využít jejich programů nebo aplikací, které jsou zprostředkovány poskytovatelem. Nejvyužívanější takovou službou je e-mailový klient, který je možný spustit v prohlížeči internetových stránek. Také je na nich možné zprovoznit chod vlastních programů. [10]

1.3.2 Webová aplikace

Internetový prohlížeč může být využit pro zprostředkování webové aplikace, která běží na webovém serveru. Díky tomu není potřeba uplatnit výkon výpočetní zařízení, na které by se musela aplikace instalovat. [11]

1.4 Lokální síť a komunikační protokol MQTT

1.4.1 LAN

Jedním ze způsobů, jak ovládat počítače, které mají možnost síťového připojení, je využití lokální sítě, pomocí které se na ně připojíme.

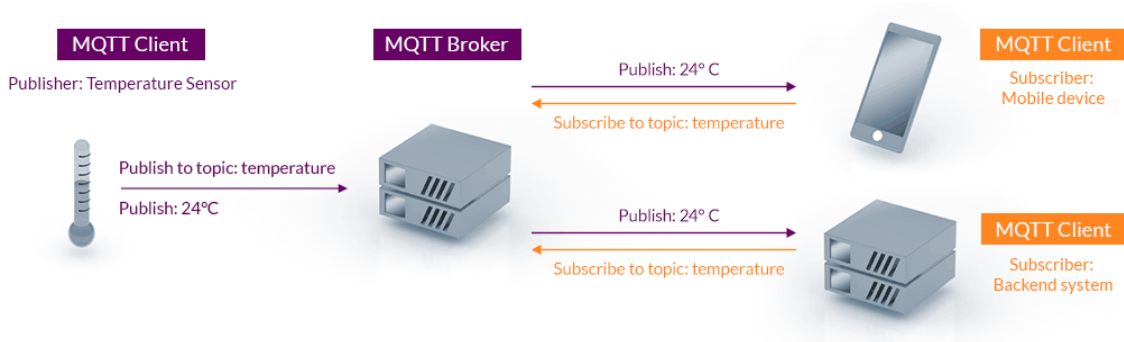
Síť označovaná jako lokální je zpravidla taková, která neumožňuje veřejný přístup, ale slouží ke zprostředkování komunikace nebo výměně dat ve vymezeném prostoru jako je například uvnitř místnosti nebo mezi nimi. [12]

1.4.2 Komunikační protokol MQTT

Elementární komunikaci, obsahující informace, lze zprostředkovat pomocí protokolu Message Queuing Telemetry Transport, známý pod zkratkou MQTT.

Ten je založen na triviálním odesílání a přijímání zpráv [13], těmi mohou být například čísla nebo textové řetězce.

Princip protokolu se skládá ze dvou typů klientů: publisher a subscriber. Publisher je schopný zprávy pouze odesílat, subscriber je způsobilý zprávy i přijímat. Protokol je postaven na interakci klienta (zařízení, které umí komunikovat skrze Transmission Control Protocol) připojeného k brokeru (označení pro server). Náplní brokeru je zprostředkovat komunikaci zpráv mezi jednotlivými klienty, řízenou na základě jejich oprávnění k přístupu a žádosti. Poté, co broker akceptuje zprávy od klienta s označením „publisher“, jsou jím odeslány klientovi s označením „subscriber“, který je přihlášen k jejich přijímání. Přenos zpráv se uskutečňuje v deklarovaném tématu, v protokolu je identifikované jako Topic, do kterého jsou oba klienti zainteresováni. [14]



Obrázek 1.8. Ilustrace komunikace protokolu MQTT. [13]

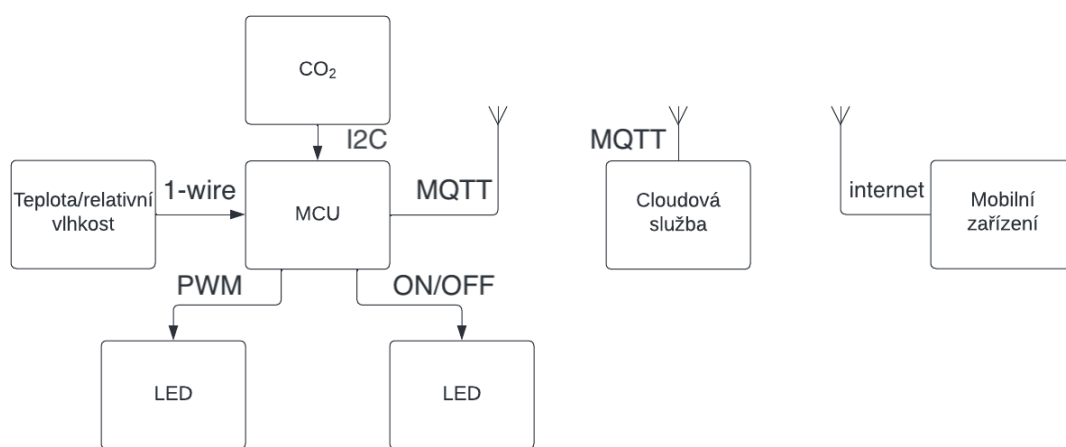
Kapitola 2

Rozbor řešení

Tato kapitola bude zaměřena na výběr řídicí jednotky a senzorů, pomocí kterých se budou měřit naše pozorované veličiny a jejich zapojení. V první části se budeme věnovat výběru řídicí jednotky tvořící mozek celého systému. Poté se budou volit senzory pro měření teploty, vlhkosti a koncentrace oxidu uhličitého. Obsahem této kapitoly bude také znázornění komunikace mezi jednotlivými prvky pomocí blokového schématu, podle kterého se bude realizovat zapojení.

2.1 Blokové schéma

Pro získání představy, dle které se bude odvíjet realizace, je vhodné vizualizovat, jak by měla být provedena. Především se uplatňují u rozsáhlých projektů, u kterých se tak předchází výskytu problémů a lze tak zjednodušit náročné procesy, které na sebe mohou a nemusí navazovat. Z grafického znázornění se bude odvíjet zapojení součástí, programování mikropočítače, směr komunikace a záležitosti s tím spojené.



Obrázek 2.1. Blokové schéma návrhu

Z přiloženého diagramu z obrázku 2.1 je zřejmé, že návrh řešení obsahuje bezdrátovou komunikaci s mobilním zařízením a prostředníkem v podobě cloudové služby. K mikropočítači budou připojeny senzory, které umožňují jednosměrně odesílat data. Podle tohoto schématu by realizace měla být minimálně ideově dodržena. Z toho plyne, že senzory budou zapojeny buď přímo do řídicí jednotky nebo do centralizovaného místa, v tomto případě do nepájivého pole, stejně tak LED.

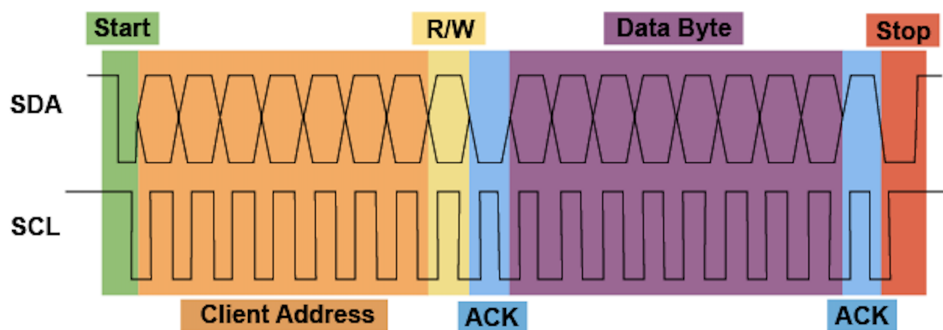
2.1.1 I2C

Přenos naměřených dat ze senzoru CO₂ 2.5 bude zajištěn sběrníkovým rozhraním pomocí I2C protokolu.

Zkratka I2C je označení pro Inter-Integrated Circuit. Tento protokol je uplatňován pro sériovou komunikaci mezi čipy skrze dvě komunikační linky. Jedna z komunikačních linek, která je využita pro přenos dat je označena SDA a druhá, která zajišťuje hodinový signál, je označena SCL. [15]

Po datové lince SDA je komunikace signálu rozdělena na dva typy zařízení a to na master a slave. Počátek komunikace oznamuje zařízení master logickým stavem high, čímž jsou upozorněna všechna slave zařízení, že dojde k jejich možné adresaci. Master vyšle adresu zařízení slave se kterým bude chtít navázat kontakt, poté následuje část signálu, která obsahuje informaci, jestli budou data čtena, nebo zapisována. V případě úspěchu správné adresace je tato část potvrzena signálem acknowledge (ACK) o úspěšném přijetí, který vyšle slave. Druhá část obsahuje data, jenž jsou rozdělena do dvou datových polí. Mezi nimi a za nimi se opět nachází část signálu pro kontrolu přijetí, kterou vyšle buď master nebo slave podle toho, zda-li šlo o zápis nebo čtení dat. Konec komunikace je oznámen opět zařízením master logickým stavem low. Druhý signál jsou hodiny, označeny jako SCL, ten zprostředkovává synchronizaci a rychlost přenosu dat. [15]

Z toho lze vyvodit, že díky adresaci lze připojit více zařízení na jednu sběrnici a ušetřit tak počet portů. Protože volbou adresy si můžeme volit s čím budeme chtít navázat kontakt.



Obrázek 2.2. Průběh signálů protokolu I2C. [16]

2.1.2 1-Wire

Senzor teploty a vlhkosti 2.1 bude komunikovat prostřednictvím protokolu 1-Wire.

Jak je již v názvu obsaženo, jde o komunikaci pomocí jedné komunikační linky.

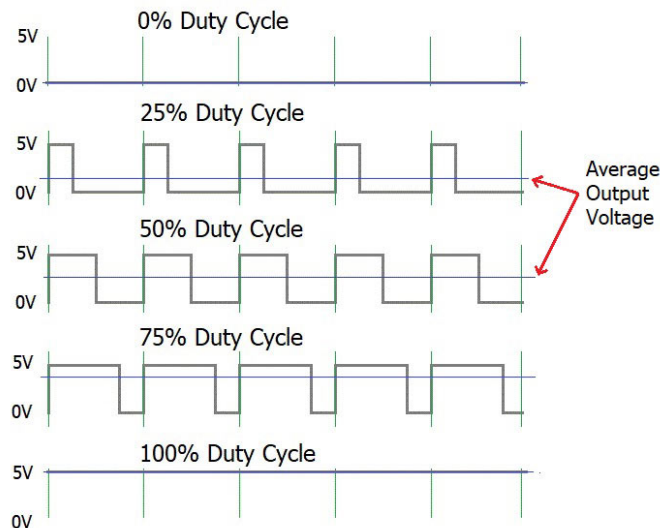
Signál nabývá dvou stavů a sleduje se v intervalech délka signálu. Delší signál reprezentuje logickou 1 a signál s kratším trváním reprezentuje logickou 0. Doba trvání signálu je v jednotkách až desítkách mikrosekund. Jedná se o komunikaci s nízkou rychlostí přenosu. [17]

Z toho vyplývá, že senzor obsahuje čip, který umí vytvářet signály nebo pulzy. Pro zpracování zakódovaných informací tímto protokolem je budeme muset umět zpracovat.

2.1.3 PWM

Řízení v digitálním prostředí, lze vykonávat pouze dvoustavově. Aby se mohla měnit hodnota digitálního signálu využívá se pulzně šířkové modulace, která je obecně známa pod zkratkou PWM. Jedná se možnost, jak měnit střední hodnotu s dvoustavových

zdroji signálu. Ve slaboproudých aplikacích nabývá dvou hodnot 0 V a V_{cc} . V dané periodě se mění doba signálu. Mění se tak relativní hodnota od minimální (0%) až po maximální (100%). V tomto případě jím bude ovládané osvětlení LED.



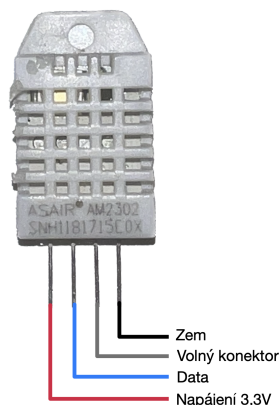
Obrázek 2.3. Interpretace průběhu signálu PWM. [18]

2.2 Výběr komponentů

Tuto část lze považovat za nekritičtější, protože nesprávný výběr komponentů by mohl limitovat budoucí funkcionalitu. Proto je vhodné si nejdříve určit vstupní a výstupní periferie, které splní požadavky pro sledování vybraných parametrů. Posléze zvolíme mikro počítač, který má dostatečné rozhraní a dokáže všechny shromažďované informace zpracovat.

2.2.1 Senzor pro měření teploty a vlhkosti

Senzorů pro měření teploty a vlhkosti je velké množství. Měření těchto veličin lze realizovat dvěma samostatnými senzory. Avšak existují řešení jako je senzor DHT22, který již splňuje oba typy měření. Toto řešení přináší řadu výhod. Hlavním přínosem tohoto senzoru je četné použití v celosvětovém měřítku a existuje přehršel množství návodů, knihoven pro různé programovací jazyky a je dobře dostupný. K porovnání se vybízí předcházející senzor DHT11, u kterého jsou měřicí parametry na úrovni, které jsou nedostačující pro tuto aplikaci.



Obrázek 2.4. Digitální senzor měření teploty a vlhkosti vzduchu DHT22

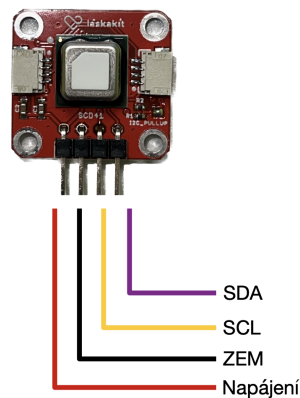
| parametr | typ/hodnota |
|-----------------|----------------------|
| periferie: | |
| napájení | 3,3-6V DC |
| výstupní signál | digitální |
| teploměr | NTC |
| vlhkoměr | polymérový kapacitor |
| měřící rozsahy: | |
| teploty | -40 až 80 °C |
| vlhkosti | 0 až 100 % |
| přesnot: | |
| teploty | ±0,5 °C |
| vlhkosti | max ±5% |
| snímací perioda | 2s |
| rozměry | |
| čelní strana | v: 20mm š: 15,1mm |
| zadní strana | v: 25,1mm š: 15,1mm |
| tloušťka | h: 5mm |

Tabulka 2.1. Parametry senzoru DHT22.

Hlavním důvodem, proč nezvolit DHT11 i přes značně nižší cenu, je přesnost měření teploty, která může být až ± 2 °C. Z hlediska teplotního komfortu pro člověka je to citelná odchylka. Porovná-li se s DHT22, který má uvedenu hodnotu chybovosti teploty v tabulce 2.1, lze konstatovat, že z hlediska samotného parametru přesnosti měření je senzor DHT22 značně výhodnější.

2.2.2 Senzor pro měření kvality vzduchu

Jelikož dostupnost senzorů pro měření CO_2 je v době návrhu řešení zadání nižší, je nutné tuto skutečnost zohlednit. Stejně tak je nezbytné vzít v úvahu podporu, kterou výrobce senzoru poskytuje. Výsledkem selekce je senzor od firmy Sensirio typ SCD41, který je rozšířen plošnou deskou Lásakait. Tím je docíleno co možná nejuniverzálnějšího použití pro jakýkoli aplikaci, protože má výstupy nejen pro nepájivé pole ale i pro typický uŠup.



Obrázek 2.5. Senzor měření kvality vzduchu CO_2 LaskaKit SCD41

| parametr | typ/hodnota |
|-----------------------------------|-----------------|
| periferie: | |
| napájení | 2,4-5,5V DC |
| sběrnice | I2C |
| měřící rozsahy: | |
| teploty | -10 až 60 °C |
| vlhkosti | 0 až 100 % |
| CO_2 | 40 - 5000 ppm |
| přesnot: | |
| pro rozsah teploty od 15 do 35 °C | |
| a | |
| vlhkosti od 20% do 65% | |
| teploty | ±0,8 °C |
| vlhkosti | max ±6% |
| CO_2 | ±40 ppm + 5% |
| odezva | 60s |
| rozměry: | |
| čelní strana | v: 21mm š: 21mm |
| tloušťka | h: 5mm |

Tabulka 2.2. Parametry senzoru SCD41.

Dle tabulky 2.2 parametrů senzoru stojí za povšimnutí jeho rozměry. Jedná se tak o dokonalou velikost pro případ vytváření modelu. Ze zkušenosti užívání SCD41 vyšlo najevo, že není vhodný pro měření teploty a vlhkosti. Proto, aby splňoval požadavky pro tuto konkrétní modelovou aplikaci, by musela být vytvořena empiricky digitální korekce. Za to však má jeden z nejlepších parametrů pro měření CO_2 v porovnání s konkurencí. Dalším nesporným přínosem je dostatečná podpora pro jakýkoli mikropočítač a to od Arduina Uno až po Raspberry Pi. Senzor pracuje na principu zmíněném v kapitole 1.2.3 a pomocí komunikační sběrnice I2C odesílá data do paměti, poté jsou pomocí ovladače zpracována tak, aby konečnému uživateli byla předložena v podobě, ze které bez větší námahy dokáže přečíst naměřené hodnoty.

2.2.3 Světelné zdroje

Zvýšení komfortu domácnosti lze provést dálkově ovládaným zdrojem světla nebo světelnou signalizací. Pro demonstraci byly vybrány světelné diody, v anglické literatuře označované Light-Emitting Diode též zkratkou LED, z důvodu kompaktní velikosti a možnosti jejich ovládní pro tento modelový případ.

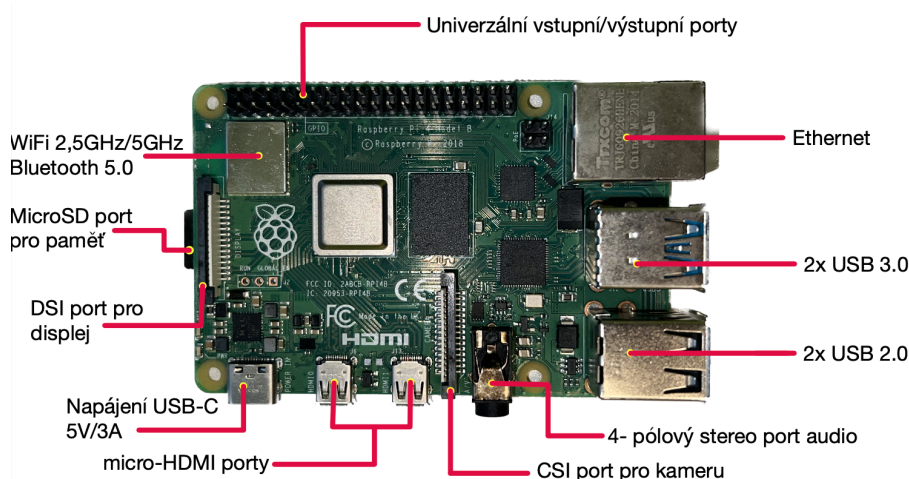
Aby uživatel nemusel neustále monitorovat stav koncentrace CO_2 byla vybrána dioda emitující světlo červené barvy, která bude signalizovat dosažení úrovně. Světelné zdroje osvětlující vymezený prostor bude zastupovat dioda emitující barvu žlutou.

2.2.4 Mikropočítač

Možností, jaký mikropočítač zvolit, je taktéž nepřeborné množství. Hlavními parametry a vlastnostmi, podle kterých je vhodné zvolit správný typ, je rozhraní portů. V předešlých podkapitolách byly zvoleny vstupní periferie, které vyžadují napájení a komunikační výstupy. Je také nanejvýše nutné počítat s možností rozšíření, které by v budoucí době mohlo nastat.

Od toho se odvíjí požadavky na parametry. Obvykle není překážkou počet digitálních portů jako spíše porty komunikační sběrnice jako je I2C a porty pro pulzně šířkovou modulaci. Toto technické řešení pro sériovou komunikaci vyžaduje dva datové kanály, které byly popsány v kapitole 2.1.1. Měl by splňovat potřebu na velikost paměti z důvodu možného velkého objemu dat a velikosti programu, který bude celý systém obsluhovat. Dalším velmi potřebným prvkem je možnost zprostředkování spojení s uživatelem. Aby jej mohl ovládat ovládacími prvky jako jsou například ovladač (joystick, tlačítko), dotyková obrazovka nebo sdílením obrazovky přes vzdálené připojení.

Všechny tyto potřebné parametry splňuje, ba i dokonce převyšuje, mikropočítač Raspberry Pi 4. S perspektivou budoucího rozšíření je to tak jedno z nejdostupnějších zařízení, které umožňuje širokou škálu aplikací. Nespornou výhodou tohoto zařízení je jeho kompaktnost a množství standardizovaných kontaktů. Je tak plně připraveno pro zapojení až dvou monitorů skrze micro-HDMI rozhraní, vstupních periférií jako je myš a klávesnice přes USB 2.0 či do počítačových sítí pomocí standardizovaného konektoru Ethernet RJ-45.



Obrázek 2.6. mikropočítač Raspberry Pi 4

Jedna z největších předností jsou bezdrátové technologie, již obsažené v rámci tohoto modelu zařízení, nevzniká tak nutnost jej rozšiřovat moduly s touto funkcí. Díky

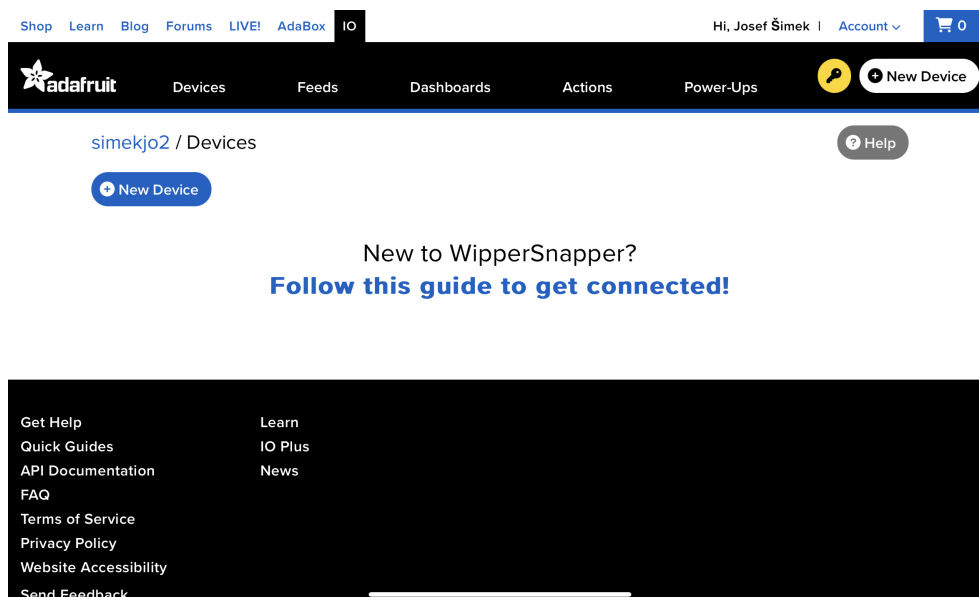
přítomnosti Wireless Fidelity (Wi-Fi) pro připojení do počítačové sítě se lze na mikropočítač vzdáleně připojovat či ho bezdrátově ovládat. Technologie Bluetooth umožňuje bezdrátově připojit například typické periferie jako je myš či klávesnice.

Dalším podstatným přínosem je možnost naimplementovat operační systém Raspberry Pi OS, dříve nazývaný Raspbian. Ten je založený na linuxovém jádru. Základní verze v sobě obsahuje vývojové prostředí Thonny Python IDE, pomocí něhož lze vyvíjet různé aplikace a programy. Operační systém je uložen na paměťové kartě typu MicroSD a je umístěna ve slotu pro paměťové karty dle popsaného obrázku 2.6.

Podle vznesených nároků na počet portů pro vstupní periferie je počet u tohoto mikropočítače dostatečný. Jde především o piny s označením GPIO, tedy vstupně-výstupně rozhraní pro všeobecné využití, jinak řečeno univerzální porty. V případě, že by bylo potřeba napájet další zařízení z těchto pinů, umožňuje Raspberry pi 4 proudový odběr až 50 mA. Takto velký odběr je vhodný například pro malé elektromotorky nebo napájení displejů.

2.3 Výběr cloudové služby pro provoz webové aplikace

Při výběru se zohlednila především dostupnost, použitelnost a podpora pro případné budoucí potíže. Volba padla na Adafruit IO, dostupná na [19]. Tato internetová služba na níž je možné využívat webovou aplikaci umožňuje přijímat zprávy MQTT protokolu a zpracovat je. Disponuje možností vytvořit uživatelské rozhraní, které se nachází v nabídce „Dashboard“ a komunikační kanály „Feeds“, přes které se budou zprostředkovávat zprávy. Pro provoz této služby byly vytvořeny knihovny pro více programovacích jazyků a platform. S jejich využitím se lze na server připojovat, odpojovat a odesílat či přijímat informace.

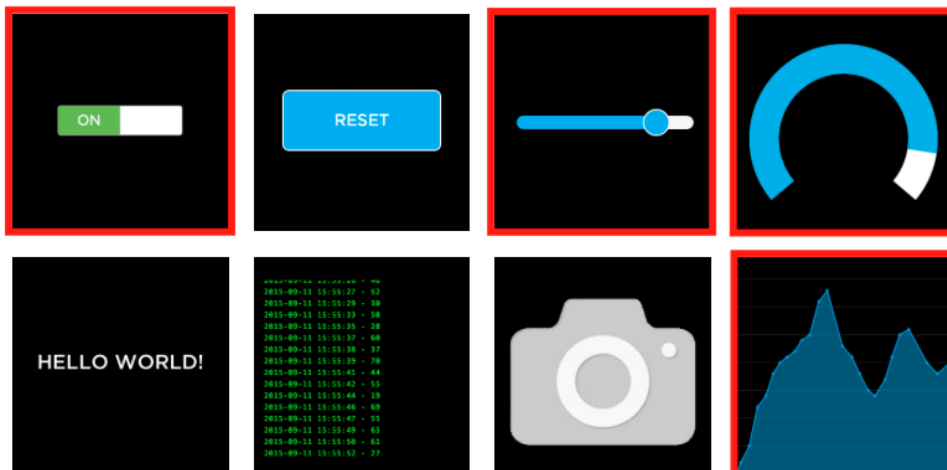


Obrázek 2.7. Prostředí cloudové služby a webové aplikace Adafruit IO

Create a new block



Click on the block you would like to add to your dashboard. You can always come back and switch the block type later if you change your mind.



Obrázek 2.8. Výběr funkčních bloků webové aplikace Adafruit IO

Grafické prostředí pro interakci s uživatelem lze vytvořit z funkčních bloků, které mají různé vlastnosti. Na obrázku 2.8 jsou zvýrazněné bloky, které se jeví jako nejvhodnější, protože umožňují zobrazení naměřených hodnot, jejich průběh a nastavování parametrů nebo stavů.

2.4 Dispozice pozorovaného prostředí

Jednou z otázek, která by mohla nastat při realizaci řešení, může být, kam umístit celý aparát. Podle této idey bude nakreslen a zkonstruován model. Skládá se z dvou částí. První část je prostor, ve kterém budou umístěny senzory a případné výstupní periferie. Druhá část obsahuje mikropočítač s nepájivým polem, do kterého bude přivedeno vzájemné propojení.

Dispozice se tak přibližuje bytové jednotce koleje Strahov, která obsahuje především jeden hlavní obývaný pokoj. Je to jeden z nejrozsáhlejších a nejčastějších pokojů, které jsou obývané studenty ČVUT v Praze. Jejich typické rozměry jsou $13m^2$. Pro budoucí měření je to vhodné místo pro měření a jeho vyhodnocení. Obvykle jej obývají dva členové, v některých případech jeden. Přednostně jsou užívány jako místo, kde obývací přespávají. K tomuto faktu je možné nahlížet tak, že pozorovatel nemůže například vědomě nakládat se situací týkající obsahu oxidu uhličitého a zasahovat, či snad předcházet jeho vysokým hodnotám.

Kapitola 3

Realizace

V této pasáži práce se nachází postup podle kterého byl zhotoven systém splňující zadání. Vychází se z rozboru řešení, ve kterém byly vybrány komponenty. V úvodu kapitoly se nachází zprovoznění mikropočítače s operačním systémem Raspberry Pi OS a aktivace připojení na vzdálenou plochu. Posléze zprovoznění senzorů s implementací do hlavního obslužného programu. V závěru kapitoly je vyhotoven postup odesílání naměřených hodnot na server a zobrazení v průběhu času.

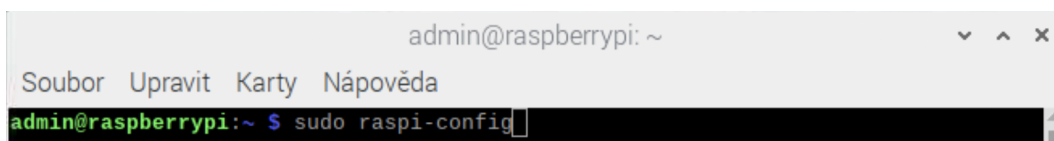
3.1 Zprovoznění mikropočítače

Základem je instalace operačního systému. Ten se pomocí jiného zařízení nainstaluje na SD kartu, která bude zároveň i pamětí mikropočítače. Po vložení do mikropočítače a připojení základních periferií jako je monitor, myš a klávesnice se může spustit. Po zapojení napájecího zdroje se začne načítat operační systém, po jehož dokončení nás uvítá počítačové prostředí. V tomto stavu můžeme začít zařízení obsluhovat a připravovat na budoucí aplikaci.

3.1.1 Vzdálený přístup

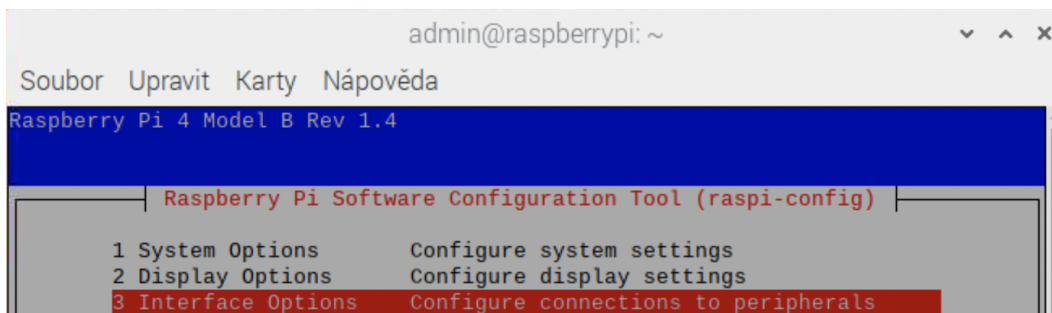
Aby mohla obsluha mikropočítač ovládat nebo analyzovat případné poruchy bez vybavení jako je monitor, myš a klávesnice, zprovozníme sdílení plochy. Na zařízení se budeme připojovat v rámci sítě LAN. Na síť se připojíme nabídkou v horní liště, kde se v pravé části nachází nastavení připojení sítí. Kliknutím na námi zvolenou síť se zařízení připojí.

Ve výchozím stavu není připojení pro sdílení obrazovky povoleno. Povolení se provede podle následujícího postupu. Příkazem `sudo raspi-config` v terminálu se otevře nabídka nastavení rozhraní. V podnabídce `interfacing options` je možné povolit Virtual Network Computing zkratka VNC. Změníme stav na `Enable`, čímž možnost rozhraní povolíme.

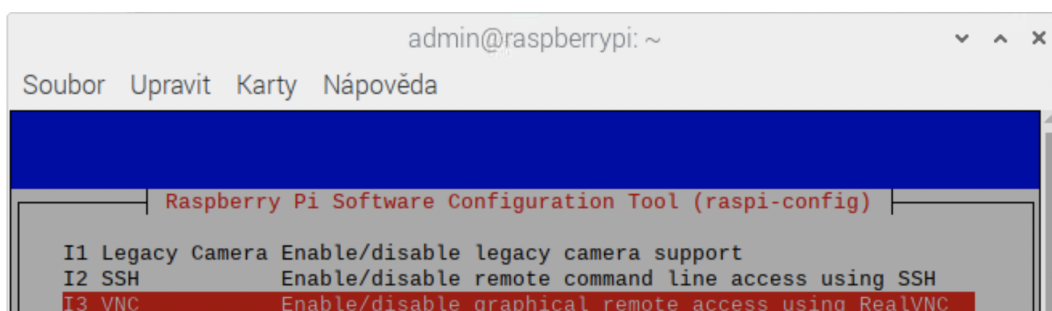


```
admin@raspberrypi: ~
Soubor Upravit Karty Nápověda
admin@raspberrypi:~ $ sudo raspi-config
```

Obrázek 3.1. Příkaz k přístupu konfigurace zařízení

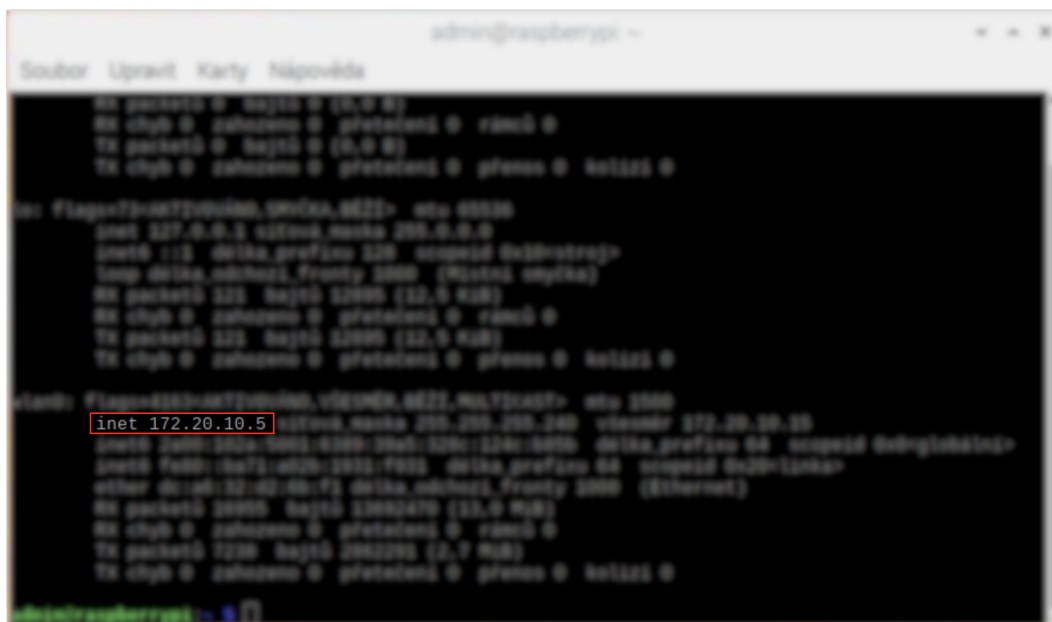


Obrázek 3.2. Nabídka konfiguračního nastavení



Obrázek 3.3. Nabídka nastavení komunikačního rozhraní

Pro vzdálené připojení v LAN síti je nezbytné získat IP adresu v místní síti. Ta se získá opět v terminálu pomocí příkazu `ifconfig`. Následně se vypíše veškeré informace o síťových rozhraních. IP adresu získáme u položky `inet`, za kterou je vypsána.



Obrázek 3.4. Zobrazení IP adresy

Na sekundárním počítači nebo chytrém zařízení se z obchodu, který umožňuje získávání aplikací, stáhne aplikace VNC viewer, který je dostupný na nejrozšířenějších operačních systémech typu Android a iOS. Tímto řešením je zajištěno, že bez ohledu na operační systém zařízení, se obsluha může v případě potřeby připojit a začít zařízení ovládat.

Po ujištění, že jsou zařízení připojena na stejné místní síti se přistoupí ke vzdálenému připojení. Po otevření aplikace se přidá zařízení, na které se chceme připojit. Po zadání IP adresy a pojmenování je možné se připojit tlačítkem **Connect**. Že se připojení podařilo je potvrzeno hláškou u části panelu, kde se nachází čas.

Poté je možné jakkoli ovládat mikropočítač bez potřeby základních periférií. Veškerá obsluha je možná i pomocí aplikace pro chytré telefony nebo tablety.

Pro odpojení spojení stačí aplikaci ukončit nebo se pomocí tlačítka v nabídce **Disconnect** odpojit. Po odpojení celý systém funguje nezávisle bez přerušení nadále.

3.2 Programování a funkce senzorů

Pro dosažení správné funkce senzorů udávané výrobcem je nutné postupovat podle dokumentace. Ta je součástí dodávky nebo je možnost stažení z internetových stránek dodavatele. Protože nedílnou součástí celého systému je program, tak sekundární ne méně podstatnou částí jsou knihovny s obslužnými instrukcemi, které zprostředkovávají provoz a ovládání senzorů. Zpravidla jsou dostupná pro více typů programovacích jazyků od různých tvůrců a jsou přístupné i z open-source.

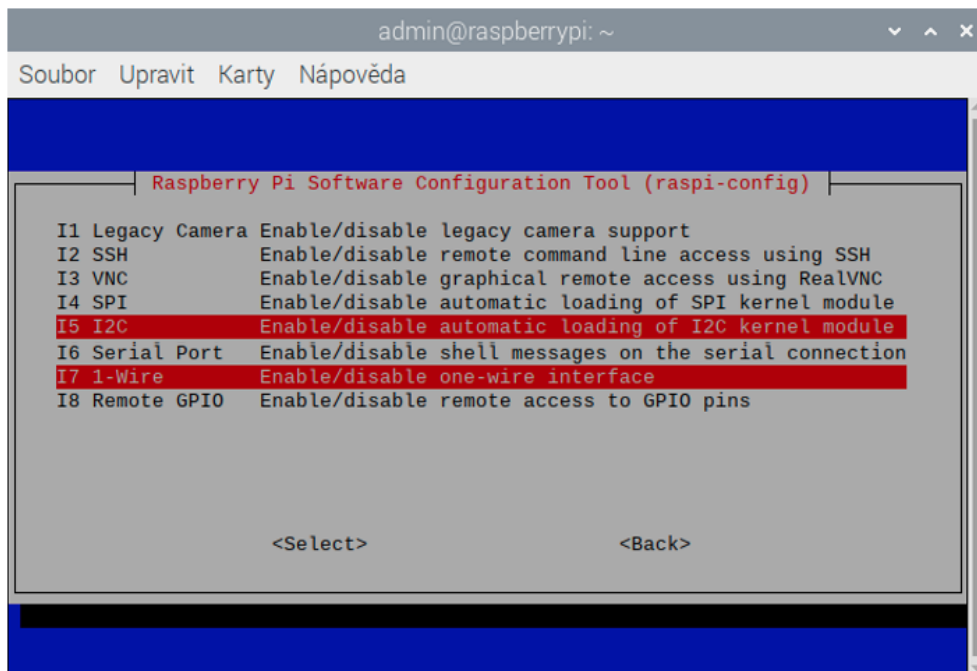
Proto by mělo být už předem rozhodnuto, jaký programovací jazyk se bude využívat. Právě jeden z faktorů, který má na výběr největší vliv je dostupnost knihoven. V tomto případě, ve kterém je více senzorů, navazujících spojení pomocí různých komunikačních sběrnic a protokolů, je o to podstatnější. Po prozkoumání nabídky jsme se rozhodli pro nejvíce vhodnou variantu programovacího jazyka s názvem Python.

Při výběru tvůrce knihoven byla učiněna volba pro společnost Adafruit jako prioritní zdroj pro tvorbu systému. Důvodem je velmi dobrá spolupráce s výrobcem zařízení, součástek a ověřené kvality volně šiřitelných knihoven, které jsou téměř zpravidla bezplatné a připravené k použití bez nutnosti úprav.

Programovací jazyk Python není potřeba instalovat, je součástí nainstalovaného operačního systému. Je však více než vhodné jej aktualizovat. Tak lze zabezpečit kompatibilitu knihoven a vyloučit tak možné nežádoucí jevy při spouštění programů a v krajním případě i absolutní nefunkčnost. V tomto případě byla použita verze Python 3.9.2.

K přípravě programu lze přistoupit uložením a dokončením instalace veškerých knihoven se souvisejícími soubory umožňujícími operace se snímačem.

Jak bylo zmíněno v podkapitolách 2.1.2 a 2.1.1 budeme pro komunikaci se senzory využívat protokoly 1-wire a I2C. Jejich rozhraní ve výchozím stavu není taktéž povoleno. Pro jejich povolení opět spustíme nabídku nastavení komunikačního rozhraní, obdobně jako v popsané podkapitole 3.1.1. Nyní změnou stavu **Enable** na řádce 5 s označením I2C a na řádce 7 s označením **1-wire** povolíme rozhraní, čím se nám umožní jimi komunikovat.



Obrázek 3.5. Povolení komunikovat rozhraním I2C a 1-Wire

3.2.1 Senzor DHT22

Nahráním knihovny `import Adafruit_DHT`, která je veřejně dostupná z repozitáře [20], do paměti programu umožníme programu využívat funkce pro obsluhu senzoru. Následnou inicializací proměnné `sensor` specifikujeme o jaký senzor se jedná, protože stejná knihovna může obsluhovat jak senzor DHT22 tak i předešlý typ DHT11. Poté určíme na jakém pinu bude připojen datový vodič, přes který budeme komunikovat se senzorem pomocí inicializace proměnné `pin`.

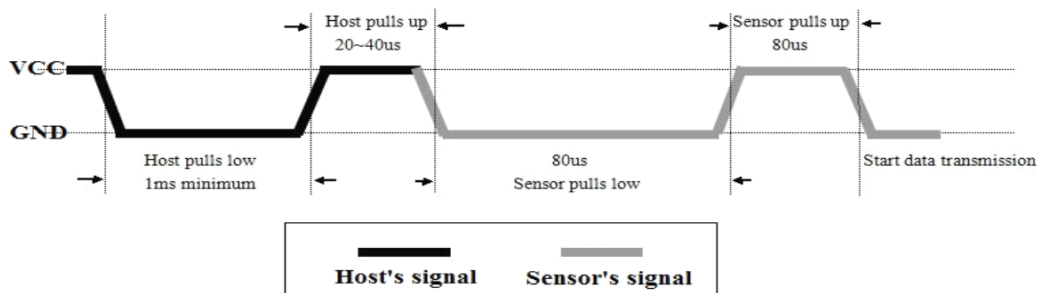
```

9 import Adafruit_DHT
10 sensor = Adafruit_DHT.DHT22
11 pin = 4

```

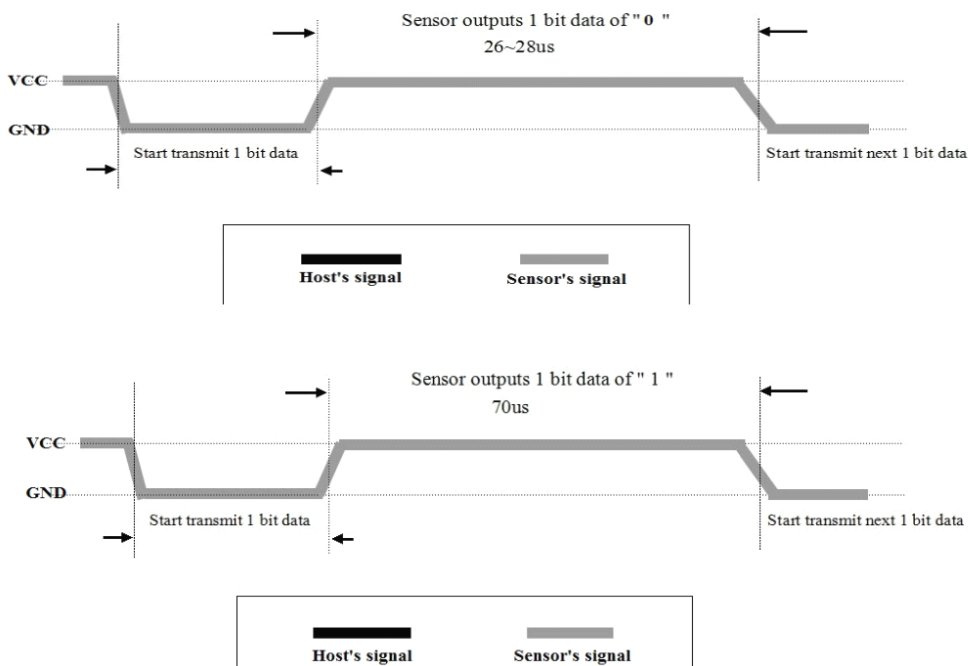
Obrázek 3.6. Nahrání knihovny Adafruit pro DHT22, senzor teploty a vlhkosti

Pokus o měření zahájíme zavoláním funkce pro opakované čtení z knihovny `adafruit` ve tvaru `Adafruit_DHT.read_retry(sensor, pin)`. Spustí se tak komunikace, jejíž začátkem je pulz LOW vytvořený mikropočítačem, který musí trvat nejméně 10 ms. Poté následuje pulz HIGH. Sensor signalizuje detekci startovního signálu a odpoví na něj LOW po dobu 80 us. Signalizace o budoucím přenosu dat se uskuteční po poklesu druhého 80 us pulzu HIGH.



Obrázek 3.7. První komunikace se senzorem DHT22 [21]

V ten moment mikropočítač očekává data o velikosti 40 bitů. Bity jsou kódovány senzorem. Stav bitu o hodnotě logické 1 představuje signál o trvání 70 us a o logické 0 30 us, v obou případech se sleduje stav HIGH. V prvním řetězci 16 bitů je zakódovaná vlhkost a v druhém teplota, posledních 8 bitů je kontrolních, ve kterých je součet všech 32 bitů po 8 bitech. Jestliže se kontrolní součet, který odeslal senzor, shoduje se součtem mikropočítače, tak jsou přijatá data za platná.



Obrázek 3.8. Přijímání signálu se zakódovanými naměřenými hodnotami [21]

Knihovna je posléze dekóduje a funkce `Adafruit_DHT.read_retry(sensor, pin)` vrací návratové hodnoty, které uloží do proměnných `humidity` a `temperature`. Jejich hodnoty jsou poté uloženy do lokálních proměnných pro aplikaci v dalších krocích.

```

121 humidity, temperature = Adafruit_DHT.read_retry(sensor, pin)
122 if humidity is not None and temperature is not None:
123     t = float(temperature)
124     h = float(humidity)

```

Obrázek 3.9. Příkaz pro obsluhu měření teploty a vlhkosti

3.2.2 Senzor SCD41

Pro kontrolu, že je senzor adekvátně připojen a že ho mikropočítač detekuje, je možné zadat do příkazového řádku konzole `i2cdetect -y 1`, ten zobrazí adresy zařízení připojených ke sběrnici. V tabulce se na řádku 6 a ve sloupci 2 vypíše číslo 62, které indikuje připojené zařízení s adresou 0x62.

```

admin@raspberrypi: ~
Soubor Upravit Karty Nápověda
admin@raspberrypi:~$ i2cdetect -y 1
 0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
10:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
20:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
30:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
40:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
50:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
60:  --  --  62  --  --  --  --  --  --  --  --  --  --  --  --
70:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
admin@raspberrypi:~$

```

Obrázek 3.10. Výpis aktivních adres na I2C sběrnici

Prvním krokem nahrajeme potřebné knihovny pro obsluhu senzoru z paměti mikro-počítače do paměti programu a to knihovny `board` a `adafruit_scd4x` příkazem `import`.

Knihovna `board` poskytuje informace o hardwarových pinech, na kterých mohou být připojeny periferie, a umožní s nimi operovat.

Následující knihovna `adafruit_scd4x`, veřejně dostupná z repozitáře [22], umožňuje relizovat interakce se senzorem. Obsahuje v sobě metody, instance tříd a definované příkazy obsahující konstanty, podle kterých jednotka senzoru rozpozná, jakou akci má provést.

Proto, aby byl program informován, že se budou využívat piny I2C sběrnice, je posléze vytvořen objekt `i2c = board.I2C()`. Ten určuje použití výchozích pinů pro komunikaci. Výchozí piny pro komunikaci s I2C zařízeními jsou v případě Raspberry Pi 4 GPIO 3 (SDA) a 5(SCL). Na těchto pinech se očekává připojené zařízení taktéž se sběrnicí I2C, v tomto případě to bude zvolený senzor SCD41 s adresou 0x62.

```

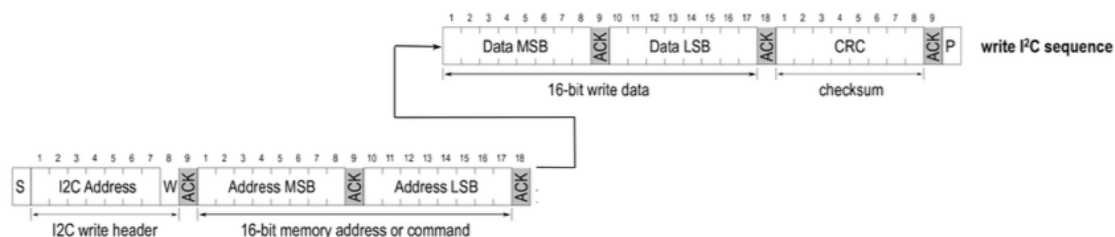
13 import board
14 import adafruit_scd4x
15 i2c = board.I2C()

```

Obrázek 3.11. Nahrání knihovny Adafruit pro SCD41, senzor CO_2

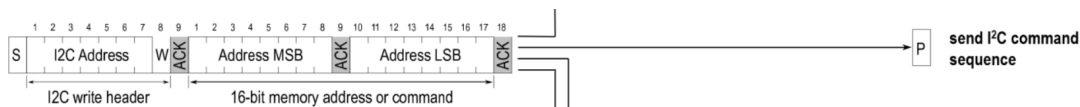
Před spuštěním měření se inicializuje objekt `scd4x`, tím se vytvoří instance třídy `SCD4X` z knihovny a spustí se proces nastavení počátečních hodnot. Argumentem `i2c` se nastaví parametry pro komunikaci se sběrnicí, jako je například adresa senzoru. Nyní je připraven na přijímání a zpracování příkazů.

Z výroby má senzor ve své paměti uloženou hladinu na úrovni 0 metrů nad mořem. Aby se zvýšila přesnost měření, uložilo se do instance proměnné třídy pomocí `scd4x.altitude`. Je vyslán hexadecimální kód 0x2427, kterým se zapíše data obsahující informaci o nadmořské výšce, na 0x62 následovaný hodnotou přiřazené hodnoty, která byla do proměnné uložena. Senzor se nachází ve výšce 330 m.



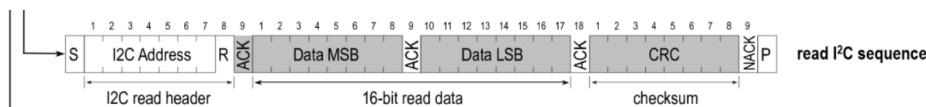
Obrázek 3.12. Příkaz pro zápis hodnoty obsahující nadmořskou výšku [23]

Základním příkazem `start_periodic_measurement()` z inicializované knihovny odešleme hexadecimální kód `0x21b1` na adresu senzoru. Senzor tak spustí měření s intervalem 5 sekund.



Obrázek 3.13. Komunikační sekvence příkazu `start_measurement` pro I2C sběrnici na adresu senzoru SCD41 [23]

Poté následuje podmínka `while not scd4x.data_ready`, která běží, dokud se z instance třídy knihovny `data_ready` nevrátí hodnota `True`, vypovídající o úspěšném naměření hodnoty. Hodnoty, které byly naměřeny jsou uloženy ve vyrovnávací paměti senzoru. K přečtení dojde zavoláním instance třídy pro danou veličinu. Pro získání naměřené hodnoty CO_2 se zavolá instance z knihovny `scd4x.CO2`, ta v sobě obsahuje volání metody `_read_data()`, která o hexadecimální konstantě `0xec05` odešle příkaz ke čtení. Z vyrovnávací paměti je prvních 9 bajtů dat vyjmuta interní metodou `read_reply` a uloženo do lokální proměnné `self_co2`.



Obrázek 3.14. Komunikační sekvence čtení dat z adresy senzoru SCD41 [23]

Po této akci se na adresu senzoru vyšle příkaz k ukončení měření `stop_periodic_measurement()`, který odešle příkaz hexadecimálním kódem o hodnotě `0x3F86` a senzor přestane pracovat.

Návratová hodnota z instance třídy plynu CO_2 se uloží do lokální proměnné `CO2`, ve které bude setrvávat na další zpracování.

```

116     scd4x = adafruit_scd4x.SCD4X(i2c)
117     scd4x.altitude = 330
118     scd4x.start_periodic_measurement()
119     while not scd4x.data_ready:
120         pass
121     CO2 = int(scd4x.CO2)

```

Obrázek 3.15. Obsluha senzoru LaskaKit SCD41

Přepočtení hodnoty z hexadecimální soustavy na decimální není potřeba programovat, protože programovacímu jazyku Python umožňuje převod vestavěná funkce `int()`. Do argumentu se vloží hexadecimální hodnota, v tomto případě hodnota naměřeného CO_2 a návratovou hodnotou bude převedené číslo v desítkové soustavě

3.3 Světelná signalizace a ovládání osvětlení

LED budou napájena a ovládána piny, které jsou součástí zařízení. Vestavěnou knihovnou `RPi.GPIO` inicializujeme porty dostupné na mikropočítači a jejich nastavením `GPIO.setup(x, GPIO.OUT)` upřesníme, že je budeme využívat jako výstupní. Vytvořením objektu `led_pwm = GPIO.PWM(13,100)` umožníme funkci nastavování hodnot pulzně šířkové modulace.

```

17 import RPi.GPIO as GPIO
18 GPIO.setwarnings(False)
19 GPIO.setmode(GPIO.BCM)
20 GPIO.setup(13, GPIO.OUT)
21 GPIO.output(13, GPIO.LOW)
22 led_pwm = GPIO.PWM(13,100)
23 led_pwm.start(0)
24
25 GPIO.setup(21, GPIO.OUT)
26 GPIO.output(21, GPIO.LOW)
27 state = GPIO.input(21)

```

Obrázek 3.16. Nastavení výstupních pinů

Blikání při dosažení nastaveného limitu CO_2 obsluhuje metoda `alertCO2`. Ta v sobě obsahuje zanamování času v době, kdy byly předešlé podmínky splněny. V případě, že rozdíl času mezi minulým a současným splněním podmínek dosáhne nebo přesáhne předem definované hodnoty, změní se aktuální stav výstupního pinu na opačný, například tedy z LOW na HIGH.

```

81 def alertCO2():
82     global switch
83     global CO2
84     while True:
85         if switch == "Zap" and CO2 >= set_alert:
86             global previous_blink
87             now_blink = time.monotonic()
88             if now_blink - previous_blink >= wait:
89                 previous_blink = now_blink
90                 state = GPIO.input(21)
91                 if state:
92                     GPIO.output(21, GPIO.LOW)
93                 else:
94                     GPIO.output(21, GPIO.HIGH)

```

Obrázek 3.17. Funkce pro blikání LED CO_2

Získáním hodnoty z webové aplikace určujeme hodnotu pulzně šířkové modulace.

```

66 if(feed_id == 'smart-home.led-pwm'):
67     print("nastavení pwm", payload)
68     led_pwm.ChangeDutyCycle(int(payload))

```

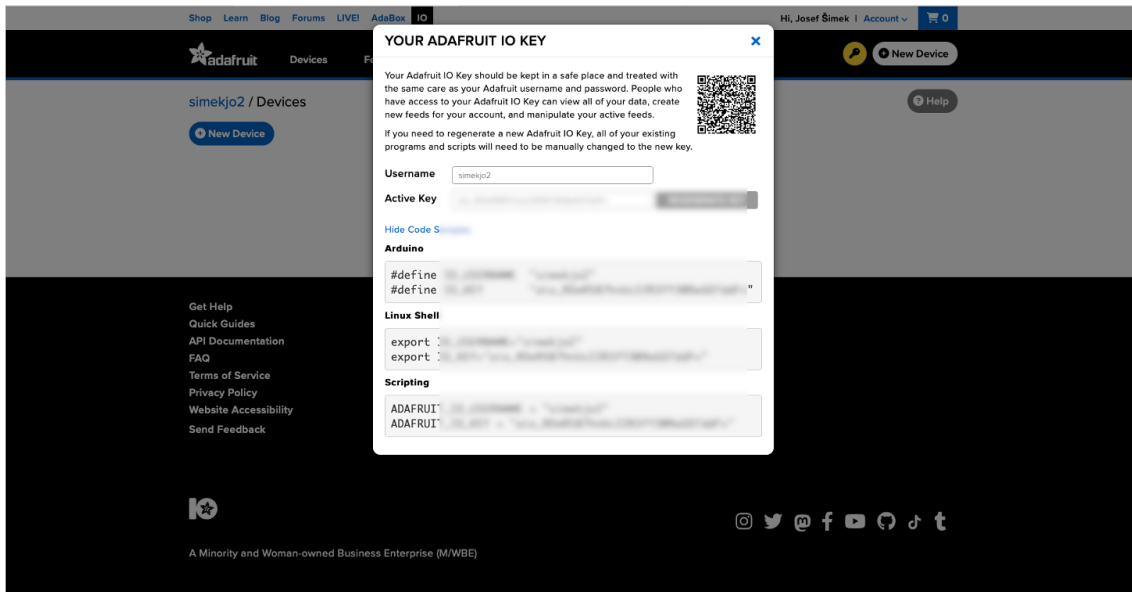
Obrázek 3.18. Nastavení hodnoty PWM pro LED z daného feedu CO_2

3.4 Obslužný program

3.4.1 Spojení s webovou aplikací

Jak již bylo zmíněno v podkapitole 2.3, tvůrce prostředí poskytuje knihovny, které stáhneme a nainstalujeme. Součástí souboru knihovny, dostupného na [19], je i ukázkový kód, který obsahuje názornou aplikaci. Ten lze použít jako platformu k následnému upravení nebo rozšíření o naše funkce. Pro účel této realizace je nejvhodnější příklad s názvem `mqtt_client_class.py`, přístupný na [24].

V prvním kroku založíme účet na Adafruit IO. Klíč a uživatelské jméno, to jsou dva parametry umožňující navázání spojení s cloudovou službou a následným posláním zpráv, získáme po kliknutí na žlutou ikonu s klíčem.



Obrázek 3.19. Získání klíče pro navázání spojení s Adafruit IO

Ty vložíme do předem připravených proměnných pro textové řetězce, ty se po spuštění programu inicializují a budou tvořit argumenty instance `mqtt_client.py`, který bude následně spuštěn na pozadí.

```

15 ADAFRUIT_IO_KEY = 'YOUR_AIO_KEY'
16
17 # Set to your Adafruit IO username.
18 # (go to https://accounts.adafruit.com to find your username)
19 ADAFRUIT_IO_USERNAME = 'YOUR_AIO_USERNAME'
```

Obrázek 3.20. Zadání přihlašovacích parametrů do proměnných

```

100 smarth = MQTTClient(USERNAME, KEY)
```

Obrázek 3.21. Implementace přihlašovacích údajů

3.4.2 Základní funkce

Následují funkce `connected`, `disconnected` a `message`. Ty se budou následně volat dle potřeby.

Funkce `connected` se volá pro potřebu připojení k Adafruit IO, a vyžádá se z daného kanálu (Feed) instancí `.subscribe` jeho poslední informace.

Funkce `Disconnected` bude zavolána pro potřebu odpojení a příkazem `sys.exit(1)` se běh programu ukončí.

Zajištění přehledu provozu komunikace se může starat funkce `message`. Ta vytiskne zprávu v případě získání nové hodnoty odebíraného feedu.


```

45 def connected(smarth):
46     print('Připojeno')
47     # Odebrání dat v daném feedu
48     smarth.subscribe('smart-home.co2-alert')
49     smarth.subscribe('smart-home.led-pwm')
50     smarth.subscribe('smart-home.zap-slash-vyp-led')
51     smarth.subscribe('smart-home.co2')
52     smarth.subscribe('smart-home.temperature')
53     smarth.subscribe('smart-home.humidity')
54
55 def disconnected(smarth):
56     # Ukončení programu při selhání spojení
57     print('Selhání spojení!')
58     sys.exit(1)
59
60 def message(smarth, feed id, payload):

```

Obrázek 3.22. Definování základních funkcí funkce

3.4.3 Odesílání hodnot

Naměřené hodnoty uložené v proměnných instancí třídy `.publish` z knihovny odešleme do daného kanálu webové aplikace. Na cloudové službě budou následně zpracovány.

```

126     smarth.publish('smart-home.co2', C02)
127
128     smarth.publish('smart-home.temperature', round(t, 1))
129
130     smarth.publish('smart-home.humidity', round(h, 1))

```

Obrázek 3.23. Příkazy k odeslání naměřených hodnot

3.5 Příprava webové aplikace

Na serveru Adafruit IO vytvoříme skupinu. Ta bude obsahovat feedy, které společně budou sloužit našim účelům. Po následném vytvoření se pod kolonkou Key zobrazí jejich unikátní klíč, jenž bude jejich identifikačním prvkem.

| Feed Name | Key | Last value | Recorded |
|--|----------------------------|------------|---------------|
| <input type="checkbox"/> CO2 alert | smart-home.co2-alert | 400 | 7 minutes ago |
| <input type="checkbox"/> CO2 ppm | smart-home.co2 | 439 | 1 minute ago |
| <input type="checkbox"/> PWM LED | smart-home.led-pwm | 100 | 8 minutes ago |
| <input type="checkbox"/> Relativní vlhkost % | smart-home.humidity | 49.9 | 1 minute ago |
| <input type="checkbox"/> Teplota °C | smart-home.temperature | 24.9 | 1 minute ago |
| <input type="checkbox"/> ZAP/VYP LED | smart-home.zap-slash-vy... | Vyp | 6 minutes ago |

Obrázek 3.24. Seznam feedů s jejich identifikačním klíčem a hodnotou

Následně přejdeme do kategorie **Dashboards**, ve které budeme přidávat bloky. Bloky s jednotlivým feedem propojíme hned v prvním kroku tvorby. Po zaškrtnutí daného odběru dat z kanálu se nám nabídne nastavení daného bloku, kterou dané blok specifikujeme.

Můžeme zde nastavit parametry jako jsou například: maximální a minimální hodnoty osy Y, jméno osy X nebo název konkrétního bloku.

Block settings ✕

In this final step, you can give your block a title and see a preview of how it will look. Customize the look and feel of your block with the remaining settings. When you are ready, click the "Create Block" button to send it to your dashboard.

Block Title (optional)

Show History

X-Axis Label

Y-Axis Label

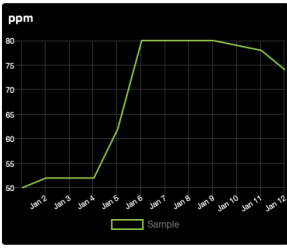
Y-Axis Minimum

Leave blank to automatically detect.

Y-Axis Maximum

Leave blank to automatically detect.

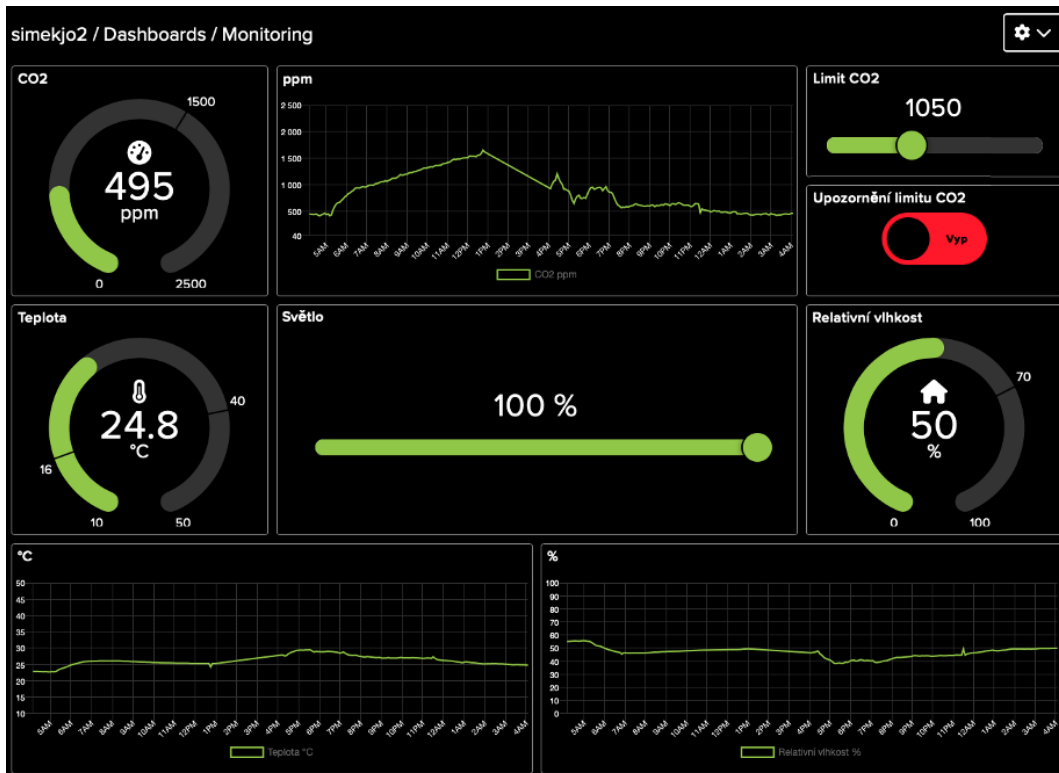
Block Preview



Line Chart The line chart is used to graph one or more feeds.

Obrázek 3.25. Nastavení parametrů daného bloku

Výsledkem je tak aplikace s uživatelským prostředím pro interakci s uživatelem, které je přístupné pro jakékoli zařízení, které se umí připojit na internet a spustit webový prohlížeč. Uživatel zde může ovládat určité prvky a zobrazovat naměřené hodnoty včetně historie.



Obrázek 3.26. Uživatelské prostředí webové aplikace

Kapitola 4

Testování

Tato část práce obsahuje zkoušku ovládacích prvků, proces měření a příklady naměřených hodnot dle uvedené situace. Měření probíhalo v podmínkách, kterými se pokusilo přiblížit co nejvíce běžnému chodu pokoje.

4.1 Provoz komunikace s cloudovou službou

4.1.1 Naměřené hodnoty

Jak bylo zmíněno v podkapitole 3.4.2, je možné odebírat nové zaznamenané hodnoty v daném datovém kanálu do mikropočítače. Přijaté zprávy lze následně vytisknout do konzole funkcí `print()`, která je vytiskne. Pomocí tohoto principu jsme ověřili, že přiřazený datový komunikační kanál získal naměřenou hodnotu

```
Naměřená a odeslaná hodnota CO2: 1473 ppm
Naměřená a odeslaná hodnota teploty: 26.80 °C
Naměřená a odeslaná hodnota relativní vlhkosti: 51.40 %
2023-05-22 13:46:06.103160
Feed smart-home.co2 zaznamenal novou hodnotu: 1473
2023-05-22 13:46:06.279490
Feed smart-home.temperature zaznamenal novou hodnotu: 26.8
2023-05-22 13:46:06.362302
Feed smart-home.humidity zaznamenal novou hodnotu: 51.4
```

Obrázek 4.1. Vytisknutí hodnot přijaté zvoleným komunikačním kanálem

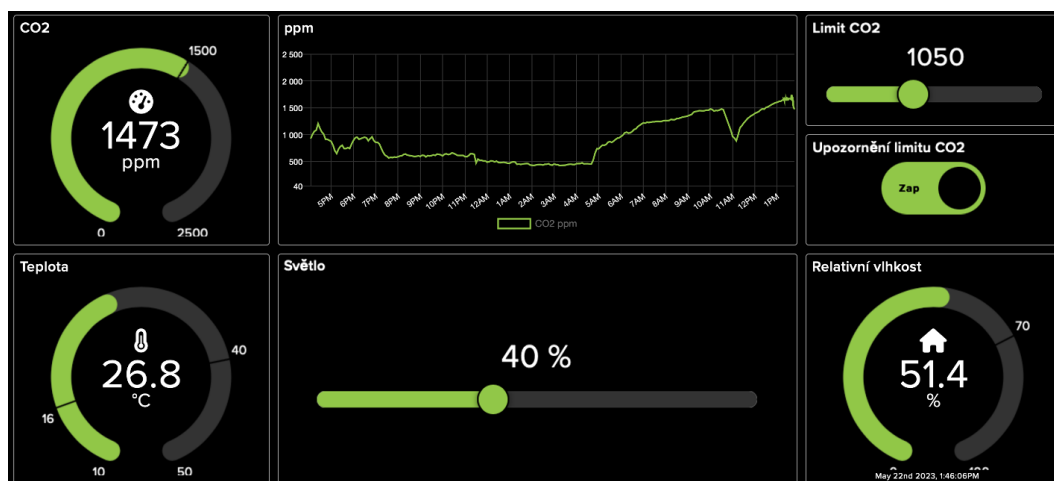
V obrázku 4.1 můžeme vidět, že po změření a odeslání zprávy, byla zaznamenaná nová hodnota.

4.1.2 Ovládací prvky

Obdobně jako v podkapitole 4.1.1 se využilo funkce tisknutí. Posuvníkem byla nastavena hodnota na upozornění limitu CO2 na hodnotu 1050 ppm, vypínačem byl změněn stav na Zap pro povolení indikace přesažení limitu světelným zdrojem LED a nastavením parametru posuvníku LED změni hodnota pulzně šířkové modulace na výstupu.

```
2023-05-22 12:31:27.267872
Feed smart-home.co2-alert zaznamenal novou hodnotu: 1050
2023-05-22 12:31:27.710144
Feed smart-home.zap-slash-vyp-led zaznamenal novou hodnotu: Zap
2023-05-22 12:31:28.511002
Feed smart-home.led-pwm zaznamenal novou hodnotu: 40
```

Obrázek 4.2. Konzole obsahující přijmutí zprávy s novými nastavenými hodnotami

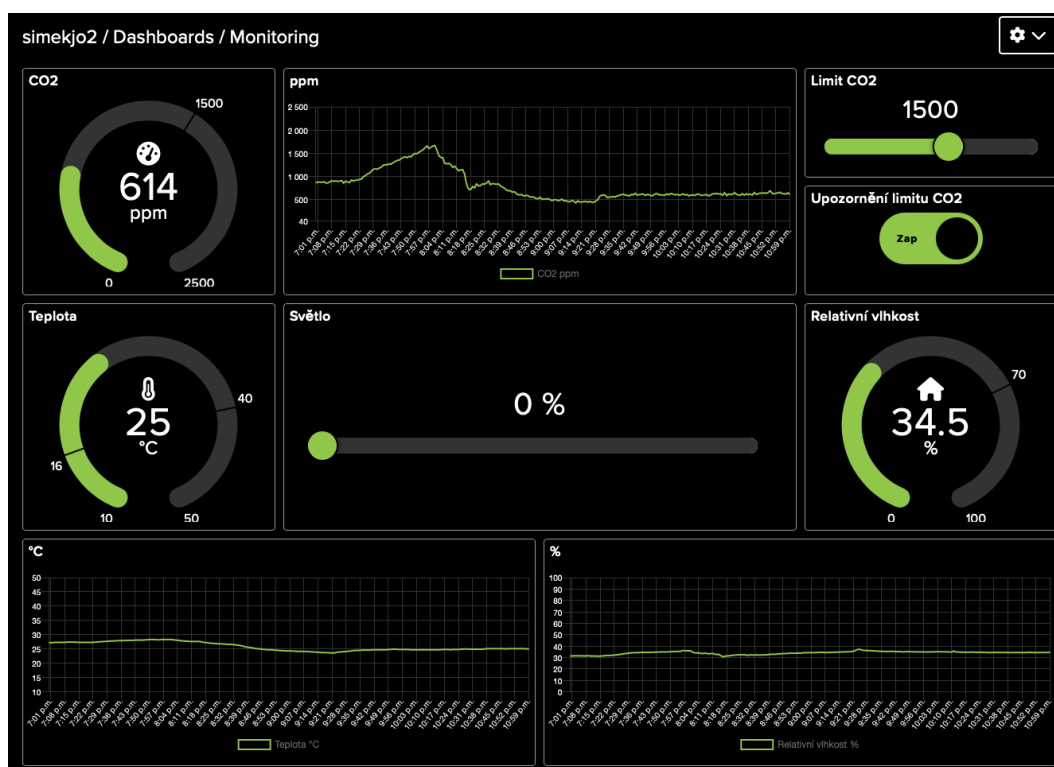


Obrázek 4.3. Interpretace naměřených hodnot

Na obrázku 4.3 je webová aplikace, která poskytuje v kruhovém měřítku informace o posledních naměřených hodnotách, které se shodují s odeslanými v obrázku 4.1. Taktéž je zde nastavenými hodnotami

4.2 Naměřené hodnoty a jejich průběh

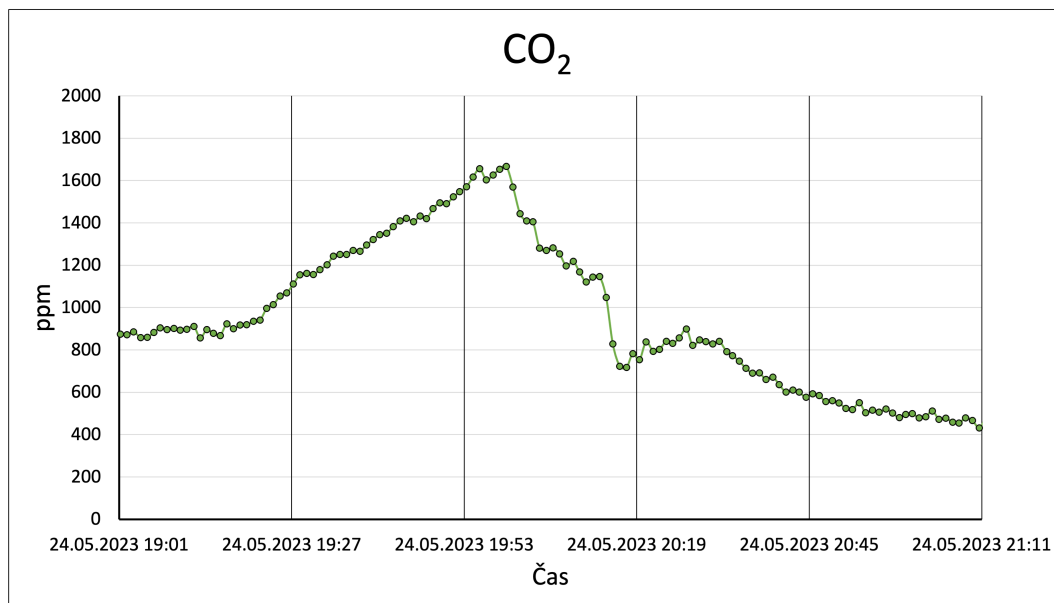
Z Adafruit IO byl stažen soubor obsahující naměřené hodnoty, zobrazené na snímku obrazovky. Ten je následně zpracován v tabulkovém programu do grafů.



Obrázek 4.4. Průběh vybraných naměřených hodnot CO_2

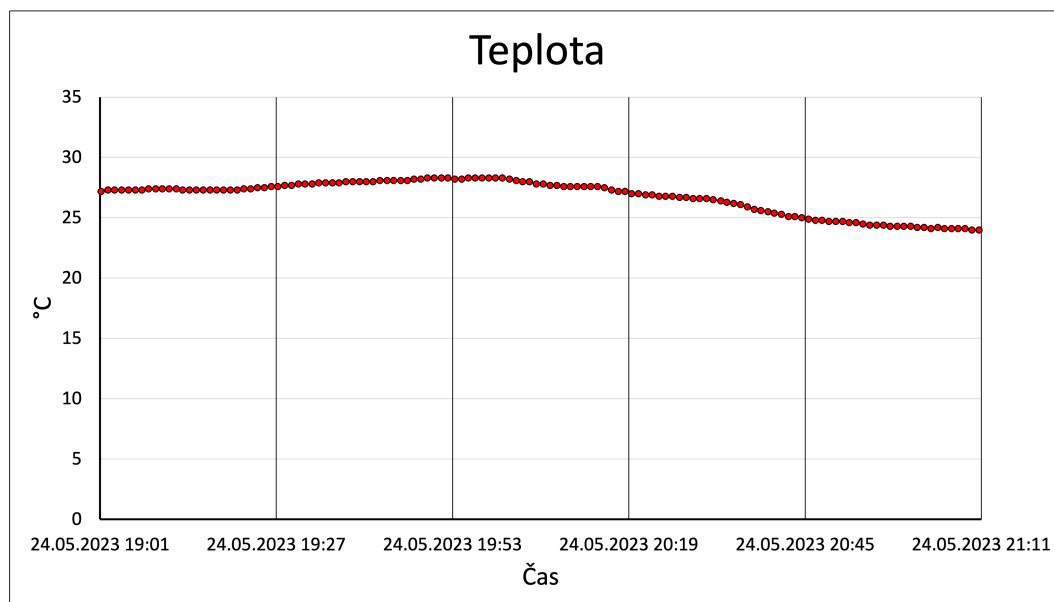
Na obrázku 4.4 si lze všimnout snímku z webové aplikace, která zaznamenává naměřené hodnoty. Nejvyšší naměřená hodnota CO_2 přesahuje 1600 ppm, teploty 28 °C a vlhkosti 35 %. Tyto nejaktuálnější přijatá data korelují s posledním bodem v průběhu.

Nyní následují průběhy naměřených hodnot vynesené do grafů, které byly zaznamenané přibližně v časovém intervalu od 19:00 do 21:00 hodin.



Obrázek 4.5. Průběh vybraných naměřených hodnot CO_2

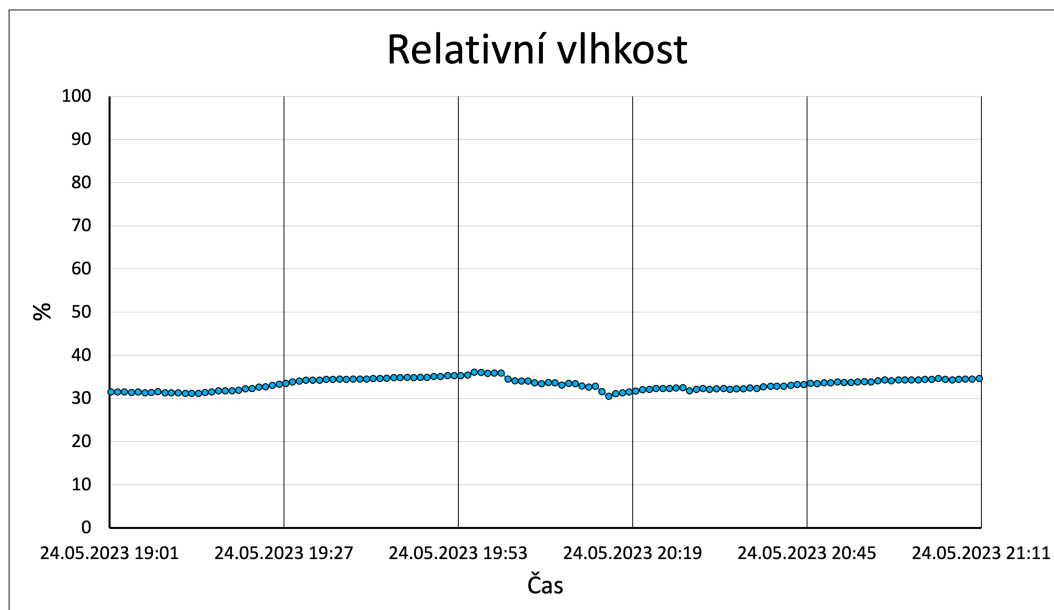
Na obrázku 4.5 je zobrazen zaznamenaný průběh plynu CO_2 v čase do grafu. V čase odhadem 19:15 započal stoupající trend indikující zvyšování koncentrace, jehož maxima bylo dosaženo kolem 20:00 hodiny, v místnosti se v tu dobu nacházely dvě osoby. Poté byl započat proces výměny vzduchu částečně otevřeným oknem, který s nejvyšší pravděpodobností zapříčinil klesající trend naměřených hodnot. Nejstrmější pokles byl nejspíše způsoben prouděním vzduchu v době, kdy odcházela druhá osoba, která při odchodu plně otevřela dveře.



Obrázek 4.6. Průběh vybraných naměřených hodnot teploty

Na obrázku 4.6 je vyneseny průběh teploty v čase do grafu. Ve zvoleném časovém intervalu stojí za povšimnutí, že současně s poklesem CO_2 (viz. obrázek 4.5) došlo

současně ke snížení teploty v místnosti. Z toho lze usoudit, že proces výměny vzduchu měl taktéž vliv na teplotu.



Obrázek 4.7. Průběh vybraných naměřených hodnot relativní vlhkosti

Na obrázku 4.7 je možné pozorovat průběh relativní vlhkosti v čase. Naměřené hodnoty se pohybovaly okolo 35 %. Mírné zvýšení relativní vlhkosti vzduchu bylo nejspíše způsobeno osobami vydechující vzduch s vyšším poměrem vlhkosti. Vezme-li se v úvahu přesnost senzoru ± 5 %, reálná hodnota může odpovídat 40 %, což je podle [2] v přijatelných mezích.

Grafy odpovídají průběhům pozorovaných ve webové aplikaci (viz. obrázek 4.4).

Závěr

V úvodní části práce byla provedena analýza tématu chytré domácnosti. Tento obor se věnuje zvýšení komfortu uživatelů chytrými zařízeními, jejichž cílem je alespoň zprostředkovat informace o stavu prostředí, jenž obývají. Pomocí ní je i možné předcházet budoucím zdravotním následkům, o kterých nemusí být obyvatelé domovů v současné době informováni. Mezi základní sledované parametry patří vlhkost, teplota a koncentrace plynu oxidu uhličitého.

Primárním cílem práce bylo navrhnout a realizovat technické řešení v podobě systému, který zajistí změření a zaznamenání zmíněných základních parametrů, ke kterým bude možné přistupovat mobilním zařízeními. Navrhnutý systém, který splnil požadavky zadání, je postaven na mikropočítači, ke kterému jsou připojeny senzory měřící zvolené parametry, a cloudové službě, jež obsahuje webovou aplikaci pro zobrazování hodnot včetně vývoje. Předností tohoto řešení je monitorování prostředí, ve kterém se nachází senzory, odkudkoli bez ohledu na typ mobilního zařízení a jeho operační systém.

Práce je také zaměřena na principy měření senzorů pro dané veličiny a jejich vlastnosti, které je od sebe odlišují. Výběr senzorů byl ovlivněn jejich parametry, dostupností a osobními zkušenostmi. Bylo zmíněno celkem šest senzorů, které se dělí do třech kategorií, jež pracují na rozdílných fundamentech. Také byl kladen důraz na komunikaci mezi mikropočítačem a zvolenými senzory prostřednictvím protokolů 1-wire a Inter-Integrated Circuit označovaného zkratkou I2C. V kapitole 3, pojmenována „Realizace“, byly demonstrovány kroky, podle kterých program postupuje a ve kterých je zahrnuto například: navázání spojení, spuštění měřicího procesu nebo přijetí naměřené hodnoty. Dále je objasněno, jak lze využít internetového připojení v kombinaci s protokolem MQTT pro přenos naměřených hodnot.

Realizovaný návrh systému uspěl při použití v praxi. Byl jím monitorován obývaný prostor. Nejvíce se uplatnilo měření koncentrace CO_2 , které pravděpodobně pomohlo předejít dosažení vysoké úrovně, která by měla negativní vliv na lidský organismus.

zdroje

- [1] MANRAL, Kavita. *What is Smart Home or Smart Building?* [online]. 2021 [vid. duben 2023]. Dostupné na <https://blog.se.com/homes/2021/04/16/what-is-smart-home-or-smart-building/>.
- [2] ANDRES, Ing. Petr. *Vlhkost vzduchu a hygiena prostředí* [online]. 2018 [vid. duben 2023]. Dostupné na <https://vetrani.tzb-info.cz/vnitřni-prostredi/18142-vlhkost-vzduchu-a-hygiena-prostredi>.
- [3] ANDRES, Ing. Petr. *Současné možnosti vzdáleného monitorování, ovládání a správy budov, domů a bytů* [online]. 2016 [vid. duben 2023]. Dostupné na <https://automatizace.hw.cz/mereni-a-regulace-mimochoodem/moznosti-vzdaleneho-ovladani-a-spravy-budov-domu-a-bytu.html>.
- [4] SERVICES, Wisconsin departmentofhealth. *Carbon Dioxide* [online]. 2023 [vid. duben 2023]. Dostupné na <https://www.dhs.wisconsin.gov/chemical/carbondioxide.htm>.
- [5] HRUBÝ, Ing. Libor. *Pasivní domy – efektivní větrání je základem pro zdravý domov* [online]. 2020 [vid. duben 2023]. Dostupné na <https://www.drevostavitel.cz/clanek/pasivni-domy-efektivni-vetrani>.
- [6] STANISLAV ĎAĎO, Marcel Kreidl. *SENZORY a měřící obvody*. Praha: Vydavatelství ČVUT, 1996. ISBN 80-01-01500-9.
- [7] VOJÁČEK, Antonín. *Přehled principů el. měření teploty - 1. díl* [online]. 2014 [vid. květen 2023]. Dostupné na <https://automatizace.hw.cz/prehled-principu-el-mereni-teploty-1-dil>.
- [8] TECHNOLOGY, Process sensing. *How Do CO2 Sensors Work?* [online]. 2023 [vid. březen 2023]. Dostupné na <https://www.processsensing.com/en-us/blog/how-CO2-sensors-work.htm>.
- [9] TECHNOLOGIES, Disruptive. *What Is a CO2 Sensor and How Does It Work?* [online]. 2023 [vid. duben 2023]. Dostupné na <https://www.disruptive-technologies.com/blog/what-is-a-co2-sensor-and-how-does-it-work>.
- [10] IT-SLOVNIK.CZ, Team. *Co je to Cloud?* [online]. 2022 [vid. květen 2023]. Dostupné na <https://it-slovník.cz/pojem/cloud>.
- [11] IT-SLOVNIK.CZ, Team. *Co je to Webová aplikace?* [online]. 2022 [vid. květen 2023]. Dostupné na <https://it-slovník.cz/pojem/webova-aplikace>.
- [12] IT-SLOVNIK.CZ, Team. *Co je to LAN?* [online]. 2022 [vid. květen 2023]. Dostupné na <https://it-slovník.cz/pojem/lan>.
- [13] MQTT. *MQTT: The Standard for IoT Messaging* [online]. 2023 [vid. květen 2023]. Dostupné na <https://mqtt.org/getting-started/>.
- [14] YASSEIN MUNEER BANI, Shadi ALJWARNEH, Mohammed Q.SHATNAWI a Razan AL-HATMI. *Internet of Things: Survey and open issues of MQTT protocol*. IEEE, 2017 [vid. květen 2023]. ISBN 978-1-5090-6778-7.

- [15] WATELECTRONICS. *What is I2C Protocol* [online]. 2022 [vid. květen 2023]. Dostupné na <https://www.watelectronics.com/i2c-protocol/>.
- [16] UNIVERSITY, Microchip. *Practical I2C: Introduction, Implementation and Troubleshooting* [online]. 2023 [vid. květen 2023]. Dostupné na <https://mu.microchip.com/practical-i2c-introduction-implementation-and-troubleshooting>.
- [17] HARRIS, Mark. *Serial Communications Protocols: Part Seven - 1-Wire* [online]. 2021 [vid. květen 2023]. Dostupné na <https://resources.altium.com/p/7-serial-communications-protocols-1-wire>.
- [18] RAJ, Aswinth. *What is PWM: Pulse Width Modulation* [online]. 2018 [vid. březen 2023]. Dostupné na <https://circuitdigest.com/tutorial/what-is-pwm-pulse-width-modulation>.
- [19] BRENTRU. *Adafruit_IO_Python GitHub repository* [online]. 2022 [vid. květen 2023]. Dostupné na https://github.com/adafruit/Adafruit_IO_Python/tree/master.
- [20] TEKKTRIK. *Adafruit_CircuitPython_DHT GitHub repository* [online]. 2023 [vid. březen 2023]. Dostupné na https://github.com/adafruit/Adafruit_CircuitPython_DHT.
- [21] LIU, Thomas. *AM2302 Datasheet PDF* [online]. 2020 [vid. duben 2023]. Dostupné na <http://www.datasheet39.com/PDF/942482/AM2302-datasheet.html>.
- [22] TEKKTRIK. *Adafruit_CircuitPython_SCD4X GitHub repository* [online]. 2022 [vid. březen 2023]. Dostupné na https://github.com/adafruit/Adafruit_CircuitPython_SCD4X/tree/main.
- [23] SENSIRION. *Data Sheet SCD40/41* [online]. 2022 [vid. březen 2023]. Dostupné na https://sensirion.com/media/documents/48C4B7FB/6426E14D/CD_DS_SCD40_SCD41_Datasheet_D1_052023.pdf.
- [24] FLAVIO-FERNANDES. *Adafruit_IO_Python/Adafruit_IO /mqtt_client.py GitHub repository* [online]. 2022 [vid. květen 2023]. Dostupné na https://github.com/adafruit/Adafruit_IO_Python/blob/master/Adafruit_IO/mqtt_client.py.