



Středoškolská technika 2023

Setkání a prezentace prací středoškolských studentů na ČVUT

MONITOROVÁNÍ VNITŘNÍHO PROSTŘEDÍ MÍSTNOSTI

Michal Grec

SPŠE V Úžlabině
V Úžlabině 320, Praha 10

Čestné prohlášení

Odevzdáním této maturitní práce na téma **Monitorování vnitřního prostředí místnosti** potvrzuji, že jsem ji vypracoval/a pod vedením vedoucího samostatně za použití v práci uvedených pramenů a literatury. Dále potvrzuji, že odevzdaná vytištěná verze písemné zprávy (protokolu) a plakátu se plně shoduje s odevzdanou elektronickou verzí.

V Praze dne.....

.....
Podpis autora/autorky práce

Anotace

Tato práce se zabývá monitorováním fyzikálních veličin, jako je vlhkost vzduchu, teplota prostředí a kvalita vzduchu v budově, které přímo ovlivňují vnitřní prostředí jednotlivých místností z hlediska obyvatelnosti a pohodlí lidí a zvířat. Monitorování zahrnuje nejen problematiku měření zmíněných veličin v různých částech objektu s dostatečně velkou přesností, ale také jejich vizualizaci v závislosti na čase uživatelům pomocí grafického rozhraní a práci s naměřenými hodnotami.

Klíčová slova

Inteligentní budovy, IoT, monitorování prostředí, uživatelské rozhraní

3 až 5 klíčových slov vystihujících práci; jedná se o výčet slov v jediném řádku, která jsou oddělena čárkami; řazení klíčových slov je zpravidla podle abecedy

Annotation

This work deals with monitoring of physical quantities such as humidity of air, temperature of environment and air quality in a building, which directly impact the inner environment of individual rooms from point of housing and wellbeing of people and animals. Monitoring does not only include the problematics of measuring of the listed quantities in different parts of a building with sufficient accuracy, but also their visualization as a function of time to users using a graphical interface and working with measured values.

Keywords

Smart buildings, IoT, environment monitoring, user interface

Obsah

1	ÚVOD A CÍL PRÁCE	5
2	VÝBĚR KOMPONENTŮ, MATERIÁLŮ A PROGRAMOVACÍCH PROSTŘEDKŮ	6
2.1	VÝBĚR JÁDRA ŘÍDÍCÍ JEDNOTKY	6
2.2	VÝBĚR MĚŘICÍCH SENZORŮ.....	7
2.2.1	<i>Senzor teploty a vlhkosti</i>	7
2.2.2	<i>Senzor kvality vzduchu</i>	8
2.3	VSTUPNÍ A VÝSTUPNÍ ZAŘÍZENÍ	8
2.4	NÁVRH KRYTU	9
2.5	PROGRAMOVACÍ PROSTŘEDKY	9
3	VÝROBA A PROGRAMOVÁNÍ	10
3.1	ŘÍDÍCÍ JEDNOTKA	10
3.1.1	<i>Návrh krytu</i>	10
3.1.2	<i>Napájení</i>	12
3.1.3	<i>Zapojení signalizačních LED</i>	13
3.1.4	<i>Kompletace</i>	13
3.1.5	<i>Software</i>	14
3.1.6	<i>Odesílání dat na místní síť</i>	16
3.2	BEZDRÁTOVÝ MODUL	17
3.3	TESTOVACÍ PROVOZ.....	18
3.4	VYUŽITÍ V PRAXI.....	19
4	ZÁVĚR	20
5	SEZNAM POUŽITÝCH ZDROJŮ.....	21
6	SEZNAM PŘÍLOH.....	22

1 Úvod a cíl práce

Vnitřní prostředí místností je velmi důležitý faktor kvality života v budově. Proto je dobré toto prostředí monitorovat, aby se obyvatelé cítili více pohodlně, bylo chráněno jejich zdraví a případně se snížily náklady za vytápění, nebo chlazení prostorů budovy. Je proto nutné monitorovat teplotu, vlhkost a kvalitu vzduchu v jednotlivých místnostech s dostatečnou přesností a frekvencí měření. Důležité je také, aby byly tyto informace přehledně zobrazovány obyvatelům budov a aby bylo možné s daty pracovat a provádět statistiku.

Cílem práce je vytvořit zařízení, které by provádělo měření v celé budově, či její části s pomocí bezdrátových senzorů a řídicí jednotky, na které by se dala data zobrazovat v uživatelsky přívětivém grafickém rozhraní. Řídicí jednotka by také měla umožňovat uživateli ovládat jednotlivé prvky a pracovat s daty pomocí vhodného vstupního zařízení. Zařízení by mělo být schopné uživateli zobrazit jak aktuální, tak předešlé výsledky měření a případně ho varovat před nebezpečím, jako je únik nebezpečného plynu, nebo příliš vysoká nebo nízká teplota.

2 Výběr komponentů, materiálů a programovacích prostředků

2.1 Výběr jádra řídicí jednotky

Jako jádro celého výrobku bylo potřeba vybrat dostatečně výkonnou výpočetní jednotku, která by měla dostatečně velký výkon, aby dokázala vykonávat několik různých programů najednou a zároveň zobrazovat celé uživatelské rozhraní. Dalším požadavkem byla podpora komunikačních protokolů Bluetooth a WiFi. Kvůli tomuto požadavku jsem se rozhodl odklonit od možnosti použít jednu z desek Arduino, která by sice umožňovala jednodušší práci se senzory díky vestavěným analogově-digitálním převodníkům, ale na druhou stranu tato deska nemá tak velký výkon a možnosti bezdrátového připojení jako jeden z jednodeskových počítačů Raspberry Pi.

Nakonec jsem se rozhodl použít jednodeskový počítač Raspberry Pi 4 model B se 4GB paměti RAM typu LPDDR4. *Tento počítač je dále vybaven čtyřjádrovým 64-bitovým procesorem ARM-Cortex A72 pracující na frekvenci 1,5 GHz. Deska také podporuje komunikační protokoly Bluetooth 5.0, BLE (Bluetooth Low Energy) a WiFi. Tento počítač disponuje čtyřmi porty USB z čehož jsou dva USB porty verze 3.0 a druhé dva USB 2.0 [1]. Tyto porty se dají použít pro připojení různých vstupních zařízení jako je klávesnice nebo myš, která se hodí pro ovládání celého výrobku. Další důležitou vlastností jsou grafické výstupní porty micro-HDMI a MIPI DSI, díky které umožňují snadné připojení k různým displejům a monitorům, na kterých je možné zobrazit uživatelské rozhraní.*



Obrázek 1: Raspberry Pi 4 [1]

2.2 Výběr měřících senzorů

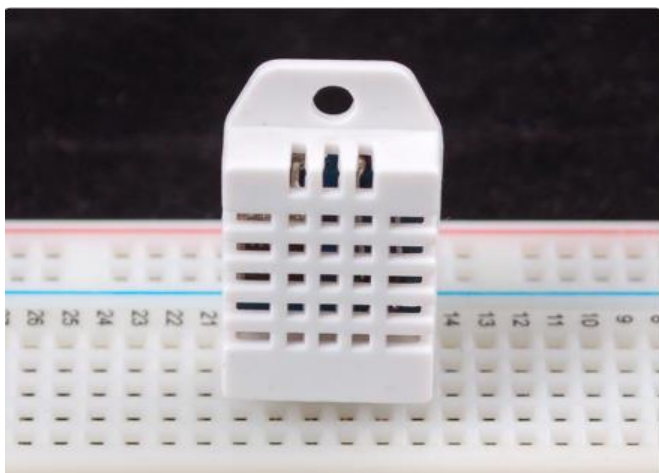
2.2.1 Senzor teploty a vlhkosti

Při výběru senzoru pro měření teploty a vlhkosti jsem vybíral hlavně podle rozsahu měření, přesnosti měření a způsobu a jednoduchosti propojení s Raspberry Pi.

Jako první jsem našel senzor BME280, který kromě teploty a vlhkosti dokáže měřit také atmosférický tlak. Funkci kontroly přetlaku nebo podtlaku jsem však neshledal pro prostředí obytných budov velmi potřebnou. *Senzor se vyznačuje rozsahem měření vlhkosti od 0 do 100 % a přesností měření vlhkosti +/- 3 % [2]*, což je pro účel této práce dostačující. Nevýhodou však je poměrně složité připojení k desce Raspberry Pi, protože je potřeba v programu pracovat se sběrnici senzoru, což celé programování komplikuje. Další nevýhodou je jeho vyšší cena oproti podobným senzorům.

Další zvažované senzory byly senzory DHT11 a DHT22. Senzory se od sebe liší zejména rozsahem měření, přesností a cenou. *DHT11 měří teplotu pouze od 0 do +50 °C s poměrně malou přesností +/-2°C. Měření vlhkosti je také omezeno, a to na rozsah od 20 do 80 % s přesností +/- 5 %. Senzor DHT22 oproti tomu měří teplotu ve větším rozsahu od -40 do +80 °C s vyšší přesností měření +/- 0,5 °C. I vlhkost tato varianta měří ve větším rozsahu a to od 0 do 100 %. Přesnost měření vlhkosti se udává +/- 2–5 % [3]*. Jedinou výhodou senzoru DHT11 je jeho nižší cena a menší rozměry.

Pro mou práci jsem se rozhodl využít senzor DHT22, protože měří požadované veličiny v dostatečně velkém rozsahu s vysokou přesností a jeho připojení k řídicí jednotce je velmi jednoduché.



Obrázek 2: Senzor DHT22 [3]

2.2.2 Senzor kvality vzduchu

Pro měření kvality vzduchu jsem vybíral mezi senzory typu MQ-x. Senzory tohoto typu se od sebe liší hlavně tím, jaký plyn dokáží detekovat. Pro mé potřeby jsem shledal nejvhodnější senzor MQ-135. *Tento senzor detekuje přítomnost nebezpečných plynů jako amoniak, oxidy dusíku, alkoholy, benzen, oxid uhličitý, nebo i kouř. Senzor má digitální i analogový výstup* [4]. Osobně jsem využil digitální výstup, který však umožňuje pouze nespojitě měření. Důvodem tohoto rozhodnutí je jednodušší zapojení a složitost programování detekce přesné koncentrace jednotlivých plynů v okolí.



Obrázek 3: Senzor MQ-135 [5]

2.3 Vstupní a výstupní zařízení

Pro snadné ovládání a čtení dat ze zařízení jsem se rozhodl použít dotykový displej, který tak plní funkci vstupního i výstupního zařízení. Konkrétně jsem použil *pětipalcový dotykový LCD displej Waveshare s kapacitní dotykovou vrstvou. Rozlišení obrazovky je 800x480 pixelů* [6] a k Raspberry Pi se připojuje pomocí rozhraní DSI. Pro některé funkce je potřeba také připojit k výrobku USB klávesnici.

Jako další výstupní prvek jsem se rozhodl použít signalizaci pomocí LED. Celkem jsem připojil tři LED, které signalizují, zda je připojeno napájení, indikaci, zda proběhlo měření teploty a také výstrahu.



Obrázek 4: Dotykový displej Waveshare [6]

2.4 Návrh krytu

Pro výrobu krytu jsem využil technologii 3D tisku. Při výběru materiálu bylo třeba vzít v potaz, že senzor MQ-135 se musí zahřát na pracovní teplotu okolo 50 °C. Dále musel započítat fakt, že jako spojovací materiál krytu budou použity šrouby a je proto větší nárok na mechanickou odolnost materiálu. Rozhodl jsem se proto pro materiál PET-G, *který se začne deformovat až při dosažení 68 °C a vyznačuje se dobrou mechanickou odolností* [7].

2.5 Programovací prostředky

Celý program pro řídicí jednotku jsem se rozhodl napsat v programovacím jazyce Python. Jazyk jsem si oblíbil hlavně díky jeho příjemné syntaxi. Pro komunikaci se senzorem DHT22 bylo potřeba importovat knihovnu *adafruit dht*. Pro vizualizaci naměřených hodnot pomocí grafů jsem použil knihovnu *matplotlib*, která umožňuje ke grafům přidat ovládací panel pro lepší přehlednost grafů. Pro tvorbu uživatelského rozhraní jsem použil knihovnu *Tkinter*, která pomáhá vytvářet okenní aplikace pro programovací jazyk Python.

Pro odesílání dat na místní síť jsem využil další dva programovací jazyky. Pro statickou část stránky programovací jazyk HTML a pro dynamickou část jazyk PHP. Program jsem zapisoval do konfiguračního souboru serveru.

Bezdrátový senzor jsem programoval v programovacím jazyce C++. Pro programování a flashování programu do senzoru jsem se rozhodl použít prostředí Arduino IDE.

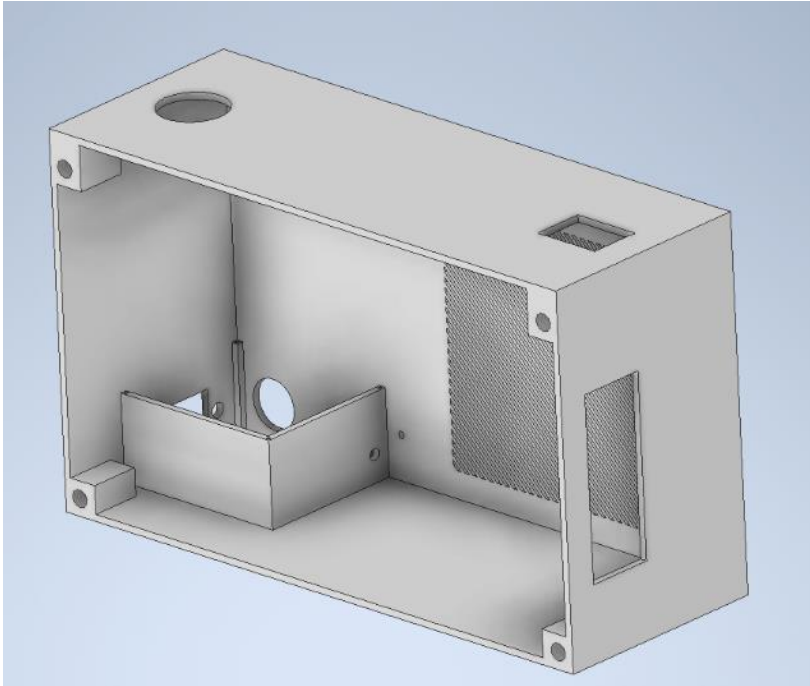
3 Výroba a programování

Cílem je vytvořit zařízení, které by provádělo měření v celé budově, či její části s pomocí bezdrátových senzorů a řídicí jednotky, na které by se dala data zobrazovat v uživatelsky přívětivém grafickém rozhraní. Řídicí jednotka by také měla umožňovat uživateli ovládat jednotlivé prvky a pracovat s daty pomocí vhodného vstupního zařízení. Zařízení by mělo být schopné uživateli zobrazit jak aktuální, tak předešlé výsledky měření a případně ho varovat před nebezpečím, jako je únik nebezpečného plynu, nebo příliš vysoká nebo nízká teplota.

3.1 Řídicí jednotka

3.1.1 Návrh krytu

Kryt výrobku jsem navrhnul v programu pro vytváření 3D modelů Autodesk Inventor. Nejprve jsem vytvořil dutý kvádr, do kterého jsem následně udělal díry pro šrouby na uchycení jádra řídicí jednotky. Potom jsem vytvaroval díry pro senzory DHT22 a MQ-135. Tyto jsem navrhnul tak, aby v nich senzory držely bez potřeby dalšího spojovacího materiálu. Uživatel si tak musí sice dát pozor, aby nezamáčkl senzory dovnitř krabičky, ale na druhou stranu se takto usnadní případná výměna vadného senzoru a vzhled celého výrobku vypadá více uceleně. Dále jsem vytvořil větrací mřížku, která pomůže chladit čipy desky Raspberry Pi, které by se jinak v uzavřeném prostoru bez větrání přehřály. Dále jsem také vytvořil otvor pro ethernetové a USB porty, aby bylo možné připojit zařízení k internetu pomocí ethernetového kabelu a také aby se daly do USB portů připojit různé vstupní zařízení, jako je klávesnice nebo myš. Poté bylo potřeba vymyslet způsob, jak spojit tuto část krytu s jeho druhou částí, kterou jsem měl v plánu následně navrhnout. Nakonec jsem se rozhodl vyzkoušet vytvořit hluboké otvory, do kterých by se zapustily distanční sloupky, které by sloužily jako závitky pro šrouby, které by obě části držely u sebe.



Obrázek 5: Návrh první části krytu [vlastní tvorba]

Po dokončení návrhu první části krytu jsem se pustil do návrhu části druhé. U této části bylo nutné vycházet z rozměrů první části. Nejpodstatnější bylo správné umístění displeje a děr pro spojení obou částí. Dále jsem vytvořil otvory pro LE diody s malými ikonami, které označují, co jaká LED signalizuje. Nakonec jsem vytvořil také jednoduché logo celého výrobku. Obě části jsem následně sám vytiskl na mé 3D tiskárně Průša i3 Mk2. První část jsem celou vytisknul z černého filamentu. Druhou část jsem tiskl po vrstvách s tím, že vrstvu s ikonkami a spodní částí loga vytiskl v oranžové barvě a zbytek loga následně opět černou. Po vytištění jsem obě části zbavil vytištěných podpor a nechtěných vláken, které vznikají při tisku.

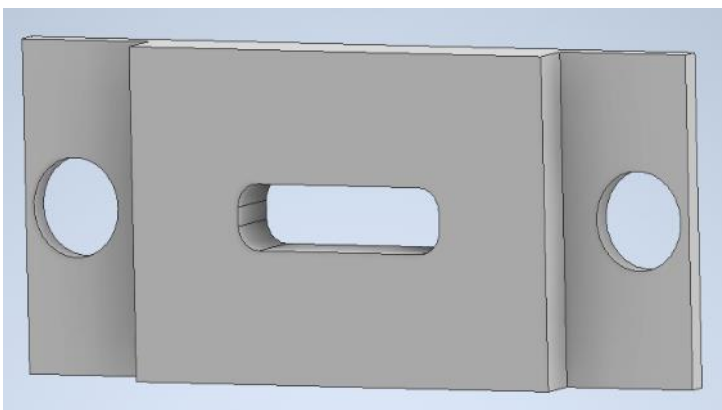


Obrázek 6: Návrh druhé části krytu [vlastní tvorba]

3.1.2 Napájení

Kvůli tomu, že napájecí konektor Raspberry Pi není možné dát dostatečně blízko ke stěně krytu jsem musel vymyslet, jak celý výrobek napájet. Díky tomuto problému jsem však byl schopný také vyřešit absenci vypínacího tlačítka Raspberry pi.

Tento problém jsem vyřešil tak, že jsem na volný USB-C konektor naletoval kolébkový vypínač. Na ten jsem následně naletoval USB-C kabel s volným koncem. Na vývody vypínače jsem také nasadil bužírky, aby se předešlo případnému nechtěnému spojení dvou drátů v případě mechanického poškození. Pro takto vytvořený prodlužovací kabel s vypínačem jsem následně vytvořil na 3D tiskárně držák, který pomůže udržet konektor u stěny krytu. Raspberry Pi 4 vyžaduje napájecí zdroj, který má na výstupu 5 V a 3 A. Z tohoto důvodu není možné použít běžnou nabíječku pro mobilní telefon, protože jinak by mohlo dojít ke ztrátě dat uložených na SD kartě, nebo dokonce k jejímu zničení.



Obrázek 7: Návrh záslepky [vlastní práce]



Obrázek 8: Napájecí obvod [vlastní práce]

3.1.3 Zapojení signalizačních LED

Do přední strany krytu jsem nalepil 3 LED. První dioda má zelenou barvu a signalizuje, že je řídicí jednotka zapojena do sítě. Druhá má barvu modrou a signalizuje, že proběhlo měření teploty a vlhkosti. Uživatel si tak může ověřit, že proběhl měřicí cyklus. Třetí LED má červenou barvu a slouží jako varovná kontrolka. Když se tato dioda začne blikat, tak to znamená, že senzor MQ-135 naměřil vysokou koncentraci některého z měřených plynů. Rezistory pro signalizační LED jsem vypočítal podle Ohmova zákona: $R = \frac{U}{I}$. Proud tekoucí do diody jsem si zvolil 15 mA a napětí na pinech GPIO je 3,3 V. Po dosazení do vzorce: $R = \frac{3,3 V}{15 \cdot 10^{-3} A} \cong 330 \Omega$.

3.1.4 Kompletace

Po dokončení všech individuálních komponent jsem začal s kompletací výrobku. Nejprve jsem vložil distanční sloupky M2,5 do svých děr. S tímto řešením jsem byl nakonec velmi spokojený, protože distanční sloupky pevně drží bez jakéhokoliv dalšího spojovacího materiálu a je možné snadno našroubovat druhou část krytu. Dále jsem nainstaloval kolébkový vypínač a napájecí konektor. Konektor jsem zafixoval pomocí tavné pistole. Následně jsem vložil senzor MQ-135 a senzor DHT22. Sensory jsem spojil s Raspberry Pi pomocí propojovacích drátů. Dalším krokem bylo přilepit k druhé části krytu signalizační LED diody. Diodám jsem následně spojil katody drátem připojil je na pin země Raspberry Pi. Jejich anody zapojil do vybraných GPIO pinů na desce Raspberry Pi. Jádru řídicí jednotky s displejem jsem do krytu uchytil pomocí M2,5 distančních sloupků a šroubů. Nakonec jsem přišrouboval k sobě obě části krytu.

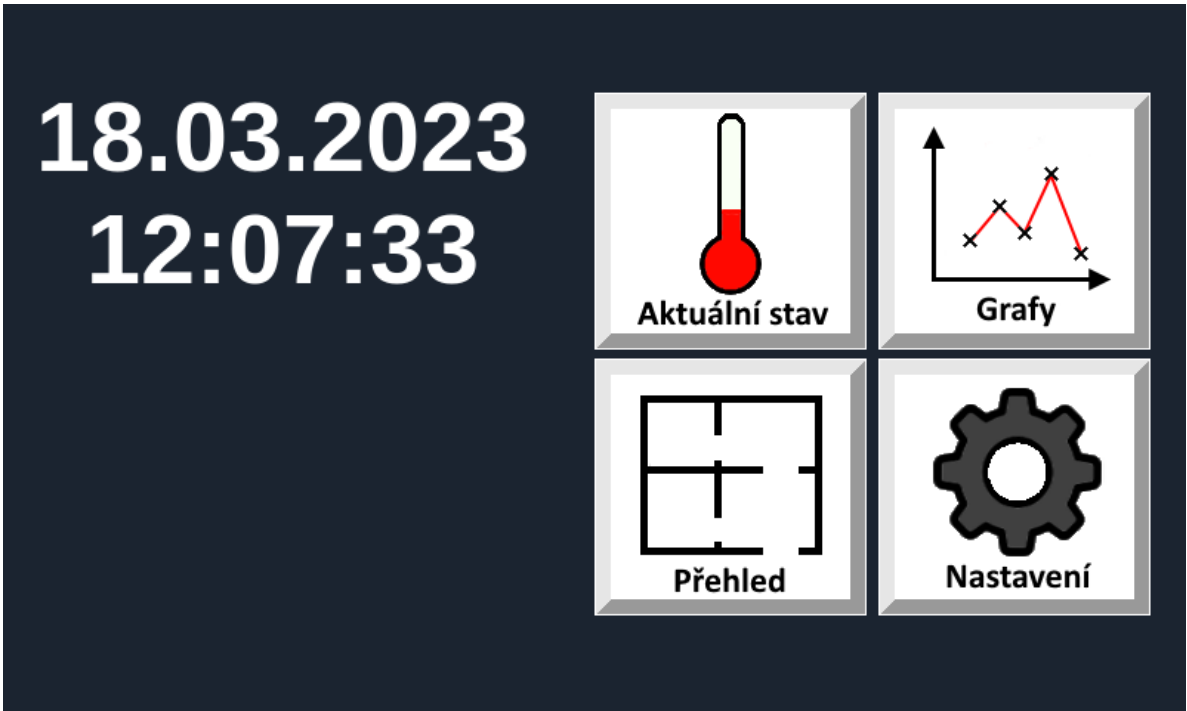


Obrázek 9: Zkompletovaný výrobek [vlastní práce]

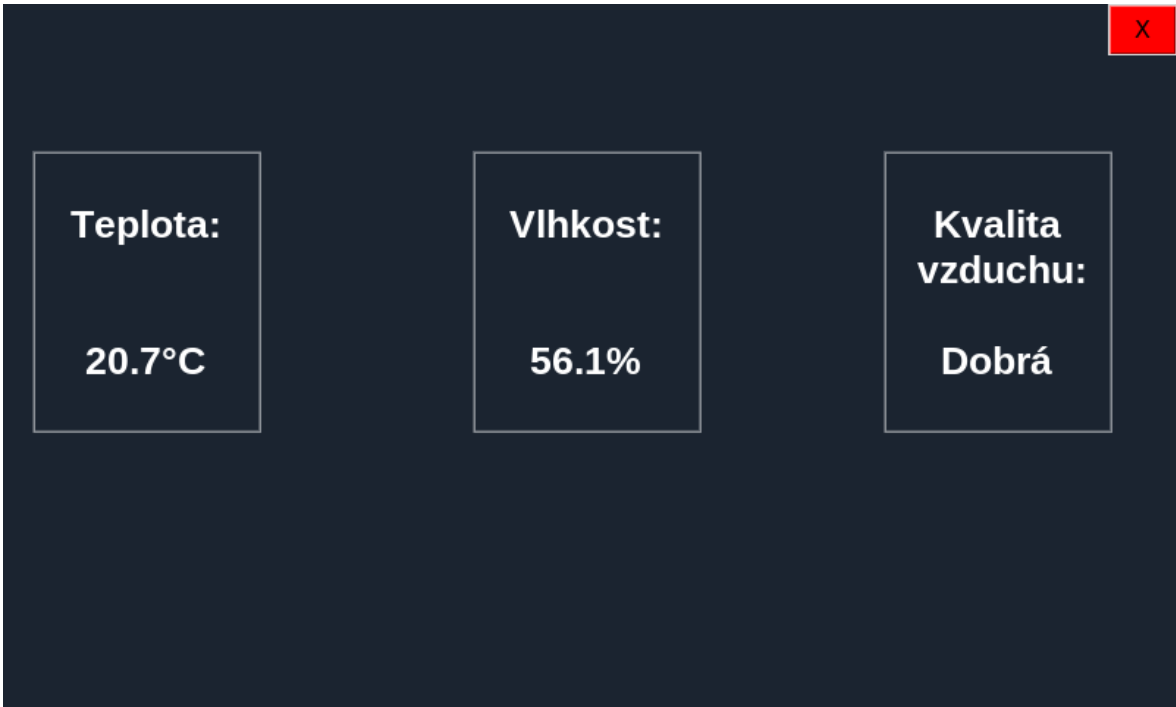
3.1.5 Software

Tvorba softwaru byla největší a nejsložitější část mé práce. Nejprve jsem se musel naučit v programovacím jazycích Python, C++, HTML a PHP. K výuce jsem použil různé webové stránky, které jsou uvedeny v přílohách. Jako první jsem začal programovat samotné uživatelské rozhraní. Po vytvoření základního okna jsem se pustil do výběru barevné palety. Zkoušel jsem několik různých kombinací barev, ale nakonec jsem se rozhodl pro tmavě modré pozadí a bílá tlačítka a text. Následně jsem začal programovat jednotlivé funkce výrobku a přiřadil k nim tlačítka, kterým jsem vytvořil ikonky v grafickém softwaru Paint.net.

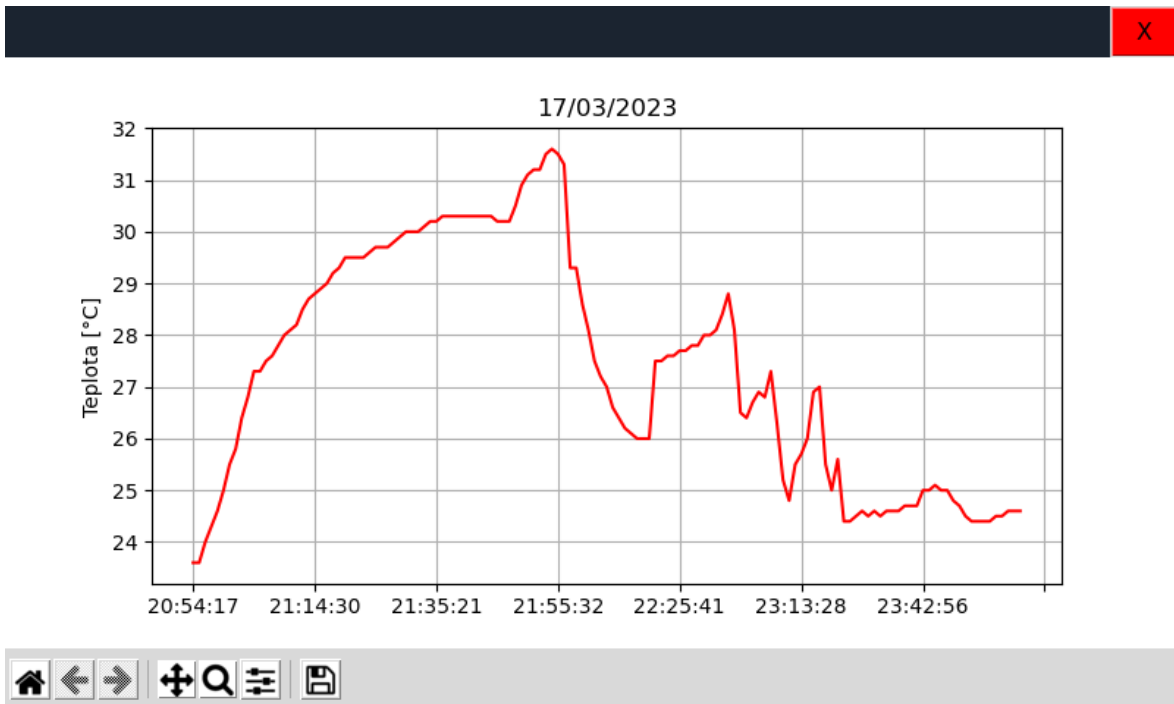
Po vytvoření funkcí s obecnými proměnnými jsem vytvořil separátní programy pro měření pomocí senzorů. Programů je více, aby se eliminovalo zpoždění celého programu a nemohlo se stát, že celý program selže kvůli poruše některého ze senzorů. Vytvořil jsem tedy jeden program pro měření vlhkosti a teploty pomocí senzoru DHT22 a druhý program pro měření kvality vzduchu senzorem MQ-135. Programy jsem vytvořil dva, protože kvalitu vzduchu je potřeba měřit velmi často, protože při úniku nebezpečného plynu by mohl dlouhý interval mezi měřeními ohrozit zdraví a majetek. Kvalitu vzduchu proto měřím každou sekundu. Měřit takto často teplotu a vlhkost mi přišlo být zbytečné a pro tvorbu grafů by bylo takto velké množství dat za jeden den náročné a pomalé. Zároveň je však nutné, aby měl uživatel přístup k, pokud možno co nejaktuálnějším hodnotám. Proto jsem se rozhodl, že senzor DHT22 bude měřit každou minutu. Program pro měření senzoru DHT22 nejdříve zapisuje naměřené hodnoty do testového souboru, pro aktuální data, kam se zároveň zapisuje i aktuální kvalita vzduchu. Dále se zapisují zvláště naměřené hodnoty teploty a vlhkosti do dalších textových souborů, kde je u každé hodnoty zapsán i čas měření. Z těchto souborů následně čte hlavní program hodnoty pro tvorbu grafů, které se vytváří pomocí knihovny *matplotlib*, a zobrazení aktuálního stavu. Do hlavní nabídky jsem také doplnil hodiny ukazující aktuální čas.



Obrázek 10: Hlavní menu [vlastní práce]



Obrázek 11: Okno sledování aktuálního stavu [vlastní práce]



Obrázek 12: Ukázka vytvořeného grafu z testovacího měření [vlastní práce]

Jako další jsem programoval funkci návrhu plánu budovy. Začal jsem programováním jednoduchého programu pro malování. Při zkoušení funkce jsem si však uvědomil, že nakreslit na displej rovnou čáru, která je nutná pro téměř každý půdorys budovy je velmi obtížné. Funkci jsem proto upravil tak, že si uživatel vybere, zda chce kreslit vodorovnou, nebo svislou čáru a program vytvoří rovnou čáru ze zadaných počátečních a koncových souřadnic. Dále jsem naprogramoval také gumu, pro mazání čar a ukládání jsem vyřešil tak, že se nakreslený návrh uloží ve formátu png a následně se dá opět jednoduše načíst zpět.

3.1.6 Odesílání dat na místní síť

Pro odesílání dat na místní síť jsem na řídicí jednotku nainstaloval *Apache web server*. Samotnou webovou stránku jsem naprogramoval v programovacích jazycích PHP a HTML. Jako první jsem naprogramoval statickou část stránky. V jazyce HTML jsem vytvořil nadpis stránky a nastavil barvu textů a pozadí. Následně jsem naprogramoval dynamickou část stránky. Tato část je naprogramovaná v programovacím jazyce PHP. Stránka tak ukazuje aktuální datum a poslední měření ze řídicí jednotky. K webové stránce se lze připojit z jakéhokoliv zařízení, které je připojeno ke stejné lokální síti a má přístup k webovému prohlížeči. Do prohlížeče stačí napsat IP adresu řídicí jednotky, kterou si člověk může zobrazit v uživatelském rozhraní. Data zobrazená na stránce se aktualizuje pokaždé, když uživatel stránku obnoví.

Monitorovací HUB

18.03.2023

Aktuální teplota: 21.8 °C

Aktuální vlhkost: 41.6%

Obrázek 13: Vzhled webové stránky [vlastní tvorba]

3.2 Bezdrátový modul

Pro bezdrátové moduly jsem nejdříve chtěl použít ESP 01 8266. Když jsem se ho však pokusil naprogramovat pomocí softwaru Arduino IDE, tak se nikdy nepodařilo s tímto ESP spojit. Zkusil jsem proto použít ESP wroom 32. K této jednotce se mi podařilo připojit bez problémů a mohl jsem proto začít programovat.

Úspěšně jsem naprogramoval měření pomocí senzoru DHT22. Potom jsem začal programovat část programu, která má komunikovat se řídicí jednotkou. Tuto část práce se mi však nepodařilo zhotovit. Podle různých návodů jsem se snažil spojit ESP se řídicí jednotkou a odesílat naměřená data, ale nikdy se komunikace nezdařila.

Kvůli nedostatku času jsem se rozhodl vytvořit alternativu. K již existujícímu zapojení jsem přidal LCD displej, na který jsem napájel piny a trimr sloužící k ovládní jasu displeje. Piny na displeji jsem následně propojil s piny ESP a v prostředí Arduino IDE jsem naprogramoval displej tak, aby neustále ukazoval aktuální teplotu a vlhkost. Následně jsem vytvořil pro modul kryt v programu Autodesk Inventor. Po vytištění na 3D tiskárně jsem do krytu upevnil displej a svázal propojovací dráty pomocí stahovacích pásků. Trimr jsem nastavil tak, aby byly výrazně vidět zobrazované znaky, ale zároveň jsem si dal pozor na to, aby neprosvítaly neaktivní zobrazovací body displeje.



Obrázek 14: Měřicí modul [vlastní práce]

3.3 Testovací provoz

Výrobek jsem testoval průběžně během tvorby. Po dokončení práce jsem také práci podrobil jednomu týdnu testovacího provozu, kdy jsem chtěl otestovat, zda dokáže výrobek běžet po několik dní bez větších výpadků, či jiných závažných chyb. Po několika hodinách měření jsem si všiml, že senzor ukazuje hodnoty měření teploty přes 30°C. Takové horko v mém pokoji jistě nebylo a musel jsem tento test ukončit. Při hledání příčiny tohoto problému jsem položil do pokoje umístil druhý senzor DHT22. Tento senzor ukazoval teploty okolo 23 °C, což odpovídalo teplotě, kterou ukazoval rtuťový teploměr, který jsem použil pro ověření hodnot. Pak jsem zkusil druhý senzor položit přímo vedle senzoru prvního. Teplota druhého senzoru začala rychle stoupat až nakonec oba senzory ukazovaly podobné hodnoty. Tím jsem ověřil, že problém není v samotném senzoru.

Nakonec jsem usoudil, že Raspberry Pi nejspíše ohřívá vnitřní část krytu více, než jsem původně předpokládal. Senzor jsem proto vyjmul z krytu a postavil ho vertikálně, aby sdílel s krytem co nejmenší plochu. Senzor takto už neukazoval tak vysoké teploty, ale stále zde byla poměrně velká odchylka. V programu pro měření jsem proto nastavil, aby se od naměřené hodnoty odečetla konstanta. Tímto jsem docílil dostatečné přesnosti měření. Výrobek následně obstál týden nepřetržitého provozu.



Obrázek 15: Centrální jednotka v provozu [vlastní tvorba]

3.4 Využití v praxi

Výrobek by měl v praxi sloužit v obytných i průmyslových objektech pro monitorování teploty, vlhkosti a kvality vzduchu. Správce objektu by díky těmto informacím mohl optimalizovat vytápění celého objektu, nebo jeho částí. Tím by se mohlo dosáhnout snížení nákladů na provoz budovy. Dále může výrobek sloužit k detekci úniku nebezpečných plynů, takže v domácnosti by mohl například upozornit obyvatele na únik zemního plynu ze sporáku, nebo jiného ohřevného zařízení. Osobně mám výrobek v plánu nainstalovat do mého bytu, abych měl přehled o teplotě a vlhkosti v jednotlivých místnostech a mohl podle toho upravit vytápění a případně zjistil, kde se drží hodně vlhkosti a mohl se tak preventivně opatřit proti vzniku plísně na zdích.

4 Závěr

Cílem práce je vytvořit zařízení, které by provádělo měření v celé budově, či její části s pomocí bezdrátových senzorů a řídicí jednotky, na které by se dala data zobrazovat v uživatelsky přívětivém grafickém rozhraní. Řídicí jednotka by také měla umožňovat uživateli ovládat jednotlivé prvky a pracovat s daty pomocí vhodného vstupního zařízení. Zařízení by mělo být schopné uživateli zobrazit jak aktuální, tak předešlé výsledky měření a případně ho varovat před nebezpečím, jako je únik nebezpečného plynu, nebo příliš vysoká nebo nízká teplota.

Během mé práce jsem se naučil mnoho nových dovedností a nasbíral jsem zkušenosti, které se mi jistě budou v praxi hodit. Naučil jsem se vytvořit funkční zařízení z jednotlivých komponentů, které by samy o sobě neměly žádnou funkci. Navzdory nesplnění všech bodů zadání mé práce jsem rád, že jsem si toto téma vybral, protože jsem se velmi zlepšil v programování, vytváření 3D modelů a řešení praktických problémů.

Největším problémem se staly bezdrátové moduly. Několik týdnů se mi nepodařilo zajistit jejich komunikaci s centrální jednotkou a nezbývalo mi na dokončení mnoho času. Nechtěl jsem však tento cíl úplně zavrhnout a narychlo jsem vytvořil alespoň prototypní samostatný měřicí modul.

Na druhou stranu se mi podařilo vytvořit systém, který dokáže měřit teplotu, vlhkost a kvalitu vzduchu a zapisovat tyto hodnoty, ze kterých následně dokáže vytvořit graf. Tento systém má navíc přívětivé uživatelské rozhraní, které jsem sám navrhnul a naprogramoval. Dalším úspěšným bodem je naprogramování webové stránky, kterou si může zobrazit kdokoliv připojený na stejnou síť, jako je řídicí jednotka.

Celý projekt bych chtěl v budoucnu dále vylepšovat a upravovat, aby měl co nejvíce funkcí a usnadnil tak životy jeho uživatelů. K budoucím krokům určitě využiji všech nových zkušeností a poučení z chyb, které jsem udělal.

5 Seznam použitých zdrojů

1.) Rpishop.cz: RaspberryPpi 4 Model B 4GB - RAM [online]. České Budějovice: Michal Prenner, 2019 [cit. 2023-02-06]. Dostupné z: <https://rpishop.cz/raspberry-pi-4/1598-raspberry-pi-4-model-b-4gb-ram-765756931182.html>

2.) Bosch-sensortec.com: BME280 - datasheet [online]. Reutlingen: Bosch Sensortec, 2022 [cit.2023-03-12]. Dostupné z: <https://www.bosch-sensortec.com/media/boschsensortec/downloads/datasheets/bst-bme280-ds002.pdf>

3.) Adafruit.com: DHT11, DHT22 and AM2302 Sensors [online]. New York City: Adafruit Industries, 2022 [cit. 2023-03-12]. Dostupné z: <https://cdn-learn.adafruit.com/downloads/pdf/dht.pdf>

4.) Hwsensor.com: MQ-135 Gas Sensor Datasheet [online]. Čeng-čou: HANWEI ELECTRONICS, 2018 [cit. 2023-03-12]. Dostupné z: https://www.electronicoscaldas.com/datasheet/MQ-135_Hanwei.pdf

5.) Dratek.cz: Senzor plynů MQ135 MQ - 135 pro Arduino [online]. Havlíčkův Brod: ECLIPSE, 2018 [cit. 2023-03-12]. Dostupné z: <https://dratek.cz/arduino/1298-senzor-plynu-mq135-mq-135-pro-arduino.html>

6.) Waveshare-com: 5inch Capacitive Touch Display for Raspberry Pi, DSI Interface, 800×480 [online]. Šen-čen: Waveshare, 2019 [cit. 2023-03-16]. Dostupné z: <https://www.waveshare.com/5inch-dsi-lcd.htm>

7.) Prusa3D.com: Prusa Tabulka Materiálů [online]. Praha: Prusa Research, 2020 [cit. 2023-01-27]. Dostupné z: <https://help.prusa3d.com/cs/materials>

6 Seznam příloh

Příloha č. 1: Seznam použitých webů pro výuku programování

- 1.) <http://tkinter.programujte.com/tkinter-hello-tkinter.htm>
- 2.) <https://www.pythontutorial.net/tkinter/>
- 3.) <https://pimylifeup.com/raspberry-pi-humidity-sensor-dht22/>
- 4.) <https://pimylifeup.com/raspberry-pi-apache/>
- 5.) <https://www.instructables.com/Connecting-Your-Raspberry-Pi-to-the-Web/>
- 6.) <https://docs.arduino.cc/learn/electronics/lcd-displays>
- 7.) <https://matplotlib.org/stable/index.html>

Příloha č. 2: Program webové stránky

```
<!DOCTYPE html>
<html>
<body>

<h1 style="color:white; font-size:60px"> Monitorovací HUB</h1>
<h2 style="color:white; font-size:40px">
<style>
body{background-color:#1B2430;}
</style>

<?php

$radky = file("/home/pi/log.txt");

echo date("d.m.Y");

echo "<br>";

echo "Aktuální teplota: ";
echo $radky[0];
echo "°C";

echo "<br>";

echo "Aktuální vlhkost: ";
echo $radky[1];
echo "%";

?>
</h2>
</body>

</html>
```

Příloha č. 3: Program měření senzoru DHT22

```
import time
import board
import adafruit_dht
from datetime import datetime
from datetime import date
import RPi.GPIO as GPIO

GPIO.setup(17, GPIO.OUT)
datum = date.today()
cas = datetime.now()

def mereni_dht22():
    dht22 = adafruit_dht.DHT22(board.D4)

    while True:
        try:
            teplota_C = dht22.temperature - 3.2
            vlhkost = dht22.humidity
            teplota_F = teplota_C * (9 / 5) + 32
            teplota_C_text = str(teplota_C)
            teplota_F_text = str(teplota_F)
            vlhkost_text = str(vlhkost)

            zapis_log = open("log.txt", "w")
            zapis_log.write(teplota_C_text + "\n" + vlhkost_text)
            zapis_log.close()

            cas = datetime.now()
            cas_string = cas.strftime("%d/%m/%Y %H:%M:%S")
            zapis_graf = open("historie_teplota.txt", "a")
            zapis_graf.write(cas_string + "\t" + teplota_C_text + "\n")
            zapis_graf.close()

            zapis_vlhkost = open("historie_vlhkost.txt", "a")
            zapis_vlhkost.write(cas_string + "\t" + vlhkost_text + "\n")
            zapis_vlhkost.close()

        except RuntimeError as error:
            print(error.args[0])
            time.sleep(2.0)
            continue
        except Exception as error:
            dht22.exit()

            raise error
            GPIO.output(17, GPIO.HIGH)
            time.sleep(1)
            GPIO.output(17, GPIO.LOW)

            time.sleep(59)
mereni_dht22()
```

Příloha č. 4: Program měření senzoru MQ135

```
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BCM)
GPIO.setup(23, GPIO.IN)
GPIO.setup(27, GPIO.OUT)

while True:
    if GPIO.input(23):
        zapis_MQ = open("MQ135.txt", "w")
        zapis_MQ.write("Dobrá")
        zapis_MQ.close()
        GPIO.output(27, GPIO.LOW)

    else:
        zapis_MQ = open("MQ135.txt", "w")
        zapis_MQ.write("Špatná")
        zapis_MQ.close()
        GPIO.output(27, GPIO.HIGH)
        time.sleep(1)
        GPIO.output(27, GPIO.LOW)
        time.sleep(1)
```

Příloha č. 5: Ukázka programu řídicí jednotky

```
def graf_t_datum():
    datum_input = Toplevel(grafy_okno)
    datum_input.geometry("800x480")
    datum_input.overrideredirect(True)
    datum_input['background'] = '#1B2430'
    close_datum_input = Button(datum_input, text="X",bg="red", font="bold", command=datum_input.destroy)
    close_datum_input.pack(anchor=tkinter.NE, ipadx=4, ipady=1)
    inputLabel = Label(datum_input, text="Zadejte požadované datum ve formátu dd/mm/yyyy" "\n" "Pokud chcete zobrazit všechna měření, napište 0",font="bold", bg="white")
    inputLabel.pack()
    graf_t_datum_okno = Toplevel(grafy_okno)
    graf_t_datum_okno.overrideredirect(True)
    graf_t_datum_okno.geometry("800x480")
    graf_t_datum_okno['background'] = '#1B2430'
    close_t_datum_okno = Button(graf_t_datum_okno, text="X",bg="red", font="bold", command=graf_t_datum_okno.destroy)
    close_t_datum_okno.pack(anchor=tkinter.NE, ipadx=4, ipady=1)

def printDatum():
    global datum
    datum = text.get(1,0, "end-1c")
    if datum in list_datum:
        zobraz_graf()

    elif datum == "0":
        graf_t()
    else:
        errorLabel = Label(datum_input, text="Měření z tohoto data neexistují", fg="RED", font=("Arial", 15, "bold"))
        errorLabel.pack()
        return 0

    datum_input.destroy()

text = tkinter.Text(datum_input, height=1, width=14, font = ("Arial", 40))
text.pack()
OK_button = ttk.Button(datum_input,text="OK", command=printDatum)
OK_button.pack(ipadx = 10)

lbl = tkinter.Label(datum_input, text="")
lbl.pack()

def zobraz_graf():
    a = 0
    vybrane_mereni_list=[]*1000
    for data in list_datum:
        if datum == list_datum[a]:
            vybrane_mereni_list.append(a)
            a = a + 1
        else:
            a = a + 1
    vybrane_hodnoty = []
    vybrane_casy = []
    b = 0
    cislo = vybrane_mereni_list[0]
    for data in vybrane_mereni_list:
        vybrane_hodnoty.append(list_hodnoty[vybrane_mereni_list[b]])
        vybrane_casy.append(list_casy[vybrane_mereni_list[b]])
        b = b + 1
```


Seznam obrázků

Obrázek 16: Raspberry Pi 4 [1]

Obrázek 17: Senzor DHT22 [3]

Obrázek 18: Senzor MQ-135 [5]

Obrázek 19: Dotykový displej Waveshare [6]

Obrázek 20: Návrh první části krytu [vlastní tvorba]

Obrázek 21: Návrh druhé části krytu [vlastní tvorba]

Obrázek 22: Návrh záslepky [vlastní práce]

Obrázek 23: Napájecí obvod [vlastní práce]

Obrázek 24: Zkompletovaný výrobek [vlastní práce]

Obrázek 25: Hlavní menu [vlastní práce]

Obrázek 26: Okno sledování aktuálního stavu [vlastní práce]

Obrázek 27: Ukázka vytvořeného grafu z testovacího měření [vlastní práce]

Obrázek 28: Vzhled webové stránky [vlastní tvorba]

Obrázek 29: Měřicí modul [vlastní práce]

Obrázek 30: Centrální jednotka v provozu [vlastní tvorba]