

Zařízení pro monitoring sdílených ubytovacích prostor

Equipment for monitoring shared accommodation spaces

Bc. Alexander Baršč

Diplomová práce

Vedoucí práce: Ing. Radim Hercík Ph.D.

Ostrava, 2024



Zadání diplomové práce

Student:

Bc. Alexander Baršč

Studijní program:

N0714A150001 Řídicí a informační systémy

Téma:

Zařízení pro monitoring sdílených ubytovacích prostor

Equipment for Monitoring Shared Accommodation Spaces

Jazyk vypracování:

čeština

Zásady pro vypracování:

Cílem práce je návrh a realizace zařízení pro detekci základních veličin provozu pronajímaných ubytovacích prostor jako jsou hotelové pokoje, pokoje na ubytovnách nebo sdílených bytů např. přes platformu AirBnb. Zařízení by mělo měřit například teplotu, úroveň hluku nebo přítomnost kouče a být schopno na základě nastavených prahových veličin notifikovat pronajímatele. Záměrem práce je vytvořit prototyp zařízení, který by umožnil pronajímateli nepřímou kontrolu dodržování podmínek pronájmu ubytovacích prostor při současném zachování soukromí uživatelů pronajímaných prostor.

1. Analýza požadavků ubytovacích podmínek sdílených bytů, hotelů a ubytoven.
2. Návrh konceptu zařízení pro monitoring ubytovacích prostor.
3. Výběr komponent a návrh HW zařízení.
4. Realizace HW prototypu zařízení.
5. Implementace SW zařízení pro monitoring ubytovacích prostor.
6. Testování zařízení pro monitoring ubytovacích prostor.
7. Zhodnocení výsledků a závěr.

Seznam doporučené odborné literatury:

[1] MATOUŠEK, David a Bohumil BRTNÍK. Aplikace algoritmů číslicového zpracování signálů. Praha: BEN - technická literatura, 2014-. ISBN 978-80-7300-478-1.

[2] VEDRAL, Josef a Jan FISCHER. Elektronické obvody pro měřící techniku. Vyd. 2. Praha: Vydavatelství ČVUT, 2004. ISBN 80-01-02966-2.

[3] ZÁHLAVA, Vít. Návrh a konstrukce desek plošných spojů: principy a pravidla praktického návrhu. Praha: BEN - technická literatura, 2010. ISBN 978-80-7300-266-4.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **Ing. Radim Hercík, Ph.D.**

Datum zadání: 01.09.2023

Datum odevzdání: 30.04.2024

Garant studijního programu: prof. Ing. Jiří Koziorek, Ph.D.

V IS EDISON zadáno: 14.11.2023 09:18:51

Abstrakt

Tato diplomová práce řeší návrh a realizaci monitorovacího zařízení pro sdílené ubytovací prostory. Cílem je vytvořit vestavěné zařízení, jenž bude detektovat pohyb osob, měřit teplotu a vlhkost, monitorovat hluk, monitorovat změny v osvětlení a detektovat přítomnost cigaretového kouře. Teoretická část se zabývá aktuální řešeními na trhu, návrhu konceptu samotného zařízení a možnostmi měření jednotlivých veličin. Praktická část popisuje návrh vhodného elektronického zapojení s využitím mikrokontroleru ESP32. Dále řeší následnou realizaci plošného spoje a jeho osazení součástkami. Finální kapitoly práce se zabývají realizaci softwarového řešení naprogramovaného v jazyce C++ s použitím IOT platformy ThingSpeak pro ukládání, vizualizaci a manipulaci se získanými daty. Na závěr je zde popsáno testování zařízení se zhodnocením získaných výsledků.

Klíčová slova

Monitorovací zařízení, sdílené ubytovací prostory, ESP32, C++, ThingSpeak

Abstract

This diploma thesis deals with the design and implementation of monitoring equipment for shared accommodation spaces. The goal was to create a device that would detect the movement of people, measure temperature and humidity, monitor noise, monitor changes in lighting and detect the presence of cigarette smoke. The theoretical part deals with current solutions on the market, design of a possible concept for the device and how to measure individual quantities . The practical part describes the design of a proper electrical schematic with application of the ESP32 microcontroller. Afterwards it deals with the subsequent realization of the printed circuit board and assembly of the components. The final chapters of the work deal with the implementation of a software solution programmed in C++ using the ThingSpeak IOT platform for storing, visualizing and manipulating the acquired data. At the end, the device testing is described here with an evaluation of the obtained results.

Keywords

Monitoring device, shared accommodation spaces, ESP32, C++, ThingSpeak

Poděkování

Rád bych poděkoval Ing. Radimu Hercíkovi, Ph.D za veškerou odbornou pomoc a konzultaci při vytváření diplomové práce. Dále bych chtěl poděkovat svým rodičům za jejich podporu a trpělivost, následně také bych chtěl ještě poděkovat mým kamarádům Petru Kozárovi a Davidovi Šipošovi, jenž mi pomohli s testováním zařízení.

Obsah

Seznam použitých symbolů a zkratek	8
Seznam obrázků	9
Seznam tabulek	11
1 Úvod	12
2 Analýza požadavků ubytovacích podmínek sdílených bytů, hotelů a ubytoven	13
2.1 Analýza stávajících řešení na trhu.	13
3 Návrh konceptu zařízení pro monitoring ubytovacích prostor	18
3.1 Koncepce zařízení	18
3.2 Způsoby detekce měřených veličin	19
4 Výběr komponent a návrh HW zařízení	23
4.1 Použité senzory a snímače	23
4.2 Schéma zapojení	27
5 Realizace HW prototypu zařízení	31
6 Implementace SW zařízení pro monitoring ubytovacích prostor	34
6.1 Úložiště pro naměřené data - ThingSpeak	34
6.2 Návrh a realizace SW	36
6.3 Vytvořené třídy	42
6.4 Použití funkcí ThingSpeaku	43
7 Testování zařízení pro monitoring ubytovacích prostor	45
8 Zhodnocení a závěr	50
Literatura	52

Přílohy **53**

A Seznam příloh **54**

Seznam použitých zkratok a symbolů

API	– Application Programming Interface
BLE	– Bluetooth Low Energy
DFN	– Dual-Flat No-Leads
EEPROM	– Electrically Erasable Programmable Read-Only Memory
EN	– Enable
FET	– Field Effect Transistor
GPIO	– General Purpose Input Output
HW	– Hardware
HTTP	– Hypertext Transfer Protocol
I2C	– Inter-Integrated Circuit
LED	– Light Emitting Diode
LDR	– Light Dependent Resistor
OOP	– Objektově orientované programování
PIR	– Passive Infrared Sensor
REST	– Representational State Transfer
SW	– Software
UART	– Universal Asynchronous Receiver-Transmitter

Seznam obrázků

2.1	Monitorovací zařízení od firmy Minut. [?]	14
2.2	Monitorovací řešení od firmy ABB [4]	15
2.3	Řídící jednotka Loxone [5]	16
2.4	Home Assistant Green [6]	16
3.1	Blokové schéma monitorovacího zařízení	19
3.2	Schématická značka fotorezistoru	20
3.3	Schématická značka fotodiody	20
3.4	Schématická značka fototranzistoru	20
3.5	Blokové schéma PIR detektoru [10]	21
4.1	ESP32 DevKit [13]	23
4.2	Senzor MQ2 [14]	24
4.3	Charakteristiky senzoru MQ2 [14]	24
4.4	Schéma zapojení modulu MQ2 [15]	25
4.5	Uživatelský registr HTU21D [16]	25
4.6	Senzor HTU21D	26
4.7	PIR modul HC-SR501 a schéma [17] [20]	27
4.8	Schéma mikrofonního zesilovače [19]	28
4.9	Zapojení step down měniče SD8942	29
4.10	Zapojení stabilizátoru AMS1117	29
4.11	Zapojení mikrokontroleru ESP32	30
5.1	Osazení součástek plošného spoje	31
5.2	Finální podoba plošného spoje po osazení	32
5.3	Pohled na pouzdro plošného spoje z boku a seshora	33
6.1	Prostředí Platform IO ve Visual Studio Code	35
6.2	Vývojový diagram programu.	38
6.3	Připojení k WiFi síti přes WiFiManager	39

6.4	Diagram třídy HTU21D	42
6.5	Diagram třídy Measurement	43
6.6	Nastavení Reactu	44
6.7	Odeslaný email z ThingSpeaku.	44
7.1	Finální montáž prototypu	45
7.2	Testovací událost: Příchod do místnosti a rozsvícení světla	46
7.3	Testovací událost: Otevření okna	47
7.4	Testovací událost: Spuštění a vypnutí hudby v reproduktorech	48
7.5	Testovací událost: Detekce kouře	49

Seznam tabulek

3.1 Porovnání fotorezistoru, fotodiody a fototranzistoru	21
--	----

Kapitola 1

Úvod

Vlastníci ubytovacích prostor, mezi které patří hotely, ubytovny nebo v posledních letech i sdílené ubytovací prostory jako Airbnb řeší problém, jak tyto prostory efektivně monitorovat, zda ubytovaní klienti dodržují pravidla ubytování, jako například neporušují lokální vyhlášky nebo nebo nepoškozují majetek vlastníka.

Tato práce řeší návrh a realizaci zařízení, jenž monitoruje co nejvíce užitečných veličin, aby mohl pronajímatel vzdáleně monitorovat pronajímaný prostor. Kapitola analýza požadavků řeší požadovanou funkcionality, jenž by takový provozovatel mohl vyžadovat po takové zařízení, dále popisuje existující řešení na trhu jako Minut, Home Assistant nebo Loxone.

Následuje poté kapitola, jenž řeší teoretický návrh takového zařízení, s jeho vstupy a výstupy a požadovaným napájením a komunikací. Jsou dále popsány způsoby jak požadované veličiny měřit.

Je popsán návrh a výběr komponent HW zařízení, popisuje motivaci za použití daného mikrokontroleru a obsahuje nejen schémata finálního zařízení, ale i schéma použitých modulů se senzory, plus jak s nimi při měření správně pracovat, jako popis samotného komunikačního protokolu.

Dále je popsán proces oživení navrženého plošného spoje, jenž byl vyroben na zakázku a následné opravy komplikací po oživení. Také je zobrazeno pouzdro, jenž bylo navrženo na 3D tiskárně.

Kapitola implementace SW se zabývá implementací požadovaných funkcí v jazyce C++ pomocí Arduino knihoven. Je zde popsán vzorový kód a použité implementované třídy. Dále se popisují možnosti zobrazení získaných dat na platformě ThingSpeak, s funkcionalitou spuštění Matlabovských kódů při určitých událostech.

Poslední kapitola se zabývá výsledky finálního testování zařízení, kde se z ThingSpeaku exportovali naměřené data dle určitých událostí, jako příchod osoby do místnosti, zhasnutí světel atd.

Kapitola 2

Analýza požadavků ubytovacích podmínek sdílených bytů, hotelů a ubytoven

Aby realizované zařízení mělo co největší přidanou hodnotu, je nutné správně analyzovat požadavky uživatelů, jenž toto zařízení budou používat a ovládaní tohoto zařízení bylo uživatelsky přívětivé. V tomto případě to budou většinou majitelé nebo zaměstnanci hotelů, sdílených bytů a ubytoven. Požadavky vyplývají ze zájmu samotného vlastníka objektu aby šlo předejít ke škodě majetku, či rušení veřejného pořádku například pořádáním večírků ubytovanými návštěvníky, můžeme je tedy shrnout do několika bodů:

- Je v prostorech rozsvíceno?
- Vyvolávají ubytované osoby přílišný hluk?
- Dochází v prostorech kouření tabákových výrobků?
- Měření teploty a vlhkosti, pro zajištění dostatečného větrání, omezení vzniku plísni atd.
- V případě že v prostorech nemá být nikdo ubytován, realizovat detekci pohybu?

Tyto požadavky musí tedy realizované zařízení řešit. Samozřejmostí je také skutečnost, že monitoring prostor musí být realizován nonstop a samotné zařízení diskrétně umístěno. Poté ovládání zařízení musí být pro uživatele intuitivní, s možností data zobrazit ve vizualizačním rozhraním.

2.1 Analýza stávajících řešení na trhu.

2.1.1 Minut

Minut je švédská společnost, zabývající se řešeními pro monitorování sdílených prostor.



Obrázek 2.1: Monitorovací zařízení od firmy Minut. [?]

Jedno z jejich řešení, je Minut Smart Home Alarm, toto zařízení ve tvaru puku obsahuje několik senzorů, pro monitoring vlhkosti, pohybu, zvuku, teploty a zda nedošlo k neoprávněné manipulaci se samotným zařízením. Toto zařízení je navrženo na instalaci na strop a lze s ním monitorovat aktuální stav v interiérech ale i v exteriérech. Samotné zařízení je napájené akumulátorem, který by po plném nabití měl zařízení udržet v provozu až rok.

Zařízení obsahuje tyto funkce:

- Měření intenzity hluku (dB)
- Detekci cigaretového kouře
- Měření teploty a vlhkosti
- Detekce pohybu
- Posílaní SMS zpráv

Zajímavostí je, že neveškerá funkcionalita toho zařízení je obsažená v základním balení, samotný senzor stojí 100 dolarů, ale pro využití veškerých funkcionalit je třeba zakoupit předplatné, které je dále členěno do několika plánů.

Tedy lze vidět praxi, že lze prodat zařízení, které sice obsahuje hardware a základní funkcionalitu, pro další služby je ale nutné si připlatit. Bohužel ze strany výrobce jsou dostupné jenom obecné parametry, ze kterých nelze vyvodit všechny zálužnosti samotného technického řešení.

2.1.2 KNX

KNX je otevřený celosvětový standard pro řízení provozu budov. Je pod záštitou asociace KNX, která se stará o certifikaci produktů a zaručuje jejich vzájemnou kompatibilitu, ať už pochází od jaké-

hokoliv výrobce. (Schneider, ABB). Například společnost ABB nabízí řešení na bázi KNX nejen pro monitorování prostor, ale systémy, jenž realizují přístupy pomocí RFID čteček, ovládání klimatizace a topení, rolet a mnoho dalšího.



Obrázek 2.2: Monitorovací řešení od firmy ABB [4]

Takové řešení pro provozovatele ubytovacího zařízení přináší spoustu výhod, například:

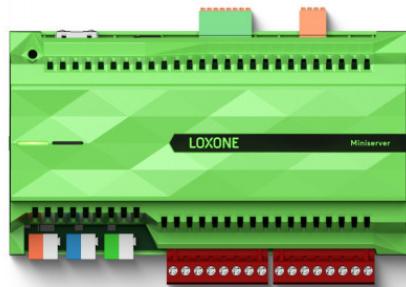
- Možnost programování přístupových karet
- Monitoring pokojů a řízení klimatu z recepce
- Úspora energie díky aktivaci elektrických zařízení místnosti pouze když je host přítomen
- Možnost správy přístupu osob do určitých prostor (například pouze některí hosté mají přístup do posilovny)

2.1.3 Loxone

Loxone je Rakouská společnost, jenž vznikla v roce 2008 s myšlenkou s vytvořením praktické domácí automatizace, kdy samotní majitelé žádné pořádné řešení na trhu nenašli.

Po úspěchu v domácnostech se Loxone rozšířil i do hotelů až po rozsáhlé budovy a výrobní haly. Také lze stáhnout software, jenž umožňuje naprogramovat svoje vlastní automatizační řešení.

V součastnosti Loxone umí řešit úlohy jako regulaci teploty v pokojích, energetický management, inteligentní osvětlení, zabezpečení domácnosti, řízení přístupu, ovládání sauny, údržba bazénu, za-vlažování trávníku.

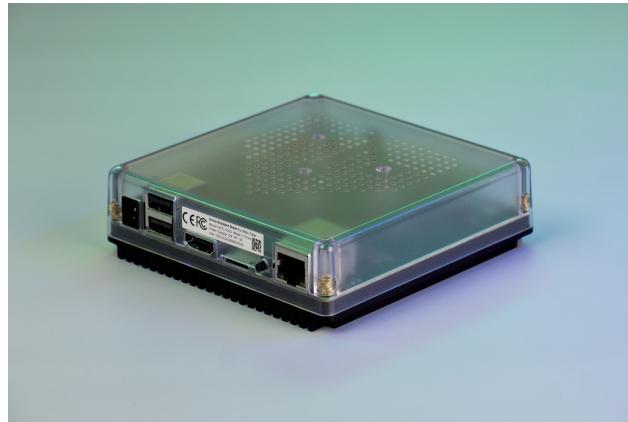


Obrázek 2.3: Řídící jednotka Loxone [5]

2.1.4 Home Assistant

Open source varianta pro automatizaci a řízení chytrých domácností, její hlavním cíle je integrovat a centralizovat různá zařízení a služby v domácnosti do jednoho jednoduše ovladatelného systému. Podporuje širokou škálu chytrých zařízení a protokolů, včetně Wi-Fi, Zigbee, Bluetooth a mnoha dalších. Můžeme tedy integrovat osvětlení, termostaty, zabezpečovací kamery a senzory.

Home Assistant je možné nainstalovat na vlastní server či Raspberry Pi, nicméně je dostupné i vlastní řešení jménem Home Assistant Green, jedná se o hub, který je navržený pro provoz Home Assistantu. Další výhodou je možnosti si celé řešení naprogramovat podle vlastního uvážení.



Obrázek 2.4: Home Assistant Green [6]

Pro shrnutí, pokud chceme levnější vstup do světa domácího monitoringu a automatizace jsou lepší řešení Minut a Home Assistant, díky jejich řádově nižším počátečním nákladům. Nicméně Minut umí pouze měřit a nelze pomocí něho cokoliv řídit, pomocí Home Assistantu jde nastavit i regulaci, ale od dražších řešení je odkázán především na komunitní podporu. Řešení KNX a Loxone

jsou vhodné v případech, kde se automatizace opravdu vyplatí díky úsporám na spotřebované energii i přes vyšší prvotní vstupní náklady.

Cílem této práce je tedy navrhnout vlastní řešení, jenž bude nezávislé na standardech velkých firem a bude schopné poskytnout dostačující funkcionality pro potencionálního uživatele.

Kapitola 3

Návrh konceptu zařízení pro monitoring ubytovacích prostor

Jak už bylo zmíněno v předchozí kapitole, je nutné požadavky pro monitoring ubytovacích prostor zpracovat do technicky realizovatelné podoby.

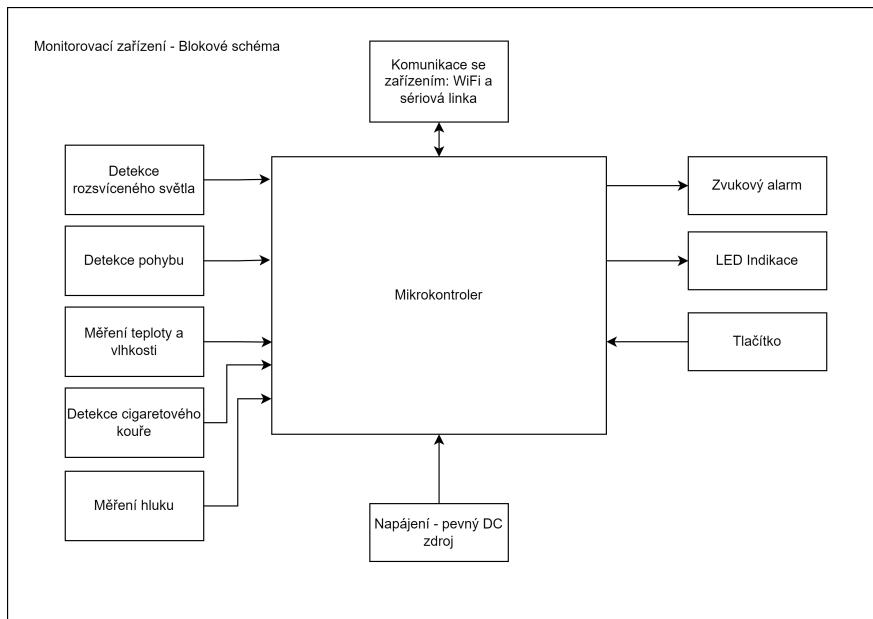
Vzhledem k tomu, že požadavky pro toto zařízení jsou pevně dané, bude se jedna o takzvaný vestavěný systém, jenž je navržený a optimalizovaný pro konkrétní funkcionalitu již od návrhu hardwaru až po samotnou softwarovou implementaci ve vhodném programovacím jazyce.

3.1 Koncepce zařízení

Blokové schéma tohoto zařízení na obrázku: 3.1 popisuje koncepci, která bude realizovaná. Vlevo se nachází vstupní (snímané) veličiny, nahoře způsob komunikace se zařízením, dole způsob napájení zařízení a nakonec vpravo výstupy tohoto zařízení.

Jako napájení bude sloužit pevný zdroj napětí, ideálně stejnosměrných 12 V (například vývod ze zabezpečovacího systému).

Detekce rozsvíceného světla bude realizována pomocí fotocitlivé součástky, s kombinací detekcí pohybu takhle moc zjistit, zda se v prostorách někdo nachází a zbytečně se neplýtvá elektřinou. Měřením teploty a vlhkosti lze zjistit zda nejsou otevřené okna, či nehrozí kvůli vlhkosti ke tvorbě plísní. Detekcí cigaretového kouře můžeme sledovat zda někdo v místnosti nekouří, aby nedošlo k nasáknutí nabytku cigaretovým kouřem. Abychom předešli hlučným večírkům, můžeme použít vhodný mikrofon pro detekci přílišného hluku. Jako výstupy lze použít zvukový alarm pomocí bzučáku a LED indikaci.



Obrázek 3.1: Blokové schéma monitorovacího zařízení

3.2 Způsoby detekce měřených veličin

V rámci měření je nutné veličiny změřit vhodným senzorem, který je převede z fyzikální formy na elektronický signál, jenž se dá dále zpracovat pomocí mikrokontroleru, dále následuje technický popis senzorů pro individuální veličiny:

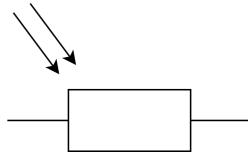
3.2.1 Detekce rozsvíceného světla

Pro detekci rozsvíceného světla potřebuje vhodnou součástku, která převede světelný tok (lm) na vhodný elektrický signál.

Fotorezistor také známý jako fotočlánek nebo LDR, je elektrická součástka, která mění svůj odpor v závislosti na intenzitě okolního světla. Jeho fungování je založeno na jevu zvaném fotoelektrický jev. Čím více světla na fotorezistor dopadá, tím vzniká více volných elektronů a zvyšuje se tím elektrická vodivost.

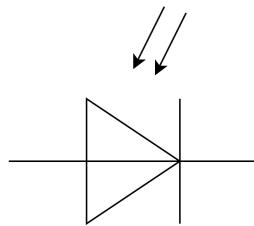
Jednou z hlavních výhod fotorezistorů je jejich jednoduchá konstrukce a nízká cena. Jsou často používány v různých elektronických zařízeních a aplikacích, jako jsou světelné senzory, automatické osvětlení, fotometrie a měření intenzity světla.

Fotorezistory mají své omezení, jako je jejich pomalá odezva na změny osvětlení a nelineární charakteristika odezvy, která vyžaduje kalibraci pro přesnější měření. Nicméně, díky své jednoduchosti a nízké ceně jsou stále oblíbenou volbou pro mnoho aplikací, kde není vyžadována extrémní přesnost. [7]



Obrázek 3.2: Schématická značka fotorezistoru

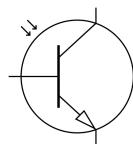
Fotodioda Funguje na principu PN přechodu, je-li upraven tak, aby na něj mohlo dopadat optické záření, je možné takto upravenou diodu využít jako fotocitlivou součástku. Používanými polovodičovými materiály jsou křemík (Si), gallium arsenid (GaAs), arsenid india (InAs) a další. Často se používají k měření osvětlení, snímání dat, rychlé fotodiody se používají v optických spojích. [8]



Obrázek 3.3: Schématická značka fotodiody

Fototranzistor Jedná se o typ tranzistoru, kde je možné průchod nosičů náboje řídit intenzitou dopadajícího optického záření narozdíl od proudu tekoucího do báze jako u běžného bipolárního tranzistoru. Kvůli jejich složitější struktuře je jejich odezva na dopadajícího světlo pomalejší než u fotodiody.

Zapojení je obdobné jako u bipolárních tranzistorů, buďto může být využit jako zesilovací prvek, kde velikost výstupního signálu je od určité úrovně intenzity osvětlení přímo uměrná s velikostí vstupního signálu než tranzistor dojde do saturace (stav, kdy je tranzistor maximálně otevřen), nebo ho lze použít ve spínacím režimu, kde výstupní signál je nulový dokud tranzistor nepřejde při velkém osvětlení do saturace. [9]



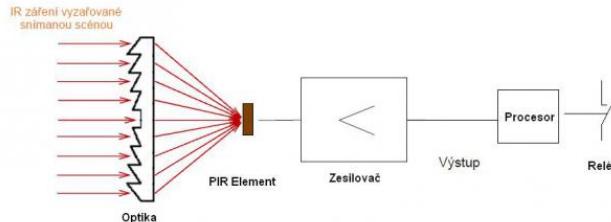
Obrázek 3.4: Schématická značka fototranzistoru

Tabulka 3.1: Porovnání fotorezistoru, fotodiody a fototranzistoru

	Fotorezistor	Fotodioda	Fototranzistor
Princip funkce	Změna odporu v závislosti na intenzitě světla	PN přechod citlivý na světelné záření	Emitorový přechod je otvíráno osvětlením
Citlivost	Nízká	Vysoká	Vysoká
Rychlosť odeadzvy	Pomalá	Rychlá	Rychlá
Použití	Jednoduché aplikace s nižšími nároky na přesnost	Měření intenzity světla, komunikace, snímání	Detekce světla, při nízkých světelných úrovních

3.2.2 Detekce pohybu osob

Pro detekci pohybu osob lze použít PIR (passive infrared detector) čidlo, jenž fungují na principu pyroelektrického jevu, kdy můžeme změnou teploty vyvolat deformaci krystalických mřížek dielektrik. V látkách s jednou polární osou symetrie lze změnou teploty vytvářet dipólový moment, který je přímo úměrný se změnou teploty. [10]



Obrázek 3.5: Blokové schéma PIR detektoru [10]

PIR Element je základní funkční prvek PIR detektoru, polovodičovou součástku ze sloučenin na bázi lithia a tantalu. Tyto detektory jsou citlivé na ozáření infračerveným světlem tak, že začnou generovat elektrický povrchový náboj Q , jehož hodnota závisí dle dopadu infračerveného záření na povrch pyroelektrického materiálu, měřený vestavěným citlivým FET tranzistorem. Tento snímač je velice citlivý ve velkém vlnovém rozsahu a proto je před ním aplikován filtr záření, jenž propouští infračervené záření o vlnových délkách v rozsahu 8 až 14 μm , kdy lidské tělo emituje do prostoru záření o vlnové délce 9,4 μm . [10]

3.2.3 Měření teploty

Senzory teploty lze rozdělit do dvou různých skupin:

- Dotykové senzory: Tyto senzory vyžadují fyzický kontakt s měřeným objektem nebo prostředím a využívají přenosu tepla mezi dvěma objekty. Jsou vhodné v situacích, kdy je snadný

přístup k měřenému objektu nebo kdy měřený objekt chemicky nebo jiným způsobem nereaguje se senzorem. Dotykové senzory mají typicky menší teplotní rozsah, obvykle v řádech stovek stupňů Celsia. Patří mezi ně kovové a polovodičové odporové senzory.

- Bezdotykové senzory: Tyto senzory se nacházejí v určité vzdálenosti od měřeného objektu a využívají takzvané pyrometrie, kdy každý objekt o určité teplotě vyzařuje určitou vlnovou délku infračerveného záření.

3.2.4 Detekce cigaretového kouře

Pro detekci kouře se používají senzory různých typů, například: [11]

- Ionizační - Ke komoře je připojený zdroj napětí, kde se komora v klidovém stavu chová jako zdroj proudu. Pokud se do prostoru komory dostane kouř, změní se hodnota protékajícího proudu obvodem.
- Fotoelektrické - Většinou mají LEDku jenž slouží jako zdroj světla, jehož paprsky buď kouř přeruší nebo odkloní na samotný přijímač. (T komora)
- Chemicko rezistivní - senzory na bázi oxidu křemičitého. Snímač se musí nejdříve ohřát elektrickým proudem, poté jakmile je senzor vystavený kouři, tak dojde ke změně jeho odporu.

Problém těchto senzorů je ten, že jsou citlivé na mnoho typů prvků a nedokážou přesně rozpoznat o jaký typ se jedná. S tím je při detekci cigaretového kouře nutné počítat.

3.2.5 Měření hluku

Pro měření hluku je vhodné použití mikrofonu, ty se dělí na několik typů podle jejich konstrukce: [12]

- Elektrodynamický mikrofon (cívkový) - Používá se cívka pohybující se v magnetickém poli permanentního magnetu, ta je spojena s membránou, která jí mechanicky rozkmitává.
- Elektromagnetický mikrofon - Využívá se změny magnetického toku protékajícího kotvou, který indukuje napětí v pevné cívce obklopující kotvu. Jedná se o rychlostní snímač, má většinou malé rozměry a hmotnost při poměrně dobré citlivosti. Jeho citlivostní rozsah je ale velice omezený a hodí se především na dorozumívací účely.
- Elektretový mikrofon - Jedná se o typ kondenzátorového mikrofonu, kde polarizace není reaktivována externím polarizačním napětím, ale je tvořena elektretem. Nevodivou hmotou která zůstává permanentně elektricky nabité, z níž je vyrobena membrána. Přímo v mikrofonní kapsli je umístěn předzesilovač s vysokým vstupním odporem FET.

Kapitola 4

Výběr komponent a návrh HW zařízení

Jako výběr vhodného mikrokontroleru jsem se rozhodoval mezi ESP32 a Raspberry Pi Pico, nakonec jsem zvolil ESP32 díky jeho výkonnějšímu dvoujádrovému procesoru s frekvencí až 240 MHz, je tedy vhodnější v rychlosti a výkonu pro náročnější aplikace. Má také lepší nabídku periférií, včetně podpory pro Wi-Fi, Bluetooth, BLE a Ethernet. Díky tomu lze ESP32 používat pro mnoho komplexních projektů.



Obrázek 4.1: ESP32 DevKit [13]

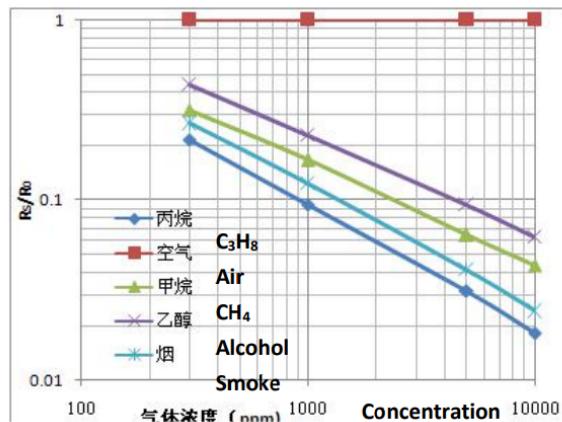
4.1 Použité senzory a snímače

4.1.1 Detektor kouře MQ2

Jedná se o polovodičový typ senzoru, jehož snímací prvek je vyroben z oxidu cíničitého. Jeho odpor se sníží v závislosti na přítomnosti plynu na který je prvek citlivý, tuto změnu můžeme pomocí vhodného obvodu změřit. Na obrázku 4.3 lze vidět poměrové charakteristiky odporu:



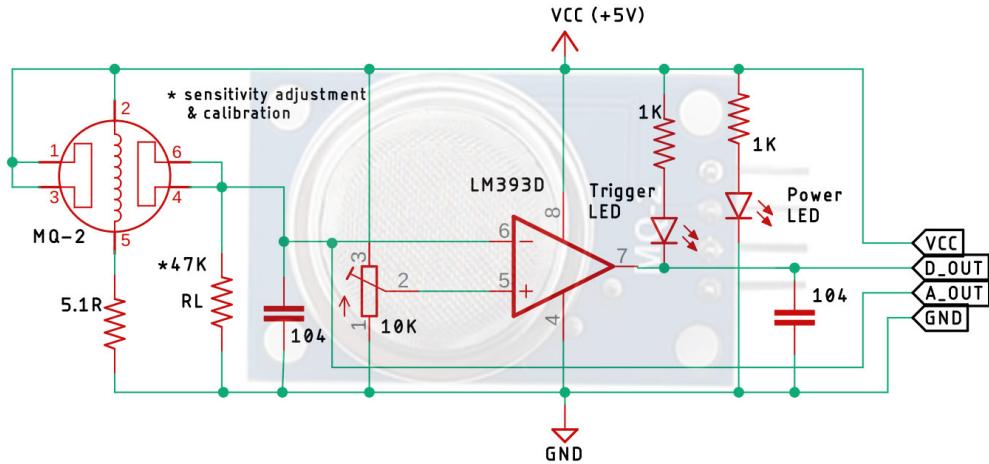
Obrázek 4.2: Senzor MQ2 [14]



Obrázek 4.3: Charakteristiky senzoru MQ2 [14]

Kde R_o je odpor senzoru na čistém vzduchu a R_s je odpor senzoru v závislosti na koncentraci daného plynu. Tento senzor se často prodává už jako samostatný modul, jenž obsahuje už veškerou požadovanou elektroniku pro napájení a zpracování signálu. To je znázorněno na obrázku: 4.4

Samotné zapojení modulu má analogový vývod ze snímače, jenž je samostatně zapojený do výstupního analogového pinu a zároveň do invertujícího vstupu komparátoru LM393D. Pomocí trimru připojeného do neinvertujícího vstupu lze nastavit hladinu, kdy se komparátor sepne. Jeho výstup funguje v režimu vysoké impedance nebo logické nuly, tedy pro jeho provoz je nutné mít k němu připojený pull-up rezistor se zdrojem napětí.



Obrázek 4.4: Schéma zapojení modulu MQ2 [15]

4.1.2 Senzor teploty a vlhkosti HTU21D

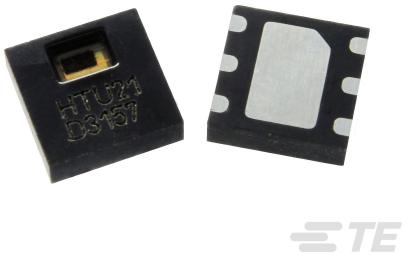
Jedná se o digitální senzor vlhkosti s teplotním výstupem v DFN pouzdře. Lze s ním měřit teploty od -40°C do 125°C , s tím že umí měřit relativní vlhkost od 0 až po 100%. Napájecí napětí jsou 3V a se senzorem lze komunikovat pomocí I2C sběrnice, modul již obsahuje zabudované pull-up rezistory. Rozlišení měřených hodnot lze nastavit, od 8 do 12 bitů pro relativní vlhkost a od 11 do 14 bitů pro měření teploty. S vyšším rozlišením naruštá doba potřebná pro měření. Rozlišení a další parametry se dají nastavit v uživatelském konfiguračním registru senzoru. [16]

Bit	#Bits	Description/Coding	Default																				
7,0	2	Measurement resolution <table border="1"> <tr> <th>Bit 7</th><th>Bit 0</th><th>RH</th><th>Temp</th></tr> <tr> <td>0</td><td>0</td><td>12 bits</td><td>14 bits</td></tr> <tr> <td>0</td><td>1</td><td>8 bits</td><td>12 bits</td></tr> <tr> <td>1</td><td>0</td><td>10 bits</td><td>13 bits</td></tr> <tr> <td>1</td><td>1</td><td>11 bits</td><td>11 bits</td></tr> </table>	Bit 7	Bit 0	RH	Temp	0	0	12 bits	14 bits	0	1	8 bits	12 bits	1	0	10 bits	13 bits	1	1	11 bits	11 bits	'00'
Bit 7	Bit 0	RH	Temp																				
0	0	12 bits	14 bits																				
0	1	8 bits	12 bits																				
1	0	10 bits	13 bits																				
1	1	11 bits	11 bits																				
6	1	Status: End of Battery ⁽¹⁾ '0': VDD>2.25V '1': VDD<2.25V	'0'																				
3, 4, 5	3	Reserved	'0'																				
2	1	Enable on-chip heater	'0'																				
1	1	Disable OTP reload	'1'																				

⁽¹⁾ This status bit is updated after each measurement

Obrázek 4.5: Uživatelský registr HTU21D [16]

Tabulka na obrázku: 4.5 popisuje význam bitů v uživatelském registru, nultý a sedmý bit slouží pro konfiguraci rozlišení měření, šestý bit slouží pro indikaci podpětí (Napájení je nižší než 2,25 V), pátý, čtvrtý a třetí jsou rezervovány a nesmí se do nich zapsiovat. Druhý bit slouží pro zapnutí integrovaného ohříváče na čipu, který zvýší hodnotu měřené teploty o $0,5$ až $1,5^{\circ}\text{C}$, takže můžeme otestovat funkčnost senzoru. První bit slouží pro vypnutí OTP načtení, které pokud není vypnuto,



Obrázek 4.6: Senzor HTU21D
[16]

tak senzor před každým měřením použije tovární nastavení. Toto použití ale výrobce nedoporučuje a pro obnovu do továrního nastavení by se měl použít příkaz pro reset čipu.

Získaná hodnota při komunikaci s čipem se musí následně přepočítat, v případě teploty S_{Temp} abychom jí získali ve stupních Celsia následujícím vzorcem:

$$T_{\text{teplota}} = -46,85 + 175,72 \cdot \frac{S_{\text{Temp}}}{2^{16}} \quad (4.1)$$

Pro získanou hodnotu relativní vlhkosti S_{RH} je výpočet relativní vlhkosti v procentech následující:

$$RV = -6 + 125 \cdot \frac{S_{RH}}{2^{16}} \quad (4.2)$$

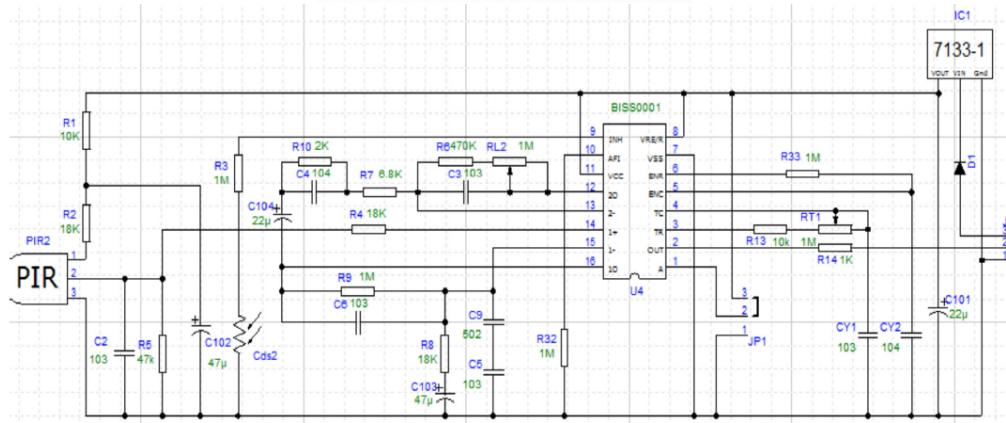
4.1.3 PIR detektor pohybu HC-SR501

PIR modul HC-SR501 je kompaktní senzor pohybu, který využívá technologii pasivní infračervené detekce k identifikaci pohybu v jeho okolí. Tento modul je oblíbený pro svou jednoduchost použití, nízkou spotřebu energie a vysokou spolehlivost.

Samotný modul má na desce nainstalované dva trimry, kde ten první slouží pro nastavení vzdálenosti citlivosti a druhý pro nastavení doby intervalu necitlivosti, kdy po zaznamenání pohybu senzor ignoruje jakýkoliv další pohyb.

4.1.4 Mikrofonní modul se zesilovačem MAX4466

Tento modul má vestavěný elektretový mikrofon se zesilovačem MAX4466, jenž je přímo navržen pro aplikace tohoto typu mikrofonů, dokáže snímat zvuk ve frekvenčním pásmu od 20 Hz do 20 kHz. Na desce se nachází i vestavěný trimr, pomocí něho lze měnit zesílení signálu z mikrofonu



Obrázek 4.7: PIR modul HC-SR501 a schéma [17] [20]

pomocí zpětné vazby. V provozu je nutné dát pozor na fakt, že výstupní audio signál bude poloviční napájecímu napětí modulu.

4.2 Schéma zapojení

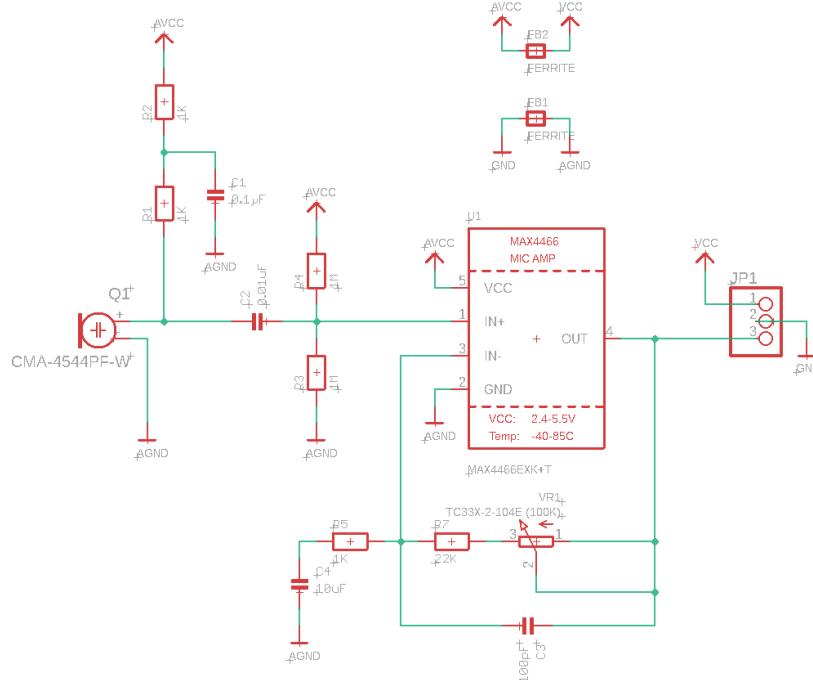
Po výběru vhodných senzorů bylo nutné realizovat schéma zapojení, jenž zároveň poskytne potřebné napájení a sběrnice pro mikrokontroler, senzory a další prvky na desce, ale také rozhraní pro samotné programování mikrokontroleru.

Srdcem zapojení je step-down měnič SD8942, jehož zapojení slouží pro dodávání stabilního výstupního napětí 5V pro zbytek obvodu, který je znázorněn na obrázku 4.9

Výstupní napětí měniče závisí na poměru odporů napěťového děliče připojeného do pinu FB, dle vzorce. 4.3

$$V_{out} = 0,6 * \left(\frac{R_{13}}{R_{14}} + 1 \right) = 0,6 * \left(\frac{110000}{15000} + 1 \right) = 4,98V \quad (4.3)$$

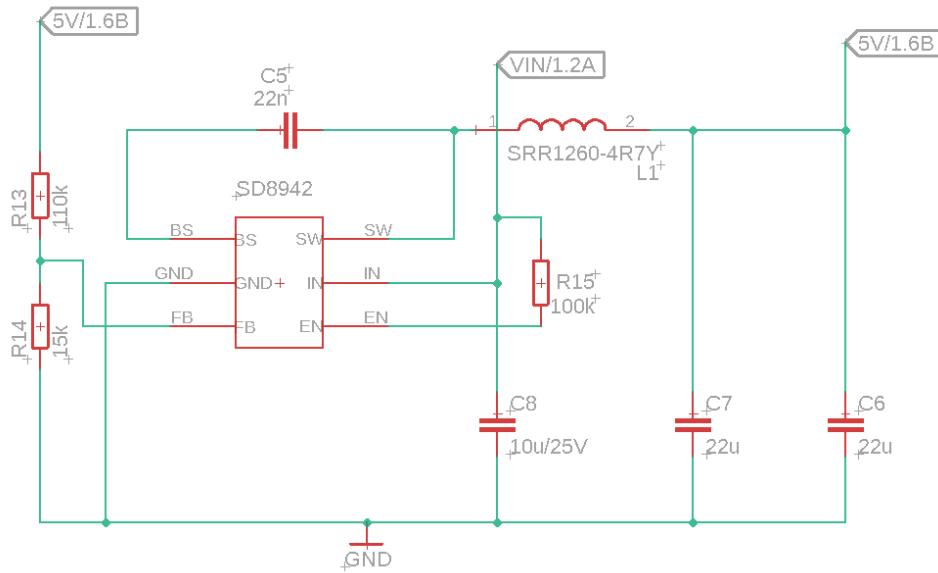
Pro napájení mikrokontroleru ESP32 je použit stabilizátor AMS1117, jenž dodává stabilní napětí 3,3V.



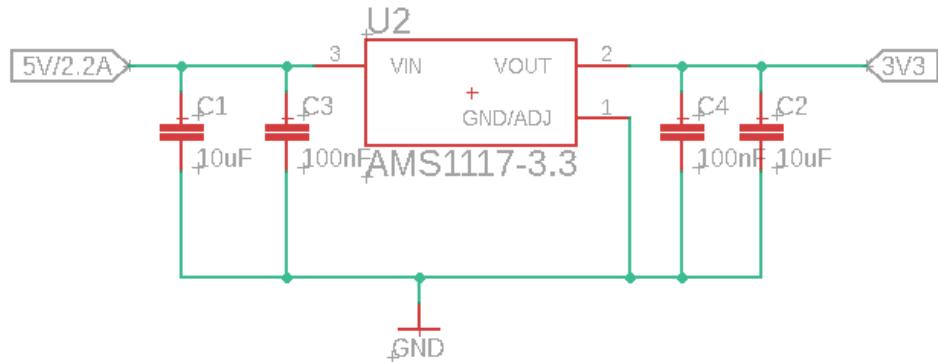
Obrázek 4.8: Schéma mikrofonního zesilovače [19]

Samotné zapojení mikrokontroleru na obrázku 4.11 obsahuje pull-up rezistory s tlačítkem na pinech EN (reset mikrokontroleru) a IO0 (tentu pin musí být na logické 0 při startu procesoru, abychom ho mohli přivést do programovacího režimu). Programování probíhá pomocí sériové linky, zapojení pak obsahuje popis pro jaký účel byl každý pin použit.

Pro mikrofon, fototranzistor a teplotní senzor je napájení přímo navedeno z GPIO pinů mikrokontroleru ESP32. Takhle se dá v případě potřeby deaktivovat napájení těchto prvků. Dále jsou piny pro komunikaci, tedy z I2C sběrnice z teplotního senzoru (ten již obsahuje pull up rezistory), z MQ2 senzoru piny pro analogový a digitální výstup, pro mikrofon a fototranzistor je přímo napojený A/D převodník. Dále jako vstup lze použít tlačítko nebo komunikovat s mikrokontrolem pomocí sériové linky. Jako výstup lze použít bzučák nebo LED. Také jsou připojeny tlačítka na piny EN (reset



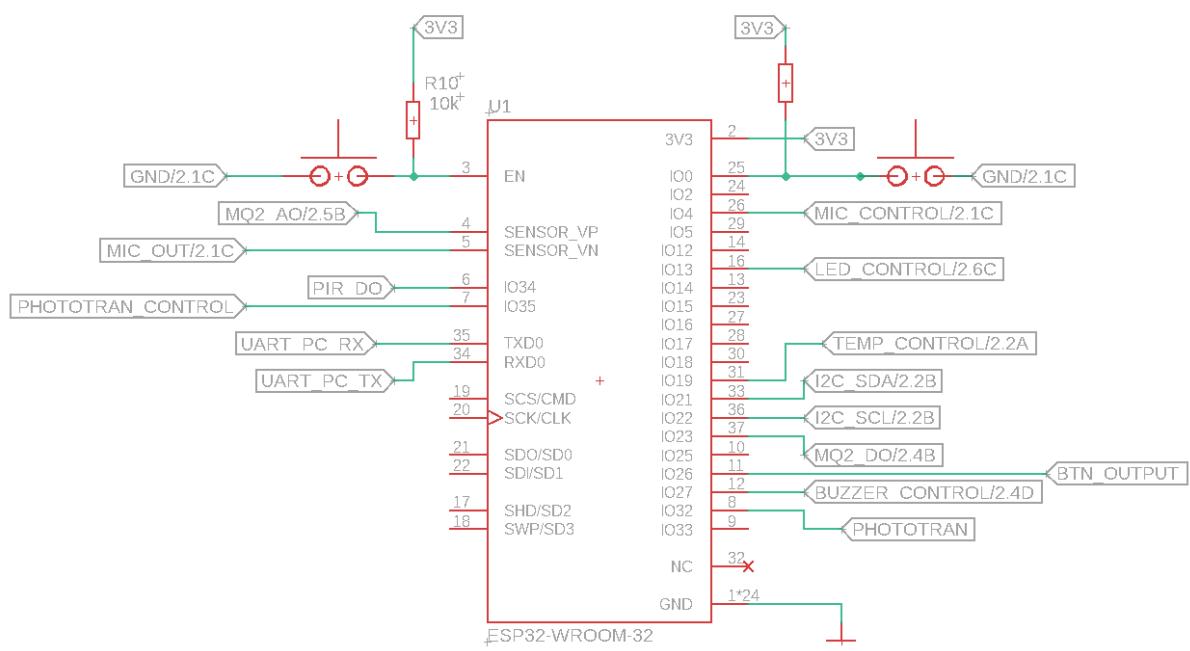
Obrázek 4.9: Zapojení step down měniče SD8942



Obrázek 4.10: Zapojení stabilizátoru AMS1117

mikrokontroleru) a IO0, jenž musí být při zapnutí mikrokontroleru na úrovni logické nuly, pokud chceme vstoupit do programovacího režimu.

Kompletní zapojení se nachází v příloze, za zmínku ještě stojí použité napěťové děliče pro MQ2 senzor, jenž převádí signál na 3,3V úroveň, kterou je mikrokontroler na svých digitálních vstupech schopen zvládnout.

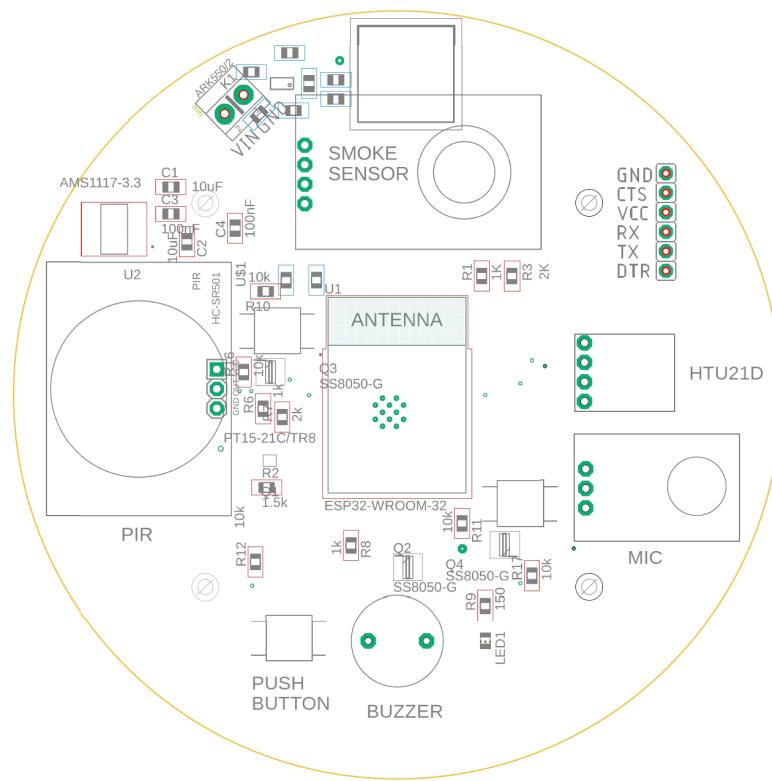


Obrázek 4.11: Zapojení mikrokontroleru ESP32

Kapitola 5

Realizace HW prototypu zařízení

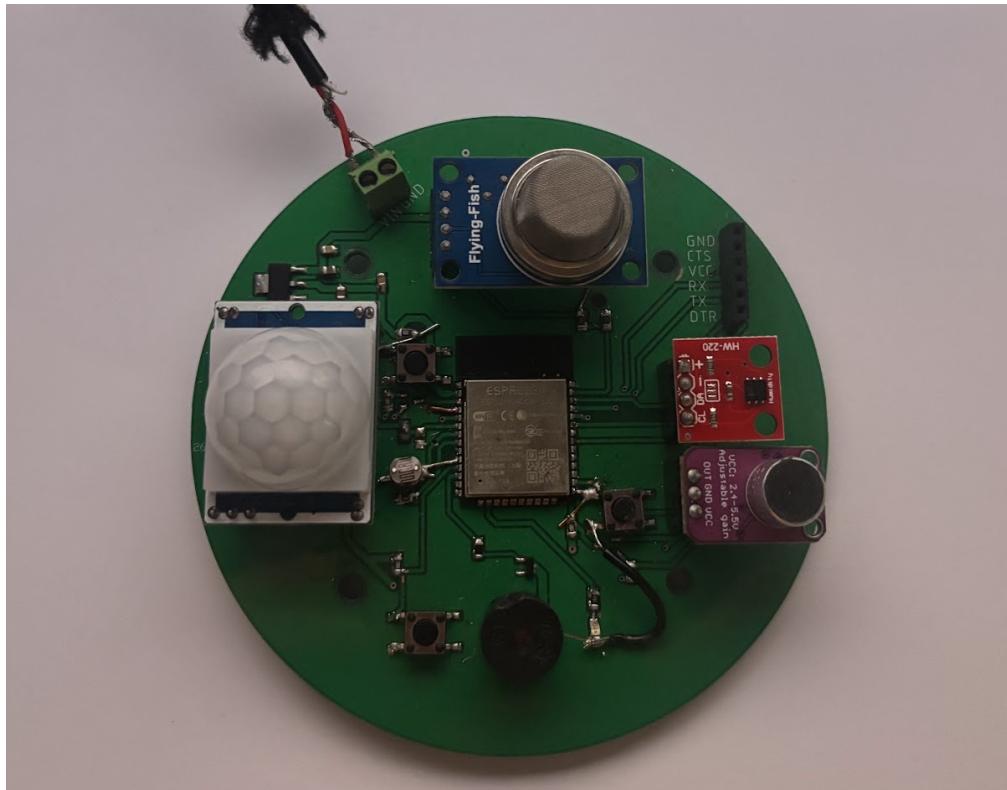
Pro realizací HW prototypu byl navržený plošný spoj v kotoučovém tvaru, jenž je především osazen SMD součástkami. Plošný spoj byl vyroben na zakázku a provedl jsem ruční osazování všech součástek kromě mikrokontroleru.



Obrázek 5.1: Osazení součástek plošného spoje

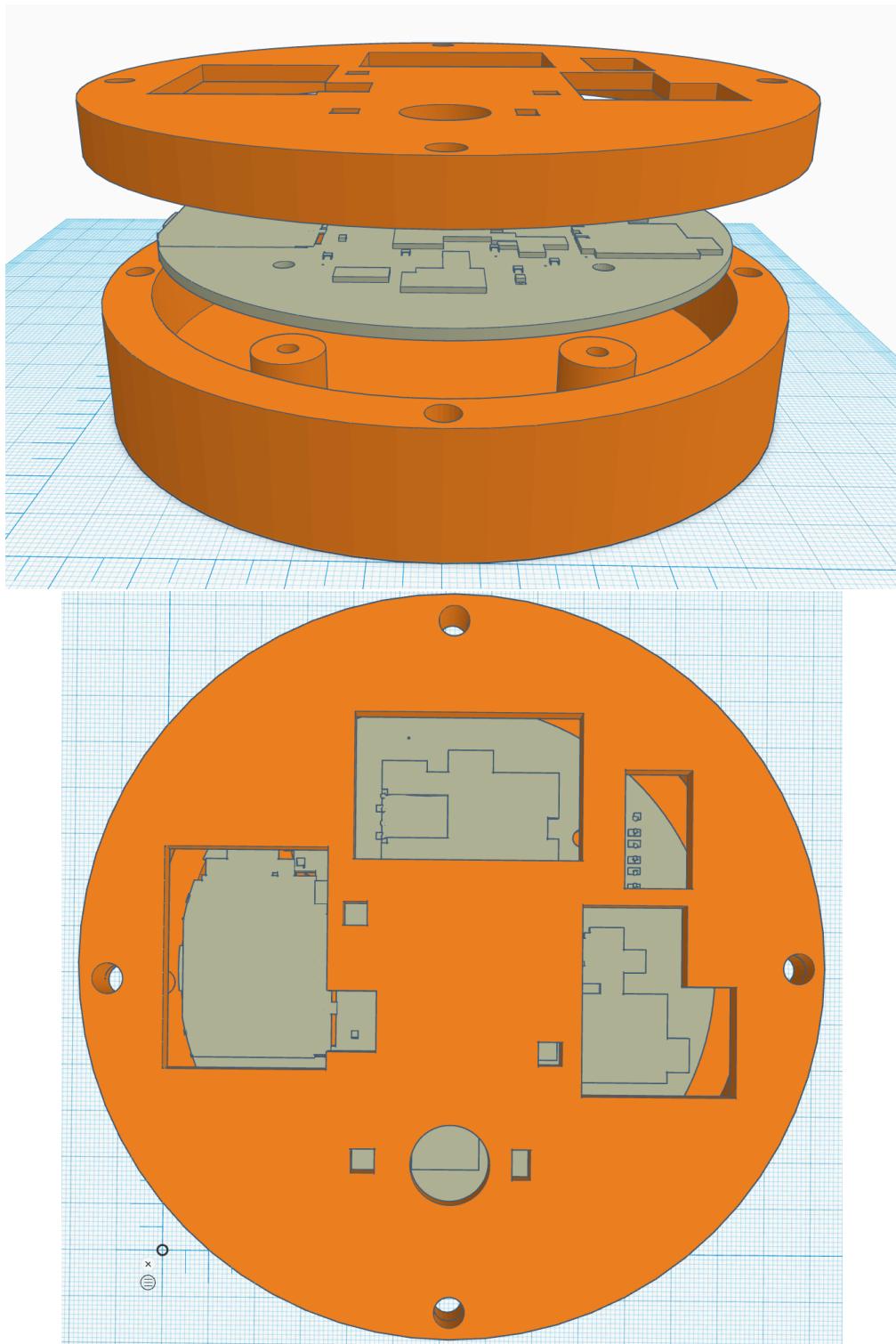
Při následném oživováním bylo zjištěno několik problémů, nejzávažnějším byl nefunkční programátor, kde reset pomocí UART převodníku nefungoval. Rozhodl jsem se tedy tranzistory pro vstup

do programovacího režimu odstranit a nechat pro vstup do programovacího režimu jenom tlačítka, které fungují spolehlivě. Další chybou bylo připojení napájení fototranzistoru na pin 35, ten slouží pouze jenom jako vstup. Na prototypu tato chyba se vyřešila přemostěním na kompatibilní pin. Poté jsem zjistil že fototranzistor nedostatečně reagoval na změny světla v okolí a vyměnil jsem ho za fotorezistor s hodnotou odporu 5 - 10 k Ω na světle, který má pro požadovanou funkcionalitu lepší parametry.



Obrázek 5.2: Finální podoba plošného spoje po osazení

Na 3D tiskárně jsem navrhl jednoduché pouzdro, které je rozdělené na dva díly. Slouží především pro usazení plošného spoje s dostatečnými otvory na vrchní straně, aby se dal snadno připojit UART převodník k programování mikrokontroleru a senzory měly dostatečné velké otvory při snímání veličin.



Obrázek 5.3: Pohled na pouzdro plošného spoje z boku a seshora

Kapitola 6

Implementace SW zařízení pro monitoring ubytovacích prostor

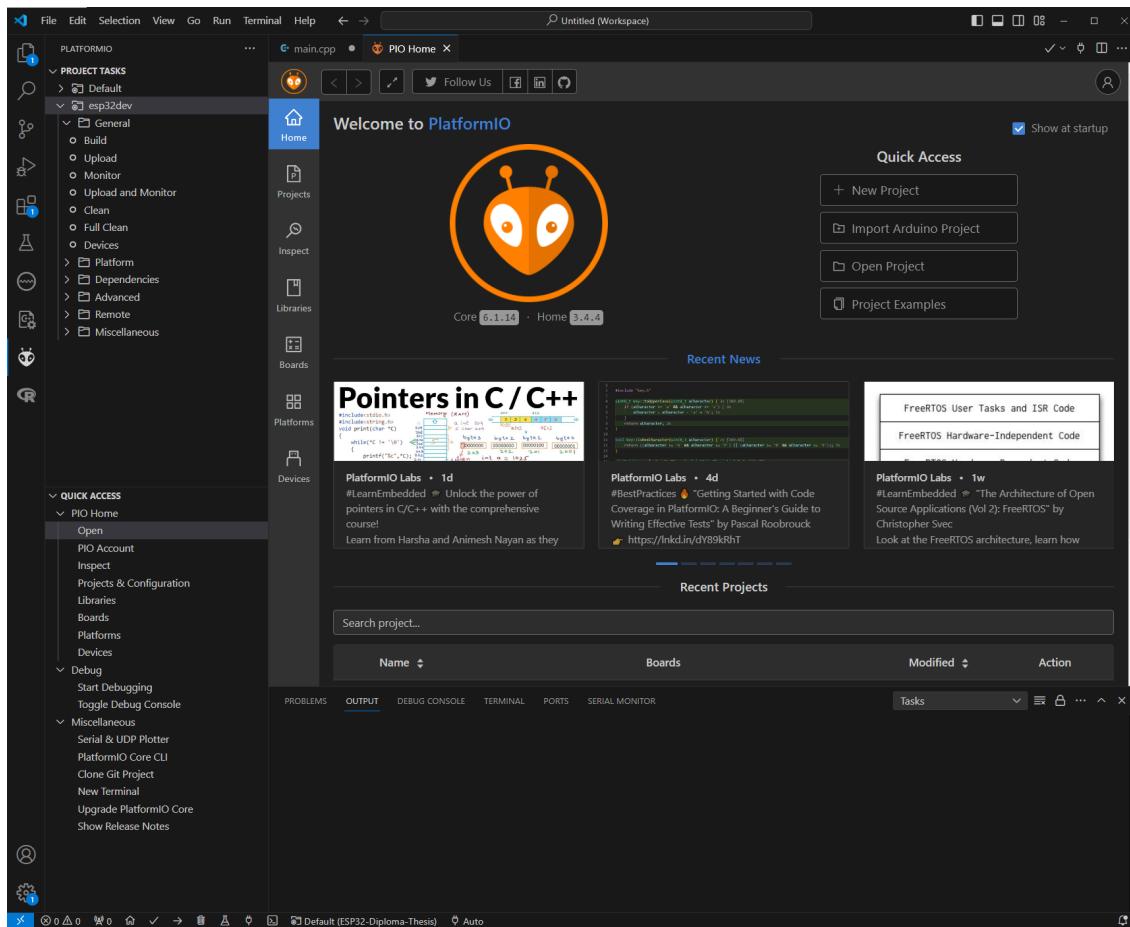
EPS32 lze programovat v mnoha programovacích jazycích, s tím že nejrozšířenějšími jsou v současnosti C (ESP-IDF) a C++ (Arduino IDE), poté existují další možnosti, jako MicroPython, Lua a další.

- ESP-IDF (C) - prostředí navržené přímo výrobcem, s vlastními knihovnami implementovanými pro Jazyk C. Výhodou je velká rychlosť jazyka, nevýhodou je fakt, že jazyk C je procedurální programovací jazyk jenž vznikl již v roce 1972 a neobsahuje funkce dnes moderních programovacích jazyků jako podporu objektově orientovaného programování.
- Arduino IDE (C++) - platforma známá především jako řada vývojových desek založených na mikrokontrolerech Atmel, tak má dnes velkou komunitu po celém světě a podpora tohoto prostředí se rozšířila i na ESP32. Výhodou je možnost použití spoustu již implementovaných knihoven, využívajících funkcí OOP, nicméně vyšší míra abstrakce je vykoupena nižším výkonom zkompilovaného kódu. [13]

Jako programovací jazyk jsem tedy zvolil C++, s tím jako vývojové prostředí jsem použil Platform IO ve Visual Studio Code, jenž má víc funkcionalit než pouhé vývojové prostředí Arduina a umožňuje snažší vývoj.

6.1 Úložiště pro naměřené data - ThingSpeak

Jako úložiště pro naměřená data byla zvolena platforma ThingSpeak od výrobců Matlabu, ta umožňuje nejen uchování dat, ale následnou možnost na nich spustit skript napsaný v Matlabu a i nastavit funkce, kdy uživatel může být informován že došlo k nějaké události (detekován pohyb, překročena hodnota nějaké veličiny).



Obrázek 6.1: Prostředí Platform IO ve Visual Studio Code

Pro zprovoznění prostředí stačí mít registrovaný u Mathworksu účet a poté lze vytvořit vlastní kanál, na který lze posílat z libovolného zařízení přes HTTP REST protokol poslat data. Velkou výhodou u ESP32 jsou již existující knihovny, pomocí kterých lze s ThingSpeakem pracovat relativně snadno.

Jeden kanál ThingSpeaku umožňuje mít najednou až 8 polí, kde se dají ukládat data pro individuální naměřené veličiny.

Poté na výpisu kódu: 6.1 je příklad metody, jenž posílá data na ThingSpeak. Metoda `setField()` slouží k nastavení hodnoty pro dané pole a metoda `writeField()` pošle zvolené data na ThingSpeak, proto je nutné mít zvolené správné číslo kanálu a klíč pro zápis na API.

Výpis kódu 6.1: Příklad metody jenž posílá data na ThingSpeak

```
void Measurement::SendMeasurement(
    unsigned long myChannelNumber,
    const char* myWriteAPIKey)
```

```

{
    ThingSpeak.setField(TEMPERATURE_FIELD, CalculateAverage(temperature));
    ThingSpeak.setField(HUMIDITY_FIELD, CalculateAverage(humidity));
    ThingSpeak.setField(AUDIO_FIELD, CalculateAverage(audio));
    ThingSpeak.setField(GAS_FIELD, CalculateAverage(gas));
    ThingSpeak.setField(PHOTORESISTOR_FIELD, CalculateAverage(photoResistor));
    ThingSpeak.setField(PIR_DETECTION_FIELD, wasMovementDetected);
    int result = ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);

    Serial.println("Result of sending data to ThingSpeak");
    Serial.println(result);
}

```

6.2 Návrh a realizace SW

Ve vývojovém diagramu na obrázku: 6.2 je popsána hlavní myšlenka programu, při startu programu dojde ke konfiguraci pinů mikrokontroleru v závislosti na jejich funkcionalitě ve funkci SetupPins.

Výpis kódu 6.2: Kód pro konfiguraci pinů

```

/// @brief Nastavi piny jenz jsou pozadovany pro korektni funkcionalitu.
void SetupPins()
{
    pinMode(TEMP_CONTROL, OUTPUT);
    pinMode(BUTTON_OUTPUT, INPUT);
    pinMode(BUZZER_CONTROL, OUTPUT);
    pinMode(LED_CONTROL, OUTPUT);
    pinMode(PHOTOTRAN_CONTROL, OUTPUT);
    pinMode(MIC_CONTROL, OUTPUT);
    pinMode(PIR_OUTPUT, INPUT);
    pinMode(MQ2_DIGITAL_OUTPUT, INPUT);

    // Zapne napajeni senzorum na desce jenz jsou napajeny pres GPIO
    digitalWrite(MIC_CONTROL, HIGH);
    digitalWrite(PHOTOTRAN_CONTROL, HIGH);
    digitalWrite(TEMP_CONTROL, HIGH);

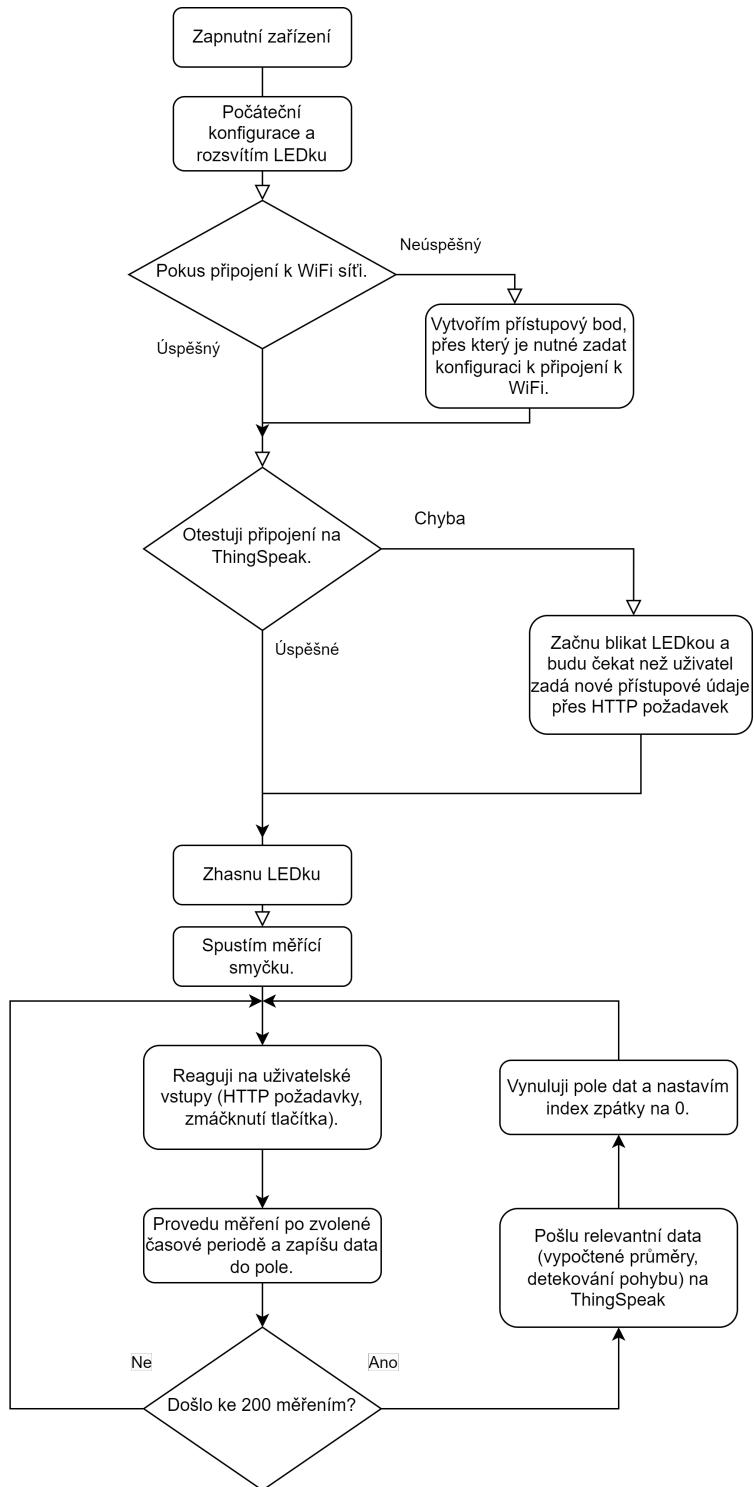
    // Nastav vystupy na logickou nulu

```

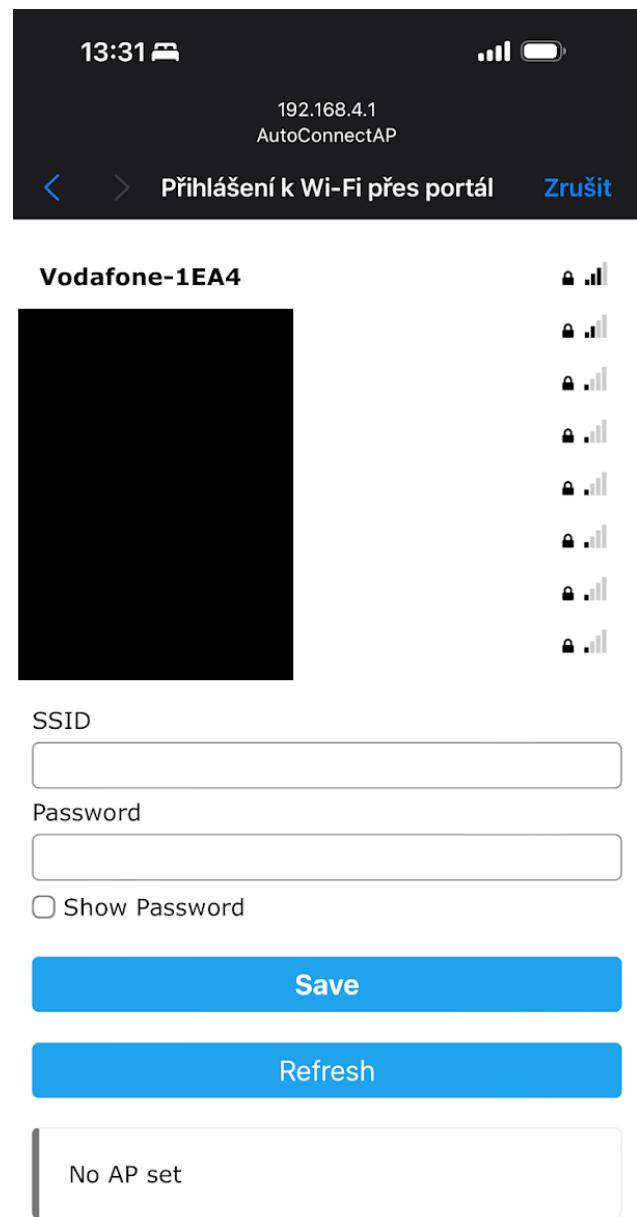
```
digitalWrite(BUZZER_CONTROL, LOW);  
digitalWrite(LED_CONTROL, LOW);  
}
```

Dále v hlavní konfiguraci programu je použita třída WiFiManager, jenž umožňuje ESP32 vytvořit přístupový bod pro konfiguraci WiFi sítě, viz obrázek 6.3 Po následném připojení k WiFi jsou konfigurační údaje uloženy v paměti mikrokontroleru a po restartu se zařízení automaticky připojí. Pro vymazání konfigurace stačí 2 sekundy podržet uživatelské tlačítko, po kterých se mikrokontroler restartuje s prázdnou konfigurací.

Následně probíhá kontrola připojení k ThingSpeak, kdy mikrokontroler pošle hodnotu z PIR čidla na ThingSpeak, pokud ten vrátí jinou HTTP odpověď než 200 (OK), dojde ke vstupu do smyčky s blikáním LEDky a mikrokontroler bude čekat než uživatel pomocí HTTP POST požadavku zadá na adresu <IPAdresaZařízení>/setThingSpeakCredentials přístupové údaje, které jsou pak uchovány v EEPROM paměti.



Obrázek 6.2: Vývojový diagram programu.



Obrázek 6.3: Připojení k WiFi síti přes WiFiManager

Výpis kódu 6.3: Ověření připojení k ThingSpeaku

```
while (!(ThingSpeak.writeField(
    existingChannelNumber,
    PIR_DETECTION_FIELD,
    digitalRead(PIR_OUTPUT),
    &existingApiKey[0]) == OK))
{
    Serial.println("Cannot send message to ThingSpeak,
try to reenter your credentials");
    while (!succesfullCredentials)
    {
        server.handleClient();
        if (millis() >= time_now + 100)
        {
            time_now += 100;
            *(uint32_t*)(GPIO_OUT_REG) ^= (1 << LED_CONTROL);
        }
    }
    break;
}
```

Výpis kódu 6.4: JSON v HTTP requestu jenž slouží pro konfiguraci ThingSpeak údajů

```
{
  "channelNumber": 2501251,
  "apiKey": "WQDRN5TAYSWZNGT3"
}
```

Poté dojde k spuštění hlavní smyčky, kde na začátku vždycky probíhá kontrola, zda nedošlo k příchodu HTTP požadavku nebo zmáčknutí tlačítka, poté pokud došlo k uplynutí vzorkovací periody (100 ms), tak dojdek novému měření. Dále se ověřuje zda senzor kouře nenaměřil nenulovou hodnotu, tím se spustí bzučák jenž je nutné poté tlačítkem deaktivovat. Když provedu 200 měření, dojde k poslání dat na ThingSpeak a vynulování pole naměřených dat, viz výpis kódu: 6.5

Výpis kódu 6.5: Hlavní smyčka programu

```
void loop()
{
    server.handleClient();

    if(buttonPressed)
    {
        EvaluateButtonPress();
    }

    if(millis() - previousMillis >= MEASURING_PERIOD)
    {
        #ifdef DEBUG
        Serial.println("Reading values:");
        #endif
        previousMillis = millis();

        measurement.Measure(htu21);

        if(measurement.index >= ARRAY_SIZE)
        {
            measurement.SendMeasurement(existingChannelNumber, &existingApiKey[0]);
            measurement.WipeMeasurements();
        }
    }
}
```

6.3 Vytvořené třídy

Pro abstrakci některých operací jako měření ze senzorů, komunikaci ze senzorů pomocí I2C atd. jsem si napsal svoje vlastní třídy, které tyto operace implementují.

HTU21D: Třída jenž provádí komunikaci se senzorem HTU21D pomocí I2C:

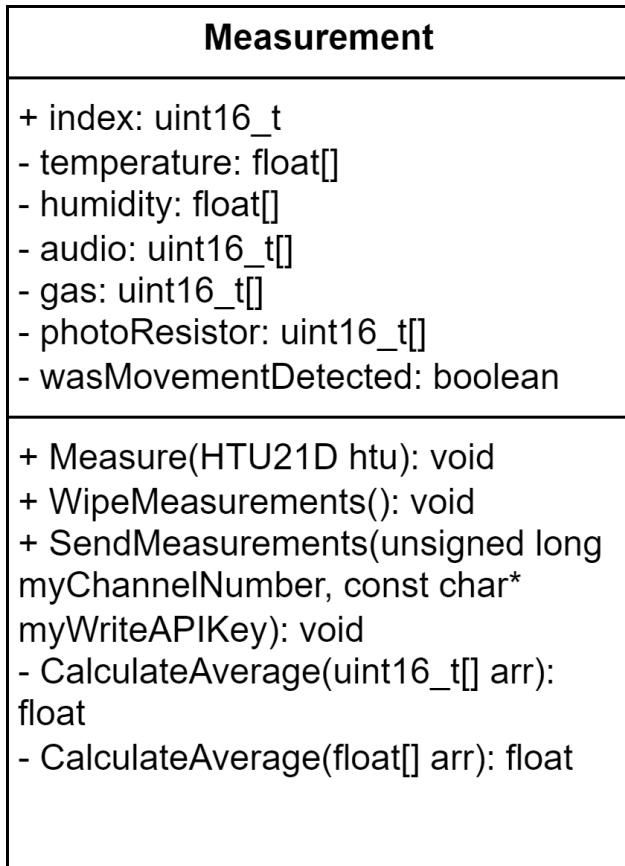
HTU21D
- _i2c_num: uint8_t
+ Begin(uint8_t i2c_num, uint8_t sda, uint8_t scl, uint32_t clk_speed): boolean + Reset(): boolean + ReadUserRegister(): uint8_t + TurnOnOffHeater(boolean value): boolean + ReadTemp(): float + ReadHumidity(): float

Obrázek 6.4: Diagram třídy HTU21D

- Begin() - spustí I2C komunikaci se senzorem.
- Reset() - nastaví senzor do továrního nastavení.
- ReadUserRegister() - vrátí hodnotu uživatelského registru.
- TurnOnOffHeater() - zapnutí/vypnutí ohříváče senzoru.
- ReadTemp() - vrátí změřenou teplotu ve °C.
- ReadHumidity() - vrátí změřenou relativní vlhkost v %.

Measurement: Třída jenž provádí měření ze senzorů a práci s naměřenými daty.

- Measure(HTU21D htu) - Provede jedno měření ze senzorů.
- WipeMeasurements() - Vynuluje všechny pole pro měření na nulu.
- SendMeasurements() - Pošle změřené hodnoty na ThingSpeak.
- CalculateAverage() - Vypočte aritmetický průměr hodnot z pole.



Obrázek 6.5: Diagram třídy Measurement

6.4 Použití funkcí ThingSpeaku

ThingSpeak umožňuje spouštění Matlabovského kódu buď v předem zvolených časech nebo při příjmutí nových dat. Tato funkcionality se dá použít například pro poslání emailu, když dojde k překročení předem definované hodnoty.

Nejdříve je třeba nutné v ThingSpeaku nastavit React, jenž spustí Matlab kód když příchozí hodnota ze senzoru kouře bude vyšší než nula. Ten pak zajistí odeslání emailu jenž je znázorněn na obrázku 6.7

[Edit React](#)

Name:	Smoke Sensor Reaction
Condition Type:	Numeric
Test Frequency:	On data insertion
Last Ran:	2024-04-15 20:35
Channel:	ESP32-Diploma
Condition:	Field 4 (Gas) is greater than 0
MATLAB Analysis:	Poslání emailu při detekci kouře.
Run:	Each time the condition is met
Created:	2024-04-15 8:33 pm

Obrázek 6.6: Nastavení Reactu

Alert: Smoke sensor triggered

Value from smoke sensor: 29.990000
Time: 2024-04-15 20:35:510 :+0000

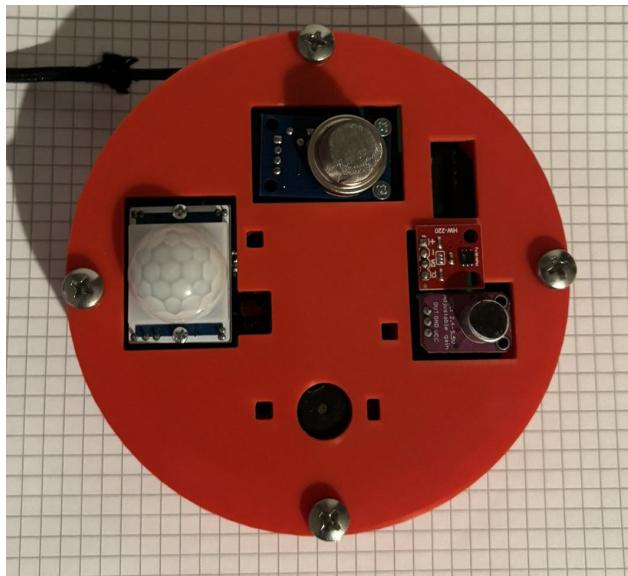
You are receiving this email because a ThingSpeak Alert was requested using your ThingSpeak Alerts API key. For more information please refer to the [ThingSpeak Alerts Documentation](#).

Obrázek 6.7: Odeslaný email z ThingSpeaku.

Kapitola 7

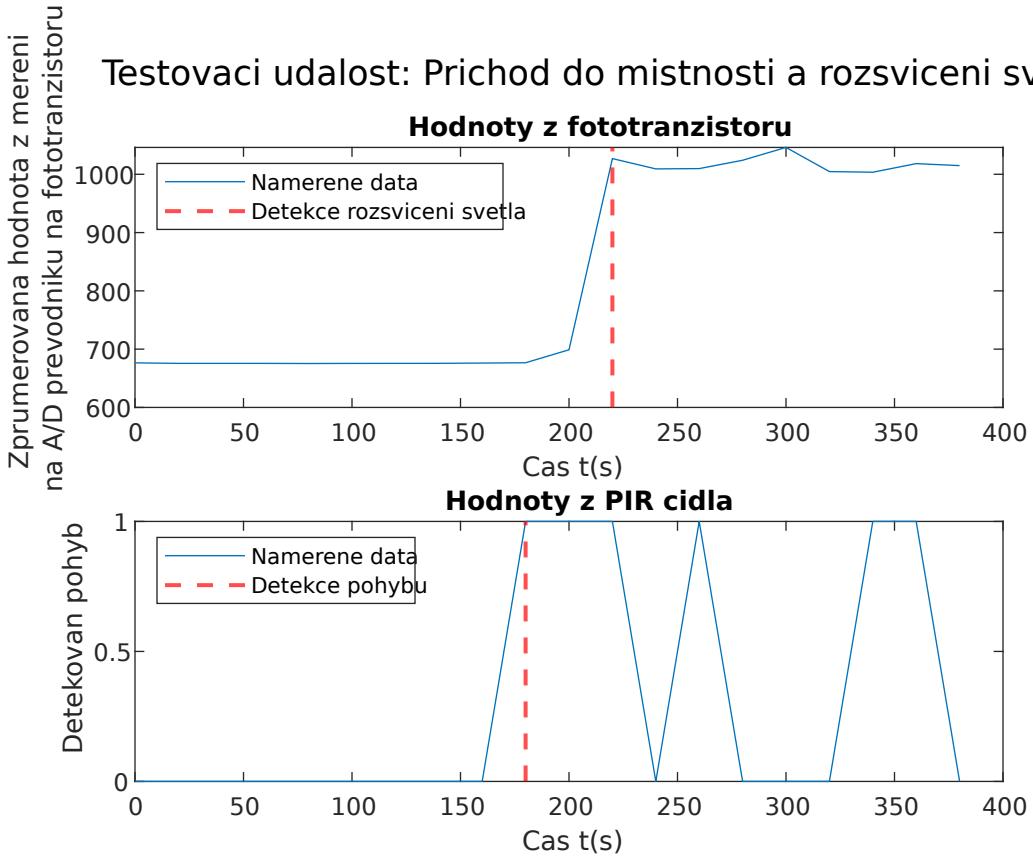
Testování zařízení pro monitoring ubytovacích prostor

Po finální montáži jsem provedl testování monitorovacího zařízení, zařízení bylo položeno na mém pracovní stole, kde jsem testoval události jako příchody, zhasnutí světel, pouštění hudby z reproduktorů a otevření oken.



Obrázek 7.1: Finální montáž prototypu

Příchod do místnosti: Zde bylo vypnuto světlo, dokud nedošlo k příchodu osoby a následnému rozsvícení světel, toto lze potvrdit z naměřených dat na fototranzistoru a z PIR čidla, ty jsou znázorněny na obrázku: 7.2, kde přerušovaná vertikální čára v horním grafu znázorňuje kdy došlo k detekci rozsvíceného světla a v dolním grafu kdy došlo k detekci pohybu.



Obrázek 7.2: Testovací událost: Příchod do místnosti a rozsvícení světla

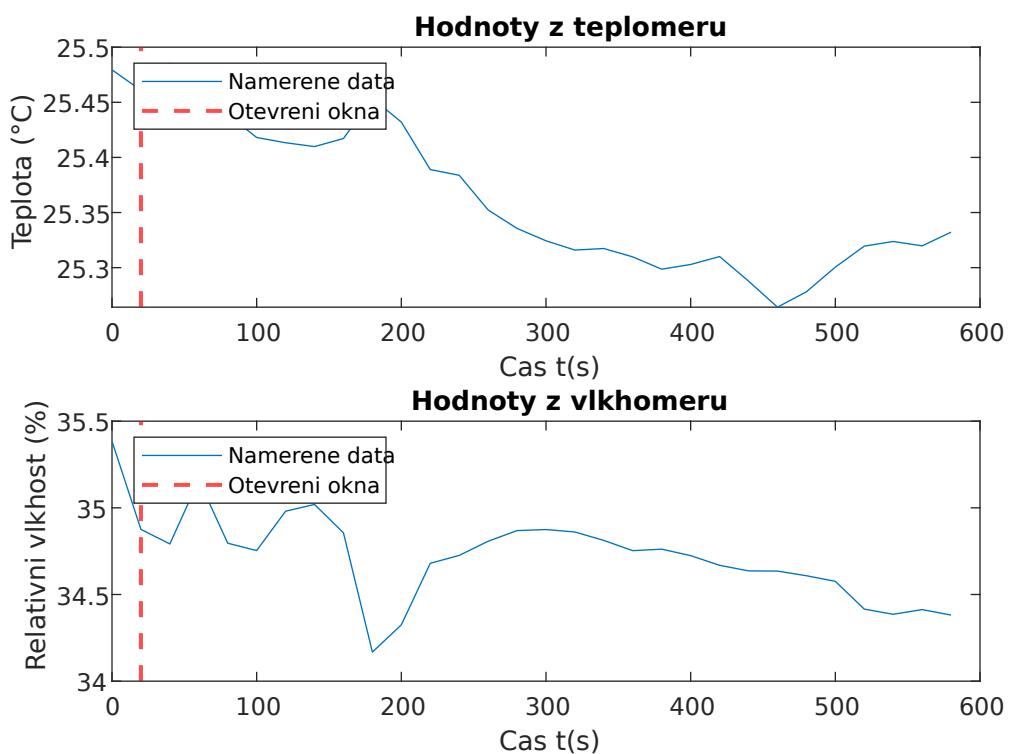
Otevření okna: Zde došlo k otevření okna v místnosti, s předpokladem že se sníží teplota v místnosti, toto lze potvrdit ze získaných dat ze senzoru HTU21D na obrázku: 7.3, kde přerušovaná vertikální čára v získaných datech znázorňuje moment kdy došlo k otevření okna s následným postupním snížením teploty.

Pouštění hudby v reproduktorech: V místnosti došlo v intervalu osmi minut k pouštění hudby přes reproduktory, to lze vidět na obrázku: 7.4. Červená přerušovaná čára se nachází v čase 220s, kdy došlo ke spuštění hudby. V následném intervalu lze vidět významný skok v minimech a maximech získaných hodnot před spuštěním hudby. V čase 700s je vyznačená zelená přerušovaná čára, kde došlo k vypnutí hudby a následnému ustálení získaných dat.

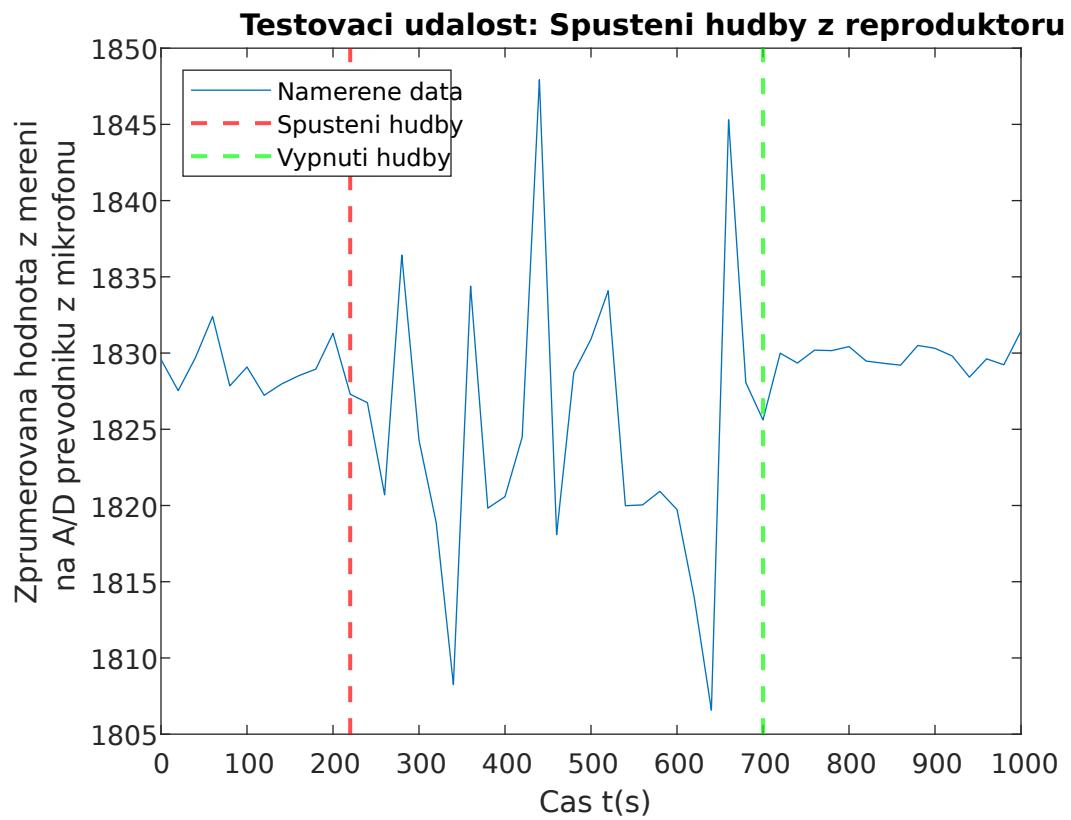
Detekce kouře: Testování probíhalo na balkóně, kde testovací subjekt (kuřák) vydechl cigaretový kouř přímo na zařízení, zařízení bylo tento kouř schopno detektovat, to se projeví špičkou v získaných hodnotách, viz obrázek: 7.5

Pro shrnutí, zařízení je schopné detektovat požadované události (pohyb, změna osvětlení v místnosti, otevření okna, kouření cigarety, pouštění hlasité hudby), ale ne na všechny z nich je schopné reagovat, například na změnu osvětlení by muselo dojít k rozšíření programu.

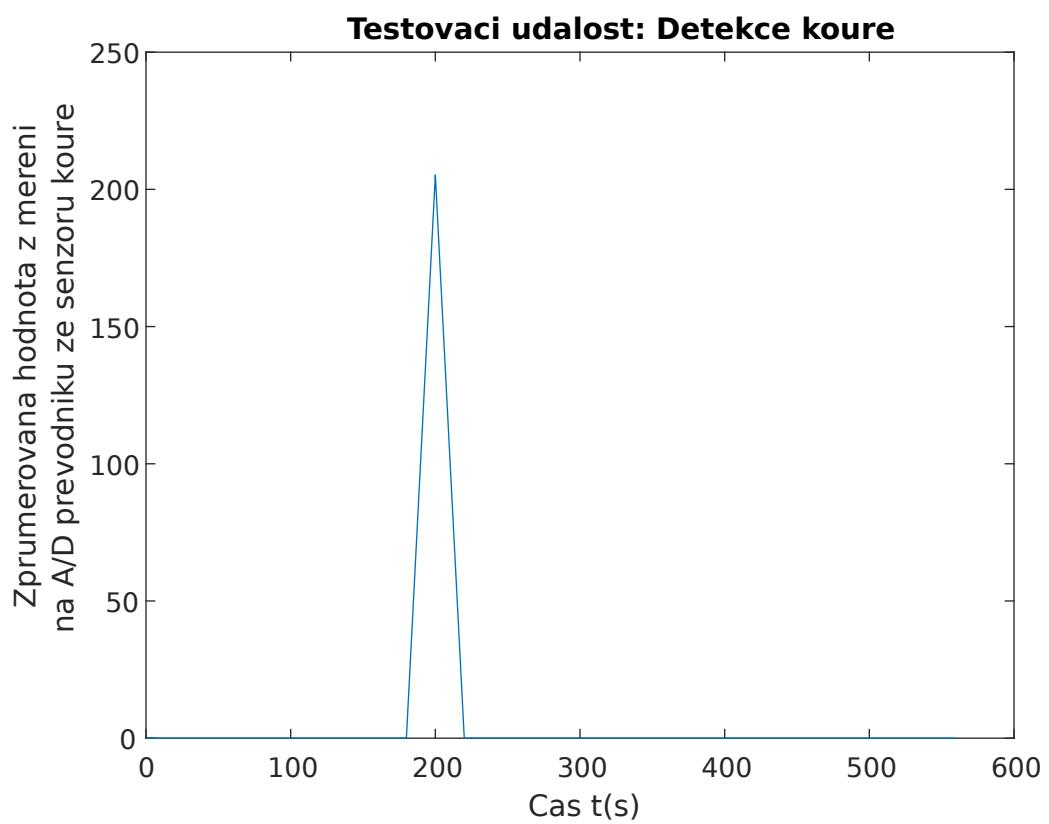
Testovaci udalost: Otevreni okna



Obrázek 7.3: Testovací událost: Otevření okna



Obrázek 7.4: Testovací událost: Spuštění a vypnutí hudby v reproduktorech



Obrázek 7.5: Testovací událost: Detekce kouře

Kapitola 8

Zhodnocení a závěr

Cílem práce bylo navrhnout a realizovat zařízení pro monitoring sdílených ubytovacích prostor. První část práce byla zaměřena na analýzu požadavků sdílených bytů, hotelů a ubytoven a porovnání stávajících řešení na trhu od různých firem. Následoval návrh blokového schématu s vhodnými vstupy a výstupy, po komunikaci až napájení. Rozepsal jsem zde vhodné možnosti měření požadovaných veličin.

Výběr komponent popisuje důvody proč byl zvolen zrovna mikrokontroler ESP32 a poté zvolené senzory pro měření požadovaných veličin. Je zde popsána jejich specifikace a způsoby práce s nimi. Dále jsem popsal zapojení obvodů celého zařízení, od stabilizátoru po samotné propojení senzorů s mikrokontrolerem. V následující části je popsán proces osazení a oživení vyrobeného plošného spoje, v tomto případě bylo třeba provést několik manuálních změn na plošném spoji, z důvodu různých menších chyb kvůli mylným původním předpokladům. Pro ochranu zařízení bylo v CAD programu navrhnuto pouzdro realizované ze dvou dílů, jenž byly vytisknuty na 3D tiskárně.

Samotné programové řešení bylo naprogramováno v jazyce C++ pomocí Arduino knihoven ve vývojovém rozhraní Platform IO a komunikace s mikrokontrolerem přes HTTP požadavky. Tyto knihovny velice usnadnily vývoj a bylo díky ním možné realizovat funkční software i za cenu vyšší abstrakce, tedy nižšího výkonu kódu. Byla využita objektová funkcionality jazyka pro implementaci vlastních tříd jenž se starají o měření a komunikaci se senzory, od čtení hodnot z A/D převodníku až po komunikaci přes I2C sběrnici. Důležitá je také realizace komunikace se službou ThingSpeak, jenž slouží jako archiv naměřených dat ze zařízení s možností spouštět libovolný Matlabovský kód při určité události, jako poslání emailu při překročení prahové hodnoty ze senzoru kouře.

Při testování zařízení se zjistilo že lze detektovat požadované události, jako pohyb osob, změna osvětlení nebo pouštění hlasité hudby, nicméně v současném stavu chybí řešení reakce na některé stavy, jako pouštění hlasité hudby, ikdyž je lze v získaných datech rozpoznat. Toto by se dalo v budoucnu vylepsit implementací nových funkcí buď na straně mikrokontroleru nebo ThingSpeaku.

Další zlepšení do budoucna bych viděl na straně plošného spoje, s tím že by senzory byly přímo osazeny na desce s požadovanou elektronikou na desce, tímto způsobem bychom mohli získat poten-

cionálně menší spotřebu a uvažovat o bateriovém napájení. Samotná konstrukční stránka zařízení by se tímto způsobem také velmi zlepšila. Samozřejmě by se mohli opravit chyby způsobené špatným návrhem, jako špatné zapojení napájecího pinu na fotorezistor. Také by se dala rozšířit uživatelská konfigurace zařízení, kde by uživatel mohl zadávat pomocí HTTP požadavku do zařízení svoje vlastní nastavení prahových hodnot.

Literatura

- [1] MATOUŠEK, David a Bohumil BRTNÍK, 2014-. *Aplikace algoritmů číslicového zpracování signálů*. Praha: BEN - technická literatura. ISBN 978-80-7300-478-1.
- [2] VEDRAL, Josef a Jan FISCHER, 2004. *Elektronické obvody pro měřící techniku*. Vyd. 2. Praha: Vydavatelství ČVUT. ISBN 80-010-2966-2.
- [3] ZÁHLAVA, Vít a Jan FISCHER, 2010. *Návrh a konstrukce desek plošných spojů: principy a pravidla praktického návrhu*. Vyd. 2. Praha: BEN - technická literatura. ISBN 978-80-7300-266-4.
- [4] ABB - Access Control In: ABB [online]. [cit. 2023-08-13]. Dostupné z: <https://new.abb.com/low-voltage/products/knx-building-systems-technology/access-control>
- [5] LOXONE. Profesionální chytrá domácnost [online]. [cit. 2024-04-29]. Dostupné z: <https://www.loxone.com/cscz/>
- [6] Home Assistant [online]. [cit. 2024-04-29]. Dostupné z: <https://www.home-assistant.io/>
- [7] Fotoodpor (fotorezistor). In: ELUC [online]. [cit. 2023-08-20]. Dostupné z: <https://eluc.ikap.cz/verejne/lekce/610>
- [8] Fotodioda In: ELUC [online]. [cit. 2023-08-20]. Dostupné z: <https://eluc.ikap.cz/verejne/lekce/611>
- [9] Fototranzistor. In: ELUC [online]. [cit. 2023-08-20]. Dostupné z: <https://eluc.ikap.cz/verejne/lekce/612>
- [10] MICHALEC, Libor. PIR detektor: skvělý sluha, ale zlý pán. Vývoj HW [online]. [cit. 2023-09-10]. Dostupné z: <https://vyvoj.hw.cz/automatizace/pir-cidlo-skvely-sluha-ale-zly-pan.html>
- [11] VOJÁČEK, Antonín. Detektory kouře - princip & IO Freescale. In: Automatizace HW [online]. [cit. 2024-02-04]. Dostupné z:

<https://automatizace.hw.cz/clanek/2006091701?fbclid=IwAR38pjCL6fvSKiJb5jV-Zn7dODo1zPoAype6Rw8lRPi7vckuyaWIDBdQnxU>

- [12] *KONSTRUKCE MIKROFONŮ*. In: Elektroakustika [online]. [cit. 2024-02-04]. Dostupné z: https://www.elektroakustika.cz/types_microphones.html
- [13] *ESP32*. In: ESPRESSIF. Espressif [online]. [cit. 2024-02-04]. Dostupné z: <https://www.espressif.com/en/products/socs/esp32>
- [14] *WINSEN. Flammable Gas Sensor (Model: MQ-2)*. Winsen [online]. [cit. 2024-02-10]. Dostupné z: [https://www.winsen-sensor.com/d/files/newpdf/mq-2-\(ver1_6\)—manual.pdf](https://www.winsen-sensor.com/d/files/newpdf/mq-2-(ver1_6)—manual.pdf)
- [15] *Interfacing MQ2 Sensor With Arduino*. Circuit Digest [online]. [cit. 2024-04-01]. Dostupné z: <https://circuitdigest.com/microcontroller-projects/interfacing-mq2-gas-sensor-with-arduino>
- [16] *HTU21D Digital High Accuracy RH/T Sensor*. TE CONNECTIVITY. TE [online]. [cit. 2024-02-17]. Dostupné z: <https://www.te.com/usa-en/product-CAT-HSC0004.html>
- [17] *LASKAKIT. PIR detektor pohybu HC-SR501*. Laskakit [online]. [cit. 2024-02-17]. Dostupné z: https://www.laskakit.cz/arduino-pir-detektor-pohybu-hc-sr501/?gad_source=1&gclid=Cj0KCQiAz8GuBhCxARIaOpzk8zHUsuAtflsvC25n50sU2omeLldGX_B4YMIjaVb09i4-uvEp3lWOcaAr5MEALw_wcB
- [18] *LASKAKIT. GY 4466 modul mikrofonu s zesilovačem MAX4466*. Laskakit [online]. [cit. 2024-03-14]. Dostupné z: https://www.laskakit.cz/gy-4466-modul-mikrofonu-s-zesilovacem-max4466/?gad_source=1&gclid=Cj0KCQjwwMqvBhCtARIaIXsZpZRSaqHXtbIja9eIcPxIWnpZwNgFjpX11S_PKpkWrdaqHl3pv6rUJkaAquWEALw_wcB
- [19] *Electret Microphone Amplifier - MAX4466 with Adjustable Gain*. Adafruit [online]. [cit. 2024-04-13]. Dostupné z: <https://www.adafruit.com/product/1063>
- [20] *DECHRISTE, Jean-Mathieu. PIR sensors: HC-SR501*. IoT Experiments [online]. [cit. 2024-04-13]. Dostupné z: <https://www.iot-experiments.com/pir-sensors-hc-sr501/>

Příloha A

Seznam příloh

Příloha I - DP_BAR0469\Přílohy\Elektronika - obsahuje schéma zapojení a osazení, plus cesty plošných spojů.

Příloha II - DP_BAR0469\Přílohy\Zdrojový kód - obsahuje projekt v PlatformIO se zdrojovým kódem řešení

Příloha III - DP_BAR0469\Přílohy\Model pouzdra - obsahuje model pouzdra (spodního a vrchního dílu) pro 3D tisk.

Příloha IV - DP_BAR0469\Přílohy\React - obsahuje kód pro poslání mailu při nenulové naměřené hodnotě na senzoru kouře.