

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

DEPARTMENT OF CONTROL AND INSTRUMENTATION

ROZPOZNÁNÍ OBLIČEJE NA ESP32 S VYUŽITÍM EXISTUJÍCÍHO FRAMEWORKU

FACE RECOGNITION ON ESP32 USING EXISTING FRAMEWORK

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Martin Mušinka

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Tomáš Beneš

BRNO 2020

Bakalářská práce

bakalářský studijní program **Automatizační a měřicí technika**

Ústav automatizace a měřicí techniky

Student: Martin Mušinka

ID: 203300

Ročník: 3

Akademický rok: 2019/20

NÁZEV TÉMATU:

Rozpoznání obličeje na ESP32 s využitím existujícího frameworku

POKYNY PRO VYPRACOVÁNÍ:

Cílem práce je vytvořit systém pro streamování obrazu a detekci obličeje na embedded platformě.

1. Vypracujte literární rešerši na téma možností detekce obličeje.
2. Seznamte se MCU ESP32.
3. Implementujte funkci streamování obrazu.
4. Implementujte funkci detekce obličejů.
5. Vytvořte demonstrační aplikaci "inteligentní domovní zvonek".
6. Systém otestujte a diskutujte dosažené výsledky.

Pro splnění semestrální práce je nutné splnit body zadání 1-3.

DOPORUČENÁ LITERATURA:

WECHSLER, Harry, c2007. Reliable face recognition methods: system design, implementation and evaluation. New York: Springer. ISBN 03-873-8464-2.

Termín zadání: 3.2.2020

Termín odevzdání: 8.6.2020

Vedoucí práce: Ing. Tomáš Beneš

doc. Ing. Václav Jirsík, CSc.
předseda rady studijního programu

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Na technológiu rozpoznávania tváre je v poslednej dobe kladený dôraz vzhľadom na jej široké spektrum využitia. Jedná sa o jednu s viacerých biometrických metód no jej výhoda spočíva v bezkontaktnom procese. V tejto práci sa pozrieme nato, ako funguje funkcia rozpoznávania tváre na mikrokontroléry ESP32 s využitím už existujúceho frameworku.

KĽÚČOVÉ SLOVÁ

esp32, mikrokontrolér, rozpoznávanie tváre, inteligentný domáci zvonček, kamera

ABSTRACT

Facial recognition technology has recently been emphasized due to its wide range of applications. It is one of several biometric methods but its advantage lies in the contactless process. In this thesis, we will look at how the Face Recognition feature works on ESP32 microcontrollers using an existing framework.

KEYWORDS

esp32, microcontroller, face recognition, smart doorbell, camera

MUŠINKA, Martin. *Rozpoznání obličeje na ESP32 s využitím existujícího frameworku*. Brno, 2020, 53 s. Bakalárska práca. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky. Vedúci práce: Ing. Tomáš Beneš

VYHLÁSENIE

Vyhlasujem, že svoju bakalársku prácu na tému „Rozpoznání obličejů na ESP32 s využitím existujícího frameworku“ som vypracoval samostatne pod vedením vedúceho bakalárskej práce, s využitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce.

Ako autor uvedenej bakalárskej práce ďalej vyhlasujem, že v súvislosti s vytvorením tejto bakalárskej práce som neporušil autorské práva tretích osôb, najmä som nezasiahol nedovoleným spôsobom do cudzích autorských práv osobnostných a/alebo majetkových a som si plne vedomý následkov porušenia ustanovenia § 11 a nasledujúcich autorského zákona Českej republiky č. 121/2000 Sb., o práve autorskom, o právach súvisiacich s právom autorským a o zmene niektorých zákonov (autorský zákon), v znení neskorších predpisov, vrátane možných trestnoprávných dôsledkov vyplývajúcich z ustanovenia časti druhej, hlavy VI. diel 4 Trestného zákonníka Českej republiky č. 40/2009 Sb.

Brno 08.06.2020

.....
podpis autora

POĎAKOVANIE

Rád by som poďakoval vedúcemu bakalárskej práce pánovi Ing. Tomášovi Benšlovi, za odborné vedenie, konzultácie, trpezlivosť a podnetné návrhy k práci.

Brno 08.06.2020

.....

podpis autora

Obsah

Úvod	11
1 Detekcia a rozpoznávanie tváre	12
1.1 Využitie	13
1.2 Metódy detekovania tváre	14
1.2.1 Metóda založená na poznatkoch a vedomostiach	14
1.2.2 Metóda založená na znakoch a vlastnostiach tváre	16
1.2.3 Metóda podľa zhody so šablónou	17
1.2.4 Metóda založená na vzhľade	17
2 Mikrokontrolér ESP32(ESP8266)	20
2.1 Espressif Systems	20
2.2 ESP8266	20
2.3 ESP32	21
2.4 Podporované frameworky	22
2.4.1 Arduino	22
2.4.2 ESP-IDF	22
3 Špecifikácia požiadaviek	23
3.1 Riešenie systému	23
3.2 PIR senzor	24
3.3 Kamera	25
3.4 Elektrický zámok	25
3.5 Puzdro	26
4 Streamovanie obrazu na ESP32	27
4.1 Soft AP kamera	27
4.2 Stacionárna kamera	27
5 Rozpoznávanie tváre na ESP32	28
5.1 ESP-WHO	28
5.1.1 MTCNN model a MobileNets	29
6 Hardvérové riešenie	30
6.1 Vývojová doska	30
6.2 Expandér pre I2C komunikáciu	31
6.3 Zapúzdrenie vývojovej dosky	33

7	Softvérové riešenie	34
7.1	Inštalácia ESP-IDF a ESP-WHO	34
7.2	Štruktúra kódu	35
7.3	Implementácia streamovania a rozpoznávania tváre	36
7.4	Hlavný zdrojový kód	38
8	Inteligentný domáci zvonček	41
8.1	Home Assistant	41
9	Dosiahnuté výsledky	43
	Záver	44
	Literatúra	45
	Zoznam symbolov, veličín a skratiek	48
	Zoznam príloh	49
A	Obrázky z prostredia Home Assistant	50
A.1	Prostredie Home Assistant	50
B	Fotografie projektu	52
C	Obsah príloženého CD	53

Zoznam obrázkov

1.1	Rozpoznávanie tváre	12
1.2	Typická tvár pre metódu založenú na poznatkoch a vedomostiach . . .	15
1.3	Distribučná metóda v Gaussovej rovine	18
2.1	ESP8266	20
2.2	ESP32	21
3.1	Blokový diagram projektu	23
3.2	PIR senzor	24
5.1	ESP-WHO	28
6.1	TTGO T-Camera	30
6.2	I2C Expander	32
6.3	Puzdro pre vývojovú dosku TTGO T-Camera	33
A.1	Home Assistant Lovelace	50
A.2	Home Assistant Negatív	50
A.3	Home Assistant Pozitív	51
A.4	Home Assistant Entita	51
B.1	Celkové zapojenie systému	52

Zoznam tabuliek

7.1	Prehľad rozlíšení a fps kamery	37
-----	--	----

Zoznam výpisov

7.1	Náhľad do zdrojového súboru app_httpd.c	35
7.2	Náhľad do zdrojového súboru app_main.c	38
8.1	MJPEG stream v prostredí Home Assistant	42

Úvod

Či sme si toho vedomí, alebo nie s funkciou rozpoznávania podľa tváre sa stretávame čoraz častejšie. Či už sa jedná o nejaké priemyselné kamery, do ktorých sa začína táto funkcia pomaly implementovať a to hlavne v zahraničných krajinách, alebo napríklad aj pri každodennom spôsobe odomykania novších smartfónov. Aj napriek tomu, že sa nejedná o najbezpečnejšiu metódu, v poslednej dobe sa na ňu kladie dôraz na výskum, keďže ide o jednu z najjednoduchších metód. Veď predsa sa aj my ľudia medzi sebou rozoznávame práve podľa tváre, či nie?

Rozpoznávanie tváre môže mať viacero účelov. Či už sa jedná o spôsob ochrany dát v už vyššie spomenutých smartfónoch alebo sa taktiež táto funkcia používa na dolapenie zločincov. Každopádne využití je viacero a nejaké tie hlavné budú prebraté v prvej kapitole.

Táto práca bude zameraná hlavne nato, ako funguje detekcia a rozpoznávanie tváre, aké sú metódy detekcie tváre, mikrokontrolér ESP32 a jeho hlavné parametre, implementáciu rozpoznávania tváre na tento mikrokontrolér a následné vytvorenie demonštračnej aplikácie pre "inteligentný domáci zvonček".

1 Detekcia a rozpoznávanie tváre

Prvým dôležitým pojmom je detekcia tváre. Detekcia tváre je súčasťou detekcie objektov, pri ktorej sa kladie dôraz práve na detekciu ľudských tvárí či už sa jedná o detekciu v reálnom čase alebo detekciu tvárí na rôznych fotografiách alebo obrazoch. Úlohou detekcie tváre je zistiť, či sa na zázname alebo fotografií nachádza alebo nenachádza ľudská tvár. Ak je tvár detekovaná môže sa teda na tejto tvári uplatniť funkcia rozpoznávania tváre.

Ľudská tvár hrá dôležitú úlohu v našej sociálnej interakcii a sprostredkúva identitu ľudí. Vďaka možnosti, že ľudská tvár by mohla byť kľúčom k bezpečnosti, získala technológia biometrického rozpoznávania tváre v posledných rokoch značnú pozornosť vzhľadom na jej široké spektrum využitia. V porovnaní s inými možnými biometrickými metódami, ako napríklad rozpoznávanie podľa odtlačku prstov, dlane alebo rozpoznávanie podľa dúhovky, má rozpoznávanie tváre výhodu vďaka svojmu bezkontaktnému procesu. Tvár dokáže byť detekovaná aj z určitej vzdialenosti bez priameho vedomia jednotlivca, čo môže viesť k odrádzaniu ľudí od zločinov vďaka zálohovaniu tvárí do rôznych databáz. [1]



Obr. 1.1: Rozpoznávanie tváre ako metóda identifikácie pre smartphone

Rozpoznávanie tváre je proces identifikácie alebo overenia totožnosti človeka prostredníctvom tváre. Zachytáva, analyzuje a porovnáva vzory založené na detailoch tváre jednotlivca. Čo vlastne znamenajú pojmy identifikácia a overenie? Pri identifikácii sa kladie otázka „Kto si?“ zatiaľ čo pri overení „Si naozaj ten, za ktorého sa vydávaš?“. Pri identifikácii sa berú do úvahy veci ako napríklad rozmiestnenie očí, nosová prepážka, kontúry pier, uší, brady atď. Tieto špecifické znaky, ktoré sa viažu na každého jedného jednotlivca, sa dokonca dajú rozpoznať v preplnenom dave a za nestabilných podmienok. [2]

1.1 Využitie

Maloobchod

Podľa údajov od spoločnosti NRF sa organizovaný zločin v maloobchode vyčísľuje na približne 30 miliárd dolárov ročne no stále sa to zhoršuje. V skutočnosti 83% maloobchodníkov uvádza medziročný nárast v tomto organizovanom zločine. Výskum však ukazuje, že maloobchodníci využívajúci rozpoznávanie tváre zaznamenali 20% pokles v krádežiach. Mnoho predajcov a obchodov s potravinami (z top 500 amerických súkromných a verejných korporácií podľa ich hrubého obratu) využíva rozpoznávanie tváre, aby nielen odradilo ľudí od krádeže, ale znížilo tak aj násilie v obchodoch.[3]

Orgány činné v trestnom konaní

V súčasnosti sa používa rozpoznávanie tváre v tomto obore na minimalizáciu falošných zatýkaní, na zaistenie bezpečnosti úradníkov a pomoci počas forenzných vyšetrovaní (ide o postupy vedúce k preukázaniu totožnosti osôb, pravosti listín a podobne.)[4]. Dozorné kamery s technológiou rozpoznávania tváre sa môžu použiť na zaistenie bezpečnosti okrskov, zatiaľ čo aplikácie na rozpoznávanie tváre sa môžu používať na mobilných telefónoch na overenie totožnosti podozrivých v teréne.

Kasína

Kasína čoraz častejšie používajú softvér na rozpoznávanie tváre na identifikáciu hráčov, ktorí majú tendenciu podvádzať a majú zákaz hrať v kasíne. Táto technológia sa používa aj na identifikáciu VIP hráčov.[5]

Banky

Rozpoznanie tváre v bankách pomáha pri identifikácii potenciálnych bankových lupičov alebo podvodníkov. Technológiu detekcie tváre možno použiť aj na určenie, ktorým zamestnancom sa poskytne prístup na bezpečné miesta v banke.[5]

Doprava

Letiská a ďalšie dopravné zariadenia používajú rozpoznávanie tváre na identifikáciu potenciálnych teroristov, jednotlivcov, po ktorých pátra napríklad INTERPOL.[5]

1.2 Metódy detekovania tváre

Technológia detekcie tváre, ktorá je súčasťou detekcie objektov je prvým a nevyhnutným krokom pri rozpoznávaní tváre.

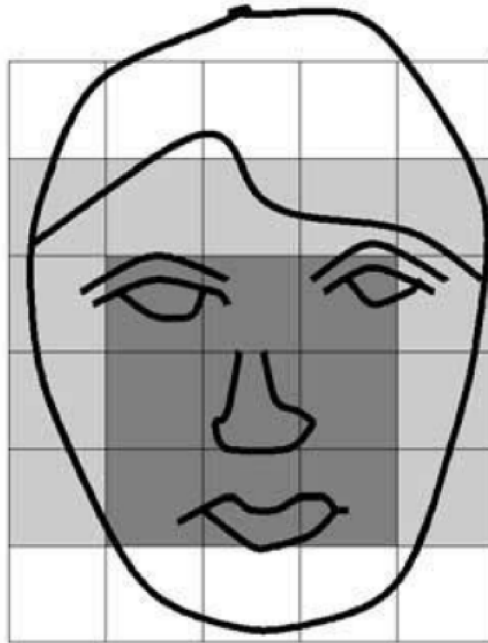
Ludia ako Ming-Hsuan Yang, ktorý je vedecký pracovník spoločnosti Google, David Kriegman, ktorý je profesor počítačovej vedy a inžinierstva na Kalifornskej univerzite a Narendra Ahuja, ktorý je profesorom výskumu na Katedre elektrotechniky a informatiky na Beckmanskoni inštitúte sa spojili a predstavili klasifikáciu metód detekcie tváre. Tieto metódy sú rozdelené do štyroch kategórií a algoritmy detekcie tváre môžu patriť do dvoch alebo viacerých skupín.

- **Metóda založená na poznatkoch a vedomostiach** (Knowledge-Based Methods) - tieto metódy vychádzajú z typických vedomostiach o ľudskej tvári. Zachytávajú vzťahy medzi rysmi tváre. Tieto metódy slúžia hlavne na lokalizáciu tváre.
- **Metóda založená na znakoch a vlastnostiach tváre** (Feature-Based Methods) - cieľom týchto algoritmov je nájsť štrukturálne vlastnosti, ktoré existujú aj keď sa zmení póza alebo svetelné podmienky. Tieto metódy slúžia hlavne na lokalizáciu tváre.
- **Metóda podľa zhody so šablounou** (Template Matching Methods) - tieto metódy uložia niekoľko štandardných vzorov tváre, ktoré popisujú tvár ako celok alebo samostatné rysy tváre. Vypočítavajú zhodu medzi vstupným obrazom a uloženým vzorom na detekciu tváre. Tieto metódy sa teda používajú na lokalizáciu tváre ale aj na detekciu tváre.
- **Metóda založená na vzhľade** (Appearance-Based Methods) - na rozdiel od porovnávania šablón sa modely získavajú z tréningových obrázkov, ktoré reprezentujú tvár. Tieto metódy sú určené hlavne na detekciu tváre. [6] [7]

1.2.1 Metóda založená na poznatkoch a vedomostiach

Tieto metódy sa vyvíjajú na základe pravidiel odvodených od poznatkov výskumných pracovníkov o ľudských tvárach. Je ľahké prísť s jednoduchými pravidlami, ktoré popisujú črty tváre a vzťahy medzi nimi napríklad tvár sa nachádza na obrázku s dvoma symetrickými očami, nosom a ústami. Vzťahy jednotlivých prvkov môžu byť vyjadrené ich relatívnymi vzdialenosťami a polohami. Jedným z problémov týchto metód sú ťažkosti pri premene ľudských vedomostí na presne definované pravidlá. Ak sú pravidlá príliš podrobné, tváre ktoré nespĺňajú presne tieto pravidlá nebudú detekované. Ak budú ale pravidlá príliš všeobecné, môže nastať veľa falošných detekcií tváre. Taktiež je tieto pravidlá ťažko upravovať vzhľadom nato, že tvár môže byť zachytená v rôznych pózach. Na druhej strane tieto metódy fungujú

dosť dobre na detekciu čelných obrázkoch tvárí. G. Yang a T. S. Huang použili na detekciu tvárí práve metódu založenú na poznatkoch a vedomostiach. Ich projekt mal názov “Human Face Detection in Complex Background”, čo v preklade znamená Detekcia ľudských tvárí v zložitom prostredí. Ich systém pozostával s troch úrovni pravidiel. Pravidlá na vyšších úrovniach boli všeobecné opisy toho, ako tvár vyzerá, zatiaľ čo pravidlá na nižších úrovniach záviseli od detailov tváre. Hierarchia multirozlišenia obrázkov je vytvorená spriemerovaním a čiastkovým vzorkovaním.



Obr. 1.2: Typická tvár pre metódu založenú na poznatkoch a vedomostiach, dôraz kladený na rozdiel intenzít v tvárových oblastiach

C. Kotropoulos a I. Pitas založili metódu lokalizácie, ktorá bola podobná ako metóda od G. Yanga a T. S. Huanga. Ak hodnota $I(x,y)$ predstavuje intenzitu obrazu o veľkosti $m \times n$ v polohe (x,y) , horizontálne a vertikálne projekcie obrazu sú definované ako:

$$HI = \sum_{y=1}^n I(x, y) \quad (1.1)$$

$$VI = \sum_{x=1}^m I(x, y) \quad (1.2)$$

Najprv sa získa horizontálny profil vstupného obrazu a potom sú zistené dve lokálne minimá, stanovené detekovaním náhlych zmien v horizontálnej intenzite, ktoré predstavujú ľavú a pravú stranu hlavy. Potom sa získa vertikálny profil a lokálne

minimá sa určia pre umiestnenie pier, špičky nosa a očí. Navrhovaná metóda bola testovaná na súbore tvári zo systému Európskej databázi pre Multimodálne overovanie pre teleserisy a bezpečnostné aplikácie, ktorá obsahovala videosekvencie 37 rôznych ľudí. Každá sekvencia obrázkov obsahovala práve jednu tvár z čelného pohľadu na rovnomernom pozadí. Táto metódy vyhodnotila správne všetkých kandidátov zobrazených na týchto sekvenciách obrázkov. [6] [7]

1.2.2 Metóda založená na znakoch a vlastnostiach tváre

Na rozdiel od metód založených na poznatkoch a vedomostiach sa vedci snažili nájsť rysy tváre na detekciu, ktoré sa nemenia. Základný predpoklad je založený na pozorovaní, že ľudia môžu bez námahy detekovať tváre a objekty v rôznych pózach a svetelných podmienkach, a preto musia existovať vlastnosti alebo znaky, ktoré sú nad týmito premenlivosťami nemenné. Bolo navrhnutých mnoho metód na prvé zistenie rysov tváre a potom na odvodenie prítomnosti tváre. Znaky tváre, ako sú obočie, oči, nos, ústa a vlasová línia, sa zvyčajne získavajú pomocou detektorov hrán. Na základe extrahovaných prvkov sa vytvára štatistický model, ktorý opisuje ich vzťahy a overuje existenciu tváre. Jedným z problémov týchto algoritmov založených na znakoch tváre je to, že obrazové prvky môžu byť vážne poškodené v dôsledku osvetlenia alebo šumu. [6] [7]

Znaky tváre

S.A. Sirohey navrhla metódu lokalizácie na segmentáciu tváre z preplneného pozadia pre identifikáciu tváre s názvom “Human Face Segmentation and Identification” čo v preklade znamená Segmentácia a identifikácia ľudskej tváre. Na odstránenie a zoskupenie hrán používa mapu hrán (detektor Canny) a heuristiku, aby sa zachovali iba tie, ktoré sú na kontúre tváre. Vytvorená elipsa sa potom prispôbí hranici medzi oblasťou hlavy a pozadím. Tento algoritmus dosahuje 80% presnosť v databáze 48 obrázkov s preplneným pozadím. [6] [7]

Farba kože

Farba ľudskej pokožky sa osvedčila ako účinná vlastnosť v aplikáciach ako je detekcia tváre. AJ keď rôzni ľudia majú rôznu farbu pleti, štúdie preukázali, že hlavný rozdiel spočíva v ich intenzite ako v ich farbonosti. [6] [7]

Textúra

Ľudské tváre majú zreteľnú textúru, ktorú je možné použiť na ich oddelenie od rôznych objektov. M.F. Augsteijn a T.L. Skujca vyvinuli metódu, ktorá odvodzuje

prítomnosť tváre prostredníctvom identifikácie textúr podobných tvári. Textúry sa počítajú pomocou štatistických znakov druhého poriadku (SGLD) na podobrazoch s rozlíšením 16 x 16 pixelov. Zvažujú sa tri typy znakov. Použili kaskádovú korelačnú neurónovú sieť na kontrolovanú klasifikáciu textúr a Kohonenovu samoorganizačnú mapu funkcií na vytvorenie zoskupení pre rôzne triedy textúr. [6] [7]

1.2.3 Metóda podľa zhody so šablounou

Pri tejto metóde je štandardný vzor tváre manuálne preddefinovaný alebo parametrizovaný funkciou. Pri vstupnom obrázku sa hodnoty kolerácie (zhody) so štandardnými vzormi vypočítavajú nezávisle pre kontúru tváre, očí, nosa a úst. Existencia tváre je teda určená na základe koleračných hodnôt. Výhodou tejto metódy je jej jednoduchá implementácia no ukázalo sa, že je neadekvátne pre detekciu tváre z dôvodu, že nedokáže efektívne vyhodnotiť tvár pri zmene pózy tváre. [6] [7]

Preddefinované šablóny

Jeden zo skorých pokusov použitia tejto metódy bol projekt s názvom “Line Extraction and Pattern Detection in a Photograph”, čo v preklade znamená Extrakcia čiar a detekcia vzorov na fotografii, na ktorom spolupracovali mená ako T. Sakai, M. Nagao, a S. Fujibayash v roku 1969. Na vytvorenie modelu tváre použili niekoľko podšablón pre oči, nos, ústa a tvár. Kolerácie medzi podobrazmi a šablónami obrysov sa vypočítajú ako prvé, aby sa dala zistiť poloha tváre. Kľúčovým zistením bolo, že zatiaľ čo variácie v osvetlení menia individuálny jas rôznych častí tváre (ako sú oči alebo čelo), relatívny jas týchto častí zostáva do značnej miery nezmenený, čo vedie k stanoveniu pomerov jasu v týchto oblastiach pre dosiahnutie správneho výsledku. [6] [7]

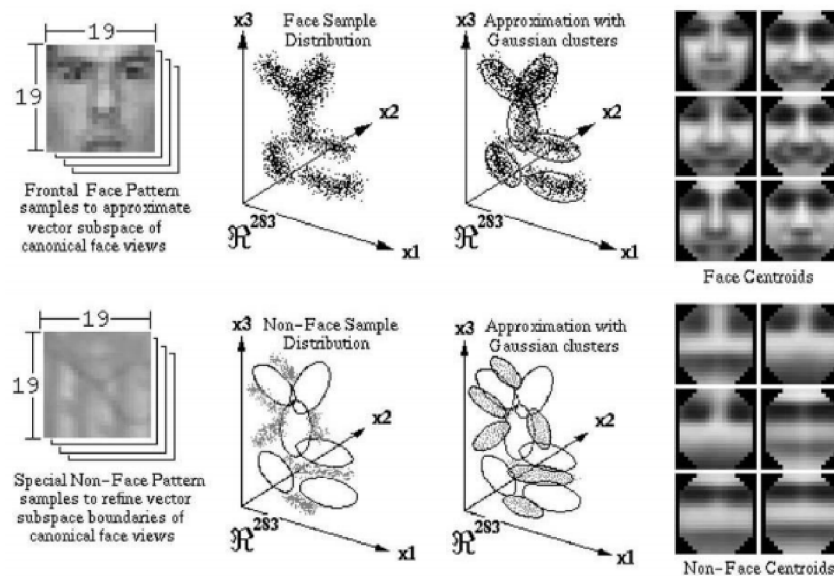
1.2.4 Metóda založená na vzhľade

Na rozdiel od metódy podľa zhody so šablounou, kde odborníci preddefinujú šablóny, sa "šablóny" v metódach založených na vzhľade učia z príkladov na obrázkoch. Metódy založené na vzhľade sú postavené na technikách zo štatistickej analýzy a strojového učenia (Machine Learning), aby našli príslušné charakteristiky na obrázkoch na ktorých sa tvár nachádza a na ktorých sa tvár nenachádza. Získané charakteristiky sú vo forme distribučných modelov alebo diskriminačných funkcií, ktoré sa následne používajú na detekciu tváre. Kvôli efektívnosti výpočtu a účinnosti detekcie sa obvykle uskutočňuje aj redukcia rozmerov/kvality obrázka. Mnohé metódy založené na vzhľade môžu byť chápané ako v pravdepodobnostnom rámci. Obrázok alebo znakový vektor odvodený z obrázka sa považuje za náhodnú premennú X a

táto náhodná premenná je pre tváre charakterizovaná funkciou $p(X_{\text{detekovanejtváre}})$ a pre obrázky, kde nie je tvár je charakterizovaná funkciou $p(X_{\text{obrázokbeztváre}})$. Je možné použiť Bayesovú klasifikáciu alebo maximálnu pravdepodobnosť na klasifikáciu umiestnenia tváre. [6] [7]

Distribučné metódy

K.-K. Sung a T. Poggio vyvinuli systém detekcie tváre založený na distribučnej metóde s názvom “Example-Based Learning for View- Based Human Face Detection” čo v preklade znamená Učenie založené na príkladoch pre detekciu ľudskej tváre na základe pohľadu, ktorý demonštroval, ako je možné zistiť rozdelenie obrázkov z triedy objektov pre obrázky z pozitívnym alebo negatívnym zistením tváre. Ich systém sa skladá z dvoj komponentov, z modelu založeného na distribučnej metóde pre vzory s tvárou a bez tváre a z viacvrstvového klasifikátora perceptrónu čo je špeciálny prípad neurónu v neurónových sieťach. Každý obrázok sa najprv normalizuje a spracuje na obrázok s rozmermi 19x19 pixelov a spracuje sa ako 361-rozmerný vektor alebo vzor. Tieto vzory sú zoskupené do šiestich zložiek s tvárou a bez pomocou modifikovaného algoritmu KMEANS ako je znázornené na nasledujúcom obrázku. Následne sa určuje vzialenosť prototypu a zosnímaného obrázku v Gaussovej rovine. [6] [7]



Obr. 1.3: Zobrazenie distribučnej metódy v Gaussovej rovine

Podporné vektorové stroje

Podporné vektorové stroje aplikoval E. Osuna, R. Freund, a F. Girosi po prvý krát v ich projekte s názvom "Training Support Vector Machines: An Application to Face Detection". Podporné vektorové stroje sa môžu považovať za nový vzor na tréningovanie klasifikátorov polynómových funkcií, neurónových sietí alebo radiálnych bázových funkcií. Zatiaľ čo väčšina metód pre výcvik klasifikátora je založená na minimalizácii chyby počas tréningu, vektorové stroje fungujú na inom princípe, ktorý sa nazýva minimalizácia štruktúrneho rizika. Jeho cieľom je minimalizovať hornú hranicu na očakávanom zlyhaní generalizácie. [6] [7]

Hidden Markov Model

Základným predpokladom Hidden Markov Modelu je, že vzory možno charakterizovať ako parametrický náhodný proces a parametre tohto procesu možno odhadnúť presným a dobre definovaným spôsobom. Pri vývoji HMM pre rozpoznávanie je potrebné najprv vytvoriť niekoľko skrytých stavov, aby sa vytvoril model. Potom je možné trénovať HMM, aby sa naučil prechodnú pravdepodobnosť medzi stavmi z príkladov, kde každý príklad je reprezentovaný ako sled pozorovaní. Cieľom tréningu HMM je maximalizovať pravdepodobnosť pozorovania tréningových údajov úpravou parametrov v modeli HMM pomocou štandardnej metódy segmentácie Viterbi a algoritmov Baum-Welch. Po vyškolení HMM určuje pravdepodobnosť výstupu triedy do ktorej patrí. Metódy založené na HMM zvyčajne spracovávajú vzor tváre ako sekvenciu pozorovacích vektorov, kde každý vektor je pásom pixelov. Počas školenia a testovania sa obrázok skenuje v určitom poradí (zvyčajne zhora nadol) a pozorovanie sa považuje za blok pixelov. HMM sa používa na rozpoznanie tváre aj na lokalizáciu. [6] [7]

2 Mikrokontrolér ESP32(ESP8266)

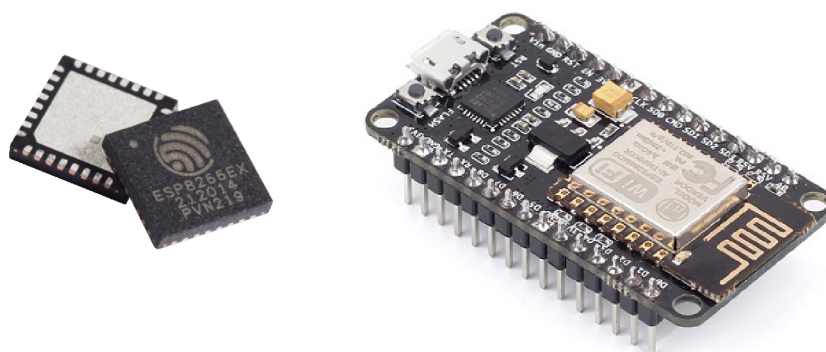
Táto kapitola je zameraná na samotný mikrokontrolér ESP32, kde je uvedené aj niečo o jeho predchodcovi ESP8266 a spomenutá bude aj firma ktorá uviedla tieto mikroprocesory na trh. V tejto kapitole budú taktiež spomenuté hlavné parametre týchto mikroprocesorov.

2.1 Espressif Systems

Espressif Systems je verejná mnohonárodná spoločnosť založená v roku 2008 so sídlom v Šanghaji a centrálnymi v Číne, Indii a Európe. Táto spoločnosť sa zameriava na vývoj výkonných a nízkoenergetických Wifi a Bluetooth mikrokontrolérov. Práve táto firma stojí za vytvorením populárnej série čipov, modulov a vývojových dosiek ESP8266 a ESP32 ktoré sa využívajú hlavne v IoT (z ang. Internet of Things - Internet vecí), jedná sa vlastne o prístroje, ktoré sú určitým spôsobom pripojené k internetu a sú schopné medzi sebou komunikovať.[8]

2.2 ESP8266

Medzi asi najväčšieho konkurenta samotného mikrokontroléru ESP32 patrí určite aj jeho mladšia generácia a to mikrokontrolér ESP8266. Jedná sa o cenovo dostupný Wifi mikrokontrolér firmy Espressif Systems ktorý bol uvedený na trh v roku 2014. Aj keď sa už môže jednať o pomerne starý mikrokontrolér stretávame sa s ním dodnes. Môžeme sa s ním stretnúť buď v podobe samotného čipu alebo na rôznych doskách či vývojových kitch. ESP8266 je osadený jednojádrový 32-bitovým mikroprocesorom Tensilica Xtensa Diamond Standard 106Micro ktorý beží na frekvencií 80Mhz. Tento

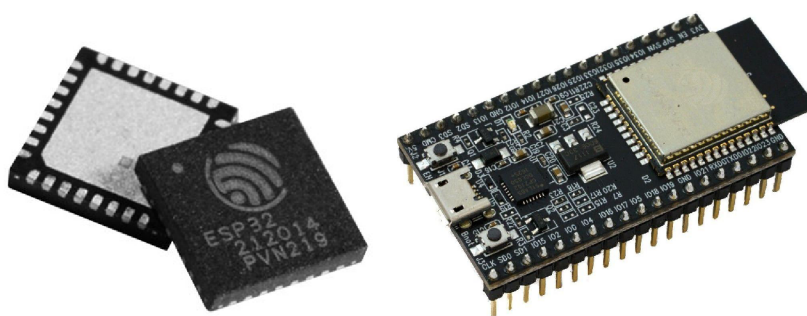


Obr. 2.1: ESP8266 čip (vľavo) a ESP8266 12-E NodeMCU Kit (vpravo)

mikrokontrolér využíva 3,3V logiku. Medzi jeho základné vlastnosti patrí aj anténa k pripojeniu na Wifi, SPI, I2C, I2S a UART komunikácia a je vybavený 16 GPIO (z ang. General Purpose Input Output - Vstupy a výstupy pre všeobecné použitie)[9].

2.3 ESP32

Mikrokontrolér ESP32 je vlastne nástupca mikrokontroléra ESP8266. Ide o novšiu, trochu drahšiu no zároveň aj omnoho výkonnejšiu verziu. Práve preto môže byť tento mikrokontrolér zakomponovaný aj v bežných domácich, no zároveň aj vo väčších a profesionálnejších projektoch. Taktiež ako pri ESP8266 aj pri ESP32 existuje mnoho rôznych dosiek a mnoho rôznych kitov.



Obr. 2.2: ESP32 čip (vľavo) a ESP32-DevKitC (vpravo)

V tejto práci bude použitý práve mikrokontrolér ESP32. Mikrokontroléry ESP32 sú vybavené dvojjadrovým 32-bitovým procesorom Xtensa LX6 pracujúceho pri 160 alebo až 240MHz a zároveň aj sekundárnym procesorom s extrémne nízkou spotrebou energie prezývaného ULP (z ang. Ultra low power) a pamäťou typu SRAM (z ang. Static Random Access Memory - statická RAM pamäť) o veľkosti 520 kibi-bajtov (1 kibibajt = 2^{10} bitov). ESP32 je vybavený bezdrôtovým Wifi a Bluetooth pripojením, zatiaľ čo ESP8266 je vybavený iba Wifi pripojením. Taktiež je dobré spomenúť periférne rozhrania. Mikrokontroléry ESP32 sú vybavené rozhraniami ako SPI, I2S, I2C, UART alebo CAN2.0, pulzne šírkovou moduláciou PWM a napríklad aj Halovou sondou. [10]

Novšie čipy ESP32 navyše podporujú bezpečnostné funkcie, ako je **secure boot** (bezpečné spustenie) a **flash encryption** (šifrovanie dát) ktoré sú potrebné na implementáciu „production ready“ (produktov pripravených na výrobu) inteligentných zariadení pre IoT. Ďalšou zaujímavosťou je napríklad aj podpora ML (z ang. Machine Learning - strojové učenie) v spolupráci s frameworkom ESP-WHO.[11]

2.4 Podporované frameworky

Medzi výrobcom podporovaným frameworkom patria iba ESP-IDF a Arduino. Vďaka podpore komunity je už možné stretnúť sa aj s alternatívnymi riešeniami ako napríklad Espruino (využívajúce JavaScript) alebo Micropython.

2.4.1 Arduino

Jedná sa o rozšírené "open-source" vývojové prostredie Arduino IDE, implementujúce Arduino software, ktoré malo slúžiť pôvodne pre mikrokontroléry ATMega od firmy Atmel. Toto prostredie je hlavne rozšírené kvôli jeho jednoduchosti a využívajú ho zväčša domáci užívatelia. Vďaka svojej popularite má aj mnoho voľne prístupných knižníc. Programy písané v tomto prostredí sa skladajú z dvoch hlavných častí a to `setup()` a `loop()`. Sekcia `setup()` slúži zväčša na inicializáciu, keďže všetko čo sa nachádza v tejto sekcii prebehne iba raz a to pri štarte. Do sekcie `loop()` sa píše tá časť kódu, ktorá sa volá opakovane. S pojmom Arduino sa môžeme stretnúť aj pri hardware, keďže sa jedná o software a hardware firmu, a to pri rôznych doskách ako napríklad Arduino Nano alebo Arduino Mega.

2.4.2 ESP-IDF

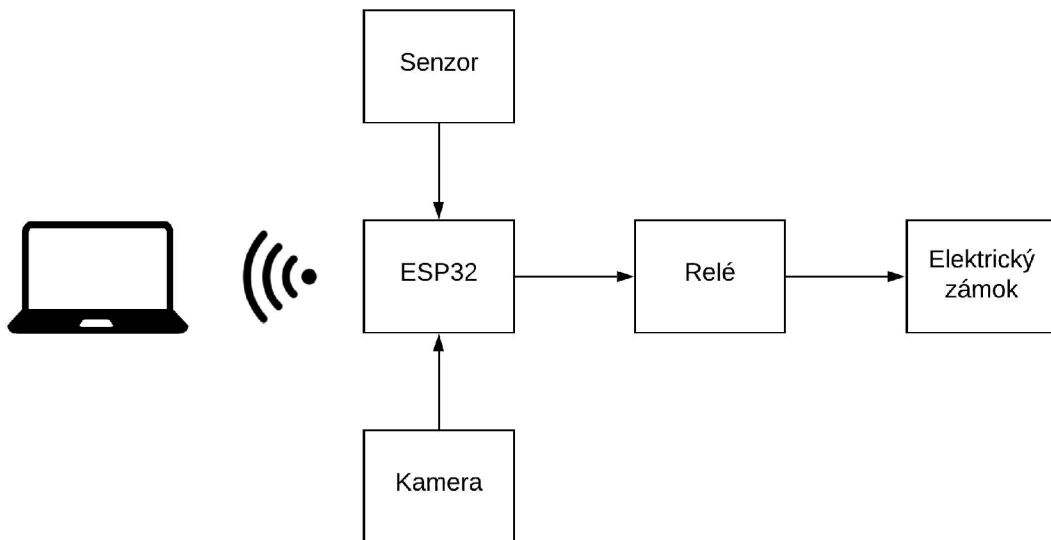
Espressif Development Framework alebo v skratke ESP-IDF je framework vyvíjaný práve firmou, ktorá stojí za výrobou mikrokontrolérov ESP32 a to Espressif Systems. Tento framework obsahuje knižnice API pre periférie, rôzne komunikačné protokoly a je založený na FreeRTOS, čo je tzv. operačný systém reálneho času (RTOS - Real-Time Operating System). S implementáciou tohto frameworku sa stretneme napríklad aj vo vývojových prostrediach ako Eclipse IDE alebo Platform IO, ktoré slúži ako rozšírenie vo Visual Studio Code. V tomto frameworku môžeme využiť aj knižnice z už vyššie spomenutého prostredia Arduino a to tak, že pridáme Arduino ako komponentu. Zvýši sa tým síce veľkosť súboru ale výhodou je využitie už napísaných knižníc. Využíva sa tu programovací jazyk Python.

3 Špecifikácia požiadaviek

V tejto kapitole budú upresnené požiadavky, ktoré sú potrebné pre uskutočnenie funkčnosti inteligentného domáceho zvončeka. Zo začiatku bude popísané riešenie celého projektu, spôsoby napájania a vhodný výber komponentov.

3.1 Riešenie systému

Cieľom projektu je vytvoriť zariadenie, ktoré bude schopné pomocou rozpoznávania tváre rozpoznať človeka a následne mu povoliť prístup do domu alebo budovy pomocou elektrického zámku. Celé zariadenie však musí spadať pod skupinu IoT zariadení čo znamená, že musí byť schopné komunikovať s okolitými zariadeniami ako napríklad mobil či počítač, musí byť nízkoenergetické a taktiež kompaktné.



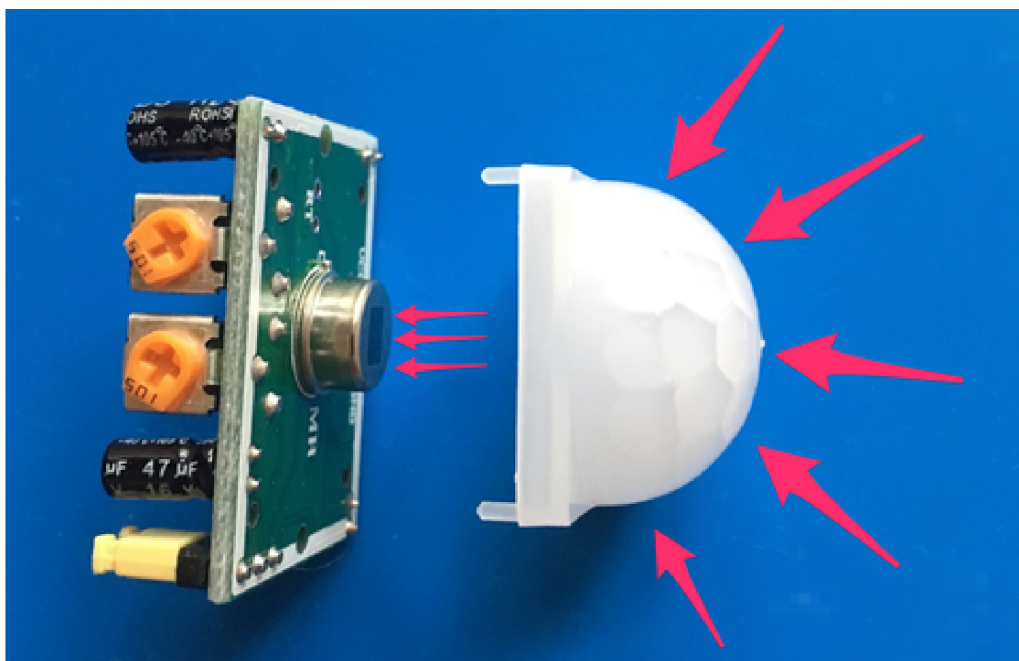
Obr. 3.1: Blokový diagram projektu

Zariadenie prostredníctvom senzoru zachytí že sa niekto nachádza v priestoroch dverí a náhle spustí svoju činnosť. Kamera zašle fotografiu majiteľovi o činnosti predo dvermi a taktiež ak je táto osoba autorizovaná, t.z. jej tvár bola rozpoznaná, bude jej umožnený vstup. V prípade že osoba nebola rozpoznaná vstup jej umožnený nebude a majiteľ dostane upozornenie, bude môcť rozhodnúť čo sa bude mať v tejto chvíli vykonať. Na výber budú možnosti ako spustiť alarm alebo povoliť vstup.

Hlavným komponentom tohto celého projektu bude práve mikrokontrolér ESP32 ktorý bude mať za úlohu všetky dáta spracovať a odosielať na server k užívateľovi.

3.2 PIR senzor

PIR (z anj. Passive Infra-Red - pasívny infračervený) senzor slúži na detekciu pohybu. S týmto senzorom sa môžeme často stretnúť pri rôznych druhov alarmov alebo aj pri osvetlení rodinných domov alebo rôznych budov, kde sa pri nastavenej senzitivite osvetlenia a zaznamenaného pohybu zopne osvetlenie na určitý čas. Tento senzor je elektronický senzor, ktorý meria infračervené svetlo vyžarované v určitom zornom poli. Všetky objekty s teplotou nad nulou vysielaajú tepelnú energiu vo forme žiarenia. Zvyčajne toto žiarenie nie je pre ľudské oko viditeľné, pretože vyžaruje na infračervených vlnových dĺžkach, ale môže byť detekované elektronickými zariadeniami určenými na tento účel. PIR senzor vlastne detekuje prítomnosť pohybu všeobecne, no neposkytuje informácie o tom kto alebo čo sa v ten moment pohlo. To v našom projekte problém nebude keďže jeho súčasťou je aj kamera a primárnou funkciou tohto senzoru je zachytiť iba samotný pohyb. Hlavnou časťou takéhoto PIR senzoru je pyroelektrický snímač ktorý vyšle signál, keď zaznamená teplotnú zmenu. [12]



Obr. 3.2: Obrázok PIR modulu s Fresnelovou šošovkou

Tento senzor je zväčša vybavený aj tzv. Fresnelovou šošovkou, ktorá zabezpečuje nasmerovanie tepelných lúčov z okolia priamo na spomínaný pyroelektrický snímač. Výstup takéhoto senzoru je na rôznych moduloch zväčša digitálny a to znamená, že na výstup priradí logickú 1 v prípade, že zaznamenal pohyb alebo logickú 0 v

prípade, že žiaden pohyb nebol zaznamenaný. Práve táto zmena z kludovej hodnoty logickej 0 na hodnotu logickú 1 pri zaznamenaní pohybu môže byť využitá pre zaslanie informácie mikrokontroléru ESP32, ktorý môže následne vyhodnotiť, že sa v priestoroch pred zvončekom nachádza človek a napríklad poslať majiteľovi upozornenie spolu aj s fotografiou.

3.3 Kamera

Asi jedným z najhlavnejších komponentov tohto projektu je kamera. Bude slúžiť na zachytenie fotografie, streamovanie videa a samotnú funkciu rozpoznávania tváre jednotlivých osôb. Pri moduloch obsahujúcich mikrokontroler ESP32 sa najčastejšie stretávame s kamerou OV2640. Jedná sa o kameru s rozlíšením 2Mpx, ktorá pre tento projekt bude bohato postačovať. Táto kamera umožňuje maximálnu rýchlosť prenosu obrázkov pri UXGA/SXGA 30fps (z ang. Frames per second - snímky za sekundu), pri SVGA taktiež 30fps a pri CIF 60fps. Na tejto kamere máme taktiež možnosť nastavenia nejakých základných parametrov ako napríklad jas, kontrast, saturáciu alebo aj otočenie obrazu po vertikálnej alebo horizontálnej osi. Pre komunikáciu s mikrokontrolérom využíva sériovú komunikáciu I2C. Jedná sa o synchronnú sériovú komunikáciu objavenú spoločnosťou Phillips v roku 1982, ktorá obsahuje 2 hlavné vodiče a to SDA a SCL. V SDA sú uschované dáta, ktoré sa majú preniesť a SCL slúži ako hodinový signál.[13]

3.4 Elektrický zámok

Keďže výstupom práce má byť je systém predstavujúci funkčnosť inteligentného domáceho zvončeka, elektrický zámok by mal byť súčasťou. Bude sa nachádzať v dverách, kde sa za určitých podmienok dvere budú môcť otvoriť. Systém pomocou implementovania rozpoznávania tváre bude rozhodovať o tom, či daná osoba má oprávnenie vstupu alebo nie a práve podľa toho rozhodne ako sa taký elektrický zámok bude správať. Jedným z adeptov je zámok Dorcas 41Aa Flex. V kludovom režime je tento zámok zamknutý, po pripojení napájacieho napájania je odomknutý. Výhodou tohto zámku je napríklad aj možnosť využitia iba krátkeho impulzu napájania, ktorý umožňuje tomuto zámku jedno otvorenie.[14] Takže v prípade ak bol daný jednotlivец rozpoznaný stačí, že systém vyšle krátky impulz do tohto zámku a dotýčny môže vstúpiť.

3.5 Puzdro

V poslednom rade bude pre tento projekt navrhnuté pudzro, do ktorého budú implementované všetky potrebné komponenty, aby bolo zaistené upevnenie hlavných častí ako napríklad kamery a PIR senzoru v okolí dverí pre možnosť zaznamenania pohybu a samotného rozpoznávania tváre.

4 Streamovanie obrazu na ESP32

Jednou z funkcií tohto projektu je taktiež aj možnosť sledovania deja odohrávaného pred inteligentným zvončekom. K tomu slúži kamera, ktorá je implementovaná na doske. Streaming (z anj. stream – prúd) je technológia kontinuálneho prenosu audiovizuálneho materiálu medzi zdrojom a koncovým užívateľom. V súčasnej dobe streaming slúži predovšetkým k prenášaniam audiovizuálneho materiálu po internete.

Pri hľadaní informácií ohľadom streamovania obrazu na mikrokontroléry ESP32 sú na výber dve metódy streamovania obrazu.

4.1 Soft AP kamera

Soft AP (z anj. Soft Access Point) je skratka pre softvérový prístupový bod. To umožňuje počítaču alebo inému zariadeniu, ktoré nebolo špeciálne vyrobené ako router plniť jeho funkciu. V prípade implementácie tejto metódy na tento projekt, by sa z dosky automaticky stal WiFi prístupový bod, na ktorý je možnosť sa pripojiť a následne získať prístup napríklad ku dátam z kamery. Zariadeniu je za potreby nakonfigurovať tzv. SSID (z anj. Service Set Identifier), čo predstavuje názov prístupového bodu a bezpečnostný kľúč, čo predstavuje heslo k tomuto prístupovému bodu. Táto metóda je možno vhodná na testovanie správania tohto projektu, no vo výsledku sa nebude používať, keďže zariadenie bude napojené na domácej sieti.

4.2 Stacionárna kamera

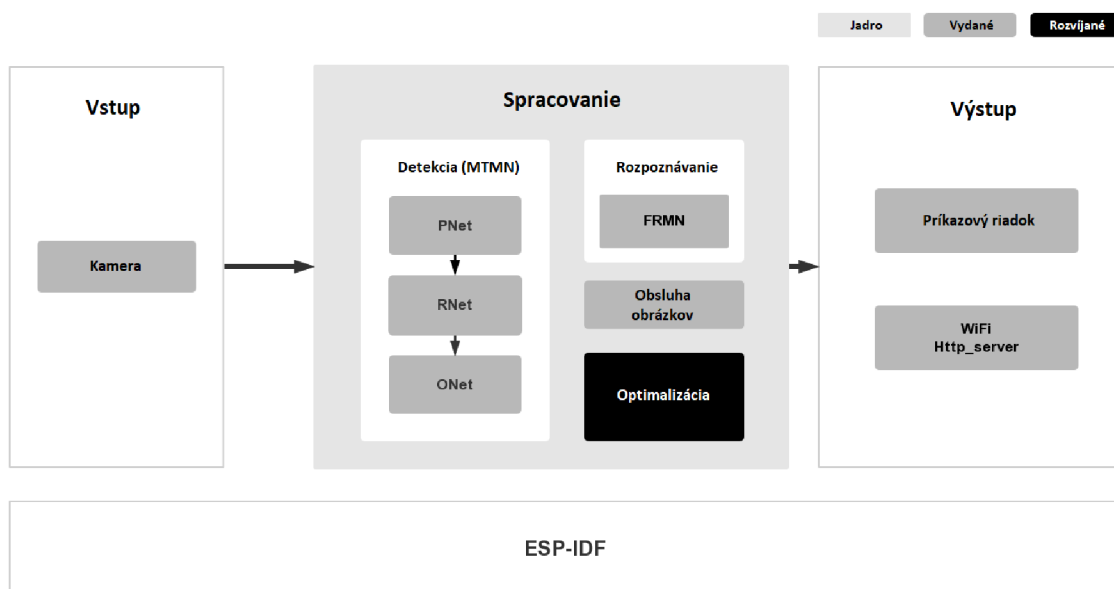
Zatiaľ, čo pri Soft AP metóde je samotné zariadenie prístupovým bodom, pri stacionárnej metóde sa toto zariadenie klasicky pripája k už vytvorenému prístupovému bodu, ako je napríklad WiFi domácej siete. Tomuto zariadeniu je za potreby nakonfigurovať SSID a bezpečnostný kľúč no zatiaľ, čo pri Soft AP to bolo treba vymyslieť, v tomto prípade je za potreby zadať informácie o sieti, na ktorú sa zariadenie bude pripájať. Práve táto metóda bude využitá v tejto práci. Jej jedinečnou výhodou je práve to, že k tomuto zariadeniu (keďže bude pripojené na domácu sieť) je možné napojiť hocijaké iné zariadenia ako napríklad inteligentné svetlá či zásuvky a pomocou týchto vecí vytvoriť rôzne automatizácie.

5 Rozpoznávanie tváre na ESP32

Samozrejme hlavnou funkciou tohto projektu je práve rozpoznávanie tváre. Pomocou čoho tvár rozpoznávame, ako to funguje a ako napríklad nahráť nové tváre bude vysvetlené v tejto kapitole.

5.1 ESP-WHO

ESP-WHO je platforma na detekciu a rozpoznávanie tváre, ktorú v súčasnosti využívajú práve mikrokontroléry ESP32 od spoločnosti Espressif Systems a spolupracuje s frameworkom ESP-IDF. Na nasledujúcom obrázku je znázornený princíp fungovania tejto platformy.



Obr. 5.1: Princíp fungovania platformy ESP-WHO

Jadrom tejto platformy je práve detekcia tváre, rozpoznávanie tváre a využitie obrázkov. Využitie obrázkov ponúka základné API na spracovanie obrázkov. Detekcia sníma obrázky zo vstupu (z kamery) a ak sa tam tvár nachádza, určuje jej polohu. Využíva model MTMN, ktorý odkazuje na MTCNN a MobileNets. Rozpoznanie tváre má za úlohu identifikovať konkrétnu osobu a vyžaduje výsledky detekcie. Pre svoju činnosť využíva model MobileFace. Optimalizácia sa stará hlavne o presnosť a rýchlosť celého procesu.

Vstup aj výstup sú flexibilné. Zdroje obrazu je možné snímať pomocou kamery. ESP-WHO zatiaľ podporuje iba 3 najčastejšie sa vyskytujúce kamerové moduly a to OV2640, OV3660 a OV6540.

Na spustenie ESP-WHO je potrebné vlastniť vývojovú dosku, ktorá integruje modul ESP32 a viac ako 4 MB externej SPI RAM pamäte. Ako testovacia doska môže byť dobrou voľbou buď ESP-WROVER-KIT alebo ESP-EYE.[15]

5.1.1 MTCNN model a MobileNets

MTCNN (Multi-task Cascaded Convolutional Neural Networks) je algoritmus pozostávajúci z 3 fáz, ktorý spravuje ohraničovacie rámčeky pri detekcii tváří spolu s 5 bodmi pre orientáciu. Každá fáza postupne zlepšuje výsledky detekcie tak, že prechádza vstupmi cez CNN, ktorý vracia ohraničujúce políčka kandidátov s ich skóre. V prvej fáze je vstupný obrázok niekoľkokrát zmenšený, aby sa vytvorila pyramída obrázka a každá zmenšená verzia obrázka prechádza cez jeho CNN. V etape 2 a 3 extrahujeme obrazové záplaty pre každú ohraničovaciu škatuľu a zmeníme ich veľkosť (24x24 v etape 2 a 48x48 v etape 3) a postúpime ich cez CNN tejto etapy. Okrem ohraničujúcich políčok a skóre, 3. etapa navyše počíta 5 tvárových orientačných bodov pre každú ohraničovaciu schránku. Ukázalo sa, že pomocou MTCNN sa dajú získať solídne výsledky a navyše je to časovo menej náročné ako napríklad SSD Mobilenet v1. [16]

MobileNets je trieda účinných modelov pre mobilné alebo iné embedded zariadenia s kamerou. Sú založené na architektúre ktorá sa využíva pri vytváraní ľahkých hlbokých neuronových sietí. MobileNets sa využívajú v širokej škále a to napríklad pri detekcii objektov, detekcii tváří alebo rozsiahlej geografickej lokalizácie. [17]

6 Hardvérové riešenie

V tejto kapitole bude rozobratý hardvér, ktorý bol použitý pri tvorbe tohto projektu. Niektoré veci, ktoré boli spomenuté v kapitole 3 boli nahradené alebo pridané.

6.1 Vývojová doska

Vzhľadom na požiadavky z kapitoly 3 bol zhotovený záver, že pre tento projekt je vyhovujúcim adeptom doska TTGO T-Camera ktorá je vybavená čipom ESP32-WROVER-B, pretože splňuje všetky z našich požiadavok.



Obr. 6.1: TTGO T-Camera od firmy LILYGO

Jedná sa o vývojovú dosku od firmy Lilygo. Patrí do série výrobkov TTGO, do ktorej spadajú aj iné vývojové dosky ako napríklad vývojové dosky s LCD OLED displejmi alebo aj vývojové dosky predstavujúce inteligentné hodinky. Táto vývojová doska TTGO T-Camera obsahuje kameru OV2640, ktorá bola spomenutá v kapitole 3.3, ktorá ponúka aj širokouhlú Fish-eye šošovku pre zväčšenie zorného poľa kamery, no pre tento projekt to potrebné nie je takže ju používať nebudeme. Vývojová doska je o rozmeroch 68x28mm a obsahuje aj nasledujúce komponenty:

- OLED displej - s rozlíšením 128x64 pixelov
- PIR senzor - spomenutý v predchádzajúcej kapitole 3.2
- Dve fyzické tlačidlá - tlačidlo RST pre resetovanie tejto vývojovej dosky a tlačidlo ktoré je napojené na vstupno-výstupný pin s číslom 34 (IO34) čipu ESP32-WROVER-B ktorý budeme pre tento projekt využívať
- Konektory pre napájanie

- USB micro-B 2.0 - pomocou tohto konektoru do vývojovej dosky nahrávame program no taktiež slúži aj ako napájanie tohoto systému
- 2PIN 1.25mm - slúži ako externé napájanie napríklad pre batériu
- Konektor typu 5PIN 2.0mm - ide o konektor s piatimi pinmi a 2mm rozstupmi medzi nimi, tri piny pre napájanie (5V; 3,3V; GND) a dva piny pre I2C komunikáciu(SDA - v našom prípade pin IO21 a SCL - v našom prípade pin IO22)

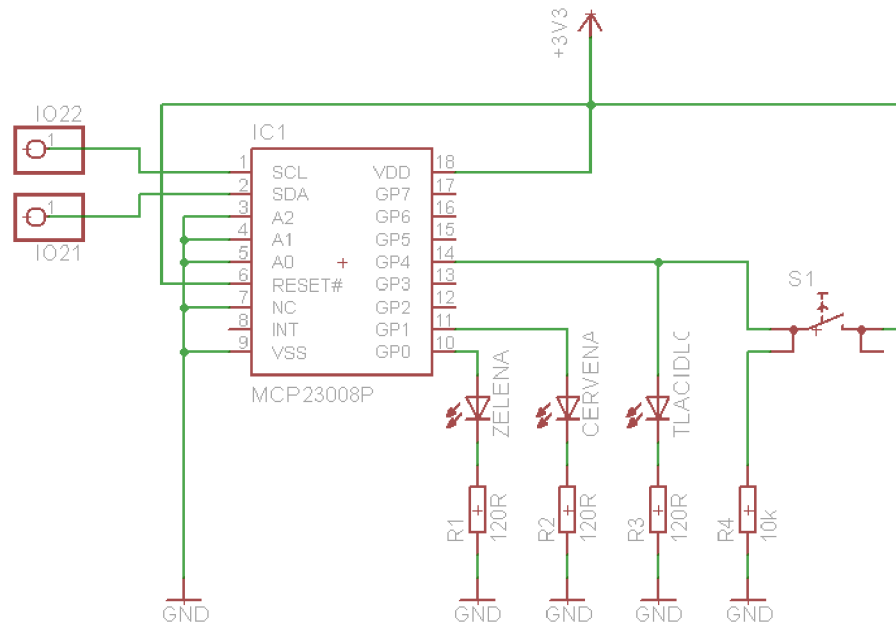
To by bolo, čo sa týka kladov tejto vývojovej dosky no je tu aj jeden zápor. Táto doska nemá žiadne použiteľné GPIO. Toto nemilé zistenie teda vedie k použitiu I2C expandéra, ktorý bude popísaný v nasledujúcej podkapitole. Ďalšie nemilé zistenie bolo, že kábel ktorý sa má pripojiť do už spomínaného 5PIN 2.0mm nie je nikde v rámci Českej a Slovenskej republiky dostupný. Za týmto káblom by bolo teda treba čakať asi mesiac, takže bola zvolená alternatíva a to prispájkovanie káblikov priamo na tento konektor. [18]

6.2 Expandér pre I2C komunikáciu

Ako už bolo spomenuté v predošlej kapitole (6.1) vývojová doska, ktorá bola zvolená neobsahuje žiadne použiteľné vstupy ani výstupy, takže pripojiť externé zariadenia ako napríklad elektrický zámok (kapitola 3.4) by bolo teda nemožné, preto bola zvolená metóda použitia práve expandéra pre I2C komunikáciu pomocou, ktorého je možné komunikovať napríklad s elektrickým zámkom. Elektrický zámok ale v tomto projekte použitý nebude z aj finančného dôvodu a faktu, že v tejto bakalárskej práci je kladený dôraz na samotnú kameru a funkciu rozpoznávania tváre, takže elektrický zámok bude nahradený LED diódovou signalizáciou pre demonštráciu správneho chodu systému.

I2C expandér je zaradenie, ktoré pomocou I2C komunikácie bude prepojené s vývojovou doskou a to nám umožní pripojiť určitý počet vstupno-výstupných zariadení. Pre tento projekt bol zvolený expandér typu MCP23008-E/P. Tento expandér obsahuje 8 použiteľných bitov pre I2C komunikáciu. Je vyhotovený z typu púzdra DIP18. Tento expandér s týmto typom púzdra bol zvolený aj práve kvôli demonštrácií systému na nepájivom poli, do ktorého perfektne zapadne. Prenosová rýchlosť takéhoto I2C expandéru je 1.7Mbps a pracuje s napájacím napätím od 1,8V do 5,5V DC napájania. Pre tento projekt zvolíme napájanie 3,3V práve z už spomínaného konektoru 5PIN 2.0mm spomenutého v kapitole 6.1. I2C expandér bol teda k vývojovej doske zapojený podľa nasledujúcej schémy zapojenia viz Obr. 7.2.

Na tejto schéme je vidieť, ako je tento expandér pripojený k napájaniu a aké jeho ďalšie piny sú využité. Tento čip využíva pin RESET s opačnou logikou, preto bolo na tento pin pripojené napájacie napätie 3,3V. Piny A0 až A2 signalizujú



Obr. 6.2: Zapojenie I2C expandéra pre demonštráciu projektu

adresu tohoto expandéra. V prípade ak budú všetky tieto adresové piny pripojené na zem, východzia adresa tohoto expandéra bude 0x20 v hexadecimálnom zápise. V prípade iného zapojenia týchto adresových pinov sa hodnota ktorá predstavuje číslo v binárnom kóde (3bitové číslo) pričíta k tejto východzej hodnote. Takže v prípade ak by A0 a A2 boli zapojené na zem a pin A1 by bol pripojený na napájacie napätie informácia by predstavovala hodnotu v binárnej sústave 010 čo je po prepočítaní do hexadecimálnej sústavy číslo 0x02, adresa tohoto expandéru by teda bola 0x22 v hexadecimálnom zápise. Toto rôzne adresovanie môže byť využité pre komunikáciu napríklad s ďalšími expandérmí kde každý bude mať k sebe priradenú svoju vlastnú adresu pre správny beh komunikácie. V tomto prípade budeme používať iba tento jeden expandér takže adresu necháme východziu. [19]

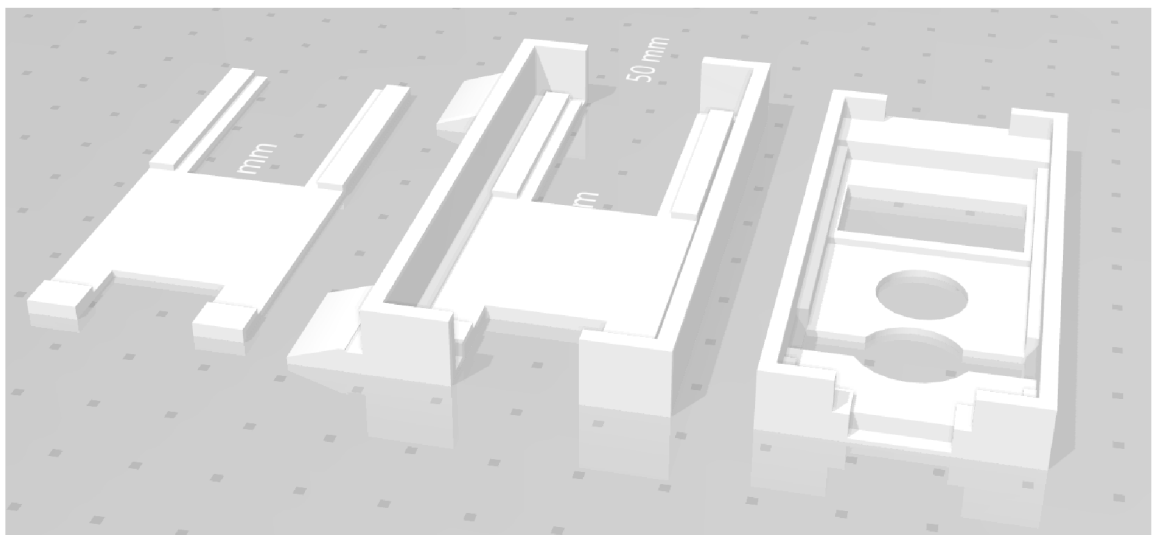
IO21 a IO22 predstavujú práve piny pre správny chod I2C komunikácie spomenutej už v kapitole 6.1 a kapitole 3.3. Z 8 pinov, ktoré môžu byť využité na tomto expandéry sa v tomto projekte budú využívať 3 a to piny GP0 GP1 a GP4 ako je už známe zo schémy zapojenia. Na všetky tieto spomenuté piny sú pripojené LED diódy s pull down zapojeným rezistorom a aj jedno tlačidlo taktiež s pull down rezistorom. LED dióda, ktorá je označená pod názvom ZELENA bude reprezentovať vlastne už spomínaný elektrický zámok, rozsvieti sa ak tvár bola detekovaná zato LED dióda, ktorá je označená názvom CERVENA sa rozsvieti, ak tvár nebola detekovaná a prístup sa vlastne neumožní. Na pine GP4 je dôležitejšie práve to tlačidlo označené S1, LED dióda s názvom TLACIDLK reprezentuje iba svetelnú signalizáciu

toho, že tlačidlo funguje správne. Tlačidlo S1 slúži ako externé tlačidlo, ktoré ak je stlačené v režime ak prebieha detekcia a rozpoznávanie tváre, tak bude daná tvár zaznamenaná a zapamätaná a bude jej následne umožnený vstup. Viac o tomto tlačidle a celkovo funkcionalitách tohto projektu bude popísané v nasledujúcej kapitole 7.

6.3 Zapúzdrenie vývojovej dosky

V kapitole 3.5 bolo spomenuté, že by sa pre tento projekt hodila aj nejaká krabička aby teda bola vývojová doska bezpečne uzavretá voči vonkajším vplyvom a taktiež aby sa mohla prakticky uchytiť na určené miesto.

Pri výbere krabičky bol použitý už existujúci dizajn zo stránky Thingiverse [20]. Je to webová stránka, na ktorej užívatelia zdieľajú svoje vlastné dizajny a poskytujú tam bezplatné návrhy hadrvéru. Vyskytujú sa tam hlavne veci pre 3D tlač, laserové frézy a podobne.



Obr. 6.3: Puzdro pre vývojovú dosku TTGO T-Camera

Funkcia celého systému je teda prezentovaná na nepájivom poli a jednotlivé komponenty sú prepojené káblíkmi.

7 Softvérové riešenie

V tejto kapitole bude popísané ako nainštalovať samotný ESP-IDF framework, z akých častí sa skladá celý kód a taktiež tu bude popísané vlastné riešenie funkčnosti systému.

7.1 Inštalácia ESP-IDF a ESP-WHO

Ako prvý krok by bolo dobre spomenúť ako nainštalovať spomínaný framework ESP-IDF a jeho platformu ESP-WHO. Keďže ESP-WHO pracuje už s frameworkom ESP-IDF, stačí získať iba ESP-WHO pretože ESP-IDF je v ňom zakomponovaný. ESP-WHO je možné získať zo stránky Github kde je vlastne aj popis ako ho správne stiahnuť[15]. Takže je potrebné otvoriť si konzolu a do nej napísať príkaz **git clone –recursive https://github.com/espressif/esp-who.git**. Slovíčko **recursive** je v tomto prípade dôležité pre správne stiahnutie všetkých priložených súborov a priečinkov. Ak sťahovanie prebehlo úspešne, je možné skúsiť jeden z viacerých príkladov, ktorý tento ESP-WHO obsahuje na overenie správnosti fungovania. Je dôležité podotknúť, že pri každom príklade je aj súbor **README.md**, v ktorom je detailnejšie popísané čo daný príklad robí a ako správne nastaviť parametre vývojovej dosky s ESP32 čipom.

Z dôvodu neobjasnených problémov pri implementovaní frameworku ESP-IDF do vývojového prostredia PlatformIO vo VSCode bola zvolená druhá varianta a to cez špeciálnu konzolovú aplikáciu ESP-IDF Tools, ktorú je možné stiahnuť z oficiálnej programátorskej príručky pre framework ESP-IDF [21]. Taktiež je potrebné mať jednu z novších verzií Pythonu. Potom pomocou príkazu **idf.py –help** je možné si zobrazíť najčastejšie používané príkazy. Na začiatku je dôležité byť v zložke, ktorú chceme buildnúť a následne nahráť do vývojovej dosky. Pomocou príkazu **cd** sa dá dostať do zvolenej zložky ktorú je potrebné buildnúť. Ešte pred samotným buildnutím je dôležité nastaviť si jednotlivé parametre. Pomocou príkazu **idf.py menu-config** sa otvorí okno, v ktorom je možné nastaviť určité parametre. V tomto okne boli nastavené piny čipu ESP32 pre konkrétnu vývojovú dosku, ktorá je v tomto projekte použitá a taktiež bol problém s tzv. "Partition table offset"ktorý bolo treba zmeniť z hodnoty 0x8000 na hodnotu 0x9000 pre správne nahratie programu do vývojovej dosky. Ak sú teda všetky parametre správne nastavené pomocou príkazu **idf.py build** je možné projekt buildnúť. Ak build prebehol bez problémov je ďalším krokom nahratie programu do vývojovej dosky a to pomocou príkazu **idf.py -p (COM) flash**, kde na miesto parametru COM je potrebné zvoliť port na ktorom je vývojová doska napojená k počítaču. Ak aj nahratie programu prebehlo úspešne, je možné sledovať, čo sa práve na vývojovej doske deje pomocou príkazu **idf.py -p**

(COM) **monitor**, kde opäť na miesto parametru COM je potrebné zvoliť port, na ktorom je vývojová doska napojená.

7.2 Štruktúra kódu

Tento projekt vychádza z modifikácie už existujúceho príkladu ESP-WHO nastavby s názvom `camera_web_server`. Pred samotnou modifikáciou tohto programu je nutné ešte nastaviť kameru na buď SoftAP alebo stacionárny mód popísaný v kapitole 4. V tomto prípade treba v súbore `sdkconfig` nastaviť SSID a heslo pre domovskú sieť. V tomto súbore je taktiež možnosť nastaviť aj už spomínaný Partition table offset alebo aj piny vývojovej dosky.

Celkový program sa skladá z viacerých častí. Medzi hlavné časti patria hlavičkové súbory (.h) a súbory so zdrojovým kódom (.c). Medzi hlavné súbory so zdrojovým kódom patria:

- `app_camera.c` - zdrojový súbor k nastaveniu kamery
- `app_httpd.c` - zdrojový súbor pre nastavenie serveru, detekcie a rozpoznávania
- `app_wifi.c` - zdrojový súbor pre nastavenie wifi
- `gpio.c` - zdrojový súbor pre nastavenie GPIO
- `mcp23008.c` - zdrojový súbor pre nastavenie I2C expandéru
- `app_main.c` - hlavný zdrojový súbor

Súbor `app_main.c` obsahoval iba volanie ostatných troch súborov so zdrojovým kódom. V tomto súbore je teda priestor na dopísanie vlastnej časti zdrojového kódu pre tento projekt. Taktiež bol modifikovaný aj súbor `app_httpd.h` a `app_httpd.c`, do ktorého boli dopísané funkcie na získanie určitých premenných pre následné spracovanie v hlavnom zdrojovom súbore. Zatiaľ, čo súbor `app_httpd.h` obsahuje iba samotné hlavičky doplnených funkcií, súbor `app_httpd.c` bol doplnený nasledovne:

Výpis 7.1: Náhľad do zdrojového súboru `app_httpd.c`

```
1 int getDetectionEnabled(){ return detection_enabled; }
2 void setDetectionEnabled(int new_value) {
3     detection_enabled = new_value;
4 }
5 int getRecognitionEnabled(){ return recognition_enabled; }
6 void setRecognitionEnabled(int new_value) {
7     recognition_enabled = new_value;
8 }
9 int getEnrollFace(){ return is_enrolling; }
10 void setEnrollFace(int new_value){ is_enrolling = new_value;}
11 int getMatchedID(){ return matched_id; }
12 int Detected(){ return detected; }
```


Tieto funkcie slúžia prevzatie určitých premenných zo súboru `app_httpd.c` tak, aby ich bolo možné v hlavnom zdrojovom súbore modifikovať. Jedná sa teda o premenné s názvami:

- `detection_enabled` - premenná ktorá hovorí o tom či je detekcia tváre aktívna
- `recognition_enabled` - zpremenená ktorá hovorí o tom či je rozpoznávanie tváre aktívne
- `is_enrolling` - premenná ktorá spúšťa nahrávanie novej tváre
- `matched_id` - premenná ktorá signalizuje počet načítaných tvári
- `detected` - premenná ktorá hovorí o tom, či bola tvár detekovaná

Premenné boli získavané z dôvodu, aby sa ponechali už vytvorené súbory v takom stave akom sú bez nejakých väčších zásahov do funkcionality. Ako boli tieto premenné využité bude popísané v kapitole 7.4.

7.3 Implementácia streamovania a rozpoznávania tváre

Pre správny chod funkcie streamovania obrazu musia byť splnené nasledujúce procesy. V prvom rade by mala byť nainicializovaná kamera. Na kamere je potrebné nainicializovať veci ako dátové piny kamery, piny pre I2C komunikáciu, frekvenciu alebo formát výstupného obrazu. Tieto všetky veci sa inicializujú v súbore `app_camera.c`. Keďže kamera funguje na princípe a je od výroby nastavená tak, že premieta obraz hore nohami, je potrebné hneď po inicializácii kameru vlastne otočiť po vertikálnej osi.

Ak je kamera nainicializovaná, treba taktiež nainicializovať aj pripojenie na WiFi a spustenie serveru. O inicializáciu WiFi pripojenie sa stará súbor s názvom `app_wifi.c`. V tomto súbore sa teda nastavujú veci ako IP adresa, samozrejme SSID a heslo danej siete a nastavenie stacionárneho režimu. Pre inicializáciu serveru slúži súbor `app_httpd.c`. Server začne bežať na dvoch portoch a to na porte 80 a porte 81. Port 80 zodpovedá webovému rozhraniu kde sa nachádzajú možnosti nastavovania kamery za ktorý zodpovedá už vytvorený HTML súbor s názvom `index_ov2640.html` alebo aj samotný stream ktorý zároveň beží na porte 81. Ak by bolo potrebné zobrazit iba stream videa trebalo by zadať IP adresu, za ktorou by nasledoval výber portu a prípona `/stream`. Táto prípona sa prezýva aj ako handler. Znamená to, že ak sa za IP adresu pridá nejaká prípona, na ktorú je vytvorený handler, vykoná sa nejaké špeciálne obsluženie tohoto príkazu. V súbore sú vytvorené nasledujúce handlers:

- `/` - slúži len na klasické zobrazenie webového rozhrania
- `/status` - slúži pre zobrazenie nastavených parametrov
- `/capture` - slúži na zachytenie aktuálneho výstupu kamery v podobe fotografie

- /stream - slúži na zobrazenie aktuálneho výstupu kamery v podobe videa

Ak sú teda všetky potrebné veci nainicializované postup je nasledovný. Pomocou kamery získame aktuálny obraz v podobe fotografie, ktorú následne program zkompresuje do kvality JPEG a odošle užívateľovi na daný port. Kamera teda vysiela vo formáte MJPEG čo je vlastne súbor viacerých fotografií za sebou nasledujúcich vo formáte JPEG. Keďže kamera dokáže vysielať v rôznych rozlíšeniach, nasledujúca tabuľka zobrazuje koľko FPS je schopná kamera spraviť a pri akom rozlíšení.

Tab. 7.1: Prehľad rozlíšení a fps kamery

Názov rozlíšenia	Rozlíšenie [pixel]	Priemerný počet obrázkov za sekundu [fps]
UXGA	1600x1200	5,8
SXGA	1280x1024	6,2
XGA	1024x768	6,4
SVGA	800x600	12,7
VGA	640x480	13,0
CIF	400x296	25,6
QVGA	320x240	25,6
HQVGA	240x176	25,6
QQVGA	160x120	25,6

Keďže ide o rozpoznávanie tváre pomocou modelu MTMN, tak ako už bolo spomenuté v kapitole 5.1.1 na ľudskej tvári nás teda zaujíma 5 bodov. Zo súboru `fr_forward.h` sa dá vyčítať, že tieto body sú pozícia ľavého oka, pozícia pravého oka, pozícia nosa, pozícia ľavej časti úst a pozícia pravej časti úst. Taktiež sú v tomto súbore nastavené aj prahové hodnoty pre vzdialenosť očí voči sebe a aj najmenšia a najväčšia vzdialenosť nosa voči očiam. Ak sú teda hodnoty týchto bodov v súlade s nastavenými prahovými hodnotami, tvár bude detekovaná a rozpoznaná. Ak je zapnutá iba funkcia detekcie tvár, v momente čo je tvár detekovaná spraví sa okolo nej ohraničenie v tvare obdĺžnika žltej farby ktorý znázorňuje kde sa tvár na obrázku nachádza. V prípade že je zapnutá aj funkcia rozpoznávania tváre, všetky tváre, ktoré nie sú zaregistrované pod príslušným ID číslom budú detekovaný červeným obdĺžnikom okolo ich tváre s prednastaveným popisom Intruder Alert, ktorý vraví o tom, že ide o cudziu osobu, ktorá nie je v systéme registrovaná. Akým povelením sa riadi funkcia pre zapnutie nahrávania tváre do systému bude popísané v kapitole 7.4. Ak teda príde povel na načítanie tváre do systému, systém si spraví 5 obrazových záznamov pre danú osobu a priradí jej dané ID číslo. Prvá načítaná osoba do systému bude mať teda ID o hodnote 0, ďalšia o hodnote 1 atď. Funkcia detekcie a rozpoznávania tváre funguje iba pri určitej hodnote rozlíšenia kvôli výkonu vývojovej dosky.

Na oficiálnej stránke ohľadom nastavby ESP-WHO je popísané, že optimálne rozlíšenie pre detekciu a rozpoznávanie tváre je QVGA s rozlíšením 320x240 pixelov. Najvyššie rozlíšenie, ktoré podporuje funkciu detekcie a rozpoznávania tváre je CIF s rozlíšením 400x296 pixelov, ktoré bolo použité aj v tejto práci. Pri rozlíšení CIF ako už je spomenuté v tabuľke v tejto kapitole dokáže kamera vyslať 25,6 snímky za sekundu. To ale neplatí pri zapnutí funkcie detekcie a rozpoznávania tváre. Pre toto nastavenie je kamera schopná vyslať len okolo 0,9 snímky za sekundu. Aj keď sa to možno zdá málo, kamera je schopná v priebehu dvoch sekúnd rozhodnúť o tom či je tvár rozpoznaná alebo nie čo je na kameru za pár dolárov celkom postačujúci výkon. Ďalšie zaujímavosti zistené počas skúšania tohoto systému budú popísané v kapitole 9.

7.4 Hlavný zdrojový kód

Ako už bolo spomenuté v tejto kapitole, súbor ktorý bol modifikovaný bol hlavne `app_main.c`, do ktorého bol dopísaný kód charakteristický pre správanie sa tohoto systému. Za začiatku kódu tohoto súboru sú implementované príslušné knižnice, sú tam aj definované GPIO piny a taktiež aj piny pre I2C komunikáciu. Celý súbor `app_main.c` bude samozrejme súčasťou príloh tohoto projektu. Z toho dôvodu bude tuna popísaná iba jeho hlavná logika, ktorá vyzerá nasledovne:

Výpis 7.2: Náhľad do zdrojového súboru `app_main.c`

```
1 while (1) {
2   if (gpio_get_level(GPIO_INPUT_IO_1) == 0) {
3     en = 1;
4     setDetectionEnabled(en);
5     setRecognitionEnabled(en);
6     while (gpio_get_level(GPIO_INPUT_IO_1) == 0);
7   }
8   while (en) {
9     if (getMatchedID() >= 0 && Detected() == 1) {
10      getMatchedID();
11      mcp23008_write_port(&mcp, 1);
12    }
13    if (getMatchedID() < 0 && Detected() == 1) {
14      getMatchedID();
15      mcp23008_write_port(&mcp, 2);
16    }
17    mcp23008_read_port(&mcp, &out);
18    button = (out & (1 << 4));
```

```

19     if (button == 1) setEnrollFace(en);
20     xThisWakeTime = xTaskGetTickCount();
21     if ((xThisWakeTime - xLastWakeTime) > interval) {
22         en = 0;
23     }
24 }
25 if (en == 0) {
26     xThisWakeTime = xTaskGetTickCount();
27     xLastWakeTime = xThisWakeTime;
28     setDetectionEnabled(en);
29     setRecognitionEnabled(en);
30     mcp23008_write_port(&mcp, 0);
31 }
32 }

```

Celý program teda beží v superslučke. Od riadku 2 až po riadok 7 sa nachádza podmienka, ktorá hovorí o tom, že ak bolo stlačené fyzické tlačidlo, ktoré sa nachádza na vývojovej doske, zapne sa funkcia detekcie tváre a zároveň aj rozpoznávania tvára. Fyzické tlačidlo, ktoré je na doske predstavuje klasické tlačidlo, ktoré sa nachádza na klasických domových zvončekoch. Riadok 6 slúži k tomu, aby sa príkazy, ktoré sú v podmienke vykonali iba raz a následne sa program zacyklil vo while cykle, z ktorého sa dostane po pustení tlačidla. Na riadkoch 8 až 24 sú vlastne príkazy, ktoré sa vykonávajú v prípade ak je detekcia a rozpoznávanie tváre zapnuté. Na začiatku sa nachádzajú hneď dve podmienky. Prvá hovorí vlastne o tom, čo sa stane ak bola tvár detekovaná a zároveň jej príslušné ID má hodnotu 0 alebo väčšiu. V tom prípade sa na výstupe I2C expandéru rozsvieti zelená LED dióda, ktorá je napojená na pin 0 I2C expandéra ako je zobrazené v kapitole 6.2. Druhá podmienka vraví o tom, čo sa stane ak tvár bola detekovaná ale nie je registrovaná v systéme. V tom prípade sa na výstupe I2C expandéra rozsvieti červená LED dióda signalizujúca to, že vstup bol danej osobe zamietnutý. Na riadku 17 sa vlastne čítajú hodnoty na výstupoch I2C expandéra a zapisujú sa do premennej **out**. Následne na riadku 18 sa pomocou bitovej logickej operácie AND vyberie štvrtý bit, na ktorom je napojené externé tlačidlo a do premennej **button** sa teda zapíše jeho hodnota. Podmienka na riadku 19 kontroluje či je tlačidlo stlačené a ak je, zapne sa funkcia pre zapamätanie tváre. Riadky 20 až 24 slúžia nato, aby vlastne od stlačenia tlačidla pre zapnutie detekcie a rozpoznávania tváre boli tieto funkcie zapnuté iba na určitý čas a po jeho uplynutí budú opäť vypnuté a kamera bude ďalej už len streamovať obraz. Je to varianta bez akýchkoľvek oneskorení, ktoré by mohli spôsobovať že celý kód by na chvíľu nevykonával žiadnu činnosť. Čas počas, ktorého je detekcia a rozpoznávanie tváre zapnuté bol nastavený z demonštračných dôvodov na 10 sekúnd a tento čas sa môže

prepísaním jednej premennej v kóde ešte viac predĺžiť alebo skrátiť. Riadky 25 až 31 hovoria o tom, čo sa diať ak je funkcia detekcie a rozpoznávania tváre vypnutá. Kamera bude vlastne iba naďalej streamovať obraz a všetky signalizačné LED diódy na výstupe I2C expandéra budú vypnuté.

8 Inteligentný domáci zvonček

Smart doorbell alebo po slovensky inteligentný zvonček je zariadenie, ktoré plní funkcionality klasického zvončeka ale zároveň je toto zariadenie pripojené na internet a obsahuje oveľa viac funkcií ako klasický zvonček. Zatiaľ, čo pri klasickom zvončeku návštevník stlačí tlačidlo a zvonček vydá zvukovú signalizáciu ako upozornenie že niekto sa nachádza pred dvermi, inteligentný zvonček obsahuje navyše aj kameru pomocou, ktorej je možné vidieť kto sa pred dvermi nachádza. Takže inteligentný zvonček slúži aj ako taká bezpečnostná kamera, ktorá môže mať taktiež aj rôzne typy pohybových snímačov, ako napríklad už spomínaný PIR senzor. Inteligentný zvonček umožňuje majiteľovi domu sledovať a dokonca aj hovoriť s návštevníkom pomocou vstavaného mikrofónu. V kombinácii s inteligentným zámkom je možné aj dvere otvoriť na diaľku.

Jeden z prvých inteligentných zvončekov predstavila spoločnosť Ring Video Doorbell v roku 2013. Od tejto doby sa do tohto priemyslu pustilo aj viacero firiem, ako napríklad Nest Hello s nejakými vlastnými funkciami. Ceny takýchto inteligentných zvončekov sa pohybujú niekde medzi 100 - 300\$. [22]

Inteligentný zvonček vytvorený v tejto práci má pár výhod a nevýhod. Medzi jeho jedinečné výhody patrí to, že je schopný rozpoznávať ľudí podľa tváre, čo zatiaľ väčšina inteligentných zvončekov nedokáže. Ďalšou výhodou môže byť aj cena, keďže celkový projekt stál niečo okolo 15-20€ oproti rôzne vyšším trojciferným sumám pri komerčných inteligentných zvončekoch. Nevýhodou môže byť fakt, že je s tým trochu viac práce, viac nastavovania, programovania a podobne. Tak ako komerčné inteligentné zvončeky môžu komunikovať s inteligentnou domácnosťou, dokáže to aj tento len ho treba zaintegrovať do nejakého systému ako je práve Home Assistant.

8.1 Home Assistant

Home Assistant je alternatíva voči komerčným inteligentným domácim hubom pre riadenie inteligentnej domácnosti. Je to vlastne open-source projekt pre lokálnu domácu sieť vytvorený komunitou. Výhodou Home Assistantu voči komerčným alternatívam je fakt, že celý systém beží lokálne a žiadne dáta sa nikam neposielajú. Je možné ho teda plne kontrolovať a monitorovať, čo môže byť v prípadoch kedy tento systém padne menšou nevýhodou. Samozrejme výhodou je opäť cena, keďže samotný program nestojí vôbec nič. Jediné náklady teda pre vytvorenie si vlastnej inteligentnej domácnosti idú do minipočítača, ktorý teda bude riadiť celú domácnosť. Odporúča sa použiť minipočítač Raspberry Pi. Celý postup inštalácie je uvedený na oficiálnej stránke Home Assistantu.[23]

V tomto projekte bol teda využitý aj Home Assistant ktorý bol nainštalovaný na minipočítač Raspberry Pi 4B s RAM pamäťou 2G,B ktorý pre správnu implementáciu Home Assistanta postačuje. Na oficiálnej stránke v záložke Integrations je taktiež uvedené ako do systému pridávať rôzne integrácie inteligentných komponentov. Pre integráciu inteligentného domáceho zvončeka vytvoreného v tomto projekte bola použitá integrácia s názvom MJPEG IP Camera.[24]

V konfiguračnom súbore s názvom configuration.yaml je teda pridať nasledujúcu časť kódu pre integráciu už existujúceho MJPEG streamu.

Výpis 8.1: MJPEG stream v prostredí Home Assistant

```
1 - platform: mjpeg
2   name: Camera
3   mjpeg_url: http://<IP ADRESS>:81/stream
```

Do kolónky `mjpeg_url` je teda potrebné zadať IP adresu na ktorej tento inteligentný zvonček vysiela stream, a potom stačí už len v záložke Prehľad pridať entitu Náhľad obrázka kde je potreba zvoliť tento inteligentný zvonček ako Camera Entity. Inteligentný zvonček je teda zakomponovaný do systému pre riadenie celej inteligentnej domácnosti.

9 Dosaiahnuté výsledky

Úlohou tejto práce bolo zostrojiť funkčnú demonštračnú aplikáciu inteligentného domáceho zvončeka. Vytváranie tejto demonštračnej aplikácie sa určite nezaobišlo bez nejakých komplikácií. Problémy sa vyskyli tak ako pri komunikáciach, tak aj pri samotnom fungovaní systému. Výsledkom tejto práce je ale funkčný prototyp inteligentného domáceho zvončeka, ktorý ale bohužiaľ neobsahuje elektrický zámok takže jeho činnosť sa demonštruje na LED diódach. V tejto práci sa teda podarilo oživiť vývojovú dosku TTGO T-Camera s mikrokontrolérom ESP32-WROVER-B, do ktorého bola následne implementovaná nadstavba ESP-WHO, ktorá zabezpečuje streamovanie obrazu ale aj detekciu a rozpoznávanie tváre. Cieľom tejto práce bolo teda prispôbiť tento už funkčný systém pre konkrétne účely. Tento inteligentný zvonček by mal byť s pridaným elektrickým zámkom plne funkčný. Ako už bolo spomenuté v kapitole 7.3 tento zvonček dokáže rozpoznať osobu do približne 2 sekúnd. Problémy ale nastávajú ak je kamera vystavená zlým svetelným podmienkam. V tom prípade systém nedokáže správne pracovať a tým pádom ani rozpoznávať tvár. Z toho dôvodu by bolo dobré tento zvonček používať s nejakým pridaným osvetlením, aby mohlo byť rozpoznávanie účinné aj za tmy. Ďalším zistením bolo, že ako je systém nastavený, tak problémy nastávajú pri zmene pózy dotyčnej osoby. Vo väčšine prípadov to ale fungovalo bez problémov, keďže sa berie do úvahy fakt, že dotyčný ak zazvoní na tento zvonček tak sa pozrie priamo čelne do kamery a systém nemá problém detekovať čelné pohľady. Problémom je ale fakt, že vývojová doska obsahuje iba jeden kamerový modul a to znamená, že nie je možné určiť hĺbkový profil čo spôsobuje fakt, že zvonček je možné oklamať fotografiou. Toto zistenie je asi jeden z najväčších problémov tohoto systému, no funguje to len za predpoklady, ak je pred kameru vystavená fotografia s rovnakou čelnou pózou. Kvôli tomuto zisteniu by nemal tento inteligentný zvonček slúžiť ako primárne zariadenie pre zabezpečenie bytu alebo domu. Taktiež skúšaním bolo zistené že zariadenie dokáže detekovať tváre z minimálnej vzdialenosti 20cm a z maximálnej vzdialenosti približne 80cm. Riešenie nahrávania nových tvárí je zatiaľ spravené pomocou externého tlačidla, čo by v budúcnosti mohlo byť nahradené nejakou aplikáciou. Aj napriek určitým problémom a zisteniam je táto demonštračná aplikácia inteligentného zvončeku plne funkčná.

Záver

Cielom tejto bakalárskej práce bolo vytvoriť demonštračnú aplikáciu predstavujúcu funkčnosť inteligentného domáceho zvončeka. V prvej časti tejto práci bola teda spravená rešerš z rôznych vedeckých internetových článkov ohľadom detekcie tváre a jej metódach. Práca pokračuje popisáním použitého mikrokontroléru ESP32, jeho hlavných vlastností, konkurencie a podporovaných frameworkov pre programovanie tohto mikrokontroléru. Následne bolo vyhotovené riešenie celého systému so špecifikáciou požiadavok pre správny chod tohoto systému. Celá práca bola zo začiatku postavená na frameworku Arduino, ktorý ale nebol vhodnou voľbou pre túto prácu. Práca teda spolu už s dosiahnutými výsledkami musela byť prerobená na druhý použiteľný framework ktorým bol ESP-IDF. Po prerobení systému pre prácu s frameworkom ESP-IDF bolo treba si ozrejmiť čo predstavuje nadstavba ESP-WHO a ako sa používa. Táto práca popisuje ako nadstavbu ESP-WHO stiahnuť a implementovať do zariadenia čo prebehlo bez nejakých väčších chýb. Práca taktiež popisuje vnútornú štruktúru spomínanej nadstavby ESP-WHO a na akých rozpoznávacích modeloch táto nadstavba funguje. Následne bol vytvorený rozbor hardvérovej časti riešenia kde popísaná zvolená vývojová doska. Po bližšom preskúmaní tejto dosky bolo však zistené, že je za potreby zaimplementovať rozšírenie pre komunikáciu čo bolo dosiahnuté pomocou I2C expandéru. Práca teda popisuje ako zvolený I2C expandér zapojiť k vývojovej doske pre správnu funkčnosť systému. Po vyriešení problémov s hardvérom prišiel na rad softvér. Práca ďalej popisuje ako vyzerá štruktúra kódu použitého pre tento systém, jeho hlavné súbory a dôležité premenné ktoré je potrebné nastaviť. Pre konkrétnu činnosť tohoto systému bol upravovaný hlavný súbor so zdrojovým kódom, ktorý bol prispôsobený pre konkrétnu činnosť tohto projektu. Po vysvetlení hlavného fungovania systému, bola spravená menšia rešerš na tému inteligentných zvončekov. Táto práca teda v poslednom rade popisuje ako takýto systém implementovať do domáceho inteligentného systému Home Assistant. Výstupom tejto práci je teda funkčná demonštračná aplikácia predstavujúca funkčnosť inteligentného zvončeka.

Literatúra

- [1] NEC, *World-leading Face Recognition for Multiple Industries* [online]. [cit. 28. 10. 2019]. Dostupné z URL: <https://www.nec.com/en/global/solutions/biometrics/face/index.html>.
- [2] Gemalto, *Facial recognition: top 7 trends (tech, vendors, markets, use cases and latest news)* [online]. poslední aktualizace 15. 12. 2019 [cit. 28. 10. 2019]. Dostupné z URL: <https://www.gemalto.com/govt/biometrics/facial-recognition>.
- [3] NRF, *Retailers see increase in organized retail crime* [online]. poslední aktualizace 18. 10. 2016 [cit. 28. 10. 2019]. Dostupné z URL: <https://nrf.com/media-center/press-releases/retailers-see-increase-organized-retail-crime>.
- [4] Wikipedia, *Forezná veda* [online]. poslední aktualizace 22. 11. 2017 [cit. 28. 10. 2019]. Dostupné z URL: https://sk.wikipedia.org/wiki/Forezn%C3%A1_veda.
- [5] Facefirst, *WHAT IS FACE RECOGNITION?* [online]. [cit. 28. 10. 2019]. Dostupné z URL: <https://www.facefirst.com/face-recognition-glossary/what-is-face-recognition/>.
- [6] TowardsDataScience, *Face Detection For Beginners* [online]. poslední aktualizace 27. 4. 2018 [cit. 5. 1. 2020]. Dostupné z URL: <https://towardsdatascience.com/face-detection-for-beginners-e58e8f21aad9>.
- [7] Researchgate, *A Review on Face Detection Methods* [online]. poslední aktualizace 1. 2. 2011 [cit. 8. 6. 2020]. Dostupné z URL: https://www.researchgate.net/publication/257338580_A_Review_on_Face_Detection_Methods.
- [8] Espressif, *About Espressif* [online]. poslední aktualizace 27. 4. 2018 [cit. 4. 11. 2019]. Dostupné z URL: <https://www.espressif.com/en/company/about-us/who-we-are>.
- [9] Wikipedia, *ESP8266* [online]. poslední aktualizace 25. 12. 2019 [cit. 4. 11. 2019]. Dostupné z URL: <https://en.wikipedia.org/wiki/ESP8266>.

- [10] Wikipedia, *ESP32* [online]. poslední aktualizace 16.12.2019 [cit. 4. 11. 2019]. Dostupné z URL: <<https://en.wikipedia.org/wiki/ESP32>>.
- [11] Hackster.io, *Face Detection and Recognition on the ESP32* [online]. [cit. 4. 11. 2019]. Dostupné z URL: <<https://www.hackster.io/news/face-detection-and-recognition-on-the-esp32-3b4b9a35c765>>.
- [12] Wikipedia, *Passive infrared sensor* [online]. poslední aktualizace 4.6.2020 [cit. 8. 6. 2020]. Dostupné z URL: <https://en.wikipedia.org/wiki/Passive_infrared_sensor>.
- [13] Waveshare, *Features* [online]. [cit. 2. 1. 2020]. Dostupné z URL: <<https://www.waveshare.com/ov2640-camera-board.htm>>.
- [14] TSSGROUP, *Popis* [online]. [cit. 2. 1. 2020]. Dostupné z URL: <<https://www.tssgroup.sk/dorcas-41aa-flex-elektricky-zamok/>>.
- [15] Github, *ESP-WHO* [online]. [cit. 11. 11. 2019]. Dostupné z URL: <<https://github.com/espressif/esp-who>>.
- [16] ITNEXT, *Realtime JavaScript Face Tracking and Face Recognition using face-api.js' MTCNN Face Detector* [online]. poslední aktualizace 16.7.2018 [cit. 8. 6. 2020]. Dostupné z URL: <<https://tinyurl.com/yxzbhyld>>.
- [17] Cornell University, *MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications* [online]. poslední aktualizace 17.4.2014 [cit. 8. 6. 2020]. Dostupné z URL: <<https://arxiv.org/abs/1704.04861>>.
- [18] Aliexpress, *LILYGO® TTGO T-Camera ESP32 WROVER & PSRAM Camera Module ESP32-WROVER-B OV2640 Camera Module 0.96 OLED* [online]. [cit. 8. 6. 2020]. Dostupné z URL: <<https://tinyurl.com/ybtjv5gh>>.
- [19] TME, *MCP23008-E/P MICROCHIP TECHNOLOGY - TECHNICKÉ INFORMÁCIE* [online]. [cit. 8. 6. 2020]. Dostupné z URL: <https://www.tme.eu/sk/details/mcp23008-e_p/multiplexory-a-analogove-prepinace/microchip-technology/>.
- [20] Thingiverse, *ESP32 WRover TTGO camera box by hermanb* [online]. poslední aktualizace 4. 4. 2019 [cit. 8. 6. 2020]. Dostupné z URL: <<https://www.thingiverse.com/thing:3540059>>.

- [21] ESPRESSIF, *Get Started* [online]. [cit. 8. 6. 2020]. Dostupné z URL:
<<https://docs.espressif.com/projects/esp-idf/en/latest/esp32/get-started/index.html>>.
- [22] Wikipedia, *Smart doorbell* [online]. poslední aktualizace 11.2.2020 [cit. 8. 6. 2020]. Dostupné z URL:
<https://en.wikipedia.org/wiki/Smart_doorbell#cite_note-3>.
- [23] Home Assistant, *Get Started* [online]. [cit. 8. 6. 2020]. Dostupné z URL:
<<https://www.home-assistant.io/getting-started/>>.
- [24] Home Assistant, *MJPEG IP Camera* [online]. [cit. 8. 6. 2020]. Dostupné z URL:
<<https://www.home-assistant.io/integrations/mjpeg/>>.

Zoznam symbolov, veličín a skratiek

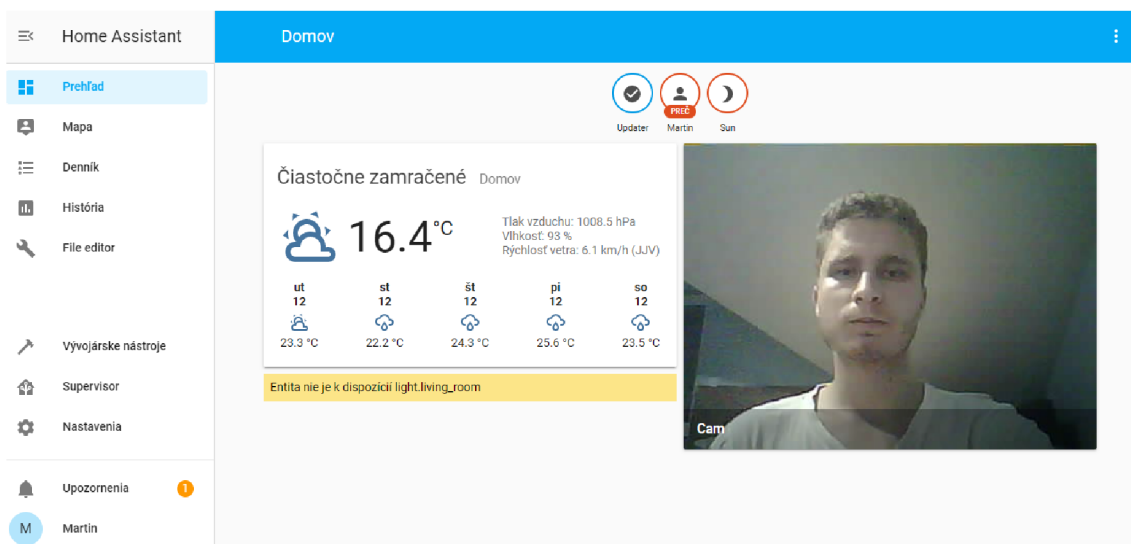
Mpx	Megapixel
GPIO	vstupno-výstupné piny – General-purpose input/output
VSCoDe	Visual Studio Code
PIR	pasívny infračervený senzor – Passive Infrared
DC	napájanie jednosmerným prúdom – Direct current
SPI	sériové periférne rozhranie – Serial Peripheral Interface
RAM	pamäť s priamym prístupom – Random-access memory
I2C	dvojžilová obojsmerná zbernica – Inter-Integrated Circuit
MTCNN	viacúčelové kaskádové konvolučné neurónové siete – Multi-task Cascaded Convolutional Neural Networks
CNN	kaskádové konvolučné neurónové siete – Cascaded Convolutional Neural Networks
SSID	názov WiFi siete – Service set identifier
WiFi	bezdrôtová sieť – Wireless fidelity
IP	internetový protokol – Internet Protocol
FPS	snímky za sekundu – Frames per second

Zoznam príloh

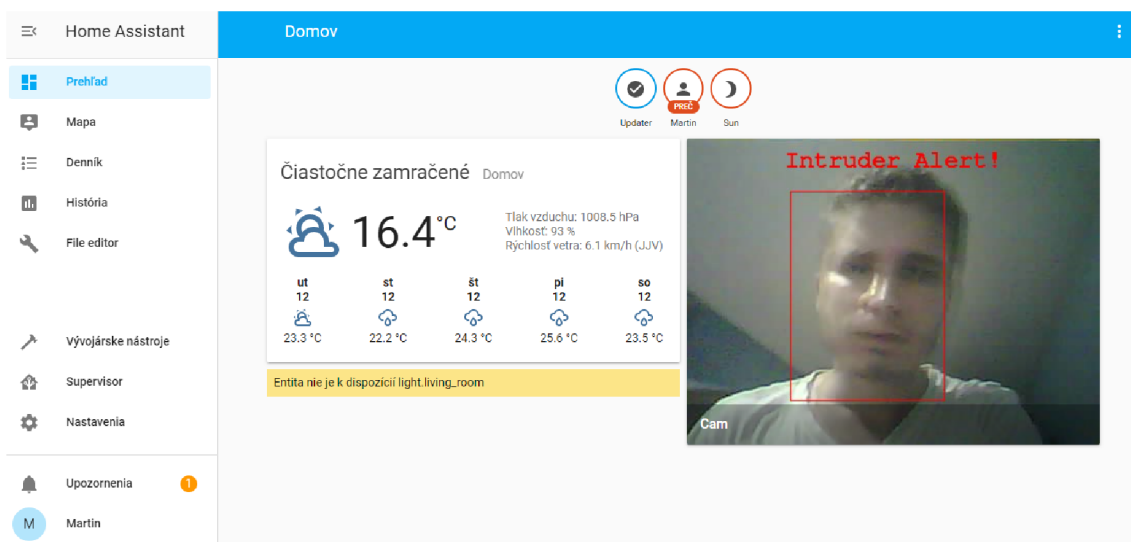
A	Obrázky z prostredia Home Assistant	50
A.1	Prostredie Home Assistant	50
B	Fotografie projektu	52
C	Obsah príloženého CD	53

A Obrázky z prostredia Home Assistant

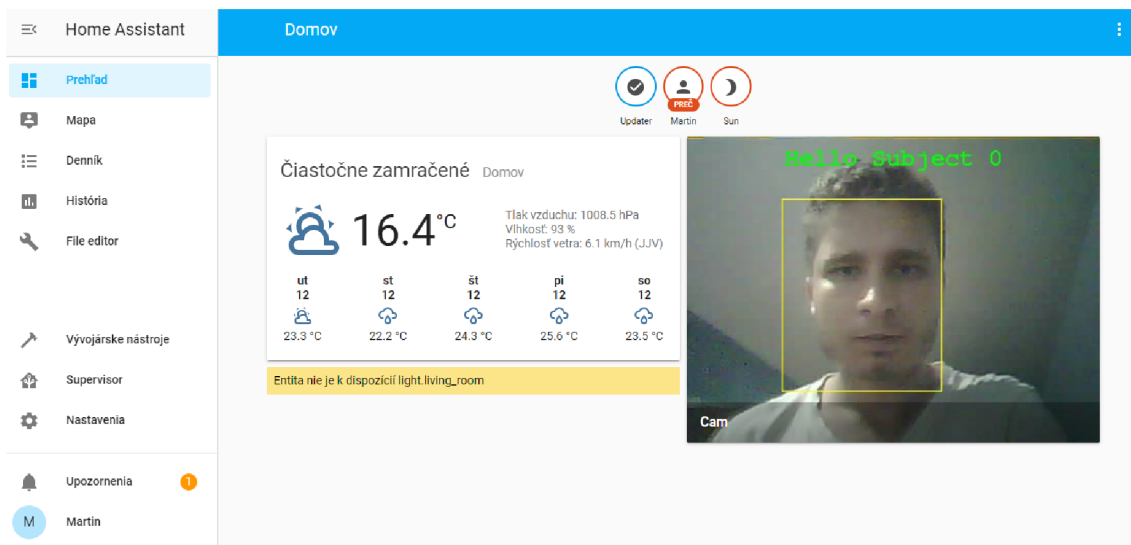
A.1 Prostredie Home Assistant



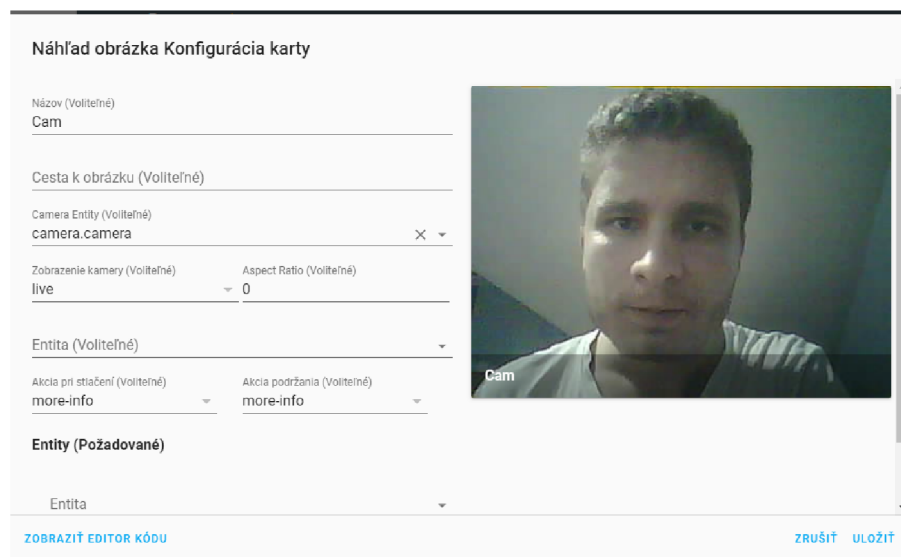
Obr. A.1: Zobrazenie tzv. Lovelace v prostredí Home Assistant



Obr. A.2: Zobrazenie osoby bez prístupu

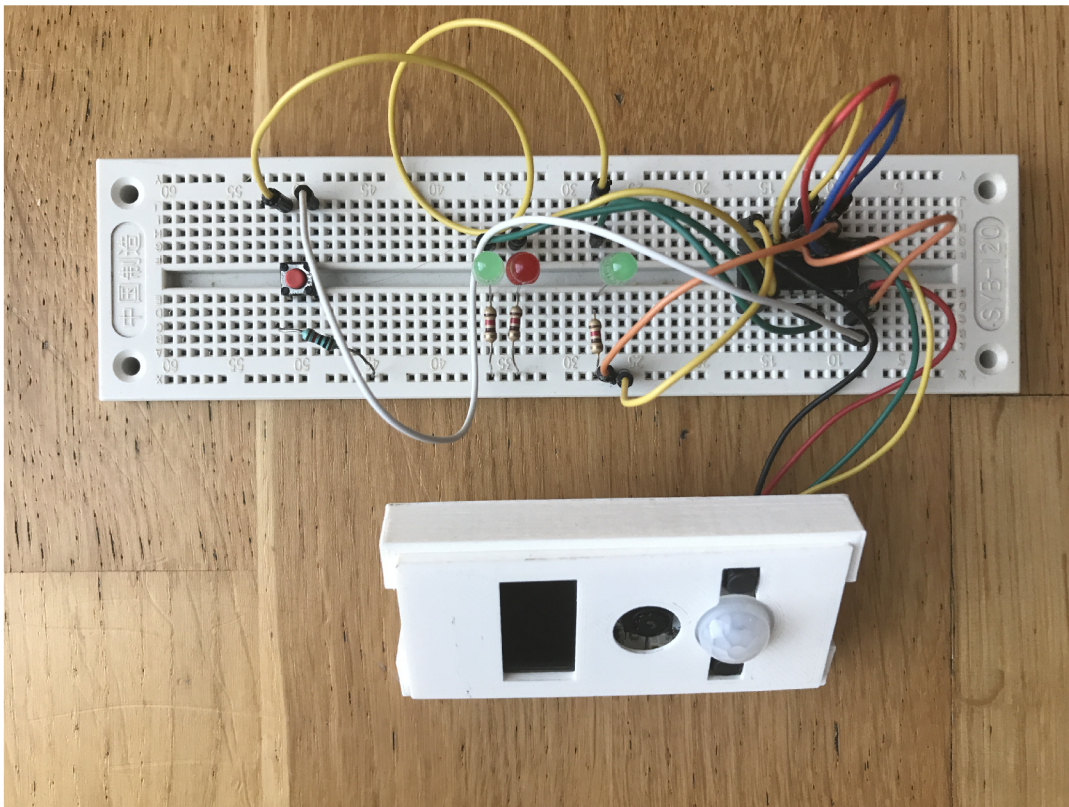


Obr. A.3: Zobrazenie osoby s prístupom



Obr. A.4: Zobrazenie nastavenia entity

B Fotografie projektu



Obr. B.1: Zobrazenie zapojenia celkového systému.

C Obsah přiloženého CD

```
/ ..... kořenový adresár přiloženého CD
├── Bakalarska_praca
│   └── Bakalarska_praca-Martin_Musinka.pdf
├── Software.zip
│   └── main
│       ├── include ..... hlavičkové súbory
│       │   ├── app_camera.h
│       │   ├── app_httpd.h
│       │   ├── app_wifi.h
│       │   ├── gpio.h
│       │   ├── mcp23008.h
│       │   └── sensor.h
│       ├── www ..... webové rozhranie
│       │   ├── compress_pages.sh
│       │   ├── index_ov2640.html
│       │   └── index_ov2640.html
│       ├── app_camera.c
│       ├── app_httpd.c
│       ├── app_main.c
│       ├── app_wifi.c
│       ├── CMakeLists.txt
│       ├── component.mk
│       ├── Kconfig.projbuild
│       └── mcp23008.c
├── CMakeLists.txt
├── Makefilep
├── partitions.csv
├── README.md
├── sdkconfig
├── sdkconfig.defaults
└── sdkconfig.old
```