VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ
DEPARTMENT OF COMPUTER SYSTEMS

INTELIGENTNÍ PŘÍSTUPOVÝ TERMINÁL NA PLATFORMĚ ESP32

INTELLIGENT ACCESS TERMINAL USING ESP32 PLATFORM

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — **ŠIMON POMYKAL**
AUTHOR

VEDOUCÍ PRÁCE — **Ing. VÁCLAV ŠIMEK**
SUPERVISOR

BRNO 2021

Vysoké učení technické v Brně
Fakulta informačních technologií

Ústav počítačových systémů (UPSY) Akademický rok 2020/2021

Zadání bakalářské práce

Student: **Pomykal Šimon**
Program: Informační technologie
Název: **Inteligentní přístupový terminál na platformě ESP32**
Intelligent Access Terminal Using ESP32 Platform
Kategorie: Vestavěné systémy

Zadání:

1. Seznamte se s problematikou přístupových systémů řídících vstup osob do hlídaných prostor. Zpracujte přehled existujících řešení a stručně shrňte jejich klíčové vlastnosti.
2. S ohledem na poznatky z bodu 1) zadání se dále zabývejte vybranými typy přístupových terminálů sloužících k ověření identity osob. Pozornost věnujte zejména různým typům použitých senzorů.
3. Prostudujte techniky vhodné pro rozpoznávání obličeje ze statického obrazu. Zaměřte se na možnosti poskytované knihovnou TensorFlow a frameworky na bázi neuronových sítí či jiných vhodných klasifikátorů.
4. Podrobně se zabývejte vestavěnou platformou ESP32. Zaměřte se na její technické parametry, principy tvorby aplikací a dostupné vývojové nástroje.
5. Navrhněte řešení přístupového terminálu, jehož koncepce bude založena na využití platformy ESP32, vhodně zvolených senzorových modulech snímajících autentizovaný subjekt a softwarové části vyhodnocující senzorická data.
6. Proveďte technickou realizaci přístupového terminálu s využitím konceptu z bodu 5) zadání. V potřebném rozsahu implementujte obslužný firmware pro prvky navrženého řešení.
7. S využitím poznatků z bodu 3) zadání implementujte aplikaci vyhodnocující senzorická data o autentizované osobě získaná prvky přístupového terminálu.
8. Vhodným způsobem demonstrujte funkčnost realizovaného řešení a vyhodnoťte dosažené výsledky. Pokuste se navrhnou případná rozšíření či vylepšení.

Literatura:

- Dle pokynů vedoucího.

Pro udělení zápočtu za první semestr je požadováno:

- Splnění bodů 1 až 5 zadání.

Podrobné závazné pokyny pro vypracování práce viz https://www.fit.vut.cz/study/theses/

Vedoucí práce: **Šimek Václav, Ing.**
Vedoucí ústavu: Sekanina Lukáš, prof. Ing., Ph.D.
Datum zadání: 1. listopadu 2020
Datum odevzdání: 12. května 2021
Datum schválení: 30. října 2020

Abstrakt

Cílem této práce je navrhnout levné řešení inteligentního přístupového systému založeného na platformě esp32. Toto řešení bude zaměřeno zejména na zabezpečení rodinných domů, bytů, garáží, zahrad apod. Přístupový systém je tvořen modulem terminálu, určeným pro autentizaci osob pomocí senzoru otisku prstu a kamerami, které monitorují vymezenou vstupní oblast daného objektu. Tyto moduly jsou připojeny ke cloudu pomocí služby AWS IoT Core. Další částí systému je cloudová aplikace, jenž vyhodnocuje data z jednotlivých modulů. Přístupový systém je primárně navržen jako součást domácího zabezpečovacího systému, ale může být do jisté míry využit i samostatně.

Abstract

The aim of this thesis is to design cheap intelligent access control system based on esp32. This system is designed for use in family houses, flats, garages, gardens etc. The designed system is composed of access control terminal module which uses fingerprint reader to authenticate people and of camera modules which monitor the area of entry These modules are connected to cloud using AWS IoT Core. Another part of the system is a cloud application which evaluates data from the system. The acces control system is meant to be part of a home security system, but can be used independently to some extent.

Klíčová slova

bezpečnostní systém, přístupový terminál, rozpoznávání obličejů, Tensorflow, MQTT, Cloud, Docker, Amazon Web Services

Keywords

home security system, access control system, face recognition, Tensorflow, MQTT, Cloud, Docker, Amazon Web Services

Citace

POMYKAL, Šimon. *Inteligentní přístupový terminál na platformě ESP32*. Brno, 2021. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Václav Šimek

Inteligentní přístupový terminál na platformě ESP32

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Václava Šimka Další informace mi poskytl Mgr. Martin Vychodil z firmy Espressif Systems. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....
Šimon Pomykal
18. května 2021

Poděkování

Rád bych poděkoval vedoucímu práce Ing. Václavu Šimkovi a Mgr. Martinu Vychodilovi z firmy Espressif Systems za odbornou pomoc při řešení této práce.

Obsah

Seznam obrázků

Kapitola 1

Úvod

Zatímco dříve postačil k zabezpečení objektu prostý zámek, nároky moderní doby jsou výrazně vyšší. Je třeba střežit nejen osobní vlastnictví, ale i omezit přístup do kritických částí zařízení, jakými jsou například řídicí místnosti v elektrárnách, nebo kanceláře obsahující cenné papíry. Tyto skutečnosti daly podnět ke vzniku sofistikovaných bezpečnostních systémů. Důležitou součástí bezpečnostního systému, střežící vstup do hlídaného objektu je přístupový terminál.

Tato práce je součástí tří externích zadání od firmy Expressif Systems - přístupový terminál, senzorový systém a kontrolní část systému s uživatelským rozhraním, které spolu tvoří kompletní domácí bezpečnostní systém.

Cílem této bakalářské práce je navrhnout levné řešení přístupového terminálu, který bude součástí domácího bezpečnostního systému. Tento terminál bude kontrolovat přístup osob do hlídaného objektu. Důraz je při výběru autentizační metody kladen zejména na její pohodlnost, společně s dostačující úrovní zabezpečení. Na základě těchto vlastností je proto zvolena metoda autentizace, využívající snímač otisků prstů. Tento primární bezpečnostní prvek přístupového systému je rozšířen o kamerové moduly, které monitorují vstupní oblast. Snímky pořízené kamerami v případě pohybu ve střeženém prostoru a údaje o činnosti modulu terminálu jsou odesílány do serverové části bezpečnostního systému, kde jsou dále zpracovány a uloženy. Za účelem zachování nízké ceny a vzhledem k jednodušší integraci s ostatními částmi bezpečnostního systému je jeho serverová část umístěna do cloudu. Součástí této bakalářské práce je také implementovat aplikaci, která provádí rozpoznávání obličeje ze snímků z kamer.

Téma této bakalářské práce mě zaujalo zejména z toho důvodu, že je velmi prakticky zaměřené. Zároveň, výsledkem implementace jsou realizované moduly, které demonstrují funkčnost řešení. Celý systém lze do budoucna rozšířit, nebo na něj navázat v dalším projektu.

V první kapitole je stručně shrnuta dosavadní situace týkající se bezpečnostních systémů. Dále jsou rozebrány různé metody autentizace, které lze v přístupových systémech použít. Následující kapitola obsahuje stručný úvod do problematiky rozpoznávání obličeje, přehled možných přístupů a popis knihovny TensorFlow. Poslední teoreticky zaměřená kapitola se zabývá platformou esp32.

Následuje praktická část, která se nejprve zabývá návrhem přístupového systému a popisem jeho vlastností. Dále je podrobně popsán implementační proces přístupového systému. Finální část práce řeší realizaci demonstračních modulů a popisuje vybrané testy systému.

Kapitola 2

Přístupové systémy

V této kapitole jsou nejprve popsány bezpečnostní přístupové systémy, speciálně pak domácí bezpečnostní přístupové systémy. Následně je shrnuta nabídka těchto systému na trhu. V druhé části kapitoly jsou rozvedeny možné způsoby autentizace osob. V této kapitole jsem čerpal z „Security Patterns for Physical Access Control Systems“ [6]. V podkapitole o PIR pohybovém senzoru jsem čerpal z jeho dokumentace [7].

Přístupové systémy jsou jednou z možností zabezpečení fyzických objektů. Potřeba chránit majetek a chránit důležité části infrastruktury jako letiště, elektrárny, vládní instituce a mnohé další, vytvořila velkou poptávku po řízení přístupu do těchto kritických objektů. Nejdříve byly bezpečnostní přístupové systémy zaměřeny spíše v oblastech jako průmysl, služby nebo doprava.

V dnešní době je často přístupový terminál součástí domácích bezpečnostních systémů. Tyto bezpečnostní systémy monitorují vnitřní a vnější prostory domu pomocí různých senzorů. Nejčastěji používaným typem senzoru je PIR pohybové čidlo, s jehož pomocí je možné detekovat pohyb osob a zvířat. Systémy používají i typy senzorů, například za účelem detekce oxidu uhličitého, kouře nebo vody.

Firmy jako Noabe (dříve Jablotron), BEDO Ajax, Smart hitech Software Solutions nebo LifeSmart poskytují domácí bezpečnostní systémy, jenž se většinou skládají z centrální bezpečnostní jednotky a modulů, které jsou k ní připojeny pomocí Wi-Fi. Tyto moduly lze libovolně kombinovat podle potřeby. Díky tomu je možné si bezpečnostní systém do jisté míry přizpůsobit.

Přístupové terminály lze také zakoupit samostatně. V tomto případě přístupové terminály často slouží nejen k zabezpečení přístupu, ale také k evidenci docházky osob do zaměstnání. Tento typ terminálů je určen primárně pro pracovní prostředí. Využita je celá škála autorizačních metod, v některých případech je jich použito několik zároveň. Nejčastěji se lze setkat s kombinací hesla a RFID[1] tokenu. Některé takto samostatně prodávané terminály lze integrovat do jiných bezpečnostní systémů, jiné s okolím nekomunikují vůbec a slouží pouze pro otevření dveří. Na základě poskytnutých funkcí, způsobu autentizace a schopnosti integrace se cena přístupových terminálů výrazně liší.

Přístupový terminál KT12M je příkladným zástupcem nejlevnější cenové kategorie. Tento terminál je vybaven dotykovou klávesnicí a RFID čtečkou. Je možné jej napájet pomocí libovolného 12V stabilizovaného zdroje. Funkcionalita tohoto terminálu zahrnuje pouze ovládání elektronického zámku a zvonku.

[1] RFID - Radio-frequency identification

Variantou přístupového terminálu, využívajícího čtečku otisků prstů je například model SF7. Tento model poskytuje podobnou funkcionalitu jako KT12M, hlavním rozdílem je nahrazení dotykové klávesnice za čtečku otisků prstů, přičemž cena tohoto modelu je několikanásobně vyšší. Oba tyto modely nelze integrovat do rozsáhlejších bezpečnostních systémů a poskytují pouze velmi omezenou funkcionalitu.

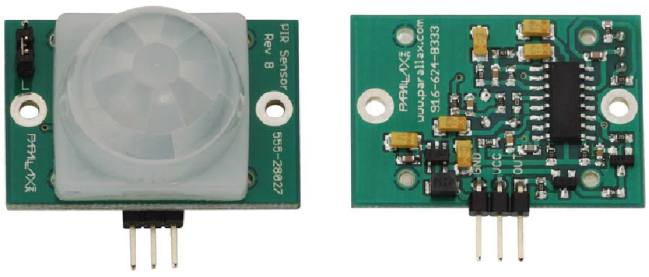
Na rozdíl od již zmíněných přístupový terminálů, nabízí model Zoneway T501 velkou škálu autorizačních prvků - RFID, klávesnice, čtečka otisků prstů, duální kamera a systém rozpoznávání obličeje. Navíc je tento modul možno využit buď samostatně (ovšem s jistými omezeními) nebo jako součást domácího bezpečnostního systému. K tomuto účelu terminál implementuje potřebné rozhraní. Ovšem cena za tyto služby je řádově větší než u předchozích systémů. V případě plného využití potenciálu terminálu je třeba řídicí modul. Celková cena je pak několikanásobně vyšší.

PIR pohybové čidlo

Passive Infra-Red (PIR) pohybové čidlo je typ senzoru, který se využívá v převážné většině bezpečnostních systémů. Tento senzor je schopen detekovat změnu intenzity dopadajícího infračerveného záření, které vydávají těla lidí a zvířat. Tohoto principu lze využít různými způsoby - otevírání dveří, spouštění alarmu a podobně. PIR čidlo může být využito jak samostatně, tak v kombinaci s jinými typy senzorů nebo kamer, které jsou řízeny na základě informací z PIR čidla.

Konkrétní model PIR pohybového čidla, který je využit v tomto projektu je, PIR Sensor with LED Signal od firmy Parallax Inc. Hlavní znaky tohoto čidla jsou následující:

- Detekce osob do 9 metrů nebo do 4,5 metrů v režimu snížené citlivosti.
- Napájecí napětí 3-6 V, proud 0,13 mA v klidovém režimu a 3 mA v aktivním.
- LED dioda pro vizuální zpětnou vazbu.
- Malá velikost.
- Jednoduché výstupní rozhraní.
- Komunikace pomocí jedno-bitového high/low výstupu.



Obrázek 2.1: PIR pohybové čidlo s LED signálem od firmy Parallax Inc [7].

2.1 Autentizace osob

V této podkapitole jsem čerpal z článku „A Review Of Authentication Methods“ [5]. V odstavci o RFID jsem čerpal z článku „THE RFID TECHNOLOGY AND ITS APPLICATIONS: A REVIEW“ [3]. V části o snímači otisků prstů jsem čerpal z jeho dokumentace [8].

Autentizace osoby je proces ověření její totožnosti. K tomuto lze využít velké množství metod. Autentizace osoby v kontextu bezpečnostního systému znamená - ověření bezpečnostních údajů získané od dané osoby vůči databázi systému. Pokud je tato akce úspěšná a systém vyhodnotí osobu jako autorizovanou, potom je jí na základě tohoto vyhodnocení umožněn přístup do hlídaného objektu. Metody autentizace osoby se dají rozdělit do tří kategorií podle poskytnutých informací:

- **Autentizace na základě znalosti** - Osoba poskytne systému tajnou informaci, známou pouze jí systému (typicky heslo, PIN).
- **Autentizace na základě vlastnictví** - Osoba prokáže svou totožnost pomocí nějakého fyzického objektu (token, ID karta, čip).
- **Autentizace na základě fyziologických rysů** - Systém identifikuje osobu na základě unikátních fyziologických rysu (otisk prstu, snímek rohovky oka, obličej).

2.1.1 Autentizace na základě znalostí

Míra zabezpečení pomocí této metody je přímo spjata s délkou tajné informace, neboli hesla, které je je použito. Pokud je použito dostatečně silné heslo je velmi časově náročné ho prolomit. Komplikované, dlouhé heslo se však uživateli špatně pamatuje. Nevýhodou hesla je také skutečnost, že jeho vyzrazení je ovlivněno lidským faktorem. Může být ukradeno pokud si jej člověk někam zapíše, může být vyzrazeno nebo zapomenuto.

Proces autentizace pomocí hesla je umožněn prostřednictvím klávesnice nebo dotykového displeje, pomocí kterého osoba heslo do systému zadá.

Důležitým prvkem při tomto způsobu autentizace je síla hesla, která udává jak odolné je heslo vůči Brute-force[2] útoku. Sílu hesla lze určit podle dvou znaků:

- **Kardinalita** - Počet znaků, který je možné při tvorbě hesla použít (včetně velkých písmen a speciálních znaků).
- **Entropie** - Síla hesla v bitech, Každý znak má určitou hodnotu. Sečtením těchto hodnot získáme sílu v bitech.

Například heslo o délce 8 znaků, kardinalitou 94 má entropii rovnu 52,74 bitu. Počítač je schopen prolomit toto heslo pomocí Brute-Force do 20 minut, kdežto prolomit heslo o délce 12 znaků, kardinalitou 94 a entropií 78,7 bitu by počítači trvalo 3018 let.

2.1.2 Autentizace na základě vlastnictví

Tato metoda ověřuje uživatele pomocí čipu, karty nebo tokenu. Na tomto fyzickém médiu je uložen certifikát, který vlastníka jednoznačně identifikuje. Uložení takovéto informace na fyzickém médiu umožňuje například technologie RFID nebo NFC[3].

[2] Brute-Force útok - Útočník se snaží heslo systematicky uhodnout.
[3] NFC - Near-field communication

.

RFID je společný název pro metody, které využívají radiových vln k automatické identifikaci objektů. K tomuto je využita kombinace RFID nosiče a RFID čtečky. Na RFID nosiči je uložen kód, který tento nosič vysílá do svého okolí. Toto vysílání je schopna zachytit RFID čtečka, která může být připojena k bezpečnostnímu systému. Výhodou RFID je velká rychlost čtení.

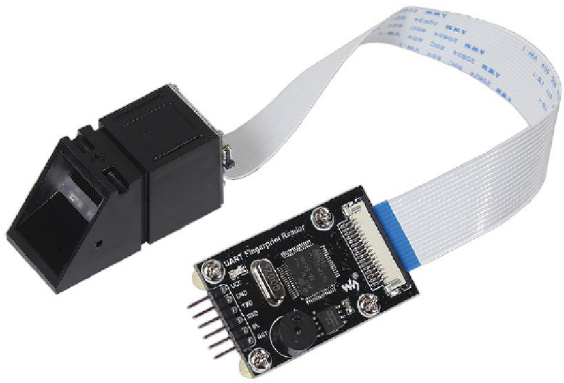
2.1.3 Autentizace na základě fyziologických rysů

Touto skupinou metod se zabývá metoda zvaná biometrie[4]. Unikátních fyziologických rysů, podle kterých lze člověk identifikovat je celé řada - obličej, duhovka oka, otisk prst, hlas, dlaň, podpis a další. Hlavní znaky biometrických metod:

- Člověk není nucen si nic pamatovat nebo něco vlastnit. Autentizace probíhá pouze na základě jeho samotného.
- Biometrické rysy jsou dostatečné unikátní pro každou osobu a je velmi těžké je napodobit.
- Většina biometrických metod autentizace osoby je vhodná pro použití pro velké množství osob (letiště, vězení, firmy).
- Biometrické rysy jsou trvalé, nemohou být ztraceny nebo zapomenuty.

Snímač otisků prstů

Snímač otisků prstů může být pouze samotná čtečka otisku, jejíž výstupem je pouze snímek v nějaké datové podobě nebo již implementovaný modul, který poskytuje výstupní rozhraní. V této bakalářské práci je použit Parallax 29126 Fingerprint Scanner. Tento již implementuje čtení otisků, jejich ukládání a zahrnuje procesor, pomocí kterého identifikuje jednotlivé snímky. Modul komunikuje pomocí rozhraní UART pomocí definovaného rozhraní. Více informací o jednotlivých parametrech modulu, jeho funkcích a komunikačním rozhraní je v dokumentaci, kerá je součástí přiloženého paměťového média.



Obrázek 2.2: Parallax 29126 Fingerprint Scanner [8].

[4]biometrie – obor zabývající se použitím matematické statistiky pro zkoumání proměnlivosti živých organismů

Kapitola 3

Rozpoznávání obličeje

V této kapitole je nejprve obecně rozvedeno, co je to rozpoznávání obličeje, stručný popis historie a jeho využití. Poté je uveden přehled jednotlivých postupů, pomocí kterých může být systém rozpoznávání obličeje implementován a jejich stručný popis.

V úvodu této kapitoly jsem čerpal z publikace "Face Recognition - Semisupervised Classification, Subspace Projection and Evaluation Methods"a to hlavně z úvodní kapitoly [11] a z části "Face Recognition: Issues, Methods and Alternative Applications"[17].V části kapitoly, která se zabývá přehledem metod, jsem čerpal z článku "A Review Of Face Recognition Methods"[12]. V poslední části kapitoly, popisující knihovnu Tensorflow a rozebírající model FaceNet jsem čerpal z oficiálních stránek Tensorflow [1] a článku „FaceNet: A Unified Embedding for Face Recognition and Clustering“ [13].

Rozpoznávání obličeje je způsob identifikace osoby na základě fyziologických rysů jejich tváře. Tato metoda určení identity patří do kategorie biometrických[1] metod společně s dalšími metodami, jako například rozpoznávání hlasu, rozpoznávání sítnice oka nebo rozpoznávání otisků prstů. Mezi hlavní výhody patří například neinvazivnost - nevyžaduje žádnou fyzickou interakci s osobou, která má být identifikována. Další výhodou je skutečnost, že identifikovaný člověk si nepotřebuje pamatovat heslo, či prokazovat se nějakým předmětem (občanský průkaz, ISIC .apod)

Počítače se začaly používat pro rozpoznání obličeje již v roce 1965. V této době byl však proces zdlouhavý a údaje o obličeji se do počítače musely zadávat manuálně. Zlom nastal v roce 1991, kdy Matthew A. Turk a Alex P. Pentland vymysleli způsob, jak detekovat obličej v obrázku, což byl první předpoklad k automatizaci procesu identifikace. Následovala dlouhá léta vývoje, kdy pro rozpoznávání obličeje našly využití hlavně bezpečnostní složky a armády různých států.

V současnosti je rozpoznávání obličeje podstatnou součástí společnosti, se kterou je možné se setkat téměř v každodenním životě. Stále významnější roli hraje při personalizaci reklam, vyhledávání osob, zabezpečení střežených objektů a mnoha dalších. Díky tomu neustále roste poptávka po rozpoznávacích systémech, které nejsou příliš drahé a jejich rozpoznávací algoritmy jsou rychlé a přesné.

Zatímco v kontrolovaném prostředí je úroveň systému rozpoznávání obličeje v současnosti již na dobré úrovni, při využití v praktických aplikacích je třeba se často potýkat s proměnnými podmínkami prostředí, jako například změnou osvětlení, úhlem, pod kterým je obličej snímán, změnou výrazu ve tváři nebo použitými módními doplňky.

[1] biometrie – obor zabývající se použitím matematické statistiky pro zkoumání proměnlivosti živých organismů

V úvodu je také nutno zmínit, že pojem "rozpoznávání obličeje", anglicky „face recognition“ můžeme rozdělit na dvě základní problematiky - verifikace obličeje a identifikace obličeje.

Verifikace obličeje

První z výše zmíněných je verifikace obličeje (anglicky "face verification"). Cílem verifikace je potvrdit identitu obličeje. Při verifikaci je vstupní obraz porovnáván pouze s obrazem, který má danou identitu. Jedná se tedy o porovnávání jedna ku jedné.

Identifikace obličeje

Na druhou stranu identifikace obličeje porovnává vstupní obraz s databází obrazů. Předmětem tohoto porovnávání je zjistit, zda je vstupní obraz pro systém známý. Snaží se tedy zjistit, jestli se identifikovaná osoba nachází v databázi či nikoli.

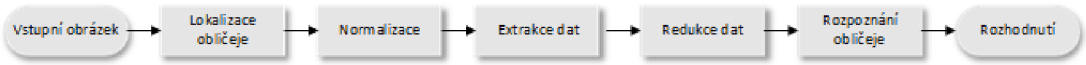
3.1 Obecná metoda rozpoznávání obličeje

Obecná metoda rozpoznávání obličeje může být chápána jako proces, který je rozdělen na několik částí.

První částí je lokalizace, neboli určení oblasti, ve které se obličej na obrázku nachází. Již tento první krok, může mít uplatnění - sledování obličeje, odhad pózy, ořezání obrázku. Následuje normalizace lokalizovaného obličeje tak, aby získaná data nebyla závislá na měřítku a náklonu obličeje. Po normalizaci se provede extrakce rysů obličeje. Tato část procesu musí být efektivní zejména z ohledu výpočetního času a využití paměti. Proto během ní přijde k redukci dimenzionality dat - zmenšení počtu proměnných, které je potřeba zpracovat. Podle typu metody je z množiny extrahovaných rysů vybrána vhodná podmnožina. Teprve nad takto připravenými daty je proveden rozpoznávací algoritmus.

Podle způsobu, jakými algoritmy data vyhodnocují, je můžeme rozdělit na template-based a geometry-based.

Template-based algoritmy porovnávají vstupní data s nějakou kolekcí dat za pomocí nástrojů, jako například Support Vector Machine (SVM) nebo Linear Discriminant Analysis (LDA). Geometry-based algoritmy vyhodnocují vstupní data na základě na sobě navzájem nezávislých rysů, jako jsou oči, nos, či ústa. Každý tento rys je vyhodnocován izolovaně a vztahy mezi nimi nejsou brány v úvahu. Nástroj, který se používá při tomto přístupu, je například Hidden Markov Model (HMM).



Obrázek 3.1: Schéma obecné metody rozpoznávání obličeje.

3.2 Metody rozpoznávání obličeje

Metod na rozpoznávání obličeje existuje v dnešní době velké množství a lze je dělit do skupin podle mnoha faktorů. I tyto skupiny se však mohou částečně překrývat. Proto neexistuje žádné oficiální obecné rozdělení. Cílem této podkapitoly není uvést kompletní výčet všech

možností a klasifikovat je do skupin, ale spíše přiblížit tuto problematiku natolik, aby bylo možné pochopit způsob zpracování problému rozpoznávání obličeje ze statického obrazu v rámci této práce.

3.2.1 Appearance-based metody

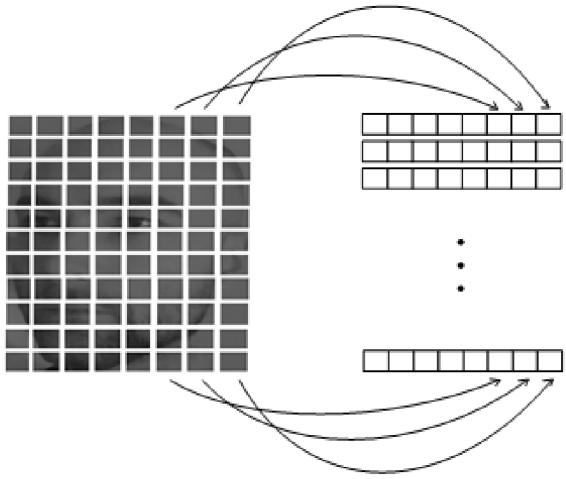
Princip, na kterém jsou tyto metody založeny, je nejvíce podobný principu, jakým obličeje rozeznává člověk. Pro tyto metody je hlavním klasifikátorem, na kterém algoritmy stojí, hodnota intenzity jednotlivých pixelů. Tyto metody se dále dělí na holistické[2] a hybridní.

Holistické metody, vstupní data pro holistickou metodu je výseč obrázku, který obsahuje celou tvář. Pro rozpoznávání tváří používá statistických podobností rozložení rozdílů hodnot intenzity pixelů..

Hybridní metody jsou přirozenější a vnímají obraz podobně jako lidé - jako celek a zároveň jako jednotlivé rysy.

Holistické metody

Holistické metody se používají k rozpoznávání údaje z obličeje jako z celku. Z obličeje jako celku je vytažen relativně malý počet rysů, které jsou odvozeny z jednotlivých pixelů obrázku. Tato množina dat zachycuje rozdíly mezi jednotlivými tvářemi. Konkrétními holistickými metodami jsou například Eigenfaces a Fisherfaces. Tyto metody dokáží efektivně pracovat s velkými databázemi. Dalšími populárními metodami jsou Principal Component Analysis (PCA) a Linear Discriminant Analysis (LDA). V metodě PCA jsou obličeje reprezentovány pomocí lineární kombinace vážených vlastních vektorů (vlastní vektor se anglicky nazývá Eigenvector). Set těchto vektorů se nazývá Eigenface.



Obrázek 3.2: Obličej reprezentovaný malým množstvím rysů.

Metoda LDA se zaměřuje na rysy, které se jsou mezi tvářemi lidí jiné, ale zároveň shodné v obrázcích jedné tváře. Tato metoda dosahuje dobrých výsledků při naplnění dvou předpokladů. Prvním předpokladem je, že globální struktura dat extrahovaných z obličeje

[2]Holismus - směr (zejména biologický) zdůrazňující celistvost a pokládající celek za něco vyššího než souhrn součástí

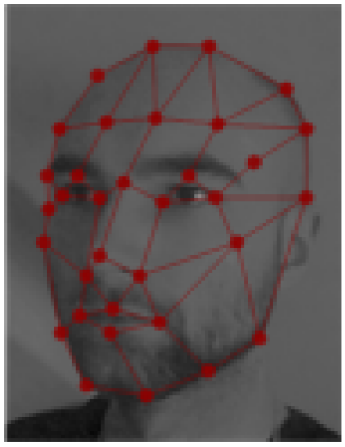
je shodná s lokální strukturou dat extrahovaných z obličeje. Druhým předpokladem je, že vstupní data[3] jsou normálním rozdělením. Nicméně v reálným aplikacích se často stává, že tyto předpoklady nejsou naplněny.

Hybridní metody

Hybridní metody využívají jak holistické, tak lokální rysy. Princip Eigenfaces je uplatněn na jednotlivé rysy, jako oči (eigen eyes), ústa (eigen mouth) nebo nos (eigen nose). Toto je umožněno skutečností, že většina metod, které se používají pro extrakci rysů, nemají žádné speciální požadavky na vstupní data. Kombinací použitých globálních a lokálních rysů je dosaženo získání dat, která lépe reprezentují unikátnost jednotlivých tváři. Výhodou oproti holistickým metodám je skutečnost, že díky částečnému lokálnímu přístupu je větší pravděpodobnost zachytit právě ty rysy tváře, které jsou nejvíce výrazné a unikátní.

3.2.2 Feature-based metody

Tyto metody využívají apriori informace nebo lokální rysy tváří k tomu, aby vytvořily množinu dat, která jednoznačně identifikuje daný obličej. Vztahy mezi prvky v této množině jsou reprezentovány pomocí grafů. Struktura grafu získaného ze vstupního obrázku je porovnána s databází. Na základě tohoto porovnání je rozlišeno, zda je obličej obsažen v databázi, či nikoli. Jedním z takovýchto algoritmů je například Elastic Bunch Graph Matching, který označuje jednotlivé rysy uzly. Tyto uzly jsou propojeny tak, aby tvořily graf, který reprezentuje tvar tváře. Obrázek 3.3 zobrazuje příklad topologické mapy tváře.



Obrázek 3.3: Topologický graf obličeje.

Experimentálně bylo zjištěno, že nejvíce významných rysů je ve tváři situováno kolem kolem očí a na čele, zatímco méně významné rysy jsou ústa, nos a tváře. Jedním z důležitých konceptů, který je v této metodě využit, jsou Gáborovy filtry[4]. Pokud je totiž filtrovaný obrázek vynásoben 2D gausovým rozložením, získáme vážený obrázek, ve kterém lokální maxima reprezentují právě důležité rysy, které jsou v centrální oblasti obličeje.

[3] data - soubory hodnot získaných z obrázku obličeje

[4] Gáborovy filtry - pásmové propusti s nastavitelnou středovou frekvencí, orientací a šířkou pásma

3.2.3 Soft Computing-Based

Metody využívající soft computing jsou skupinou metod pro analýzu nejenom obličeje, ale obecně vstupního obrazu. Tyto metody zahrnují použití umělých neuronových sítí (ANN)[5], fuzzy logiky a genetických algoritmů (GA)[6].

ANN metody

Umělé neuronové sítě, někdy také jednoduše nazývané neuronové sítě, jsou jedny z nejúspěšnějších rozhodovacích systémů. Tyto sítě mohou být natrénovány (naučeny) řešit komplexní problémy. Tohoto se dá velice dobře využít v problematice, jakou je rozpoznávání obličeje. Konkrétně se neuronové sítě využívají například k získávání relevantních informací ze vstupního obrázku obličeje nebo finální klasifikaci. Neuronové sítě se dělí na mnoho skupin. Mezi ty, které se používají v oblasti rozpoznávání obličeje patří například:

- FFNN - Feed-Forward Neural Network.
- Hybridní neuronové sítě.
- Hierarchické neuronové sítě.
- Konvoluční neuronové sítě.

3.2.4 Metody založené na fuzzy logice

Algoritmy jako PCA nebo LDA mapují data lineárně z vícerozměrného obrázku, který je z principu nelineární. Toto způsobuje, že lineární metody nedosahují příliš dobrých výsledků, co se týče přesnosti při rozpoznávání obličeje. Řešením výše uvedeného problému je nový přístup k extrakci dat z obrázku založený na fuzzy logice.

3.2.5 GA metody

GA metody jsou metody, které využívají genetického algoritmu. GA je stochaistický vyhledávací a optimalizační algoritmus, který je založený na teorii evoluce a přirozeného výběru. Tento algoritmus nazývá každé řešení chromozomem a využívá vyhledávacího algoritmu postaveného na principech, jako dědičnost, mutace, přirozený výběr nebo křížení. V podstatě se jedná o heuristickou metodu, která se řídí myšlenkou silnější přežije.

Algoritmus začíná s počáteční náhodnou množinou, zvanou populace. Každý prvek této množiny, zvaný chromozom, reprezentuje jedno možné řešení problému. Jednotlivým chromozomům jsou přiřazeny hodnoty, které vyjadřují, jak dobrý je chromozom kandidátem na to, aby se stal řešením. Následně je provedena evoluce pomocí operací, které vytvoří novou populaci chromozomů. Tento proces je opakován, dokud není získáno téměř optimální řešení nebo dokud není dosaženo nějakého iteračního limitu.

3.3 Tensorflow

Tensorflow je open-source knihovna, která se zaměřuje na machine-learning. Tato knihovna má velkou škálu využití. Jedním z jejich hlavních zaměření je trénování neuronových sítí.

[5] ANN - Artificial Neural Network
[6] GA - Genetic Algorithm

Pomocí této knihovny mohou být implementovány modely fungující na bázi neuronových sítí, které jsou schopny vybrat vhodné vlastnosti obličeje a vytvořit z něj soubor dat, který jej reprezentuje. Na základě tohoto souboru lze poté provést klasifikace obličeje. V tensorflow je již vytvořena řada modelů. Tyto modely jsou již častokrát před trénovány na obrovských kvantech dat tak, aby dosahovaly co nejlepších výsledků. Velké množství těchto modelů je volně dostupných v Github repozitáři tensorflow/models[7].

FaceNet

Jedním z modelů implementovaných v TensorFlow je FaceNet. Tento model byl vyvinut v roce 2015 společností Google. FaceNet vytvoří z obrázku tváře vektor, zvaný embedding, který reprezentuje jeho unikátnost. Tyto vektory mohou být poté porovnávány například pomocí SVM. Jedním ze znaků modelu FaceNet je, že se dokáže relativně dobře přizpůsobit faktorům, jako jsou světlo a úhel, pod kterým je tvář snímána.

[7] https://github.com/tensorflow/models

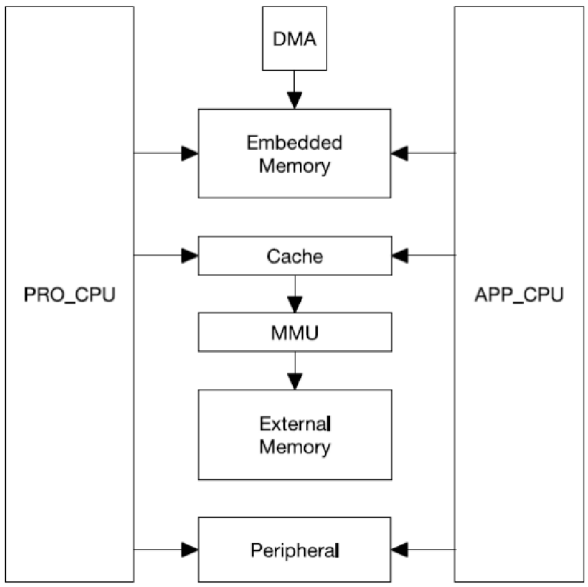
Kapitola 4

Platforma ESP32

V úvodu této kapitoly jsem čerpal z článku „Comparative Analysis and Practical Implementation of the ESP32 Microcontroller Module for the Internet of Things“ [9]. V podkapitolách o použitých typech mikrokontrolérů jsem čerpal z jejich specifikací, [14] a [9]

IoT aplikace jsou v posledních letech stále více na vzestupu a s potřebou vzdáleného ovládání nejrůznějších druhů zařízení, ukládání dat do cloudových úložišť, jejich zpracování, analýzy dat je vyžadována neustálá inovace v oblasti mikrokontrolérů, které tyto služby mohou nabídnout. Základním požadavkem na tyto jednotky je komunikační rozhraní, nejčastěji bezdrátové rozhraní, které umožňuje efektivní připojení do sítě, čímž dokáže zajistit spolehlivý přenos dat. Tyto pokročilé komunikační schopnosti výrazným způsobem ovlivnily oblasti průmyslové automatizace, řízení procesů, výroby, ale také logistiku a inteligentní dopravu. Nemalou měrou se podílí také v oblasti domácí automatizace, chytrých spotřebičů a také např. v oblasti lékařství. Obecně při tvorbě aplikací je brán zřetel na rozměry těchto zařízení, jejich hmotnost, výkonnost, nízká cena a nízká spotřeba elektrické energie, přičemž platí, že čím menší je velikost a hmotnost zařízení, tím širší je oblast jeho aplikací. Na trhu je v dnešní době mnoho zařízení typu IoT. Mohou to být různé Xbee moduly, Arduino desky s komunikačním rozhraním, WhizFi moduly a další. Tato zařízení jsou poměrně drahá a relativně velká z pohledu hmotnosti a zástavbových rozměrů. Společnost Espressif vydala v září roku 2016 nové zařízení s názvem ESP32, které výše uvedené nevýhody řeší a je to nástupce předchozích mikrokontrolérů řady ESP8266. Ty mikrokontroléry byly velmi oblíbené z pohledu jednoduchosti programování, snadného použití ve výkonově nenáročných aplikacích. Platforma ESP32 nabízí už vylepšené řešení, které je možné implementovat do složitějších projektů. Je přímo navržená pro IoT aplikace (Internet od Things) a projekty, které jsou předurčené pro embedded systémy[1]. Jedná se o nízkonákladové mikrokontroléry s nízkou spotřebou elektrické energie, disponující rozhraním Wi-Fi, Bluetooth a výkonným dvoujádrovým mikroprocesorem Tensilica Xtensa LX6 s harvardskou architekturou. Základní blokovou strukturu mikrokontroléru je možné vidět na obrázku 4.1.

[1]Embedded system - Kombinace procesoru, výpočetní paměti a periferních zařízení

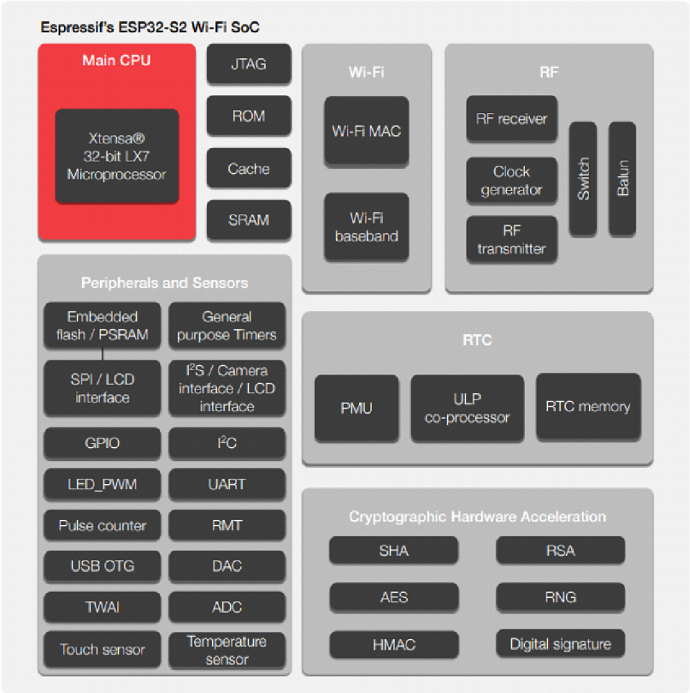


Obrázek 4.1: Blokové schéma ESP32-S2 [9].

Embedded paměť, externí paměť a periferie jsou umístěny na datové sběrnici a/nebo na instrukční sběrnici jednotlivých jader mikroprocesoru. Mikrokontrolér má jedno jádro označeno jako PRO_CPU pro protokoly a druhé jádro APP_CPU pro aplikace. Integrované paměti jsou o velikosti 448 kB ROM, 520 kB SRAM a dále jsou zde dvě paměti RTC o velikosti 8 kB. Externí paměť pak podporuje až čtyři krát 16 MB flash paměti. ESP32 využívá buď vnitřní interní smyčku PLL (Phase Lock Loop) na frekvenci 300 MHz nebo je možné využít externí krystal. Pro obě jádra je možné využít obvod oscilátoru, jako zdroj hodin o frekvenci 2 až 40 MHz s označením CPU_CLK pro obě jádra mikroprocesoru. Hodiny mohou dosahovat výkonu až 160 MHz, popř. při nenáročných aplikacích také nižší, čímž se sníží spotřeba elektrické energie mikroprocesoru. Všechny další hodinové signály pro periferie vychází z frekvence hlavních hodin. ESP32 nabízí ještě další hodinové signály s nízkou spotřebou, jako např. interní RTC_CLK o výchozí frekvenci 150 kHz s možností využít režim s extrémně nízkou spotřebou, tzv. deep sleep režim. K dispozici jsou také čtyři 64bitové obecné časovače, využívající hodiny o frekvenci 80 MHz, dále časovače pro řízení PWM řadiče, kde je k dispozici 8 vysokorychlostních a 8 nízkorychlostních kanálů PWM.

4.1 ESP32-S2-saola

ESP32-S2-Saola je malý typ mikrokontroléru od firmy Espressif, který obsahuje bezdrátový modul s čipem ESP32-S2. Jedná se o výkonný jedno-jádrový 32 bitový mikroprocesor Xtensa LX7 s taktovací frekvencí 240 MHz. Samotný mikroprocesor obsahuje 128 kB ROM paměť, 320 kB SRAM paměť a nabízející množství periferií, které může uživatel využít při tvorbě své aplikace. Vývojovou desku lze rozdělit do několika částí a to části tvořenou mikroprocesorem, částí periferií a snímačů, RTC, hardwarovými akcelerátory a rozhraním Wi-Fi a částí RF, viz obrázek 4.2.



Obrázek 4.2: Blokové schéma ESP32-S2 [14].

Připojení vývojové desky k PC je pomocí USB rozhraní, které je rovněž využíváno k napájení celého systému a periferií. Fyzicky je připojení realizováno USB Micro konektorem se sériovým UART převodníkem. Deska je kompatibilní s mnoha programovacími prostředími, jako je Arduino IDE, ESP-IDF, Node MCU apod., která budou popsána v následujících kapitolách.

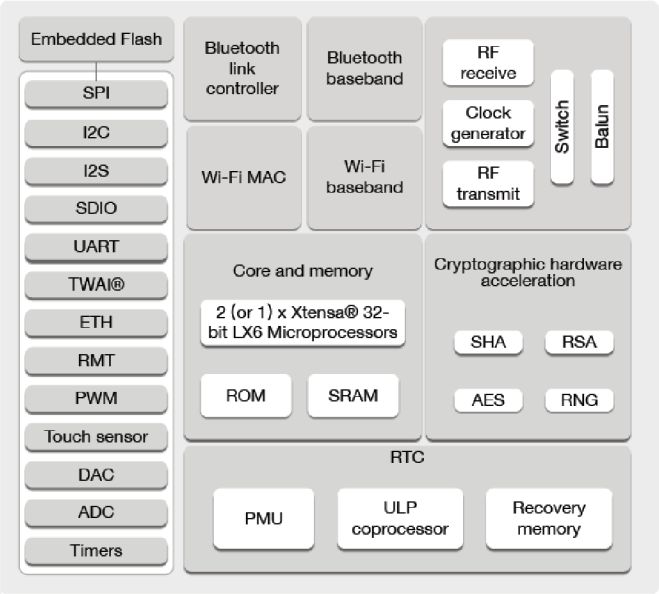
Wi-Fi bezdrátové rozhraní podporuje standardy IEEE 802.11 b/g/n, které poskytuje přenosovou rychlost až 150 Mbps, což je pro vytvářenou aplikaci v praktické části této práce zcela dostačující. V tabulce 4.3 je přehled všech dostupných periferií ESP32-S2.

MCU	- ESP-32-S2 embedded, Xtensa single-core 32-bit LX7 microprocessor - 128 KB ROM - 320 KB SRAM - 16 KB SRAM in RTC
Hardware	- Interfaces: GPIO, SPI, LCD, UART, I2C, I2S, Camera interface, IR pulse counter, LED PWM, USB, 1.1 OTG, ADC, DAC, touch sensor, temperature sensor - Dimensions (18 x 31 x 3.3) mm - Operating Voltage/Power supply 3.0-3.6V - 40 MHz crystal oscilator - 2 MB PSRAM - 4 MB SPI Flash - Operating temperature range -40-85C
WIFI	- 802.11 b/g/n - Bit rate: 802.11n up to 150Mbps - A-MPDU and A-MSDU aggregation - Center frequency range of operating channel 2412-2484Mhz

Obrázek 4.3: Přehled hardwarových vlastností mikrokontroléru ESP32-S2.

4.2 ESP-EYE

ESP-EYE je vývojová deska určená pro zpracování obrazu a zvuku, která je vhodná pro tvorbu IoT aplikací. Tato deska je osazena ESP32 čipem, 2-Megapixelovou kamerou a mikrofonem. Dále disponuje 8 MB PSRAM a 4 MB flash pamětí. Umožňuje posílání obrazu pomocí Wi-Fi a flashování pomocí Micro-USB portu. Blokové schéma je na obrázku 4.4.



Obrázek 4.4: Blokové schéma ESP32 [9].

4.3 Vývojová prostředí

Asi nejznámějším prostředím pro vývoj aplikací v prostředí embedded systémů je Arduino IDE, které je sice primárně zaměřeno na vývoj zařízení kompatibilních s arduinem. S rozšířeními a knihovnami třetích stran je možné využít Arduino IDE i pro vývoj mikrontroléru založeném na platformě ESP32. Alternativou k Arduino IDE je například Platform IO, které oproti Arduino IDE nabízí větší řadu nástrojů a možností pro pohodlnější a produktivnější vývoj aplikací. Výhodou této platformy je možnost integrace do Visual Studio Code, které poskytne programátorovi ještě větší komfort.

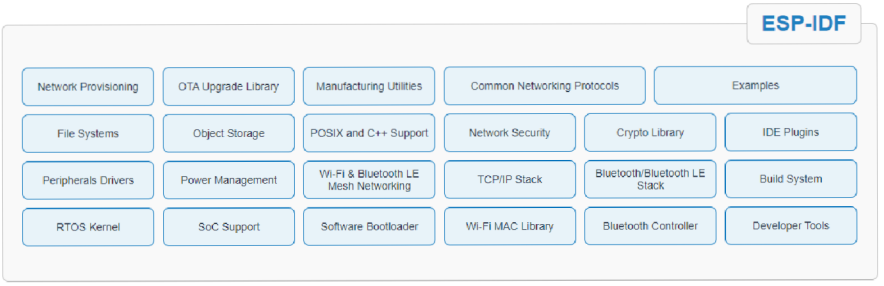
Pro vývoj embedded části inteligentního přístupového systému jsem se rozhodl použít prostředí Espressif IoT Development Framework (ESP-IDF) a to z několika důvodů. Hlavní výhodou ESP-IDF je, že obsahuje velké množství knihoven přímo určených pro mikrokontroléry ESP32, tímto lze předejít využívání knihoven třetích stran a s tím spjatých problémů s kompatibilitou. Prostředí ESP-IDF je podrobněji popsáno v následující podkapitole.

Espressif IoT Development Framework

ESP-IDF je vývojové prostředí od společnosti Espressif Systems, volně dostupné na Githubu, určené pro vývoj projektů založených na platformách ESP32, ESP32-s a ESP32-c. Toto prostředí obsahuje nástroje CMake a Ninja build tools pro překlad a spuštění kódu napsaném v jazyce C/C++.

ESP-IDF také poskytuje knihovny, pomocí kterých se dají využívat jednotlivé komponenty vývojových desek. Tyto knihovny jsou děleny do komponent, které se dají do projektů zahrnout. Využití těchto knihoven je demonstrováno přehlednou dokumentací a demonstračními projekty, které jsou součástí ESP-IDF. Přehled komponent, poskytovaných prostředím ESP-IDF je na obrázku 4.5. Je nutno podotknout, že toto prostředí nemá žádné GUI a proto je ho třeba při vývoji zkombinovat s textovým editorem jako je například VS Code. VS Code také podporuje ESP IDF plugin, který umožňuje překládat projekty přímo ve VS Code.

Jedním důležitým nástrojem, který je později využit při impelentaci je Project Configuration Menu. Jedná se o nástroj ESP-IDF prostředí, který programátorovi umožní předat zařízení nastavení, které se uloží v sdkconfig složce v root adresáři projektu. Na základě sdkconfigu se při překladu ke kódu vygeneruje zdrojový soubor sdkconfig.h, který umožní přistupovat k parametrům nastavených pomocí Project Configuration Menu.



Obrázek 4.5: Přehled komponent poskytovaných prostředím ESP IDF [16].

Kapitola 5

Návrh systému

Tato kapitola detailně popisuje průběh návrhu a systému jako celku a také jeho jednotlivých částí. Dále se také zabývá využitými technologiemi.

V podkapitole zabývající se MQTT[1] jsem čerpal z dokumentace MQTT [10], v podkapitole o technologii Docker z dokumentace dostupné oficiálních webových stránkách [4].

5.1 Základní koncept

Z podstaty toho, že systém má sloužit jako bezpečnostní přístupový terminál, jeho primární funkcí je zajistit bezpečnost vstupu do střeženého objektu identifikací osoby, která do něj vstupuje. Tohoto lze docílit pomocí celé řady identifikačních metod - pin, token, čip, karta nebo biometrie. Právě poslední z dříve uvedených možností je použita jako hlavní identifikační metoda, která jednoznačně a spolehlivě prokáže, zdali identifikovaná osoba je oprávněná vstoupit, či nikoli. Konkrétně je použit snímač otisků prstů. Oproti ostatním způsobům identifikace má biometrie jednu hlavní výhodu - člověk, který se chce prokázat, si nemusí pamatovat žádné heslo, PIN, nemusí u sebe neustále nosit klíče, kartu nebo čip. Právě tato "pohodlnost", která však zároveň žádným způsobem neohrozí bezpečnost objektu, byla jedním z hlavních důvodů, proč jsem se rozhodl právě pro tento způsob zabezpečení. Tato funkce primárního zabezpečení je zahrnuta v modulu terminálu.

Další důležitou částí je vedení a ukládání záznamů z činnosti terminálu pro případnou zpětnou analýzu potenciálních problémů, jak technického, tak bezpečnostního rázu, které by mohly nastat. Vzhledem k tomu, že uschovávat větší objem dat přímo v modulu terminálu je do značné míry nepraktické (byl by třeba dodatečný hardware přímo v modulu terminálu), nabízí se možnost uchovávat tato data na lokálním serveru, ke kterému by byl terminál připojen. Tento server by byl fyzicky umístěn v objektu, který by terminál střežil. Další možností je uchovávat data terminálu na cloudovém[2] serveru. Obě tyto možnosti mají svoje výhody a nevýhody. Vzhledem k tomu, že je tento projekt koncipován jako součást většího projektu a k vývoji všech částí je potřeba společný přístup. Bylo rozhodnuto pro vytvoření serveru v cloudové podobě.

Nicméně při implementaci byla snaha o to, aby systém nebyl závislý na zvolené cloudové platformě a bylo jej možné spustit i na jiných zařízeních s žádnými nebo s minimálními úpra-

[1] MQTT - Message Queuing Telemetry Transport

[2] Cloud - rozsáhlá síť vzájemně propojených vzdálených serverů po celém světě, které fungují jako jeden ekosystém

vami. Proto je při vývoji serverové aplikace vyhodnocující data z kamer použita technologie Docker.

Systém také poskytuje sekundární způsob zabezpečení - monitoring vyhrazeného vstupního prostoru pomocí kamerových modulů. Tyto kamerové moduly budou v případě pohybu zachyceného PIR čidlem snímat oblast vstupu do objektu a tyto snímky se budou ukládat na serveru. Snímky mohou v případě pokusu o neoprávněné vniknutí do objektu sloužit ke zpětné identifikaci pachatelů nebo jako důkazy o trestném činu. Vzhledem k variabilitě prostředí, kde může být zařízení umístěno, je systém navržen tak, aby k jednomu terminálu bylo možné připojit více kamer. Tímto lze docílit lepšího pokrytí prostoru a zajistit větší pravděpodobnost zachycení tváří osob vstupujících do střeženého objektu.

Poslední funkční částí projektu je aplikace, která bude zpracovávat fotografie zachycené kamerami připojenými k terminálu. Na těchto fotografiích bude prováděna detekce a rozpoznávání obličejů. Tato funkce ve finále umožní kontaktovat odpovědnou osobu o tom, že se ve vstupní oblasti do objektu pohybují neznámí lidé.

Aby bylo možné tento bezpečnostní systém využít v dalších projektech, či ovládat pomocí nějakého GUI, je třeba, aby poskytoval rozhraní, která by k těmto účelům mohla být použita. Také je třeba, aby nejenom informace z jednotlivých senzorů, ale celá výstupní logika byla aspoň částečně zpracována. Z tohoto důvodu je také nutné definovat protokol, pomocí kterého je možné s terminálem komunikovat a také určit v jakém formátu budou data z terminálu uschovávána.

5.2 Komunikace

Jelikož je systém rozdělen na několik částí, jeho nedílnou součástí je komunikace mezi těmito částmi. Aby tyto části mohly spolu komunikovat, je nutné si nejprve ujasnit několik věcí. První věcí je pomocí jakého síťového protokolu bude systém komunikovat. První možností by bylo použít pouze TCP a vytvořit vlastní protokol přímo optimalizovaný potřebám terminálu. Tato úvaha byla zavržena. Tvorba vlastního protokolu by totiž přinesla spoustu práce navíc a také skutečnost, že výsledný protokol by vzhledem k již implementovaným a zaběhnutým možnostem, které se nabízí, nebyl tak kvalitní, jako ostatní. Také by došlo k potenciální komplikaci, pokud by systém měl v budoucnu komunikovat s jinými systémy. Jako komunikační protokol systému jsem zvolil MQTT[3].

5.2.1 MQTT

MQTT je komunikační protokol pro výměnu zpráv mezi zařízeními, ideální pro M2M[4] komunikaci v prostředí s omezenými zdroji doporučen standardem ISO/IEC. Tento protokol je ideální pro vývoj IoT[5] aplikací. Jeho hlavními výhodami jsou lehkost a kompaktnost. Protokol MQTT je orientovaný na rozsáhlé sítě s malým datovým tokem a přizpůsoben tak, aby byl co nejmenší co do datového objemu.

Veškerá komunikace v MQTT musí projít přes MQTT brokera, který funguje jako poštovní kancelář. MQTT broker běží na nějakém fyzickém zařízení nebo v cloudu. Zprávy jsou jednotlivými klienty posílány MQTT brokerovi a následně jsou zařízením přerozdělovány pomoci tzv. „topiců“, což jsou vlastně komunikační kanály identifikované názvem. Po připojení k MQTT brokerovi může klient (zařízení) posílat zprávy na určitý topic. Tyto zprávy

[3]MQTT - Message Queuing Telemetry Transport
[4]M2M - Machine to Machine
[5]IoT - Internet of things

obdrží ostatní klienti, kteří jsou přihlášení k odběru tohoto topicu. MQTT implementuje pouze velmi jednoduché rozhraní:

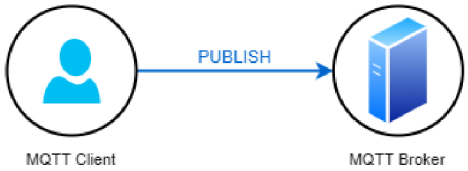
- **Connect** - Připojení klienta k MQTT brokerovi.
- **Subscribe** - Přihlášení klienta k odběru topicu.
- **Unsubscribe** - Odhlášení klienta od odběru topicu.
- **Publish** - Publikace zprávy do topicu.
- **Disconnect** - Odpojení klienta od MQTT brokera.

5.2.2 Formát zpráv

Data, která jsou v systému předmětem komunikace, jsou dvojího typu. Prvním typem jsou příkazy a odpovědi. Pro tento typ dat jsem zvolil formát JSON. Tento formát podporuje přenos všech datových typů, kterých bude v systému potřeba: string, int a array. Druhým typem jsou neformátovaná binární data z kamer.

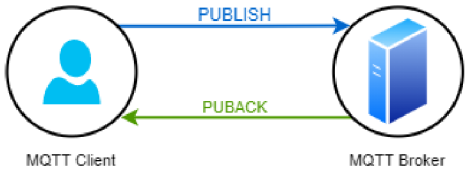
MQTT poskytuje tři úrovně Qos[6], které garantují doručení zprávy:

- **QoS 0** - Garantuje nejefektivnější způsob doručení, ale negarantuje, že zpráva bude doručena.



Obrázek 5.1: Quality of Service úrovně 0.

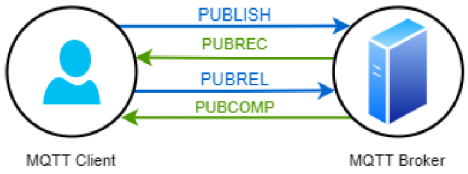
- **QoS 1** - Garantuje, že zpráva je doručena alespoň jednou.



Obrázek 5.2: Quality of Service úrovně 1.

- **QoS 2** - Garantuje, že zpráva je doručena právě jednou, ovšem za cenu rychlosti a efektivity.

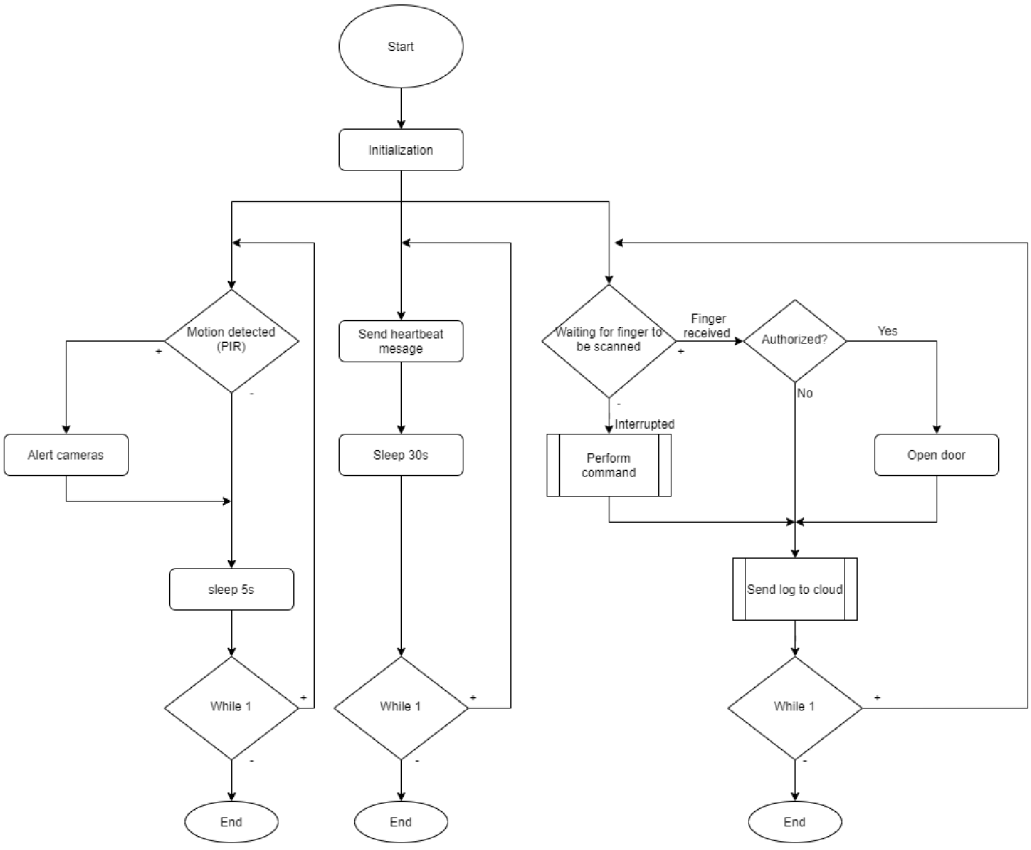
[6]Qos - Qualify of Service



Obrázek 5.3: Quality of Service úrovně 2.

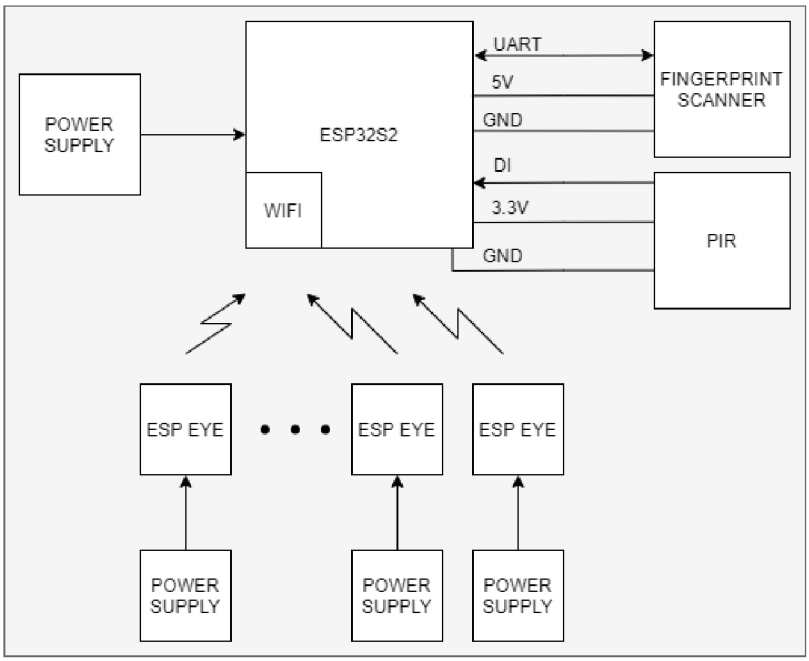
5.3 Embedded část systému

Vstupní terminál jsem rozdělil na dvě zařízení, které spolu komunikují prostřednictvím Wi-Fi. Hlavní část systému je tvořena PIR čidlem a čtečkou otisků prstů, které jsou ovládány pomocí mikrokontroléru ESP32-S2. Tento modul identifikuje osoby, které se snaží vstoupit do střeženého objektu na základě otisku prstu a v případě úspěšné identifikace autorizované osoby vyšle signál pro otevření vstupních dveří. Terminál navíc poskytuje příkazy, pomocí kterých je možné přidávat, mazat a vypisovat otisky prstu, které jsou autorizované. Veškerá aktivita modulu terminálu je zaznamenávána a tyto záznamy jsou odesílány na server. Zařízení umožňuje detekovat pohyb pomocí vhodně umístěného PIR čidla. V případě pohybu jsou upozorněny kamery, které patří k terminálu. Návrh terminálu je zobrazen na obrázku 5.4.



Obrázek 5.4: Schéma hlavní části přístupového systému.

Podpůrnou část systému tvoří kamerové moduly tvořené mikrokontrolérem ESP-EYE, který je osazen kamerou. Tento typ zařízení, v případě obdržení varovné zprávy z hlavní části systému, periodicky snímá vstupní oblast. Následovně jsou tyto snímky odesílány do serverové části pro další zpracování. Blokové schéma embedded části systému je na obrázku 5.5.



Obrázek 5.5: Blokové schéma embedded části systému.

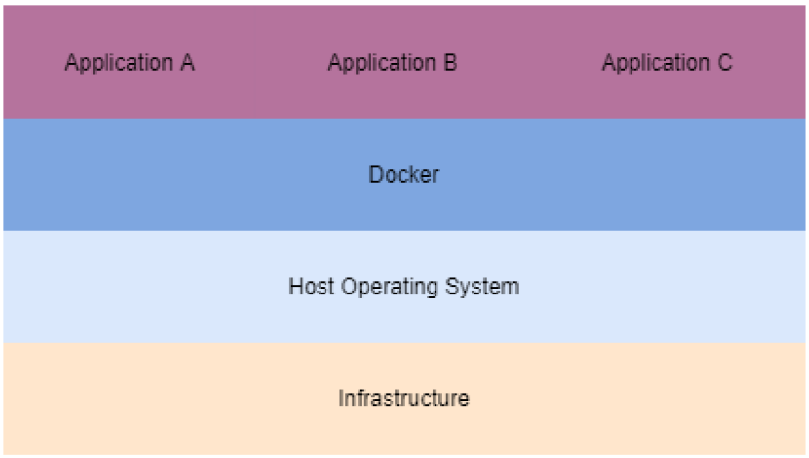
5.4 Serverová část systému

Součástí serverové části systému je MQTT broker, na který se jednotlivé zařízení připojí a aplikace, která má za úkol vyhodnocovat data získaná z kamer. Umístěním této aplikace na server je docíleno přesunutí výpočetně náročných operací mimo embedded prostředí, které má pouze omezené výpočetní možnosti. Zároveň je třeba vytvořit datové úložiště, do kterého se budou ukládat data z jednotlivých kamer a záznamy o činnosti systému. V neposlední řadě je nutné při vývoji dbát na to, aby koncová část systému, na kterou navazuje další projekt, poskytovala vhodné rozhraní pro získávání informací ze systému a zároveň jej umožňovala ovládat.

Aplikaci, která bude rozpoznávat obličeje jsem se rozhodl doplnit technologií Docker, aby bylo možné v případě potřeby jednoduše měnit zařízení, na kterém je spuštěna.

5.5 Docker

Docker je otevřená platforma, která umožňuje oddělit prostředí pro běh aplikace. Toto prostředí zvané kontejner je částečně virtualizované a zapouzdřené. Jeho součástí je pouze aplikace a soubory potřebné pro její běh. Docker kontejner neobsahuje virtualizovaný operační systém, důsledkem je větší flexibilita a nižší náklady na provoz.

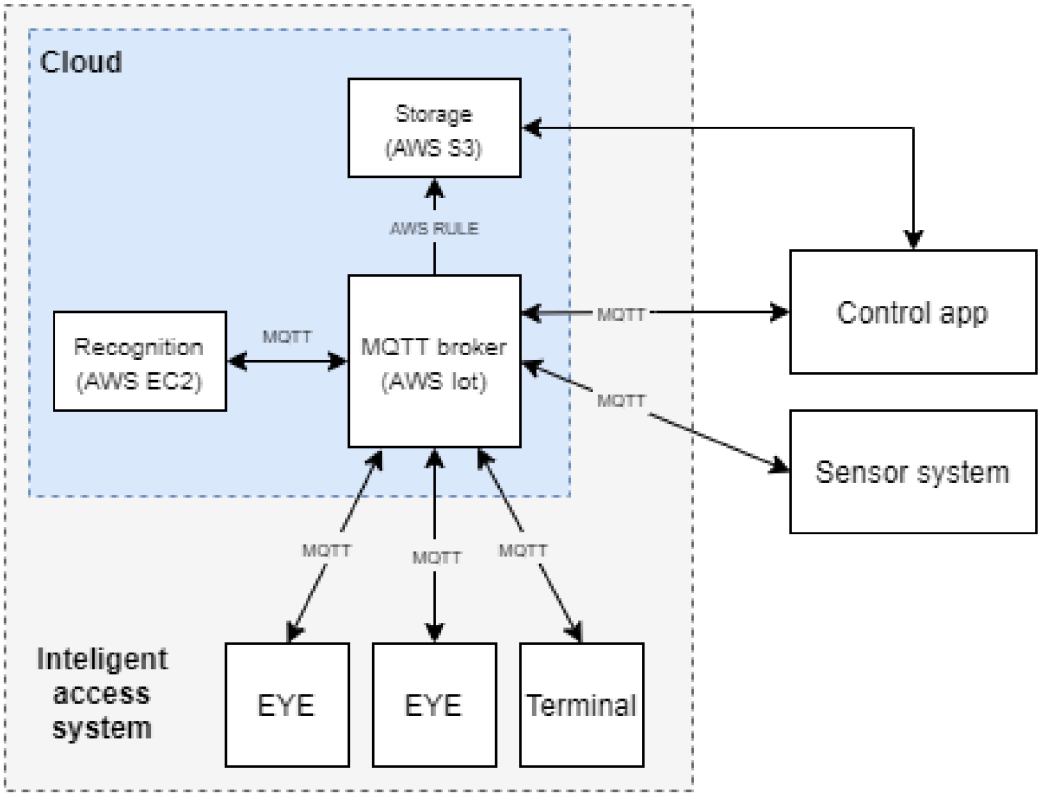


Obrázek 5.6: Schéma zobrazující oddělení aplikace od systému v prostředí Docker.

5.6 Finální podoba systému

Je nutno podotknout, že celkový návrh topologie systému prošel při vývoji jednou hlavní změnou. Původní varianta byla použít komunikační ESP Mesh sítě, což je komunikační protokol postaven nad protokolem Wi-Fi, který by umožnil vytvořit z mikrokontrolérů lokální síť. Jeden z mikrokontrolérů by fungoval jako root, který by sloužil jako přístupový bod pro komunikaci s vnějším prostředím. Tato síťová struktura měla několik výhod, ale zároveň i nevýhod. V průběhu implementace se však vyskytlo několik problémů, které na první pohled nebyly zjevné. Jedním z nich byla například maximální velikost zpráv v rámci mesh sítě, která byla řádově nižší, než je potřeba pro fotografie z kamer. Menší kvalita fotografií za účelem snížení jejich velikosti nepřicházela v úvahu, protože by mohla degradovat úspěšnost rozpoznávání obličeje. Hlavní výhoda mesh sítě spočívala v tom, že komunikace mezi jednotlivými zařízeními by probíhala pouze v lokální síti, tudíž by vznikla jistá míra nezávislosti na internetovém připojení.

Systém je tedy implementován tak, aby komunikace jednotlivých zařízení byla na sobě nezávislá a každé z nich bylo přímo připojeno pomocí MQTT ke cloudové části. Finální návrh celého systému je na obrázku 5.7.



Obrázek 5.7: Celkové schéma systému. Modrou barvou je vyznačena cloudová část, šedou barvou pak systém implementovaný v této bakalářské práci. Control app a Sensor system jsou implementovány jako jiné bakalářské práce.

Kapitola 6

Implementace

V této kapitole je detailně rozebrán implementační proces celého systému a jeho jednotlivých částí, jenž jsou popsány v návrhu.

6.1 Embedded část

Pro vývoj modulů v této části bylo využito vývojové prostředí ESP IDF, které je popsáno v podkapitole 4.3. Veškerá implementace v embedded části je v jazyce C99. Embedded část přístupového systému je rozdělena do dvou částí - modulu terminálu (projekt „terminal“) a modulu kamery (projekt „kamera“).

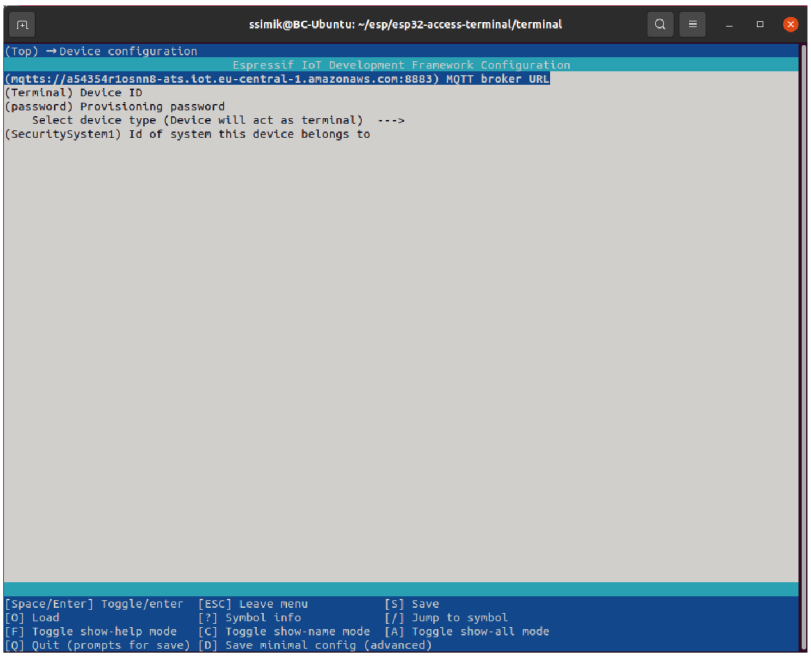
6.1.1 Přednastavení modulů

Tato část je společná pro oba moduly. Jako první je nutno zařízení nastavit parametry, které zařízení identifikují v rámci systému. K tomuto je využito Project Configuration Menu zmíněné v podkapitole 4.3. K předání potřebných parametrů je vytvořen Kconfig.projbuild soubor, pomocí kterého je přídána do Configuration Menu vlastní sekce, pojmenovaná Device configuration. V této sekci lze nastavit parametry potřebné pro správný chod zařízení. Soubor Kconfig.projbuild je shodný pro všechny části systému.

Parametry nastavené pomocí menuconfig:

- **MQTT_URL** - URL MQTT brokera včetně portu.
- **MQTT_ID** - identifikátor zařízení v rámci systému.
- **PROV_PASSWORD** - heslo použité při provisioningu zařízení.
- **SYSTEM_ID** - název bezpečnostního systému.
- **DEVICE_TYPE** - typ zařízení (může být terminál nebo kamera).
- **ROOT_TERMINAL_ID** - název terminálu, ke kterému kamera patří (lze vyplnit pouze u zařízení typu kamera).

Na následujícím obrázku 6.1 je možné vidět Project Configuration Menu, otevřený v systémové konzoli příkazem „idf.py menuconfig“.



Obrázek 6.1: Project Configuration Menu otevřené v sekci Device configuration.

6.1.2 Komunikace

Moduly komunikují pomocí Wi-Fi s MQTT brokerem. Tuto komunikaci implementuje komponenta „communication“, která obsahuje následující soubory:

- **mqtt_handle.c** - implementuje inicializaci MQTT klienta, připojení k MQTT serveru, odebírání topiců a zpracovávání MQTT událostí, mezi které patří i příchozí zprávy.
- **mqtt_handle.h** - obsahuje deklarace proměnných pro názvy kanálů, pomocí kterých zařízení komunikuje, frontu, do které jsou ukládány příchozí příkazy, deklarace funkcí implementovaných v mqtt_handle.c a další definice maker potřebných pro MQTT komunikaci.
- **communication.c** - implementuje provisioning a připojení k Wi-Fi.
- **communication.h** - obsahuje kódy jednotlivých příkazů, pomocí kterých lze zařízení ovládat, deklaraci funkce a proměnných potřebných pro inicializaci Wi-fi.

Aby bylo možné se pomocí modulu připojit k AWS Iot Core, je třeba do složky communication umístit platné certifikáty a privátní klíč s názvy:

- **amazon_cert.pem** - ROOT CA certifikát AWS.
- **cert.pem.crt** - certifikát přidružený k **MQTT_ID**, pod kterým se zařízení připojuje.
- **private.pem.key** - privátní klíč přidružený k **MQTT_ID**, pod kterým se zařízení připojuje.

Tyto soubory jsou poté přidány do Cmakefile komponenty, která se při překladu uloží v .rodata sekci flash paměti. Na tyto autorizační údaje je možné se pak odkazovat ve zdrojovém kódu. Obrázek 6.2 zobrazuje deklaraci proměnných, které obsahují autorizační údaje v souboru mqtt_handle.c.

Protože certifikáty a privátní klíče jsou uchovávány přímo na mikrokontroleru, je vhodné jej zabezpečit, pomocí technologií Secure Boot V2 a Flash Encryption, které systém chrání před pokročilými útoky fyzického charakteru viz dokumentace [15].

```
extern const uint8_t cert_pem_crt_start[] asm("_binary_cert_pem_crt_start");
extern const uint8_t cert_pem_crt_end[] asm("_binary_cert_pem_crt_end");

extern const uint8_t private_pem_key_start[] asm("_binary_private_pem_key_start");
extern const uint8_t private_pem_key_end[] asm("_binary_private_pem_key_end");

extern const uint8_t amazon_cert_pem_start[] asm("_binary_amazon_cert_pem_start");
extern const uint8_t amazon_cert_pem_end[] asm("_binary_amazon_cert_pem_end");
```

Obrázek 6.2: Deklarace proměnných obsahujících certifikáty a privátní klíč.

Kritickou částí kódu je inicializace Wi-Fi modulu. K tomuto je využita knihovna „esp_wifi.h“. Předtím než je možné se připojit k routeru, je nutný tzv. provisioning - získání SSID a hesla routeru, ke kterému se má zařízení připojit. K účelu provisioningu je využita kombinace mobilní aplikace od společnosti Espressif Systems a komponenty „wifi_provisioning“, která je součástí ESP IDF. Aplikace použitá k provisioningu se jmenuje „ESP SoftAP Prov“ a je volně dostupná na „Obchod Play“. Zařízení při inicializaci Wi-Fi nejprve zkontroluje, jestli již nejsou údaje o routeru, ke kterému se má připojit, uloženy v NVS[1]. Pokud tyto údaje nenalezne, zařízení se nastaví do SoftAP módu a čeká, dokud mu tyto údaje nejsou poskytnuty. Při provisioningu jsou SSID a heslo routeru uloženy do NVS paměti. Po získaní těchto údajů a to buď pomocí provisioningu nebo z NVS paměti se zařízení připojí k danému routeru.

Wi-fi provisioning může být vyvolán manuálně stlačením tlačítka RST třikrát za sebou, v intervalu tří vteřin. Toto je impelementováno pomocí proměnné „restart_counter“, která je uložena v NVS. Tato proměnná se po startu zařízení inkrementuje. Po uplynutí pěti vteřin od připojení k Wi-Fi je vynulována. Proměnná je kontrolována při inicializaci Wi-Fi a pokud je větší než tři (zařízení se třikrát po sobě restartovalo v intervalu zhruba tří vteřin), tak se namísto připojení k Wi-Fi pomocí údajů již uložených v NVS, spustí provisioning. Pokud je provisioning úspěšný, údaje k Wi-Fi jsou v NVS nahrazeny novými. V případě neúspěšného provisioningu je možné zařízení restartovat jednou. V tomto případě se pokusí k Wi-Fi připojit pomocí původního SSID a hesla.

Po připojení k Wi-Fi se zařízení připojí k MQTT brokeru a začne přijímat zprávy na jednotlivých kanálech (topics). MQTT Klient je nastaven pomocí parametrů z Project Configuration Menu, zkompilovaných certifikátů a privátního klíče. 6.3.

[1]NVS - non-volatile storage

```
esp_mqtt_client_config_t mqtt_cfg = {
    .uri = CONFIG_MQTT_URL,
    .cert_pem = (const char *)amazon_cert_pem_start,
    .client_id = CONFIG_MQTT_ID,
    .client_cert_pem = (const char *)cert_pem_crt_start,
    .client_key_pem = (const char *)private_pem_key_start,
};
```

Obrázek 6.3: Nastavení MQTT klienta pomocí parametrů z Project Configuration Menu, přidání certifikátů a privátního klíče.

V případě terminálu jsou důležitými částmi kódu obsaženého v této komponentě struktura „Command“ viz. obrázek 6.4 a struktura „Command_Q“ viz obrázek 6.5. Tyto struktury slouží pro zpracování příkazů odeslaných z MQTT brokera.

```
/*Structure used for parsing and processing commands form MQTT messages*/
struct Command {
    int sn;
    int cmd;
    int arg1;
    int arg2;
};
```

Obrázek 6.4: Datová struktura Command, která je použita pro ukládání přijatých příkazů.

```
/*Queue of commands*/
struct Command_Q {
    struct Command q[COMMAND_Q_SIZE];
    int size;
    int first;
} commands;
```

Obrázek 6.5: Datová struktura Command_Q, které je použita pro zpracování přijatých příkazů.

Při inicializaci komponenty „communication“ dochází v neposlední řadě také k vytvoření a uložení topic stringů, které reprezentují názvy kanálů, pomocí kterých zařízení komunikuje. Tyto názvy jsou částečně vytvořeny z parametrů zadaných v Project Configuration Menu viz. obrázek 6.6. Jelikož jednotlivé části sytému komunikují pomocí jiných topiců, je tato tvorba rozdělena podle DEVICE_TYPE.

```
#if CONFIG_TERMINAL
    snprintf(topic_in, BUFFER_SIZE, "/%s/%s/in",CONFIG_SYSTEM_ID ,CONFIG_MQTT_ID);
    snprintf(topic_out, BUFFER_SIZE, "/%s/%s/out",CONFIG_SYSTEM_ID, CONFIG_MQTT_ID);
    snprintf(topic_scan, BUFFER_SIZE, "/%s/%s/scan",CONFIG_SYSTEM_ID, CONFIG_MQTT_ID);
    snprintf(topic_alert, BUFFER_SIZE, "/%s/%s/alert",CONFIG_SYSTEM_ID, CONFIG_MQTT_ID);
    snprintf(topic_heartbeat, BUFFER_SIZE, "/%s/%s/heartbeat",CONFIG_SYSTEM_ID, CONFIG_MQTT_ID);
#elif CONFIG_CAMERA
    snprintf(topic_in, BUFFER_SIZE, "/%s/%s/%s/in", CONFIG_SYSTEM_ID,CONFIG_ROOT_TERMINAL_ID, CONFIG_MQTT_ID);
    snprintf(topic_out, BUFFER_SIZE, "/%s/%s/%s/out", CONFIG_SYSTEM_ID,CONFIG_ROOT_TERMINAL_ID, CONFIG_MQTT_ID);
    snprintf(topic_data, BUFFER_SIZE, "/%s/%s/data",CONFIG_SYSTEM_ID, CONFIG_ROOT_TERMINAL_ID);
    snprintf(topic_alert, BUFFER_SIZE, "/%s/%s/alert", CONFIG_SYSTEM_ID,CONFIG_ROOT_TERMINAL_ID);
    snprintf(topic_heartbeat, BUFFER_SIZE, "/%s/%s/%s/heartbeat", CONFIG_SYSTEM_ID,CONFIG_ROOT_TERMINAL_ID, CONFIG_MQTT_ID);
#endif
```

Obrázek 6.6: Vytvoření jednotlivých topic stringů potřebných pro komunikaci zařízení.

6.1.3 Terminál

Terminál je tvořen mikrokontrolérem ESP32-S2, ke kterému je pomocí GPIO[2] připojeno PIR pohybové čidlo viz. podkapitola 2. Zároveň je k mikrokontroléru pomocí rozhraní UART připojen snímač otisků prstů „29126 Fingerprint Scanner“ viz. podkapitola 2.1.3.

Veškerá komunikace s MQTT brokerem a ostatními zařízeními probíhá na následujících kanálech (topics):

- **/SYSTEM_ID/MQTT_ID/in** - příkazy pro ovládání terminálu.
- **/SYSTEM_ID/MQTT_ID/out** - odpovědi na jednotlivé příkazy.
- **/SYSTEM_ID/MQTT_ID/scan** - záznamy o snímání otisků prstů.
- **/SYSTEM_ID/MQTT_ID/alert** - na tento topic je v případě pohybu detekovaného PIR čidlem poslána zpráva, která upozorní kamery připojené k terminálu.
- **/SYSTEM_ID/MQTT_ID/heartbeat** - na tento topic terminál periodicky posílá zprávy, které lze sledovat a určit podle nich dostupnost zařízení.

Pro komunikaci se snímačem otisků prstů poskytuje firma Parallax Inc. vzorovou knihovnu. Tato knihovna však nebyla kompatibilní s mikrokontrolérem ESP32-S2, proto jsem ji musel vhodně upravit a rozšířit. Výsledkem této činnosti je komponenta fp_scanner, která je rozdělena do souborů:

- **fp_scanner.h** - obsahuje deklarace funkcí pro komunikaci se snímačem otisků prstů.
- **fp_scanner.c** - obsahuje implementaci jednotlivých funkcí pro komunikaci se snímačem otisků prstů.

Komponenta fp_scanner poskytuje rozhraní pro komunikaci se snímačem otisků prstů, který je kompatibilní s mikrokontrolérem ESP32-S2.

Po inicializaci terminálu, která je popsána výše, se program rozvětví do tří úloh. Hlavní úlohou je terminal_task, která se stará o obsluhu snímače otisků prstů. Úloha je koncipována tak, že snímač otisků prstů je primárně v identifikačním režimu, kdy očekává přiložení prstu. Pokud je ke snímači přiložen prst, naskenuje se a pokud je identifikován jako autorizovaná osoba, terminál nastaví výstupní signál na GPIO pinu číslo 20 na hodnotu logické jedničky po dobu pěti vteřin. Tento výstupní pin by poté byl použit k otevření vstupních

[2]GPIO - General-purpose input/output

dveří. Při každém sejmutí otisku prstu je poslán záznam na MQTT brokera, ke kterému je terminál připojen.

Další úloha upozorňuje kamery náležících k terminálu o pohybu ve vstupním prostoru. K tomuto je využito PIR pohybové čidlo viz. 2. Výstup PIR pohybového čidla je připojen na GPIO pin číslo 19. V případě, kdy je hodnota na tomto pinu rovna logické jedničce, znamená to, že PIR čidlo zachytilo pohyb. V tomto případě je odeslána zpráva do „.../alert“ topicu. Na základě této zprávy začnou kamery příslušící terminálu snímat danou oblast.

Poslední úlohou terminálu je periodické posílání zpráv do topicu „.../heartbeat“, pomocí kterých je možné sledovat dostupnost terminálu a případně reagovat na vzniklé problémy.

6.1.4 Kamera

Tato část systému je tvořena mikrokontrolérem ESP-EYE. Komunikace kamery s MQTT brokerem a ostatními částmi systému probíhá na následujících kanálech (topics):

- **/SYSTEM_ID/ROOT_TERMINAL_ID/MQTT_ID/in** - příkazy pro ovládání terminálu.
- **/SYSTEM_ID/ROOT_TERMINAL_ID/MQTT_ID/out** - odpovědi na příkazy.
- **/SYSTEM_ID/ROOT_TERMINAL_ID/alert** - na tento topic je v případě pohybu detekovaného PIR čidlem obdržena zpráva na základě které kamera začne snímat.
- **/SYSTEM_ID/ROOT_TERMINAL_ID/data** - snímky z jednotlivých kamer..
- **/SYSTEM_ID/ROOT_TERMINAL_ID/MQTT_ID/heartbeat** - kamera periodicky posílá zprávy, které lze sledovat a určit podle nich dostupnost zařízení.

Pro ovládání kamery bylo třeba ESP IDF rozšířit komponentou „camera_driver“, pomocí které je kamera inicializována. Tato komponenta využívá ovladače kamery, uložené v adresáři „ESP32-camera“. Ovladače jsou importovány z frameworku ESP WHO, což je nádstavba ESP IDF, která se zabývá právě detekcí a rozpoznáváním obličeje.

Snímky z kamer jsou ve formátu .jpeg s rozlišením 1280 x 720. Průměrná velikost snímku se pohybuje zhruba kolem 45 kB. Snímek je pomocí MQTT odesílán binárně.

Kamera také posílá heartbeat zprávy, obdobně jako terminál.

6.2 Serverová část - Amazon Web Services

Veškeré informace týkající se Amazon services, službách, které poskytuje a jejich využití, jsou čerpány z oficiální dokumentace AWS[3][2].

Jak již bylo zmíněno, projekt je součástí většího systému, jehož jednotlivé části by spolu měly komunikovat. Proto bylo dohodnuto implementovat serverovou část v cloudu tak, aby na tento cloud měli přístup všichni, za účelem usnadnění integrace a testování společné spolupráce jednotlivých částí. Jako cloudová platforma bylo zvoleno AWS.

AWS je dceřinná společnost Amazonu, která nabízí cloudové služby a koncové rozhraní API, pomocí kterých lze k těmto službám přistupovat. Tyto služby samozřejmě nejsou

[3] AWS - Amazon Web Services

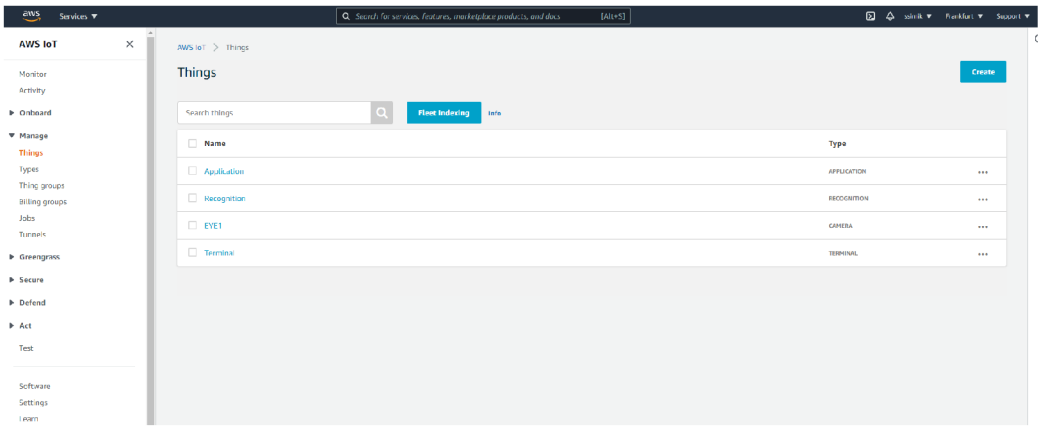
zdarma - většina služeb má definovaný limit (podle různých metrik), při jehož vyčerpání se stane placenou. Za některé služby se musí platit vždy. Pro využívání služeb je třeba vytvořit si vlastní AWS účet. AWS účet musí být spojen s platnou kreditní kartou. Z tohoto důvodu nebude k bakalářské práci přiložen AWS účet, který byl použit při vývoji, jenž je připraven k nasazení systému a na němž jsem systém testoval.

6.2.1 AWS IoT Core

Důležitým prvkem serverové části systému je MQTT broker. Služba IoT Core obsahuje nejen MQTT brokera, ale také slouží jako prostředník pro připojení klientů k ostatním cloudovým službám. Maximální velikost jedné MQTT zprávy je limitována na 128 kB, nicméně největší velikost dat, která je posílána, je snímek z kamery, jehož velikost se pohybuje kolem 50 kB.

k AWS IoT Core se dá připojit pomocí koncového bodu, který je k účtu přidělen při registraci. Konkrétně je využitý port "8883", kde musí být klient autorizován pomocí certifikátu podepsaného privátním klíčem. Komunikace je zabezpečená pomocí TLS v1.2[4].

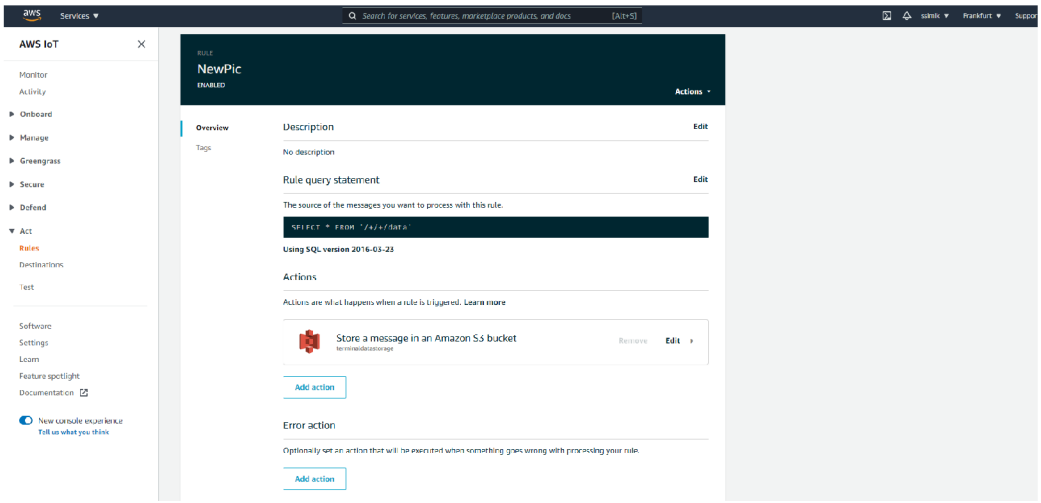
Před Připojením jednotlivých částí k IoT Core je však nejprve nutné jednotlivá zařízení zaregistrovat, vytvořit jim bezpečnostní certifikáty a klíče s potřebnými oprávněními. Tyto kroky jsou detailně popsány v AWS dokumentaci. Na následujícím obrázku je stav stránky po zaregistrování všech potřebných zařízení.[2]



Obrázek 6.7: Seznam zaregistrovaných zařízení na AWS IoT Core.

V IoT Core je možné nastavit pravidla (rules) pro jednotlivá témata (topics). Tyto pravidla se uplatní vždy, když na dané téma přijde nová zpráva. K pravidlu lze přidružit akci, která se provede s obsahem zprávy. Tohoto je využito pro ukládání fotografií z kamer a důležitých zpráv systému do služby AWS S3.

[4]TLS - Transport layer security



Obrázek 6.8: Detail pravidla pro ukládání fotografií z kamer systému.

6.2.2 AWS S3

AWS S3[5] implementuje úložiště dat, které poskytuje API rozhraní pro jednoduchou manipulaci se skladovanými daty. K datům je možné přistupovat buď přímo v prohlížeči nebo pomocí výše zmiňovaného API. V této službě je vytvořeno specifické úložiště (Bucket) - úložiště dat, s názvem „terminaldatastorage“, do kterého jsou data ukládána do následující adresářové struktury:

- **/Securitysystem1/Terminal/data/** - adresář pro fotky z kamer.
- **/Securitysystem1/Terminal/in/** - adresář pro záznam příkazů poslaných terminálu.
- **/Securitysystem1/Terminal/out/** - adresář pro záznam odpovědí na příkazy.
- **/Securitysystem1/Terminal/out/** - adresář pro záznam naskenovaných otisků prstů.

Záznamy jsou zde uloženy v souboru ve formátu JSON a data z kamery ve formátu .jpeg. Při implementaci této vzniklo riziko, že velikost uložených dat by mohla přesáhnout limit AWS S3, po kterém by bylo třeba za službu platit. Tento limit je 5 GB. Proto byl proveden krátký test.

6.3 Serverová část - Aplikace

Aplikace využívá TensorFlow-CPU protože je primárně určena pro nasazení v cloudovém prostředí. Součástí řešení je aplikace, která detekuje a rozpoznává obličeje na snímcích z kamer. Tato aplikace je naprogramována v jazyce Python. Aplikace je rozdělena do dvou zdrojových souborů:

- **prefactorization.py** - Obsahuje funkce pro extrakci všech důležitých informací z obrázků.

[5] S3 - Simple Storage Service

- **recognition.py** - Implementuje dva objekty - "Mqtt_client", který slouží pro komunikaci s MQTT brokerem a zároveň obsahuje instanci objektu "Recognizer", která řeší část týkající se rozpoznávání obličeje.

Pro MQTT komunikaci byla využita knihovna Paho-mqtt. K práci s obrázky pak knihovna Pillow.

Detekce a rozpoznávání obličeje

Aplikace monitoruje dostupnost zařízení systému a provádí identifikaci obličeje. Kontrola dostupnosti zařízení je provedena periodickou kontrolou jejich heartbeat zpráv. Pokud je obdržena první taková zpráva, zařízení je přidáno na seznam monitorovaných.

Při identifikaci obličeje je nejprve třeba vytvořit soubor dat, oproti kterému bude obličej porovnán. Zároveň bude třeba tento soubor dat během chodu aplikace měnit, pokud bude uživatel chtít přidat, či odebrat autorizovanou osobu. V adresáři /photos/authorized jsou jednotlivé adresáře, které obsahují fotografie autorizovaných osob. Jméno osoby, je názvem adresáře.

Úpravu seznamu autorizovaných osob a přehled zařízení lze získat komunikací na následujících kanálech:

- /**SYSTEM_ID**/**MQTT_ID**/add/**jmeno** - vytvoří se nový adresář s názvem **jmeno** a fotka se do něj uloží. Pokud adresář již existuje, fotka se do něj přidá. **SYSTEM_ID** a **MQTT_ID** jsou parametry použité při tvorbě instance objektu „Mqtt_client“ a reprezentují název bezpečnostního systému a název klienta aplikace.
- /**SYSTEM_ID**/**MQTT_ID**/remove/**jmeno** - odstranění autorizované osoby. Pokud existuje, adresář s názvem **jmeno** i se všemi soubory, které obsahuje, je odstraněn (Na obsahu zprávy nezáleží).
- /**SYSTEM_ID**/**MQTT_ID**/getauthorized - pokud je poslána zpráva na tento topic, zařízení odešle seznam autorizovaných osob (Na obsahu zprávy nezáleží).
- /**SYSTEM_ID**/**MQTT_ID**/authorized - topic, kam je poslán seznam autorizovaných osob.
- /**SYSTEM_ID**/getstatus - žádost o přehled zařízení a jejich dostupnost (Na obsahu zprávy nezáleží).
- /**SYSTEM_ID**/status - odpověď na žádost o přehled zařízení.
- /**SYSTEM_ID**/**MQTT_ID**/getauthorized - pokud je poslána zpráva na tento rozpoznávací algoritmus se znovu natrénuje pomocí dat uložených v adresáři „/dataset“ (Na obsahu zprávy nezáleží).

Prvním krokem k úspěšnému rozpoznání obličeje je jeho detekce. K tomuto je využita MTCNN[6], konkrétně implementace od Iván de Paz Centeno, která je dostupná na Githubu v projektu ipazc/mtcnn[7]. Pomocí této knihovny je z každé fotky vyříznuta pouze oblast s obličejem.

[6]MTCNN - Multi-Task Cascaded Convolutional Neural Network
[7]https://github.com/ipazc/mtcnn

Obrázek 6.9: Oříznutí obličeje detekovaného pomocí MTCNN.

Nyní je je potřeba z oříznutého obrázku obličeje dostat údaje, které budou reprezentovat jeho podobu. Tohoto lze docílit využitím FaceNet modelu, který je součásti knihovny Keras (knihovna, která slouží jako rozhraní pro TensorFlow). Tento model je schopen vytvořit z obrázku obličeje vektor, jenž reprezentuje jeho unikátnost. Tento vektor se nazývá „embedding“. FaceNet očekává na vstupu obrázek o rozměrech 160x160 pixelů, proto jsou všechny obrázky ořezaných obličejů transformovány na tyto rozměry. Při vytváření je ke každému vektoru přiřazen název osoby, ke které patří (podle názvu složky, ze které byl obrázek nahrán) a každý vektor je normalizován.

Konečné porovnávání provádí lineární SVM[8], který je natrénován pomocí naší databáze autorizovaných osob. Při obdržení nové fotografie je vytvořen embedding způsobem, který je uveden výše. Tento embedding SVM porovná s databází zpracovaných údajů o autorizovaných obličejích. Výsledkem je jméno osoby, ke které systém fotografii přirovnal a shoda v procentech.

Pokud je shoda menší než stanovený limit je odeslána zpráva na topic „/SYSTEM_ID/warning“. Tato zpráva signalizuje, že systém detekoval neznámou osobu.

Limit pro rozhodování, zda-li je osoba autorizovaná, je stanoven experimentálně na 60 % viz. následující kapitola.

Pokud systém detekuje na snímku autorizovanou osobu a počet fotek této osoby v datasetu je menší než deset, potom se na základě tohoto snímku dotrénuje. Díky tomuto dojde ke zvýšení přesnosti procesu rozpoznávání obličeje.

[8]SVM - Support Vector Machine

Kapitola 7

Realizace

Za účelem testovaní implementovaného systému byly vytvořeny demonstrační moduly kamery a terminálu. Pomocí těchto modelů je v této kapitole demonstrována funkcionalita systému. Tyto moduly jsou zobrazeny na obrázcích 7.1 a 7.2. Z důvodu časté aktualizace kódu na těchto zařízeních a potřeby monitorovat jejich činnost pomocí sériové linky nejsou tyto zařízení zabezpečena pomocí technologií Secure Boot V2 a Flash encryption. Jistým důvodem byly také obavy, že po nesprávném provedení zabezpečení mikrokontrolérů by moduly nebylo možné použít.

Obrázek 7.1: Laboratorní modul kamery.

Obrázek 7.2: Laboratorní modul terminálu.

Pomocí těchto modulů bylo provedeno několik testů, které jsou popsány níže. Většina testů byla spíše praktického charakter. Jelikož se jedná o bezpečnostní systém, důležitá je především stabilita a tolerance chyb. Tyto vlastnosti bylo možné jednoduše otestovat pomocí realizačních modulů. Moduly byly umístěny do prostředí, kde byly po delší doby vystaveny simulovaným podmínkám co nejvíce blížícím se reálnému provozu. Ovšem možnost tohoto testování byla omezena na domácí prostředí kvůli aktuální epidemiologické situaci.

Součástí demonstračního modulu terminálu je LED dioda, která simuluje otevření dveří v případě sejmutí otisku prstu autentizované osoby. Toto je možné pozorovat na obrázku 7.3. Více obrázků z testování systému je obsaženo v přílohách.

Obrázek 7.3: Rozsvícení LED diody umístěné na demonstračním modulu terminálu při sejmutí autentizovaného otisku prstu.

7.1 Nasazení rozpoznávací aplikace do EC2

Původně měla být rozpoznávací aplikace nasazena na virtuálním stroji, spuštěném ve službě EC2. Nabídka virtuálních strojů, které jsou zdarma je však výkonnostně značně omezena. EC2 v podstatě nabízí pouze jednojádrový procesor s frekvencí 2,5 GHz a RAM o kapacitě 1 GiB. Přehled virtuálních strojů je zobrazen na obrázku 7.4.

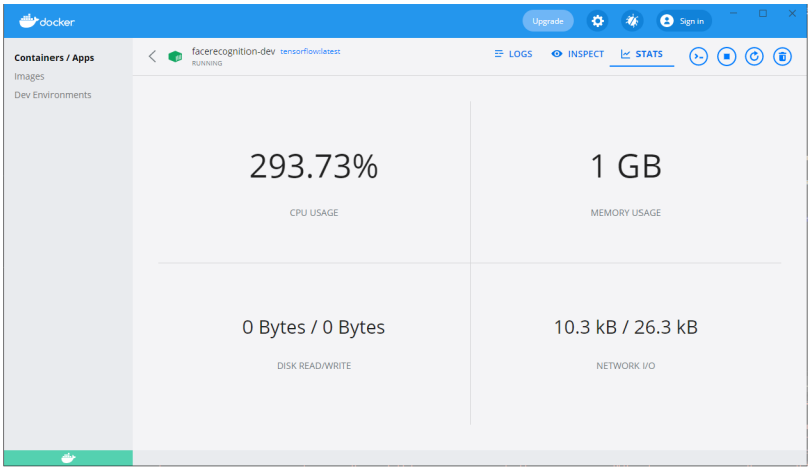
	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
☐	t2	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
☑	t2	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate	Yes
☐	t2	t2.small	1	2	EBS only	-	Low to Moderate	Yes
☐	t2	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
☐	t2	t2.large	2	8	EBS only	-	Low to Moderate	Yes
☐	t2	t2.xlarge	4	16	EBS only	-	Moderate	Yes
☐	t2	t2.2xlarge	8	32	EBS only	-	Moderate	Yes
☐	t3	t3.nano	2	0.5	EBS only	Yes	Up to 5 Gigabit	Yes
☐	t3	t3.micro	2	1	EBS only	Yes	Up to 5 Gigabit	Yes
☐	t3	t3.small	2	2	EBS only	Yes	Up to 5 Gigabit	Yes
☐	t3	t3.medium	2	4	EBS only	Yes	Up to 5 Gigabit	Yes
☐	t3	t3.large	2	8	EBS only	Yes	Up to 5 Gigabit	Yes
☐	t3	t3.xlarge	4	16	EBS only	Yes	Up to 5 Gigabit	Yes
☐	t3	t3.2xlarge	8	32	EBS only	Yes	Up to 5 Gigabit	Yes
☐	t3a	t3a.nano	2	0.5	EBS only	Yes	Up to 5 Gigabit	Yes

Obrázek 7.4: Nabídka virtuálních strojů, které lze spustit v rámci služby EC2.

Po nasazení rozpoznávací aplikace na tento virtuální stroj se ukázalo, že při jejím trénovaní se server zasekne a zhroutí.

Proto byla provedena jednoduchá analýza využití zdrojů při jejím běhu pomocí aplikace Docker desktop. Při této analýze bylo zjištěno, že při trénování dochází k využití RAM paměti, které přesahuje 1 GB. Toto je důvod proč aplikace nelze nasadit na virtuální stroje,

které AWS poskytuje v zdarma. Obrázek 7.5 zachycuje průběh testu v momentě velké spotřeby paměti.



Obrázek 7.5: Využití paměti aplikace přesáhne 1 GB při analýze v aplikaci Docker desktop.

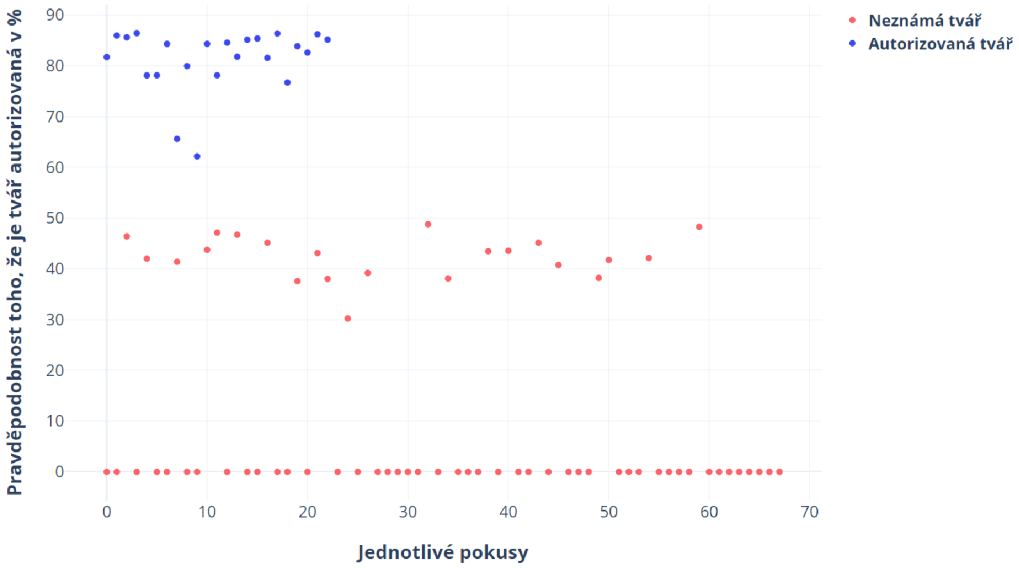
Rozpoznávací aplikaci se z tohoto důvodu nepodařilo zprovoznit v prostředí EC2. Při praktickém použití systému by tedy služba EC2 musela být placena z AWS účtu tak, aby bylo možné spustit výkonnější stroj, nebo by aplikace mohla být umístěna na lokálním serveru, či počítači. Pokud by primárním zájmem bylo udržení minimální ceny systému bylo by možné umístit aplikaci na jinou vhodně zvolený cloudový server, který poskytuje možnost provozování dostatečně výkonného stroje.

Při vývoji a testování přístupového systému byla aplikace spouštěna na „GE62VR 7RF Apache Pro“ (16 GB RAM).

7.2 Testování aplikace pro rozpoznávání obličeje

Účelem tohoto testování je určit práh pravděpodobnosti, na základě kterého je rozhodnuto, zdali rozpoznávaný obličej patří mezi autorizované. K tomuto účelu byla vytvořena testovací sada, která obsahuje 24 fotografií jedné osoby (autorizované) a 66 fotografií různých (neznámých) osob. Pomocí této sady byla aplikace otestována několika testy s různými datasety (dataset obsahuje fotky jednotlivých autorizovaných osob), pomocí kterých byl rozpoznávací algoritmus trénován. Testovací sada, stav datasetu při testu, a výsledky jednotlivých testů, jsou obsaženy v přiloženém datovém médiu v adresáři „recognition_testy“.

Pomocí testování bylo zjištěno, že rozpoznávací algoritmus je velmi nepřesný s malým počtem fotografií v datasetu. Důsledkem je přidání 20 fotografií rozdělených do adresářů „unknown“ a „unknown2“. Pomocí těchto fotografií je dataset uměle rozšířen. Na grafu 7.6 je výsledek testů s takto rozšířeným datasetem a jednou autorizovanou osobou, jejíž adresář obsahuje 10 fotografií.

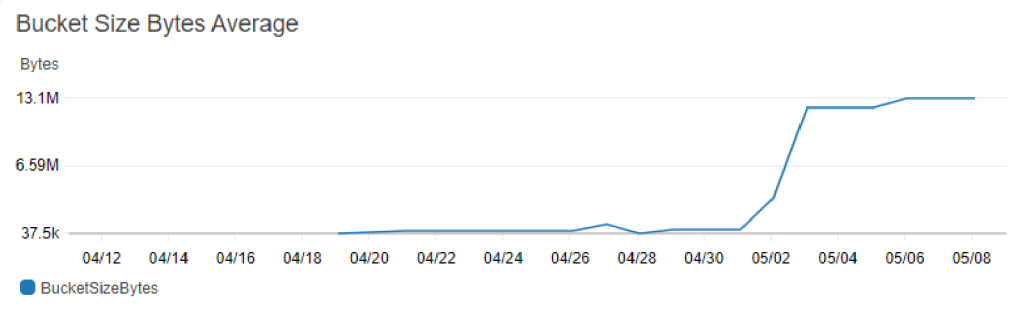


Obrázek 7.6: Graf jednotlivých předpovědí v rámci testovacího setu, který obsahuje 24 fotografií autorizované osoby a 66 fotografií neznámých osob.

Na obrázku je možné vidět že oblast mezi 50-60 % rozděluje odhady autorizovaných a neznámých osob. V této oblasti je zvolen práh 60 %, pomocí něhož je určeno zda je osoba autorizovaná. Na základě těchto testů je do aplikace přidána funkcionalita, která vždy při odhadu nad 80 % přidá fotografii do datasetu autorizované osoby pokud jich dataset obsahuje méně než 10. Tímto je docíleno zvýšení spolehlivosti odhadu u autorizovaných osob, jejichž dataset obsahuje menší počet fotografií.

7.3 Testování velikosti výstupních dat

Tato část testuje skutečnost popsanou v kapitole 6.2.2. Po provedení zátěžového testu, který trval zhruba jeden den, se velikost dat v úložišti zvětšila přibližně o 11,5 MB. Test byl proveden pouze s jednou kamerou připojenou k terminálu. Na základě těchto údajů by měla velikost úložiště systému se třemi kamerami vydržet alespoň 3 měsíce, než by data musela být přesunuta jinam. Test byl uskutečněn pomocí nástroje CloudWatch, který je součástí AWS. Z tohoto nástroje byl také získán graf vývoje velikosti úložiště viz obrázek 7.7.



Obrázek 7.7: Graf vývoje velikosti úložiště.

Kapitola 8

Závěr

Cílem této práce bylo vytvořit přístupový terminál, který bude možné využít jako součást domácího bezpečnostního systému tvořeného přístupovým systémem, senzorovým systémem a řídicí aplikací. Tento záměr byl splněn. V rámci této práce se podařilo implementovat přístupový terminál a na základě této implementace sestrojit demonstrační moduly, pomocí kterých bylo možné otestovat jeho funkcionalitu v simulovaném prostředí a to i navzdory omezeným možnostem při současné epidemiologické situaci (koronavirová epidemie 2020-2021).

Přínos práce vidím zejména v tom, že terminál poskytuje oproti jiným řešením stejné cenové kategorie nadstandardní metody zabezpečení. Další výhodou je možnost využít jej samostatně nebo jako integrovanou součást domácího bezpečnostního systému.

Jistou komplikací, která nastala v průběhu vývoje celého zařízení, byla nemožnost připojení vytvořeného řešení k řídicí aplikaci, která byla předmětem jiné bakalářské práce, jenž byla v průběhu jejího řešení zrušena. Tato skutečnost znemožnila plně využít potenciál získávaných dat.

Z tohoto důvodu bych chtěl na tuto práci navázat tvorbou vlastní řídicí aplikace s jejíž pomocí by bylo možné vytvořit prototyp domácího bezpečnostního systému a nasadit jej do reálného prostředí. Výsledný bezpečnostní systém by mohl být řádově levnější než jiná obdobná řešení, která jsou současně dostupná na trhu.

Práce mi poskytla příležitost rozvinout programovací schopnosti, zvláště v oblasti vestavěných systémů. Dalším osobním přínosem je velké množství technologií - MQTT, TLS, cloud computing, Docker, TensorFlow, se kterými jsem se díky projektu více seznámil, společně s pochopením základních principů, na kterých fungují algoritmy rozpoznávání obličeje.

Jistou zkušeností byl také proces vývoje, při kterém došlo k několika problémům (nevhodnost mesh technologie, nedostatečná RAM virtuálního stroje dostupného na EC2, zrušení vývoje řídicí aplikace), na které bylo třeba reagovat a vývoj jim přizpůsobit.

Pravidelná spolupráce s externím konzultantem z firmy Espressif Systems mi při vývoji poskytla důležitou zpětnou vazbu, která mě nasměrovala k řešení praktických aspektů systému.

Literatura

[1] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z. et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*, 9. listopadu 2015 [cit. 10.5.2021]. Software available from tensorflow.org. Dostupné z: https://www.tensorflow.org/.

[2] Amazon Web Services, I. *AWS Documentation* [online]. 2021 [cit. 10.5.2021]. Dostupné z: https://docs.aws.amazon.com/index.html?nc2=h_ql_doc_do.

[3] Chechi, D., Kundu, T. a Kaur, P. THE RFID TECHNOLOGY AND ITS APPLICATIONS: A REVIEW. *International Journal of Electronics, Communication Instrumentation Engineering Research and Development (IJECIERD)*. Září 2012, sv. 2, s. 109–120, [cit. 10.5.2021].

[4] Docker, I. *Docker documentation* [online]. 2013 [cit. 10.5.2021]. Dostupné z: https://www.espressif.com/en/products/sdks/esp-idf.

[5] Farik, M., Lal, N. a Prasad, S. A Review Of Authentication Methods. *International Journal of Scientific Technology Research*. Listopad 2016, sv. 5, s. 246–249, [cit. 10.5.2021].

[6] Fernández, E., Ballesteros, J., Desouza Doucet, A. a Larrondo Petrie, M. Security Patterns for Physical Access Control Systems. In:. červenec 2007, sv. 4602, s. 259–274 [cit. 10.5.2021]. DOI: 10.1007/978-3-540-73538-0_19. ISBN 978-3-540-73533-5.

[7] Inc., P. *555-28027-PIR-Sensor-Prodcut-Doc-v2.2*. 2.2. Parallax Inc., 2012 [cit. 10.5.2021].

[8] Inc., P. *29126-Fingerprint-Scanner-Product-Guide-v1.0*. 1.0. Parallax Inc., 2017 [cit. 10.5.2021].

[9] Maier, A., Sharp, A. a Vagapov, Y. Comparative Analysis and Practical Implementation of the ESP32 Microcontroller Module for the Internet of Things. Září 2017, [cit. 10.5.2021]. DOI: 10.1109/ITECHA.2017.8101926.

[10] OASIS. *MQTT - The Standard for IoT Messaging* [online]. 2021 [cit. 10.5.2021]. Dostupné z: https://mqtt.org/.

[11] Ramakrishnan, S. Introductory Chapter: Face Recognition - Overview, Dimensionality Reduction, and Evaluation Methods. In: Ramakrishnan, S., ed. *Face Recognition* [online]. Rijeka: IntechOpen, 2016, kap. 1 [cit. 10.5.2021]. DOI: 10.5772/63995. Dostupné z: https://doi.org/10.5772/63995.

[12] Roomi, M. a Beham, D. A Review Of Face Recognition Methods. *International Journal of Pattern Recognition and Artificial Intelligence.* Duben 2013, sv. 27, [cit. 10.5.2021]. DOI: 10.1142/S0218001413560053.

[13] Schroff, F., Kalenichenko, D. a Philbin, J. FaceNet: A unified embedding for face recognition and clustering. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* [online]. IEEE. červen 2015, [cit. 10.5.2021]. DOI: 10.1109/cvpr.2015.7298682. Dostupné z: http://dx.doi.org/10.1109/CVPR.2015.7298682.

[14] Systems, E. *ESP32-S2 Family Datasheet.* 1.1. Espressif Systems, 2020 [cit. 10.5.2021].

[15] Systems, E. *ESP-IDF Programming Guide* [online]. 2021 [cit. 10.5.2021]. Dostupné z: https://docs.espressif.com/projects/esp-idf/en/latest/esp32/index.html.

[16] Systems, E. *IoT Development Framework | Espressif Systems* [online]. 2021 [cit. 10.5.2021]. Dostupné z: https://www.espressif.com/en/products/sdks/esp-idf.

[17] Wójcik, W., Gromaszek, K. a Junisbekov, M. Face Recognition: Issues, Methods and Alternative Applications. In: *Face Recognition - Semisupervised Classification, Subspace Projection and Evaluation Methods* [online]. 2016 [cit. 10.5.2021]. DOI: 10.5772/62950. Dostupné z: https://app.dimensions.ai/details/publication/pub.1042436820andhttps://www.intechopen.com/citation-pdf-url/51031.

Příloha A

Obsah paměťového média

/ **pristupovy_terminal**
/ / **terminal**
/ / **eye**
/ **recognition_aplikace**
/ **recognition_testy**
/ **technicke_dokumentace**
/ **technicka_zprava**
/ **demonstracni_video.mkv**
/ **manual.md**

Příloha B

Analýza ceny řešení

V této příloze lze nalézt ceny jednotlivých položek, ze kterých je přístupový terminál tvořen a srovnání s jinými přístupovými terminály v podobné cenové kategorii. Ceny jsou uvedeny při kurzu 1 USD / 20,22 CZK.

- PIR pohybové čidlo[1] - 313 Kč
- Fingerprint scanner[2] - 835 Kč
- ESP32-S2-Saola[3] - 178 Kč
- ESP-EYE[4] - 418 Kč
- Příslušenství (kabely, krabičky) - do 800 Kč (odhad pro odolný obal terminálu, tak aby srovnání bylo přesnější)
- **Celková cena: 2544 Kč**

Pro srovnání:

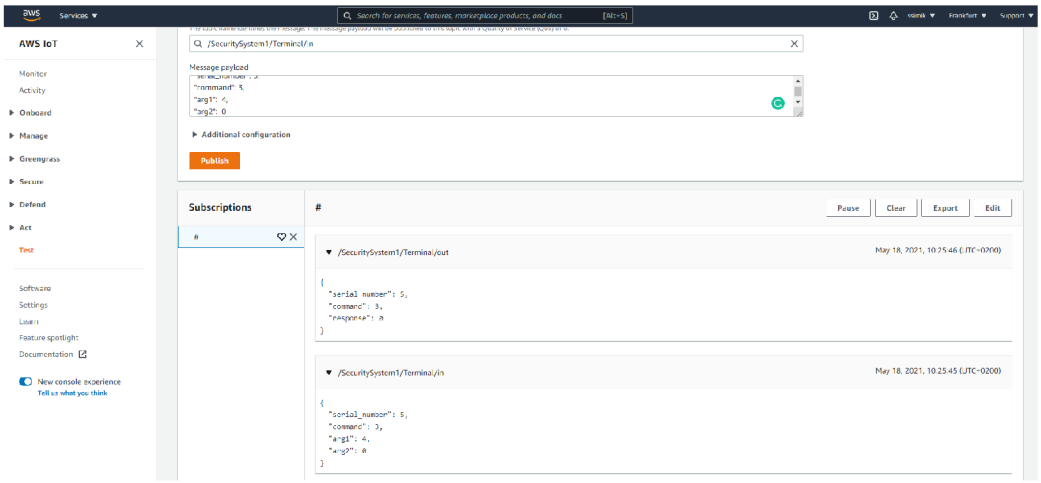
- ZONEWAY F8[5] (klávesnice, snímač otisků prstů, možnost integrace do většího bezpečnostního systému) - 3617 Kč
- ZONEWAY TF1W[6] (klávesnice, snímač otisků prstů, RFID, možnost dálkového ovládání) - 2799 Kč
- Sygonix SY-3776414[7] (snímač otisku prstů, RFID) - 2790 Kč

[1] https://www.parallax.com/product/pir-sensor-with-led-signal/
[2] https://www.parallax.com/product/fingerprint-scanner/
[3] https://cz.mouser.com/
[4] https://www.digikey.com/en/products/detail/espressif-systems/ESP-EYE/9838645
[5] https://www.mall.cz/
[6] https://www.mall.cz/
[7] https://www.conrad.cz/

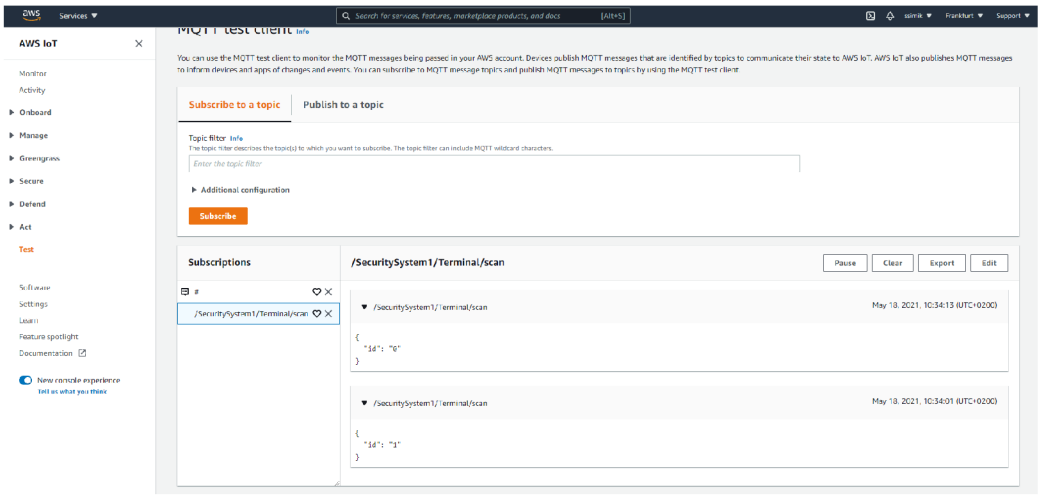
Příloha C

Chování systému

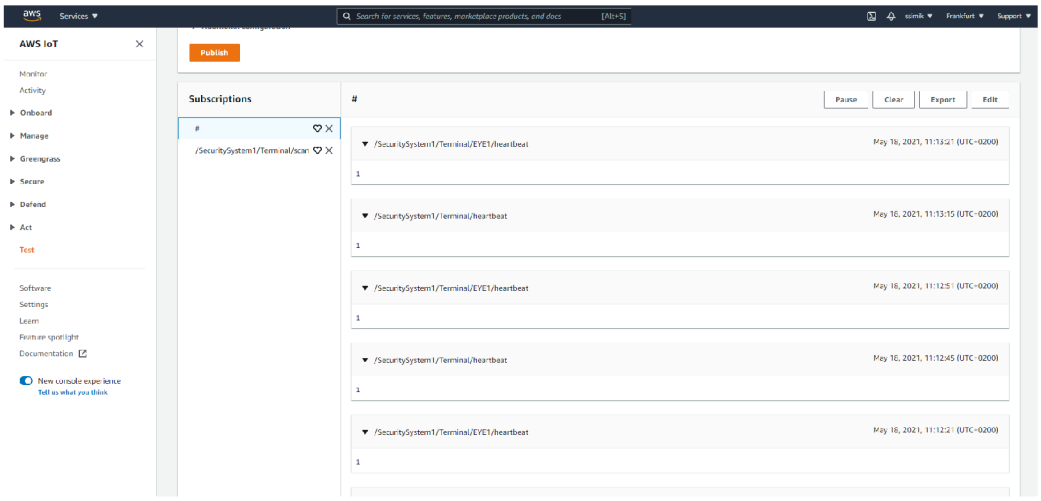
Tato příloha obsahuje několik obrázků zobrazujících chování systému.



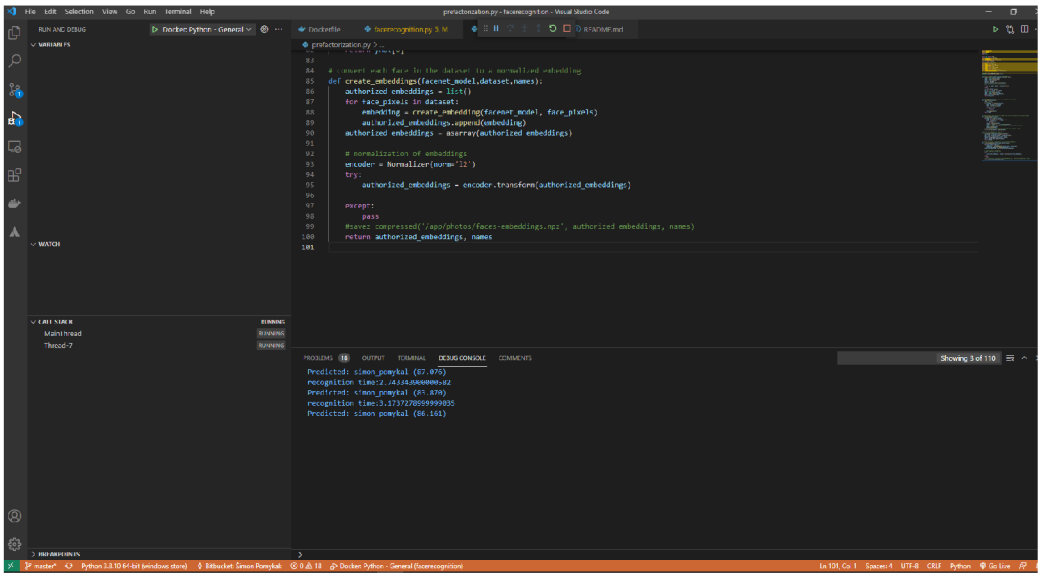
Obrázek C.1: Zaslání příkazu modulu terminálu pomocí MQTT zprávy a odpověď na tento příkaz.



Obrázek C.2: Zprávy z terminálu obdržené při jednotlivých sejmutích otisku prstu.



Obrázek C.3: Heartbeat zprávy zasílaná jednotlivými zařízeními.



Obrázek C.4: Úspěšné rozpoznání několika snímků k kamery, které byly zachyceny při detekci pohybu pomocí PIR čidla.