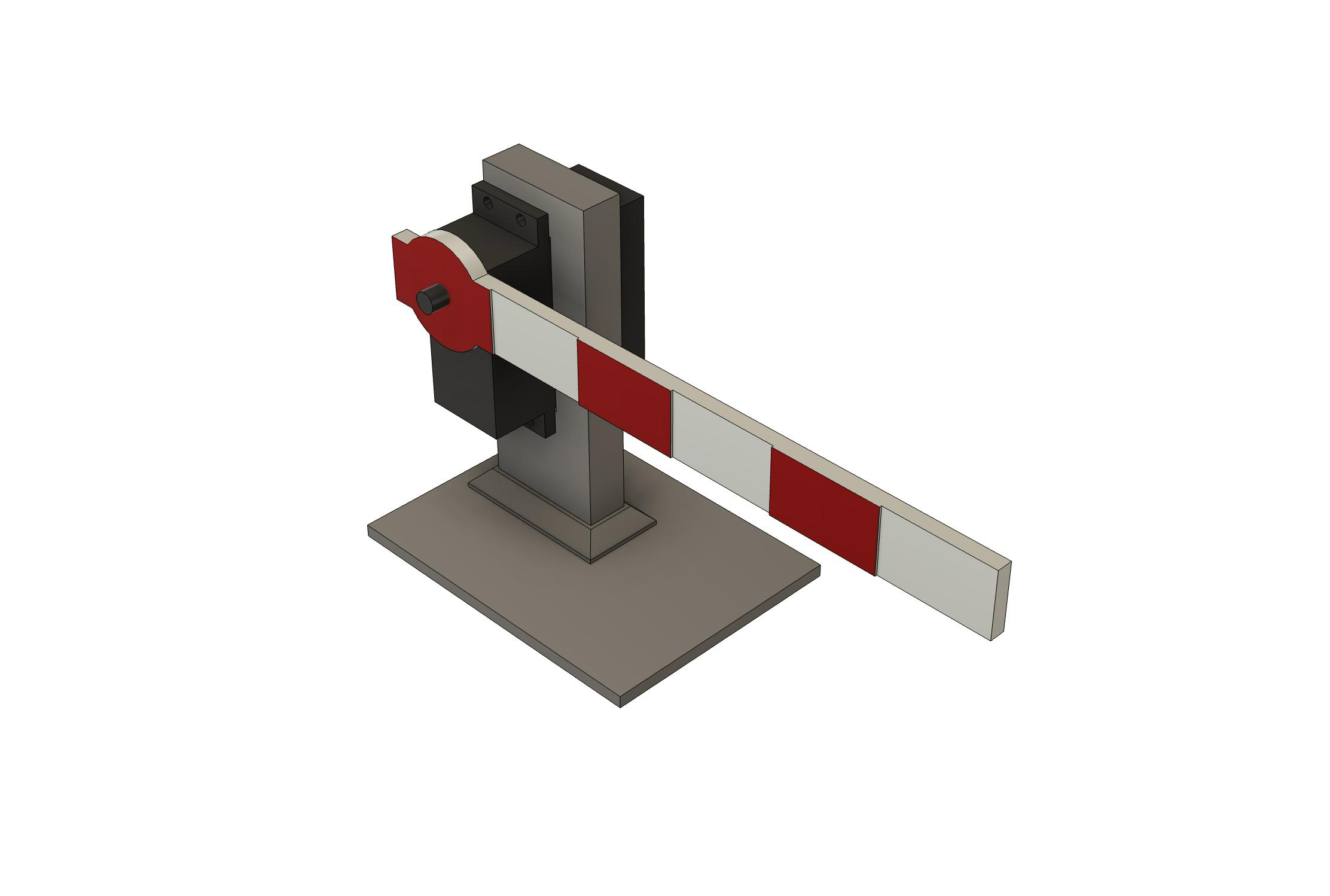
# Kamerový systém s detekcí obličeje

Jan Koubek C3b, obor C

Projekt se zabývá problematkou rozpoznání obličeje, pomocí kterého se ovládají výstupní zařízení (v tomto případě LED, servozávora).



## Obsah

[Kamerový systém s rozpoznáním obličeje 1](#_Toc402845875)

[Úvod 1](#_Toc127669612)

[Technické specifikace 2](#_Toc756743077)

[Ekonomická rozvaha 3](#_Toc1208501952)

[Vývoj 4](#_Toc717380025)

[Testování 4](#_Toc1323946877)

[Scénář 1: Spuštění programu na Raspberry Pi 5](#_Toc658221454)

[Scénář 2: Inicializace kamery, vytvoření rozpoznávacího objektu a otevření okna 5](#_Toc1943606099)

[Scénář 3: Nasnímání a rozpoznání obličejů 5](#_Toc336498841)

[Scénář 4: Uložit a načíst rozpoznávací objekt 6](#_Toc302838205)

[Scénář 5: Přidat výstupní zařízení 7](#_Toc88855250)

[Manuál 7](#_Toc1755666388)

[Konfigurace systému před spuštěním 8](#_Toc1273432317)

[Diagram číslování GPIO pinů 8](#_Toc804923032)

[Zapnutí PWM na Raspberry Pi 10](#_Toc764959760)

[Práce s programem 10](#_Toc745070367)

[Spuštění 11](#_Toc920950154)

[Přehled příkazů systému 11](#_Toc349655187)

[Základní způsoby užití programu 11](#_Toc1132394567)

[Inicializace kamery 12](#_Toc1247364692)

[Vytvoření/načtení rozpoznávacího objektu 12](#_Toc476273195)

[Uložení rozpoznávacího objektu 12](#_Toc1012283064)

[Přidání nového obličeje 12](#_Toc295877417)

[Otevření okna s vizualizací rozpoznání obličejů 12](#_Toc722488744)

[Přidání výstupních zařízení 12](#_Toc1733614573)

[Úprava nastavení rozeznávání 13](#_Toc346297863)

[Licence 13](#_Toc576029382)

[GitHub repozitář 13](#_Toc1943417985)

[Závěr 13](#_Toc1804947990)

## Úvod

Jedná se o software napsaný v jazyce Java, který je určený pro systémy Raspberry Pi (dále jen RPi). Program pracuje s USB kamerou, pomocí které se dokáže naučit rozeznávat obličeje. Jednotlivé obličeje se nejprve nasnímají kamerou, pomocí uživatelského rozhraní se jim přiřadí jméno a úroveň přístupu. Program následně umí rozpoznat jednotlivé obličeje mezi sebou. Program dále umí pracovat s výstupními zařízeními zapojenými do pinů GPIO, které mění svůj stav na základě rozpoznání autorizovaného obličeje. Celý program je vysoce konfigurovatelný a umí pracovat s prakticky neomezeným množstvím zařízení fungujících na principu jednoduchého digitálního výstupu nebo PWM (pulse-width-modulation) výstupu, který je plně konfigurovatelný. V případě tohoto projektu výstup PWM ovládá servomechanismus spojený se závorou. Nedílnou součástí projektu je i indikátor v podobě LED diody, která například varuje před nadcházejícím zavření závory, apod. Počet obličejů, který program může rozpoznat, není nijak omezený (Resp. je omezený pamětí a prostředky hostitelského počítače). Jednotlivé instance natrénovaného rozpoznávacího objektu, který umí rozpoznat obličeje, se ukládají souborů a mohou být znovu použity při dalším spuštění programu bez nutnosti opětovně trénovat algoritmus. Díky ukládání do souborového systému může být k dispozici veliké množství instancí rozpoznávacího objektu najednou. Rozpoznávání je vysoce konfigurovatelné, včetně nastavení citlivosti, tolerance jistoty rozpoznání a prodlevy mezi zapnutým a vypnutým stavem. Toto zaručuje vysokou flexibilitu využití napříč různými podmínkami snímání kamery a využití výstupních zařízení.

## Technické specifikace

Cílové zařízení: Raspberry Pi 4 Model B Rev 1.5 se 4 ARM jádry a pamětí 8 GB RAM, případně RPi 5 8 a 16 GB RAM (s ohledem na využití rozpoznávací technologie)

Softwarové požadavky:

* OS: Raspberry Pi OS (Debian Bookworm 12.9) nebo novější
* Java Runtime Environment verze 23.0.1 nebo novější

## Ekonomická rozvaha

Zařízení je technologický experimentální systém a není záměrem srovnávat jej s komerčními produkty na trhu. Vzhledem k použití trivializovaného demonstračního HW v podobě samotného počítače a jednoduché brány, tištěné na 3D tiskárně se jedná především o demonstrační zařízení. Pro srovnání s komerčními produkty by bylo třeba vyrobit a otestovat zařízení schopné realizovat funkčnost v rozměrech a podání pro fyzické osoby. Tuto variantu projekt v této fázi nezvažoval.

## Vývoj

Celý program byl vytvořen pomocí programovacího jazyku Java, který jsem zvolil z důvodu vlastní zkušenosti s jazykem. Nejdůležitější využitou externí knihovnou je [OpenCV](https://opencv.org/), která byla zkompilována s experimentálními moduly. V projektu realizuje snímání obrazu z kamery, rozpoznání obličejů a úpravu obrazu. Další využitou knihovnou je [Pi4J](https://www.pi4j.com/), která v projektu zajišťuje výstup do GPIO pinů.

Pro vývoj jsem použil Visual Studio Code na platformě MAC OS s využitím rozšíření pro vzdálený vývoj (remote development) přes protokol SSH a remote VSC server. Toto řešení jsem zvolil s ohledem na komfort při vývoji na platformě MAC a zároveň ladění programu na cílovém prostředí RPi.

Během vývoje jsem se inspiroval dokumentací OpenCV a Pi4J s cílem spolehlivého, optimalizovaného a přenosného kódu.

Vývoj probíhal v celkem 3 etapách na základě iterací zadání. V první etapě vývoje se jednalo o “proof-of-concept”. Program uměl pouze zjistit a zvýraznit obličej nasnímaný kamerou. Ve druhé etapě přibylo konzolové rozhraní, plně implementované rozpoznávání obličejů a výstup pomocí LED světla. Ve třetí etapě pak přibyla podpora pro PWM výstupní zařízení, byla sestrojena závora a proběhla spousta testování a ladění, aby projekt dosáhl požadovaného stavu.

Kód je logicky rozdělen do tříd s vysokou mírou abstrakce, která umožňuje velkou rozšiřitelnost. Je použit návrhový vzor Singleton pro přístup ke statickému kontextu odkudkoli z kódu.

## Testování

Funkčnost projektu byla testována několika spolužáky a rodinou. Otestovaly se následující scénáře:

### Scénář 1: Spuštění programu na Raspberry Pi

Podmínky: Správně zkonfigurovaný program a RPi, vstupní bod terminál OS.

|  |  |  |  |
| --- | --- | --- | --- |
| **Název kroku** | **Popis** | **Očekávaný stav** | **Výsledek** |
| Spustit program | Použít příkaz java –jar face-recognition.jar | Program se spustí | OK |

### Scénář 2: Inicializace kamery, vytvoření rozpoznávacího objektu a otevření okna

Podmínky: Funkční scénář 1, vstupní bod terminál OS.

|  |  |  |  |
| --- | --- | --- | --- |
| **Název kroku** | **Popis** | **Očekávaný stav** | **Výsledek** |
| Spustit program | Použít příkaz java –jar face-recognition.jar | Program se spustí | OK |
| Inicializace kamery | Zadat příkaz “ic” a zvolit index 0 | Kamera se inicializuje bez problému | OK |
| Vytvoření rozpoznávacího objektu | Zadat příkaz “cr” | Rozpoznávací objekt se vytvoří bez problému | OK |
| Otevření okna | Zadat příkaz “ow” | Otevře se okno s výstupem kamery | OK |

### Scénář 3: Nasnímání a rozpoznání obličejů

Podmínky: Funkční scénář 2, provést kroky scénáře 2, 2 přítomné osoby, vstupní bod rozhraní aplikace.

|  |  |  |  |
| --- | --- | --- | --- |
| **Název kroku** | **Popis** | **Očekávaný stav** | **Výsledek** |
| Nasnímat první obličej | Příkaz tf | Odezva “Started tracking face” | OK |
| Uložit první obličej | Příkaz sf, authorized: yes | Přidá se první osoba | OK |
| Nasnímat druhý obličej | Příkaz tf | Odezva “Started tracking face” | OK |
| Uložit druhý obličej | Příkaz sf, authorized: no | Přidá se druhá osoba | OK |
| Rozpoznání | Vyzkoušet, zda program rozezná od sebe obličeje | Přiřazení jmen správné | OK |

### Scénář 4: Uložit a načíst rozpoznávací objekt

Podmínky: Funkční scénář 3, provést kroky scénáře 3, vstupní bod rozhraní aplikace, konec scénáře 3.

|  |  |  |  |
| --- | --- | --- | --- |
| **Název kroku** | **Popis** | **Očekávaný stav** | **Výsledek** |
| Uložit rozpoznávací objekt | Příkaz sr, cílový adresář recognizers/test1 | Vytvoří se na daném místě adresář | OK |
| Ukončit program | Příkaz exit | Program se ukončí | OK |
| Spustit program | Použít příkaz java –jar face-recognition.jar | Program se spustí | OK |
| Načíst rozpoznávací objekt | Příkaz lr, stejný adresář | Rozpoznávací objekt se načte | OK |
| Vyzkoušet rozpoznání osob | Ověřit správnost rozpoznání | Rozpoznání správné | OK |

### Scénář 5: Přidat výstupní zařízení

Podmínky: Funkční scénář 4, provést kroky scénáře 4, zapojená závora do výchozích pinů, vstupní bod rozhraní aplikace.

|  |  |  |  |
| --- | --- | --- | --- |
| **Název kroku** | **Popis** | **Očekávaný stav** | **Výsledek** |
| Přidat GPIO zařízení | Příkaz ag | Otevře se dialog na přidání GPIO zařízení | OK |
| Zkonfigurovat zařízení | PWM, “blinking”: no, kanál 0, výchozí hodnoty | Zařízení se přidá | OK |
| Ověřit funkčnost zařízení | Rozpoznání osoby 1 | Zařízení se přepne do otevřeného stavu | OK |
| Ověřit přepnutí stavu na zavřeno | Skrýt obličej osoby 1 | Zařízení se po několika vteřinách zavře | OK |
| Ověřit nepropuštění osoby 2 | Rozpoznání osoby 2 | Zařízení se nepřepne do stavu otevřeno | OK |

## Manuál

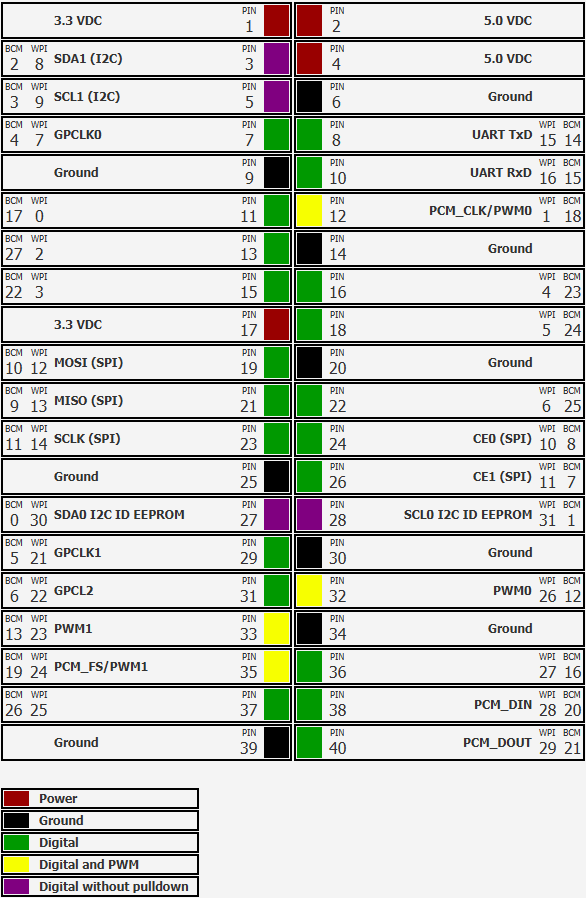
### Konfigurace systému před spuštěním

Předpokládá se plně konfigurované RPi. Spustitelný soubor JAR se musí vyskytovat ve stejné složce jako všechny závislosti. Závislostmi jsou složky “classes”, “lib” a “opencv”.

Před spuštěním programu zapojte všechny výstupní zařízení do správných pinů. Na poskytnutém RPi je nastavený PWM pin 12 na kanál 0. Pro správné fungování na poskytnutém RPi zapojte žlutý kabel servo motoru do BCM pinu 12, zbytek kabelů do libovolných stejně zbarvených pinů. LED světlo zapojte černým koncem do BCM pinu 4.

Řiďte se [diagramem číslování GPIO pinů níže](#_Diagram_číslování_GPIO).

#### Diagram číslování GPIO pinů



Označení jednotlivých GPIO pinů se provádí pomocí hodnoty **BCM** (viz. tabulka výše). Zařízení s jednoduchým digitálním výstupem se mohou zapojit do libovolného zeleného pinu. PWM zařízení fungují pouze při zapojení do žlutých “Digital and PWM” pinů

#### Zapnutí PWM na Raspberry Pi

Pro použití PWM výstupních zařízení se musí na Raspberry Pi nejprve PWM zkonfigurovat.

Kvůli hardwarové implementaci ovládání PWM kanálů se musí PWM zapnout v konfiguračním souboru /boot/firmware/config.txt, kde se použije jedna z následujících konfigurací na základě Vašich zvolených PWM pinů ([viz. diagram číslování GPIO pinů](#_Diagram_číslování_GPIO)):

[all]  
dtoverlay=pwm  
# GPIO 18 = channel 0  
  
[all]  
dtoverlay=pwm-2chan  
# GPIO 18 = channel 0  
# GPIO 19 = channel 1  
  
[all]  
dtoverlay=pwm-2chan,pin=12,func=4,pin2=13,func2=4  
# GPIO 12 = channel 0  
# GPIO 13 = channel 1

Tato konfigurace přiřadí dané GPIO piny na dané PWM kanály. RPi poskytuje pouze 2 současné PWM kanály. Poskytnuté RPi má předem nastavený GPIO pin 12 na kanál 0.

### Práce s programem

#### Spuštění

Pro spuštění otevřete konzolové rozhraní ve složce se souborem “face-recognition.jar” a všemi závislostmi, následně zadejte tento příkaz:

*java -jar face-recognition.jar*

Po spuštění příkazu se spustí textové rozhraní programu.

#### Přehled příkazů systému

|  |  |  |
| --- | --- | --- |
| **Příkaz** | **Alternativní způsob zápisu** | **Vysvětlení** |
| help |  | Zobrazí všechny dostupné příkazy |
| configuration | conf | Zobrazí nabídku s možnou úpravou různých nastavení |
| addgpio | ag | Zobrazí dialog pro přidání nového výstupního zařízení |
| openwindow | ow | Otevře okno zobrazující výstup kamery s rozpoznanými obličeji |
| initcamera | ic | Zobrazí dialog pro inicializaci kamery |
| createrecognizer | cr | Vytvoří nový rozpoznávající objekt |
| loadrecognizer | lr | Zobrazí dialog pro načtení rozpoznávacího objektu ze souborového systému |
| saverecognizer | sr | Zobrazí dialog pro uložení rozpoznávacího dialogu do souborového systému |
| trackface | tf | Začne snímat nový obličej |
| saveface | sf | Přidá nově nasnímaný obličej do rozpoznávacího objektu |
| faces | f | Zobrazí a rozpozná všechny obličeje, které kamera momentálně vidí |
| smile | s | Zobrazí počet úsměvů, který kamera aktuálně vidí |
| exit |  | Ukončí program po 5 vteřinách pro reset výstupních zařízení |

#### Základní způsoby užití programu

##### Inicializace kamery

Inicializaci kamery lze provést pomocí příkazu ic. Po zadání příkazu budete muset zadat index kamery, které chcete použít. Tento index naleznete zjištěním čísla souboru video ve složce /dev, který odpovídá Vaší kameře. Pouze mimořádně se však kamera nalézá na jiném indexu než 0

##### Vytvoření/načtení rozpoznávacího objektu

Před vytvoření/načtení rozpoznávacího objektu **musí** být nejprve inicializována kamera. Vytvoření nového rozpoznávacího objektu se provede příkazem cr. Načtení z uložiště se provede pomocí příkazu lr a následném zadání relativního umístění **adresáře** s rozpoznávacím objektem.

##### Uložení rozpoznávacího objektu

Aby se program nemusel při každém spuštění učit znovu a znovu stejné obličeje, lze použít příkaz sr, který do zadaného relativního umístění adresáře vytvoří adresář s rozpoznávajícím objektem.

##### Přidání nového obličeje

Pro přidání nového obličeje se musí nejprve nový obličej nasnímat pomocí příkazu tf. Po uplynutí cca. 30 vteřin obličej uložíte do rozpoznávacího objektu příkazem sf. Nastavte osobu jako autorizovanou pokud chcete, aby se výstupní zařízení spouštěli po rozpoznání tohoto obličeje.

##### Otevření okna s vizualizací rozpoznání obličejů

Pokud má Váš operační systém podporu pro grafické rozhraní, můžete použít příkaz ow pro otevření okna s živým záznamem z kamery a vizualizací rozpoznání obličejů a úsměvů.

##### Přidání výstupních zařízení

Pro přidání výstupních zařízení použijte příkaz ag. Následně vyberte odpovídající nastavení pro Vaše zařízení. Při přidávání Digital Output zařízení budete muset doplnit číslo GPIO pinu, do kterého je zařízení zapojené ([viz. diagram číslování GPIO pinů](#_Diagram_číslování_GPIO)). Při přidávání PWM zařízení budete muset zadat PWM kanál ([viz. Zapnutí PWM na Raspberry Pi](#_Zapnutí_PWM_na)). Pokud jste zapojili všechny výstupní zařízení dle doporučeného zapojení ([viz. Konfigurace systému před spuštěním](#_Konfigurace_systému_před)), nastavte Digital Output zařízení (LED) pin 4 a možnost “blinking” na “y”, následně nastavte PWM zařízení na channel 0, “blinking” na ”n”, a zbytek možností podle výchozích hodnot.

##### Úprava nastavení rozeznávání

Pro úpravu citlivosti, délky opoždění vypnutí nebo maximální tolerance jistoty rozeznání obličeje napište příkaz conf. Následně vyberte nastavení, které chcete upravit, a nastavte ho na požadovanou hodnotu.

## Licence

Celý projekt je publikován pod licencí GNU GPLv3. S touto licencí se můžete seznámit na oficiálních stránkách GNU:

<https://www.gnu.org/licenses/gpl-3.0.html>

## GitHub repozitář

<https://github.com/KoubekJ1/face-recognition>

## Závěr

Projekt trval celkem cca. 40 hodin práce. Pokud bych projekt měl začít znovu, vybral bych si programovací jazyk C místo Javy kvůli jednoduššímu kompilování knihoven a lepšímu přístupu k hardwaru. Kód by mohl mít lepší architekturu a více použitých návrhových vzorů. Tyto nepřesnosti byli způsobeny zejména chybným plánováním. I přes to jsem s projektem vcelku spokojen. Ponořil jsem se do témata, které pro mě bylo zcela nové, a naučil jsem mnoho věcí spojených s propojením hardwaru se softwarem, kompilování knihoven a prací s build systémy, a o práci s Raspberry Pi a Linuxem samotným. Celý systém má prostor pro vylepšení, hlavně co se týče výkonu a rozpoznávání úsměvů.