



TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky
a mezioborových studií ■

Možnosti rozpoznávacích algoritmů v robotu NAO

Bakalářská práce

Studijní program: B2646 – Informační technologie
Studijní obor: 1802R007 – Informační technologie

Autor práce: **Jan Vacek**
Vedoucí práce: Ing. Miroslav Holada, Ph.D.





TECHNICAL UNIVERSITY OF LIBEREC
Faculty of Mechatronics, Informatics
and Interdisciplinary Studies ■

Possibilities of recognition algorithms in a NAO robot

Bachelor thesis

Study programme: B2646 – Information Technology
Study branch: 1802R007 – Information Technology

Author: **Jan Vacek**
Supervisor: Ing. Miroslav Holada, Ph.D.



ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: Jan Vacek
Osobní číslo: M14000079
Studijní program: B2646 Informační technologie
Studijní obor: Informační technologie
Název tématu: Možnosti rozpoznávacích algoritmů v robotu NAO
Zadávající katedra: Ústav informačních technologií a elektroniky


Z á s a d y p r o v y p r a c o v á n í :

1. Seznamte se s roboty NAO na pracovišti školitele a s dosud řešenými projekty majícími souvislost s těmito roboty.
2. Prozkoumejte problematiku rozpoznávacích algoritmů implementovaných na senzorický systém robotů NAO. Zaměřte se především na možnosti zpracování obrazu.
3. Navrhňte software, který umožní snadno testovat rozpoznávací možnosti robotu.
4. Navržené řešení prakticky ověřte.


Rozsah grafických prací: Dle potřeby dokumentace
Rozsah pracovní zprávy: cca 30-40 stran
Forma zpracování bakalářské práce: tištěná/elektronická
Seznam odborné literatury:

- [1] ALDEBARAN. Aldebaran documentation: NAO Documentation [online]. 2016 [cit. 2016-10-16]. Dostupné z: http://doc.aldebaran.com/2-1/home_nao.html
- [2] NOVÁK, Petr. Mobilní roboty: pohony, senzory, řízení. 1. vyd. Praha: BEN - technická literatura, 2005. ISBN 80-7300-141-1.

Vedoucí bakalářské práce: Ing. Miroslav Holada, Ph.D.
Ústav informačních technologií a elektroniky
Konzultant bakalářské práce: Bc. Šimon Škapík
Datum zadání bakalářské práce: 12. září 2016
Termín odevzdání bakalářské práce: 15. května 2017


prof. Ing. Zdeněk Plíva, Ph.D.
děkan




prof. Ing. Ondřej Novák, CSc.
vedoucí ústavu

V Liberci dne 12. září 2016

Prohlášení

Byl jsem seznámen s tím, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé bakalářské práce pro vnitřní potřebu TUL.

Užiji-li bakalářskou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Bakalářskou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím mé bakalářské práce a konzultantem.

Současně čestně prohlašuji, že tištěná verze práce se shoduje s elektronickou verzí, vloženou do IS STAG.

Datum: 13. 5. 2017

Podpis: 

Poděkování

Na tomto místě bych rád poděkoval svému vedoucímu práce, panu Ing. Miroslavu Holadovi, Ph.D., za vedení bakalářské práce a věcné připomínky. Dále bych chtěl poděkovat celé své rodině, zejména svým rodičům, kteří mě, nejen během této bakalářské práce, podporovali.

Abstrakt

Tato práce je zaměřena na možnosti rozpoznávacích algoritmů implementovaných v humanoidním robotu NAO firmy Aldebaran. Zabývá se popisem a použitím algoritmů implementovaných na senzorický systém robotu NAO, zejména pak možnostmi zpracování obrazu. Práce je dále zaměřena na návrh a realizaci programu, který by umožnil snadné testování rozpoznávacích možností robotu.

Program pro testování algoritmů je řešen jako spustitelná aplikace pro platformu Windows napsaný v jazyce Python. Jednotlivé rozpoznávací algoritmy je pak možné spouštět přímo pomocí tlačítek na robotu. Grafická část programu, běžící na počítači s Windows, slouží pro navázání spojení s robotem NAO a spuštění samotného skriptu.

Klíčová slova

humanoidní robot NAO, rozpoznávací algoritmy, zpracování obrazu

Abstract

This thesis is focused on capabilities of recognition algorithms implemented on humanoid robot NAO made by Aldebaran Robotics. It handles description and applications of algorithms implemented on the sensoric system of NAO robot, precisely image processing. The thesis is also focused on the application design and realization. Thanks to the application, it will be easy to test all of the recognition capabilities of the robot.

The algorithm testing program is realised as an executable application made for Windows platform. The application is coded in Python. Each recognition algorithm can be started by pressing individual buttons on the robot. Main purpose of the Windows application is to secure the communication with the robot which is followed by initiation of the script itself.

Keywords

humanoid robot NAO, recognition algorithms, image processing

Obsah

1	Úvod.....	12
1.1	Motivace.....	12
1.2	Cíle práce.....	12
2	Stávající stav rozpoznávacích algoritmů	13
2.1	Příklady použití.....	13
2.1.1	Rozpoznávání dopravních značek.....	13
2.1.2	Rozpoznávání SPZ.....	14
2.1.3	Rozpoznávání textu	14
2.1.4	Rozpoznávání obličejů	14
2.1.5	Rozpoznání řeči.....	15
2.1.6	Rozpoznávání hudby.....	15
3	Humanoidní robot NAO	16
3.1	Technické parametry.....	16
3.2	Hardwarové vybavení.....	17
3.2.1	Záznam zvuku	17
3.2.2	Reprodukce zvuku	17
3.2.3	Záznam obrazu	18
3.2.4	LED diody	18
3.2.5	Infračervené senzory	20
3.2.6	Dotykové senzory	20
3.2.7	Ultrazvukové senzory	21
3.3	Softwarové vybavení.....	22
3.4	Implementované rozpoznávací algoritmy.....	22
3.4.1	Možnosti sledování.....	22
3.4.2	Rozpoznatelné objekty	23
4	Návrh a možnosti realizace softwarového řešení.....	25
4.1	Možnosti návrhu	25
4.2	Možnosti realizace návrhu	27
4.2.1	Využití Choregraphe	27
4.2.2	Využití SDK.....	27
4.3	Finální volba	28
4.3.1	Realizace grafického rozhraní.....	28

4.3.2	Ovládání robota	30
5	Funkce realizovaného programu	31
5.1	Výběr režimu	31
5.2	Výběr typu úlohy	31
5.2.1	Rozpoznání objektů	31
5.2.2	Pohyb	32
5.3	Výběr varianty úlohy	32
5.3.1	Rozpoznání osob a obličejů	33
5.3.2	Rozpoznání červeného míčku	33
5.3.3	Rozpoznání NAO marků	34
6	Závěr	36
7	Seznam použité literatury	37
8	Přílohy	38

Seznam obrázků

Obrázek 1 - rozpoznání dopravních značek - převzato z [4]	13
Obrázek 2 - proces zpracování obrazu před rozpoznáním SPZ - převzato z [5]	14
Obrázek 3 - klíčování obličejových rysů - převzato z [6]	15
Obrázek 4 – Robot NAO	16
Obrázek 5 - mikrofony.....	17
Obrázek 6 – hlasitý reproduktor	17
Obrázek 7 - kamery	18
Obrázek 9 - rozmístění diod - oči	19
Obrázek 8 - rozmístění diod – dotykové senzory na hlavě	19
Obrázek 10 - rozmístění diod - reproduktory	19
Obrázek 11 - infračervené senzory	20
Obrázek 12 - ultrazvukové senzory.....	21
Obrázek 13 - NAO mark – převzato z [1].....	24
Obrázek 14 - návrh ovládání programu	26
Obrázek 15 - schéma komunikace mezi PC a robotem.....	28
Obrázek 16 - výsledné grafické rozhraní.....	29
Obrázek 17 - vývojový diagram kontroly validity vstupů z GUI	29
Obrázek 18 - ovládání programu - výběr režimu	31
Obrázek 20 - ovládání programu - výběr pozice	32
Obrázek 19 - ovládání programu - výběr objektu	32
Obrázek 21 - NAO mark (64) použitý pro testování - převzato z [1]	35

Seznam zkratek

- ABS-PC – Akrylonitril-butadién-styrén-polykarbonát – typ umělé hmoty
- PA-66 – Polyamid 66 – typ umělé hmoty
- XCF-30 – typ umělé hmoty, podobný PA-66, vyztužený uhlíkovými vlákny
- SDK – Software Development Kit – balíček funkcí pro vývoj software
- OCR – Optical Character Recognition – optické rozpoznávání znaků, metoda např. pro digitalizaci dokumentů
- USB – Universal Serial Bus – univerzální sériová sběrnice
- LED – Light Emitting Diode – světlo vyzařující dioda

1 Úvod

1.1 Motivace

Jednou z nejmodernějších věd v oboru mechatroniky je právě robotika. V současné době se setkáváme s různými implementacemi robotů do běžného života. A to například v podobě bezpilotních dronů, či autonomních vozidel. Stále častěji se setkáváme s takzvanými kolaborativními roboty, jejichž cílem je výpomoc člověku. Zejména při úkonech, které vyžadují vysokou a stále stejnou přesnost.

Samostatnou skupinou robotů jsou humanoidní roboti. Humanoidní robot neboli robot, který je svým vzhledem a stavbou těla podobný člověku, je v současné době využíván jako výzkumná pomůcka v mnoha odvětvích vědy. Pro zdokonalení humanoidních robotů je důležité pochopit nejen strukturu a chování lidského těla, ale také princip lidského poznání, které vypovídá o tom, jak je člověk schopen vnímat své okolí a jak získává své motorické dovednosti. Právě na schopnosti vnímání okolí, přesněji na zpracování obrazové informace, je zaměřena tato práce.

1.2 Cíle práce

Cílem práce je seznámit se s humanoidními roboty NAO a možnostmi rozpoznávacích algoritmů implementovaných na jejich senzorický systém. Dalším cílem je na základě získaných poznatků vytvořit aplikaci, která by umožnila testování výrobcem implementovaných algoritmů. Ovládání samotné aplikace by mělo být natolik snadné, aby k jeho pochopení nebyl zapotřebí složitý návod k obsluze.

Tato dokumentace je rozdělena na řešeršní část a část popisující návrh a vývoj aplikace pro testování rozpoznávacích možností robotu. Rešeršní část je zaměřena na popis současného stavu rozpoznávacích algoritmů, včetně názorných příkladů jejich využití v reálném životě. Rešeršní část dále podrobně popisuje hardwarové a softwarové vybavení robota NAO, výrobcem implementované rozpoznávací algoritmy a jejich možnosti. Druhá část popisuje vývoj testovací aplikace, od návrhu až po výsledné řešení.

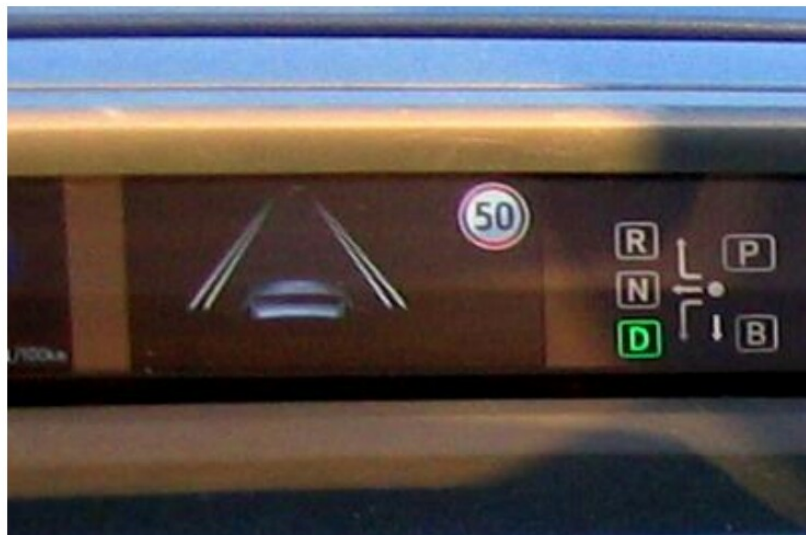
2 Stávající stav rozpoznávacích algoritmů

Rozpoznávací algoritmus je takový algoritmus, který analyzuje vstupní data, například obraz nebo zvuk, a na základě získaných údajů poskytuje uživateli požadované informace. Rozpoznávací algoritmy mohou zpracovávat data různého charakteru.

2.1 Příklady použití

2.1.1 Rozpoznávání dopravních značek

V dnešní době se stává téměř běžným, že nově vyrobené automobily jsou schopné rozpoznávat svislé dopravní značení. Kamera ve vozidle snímá prostor před vozem, v tomto prostoru pak pomocí algoritmu pro optické rozpoznávání znaků, z angličtiny OCR – Optical Character Recognition, vyhledává dopravní značky. Informace o rozpoznané dopravní značce je pak zobrazena řidiči na displeji. Díky této aplikaci se pak řidič nemusí tolik soustředit na zapamatování dopravní značky.



Obrázek 1 - rozpoznání dopravních značek - převzato z [4]

2.1.2 Rozpoznávání SPZ

Další poměrně užitečnou demonstrací využití rozpoznávacích algoritmů v praxi je rozpoznávání SPZ neboli státní poznávací značky. V tomto případě se algoritmus soustředí na nalezení tabulky registrační značky. Po úspěšném nalezení značky se pak pomocí OCR snaží rozpoznat jednotlivé znaky na poznávací značce. S aplikací tohoto algoritmu se můžeme setkat ve většině nákupních center, kde informace o registrační značce slouží k automatickému otevírání závory. Tento algoritmus by se dal využít například jako zabezpečovací prvek na firemních parkovištích apod., kde by bezpečnostní systém dovolil vjezd na parkoviště jen povoleným vozidlům.



Obrázek 2 - proces zpracování obrazu před rozpoznáním SPZ - převzato z [5]

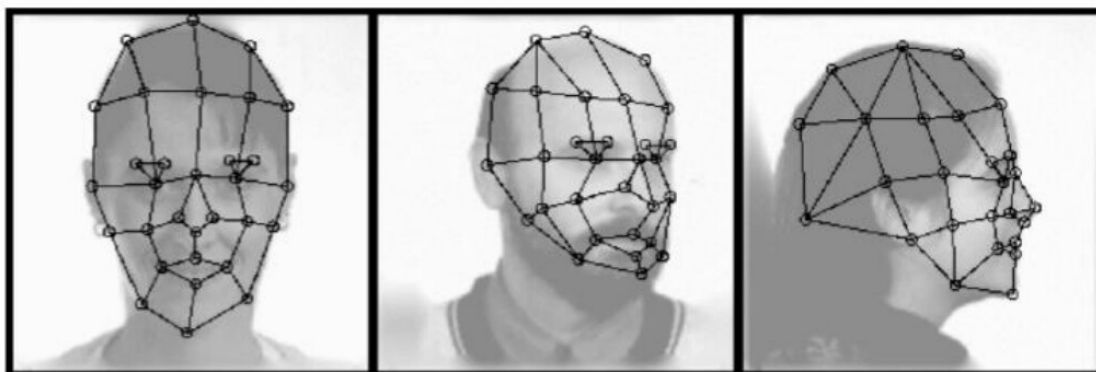
2.1.3 Rozpoznávání textu

Algoritmy pro rozpoznávání textu opět využívají princip optického rozpoznání znaků. Kvalita výsledného rozpoznání velmi závisí na kvalitě pořízeného obrázku. Tento algoritmus je využíván zejména k digitalizaci tištěných dokumentů, dále může být využit k okamžitému překladu různých značek a nápisů.

2.1.4 Rozpoznávání obličejů

Mezi další praktická využití rozpoznávacích algoritmů patří i rozpoznávání obličejů. Většina algoritmů funguje na principu identifikace obličejových rysů z obrázku

obličej rozpoznávaného subjektu. Algoritmus může například analyzovat relativní pozici, velikost nebo tvar očí, nosu, čelisti nebo lícních kostí. Informace o těchto rysech jsou využity pro nalezení jiných obrázků se shodnými rysy. V praxi může aplikace tohoto algoritmu sloužit k zabezpečení budov, nebo může pomoci při identifikaci pachatele.



Obrázek 3 - klíčování obličejových rysů - převzato z [6]

2.1.5 Rozpoznání řeči

Algoritmy pro rozpoznávání řeči mohou v dnešní době sloužit k mnoha účelům. Mezi ně patří například převod řeči na text nebo hlasové ovládání různé elektroniky, od palubních počítačů v autě přes mobilní telefony až po chytrou domácnost. Algoritmy rozpoznávání řeči se dělí na dvě skupiny, na algoritmy závislé a nezávislé na mluvčím.

Algoritmy závislé na mluvčím jsou sice přesnější, avšak je nutné namluvit několik hodin nahrávek, aby byl algoritmus schopen se naučit model řeči mluvčího. Pro rozpoznání české řeči slouží například komerční řešení NewtonDictate, který je výsledkem práce na Technické univerzitě v Liberci.

2.1.6 Rozpoznávání hudby

Dalším využitím rozpoznávacích algoritmů je rozpoznání právě přehrávané hudby. Uživatel pomocí speciální aplikace zaznamená úryvek skladby, jejíž název chce zjistit. Aplikace nahrávku převede na řadu číselných hodnot, které se pokouší nalézt v databázi skladeb a získat tak její název.

3 Humanoidní robot NAO

Humanoidní robot NAO je výtvořem francouzské společnosti Aldebaran Robotics, která byla v roce 2012 odkoupena japonskou společností SoftBank Robotics. NAO je prvním humanoidním robotem této značky. Jeho první generace byla široké veřejnosti představena v roce 2006. Robot byl od té doby průběžně vyvíjen, až do současné páté generace. Od uvedení na trh až do současnosti se celosvětově prodalo přes 10 000 robotů NAO.

3.1 Technické parametry

Robot NAO je 573 milimetrů vysoký a 275 milimetrů široký. Hloubka, která je počítána s nataženými rukama robota činí 311 milimetrů. Robot NAO váží 5,4 kilogramu. Robot je vyroben především z plastových materiálů, konkrétně jsou použity plasty s označením ABS-PC, PA-66 a XCF-30.

Veškeré výpočetní operace má na starost integrovaný procesor od firmy Intel. Jedná se o typ Intel ATOM s modelovým označením Z350, jehož pracovní frekvence činí 1,6 GHz. Procesoru sekunduje 1 GB operační paměti RAM. Pro operační systém a uživatelsky vytvořené programy jsou k dispozici 2 GB FLASH paměti doplněné integrovanou Micro SDHC kartou s kapacitou 8 GB. Jako zdroj elektrické energie je v robotu umístěna li-ion baterie s kapacitou 2250 mAh. Nabití baterie by, v balení obsaženou nabíječkou, nemělo trvat déle než 3 hodiny, přičemž výdrž na jedno nabití se pohybuje, v závislosti na intenzitě používání, od 60 do 90 minut.

Pro nastavení síťového rozhraní je v robotově hlavě umístěn standardní RJ-45 konektor. Pro komunikaci s počítačem je v robotu dostupná WiFi, podporující standard IEEE 802.11 a/b/g/n. Pro připojení rozšiřujících periférií jako jsou například Microsoft Kinect, 3D senzor společnosti Asus nebo Arduino je určen USB konektor umístěný též v hlavě robota.

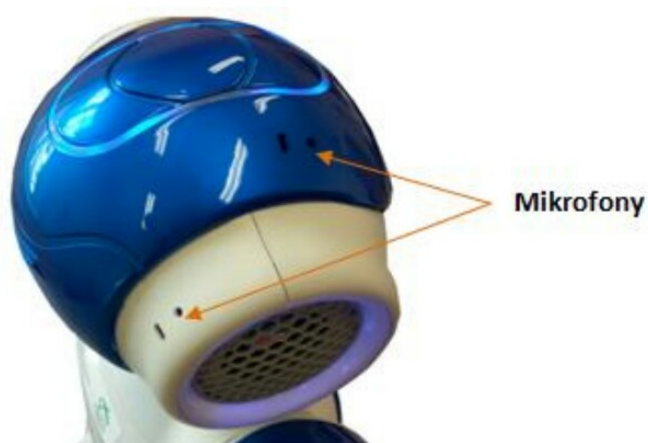


Obrázek 4 – Robot NAO

3.2 Hardwarové vybavení

3.2.1 Záznam zvuku

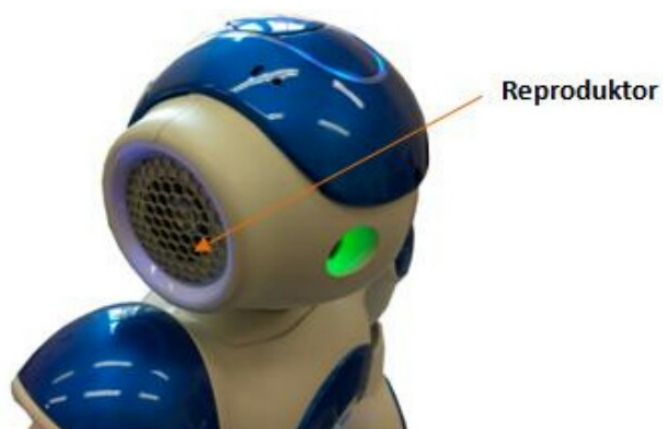
Pro záznam zvuku jsou na hlavě robota NAO umístěny celkem 4 mikrofony. Mikrofony mohou být použity pro záznam hlasových příkazů, na základě kterých může robot vykonávat další úlohy či pro záznam zvukové nahrávky, kterou robot následně přehraje. Mikrofony je také možné využít pro orientaci v prostoru, přesněji pro sledování zvuku. Přesnost určení silně závisí na členitosti okolí. Čím členitější okolí je, tím více dochází k odrazům zvukového signálu od okolních objektů, které mají za následek zkreslení orientace.



Obrázek 5 - mikrofony

3.2.2 Reprodukce zvuku

O reprodukci veškerých zvuků, ať už jde o syntetizaci řeči, chybová a systémová hlášení či o přehrávání audio souborů ve formátu MP3 nebo OGG, se stará dvojice reproduktorů umístěných na stranách hlavy robota NAO.

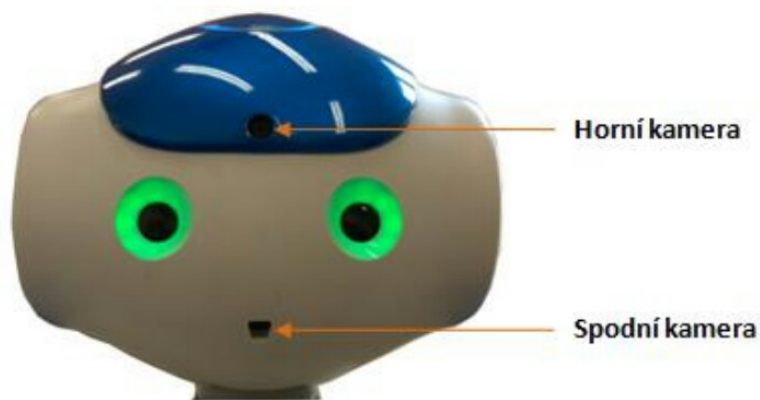


Obrázek 6 – hlasitý reproduktor

3.2.3 Záznam obrazu

Robotu NAO jsou k dispozici dvě kamery umístěné v hlavě. Tyto kamery robotu slouží k prostému pořízení fotky či videa, orientaci v prostoru nebo k rozeznání a sledování objektů. Horní kamera zaznamenává obraz přímo před robotem. Druhá, spodní kamera je umístěna tak, aby robotu umožnila rozpoznat překážky či objekty nacházející se v jeho bezprostřední blízkosti, a to zejména u jeho nohou.

Obě kamery mají totožné parametry. Jejich maximální rozlišení činí 1280x960 pixelů, při kterém jsou schopné pořídit záznam o snímkové frekvenci 30 snímků za vteřinu. Horizontální zorné pole kamer je téměř 61 stupňů, vertikální zorné pole pak necelých 48 stupňů.



Obrázek 7 - kamery

3.2.4 LED diody

Na těle robota se nachází celkem 51 LED diod, které mohou sloužit například k vyjádření nálady robota, nachází-li se v autonomním režimu, nebo k signalizaci dokončené uživatelské úlohy. Každá z diod může být individuálně rozsvícena či zhasnuta.

Na hlavě nalezneme 12 LED diod, které jsou umístěny kolem dotykových senzorů. Tyto diody slouží jako vizuální zpětná vazba pro uživatele. Vždy, když dojde ke stisku některého z dotykových senzorů, dojde k rozsvícení příslušných diod kolem stisknutého senzoru.



Obrázek 8 - rozmístění diod – dotykové senzory na hlavě

V každém oku se nachází 8 LED diod, celkem tedy 16 diod. Na rozdíl od diod umístěných kolem dotykových senzorů na hlavě, jsou diody v očích schopné svítit více než jen jednou barvou. Kromě základních barev jako je červená, modrá a zelená, jsou k dispozici předdefinované kombinace těchto tří barev.



Obrázek 9 - rozmístění diod - oči

Kolem každého reproduktoru je v kruhu umístěných 10 LED diod. Stejně jako diody umístěné v očích, jsou diody kolem reproduktorů schopny svítit více než jednou barvou.



Obrázek 10 - rozmístění diod - reproduktory

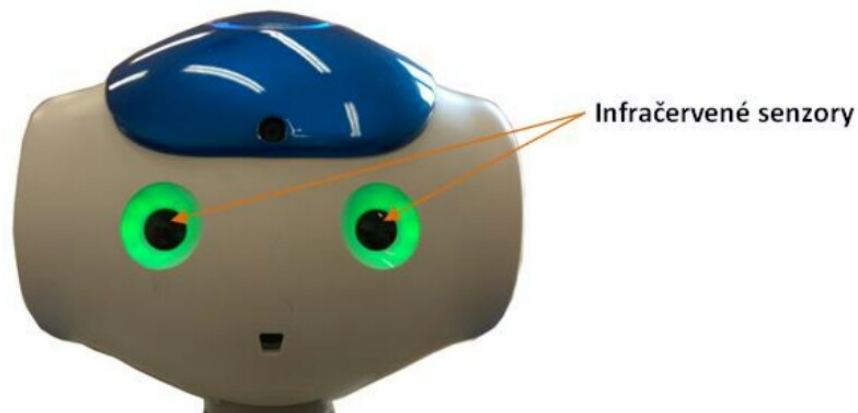
Pod hlavním tlačítkem v hrudi robota se ukrývá jedna vícebarevná dioda signalizující zapnutí robota.

Zbývající dvojice diod se nachází v nohách robota. Tyto diody jsou ukryté v nárttech pod plastovým průsvitným krytem s logem výrobce.

3.2.5 Infračervené senzory

Robot NAO má k dispozici dva infračervené senzory umístěné v očích. Komunikaci pomocí infračervených senzorů zajišťuje modul *ALInfrared*, který poskytuje tři možné způsoby využití. Jedním z nich je použití robota jako dálkové ovládání, další možností je nastavení robota na příjem příkazů z dálkového ovládání a posledním, avšak výrobcem nedoporučeným využitím je vzájemná komunikace několika robotů.

Pro odesílání a přijímání ovládacích kódů nebo komunikaci s ostatními roboty využívá modul *ALInfrared* balíček Linux Infrared Remote Control (LIRC).



Obrázek 11 - infračervené senzory

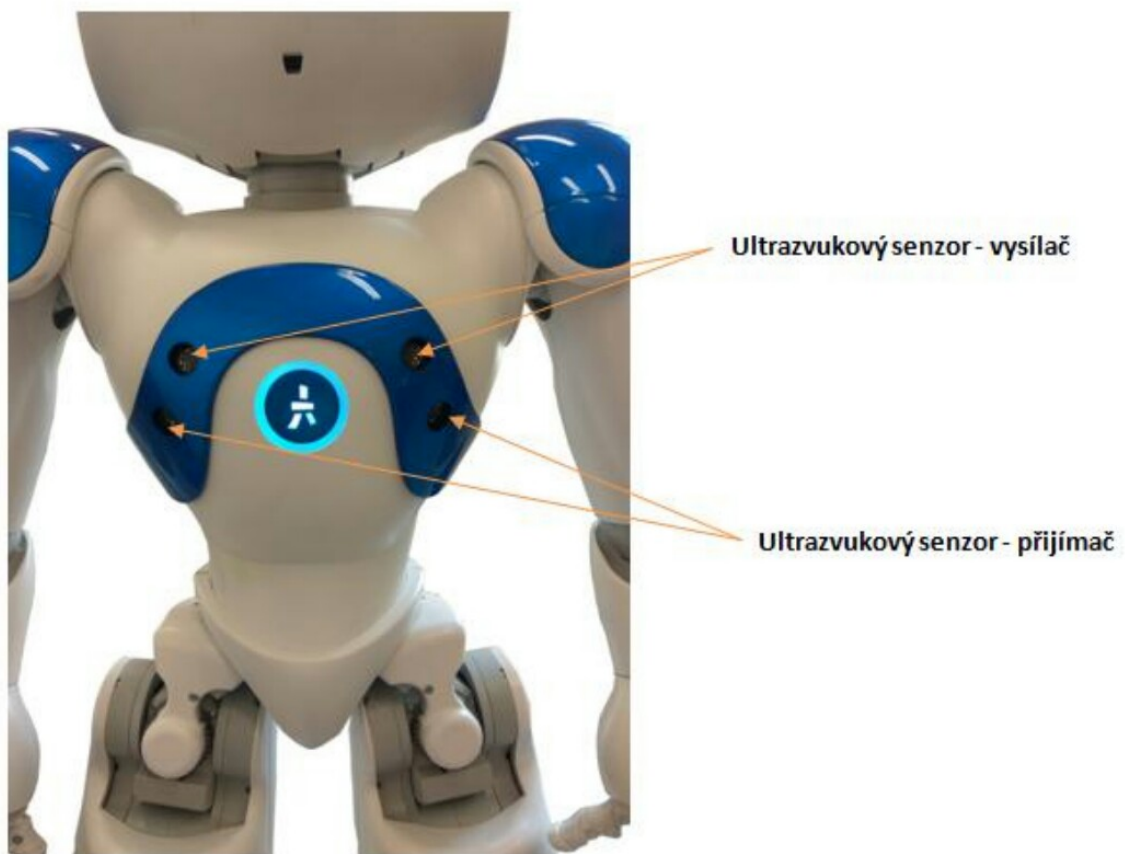
3.2.6 Dotykové senzory

Na těle robota se nachází devět dotykových senzorů, které jsou určeny pro řízení chodu běžícího programu, které jsou umístěny po těle robota. Tři z nich se nachází na hlavě robota, zbylých šest se nachází na rukách robota, tři na každé ruce. Až na senzory umístěné na hlavě, jejichž stisknutí je doprovázeno rozsvícením LED diod nacházejících se po obvodu dotykových plošek, není uživateli poskytnuta žádná zpětná vazba potvrzující stisknutí.

3.2.7 Ultrazvukové senzory

Robot NAO je vybaven dvojicí ultrazvukových senzorů, neboli sonarů, které jsou umístěny v hrudi robota. Ultrazvuk je využíván k odhadu vzdálenosti od překážek nacházejících se v robotově okolí.

Sonary pracují na frekvenci 40 kHz. Jejich rozlišení se pohybuje, v závislosti na vzdálenosti, od 1 do 4 cm. Robot je schopen korektně určit vzdálenost od překážky pokud se překážka nachází ve vzdálenosti 20 až 80 cm. Pro překážky vzdálené méně než 20 cm není robot schopen vzdálenost vypočítat. Vzdálenost přesahující 80 cm je pouhým odhadem.



Obrázek 12 - ultrazvukové senzory

3.3 Softwarové vybavení

V robotu použitý operační systém se nazývá NAOqi OS. Jedná se o speciální linuxovou distribuci založenou na distribuci Gentoo. NAOqi OS je tzv. vestavná (embedded) distribuce vytvořena přímo pro potřeby společnosti Aldebaran. Operační systém poskytuje a spouští mnoho knihoven a programů, které jsou potřebné pro chod robota.

Výrobce poskytuje soubor s virtualizovaným operačním systémem NAOqi OS, který lze spustit pomocí programu VirtualBox od společnosti Oracle. Díky tomu je možné testovat a ladit programy bez nutnosti fyzického přístupu k robotu.

3.4 Implementované rozpoznávací algoritmy

V robotu NAO je výrobcem implementováno několik rozpoznávacích algoritmů, které jsou uživateli plně k dispozici. Díky těmto rozpoznávacím algoritmům je robot schopen rozpoznat a následovat různé objekty. O tuto funkcionalitu se stará modul s názvem *ALTracker*, který nahradil starší moduly pro sledování jednotlivých cílů (*ALRedBallTracker*, *ALFaceTracker*). Jeho cílem je navázání propojení mezi samotným detekováním požadovaného objektu a pohybem pro udržení vizuálního kontaktu s rozpoznaným objektem. Existuje několik způsobů, kterými robot může rozpoznaný objekt následovat.

3.4.1 Možnosti sledování

V režimu *Head* je objekt sledován pouhým otáčením a nakláněním hlavy robota. Tento způsob sledování je výchozím nastavením modulu *ALTracker*.

Uchování vizuálního kontaktu s rozpoznaným objektem je dosaženo pomocí pohybu celého těla robota. V tomto režimu se NAO za objektem otáčí celým svým tělem. Případně může dojít ke změně polohy, ve které se robot momentálně nachází. Například ze stoje do dřepu a podobně. Během sledování objektu robot stabilitu udržuje autonomně.

Režim *Move* je oproti režimu *WholeBody* rozšířen o možnost chůze, kdy se robot snaží dodržovat uživatelem definovanou vzdálenost s objektem. Je nutné zajistit, aby byla cesta k objektu bez překážek, ty robot nedokáže během procesu sledování autonomně detekovat.

3.4.2 Rozpoznatelné objekty

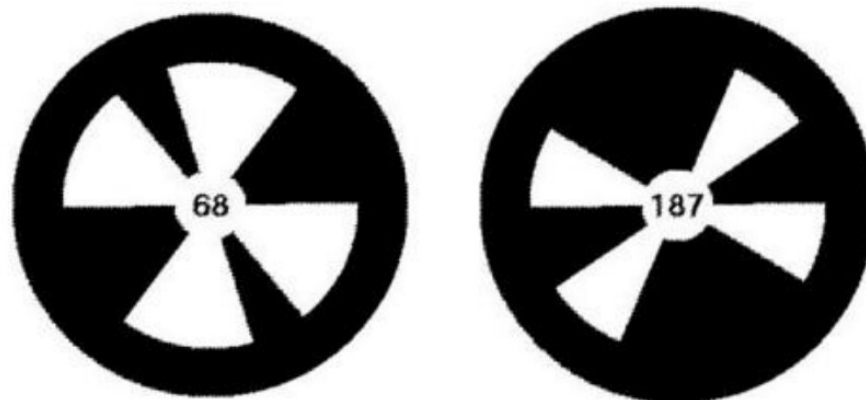
Modul *ALTracker* umožňuje rozpoznat a sledovat různé objekty, které se specifikují pomocí parametru *targetName*. Na základě tohoto parametru pak modul volá jiný modul specifický pro každý cíl.

O rozpoznání červeného míčku se stará modul *ALRedBallDetection*. Princip tohoto modulu je založen na detekci červených pixelů z obrázku pořízeného kamerou robota. Tyto body jsou filtrovány v závislosti na jejich barevné odlišnosti od červené barvy v barevném modelu YUV. Filtr používá vypočtenou prahovou hodnotu pro umožnění detekce i při proměnlivých světelných podmínkách. Ze všech detekovaných červených pixelů jsou zachovány pouze ty, které tvoří kruhový tvar. Pokud je nalezena množina pixelů splňující tuto podmínku, dojde ke změně hodnoty *redBallDetected* v paměti robota. Robot by měl být chopen detekovat červené míčky o průměru v rozmezí od 1 cm do 1 m. V praxi byly testovány červené míčky o dvou rozměrech. Menší o průměru 4 centimetry a větší o průměru 6 centimetrů. Větší byl robot schopen rozeznat a sledovat až na vzdálenost 2,5 metru. U menšího se schopnost sledování zkrátila na 1,5 metru. Pro schopnost rozpoznání míčku je velmi důležitá intenzita osvětlení prostoru. Při dobrých světelných podmínkách rozpoznání funguje celkem dobře. Problém také nastane v okamžiku, kdy robot musí rozeznat červený míček na červeném nebo podobně barevném pozadí.

Modul *ALFaceDetection* umožňuje robotu detekci a případné rozpoznání obličeje nacházejícího se před ním. Tento modul pracuje na principu založeném na algoritmu pro detekci a rozpoznání obličejů společnosti OMRON. Kromě samotné detekce obličeje algoritmus poskytuje uživateli různé informace o nalezeném obličejí. Například jeho pozici a řadu úhlových souřadnic hlavních obličejových rysů jako jsou oči, nos či ústa. Aby byl robot schopen rozpoznávat jednotlivé obličeje, je nutné nejprve projít procesem učení. V této fázi robot pořídí pro každý obličej 5 fotografií, z nichž se robot pokusí daný obličej zapamatovat. Pro zlepšení výsledků rozpoznávání je doporučeno

opakovat učení již uložených obličejů. Rozpoznávací schopnosti robota silně souvisí s osvětlením v prostředí, kde se robot momentálně nachází. V dobře osvětleném prostoru je robot NAO schopen rozpoznat a sledovat obličej až na vzdálenost 1,5 metru.

Pro detekci a rozpoznání speciálních značek slouží modul s názvem *ALLandMarkDetection*. Tyto speciální značky se nazývají NAO marky, které se skládají z černého kruhu s bílými trojúhelníkovými lopatkami vycházejícími ze středu kruhu. Specifické umístění trojúhelníkových lopatek slouží k odlišení jednoho NAO marku od ostatních. Modul *ALLandMarkDetection* může sloužit k různým účelům, například rozmístění těchto speciálních značek může v kombinaci s ostatními senzory výrazně pomoci k orientaci robota v operačním prostoru.



Obrázek 13 - NAO mark – převzato z [1]

Sledování osob v bezprostředním okolí robota NAO umožňuje modul *ALPeoplePerception*, který kromě samotného rozpoznání a sledování poskytuje uživateli základní informace o nalezených osobách. Veškeré vizuální informace modul získává z obou kamer umístěných v hlavě robota a případně z 3D senzoru, je-li dostupný. Po detekci jsou veškeré atributy modulem *ALPeoplePerception* nepřetržitě aktualizovány, přičemž veškerá shromážděná data jsou uložena v paměti a přístupná pomocí modulu *ALMemory*. Všechny nově detekované osoby nacházející se v aktuálním video snímku, jsou pokud možno přiřazeny k již detekovaným osobám. Pokud není nalezená žádná shoda, uloží se informace o nových osobách do paměti. Pokud se některá z detekovaných osob dostane mimo zorné pole robota, nedojde k jejímu okamžitému odstranění ze seznamu osob. Dobu, po kterou musí být osoba mimo dohled, aby byla zapomenuta, lze libovolně nastavit. Proces detekce a sledování je velice náročný na výkon. Pro zlepšení celého procesu je možné omezit maximální vzdálenost, do které bude robot osoby detekovat.

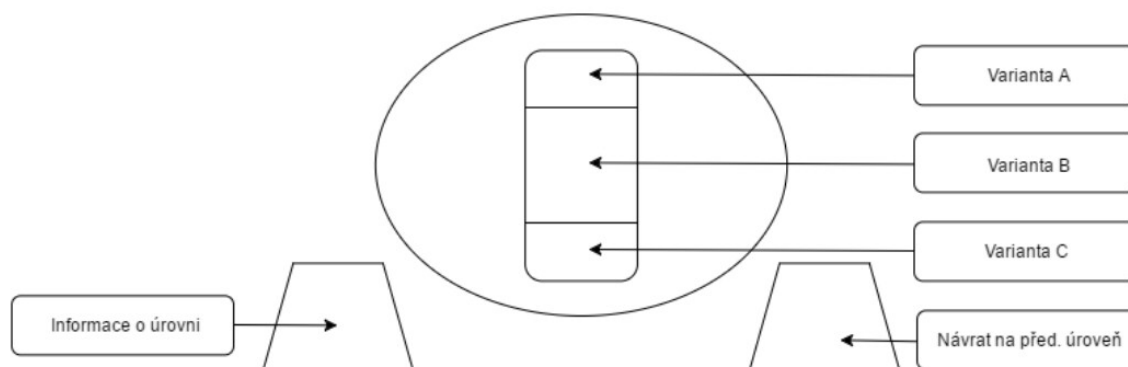
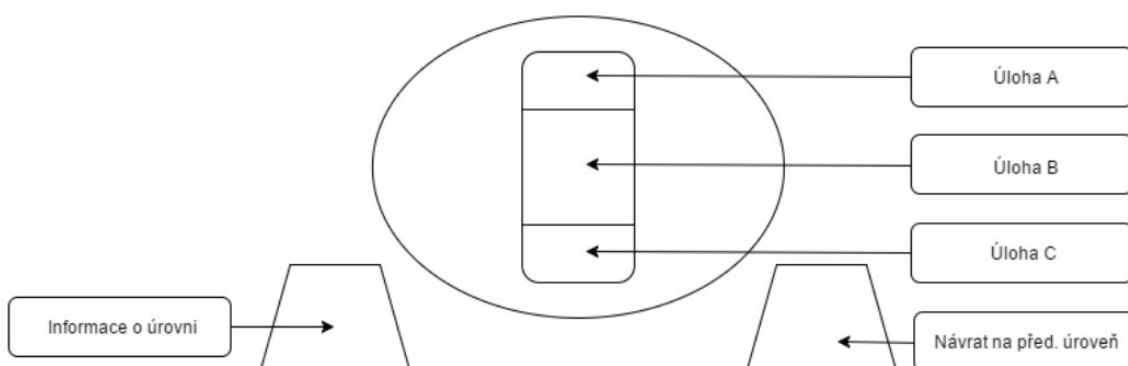
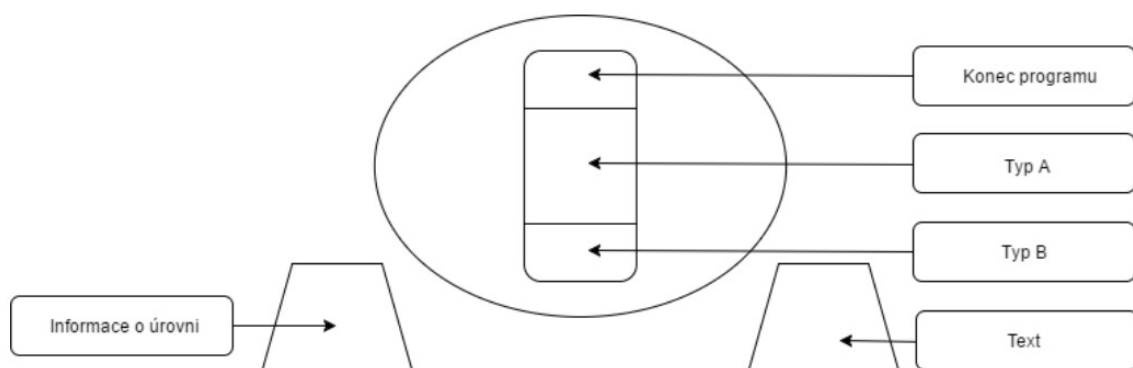
4 Návrh a možnosti realizace softwarového řešení

4.1 Možnosti návrhu

Od začátku se nabízelo několik možných metod ovládání testovací aplikace. Uživatelsky nejintuitivnější se jevila možnost ovládání kompletně skrze grafickou nadstavbu běžící v počítači nebo v chytrém mobilním telefonu či tabletu. Jako další možnost přicházelo v úvahu ovládání pomocí dotykových senzorů na těle robota. Tato možnost by oproti předchozí variantě umožnila přímý kontakt a kontrolu nad robotem. Ovládání přímo na robotu je vhodnější i pro účely prezentace, kdy není zapotřebí odbíhat k řídicímu počítači.

Pro zajištění jednoduchosti ovládání aplikace pro testování a možnost přímého kontaktu s robotem bylo zvoleno ovládání pomocí dotykových senzorů rozmístěných po těle robota. Zajištění zpětné vazby uživateli při každé interakci s dotykovými senzory by mohlo být realizováno hlasovou informací o příslušné volbě.

Samotná logika aplikace by pak byla rozdělena do několika úrovní. Na hlavní úrovni uživatel specifikuje typ požadovaného úkonu. Výběrem činnosti se program dostává do další úrovně, kde uživatel vybere konkrétní úlohu. V poslední úrovni jsou k dispozici různé varianty na, v předchozí úrovni, konkretizovanou úlohu. Mezi těmito úrovněmi bude možné libovolně přecházet za pomoci dotykových senzorů. Pro snadnou orientaci v úroňovém systému programu bude vyhrazen dotykový senzor. Stisknutím senzoru bude uživatel robotem přesně informován o aktuálně zvolené úrovni.



Obrázek 14 - návrh ovládání programu

4.2 Možnosti realizace návrhu

Ve fázi návrhu se, pro zvolený způsob řešení ovládnání aplikace, nabízí několik možností samotné realizace návrhu.

4.2.1 Využití Choregraphe

Jednou z možností byla realizace ve vývojovém prostředí Choregraphe poskytnutém výrobcem robota. Výhodou tohoto řešení je to, že zde není potřeba řídicího počítače. Program je tedy nahrán přímo do paměti robota, ze které je následně spouštěn. Vzhledem k tomu, že zmíněné vývojové prostředí umožňuje vývoj výhradně pomocí předpřipravených programových bloků, veškeré komplexnější úlohy se stávají málo přehlednými. Z tohoto důvodu byla tato možnost zamítnuta.

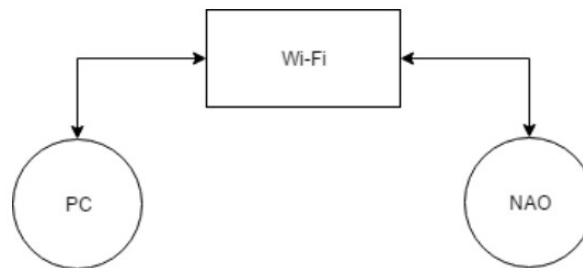
4.2.2 Využití SDK

Jediným v úvahu připadajícím řešením je tedy využití některého z podporovaných jazyků v kombinaci s SDK od výrobce. Pro realizaci byl zvolen programovací jazyk Python, konkrétně ve verzi 2.7. Tato verze sice není nejaktuálnější, ale jako jediná je plně podporována dostupným SDK. V tomto případě je možné testovací program realizovat dvěma způsoby. Obě varianty, na rozdíl od programu realizovaného v Choregraphe, vyžadují dostupnou Wi-Fi síť, ke které budou robot i řídicí počítač připojeny.

V prvním případě by byl soubor s programem uložen v paměti robota. Spuštění skriptu by bylo provedeno z počítače, který by pak k samotnému chodu programu nebyl zapotřebí. Nevýhodou tohoto přístupu je nedostačující výkon hardwarového vybavení robota pro spuštění programu v jazyce Python přímo v robotu.

Druhou možností je spuštění testovacího programu pouze v počítači, který by s robotem navázal spojení a využíval jej jako vstupně-výstupní zařízení. Pro navázání spojení by sloužilo jednoduché okno grafického rozhraní. Prostřednictvím grafického rozhraní aplikace by uživatel zadal IP adresu a port, na kterých bude s robotem komunikovat. IP adresa je uživateli sdělena po stisknutí hlavního tlačítka na hrudi robota. Port je pro komunikaci vždy stejný, a to 9559. Program by kontroloval správnost zadaných údajů. Pokud proběhne vše v pořádku, dojde k navázání spojení. O úspěšném

spojení je uživatel robotem hlasově informován. Dotykové senzory na hlavě a rukách robota slouží k veškerému ovládání programu. Jediným nedostatkem této varianty je nepřetržitá potřeba počítače.



Obrázek 15 - schéma komunikace mezi PC a robotem

4.3 Finální volba

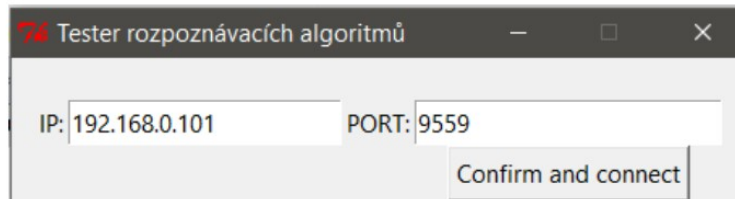
I přes nutnost nepřetržité dostupnosti počítače byla zvolena varianta s programem běžícím v počítači. Toto řešení minimálně zatěžuje hardware robota, díky čemuž není výrazně ovlivněna výdrž baterie.

4.3.1 Realizace grafického rozhraní

Na rozdíl od ostatních jazyků, jako je například Java nebo C#, není pro jazyk Python dostupný žádný funkční grafický editor pro úpravu rozložení grafických prvků. Veškeré přidávání ovládacích prvků a úprava jejich rozložení probíhá úpravou zdrojového kódu, což je z časového hlediska velmi neefektivní.

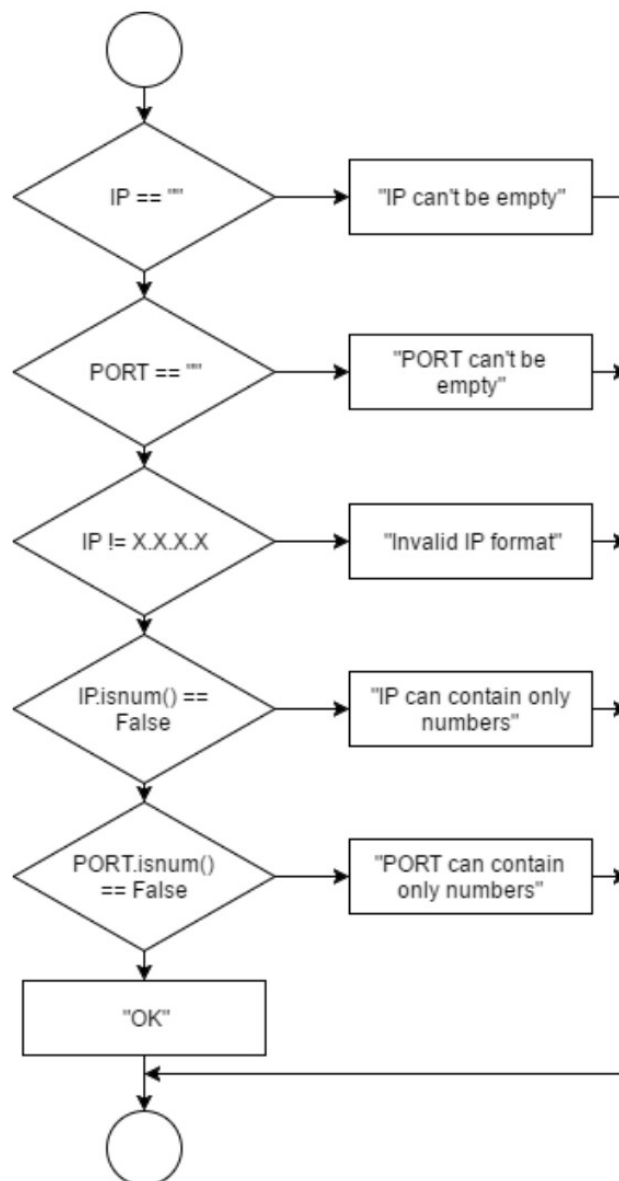
Pro realizaci grafického rozhraní byla zvolena knihovna Tkinter, která je součástí základní instalace jazyku Python. Tkinter není zdaleka jedinou knihovnou pro realizaci grafického uživatelského rozhraní, mezi alternativy patří například knihovna Qt nebo wxPython, které je ovšem potřeba doinstalovat a nejsou tak stabilní jako zvolená knihovna Tkinter.

Pro zadání informací o IP adrese a portu robota slouží dvě textová pole. Před samotným navázáním spojení je nutné zkontrolovat platnost IP adresy a portu.



Obrázek 16 - výsledné grafické rozhraní

Aplikace kromě správné délky adresy také kontroluje, zda IP adresa a port obsahují pouze číslice. V případě chybně zadaných hodnot bude uživateli zobrazena konkrétní chyba ve stavovém řádku aplikace. Pokud budou hodnoty vyhodnoceny jako validní, dojde k samotnému pokusu o navázání spojení s robotem.



Obrázek 17 - vývojový diagram kontroly validity vstupů z GUI

4.3.2 Ovládání robota

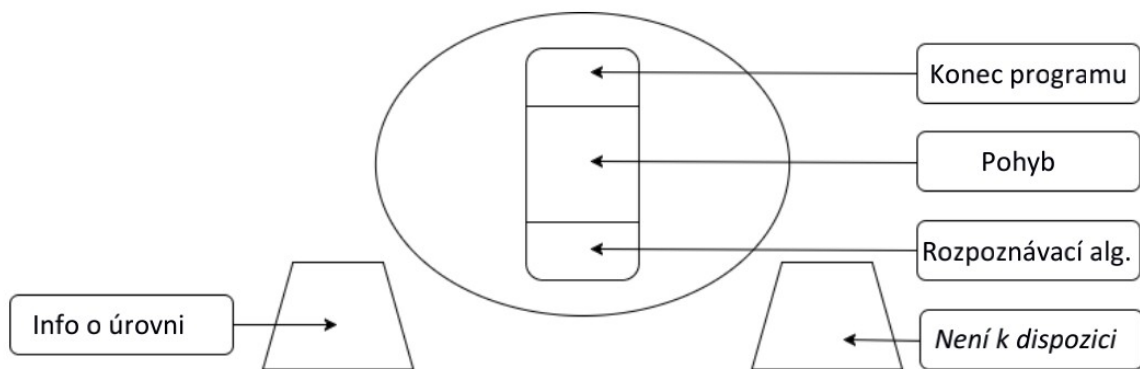
Po úspěšném spojení s robotem aplikace ve smyčce čeká na další interakci uživatele. Jak bylo již zmíněno, veškeré ovládání nyní probíhá prostřednictvím dotykových senzorů na těle robota. Trojice senzorů umístěných na hlavě slouží k volbě jednotlivých režimů. Senzor na levé ruce je určen pro návrat na předchozí úroveň. Pro zjištění aktuální úrovně slouží senzor na pravé ruce robota.

5 Funkce realizovaného programu

V této kapitole jsou podrobně popsány všechny úrovně programu, včetně jednotlivých dílčích variant pro testování samostatných algoritmů.

5.1 Výběr režimu

První úroveň programu umožňuje výběr mezi dvěma režimy a ukončení programu. První režim je zaměřený na testování rozpoznávacích algoritmů. Do tohoto režimu se uživatel dostane stisknutím prvního senzoru od předu na hlavě. Druhý režim slouží k provádění základních pohybových úkonů, v případě, že je nutné změnit pozici, ve které se robot aktuálně nachází. Tento režim je přístupný stisknutím prostředního senzoru na hlavě robota. Poslední sensor na hlavě slouží k ukončení aplikace.



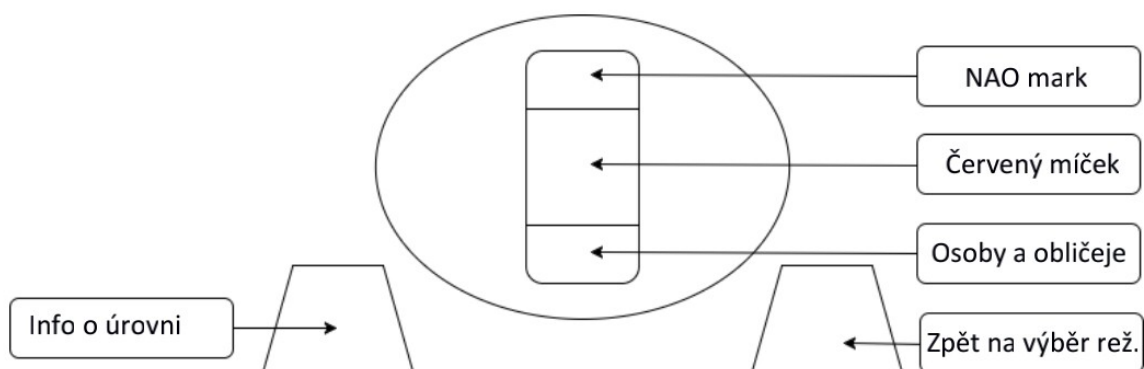
Obrázek 18 - ovládání programu - výběr režimu

5.2 Výběr typu úlohy

Na druhé úrovni jsou uživateli k dispozici konkrétní typy úloh v závislosti na režimu zvoleném v předchozí úrovni. Pro návrat na výběr režimu slouží sensor na levé ruce robota.

5.2.1 Rozpoznání objektů

Na druhé úrovni, v režimu rozpoznání objektů, je uživateli umožněno vybrat si sadu příkladů pro konkrétní objekt, který chce, aby robot rozpoznal.

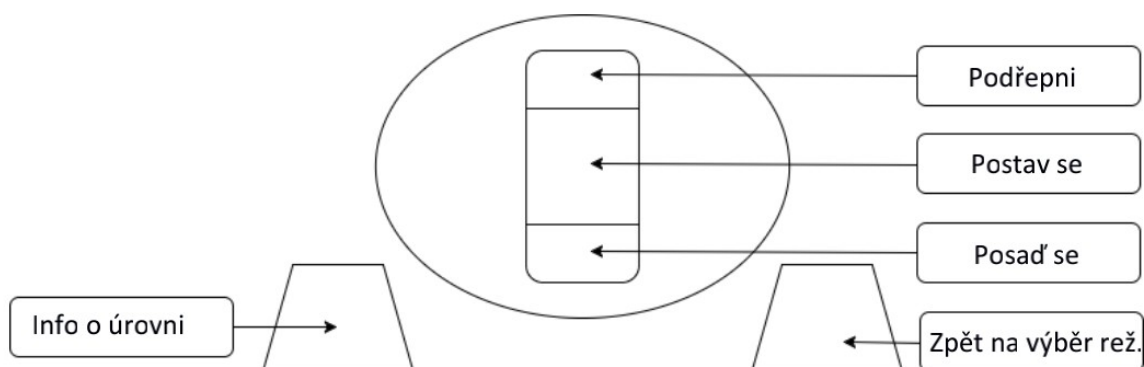


Obrázek 19 - ovládání programu - výběr objektu

K dispozici jsou tři sady menších testovacích aplikací, na každý rozpoznatelný objekt je jedna sada. První sada je zaměřená na testování rozpoznání osob a obličejů. Druhá sada je určená pro rozpoznání a sledování červených míčků. Poslední ze tří dostupných sad je sada testující rozpoznávací schopnosti u speciálních značek, neboli NAO marků.

5.2.2 Pohyb

Druhá úroveň pohybového režimu umožňuje změnit pozici robota do tří základních pozic. Prvním pokynem je pokyn „Posaď se“, kdy se, jak již název napovídá, robot posadí. Druhým příkazem je „Postav se“ kdy se robot postaví. Poslední pozicí je podřep, ve kterém robot zabírá nejméně místa.



Obrázek 20 - ovládání programu - výběr pozice

5.3 Výběr varianty úlohy

Třetí úroveň výběru je k dispozici pouze pro režim rozpoznávacích algoritmů. V této úrovni uživatel volí konkrétní variantu algoritmu pro objekt vybraný ve druhé úrovni programu.

5.3.1 Rozpoznání osob a obličejů

Uživateli jsou k dispozici tři podprogramy pro testování algoritmu pro rozpoznání osob a obličejů. První podprogram počítá uživatele nacházející se před robotem. Druhý podprogram umožňuje sledovat jeden obličej otáčením hlavy. Poslední podprogram pro sledování jednoho obličeje využívá celé tělo robota.

První podprogram je zaměřený na schopnost rozpoznání více obličejů současně. Dokud se před robotem NAO nachází jeden nebo více obličejů, algoritmus se snaží o jejich rozpoznání a následně pomocí syntetizátoru řeči sdělí přesný počet před ním se nacházejících osob. Pokud robot nenalezne žádný obličej, je uživatel informován. Rozpoznávání je prováděno ve smyčce, dokud není stisknut přední senzor na hlavě robota.

Další podprogram se zaměřuje na rozpoznání pouze jednoho obličeje, který je pak otáčením hlavy sledován. Pro ukončení sledování a podprogramu samotného je nutné, aby uživatel stiskl přední senzor na robotově hlavě.

Ve třetím podprogramu, stejně jako v předchozím případě, kdy byl obličej sledován pouhým otáčením hlavy, je robot zaměřen pouze na jeden obličej, který je tentokrát sledován celým tělem. Robot se tedy za rozpoznaným obličejem otáčí celým svým tělem a v případě potřeby může změnit polohu, ve které se momentálně nachází. Ukončení sledování a návrat na výběr testovacích variant je provedeno opět stisknutím předního senzoru na hlavě robota.

5.3.2 Rozpoznání červeného míčku

Obdobně jako u rozpoznávání obličejů jsou také pro testování rozpoznávání červeného míčku k dispozici tři jednoduché podprogramy. Účelem prvního podprogramu je prosté sledování míčku hlavou. Druhý podprogram kromě sledování hlavou využívá i sledování levou rukou robota. V rámci posledního podprogramu je červený míček sledován chůzí robota.

Princip prvního podprogramu je velmi podobný variantě na sledování obličejů hlavou s tím rozdílem, že místo obličejů je sledován červený míček o průměru 60 milimetrů. Stejně jako ve všech předchozích případech, je ukončení podprogramu umožněno stisknutím předního senzoru na hlavě.

Ve druhém podprogramu je pouhé sledování hlavou rozšířeno o levou ruku robota, což by efektor pohybu. Pozice míčku je do značné míry kopírována pohyby roboty levé ruky. V tomto případě je sledování nastaveno na menší, tedy 40 milimetrový, červený míček. Pokud uživatel vloží míček do dlaně robota a stiskne senzor na levé ruce, robot poděkuje a míček uchopí. Po uchopení dojde k pozastavení sledovacího algoritmu. Pro opětovné spuštění a uvolnění míčku je třeba stisknout přední senzor na hlavě. Ukončení podprogramu je pro změnu provedeno stisknutím zadního senzoru na hlavě robota.

Poslední variantou aplikace algoritmu je chůze za červeným míčkem. Tato varianta je určena pro 60 milimetrový míček. Testováním v praxi bylo ověřeno, že funkci lze použít i pro menší, 40 milimetrový míček. Ukončení sledování se provede stisknutím předního senzoru na hlavě robota.

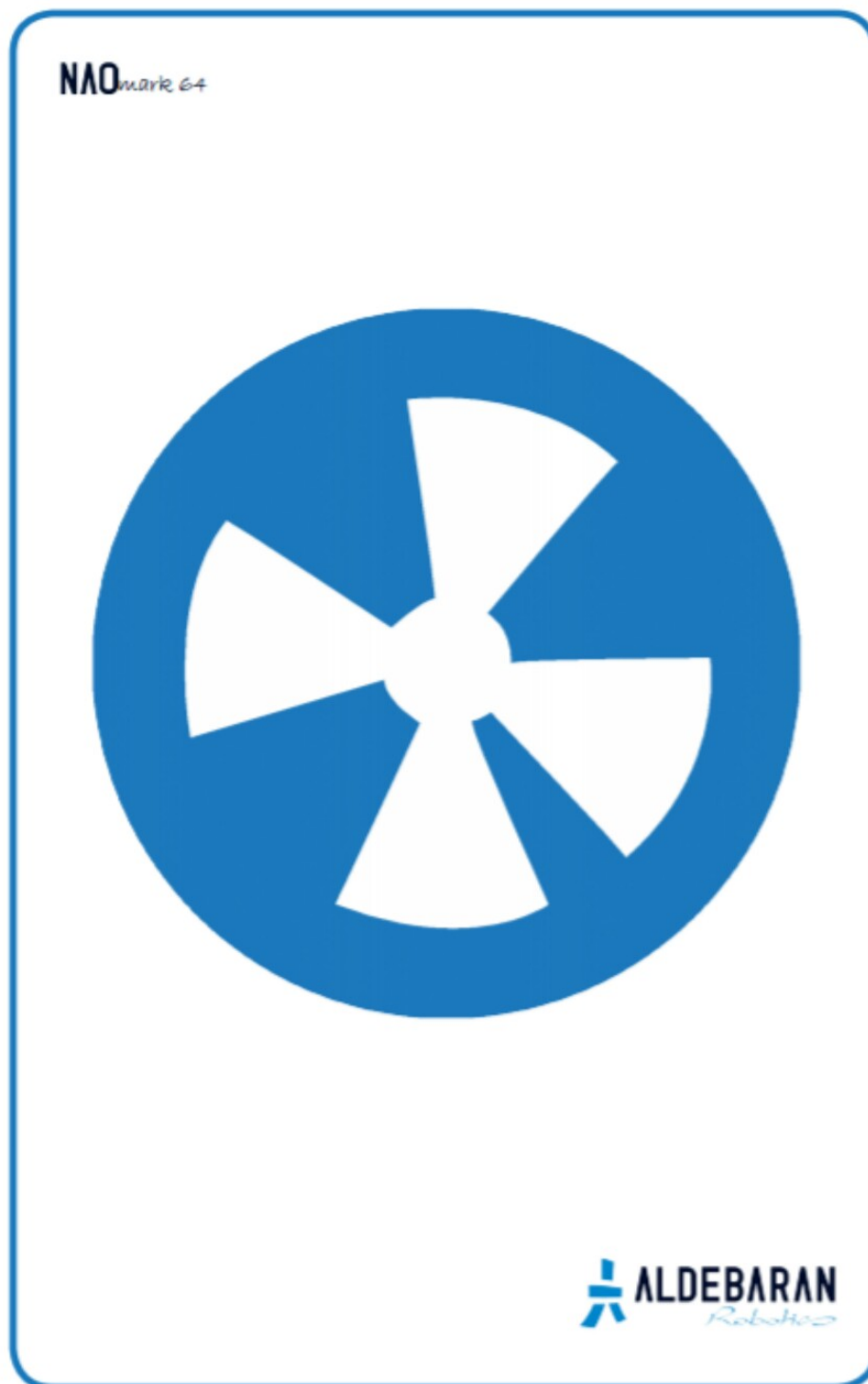
5.3.3 Rozpoznání NAO marků

Pro testování rozpoznávání speciálních NAO marků jsou taktéž k dispozici tři podprogramy. Výsledkem prvního podprogramu je informace o jedinečném čísle viditelných NAO marků. Druhý podprogram slouží ke sledování speciálních značek celým tělem robota. V posledním, třetím, podprogramu jsou speciální značky sledovány pomocí chůze.

Díky prvnímu podprogramu dokáže robot rozeznat jeden a více NAO marků. Pro každý detekovaný pak uživateli pomocí řeči sdělí unikátní ID. V praxi byly otestovány dva NAO marky, první s číslem 64 a druhý s číslem 68. Úspěšnost rozpoznání, je podmíněna nehybností značek. V případě pohybu by měl být pohyb pomalý. S rostoucí rychlostí pohybu úspěšnost rozpoznání klesá. Rozpoznávání je opět ukončeno stisknutím předního senzoru na hlavě robota.

Druhý ze tří podprogramů umožňuje sledování pouze jednoho konkrétního NAO marku. Robot se soustředí na NAO mark s číslem 64. Stejně jako u jiných objektů je konkrétní značka sledována celým tělem robota. Robot je schopen se za značkou otáčet a případně přizpůsobit pozici svého těla.

Ve třetím podprogramu je, stejně jako v předchozím, robot zaměřen na konkrétní značku, opět s číslem 64. Obdobně jako u chůze za červeným míčkem si robot od značky udržuje nastavenou vzdálenost.



Obrázek 21 - NAO mark (64) použitý pro testování - převzato z [1]

6 Závěr

V rámci bakalářské práce byla navržena a realizována aplikace určená pro testování možností rozpoznávacích algoritmů implementovaných na senzorický systém robotu NAO. Samotná aplikace je napsána v programovacím jazyce Python. Ovládání aplikace je členěné do tří úrovní. V první úrovni uživatel volí, zda chce testovat jednotlivé rozpoznávací algoritmy, nebo zda chce změnit pozici, ve které se robot nachází. Režim testování rozpoznávacích algoritmů je rozdělen do tří kategorií, kdy každá kategorie reprezentuje jeden rozpoznatelný objekt. V poslední úrovni se pro každý objekt nachází trojce podprogramů upřesňující režim testování zvoleného algoritmu. Veškerá navigace programem probíhá skrze dotykové senzory na těle robota.

Do budoucna by mohla být práce rozšířena o komplexnější a složitější úlohy zaměřené na testování rozpoznávacích algoritmů. Jako například vyhledávání míčků v prostoru apod.

7 Seznam použité literatury

- [1] NAO Documentation. *Aldebaran Documentation* [online]. [cit. 2017-05-02].
Dostupné z: http://doc.aldebaran.com/2-1/home_nao.html
- [2] An Introduction To Tkinter [online]. 2005 [cit. 2017-05-08]. Dostupné z:
<http://effbot.org/tkinterbook/tkinter-index.htm>
- [3] NOVÁK, Petr. *Mobilní roboty: pohony, senzory, řízení*. Praha: BEN - technická literatura, 2005. ISBN 80-7300-141-1.
- [4] By Hans Haase - Own work, CC BY-SA 4.0,
<https://commons.wikimedia.org/w/index.php?curid=47753356>
- [5] By Amir shahram Hematian - LPR device, Public Domain,
<https://commons.wikimedia.org/w/index.php?curid=11780808>
- [6] Wiskott, Laurenz, et al. "Face recognition by elastic bunch graph matching." *IEEE Transactions on pattern analysis and machine intelligence* 19.7 (1997): 775-779.

8 Přílohy

Příloha A – přiložené CD obsahuje

- **Bakalářskou práci – ve formátu .pdf**
- **Zdrojový kód programu – ve formátu .py**
- **Spustitelnou verzi programu – ve formátu .exe**
- **Vývojové diagramy a fotografie – ve formátu .png**

Příloha B – zdrojový kód grafického rozhraní

```
1. def gui():
2.     global ip
3.     global port
4.     global status
5.
6.     # Vytvoření hlavního dialogového okna
7.     form = Tk()
8.     # Nastavení titulku okna
9.     form.title("Tester rozpoznávacích algoritmů")
10.
11.    # Popisek a textbox pro IP
12.    ip = StringVar(value="192.168.0.101") # "kontejner" pro uchování obsahu textboxu
13.    ip_label = Label(form, text="IP:")
14.    ip_entry = Entry(form, width=0, textvariable=ip)
15.
16.    # Popisek a textbox pro PORT
17.    port = StringVar(value="9559") # "kontejner" pro uchování obsahu textboxu
18.    port_label = Label(form, text="PORT:")
19.    port_entry = Entry(form, width=0, textvariable=port)
20.
21.    # Potvrzovací tlačítko
22.    confirm_button = Button(master=form, text="Confirm and connect",
23.                            command=process_input)
24.
25.    # Informační řádek
26.    status = StringVar()
27.    status_label = Label(form, text="", textvariable=status)
28.
29.    # Umístění grafických prvků do mřížky
30.    confirm_button.grid(in_=form, column=5, row=3, columnspan=1,
31.                       padx=0, pady=0, padx=0, pady=0, rowspan=1, sticky="")
32.    ip_label.grid(in_=form, column=2, row=2, columnspan=1,
33.                padx=0, pady=0, padx=0, pady=0, rowspan=1, sticky="nsew")
34.    port_label.grid(in_=form, column=4, row=2, columnspan=1,
35.                  padx=0, pady=0, padx=0, pady=0, rowspan=1, sticky="nsew")
36.    ip_entry.grid(in_=form, column=3, row=2, columnspan=1,
37.                padx=0, pady=0, padx=0, pady=0, rowspan=1, sticky="nsew")
38.    port_entry.grid(in_=form, column=5, row=2, columnspan=1,
39.                   padx=0, pady=0, padx=0, pady=0, rowspan=1, sticky="nsew")
40.    status_label.grid(in_=form, column=1, row=1, columnspan=6,
41.                    padx=0, pady=0, padx=0, pady=0, rowspan=1, sticky="nsew")
42.
43.    # Nastavení rozměrů jednotlivých buněk mřížky
44.    form.grid_rowconfigure(1, weight=0, minsize=17, pad=0)
45.    form.grid_rowconfigure(2, weight=0, minsize=28, pad=0)
46.    form.grid_rowconfigure(3, weight=0, minsize=34, pad=0)
47.    form.grid_columnconfigure(1, weight=0, minsize=14, pad=0)
48.    form.grid_columnconfigure(2, weight=0, minsize=20, pad=0)
49.    form.grid_columnconfigure(3, weight=0, minsize=169, pad=0)
50.    form.grid_columnconfigure(4, weight=0, minsize=39, pad=0)
51.    form.grid_columnconfigure(5, weight=0, minsize=190, pad=0)
52.    form.grid_columnconfigure(6, weight=0, minsize=16, pad=0)
53.    form.resizable(0, 0)
54.
55.    form.mainloop()
```