

**SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE  
FAKULTA ELEKTROTECHNIKY A INFORMATIKY**

**ZADANIE ČÍSLO 1 Z PREDMETU  
SEMINÁRNA PRÁCA**

**SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE**  
**FAKULTA ELEKTROTECHNIKY A INFORMATIKY**

**ZADANIE ČÍSLO 1 Z PREDMETU**  
**SEMINÁRNA PRÁCA**

Študijný program:	Aplikovaná informatika
Predmet:	I-HMI – Human mashine interface HMI
Prednášajúci:	Ing. Martin Dekan, PhD.
Cvičiaci:	Ing. Michal Dobiš, PhD.

**Bratislava 2024**

**Bc. Filip Lobpreis, Bc. Maroš Kocúr**

# Obsah

<b>Úvod</b>	<b>1</b>
<b>1 Užívateľské okno</b>	<b>2</b>
1.1 Pôvodný dizajn . . . . .	2
1.2 Nový dizajn . . . . .	3
<b>2 Lidar Kamera fúzia</b>	<b>6</b>
2.1 Problematika . . . . .	6
2.2 Riešenie . . . . .	6
2.3 Kamerová časť . . . . .	6
2.4 Parkovacia kamera . . . . .	7
<b>3 Ovládanie gestami</b>	<b>8</b>
3.1 Navádzanie na body . . . . .	11
<b>Záver</b>	<b>13</b>
<b>Zoznam použitej literatúry</b>	<b>14</b>
<b>Prílohy</b>	<b>14</b>
<b>A Štruktúra elektronického nosiča</b>	<b>15</b>
<b>B Algoritmus</b>	<b>16</b>
<b>C Výpis subline</b>	<b>17</b>

# Zoznam obrázkov a tabuliek

Obrázok 1.1	Povodni výzor aplikácie a ovládanie robota. . . . .	2
Obrázok 1.2	Nove dizajny aplikácie zobrazujú základný (a), kávový (b) a mokka (c) mód. . . . .	3
Obrázok 1.3	Ledka zobrazujúca stav pripojenia k robotu (a), odpojenia od robota (b) a núdzového zastavenia (c). . . . .	4
Obrázok 1.4	. . . . .	4
Obrázok 1.5	Dynamické zobrazenie tlačidiel pre ovládanie robota na dotykovom displeji. . . . .	5
Obrázok 1.6	Pomoc pre užívateľa na zorientovanie sa v používaní aplikácie. .	6
Obrázok 2.1	Alarmy použité na upozornenie operátora o objektoch v blízkosti robota . . . . .	7
Obrázok 2.2	Parkovací asistent . . . . .	8
Obrázok 3.1	Kĺby ľudského tela . . . . .	9
Obrázok 3.2	Relatívne zobrazenie polohy rúk na ovládanie robota. . . . .	10
Obrázok 3.3	Zobrazenie mapy a bodov na mape. . . . .	11



# Zoznam algoritmov

B.1	Vypočítaj $y = x^n$	16
-----	---------------------	----

# Zoznam výpisov

C.1 Ukážka sublime-project . . . . .	17
--------------------------------------	----

# Úvod

V prvom zadaní je našou úlohou vytvoriť aplikáciu, ktorá bude zobrazovať aktuálny stav robota pomocou tele riadenia a popri tom bude spĺňať základné normy pre HMI. Tato aplikácia bude slúžiť ako nástroj pre ovládanie robota a zobrazovanie jeho stavu.

Toto zadanie sa bude zaoberať tromi hlavnými problematikami:

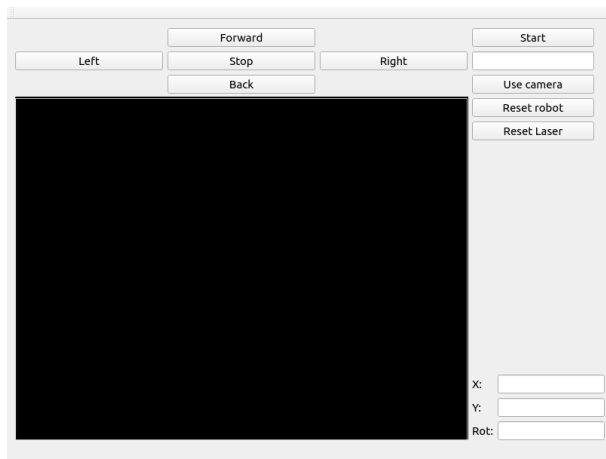
- Návrh aplikácie - v tejto časti sa budeme venovať návrhu aplikácie, respektíve opraveniu štruktúry už existujúcemu grafickému rozhraniu.
- Fúzia kamery a lidar - druhá sekcia sa zaoberá fúziou dát z kamery a lidar.
- Vizuálna detekcia - posledná sekcia sa zaoberá tele riadením robota.



# 1 Uživatelské okno

## 1.1 Pôvodný dizajn

V prvej časti prvého zadania bolo našou úlohou upraviť užívateľské rozhranie, tak aby spĺňalo normy zaoberajúce sa rozhraním človeka a stroja (HMI). Na začiatku sme začínali s existujúcim rozhraním zobrazeným na obrázku 1.1.



Obr. 1.1: Povodni výzor aplikácie a ovládanie robota.

Tento dizajn ma niekoľko nedostatkov, ktoré sme mali za úlohu identifikovať a opraviť. Prvým problémom je, že tlačidlá pre ovládanie robota sa nachádzajú nad zobrazovacou plochou. Toto umiestnenie je nevhodné, pretože blokuje užívateľovi vo výhlade na robota. Druhým problémom je, že tlačidlá pre ovládanie robota sú príliš malé a neprehľadné. Tretím problémom je, že tlačidlá pre ovládanie robota sú príliš blízko k sebe, čo môže spôsobiť neúmyselné stlačenie viacerých tlačidiel naraz.

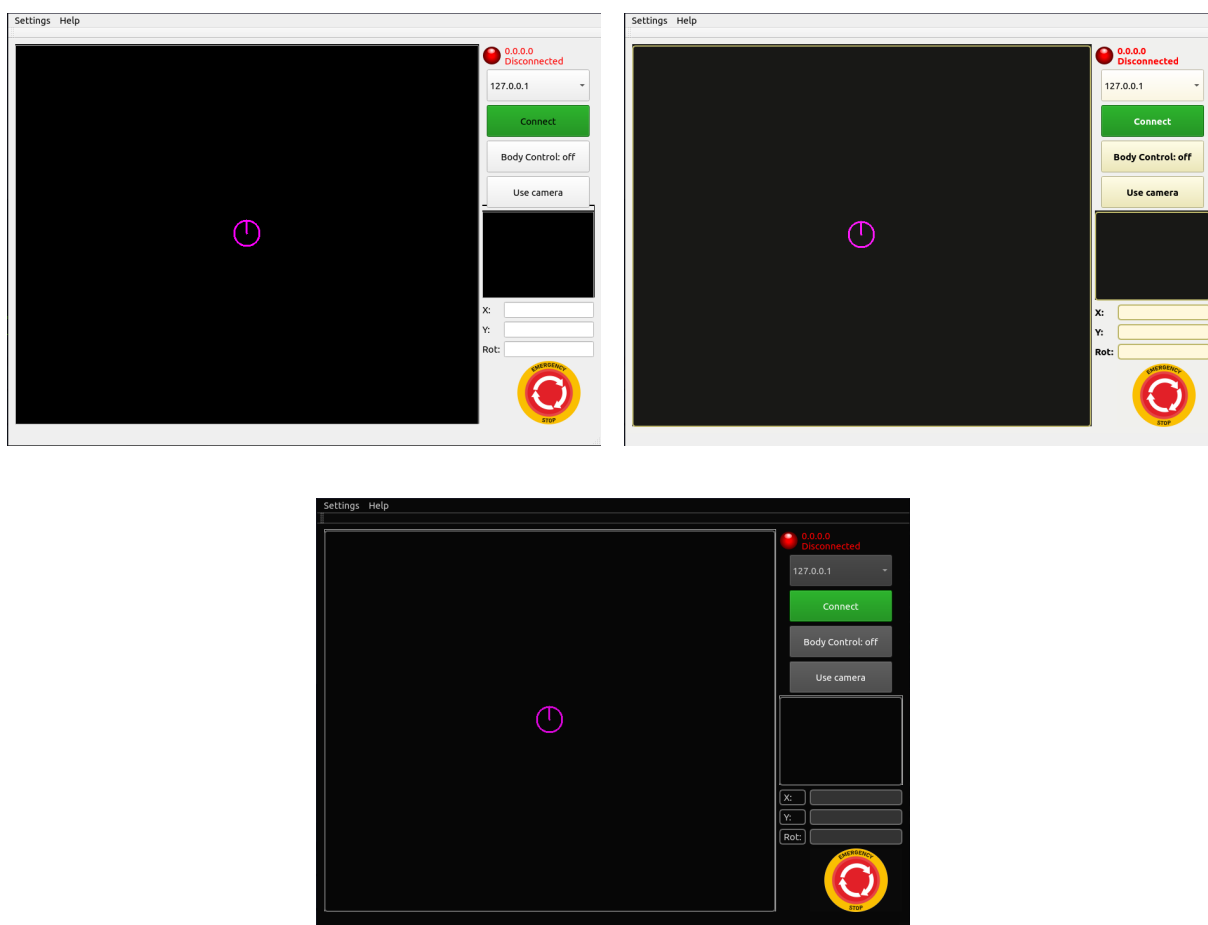
Ďalšie chybné prvky sa nachádzajú v samotnej funkcionalite aplikácie. Užívateľ sa vie pripojiť len na jedného robota a to toho ktorý je v simulácii. Na pripojenie robota, ktorý je v reálnom svete je potrebné zmeniť IP adresu robota v kóde a ten následne prekompilovať. Ak sa tlačidlo **Start** stlačí viackrát ako raz, tak aplikácia spadne. Tlačidlá **Reset robot** a **Reset Laser** sú nepotrebné, pretože nemajú žiadnu funkcionalitu.

Následne poloha zobrazenia súradníc robota je nevhodná, pretože na ňu užívateľ nebude vidieť, keďže sú tlačidlá na ovládanie robota lokalizované nad obrazovkou zobrazujúcou dáta z lidarů alebo kamery.

Nedostatky tejto aplikácie sú ďalej len samotná implementácia zobrazenia dát z lidarů a kamery. Tieto dáta nie sú spracované na vzájomné prepojenie.

## 1.2 Nový dizajn

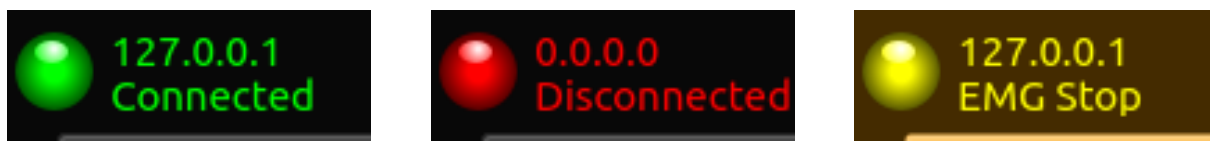
Na základe identifikovaných problémov sme navrhli nový dizajn aplikácie. Tento dizajn je zobrazený na obrázkoch 1.2. Na obrázku 1.2 a je zobrazený základný dizajn aplikácie. Tento dizajn sa dá zmeniť cez nastavenia *Settings > Change style sheet*. Tlačidlá pre ovládanie robota boli v základnej verzii vymazane. Robot sa ovláda pomocou klávesnice. A to klávesami W, A, S, D alebo šípkami. Pre zastavenie robota je potrebné stlačiť klávesu R. Tato klávesa bola zvolená preto, aby sa zabránilo nežiadanému tlačeniu klávesy a tým zastavením robota. Ďalej bol pridaný prvok na zmenu IP adresy robota.



Obr. 1.2: Nove dizajny aplikácie zobrazujú základný (a), kávový (b) a mokka (c) mód.

Ako ďalšie prvky bol pridaný identifikátor stavu robota, ktorý zobrazuje či je robot pripojený, odpojený alebo v núdzovom zastavení. Tieto stavy sú zobrazené na obrázkoch 1.3

Ďalší zmenený prvok je farebné zobrazenie tlačidla pripojenia a odpojenia robota. Keďže sú červená a zelená zlé farby na kombináciu, bola zvolená aj zmena textu zobrazeného na tlačidle. Pri pripojení robota je zobrazený text **Connect** a pri odpojení robota

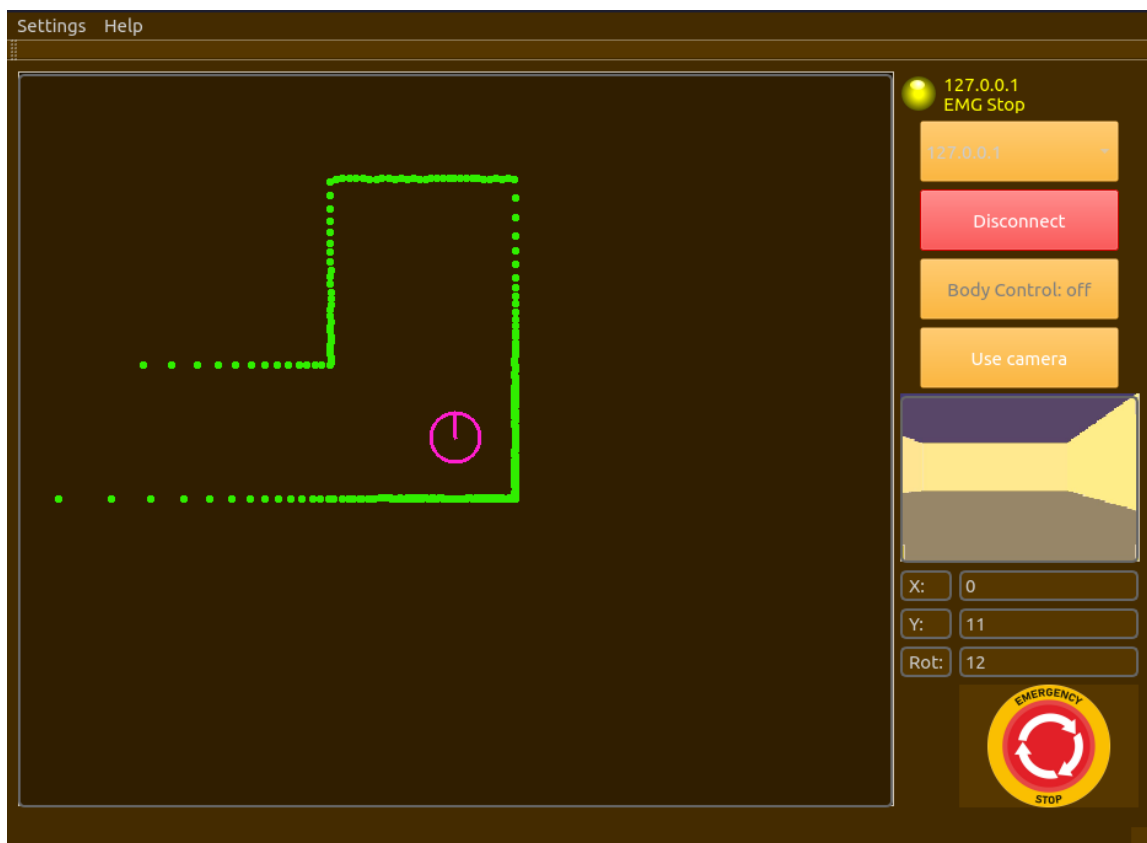


Obr. 1.3: Ledka zobrazujúca stav pripojenia k robotu (a), odpojenia od robota (b) a núdzového zastavenia (c).

je zobrazený text **Disconnect**.

Taktiež bolo pridané softvérové tlačidlo pre núdzové zastavenie robota. Toto tlačidlo podľa normy HMI musí byť umiestnené na viditeľnom mieste a musí byť veľké. Taktiež nemôže byť softvérové ale inú možnosť sme v tomto prípade nemali (a taktiež to bolo v zadaní).

Pri stlačení tohto núdzového tlačidla sa zablokuje všetky vstupy z tlačidiel, klávesnice, a kamery. Jedinými povolenými funkciami sú prepnutie medzi kamerou a lidarom a zrušenie núdzového stavu. Zobrazenie núdzového zastavenia je zobrazený na obrázku 1.4

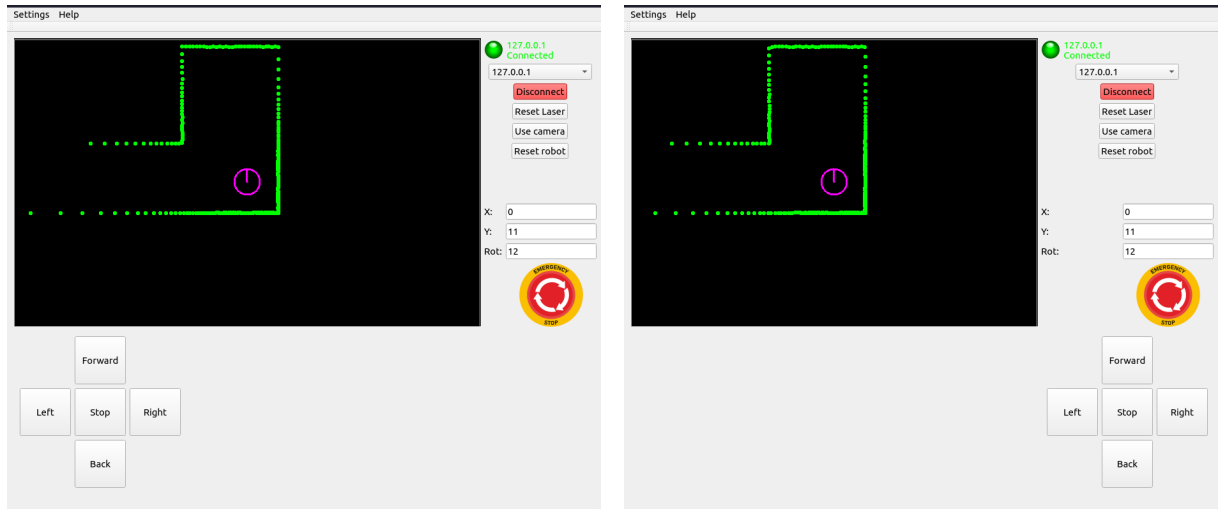


Obr. 1.4

Na obrázku 1.4 je zobrazené už aj zobrazenie dát z lidarom. Taktiež môžeme jedno-

značne identifikovať nezablokované tlačidlá.

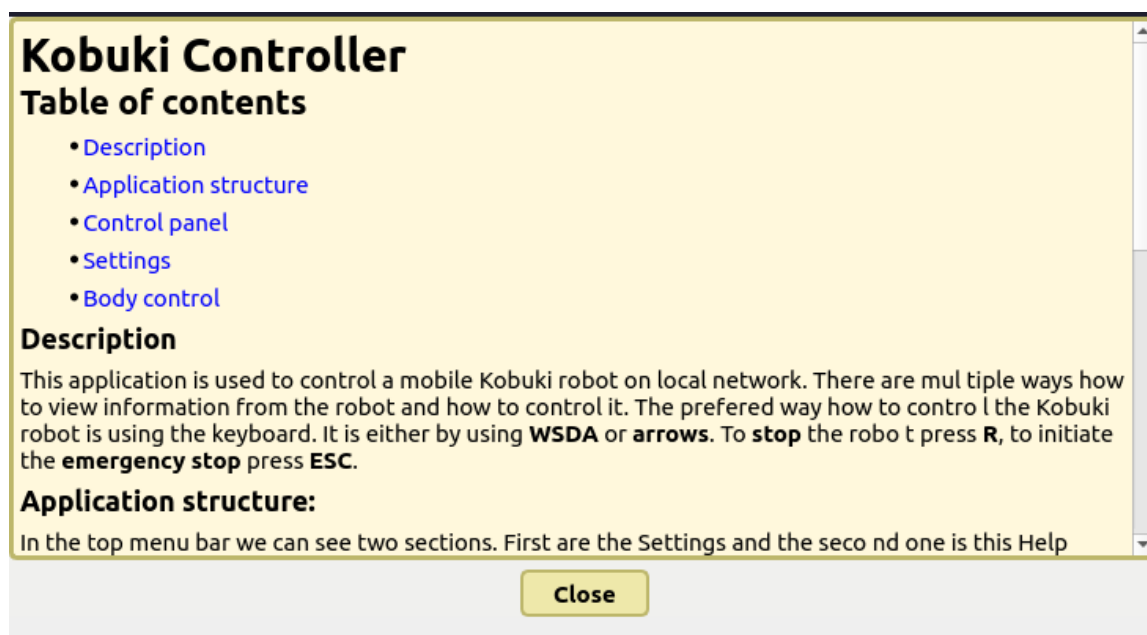
Ak by užívateľ používal túto aplikáciu na dotykovom displeji a chcel by ovládať robot pomocou tlačidiel, tak si môže zobraziť tlačidlá pre ovládanie robota. Tieto tlačidlá sú zobrazené na obrázku 1.5



Obr. 1.5: Dynamické zobrazenie tlačidiel pre ovládanie robota na dotykovom displeji.

Ako vidíme na obrázku 1.5 tak sa tlačidlá pre ovládanie robota zobrazujú len vtedy keď je v nastavení zvolený mód s tlačidlami. Taktiež si užívateľ môže zvoliť, či chce mať tlačidlá na ľavej alebo pravej strane obrazovky. Tieto tlačidlá sú dostatočne veľké na to, aby ich užívateľ mohol ovládať aj na dotykovom displeji.

Pre jednoduchosť používania aplikácie a plochej krivke učenia sme pridali možnosť zobrazenia pomoci. Tato pomoc je zobrazená na obrázku 1.6. V tejto nápovede sme zvýraznili tie najdôležitejšie prvky, aby si ich užívateľ vedel nájsť a vedel ich používať v čo najkratšom čase.



Obr. 1.6: Pomoc pre užívateľa na zorientovanie sa v používaní aplikácie.

## 2 Lidar Kamera fúzia

### 2.1 Problematika

V tejto úlohe bolo potrebné vizualizovať na kamere dáta z lidar, primárne tie ktoré niesu viditeľné kamerou, aby sme upozornili operátora na možné kolízie a prekážky ktoré sa nachádzajú v pracovnom prostredí robota. Taktiež sme implementovali parkovaciu kameru, ktorá sa zapne pri cúvaní s robotom, následne pri zmene smeru dopredu sa opäť zapne pôvodné okno, ktoré mal používateľ zapnuté.

### 2.2 Riešenie

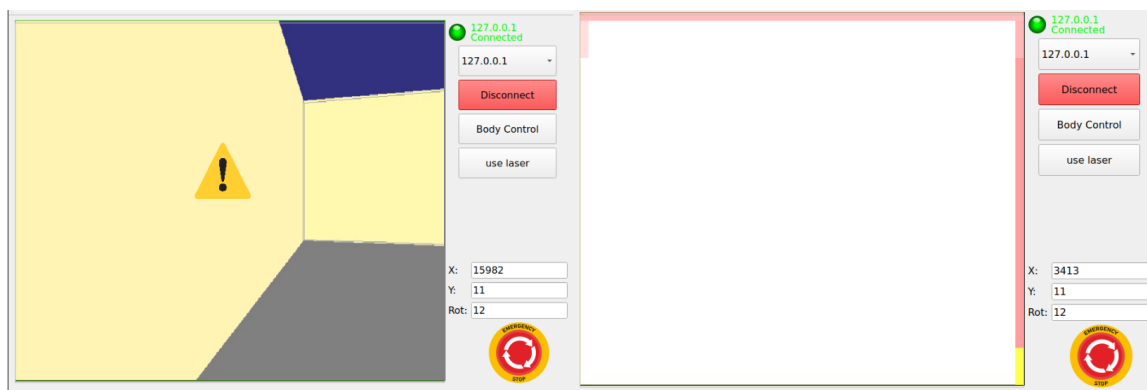
Robota sme si rozdelili na 8 častí a z nich sme následné spracovali dáta pre vizualizáciu. Robota delíme na prednú/zadnú časť pravú/ľavú časť po 40 stupňov a pomedzi ne sú ďalšie 50 stupňov sektory, ktoré nám poskytujú ďalšie informácie o predmetoch v okolí robota. Taktiež v HMI je implementovaný detektor kolízií. Parkovacia kamera vykresľuje primárne body za robotom, ale ponecháva operátorovy informáciu aj o prekážkach pred robotom.

### 2.3 Kamerová časť

Detegovali sme prekážky ktoré sa nachádzajú už 80cm od robota, a ak sa predmet nachádzal bližšie než 50cm od robota, zvýšili sme úroveň alarmu. Na kamere sa nevyk-

reslujú dáta nachádzajúce sa priamo za robotom, ale zvýrazňujeme 7 sektorov v okolí robota.

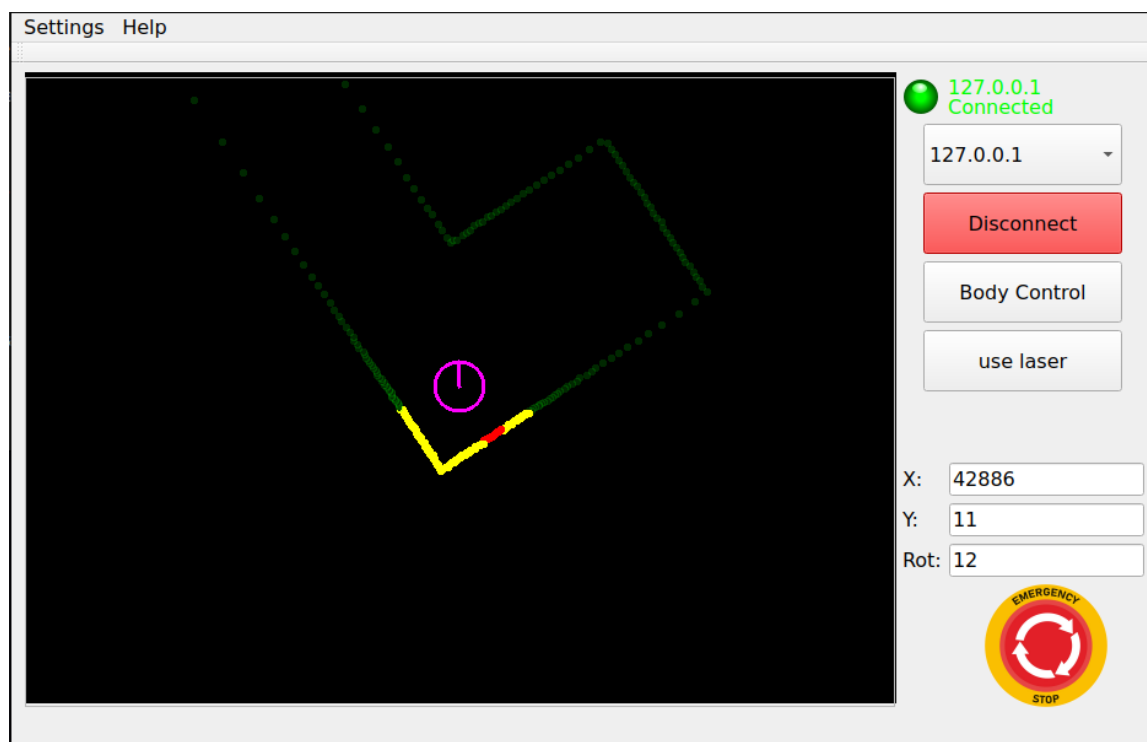
Alarm výzera následovne:



Obr. 2.1: Alarmy použité na upozornenie operátora o objektoch v blízkosti robota

## 2.4 Parkovacia kamera

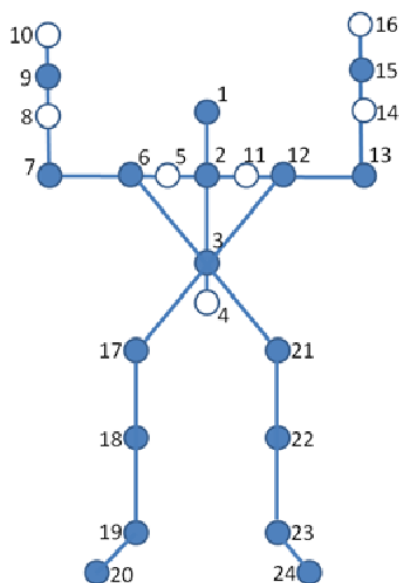
Pri zmene smere robota na chod vzad, sa automaticky zapne parkovacia kamera, na ktorej sa zafarbujú predmety ležiace za robotom. Body postupne menia farbu z bezpečnej zelenej cez žltú na červenú farbu.



Obr. 2.2: Parkovací asistent

### 3 Ovládanie gestami

Ovládač reaguje na operátorove zápästie a lakeť pravej aj ľavej ruky. Pohyb je v prirodzenom pracovnom priestore kĺbov ruky. S ľavou rukou sa nastavuje rýchlosť v pred aj vzad od 300 po -300 mm/s, zväčšením uhla medzi laktom a zápästím sa rýchlosť pridáva a naopak. Pravou rukou sa nastavuje rotačná rýchlosť rovnakým spôsobom ako pri rýchlosti, stým že rozmedzie je od -45 stupňov až 45 stupňov za sekundu. Týmto typom ovládania je možné robotovi poslať tri typy rýchlosti.



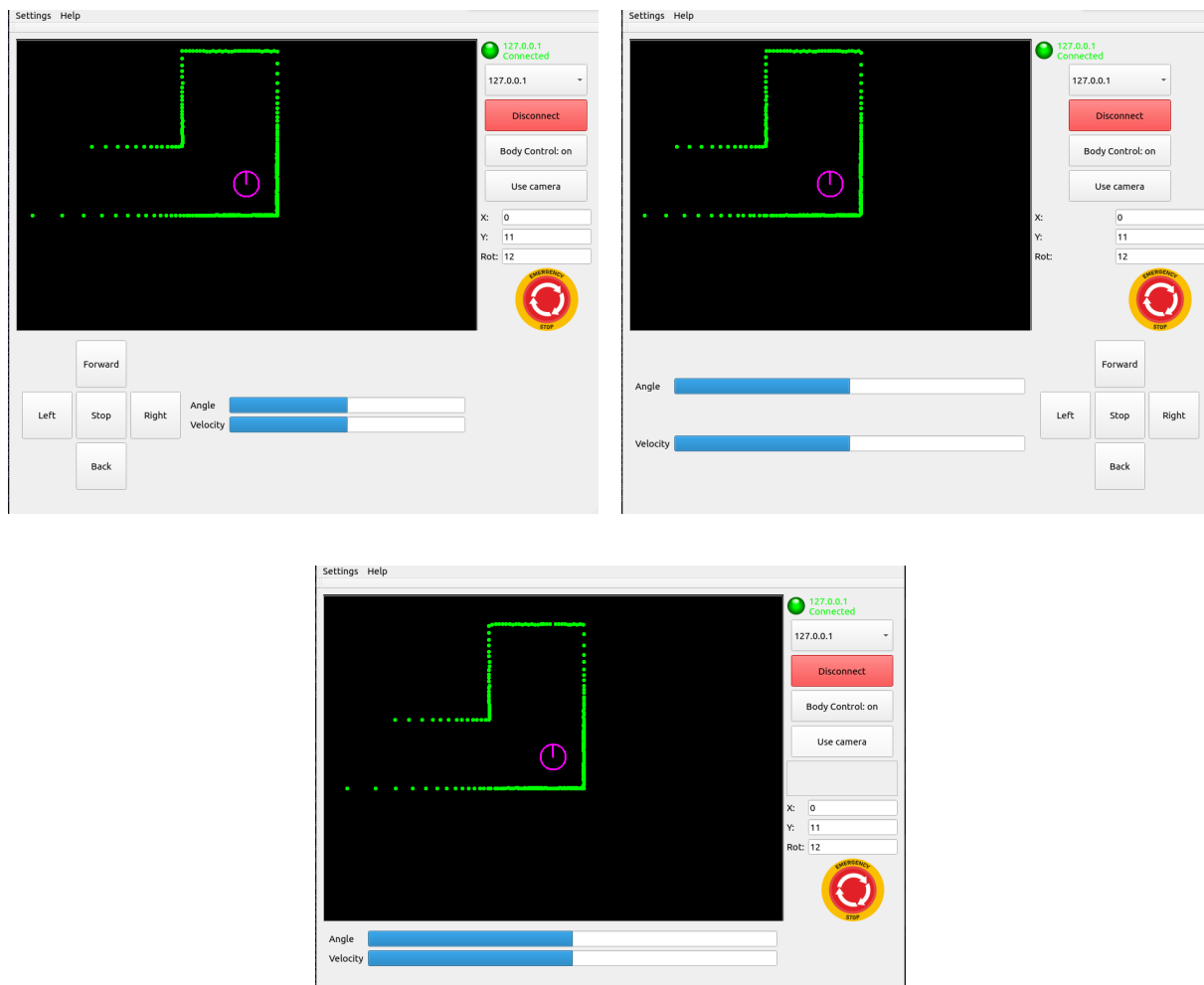
Obr. 3.1: Kĺby ľudského tela

Na 3.1 sú znázornené kĺby ktoré sa na ľudskom tele detegujú. My sa zaoberáme kĺbom 7 voči 9 a 13 voči 15, kde obe ruky sa sklápajú do vnútra. Nulové rýchlosti sú pri 45 stupňovom uhle.

Aby mal užívateľ aj spätnú väzbu o tom, ako aplikácia vníma natocenia jeho rúk, tak si môže pridať zobrazenie relatívnej hodnoty. Toto zobrazenie vidíme na 3.2.

Zaroven, ako si môžeme všimnúť, tak mali frame, krory sme mali na pravo od hlavnej zobrazujúcej plochy nam zanikol. Stalo sa to z dôvodu, že sme nemali dostatok miesta na jeho vykreslenie. Preto, aby sme neposkytovali necitateľne informácie, tak sme sa rozhodli tento frame odstrániť.



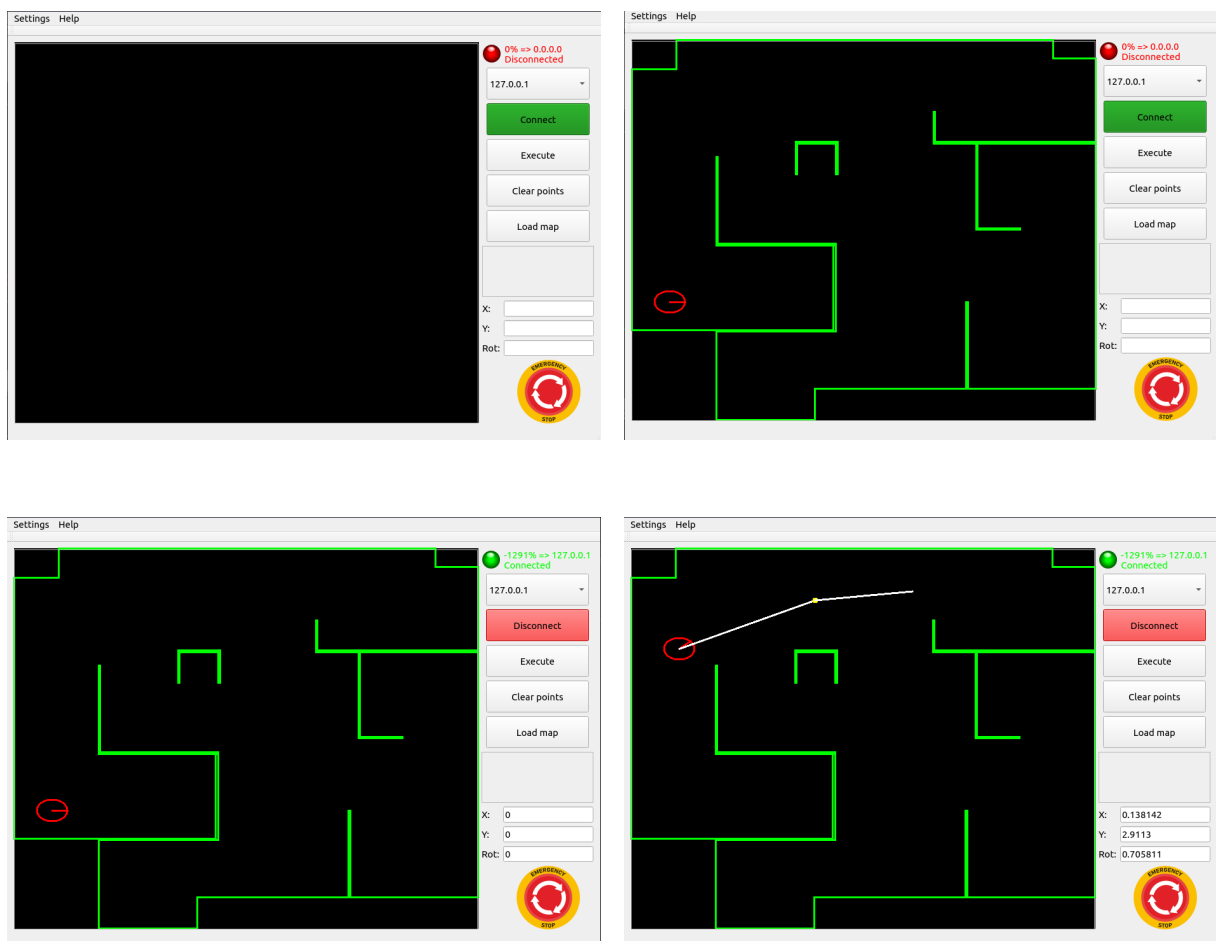


Obr. 3.2: Relativne zobrazenie polohy ruk na ovladanie robota.

### 3.1 Navádzanie na body

Navádzanie na body je spravené cez body zadané operátorom. Body si ďalej berieme z vektora a navádzame ne cez kružnice. Na príchodných bodoch robot nezastavuje. Príchodzie body sú zobrazené na mape bodmi pre lepšiu viditeľnosť. Tieto body môže operátor vymazať alebo začať vykonávať trajektóriu. Ak užívateľ opätovne klikne na dane body, tak ich zruší. Tieto body môže užívateľ presúvať pomocou stredného tlačidla na myši. Ak by ich užívateľ presunul do takej pozície, aby robot narazil do prekážky, tak sa trajektória vykreslí na červeno a nebude vedieť túto trajektóriu vykonať.

Načítavanie mapy je inicializované operátorom. Výzor aplikácie je zobrazený na 3.3.



Obr. 3.3: Zobrazenie mapy a bodov na mape.

Ako vidíme na obrazoch 3.3, zobrazujeme mapu, body na mape, trajektóriu a stav robota. Taktiež si môžeme všimnúť, že sme pridali do aplikácie aj zobrazenie aktuálneho percentuálneho stavu batérie robota. Tento stav ale pri simulátore nefunguje a preto vidíme na obrázku stav batérie ako *-1291*.

Navádzanie robota na dane body je spravené cez počítanie odometrie a následne výpočet trajektórie použitím P regulátora s antiwindup zapojením. Navigáciu nebudeme ďalej rozoberať, keďže nie je hlavnou súčasťou tohto predmetu a zadania. Pre bližšie informácie k navigácii sa obráťte na dokumentáciu k projektu z predmetu *Riadenie mobilných robotov*.

# Záver

Conclusion is going to be where?

Here.

# Prílohy

A	Štruktúra elektronického nosiča . . . . .	15
B	Algoritmus . . . . .	16
C	Výpis subline . . . . .	17

# A Štruktúra elektronického nosiča

*/CHANGELOG.md*

- file describing changes made to FEIstyle

*/example.tex*

- main example *.tex* file for diploma thesis

*/example\_paper.tex*

- example *.tex* file for seminar paper

*/Makefile*

- simply Makefile – build system

*/fei.sublime-project*

- is project file with build in Build System for Sublime Text 3

**/img**

- folder with images

**/includes**

- files with content

*/bibliography.bib*

- bibliography file

*/attachmentA.tex*

- this very file

# B Algoritmus

---

**Algoritmus B.1** Vypočítaj  $y = x^n$

---

**Require:**  $n \geq 0 \vee x \neq 0$

**Ensure:**  $y = x^n$

$y \leftarrow 1$

**if**  $n < 0$  **then**

$X \leftarrow 1/x$

$N \leftarrow -n$

**else**

$X \leftarrow x$

$N \leftarrow n$

**end if**

**while**  $N \neq 0$  **do**

**if**  $N$  is even **then**

$X \leftarrow X \times X$

$N \leftarrow N/2$

**else**  $\{N$  is odd $\}$

$y \leftarrow y \times X$

$N \leftarrow N - 1$

**end if**

**end while**

---

# C Výpis sublime

```
../.. / fei .sublime-project
```

Výpis C.1: Ukážka sublime-project