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

**Riadenie mobilných robotov**

Adrian Komorný

Mário Vido

**Úloha 1. Lokalizácia a polohovanie robota v prostredí (10b)**

Úlohou v tomto zadaní bolo zistiť kde sa robot nachádza a následne ho poslať na želané súradnice.

Polohu robota sme zistili pomocou enkodera a uhol natočenie pomocou Gyra. Ako prvé sme si inicializovali premenne kde sme si uložili predošlú hodnotu ktorú sme využili na vypočítanie súradníc a uhla natočenia.

Obrázok, na ktorom je text, snímka obrazovky, písmo

Automaticky generovaný popis

V tejto funkcii si ukladáme začiatočné pozície a začiatočný uhol natočenia. Init slúži ako pomocná premenná ktorá slúži aby nám cyklus v hlavnom loope sa vykonal len raz.

Potrebovali sme postupne ako sa mobilný robot hýbal ukladať hodnoty pravého a ľavého enkodera. Pri ukladaní sme si museli dávať pozor lebo hodnota enkodera mohla pretiecť keď bola vyššia ako 65535. Tento problém sme vyriešili tak že sme spravili if cykly ktoré nám zistia ak sa enkoder dostal pod alebo nad určitú hranicu. Keď sa dostal nad alebo pod túto hranicu tak sme odčítali tú hodnotu od maximálnej hodnoty enkodera.

Obrázok, na ktorom je text, snímka obrazovky, písmo

Automaticky generovaný popis

Následne sme potrebovali ešte zistiť uhol natočenia robota. Ten sme zistili pomocou gyra. Ako posledné sme zistili aktuálne pozície súradníc x, y a vzdialenosť pomocou vzorcov s prezentácie

Obrázok, na ktorom je text, snímka obrazovky, písmo

Automaticky generovaný popis

Keď sme mali zistenú polohu potrebovali sme robot dostať do určitého bodu. Nato sme využili premenné koncovýX a koncovýY. Ako prvé sme potrebovali zistiť veľkosť uhla ktorý sa nachádza medzi naším robotom a bodom kam sa chceme dostať.

Obrázok, na ktorom je text, snímka obrazovky, písmo

Automaticky generovaný popis

Keď sme mali uhol sme potrebovali robot otočiť až po bod kým sa nenachádzal v rozmedzí +- 3 stupňov od bodu kde sa chceme dostať. Ak sa robot nachádazl mimo tohto rozmedzia tak sme robot otáčali.

Obrázok, na ktorom je text, snímka obrazovky, písmo

Automaticky generovaný popis

Použili sme P regulátor na regulovanie rýchlosti otáčanie.

Keď sa robot dostal do toho rozmedzia tak sme poslali robota dopredu. Na pohyb sme si vytvorili rampy ktoré nám postupne zrýchľovali robota a spomaľovali.  
Obrázok, na ktorom je text, snímka obrazovky

Automaticky generovaný popis

Zastavenie robota sme vyriešili tak, že keď sa robot dostal do okolia bodu tak sme mu nastavili 0 rýchlosť. Keďže mi sme písali všetky úlohy do jedného projektu pri zastavení robota máme aj pole ktoré využívame v 4. úlohe. Túto funkcionalitu priblížime bližšie neskôr.

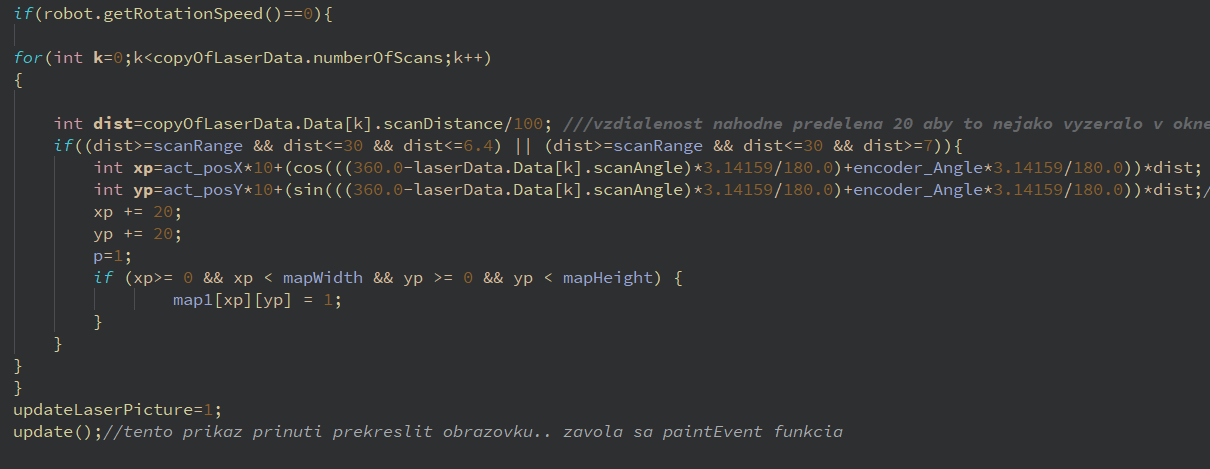
Obrázok, na ktorom je text, snímka obrazovky, softvér

Automaticky generovaný popis

**Druhú úlohu sme nevypracovali.**

**Úloha 3. Mapovanie**

V tejto úlohe sme mali namapovať prostredie ktorým prechádzal robot pomocou snímačov robota. Na spracovanie bodov kde sa nachádza stena sme použili už existujúcu funkciu lidaru kde voláme dáta s lidaru. Tu sme vytvorili script ktorý nám vypočíta polohu steny podľa dát a uloží nám ich do poľa.



Nato aby sa nám dobre zosnímali dáta sme potrebovali najskôr určiť pár podmienok. Merať sme začali len od určitej vzdialenosti od robota do určiej vzdialenosti a nemali sme marať body zo vzdialenosti 6.4 až 7 dm.   
Na uloženie mapy sme vytvorili nové tlačidlo ktoré nám uložilo body s poľa do textového dokumentu ako 1 a 0. Tu sme si nastavili akú veľkú chceme mapu a názov uloženého súboru.

Obrázok, na ktorom je text, snímka obrazovky, písmo

Automaticky generovaný popis

**Úloha 4. Plánovanie dráhy**

Úlohu 4 sme začali načítaním 2d mapy. zhotovenej za pomoci 0 a 1 a dvoch ďalších znakov. 0 predstavovala prázdny štvorček miesta, 1 steny 3 koniec kde sa ma robot dostať a ~ tento znak označoval miesto začiatku robota, a je to zároveň posledný ascii znak. Flood algoritmus sa začal napĺňať od konca, logika fungovala na princípe prehľadávania každého políčka na mape a zároveň, sa pointer pozeral ci dane políčko x,y je na 0 alebo 1 súradnici. To znamená ak pointer x,y ležal na už zaplavenej súradnici tak sa algoritmus pozrel na všetkých susedov, a ak boli 0 tak ich prepísal o ++1.

Hlavná while podmienka je nasledovná, ak susedia začiatočného bodu sú nenulový tak sa algoritmus zruší, to znamená ak sa nejaké číslo priblíži ku štartu mapa je vyplnená.

Nasledovne sa pointer nastaví na začiatočný pod a za pomoci 4 „bufferovych“ premenných sa bude do každej načítavať súčet riadkov v jednom smere, dokým platí ,napr. pre premennú hore\_ na začiatku je nulová a pozerá sa ci políčko nad pointerom je x-1, ak to platí ideme správne, pozrú sa všetky smery, a ktorý je najdlhší tak pointer sa presunie na najzvalenejšie miesto v poradí suseda x-1.

Takto bude algoritmus fungovať až pokiaľ nepríde do cieľa. Každý z kľúčových bodov kde sa otáča, sa zapíše a nakoniec sa robot posiela na dané súradnice za sebou až pokiaľ nepríde do cieľa.