

»**Mladi za napredek Maribora 2019**«

**36. srečanje**

# ***Reševalni robot***

Raziskovalno področje: **Elektrotehnika in elektronika**

Inovacijski predlog

PROSTOR ZA NALEPKO

Avtor: JAKA WALDHÜTTER

Mentor: MIRAN WALDHÜTTER

Šola: SREDNJA ELEKTRO-RAČUNALNIŠKA ŠOLA MARIBOR

Število točk: 158

Mesto: 1

Priznanje: srebrno

**Maribor, 2019**

»**Mladi za napredek Maribora 2019**«

**36. srečanje**

# ***Reševalni robot***

Raziskovalno področje: **Elektrotehnika in elektronika**

Inovacijski predlog

PROSTOR ZA NALEPKO

**Maribor, 2019**

# 1 KAZALO VSEBINE

<b>1</b>	<b>Kazalo vsebine</b>	<b>2</b>
<b>2</b>	<b>Kazalo slik</b>	<b>4</b>
<b>3</b>	<b>Povzetek</b>	<b>6</b>
<b>4</b>	<b>Zahvala</b>	<b>8</b>
<b>5</b>	<b>Vsebinski del</b>	<b>9</b>
5.1	Uvod	9
5.2	Zakaj uporabiti ta sistem (konkurenca oz. obstoječi roboti)	10
5.3	RoboCup	11
5.4	Sestava in posamezni elementi robota	12
5.4.1	Osrčje robota	12
5.4.2	Napajanje	13
5.4.3	Motorji	14
5.4.4	Senzor razdalje	15
5.4.5	Zaznavanje termalnih žrtev – senzor temperature	15
5.4.6	Zaznavanje črnega polja	16
5.4.7	Kompas in giroskop	17
5.4.8	Vizualne žrtve	17
5.4.9	Razdeljevalec reševalnih paketov	18
5.5	Programiranje in delovanje posameznih elementov	19
5.5.1	Nadzor motorjev	19

5.5.2	Senzor razdalje	19
5.5.3	Temperaturni senzor	19
5.5.4	Senzor barve	20
5.5.5	Raspberry Pi in prepoznavna vizualnih žrtev	21
5.5.6	Zasnova celotnega programa	22
<b>5.6</b>	<b>3D modeliranje in ostali deli robota</b>	<b>22</b>
<b>5.7</b>	<b>Izvedba</b>	<b>23</b>
<b>5.8</b>	<b>Implementacija v resničnem življenju</b>	<b>24</b>
5.8.1	Spremembe in ideje pri implementaciji v resničnem življenju	24
5.8.2	Implementacija na različnih robotih	26
<b>6</b>	<b>Zaključek</b>	<b>29</b>
<b>7</b>	<b>Družbena odgovornost</b>	<b>30</b>
<b>8</b>	<b>Viri</b>	<b>31</b>
8.1	Spletne strani:	31
8.2	Dokumenti na spletu	32
8.3	Viri slik	32

## 2 KAZALO SLIK

Slika 1: V2.....	10
Slika 2: ROBOCUE .....	10
Slika 3: GOSAFER .....	10
Slika 4: Primer labirinta .....	12
Slika 5: Arduino mega.....	13
Slika 6: Lipo baterija .....	13
Slika 7: Polnilna enota za LIPO baterije .....	14
Slika 8: Povezava motorjev z mikrokrmilnikom Arduino Mega .....	14
Slika 9: Laser PING range finder .....	15
Slika 10: Modul s temperaturnim senzorjem .....	16
Slika 11: Barvni senzor .....	16
Slika 12: IMU 10 DOF.....	17
Slika 13: Raspberry Pi z kamera modulom .....	18
Slika 14: Razdeljevalnik paketov .....	18
Slika 15: Ukaz motorjem.....	19
Slika 16: Uporaba ping senzorja .....	19
Slika 17: Program temperaturnega modula .....	20
Slika 18: Odsek programa za senzor TCS3200.....	21
Slika 19: Primer obdelave slike.....	21
Slika 20: 3D model robota .....	22

---

Slika 21: Prototip robota .....	23
Slika 22: Preizkus senzorjev .....	24
Slika 23: Panasonic Grid eye zaznavanje oseb .....	25
Slika 24: MQ-9.....	26
Slika 25: XRL.....	27
Slika 26: RSTAR.....	27
Slika 27: Velox.....	27
Slika 28: QUINCE.....	27

### 3 POVZETEK

Inovacijski predlog je sistem za robote skavte, ki bi namesto ljudi raziskovali območja nesreč in iskali morebitne preživele ponesrečence.

Vsak dan se na svetu zgodi veliko nesreč. Takšne, ki se zgodijo v eni stavbi in takšne, ki prizadenejo cela mesta. Ob reševanju ponesrečencev iz teh območij se pogosto poškodujejo tudi reševalci, ki zaradi nepoznavanja območja v le to pridejo nepripravljeni. Na svetu zato marsikdo razvija robota, ki bi se lahko podal v te situacije in reševal ljudi iz teh situacij. Tehnologija še ni dovolj razvita, da bi lahko roboti varno in učinkovito reševali ljudi v različnih okoliščinah ali pa je cena takšnih robotov takšna, da je, na žalost, človeško življenje pogosto cenejše.

Zamislil sem si sistem, ki bi lahko bil splošno dostopen, vendar še vedno uporaben. S vpogledom v nevarno območje, bi lahko reševalne akcije optimizirali in rešili veliko več ljudi, kot jih rešujemo zdaj. Reševalci bi se lahko v naprej pripravili in določili prioriteto ponesrečencev. Tisti, ki so nujno potrebni pomoči, bi jo prejeli hitreje, kot tisti z le manjšimi poškodbami.

Roboti, ki bi uporabljali moj sistem, bi imeli veliko prednosti pred že obstoječimi reševalni roboti. So cenejši, lažji, bolj mobilni, ne ogrožajo ljudi, lahko delujejo hitreje in porabljajo manj energije.

Ob raziskovanju sem začel izdelovati robota, ki bo ta sistem uporabljal. Robot bo izdelan za namene tekmovanja RoboCup v kategoriji Rescue. Naloge tega tekmovanja sovpadajo s problemi, ki sem jih želel rešiti. Robot se mora samostojno premikati po labirintu in poiskati žrtve. Prepoznati mora tudi črke na steni.

Robot, ki sem ga začel izdelovati, uporablja za glaven nadzor mikrokontroler Arduino Mega 2560. Vsebuje senzorje za razdaljo, temperaturo, ter barve. Za branje in zaznavanje črk uporablja mikroračunalnika Raspberry Pi 3 modela B+. Za premikanje uporablja pametne servo motorje, s katerimi dobimo res natančen nadzor nad robotom. Vključen bo imel tudi kompas in giroskop, ki bo robotu omogočal boljšo predstavo o poziciji v okolju.

Program bo deloval tako, da bo celotno območje razdelil v kvadratna polja. Premikal se bo po poljih in se v vsakem ustavil. Takrat bo preveril prisotnost sten in si tako narisal sliko labirinta,

po kateri bo se lahko natančno koordiniral – vrnil se bo lahko na neraziskana območja, razlikoval pa bo tudi med žrtvami, ki jih je že obiskal in med takšnimi, katerih še ni.

Namen naloge je izdelava sistema, ki ga lahko uporabimo v različnih okoliščinah in v različnih robotih.

Roboti, ki bi uporabljali moj sistem bi delovali tako, da bi konstantno preverjali prisotnost sten. Ponesrečence bi zaznavali s temperaturnim senzorjem Panasonic Grid eye. Dodamo lahko tudi prepoznavo obrazov, s katero lahko zagotovimo prepoznavanje žrtev. Kamero bi lahko uporabili tudi tako, da ob morebitni zaznavi ponesrečenca, naredi sliko in reševalci jo potrdijo ali zavrnejo. Med izrisovanjem mape – v mapi je tudi vidno, kje je robot zaznal ponesrečence - bi robot mapo pošiljal reševalcem. Robot bi lahko imel vključene tudi senzorje za zaznavanje različnih plinov in ogenj. Vsi ti podatki bi reševalcem pomagali pripraviti se na reševalno akcijo.



## **4 ZAHVALA**

Za nasvete, pomoč in potrpljenje se zahvaljujem svojemu mentorju, ki mi je stal ob strani skozi vse težave in mi svetoval ter pomagal izoblikovati ta inovacijski predlog. Zahvaljujem se tudi svoji družini, ki me je ob vsem podpirala.

## 5 VSEBINSKI DEL

### 5.1 Uvod

Vsak dan se po celem svetu dogajajo grozne nesreče – naravne ali takšne, ki jih je povzročil človek. Zaradi nepripravljenosti in nepoznavanja okolja se poškodujejo tudi ljudje, ki bi drugače preživeli, če bi jih reševalci prej našli. V nesrečah zaradi nevednosti umirajo tudi reševalci, ki želijo pomagati.

Človeško življenje je edina stvar, ki je na tem svetu ne moremo zamenjati in verjamem, da lahko takšne smrti preprečimo.

Odločil sem se, da želim ustvariti robota, ki bo izboljšal reševalne akcije in bo pomagal k hitrejšemu in boljšemu sistemu za reševanje ljudi iz nesreč.

Človeško življenje je neprecenljivo. Vendar marsikje na svetu to ne drži. Reševalni roboti na trgu so izjemno dragi in ne izpopolnjeni, osredotočajo pa se na nepomembne stvari oz. na kompleksne situacije, ki jih trenutno še ne moremo rešiti. Vsaj ne na način, ki bi omogočal dostopnost rešitve. Želel sem ustvariti robota, ki

- lahko deluje v kriznih območjih in območjih naravnih nesreč,
- je cenovno bolj dostopen,

ampak sem hitro opazil, da je takšen robot zame, kljub vsemu še vedno prevelik finančen zalogaj.

V srednji šoli sem naletel na tekmovanje RoboCup. To je robotsko tekmovanje, ki ima veliko podkategorij, med drugim tudi Rescue Maze. To tekmovanje je usmerjeno k iskanju rešitev problemov točno na tem področju, saj tekmovališče simulira kraj nesreče po katerem se mora robot samostojno premikati in iskati žrtve. Sodeloval sem na dveh svetovnih tekmovanjih in letos se odpravljam na tretjega. Lani sem z sotekmovalcem dobil priznanje za najboljšo inovacijo.

Letos sem se odločil, da bom robota nadgradil, dopolnil, ob tem pa bom izdelal sistem, katerega lahko implementiramo v resničnem življenju in bo izboljšal delovanje trenutnih reševalnih služb in reševalnih akcij.

## 5.2 Zakaj uporabiti ta sistem (konkurenca oz. obstoječi roboti)

Zavedal sem se, da na svetu obstaja veliko reševalnih robotov, primeri katerih so na slikah 1, 2 in 3.



Slika 1: V2<sup>1</sup>



Slika 2: ROBOCUE<sup>2</sup>



Slika 3: GOSAFER<sup>3</sup>

Opazil sem, da se večina osredotoča na izdelavo robota, ki ga bo kontroliral človek, kar je slabo, saj to pomeni, da dopuščamo možnost napak, kot pravi rek: »motiti se je človeško«. Takšni roboti potrebujejo človeškega upravljalca, ki potrebuje mesece učenja in preučevanja robota, imeti mora izjemen občutek za predstavo, za okolje in za ocenjevanje situacije. Kakor hitro pa

---

<sup>1</sup> Iranski robot V2, ustvarjen za pomoč ranjenih vojakov

<sup>2</sup> ROBOCUE – Robot razvit na japonskem za pomoč reševanja ljudi iz potresnih območij

<sup>3</sup> GOSAFEE – Robot podjetja The Parosha Cheetah, zasnovan za reševanje ljudi iz nevarnih situacij

v enačbo dodamo človeški element dopuščamo tudi možnost slabega dneva, utrujenosti, zmanjšane pozornosti itd.

Osredotočajo se predvsem na izdelavo robota, ki bo lahko reševal ljudi iz teh situacij. To je zelo velika slabost, saj s tem v večini primerov ponesrečenca, zaradi nepredvidljivih in kompleksnih situacij, še bolj poškodujemo in ga postavljamo v veliko večjo nevarnost. Mislim, da s tem ne izkoriščamo vseh možnosti in potencialov robotike. Z osredotočanjem na te stvari, pa povzročamo samo še večje stroške.

Marsikateri reševalni roboti so tudi zelo težki, kar je lahko zelo nevarno, saj s takim robotom povzročamo veliko obremenitev na že tako poškodovano strukturo.

Takšni roboti so zelo dragi, kar preprečuje njihovo splošno dostopnost.

Po teh ugotovitvah sem ugotovil, da želim ustvariti robota izvidnika (skavta), ki bo deloval avtonomno oz. samostojno. Robot se bo samostojno premikal po kraju nesreče, poiskal žrtev in poslal njeno lokacijo reševalcem. Pregledal bo če je območje varno, preverjal bo prisotnost nevarnih plinov in požarov, prav tako pa bo ob najdbi žrtve žrtvi dostavil reševalni paket prve pomoči, hrano ter pijačo, s čimer bo ponesrečenec lahko pričakal prihod reševalcev.

Ta ideja se je skladala z nalogami kategorije Rescue Maze na tekmovanju RoboCup:

### **5.3 RoboCup**

Na tekmovanje robotike RoboCup sem naletel v drugem letniku srednje šole. Tekmovanje ima več kategorij, meni najbolj primerna pa je Rescue Maze B. V tej kategoriji je simulirana situacija nesreče kjer je območje prenevarno, da bi v vanj vstopil človeški reševalec. Ustvariti moramo robota, ki bo raziskal čim več labirinta ter poiskal žrtve. Žrtve bodo podane v obliki ogrevanih teles ali natiskanih črk na stenah. Ob prepoznavi žrtve mora robot spustiti simuliran reševalni paket.

Labirint je v dveh nadstropjih, ki sta povezani s klancem. Sestavljen bo tudi iz samostoječih sten, ki se ne dotikajo drugih sten. Tako, ob preprostem sledenju desni (ali levi) steni, teh sten ne zaznamo. Primer takšnega labirinta je viden na sliki 4.



Slika 4: Primer labirinta

Po celem labirintu so razmetane ovire, kot so zobotrebeci ali lesene palice, ovire so pa lahko tudi v obliki valjev s premerom 20 centimetrov. To simulira oz. predstavlja ostanke uničenih sten ali podobne probleme, ki jih lahko najdemo v resnični situaciji. Na tleh so tudi črna polja, ki predstavljajo nevarna območja, katerih robot ne sme prečkati. Srebrna polja predstavljajo »checkpointe« – točke na katere se lahko robot v primeru napake vrne.

Glavna namen tekmovanja je ustvariti najboljši način oz. najti najboljšo rešitev že znanemu problemu.

## 5.4 Sestava in posamezni elementi robota

### 5.4.1 Osrčje robota

Za glavni nadzor robota sem izbral mikrokrmilnik Arduino Mega 2560, saj ima največ (54 digitalnih in 16 analognih) vhodov izmed vseh Arduino mikrokrmilnikov. Vidimo ga lahko na sliki 5. Mikrokrmilniki Arduino so univerzalni mikrokrmilniki, ki delujejo na odprtokodni kulturi, zato so idealni za takšne projekte, izbral pa sem ga tudi zato, ker imam z njim največ izkušenj.



Slika 5: Arduino mega

### 5.4.2 Napajanje

Za napajanje sem uporabil tri celično LiPo (Litij-ionske) baterije. To so polnilne baterije (akumulatorji), ki so sestavljene iz treh celic, napetosti 3,7 V. Vse celice skupaj nam tako dajo 11,1 V. Napetost polne baterije doseže 12,6 V, napetost prazne pa ne sme pasti pod 9 V, saj se v tem primeru baterija lahko trajno pokvari. V vezju sem zato dodal posebno varovalno vezje, ki ob padcu napetosti pod 9 V prekine povezavo baterije z krmilnikom.

Primer baterije, ki jo uporabljam je na sliki 6. Veliki žici (rdeča in črna) dajeta skupno napetost (11,1 V); manjši konektor pa je namenjen varnemu polnjenju in nadzoru posameznih celic.



Slika 6: Lipo baterija

Polnjenje Lipo baterij je lahko zelo nevarno, saj moramo konstanto preverjati stanje posameznih celic in obdržati vse celice v ravnovesju. Če na to ne pazimo, se lahko baterija trajno poškoduje. Za polnjenje zato uporabljamo posebno polnilno enoto, ki jo lahko vidimo na sliki 7.

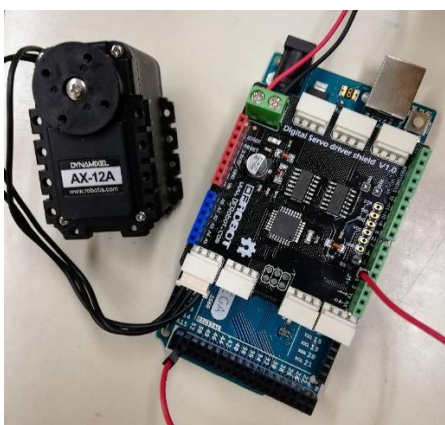


Slika 7: Polnilna enota za LIPO baterije

### 5.4.3 Motorji

Za premikanje robota sem izbral »pametne« servo motorje, imenovane AX-12A. To so motorji, ki se lahko konstanto vrtijo (kot normalni motorji) hkrati pa lahko preverjajo svojo pozicijo. Izdeluje jih podjetje dynamixel.

Motorčki za komunikacijo uporabljajo poseben protokol, zato ne morejo komunicirati direktno z mikrokrmilnikom. V ta namen so proizvajalci ustvarili poseben Shield (dodatek), ki ga lahko uporabimo za komunikacijo. Nanj lahko priključimo 6 pametnih motorčkov, s katerimi lahko Arduino komunicira preko samo enega priključka (v našem primeru digitalni priključek 53). Shield na Arduino Megi, lahko vidimo na sliki 8. Rdeča žica je zgoraj omenjena povezava med motorji in mikrokontrolerjem.



Slika 8: Povezava motorjev z mikrokrmilnikom Arduino Mega

#### 5.4.4 Senzor razdalje

Odločil sem se, da bo robot, za čim bolj dodelano premikanje uporabljal osem senzorjev – po dva na vsaki strani. Tako bom dosegel natančnejše meritve in zagotovil možnost poravnavanja na vsako steno. Robot, bo tudi dobil realno predstavo sveta okoli sebe – z lahkoto bo prepoznal razliko med ovirami in stenami.

Uporabil bi lahko samo tri senzorje, vendar se je to izkazalo kot manj natančno. Ker so vse stene v kockah, je premikanje veliko hitrejšo če se robot premika v ravni črti. Z uporabo osmih senzorjev lahko robota poravnamo in mu tako zagotovimo najhitrejšo in najbolj učinkovito pot po labirintu.

Za senzorje sem izbral laserske senzorje, za karseda natančne meritve. Laserski senzorji so tudi veliko manjši in tako bolj primerni mojemu cilju.

Senzorji so v paketu PING, podjetja Parallax. PING je vrsta uporabe oz. komunikacije s senzorjem, kjer senzor upravljamo preko enega samega priključka na mikrokrmilniku. Takšno varianto sem izbral saj je pri takšni komunikaciji najlažje poiskati napake in zamenjati problematične senzorje. Prav tako bi pri I2C komunikaciji, katero uporablja večina senzorjev, lahko prišlo do problemov, kakršne je pri takšnem številu senzorjev težje odpraviti. To težavo sem opazil iz prejšnjih let, prav tako pa jo opažam pri tekmovalcih iz drugih držav na tekmovanju. Primer uporabljenega senzorja je na sliki 9.



Slika 9: Laser PING range finder

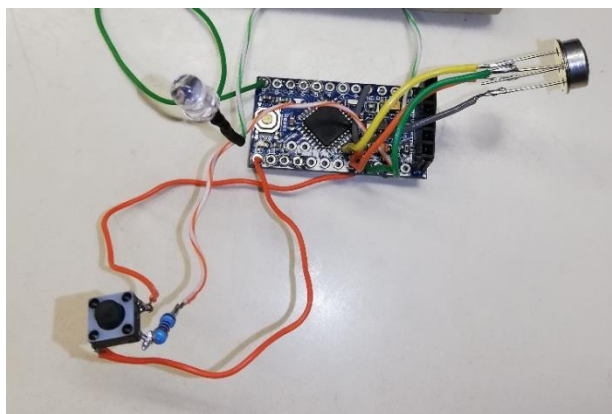
#### 5.4.5 Zaznavanje termalnih žrtev – senzor temperature

Na polju bodo temperaturene žrtve, ki bodo imele temperaturo, vsaj za 10° višjo kot so ostali deli labirinta – okoli 30° C. Za zaznavanje termalnih žrtev za namene tekmovanja sem uporabil



senzor Melexis' MLX90614ESF-BAA. To je Infra-rdeči senzor temperature, ki temperaturo objekta zaznava brez dotika.

Da bi poenostavil stvari in razbremenil glaven mikrokrmilnik (Arduino mega) sem se odločil, da bom vsak temperaturni senzor povezal na samostojen mikrokrmilnik Arduino pro mini, ki bo glavnemu mikrokrmilniku poslal signal le ob prekoračitve neke določene temperaturne meje (npr. ko bo temperatura objekta prekoračila 30° C). V ta paket sem dodal tudi gumb, s katerim lahko kalibriram mejno temperaturo in svetilno diodo (LED), preko katere lahko vidim, če modul zaznava žrtev. Ta modul lahko vidimo na sliki 10.



Slika 10: Modul s temperaturnim senzorjem

#### 5.4.6 Zaznavanje črnega polja

Za zaznavanje črnega polja uporabljam barvni senzor TCS3200, ki ga izdeluje podjetje DFROBOT. Senzor nam poda vrednosti RGB, iz katerih lahko natančno določamo realne barve objektov. Da preprečim motnje razsvetljave oz. zunanje svetlobe, bom naredil tulec, v katerega bom postavil senzor tako, da bo karseda natančno zaznaval barve. Senzor je viden na sliki 11.



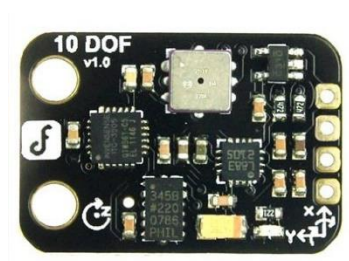
Slika 11: Barvni senzor

### 5.4.7 Kompas in giroskop

Da bi izboljšal delovanje in hitrost robota sem se odločil da bom vanj dodal digitalni kompas, katerega bo robot uporabljal za obračanje in lažjo navigacijo po prostoru.

Robot mora vedeti kdaj prestopi iz enega nadstropja v drugo in v ta namen sem želel dodati tudi giroskop – po klancu bi zaznal naklon.

Oba problema sem rešil z enim samim modulom, ki je združen kompas, giroskop, vključene pa ima tudi druge uporabne senzorje, kot so barometer, magnetometer ipd. To je senzor IMU 10 DOF, podjetja DFROBOT, ki ga lahko vidimo na sliki 12.



Slika 12: IMU 10 DOF

### 5.4.8 Vizualne žrtve

Na tekmovalnem labirintu bodo na stene prilepljene črke S, H in U. Črke nam podajajo stanje žrtve - S stabilen (stable), H – ranjen (harmed) in U – nepoškodovan (unharmmed). Robot mora za črko S spustiti le en reševalni paket, za črko H dva, pri črki U pa se mora le ustaviti in pokazati, da jo je zaznal (utripanje s svetilno diodo (LED) ali zvočni efekt).

Za zaznavanje črk sem se odločal med mikroračunalnikom Raspberry Pi 3 model B+ s kamero in kamero z mikrokontrolerjem OpenCam MV. Program za prepoznavo črk na Raspberry Pi-u že obstaja, zato sem se odločil, da bom za prepoznavo uporabil ta mikroračunalnik.

Pri izbiri kamere sem bil pazljiv na število sličic, ki jih lahko kamera posname na sekundo (FPS). Odločil sem se za kamero, ki je bila ustvarjena specifično za ta mikroračunalnik (Raspberry Pi Camera module V2). To kamero sem uporabil, saj je zelo majhna kljub temu pa zelo zmogljiva, ker pa je ustvarjena specifično za ta mikroračunalnik, z njim dela najboljše. Pri uporabi drugih (USB) kamer, bi te delovale veliko počasneje, delovale pa bi z veliko manj

sličicami na sekundo, porabile pa bi veliko več energije in centralno procesorske enote. Kamero povezano na Raspberry Pi lahko vidimo na sliki 13.

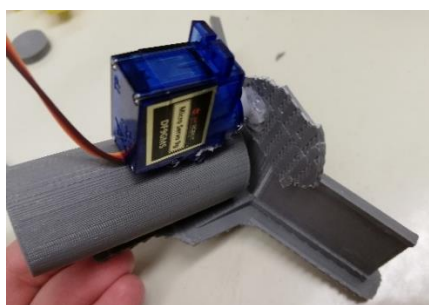


Slika 13: Raspberry Pi z kamera modulom

#### 5.4.9 Razdeljevalec reševalnih paketov

Ob zaznavi žrtve mora robot podati simuliran reševalni paket – ta mora pasti na mesto, ki je oddaljeno največ 15 cm od žrtve. Paket je lahko v katerikoli obliki, imeti mora le prostornino večjo od  $1\text{ cm}^3$ .

Odločil sem se, da bom reševalne pakete simuliral s 3D natiskanimi kolobarji. Razdeljevalec bo sestavljen iz servo motorja, ki bo ob dnu potiskal kolobarje na tisto stran na kateri se nahaja žrtev. Na vsaki strani bo klanec, ki bo paket potisnil čim bolj proti cilju (žrtvi). Razdeljevalec lahko vidimo na sliki 14.



Slika 14: Razdeljevalnik paketov

## 5.5 Programiranje in delovanje posameznih elementov

### 5.5.1 Nadzor motorjev

Za nadziranje motorjev, preko ukaza `myservo.rotate` pošiljam vrednosti s katero se morajo premikati. Prva številka predstavlja ID motorja, druga pa hitrost. 300 je polna hitrost, normalna hitrost bo okoli 150 – s tem so senzorji še dovolj natančni. Primer programa, za preprost premik naravnost lahko vidimo na sliki 15.

```
myservo.rotate(2, -150);  
myservo.rotate(1, 150);
```

Slika 15: Ukaz motorjem

### 5.5.2 Senzor razdalje

Laser PING podjetja parallax deluje tako, da pošlje svetlobni žarek, ki se odbije od stene, nato pa zmeri čas, ki ga je žarek potreboval, da je prepotoval pot tja in nazaj. To deli s hitrostjo svetlobe v zraku in nato z dva, da dobi čas, ki ga ja žarek potreboval za potovanje v eno smer (do stene). Vrednost nam tako pretvori v centimetre. Na začetku so spuščamo signali s katerimi določimo natančnost.

```
long checkPing(int pingOnPin) {  
    long duration;  
    pinMode(pingOnPin, OUTPUT);  
    digitalWrite(pingOnPin, LOW);  
    delayMicroseconds(2);  
    digitalWrite(pingOnPin, HIGH);  
    delayMicroseconds(5);  
    digitalWrite(pingOnPin, LOW);  
  
    | pinMode(pingOnPin, INPUT);  
    duration = pulseIn(pingOnPin, HIGH);  
    return duration / 29 / 2;  
}
```

Slika 16: Uporaba ping senzorja

### 5.5.3 Temperaturni senzor

Deluje na SMBus komunikaciji. To je komunikacija, ki izhaja iz I2C komunikacije. Povezava deluje s štirimi žicami – dve za napetost in dve za komunikacijo. Senzor lahko prebere temperaturo okolja in temperaturo objekta, jaz pa na robotu uporabljam le temperaturo objekta.

Za mikromilnik Arduino obstajajo že napisane knjižnice za ta senzor, katere pri temperaturnem modulu izkoriščam.

Senzorju pošljem ukaz `mlx.ReadObjectTempC()`, nato nam senzor vrne podatek o temperaturi, tega pa primerjam z vrednostjo `maxT`. Če je prebrana vrednost večja kot vrednost spremenljivke `maxT`, modul pošlje signal vrednosti 1 (HIGH) na digitalni priključek, ki je povezan z glavnim mikromilnikom. Istočasno prižge tudi svetilno diodo (LED), kar nam olajša odpravljanje napak. Na modul je priključen gumb, s katerim lahko spreminjamo spremenljivko `maxT`. Ob pritisku na gumb, modul prebere temperaturo objekta, nato odšteje dve stopinji in to vrednost shrani kot vrednost spremenljivke. Pred tem vklopi svetilno diodo, da lahko vemo, da se dogaja kalibracija.

```
void loop() {  
  Serial.println(mlx.readObjectTempC());  
  Serial.println(maxt);  
  if (digitalRead(kalibracija) == HIGH) {  
    digitalWrite (ledkali, HIGH);  
    delay(500);  
    maxt = (mlx.readObjectTempC() - 2);  
    delay(500);  
    digitalWrite (ledkali, LOW);  
  }  
  //Serial.println(checkPing());  
  if (mlx.readObjectTempC() > maxt) {  
    digitalWrite (led, HIGH);  
    digitalWrite (temp, HIGH);  
  }  
  else {  
    digitalWrite (led, LOW);  
    digitalWrite (temp, LOW);  
  }  
}
```

Slika 17: Program temperaturnega modula

### 5.5.4 Senzor barve

Senzor barve je sestavljen iz 64ih fotodiod, ki so v štirih skupinah po 16. Vsaka skupina ima drugačen filter. Prva skupina ima rdeč filter in tako zaznava rdečo svetlobo, druga ima zelenega, tretja pa modrega. Zadnja skupina 16-ih diod je brez filtra in zaznava celotno svetlobo, ki jo zaznavajo diode. Z izbiro filtriranih fotodiod lahko dobimo vrednosti posameznih barv. Filtre izbiramo z digitalnima priključkoma S2 in S3. S priključkoma S0 in S1, izbiramo kako vrednost nam bo senzor podal – 100%, 20% ali 2% vrednost originalne frekvence. Na sliki 20 lahko vidimo program za izpis vrednosti rdeče barve in spremembo filtra v zelen filter.

```

Serial.print("red=");
Serial.println(countR, DEC);
digitalWrite(s2, HIGH);
digitalWrite(s3, HIGH);

```

Slika 18: Odsek programa za senzor TCS3200

### 5.5.5 Raspberry Pi in prepoznavanje vizualnih žrtev

Vizualno zaznavanje žrtev bo sestavljeno iz dveh delov – Tesseract in OpenCV.

OpenCV je vtičnik za prepoznavo besedila v naravnih okoljih – za projekt ni nujno potreben, vendar vse skupaj optimizira in pospeši celotno branje besedila. Besedilo zazna in ga obreže tako, da imamo le izrez besedila oz. v našem primeru črke. Ta izrez tudi pravilno obrne in tako omogoči najboljše in najhitrejše delovanje Tesseract vmesnika.

Tesseract je sistem za prepoznavo črk in drugih optični znakov, ki ga je razvilo podjetje Google. Je vtičnik kompatibilen z mikroračunalnikom Raspberry Pi in je zato kot nalašč za ta projekt.

Na sliki 19 vidimo primer obdelave slike s pomočjo OpenCV sistema.

```

OpenCV OCR and text recognition with Tesseract
def decode_predictions(scores, geometry):
    # grab the number of rows and columns from the scores volume, then
    # initialize our set of bounding box rectangles and corresponding
    # confidence scores
    (numRows, numCols) = scores.shape[2:4]
    rects = []
    confidences = []

    # loop over the number of rows
    for y in range(0, numRows):
        # extract the scores (probabilities), followed by the
        # geometrical data used to derive potential bounding box
        # coordinates that surround text
        scoresData = scores[0, 0, y]
        xData0 = geometry[0, 0, y]
        xData1 = geometry[0, 1, y]
        xData2 = geometry[0, 2, y]
        xData3 = geometry[0, 3, y]
        angleData = geometry[0, 4, y]

        # loop over the number of columns
        for x in range(0, numCols):
            # if our score does not have sufficient probability,
            # ignore it
            if scoresData[x] < args["min_confidence"]:
                continue

            # compute the offset factor as our resulting feature
            # maps will be 4x smaller than the input image
            (offsetX, offsetY) = (x * 4.0, y * 4.0)

            # extract the rotation angle for the prediction and
            # then compute the sin and cosine
            angle = angleData[x]
            cos = np.cos(angle)
            sin = np.sin(angle)

            # use the geometry volume to derive the width and height
            # of the bounding box
            h = xData0[x] - xData2[x]
            w = xData1[x] - xData3[x]

            # compute both the starting and ending (x, y)-coordinates
            # for the text prediction bounding box
            endX = int(offsetX + (cos * xData1[x]) + (sin * xData2[x]))
            endY = int(offsetY - (sin * xData1[x]) + (cos * xData2[x]))
            startX = int(endX - w)
            startY = int(endY - h)

            # add the bounding box coordinates and probability score
            # to our respective lists
            rects.append((startX, startY, endX, endY))
            confidences.append(scoresData[x])

    # return a tuple of the bounding boxes and associated confidences
    return (rects, confidences)

```

Slika 19: Primer obdelave slike<sup>4</sup>

<sup>4</sup> Primer na sliki je iz spletne strani: <https://www.pyimagesearch.com/2018/09/17/opencv-ocr-and-text-recognition-with-tesseract/#>

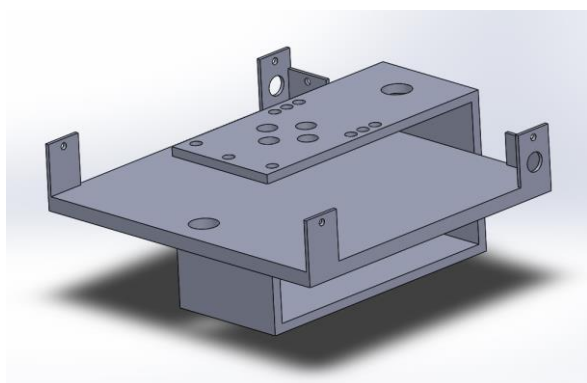
Ob zaznavi določene črke bo Raspberry PI uporabil enega svojih vhodno-izhodnih priključkov, da bo preko njega poslal vrednost »1«. To bo prebral glavni mikrokrmilnik, ki bo, odvisno od črke, poskrbel za pravo dejanje (spustil nobenega, enega ali pa dva paketa).

### 5.5.6 Zasnova celotnega programa

Program bo razdelil labirint na polja, v obliki kvadratov, katerih velikost se bo kalibrirala na začetku tekmovanja<sup>5</sup>. Premikal se bo po funkcijah naprej, nazaj, levo ali desno. Ob vsakem premiku naprej se bo premaknil za eno kocko in takrat preveril prisotnost sten. S tem si bo lahko izrisal mapo labirinta. To mu preprečuje večkratno zaznavanje, že zaznanih žrtev ali ponovno obiskovanje, že obiskanega polja. Tako bo lahko videl, katerih območij še ni raziskal. Lahko se bo tudi uspešno vrnil na začetno pozicijo. Termalne in vizualne žrtve bo iskal cel čas potovanja. Prav tako bo konstantno preverjal za prisotnost črnega polja.

## 5.6 3D modeliranje in ostali deli robota

Za robota sem izoblikoval ohišje v programu SolidWorks 2017. Sestavljeno je iz dveh delov. Prvi deluje kot ohišje za senzorje, motorje in ploščice; drugi pa služi kot ploščad z gumbi za kalibracijo in držalo za baterijo. Ohišje bi sicer lahko bilo v enem delu, vendar 3D printer, ki mi je bil na voljo, ni bil sposoben ustvariti kvalitetnega ohišja v enem kosu. 3D modela robota se poklopita v eno ohišje, ki je vidno na sliki 20.

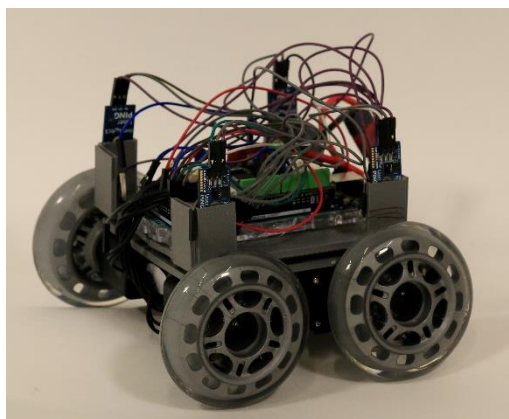


Slika 20: 3D model robota

---

<sup>5</sup> Vsako polje ima približno površino 30ih cm

Na sliki 21 je viden prototip robota, na sliki pa lahko vidimo tudi kolesa, ki sem jih izbral za robota. Ta kolesa sem izbral, saj so imela daleč najboljši oprijem. V primeru kakšnih drugih koles (npr. Mecanum) se oprijem izgubi in robot ne more prepeljati klanca.



Slika 21: Prototip robota

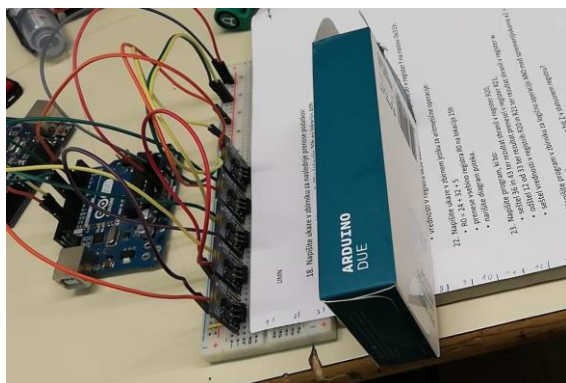
## 5.7 Izvedba

V teoriji stvar deluje odlično, vendar se v realnosti pojavijo marsikateri napake na katere se ne moreš pripraviti. Pojavljajo se problemi z dobavitelji (inventure, prazniki, počitnice itd.), zato vsi elementi niso prišli dovolj hitro, da bi lahko robota izpeljal do konca. Osredotočiti sem se tudi želel na postopno delo. V preteklih letih so bile težave, saj se je cel robot zgradil naenkrat, nato pa ni nič delovalo. Odločil sem se, da bom tokrat razvijal sisteme postopoma. Najprej bom dopolnil sistem za vožnjo in premikanje po labirintu, nato bom vključil termalne žrtve, šele zatem pa bom začel delati na ostalih delih robota.

Začel sem z razvijanjem sistema za vožnjo in po nekaj neuspešnih poskusih opazil napako v senzorjih. Niti en senzor ni deloval pravilno. V robota sem vključil 8 senzorjev, vsak od njih pa je »deloval s svojo glavo«. Kljub identični kodi so se meritve razlikovale od 1 do 5 cm. Na sliki 22 je vidno preizkušanje senzorjev.

Pet sem jih postavil v vrsto drug ob drugega, nato sem prednje postavil škatlico in preveril meritve senzorjev. Sumil sem, da bi lahko nastajali problemi tudi zaradi kota senzorjev ( $55^\circ$ ), zato sem dva izklopil in tako povečal medsebojno razdaljo senzorjev. Meritve so ostale enake.





Slika 22: Preizkus senzorjev

Ker senzorji niso delali, sem se odločil za uporabo ultrazvočnih senzorjev, ki so kompatibilni z kodo PING. Zato sem prisiljen spremeniti nekaj stvari, vendar verjamem, da mi bo pred tekmovanjem uspela pravočasna izvedba vsega.

## 5.8 Implementacija v resničnem življenju

Robot deluje na principih, ki bi se jih dalo uporabljati oz. implementirati v vse možne robote. Na svetu obstaja veliko podjetij, ki delajo le na robotu, ki se lahko premika po nevarnih območjih. Z implementacijo mojih sistemov, lahko ti roboti rešujejo življenja. Razdelitev prostora na kvadratna polja in s tem orientacija po njem, se lahko implementira v vsako hišo oz. v vsako območje nevarne nesreče.

Pri implementaciji v resničnem življenju, pa bi par stvari spremenil in dodal.

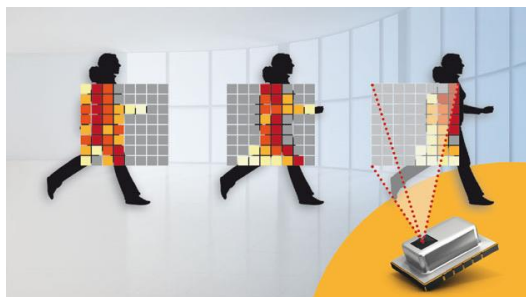
### 5.8.1 Spremembe in ideje pri implementaciji v resničnem življenju

#### 5.8.1.1 Zaznavanje ponesrečencev

Za uporabo sistema v realnem svetu moram najprej spremeniti zaznavanje ponesrečencev. Senzor, ki ga trenutno uporabljam je za tekmovanje zelo uporaben, vendar v resničnem svetu, ne bi uporaben, saj bi lahko vsako najmanjšo ogrevano telo zaznal kot človeka.

Pri robotu, ki bi raziskoval resnična območja nesreč, bi kombiniral temperaturni senzor Panasonic Grid eye z zaznavanjem in iskanjem obraza (na mikroračunalniku Raspberry Pi).

Panasonic Grid eye je temperaturni senzor, ki ima 64 pixlov. V vsakem lahko posebej zaznava višino temperature, zaradi tega pa lahko zaznava gibanje in silhuete oseb (glej sliko 23). Tako lahko senzor razlikuje med požarom ter človekom.



Slika 23: Panasonic Grid eye zaznavanje oseb

Uporaba tega senzorja je v osnovi že dovolj, vendar sem želel vnesti dodatno varovalo. Na mikroračunalniku Raspberry Pi je možna zaznava obrazov z njihovo kamero. Kombinacija tega sistema s temperaturnim senzorjem Grid eye, bi ustvarilo odlično avtomatizirano zaznavanje človeških ponesrečencev.

Če bi se izkazalo, da zaznava obrazov ni natančna bi mikroračunalnik naredil sliko, na kateri bi lahko reševalci videli, če je senzor res zaznal ponesrečenca. S potrditvijo oz. zavračanjem žrtve, bi lahko žrtev izbrisali iz mape. Ob sliki bi lahko reševalci tudi določili prioriteto žrtve. Če žrtev krvavi ali je v kritičnem stanju lahko žrtev prioritiziramo in se tako posvetili najprej tistim, ki so pomoči najbolj nujno potrebni.

### 5.8.1.2 Senzor plina

V robota želim dodati senzor plina, s katerim bi robot zaznaval pline prisotne v ozračju, te podatke bi sporočil reševalcem, ti pa bi se tako lahko pravilno pripravili. Območja z vnetljivimi plini, bi označili z večjo prioriteto pri reševanju. Takšnih senzorjev je zelo veliko, vsak od njih zaznava različne pline, ampak so ustvarjeni tako, da dobimo nazaj vrednosti preko analognih priključkov mikrokrmilnika Arduino – to pomeni, da jih lahko na mikrokrmilnik priključimo več, s tem pa lahko dobimo zelo natančne podatke o stanju in nevarnosti v sobi oz. nevarnem območju. Primer takšnega senzorja lahko vidimo na sliki 24.



Slika 24: MQ-9<sup>6</sup>

### 5.8.1.3 Komunikacija z reševalci

V robotu, ki bi deloval v resničnem svetu, bi seveda moral vključiti brezžično povezavo, s katero lahko mapo, ki jo je robot ustvaril nekako posreduje reševalcem. Preko te povezave, bodo reševalci prav tako dobivali podatke o plinih v ozračju in o morebitnih požarih. Robot jim bi tudi pošiljal slike ponesrečencev, preko katerih lahko zagotovimo, da se robot ni zmotil ali preverimo stanje žrtev.

Velik problem pri brezžični komunikaciji je oddaljenost robota, zaradi nje se brezžična povezava pogosto prekinja, prav tako pa se pojavlja tudi problem ovir – če je robot v objektu z veliko stenami bo skozi stene brezžično povezavo težko obdržal.

Jaz bi uporabil komunikacijski standard LoRaWAN<sup>7</sup>. Ta je sestavljen iz LPWAN (Low Power Wide Area Network). Sistem uporablja nižjo frekvenco signalov – nižje frekvence lažje potuje skozi stene, istočasno pa ima zelo velik obseg, zato je idealna za takšnega robota. Z eno samo bazno postajo lahko pokrijemo tudi do  $16\text{ km}^2$ . Zaenkrat se uporablja predvsem v namene IoT (internet stvari) omrežij, vendar verjamem, da bi lahko bil uporaben tudi tukaj. Čipi so zelo majhni, kar omogoča preprosto implementacijo v robota katerekoli velikosti.

### 5.8.2 Implementacija na različnih robotih

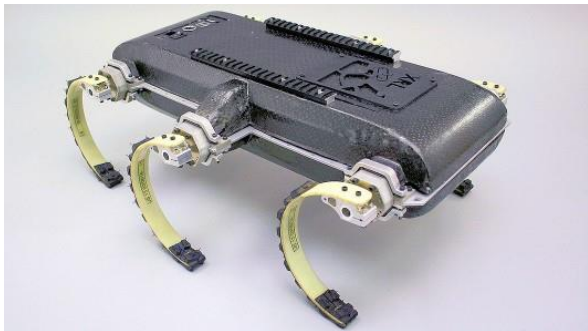
S temi spremembami imam odličen sistem za optimiziranje reševalnih akcij vseh vrst. Na žalost pa drži tudi to, da robot v trenutnem stanju in s trenutnim načinom premikanja ni pripravljen za resnična nevarna območja. Ampak to ni bil namen mojega inovacijske predloga. Sestaviti sem želel rešitev, ki bi se jo dalo implementirati marsikje. Na spodnjih slikah so primeri

---

<sup>6</sup> Senzor za zaznavo ogljikovega monoksida in vnetljivih plinov MQ-9. Proizvaja ga podjetje pololu

<sup>7</sup> Več o tem si lahko preberete na spletni strani: <http://iot.telos.si/kaj-je-lora/>

robotov, v katerih lahko ta sistem uporabimo. Roboti na teh slikah so bili zgrajeni le za raziskovanje premikanja po neidealnih terenih – sneg, gozd, območja naravnih nesreč itd.



Slika 25: XRL<sup>8</sup>



Slika 26: RSTAR<sup>9</sup>



Slika 27: Velox<sup>10</sup>



Slika 28: QUINCE<sup>11</sup>

Ti roboti lahko v kombinaciji z mojim sistemom rešujejo življenja. Z raziskovanjem območja nesreče in izrisom natančne mape, ki jo posredujejo reševalcem, lahko olajšajo in učinkovito optimizirajo reševalne akcije. Z avtomatizirano prioritizacijo žrtev in sob oz. območij se lahko

---

<sup>8</sup> Robot X-RHex Lite (XLR) je robot, ki se lahko pomika po vsakšnjem težavnem terenu – ustvarila ga je skupina KOD\*LAB

<sup>9</sup> RSTAR je robot, ki lahko spreminja svojo obliko in se s tem premika skozi ozke špranje ali pa po neravnem terenu vseh vrst. Ustvarili so ga pri GRASP laboratoriju

<sup>10</sup> Velox je robot, ki oponaša biologijo narave. Njegovo premikanje je bilo ustvarjeno po zgledu stonoge, kače in lignjev. Premika se lahko po snegu, pesku, v vodi ali pa na suhem. Ustvarili so ga pri podjetju Pliant energy systems.

<sup>11</sup> QUINCE je en primer pogostih sistemov premikanja. Premika se lahko po vseh možnih terenih, prav tako pa se lahko premika po stopnicah. Ustvarili so ga raziskovalci iz raziskovalne ekipe iz mednarodnega inštituta za reševalne sisteme

reševalci posvetijo tistim, ki so pomoči najbolj nujno potrebni. Če nosijo roboti s seboj reševalne pakete, pa čakanje olajšajo vsem ponesrečencem.

Uporabljajo se lahko tudi v mestih, katere so prizadeli potresi. V takšne robote, bi lahko dodali tudi GPS oddajnike, ki bi delo reševalcem še dodatno olajšali. Prevažajo lahko tudi življenjske potrebščine, s katerim lahko tistim, ki so ujeti v stavbah nedostopnim ljudem življenja rešijo tudi brez pomoči reševalcev.

## 6 ZAKLJUČEK

V času raziskovanja sem se spoznal z veliko novimi stvarmi. Prvič sem se tudi seznanil mikroračunalnikom Raspberry Pi. Spoznal sem veliko novih senzorjev in nove načine komunikacije.

Ugotovil sem, da bo moja vizijo težko pripeljati do konca. Kot elektrotehnik sem si zadal zelo težko nalogo, ki jo moram hitro izpeljati do konca. Zato se mi je pridružil programer, ki mi bo robota za namene tekmovanja pomagal izpeljati karseda kvalitetno. S tem, pa bo razvit tudi del sistema na katerem lahko delam in gradim, tako pa razvijem robota oz. sistem, ki bo v resničnem svetu rešil nešteto življenj.

Ta inovacijski predlog me je seznanil z veliko težavami na tem svetu. Spoznal sem kruto resničnost in sem jo poskusil olepšati. Ko bom z končanim izdelkom popolnoma zadovoljen se bom poskusil povezati s svetovnimi organizacijami in različnimi proizvajalci robotov ter tako pripeljati moj sistem v realnost, razširiti pa ga želim po svetu. Verjamem, da lahko reši in olajša življenja mnogih ljudi. Takšni roboti so idealni za veliko različnih okoliščin. Uporabimo jih lahko v stavbah, kjer so se zgodile razne nesreče, v mestih katere so zadele velike naravne katastrofe (cunamiji, potresi ipd.), uporabljamo jih lahko v vojnih območjih oz. mestih katere so vojne hudo prizadele. Z roboti skavti, ki bi uporabljali ta sistem so potenciali skorajda neomejeni.

## 7 DRUŽBENA ODGOVORNOST

Sistem, ki ga razvijam bo rešil veliko življenj. Pred sistemi, ki obstajajo ima veliko prednosti. Roboti, ki uporabljajo moj sistem so lahko zelo mali in lahki, kar jih pomeni, da je njihov prenos zelo lahek. Vsaka reševalno vozilo lahko ima veliko takšnih robotov brez da bi se z njimi zaseden prostor pogrešal. Zdajšnji reševalni roboti pogosto potrebujejo posebno prikolico, ki je namenjena posebej prevozu le njih – to povzroča tudi dodatne stroške. Velikost robotov skavtov in s tem njihova mobilnost omogoča njihovo uporabo tudi navadnim gasilcem.

Roboti z mojim sistemom so lahko zelo poceni, kar povzroča njihovo dostopnost po celem svetu. Zaradi nizke cene se bodo vse države strinjale da je človeško življenje bolj dragoceno.

Zaradi tega, ker ne potrebujejo človeškega upravljalca in s tem dvostranske povezave, ter zaradi delovanja, ki ne vključuje prevažanja človeških teles, roboti porabijo manj energije. S tem se znižajo stroški in podaljša njihovo delovanje.

Takšni roboti ne ogrožajo človeških življenj (ne poskušajo jih premikati ali dvigovati), kar zmanjša tveganja in omogoči delovanje brez nevarnosti. Delujejo brez stalnega nadzora, vendar ga omogočajo. Če želijo reševalci pogledati skozi »oči robota« lahko ob kateremkoli trenutku pogledajo sliko iz kamere na robotu, lahko pa prevzamejo tudi popoln nadzor nad robotom in tako raziščejo območje, ki jih zanima. To je zaradi uporabljene vrste komunikacije možno tudi, če je robot oddaljen več kilometrov.

Sestavil sem sistem, ki se ga da implementirati oz. uporabiti. To sem naredil, saj verjamem, da je na današnjem svetu veliko težav na katere jaz sploh nisem pomislil. S tem omogočam ljudem, ki za te težave vedo, da ustvarijo robota, ki bo najbolj ustrezal njihovim okoliščinam. Vse kode in celotna rešitev bodo objavljeni in bodo delovali na odprtokodni zasnovi. S tem lahko na sistemu gradijo vsi in ga izboljšujejo.

## 8 VIRI

### 8.1 Spletne strani:

<https://www.arduino.cc/en/tutorial/ping> 18.10.2018

[https://www.dfrobot.com/wiki/index.php/Digital\\_Servo\\_Shield\\_for\\_Arduino\\_SKU:DRI0027](https://www.dfrobot.com/wiki/index.php/Digital_Servo_Shield_for_Arduino_SKU:DRI0027)  
20.10.2018

[https://www.dfrobot.com/wiki/index.php/TCS3200\\_Color\\_Sensor\\_\(SKU:SEN0101\)](https://www.dfrobot.com/wiki/index.php/TCS3200_Color_Sensor_(SKU:SEN0101))

<https://howtomechatronics.com/tutorials/arduino/arduino-color-sensing-tutorial-tcs230-tcs3200-color-sensor/>

<https://raspberrypi.stackexchange.com/questions/88265/how-to-get-the-pi-to-recognize-numbers-and-letters-from-images>

<https://www.pyimagesearch.com/2018/08/20/opencv-text-detection-east-text-detector/>

<https://www.pyimagesearch.com/2018/09/17/opencv-ocr-and-text-recognition-with-tesseract/#>

<https://eu.industrial.panasonic.com/video/grid-eye-passive-infrared-array-sensor>

<https://howtomechatronics.com/tutorials/arduino/arduino-color-sensing-tutorial-tcs230-tcs3200-color-sensor/>

<https://randomnerdtutorials.com/arduino-color-sensor-tcs230-tcs3200/>

<https://www.hackster.io/mjrobot/arduino-color-detection-57e4ce>

<https://www.instructables.com/id/Everything-you-need-to-know-about-colour-sensors/>

<https://www.arduino.cc/en/tutorial/ping>

<http://www.robaid.com/robotics/quince-search-and-rescue-robot-developed-in-japan.htm>

[https://www.eurekalert.org/pub\\_releases/2018-07/aabu-ncc071718.php](https://www.eurekalert.org/pub_releases/2018-07/aabu-ncc071718.php)



<http://lifestyle.co/this-unusual-nature-inspired-robot-is-equally-at-home-on-land-or-in-the-water/>

<https://sites.google.com/site/bioinspiredrob/crawling-running-wriggling-robots>

<https://www.grasp.upenn.edu/projects/x-rhex-lite-xrl>

<https://kodlab.seas.upenn.edu/past-work/rhex/xrl/>

<https://www.pliantenergy.com/home-1/>

<https://www.mysensors.org/build/gas>

<https://www.intechopen.com/books/search-and-rescue-robotics-from-theory-to-practice/tactical-communications-for-cooperative-sar-robot-missions>

<http://iot.telos.si/kaj-je-lora/>

<https://lora-alliance.org/>

<https://sl.wikipedia.org/wiki/LoRa>

Vse spletne strani so bile nazadnje obiskane 9.02.2019

## 8.2 Dokumenti na spletu

[https://ws680.nist.gov/publication/get\\_pdf.cfm?pub\\_id=32584](https://ws680.nist.gov/publication/get_pdf.cfm?pub_id=32584)

<https://www.sparkfun.com/datasheets/Sensors/Temperature/SEN-09570-datasheet-3901090614M005.pdf>

<https://www.parallax.com/sites/default/files/downloads/28041-LaserPING-2m-RangeFinder-Guide.pdf>

Vsi spletni dokumenti so bili obiskani 9.02.2019

## 8.3 Viri slik

Slika 1:

[https://www.google.com/search?q=rescue+robot&safe=off&source=lnms&tbm=isch&sa=X&ved=0ahUKEwjUuuuDm47gAhWH6aQKHTxbC7sQ\\_AUIDigB&biw=1680&bih=908#imgsrc=vpbjqGEAb38JM:](https://www.google.com/search?q=rescue+robot&safe=off&source=lnms&tbm=isch&sa=X&ved=0ahUKEwjUuuuDm47gAhWH6aQKHTxbC7sQ_AUIDigB&biw=1680&bih=908#imgsrc=vpbjqGEAb38JM:)

Slika 2:

[https://www.google.com/search?q=rescue+robot&safe=off&source=lnms&tbm=isch&sa=X&ved=0ahUKEwjUuuuDm47gAhWH6aQKHTxbC7sQ\\_AUIDigB&biw=1680&bih=908#imgsrc=zFe9Jqme-dIgPM:](https://www.google.com/search?q=rescue+robot&safe=off&source=lnms&tbm=isch&sa=X&ved=0ahUKEwjUuuuDm47gAhWH6aQKHTxbC7sQ_AUIDigB&biw=1680&bih=908#imgsrc=zFe9Jqme-dIgPM:)

Slika 3:

[https://www.google.com/search?q=rescue+robot&safe=off&source=lnms&tbm=isch&sa=X&ved=0ahUKEwjUuuuDm47gAhWH6aQKHTxbC7sQ\\_AUIDigB&biw=1680&bih=908#imgsrc=VyhOfxiZAEQAI:](https://www.google.com/search?q=rescue+robot&safe=off&source=lnms&tbm=isch&sa=X&ved=0ahUKEwjUuuuDm47gAhWH6aQKHTxbC7sQ_AUIDigB&biw=1680&bih=908#imgsrc=VyhOfxiZAEQAI:)

Slika 4:

[https://www.google.com/search?q=rescue+maze&safe=off&tbm=isch&tbs=rimg:CUNz3PTpt\\_15AIjiMqoI4nECYWyzSmhZsXihLXf4dQcUB3V4jx6b0zBHxKEia7eTuov2DzMVReMpE9RZTLjvf4CRXeyoSCcyqgjcQJhbESBE2YoK9qJHKhIJLNKaFmxeKEsRh3lr-5nRjAsqEgld\\_1h1BxQHdXhHskFPLhqeTcyoSCSPHPvTMEdcoERSykhFEh0UKhIJSJrt5O6i\\_1YMRaY1iRRvxBGMqEgnMxVF4ykT1FhH\\_14-GBwroFgioSCVMuO9\\_1gJFd7ESCE8\\_1-YhI0U&tbo=u&sa=X&ved=2ahUKEwi3kLXsyY7gAhWS6qQKHdBHCEkQ9C96BAgBEBs&biw=1680&bih=908&dpr=1#imgsrc=Uy473-AkV3verM:](https://www.google.com/search?q=rescue+maze&safe=off&tbm=isch&tbs=rimg:CUNz3PTpt_15AIjiMqoI4nECYWyzSmhZsXihLXf4dQcUB3V4jx6b0zBHxKEia7eTuov2DzMVReMpE9RZTLjvf4CRXeyoSCcyqgjcQJhbESBE2YoK9qJHKhIJLNKaFmxeKEsRh3lr-5nRjAsqEgld_1h1BxQHdXhHskFPLhqeTcyoSCSPHPvTMEdcoERSykhFEh0UKhIJSJrt5O6i_1YMRaY1iRRvxBGMqEgnMxVF4ykT1FhH_14-GBwroFgioSCVMuO9_1gJFd7ESCE8_1-YhI0U&tbo=u&sa=X&ved=2ahUKEwi3kLXsyY7gAhWS6qQKHdBHCEkQ9C96BAgBEBs&biw=1680&bih=908&dpr=1#imgsrc=Uy473-AkV3verM:)

Slika 5:

[https://store-cdn.arduino.cc/uni/catalog/product/cache/1/image/500x375/f8876a31b63532bbba4e781c30024a0a/a/0/a000067\\_front\\_1\\_.jpg](https://store-cdn.arduino.cc/uni/catalog/product/cache/1/image/500x375/f8876a31b63532bbba4e781c30024a0a/a/0/a000067_front_1_.jpg)

Slike 6-8:

Avtor naloge

Slika 9:

[https://www.google.com/search?q=laserping&safe=off&rlz=1C1GGRV\\_enSI752SI753&source=lnms&tbm=isch&sa=X&ved=0ahUKEwiA1Pywq5fgAhXRZFAKHrtvCRAQ\\_AUIDigB&biw=1600&bih=758#imgsrc=ShwBW0qiT7hPXM:](https://www.google.com/search?q=laserping&safe=off&rlz=1C1GGRV_enSI752SI753&source=lnms&tbm=isch&sa=X&ved=0ahUKEwiA1Pywq5fgAhXRZFAKHrtvCRAQ_AUIDigB&biw=1600&bih=758#imgsrc=ShwBW0qiT7hPXM:)

Slika 10:

<https://cdn.sparkfun.com/assets/parts/3/3/5/1/09570-01.jpg>

Slika 11:

[https://www.google.com/search?q=tcs3200&safe=off&rlz=1C1GGRV\\_enSI752SI753&tbm=isch&source=iu&ictx=1&fir=vDHgTJ8mlyBVeM%253A%252CILf\\_1BXv2d-mUM%252C\\_&usg=AI4\\_-kTYnSYiKGL77lpXCRr73aqF9lqhyQ&sa=X&ved=2ahUKEwjtpC-spfgAhWQb1AKHWFMCwcQ9QEwBHoECAAQCA#imgsrc=vDHgTJ8mlyBVeM:](https://www.google.com/search?q=tcs3200&safe=off&rlz=1C1GGRV_enSI752SI753&tbm=isch&source=iu&ictx=1&fir=vDHgTJ8mlyBVeM%253A%252CILf_1BXv2d-mUM%252C_&usg=AI4_-kTYnSYiKGL77lpXCRr73aqF9lqhyQ&sa=X&ved=2ahUKEwjtpC-spfgAhWQb1AKHWFMCwcQ9QEwBHoECAAQCA#imgsrc=vDHgTJ8mlyBVeM:)

Slika 12:

[https://www.google.com/search?q=raspberry+pi+camera&safe=off&source=lnms&tbm=isch&sa=X&ved=0ahUKEwiU6Yby0JjgAhVGIIAKHX6aB\\_gQ\\_AUIDigB&biw=1680&bih=908#imgsrc=IxUqod48LTP7uM:](https://www.google.com/search?q=raspberry+pi+camera&safe=off&source=lnms&tbm=isch&sa=X&ved=0ahUKEwiU6Yby0JjgAhVGIIAKHX6aB_gQ_AUIDigB&biw=1680&bih=908#imgsrc=IxUqod48LTP7uM:)

Slika 13:

[https://images-na.ssl-images-amazon.com/images/I/614%2Brmf38hL.\\_SL1001\\_.jpg](https://images-na.ssl-images-amazon.com/images/I/614%2Brmf38hL._SL1001_.jpg)

Slika 14: [https://images-na.ssl-images-amazon.com/images/I/81DbLzpiB5L.\\_SL1200\\_.jpg](https://images-na.ssl-images-amazon.com/images/I/81DbLzpiB5L._SL1200_.jpg)

Slika 15: [https://images-na.ssl-images-amazon.com/images/I/71FU1KkRGyL.\\_SL1200\\_.jpg](https://images-na.ssl-images-amazon.com/images/I/71FU1KkRGyL._SL1200_.jpg)

Slika 16:

[https://api.qrserver.com/v1/create-qr-code/?data=Ime+in+priimek%0AVarnostni+kontakt+\(%C5%A1tevilka\)%0AAlergije%3A%0A-%0A-%0ABolezni%3A%0A-&size=220x220&margin=0](https://api.qrserver.com/v1/create-qr-code/?data=Ime+in+priimek%0AVarnostni+kontakt+(%C5%A1tevilka)%0AAlergije%3A%0A-%0A-%0ABolezni%3A%0A-&size=220x220&margin=0)

Slika 20: [https://http2.mlstatic.com/sensor-touch-toque-capacitivo-fc-91-ttp223-arduino-pic-D\\_NQ\\_NP\\_120825-MLB25497429377\\_042017-F.jpg](https://http2.mlstatic.com/sensor-touch-toque-capacitivo-fc-91-ttp223-arduino-pic-D_NQ_NP_120825-MLB25497429377_042017-F.jpg)

Sliki 21 in 22: Avtor naloge

Slika 23: [https://www.google.com/url?sa=i&source=images&cd=&cad=rja&uact=8&ved=2ahUKEwikjIjOwqngAhVS2KQKHdRgBEEQjRx6BAgBEAU&url=https%3A%2F%2Fwww.industrial.panasonic.com%2Fproducts%2Fsensors-optical-devices%2Fsensors-automotive-and-industrial-applications%2Finfrared-array&psig=AOvVaw2gItFIDDU2E5Z6XpzxJ\\_8f&ust=1549625387032516](https://www.google.com/url?sa=i&source=images&cd=&cad=rja&uact=8&ved=2ahUKEwikjIjOwqngAhVS2KQKHdRgBEEQjRx6BAgBEAU&url=https%3A%2F%2Fwww.industrial.panasonic.com%2Fproducts%2Fsensors-optical-devices%2Fsensors-automotive-and-industrial-applications%2Finfrared-array&psig=AOvVaw2gItFIDDU2E5Z6XpzxJ_8f&ust=1549625387032516)

Slika 24: <https://www.robotistan.com/carbon-monoxide-flammable-gas-sensor-mq-9-4153-17-B.jpg>

Slika 25: <http://images.gizmag.com/inline/x-rhex-xrl-robot-jumping-kodlab-upenn-0.jpg>

Slika 26: [https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcTso54OclfOCXcQxC-9ntix\\_XG7usMYFGkp8aK\\_kK9u-i\\_t9](https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcTso54OclfOCXcQxC-9ntix_XG7usMYFGkp8aK_kK9u-i_t9)

Slika 27: [http://lifestyle.co/wp-content/uploads/2018/12/1\\_pliant\\_energy-640x640.jpg](http://lifestyle.co/wp-content/uploads/2018/12/1_pliant_energy-640x640.jpg)

Slika 28: <http://www.robaid.com/wp-content/gallery/robots3/quince-search-and-rescue-robot.jpg>

Vse slike so bile dosegljive na spletu dne 9.2.2019