

ŠOLSKI CENTER CELJE



Srednja šola za strojništvo, mehatroniko in medije

Raziskovalna naloga

# AVTONOMNI REŠEVALNI ROBOT

Avtorja:

Luka TOMŠIČ, M-4. c

Dominik HRASTNIK, M-4. c

Mentorji:

mag. Andro GLAMNIK, univ. dipl. inž.

mag. Matej VEBER, univ. dipl. inž.

Robert OJSTERŠEK, mag. inž.

Mestna občina Celje, Mladi za Celje

Celje, 2017

## **POVZETEK**

V današnjem času se vse več pozornosti namenja razvoju in izdelavi avtonomnih strojev in naprav, ki bi človeku olajšali delo ter ga nadomestili v morebitnih nevarnih situacijah. Lep zgled nam daje večina avtomobilskih korporacij z avtomobili, pri katerih za upravljanje ni potreben človeški faktor. Vso tovrstno tehnologijo je mogoče najti tudi v robotiki, ki nam je vzbudila zanimanje za koncept avtonomnega reševalnega vozila. Raziskovalni učni projekt predstavlja konstruiranje in avtomatizacijo robota, ki se mora držati določenih pravil.

Ključne besede: avtonomnih strojev, olajšali delo, avtonomnega reševalnega vozila, avtomatizacijo robota

## KAZALO VSEBINE

1	UVOD .....	1
1.1	PREDSTAVITEV .....	1
1.2	HIPOTEZE.....	1
1.3	METODE RAZISKOVANJA.....	2
2	UPORABA ROBOTOV .....	3
2.1.1	INDUSTRIJA.....	3
2.1.2	NAPREDNEJŠA ZNANSTVENA PODROČJA .....	3
2.1.3	NEVARNO OKOLJE .....	4
2.1.4	IZOBRAŽEVALNI NAMEN .....	4
3	PREDSTAVITEV RAZISKOVALNE NALOGE.....	5
3.1	POVOD .....	5
3.2	IZDELAVA NOVEGA MOBILNEGA ROBOTA .....	8
3.2.1	KONSTRUKCIJA.....	9
3.3	PROGRAM.....	12
3.3.1	RASPBERRY PI.....	12
3.3.1.1	RASPBERRY PI-PROGRAM.....	13
3.3.2	ARDUINO .....	16
3.3.2.1	ARDUINO DUE .....	17
3.3.2.2	ARDUINO PROGRAM.....	19
4	ZAKLJUČEK.....	20
5	ZAHVALA.....	22
6	VIRI.....	23
7	IZJAVA.....	24

## KAZALO SLIK

Slika 1: Dotik z robotiko .....	2
Slika 2: Uporaba robotov .....	3
Slika 3: Uporaba industrijskega robota v učne namene .....	4
Slika 4: ROBOCUP JUNIOR .....	5
Slika 5: Lego robot .....	6
<i>Slika 6: Končen lego robot</i> .....	7
Slika 7: Izdelava konstrukcije .....	8
Slika 8: Osnovna konstrukcija.....	9
Slika 9: Sestavni deli mobilnega robota .....	10
Slika 10: Mobilni robot – prikaz s spodnje strani .....	11
Slika 11: Mobilni robot – prikaz z zgornje strani.....	12
Slika 12: Program RPI.....	14
Slika 13: Program RPI.....	15
Slika 14: Raspberry PI 3B .....	16
Slika 15: Arduino due z adafruit motor shieldom .....	19

## KAZALO TABEL

Tabela 1: Specifikacija Arduina .....	17
Tabela 2: Potrditev hipotez .....	20

## **1 UVOD**

### **1.1 PREDSTAVITEV**

Robot je stroj, ki ga nadzoruje računalnik in ga lahko programiramo, da samostojno izvaja določena opravila. Pogosto ga uporabljamo v industriji za prenašanje materiala ali za izvajanje ponavljajočih se opravil. Nekateri roboti so opremljeni s tipali, na primer za dotik ali svetlobo, in so sprogramirani tako, da lahko na osnovi njihovih podatkov sprejemajo enostavne odločitve. Seveda se pojavlja vse več avtonomnih robotov, ki brez človeške pomoči spremljajo okolico in se ji glede na določene vhodne signale ustrezno prilagajajo. Z besedno zvezo mobilni robot pa označujemo robote, ki jih je mogoče prestaviti iz točke A v točko B oziroma je tega dejanja sposoben sam, po prej napisanem algoritmu. Hkrati pa nam roboti kot glavne attribute predstavljajo olajšanje našega vsakdanjega življenja, delujejo v pogojih, v katerih ljudje ne morejo, in kar je najpomembneje, zagotavljajo nam veliko natančnost pri velikem številu ponovitev.

### **1.2 HIPOTEZE**

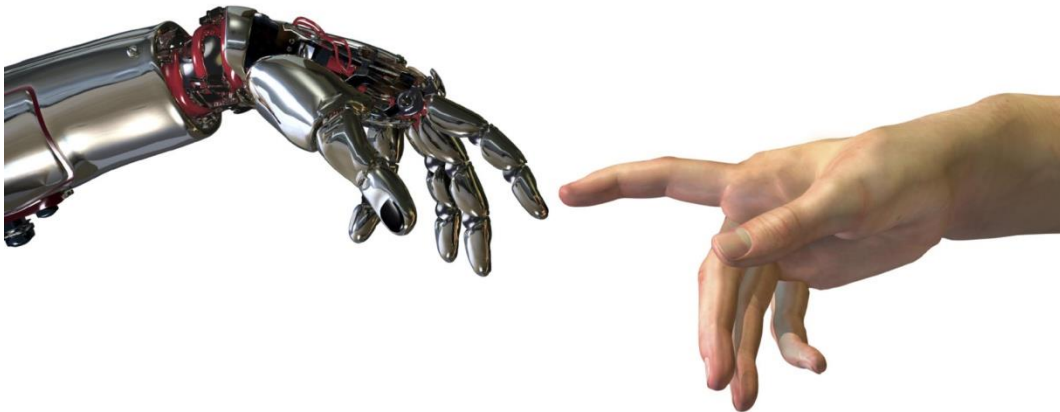
Cilj najine raziskovalne naloge je zasnovati in izdelati mobilni robot, ki bo najbolje ustrezal vsem kriterijem tekmovanja. V ta namen sva si zastavili ustrezne hipoteze. Z njihovo pomočjo želiva dokazati, da lahko s pomoči tehnologije 3D-tiskanja izdelava funkcionalen sklop. Hkrati pa želiva spoznati delovanje drugih mobilnih robotov in na podlagi tega zasnovati ter izdelati najprimernejši mobilni robot.

Za nadaljnje raziskovanje smo si zastavili sledeče hipoteze:

- S primerjanjem različnih mobilnih robotov pridemo do najprimernejše rešitve.
- Robot bo ustrezal vsem zastavljenim pogojem.
- Z uporabo razvojnih plošč Arduino in Raspberry so nama omogočene širše možnosti pri razvoju programa.
- Tiskanje delov bo najhitrejša in najcenejša metoda izdelovanja konstrukcije.
- Celotna izdelava ne bo predraga.

### 1.3 METODE RAZISKOVANJA

V raziskovalni nalogi sva se posluževala dveh metod raziskovanja: metode analize, katere namen je razčleniti neko celoto na njene osnovne gradnike. To sva uporabili pri razčlenitvi celotnega mobilnega robota. Primerjalna metoda pa nama je služila za primerjanje različnih mobilnih robotov.

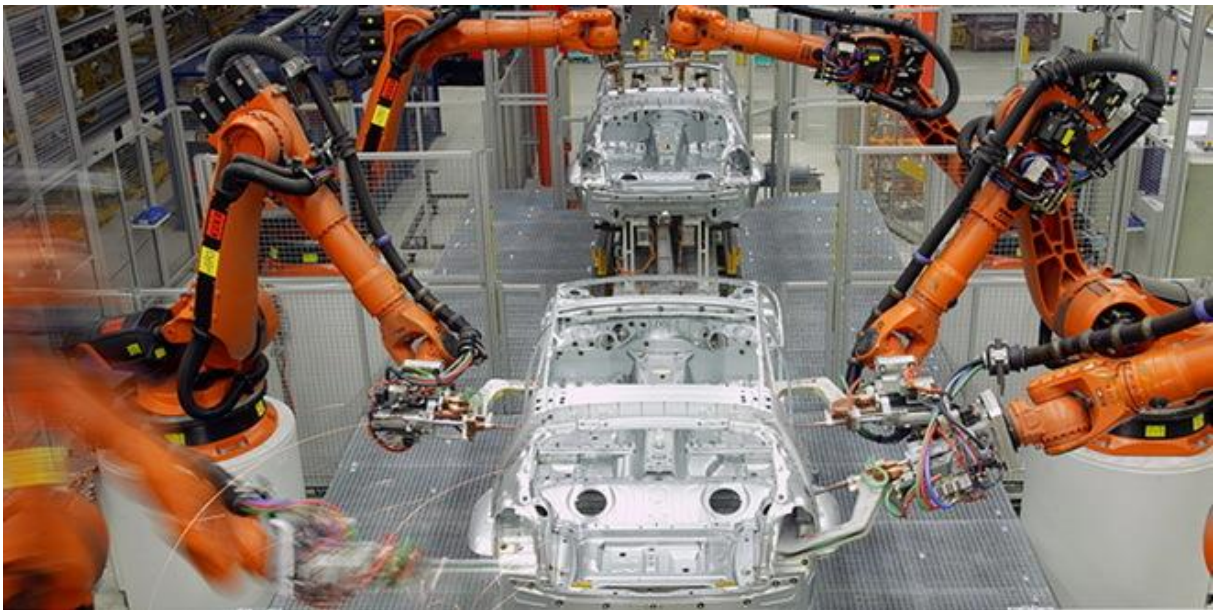


*Slika 1: Dotik z robotiko*  
(<http://rcj.robocup.org/>)

## 2 UPORABA ROBOTOV

### 2.1.1 INDUSTRIJA

Robote uporabljamo vse od preprostih nalog, kot so prelaganje stvari, do naprednejših, namenjenih natančnemu skeniranju, preciznemu gibanju, ki se lahko najdejo v izdelovalni industriji.



*Slika 2: Uporaba robotov*

*(<https://www.oelcheck.de/en/>)*

### 2.1.2 NAPREDNEJŠA ZNANSTVENA PODROČJA

Kjer je človeško gibanje omejeno, na primeru v atmosferi drugega planeta, se pojavijo »bitja«, sposobna tam delati in živeti. Ob omejitvi poletov astronautov v zunanje vesolje so morali znanstveniki odkriti način, da bi lahko raziskovali in zbirali informacije na drugih planetih. Rešitev so našli v mobilnih robotih. Kot primer lahko vzamemo rover Curiosity, ki že dober čas deluje na drugem planetu. Sposoben samodejnega napajanja s sončnimi celicami, konstantnim oddajanjem in sprejemanjem signalov, premikanja in analiziranja površine je eden izmed najnaprednejših izdelanih mobilnih robotov.



### **2.1.3 NEVARNO OKOLJE**

Okolja, ki človeku predstavljajo potencialno nevarnost, je vedno mogoče doseči s pomočjo ustreznih robotov. Seveda je veliko lažje prilagoditi robota na požar, bližino eksploziv, visoko radiacijo in pritisk kot izpostavljati tem nevarnim faktorjem človeka. Tako lahko robote vodimo indirektno preko zunanega kontakta s krmilnikom ali pa so popolnoma avtonomni. Z njimi lahko človek posreduje v prej naštetih prostorih brez svoje neposredne prisotnosti.

### **2.1.4 IZOBRAŽEVALNI NAMEN**

Različne oblike in primere robotov je moč najti tudi v šolskem okolju, kjer primarno služijo za izobraževanje dijakov. Na njih se dijaki učijo delovanja robotov ter njihovih funkcij, ki jih lahko z dodatno nadgradnjo izboljšajo.



*Slika 3: Uporaba industrijskega robota v učne namene*

*(osebni vir)*

### 3 PREDSTAVITEV RAZISKOVALNE NALOGE

#### 3.1 POVOD

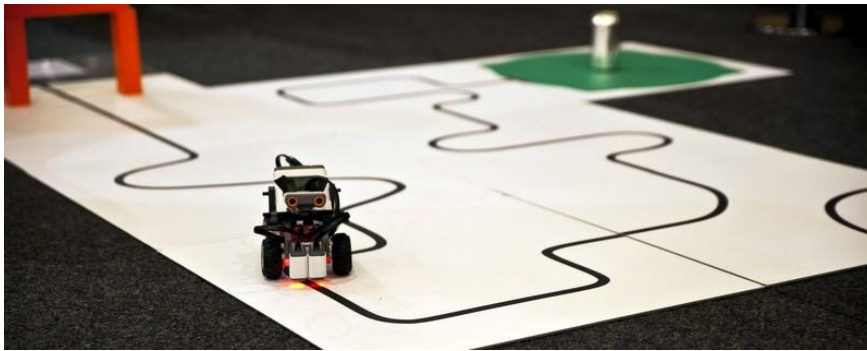
Raziskovalna naloga je nastala kot rezultat udeležbe na tekmovanju, ki od udeležencev zahteva izdelavo robota, optimiziranega po določenih pravilih oziroma kriterijih. Vse skupaj se je začelo z izdelavo robota na podlagi krmilnika LEGO MINDSTORM in z udeležbo na tekmovanju ROBOCUP JUNIOR.



Slika 4: ROBOCUP JUNIOR

(<http://rcj.robocup.org/>)

RoboCup Junior je tekmovanje, na katerem mora robot premagati poligon oziroma progo, sestavljeno iz različnih nalog in ovir. Celotno progo lahko razdelimo na dva dela. V prvem delu ima robot nalogo, da se pelje po poti, označeni s črnim in z zelenim lepilnim trakom, širine od 2 cm do 3 cm, se izogiba različnim oviram, preko katerih se mora zapeljati oziroma jih obvoziti. Če mobilni robot uspešno premaga vse zadane ovire, ga proga pripelje do vhoda v prostor oziroma areno, kjer poteka drugi del tekmovanja – reševanje žrtev. V areni mora najti tako imenovane žrtve. Le-te nam predstavljajo srebrne kroglice, premera 50 mm. Pred začetkom vsakega kroga tekmovanja se izžreba, v katerem kotu v areni bo postavljeno odlagališče, prav tako pa se v areno kroglice spustijo naključno. Seveda se tako iskanje žrtev oteži in zakomplicira.



*Slika 5: Lego robot*

*(osebni vir)*

V šolskem letu 2015/16 smo se nekateri dijaki tretjega letnika programa tehnik mehatronike odločili za sodelovanje pri interesni dejavnosti, imenovani Mobilni roboti. V sklopu le-te smo pod vodstvom Roberta Ojsterška in Mateja Vebra začeli s preprosto lekcijo o mobilnih robotih. Že na začetku je bil vsem skupen interes, da s pomočjo setov Lego Mindstorm in krmilnika EV3 začnemo spoznavati delovanje mobilnih robotov ter jih tudi sami izdelamo. Za uspešnejšo realizacijo idej smo se razdelili v tri manjše skupine in tako preizkušali različne ideje ter koncepte. Seveda smo še vedno aktivno sodelovali, si delili programe, zamisli in izkušnje ter s tem najhitreje prišli do rezultatov. V prvi fazi smo se skoncentrirali zgolj na prvi del tekmovanja oziroma na premagovanje prvega dela proge z ovirami.

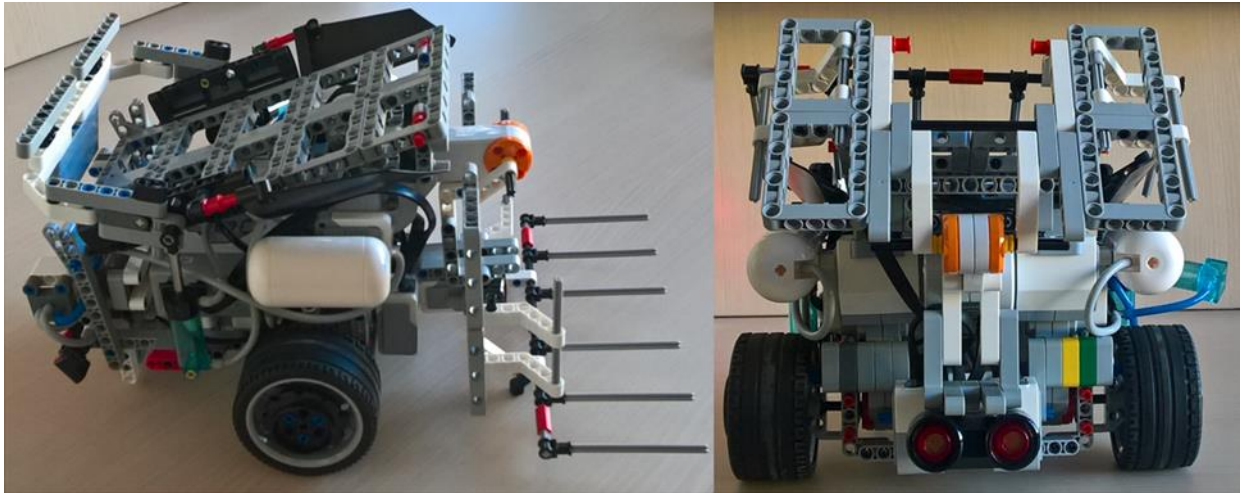
Med skupinami smo se interno dogovorili, da bomo za udeležbo na državnem tekmovanju uporabili in nadgradili robota, ki bo najboljši na šolskem tekmovanju. Nadgradnje je bil deležen tako programski kot tudi konstrukcijski del zmagovalnega mobilnega robota.

Lotili smo se tudi mehanizma za pobiranje žrtev. Vsi v ekipi smo bili naklonjeni ideji, da bi celoten mehanizem izvedli na najenostavnejši način in se tako izognili komplikacijam, ki naraščajo vzporedno s samo kompleksnostjo mehanizma. Že od samega začetka se je pojavljala ideja o vilicam podobnem dvigajočem se mehanizmu. Tako imenovane vilice so žrtve dvignile na zadek robota, kjer se je nahajal tudi pnevmatski mehanizem za odlaganje žrtev v za to predpisan prostor. S pomočjo dvopoložajnega pnevmatskega ventila se je aktivirala loputa, nakar so pobrane žrtve popadale v odlagalni prostor.

*AVTONOMNI MOBILNI ROBOT*  
*Raziskovalna naloga*

---

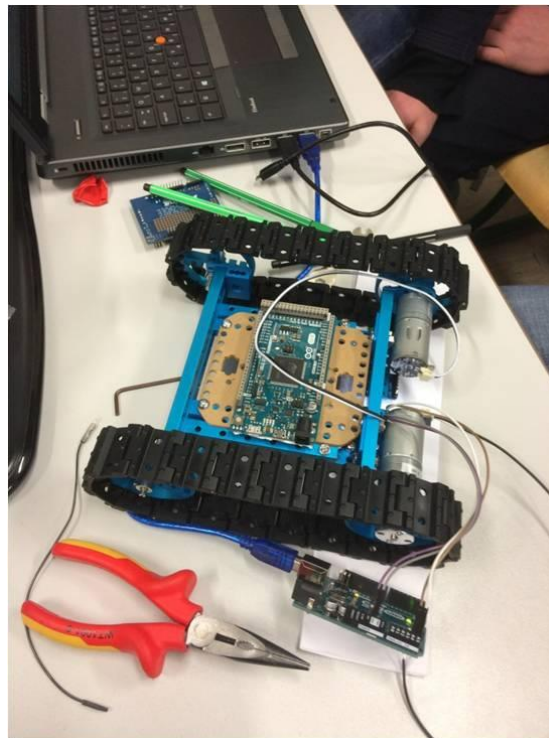
S tako zasnovanim mobilnim robotom smo se udeležili tekmovanja v Mariboru. Ker je bilo državno prvenstvo odprtega tipa, se ga je udeležilo tudi nekaj predstavnikov iz sosednjih držav. Na tem tekmovanju smo v kategoriji ROBOCUP JUNIOR osvojili prvo mesto. S prvim mestom smo si prislužili sodelovanje na svetovnem tekmovanju na Japonskem in udeležbo na robotskem taboru na Reki na Hrvaškem.



*Slika 6: Končen lego robot  
(osebni vir)*

### 3.2 IZDELAVA NOVEGA MOBILNEGA ROBOTA

Za udeležbo na svetovnem tekmovanju na Japonskem smo se v skupini dogovorili, da poskusimo izdelati povsem nov koncept mobilnega robota, in sicer tako po konstrukcijski kot tudi programski plati. Za začetek smo se odločili, da bo robot voden s pomočjo razvojnih plošč Arduino in Raspberry Pi. Odločitev za dva mikroračunalnika je padla ob ideji, da za vodenje robota ne uporabimo svetlobnih senzorjev, temveč kamero. V ta namen smo za obdelavo in prejemanje signalov iz kamere uporabili Raspberry Pi. Ta je pošiljal signale na Arduino preko komunikacije  $I^2C$ , na podlagi katerih je Arduino po prednastavljenih parametrih kontroliral hitrost vrtenja motorjev.



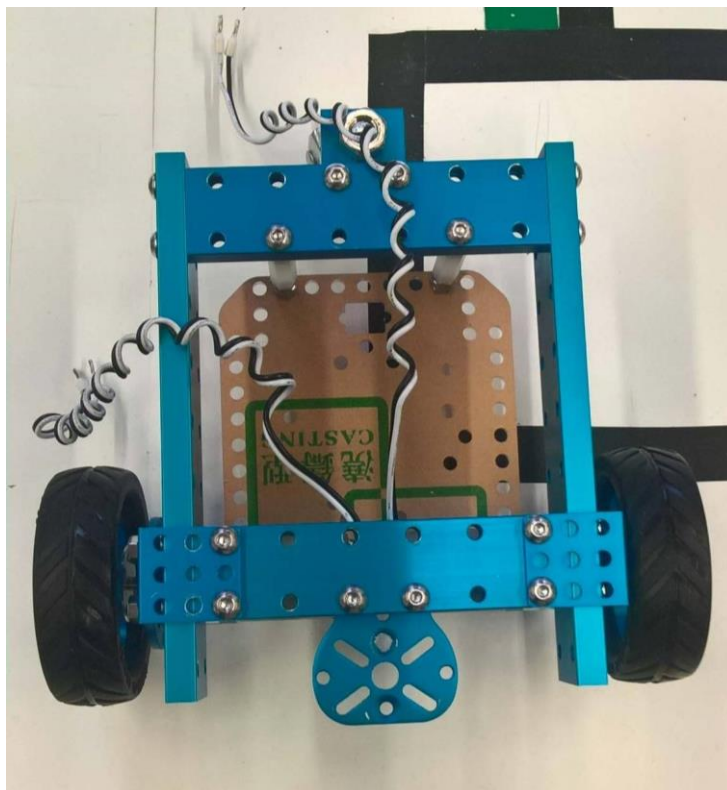
*Slika 7: Izdelava konstrukcije*

*(osebni vir)*

S časom je nastalo veliko konceptov, ki pa se v praksi žal niso obnesli. Prav tako smo ovrgli tudi nekaj programov. Vse omenjeno pa nam je predstavljalo izziv in napredek pri iskanju boljših rešitev.

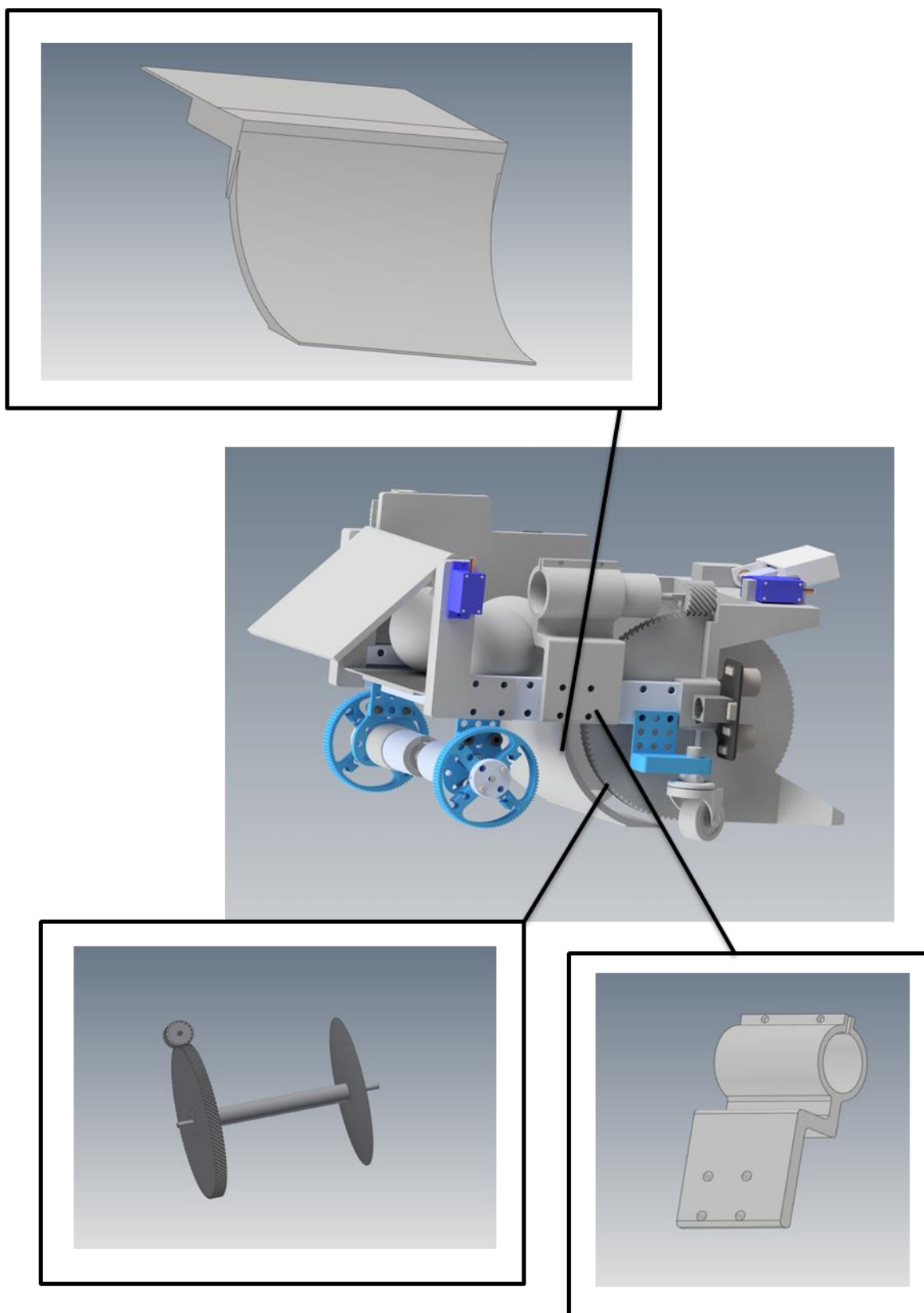
### 3.2.1 KONSTRUKCIJA

Za ogrodje smo sprva uporabili aluminijaste profile, na katerih smo lahko pričeli z izgradnjo konstrukcije in s preizkušanjem programov. Ker je celotna zadeva z aluminijastimi profili delovala neokretno in nam ni ponujala veliko možnosti za nadgrajevanje konstrukcije, smo se odločili za izdelavo 3D-modelov konstrukcije s pomočjo 3D-tiskanja. Takšen način izdelave nam ponuja več svobode pri konstruiranju in oblikovanju mobilnega robota.



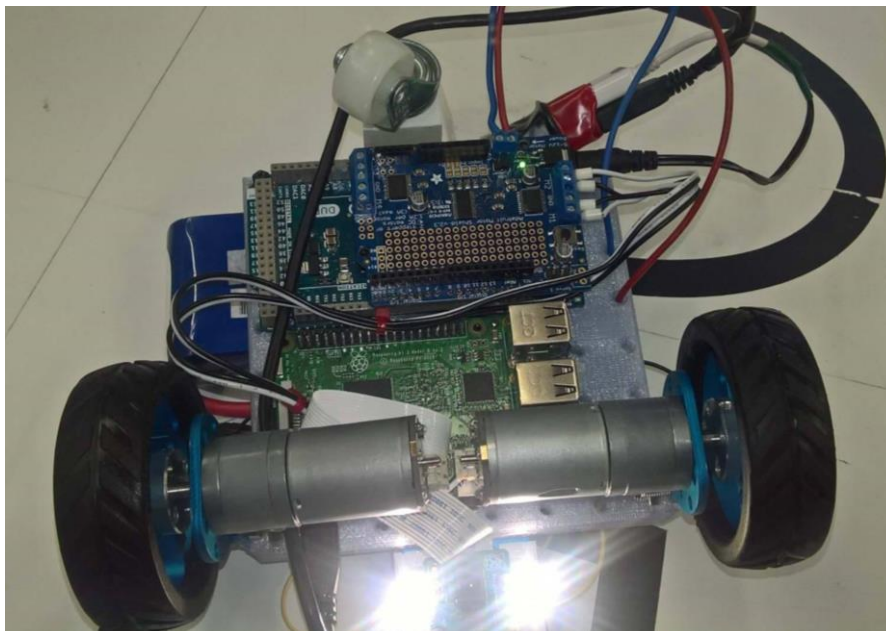
*Slika 8: Osnovna konstrukcija  
(osebni vir)*





Slika 9: Sestavni deli mobilnega robota (osebni vir)

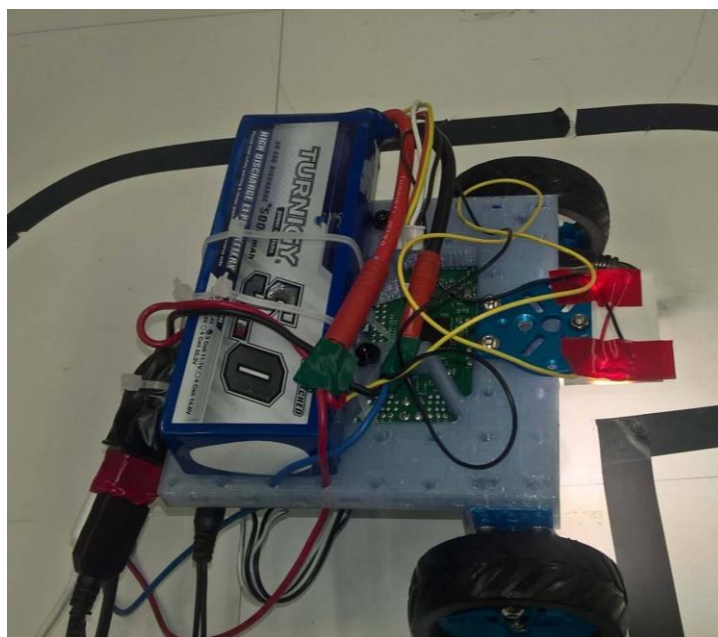
Na spodnji sliki je prikazana pritrditev mikroročunalnikov Arduino in Raspberry PI ter DC-motorjev, ki preko zobniškega reduktorja zmanjšujejo vrtljaje v razmerju 1 : 46. Nameščena je tudi kamera z dodanima osvetliltvama za boljše branje podatkov zaradi manjših motenj.



*Slika 10: Mobilni robot – prikaz s spodnje strani*  
*(osebni vir)*

Zaradi različnih porabnikov sva uporabila baterijo 11.7 V, 5000 mAh. Nisva uporabila svinčene, saj imajo danes baterije veliko večji izkoristek, lahko jih tudi večkrat polnimo in so veliko lažje. Najina baterija je LiPo.





Slika 11: Mobilni robot – prikaz z zgornje strani  
(osebni vir)

### 3.3 PROGRAM

Za programiranje sva si izbrali programska jezika Arduino in Pyton. Za komunikacijo med Arduino in Raspberryjem sva izbrala  $I^2C$ , saj je ta hitra ter zanesljiva. S to kombinacijo mikroračunalnikov sva izdelala program, pri katerem kamera, ki je priključena na Raspberry, pri vožnji po črti opazuje, koliko je črna črta oddaljena od sredine slike in na katerem koncu je zelen kvadrat. Dobljene koordinate pošlje preko komunikacije  $I^2C$  do Arduina, ki dobljene informacije uporabi za vodenje DC-motorjev s pomočjo motor shilda.

#### 3.3.1 RASPBERRY PI

Za zaznavanje črte in zelenih kotov sva si izbrala mikroračunalnik Raspberry PI 3B, saj je eden izmed najboljših mikroračunalnikov za procesiranje slik. Deluje z operacijskim sistemom Linux, ki se večinoma uporablja v industriji, saj je primeren za vodenje in programiranje raznih industrijskih ter robotskih sistemov.

Raspberry PI 3B deluje na podlagi procesorjev 1.2 GHz 64-bit quad-core in 1 GB RAM, ki sta zelo primerna, saj za hitro procesiranje slik potrebujemo zmogljivo in hitro procesno enoto.

### **3.3.1.1 RASPBERRY PI-PROGRAM**

Na začetku programa določimo vse potrebne knjižnice za celoten program. Nato vključimo kamero in ji povemo, kaj naj opazuje; v našem primeru črno ter zeleno barvo. Za boljšo predstavo, kaj robot točno vidi, prikažemo prenos slike in na njem izrisujemo pravokotnik, okoli katerega robot sledi. Sredino tega pravokotnika postavimo na koordinatni sistem v x-osi, kjer je izhodišče na sredini slike, in izpisujemo koordinate v negativni oz. pozitivni smeri. Te podatke pošljemo s pomočjo  $I^2C$  komunikacije do Arduina.

AVTONOMNI MOBILNI ROBOT  
Raziskovalna naloga

---

```
from picamera.array import PiRGBArray #As there is a resolution problem in raspberry pi, w
from picamera import PiCamera
import RPi.GPIO as GPIO
import time
import cv2
import cv2.cv as cv
import numpy as np
import serial

import smbus
import time
# For RPi version 1 use "bus = smbus.SMBus(0)"
bus = smbus.SMBus(1)
# This is the address we setup in the Arduino Program
address = 0x04

#Image analysis work
def segment_colour(frame): #returns only the red colors in the frame
    hsv_roi = cv2.cvtColor(frame, cv2.cv.CV_BGR2HSV)
    mask_1 = cv2.inRange(hsv_roi, np.array([0, 0, 0]), np.array([50, 50, 50]))
    ycr_roi=cv2.cvtColor(frame,cv2.cv.CV_BGR2YCrCb)
    mask_2=cv2.inRange(ycr_roi, np.array((0.,165.,0.)), np.array((255.,255.,255.)))
    hsv_roi = cv2.cvtColor(frame, cv2.cv.CV_BGR2HSV)
    mask_3 = cv2.inRange(hsv_roi, np.array([39, 96, 16]), np.array([100, 255, 100]))
    ycr_roi=cv2.cvtColor(frame,cv2.cv.CV_BGR2YCrCb)
    mask_4=cv2.inRange(ycr_roi, np.array((0.,165.,0.)), np.array((255.,255.,255.)))

    mask = mask_1 | mask_2 | mask_3 | mask_4
    kern_dilate = np.ones((8,8),np.uint8)
    kern_erode = np.ones((3,3),np.uint8)
    mask= cv2.erode(mask,kern_erode) #Eroding
    mask=cv2.dilate(mask,kern_dilate) #Dilating
    cv2.imshow('mask',mask)
    return mask

def find_blob(blob): #returns the red colored circle
    largest_contour=0
    cont_index=0
    contours, hierarchy = cv2.findContours(blob, cv2.RETR_CCOMP, cv2.CHAIN_APPROX_SIMPLE)
    for idx, contour in enumerate(contours):
        area=cv2.contourArea(contour)
        if (area >largest_contour) :
            largest_contour=area

            cont_index=idx
            #if res>15 and res<18:
            # cont_index=idx

    r=(0,0,2,2)
    if len(contours) > 0:
        r = cv2.boundingRect(contours[cont_index])

    return r,largest_contour

def target_hist(frame):
    hsv_img=cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)

    hist=cv2.calcHist([hsv_img],[0],[None],[50],[0,255])
    return hist
```

Uvoz  
knjižnic

Zaznavanje črne, zelene  
barve

Slika 12: Program RPI

(osebni vir)

AVTONOMNI MOBILNI ROBOT  
Raziskovalna naloga

---

```
# capture frames from the camera
for image in camera.capture_continuous(rawCapture, format="bgr", use_video_port=1):
    #grab the raw NumPy array representing the image, then initialize the times
    frame = image.array
    frame=cv2.flip(frame,1)
    global centre_x
    global centre_y
    centre_x=0.
    centre_y=0.
    hsv1 = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
    mask_red=segment_colour(frame) #masking red the frame
    loct,area=find_blob(mask_red)
    x,y,w,h=loct

    if (w*h) < 10:
        found=0
    else:
        found=1
        simg2 = cv2.rectangle(frame, (x,y), (x+w,y+h), 255,2)
        centre_x=x+((w)/2)
        centre_y=y+((h)/2)
        cv2.circle(frame,(int(centre_x),int(centre_y)),3,(0,110,255),-1)
        centre_x-=80
        centre_y=6--centre_y
        print centre_x,centre_y
        def try_io(call, tries=10):
            assert tries > 0
            error = None
            result = None

            while tries:
                try:
                    result = call()
                except IOError as e:
                    error = e
                    tries -= 1
                else:
                    break

            if not tries:
                raise error

        try_io(lambda: bus.write_byte(address, centre_x))

    initial=400
    flag=0

    cv2.imshow("draw",frame)
    rawCapture.truncate(0) # clear the stream in preparation for the next fram

    if(cv2.waitKey(1) & 0xff == ord('q')):
        break
```

Zapisovanje koordinat  
pozicije črte ter pošiljanje  
preko I<sup>2</sup>C

V primeru  
napake pri  
komunikaciji  
preko I<sup>2</sup>C  
pošljemo  
podatek še  
desetkrat.

Zaključek  
programa

Slika 13: Program RPI

(osebni vir)



Slika 14: Raspberry PI 3B

(osebni vir)

### 3.3.2 ARDUINO

Za Arduino sva se odločila, saj je odprtokodna platforma. Temelji na preprosti strojni in programski opremi, ki jo je enostavno uporabljati. Svetovna skupnost oblikovalcev, študentov, umetnikov, programerjev in strokovnjakov se je zbrala okrog te odprtokodne platforme in če se njihovi prispevki seštejejo, pridemo do neverjetne količine dostopnega znanja, ki je lahko v veliko pomoč tako začetnikom kot tudi profesionalcem.

Arduino je nastal na inštitutu Ivrea Interaction oblikovanje kot enostavno orodje za hitro izdelavo prototipov. Takoj, ko je dosegel širšo skupnost, je začel spreminjati svet, se prilagajati novim potrebam in izzivom.

Programska oprema je enostavna za uporabo za začetnike, vendar je dovolj prožna tudi za napredne uporabnike. Deluje na računalnikih z operacijskimi sistemi Mac, Windows in Linux. Uporablja se ga za izgradnjo nizkocenovnih znanstvenih instrumentov, katerih namen je dokazati kemične in fizikalne principe, hkrati pa se ga uporablja tudi v nezahtevni robotiki.

### 3.3.2.1 ARDUINO DUE

Za krmiljenje motorjev sva izbrala Arduino due, saj je daleč najhitrejši izmed vseh Arduino mikroročunalnikov. Uporablja ATSAM3X8E 48 MHz procesor, katerega hitrost procesiranja zadostuje najinim potrebam, med katerimi je najpomembnejše hitro branje podatkov, dobljenih preko  $I^2C$  komunikacije, in hitro odzivanje nanje.

*Tabela 1: Specifikacija Arduina*

Mikrokrmilnik	AT91SAM3X8E
Delovna napetost	3.3 V
Vhodna napetost (priporočena)	7–12 V
Vhodna napetost (omejena)	6–16 V
digitalni I/O-pini	54 (od katerih je 12 predvidenih PWM izhodov)
Analogni vhodi	12
Analogni izhod	2 (DAC)
Popolni DC izhodiščni tok na vseh I/O	130 mA
DC-tok za 3.3 V pin	800 mA
DC-tok za 5 V pin	800 mA
Flash Memory	512 KB vse na voljo za zahteve za uporabnike
SRAM	96 KB (dva shranjevalnika: 64 KB in 32 KB)

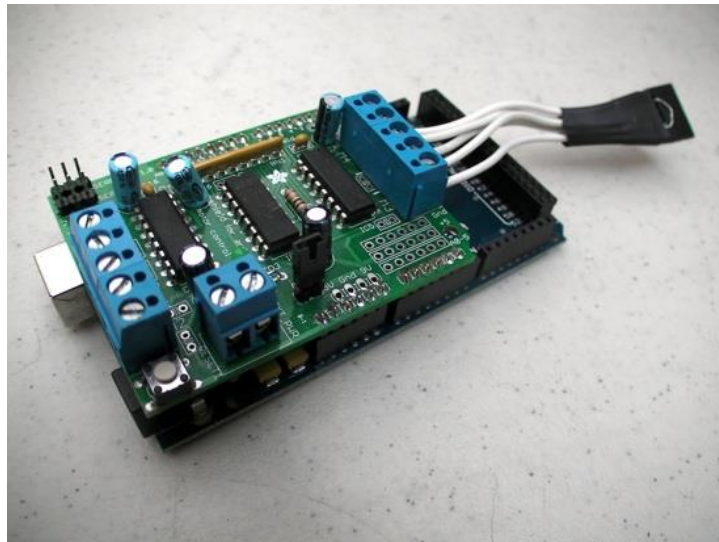
*AVTONOMNI MOBILNI ROBOT*  
*Raziskovalna naloga*

---

Hitrost ure	84 MHz
Dolžina	101.52 mm
Širina	53.3 mm
Teža	36 g

### 3.3.2.2 ARDUINO PROGRAM

Za začetek programa določimo vse spremenljivke ter vnesemo knjižnico za Adafruit motor shield, s pomočjo katerega vodimo motorje. Nato določimo Arduino svojo addresso, ki je potrebna za  $I^2C$  komunikacijo, in mu povemo, da je "Slave" oz. "Suženj" Raspberryju. V nadaljevanju programa imamo kodo, ki je potrebna za sprejem podatkov iz Raspberryja. Nato sva naredila preračun, ki pretvori dobljene podatke nazaj v prvotne negativne številke, saj je pošiljanje negativnih vrednosti preko  $I^2C$  komunikacije nemogoče. Dobljeno vrednost vnesemo v program za vodenje motorjev, ta pa nam določa, kakšna sta moč in smer vrtenja.








*Slika 15: Arduino due z adafruit motor shieldom  
(osebni vir)*



#### 4 PREDSTAVITEV REZULTATOV

S pomočjo zastavljenih hipotez in raziskovalnih metod nama je raziskavo uspelo pripeljati do konca. Kljub mnogim zapletom, težavam pri programiranju in izdelavi opisane aplikacije sva uspela raziskati ter s pomočjo primerjanja različnih lastnosti posameznih komponent priti do primerne funkcionalnosti in oblike zelenega robota. Problem komunikacije med Arduino in Raspberry-jem sva po poglobljanju v strokovno literaturo programske rešila z ponavljanjem pošiljanja signala. Potrdiva lahko tudi hipotezo, povezano s približevanjem oblik in funkcionalnosti dejanski robotski aplikaciji.

*Tabela 2: Potrditev hipotez*

Strojni vid je najprimernejša rešitev za najino aplikacijo.	
Pri izdelavi aplikacije se bova poskušala čim bolj približati dejanski obliki in funkcionalnosti aplikacije.	
Z programom bova odpravila napako na komunikaciji.	
Robot bo brez problemov prevozil labirint.	
Robot bo hitro rešil žrtve.	

## **5 ZAKLJUČEK**

Z zastavljenimi cilji in hipotezami nama je izdelavo reševalnega robota uspelo tudi realizirati. Zadane cilje in hipoteze sva, kljub velikim težavam pri programiranju, uspešno uresničila s trdnim timskim delom. Kljub veliki konkurenci na tekmovanju sva prepričana, da se bo robot dobro izkazal in dosegel tudi dobre rezultate.

## **6 ZAHVALA**

Zahvaljujemo se mag. Andru Glamniku, univ. dipl. inž., mag. Mateju Vebru, univ. dipl. inž., in Robertu Ojsteršku, mag. inž., za pomoč pri konstruiranju, programiranju, podpori in koordinaciji celotnega projekta.

Zahvaljujemo se tudi naši profesorici slovenščine Brigiti Renner, prof., ki si je vzela čas in skrbno prebrala ter lektorirala raziskovalno nalogo.

## **7 VIRI**

[1] Arduino Due (online). (citirano 6. 3. 2017). Dostopno na naslovu:

<https://www.arduino.cc/en/Main/ArduinoBoardDue>

[2] Python (online). (citirano 6. 3. 2017). Dostopno na naslovu: <https://www.python.org/>

[3] Raspberry PI 3B (online). (citirano 6. 3. 2017). Dostopno na naslovu:

<https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>

[4] Robocup junior (online) (citirano 6. 3. 2017). Dostopno na naslovu: <http://rcj.robocup.org/>

## **8 IZJAVA**

Mentorja (-ici), Andro Glamnik in Matej Veber, v skladu z 2. in 17. členom Pravilnika raziskovalne dejavnosti »Mladi za Celje« Mestne občine Celje, zagotavljam, da je v raziskovalni nalogi z naslovom

Izdelava inteligentnega mobilnega robota,

katere avtorja (-ici) sta Luka Tomšič, Dominik Hrastnik.

-besedilo v tiskani in elektronski obliki istovetno,

-pri raziskovanju uporabljeno gradivo navedeno v seznamu uporabljene literature,

-da je za objavo fotografij v nalogi pridobljeno avtorjevo (-ičino) dovoljenje in je hranjeno v šolskem arhivu,

-da sme Osrednja knjižnica Celje objaviti raziskovalno nalogo v polnem besedilu na knjižničnih portalih z navedbo, da je raziskovalna naloga nastala v okviru projekta Mladi za Celje,

-da je raziskovalno nalogo dovoljeno uporabiti za izobraževalne in raziskovalne namene s povzemanjem misli, idej, konceptov oz. besedil iz naloge ob upoštevanju avtorstva in korektnem citiranju,

-da smo seznanjeni z razpisni pogoji projekta Mladi za Celje.

Celje, 8. 3. 2017

žig šole

Podpis mentorjev(-ic)

Podpis odgovorne osebe