

Šolski center Celje
Srednja šola za strojništvo, mehatroniko in medije

MULTIFUNKCIJSKO AVTONOMNO VOZILO

RAZISKOVALNA NALOGA

Avtorji:

Dominik NEMEC, M-3. c
Peter PLANKO, M-3. c
Tilen ZIDAR, M-3. c

Mentorji:

Robert OJSTERŠEK, dipl. inž. UN
Marjan JAMNIŠEK, p. u.
Matjaž CIZEJ, univ. dipl. inž.
Stevo ROMANIČ dipl. inž.

Mestna občina Celje, Mladi za Celje
Celje 2015

Povzetek

Za projektno delo smo najprej raziskali trg in nato iz dveh želja naredili eno. Vozilo smo najprej narisali na list papirja in nato še zmodelirali v SolidWorksu. Ko smo imeli narejen model, smo začeli žagati, brusiti in variti. Vse skupaj smo nato očistili, prebarvali s temeljno barvo in nato še s črnim in rdečim lakom. Nato smo zvezali elektroniko in naredili svojo aplikacijo za telefon, ki vodi vozilo. Tako je postal naš izdelek konkurenčen na trgu.

KAZALO

Povzetek	2
Uvod	5
Hipoteze	6
1 Potek izdelave.....	7
1.1 Konstruiranje vozila	7
1.2 Začetki izdelave.....	7
1.3 Izdelava vzmetenja	8
1.4 Razrez in valjanje pločevine.....	8
1.5 Izdelava zaboja za baterije.....	9
1.6 Sestavljanje izdelka	9
1.7 Vgradnja inkrementalnega dajalnika pomika	10
1.8 Izdelava ohišja senzorjev.....	10
2 Elektronika in programiranje.....	11
2.1 Vezava baterij	11
2.2 Zaboj za krmilnik	12
2.3 Program	14
2.4 Mobilna aplikacija	19
2.5 Priklop H-mostiča.....	19
2.6 Opis pogonskega motorja.....	21
2.7 Pomen inkrementalnega dajalnika.....	21
2.8 Vezava ultrazvočnih senzorjev PING	22
3 Nadgradnja	23
3.1 Snemanje filma.....	23
3.2 Varnost in reševanje	23
3.3 Prevažanje	24
3.4 Uporaba v industriji in gospodarstvu	25
3.5 V kmetijstvu in gozdarstvu.....	28
3.6 Avtonomno vozilo v turizmu.....	28
3.7 Vozilo za lastno uporabo	29
4 Zaključek	30
5 Zahvala	31
6 Viri in literatura	32

KAZALO SLIK

Slika 1: 3D-model	7
Slika 2: Začetek konstruiranja	7
Slika 3: Vzmetenje zadnjih koles	8
Slika 4: Odbijač	8
Slika 5: Zaboja za baterije	9
Slika 6: Sestavljanje	9
Slika 7: Inkrementalni dajalnik na pogonskem motorju	10
Slika 8: Ohišje senzorja	10
Slika 9: Vezava baterij	11
Slika 10: Vezava BMS-modula	12
Slika 11: Škatla s krmilnikom	12
Slika 12: ArduinoMega	13
Slika 13: WifiShield	13
Slika 14: ArduinoMega in WifiShield	13
Slika 15: Aplikacija	19
Slika 16: H-mostič	19
Slika 17: Prilagoditveno vezje za H-most	20
Slika 18: Izvedba vezja	20
Slika 19: Inkrementalni dajalnik	21
Slika 20: Ultrazvočni senzor razdalje	22
Slika 21: Robotska roka	23
Slika 22: Raziskovanje v nevarnih okoliščinah	23
Slika 23: Prenašanje eksploziva	24
Slika 24: MPS-postaja	25
Slika 25: Industrijski voziček	25
Slika 26: GPS-senzor	26
Slika 27: Girokop	26
Slika 28: Nadzor v težko dostopnih prostorih	27
Slika 29: Roka s hitromenjalno pripravo	27
Slika 30: Vozilo za razvažanje grozdja	28
Slika 31: Robot za strežbo hrane	29

KAZALO TABEL

Tabela 1: Opisna tablica motorja	21
----------------------------------------	----

Uvod

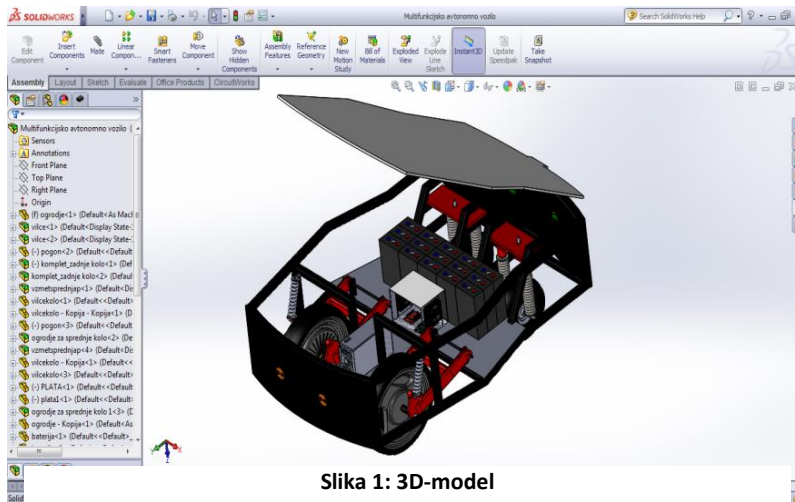
Na začetku projektne dela smo izbirali med predelavo bencinskega gokarta v električnega in izdelavo industrijskega vozička za prevažanje naročenih izdelkov do vsakega delavca. Medtem smo še raziskali trg in vse skupaj združili ter pričeli z izdelovanjem multifunkcijskega avtonomnega vozila.

Hipoteze

- Izdelamo lahko izdelek, ki bo dostopen vsem industrijam.
- Izdelamo lahko izdelek, ki bo imel do sedaj največ funkcij.
- Izdelamo lahko izdelek, ki bo dostopen tudi posameznikom.
- Izboljšamo lahko natančnost pozicioniranja.

1 Potek izdelave

1.1 Konstruiranje vozila



Slika 1: 3D-model

Delo smo začeli z risanjem oblike vozila na list papirja. Narisali smo nekaj risb in nato izbrali najprimernejšo obliko, na kateri smo napravili še manjše popravke. Po končanem risanju na papir smo vse skupaj ponovno narisali na računalniku v 3D-modelu. Za risanje smo uporabljali

računalniški program SolidWorks.

Tako smo narisani 3D-model začeli izdelovati. Za ohišje smo uporabili navadne pohištvne cevi 30 mm x 30 mm x 2 mm, zato so bili tudi stroški manjši. Medtem ko smo izdelovali ogrodje, smo dobili večja kolesa z vgrajenim 1.5 kW motorjem in vilicami, en par manjših koles in tri pare vzmeti.

1.2 Začetki izdelave



Slika 2: Začetek konstruiranja

Pri izdelavi smo izmenjavali izkušnje, dajali nasvete in na koncu našli skupno točko ter misel prenesli na izdelek. Ogrodje smo sprva točkovno zavarili in ko smo se z obliko vsi strinjali, smo jo zvarili. Pri izdelavi oblike smo imeli največ težav z velikostjo izdelka. Poskušali smo narediti čim manjše, stabilno in kompaktno ogrodje. Varili smo z MIG/MAG varilnim Iskrinim aparatom. Po končanem varjenju smo zware očistili s kotnim brusilnikom.

1.3 Izdelava vzmetenja



Slika 3: Vzmetenje zadnjih koles

Potrebno je bilo tudi izdelati vpetje z amortizacijo »zadnjih« manjših koles. To smo naredili tako, da smo postružili kovinsko palico in vanjo naredili izvrtino za amortizer ter dve manjši izvrtini z navojem za pritrditev oz. lažjo menjavo amortizerja. Ta kos – »pušo« smo privarili na kovinsko 10 mm debelo ploščo z že narejeno izvrtino in sedežem za tlačni ležaj. To smo pritrdili na še eno prav tako pripravljeno ploščo s sedežem za ležaj s posebnim vijakom, ki smo ga izdelali sami. Tako lahko to kolo nosi veliko težo in ima možnost obrata za 360°.

1.4 Razrez in valjanje pločevine

Nato smo odrezali še 2 mm debelo pločevino, na katero smo postavili baterije in elektronski del. Ko smo končali, smo celotno ogrodje temeljito očistili, prebarvali s temeljno barvo, z rdečim in črnim lakom ter bleščicami.



Slika 4: Odbijač

Med sušenjem barve smo poskrbeli tudi za zunanjo podobo vozila. 1 mm debelo pločevino smo zvaljali, tako da se je lepo prilegala okvirju. Vanjo smo izvrtali dve izvrtini za senzorje. Izdelali smo tudi šolski znak, tako da smo bakreni prah stisnili in žgali v peči – postopek sintranja materialov.

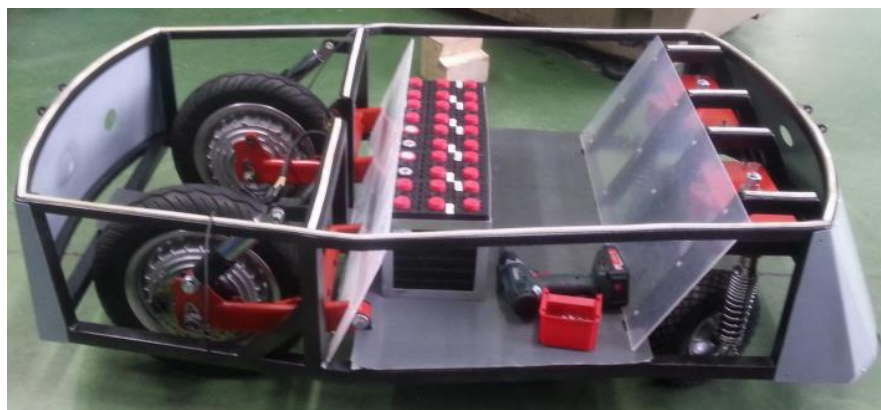
1.5 Izdelava zaboja za baterije



Slika 5: Zaboj za baterije

Hkrati smo izdelali aluminijasto škatlo, v katero smo postavili baterije. Na ohišje smo pritrdili malo večji aluminijasti okvir, v katerega postavimo škatlo s petnajstimi baterijami. Tako je možno enostavno in hitro odstraniti vse baterije hkrati.

1.6 Sestavljanje izdelka



Slika 6: Sestavljanje

Po končanem barvanju smo pritrdili pločevino in začeli izdelovati plašč ter zaboje za elektroniko iz pleksi stekla. Izdelali smo pokrov iz aluminijaste pohodne pločevine, ki se bo odpiral s pomočjo plinskih amortizerjev.

1.7 Vgradnja inkrementalnega dajalnika pomika



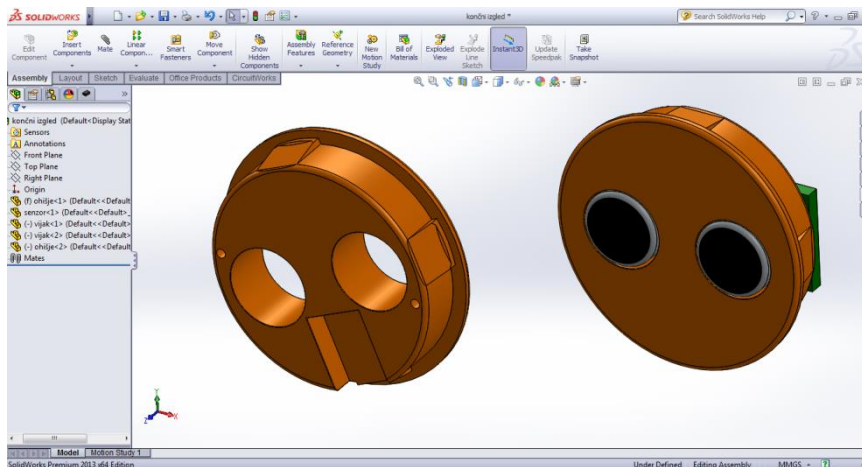
Slika 7: Inkrementalni dajalnik na pogonskem motorju

V sprednja pogonska kolesa smo morali vgraditi tudi ležaje z inkrementalnim dajalnikom. S tem smo ugotovili točen položaj koles in si olajšali delo s samim programiranjem.

1.8 Izdelava ohišja senzorjev

Medtem smo naredili svoj 3D-tiskalnik, s katerim smo si natisnili prototipe in zmanjšali stroške izdelave.

Izdelali smo ohišja za vgradnjo ultrazvočnih senzorjev razdalje. Najprej smo v SolidWorksu narisali model, nato pa ga pretvorili s posebnim programom v G-kodo in tako je 3D-tiskalnik prepoznal obliko ter natisnil model. Prednost je v tem, da je takšna izdelava prototipa veliko cenejša, kot če bi ga izdelali s CNC-strojem.

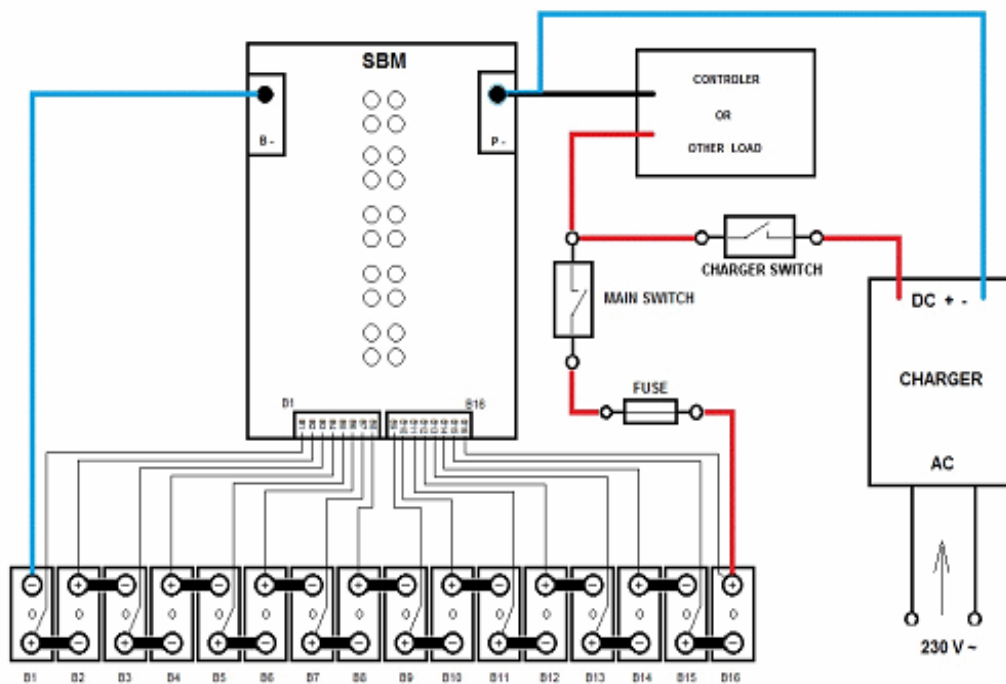


Slika 8: Ohišje senzorja

2 Elektronika in programiranje

2.1 Vezava baterij

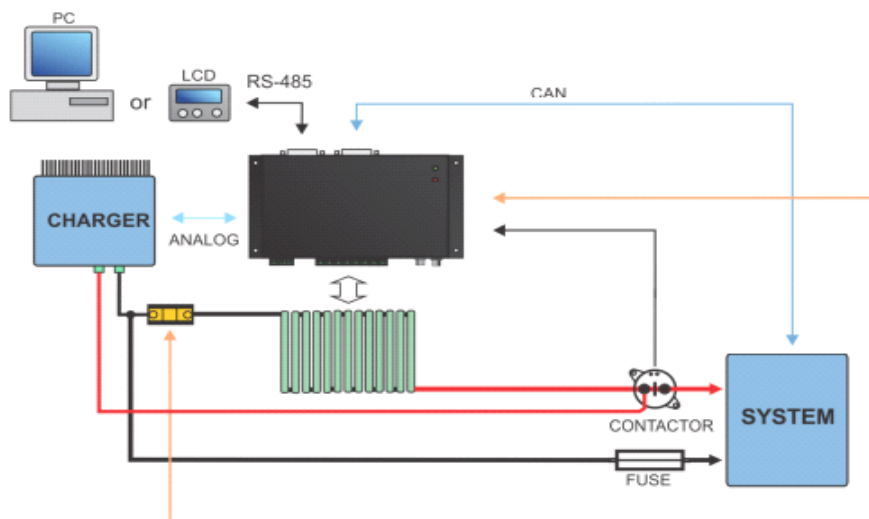
Ko je bila zgradba končana, smo se lotili ožičenja industrijskega vozička. Kupili smo dva 1.5 kW motorja in 15 baterij. Uporabili smo LiFePO4 SE60AHA baterije, ki smo jih zvezali takole:



Slika 9: Vezava baterij

Ena od teh baterij proizvaja približno 3.2 V in 60 Ah. Na zgornji sliki je vsaka baterija povezana posamično na BMS (Battery Management System). Za BMS smo uporabili REC BMS 9R, ki ima možnost priklopa do 15 baterij. Njegova maksimalna napetost po bateriji je 3.5 V ob zagonu in 3.6 V ob zaključku. Maksimalna napetost vseh baterij, ki jo zdrži, je 62.5 V. Zaščiten je z IP32, kar pomeni 3. stopnjo varnosti proti stvarem, večjim od 2.5 mm (orodja, debele žice itd.) in 2. stopnjo varnosti proti rahlim kapljicam tekočine. BMS se veže po spodnji sliki:

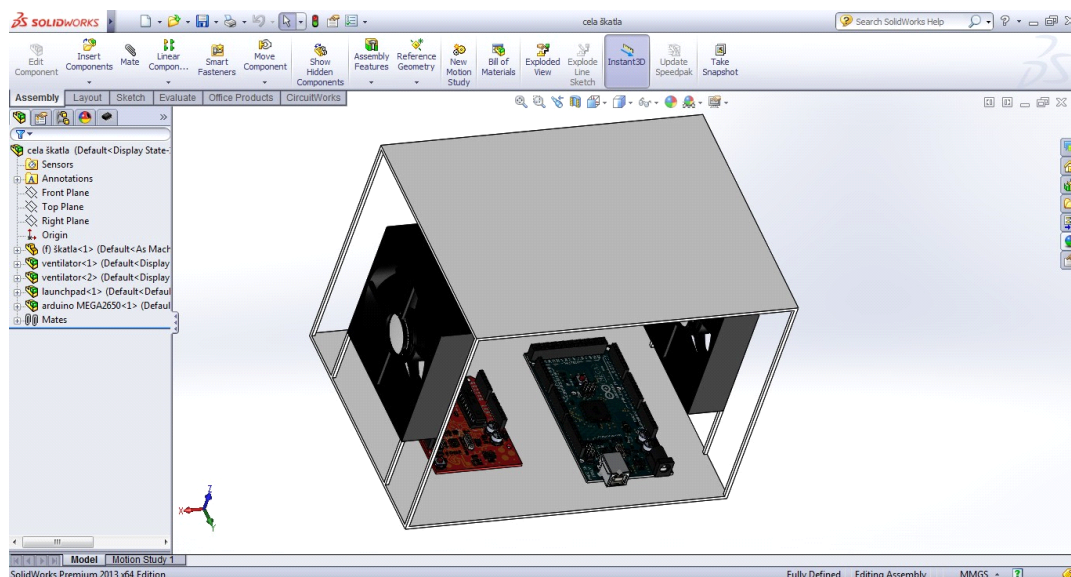
Multifunkcijsko avtonomno vozilo



Slika 10: Vezava BMS-modula

2.2 Zabož za krmilnik

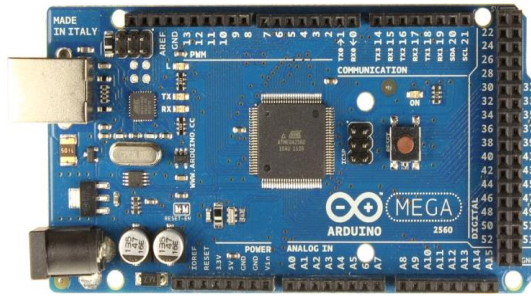
Za elektronske komponente je bilo potrebno narediti ustrezno veliko varovalno-prezračevalno-hladilno komoro. Naredili smo jo z uporabo prozorne plastike, na kateri dva ventilatorja skrbita za pretok zraka skozi njo.



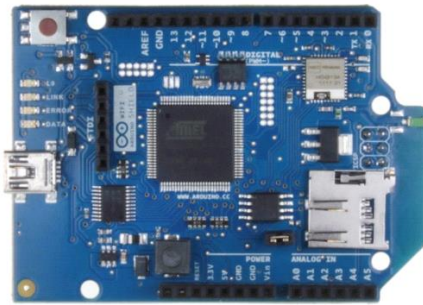
Slika 11: Škatla s krmilnikom

Za krmilnik smo uporabili ArduinoMega in ArduinoWifiShield.

Multifunkcijsko avtonomno vozilo

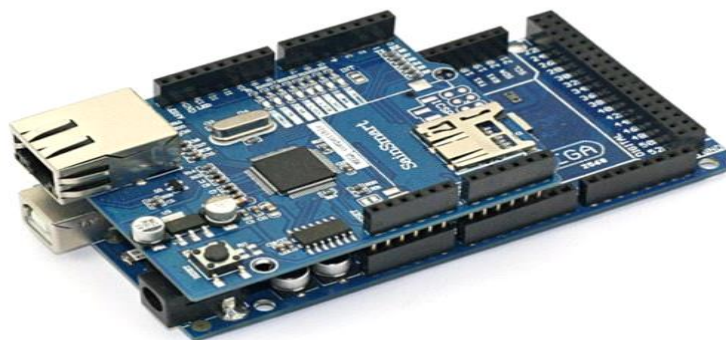


Slika 12: ArduinoMega



Slika 13: WifiShield

ArduinoMega deluje na 5 V, zato smo morali zmanjšati napetost iz baterij na krmilnik. Mega ima 54 digitalnih vhodov/izhodov (15 od njih se lahko uporablja kot PWM), 16 analognih vhodov, 4 UARTs, 16 MHz oskulator, USB-povezavo, napajalni priključek, ICSP-header in tipko za reset. Krmilnik komunicira z računalnikom preko programa ArduinoSoftware. Na računalniku smo napisali kodo, z njo in s pomočjo WifiShielda komunicira z mobilno napravo. WifiShield vstavimo direktno na krmilnik.



Slika 14: ArduinoMega in WifiShield

2.3 Program

Krmilnik in WifiShield komunicirata preko pina 2, zato nanj nismo smeli ničesar vezati. Napajanje dobi iz krmilnika, zato ga ni potrebno posebej napajati. Koda za krmilnik je:

```
#include<Adafruit_CC3000.h>
#include<ccspi.h>
#include<SPI.h>
#include<string.h>
#include "utility/debug.h"
#include<stdlib.h>

intmotorCommand[4];

#define ADAFRUIT_CC3000_IRQ 3
#define ADAFRUIT_CC3000_VBAT 8
#define ADAFRUIT_CC3000_CS 10

intspeed_motor1 = 6;
intspeed_motor2 = 5;
intdirection_motor1 = 7;
intdirection_motor2 = 4;

#define WLAN_SSID "yourNetwork" // cannotbelongerthan 32 characters!
#define WLAN_PASS "yourPassword"
#define WLAN_SECURITY WLAN_SEC_WPA2 // Thiscanbe WLAN_SEC_UNSEC, WLAN_SEC_WEP,
WLAN_SEC_WPA or WLAN_SEC_WPA2

Adafruit_CC3000 cc3000 = Adafruit_CC3000(ADAFRUIT_CC3000_CS, ADAFRUIT_CC3000_IRQ,
ADAFRUIT_CC3000_VBAT,
SPI_CLOCK_DIV2);

#define LISTEN_PORT 8888

Adafruit_CC3000_Server robotServer(LISTEN_PORT);

voidsetup(){
for(int i=4;i<=7;i++)
{
pinMode(i, OUTPUT); //set pin 4,5,6,7 to output mode
}

// Connect to WiFinetwork
cc3000.connectToAP(WLAN_SSID, WLAN_PASS, WLAN_SECURITY);
Serial.println("Connected to WiFinetwork!");

// Check DHCP
Serial.print(F("Requestingaddressfrom DHCP server..."));
for(t=millis(); !cc3000.checkDHCP() && ((millis() - t) <dhcpTimeout); delay(1000));
if(cc3000.checkDHCP()) {
Serial.println(F("OK"));
} else {
Serial.println(F("failed"));
return;
}

robotServer.begin();
```

Multifunkcijsko avtonomno vozilo

```
}

voidloop(){
Adafruit_CC3000_ClientRefclient = robotServer.available();
char c = client.read();
result = result + c;
client.println("HTTP/1.1 200 OK");
client.println("Content-Type: text/html");
client.println("Connection: close");
client.println();

    format_result(motorCommand,result);
Serial.println("Motor 1 speed: " + String(motorCommand[0]) + " anddirection: " + String(motorCommand[2]));
Serial.println("Motor 2 speed: " + String(motorCommand[1]) + " anddirection: " + String(motorCommand[3]));

    send_motor_command(speed_motor1,direction_motor1,motorCommand[0],motorCommand[2]);
    send_motor_command(speed_motor2,direction_motor2,motorCommand[1],motorCommand[3]);

voidsend_motor_command(intspeed_pin, intdirection_pin, intpwm, booleanreverse) {
analogWrite(speed_pin,pwm); // Set PWM control, 0 for stop, and 255 formaximumspeed
if(reverse)
    {
digitalWrite(direction_pin,HIGH);
    }
else
    {
digitalWrite(direction_pin,LOW);
    }
}
}
```

S to kodo se povežemo na splet in sprejemamo podatke, ki jih pošilja telefon. Program, ki smo ga uporabili za izdelavo aplikacije na telefonu, se imenuje Android Studio. V bistvu se program poveže preko wifija na krmilnik in pošilja podatke, kaj naj dela. Motorje kontroliramo s pomočjo giroskopa, ki je vdolan v telefon. Program bere pozicijo telefona, v katero smer je nagnjen. Npr. če je telefon nagnjen naprej, bo voziček šel naprej, če je nagnjen nazaj, bo šel nazaj, za levo se bo desni motor vrtel naprej, levi pa bo na miru, za desno pa se bo levi motor vrtel naprej, desni pa bo na miru. Koda za aplikacijo na telefonu je sledeča:

```
publicclassCarControllerextendsActivityimplementsSensorEventListener {

    SensorManagersensorManager = null;

    EditTextIpAddress;
    EditTextPortNum;
    TextViewtextIn;

    TextView X;
    TextView Y;
    TextView Z;

    /** Calledwhentheactivity is firstcreated. */
    @Override
```


Multifunkcijsko avtonomno vozilo

```
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    sensorManager = (SensorManager) getSystemService(SENSOR_SERVICE);

    setContentView(R.layout.main);
    IpAddress = (EditText)findViewById(R.id.enteredIPAddress);
    PortNum = (EditText)findViewById(R.id.enteredPort);

    textIn = (TextView)findViewById(R.id.textin);

    X = (TextView)findViewById(R.id.X);
    Y = (TextView)findViewById(R.id.Y);
    Z = (TextView)findViewById(R.id.Z);

    /// Connectbuttonlistener///
    ButtonconnectButton = (Button)findViewById(R.id.connectButton);
    connectButton.setOnClickListener(connectButtonOnClickListener);

    /// Disconnectbuttonlistener ///
    ButtonDisconnectButton = (Button)findViewById(R.id.disconnectButton);
    DisconnectButton.setOnClickListener(DisconnectButtonOnClickListener);

    StringAppTAG = bk.CarController.Globals.getTag();
    Log.d(AppTAG, "Loaded");

}

@Override
public void onResume(){
    super.onResume();
    sensorManager.registerListener(this,
    sensorManager.getDefaultSensor(Sensor.TYPE_ORIENTATION),SensorManager.SENSOR_DELAY_GAME);
}
@Override
public void onStop(){
    super.onStop();
    sensorManager.unregisterListener(this,
    sensorManager.getDefaultSensor(Sensor.TYPE_ORIENTATION));
}

@Override
public void onAccuracyChanged(Sensor sensor, int accuracy) {
    // TODO Auto-generated method stub

}

@Override
public void onSensorChanged(SensorEvent event) {
    // TODO Auto-generated method stub
    synchronized (this) {
        switch(event.sensor.getType()){
```


Multifunkcijsko avtonomno vozilo

```
caseSensor.TYPE_ORIENTATION: {
    X.setText("X: "+Float.toString(event.values[0]));
    Y.setText("Y: "+Float.toString(event.values[1]));
    floatfy = event.values[1];
    int y = (int)fy;

    if( y > 20){

        //System.out.println("LEFT");
        bk.CarController.Globals.setLeft("L");
        //bk.CarControler.Globals.setLeft("L");
        bk.CarController.Globals.setRight("X");
    }
    if(y<-20){
        bk.CarController.Globals.setLeft("X");
        bk.CarController.Globals.setRight("R");
    }
    if((y<20)&&(y>-20)){
        bk.CarController.Globals.setLeft("X");
        bk.CarController.Globals.setRight("X");
    }
}

Z.setText("Z: "+Float.toString(event.values[2]));

floatfz = event.values[2];
int z = (int)fz;

if( z > 60){
    //System.out.println("BACK");
    bk.CarController.Globals.setForward("X");
    bk.CarController.Globals.setBackward("B");
}
if(z<30){
    //System.out.println("FRWD");
    bk.CarController.Globals.setForward("F");
    bk.CarController.Globals.setBackward("X");
}
if((z>30)&&(z<60)){
    bk.CarController.Globals.setBackward("X");
    bk.CarController.Globals.setForward("X");
}

break;
}
caseSensor.TYPE_ACCELEROMETER: {
    break;
}
}
}
```

```
GameLoopgameLoop = newGameLoop();
```

Multifunkcijsko avtonomno vozilo

```
Button.OnClickListenerconnectButtonOnClickListener = newButton.OnClickListener(){
    @Override
    publicvoidonClick(View arg0){

        Log.d(bk.CarController.Globals.getTag(),"GlobalSocket is Opening");
        System.out.println(bk.CarController.Globals.getSocket().isConnected());
        try{

            intport = Integer.parseInt(PortNum.getText().toString());
            System.out.println(port);
            Socketgsocket = new
Socket(IPAddress.getText().toString(),Integer.parseInt(PortNum.getText().toString()));
            bk.CarController.Globals.setSocket(gsocket);

        }catch(Exception e){
            Log.e(bk.CarController.Globals.getTag(),e.toString());
        }

        textIn.setText("Connected");

        bk.CarController.Globals.setSending(true);
        gameloop.start();

    } };

Button.OnClickListenerDisconnectButtonOnClickListener = newButton.OnClickListener(){

    @Override
    publicvoidonClick(View v) {
        // TODO Auto-generatedmethodstub

        Log.d(bk.CarController.Globals.getTag(),"GlobalSocket is Closing");
        System.out.println(bk.CarController.Globals.getSocket().isConnected());
        try {
            bk.CarController.Globals.getSocket().shutdownOutput();
            bk.CarController.Globals.getSocket().close();
        } catch (Exception e) {
            // TODO Auto-generatedcatchblock
            Log.e(bk.CarController.Globals.getTag(),e.toString());
        }

        textIn.setText("Disconnected");
        bk.CarController.Globals.setSending(false);
        gameloop.requestStop();

    }

};

}
```

Ta koda poveže telefon s krmilnikom preko spleta in preko krmilnikovega IP-naslova ter port številke. Ko imamo vse pravilno izpolnjeno, lahko začnemo s komuniciranjem.

2.4 Mobilna aplikacija

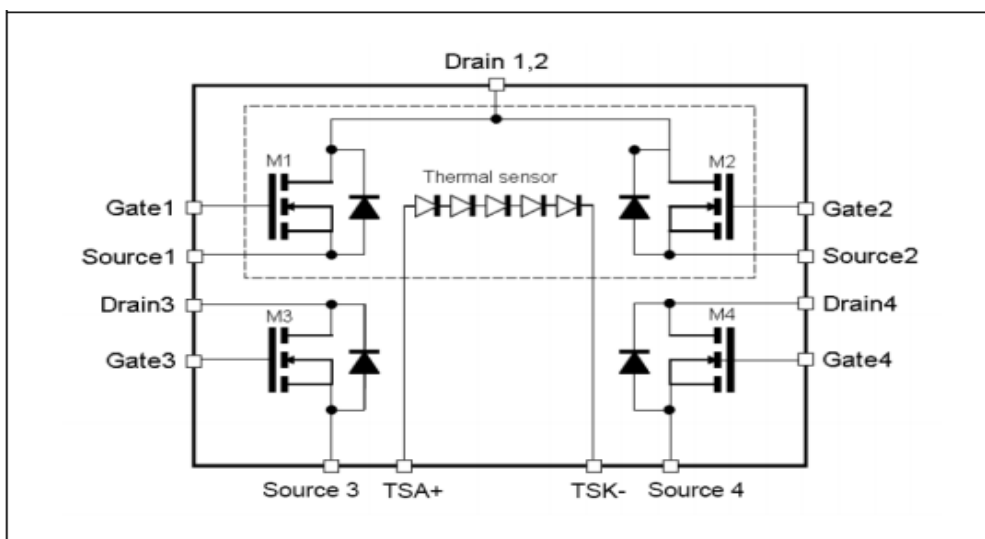


Slika 15: Aplikacija

Ko kliknemo connect se telefon poveže na krmilnik in začne pošiljati svojo pozicijo. Ker pa 1.5 kW motorja ne moremo direktno povezati na krmilnik, smo ju zvezali preko H-mostiča in Motor DriverShielda VNH7013.

2.5 Priklop H-mostiča

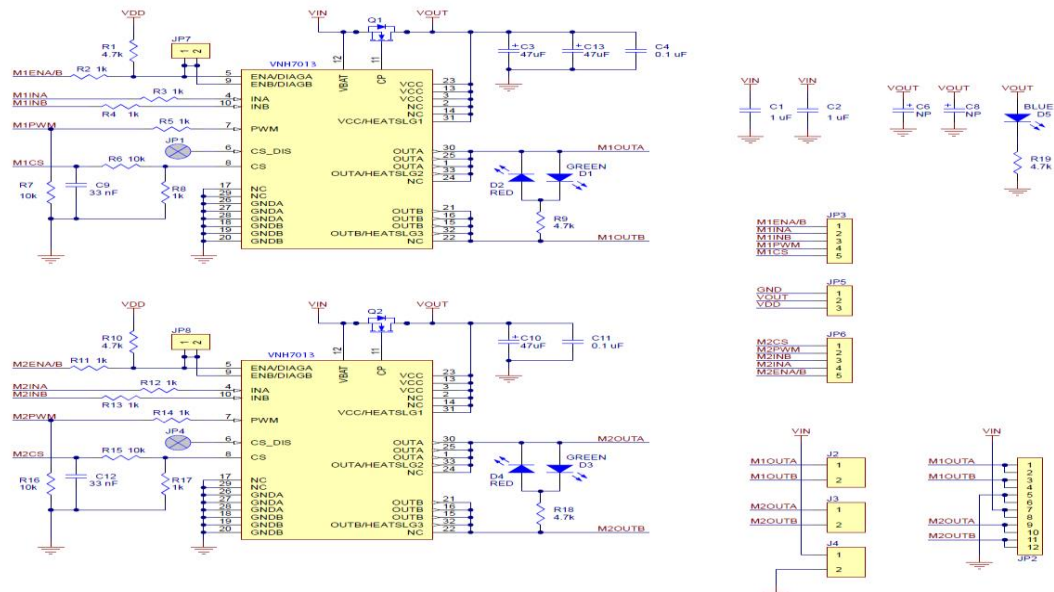
H-mostič je neke vrste most, s katerim kontroliramo, v katero smer se bo naš DC-motor vrtel. Motor DriverShield pa nam omogoča lažje krmiljenje DC-motorja, saj je kompatibilen z našim krmilnikom. Njegova maksimalna operacijska napetost je 72 V, tok pa je 40 A. V njem je že narejen H-mostič, zato ga ni bilo potrebno dodatno narediti. Spodnja slika prikazuje H-mostič in njegove komponente.



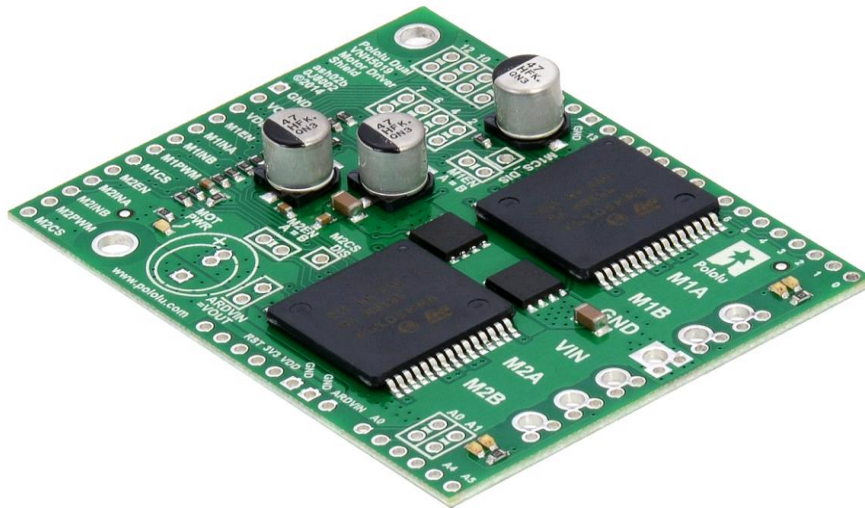
Slika 16: H-mostič

Multifunkcijsko avtonomno vozilo

Na sliki so vidni temperaturni senzori. Spodaj je prikazano, kako se VNH7013 poveže na krmilnik in kako zgleda v resničnem svetu.



Slika 17: Prilagoditveno vezje za H-most



Slika 18: Izvedba vezja

2.6 Opis pogonskega motorja

Motorja, ki sta uporabljena, sta ES15 DC. Najpogosteje se takšni motorji uporabljajo v električnih skuterjih, saj so že vdelani v kolesa. Njihova delovna napetost je 48 V, tok pa 30 A.

Tabela 1: Opisna tablica motorja

OPIS	ES15
Ocenjena moč	1.500 W
Ocenjena napetost	48 V
Ocenjen tok	30 A (Max ~45 km/h)
Ocenjen navor	20 Nm
Ocenjen RPM	420 rpm
Napetost brez bremena	2.9 A
Hitrost brez bremena	660 rpm
Maksimalna moč	3800 W
Maksimalni navor	70 Nm
Učinkovitost	85 %
Teža	18 kg
Velikost koles	12" x 3"

2.7 Pomen inkrementalnega dajalnika

Za lažjo regulacijo hitrosti vrtenja motorjev smo uporabili dva inkrementalna dajalnika pomika, ki smo ju vgradili neposredno na kolesa preko dveh krogličnih ležajev. Inkrementalna dajalnika pomika merita trenutno rotacijo in zasuk elektromotorja, kar lahko posledično uporabimo za merjenje hitrosti in položaja celotne platforme.



Slika 19: Inkrementalni dajalnik

2.8 Vezava ultrazvočnih senzorjev PING

Glavni premiki tega vozila so naprej, nazaj, levo in desno. Vprašanje je, kaj se zgodi, če nastopi nepričakovana ovira in nimamo časa reagirati. Zato smo na krmilnik še dodatno povezali 6 ultrazvočnih senzorjev razdalje.



Slika 20: Ultrazvočni senzor razdalje

Delujejo na 5 V, kar pomeni, da jih lahko direktno povežemo na krmilnik. Imajo 3 pine, napajanja 5 V, minus in signal, ki ga priklopimo na analogne vhode na krmilniku. Delujejo na razdalji od 3 cm do 4 m. Nameščeni so: dva spredaj, dva zadaj in eden na vsaki strani vozila. Za njih smo izdelali nosilce s pomočjo 3D-tiskalnika. Vse to pa lahko gledamo preko LCD-zaslona, ki prikazuje podatke, kot so napolnjenost baterije, tok, napetost, moč itd. Njegova velikost zaslona je 2.8" in se lahko priklopi na Arduino krmilnik.

3 Nadgradnja

Za raziskovalno nalogo smo naredili avtonomno vozilo, ki ima na vrhu ravno platformo, zasnovano tako, da jo lahko uporabimo v različnih aplikacijah.

3.1 Snemanje filma

Nanj bomo postavili robotsko roko, ki bo delovala na 48 V in v roki držala kamero. Kamera bo služila kot snemalno avtonomno vozilo za snemanje filmov ali video spotov.



Slika 21: Robotska roka

Roka se premika v isti smeri kot kolesa, na katerih je vozilo. Če bi se vozilo premikalo naprej, nazaj, levo ali desno ob istem času, bi robotska roka snemala in se premikala ter sledila dogajanju na prizorišču.

3.2 Varnost in reševanje

Drugi primer uporabne aplikacije vozila je uporaba v različnih nevarnih okoliščinah. Lahko si zamislimo, da se je nekje v gozdu zanel požar in zaradi gostega dima je vidljivost zelo omejena. To vozilo bi lahko brezžično preko telefona vodili po gozdu in videli, kje je npr. ujeta kakšna oseba in jo lažje rešili.



Slika 22: Raziskovanje v nevarnih okoliščinah

Roko bi lahko uporabljali vojaki, saj jih v vojni lahko iz kota pričakuje sovražnik. Lahko bi uporabili to vozilo, in sicer kot izvidnik. S tem bi vnaprej vedeli, kaj lahko pričakujejo. Vozilo bi lahko uporabljali tudi policisti, tam, kjer jim grozi nevarnost eksplozije. To vozilo bi se lahko približalo domnevemu eksplozivu in z roko ter kamero preverilo, če gre za eksploziv ali ne.



Slika 23: Prenašanje eksploziva

3.3 Prevažanje

Na platformi, ki je ravna, bi lahko prevažali tudi različne stvari. Nanjo lahko postavimo invalidski voziček, ki bi prevažal invalide po neravni površini. Na ravni zgornji površini imamo tudi šest kavljev, za katere lahko kaj privežemo. Npr. če želimo težak tovor premestiti, ga postavimo na vozilo in zaradi varnosti privežemo z vrvjo.

V našem primeru bomo na platformo pričvrstili MPS-postajo za šiljenje svinčnikov in lesenih zatičev.



Slika 24: MPS-postaja

3.4 Uporaba v industriji in gospodarstvu

Vozilo bi lahko uporabili kot lunarno vozilo, ki bi imelo nameščeno kamero ter robotsko roko. Z njim bi lahko pobirali različne kamnine oziroma odvzemali različne vzorce kamnin. Uporabljali bi ga lahko tudi v različnih proizvodnjah namesto tekočih trakov, saj lahko z njim razvažamo različne končne izdelke. S tem vozilom jih pripeljemo točno tja, kamor želimo, kadar nam tekoči trakovi tega ne omogočajo, saj so fiksno pritrjeni.



Slika 25: Industrijski voziček

Na vozilo bomo namestili GPS-senzor, da se bo premikalo preko GPS-signala in koordinat v prostoru. Narejeno bo tako, da bo vozilo poiskalo točko, kjer se nahaja, določili bi le končno

točko. Vozilo bi se pripeljalo po trajektoriji od točke A do točke B.



Slika 26: GPS-senzor

GPS-oddajnik bi moral biti zelo natančen, saj lahko zaradi manjše nenatančnosti zaide s poti in to celo za več kot 2 m. Praktična uporaba bi bila predvsem v zaprtih prostorih, kjer je GPS precej natančnejši. V vozilo se lahko vgradi tudi detektor kovin. Preko našega brezžičnega vodenja bi ga lahko vodili po površinah, kjer so potencialne arheološke znamenitosti.

Vozilo se lahko uporablja tudi kot merilnik nagibov, saj bi lahko nanj namestili giroskop, ki bi se brezžično povezal na naš računalnik in v določenem programu izpisoval graf nagiba površine, po kateri bi se vozilo premikalo. Bolj kot je strma površina, višja bi bila krivulja na grafu.



Slika 27: Giroskop

Vozilo bi lahko uporabljali tudi pri gradbenih delih. Z njim bi prevažali težke predmete, kot so vreče cementa, kovinske cevi, težki okvirji za okna, strešne kritine ipd. Za takšno delo bi na gradbišču potrebovali sicer dvigalo. Prav tako bi ga lahko uporabili za preverjanje širših betonskih cevi za kanalizacijo ter izpeljevanje potokov pod zemljo. Za to delo bi potreboval 360-stopinjski pogled okoli sebe, kar bi dosegli s kamerami. Takšne cevi bi lahko tudi popravljali, zatesnili .. Potrebovali bi močno robotsko roko ter material za zatesnitev tovrstnih cevi.

Multifunkcijsko avtonomno vozilo



Slika 268: Nadzor v težko dostopnih prostorih

Naše avtonomno vozilo bi lahko pomagalo v letalski industriji zaradi velike platforme, počasnejšega gibanja ter natančnega upravljanja. Z ojačitvijo platforme bi lahko prevažalo manjše letalske motorje in sestavne dele letal, kot so okna, manjša zakrilca in kokpiti. Z varilno roko bi lahko varili letalske rezervoarje. Prednost tega vozila je v tem, da bi imelo na roki dve kameri manjše velikosti za prepoznavanje varilne površine. To bi mu omogočalo lažji dostop kot človeku, saj bi z dovolj dolgo roko takšna zahtevna dela opravljalo od zunaj. Če bi bilo uporabljeno v letalski industriji, bi potrebovalo na platformi podoben del kot ga imajo kopači: lahko bi po hitrem postopku zamenjali več nastavkov za različna dela na letalih, ob nujni potrebi pa bi brez nastavkov na letalo s hitrejšimi elektromotorji prevažalo kovčke, potovalne torbe, zaloge vode, hrane ipd.



Slika 29:Roka s hitromenjalno pripravo

Ker se naše avtonomno vozilo upravlja na daljavo, bi lahko transportirali nevarne odpadke, kot so radioaktivne palice iz jedrskih elektrarn, odslužene bombe, kemijski odpadki ... Tako življenja ne bi bila ogrožena, saj ne bi bili v neposredni bližini odpadkov. Nadgradili bi ga z

robotsko roko za razlaganje odpadkov na lažje dostopnih mestih, s koši za nevarne odpadke, ki bi tesnili, ter z opremo, s katero bi nekatere večje odpadke pripeli. Za najnevarnejše odpadke bi potrebovali kamero z brezžičnim prenosom na prenosni računalnik, mobilni telefon in podobne naprave. Velika prednost tega vozila bi bila, da bi za varnost ljudi potrebovali posebej označeno pot, varovano z ograjami, da bi brez škode odpadke prepeljalo iz enega mesta na drugo. Tako ljudje ne bi imeli bližnjega stika z odpadki med transportom, kar bi zmanjšalo možnost zastrupitve ali poškodb.

3.5 V kmetijstvu in gozdarstvu

To vozilo lahko uporabimo tudi kot prevoznik grozdja v vinogradih, saj bi mu lahko namestili visokoprofilna kolesa ter nanj postavili velik zaboj za obrano grozdje.



Slika 30: Vozilo za razvažanje grozdja

Vozilo bi lahko zaradi velike platforme uporabljali kot gozdarsko vozilo z nastavki za podiranje manjših dreves, nato pa bi jih po delih lahko transportiralo do lažje dostopnih površin za večja vozila. Tako bi pomagalo pri učinkovitejšem gozdarstvu, ker bi ga lahko upravljali na daljavo, na njem pa bi imeli nameščene tudi štiri kamere za lažjo vidljivost in okoli vozila. S tem bi tudi povečali varnost pri podiranju dreves.

3.6 Avtonomno vozilo v turizmu

Vozilo bi lahko uporabljali v hotelirstvu za prevažanje kovčkov; za vsako sobo bi imeli napisan program, ki bi mu omogočal dostop brez daljinskega upravljanja. Ko gosti pridejo v recepcijo, bi mu v poseben majhen računalnik vnesli podatek o sobi ter mu naložili kovčke.

Ko bi s pomočjo dvigala prtljago pripeljalo do sobe, bi mu gostje le vstavili kartico za odklepanje sob, da bi potrdili prejem. To bi lahko prek brezžične povezave prejeli receptorji, ki bi mu podali ukaz, naj se vrne. Vozilo bi bilo uporabno tudi na ladjah za križarjenje, saj na njih pogosto nastavljajo koticke s hrano po zgornjih palubah, kar vzame zaposlenim veliko časa. Naše vozilo bi lahko imelo nastavljeno pot, po kateri bi imelo dostop do mest na ladji, kjer čez dan nastavljajo hrano. V kuhinji bi mu bilo le treba naložiti hrano, z dvigali za osebje bi pripeljalo hrano na zgornjo palubo, po zastavljeni poti bi se pripeljalo do določenega mesta, kjer bi ga čakal član osebja, ki bi razložil hrano z vozila, ter mu naložil staro, ki bi jo peljalo v kuhinjo.



Slika 31: Robot za strežbo hrane

Ko ne bi prevažalo hrane, bi preprosto krožilo po zastavljeni poti na zgornjih palubah in bi – zaradi nižje hitrosti – pobiralo prazne kozarce potnikov (dajali bi jih na posebej fiksiran pladenj), v poseben koš na platformi bi mu ljudje lahko oddali odpadke. Prihranilo bi veliko časa in dela na ladjah.

3.7 Vozilo za lastno uporabo

Vozilo bi lahko imeli tudi za lastno uporabo, saj bi lahko nanj postavili npr. sedež in se z njim preprosto prevažali.

4 Zaključek

Na začetku raziskovalne naloge nismo bili povsem prepričani v naše sposobnosti, saj smo si zadali kar velik projekt, a vendar nam je s pomočjo mentorjev uspelo. Pri izdelavi projekta smo se srečali z velikimi in malimi težavami, ki smo jih s timskim delom uspešno odpravili, se iz njih marsikaj pozitivnega naučili in naredili kakovosten izdelek. Prednost izdelka je predvsem v tem, da je zmogljiv, opremljen s sodobno tehnologijo in ima močno konstrukcijo, zato je po posvetovanju z izdelovalci industrijskih vozil izdelek konkurenčen na trgu.

5 Zahvala

Zahvaljujemo se mentorju g. Robertu Ojsteršku, dipl. inž. mehatronike UN, ki nam je pomagal pri vezavi in nas ob tem naučil marsikaj novega.

Zahvaljujemo se g. Marjanu Jamnišku, p. u., ki nam je pomagal pri izdelavi konstrukcije in nas ves čas spodbujal.

Zahvaljujemo se g. Matjažu Cizeju, univ. dipl. inž., za pomoč pri vezavi močnejšega toka, in samem nakupu elektronskih komponent.

Zahvaljujemo se ge. Brigiti Renner za lektoriranje naloge.

6 Viri in literatura

[1] *Arduino UNO* (online). (citirano 1. 3. 2015). Dostopno na naslovu:

<http://gadgetgangster.com/news/53-reviews/316-review-arduino-uno-part-i.html>

[2] *PingParallax* (online). (citirano 28. 2. 2015). Dostopno na naslovu:

<http://www.parallax.com/product/28015>

[3] *Robotske platforme* (online). (citirano 2. 3. 2015). Dostopno na naslovu:

https://www.google.si/search?q=robotics+platform&client=opera&hs=rja&source=lnms&tbm=isch&sa=X&ei=G5b-VLX8MujMygP-rYD4CQ&ved=0CAcQ_AUoAQ&biw=1366&bih=660

[4] *STM VNH* (online). (citirano 28. 2. 2015). Dostopno na naslovu:

<http://www.st.com/web/en/search/partNumberKeyword>

IZJAVA

Mentor, Robert Ojsteršek dipl. inž. mehatronike UN, v skladu z 2. in 17. členom Pravilnika raziskovalne dejavnosti »Mladi za Celje« Mestne občine Celje, zagotavljam, da je v raziskovalni nalogi naslovom

Multifunkcijsko avtonomno vozilo,

katere avtorji so Tilen Zidar, Dominik Nemeč, Peter Planko:

- 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 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 oziroma besedil iz naloge ob upoštevanju avtorstva in korektnem citiranju,
- da smo seznanjeni z razpisni pogoji projekta Mladi za Celje

Celje, 12.3.2015

žig šole

Podpis mentorja

Podpis odgovorne osebe