Mestna občina Celje
Komisija Mladi za Celje

# DALJINSKO UPRAVLJANJE

RAZISKOVALNA NALOGA

Avtorji:
Marko Mlaker, E4-d
Nejc Horvat, E4-d
Primož Šibanc, E4-d

Mentor:
g. Borut Slemenšek, univ. dipl. inž.

Somentor
g. Gorazd Breznik

Celje, marec 2013
Šolsko leto 2012/2013

Mestna občina Celje
Komisija Mladi za Celje

# DALJINSKO UPRAVLJANJE

Mentor:
g. Borut Slemenšek, univ. dipl. inž.

Somentor
g. Gorazd Breznik

Avtorji:
Marko Mlaker, E4-d
Nejc Horvat, E4-d
Primož Šibanc, E4-d

Celje, marec 2013
Šolsko leto 2012/2013

# Kazalo vsebine



# Kazalo slik

## Povzetek

S pomočjo osnovnih elektrotehničnih sestavin, predhodnih izkušenj ljudi s svetovnega spleta in lastno ustvarjalnostjo smo izdelali programsko kodo za avtomobil, ki deluje na odprtokodnem krmilniku in je sposoben prenašati sliko na računalnik s pomočjo brezžične tehnologije, prav tako pa ga je mogoče upravljati na daljavo.

## Abstract

With the help of basic electrical components, other people's experience and our own creativity, we developed a program code for a radio controlled device, which is based on wireless technology, operates on open source microcontroller and is capable of transmitting live picture feed to computer. It can also be remote controlled.



## Ključne besede

Arduino
Radijsko vodene naprave
Brezžičen prenos slike
Krmiljenje na daljavo

## Keywords

Arduino
Radio controlled devices
Wireless video capture
Remote control

# 1. Uvod

Pri četrti izpitni enoti poklicne mature smo se Marko Mlaker, Nejc Horvat in Primož Šibanc odločili, da bomo prikazali vožnjo radijsko vodenega avtomobila, ki bo ukaze, ki so neposredno povezani s samim nadzorom avtomobila, prejemal preko brezžične računalniške povezave z uporabo programa Drive Different.

## 1.1 Predstavitev problema

Uspešno želimo izdelati vohunski avtomobil, ki se ga da voziti preko računalnika. Namen je ugotoviti način komunikacije navadne radijsko vodene naprave z računalnikom, tako da bodo stroški čim manjši, način izvedbe pa čim bolj preprost. To pomeni, da je uporaben tako za domačo kot tudi za zabavno rabo. Z določenimi popravki je možno avtomobil prilagoditi tako, da bo njegova uporaba tudi na profesionalnem nivoju.

## 1.2 Cilji

Postavili smo si točno določene cilje, ki so se nam zdeli nujni pri uresničitvi projekta vodenja avtomobila:
- Avtomobil je treba voziti preko računalnika z grafičnim vmesnikom, prijaznim do uporabnika
- Poleg vožnje in upravljanja avtomobila želimo še sproten prenos slike
- Upravljanje z avtomobilom bo možno tudi preko spletne strani

# 2. Vsebina

## 2.1 Teoretično ozadje

Začetek projekta se je pričel z ugotavljanjem, kako avtomobilčki delujejo, kaj vsebujejo in kako izgleda notranja zgradba. Začeli smo z raziskovanjem po različnih spletnih virih v upanju, da bomo prišli do koristnih informacij, ki bi nam olajšale delo. Prišli smo do ugotovitve, da je izpeljava možna celo na več načinov, kar je bila za nas dobra novica, saj smo lahko ostali pri naši ideji za maturitetno nalogo.
Tako smo začeli z nabiranjem gradiva in zapiskov ljudi, ki so se s podobnimi idejami ukvarjali že pred nami. Zaradi ozkega spektra nam koristnih informacij smo začeli z lastnim preučevanjem delovanja radijsko vodenih avtomobilov, kako naj bi to izgledalo v praksi, dokupili pa smo tudi osnovne dele, kot so arduino (odprtokoden elektronski mikrokrmilnik, na katerem je mogoče ustvariti interaktivne elektronske objekte), avtomobil, ki omogoča radijsko vodenje, žice, pribor za spajanje, testno ploščico in ostale stvari.

## 2.2 Postopek računalniške povezave

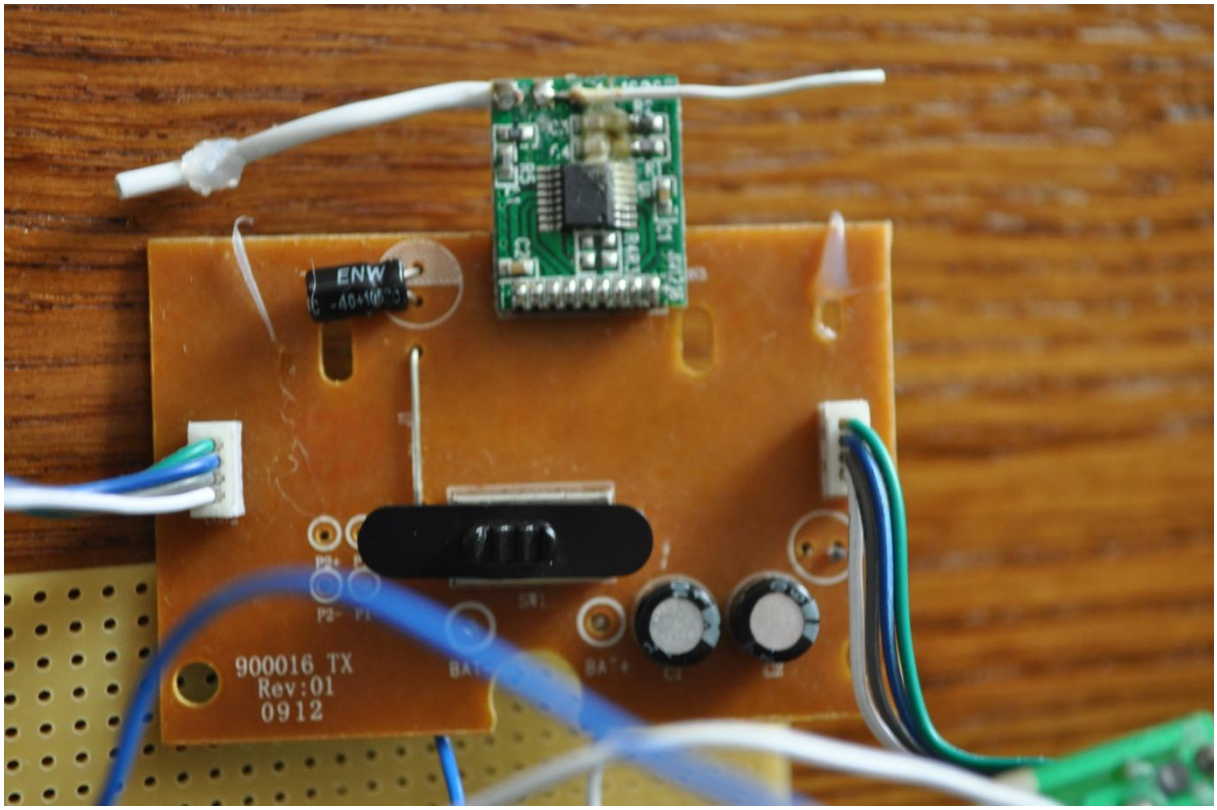
Za povezavo avtomobila z računalnikom potrebujemo:

- dvokanalno radijsko vodeno napravo, v našem primeru je to avtomobil Carrera Red Jumper
- čip UL N2803A
- približno dva metra bakrenih žic (lahko tudi manj, odvisno od racionalnosti uporabe)
- letvice za elektrotehniko
- Arduino UNO - SMD edition ali pa navadnega (navaden omogoča menjavo čipa)
- osnovni elektrotehnični pribor (klešče, izvijači, multimeter, nožek, testna ploščica)

Cena projekta, ki ne zavzema elektrotehničnega pribora znaša približno 50€. Za naš projekt smo posegli po malce dražjih sestavnih delih, zato je bil denarni vložek višji.

*Slika 1: Carrera Red Jumper - avtomobil, ki smo ga uporabili za vodenje.*

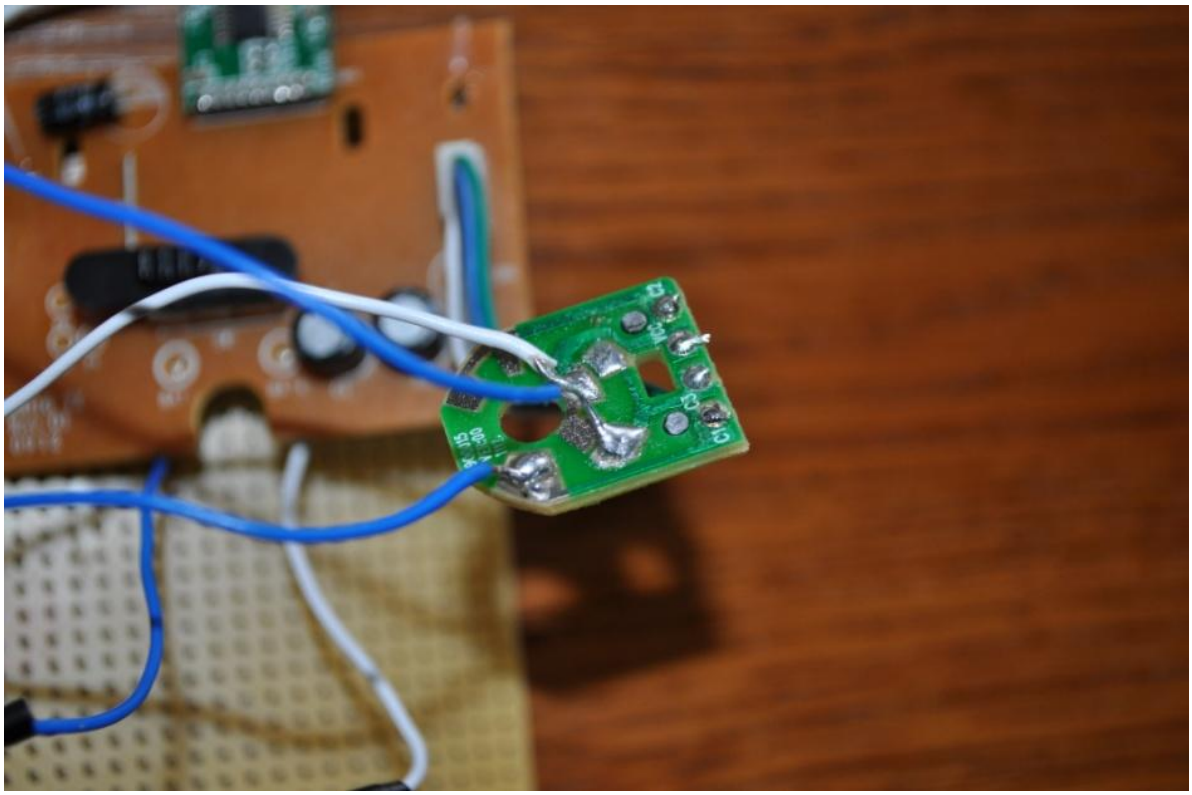
Najprej vstavimo v radijsko voden avtomobil in njegov oddajnik baterije, da se prepričamo, če v osnovi avtomobil deluje. Ko preverimo delovanje funkcij, ga ugasnemo in razdremo oziroma odstranimo plastični del oddajnika, da dobimo le krmilno ploščico s tipkami.



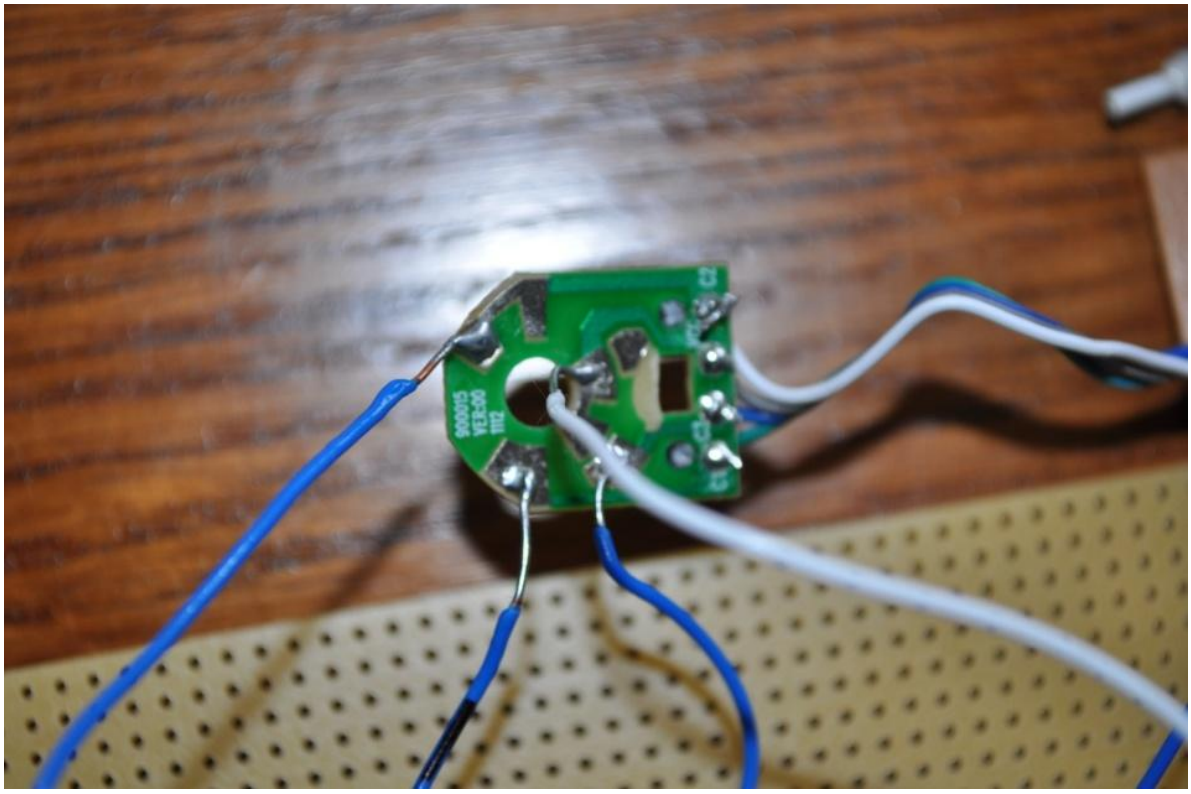

*Slika 2: Krmilno vezje radijsko vodene naprave.*

Sedaj moramo ugotoviti, kje se nahaja tok, ko je daljinski upravljalnik vključen, vendar tipka ni pritisnjena. To moramo narediti za vsako tipko posebej, ker moramo tja, kjer primarno ni električne napetosti, spojiti žice, po katerih bomo kasneje pošiljali signale.

Ko ugotovimo povezave, nanje spojimo žice, ki jih bomo kasneje potrebovali za krmiljenje.



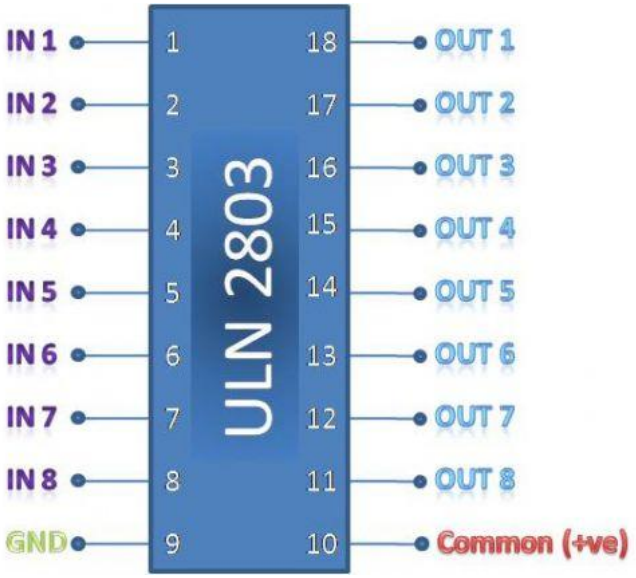
*Slika 3: Tipala krmiljenja levo in desno.*



*Slika 4: Tipala krmiljenja naprej in nazaj.*

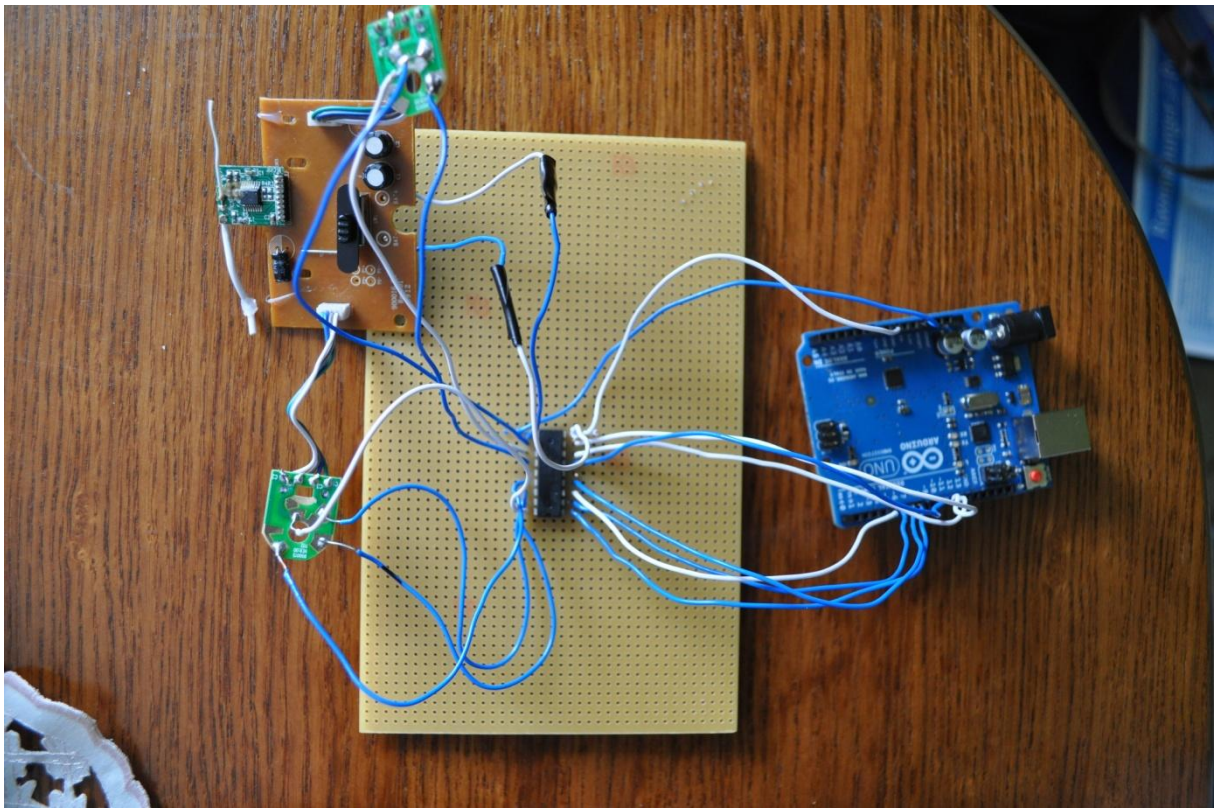
Ko smo opravili povezave do krmilnih ploščic, postavimo čip UL N2803A v testno ploščico. Sedaj povežemo žice z oddajnika na čip po naslednjih korakih:

1. Žice z oddajnika na čip povežemo na desno stran čipa, ki vsebuje manjšo vzboklino na vrhu.
2. Prva desna noga je napajanje. Tja povežemo plus (simbol +) z oddajnika.
3. Prostih imamo še osem nogic za priklop, izberemo pa lahko katerokoli. Izberemo želeno in tja priključimo žice s tipke levo, desno, naprej in nazaj.
4. Povezati moramo še nogice, ki smo jih izbrali za smeri na mikrokrmilnik arduino in na drugo stran čipa. Na arduinu izberemo digitalne pine. V našem primeru smo izbrali pine od pin13 do pin8 + pin4.
5. Za povezavo recimo izberemo pin13, povežemo na INput1 in tako bo signal poslan z računalnika na pin13 poslan na INput1 in bo odšel kot izhod na OUTput1.
6. Za konec povežemo še minus (-) z oddajnika na GND na čipu in iz čipa na GND na arduina.

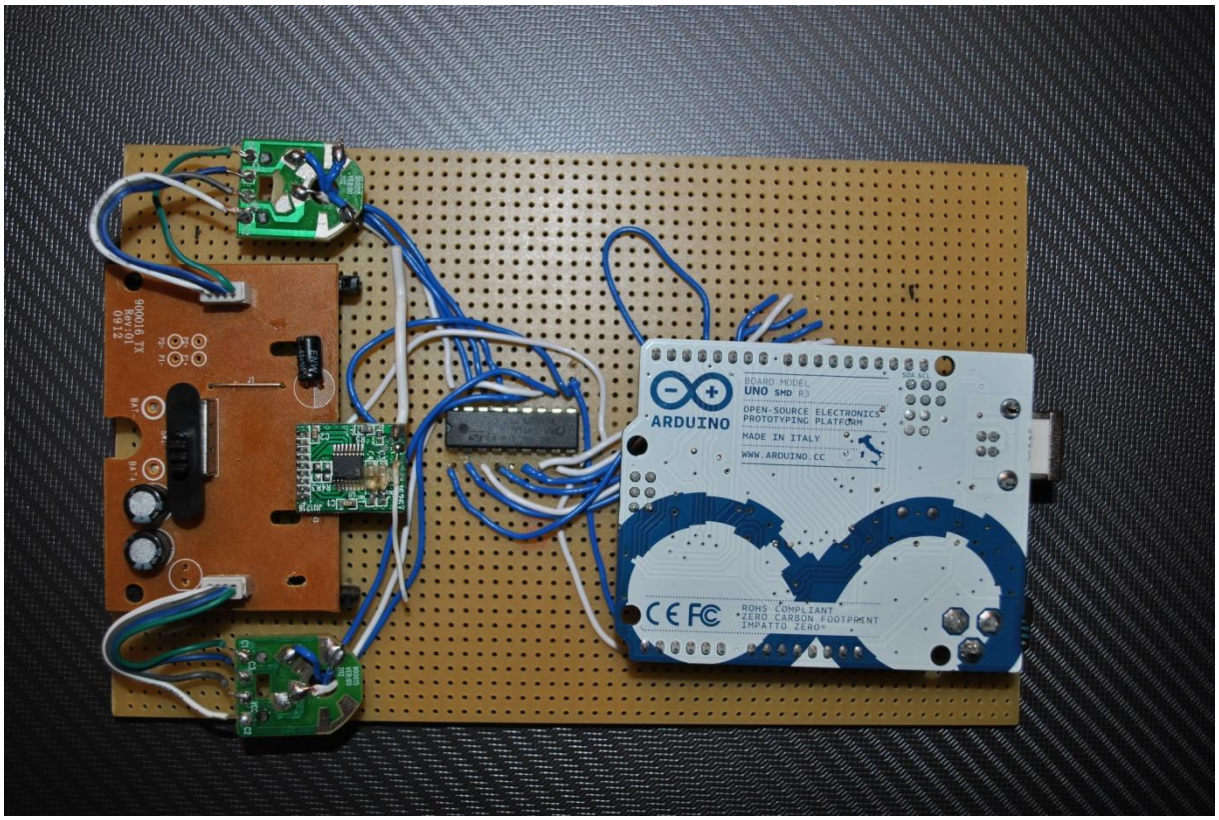



*Slika 5: Čip ULN2803A s prikazanimi nogicami.*

Naše povezave bi morale izgledati tako:



*Slika 6: Vzpostava vseh povezav za komuniciranje, ki še niso urejene.*




*Slika 7: Urejene vzpostavljene povezave.*

Tako imamo ustvarjene fizične povezave z mikrokrmilnika do oddajnika, kar pomeni, da je delo s spajanjem končano in nas čaka le še pisanje programa.

Najprej moramo namestiti gonilnike, da naš računalnik prepozna Arduina. To nam je povzročalo veliko težav, čeprav smo delali po postopku, ki je opisan na njihovi uradni spletni strani. Po številnih neuspešnih nameščanjih, nam je na koncu le uspelo. Zanimivo je, da postopka nismo spreminjali.

Ko so gonilniki nameščeni, potrebujemo še programsko razvojno orodje. Dobili smo ga brezplačno, prav tako na njihovi uradni spletni strani, pod zavihkom »Download«. Dobimo ga s paketom gonilnikov, ki smo jih potrebovali v prejšnjem koraku. Prednost Arduina je v tem, da je odprtokoden, kar pomeni, da je koda vidna uporabniku, in jo je mogoče urejati. Programiranje je tukaj podobno programiranju programskega jezika C++, ki nam je bilo sprva tuje in nam je povzročalo nemalo preglavic. S pomočjo vodičev, ki so dostopni na svetovnem spletu in ob ponujenih osnovnih primerih smo le uspeli ustvariti program, katerega smo poslali na arduina.

### 2.3 Programska koda mikrokrmilnika

Ko smo usposobili program Arduino, smo potrebovali dodaten program, ki bo tekel v Windows okolju in ki bo skrbel za ustrezno komunikacijo med Arduinom in Windows operacijskim sistemom. Za to smo izbrali program Microsoft Visual Studio 2010. Ustvarili smo Windows form aplikacijo - okensko aplikacijo, ki je enostavna za narediti, le poiskati smo morali, kako je poskrbljeno s serijsko komunikacijo, ki je bila za nas nova, saj v šoli nismo dobili tega znanja.

Tako nam je uspelo napisati program, ki smo ga poimenovali Drive Different. Ideja za ime projekta se nam je porodila, ko smo iskali sliko, katero bi namestili za ozadje našega programa.

Pripadajoča programska koda mikrokrmilnika v celoti:

```
void setup()
{
        Serial.begin(9600); //hitrost prenosa, enako na obeh
  pinMode(13, OUTPUT);   //levo
  pinMode(12, OUTPUT);   //vedno levo in desno
  pinMode(11, OUTPUT);   //desno
  pinMode(10, OUTPUT);   //hitro naprej
  pinMode(9, OUTPUT);    //naprej
  pinMode(8, OUTPUT);    //nazaj
  pinMode(4, OUTPUT);    //vedno naprej in nazaj
}

void loop()
{
        if(Serial.available()>0)
        {
                int tipka = Serial.read();
                    if(tipka == 1)
                    {
                            digitalWrite(12, 1);   //desno prizges
                  digitalWrite(11, 1);
                    }

                    if(tipka == 2)
                    {
                            digitalWrite(12, 1); //levo
                  digitalWrite(13, 1);
                    }

                    if(tipka == 3)
                    {
                            digitalWrite(4, 1);  //naprej
                  digitalWrite(10, 1);
                    }
```

```
          if(tipka == 4)
          {
                    digitalWrite(4, 1); //nazaj
        digitalWrite(9, 1);
          }

if(tipka == 9)
          {
                  digitalWrite(4, 1); //nazaj hitro
        digitalWrite(8, 1);
          }

 if(tipka == 20)
 {
        digitalWrite(4, 1); //naprej hitro
        digitalWrite(10, 1);
        digitalWrite(9, 1);
        digitalWrite(8, 1);
 }

          //ugasnes

          if(tipka == 5)
          {
                    digitalWrite(12, 0);   //desno
        digitalWrite(11, 0);
          }

          if(tipka == 6)
          {
                    digitalWrite(12, 0); //levo
        digitalWrite(13, 0);
          }

          if(tipka == 7)
          {
                    digitalWrite(4, 0);  //naprej
        digitalWrite(10, 0);
          }

          if(tipka == 8)
          {
                    digitalWrite(4, 0); //nazaj
        digitalWrite(9, 0);
          }



          if(tipka == 10)
          {
                  digitalWrite(4, 0); //nazaj hitro
```

```
                digitalWrite(8, 0);
                  }

            if(tipka == 21)
            {
                digitalWrite(4, 0); //naprej hitro
                digitalWrite(10, 0);
                digitalWrite(9, 0);
                digitalWrite(8, 0);
            }
        }
}
```

## 2.4 Programska koda razvite okenske aplikacije


*Slika 8: Izgled Drive Different aplikacije.*

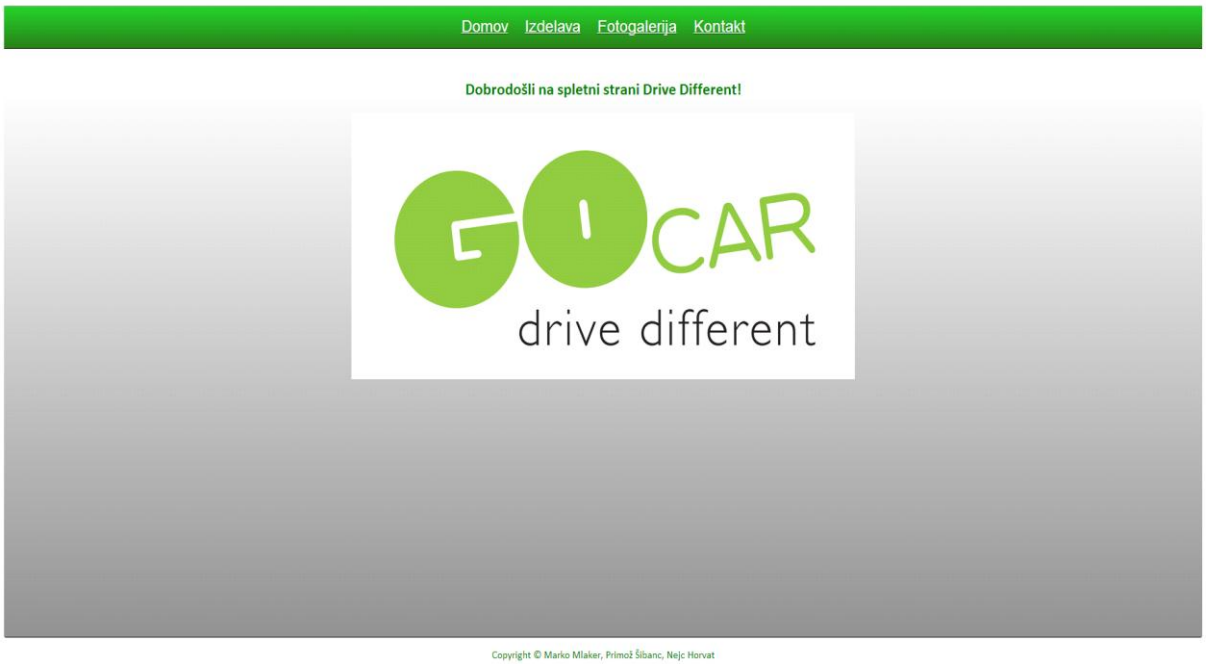
Programska koda, napisana v jeziku C#:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace Mlaker
{
   public partial class Form1 : Form
   {
      public Form1()
      {
         InitializeComponent();
      }

      private void Form1_KeyDown(object sender, KeyEventArgs e)
      {
         if (e.KeyCode == Keys.Left)
         {
            Poslji(2);
         }
         else if (e.KeyCode == Keys.Right)
         {
            Poslji(1);
```

```
    }
    else if (e.KeyCode == Keys.Up)
    {
        Poslji(4);
    }
    else if (e.KeyCode == Keys.Down)
    {
        Poslji(3);
    }
    else if (e.KeyCode == Keys.Space)
    {
        Poslji(9);
    }
    else if (e.KeyCode == Keys.ShiftKey)
    {
        Poslji(20);
    }
}

void Poslji(byte poslji)
{
    try
    {
        byte[] b = new byte[1];
        b[0] = poslji;
        serialPort1.Write(b, 0, 1);
    }

    catch
    {
        MessageBox.Show("Napaka pri posiljanju");
    }
}

private void buttonOdpri_Click(object sender, EventArgs e)
{
    if (buttonOdpri.Text == "Odpri")
    {
        serialPort1.PortName = textBoxPort.Text;
        serialPort1.BaudRate = int.Parse(textBoxHitrost.Text);
        serialPort1.Open();
        buttonOdpri.Text = "Zapri";
    }
    else
    {
        serialPort1.Close();
        buttonOdpri.Text = "Odpri";
    }
}
```

```
        private void Form1_KeyUp(object sender, KeyEventArgs e)
        {
            if (e.KeyCode == Keys.Left)
            {
                Poslji(6);
            }
            else if (e.KeyCode == Keys.Right)
            {
                Poslji(5);
            }
            else if (e.KeyCode == Keys.Up)
            {
                Poslji(8);
            }
            else if (e.KeyCode == Keys.Down)
            {
                Poslji(7);
            }
            else if (e.KeyCode == Keys.Space)
            {
                Poslji(10);
            }

            else if (e.KeyCode == Keys.ShiftKey)
            {
                Poslji(21);
            }
        }
    }
}
```

## 2.5 Izdelava spletne strani



Slika 9: Končna različica spletne strani.

Spletna stran Drive different je nastajala v okviru projekta raziskovalne naloge Drive different. Izdelava spletne strani je potekala v razvojnem orodju Microsoft Visual Studio 2010.

Spletna stran sestoji iz glavne strani, ter štirih spletnih obrazcev.

Oblika naše spletne strani je nastala s spletnim jezikom CSS (Cascading Style Sheets), sama spletna stran pa v jeziku ASP.NET (Active server pages).

Za obliko in izgled glavne strani smo uporabili CSS jezik, ki je viden na slikah spodaj:

```
   .navigacija          /*navigacijska vrstica*/
{
   border-top: 1px solid #8f9294;
   background: #287d18;
   background: -webkit-gradient(linear, left top, left bottom, from(#23cf29), to(#287d18));
   background: -webkit-linear-gradient(top, #23cf29, #287d18);
   background: -moz-linear-gradient(top, #23cf29, #287d18);
   background: -ms-linear-gradient(top, #23cf29, #287d18);
   background: -o-linear-gradient(top, #23cf29, #287d18);                    /* Barva odzadja navigacijske vrstice*/
   padding: 16px 32px;
   -webkit-border-radius: 0px;
   -moz-border-radius: 0px;
   border-radius: 0px;
   -webkit-box-shadow: rgba(0,0,0,1) 0 1px 0;
   -moz-box-shadow: rgba(0,0,0,1) 0 1px 0;                                    /* oblika robov navigacijske vrstice*/
   box-shadow: rgba(0,0,0,1) 0 1px 0;
   text-shadow: rgba(0,0,0,.4) 0 1px 0;
   text-align:center;
}
.navigacija li                                                               /* stil pisave v navigacijski vrstici*/
{
   font-size: 24px;
   font-family: 'Lucida Grande', Helvetica, Arial, Sans-Serif;
   vertical-align: middle;
   list-style:none;
   margin:0;
   padding:0;
   text-align:center;
   display:inline;
}
.navigacija a{                                                               /* odmik med naslovi v navigacijski vrstici*/
    padding:10px;
}
```

*Slika 10: Prikaz CSS kode navigacije.*

```
.contentplaceholder                                                  /* velikost, barva, odzadje ContentPlaceHolder-ja*/
{
    width:auto;
    height:815px;
    background: #8f8f8f;
    background: -webkit-gradient(linear, left top, left bottom, from(#ffffff), to(#8f8f8f));
    background: -webkit-linear-gradient(top, #ffffff, #8f8f8f);
    background: -moz-linear-gradient(top, #ffffff, #8f8f8f);
    background: -ms-linear-gradient(top, #ffffff, #8f8f8f);
    background: -o-linear-gradient(top, #ffffff, #8f8f8f);
    padding: 8.5px 17px;
    -webkit-border-radius: 3px;
    -moz-border-radius: 3px;
    -webkit-box-shadow: rgba(0,0,0,1) 0 1px 0;
    -moz-box-shadow: rgba(0,0,0,1) 0 1px 0;
    box-shadow: rgba(0,0,0,1) 0 1px 0;
}

li:hover                                                             /*Hover funkcija pisave v navigacijski vrstici*/
{
    background-color:gray;
}

li:active                                                           /* Klik na naslov v navigacijski vrstici*/
{
    background-color:#0046A8;
}
```

*Slika 11: Prikaz CSS kode vsebinskega prostora.*

Za lažjo preglednost so določeni deli komentirani (v zeleni barvi).

# 3. Zaključek

Naš projekt smo pripeljali do konca. Pri delu smo se naučili veliko stvari na področju elektronike, ki nam je manj poznana, in se nam je od 2 letnika izobraževanja že kar oddaljila. Pri vezavi električnih sestavin smo naleteli na težave. Sprva smo mislili uporabiti releje tudi kot stikala, le-ta pa so bila velika, počasna, hitrost avtomobila pa ne bi morali upravljati.

Ta problem smo odpravili z uporabo čipa, imenovanega ULN2803A. Čip deluje kot rele, a je po velikosti manjši in od ponujenih osmih tipk hkrati.

Težave in napake so se pojavile tudi pri pisanju programske kode, saj je sprva delovala samo desna smerna tipka. To smo odpravili s podrobnejšim pregledom kode.

Skoraj pa je ostala napaka hitre vožnje. Namesto, da bi avtomobil lahko peljali s povečano hitrostjo naprej, se je pomikal nazaj. Omenjeno napako smo odpravili s temeljitim pregledom kode in samih komponent. Tako smo ugotovili, da je za to dejanje potrebno hkratno pošiljanje kar štirih signalov.

Možnosti za izboljšavo projekta je ogromno, a smo bili omejeni predvsem s finančnimi sredstvi. Ena od teh možnosti je izboljšava oz. podaljšanje dometa, a bi bil vložek na tej točki predrag, tega pa nam naš proračun žal ni omogočal. Izboljšave bi lahko storili s komponentami Xbee, ki se priključijo na mikrokrmilnik in računalnik ali pa z mikrokrmilnikom, ki deluje na brezžični tehnologiji - arduina bi priključili na sam avtomobil in bi tam izvedli potrebne prilagoditve. Tako sploh ne bi potrebovali oddajnika avtomobila, saj bi za to poskrbelo brezžično omrežje našega računalnika oz. usmerjevalnika.

Dodali bi mu lahko tudi GPS oddajnik, ki se priključi na mikrokrmilnik in bi tako oddajal signal nahajališča, to pa bi lahko izrisovali na zemljevid v realnem času. Tako bi točno vedeli, kje se kaj dogaja. S tem bi naš avtomobil postal pravi vohunski avtomobil z dometom od 100 do 200 metrov, ob morebitni uporabi ojačevalnikov brezžičnega signala pa bi se dal domet znatno povečati.

Namesto android mobilnika za kamero bi lahko uporabili, posebno, namensko IP kamero. S tem bi zavzeli manj prostora na avtomobilu, vendar so elegantne rešitve v večini primerov tudi dražje, kar drži tudi tukaj.

# 4. Viri in literatura


OSNOVE PROGRAMIRANJA ZA ARDUINO. [online]. Zmaga: Spletno pridobivanje znanja. [Zadnja sprememba 11. jan. 2013]. [Citirano 13. mar. 2013; 15:55]. Dostopno na spletnem naslovu: http://www.zmaga.com/content.php?l=0&id=4511

JESPERSEN, T. Wifi controlled RC car with the Arduino. [online]. TKJ Electronics: Development with ease. [Zadnja sprememba 8. feb. 2011]. [Citirano 13. mar. 2013; 16:00]. Dostopno na spletnem naslovu: http://blog.tkjelectronics.dk/2011/02/wifi-controlled-rc-car-with-the-arduino/

HOW TO BUILD A PC CONTROLLED RC CAR. [online]. [Zadnja sprememba avg. 2004]. [Citirano 13. mar. 2013; 16:03]. Dostopno na spletnem naslovu: http://www.jbprojects.net/articles/rc/

WIFI ROBOT. [online]. [Zadnja sprememba: avg. 2008]. [Citirano 13. mar. 2013; 18:53]. Dostopno na spletnem naslovu: http://jbprojects.net/projects/wifirobot/

RC CAR CONTROL PROGRAMMING. [online]. [Zadnja sprememba 11. jan. 2012]. [Citirano 13. mar. 2013; 16:55]. http://www.codeproject.com/Articles/126859/RC-Car-Control-Programming

PEEK, B. Computer-Controlled R/C Car with Camera. [online]. [Zadnja sprememba 21. jan. 2007; 23:51]. [Citirano 13. mar. 2013; 17:08]. Dostopno na spletnem naslovu: http://channel9.msdn.com/coding4fun/articles/Computer-Controlled-RC-Car-with-Camera



WIFI ROBOT. [online]. [Zadnja sprememba: avg. 2008]. [Citirano 13. mar. 2013; 18:53]. Dostopno na spletnem naslovu: http://jbprojects.net/projects/wifirobot/



PUT YOUR RC CAR UNDER COMPUTER CONTROL. [online]. Dostopno na spletnem naslovu: http://www.instructables.com/id/Put-your-RC-car-under-computer-control/step10/Done-at-last/