

ŠOLSKI CENTER VELENJE
ELEKTRO IN RAČUNALNIŠKA ŠOLA

MLADI RAZISKOVALCI ZA RAZVOJ ŠALEŠKE DOLINE

RAZISKOVALNA NALOGA

PNEVMATSKI SIMULATOR VOŽNJE

Tematsko področje: Tehnika

Avtorji:

Aleksander Turnšek, 3. letnik
Blaž Sitar, 3. letnik
Samo Železnik, 3. letnik

Mentor:
Gregor Hrastnik

Somentorja:
Uroš Remenih
Benjamin Krušič

Velenje, 2012

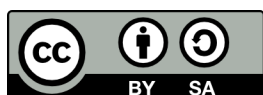
Raziskovalna naloga je bila opravljena na ŠC Velenje, Elektro in računalniška šola, 2012.

Mentor: Gregor Hrastnik, univ. dipl. inž. rač. in inf.

Somentorja: Uroš Remenih, abs. FE, smer telekomunikacije
Benjamin Krušič, dipl.ing.str.

Datum predstavitev: marec 2012

Uporabili smo licenco Creative Common (Priznanje avtorstva - <http://creativecommons.si/node/249>,
<http://creativecommons.org/licenses/by-sa/2.5/si/>).



BY: Aleksander Turnšek, Blaž Sitar, Samo Železnik, Uroš Remenih

KLJUČNA DOKUMENTACIJSKA INFORMACIJA

KG Simulator, pnevmatski cilindri, Live for Speed, X-simulator

AV Aleksander TURNŠEK / Blaž SITAR / Samo ŽELEZNIK

Mentor HRASTNIK, Gregor

KZ 3320 Velenje, SLO, Trg mladosti 3

ZA ŠC Velenje, Elektro in računalniška šola, 2012

LI 2012

IN *Pnevmatski simulator vožnje*

TD RAZISKOVALNA NALOGA

OP *IX, 33 s., 0 tab., 26 sl., 0 prilog*

IJ SL

JI En

AI sl

Namen raziskovalne naloge je bil raziskati, ali je možno izvoziti digitalne podatke ter jih pretvoriti v elektro signale, katere lahko uporabimo za krmiljenje pnevmatskih ventilov. Raziskovalna naloga zahteva znanje iz tako programskega kot tudi elektro in strojnega področja. Pri dokončanem simulatorju računalnik pošlje podatke o silah premikanja iz računalniške igre na kontrolno ploščo s pomočjo programa "x-simulator", ta pa pošlje podatke na relejni krmilnik za pnevmatske cilindre, ki premikajo ogrodje simulatorja. Ti cilindri nam nato s pomočjo dviganja in spuščanja predstavijo občutek vožnje kot v pravem dirkalnem avtomobilu. V našem primeru to deluje le v teoriji, saj se sestave samega simulatorja nismo lotili. Postavili smo si tudi dve hipotezi. Ena izmed njih govori o tem, da lahko izvozimo podatke o silah iz računalniške igre, druga hipoteza pa govori o tem, da lahko podatke o teh silah prenesemo na krmilno ploščo. Za izdelavo krmilne plošče smo morali dele kupiti ter jih sestaviti. Programsko opremo za izvoz podatkov iz iger smo dobili brezplačno na internetu.

KEY WORDS DOCUMENTATION

ND ŠC Velenje, Poklicna in tehniška elektro in računalniška šola, 2011

CX Simulator, pneumatic cylinders, Live for Speed, X-simulator

AU Aleksander TURNŠEK / Blaž SITAR / Samo ŽELEZNIK

Mentor: HRASTNIK, Gregor

PP 3320 Velenje, SLO, Trg mladosti 3

PB ŠC Velenje, Elektro in računalniška šola, 2012

PY 2012

TI *Pnevmatski simulator vožnje*

DT RESEARCH WORK

NO IX, 33 p., 0 tab., 26 fig., 0 app.

LA SL

AL sl/en

AB The purpose of the study was to investigate whether it is possible to export digital data and convert it into electrical signals, which can be used for controlling the pneumatic valves. Research work requires knowledge of programming as well as electrical and mechanical fields. In case of the complete simulator, computer sends data of movement forces from the computer game to the control panel, using the "x-simulator" programme, which sends data to the relay controller for pneumatic cylinders that move the simulator's frame. These cylinders then use the lift and descent functions to emulate the feeling of driving a real racing car. In our case, this only works in theory, since we did not deal with the assembly of the simulator itself. We came up with two hypotheses. One of them says that you can export data of the forces from the computer game, whereas the second hypothesis says that information of the forces can be transferred to the control panel. In order to manufacture the control panel, we had to buy the parts and assemble them. Software for data export from the games was given to us for free on the Internet.

Kazalo vsebine

1 UVOD.....	7
1.1 Hipoteze.....	7
2 PREGLED STANJA TEHNIKE.....	8
2.1 Kaj so simulacije?.....	8
2.2 Zgodovina simulacij.....	8
2.3 Trenutno razvijanje.....	9
2.4 O simulatorjih.....	9
2.5 Simulacija Live for speed.....	11
3 CILJI, METODE RAZISKOVANJA.....	14
3.1 X-Simulator.....	14
3.2 Ogradje simulatorja v programu Solidworks.....	19
3.3 Ogradje simulatorja v programu Blender.....	20
3.4 Kontrolni plošči	23
3.5 Velleman k8055d.....	23
3.5.1 Programska oprema.....	23
3.5.2 Strojna oprema.....	26
3.5.3 Sestavljanje.....	27
3.6 Tranzistorska plošča.....	27
4 REZULTATI IN RAZPRAVA.....	28
5 ZAKLJUČEK.....	29
6 POVZETEK.....	30
7 ZAHVALA.....	31
8 VIRI IN LITERATURA.....	32
9 AVTORJI RAZISKOVALNE NALOGE.....	33
9.1 Samo Železnik.....	33
9.2 Blaž Sitar.....	33
9.3 Aleksander Turnšek.....	33

Kazalo slik

Slika 1: Informacije o dirkalniku.....	11
Slika 2: Nastavljanje barv.....	11
Slika 3: Dirkalnik s teksturo in poljubnimi barvami.....	11
Slika 4: Nastavitve zavor pri dirkalniku.....	12
Slika 5: Nastavitve podvozja na dirkalniku.....	12
Slika 6: Nastavitve upravljanja koles.....	12
Slika 7: Nastavitve prenosov in sklopke.....	13
Slika 8: Nastavitve gum spredaj in zadaj.....	13
Slika 9: Nastavitve potnikov v dirkalniku.....	13
Slika 10: Uporabniški vmesnik programa Force sender.....	15
Slika 11: Vtičnik outsim.....	15
Slika 12: Nastavljanje sekvenc za pozitivno smer.....	16
Slika 13: Nastavljanje sekvenc za negativno smer.....	16
Slika 14: Izpis podatkov o silah iz igre preko programa force sender.....	17
Slika 15: Izpis podatkov, ki jih program pošilja na kontrolno ploščo.....	17
Slika 16: Nastavljanje sekvenc ob mirovanju.....	18
Slika 17: Izgled celotnega zaslona: levo zgoraj program force profiler, desno zgoraj simulacija Live for Speed, levo spodaj program force sender, desno spodaj vtičnik outsim.....	18
Slika 18: Načrt ogrodja v programu SolidWorks.....	19
Slika 19: Omogočanje uvoza datotek .STL formata.....	20
Slika 20: Izbira Cycles pogona za upodabljanje.....	21
Slika 21: Upodobljeno ogrodje iz zadnjega pogleda.....	21
Slika 22: Upodobljeno ogrodje iz sprednjega pogleda.....	22
Slika 23: Upodobljeno ogrodje iz stranskega pogleda.....	22
Slika 24: Otroški uporabniški vmesnik.....	24
Slika 25: Visual basic uporabni vmesnik.....	24
Slika 26: Kompakten uporabniški vmesnik.....	25

UVOD

Namen raziskovalne naloge je bil iz računalniške simulacije Live for Speed pridobiti informacije o silah, ki delujejo na avtomobil in voznika. Med izdelavo raziskovalne naloge smo morali predelati kar nekaj snovi o samih simulatorjih in simulacijah ter programih, ki iz igre pridobijo podatke o silah. Da smo se prepričali če smo pravilno pridobili podatke o silah iz računalniške simulacije, smo izedlali kontrolno ploščo, ki nam preko LED-diod prikazuje delovanje sil. S pomočjo te kontrolne plošče lahko izdelamo pravi simulator s cilindri. V raziskovalni nalogi smo predstavili programski paket za izpisovanje podatkov o silah "x-simulator", kontrolno in tranzistorsko ploščo, računalniško simulacijo dirkanja Live for Speed, CAD-orodje SolidWorks in na splošno o dirkalnih simulatorjih.

Hipoteze

Izbrali smo si dve hipotezi, ki se navezujejo na izbran projekt, in sicer:

- ⤴ Lahko izvozimo podatke o silah iz računalniške igre.
- ⤴ Lahko podatke o teh silah prenesemo na krmilno ploščo.

Prav te hipoteze smo si izbrali zato, ker je bil to naš cilj že od začetka, saj smo vedeli da za dokončanje celotnega simulatorja verjetno ne bomo imeli dovolj časa ali pa dovolj materialnih sredstev.

PREGLED STANJA TEHNIKE

Kaj so simulacije?

Sim (simulacije) je skupni izraz za računalniško opremo, ki poskuša čim bolj natančno posnemati dirkanje avtomobila s spremenljivkami iz resničnega sveta, kot so poraba goriva, poškodbe, obraba in oprijem gum in nastavitve vzmetenja. Da bi lahko bili v simulacijah konkurenčni, mora voznik razumeti vse vidike ravnanja z avtomobilom, kar naredi dirkanje v realnem svetu tako zahtevno. Nekaj najbolj zahtevnih stvari je zaviranje na meji, nadzor avtomobila ob zdrsih gum in kako pravilno pripeljati v ovinek in iz njega, da bo ohranil optimalno hitrost. To je nivo zahtevnosti, ki loči simulacije od arkadnih dirkaških iger, kjer so iz formule vzete vse spremenljivke iz realnega življenja in je glavni cilj ustvariti občutek hitrosti, ne pa občutka realizma.

Na splošno velja, da so igre, kot so Grand Prix Legends, iRacing.com, rFactor, F1 Challenge, GTR2, Live For Speed, netKar Pro, Project C.A.R.S. manj popularne kot na primer Need for Speed serija zaradi tega, ker je v njih potrebno vložiti ogromno truda in časa, ker potrebujejo veliko spretnosti, da jih obvladamo. Kljub temu so nekatere simulacije uspele, dve najbolj popularni sta NASCAR Racing 2003 in Richard Burns Rally, ki sta doživeli svetovno slavo. Nekatere igre pa niso uspele tudi zaradi omejitve strojne opreme pri računalnikih. Simulacije imajo zahtevne fizikalne pogoje, ki zahtevajo veliko od računalnika, zato ima marsikatera simulacija slabše grafične lastnosti od nekaterih arkadnih iger, da lahko deluje nemoteno na večini računalnikov, ki dosegajo minimalne zahteve za poganjanje igre. Ker moramo biti pri simulacijah ob vožnji zelo natančni, je zelo priporočljivo imeti volan in pedalke za plin in zavoro. Ker pa je tehnologija v zabavni elektroniki že tako napredovala, pa lahko sedaj kupimo tudi volane z obračalnim krogom do 1080°, povratnim učinkom, pedalke s pedalko za plin, zavoro in sklopko, ki nam zelo koristi če imamo ročne menjalnike, ki so na voljo v H-stilu (kot v pravem avtomobilu) in imajo 7 ali 8 stopenj, ali pa imamo sekvenčni menjalnik, ki ima le pozicijo za prestavljanje v višjo prestavo in pozicijo za prestavljanje v nižjo prestavo. Večina arkadnih iger pa se brez problema igra na tipkovnici, miški ali na joystick kontrolerju.

Z razvojem igranja preko mreže imamo možnost tekmovati proti človeškim nasprotnikom, ki za razliko od računalniških voznikov z umetno inteligenco med igranjem mislijo, načrtujejo taktike in se tudi pod pritiskom zmotijo, je to najbližje, kako lahko pridemo do resničnega dirkanja, ne da bi sedli v sedež pravega dirkalnika. Tudi tisti, ki dirkajo na resničnih tekmovanjih, uporabljajo simulacije za treninge ali pa za zabavo. Z razvojem fizikalnih pogonov, okoli se gradijo igre in z izboljšanjem strojne opreme, bo programska oprema prišla najbližje realnosti.

Zgodovina simulacij

Dirkaške video igre so vedno poskušale zagotoviti simulacijsko vožnjo, dokler Namco z založbo Atari ni v letu 1982 izdal igre "Pole position", ki so jo oglaševali kot "neverjeten vozniški realizem". Igra je zagotavljala izkušnjo Formule 1 za volanom. Vsebovala je voznike z umetno inteligenco proti katerim smo lahko dirkali, nesreče ob trkih in kvalifikacijski krog, kjer je moral igralec najprej opraviti poizkusni krog, preden je lahko sodeloval na Veliki nagradi. Igra je bila pionirska tudi pri tretjeosebni kameri, ki je sledila avtomobilu, proga pa se je sproti risala. Igra je simulirala vožnjo naravnost, medtem ko se je položaj proge spreminjal iz leve proti desni, igralec pa ji je moral slediti.

Trenutno razvijanje

Ker je povpraševanja po dirkaških simulacijah vedno več, se v tem trenutku razvija precej novih simulacij, ki so že ugledale luč sveta, kot alfa ali beta verzije ali pa jih razvijalci skrbno skrivajo pred množicami. Trenutne igre v razvoju so iRacing.com, project C.A.R.S. (community assisted racing simulator), Live for Speed, XMotor Racing, rFactor2, Asseto Corsa, GTR3 in še mnoge druge.

Leta 2003 je tričlanska skupina neodvisnih razvijalcev, ki jo sestavljajo Scawen Roberts, Eric Bailey in Victor van Vlaardingen, izdala testno različico dirkalne simulacije Live For Speed. Med igralci je ta različica vzbudila veliko zanimanja, zato so se razvijalci odločili, da projekt še izboljšajo s serijo popravkov. Korak za korakom so igro izboljševali in s tem pridobili veliko igralcev. Razvijalci imajo v planu 4 velike različice igre. Prvo so poimenovali S1, trenutna pa je različica S2, ki je že v fazi zaključka, saj je bila izdana že leta 2005. V letošnjem letu se pričakuje že različica S3, ki nam bo prinesla nove avtomobile, proge in kar je najbolj pomembno prenovljen fizikalni pogon in fizikalni model gum.

Glavni meni igre bi lahko označili kot preprostega, kar na nek način tudi je. Avtorji se niso ukvarjali z animacijami v ozadju ali čim podobnim. Na voljo so enoigralski način, večigralski način, hot lapping, nastavitve in posnetki dirk (enoigralski in večigralski). V enoigralskem načinu se po hitrem postopku znajdete v meniju, kjer lahko nastavite dirko po želji. Pred dirko je potrebno nastaviti število in težavnostno stopnjo računalniških nasprotnikov, možnost kvalifikacij, dolžino dirke, progo in avtomobil. Na začetku so na voljo le trije avtomobili - ostale je potrebno odkleniti z nakupom licence S1 ali S2.

O simulatorjih

Simulator posnema sile, predvsem pa voznika.

Kako ljudje zaznamo gibanje?

Prvi senzorji so oči. Z njimi gledamo iz avta na okolico, kjer se premikamo. Vendar pa oči in računalnik nista najboljši simulator. Niti z najboljšo igro in najboljšo grafično kartico in zaslonom na voljo ne moremo tako dobro prikazati simulacije, kot jo lahko s pravim dirkalnikom, vendar pa se s tem zelo približamo resničnosti. Obstaja pa še en senzor – notranje uho. Ta zazna bilance in gibanje. Torej mora simulator simulirati očesi in notranje uho. To pa dosežemo s pravo sliko na ekranu in zvokom. Nenazadnje pa zaznavamo tudi z rokami in hrbtom. Ti čutijo vibracije iz voznikovih koles in sedeža in so najhitrejši senzorji gibanja. Simulator poskuša simulirati sile, kot so pospeševanje, zmanjševanje hitrosti in obračanje na voznika. To dosežemo s cilindri, na katerih stoji simulator. Obstaja več vrst simulatorjev, ki uporabljajo različne tehnike za doseg tega cilja. Najenostavnejši so tisti, ki uporabljajo samo volan in sedež.

Simulatorji s tremi vrstami nagibanja omogočajo nagibanje levo in desno, naprej in nazaj ter zasuk v levo ali desno stran. Simulator s štirimi vrstami nagibanja omogoča enake sposobnosti nagibanja kot simulator s tremi, vendar še ima dodano dvigovanje in spuščanje simulatorja. Oba simulatorja omogočata pri pospeševanju in zaviranju enake sposobnosti, torej nagibanje sedeža naprej in nazaj. Ko se sedež nagne nazaj, nas teža potisne na sedež. Takrat simulator simulira pospeševanje. Ko pa se sedež nagne naprej, nas teža potegne s sedeža. Takrat simulator simulira zaviranje.

Mnogi simulatorji imajo os nagiba v levo in desno stran pod sedežem, ker je to zelo preprosta metoda in se z njo prihrani veliko prostora. Vendar pa ima ta pomanjkljivost. Če se sedež preveč

nagne v določeno stran, nas začne vleči iz sedeža. Pri takih situacijah se največkrat izniči učinek pospeševanja. Občutek pospeševanja se nato sklicuje na čute voznika. Enaka težava pa se pojavi tudi pri zaviranju.

Nekateri simulatorji s štirimi vrstami nagibanja pa imajo to os nagibanja nad voznikom. Ko se sedež pomakne nazaj, voznika potisne na sedež. To se ujema s silo težnosti in zaznavanjem pospeška. Ker so vsi čuti uravnoteženi, se pospešek in zaviranje čutita tudi takrat, ko se voznik nagne v določeno stran. Višje kot je os nagibanja, boljši občutek je.

Tako lahko uporabimo rotacijske predloge za simulacijo pospeševanja in zaviranja. Včasih res zgleda nenavadno, ko imamo noge višje od glave, ampak za pravičen občutek pospeševanja je to najboljši način.

Da bi bila simulacija čim boljša, se ne bi smelo videti okolice okoli sebe. Pri tem nam veliko pomaga, če imamo simulator v temnem prostoru ali pa si kupimo enobarvno zaveso. Tudi z uporabo čelade lahko veliko pripomoremo k boljšem občutku vožnje, saj nam ta zmanjša vidni kot in ga usmeri v zaslon, poleg tega pa daje vozniku občutek pravega dirkača. Torej se v tem primeru čelada ne uporablja za varnost, ampak za boljše počutje voznika.

Med samim raziskovanjem že izdelanih simulatorjev s štirimi vrstami nagibanja smo izvedeli, da je najboljši kot pri nagibanju v levo in desno stran kot 30° , saj še kljub svoji velikosti daje vozniku občutek varnosti. Tudi kot nagibanja naprej in nazaj je 30° , saj lahko z njim prikažemo tako pospeševanje kot zaviranje. Kot zasuka v levo ali desno stran pa je 90° , medtem kot je razdalja dvigovanja in spuščanja najboljša v višini 500 mm. Ker bi lahko takšen simulator izdelali doma, smo si še pogledali določene specifikacije simulatorja. Ugotovili smo, da:

- potrebuje napetost $230\text{ V} \pm 10\%$ / 50 Hz,
- največji tok 16 A,
- največjo porabo 3,35 kW,
- srednjo porabo, ko je gibanje vključeno $800 \div 1400\text{ W}$,
- porabo, ko je gibanje izključeno 400 W,
- odvodni tok (na tleh) največ 35 mA,
- temperaturo okolja $10 \div +30^\circ\text{ C}$,
- največjo obremenitev (skupaj): 300 kg,
- največjo obremenitev (samo oseba): 120 kg,
- teža 320 kg.

Ugotovili smo še, da je najbolje imeti premer osnovne plošče 1730 mm, širino brez osnovne segmente stranske plošče 1450 mm, višino (v osnovnem položaju) 1800 mm, največjo višino (v gibanju) 2650 mm in minimalni varnostni krog premera 2500 mm.

Simulacija Live for speed

To simulacijo dirkanja smo izbrali zato, ker uporabniku ponuja bogato paleto nastavitvev za dirkalni avto, s katerim se vozi. Uporabnik lahko nastavlja najrazličnejše podrobnosti avtomobila. Nastavitve so prilagojene uporabniku in so predstavljene na razumljiv način, da se lahko uporabnik znajde med vsemi temi nastavitvami. Odločili smo se, da opišemo vse te nastavitve.

Prvo področje pri avtomobilu so informacije o samem avtomobilu. Na levi imamo prikazane podatke o motorju, moči motorja, navoru, skupni masi dirkalnika, specifični moči, razporeditve mase dirkalnika in prenosih. Na desni strani pa imamo prikazane podatke o gorivu. Podatke o gorivu lahko poljubno nastavljamo. Lahko si nastavimo količino goriva na startu in količino goriva na postanku v boksih. Zadnja nastavitvev pri informacijah pa je o zamenjavi gum. Lahko si nastavimo, koliko odstotkov gume lahko obrabimo, medtem ko dirkamo. Če bomo med dirkanjem zelo obrabili gume, nam bo igra na podlagi podatka o zamenjavi gum javila, da moramo zamenjati gume.



Slika 1: Informacije o dirkalniku

Drugo področje so barve. Uporabnik si lahko poljubno izbere barve s pomočjo RGB-modela. Uporabnik si najprej naloži teksturo dirkalnika, ki jo lahko prenese z uradne spletne strani, lahko pa jo izbere iz menija, ki ga ponuja igra. Tekstura so v bistvu vsi napisi, ki so prikazani na dirkalniku. Nato uporabnik poljubno nastavi barve karoserije in koles. Lahko pa si sam izdela svoje barve, brez teksture.



Slika 2: Nastavljanje barv.



Slika 3: Dirkalnik s teksturo in poljubnimi barvami

Naslednje področje so zavore. Tu lahko nastavljamo maksimalno količino navora na kolo na dirkalniku. Druga nastavitvev nam pa pove o zavornem ravnotežju. Iz izkušenj je zavorno ravnotežje najboljše nastaviti na sprednja kolesa, saj se pri zaviranju dirkalnik nagne naprej in tako lahko izkoristimo težo dirkalnika pri zaviranju.



Slika 4: Nastavitve zavor pri dirkalniku

Naslednje področje nastavitvev je podvozje. Tu imamo nastavitve razdeljene na sprednji in zadnji del dirkalnika. Pri obeh lahko nastavljamo višino podvozja, trdoto, amortizacijo in stabilizator podvozja. Večja kot je višina podvozja, višje je dirkalnik od tal. Trdota podvozja se najbolj vidi pri zaviranju, saj ob večjih količinah začne dirkalnik poskakovati. Amortizacija se najbolj pozna tam, kjer z dirkalnikom skočimo ali zapeljemo na oviro. Stabilizator pa nam vse skupaj stabilizira, tako da lahko vozimo kar se da normalno.



Slika 5: Nastavitve podvozja na dirkalniku

Naslednje področje pri nastavitvah je upravljanje. Tukaj lahko nastavljamo največji obrat gum pri polnem obratu volana. Lahko nastavimo tudi paralelno upravljanje koles. To pomeni, da se lahko pri določenem obratu notranje kolo bolj obrne kot zunanje. Lahko pa nastavljamo tudi zaprtost gum. Zraven nastavitvev imamo tudi podatek, za koliko se obrne volan od začetka do konca (v našem primeru je to za 720°).



Slika 6: Nastavitve upravljanja koles

Naslednje področje pri nastavitvah so prenosi. Tukaj lahko nastavljamo vrsto sklopke, blokada in pa prenalaganje. Na desni strani pa lahko nastavljamo prenose med prestavami v avtu. Manjše kot so razlike, boljše teče prestavljanje. Paziti moramo, da niso vrednosti prevelike, saj potem izgubimo na hitrosti avtomobila, vendar pa moramo tudi paziti, da niso premajhne, saj potem dirkalnik ne dobi dovolj obratov za normalno delovanje, zato spet izgubimo na hitrosti.



Slika 7: Nastavitve prenosov in sklopke

Predzadnje področje nastavitvev so nastavitve gum. Tukaj si lahko izberemo proizvajalca gum, lahko nastavljamo pritisk v gumah in nagib gum. Te nastavitve lahko nastavljamo simetrično ali asimetrično. Pri simetričnem nastavljanju nastavljamo pritisk in nagibom gum pri obeh gumah hkrati, pri asimetričnem pa vsako gumo posebej. Te nastavitve nam pridejo prav predvsem pri trdoti gum in pri sami vzdržljivosti gum, saj se nam lahko zgodi, da nam guma med dirkanjem počí.



Slika 8: Nastavitve gum spredaj in zadaj

Zadnje nastavitve pri dirkalnikih so potniki. Tu ni veliko nastavljanja, lahko si pa samo izberemo koliko potnikov bomo vozili s sabo in kje bodo sedeli. Dodatni potniki so najboljši za razporeditev teže avtomobila, predvsem pri drsenju skozi ovinke.



Slika 9: Nastavitve potnikov v dirkalniku

CILJI, METODE RAZISKOVANJA

Cilj, ki smo si ga zastavili, je izdelava kontrolne plošče ter pomožne kontrolne plošče in prenos podatkov na te plošče iz računalniške simulacije dirkanja Live for speed.

X-Simulator

Glavni del raziskovalne naloge je bil prenesti podatke iz igre Live for Speed na našo kontrolno ploščo k8055d. To je precej zapleten proces, ker vsebuje veliko korakov in ker smo imeli nekaj težav s programom.

X-simulator (okrajšava za Cross Simulator software) je brezplačen program, ki ga je izdelal "sirnoname" in je dostopen na www.x-simulator.de. Orodje nam omogoča, da iz več kot 15 različnih iger črpamo podatke o silah, ki nastajajo pri vožnji ali pri letenju. Omogoča nam povezovanje z več kontrolnimi ploščami, ki si jih lahko naredimo sami ali pa jih kupimo že sestavljene. Glede funkcionalnosti ter prilagodljivosti nam omogoča dodajanje neomejenega število osi, ki lahko delujejo v realnem času, dodamo lahko tudi motorje in ostale električne naprave, v kolikor je naš računalnik dovolj zmogljiv. Naredimo lahko svoje profile za igre, ki niso podprte, tako da lahko v teoriji ta program uporabimo s katero koli igro, ki deluje na podlagi knjižnice DirectX ter ima fizikalni sistem dovolj napreden, da iz nje črpamo podatke o silah. Ko smo na računalnik priključili kontrolno ploščo, jo je program pod zavihkom "interface settings" samodejno zaznal in nam prikazal vse njene digitalne ter analogne vhode in izhode. Ta program pride v paketu s štirimi orodji. Prvo orodje je Force Sender, ki iz igre izpisuje podatke o silah, te pa lahko vidimo na zaslonu v posebnem oknu. Drugo orodje, ki je tudi za nas najpomembnejše, je "Force Profiler", v katerem nastavljamo osi. Mi smo nastavili 3 osi (za vsak cilinder eno) in na vsako os nastavili učinek sile trka in silo teže. Na sprednja dva cilindra smo nastavili sile stranskega nagibanja avtomobila, na zadnjo os pa smo nastavili pospeševanja, zaviranja ter sile sunka menjanja prestav. Za vsak cilinder je bilo potrebno nastaviti po dva izhoda na naši plošči. En izhod je določal dviganje, drugi pa spuščanje cilindrov.

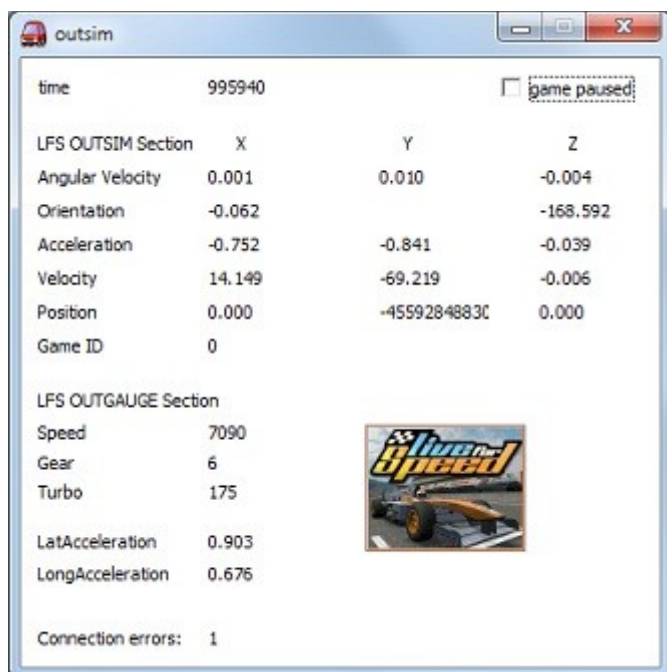
Začeli smo tako, da smo priključili našo kontrolno ploščo in zagnali programsko opremo, ki smo jo dobili ob nakupu plošče. Zagnali smo programsko opremo in preko nje povezali ploščo. Plošča je pravilno delovala, če so ob pritisku na gumb TEST LED-diode začele utripati. Takrat smo program za testiranje zaprli in zagnali prvega izmed programov iz paketa X-Simulator.

Prvi program je bil »Force sender«, (slika 10) ko se nam je program odprl, smo se vpisali z naključnim uporabniškim imenom in geslom. Nato se nam je odprlo okno, kjer smo nastavili fizično pot do zagonske datoteke igre Live for Speed. IP-naslov smo nastavili na 127.0.0.1, kar nam omogoča da podatke pošiljamo na lokalni računalnik. Če bi imeli dva računalnika, pa bi lahko nastavili tako, da podatke prenaša tudi preko mreže. Nato smo si izbrali še vtičnik, preko katerega bomo videli, kakšne sile program izpisuje iz igre. Nato smo preko programa »Force sender« zagnali igro, ki smo jo nastavili v okenski način, da smo lahko spremljali izpisovanje sil. V igri smo nastavili avtomobil, progo in umetno inteligenco ali računalniškega voznika, ki je bil nastavljen, da vozi 100 krogov in tako vseeno pošilja podatke.



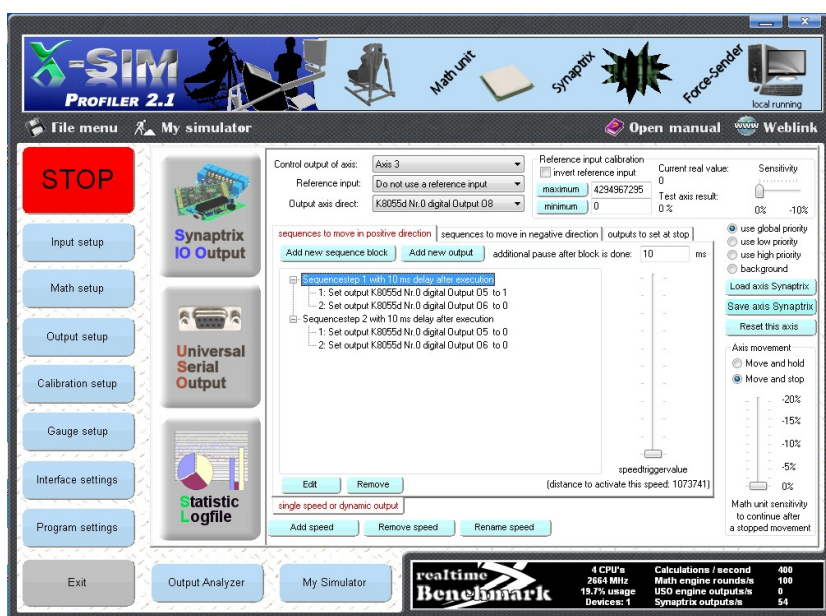
Slika 10: Uporabniški vmesnik programa Force sender

Na vtičniku (slika 11) so se nam izpisovali podatki o hitrosti, prestavi, turbu in različne vrednosti za vsako os x, y in z. Ena izmed vrednosti je bila orientacija po y osi, pozicija avtomobila, pospešek, hitrost, nagib naprej in nazaj.

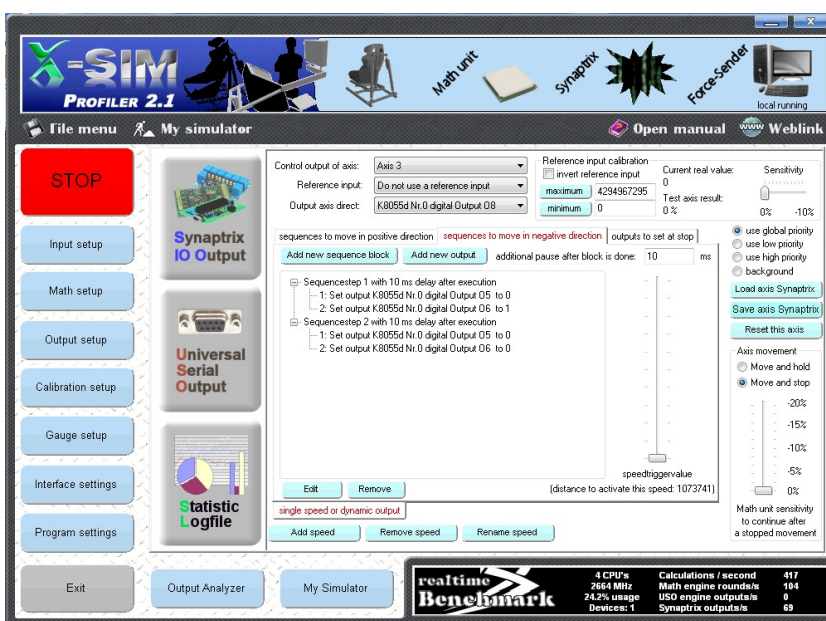


Slika 11: Vtičnik outsim

Naslednji korak je bil zaganjanje programa »Force profiler« (slika 12), ki je glavni program za prenašanje podatkov na našo kontrolno ploščo. V programu smo morali nastaviti sile in sekvence. Na sekvenčni korak 1 smo nastavili npr. izhod 1 in izhod 2 in smo njuni vrednosti nastavili na 1 in 0, v drugem sekvenčnem koraku pa smo obema izhodoma vrednost nastavili na 0. To je, ko želimo, da se cilinder premika v pozitivni smeri. Če želimo cilinder premikati v negativno smer, pa smo morali narediti enako stvar za izhod 1 in 2 v sekvenčnem koraku 1 dodali vrednost 0 in 1 sekvenčnemu koraku 2 dodali vrednost 0. Tako smo za vsako smer cilindra dobili po 2 sekvenčna bloka z dvema izhodoma. V zavihku, ko se cilinder ne premika, pa smo obema izhodoma dodali vrednost 0. To smo ponovili še za ostali dve osi, vendar za različne izhodi (npr. izhod 3 in izhod 4).



Slika 12: Nastavljanje sekvenc za pozitivno smer

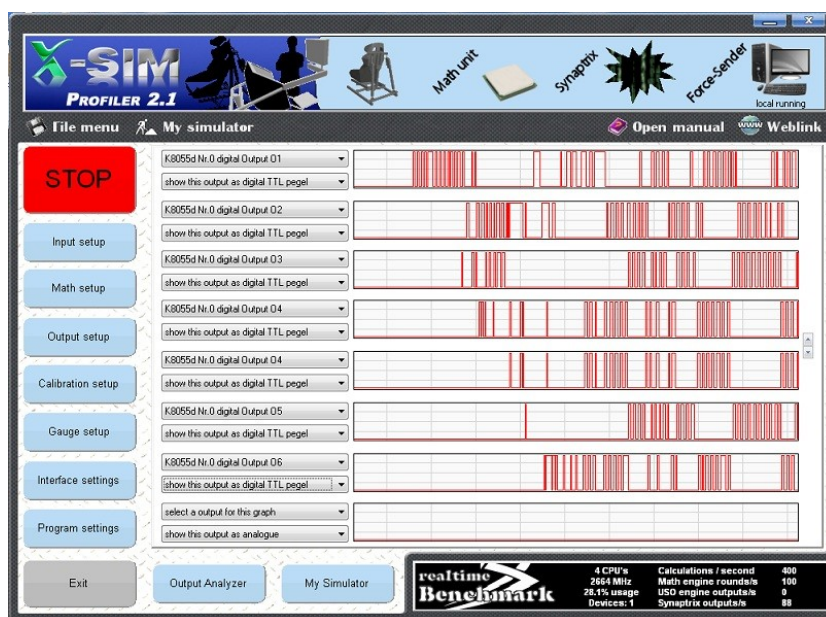


Slika 13: Nastavljanje sekvenc za negativno smer

Ko je bilo v programu vse nastavljeno, smo pritisnili na gumb start, v zgornjem delu programa se je zagnala animacija simulatorja in pošiljanja podatkov. V programu so se nam pod zavihkom input izpisovali podatki iz igre, ki smo jim lahko dodajali različne funkcije, kot na primer wash-out, ki nam je spremembo vrednosti iz 1 v 0 iz digitalne oblike pretvorila v analogno in tako smo iz ostre spremembe dobili gladko analogno spremembo. Takšnih amplitudnih orodij je bilo kar nekaj. Silam smo lahko dodajali množilnike in delilnike, ki so nam sile ojačili ali omilili.

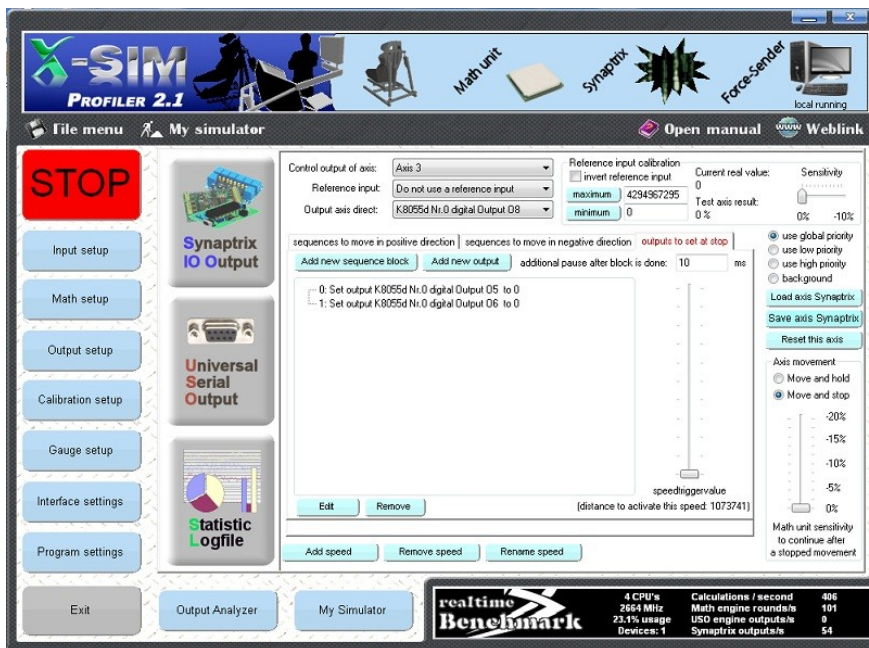


Slika 14: Izpis podatkov o silah iz igre preko programa force sender

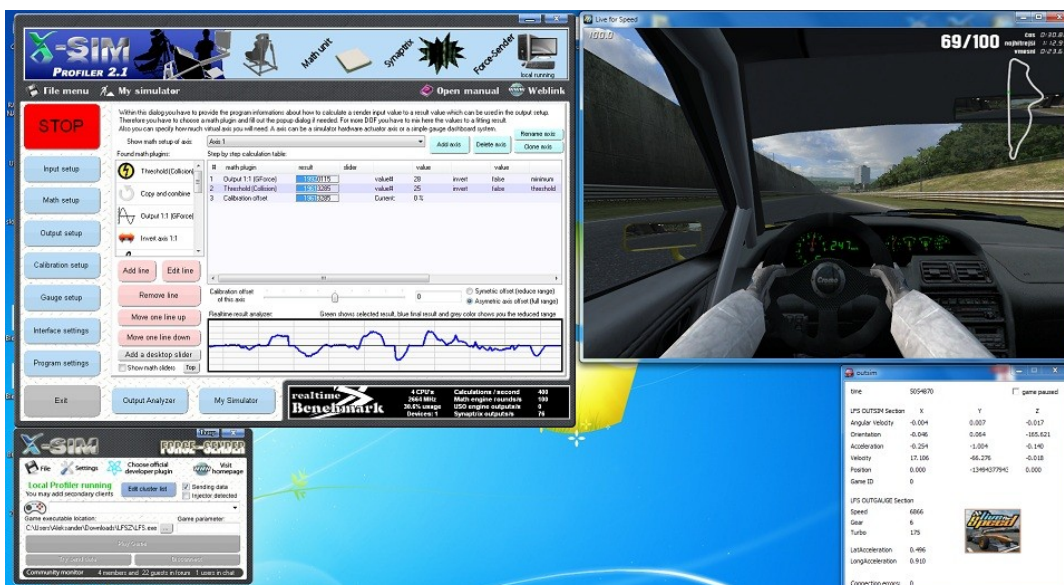


Slika 15: Izpis podatkov, ki jih program pošilja na kontrolno ploščo

Ko je bil program nastavljen, smo na naši kontrolni plošči videli rezultate v obliki utripanja LED-diod, ki smo jim nastavili določene funkcije. Ker ima program veliko funkcij in ker izpiše veliko sil, smo imeli za virtualni tretji cilindar nekaj problemov, saj smo imeli za vsak cilindar 2 LED lučki. Ena lučka je bila za pozitivno smer cilindra, druga lučka pa za negativno smer. Problem je bil, ker sta goreli obe lučki, kar pa v realnosti ne gre, saj ventil preklaplja med stanjem 1 ali 0, obeh stanj hkrati pa ne more imeti.



Slika 16: Nastavljanje sekvenc ob mirovanju



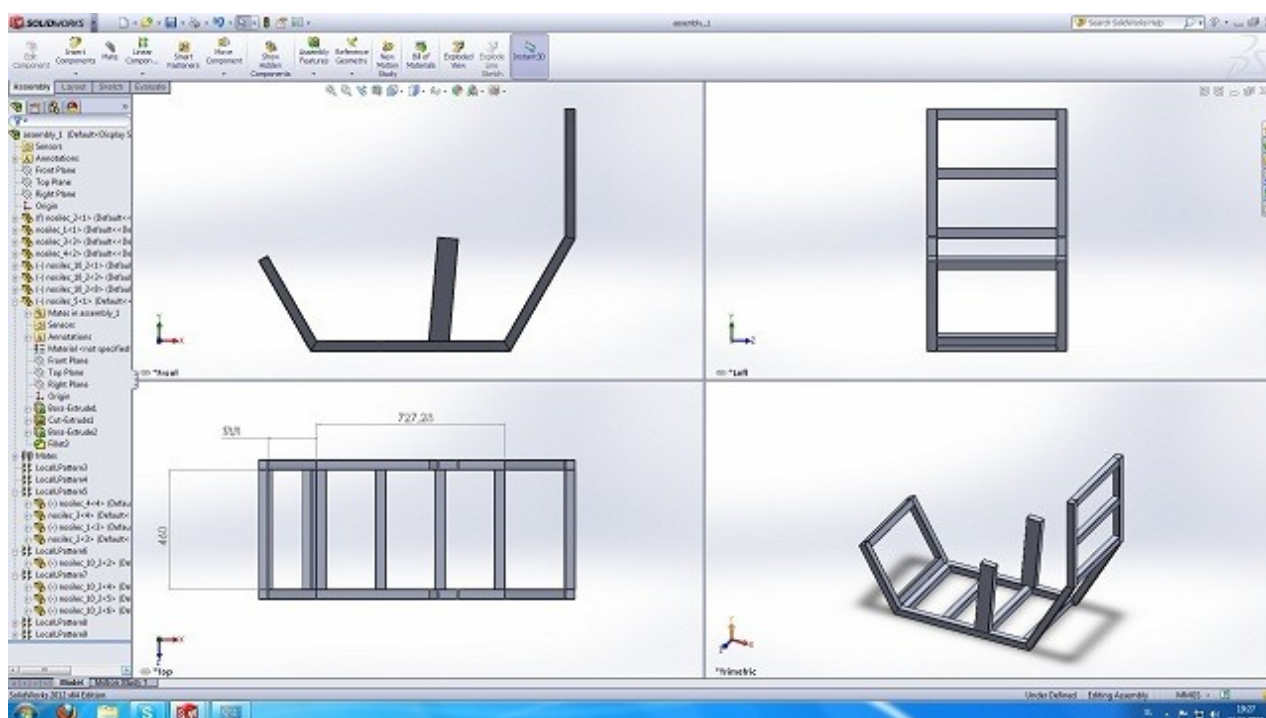
Slika 17: Izgled celotnega zaslona: levo zgoraj program force profiler, desno zgoraj simulacija Live for Speed, levo spodaj program force sender, desno spodaj vtičnik outsim

Ogrodje simulatorja v programu Solidworks

Ogrodje za simulator (slika 18) smo morali pred izdelavo narisati v CAT-orođju. Uporabili smo orodje SolidWorks, ki je eden najboljših programov za načrtovanje konstrukcij. Model smo začeli izdelovati po nekaj urah vaje v programu. Skico ogrodja smo najprej narisali na list papirja, napisali mere in izračunali kote. Nato smo izbrali material in premer cevi, iz katerih bomo izdelovali ogrodje. Iz načrta na papirju smo v programu SolidWorks izdelali vsak del posamezno in jih na koncu sestavili v celoto.

PRIMER:

Za del ogrodja smo morali najprej na ploskev narisati kvadrat. Ko je bil kvadrat na ploskvi, smo ga kotirali in mu določili dolžine stranic (v našem primeru 40 mm). Drugi korak je bil vrisati notranjo steno cevi, ki smo jo naredili na enak način kot zunanjo, ampak smo stranice nastavili za 3 mm manj, ker je to debelina stene železne cevi. Ko smo imeli pogled od zgoraj, smo zapolnili prostor med zunanjo in notranjo steno. Ko je bil prostor zapolnjen, smo z orodjem extract dodali še dolžino cevi. Dolžino smo prav tako določili s kotiranjem in nastavljanjem dolžine (npr.: 500 mm). Ko je bila osnovna oblika končana, smo z orodjem za filet posneli zunanje robove železni cevi, kar nam je dalo izgled železne cevi. Na koncu smo elementu dodali že material železo, kar nam pomaga pri računanju končne teže naše konstrukcije, ki je zelo pomembno zaradi nosilnosti cilindrov.

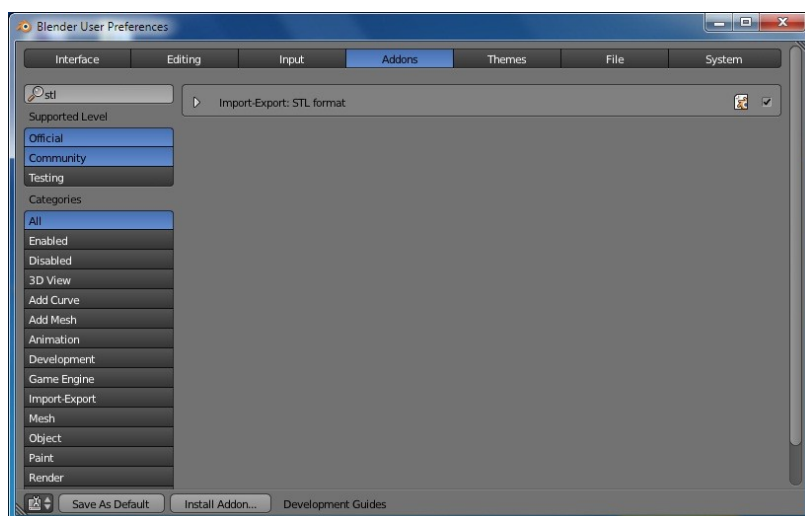


Slika 18: Načrt ogrodja v programu SolidWorks

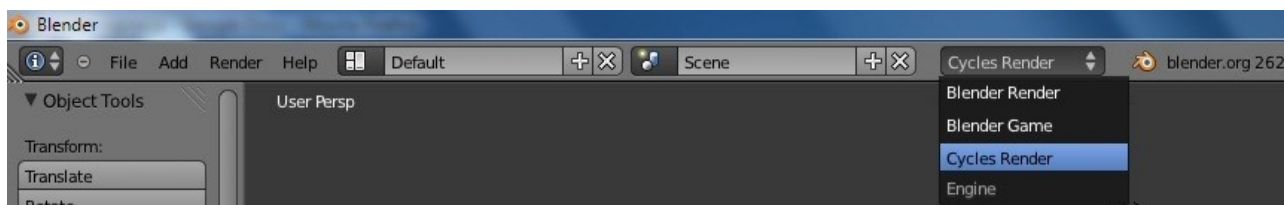
Ogrodje simulatorja v programu Blender

Ko smo končali z izdelavo konstrukcije simulatorja v Solidworksu, smo se odločili, da te modele uvozimo še v program za 3d modeliranje Blender. Za to smo se odločili, ker smo tam lahko model preuredili tako, da nam ne služi kot pomoč pri morebitni izdelavi same konstrukcije, temveč nam služi kot predstavitvena slika, s katero bi lahko izdelani simulator predstavili morebitnim sponzorjem ter ga oglaševali.

Vse dele smo najprej morali v Solidworksu shraniti v .STL-format, ki se ga lahko uvozi v Blender. To smo naredili tako, da smo na "File", nato na "Save As" ter nato pod "Vrsta datoteke" izbrali .STL. Za tem smo morali odpreti Blender ter vključiti dodatek, ki omogoča odpiranje .STL datotek. V orodni vrstici kliknemo na "File", nato na "User Preferences". Odpre se nam okno z nastavitvami. Izberemo zavihek "Addons" ter v iskalno okno vpišemo STL. Najde nam "Import-Export: STL format". Če še ta dodatek ni vključen, ga vključimo s klikom na okence, ki se nam obkljuka (slika 19). Da uvozimo model kliknemo na "File" v orodni vrstici, nato na "Import" ter nato na "Stl (.stl)". Izberemo željeni del, kliknemo nanj, ta se nam bo pa odprl v 3d oknu. Mi smo odprli vse modele na enkrat. Če modeli niso bili pravilno postavljeni ali obrnjeni, smo jih popravili tako, da smo dobili sestavljen simulator. Zmodelirali smo še nekaj ključnih delov, kot so volan, pedalke, sedež ter ekrani. To smo zmodelirali v Blenderju, ker to za nas v Solidworksu ni bilo ključno, če pa želimo prikazati simulator v takšni obliki za predstavljanje izdelka, je pa dobro imeti vse dele. Določene modele, kot so cilindri, smo še popravili, ker se niso najbolje prilagodili Blenderju ter je prišlo do napak ter deformacij. To smo naredili tako, da smo s funkcijami Blenderja popravili normale ter odstranili podvojena oglišča. Tiste dele, ki se jih pa ni dalo popraviti s funkcijami Blenderja smo zbrisali ter jih zmodelirali na novo. Za upodabljanje objektov smo uporabili "Cycles" pogon za upodabljanje (slika 20), ki nam omogoča veliko bolj realistično izrisovanje kot pa izrisovalnik v Blenderju. Vsakemu modelu smo dodali teksturo ter barvo. Naredili smo še preprosto sceno s postavljanjem belih ploskev ter naredili osvetlitev z različnimi barvami. Postavili smo še kamero ter začeli upodabljati slike iz različnih zornih kotov (slike 21, 22 in 23).



Slika 19: Omogočanje uvoza datotek .STL formata



Slika 20: Izbira Cycles pogona za upodabljanje



Slika 21: Upodobljeno ogrodje iz zadnjanskega pogleda



Slika 22: Upodobljeno ogrodje iz sprednjega pogleda



Slika 23: Upodobljeno ogrodje iz stranskega pogleda

Kontrolni plošči

Če smo želeli izvoziti podatke iz računalnika na simulator, smo rabili kontrolno ploščo, ki glede na prejete podatke pošilja električne signale na relejno ploščo, ta pa spušča ter dviga cilindre. Najprej smo poskusili podatke izvoziti na eno kontrolno ploščo, ki smo jo že imeli, a se je izkazalo, da naš program "x-simulator" podpira le nekaj točno določenih plošč. Našli smo seznam vseh podprtih plošč, preverili, katera nam najbolj ustreza tako funkcionalno kot tudi cenovno ter se na koncu odločili za ploščo Velleman k8055d.

Velleman k8055d

Velleman k8055d je USB eksperimentalna vmesniška plošča, primerna za avtomatizacijo doma, izobraževanje ter učenje, testiranje naprav, upravljanje drugih elektronskih naprav ter še za mnogo drugega. Ploščo se lahko upravlja tudi s php-jem preko spletne strani. Vso napajanje dobi preko USB-priključka iz računalnika, kjer dobi tudi vse podatke, tako da ni potrebnega dodatnega napajanja. Če želimo imeti še več vhodov ter izhodov za upravljanje naprav, lahko sočasno uporabljamo 4 takšne plošče na enem računalniku. Vsa komunikacija poteka preko DLL-datotek ("Dynamic Link Library").

K8055 kontrolna plošča ima 5 digitalnih vhodnih kanalov in 8 digitalnih izhodnih kanalov. Poleg tega sta dva analogna vhoda in dva analogna izhoda z 8-bitno ločljivostjo. Število vhodov/izhodov je mogoče še razširiti s povezovanjem več (do največ štirih) kartic za priključke USB računalnika. Vse komunikacijske rutine so v dinamični povezovalni knjižnici (DLL). Lahko pišemo aplikacije za Windows (98SE, 2000, Me, XP) operacijske sisteme v jezikih Delphi, Visual Basic, C++ Builder ali z katerim koli drugim 32bitnem razvojnem orodju, ki podpira klicanje DLL-jev.

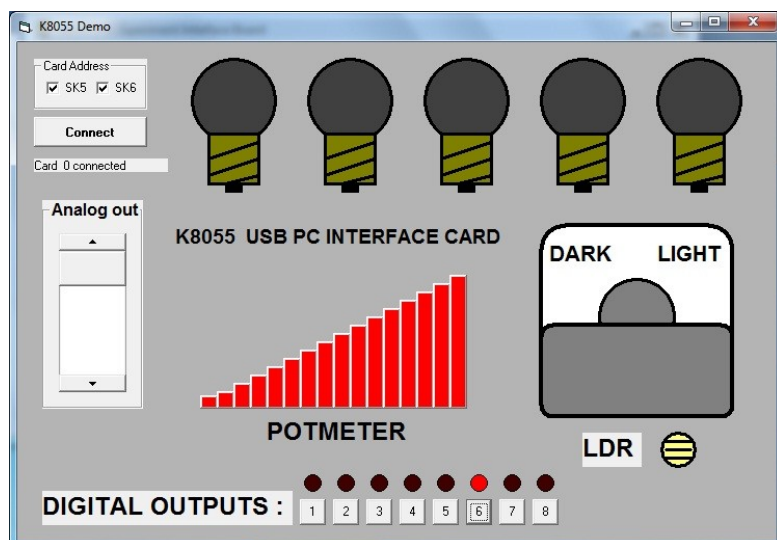
Programska oprema

Če kupimo dele za to ploščo, dobimo zraven priložen CD z gonilniki ter programsko opremo. Minimalne zahteve za to so:

- procesor iz razreda Intel Pentium,
- povezava USB1.0 ali višje,
- Windows 98SE ali višje (razen Win NT),
- CD ROM optični pogon ter miška.

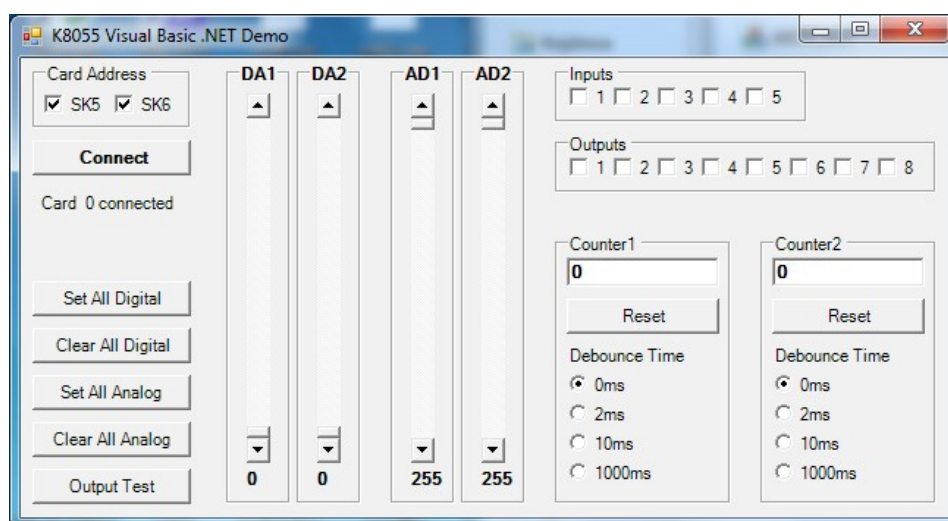
Zraven plošče Velleman k8055d smo dobili CD z navodili za uporabo programske opreme ter navodila za sestavo plošče in nekaj demo programi za uporabo naše plošče. Navodila za sestavo plošče so identična tistim, ki so bila priložena v natisnjeni verziji, le da so bila navodila v elektronski obliki v več jezikih. Shranjena so bila v .pdf formatu. Demo programi so nam bili v veliko pomoč, saj smo z njimi lahko povezali ploščo preko gonilnikov. Dobili smo 3 podobne programe. Programi imajo enake funkcije, ampak različen uporabniški vmesnik. Prvi program ima uporabniški vmesnik, ki je najbolj grafičen in je verjetno namenjen mlajšim uporabnikom. Ko

pritisnemo na eno izmed petih gumbov na plošči, nam v uporabniškem vmeniku zasveti žarnica. Če spreminjamo položaje potenciometroma, se nam povečuje skala pri napisu POTMETER ali pa premika kazalec na merilcu, na katerem piše DARK/LIGHT. Pri napisu DIGITAL OUTPUT se nam ob pritisku na gumb prižge LED-dioda na kontrolni plošči in v programu.



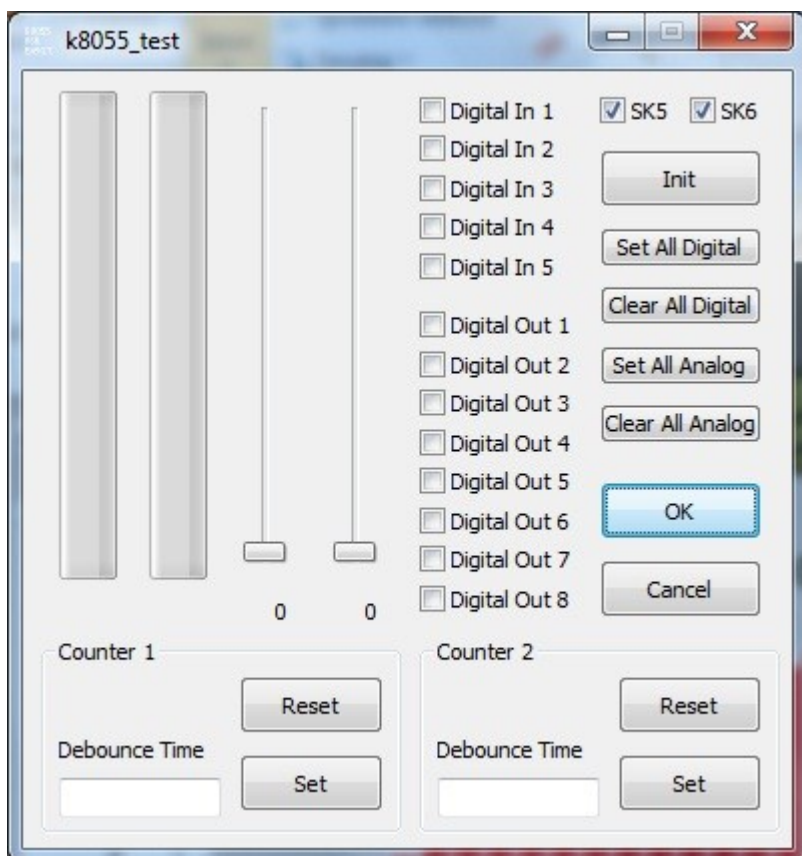
Slika 24: Otroški uporabniški vmesnik

Druga aplikacija je zelo podobna prvi, a ima malo bolj profesionalni in kompakten izgled, saj ne prikazuje sličic, ampak le pomembne informacije, kot so npr.: ob pritisku gumba se v okencu prikaže kljukica. Tudi LED-diode prižigamo le s check boxi in ne z gumbi. Ta aplikacija ima še nekaj dodatnih možnosti kot so prižiganje vseh digitalnih ali vseh analognih izhodov ali pa njihovo ugašanje. Ponuja nam tudi test izhodov in zakasnitev zaznavanja pritiska gumba. To lahko nastavimo tako da izberemo radio gumb z 0 ms zakasnitve, 2 ms zakasnitve, 100 ms zakasnitve ali 1000 ms zakasnitve.



Slika 25: Visual basic uporabni vmesnik

Zadnja aplikacija, ki je ponujena zraven plošče pa ima povsem enake funkcionalnosti kot prejšnja, samo, da je ta v najbolj kompaktnem načinu, a je še vedno uporabna, čeprav ni najbolj pregledna. Edina razlika je, da lahko sami nastavimo čas zakasnitve gumba.



Slika 26: Kompakten uporabniški vmesnik

Razvijalci programske opreme so se potrudili in naredili programsko opremo z uporabniškim vmesnikom tudi za Linux operacijske sistemi in mobilne naprave z Androidom ter iOS operacijskimi sistemi.

Za namestitev programske opreme na Linux operacijski sistem moramo s spletnega naslova linuxk8055.free.fr prenesti odprtokodno programsko opremo za našo kontrolno ploščo. Najprej moramo prenešeno kompresirano datoteko razširiti. Za tem moramo kodo "sestaviti". To pa naredimo tako, da zaženemo terminal, se premaknemo v mapo, kjer smo razširili datoteke ter z ukazi `./configure`, `make` ter `make install` dokončamo proces namestitve. Predpogoj za namestitev je imeti program "libusb 0.1.8" ali novejšo različico ter jedro operacijskega sistema "kernel 2.6" ali novejšega. Za tem lahko z ukazom `man k8055` vidimo navodila za uporabo ukazov, s katerimi lahko beremo ali pa pošiljamo podatke na ploščo. Obstaja tudi uporabniški vmesnik za to programsko opremo. Tega lahko dobimo na spletnem naslovu <http://sites.google.com/site/vellemank8055/k8055guidownload>.

Kompresirano datoteko ponovno razširimo, naložimo pa še naslednje programe, ki so potrebni za delovanje:

- gcc
- libk8055 v0.3 ali novejši
- wxWidgets
- libusb-0.1.9 ali novejši in kernel 2.4.18

Ko smo vse namestili, lahko opazimo nov direktorij "src". Odpremo terminal ter gremo v ta direktorij. Napišemo ukaz make ter namestitev je končana. V tej mapi uporabimo ukaz ./k8055, da zaženemo vmesnik.

Žal nimamo nobene naprave z iOS-operacijskim sistemom, da bi lahko preizkusili delovanje aplikacije za kontrolno ploščo tam, smo jo pa zato lahko preizkusili na napravi z operacijskim sistemom Android.

Aplikacija je namenjena samo tabličnim računalnikom z operacijskim sistemom Android, ker za prikaz uporabniškega vmesnika potrebuje velik zaslon. Slaba lastnost je, da potrebujemo kabel, da se lahko povežemo na ploščo, ampak takšen kabel je zelo težko najti in moramo zato dokupiti pretvornike. Uporabniški vmesnik je zelo podoben uporabniškemu vmesniku na osebnih računalnikih in tudi omogoča enake funkcije. Z njim je mogoče prižigati in ugašati LED-diode ob digitalnih izhodih in analognih izhodih, prepozna pritisnjen gumb na plošči in položaj potenciometra. V aplikacijo lahko tudi naložimo program, ki se bo izvajal na plošči in tako ne potrebujemo vedno računalnika za upravljanje plošče.

Strojna oprema

Plošča vsebuje:

- 5 digitalnih vhodov (0= ozemljeno, 1= odprto) (testni gumbi za na ploščo so priloženi),
- 2 analogna vhoda z možnostjo ojačanja ter oslavljenja signala (notranji test z +5V je priskravljen),
- 8 digitalnih odprtih zbiralnikov izhodnih stikal (maksimalno 50V/100mA) (LED indikator na plošči),
- 2 analogna izhoda:
 - 0 to 5 V, izhodni upor 1K5
 - PWM od 0 do 100 % odprt kolektor odda maksimalno 100 mA/40 V (LED-indikator na plošči),
- povprečen čas procesiranja: 20 ms na ukaz,
- napajanje preko USB-ja: približno 70 mA,
- dimenzije: 145 x 88 x 20 mm / 5.7 x 3 x 0.8".

Ploščo sestavljajo:

- 2 premostitvena kabla
- 2 diodi
- 43 uporov
- 6 keramičnih kondenzatorjev
- 4 IC podnožja (integrated circuit)
- 5 gumbov

- 2 potenciometra
- 1 kondenzator 220 nF
- 9 LED-diod
- 2 tranzistorja
- 1 USB-priključek
- 1 elektrolitski kondenzator
- 1 quartz kristal
- 24 navojnih priključkov
- 4 IC-čipi (integrated circuit)
- 8 "headerjev"
- plošča iz tiskanega vezja

Sestavljanje

Naročili smo dele za to ploščo, ko so prišli, smo jih sestavili ter preizkusili. Sestavljanje je potekalo gladko. Za delo smo rabili orodja ter pripomočke, ki smo jih dobili v šoli ali pa jih prinesli od doma:

- ščipalke,
- spajkalnik,
- spajkalna žica,
- izvijač.

Navodila za izdelavo so bila napisana jasno, zato smo jim brez večjih težav sledili. Najprej smo morali prispajkati premostitveni kabel. Za tem smo morali prispajkati dve diode, paziti smo pa morali na njihovo polariteto v navodilih preveriti upornost ter mesto, kam spada. To je trajalo najdlje, saj jih je 43 ter imajo različne upornosti. Za tem smo prispajkali 7 keramičnih kondenzatorjev, 4 IC-podnožja, 5 gumbov, 2 potenciometra, en 220 nF kondenzator, 11 LED-diod, pri katerih smo morali paziti na polariteto, 2 tranzistorja, električni kondenzator, pri katerem smo morali paziti na polariteto, Quartz kristal, 24 navojnih priključkov ter 4 IC-čipe.

Na koncu smo ploščo preko USB-vmesnika povezali z računalnikom kamor smo namestili tudi programsko opremo za preizkušanje delovanja ter krmiljenje, ki je bila priložena.

Tranzistorska plošča

Ker Velleman k8055d plošča ne oddaja dovolj močnega signala smo morali izdelati še eno pomožno kontrolno ploščo, na kateri bomo signal ojačili iz 12 V na 24 V, kar je dovolj za prestavljanje pozicije ventilov, če bi jih v prihodnosti uporabljali, kar je tudi eden od naših ciljev za prihodnost. Plošča je sestavljena iz 20 navojnih priključkov, 8 LED-diod, 8 kondenzatorjev, 7 tranzistorjev, 8 diod in 9 premostitvenih kablov. Od leve proti desni imamo 10 navojnih priključkov za vhode, nato imamo 8 LED-diod, ki nam sporočajo, ali se podatki prenašajo. Naslednji elementi so 9 premostitvenih kablov, ki premoščajo prostor za mikroprocesor. Naslednji na vrsti je 8 kondenzatorjev in za njimi 7 tranzistorjev. Pred zadnjimi 10 izhodi pa je še 8 diod. Plošča je povsem preprosta in potrebuje le 12 V in 24 V napajanje ter do 8 podatkovnih kablov iz kontrolne plošče do naše plošče in iz naše plošče do ventilov za cilindre.

REZULTATI IN RAZPRAVA

Simulatorji so trenutno še dokaj nepoznano področje, saj je večina projektov simulatorjev narejenih kot domač projekt in le redkim uspe priti na tržišče. V času pisanja naloge je na tržišču le nekaj komercialnih simulatorjev vožnje, ki pa so pogosto narejeni za dirkaške ekipe, da lahko trenirajo tudi ko niso v pravem avtomobilu. Razlog za to je predvsem cena, saj se le te gibajo od 20.000 € pa vse tja do 200.000 € za najboljše simulatorje.

Idejo za raziskovalno nalogo smo dobili dijaki pri praktičnem pouku, ko smo se pogovarjali o računalniških simulacijah. Ko smo profesorju predstavili idejo, nam je bil pripravljen pomagati. Na začetku smo se morali odločiti, kaj si bomo zastavili kot problem. Hoteli smo narediti celoten simulator, a smo kaj kmalu ugotovili, da so deli zanj zelo dragi, sponzorje za njih je pa težko dobiti ter dobava sponzorskih delov traja dolgo časa. Tako smo si zastavili cilj za to leto, da izvozimo podatke iz igre, ter da jih pošljemo na kontrolno ploščo v realnem času. Naš glavni cilj je bil iz simulacije Live for Speed dobiti podatke iz fizikalnega pogona o silah, ki delujejo na avtomobil in voznika v igri. Ugotovili smo, da se kljub temu, da nimamo dostopa do izvorne kode simulacije, še vseeno da legalno dostopati do podatkov fizikalnega pogona, kar pa je bila glavna hipoteza naše raziskovalne naloge. Do podatkov fizikalnega pogona smo dostopali s pomočjo programskega paketa "x-simulator", ki ga je izdelal "sirnoname" in tako ob zbranih podatkih potrdili prvo hipotezo. Kljub temu, da je program "Force Profiler" odlično narejen pa tudi ni šlo brez težav kot so neželeno ustavljanje programa, sesuvanje in tudi izpisovanje napačnih podatkov, kar pa nam je vzelo precej časa in razmišljanja, da smo lahko napake odpravili. Ker smo veliko delali v šoli smo morali imeti programe in igro nameščeno na prenosne računalnike, ki pa niso tako zmogljivi kot namizni osebni računalniki, zato smo tudi v tem pogledu imeli veliko težav z delovanjem.

Tudi za drugo hipotezo nam je v veliko pomoč prišel programski paket "x-simulator" preko katerega smo podatke iz simulacije Live for Speed prenašali na našo kontrolno ploščo Velleman k8055d in smo jo na koncu lahko potrdili. Čeprav nismo imeli veliko izkušenj s področij zadanih problemov, je delo potekalo brez večjih zapletov. To nam je omogočila dobra dokumentacija za kontrolno ploščo ter pomoč profesorjev ter drugih oseb, ki imajo izkušnje na teh področjih. Ker pa smo imeli nekaj idej, pa smo že tudi risali načrte za naš simulator, ki bi lahko ob pomoči sponzorjev ugledal luč sveta, saj imamo že zanj pripravljeno celotno krmilno opremo od računalnika z nameščenim programskim paketom "x-simulator", kontrolno ploščo Velleman k8055d ter tranzistorsko ploščo in nam tako manjka samo še glavni del – fizični simulator.

ZAKLJUČEK

Pri nastajanju raziskovalne naloge smo se veliko naučili o simulacijah, simulatorjih in o kontrolnih ploščah. Kljub temu, da smo z programskim paketom X-Simulator začeli delati že pred prijavo raziskovalne naloge, ga dolgo nismo znali pravilno uporabljati, dokler na spletu nismo našli dobrega vodiča, ki nam je v zadnjem trenutku pomagal pravilno prenesti informacije o silah do kontrolne ploščice. Ta raziskovalna naloga je dober povod za izdelavo celotnega simulatorja s cilindri, dirkalnim sedežem, volanom in velikim ekranom. Projekt bi nam ob pomoči sponzorjev lahko uspel, saj so simulatorji zelo zahtevni za izdelavo in predvsem dragi. To nalogo nameravamo nadaljevati ter simulator dokončati, saj nas to področje zelo veseli. V začetku smo imeli v načrtu narediti vse v letošnjem letu, a smo kaj kmalu videli, da bo premalo časa, saj je material drag, da ga dobimo od sponzorjev, pa traja dlje, kot smo sprva pričakovali.

POVZETEK

Namen raziskovalne naloge je bil raziskati, ali je možno izvoziti digitalne podatke ter jih pretvoriti v elektro signale, katere lahko uporabimo za krmiljenje pnevmatskih ventilov. Raziskovalna naloga zahteva znanje iz tako programskega kot tudi elektro in strojnega področja. Pri dokončanem simulatorju računalnik pošlje podatke o silah premikanja iz računalniške igre na kontrolno ploščo s pomočjo programa "x-simulator", ta pa pošlje podatke na relejni krmilnik za pnevmatske cilindre, ki premikajo ogrodje simulatorja. Ti cilindri nam nato s pomočjo dviganja in spuščanja predstavijo občutek vožnje kot v pravem dirkalnem avtomobilu. V našem primeru to deluje le v teoriji, saj se sestave samega simulatorja nismo lotili. Postavili smo si tudi dve hipotezi. Ena izmed njih govori o tem, da lahko izvozimo podatke o silah iz računalniške igre, druga hipoteza pa govori o tem, da lahko podatke o teh silah prenesemo na krmilno ploščo. Za izdelavo krmilne plošče smo morali dele kupiti ter jih sestaviti. Programsko opremo za izvoz podatkov iz iger smo dobili brezplačno na internetu.

ZAHVALA

Najprej bi se radi zahvalili našim mentorjem Gregorju Hrastniku, Urošu Remenihu ter Benjaminu Krušiču, ki so nam vseskozi pomagali pri raziskovanju. Zahvaljujemo se profesoricama Jolandi Melanšek ter Lidiji Šuster za lektoriranje naloge. Zahvalili bi se tudi staršem, ki so nam med izdelavo raziskovalne naloge vedno stali ob strani. Zahvala gre Šolskemu centru Velenje, ki nam je omogočil, da smo lahko raziskovalno nalogo naredili. Zahvalili pa bi se tudi vam, ki ste si vzeli nekaj časa in prebrali našo raziskovalno nalogo.

VIRI IN LITERATURA

- [1] http://www.x-simulator.de/wiki/X-SIM_2.0_Software_Documentation, 28. 1. 2012
- [2] <http://www.velleman.eu/products/view/?country=be&lang=en&id=351346>, 18. 2. 2012
- [3] http://en.wikipedia.org/wiki/Sim_racing, 18. 2. 2012
- [4] http://www.x-simulator.de/wiki/Go_here_to_learn_more_about_the_step_by_step_synaptrix_control, 24. 2. 2012
- [5] <http://linuxk8055.free.fr/>, 6. 3. 2012
- [6] <http://sites.google.com/site/vellemank8055/k8055guidownload>, 6. 3. 2012
- [7] http://www.motion-sim.cz/index.php?option=com_content&view=article&id=69%3Atechnical-data-&catid=40%3Ageneral-information&Itemid=65&lang=en, 15. 11. 2011
- [8] http://www.motion-sim.cz/index.php?option=com_content&view=article&id=55%3Aspecification&catid=40%3Ageneral-information&Itemid=65&lang=en, 15. 11. 2011
- [9] http://www.motion-sim.cz/index.php?option=com_content&view=article&id=85%3Ahow-it-work&catid=40%3Ageneral-information&Itemid=65&lang=en, 15. 11. 2011

AVTORJI RAZISKOVALNE NALOGE

Kratka predstavitev avtorjev raziskovalne naloge.

Samo Železnik

Sem Samo Železnik, dijak 3. letnika Elektro in računalniške šole, Šolskega centra Velenje. Za raziskovalno nalogo sem se odločil, ker rad izdelujem raziskovalne naloge ter me je zanimalo področje simulacije dirkanja.

Blaž Sitar

Sem Blaž Sitar, dijak 3. letnika Elektro in računalniške šole, Šolskega centra Velenje. Za raziskovalno nalogo sem se odločil, ker rad izdelujem raziskovalne naloge.

Aleksander Turnšek

Sem Aleksander Turnšek, dijak 3. letnika Elektro in računalniške šole, Šolskega centra Velenje. Za raziskovalno nalogo sem se odločil, ker rad izdelujem raziskovalne naloge ter rad igram računalniške simulacije.