

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Aljoša Kopina

**Računalniško krmiljen motoriziran
snemalni voziček**

DIPLOMSKO DELO NA UNIVERZITETNEM ŠTUDIJU

Mentor: prof. dr. Dušan Kodek

Ljubljana, 2012

Rezultati diplomskega dela so intelektualna lastnina Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavlanje ali izkoriščanje rezultatov diplomskega dela je potrebno pisno soglasje Fakultete za računalništvo in informatiko ter mentorja.

Besedilo je oblikovano z urejevalnikom besedil \LaTeX .



Št. naloge: 01818/2012

Datum: 15.03.2012

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **ALJOŠA KOPINA**

Naslov: **RAČUNALNIŠKO KRMILJEN MOTORIZIRAN SNEMALNI VOZIČEK
COMPUTER CONTROLLED MOTORIZED CAMERA DOLLY**

Vrsta naloge: Diplomsko delo univerzitetnega študija

Tematika naloge:

V filmski industriji se za snemanje dinamičnih kadrov že dolgo uporabljajo snemalni vozički. Ker so za domačo in polprofesionalno uporabo taki vozički preokorni in predragi, se na trgu pojavljajo pomanjšane inačice vozičkov na linearnih aluminijastih tračnicah. Razvijte in izdelajte sistem za računalniško krmiljenje motorjev za premikanje vozička ter hkrati tudi povečave in ostrenja kamere. Za upravljanje s sistemom razvijte aplikacijo, ki teče na tabličnem računalniku z operacijski sistemom Android. Uporabite že obstoječo mehansko konstrukcijo, koračne motorje in mikrokrmilnik, ki preko Bluetooth modula omogoča povezavo z tabličnim računalnikom. Pravilnost delovanja preverite in podajte možne načine za izboljšave.

Mentor:


prof. dr. Dušan Kodek



Dekan:


prof. dr. Nikolaj Zimic

IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani Aljoša Kopina,

z vpisno številko **63050148**,

sem avtor diplomskega dela z naslovom:

Računalniško krmiljen motoriziran snemalni voziček

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom prof. dr. Dušana Kodeka;
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela;
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki "Dela FRI".

V Ljubljani, dne 14. septembra 2012

Podpis avtorja:

Zahvala

Zahvaljujem se vsem, ki so na kakršenkoli način pomagali pri nastajanku te diplome. Še posebej Mihi Koklju za izdelavo konstrukcije, Urši Lutman za prevajanje iz Aljoščine v Slovenščino, Jožetu Guni in sestri Tjaši za posojilo tabličnega računalnika ter sestri Lei za vejice. Zahvaljujem se tudi moji mami in vsem prijateljem za njihovo podporo ter zaupanje in mentorju Dušanu Kodeku za pomoč. Denis Moder, hvala za kabel.

Kazalo

Povzetek

Abstract

1	Uvod	1
2	Uporabljena strojna oprema in njeno delovanje	5
2.1	Konstrukcija - mehanski del	5
2.2	Mikrokrmilnik ChipKIT Uno32	7
2.3	Krmilnik koračnega motorja - pololu A4988	8
2.4	Končna stikala	9
2.5	Bluetooth modul	10
2.6	Krmilno vezje	10
3	Osnovni principi delovanja	13
3.1	Osnovni koncepti	13
3.2	Knjižica za delo z motorjem	16
3.3	Izračun časovnih parametrov	17
3.4	Ukazi mikrokrmilniku	19
4	Aplikacija	21
4.1	Phonegap™	24
4.2	Upravljanje z motorji - ročni način	24

KAZALO

4.3	Časovnica - predogled	25
4.4	Nastavitve	25
4.5	Bluetooth funkcije	26
4.6	Podatkovna baza	28
4.7	Področja uporabe	28
5	Sklepne ugotovitve	31

Kazalo slik

2.1	Pregled celotnega sistema	6
2.2	ChipKIT UNO 32	7
2.3	Pololu A4988 krmilnik za koračni motor	8
2.4	Levo: shema opto sklopnika, desno: dejanski element	9
2.5	Bluetooth modul LM-400	10
2.6	Krmilno vezje	11
2.7	Levo: krmilno vezje s kabli, desno: krmilno vezje spodnja stran .	12
2.8	Snemalna glava	12
3.1	Zgoraj: graf pozicije v odvisnosti od časa, spodaj: graf hitrosti v odvisnosti od časa	14
3.2	Dejanska pot motorja v primerjavi z idealno potjo	15
4.1	Izgled aplikacije na tabličnem računalniku	22
4.2	Vpogled v nastavitve	27

KAZALO SLIK

Kazalo tabel

3.1	Parametri ključnih točk uporabljenih za generacijo grafov na sliki 3.1	15
3.2	Opis ukazov, ki jih preko bluetootha pošilja Android aplikacija .	20
4.1	Opis regij, osnovnega pogleda aplikacije	23

Seznam uporabljenih kratic in simbolov

MPIDE - Multi Platform Integrated Development Environment

HTML5 - Hyper Text Markup Language 5

CSS - Cascading Style Sheets

SQL - Structured Query Language

Povzetek

Diplomsko delo obravnava izdelavo krmilnega sistema za nadzor motoriziranega snemalnega vozička, na katerega namestimo kamero oz. fotoaparatus. Tak sistem omogoča fleksibilno in zanesljivo ponovljivo slikanje ali snemanje različnih kadrov. Voziček s kamero se lahko z uporabo koračnih motorjev premika po tračnicah ter vrti okrog različnih osi. Upravljanje s sistemom poteka preko aplikacije na tabličnem računalniku z operacijskim sistemom Android.

V diplomskem delu je opisan princip delovanja krmilnega sistema, uporabljena strojna oprema in načrtovanje krmilnega vezja ter aplikacija za upravljanje in nadzor celotnega sistema.

Uspešnost celotnega projekta ocenjujemo kot zadovoljivo, vendar pa je še vedno veliko prostora za izboljšave. Te so smiselne predvsem pri nadgradnji aplikacije za nadzor z dodatnimi funkcijami, ki bi uporabniku omogočile lažje in hitrejše delo s sistemom. S tem bi izdelek z vidika uporabnosti morda lahko konkuriral na trgu snemalne opreme in pripomočkov.

Ključne besede

Avtomatizacija, koračni motorji, Android, kamera na vozičku

Abstract

This thesis is about developing an automated motorized camera dolly system. These type of systems give the user an option to automate filming processes and create identical camera movement for as many repetitions as needed. Our design of the camera dolly uses stepper motors to move the camera on the slider and rotate it around multiple axis. The user controls the system using an application running on Android based tablet computer.

There are three main chapters in this thesis in which we describe our system from the conceptual point of view, the hardware part and finally the software application section.

In conclusion we evaluate the system as successfully developed, but as always there is room for improvement, mostly on the side of the application. Adding additional functionalities would make users interaction with the system even better and maybe we could make a competitive product for the market of filming tools.

Keywords

Automatization, stepper motors, Android, camera dolly

Poglavje 1

Uvod

Pogled v zgodovino video produkcije in filmske industrije nam razkrije, da so bili filmi na začetku sestavljeni zgolj iz statičnih kadrov, za katere premikanje kamere ni bilo potrebno. Konec 19. stoletja so se pričeli prvi poizkusi snemanja dinamičnih kadrov, tako da so kamero pritrdili na vlak. Kmalu za tem je sledil razvoj raznih nosilcev, vrtečih (rotacijskih) glav, vozičkov in žerjavov ter ostalih pripomočkov za stabilizacijo dinamičnih posnetkov. V filmski industriji se pogosto uporablja t.i. camera dolly - tračnice, na katerih je voziček, na njem pa je pritrjena kamera. Za upravljanje z vozičkom in snemalno opremo potrebujemo od dva do pet ljudi, odvisno od kompleksnosti konstrukcije, uporabljene opreme in tehnične zahtevnosti kadra. Za domačo in polprofesionalno uporabo so taki vozički ter ostali podobni pripomočki predragi in preveč okorni. Postavitev samih tračnic je zamudna, upravljanje z opremo pa zahteva veliko resursov. Za potrebe različnih, tudi zahtevnejših projektov, lahko podoben snemalni učinek dosežemo z uporabo pomanjšane verzije tračnic ter vozička, ki ga premikamo po linearnih aluminijskih vodilih, ki jih bomo za potrebe tega dela imenovali tračnice, da preprečimo zamenjavo z izrazov vodilo (bus). Glavna prednost takega sistema je predvsem nižja cena, odlikujejo pa ga tudi hitra priprava in postavitev ter enostavnejša uporaba na zahtevnejših terenih. Kot slabost morda velja omeniti prostorsko omejenost,

saj so konstrukcije načeloma relativno kratke (do nekaj metrov) ter možnost omejenega premikanja vozička le po daljici tračnic. A tudi z uporabo teh naprav se pri montaži videa pogosto pojavijo težave zaradi malenkostnih razlik v položaju kamere tekom snemanja. To je še posebej moteče v primeru, ko želimo večkrat posneti, glede na položaj kamere identičen, a vsebinsko različen kader, ter posnetke nato združiti. Montažo lahko močno poenostavimo, in s tem tudi izboljšamo končni produkt tako, da zagotovimo identičnost kadrov oz. pozicijo kamere v poljubnem času snemanja. Navedeno je mogoče doseči z nadgradnjo strojnega sistema, ki s pomočjo več motorjev omogoča vrtenje kamere okoli vertikalne in horizontalne osi. Takšna naprava nam tako omogoča popolni nadzor nad potjo kamere in koti snemanja, s tem pa dosežemo tudi učinek absolutne ponovljivosti dinamičnih snemalnih kadrov.

V času idejne zasnove tega diplomskega dela na tržišču še ni bilo produkta, ki bi omogočal premikanje vozička ter kamere po vertikalni in horizontalni osi, hkrati pa tudi upravljanje s povečavo in ostrenjem. Ker je od idejne zasnove do končne izdelave projekta minilo kar nekaj časa, lahko danes na tržišču že zasledimo podobne produkte podjetij Kessler [6] in Ditogear [3], ki pa so namenjeni skoraj izključno profesionalnim uporabnikom in imajo tudi temu primerno tržno ceno.

Na osnovi natančne preučitve potreb snemalnih situacij smo zastavili izdelavo konstrukcije, ki omogoča premikanje nosilca za kamero. Pri načrtovanju, tako mehanskih gradnikov kot tudi same aplikacije, smo se poizkusili približati prednostim, ki jih omogoča profesionalna video produkcijska oprema. Konstrukcija tako vsebuje pet koračnih motorjev, ki omogočajo pomik po tračnicah, premikanje pogleda kamere po vertikalni in horizontalni osi, približevanje in oddaljevanje ter spreminjanje ostrenja. Za upravljanje sistema smo razvili aplikacijo, ki se izvaja na tabličnem računalniku z operacijskim sistemom Android, z manjšimi popravki pa bi jo lahko uporabljali tudi na osebni računalnikih.

V času razvoja smo naleteli na večje število delnih rešitev, ki pa v večini

primerov niso bile dovolj natančne ali pa niso imele vseh zelenih funkcionalnosti. V večini primerov bi tako prilagajanje že obstoječih delnih rešitev trajalo dlje kot razvoj svoje, lastne rešitve. Od obstoječih rešitev smo tako uporabili le knjižnico, preko katere na enostaven način dostopamo do bluetooth sistemskih klicev v operacijskem sistemu Android [4].

Glavni izzivi diplomskega projekta:

- vzpostavitev hkratnega krmiljenja petih, drug od drugega neodvisnih koračnih motorjev;
- implementacija zveznih prehodov med spreminjanjem hitrosti motorja ter pospeški in pojemki;
- vzpostavitev brezžične komunikacije med aplikacijo na tabličnem računalniku in krmilnikom motorjev;
- izdelava delujoče verzije s podprtimi načrtovanimi funkcionalnostmi v čim manj iteracijah ter časovno omejenem roku.

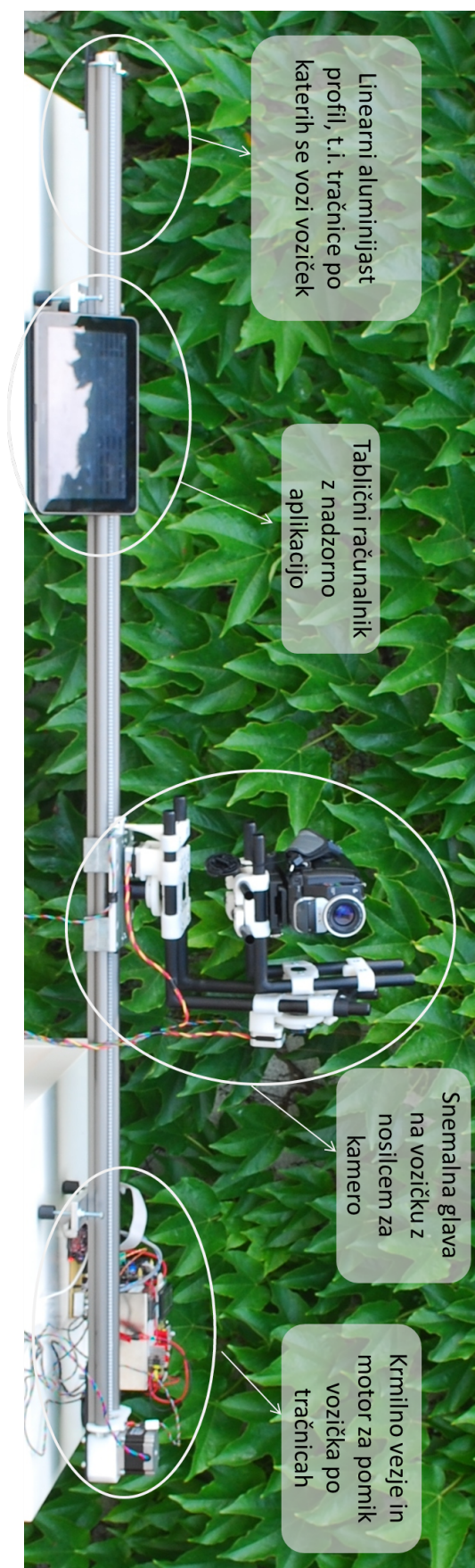
V naslednjih poglavjih nazorno in podrobno predstavljamo izdelavo projekta motorizirane konstrukcije za avtomatizacijo snemalnih procesov z vseh aspektov. V prvem delu je predstavljen strojni del rešitve, v katerem je opisana konstrukcija ter vsa uporabljena mehanska in strojna oprema. Naslednje poglavje zajema koncept implementacije zveznih premikov med različnimi hitrostmi, v zaključnem poglavju pa so pojasnjeni še procesi delovanja aplikacije za nadzor. To poglavje lahko služi tudi kot uporabniški vodič za potencialne uporabnike sistema. V prilogah se nahaja izvorna koda, tako krmilnega kot uporabniškega dela, sistema.

Poglavje 2

Uporabljena strojna oprema in njeno delovanje

2.1 Konstrukcija - mehanski del

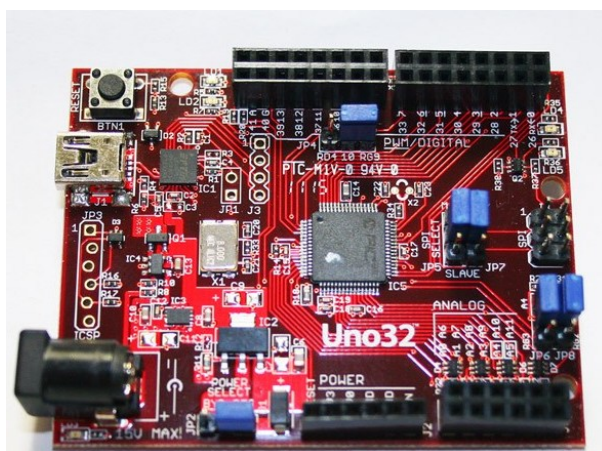
Že iz uvoda je razvidno, da zajema celoten projekt zajema načrtovanje in izdelavo konstrukcije ter razvoj programske opreme za nadzor mehanizacije. Konstrukcijo je načrtoval in izdelal Miha Kokalj v okviru njegove diplomske naloge na Naravoslovnotehniški fakulteti[8], aplikacija za nadzor pa je tema tega diplomskega dela. Konstrukcija je sestavljena iz 1,8 m dolgih tračnic na katerih se nahaja voziček. Tega s pomočjo jermena premikamo z uporabo koračnega motorja. Na voziček je pritrjena motorizirana glava, na kateri se nahajajo ostali štirje koračni motorji, s katerimi premikamo kamero. Kot zanimivost naj na tem mestu omenimo, da so zobniki, ki služijo za prenose na motorjih, nosilci za osi ter nosilci za kamero in pripomočke, natisnjeni iz ABS plastike z uporabo izboljšane verzije tiskalnika RepRap v2. Celoten sistem je prikazan na sliki 2.1



Slika 2.1: Pregled celotnega sistema

2.2 Mikrokrmilnik ChipKIT Uno32

Prvotni namen je bil izdelati kontrolno vezje z uporabo razvojne ploščice Arduino UNO, ki temelji na mikroprocesorju ATmega328, a se je izkazalo, da ta ni dovolj zmogljiv, saj ima premalo digitalnih vhodov/izhodov ter procesorske moči - prenizko frekvenco delovanja in prepočasne operacije z 32-bitnimi števili. Slednje je predstavljalo težavo predvsem pri krmiljenju koračnih motorjev z višjimi frekvencami. Zaradi poznavanja razvojnega okolja in prejšnjih pozitivnih izkušenj z Arduino, smo iskali nekaj podobnega, le bolj zmogljivega in odgovor je bil ChipKIT Uno32, ki je prikazan na sliki 2.2. Glavna prednost s stališča računske zmogljivosti je 32-bitni, 80MHz mikroprocesor PIC32MX320F128, ki ima na voljo 128KB Flash in 16KB SRAM pomnilnika ter 42 vhodov/izhodov.

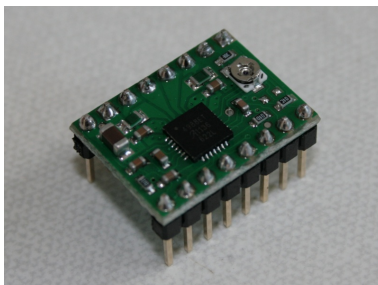


Slika 2.2: ChipKIT UNO 32

Razvojno okolje je nekoliko prilagojena verzija okolja ki ga uporablja Arduino, imenovano MPIDE. V njem lahko uporabljamo programski jezik C/C++, na voljo pa je tudi kar nekaj knjižnic, ki nam poenostavijo enostavne naloge kot so upravljanje s serijsko povezavo, pisanje in branje vhodov ter izhodov, delo z nizi in podobno.

2.3 Krmilnik koračnega motorja - pololu A4988

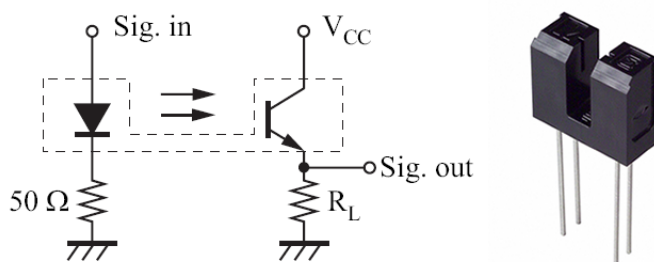
Uporaba namenskega krmilnika za upravljanje z motorji ima več prednosti. Z njegovo uporabo "sprostimo" dva digitalna izhoda na glavnem mikrokrmilniku, saj nam ni potrebno krmiliti vseh štirih signalov za nadzor motorja neposredno, ampak krmilimo le signala step in direction. Ob pozitivni fronti signala step krmilnik ustrezno spremeni izhodne krmilne signale, priključene na motor, ki sprožijo dejanski korak, signal direction pa določi smer koraka. Druga prednost je možnost uporabe "microstepping" načinov, s katerimi povečamo resolucijo korakov ter dosežemo večjo natančnost in odpravimo resonančne motnje [9]. Krmilnik nam poleg polnokoračnega načina omogoča še pol, četrt, osminko in šesnajstinsko koračni način. Krmiljenje motorja z namenskim krmilnikom pripomore k večji modularnosti, robustnosti ter razširljivosti sistema. Izgled uporabljenega krmilnika je prikazan na sliki 2.3. V izdelanem krmilnem vezju smo, predvsem zaradi zagotavljanja večje modularnosti, za vsak motor uporabili svoj krmilnik.



Slika 2.3: Pololu A4988 krmilnik za koračni motor

2.4 Končna stikala

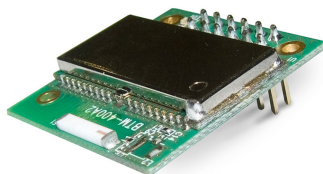
Zaradi zahteve po ponovljivosti gibanja je bilo potrebno zagotoviti referenčno pozicijo. Za to začetno pozicijo smo določili položaj tako, da je voziček na tračnicah pomaknjen do motorja, nosilec za kamero pa poravnan tako, da je kamera obrnjena pravokotno na tračnice ter vertikalno poravnana. Za premikanje vozička ter snemalne glave v začetni položaj smo uporabili optična stikala t.i. opto-sklopnike, sestavljene iz LED diode in fototranzistorja. Princip njihovega delovanja je preprost. Ko med diodo in fototranzistor pride objekt in prekine svetlobni žarek, se fototranzistor "zapre", kar se odraža v spremembi napetosti. V konstrukcijo smo vgradili tri tovrstna stikala, ki jih lahko uporabimo tudi v primeru ko pride do preskoka koraka pri katerem od motorjev zaradi prevelike obremenitve. Mehanskih končnih stikal nismo uporabili, saj imamo na obeh straneh tračnic kovinske zaključke, ki preprečujejo vozičku nadaljnje premikanje, moč na motorjih pa ni dovolj visoka, da bi prišlo do mehanskih poškodb tračnic ali vozička. V primeru, da bi želeli izdelek dodelati do nivoja ustreznega za komercialno tržišče, bi bilo potrebno dodati še mehanska končna stikala na oddaljenem koncu tračnic.



Slika 2.4: Levo: shema opto sklopnika, desno: dejanski element

2.5 Bluetooth modul

Za brezžično komunikacijo med krmilnikom in tabličnim računalnikom smo uporabili bluetooth modul LM-400, ki je prikazan na sliki 2.5. Prvotna izbira je bil Seeduino Bluetooth ščit za Arduino, ki deluje tudi v kombinaciji z ChipKIT UNO razvojno ploščico. Kljub uspešnim testiranjem pri povezovanju krmilnika z osebnim računalnikom, ter ostalimi Bluetooth napravami, nismo uspeli vzpostaviti povezave z napravami z operacijskim sistemom Android. Izkazalo se je, da gre za strojno napako v ščitu, zaradi česar smo se odločili za zamenjavo z LM-400. Modul je bilo pred prvo uporabo potrebno ustrezno nastaviti: spremeniti bitno hitrost prenosa (baud rate), ugasniti strojno kontrolo pretoka (hardware flow control),... To smo dosegli z uporabo AT ukazov, poslanih preko serijskih vrat ChipKITa. Ob normalnem delovanju modra lučka na modulu utripa, ko čaka na povezavo, in sveti, kadar je povezava vzpostavljena.

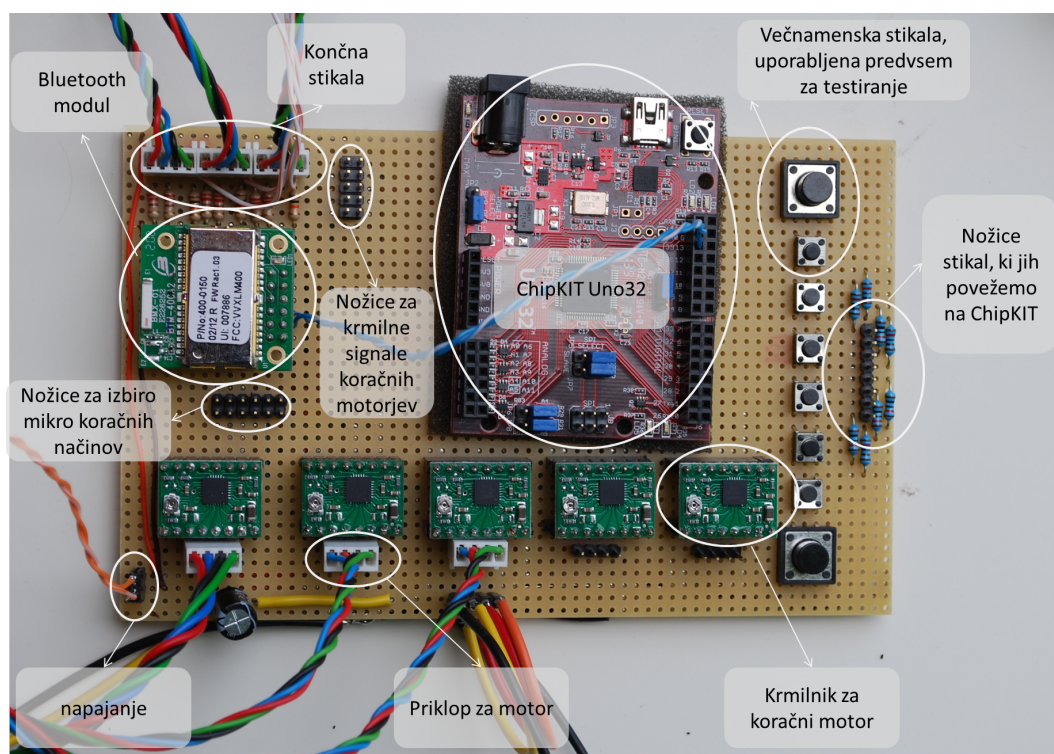


Slika 2.5: Bluetooth modul LM-400

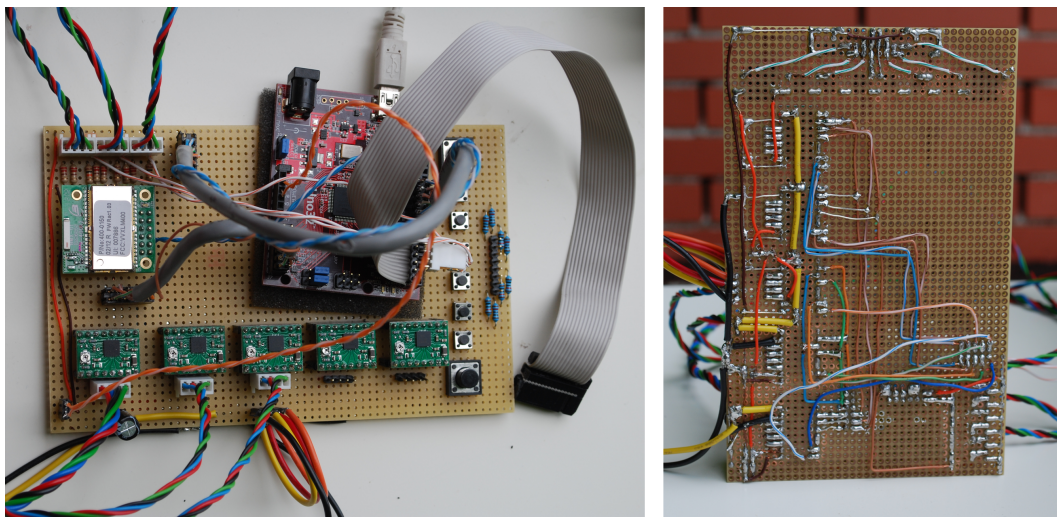
2.6 Krmilno vezje

Zaradi lažje možnosti spreminjanja povezav in prestavljanja posameznih komponent v fazi testiranja smo krmilno vezje izdelali na ploščici za izdelavo prototipov. Končna verzija vezja je prikazana na sliki 2.6, kjer so označeni vsi ključni elementi, opisani v tem poglavju. Iz vezja smo za namene večje preglednosti odstranili kable, ki ChipKIT UNO povezujejo z ostalimi deli vezja.

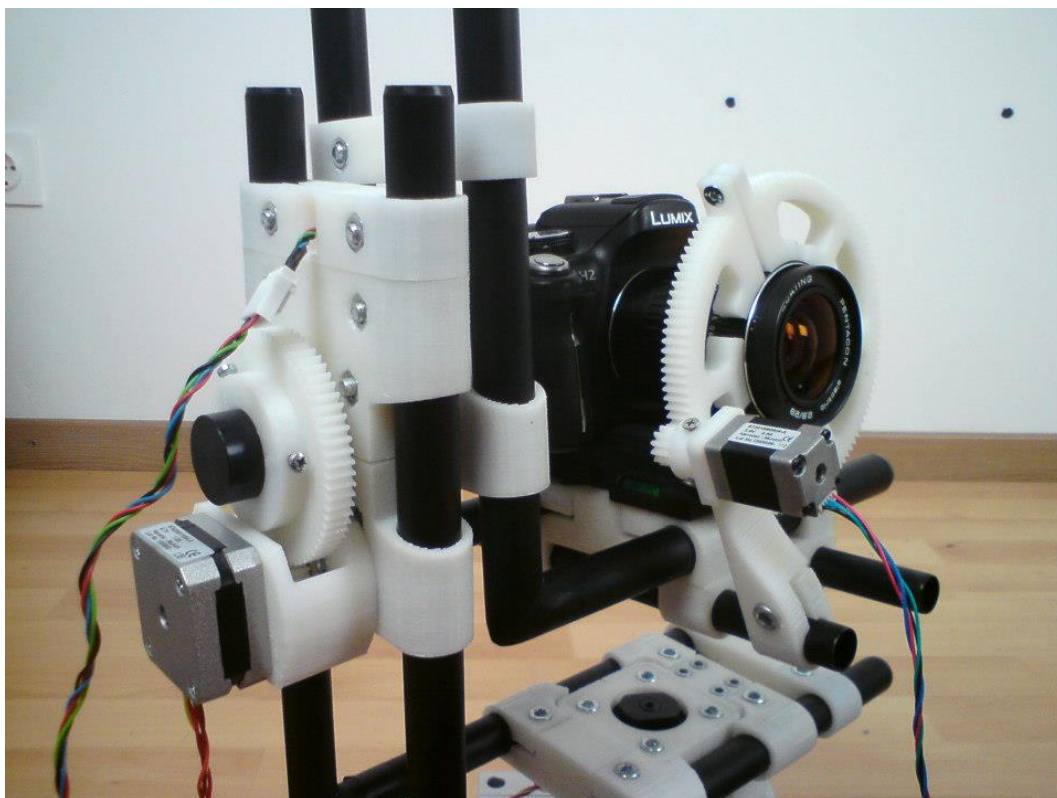
Na sliki 2.8 je prikazana spodnja stran razvojne ploščice ter zgornja stran z vsemi povezavami. Na ploščici imamo na voljo 8 stikal, ki smo jih uporabljali večinoma v času testiranja krmiljenja motorjev, saj takrat še ni bila implementirana možnost brezžične komunikacije in oddaljenega nadzora. Stikala so se izkazala za uporabna tudi v kasnejših fazah razvoja, saj nam omogočajo hitrejšo poganjanje funkcij, ki bi jih sicer prožili z ukazi poslanimi preko brezžične povezave. Z uporabo stikal nam ni potrebno poganjati posebne aplikacije in vzpostavljati brezžične povezave vsakokrat ko naložimo nov krmilni program. V prihodnje jih lahko uporabimo za klice funkcij, ki uporabniku pomagajo v primeru težav, npr. tipka za ponovni zagon modula ali krmilnega vezja, tipka za ustavitev vseh motorjev ali za povrnitev motorjev v prvotni položaj.



Slika 2.6: Krmilno vezje



Slika 2.7: Levo: krmilno vezje s kabli, desno: krmilno vezje spodnja stran



Slika 2.8: Snemalna glava

Poglavje 3

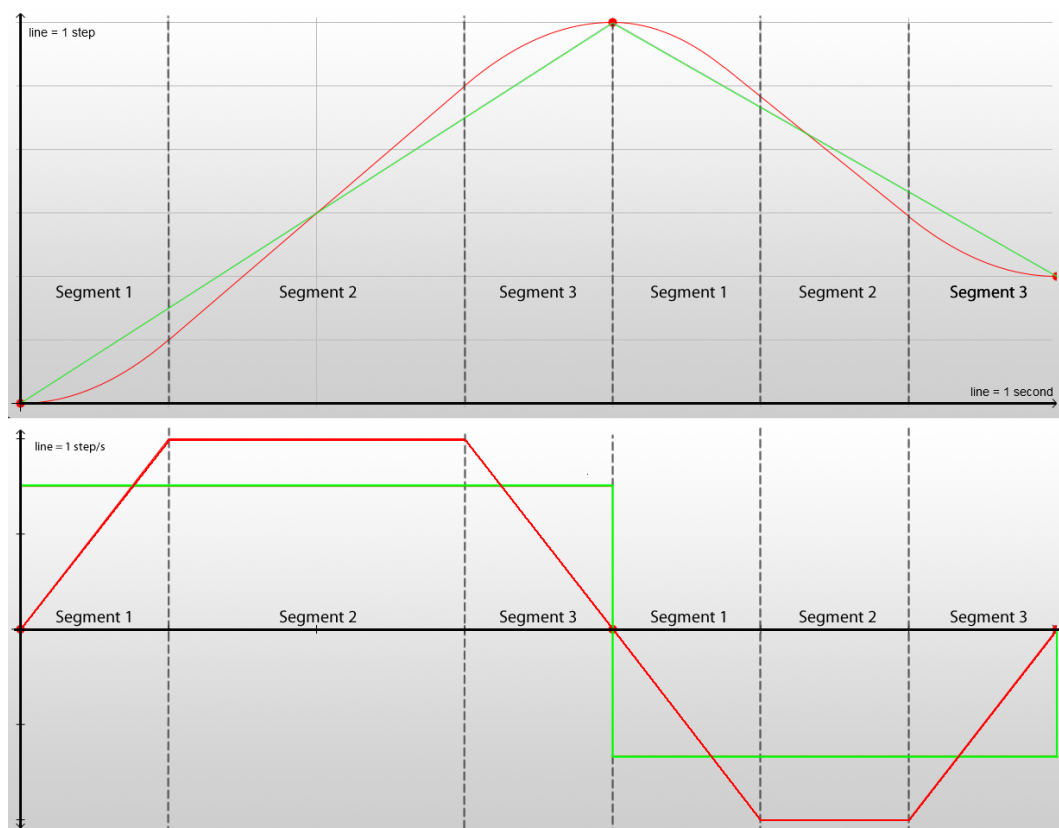
Osnovni principi delovanja

Sistem nam omogoča krmiljenje vsakega od petih motorjev, neodvisno od ostalih. Za lažji opis delovanja v tem poglavju opisujemo nadzor le enega motorja, saj za nadzor vseh motorjev uporabljamo enako strojno opremo in enako knjižnico, ki smo jo napisali za potrebe krmiljenja koračnih motorjev.

3.1 Osnovni koncepti

Gibanje motorja v času opišemo s časovnico v obliki grafa, v katero ima uporabnik možnost vstavljanja ključnih točk (key frame). Ključna točka opisuje v katerem položaju naj se motor nahaja v določenem času, med posameznimi točkami pa vrednosti dobimo z uporabo interpolacije. V začetnih fazah razvoja projekta smo nameravali uporabiti linearno interpolacijo zaradi enostavnega izračuna parametrov ter enostavnega nadzora motorjev. Tako bi vsaka ključna točka vsebovala le dva podatka: čas in pozicijo motorja v izbranem času, sistem pa bi se med podanimi točkami premikal linearno, s konstantno hitrostjo. Zaradi nenadnih sprememb hitrosti, včasih pa tudi smeri gibanja, je bilo premikanje kamere nezvezno, kar se je na posnetkih izražalo v obliki ostrih prehodov, posledično pa posnetki niso delovali profesionalno. Za odpravo tega pojava smo ključnim točkam dodali še dva parametra: faktor gladkosti prehoda v

točko ter iz nje. Ta faktorja predstavljata odstotek časa med dvema ključnima točkama, ko se motor giblje enakomerno pospešeno, s čimer dosežemo zvezne prehode med različnimi hitrostmi. Na ta način smo razdelili gibanje med dvema ključnima točkama na tri segmente: v prvem je gibanje enakomerno pospešeno, v drugem enakomerno s konstantno hitrostjo, v tretjem pa gre ponovno za enakomerno pospešeno gibanje, pri čemer je končna hitrost enaka začetni hitrosti prvega segmenta med naslednjima dvema ključnima točkama. Slika 3.1 nazorno pokaže kako poteka deljenje na segmente ter kako izgledata grafa hitrosti in pozicije v odvisnosti od časa. V segmentih 1 in 3 gre za enakomerno pospešeno gibanje, v segmentu 2 pa za enakomerno gibanje.

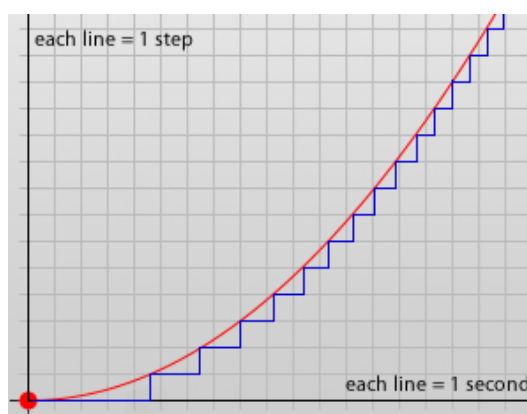


Slika 3.1: Zgoraj: graf pozicije v odvisnosti od časa, spodaj: graf hitrosti v odvisnosti od časa

ključna točka:	čas	pozicija	faktor gladkosti v točko	faktor gladkosti iz točke
1:	0	0	/	0.25
2:	4	6	0.25	$\frac{1}{3}$
3:	7	2	$\frac{1}{3}$	/

Tabela 3.1: Parametri ključnih točk uporabljenih za generacijo grafov na sliki 3.1

Na tem mestu je potrebno pojasniti, da je najmanjša možna enota premika en korak. Velikost koraka je odvisna od izdelave motorja in ga najdemo v dokumentaciji proizvajalca. V našem primeru ima motor, ki pomika voziček po tračnicah, 200 korakov na obrat, kar pomeni da se ob vsakem koraku os motorja zavrti za 0.9 stopinje. Ostali motorji uporabljeni v tem projektu imajo 400 korakov na obrat. Zaradi gibanja po korakih gibanje ni zvezno. Vtis zveznosti lahko ustvarimo z uporabo mikrokoračnih načinov, s katerimi zmanjšamo velikost koraka, uporabo motorjev z dovolj velikim številom korakov na rotacijo ter ustreznim prestavnim razmerjem. Na ta način se s prostim očesom ne opazi nezveznosti oz. premikanja motorjev po korakih. Slika 3.2 prikazuje dejansko pot motorja v primerjavi z idealno.



Slika 3.2: Dejanska pot motorja v primerjavi z idealno potjo

3.2 Knjižica za delo z motorjem

Kot že omenjeno, je najmanjša možna enota premika motorja en korak in krmilni program skrbi, da se signali za vsak korak pošljejo pravočasno. Zaradi lažjega nadzora več motorjev in boljše preglednosti smo napisali posebno knjižnico, ki nam omogoča upravljanje z motorji. Za vsak motor hranimo sledeče podatke:

- podatke o vseh segmentih preko katerih se bo motor pomaknil: tabelo časov med posameznimi segmenti, tabelo pospeškov segmentov, tabelo začetnih hitrosti segmentov;
- podatke o trenutnem stanju: trenutno pozicijo, hitrost, pospešek ter razdaljo do cilja;
- časovne podatke: čas zadnjega koraka, čas naslednjega koraka, čas zamenjave segmenta, čas med koraki, zaporedno številko trenutnega segmenta.

Knjižnico sestavljajo metode in funkcije za spreminjanje omenjenih parametrov, ki jih potrebujemo pri pripravi snemalnega projekta, funkcije in procedure, ki skrbijo za pošiljanje signalov motorju, in vse časovne izračune, ki so potrebni, da se motor giblje z želeno hitrostjo in pospeškom. V grobem bi lahko algoritem razdelili na:

- začetno inicializacijo, kjer se shranijo podatki o segmentih in postavijo začetni parametri;
- primerjanje trenutnega časa s podatkom o času naslednjega segmenta. V kolikor je čas za nov segment se izračunajo novi časovni parametri za naslednji korak;
- primerjanje trenutnega časa s časom naslednjega planiranega koraka. Ob ustreznem času se sproži podprogram, ki poskrbi za pošiljanje krmilnih signalov do krmilnika motorja, ter ustrezno popravi spremenljivke o

trenutni poziciji in času zadnjega koraka, hkrati pa pokliče funkcijo ki izračuna čas novega koraka.

Poskrbeti je bilo potrebno, da se glavna zanka izvaja tako hitro, da je omogočena dovolj visoka frekvenca klicanja funkcije, ki preverja časovne parametre, da ne zamudimo kakšnega koraka. V praksi se je izkazalo, da so bile prisotne omejitve na strani motorjev. Krmilnik bi lahko pošiljal signale nekajkrat hitreje kot so se motorji zaradi fizičnih omejitev zmožni vrteti. Za glavni motor, ki premika voziček po tračnicah, se je preskakovanje korakov začelo kmalu po višanju števila obratov preko 1200 obratov na minuto, kar je bilo za naše potrebe povsem zadovoljivo.

3.3 Izračun časovnih parametrov

Za osnovo pri izračunu časa do naslednjega koraka smo vzeli enačbo za pot pri enakomernem in enakomernem pospešenem gibanju - enačba (3.1). Ko govorimo o pozitivni vrednosti hitrosti pomeni, da se motor vrti v smeri urinega kazalca, ob negativni hitrosti pa v obratni smeri urinega kazalca. Ker se koračni motor pomika vedno po en korak, je naslednja pozicija znana. To je en korak več kot je trenutno stanje, kadar se gibljemo s pozitivno hitrostjo in en korak manj kadar je hitrost negativna, zanima pa nas čez koliko časa naj se zgodi korak. Do odgovora pridemo z reševanjem kvadratne enačbe, ki ima v našem primeru največ eno smiselno rešitev, ki je odvisna od predznaka pospeška in začetne hitrosti. V primeru, da je pospešek enak nič, imamo enakomerno gibanje, čas med posameznimi koraki se ne spreminja in lahko ga izračunamo po trivialni enačbi (3.2). Časi med koraki se spreminjajo, in sicer krajšajo v primeru pospešenega gibanja, ali daljšajo v primeru pojemajočega gibanja. Izračunamo jih lahko z uporabo enačbe (3.3). Izračun poteka tako, da najprej izberemo predznak znotraj enačbe pod korenem. Pozitiven predznak uporabimo v kolikor je pospešek pozitiven, negativen pa v kolikor je vrednost pospeška negativna. Nato izračunamo vrednost enačbe pod korenem in preve-

rimo, če je pozitivna, sicer realna rešitev ne obstaja in motor te točke ne doseže. Tak primer se lahko zgodi le kadar sta predznaka pospeška in začetne hitrosti obratna, saj se v tem primeru motor približuje neki točki, v njej ima hitrost nič, nato pa se začne gibati v drugo smer. Pri izbiri predznakov izven korena izberemo pozitivni predznak kadar je začetna hitrost pozitivna, ter negativni predznak, kadar je začetna hitrost negativna.

$$S_{pos} = v_0 t + \frac{at}{2} \text{ oz. kadar } a = 0 \Rightarrow S_{en} = v_0 t \quad (3.1)$$

$$t_{0,1} = t_{1,2} = t_{n,n+1} = \frac{1}{v_0} \quad (3.2)$$

$$t_{n,n+1} = \frac{(-2v_0 \pm \sqrt{2v_0^2 \pm 8a(n+1)}) - (-2v_0 \pm \sqrt{2v_0^2 \pm 8an})}{2a} \quad (3.3)$$

Spremenljivke uporabljene v enačbah 3.1, 3.2 in 3.3:

- S_{pos} - pot pri pospešenem gibanju, S_{en} - pot pri enakomernem gibanju;
- v_0 - začetno hitrost;
- t - čas;
- a - pospešek;
- $n+1$ - pozicija motorja po naslednjem koraku, n - pozicija motorja v trenutnem stanju.

3.4 Ukazi mikrokrmilniku

Uporabnik z uporabo aplikacije pošilja ukaze mikrokrmilniku preko brezžične Bluetooth povezave. Tabela 3.2 prikazuje najpogosteje uporabljene ukaze ter podaja kratko razlago izvedene operacije ob prejetem ukazu. Stolpci P1, P2, P3 in P4 predstavljajo parametre, ki so del poslanega ukaza, med seboj pa jih ločimo z vejico. Tako bi se ukaz za premik motorja za povečavo za pol obrata (200 korakov), s hitrostjo 100 korakov/s glasil: "p,4,100,200".

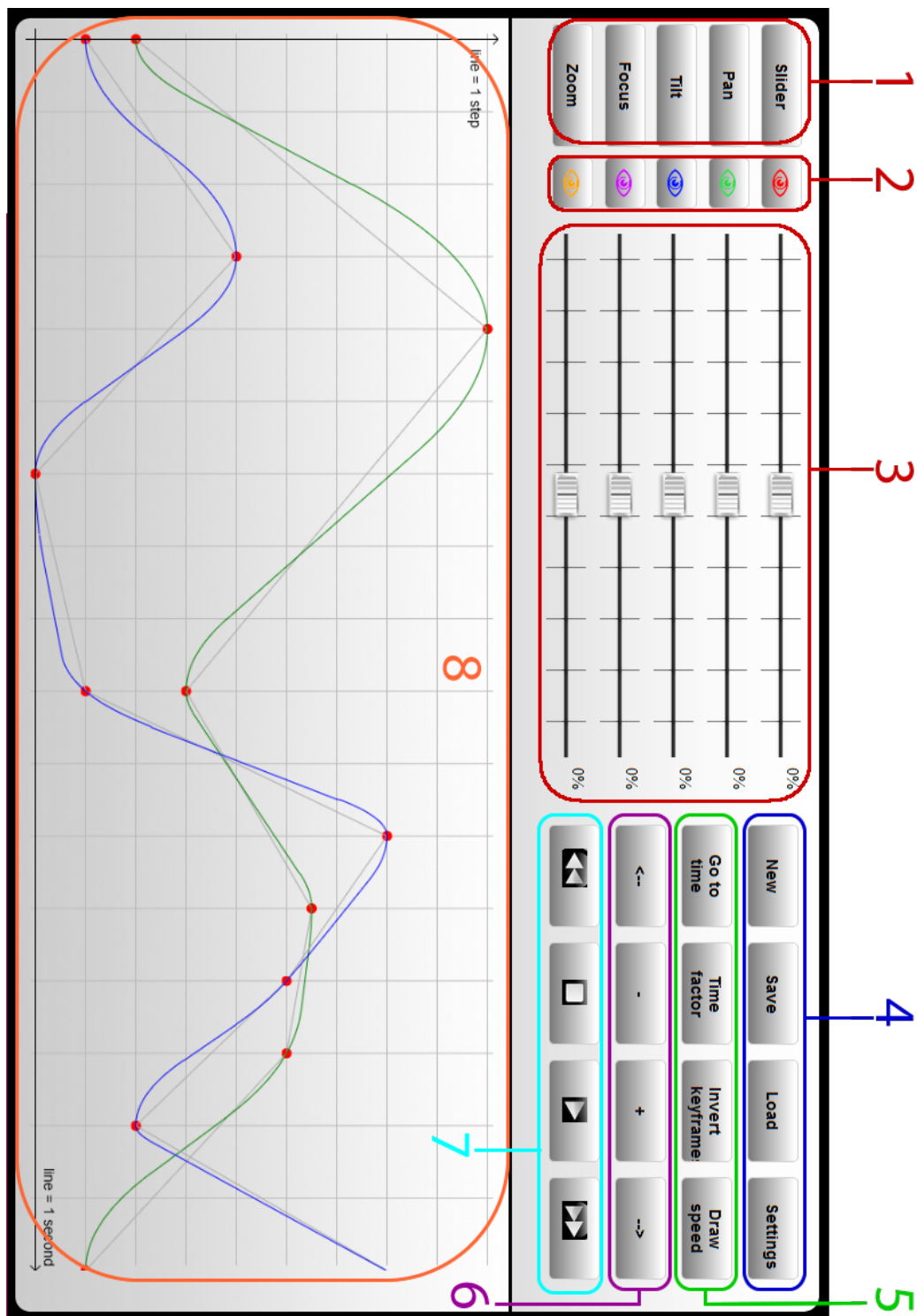
ukaz	P1	P2	P3	P4	opis
a	/	/	/	/	mikrokrmilnik pošlje podatke o položaju vseh motorjev
s	mID	/	/	/	mikrokrmilnik pošlje podatke o položaju motorja mID
r	mID	hitrost	/	/	začne premikati motor mID s hitrostjo: hitrost
x	mID	/	/	/	ustavi motor mID
p	mID	hitrost	razdalja	/	začne premikati motor mID s hitrostjo hitrost in ga ustavi, ko se premakne za razdalja
m	mID	način	/	/	nastavi mikrokoračni način motorju motorID. 0 predstavlja polnokoračni način, 4 predstavlja šesnajstinsko koračni način, vmesne stopnje pa 1,2 in 3.
d	mID	št seg.			mikrokrmilnik inicializira prostor za št. segmentov pri motorju mID po tem pričakuje podatke za vsak segment za omenjen motor. Ti se pošljejo z ukazem 'k'
k	motorID	čas	pospešek	hitrost	podatki o segmentu za motor, ki se shranijo v tabelo segmentov in se uporabijo ob prejetju ukaza g.
g	/	/	/	/	začne s snemanjem

Tabela 3.2: Opis ukazov, ki jih preko bluetootha pošilja Android aplikacija

Poglavje 4

Aplikacija

Aplikacija za upravljanje z avtomatiziranim snemalnim vozičkom se je tekom tega diplomskega dela neprestano nadgrajevala z idejami za izboljšavo, ki so se porajale ob praktičnem preizkušanju naprave. Najbolj očitne so spremembe grafičnega vmesnika in dodajanje novih funkcionalnosti, ki jih v prvotnem načrtu ni bilo. Osnovni pogled uporabniškega vmesnika aplikacije je prikazan na sliki 4.1. Uporabniški vmesnik ima veliko gumbov, združenih v logične enote, kar omogoča bolj učinkovito upravljanje celotnega sistema. V tabeli 4.1 so zbrani kratki opisi posameznih delov glavnega pogleda.



Slika 4.1: Izgled aplikacije na tabličnem računalniku

skupina	opis
1	odpre okno za dodajanje ključne točke v časovnico izbranega motorja
2	vklopi ali izklopi prikaz časovnice motorja
3	drsniki služijo "ročnemu" premikanju motorjev
4	shranjevanje, nalaganje in brisanje shranjenih časovnic ter dostop do nastavitev sistema
5	napredne funkcije sistema
6	manipulacija s časovnico za izbiro bolj ali manj podrobnega pregleda ter premikanje po časovnici
7	gumbi za začetek in konec snemanja ter premikanja med ključnimi točkami
8	predogled časovnice za izbrane motorje

Tabela 4.1: Opis regij, osnovnega pogleda aplikacije

4.1 PhoneyapTM

PhoneyapTM je odprtokodna platforma za razvoj mobilnih aplikacij z uporabo tehnologij in programskih jezikov HTML5, JavaScript ter CSS. Razvoj aplikacije za mobilno napravo tako poteka podobno kot bi izdelovali spletno aplikacijo. To nam omogoča, da večji del aplikacije razvijemo naenkrat za naprave, ki poganjajo operacijski sistem Android, Windows phone, Symbian, iOS in BlackBerry [11], nato pa prilagodimo določene dele aplikacije, ki zahtevajo dostop do sistemskih klicev preko knjižnic in vtičnikov. V primeru aplikacije v okviru tega diplomskega dela je bilo potrebno dodati podporo za Bluetooth, kar smo dosegli z uporabo knjižnice "bluetoothPlugin" za Android, ki že vsebuje osnovne funkcije za povezovanje z drugimi napravami in pošiljanje podatkov.

4.2 Upravljanje z motorji - ročni način

Najenostavnejši način uporabe predstavlja uporaba aplikacije za daljinski nadzor posameznih motorjev, kjer iz aplikacije brezžično pošiljamo ukaze za pomik posameznega motorja na mikrokrmilnik. Na voljo imamo dva načina premikanja posameznega motorja:

- premik do točke - v tem načinu preko vmesnika pošljemo krmilniku informacijo za koliko korakov naj se motor premakne, mikrokrmilnik pa poskrbi, da se motor pomakne do te točke s hitrostjo, ki je enaka polovici maksimalne. Maksimalno hitrost lahko spreminjamo v nastavitvah.
- premik s konstantno hitrostjo - v tem načinu preko drsnika izberemo hitrost s katero naj se motor premika, dokler ga ne ustavimo ali mu pošljemo ukaza z novo informacijo o hitrosti. Za bolj natančen nadzor motorjev so drsniki v eksponentni skali, računamo pa odstotek maksimalne hitrosti, ki jo lahko za vsak motor posebej nastavimo v nastavitvah. V praksi se je ta način izkazal za bolj uporabnega, zato so v

trenutni verziji grafičnega vmesnika uporabniku na voljo veliki drsniki, s katerimi lahko spreminja hitrost motorjev s klikanjem in tako najde položaj kamere, ki ga želi dodati v časovnico kot eno izmed ključnih točk.

Podatke o trenutni lokaciji dobimo na način, da pošljemo ukaz za izpis stanja. Na ta ukaz bo mikrokrmilnik odgovoril s podatki o oddaljenosti od izhodišča v obliki števila korakov. Te podatke lahko uporabimo za dodajanje ključnih točk v časovnico. Izbiramo lahko med dodajanjem podatkov posameznim motorjem ali avtomatskim dodajanjem ključne točke na vseh motorjih.

4.3 Časovnica - predogled

Velik del uporabniškega vmesnika predstavlja okno za predogled časovnice. Prikaz je v obliki grafa na katerem je neodvisna spremenljivka čas, odvisna pa položaj ali hitrost motorjev. Med grafoma preklapljammo z gumbom "Draw speed", ki se spremeni v "Draw position" ko smo v načinu za predogled hitrosti. Glede na svoje preference in potrebe lahko izberemo enega ali več motorjev za katerega si želimo ogledati potek premika v času snemanja - gumbi v skupini 2. Hkrati imamo tudi možnost dodajanja novih ključnih točk - gumbi v skupini 1 - ter spreminjanja parametrov obstoječih ali brisanja ključnih točk, ki jih ne želimo imeti več. Do pogleda za spremembo parametrov ali izbris ključne točke pridemo s klikom na rdečo oznako v točki na grafu. Za bolj natančen pregled v primeru manjših premikov smo implementirali tudi funkcije za povečavo ter premikanje po časovnici - skupina gumbov 6.

4.4 Nastavitve

Za večjo fleksibilnost pri delu z aplikacijo imamo možnost izbire različnih nastavitev. Nastavljamo lahko maksimalno hitrost posameznega motorja, na

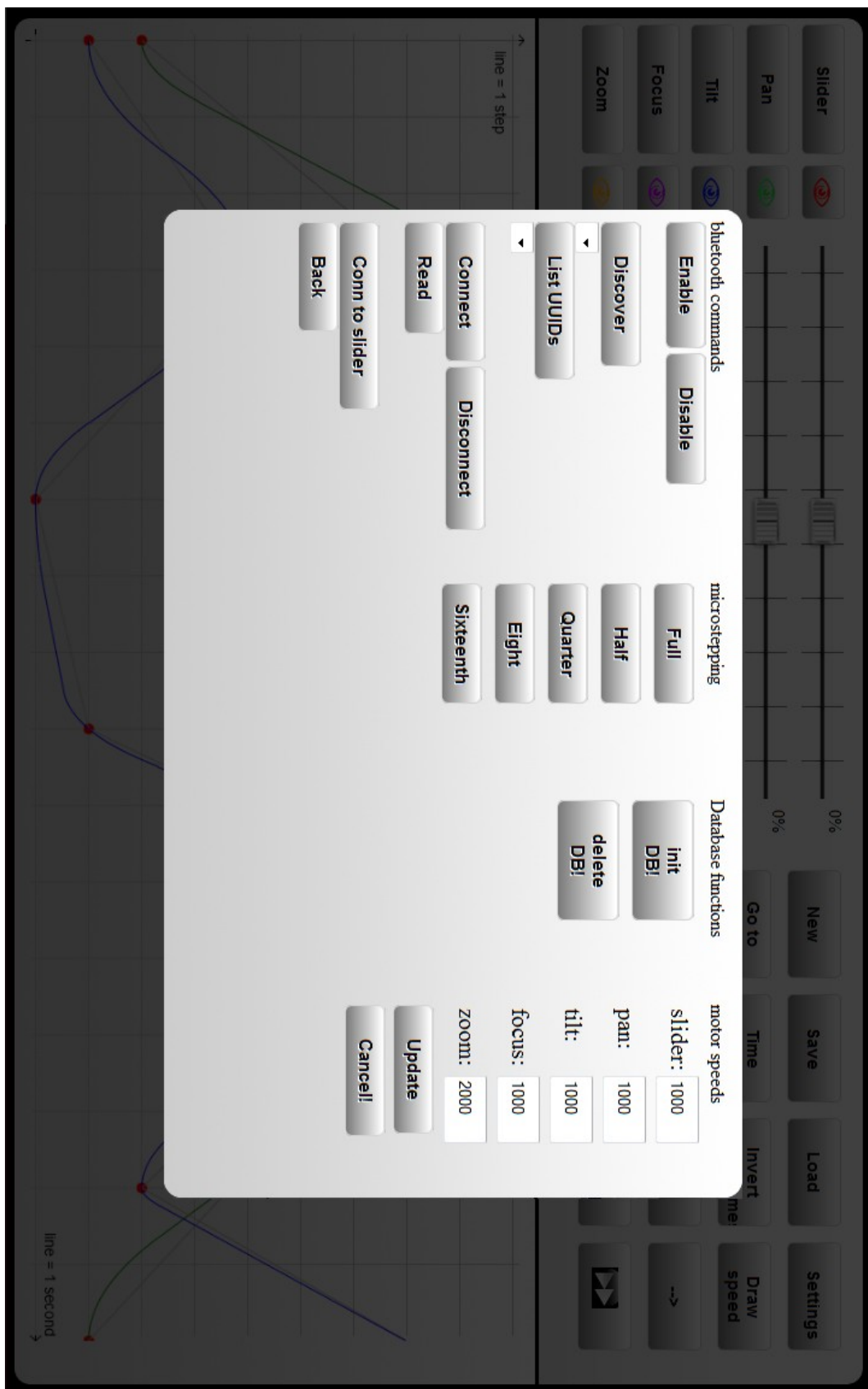
podlagi katere se izračunava trenutna hitrost premika pri ročnem načinu uporabe tako, da je le ta prilagojena obremenitvi in postavitvi tračnic. Izbiramo lahko med polnokoračnim ter štirimi mikrokoračnimi načini. Hkrati lahko nastavimo do tri različne koračne načine - enega za pomikanje po tračnicah, drugega za pomikanje po vertikalni in horizontalni osi ter tretjega za povečavo in ostrenje skupaj. V tem pogledu imamo možnost spreminjanja upravljanja s povezavo Bluetooth ter izvajamo administrativne operacije nad podatkovno bazo. Pogled za upravljanje z nastavitvami, kot vsi ostali pogledi se uporabniku prikažejo v t.i. overlay načinu, ki je prikazan na sliki 4.2, kjer se osnovni pogled zatemni, nato pa prikaže trenutni pogled navidezno na osnovnem pogledu. Ob shranjevanju ali preklicu narejenih sprememb pa se ponovno vrnemo v osnovni pogled.

4.5 Bluetooth funkcije

Za namene brezžične komunikacije med aplikacijo, ki teče na tabličnem računalniku in krmilnikom ChipKIT, smo uporabili že omenjen vtičnik `bluetoothPlugin` in z njim implementirali sledeče funkcionalnosti:

- vklop/izklop Bluetooth povezave na sami napravi;
- iskanje vidnih naprav v bližnji okolici;
- seznanjanje (*angl. pairing*) novih naprav;
- odpiranje podatkovne povezave s seznanjeno napravo;
- pošiljanje podatkov;
- sprejemanje podatkov;
- rušenje odprte podatkovne povezave.

Te metode zadostujejo za vzpostavitev komunikacije in pošiljanje navodil mikrokontrolniku ter prejemanje povratnih informacij.



Slika 4.2: Vpogled v nastavitve

4.6 Podatkovna baza

Možnost shranjevanja trenutnega dela in ponovna uporaba shranjenih časovnic ter ostalih nastavitev za nadaljnje delo, je funkcionalnost aplikacije, ki uporabniku prihrani ogromno časa. Omogoča nam nalaganje vnaprej pripravljenih časovnic, kar uporabniku omogoči obnovitev predhodno shranjenega scenarija v le nekaj klikih. Podatki se shranjujejo v podatkovno bazo SQLite, ki ne potrebuje strežnika, saj shranjevanje poteka lokalno v datoteko na Android napravi. Ob prvi uporabi aplikacije se podatkovna baza inicializira in shrani testni scenarij. Kasneje pa ima uporabnik na voljo shranjevanje aktivne časovnice ter brisanje ali nalaganje katere izmed shranjenih časovnic. Da bi preprečili potencialno izgubo podatkov, se ob ustvarjanju nove, uporabniku ponudi možnost shranjevanja trenutno aktivne časovnice.

4.7 Področja uporabe

Eden izmed ciljev ob izdelavi diplomskega dela je bil dodelati celoten sistem do te mere, da bo zaživel v praksi ter bo uporaben pripomoček pri snemanju. Možnosti uporabe sistema so omejene predvsem s fizično dolžino tračnic ter domišljijo in kreativnostjo uporabnika. Predvideni načini uporabe pa vključujejo:

- snemanje s časovnimi presledki (time lapse);
- snemanje kadrov za videospote in druge filmske kadre kjer želimo skrajšati čas montaže posnetkov z identičnimi premiki kamere;
- snemanje določene serije produktov z uporabo identičnih premikov kamere za poenoten prikaz v obliki video kataloga;
- ponavljanje identičnih gibov kamere, s katerimi lahko dosežemo določene posebne efekte pri video montaži, kot npr. kloniranje osebe v kadru;

- pomoč pri vstavljanju 3D računalniške grafike v posneto sceno brez 3D sledenja (tracking).

Sistem je uporaben tudi za terensko delo, saj teža celotne konstrukcije ne presega 10 kilogramov, večurno avtonomijo pa lahko dosežemo z uporabo baterijskega napajanja.

Poglavje 5

Sklepne ugotovitve

Pot od prve ideje do trenutne verzije sistema za motoriziran nadzor snemalnega vozička je bila polna izboljšav, problemov in njihovih rešitev. V času razvoja ideje podobnih produktov na tržišču še ni bilo, zato je faza razvoja zahtevala precej ustvarjalen in inovatorski pristop vseh sodelujočih, ki pa je terjal veliko časa in tudi učenja na princip poskusov in napak. Ker je razvoj programske in strojne opreme potekal v časovni prepletenosti, smo se velikokrat soočili s težavami zaradi predpostavljajanja, da nova oprema deluje brezhibno in da lahko zaupamo dokumentaciji proizvajalca. Tako smo imeli opravka z Bluetooth modulom, ki je bil nezdržljiv z operacijskim sistemom Android, napačno dokumentacijo pri naslednjem uporabljenem modulu, strojno okvaro na zaslonu pri prvem tabličnem računalniku na katerem smo testirali aplikacijo,... Ugotavljanje vzrokov nam je v tovrstnih situacijah vzelo veliko časa, a z dobro mero vztrajnosti in potrpežljivosti so vsa vprašanja dobila svoje odgovore in vsi problemi svoje rešitve.

Pri izdelavi diplomskega dela so mi prav prišle izkušnje, ki sem jih pridobil z izdelavo spletnih mest ter poznavanje platforme Arduino, ki sem ga uporabljal za izdelavo seminarja tekom študija. Z izdelavo aplikacije za mobilne naprave ter izdelavo krmilnega vezja sem se srečal prvič, zato mi je bil to precejšen izziv, a verjamem, da mi bodo pridobljena znanja v prihodnosti zagotovo še

koristila.

Z delovanjem trenutne verzije produkta sem zadovoljen, uspešno smo posneli tudi že nekaj daljših testnih posnetkov. Gotovo obstaja še marsikatera pomanjkljivost, ki se bo pokazala z nadaljnjo uporabo, kar bomo v naslednjih razvojnih iteracijah odpravili. Nedvomno se bo med nadaljnjim testiranjem porodila tudi marsikatera ideja za poenostavitev upravljanja s sistemom in pohitritev celotnega snemalnega procesa. K izboljšavam sistema bomo tako stremeli tudi v prihodnje, saj se bo po zaključku tega diplomskega dela, delo na projektu nedvomno nadaljevalo.

Literatura

- [1] (2011) Allegro MicroSystems, "DMOS Microstepping Driver with Translator and Overcurrent Protection". Dostopno na:
http://www.pololu.com/file/0J450/a4988_DMOS_microstepping_driver_with_translator.pdf
- [2] (2012) Digilent[®], "chipKIT[™] Uno32[™] Board Reference Manual". Dostopno na:
http://www.digilentinc.com/Data/Products/CHIPKIT-UNO32/chipKIT-Uno32-RevC_rm.pdf
- [3] (2011) DitoGear, "Timelapse dolly, slider, remote pan tilt head, motion control". Dostopno na:
<http://ditogear.com/products/>
- [4] (2012) GitHub, "BluetoothPlugin". Dostopno na:
<https://github.com/phonegap/phonegap-plugins/tree/master/Android/BluetoothPlugin>
- [5] C. Hellebuyck, *Getting Started with chipKIT[™]*. USA: Electronics products, 2011.
- [6] (2012) Kessler, "CineDrive". Dostopno na:
<http://kesslercinedrive.com/>
- [7] (2009) Kingbright, "KTIR0611S (Ver.9)". Dostopno na:
<http://www.kingbright.com/manager/upload/pdf/KTIR0611S%28Ver.9%29.pdf>

- [8] M. Kokalj, *Načrtovanje in izdelava motoriziranega sistema za pomik kamere: diplomsko delo*, Ljubljana, [M. Kokalj], v tisku.
- [9] S. Lavric, "Krmiljenje koracnih motorjev v teoriji in praksi," *Svet elektronike: revija za elektroniko, avtomatiko, racunalništvo in telekomunikacije*, vol. 7, št. 70, str. 46-49, 2000. Dostopno na:
http://www.svet-el.si/download/200/024_00.Krmiljenje%20koracnih%20motorjev%20v%20teoriji%20in%20praksi.pdf
- [10] (2009) LM Technologies, "AT command -Manual v4.52". Dostopno na:
<http://www.lm-technologies.com/downloads/modules/bluetooth-data-modules/LM400/at-command-manual/AT-command-Manualv4.52.pdf>
- [11] (2012) Phonegap, "Phonegap". Dostopno na:
<http://phonegap.com/>
- [12] (2011) M. Pilgrim, "The past present and future of local storage for web applications". Dostopno na:
<http://diveintohtml5.info/storage.html>
- [13] (2008) Sparkfun electronics, "LM400 – Bluetooth Module General". Dostopno na:
<http://www.sparkfun.com/datasheets/Wireless/Bluetooth/LM400%20Data%20Sheet%20DSS%20ver%202012.pdf>