



Št. naloge: 01687/2010

Datum: 01.09.2010

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

Kandidat: **MATEJ CIGALE**

Naslov: **KOMBINACIJSKA TIPKOVNICA Z MIŠKO**
COMBINATION KEYBOARD AND MOUSE

Vrsta naloge: Diplomsko delo univerzitetnega študija

Tematika naloge:

Kljub velikemu napredku računalniške tehnologije je tipkovnica še vedno osnovna naprava, preko katere človek komunicira z računalnikom. Standardna QWERTY tipkovnica je nastala leta 1873 in se od takrat skoraj ni spremenila. Čeprav je že dolgo znano, da je njen razpored tipk slab, se nobena od kasnejših izvedb ni uveljavila. Pač pa se je kot alternativa pojavilo več vrst enoročnih kombinacijskih tipkovnic, ki kljub manjšemu številu tipk omogočajo skoraj enako hitrost tipkanja. Zasnujte kombinacijsko tipkovnico, ki ima razpored tipk prilagojen porazdelitvi črk v slovenskem jeziku. Za izdelavo tipkovnice uporabite primeren mikrokrmilnik in vmesnik na osnovi USB standarda. Izdelajte programsko opremo, ki omogoča priključitev tipkovnice na osebni računalnik. Tipkovnico preizkusite in preverite pravilnost njenega delovanja.

Mentor:

prof. dr. Dušan Kodek



Dekan:

prof. dr. Nikolaj Zimic

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Matej Cigale

Kombinacijska tipkovnica z miško

DIPLOMSKO DELO
NA UNIVERZITETNEM ŠTUDIJU

Mentor: prof. dr. Dušan Kodek

Ljubljana 2010



Št. naloge: 01687/2010

Datum: 01.09.2010

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

Kandidat: **MATEJ CIGALE**

Naslov: **KOMBINACIJSKA TIPKOVNICA Z MIŠKO**
COMBINATION KEYBOARD AND MOUSE

Vrsta naloge: Diplomsko delo univerzitetnega študija

Tematika naloge:

Kljub velikemu napredku računalniške tehnologije je tipkovnica še vedno osnovna naprava, preko katere človek komunicira z računalnikom. Standardna QWERTY tipkovnica je nastala leta 1873 in se od takrat skoraj ni spremenila. Čeprav je že dolgo znano, da je njen razpored tipk slab, se nobena od kasnejših izvedb ni uveljavila. Pač pa se je kot alternativa pojavilo več vrst enoročnih kombinacijskih tipkovnic, ki kljub manjšemu številu tipk omogočajo skoraj enako hitrost tipkanja. Zasnujte kombinacijsko tipkovnico, ki ima razpored tipk prilagojen porazdelitvi črk v slovenskem jeziku. Za izdelavo tipkovnice uporabite primeren mikrokrmilnik in vmesnik na osnovi USB standarda. Izdelajte programsko opremo, ki omogoča priključitev tipkovnice na osebni računalnik. Tipkovnico preizkusite in preverite pravilnost njenega delovanja.

Mentor:

prof. dr. Dušan Kodek



Dekan:

prof. dr. Nikolaj Zimic

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.

IZJAVA O AVTORSTVU
diplomskega dela

Spodaj podpisani Matej Cigale,
z vpisno številko 63020020,
sem avtor diplomskega dela z naslovom:

Kombinacijska tipkovnica z miško

S svojim podpisom zagotavljam, da:

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

V Ljubljani, dne

Podpis avtorja:

Zahvala

Na prvem mestu bi se rad zahvalil mentorju prof. dr. Dušanu Kodeku, za konstruktivne predloge pri vsebini besedila in njegovo pripravljenost na sodelovanje.

Zahvalil bi se mami, Ireni Cigale, ki je pridno pregledala moje pisanje in popravila nekaj napak. Potrebno se je zahvaliti tudi ostalim družinskim članom, ki so bili potrpežljivi z mano.

Zahvala gre tudi sestri Mojci Cigale Kršinar, ki je vztrajno skrbela, da so vse moje stvari prišle za menoj.

Kazalo

Povzetek	1
Abstract	3
1 Uvod	5
2 Delovno okolje	6
2.1 Mikrokrmilnika PIC18F4550 in PIC18F4450.....	6
2.2 Programator PICKit 2.....	6
2.3 MPLAB IDE in C18 C Compiler.....	7
3 Zasnova tipkovnice	8
3.1 QWERTY in Dvorak.....	8
3.2 Kombinacijske tipkovnice.....	10
3.3 Oblika in delovanje tipkovnice.....	11
3.4 Porazdelitev črk v slovenskem jeziku.....	12
3.5 Postavitev znakov na tipkovnico.....	14
3.6 Slabosti zasnove.....	16
4 Osnovna USB naprava	17
4.1 Nekaj besed o USB.....	17
4.2 Povezava naprave z USB.....	18
4.3 Deskriptorji.....	19
4.4 Naprave za povezovanje s človekom.....	19
4.5 Vežje, ki ga zahteva USB.....	20
5 Tipkovnica	22
5.1 Povezava potenciometrov.....	22
5.2 Povezava gumbov in diod.....	22
5.3 Končano vežje.....	23
5.4 Potek programa.....	24
5.5 HardwareProfile.h.....	24
5.6 Izbiranje poslanega znaka.....	26
5.7 Deskriptor HID poročila.....	26
5.8 Analogno digitalni pretvornik in miška.....	27
5.9 Prehod na mikrokrmilnik PIC 18F4450.....	27
6 Sklepne ugotovitve	28
6.1 Izboljšave.....	28
6.2 Obrazložitev postavitve.....	29
6.3 Težave realizacije.....	29
A Shema vezja	30
Seznam slik	31
Seznam tabel	32

Seznam uporabljenih kratic in simbolov

USB	Univerzalno serijsko vodilo – Universal serial bus
HID	Naprava za komunikacijo s človekom – Human interface device
QWERTY	standardni razpored tipk na tipkovnici

Povzetek

Cilj predstavljene diplomske naloge je zasnova in izdelava naprave, ki jo lahko uporabljamo namesto miške in tipkovnice, osnovane na vodilu USB.

V delu si ogledamo preteklost tipkovnice, ki jo vsi poznamo, in njen razvoj. Srečamo se z najbolj znano izboljšavo, Dvorakovo poenostavljeno tipkovnico. Ugotovitev, ki jih je Avgust Dvorak uporabil pri zasnovi svoje tipkovnice, so tudi uporabljene v oblikovanju naše tipkovnice. Za čim boljšo porazdelitev tipk po tipkovnici moramo poznati lastnosti slovenskega jezika. Njegove lastnosti, ki so za nas pomembne, so zato kratko predstavljene.

Tipkovnica nima polnega nabora tipk, zato ji rečemo tudi kombinacijska, saj znake določamo s pomočjo kombinacij tipk. V delu je tudi opisan postopek, po katerem je bila razvita naša tipkovnica.

Za razvoj tipkovnice je potrebno poznavanje protokola USB in njemu podrejenega standarda HID, ki definira posebno skupino naprav, imenovanih naprave za povezovanje s človekom. Zapisane so osnovne zahteve, ki jih mora izpolnjevati naprava, če želi komunicirati preko vodila USB, in kako poteka komunikacija z gostiteljem. Kratko so predstavljeni deskriptorji naprave, preko katerih gostitelj izve, za kakšno napravo gre. Podatki, zbrani v nalogi lahko služijo kot kratek uvod v delovanje protokola USB.

Podana je tudi konkretna realizacija kombinacije tipkovnice in miške z mikrokontrolnikom PIC 18F4550, podjetja Mikrochip. Prikazani so vsi deli, ki jih potrebujemo za delujočo napravo priključeno na USB. Poleg tega so opisana še ostala vezja, ki so potrebna za realizacijo tipkovnice in miške. Zapisan je način, s katerim določamo delovanje tipkovnice.

Ključne besede: USB, HID, PIC18F4550, tipkovnica, miška, Dvorak, QWERTY, porazdelitev črk.

Abstract

The object of this thesis is design and construction of a device that can replace a standard keyboard and mouse.

The history of the standard keyboard is presented and the most known alternative the Dvorak Simplified Keyboard is described. The basic principals that August Dvorak used in designing his keyboard are also used in our design. In other to achieve this the knowledge of the characteristics of Slovenian language are required. The characteristics that we require to design our keyboard are presented in this work.

This keyboard does not employ a full set of keys, it is a cording keyboard because characters are derived from combinations of keys. The procedure used to design our keyboard is also given in this thesis.

The development of a keyboard requires knowledge of the USB protocol and the standard HID that defines a class of devices named Human Interface Devices. The requirements that the device has to meet to successfully communicate over the Universal Serial Bus. The descriptors that help the host learn about the device are also briefly explained.

A full design of the keyboard and mouse based around the Microchip's PIC18F4550 microcontroler serves as an example for the realization of the device. The circuits needed to build a functional device connected to the USB are listed and the basic program is outlined as are the specific parts and code that are used in this project.

Key words: USB, HID, PIC18F4550, keyboard, mouse, Dvorak, QWERTY, Letter frequency.

Poglavje 1

Uvod

Računalniki so v zadnjih letih postali vse manjši in manjši. Velik napredek v razvoju tehnologije jim je omogočil, da imajo že majhne naprave, kot naprimer mobiteli, večjo procesorsko moč kakor namizni računalniki preteklega desetletja. Edina resna ovira pri manjšanju računalnikov tako že dolgo ni velikost njihove notranje zgradbe, temveč velikost vhodno-izhodnih naprav, s katerimi človek komunicira z računalnikom.

Naprava, ki sem jo v svoji diplomski nalogi vzel pod drobnogled, je tipkovnica, ki je eden najstarejših delov računalnika. Njeno osnovno oblikovanje sega še v obdobje prvih tipkalnih strojev. Prav presenetljivo je, kako malo truda je bilo vložena v ta nadvse pomemben del računalnika, ki je še vedno glavni način, preko katerega sporočamo svoje želje novodobnim napravam.

Tipkovnica, ki sem si jo predstavljal, mora zadostovati nekaterim zelo pomembnim nalogam. Mora zavzeti malo prostora, saj je prostor pri majhnih napravah, kot so na primer mobilni telefoni, omejen. Zaželeno je, da je uporaba dovolj enostavna, da ne bo zahtevala strme krivulje učenja. Če je le mogoče, naj dovoljuje enostavno popravljanje napak uporabnika. Njena konstrukcija naj bo taka, da vzpodbuja pravilno lego prstov, dlani, rok in celega telesa, in s tem omogoča udobnejše in učinkovitejše delo.

Poleg tega bi bilo priročno, če bi uporabniku omogočili še enostavno upravljanje z "miškinim" kazalcem, kar še dodatno pohitri delo z računalnikom.

Za vzorec vhodno-izhodne naprave, ki bi oblikovno lahko služila tem namenom, sem si izbral igralni plošček (gamepad), ki s svojo obliko in načinom uporabe omogoča dokaj sproščeno delo. To pa ne pomeni, da ideje ne bi bilo mogoče dokaj enostavno prenesti na druge oblike, kot so na primer tablični računalniki, mobilni telefoni in podobno.

Za izdelavo primera sem izbral mikrokrmilnik podjetja Microchip in sicer 18F4550 oziroma 18F4450.

Poglavje 2

Delovno okolje

2.1 Mikrokontroler PIC18F4550 in PIC18F4450

Mikrokontroler PIC18F4550 podjetja Microchip predstavlja enega izmed zmogljivejših mikrokontrolerov tega proizvajalca. Za razvoj je zanimiv predvsem zaradi lepega števila integriranih modulov. V mojem primeru je to predvsem USB komunikacijski modul (Universal Serial Bus communications module), saj je ta nujno potreben za komunikacijo preko USB protokola. Druge prednosti mikrokontrolerov so tudi preprosto serijsko programiranje v vezju (In-Circuit Serial Programming) in seveda velika količina uporabnikov tega čipa.

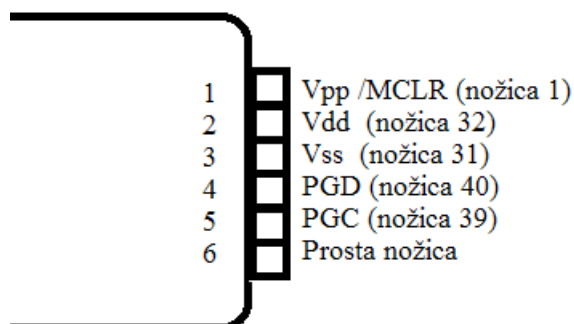
Mikrokontroler PIC18F4450 je novejša, okrnjena verzija PIC18F4550, ki je zaradi tega tudi cenejša. K sreči se izkaže, da je prehod med tema dvema mikrokontroleroma enostaven, saj obsega samo spreminjanje nekaterih nastavitev.

Razvoj moje diplomske naloge je potekal večinoma na PIC18F4550, na koncu pa sem prešel na cenejšo verzijo PIC18F4450. V obeh primerih gre za čip s štiridesetimi nožicami, kar omogoča enostavno povezovanje.

2.2 Programator PICKit 2

Gre le za enega izmed številnih programatorjev na trgu. Zanimiv je predvsem zaradi nizke cene in velikega števila podprtih Mikrokontrolerov. Tudi njegova povezava s čipom je nadvse preprosta. Ustrezne nožice (pin) priključka le povežemo na prave vhode na PIC.

Povezavo programatorja PICKit 2 na PIC18F4550 vidimo na sliki 3.1. Izhod 1 na programatorju je MCLR, ki ga povežemo na nožico 1 na mikrokontrolerju. Drugi in tretji izhod programatorja sta Vdd in Vss, torej napajalna napetost in ozemljitev. Četrti in peti izhod sta namenjena programiranju mikrokontrolerja in ju povežemo na nožici 40 in 39. Na šesti izhod ni vezana nobena funkcija programatorja.



Slika 3.1: Pomen posamezne nožice programatorja PICKit 2.

PIC omogoča neposredno napajanje vezja, kar v mojem primeru ni zaželeno, saj USB specifikacija prepoveduje, da bi naprava napajala vodilo. K sreči ima PICKit 2 zaznavanje lastnega napajanja naprave. Če tega zaznavanja ne bi imela oziroma bi želel, da se naprava kdaj poganja z interno baterijo, bi moral realizirati tudi zaščitno vezje, ki bi preprečilo morebitne težave na vodilu.

2.3 MPLAB IDE in C18 C Compiler

MPLAB Integrirano razvojno okolje (Integrated Development Environment - IDE) je programsko orodje, ki omogoča razvoj na Microchipovih produktih. Omogoča razvoj, programiranje in razhroščevanje (debugging).

C18 C Compiler je le eden izmed številnih prevajalnikov, ki je na voljo uporabniku. Uporabljal sem brezplačno različico, ki ne omogoča vseh optimizaciji. Kljub temu, da to ni najboljši prevajalnik, sem ga izbral, saj je brezplačen in dobro integriran v razvojno okolje.

Poglavje 3

Zasnova tipkovnice

3.1 QWERTY in Dvorak

QWERTY tipkovnica je nastala s tipkalnim strojem. Njen izumitelj Christopher Latham Sholes jo je leta 1873 ustvaril izkustveno. Razvil jo je, da bi zmanjšal število zatikanj pri ročicah tipkalnega stroja. Dodatno jo je spremenil, da je omogočil prodajalcem tipkovnic čim lažje pisanje besede typewriter (tipkalni stroj), zato je vse črke, ki so za to potrebne, premaknil v zgornjo vrstico.

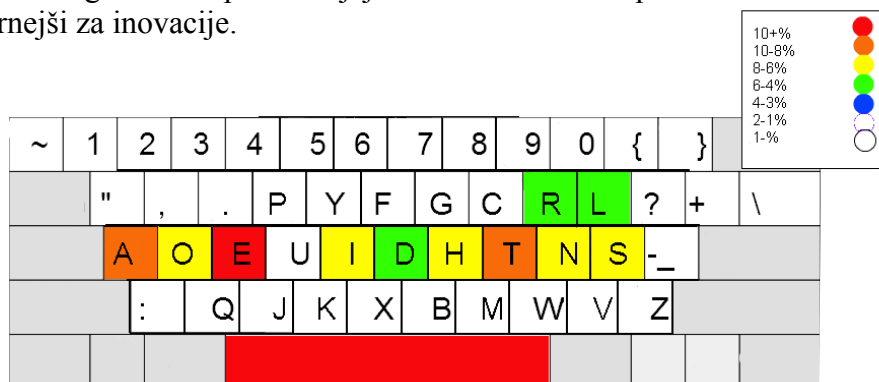
S prihodom prvih računalnikov se oblikovalci niso toliko ozirali na sam način tipkanja, temveč so postavitev tipk preprosto prevzeli, in tako omogočili lahek prehod iz tipkalnega stroja na računalnik.

Dodatna motivacija za ohranjanje istega razporeda tipk je povezovalni efekt (network effect [3]). Preprosto povedano to pomeni, da je nek produkt vreden več zato, ker ga uporablja večje število ljudi. V primeru tipkovnice je to dokaj očitno, saj je prehod iz enega delovnega mesta na drugega lažji, če se uporablja enaka tipkovnica.

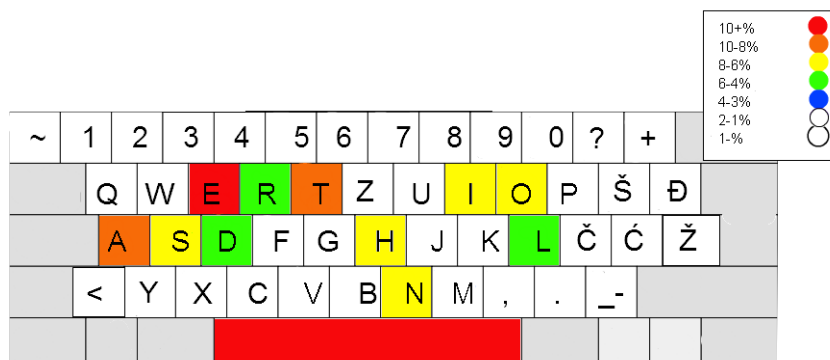
Leta 1936 je August Dvorak patentiral svoj razpored tipk - Dvorak Simplified Keyboard. Ta je bila razvita na podlagi njegovega raziskovanja, kako pravzaprav tipkamo, kateri načini tipkanja so najhitrejši, ter seveda raziskovanja pogostosti posameznih znakov in zaporedij znakov v angleščini. Dokopal se je do nekaterih pomembnih spoznanj:

- Najhitreje je, če si tipke sledijo izmenično med rokama.
- Najhitreje tipkamo, če so pogosti znaki in pari znakov na srednji vrstici (home row).
- Manj uporabljeni znaki naj bodo na spodnji vrstici, ki je najteže dostopna.
- Desnica bi morala tipkati več, saj je večina ljudi desničarjev.
- Pogoste dvojice znakov naj ne bodo na sosednjih prstih.
- Za uporabnika je najlažje, če tipka proti sredini tipkovnice, od mezinca proti kazalcu.

Dvorakova inovacija se ni uveljavila. Kot razloge za neuspeh se navaja cena za ponovno izobraževanje uporabnikov, začetna upočasnitev in seveda cena za prehod. Zagovorniki inovacije kot razloge za neuspeh navajajo tudi raziskave nasprotnikov in čas nastanka, ki ni bil najprimernejši za inovacije.



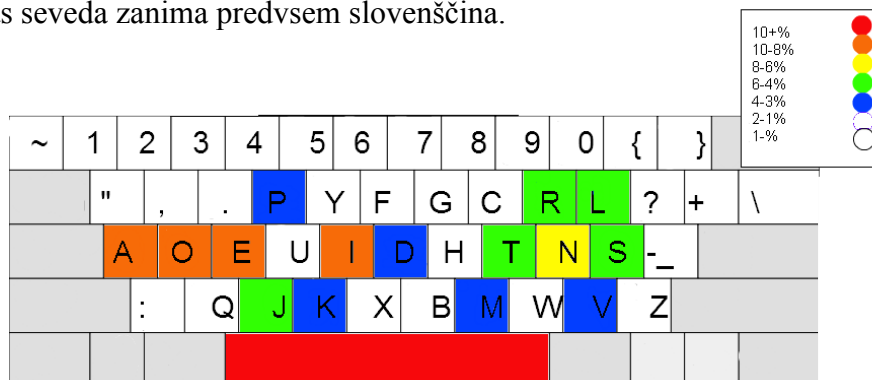
Slika 3.1: Dvorakova poenostavljena tipkovnica z verjetnostjo pritisnjenih znakov v angleščini.



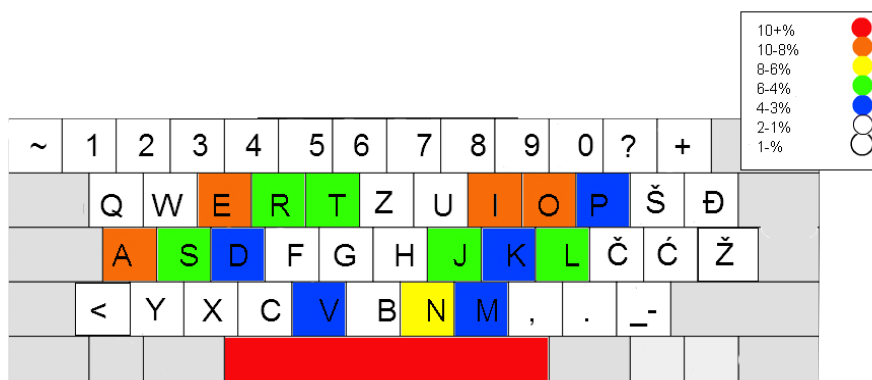
Slika 3.2: Standardna – QWERTY – tipkovnica z verjetnostjo pritisnjenih znakov v angleščini.

Na slikah 3.1 in 3.2 lahko vidimo razporeditev pogostosti znakov v primeru obeh tipkovic, ko govorimo o angleškem jeziku. Čeprav gre le za podatek o pogostosti znakov, lahko vidimo slabosti QWERTY nabora znakov.

A to seveda ne pomeni, da je Dvorakova tipkovnica boljša, ko pride do uporabe v drugih jezikih. Nas seveda zanima predvsem slovenščina.



Slika 3.3: Dvorakova poenostavljena tipkovnica z verjetnostjo pritisnjenih znakov v slovenščini.



Slika 3.4: Standardna – QWERTY – tipkovnica z verjetnostjo pritisnjenih znakov v slovenščini.

Na slikah 3.3 in 3.4 lahko vidimo razporeditev pogostosti znakov za slovenski jezik. Očitno je, da je že samo v tem pogledu Dvorakova tipkovnica komajda kaj boljša kakor običajna. Seveda pa se iz tako preprostega diagrama ne vidi drugih, prav tako pomembnih faktorjev, kot so izmenjava rok, razporeditev parov znakov in podobno.

3.2 Kombinacijske tipkovnice

QWERTY in Dvorakovo poenostavljeno tipkovnico bi lahko imenovali tipkovnice s polnim naborom tipk, saj ima pri obeh primerih tipkovnic približno toliko tipk, kot ima znakov. Navadno so izjema samo nekateri znaki, ki pri navadni uporabi niso pogosti.

A to ni nujno potrebno. Dovolj bi bilo tudi manj tipk, do posameznih znakov pa bi potem morali dostopati s kombinacijo tipk. To pravzaprav ni tako nenavadno, kot se morda sliši. Mobilni telefoni imajo osem tipk, s katerimi lahko vnašamo črke, do dodatnih tipk pa dostopamo z večkratnim pritiskom. Tudi navadna tipkovnica je pravzaprav kombinacijska, saj imamo posebno tipko, s katero dostopamo do velikih črk.

Eden izmed bolj razširjenih primerov kombinacijske tipkovnice je gotovo Stenograf, ki pa ima slabo lastnost, da izpisa ni mogoče enostavno pretvoriti v tekst, ki je razumljiv navadnemu uporabniku. Drug, podoben primer, je Hall Braille Writer, ki se uporablja za tipkanje znakov v Braillovi pisavi. Pri slednjem gre seveda za nekoliko drugačne razmere, saj se prilagaja značilnostim pisave za slepe, in je nekoliko neuporaben za večino prebivalstva.

Velika večina jih temelji na ideji enoročne uporabe. Primer takih rešitev so na primer: The Data Egg [4], FrogPad [5] in veliko število ljubiteljskih projektov, pa tudi takih, ki so prešli v prosto prodajo.

Kombinacijska tipkovnica je torej zelo zanimiv način vnosa teksta. Toliko bolj, če upoštevamo nekatere raziskave, ki pričakujejo, da je hitrost enoročnega kombinacijskega tipkanja 88% hitrosti dvoročnega [6].

Toda kombinacijske tipkovnice imajo tudi slabe strani. Najočitnejša je, da njihova uporaba ni tako preprosta kakor uporaba navadne tipkovnice, saj se mora uporabnik naučiti nekaterih pravil, če želi dostopati do črk. To vsekakor ni nepomembno. Izdelava naprave, ki je ljudje ne bodo uporabljali, je nesmiselna. Da pa bi povprečen uporabnik tipkovnico uporabljal, mora čim hitreje priti do trenutka, ko bo lažje uporabiti tako tipkovnico kakor vzeti navadno. Tega problema so se razvijalci lotili na najrazličnejše načine. Večinoma so bodisi poizkušali posnemati način uporabe, ki nam je znan iz običajne tipkovnice, ali pa tipke razpostavili tako, da si uporabnik lahko kombinacije zapomni s pomočjo slike.

Sam sem ubral nekoliko drugačen pristop. Glede na Dvorakova spoznanja sem se najprej odločil, da morajo imeti pogosto izbrane črke enostavne kombinacije za izbiro. Nadalje sem vse samoglasnike postavil pod levo roko, in tako dosegel, da se roki izmenjujeta, saj so v slovenščini grupe samoglasnikov zelo redke.

Pri obliki sem se zgledoval po igralnem ploščku in gumbe, s katerimi izpisujemo znake položil pod kazalca, sredinca in prstanca. Pri uporabi igralnega ploščka so ti gumbi pod napravo in uporabnik jih ne vidi.

Ker uporabnik tipk pri vpisu ne vidi, je pomembno, da se te hitro izpišejo na zaslonu. Poleg tega pa je tudi pomembno, da uporabnik lahko čim lažje popravi napačno izbrano tipko. Odločil sem se, da bo naprava izbrano kombinacijo takoj poslala računalniku, potem pa omogočala spreminjanje, če bo uporabnik spremenil trenutno izbrano kombinacijo. To ima seveda nekatere slabosti. Prva in najbolj očitna je, da naprava ne bo delovala pravilno, če program na drugi strani ne omogoča brisanja črk, saj je prva izbrana kombinacija zaradi hitrosti izpisa neredko napačna. Seveda je le malo programov, ki ne omogočajo brisanja napisanega besedila, tako ta težava ni tako velika, kakor se zdi na prvi pogled. Druga slabost je, da se s takim načinom uporabe najvišja hitrost vnosa besedila lahko drastično upočasni. Za zapis naslednje črke je namreč treba izpustiti pred tem pritisnjene tipke. Z nadaljnjim razvojem in raziskavami bi bilo mogoče nekatere izmed teh napak omiliti, če ne že povsem odpraviti, vendar bi to zahtevalo daljše raziskovalno obdobje z večjim številom uporabnikov.

3.3 Oblika in delovanje tipkovnice

Oblika tipkovnice je prva izmed pomembnejših odločitev, ki jih moramo sprejeti. Moja zasnova temelji na igralnem ploščku (gamepad). Sama oblika je privlačna, ker je oblikovana za nekoliko bolj naravno držo. Če roke sprostimo, namreč zlezejo v dokaj podoben položaj kakor pri uporabi igralnega ploščka. Tudi dlani so pri uporabi v sproščenem položaju. Oblika je torej za tipkovnice zelo primerna. Poleg tega omogoča uporaba take tipkovnice tudi udobnejši položaj telesa, saj nismo omejeni z dostopom do mize.

Druga odločitev, ki sem jo moral sprejeti, je število izbirnih gumbov. Izbral sem šest gumbov na spodnji strani, s katerimi določamo izbrano črko. Za šesterico sem se odločil, ker je to najmanjša soda kombinacija, ki omogoča izbiro vseh črk v abecedi brez vpeljave časovne funkcije. S štirimi gumbi ne moremo z eno potezo dostopati do vseh črk. ($2^4=16$). Pet gumbov bi zadoščalo, a bi to pomenilo lihost izbirnih gumbov. Liho število izbirnih gumbov samo po sebi ni slabo, vendar vnaša v obliko nesimetrijo, kar oteži postavitve gumbov. Poleg tega si s šestimi gumbi omogočimo dovolj svobode, da lahko zanemarimo nekatere kombinacije, ki niso lahko dostopne.

Gumbe za izbiro črk sem postavil pod kazalca, sredinca in prstanca. Mezinca tako ostaja prost, da z njim držimo napravo. Na zgornji strani naprave sem postavil analogno paličico, ki omogoča nadzor miške. S pritiskom na paličico uporabimo levi miškin gumb. Poleg paličice je postavljen še dodaten gumb, s katerim nadzorujemo desni miškin gumb. Na drugi strani naprave so še trije gumbi. Gumb za presledek in enter. Gumb za alternativo, s katerim lahko pišemo številke in druge posebne znake. Ter gumb, s katerim vključimo držanje Shift gumba. Ta je uporaben le v redkih primerih, saj velike črke pišemo tudi s podaljšanim držanjem izbrane črke.

Delovanje tipkovnice je nekoliko drugačno, kakor smo ga vajeni. Ob pritisku na gumb, ali njih kombinacijo, se črka izpiše. Če črka ni taka, kakor smo si želeli, imamo kratko časovno obdobje, v katerem lahko izbrano kombinacijo zamenjamo z nečim drugim, torej pritisnemo ali spustimo gumbe. To lahko ponavljamo nekaj časa. Znak prenehamo spreminjati, ko izpustimo kombinacijo tipk, ko torej ni izbrana nobena črka. Kadar kombinacijo držimo dlje časa, se črka spremeni v veliko tiskano. To s seboj nosi nekatere posledice. Vnos znakov je tako lažji, saj lahko vidimo svojo napako in jo, po potrebi, popravimo. Slabost je, da je pisanje znakov upočasnjeno, kajti za vsak vnesen znak moramo popolnoma izpustiti vse gumbe, ali čakati, da preteče čas, v katerem se gumb lahko spreminja.

Hitrost tipkanja lahko pri taki obravnavi vnosa pospešimo na dva načina. Prva možnost je, da zmanjšujemo čas, v katerem lahko spreminjamo izbrano črko. To je očitno slabo, saj s preveliko hitrostjo otežimo učenje in uporabo novemu uporabniku. Drugi način je, da kombinacije razpostavimo po tipkovnici tako, da se sorodne kombinacije znakov ne prekrivajo, torej, da moramo, pri pogostih kombinacijah znakov, gumbe popolnoma izpustiti. Ta možnost do določene mere pohitri vnos znakov, vendar tudi zmanjša največjo mogočo hitrost tipkanja, ker zahteva popolno spreminjanje prijema.

V svojem izdelku sem se odločil za daljši čas, v katerem je mogoče spreminjati znake, in za tako postavitve, ki pogoste pare znakov postavi na nepresečne položaje. Druga mogoča rešitev bi bile presečne kombinacije in zmanjševanje časa za popravilo vnosa, če uporabnik pravilno vnaša kombinacije.

V vsakem primeru pa potrebujemo za uspešno postavitve črk na tipkovnico podatke o pogostostih posameznih črk, dvojic in trojic v slovenščini.

3.4 Porazdelitev črk v slovenskem jeziku

Frekvenca črk in parov je, glede na povedano, vsekakor eden izmed pomembnejših podatkov, ki jih potrebujemo. Za čim boljši podatek o frekvenci je potrebno uporabiti dovolj velik vzorec, čim večjega števila avtorjev, da s tem odpravimo odstopanja.

Uporabil sem podatke Primoža Jakopina iz Doktorske disertacije: „Zgornja meja entropije pri leposlovnih besedilih v slovenskem jeziku“ [1], ki je za potrebe analize uporabil dva vzorca. Prvi vzorec je obsegal 16.784.110 znakov, drugi pa 2.497.308 znakov. Prvi vzorec vsebuje šestdeset del v slovenskem jeziku. Drugi vzorec pa vsebuje popoln vzorec posameznega avtorja. V tem primeru gre za vsa dela pisatelja Cirila Kosmača. Iz dela navajam frekvence znakov v slovenščini. Tabela 3.1 vsebuje verjetnosti znakov, če združimo oba vzorca. Tako sem se odločil, saj sem sklenil, da večji vzorec odtehta pristranskost, ki se ustvari z močnejšim nastopanjem izbranega avtorja.

Tabela 3.1: Porazdelitev znakov v slovenščini.

Znak	% male	% velike	% skupaj	Znak	% male	% velike	% skupaj
A	10,34	0,12	10,46	L	5,19	0,07	5,26
B	1,84	0,11	1,95	M	3,15	0,15	3,3
C	0,64	0,03	0,66	N	6,14	0,19	6,33
Č	1,43	0,04	1,47	O	8,98	0,11	9,08
D	3,3	0,08	3,38	P	3,12	0,24	3,37
E	10,68	0,03	10,71	R	4,95	0,07	5,02
F	0,08	0,03	0,11	S	4,88	0,18	5,06
G	1,58	0,05	1,64	Š	0,97	0,03	1
H	1,01	0,04	1,05	T	4,12	0,22	4,34
I	8,94	0,11	9,04	U	1,85	0,03	1,88
J	4,6	0,08	4,68	V	3,61	0,16	3,77
K	3,51	0,19	3,7	Z	2	0,11	2,11
				Ž	0,62	0,01	0,64

Seveda ta vzorec ni nujno najboljši, saj vsebuje le leposlovna besedila. Veliko primernejše bi bilo, če bi uporabili vzorec, ki bi vseboval tudi drugačne publikacije v realnih odmerkih. A že sama postavitve primerne vzorca bi bila zelo težavno opravilo. Do odstopanj seveda prihaja hitreje, kakor si mislimo, kar je iz navajanega dela očitno.

V tabeli 3.2 so posamezne verjetnosti črk predstavljene v intervalih. Podatke za slovenščino sem dobil iz dela Primoža Jakopina. Tako lahko lažje odčitamo verjetnosti posameznih črk. Za primerjavo sem dodal tudi podatke za angleščino, ki sem jih povzel po članku „Letter frequency“ na wikipediji [7].

Razlike med vzorcema so za moje potrebe zanemarljive, kar je iz tabele 3.2 razvidno. Hitro lahko vidimo, da se pogostost znakov spreminja, zamenja se lahko celo njihov vrstni red, a vendar imajo znaki, ne glede na vzorec jezika, podobne verjetnosti. Če to ne bi veljalo, ne bi bilo mogoče izdelati tipkovnice, ki bi jo lahko uporabljali vsi slovensko govoreči uporabniki.

Prav tako lahko iz tabele 3.2 razberemo, zakaj moramo za drug jezik črke ponovno razporediti po tipkovnici. Odstopanja so tukaj mnogo večja. Kljub temu opazimo nekatere podobnosti v verjetnosti črk. To je seveda, do neke mere, razumljivo. Žal pa je posledica naključja in še zdaleč ne velja za vse jezike, ki uporabljajo latinsko pisavo.

Tabela 3.2: Primerjava vzorcev. Za primerjavo so podani še podatki za znake v angleščini.

%	Prvi vzorec	Drugi vzorec	Oba vzorca	Iz podatkov za angleščino
10+	AE	AE	AE	E
10-8	IO	IO	IO	TA
8-6	N	N	N	OINSH
6-4	JLRST	JLRST	JLRST	RDL
4-3	DKMPV	DKMPV	DKMPV	
3-1	BGUZČH	BGUZČHŠ	BGUZČHŠ	CUMWFGYPB
1-0	CFŠŽ	CFŽ	CFŽ	VKJXQZ

Drugi pomemben podatek pri razporeditvi znakov na tipkovnici so pogoste dvojice in trojice znakov, ki jih vidimo v tabeli 3.2. Ta podatek nam pomaga odpraviti kombinacije, katerih vnos je težaven, in olajšati vnos znakov, ki se pogosto pojavljajo skupaj. Dober primer tega je v slovenščini kombinacija "je", ki je daleč najpogostejša. Bilo bi zelo neprijetno, če bi bila ta kombinacija počasi oziroma težko dostopna. Pri teh podatkih lahko pride do odstopanj, ki bi lahko spremenila samo postavitev znakov po tipkovnici. V tabeli so navedeni samo primeri, ki ne vsebujejo presledkov, saj je za presledke odgovorna posebna tipka, torej presek ni mogoč.

Tabela 3.3: Dvojice in trojice znakov. Leva tabela je iz prvega vzorca, desna iz drugega vzorca.

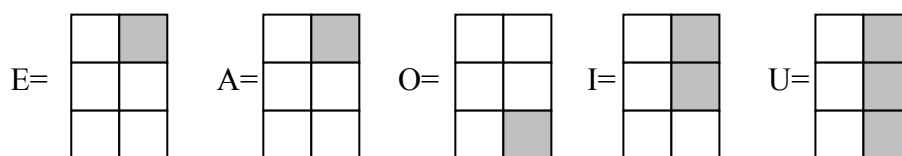
1.	je	264.492	pre	43.926	1.	je	39.931	ako	6.389
2.	na	146.630	pri	40.525	2.	se	24.141	nil	5.828
3.	ni	145.845	bil	39.837	3.	in	22.032	pre	5.321
4.	se	143.604	ako	35.993	4.	il	21.711	bil	5.207
5.	ra	136.715	nje	35.076	5.	ni	21.238	ila	4.982
6.	po	134.618	ost	34.755	6.	la	21.156	pri	4.868
7.	st	134.288	ega	33.266	7.	ra	20.310		
8.	al	133.657	sta	33.002	8.	po	19.984		
9.	il	133.299			9.	al	19.966		
10.	pr	129.557			10.	re	18.778		
11.	ko	128.468			11.	na	18.629		
12.	in	126.556			12.	ko	17.723		
13.	re	124.223			13.	st	17.144		
14.	la	123.822			14.	ka	16.891		
15.	ne	116.207			15.	pr	16.753		
16.	li	113.499			16.	ne	15.524		
17.	da	105.078			17.	da	15.182		
18.	ti	104.037			18.	el	14.927		
19.	en	103.772			19.	li	14.428		
20.	el	101.436			20.	aj	13.635		
21.	te	97.897			21.	ri	13.544		
22.	ka	94.523			22.	en	13.156		
23.	ri	94.117			23.	za	13.124		
24.	no	93.439			24.	ti	12.966		
25.	em	92.866			25.	ta	12.768		

Tabela 3.3: Dvojice in trojice znakov. Leva tabela je iz prvega vzorca, desna iz drugega vzorca (nadaljevanje).

26.	ve	92.426	26.	em	12.749
27.	ov	89.293	27.	te	12.280
28.	aj	88.774	28.	no	12.215
29.	ta	82.405	29.	ak	11.796
30.	za	79.989	30.	ve	11.355
31.	od	79.756	31.	ar	11.039
32.	le	79.306	32.	le	10.933
33.	bi	78.900	33.	bi	10.323
34.	lo	78.441	34.	od	9.890
35.	an	76.467	35.	lo	9.776
36.	nj	75.104	36.	ro	9.560

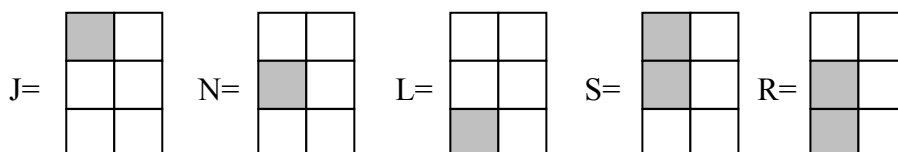
3.5 Postavitev znakov na tipkovnico

S temi podatki lahko začnemo razpostavljati znake na tipkovnico. Prva odločitev je, da najprej razpostavimo vse samoglasnike. Samoglasniki predstavljajo približno 41% delež vseh znakov. Odločil sem se, da jih postavim na desno stran tipkovnice. Za tako odločitev imamo več razlogov. Prvi je, da je za uporabnika prikladno, če znaki sledijo nekemu smislu. To, da so vsi samoglasniki dostopni z desno roko, si je lahko zapomniti. Drugi razlog je, da je bolje, če roki pri uporabi izmenjujemo. Ker bodo vsi samoglasniki na desni strani, najpogostejši soglasniki pa na levi, bo to omogočilo enakomerno obremenitev rok. Z uporabo enega samega gumba so nam dostopne tri črke. Najprej postavimo najpogostejšo črko, E. Ker je najmočnejši prst kazalec, jo postavimo pod njega. Sledita mu A in O. Za I potrebujemo kombinacijo. Najmočnejša, samo leva, je kazalec in sredinec. U je, v primerjavi z ostalimi, mnogo manj pogosta. Zanj uporabimo vse tri prste.



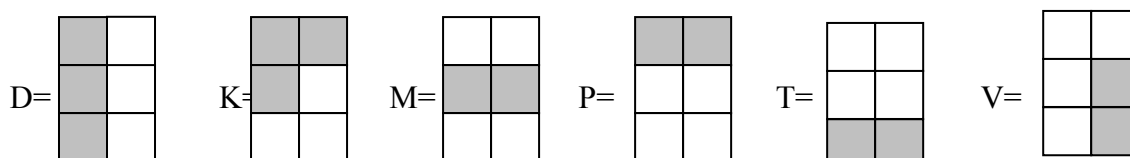
Slika 3.5: Razporeditev samoglasnikov.

Samoglasnikom sledijo najpogostejši soglasniki. J, L, N, R, S. Zaradi pogostosti kombinacije JE, sem J postavil v isto vrsto kakor E, torej pod kazalec. N, kot najmočnejši, je postavljen pod sredinec, L pa pod prstanec. S je kombinacija kazalca in sredinca, R pa kombinacija sredinca in prstanca. Seštevek verjetnosti vseh znakov skupaj je približno 26%. To pomeni, da je za zdaj leva roka veliko manj obremenjena kot desna. Presledek, ki zavzame 18% vseh pritiskov, skupno obremenitev leve roke dvigne na 44%. Seveda pa je ta pod palcem.



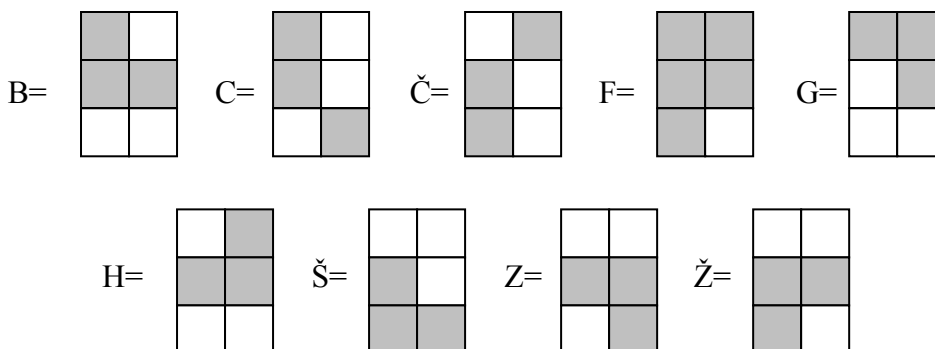
Slika 3.6: Slika najpogostejših soglasnikov.

Naslednje črke, ki še imajo velik odstotek, so D,K,M,P,T,V. Med njimi ima T nekoliko večjo verjetnost. Odločil sem se za razporeditev razvidno iz slike 3.5. Pri tej postavitvi sem se trudil tvoriti nepresečne množice za vpis črk, ki pogosto nastopajo v paru. P je pristal na svojem mestu, ker tako uide pogosti povezavi z R. T in M pa, ker so izbrane kombinacije s samoglasniki najmanj verjetne. V je na desni strani, saj je to edina preostala dvoprstna kombinacija. Verjetnost te skupine je 22%.



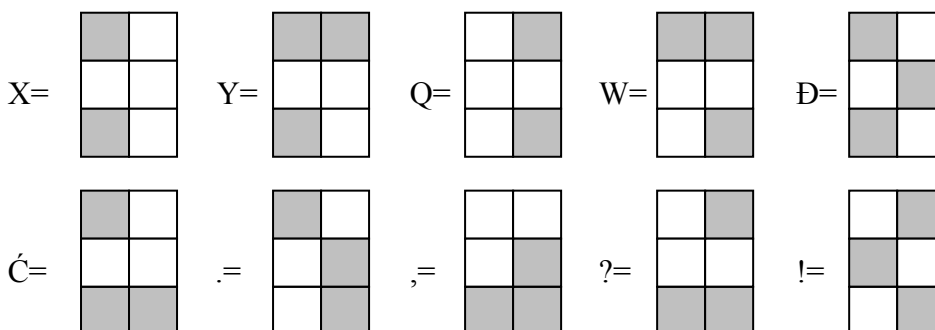
Slika 3.7: Postavitve črk D,K,M,P,T in V.

Preostale črke, torej B,C,Č,F,G,H,Š,Z in Ž, sem razporedil po prostih mestih. Teh deset črk ima verjetnost pojavitve zgolj 10%, torej si z njimi lahko privoščimo kombinacije, ki niso tako prijetne za uporabo.



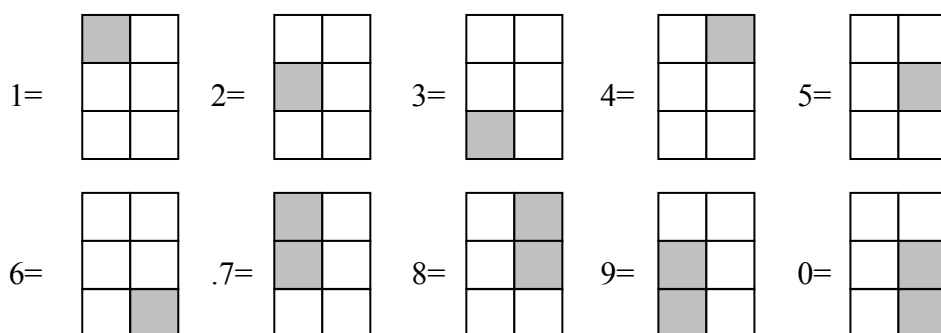
Slika 3.8: Postavitve črk B,C,Č,F,G,H,Š,Z in Ž.

Ostanejo nam samo še tuje črke, ki jih v slovenščini komajda zasledimo – X,Y,Q,W,Đ,Ć in ločila. Pri ločilih moramo biti pozorni na dejstvo, da le ta tvorijo pare. Tako je vrednost, ki jo pošljemo za piko, enaka vrednosti, ki jo pošljemo, če želimo poslati dvopičje. Razlikujeta se zgolj v delu poročila, ki določa veliko začetnico. Posledica tega je, da ne moremo povezati vejice in pike na isto kombinacijo.



Slika 3.9: Postavitve črk X,Y,Q,W,Đ,Ć in ločil.

Do številčk dostopamo s pritiskom izbirnega gumba "alt". Tu imamo tudi možnost izbire drugih znakov, ki so razporejeni tako kakor na običajni tipkovnici. Dobimo jih s pritiskom na tipko za velike črke. Taka razporeditev ni najboljša, vendar se prilagaja zahtevam protokola, ki je oblikovan na osnovi standardne tipkovnice.



Slika 3.10: Postavitev številčk. Izpisujemo jih s pritiskom na poseben gumb "alt" na zgornji strani tipkovnice.

3.6 Slabosti zasnove

Seveda ni treba posebej poudarjati, da ima taka zasnova tipkovnice svoje težave. Poleg že omenjenih težav s hitrostjo vnosa znakov, obstaja tudi mnogo drugih. Najbolj pereča je vsekakor ozkost zasnove tipkovnice. Razporeditev znakov je prilagojena tipu tipkovnice, načinu upravljanja z znaki in seveda jeziku. To vsekakor omejuje možnost uporabe.

Če bi tipkovnico spremenili, bi morali seveda celoten postopek postavljanja začeti na začetku. To velja, če bi želeli spremeniti jezik, obliko ali način prepoznavanja znakov v tipkovnici. Če bi se odločili, da bi ob zaporednem pritisku kombinacije izpisali drugačen znak, kar bi bilo v slovenščini smiselno, bi bil zaporedni pritisk s kazalcem lažja in boljše kombinacija kakor v nasprotnem primeru. Vendar tak način uporabe ni primeren, če obstaja sprejemljiva verjetnost, da se črka ponovi. V slovenščini je to zelo redek pojav, a to ne velja za vse jezike.

Prav tako je očitno, da bi, če bi znake prerazporedili pod druge prste, morali ponovno premisliti razporeditev znakov na tipkovnici. To vsekakor podraži razvojni cikel produkta. Razvijati ga je treba za vsako jezikovno področje posebej. Seveda ga je mogoče prilagajati osebnim potrebam uporabnika s preslikavami znakov v samem računalniku. To je do določene stopnje mogoče tudi z mojim izdelkom, vendar ima tak način pomanjkljivosti. Prednost in zahteva moje tipkovnice je v tem, da deluje brez dodatnih gonilnikov, da je torej povsem plug&play. To je dobro za uporabnika, saj poenostavi uporabo, kar je zelo pomembno. Od povprečnega uporabnika namreč ne moremo pričakovati, da bo uporabljal napravo, s katero se mora boriti, da jo pripravi do delovanja.

Poglavje 4

Osnovna USB naprava

4.1 Nekaj besed o USB

Ker je zaželeno, da je tipkovnica dostopna čim večjemu krogu uporabnikov, je odločitev, da se uporablja Univerzalno serijsko vodilo - USB (Universal Serial Bus) povsem logična. Poleg tega, da je to eden izmed redkih priključkov, ki jih lahko pričakujemo na katerem koli računalniku, ima tudi druge, zelo pomembne lastnosti, ki so zaželeni pri moji realizaciji.

Obstajajo trije aktualni standardi USB: 1.0, 1.1 in 2.0. Obstaja tudi standard 3.0, vendar še ni dovolj razširjen, da bi lahko opravičili njegovo uporabo v tipkovnici. Standard, ki je trenutno v širši uporabi, je 2.0, ki omogoča prenose do 480 megabitov na sekundo in priključitev, z uporabo vozlišč (hub), do 127 naprav (device) na gostitelja (host).

Gostitelj, v večini primerov računalnik, je lahko samo eden, in vodi celotno komunikacijo po vodilu. Naprave se lahko le odzivajo na njegove zahteve. Gre torej za asimetrično zasnovano vodilo. Komunikacija po vodilu poteka s pomočjo cevi (pipes), ki so logični kanali. Cev se na napravi konča v končni točki (endpoint), ki jih ima naprava lahko več. Imamo dva tipa cevi: tokovne (stream), ki so namenjene enosmernemu prenosu podatkov, ter sporočilne (message), ki so dvosmerne.

USB standard 2.0 definira štiri tipe prenosov po ceveh. Gre za asinhroni (isochronous), ki je namenjen napravam, pri katerih je pomembna stalnost pretoka, ne pa morebitne napake. Primer so kamere in mikrofoni. Prekinilni (interrupt) prenos je namenjen napravam, ki potrebujejo stalno komunikacijo z gostiteljem, nimajo pa velikih zahtev po pasovni širini. To so tipkovnice, miške ter podobne naprave. Masovni (bulk) so veliki prenosi, ki zavzemajo vso preostalo pasovno širino. Za te prenose je značilno, da ne smejo biti časovno kritični, saj imajo ostali načini prednost. Zadnji tip prenosov po vodilu so krmilni prenosi (control transfers), ki se uporabljajo za preproste ukaze namenjene napravi. Med zgoraj naštetimi samo krmilni prenosi uporabljajo sporočilne cevi, ostali pa tokovne in so torej enosmerni.

Za mojo diplomsko nalogo so pomembne nekatere lastnosti, ki jih ima USB protokol. Kot na primer dokaj preprosta strojna oprema na strani naprave, ki se mora samo odzivati na zahteve. To seveda pomeni, da je gostitelj na računalniku bolj kompleksen. Operacijski sistem lahko naprave na USB vodilu krmili s standardnimi gonilniki, kar olajša razvoj naprav.

USB ima, kot vsi standardi, tudi slabe strani. Največja je seveda omejena oddaljenost naprave od računalnika, saj je ta omejena na 5 metrov. Poleg tega pa obstaja tudi pomanjkanje možnosti povezave dveh računalnikov preko USB in nezmožnost gostitelja, da pošlje sporočilo več napravam hkrati. Za razvoj naprave pa so ovire predvsem težavnost protokola, ki je potreben za komunikacijo z računalnikom, in zahteva, da morajo vsi izdelovalci, ki proizvajajo naprave, pridobiti identifikacijo proizvajalca (Vendor ID). Za ljubiteljske projekte zadošča identifikacija proizvajalca USB krmilnika.

Pri opisovanju cevi je pomembno, da vemo, da cev noter (IN) potuje iz naprave v gostitelja, ven (OUT) pa iz gostitelja v napravo.

4.2 Povezava naprave z USB

Če želimo, da gostitelj napravo zazna in jo kasneje tudi uporablja, mora naprava opraviti nekatere zahteve. To pomeni vzpostavitev povezave med napravo in gostiteljem ter njeno oštevilčenje (enumeration), kakor se imenuje določanje napravinega naslova. Da to uspe, se mora naprava pravilno odzvati na zahteve gostitelja.

Običajno vzpostavljanje povezave poteka po naslednjih stopnjah:

1. Napajana (Powered state): Naprava je vključena v USB vrata in prejema napajanje, če ga potrebuje.
2. Vozlišče zazna napravo: Začuti spremembo napetosti na D+ ali D- povezavi.
3. Gostitelj izve za obstoj nove naprave: Izve samo, da se je na vratih zgodil dogodek.
4. Vozlišče določi tip naprave: S pomočjo upora vezanega na D+ ali D-, zazna ali gre za napravo s polno hitrostjo (full-speed device) ali za napravo, ki podpira le nizko hitrost (low-speed device).
5. Vozlišče ponastavi napravo.
6. Ali naprava s polno hitrostjo podpira visoko hitrost:
To se poizve z zaporedjem posebnih stanj. J in K, s katerimi naprava in vozlišče določita svoje sposobnosti.
7. Vozlišče vzpostavi povezavo med napravo in vodilom.
8. Gostitelj pošlje *Get_Descriptor* zahtevo, da izve največjo velikost paketov: Gostitelj ne čaka na celoten deskriptor, ampak samo na podatke, ki ga zanimajo.
9. Gostitelj določi naslov.
10. Gostitelj poizve o sposobnostih naprave: Pošlje *Get_Descriptor* na novi naslov naprave. Tokrat sprejme celoten deskriptor.
11. Gostitelj določi in naloži gonilnike.
12. Gonilnik izbere nastavitve (configuration).

V teh korakih naprava potuje skozi štiri od šestih stanj.

- Napajano (Powered) – koraki med 1 in 7.
- Običajno (Default) – koraka 8 in 9.
- Naslovljeno (Address) – koraki 10, 11 in 12.
- Nastavljeno (Configured) – naprava deluje.
- Priključeno (Attached) – če naprava ne prejema napetosti, ker je vozlišče zaznalo napako ali pa mu je tako ukazal gostitelj.
- Odloženo (Suspend) – če naprava ne zazna dogajanja na vodilu več kot 3 milisekunde.

To so koraki, ki jih mora uspešno opraviti vsaka naprava, ki želi komunicirati preko USB vodila. Če oštevilčenje ni uspešno, naprava in gostitelj ne moreta komunicirati. V splošnem naprava ne sme pričakovati dogodkov ali njihovega zaporedja, temveč se mora samo pravilno odzivati na zahteve gostitelja. Pravilno odzivanje vključuje pošiljanje pravilne količine podatkov, opuščanje krmilnih prenosov, če je to potrebno, pošiljanje praznega paketa, ko je to potrebno in pošiljanje zavrnitve na zahteve, ki jih naprava ne pozna.

4.3 Deskriptorji

Deskriptor je podatkovna struktura, ki jo naprava pošlje gostitelju. Shranjeni so v spominu naprave, ter se z uporabo ne spreminjajo. V njih so informacije o napravi, ki gostitelju pomagajo določiti, za kakšno napravo gre, ter kako naj z njo komunicira.

Gostitelj najprej pošlje zahtevo po deskriptorju naprave, ki ga mora vsebovati vsaka naprava. Ta vsebuje splošne podatke o napravi, kot so velikost deskriptorja, verzija USB specifikacije, na podlagi katere je bila razvita, številko proizvajalca in naprave, največjo dovoljeno velikost paketa za končno točko 0 in število mogočih nastavitvev. Druga polja omogočajo določanje, za kakšno napravo gre.

Naslednji deskriptor, ki ga zahteva gostitelj, je deskriptor nastavitve. V njem so ugnezdjeni tudi drugi, podrejeni, deskriptorji, torej deskriptorji povezave in deskriptorji končne točke. Gostitelj lahko da zahtevo le za deskriptor nastavitve, ter poda velikost, ki naj se prenese. Ne more torej zahtevati samo deskriptorja povezave. Nastavitev ima lahko naprava več, vendar je lahko aktivna samo ena. Deskriptor nastavitve vsebuje podatke o svoji velikosti in tipu, ter o velikosti celotnega polja deskriptorja, z vsemi podrejenimi deskriptorji. Poleg tega vsebuje še število deskriptorjev povezav, nastavitve za vir napetosti, in največjo potrebno moč, ki jo naprava zahteva. Vsak deskriptor nastavitve vsebuje vsaj en deskriptor povezave. Ta poda podatke o tem, katere naloge in sposobnosti ima naprava. Več povezav je lahko aktivnih hkrati. Poleg velikosti in tipa vsebuje še opis povezave, ter tip naprave, ki jo ta naprava implementira.

Vsak deskriptor povezave ima lahko enega ali več deskriptorjev končne točke, ki definirajo končne točke. Končne točke 0 ni potrebno definirati, saj je nastavitev enaka za vse naprave. Deskriptor končne točke vsebuje dolžino, tip, naslov končne točke, torej njeno številko in smer, podatek o tem, katere načine prenosa podpira, največja velikost paketa, ki jo podpira, in največji čas med posameznimi zahtevami gostitelja.

Poleg standardnih deskriptorjev naprava lahko vsebuje še deskriptorje niza (string), v katerih so opisi, ter deskriptorje, ki so odvisni od tipa naprave. Primer takega deskriptorja je deskriptor HID.

Pravilno nastavljeni deskriptorji so nujno potrebni za pravilno delovanje naprave in so ena izmed najpomembnejših nalog pri razvoju same naprave.

4.4 Naprave za povezovanje s človekom

Da bi povečali inovativnost pri razvoju novih naprav, olajšali razvoj naprav, pa tudi njihovo uporabo, je bil razvit poseben standard, ki je opisan v dokumentaciji: "*Device Class Definition for Human Interface Devices (HID)*" [8]. Naprave za povezavo s človekom (Human Interface Devices – HID), kot jih definira standard, so naprave, ki omogočajo komunikacijo med človekom in računalnikom. To so lahko tipkovnice, miške in igralni pripomočki, pa tudi drugačni gumbi, stikala in drsniki. Mogoče so tudi druge naprave, ki ne zahtevajo neposrednega stika s človekom, ampak samo podajajo podatke na podoben način. Omejitev je le, da uporabljajo le en krmilni prenos in en vhodni prekinitveni prenos. Dodatno lahko uporabljajo še dodaten izhodni prekinitveni prenos.

Moč klasifikacije je v tem, da je na računalniku samo eden gonilnik, ki upravlja z vsemi HID napravami. To poenostavi nove naprave, saj lahko operacijski sistem upravlja z napravo brez posebnih gonilnikov, ampak preko gonilnika za HID naprave. Dodatna prednost je, da uporabniku ni potrebno nameščati posebnih gonilnikov za uporabljanje naprave.

Gonilnik HID naprav prebere poseben del bralnega spomina na napravi, kjer je shranjen opis naprave. Del tega opisa je *Report descriptor*, na podlagi katerega gonilnik izve, kakšne podatke naj pričakuje od naprave in kako naj jih interpretira. Ko naprava pošilja podatke gostitelju, jih ta razume tako, kakor so bili definirani v deskriptorju.

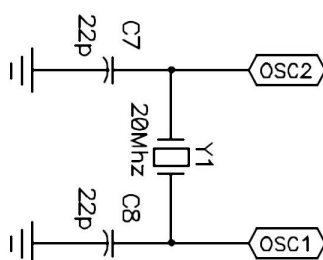
To si lahko predstavljamo kot nekakšno tabelo. V deskriptorju so opisi posameznih polj. V poročilih, ki jih gostitelj prejema iz naprave pa so trenutne vrednosti, ki opišejo stanje naprave.

Spoštovanje HID specifikacije je pomembno, saj le ta omogoča avtomatsko prepoznavanje naprave, ko jo priključimo v računalnik. Dodatna želja bi bila, da bi naprava, vsaj v omejeni meri, spoštovala tudi zagonski protokol (boot protocol), ki je potreben za uporabo naprave na sistemih, ki nimajo implementiranega celotnega HID gonilnika.

4.5 Vezje, ki ga zahteva USB

Za delovanje PIC mikrokrmilnika v povezavi z USB gostiteljem na računalniku, je potrebnih nekaj posebnih vezij.

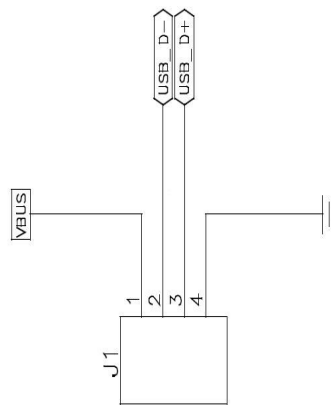
Prva je zunanja ura. PIC 18F4550 ima notranje mehanizme za določanje urinega signala, ki pa ne zadostujejo standardom USB. Na nožici OSC1 in OSC2 (nožica 13 in 14 na 40-pin PDIP ohišju) je potrebno pritrčiti 20MHz oscilator, in ga povezati z ustreznimi kondenzatorji. Priporočljivo je, da je oscilator eden izmed prvih delov, ki ga pritrčimo na vezje, saj je zaželeno, da je čim bližje mikrokrmilniku. Kapacitivnost kondenzatorjev je poljubna. Višja kapacitivnost poveča stabilnost, a zveča tudi zamik vzpostavitve.



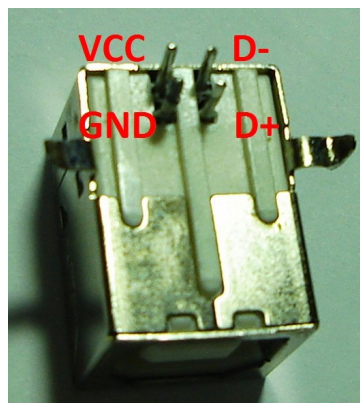
Slika 4.1: Priključitev oscilatorja.

Naslednje podvezje je kondenzator, ki ga moramo pritrčiti na 18F4550. V specifikaciji nisem dobil točnega podatka o želeni kapacitivnosti, zato sem uporabil kar takega, kot je v vezju PICDEM FS USB BOARD [9]. Gre za 470 nF kondenzator, ki ga je potrebno pritrčiti na nožico 18, imenovano V_{USB} .

Zadnji del je USB priključek. Sam sem uporabljal standardni B priključek, ki je sicer nekoliko okoren za to uporabo. Pomemben podatek je tudi, kam se na priključek povezujejo različne žice. To lahko vidimo na sliki 5.3.



Slika 4.2: Shema USB vtiča tipa B.



Slika 4.3: Fotografija priključka USB vtiča B z označenimi nožicami.

S tem vezjem se že lahko povežemo na računalnik in zaženemo testni program, da vidimo, če osnovno vezje deluje tako, kot je zaželeno. S tem vezjem lahko naredimo tudi osnovno zasnovno programa. To vezje je osnova ne le za tipkovnico, ampak za katerokoli napravo, ki bo z računalnikom povezana preko vodila USB.

Poglavje 5

Tipkovnica

5.1 Povezava potenciometrov

Kot prvi prilagodljivi element moramo, predvsem zaradi velikosti, na ploščico namestiti palčko, s katero bomo upravljali z miško. Pri postavitvi je treba biti pozoren tudi na to, da je treba za vstavitve nekaterih nožic narediti na ploščici dodatne luknje.

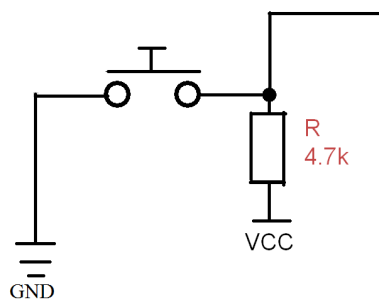
Sama palčka je sestavljena iz dveh potenciometrov in vzmeti, ki jo povrnejo v središčno lego.

Potenciometer je gradnik, ki nam, glede na položaj ročice, poda spremenljivo vrednost napetosti na nožici. Potenciometer ima tri nožice. Eno za napajalno napetost, eno za ozemljitev in tretjo, na kateri odčitamo dobljeno napetost.

To napetost moramo, če jo želimo uporabiti v našem programu, pretvoriti iz analogne v digitalno vrednost. Tukaj smo nekoliko omejeni pri izbiri nožic, saj ima PIC 18F4550 samo 13 pretvornikov iz analognega v digitalni signal. Sam sem ju povezal na nožici 7 in 8, torej na pretvornika 4 in 5.

5.2 Povezava gumbov in diod

Za povezavo izbirnih gumbov sem uporabil PORTD nožice, saj so na obeh straneh ohišja. Izbira povezav je nepomembna, ker lahko poljubno izberemo katerokoli nožico in jo potem povežemo s poljubno spremenljivko v HardwareProfile.h. Gumb sem razporedil izkustveno, glede na moje prste, in jih povezal z nožicami. Poizkušal sem jih povezati tako, da je čim manj presekov. Slika 5.4 prikazuje shemo gumba, povezanega na mikrokrmilnik. Opozoriti je treba, da je gumb ob pritisku 0, ko ni pritisnjen pa 1.



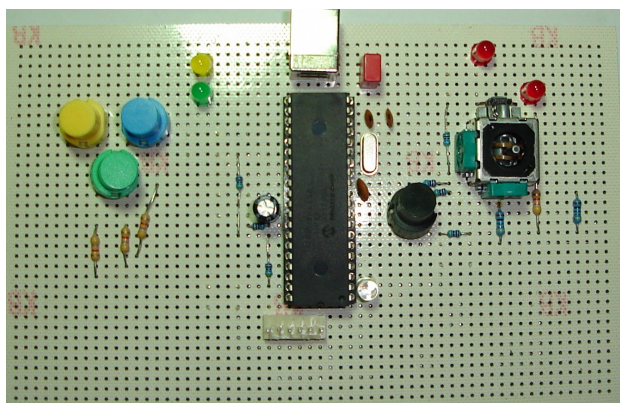
Slika 5.1: Shema gumba.

Ostale gumbes sem porazdelil predvsem ozirajoč se na postavitve le teh na ploščici, da je bilo povezovanje čim lažje s čim manj presekov, saj to oteži in podraži razvoj ploščic.

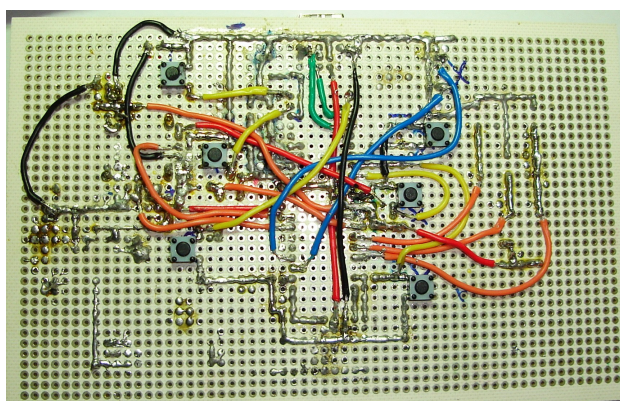
Diode so povezane na PORTA nožice. To delitev sem izbral predvsem zaradi lažjega programiranja. Za učinkovitejše povezovanje vezja bi jih verjetno morali povezati na kake druge nožice, a kot sem omenil, že pri gumbih so si nožice med seboj izmenljive. Izjema so le nekatere, ki imajo posebno funkcijo.

5.3 Končano vezje

Ko vezju za USB dodamo vse dodatke, ki jih želimo, je naše vezje končano. Na sliki 5.2 vidimo zgornjo stran vezja. Osrednji element je mikrokrmilnik, poleg njega pa lahko vidimo kristal. V zgornjem desnem delu vezja je palčka. Na levi strani imamo funkcijske gumbe. Zeleni gumb je presledek, rumeni vključi velike začetnice, z modrim pa dostopamo do številk. Črni gumb na desni strani vezja deluje kot desni miškin gumb. Levi miškin gumb je v sami palčki in ga poženemo z pritiskom nanjo.



Slika 5.2: Zgornja stran vezja.



Slika 5.3: Spodnja stran vezja.

Na sliki 5.3 lahko vidimo sliko spodnjega dela vezja. Na levi strani vidimo desne spodnje gumbe, na desni pa leve spodnje gumbe. Vidimo nesimetričnost postavitve gumbov. To je posledica postavitve palčke, ki spremeni postavitev dlani na vezju.

5.4 Potek programa

Diagram poteka programa si lahko ogledamo na sliki 5.4. Program se začne z inicializacijo sistema. Začne se standardni program inicializacije USB, sledi mu specifična inicializacija za uporabniški program.

Nato se v neskončni zanki izvajajo opravila USB in uporabnikova koda. Opravila USB vsebujejo procese, ki skrbijo za pravilno delovanje USB standarda, sprejemanje in odgovarjanje na SETUP pakete ter odzivanje na posebne dogodke. Paziti je treba, da se zanka odvijati dovolj hitro, saj se mora naprava pravočasno odzivati na prihod SETUP paketa. To lahko omejuje kompleksnost algoritma.

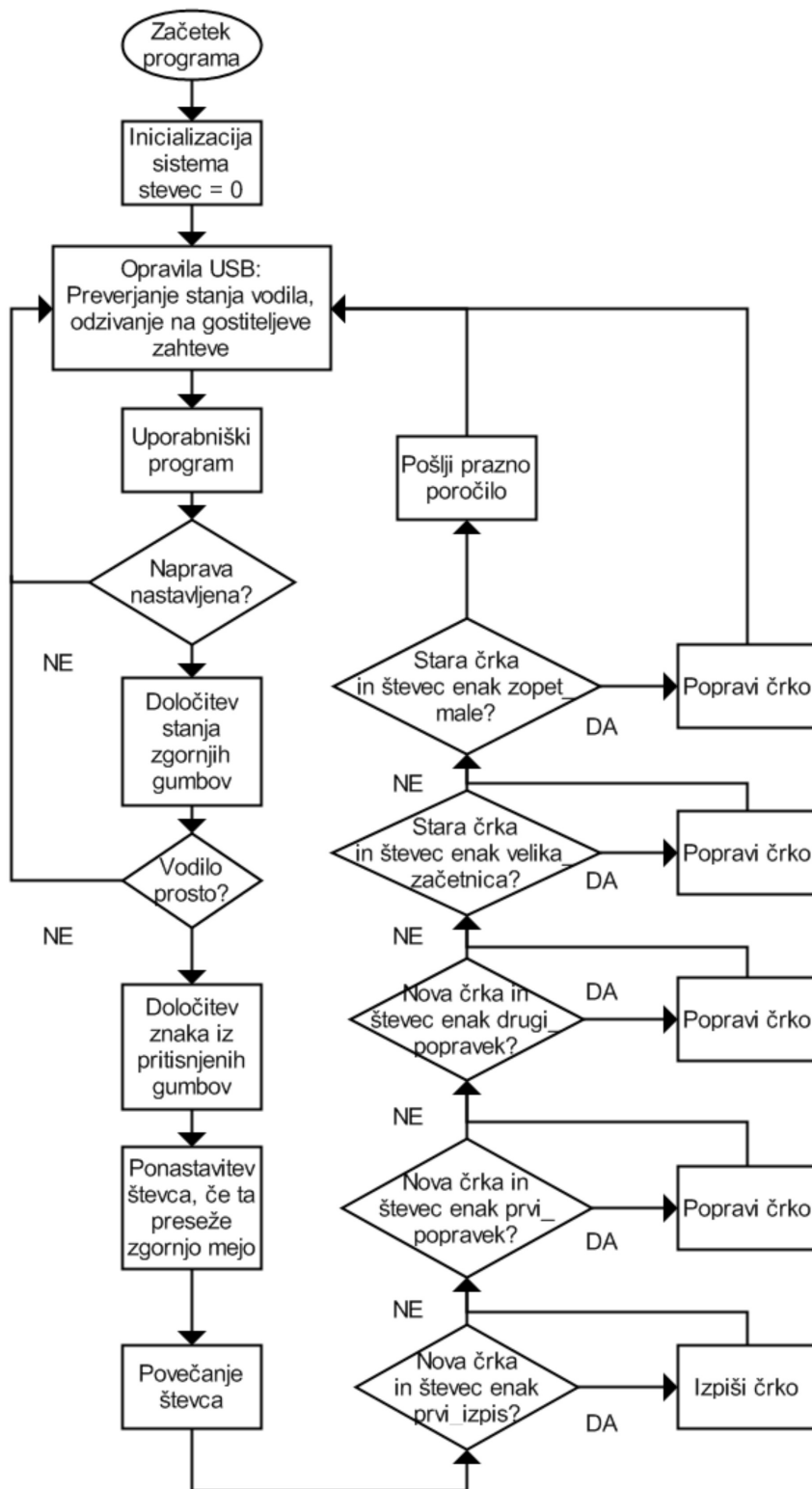
Uporabniška aplikacija se poganja le, če je naprava v "CONFIGURED_STATE" in ne v "Suspend". Oboje teh stanj je definiranih v USB Standardu. To je pravzaprav le preverjanje, če je gostitelj pripravljen na naše podatke.

Uporabniška aplikacija je serija odločitvenih stavkov, ki se jih lotimo z vsako iteracijo programa. V njej najprej preverimo, če je potrebno ponastaviti števec. To storimo, kadar presežemo zgornjo mejo, v mojem primeru je to 896, ali če ni pritisnjena nobena tipka. Nato se števec poveča in se primerja z vrsto konstant. Če je števec enak konstanti *prvi_izpis*, se izpiše črka. Števec se nato primerja s spremenljivkama *prvi_popravek* in *drugi_popravek*. Za *prvi_popravek* sem uporabil vrednost 128, za *drugi_popravek* pa 192. Če je izbrana kombinacija tipk drugačna kot prej, se pošlje ukaz za brisanje ene črke in popravljen črka. Če pa se kombinacija ne spremeni, se, ko je števec enak konstanti *velika_zacetnica*, črka nadomesti z veliko začetnico. To se v mojem programu zgodi pri vrednosti 448. Naslednja sprememba se zgodi, ko je vrednosti števca enaka konstanti *zopet_male*, v mojem primeru 704, ko se velika črka pobriše in zapiše mala. Če noben izmed teh pogojev ni izpolnjen, se pošlje prazno poročilo. Če tega ne bi naredili, bi se zadnja izpisana kombinacija ponavljala v neskončnost.

Med vsem tem dogajanjem se v določenih trenutkih odziva tudi miška. Ker se mora miška odzivati neodvisno od tipkovnice, se to zaporedje zgodi vedno, ko pridemo na vrednost pet po modulu šestnajst. V naslednjih petih ponovitvah odčitamo analogne vrednosti, ki jih dobimo iz potenciometrov, potem pa še vrednosti gumbov in jih v zadnji, deveti ponovitvi, pošljemo računalniku.

5.5 HardwareProfile.h

Ta datoteka vsebuje opis vezja. Z njenim spreminjanjem je mogoče enostavno opisati sestavljeno vezje. Tak način opisa omogoča, da je program neodvisen od samega vezja, pod pogojem, da so definirane vse nožice, ki jih potrebujemo. Tudi samo vezje je skoraj neodvisno od programa, saj lahko nožice poljubno prestavljamo. Omejitev velja le za pretvornike analognih signalov v digitalne, ker ti niso mogoči na vseh nožicah. Gre za dobro prakso v programiranju vezij. Datoteka je skupek definicij krajših procedur, ki jih kasneje lahko kličemo. Seveda bi lahko to datoteko popolnoma obšli in klice uporabljali, kjer jih pač potrebujemo, a bi s tem otežili spreminjanje vezja.



Slika 5.4: Diagram poteka programa.

5.6 Izbiranje poslanega znaka

Izbiranje znaka, ki ga bomo poslali iz kombinacije pritisnjenih tipk, je eno izmed pomembnejših nalog programa. Za čim boljše prilagodljivost je priporočljivo, da je funkcija lahko razumljiva in enostavno prilagodljiva.

Najprej se ustvari odločitveni vektor iz pritisnjenih tipk in posebne spremenljivke, ki nam določa alternativno izbiro znakov. Slednjo spremenljivko se spreminja v drugi funkciji v odvisnosti od pritisnjenega gumba na zgornji strani ploščice. Njena vrednost se ves čas javlja uporabniku na LED diodi.

Dobljeni vektor se, s pomočjo switch-case odločitvenega stavka, preoblikuje v bajtno kodo znaka, ki ga želimo. V tem odločitvenem stavku spreminjamo vrednosti, če želimo spremeniti razporeditev znakov.

Opozoriti je treba na posebne lastnosti razporeda znakov, ki jih lahko pošiljamo. Znaki so zakodirani tako, da sta na za vsako vrednost dva znaka, ki ju ločimo z uporabo posebnega znaka na prvem mestu poslanega poročila, ki je lahko bodisi 0, za male črke, ali 2, za velike. To privede do nekoliko neprijetnega položaja pri znakih, ki niso črke. Ti so namreč razporejeni tako, kakor jih lahko vidimo na tipkovnici. Če želimo poslati znak "%", moramo poslati poročilo, ki vsebuje vrednost za številko 5 in oznako, da gre za veliko črko. To nekoliko omeji naše možnosti pri pošiljanju posebnih znakov, saj se moramo prilagajati razporedu znakov na tipkovnici.

5.7 Deskriptor HID poročila

Deskriptor HID poročila pove gostitelju, katere in kakšne podatke naj pričakuje iz naprave, oziroma kakšne podatke naprava pričakuje od njega. Ker je HID standard narejen tako, da omogoča čim večje število najrazličnejših naprav, mora biti tudi deskriptor dovolj fleksibilen, da to omogoča.

Deskriptor se začne s poljem *Usage page*, ki je nekakšen splošen opis tipa naprave. Sledi mu *usage* polje, ki natančneje opredeli samo napravo. Naslednje polje je navadno *Collection*, posebno polje, predstavljamo si ga lahko kot nekakšen oklepaj, ki nam združi podatke, ki jih bomo pošiljali po povezavi. Za samo strukturo podatkov moramo določiti smer, minimum in maksimum, ki ga naprava pošilja, minimum in maksimum, kakor naj ga razume gostitelj, ter velikost in obliko podatkov, ki jih bomo poslali. Kadar je potrebno, dodamo še podlago (padding), da je poročilo večkratnik osmih bitov. Za vsak del poročila moramo sestaviti poseben del deskriptorja, tako da gostitelj ve, kakšne podatke sprejema. Ko je deskriptor poročila sestavljen, podatke pošiljamo kot navadno tabelo, če se seveda držimo svojih določil.

V mojem primeru je poročilo razdeljeno na dva dela s pomočjo polja *report ID*. To omogoča nekoliko preglednejše pošiljanje podatkov pri sami realizaciji, lahko pa se uporablja tudi, ko imamo zelo veliko podatkov, ki jih ne pošiljamo ob istem času, da zmanjšamo zasedenost vodila. Poročili se med seboj ločita po prvem polju, kjer je prvo označeno z 1, drugo pa z 2. Lahko bi seveda to povsem obšli in vedno pošiljali celotno poročilo.

Tak način ni povsem moja ideja, saj sem ga povzel po drugi realizaciji. Moja osnovna ideja je temeljila na dveh končnih točkah, žal pa nisem imel uspeha s poganjanjem te naprave.

5.8 Analogno digitalni pretvornik in miška

Pretvorba položaja palčke v premik miškinega kazalca se dogaja v več fazah. To moramo storiti zato, ker ne vemo, koliko časa bo pretvornik potreboval, da se nastavi. Lahko bi se zgodilo, da bi pretvornik potreboval predolgo časa in naprava ne bi pravočasno odgovorila na *Setup* paket. To bi povzročilo nepravilno odzivanje naprave.

V prvi fazi s posebnim ukazom nastavimo pretvornik za prvi potenciometer. Sam mikrokrmilnik omogoča več različnih nastavitvev, tako da lahko rezultate prilagodimo svojim potrebam. Za vezje, ki ga imamo, popolnoma zadoščajo običajne nastavitve. Po tem koraku pretvornik potrebuje nekaj časa, da se pripravi na pretvorbo. V naslednji iteraciji preberemo njegovo vrednost in jo pretvorimo, počakamo, da se pretvorba konča, nato pa jo preberemo in shranimo. Zdaj moramo celotni postopek ponoviti, saj je lahko v določenem trenutku delujoč samo en analogno digitalni pretvornik. V petem koraku odčitamo pritisnjene gumbе in rezultate zberemo skupaj v poročilo, ki ga pošljemo gostitelju.

5.9 Prehod na mikrokrmilnik PIC 18F4450

Prehod na mikrokrmilnik PIC 18F4450 je povsem preprost, saj je program popolnoma združljiv z njim, potrebno je le popraviti nastavitvene bite mikrokrmilnika, ki se med tipoma krmilnikov nekoliko razlikujejo. Prehod je zanimiv predvsem zaradi nižje cene mikrokrmilnika PIC 18F4450, ki je pri velikih serijah 30% cenejši. To zmanjša končno ceno naprave.

Poglavje 6

Sklepne ugotovitve

6.1 Izboljšave

Ker gre za preprost primer delovanja USB HID protokola, je seveda mogočih izboljšav nešteto. Naštel bom le nekaj takih, ki so se mi porodile med samo realizacijo.

Najprej bi omenil samo obliko naprave. Ta je kljub trudu nadvse neudobna za uporabo. Potrebna bi bila drugačna oblika, ki bi se bolje prilegala človeškim rokam. Mogoča izboljšava na tem področju bi bila prenos dveh tipk pod palca, ki sta pri trenutni obliki uporabljena veliko premalo. To bi seveda pridobilo želen učinek šele, ko bi celotno tipkovnico preoblikovali v obliko, ki je bolj podobna začetni ideji, torej igralnemu ploščku. S tem bi seveda nekoliko izgubili na splošnosti naprave. Izboljšave bi lahko bile deležne tudi alternativne tipke, ki so sedaj omejene zgolj na znake, ki so povezani s številkami.

Razporeditev tipk še ni nujno najboljša. Kot je že omenjeno, je vzorec, po katerem sem prevzel pogostost črk, izpeljan iz literarnih del. To samo po sebi ni slabo, ne odraža pa nujno pravega stanja na področju jezika. Seveda pa bi bila potrebna obsežnejša strokovna analiza s področja medicine, da bi tipkovnico oblikovali tako, da bi se čim bolje prilegala rokam in bi pokazala, katere kombinacije so resnično najboljše. Tudi sam način izbire zapisane črke je morebiti nekoliko vprašljiv, saj bi lahko z drugačnimi prijemi pospešili vnos črk. Z uporabo majhnega pomnilnega vezja bi lahko realizirali napravo, ki bi omogočala časovni zamik pritiskov na tipke in s tem lažjo uporabo.

Algoritem, ki nadzoruje miško, je preprost. Da bi bolje izkoristili napravo, bi bilo potrebno poiskati načine, v katerih primerih premikanje miške pospešiti ali upočasniti. Gumbi se zdijo malo neprimerni, poleg tega pa ostaja primanjkljaj koleščka, ki je zelo pomemben del vsake miške. Težava slednjega je predvsem visoka cena vezij, ki bi jih lahko uporabili v ta namen.

Zadnje in morebiti najpomembnejše področje pa je seveda na drugi strani žice. Naprava se prilagaja zahtevam po skladnosti računalniku. Seveda pa lahko postorimo mnogo, da se tudi računalnik odziva na naše želje. Z razvojem namenske programske opreme bi morebiti lahko dinamično spreminjali hitrost sprejemanja znakov in se tako prilagajali uporabnikovi spretnosti z napravo. Če bi uporabniški vmesnik prilagodili uporabi analogne palčke, bi lahko z njim upravljali skorajda tako hitro kot z miško. Nekatere lastnosti, ki bi jih lahko nadzoroval pameten program za vnos teksta, bi lahko samodejno nadomestile mnoge naloge, ki sedaj čepijo na tipkovnici. In nenazadnje bi veliko pridobili že s samim programom, ki bi novim uporabnikom na zabaven način približal uporabo tipkovnice.

Končna realizacija bi morala za zgled jemati obstoječe konzole, saj bi tako dobili močno že obstoječe zaledje, ki bi sprejelo nov način komunikacije z zunanjim svetom.

6.2 Obrazložitev postavitve

Zanimiva lastnost je, da pravzaprav ne gre za samo napravo, ki bi bila utrjena v eni sami obliki, ampak bi jo lahko prilagodili neštetim možnostim. Lahko bi uporabili dve fizično ločeni ročici, za vsako roko svojo, in tako omogočili še prestejši premik rok.

Povsem mogoče je celotno obliko preoblikovati v kakršno koli obliko, ki je primerna za situacijo, v kateri bi tipkovnico uporabljali, dokler bi se držali enakega načina vnosa kombinacij. Povsem mogoče je tipkovnico prenesti na ročaje stola, saj je prehod, ki je pri tem potreben za uporabnika, da preide na tako spremenjeno tipkovnico minimalen, kajti premikajo se le prsti, ki pa sledijo enakemu premikanju in zato lahko govorimo o nekakšnem mišičnem spominu.

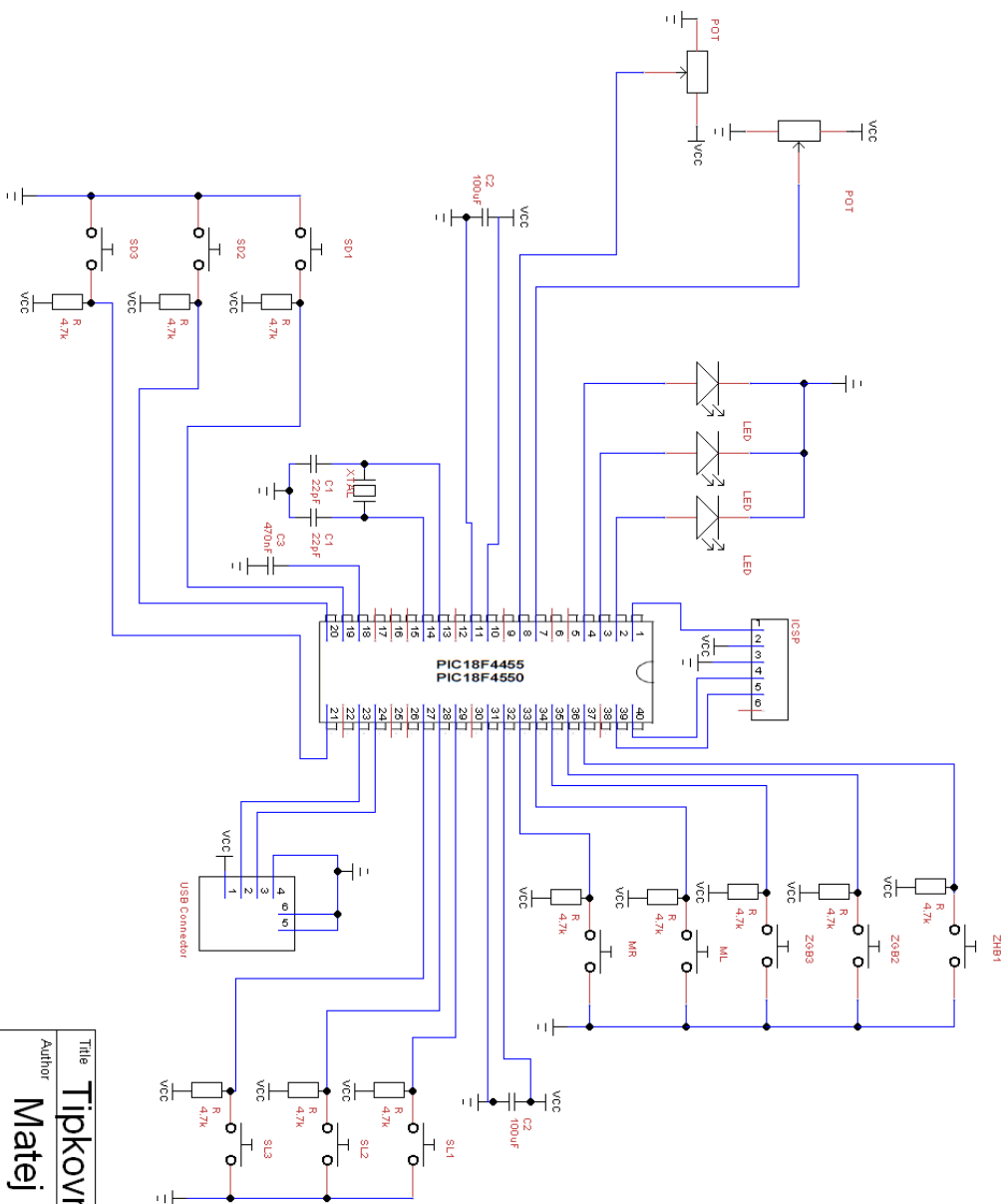
Idejo lahko brez posebnega truda prenesemo na volan avtomobila, zadnjo stran mobilnega telefona, dlančnika ali tabličnega računalnika, na katerih je navadno dovolj prostora. Povsem mogoče je tipke prenesti tudi na ročaje stola ali mize, pravzaprav kamorkoli, kjer potrebujemo vnos besedila, a ne želimo obremenjevati okolice s tipkovnico.

6.3 Težave realizacije

Kot pri vsakem projektu, je tudi pri tem nekaj področij, s katerimi preprosto nisem zadovoljen. Priznati moram, da sem se pri sami realizaciji nekoliko zataknil pri deskriptorjih, ki so mi povzročili nemalo težav. Zato mi kljub trudu ni uspelo realizirati naprave tako, kot sem si na začetku zastavil. Tako pravzaprav ne gre za kombinirano napravo, ampak za eno napravo, ki pošilja dva različna tipa poročil. Razlika se morebiti ne zdi pomembna, a naprava tako ne deluje, ko niso prisotni gonilniki za HID naprave, ne deluje torej zagonski protokol. To se morebiti ne zdi tako pereča težava, pomeni pa, da za zdaj naprava ne more biti edina tipkovnica priključena na računalnik, poleg tega pa nimamo nobenega zagotovila, da bo naprava delovala na vseh operacijskih sistemih, na primer takih, katerih gonilniki so le površne realizacije samega standarda.

Dodatek A

Shema vezja



Title	Tipkovnica z miško	
Author	Matej Cigale	
File	sestviJprabnik\Desktop\Diploma\Moje vezje1.dsn	Document
Revision	1.0	Sheets
Date	15. 10. 2010	1 of 1

Seznam slik

Slika 3.1: Pomen posamezne nožice programatorja PICKit 2.....	6
Slika 3.1: Dvorakova poenostavljena tipkovnica z verjetnostjo pritisnjenih znakov v angleščini.....	8
Slika 3.2: Standardna – QWERTY – tipkovnica z verjetnostjo pritisnjenih znakov v angleščini.....	9
Slika 3.3: Dvorakova poenostavljena tipkovnica z verjetnostjo pritisnjenih znakov v slovenščini.....	9
Slika 3.4: Standardna – QWERTY – tipkovnica z verjetnostjo pritisnjenih znakov v slovenščini.....	9
Slika 3.5: Razporeditev samoglasnikov.....	14
Slika 3.6: Slika najpogostejših soglasnikov.....	15
Slika 3.7: Postavitev črk D,K,M,P,T in V.....	15
Slika 3.8: Postavitev črk B,C,Č,F,G,H,Š,Z in Ž.....	15
Slika 3.9: Postavitev črk X,Y,Q,W,Đ,Ć in ločil.....	16
Slika 3.10: Postavitev števil.....	16
Slika 4.1: Priključitev oscilatorja.....	20
Slika 4.2: Shema USB vtiča tipa B.....	21
Slika 4.3: Fotografija priključka USB vtiča tipa B z označenimi nožicami.....	21
Slika 5.1: Shema gumba.....	22
Slika 5.2: Zgornja stran vezja.....	23
Slika 5.3: Spodnja stran vezja.....	23
Slika 5.4: Diagram poteka programa.....	25

Seznam tabel

Tabela 3.1: Porazdelitev znakov v slovenščini..... 12

Tabela 3.2: Primerjava vzorcev.

Za primerjavo so podani še podatki za znake v angleščini..... 13

Tabela 3.3: Dvojice in trojice znakov. Leva tabela je iz prvega vzorca,
desna iz drugega vzorca. 13

Tabela 3.3: Dvojice in trojice znakov. Leva tabela je iz prvega vzorca,
desna iz drugega vzorca (nad). 14

Viri

[1] P. Jakopin, „Zgornja meja entropije pri leposlovnih besedilih v slovenskem jeziku: doktorska disertacija“ Ljubljana Fakulteta za Elektrotehniko, 1999.

[2] J. Axelson, „USB Complete: Everything You Need to Develop USB Peripherals, Third Edition“ Lakeview Research LLC, 5310 Chinook Ln., Madison WI 53704, 2005.

[3] (5.9.2010) QWERTY. Dostopno na:
<http://en.wikipedia.org/wiki/QWERTY>

[4] (11.9.2010) Introducing the "Data Egg". Dostopno na:
<http://www.xaphoon.com/dataegg/>

[5] (11.9.2010) FrogPad. Dostopno na:
<http://www.frogpad.com/>

[6](12.9.2010) Half-QWERTY: A One-handed Keyboard Facilitating Skill Transfer From QWERTY. Dostopno na:
<http://edgarmatias.com/papers/ic93/index.html>

[7] (17.10.2010) Letter frequency. Dostopno na:
http://en.wikipedia.org/wiki/Letter_frequency

[8] (28.8.2010) Device Class Definition for Human Interface Devices (HID) Version 1.11. Dostopno na:
www.usb.org/developers/devclass_docs/HID1_11.pdf

[9] (12.6.2010) PICDEM FS USB Demonstration Board User's guide. Dostopno na:
ww1.microchip.com/downloads/en/DeviceDoc/51526b.pdf