

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Miha Gorše

**Z mobilnim telefonom
krmiljen intervalometer**

DIPLOMSKO DELO
NA UNIVERZITETNEM ŠTUDIJU

Mentor: prof. dr. Dušan Kodek

Ljubljana, 2010



Št. naloge: 01682/2010

Datum: 01.09.2010

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

Kandidat: **MIHA GORŠE**

Naslov: **Z MOBILNIM TELEFONOM KRMILJEN INTERVALOMETER
A MOBILE PHONE CONTROLLED INTERVALOMETER**

Vrsta naloge: Diplomsko delo univerzitetnega študija

Tematika naloge:

Sposobnost za brezžično krmiljenje naprav raznih vrst je v današnjem času zaželena in pogosta lastnost. Narediti napravo s to sposobnostjo zahteva poleg ustrezno zasnovane naprave tudi ustrezen daljinski upravljalnik. Razvoj takega upravljalnika je zahteven in je zato smiselno iskati drugačne rešitve. Zaradi nizke cene in vsesplošne razširjenosti je zelo primerna alternativa mobilni telefon. Današnji mobilni telefoni imajo namreč vgrajene sposobnosti kot je bluetooth, ki je zelo primeren za brezžično krmiljenje. To sposobnost uporabite pri razvoju fotografskega intervalometra, s katerim je mogoče prožiti zaporedje posnetkov ob različnih časovnih intervalih. Razvijte in izdelajte vso potrebno strojno in programsko opremo takega intervalometra. Ta naj poleg delovanja na daljinsko krmiljenje omogoča tudi samostojno delovanje preko ustreznega uporabniškega vmesnika. Intervalometer preizkusite in preverite pravilnost njegovega 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 objavljane ali izkoriščanje rezultatov diplomskega dela je potrebno pisno soglasje Fakultete za računalništvo in informatiko ter mentorja.

Besedilo je oblikovano z urejevalnikom besedil $\text{\LaTeX}$.



Št. naloge: 01682/2010

Datum: 01.09.2010

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

Kandidat: **MIHA GORŠE**

Naslov: **Z MOBILNIM TELEFONOM KRMILJEN INTERVALOMETER
A MOBILE PHONE CONTROLLED INTERVALOMETER**

Vrsta naloge: Diplomsko delo univerzitetnega študija

Tematika naloge:

Sposobnost za brezžično krmiljenje naprav raznih vrst je v današnjem času zaželena in pogosta lastnost. Narediti napravo s to sposobnostjo zahteva poleg ustrezno zasnovane naprave tudi ustrezen daljinski upravljalnik. Razvoj takega upravljalnika je zahteven in je zato smiselno iskati drugačne rešitve. Zaradi nizke cene in vsesplošne razširjenosti je zelo primerna alternativa mobilni telefon. Današnji mobilni telefoni imajo namreč vgrajene sposobnosti kot je bluetooth, ki je zelo primeren za brezžično krmiljenje. To sposobnost uporabite pri razvoju fotografskega intervalometra, s katerim je mogoče prožiti zaporedje posnetkov ob različnih časovnih intervalih. Razvijte in izdelajte vso potrebno strojno in programsko opremo takega intervalometra. Ta naj poleg delovanja na daljinsko krmiljenje omogoča tudi samostojno delovanje preko ustreznega uporabniškega vmesnika. Intervalometer preizkusite in preverite pravilnost njegovega delovanja.

Mentor:

prof. dr. Dušan Kodek



Dekan:

prof. dr. Nikolaj Zimic

IZJAVA O AVTORSTVU

diplomskega dela

Spodaj podpisani Miha Gorše,

z vpisno številko 63020046,

sem avtor diplomskega dela z naslovom:

Z mobilnim telefonom krmiljen intervalometer

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 13.10.2010

Podpis avtorja:

Zahvala

Zahvaljujem se mentorju za pomoč pri uresničitvi ideje.

Prav tako gre posebna zahvala vsem mojim bližnjim, ker so me vzpodbujali in mi stali ob strani.

Zahvaliti se moram tudi Sandri, ker je odstopila svoj odsluženi mobilni telefon v raziskovalne namene.

*Diplomsko nalogo posvečam vsem z idejami na
papirju.*

Kazalo

Povzetek	1
Abstract	2
1 Uvod	3
2 Mobilni telefon in Symbian	4
2.1 Splošno	4
2.2 PyS60 - Python za Symbian	5
2.3 Razvoj	6
2.3.1 Python in bluetooth	8
2.3.2 Uporabniški vmesnik	11
3 Intervalometer	14
3.1 Vezje	14
3.1.1 Arduino	14
3.1.2 Tipkovnica	15
3.1.3 LCD zaslon	16
3.1.4 Ostali elementi vezja	17
3.2 Bluetooth modul	19
3.3 Program	20
4 Upravljanje intervalometra	25
4.1 Mobilni telefon	25
4.2 Intervalometer	27
5 Sklepne ugotovitve	29
A Priloge	31
A.1 Slike naprave	31

B Priložen CD	36
Seznam slik	37
Literatura	39

Seznam uporabljenih kratic in simbolov

RFCOMM	(angl.) Radio Frequency Communication, protokol za prenos preko povezave bluetooth.
Bluetooth	(angl.) protokol za brezžično komunikacijo.
SMS	(angl.) Short Message Service, kratka sporočila s 160 znaki, ki jih pošiljamo preko mobilnega telefona.
OBEX	(angl.) Object Exchange Protocol, protokol za pošiljanje datotek in drugih objektov.
Bulb	(angl.) funkcija, vgrajena v fotoaparate, namenjena daljšim osvetlitvenim časom.
PIN	(angl.) Personal Identification Number, povezovalna koda za bluetooth protokol.
RX	(angl.) Receive, sprejemnik.
TX	(angl.) Transmit, oddajnik.
LCD	(angl.) Liquid Crystal Display, zaslon s tekočimi kristali.
USB	(angl.) Universal Serial Bus, univerzalno serijsko vodilo.
GSM	(angl.) Global System for Mobile Communications, omrežje mobilne telefonije.

Povzetek

Intervalna fotografija ni zelo razširjena, saj je za to potrebno dobro načrtovanje in veliko časa. Fotografije se nato pogosto še sestavijo v video posnetek, kar ni preprosto opravilo. Seveda intervalometri v takšni in drugačni obliki že obstajajo. Nismo pa še zasledili takšnega, ki se ga lahko krmili z mobilnim telefonom.

To diplomsko delo je bolj kot intervalni fotografiji posvečeno izdelavi aplikacije za mobilni telefon z operacijskim sistemom Symbian. Zasnovan in izdelan je bil intervalometer, s katerim upravljamo fotoaparati. Mobilni telefon in intervalometer komunicirata brezžično preko povezave bluetooth. Mobilni telefon in intervalometer sta torej povezana tako, da mobilni telefon služi predvsem za daljinsko upravljanje. Implementirano je tudi krmiljenje s sporočili SMS. Intervalometer se lahko uporablja tudi kot samostojna naprava, saj ga uporabnik upravlja z vgrajeno tipkovnico in LCD zaslonom.

Cilj diplomske naloge je bil raziskati možnosti brezžičnega krmiljenja oziroma upravljanja. Ta cilj je bil dosežen z uporabo mobilnega telefona in razvojne ploščice Arduino, ki komunicirata brezžično preko bluetooth povezave. Na koncu je nastal polno delujoč prototip intervalometra, ki se lahko uporabi za praktičen prikaz delovanja in za prikaz njegove uporabe.

Ključne besede:

intervalometer, intervalno fotografiranje, Symbian, Python za Symbian, Arduino, bluetooth, brezžično krmiljenje

Abstract

Intervalometer or time lapse photography is not very widely used because it requires good planning and a lot of time. Photos are then often put together into a video file which is not a simple procedure. One can buy a wired remote for the camera with time lapse function. But we haven't found one that can be controlled with a mobile phone.

In this thesis we are focusing on building an application for Symbian smartphone, rather than the time lapse photography itself. A photographic intervalometer that can be connected to a camera was designed and constructed. The mobile phone and the intervalometer communicate wirelessly via Bluetooth. Mobile phone is used to remotely control the intervalometer. It can also use SMS messages for this purpose. The intervalometer can also be used as a stand-alone device and the user operates it with the built-in keyboard and LCD screen.

The aim of this thesis was to explore the possibility of wireless communication and control. This goal was achieved using mobile phone and the Arduino development board which communicate wirelessly via Bluetooth. In the end we got a fully functional intervalometer prototype which can be used to demonstrate how the device works and how it can be used.

Key words:

time lapse, time lapse photography, Symbian, Python for Symbian, Arduino, Bluetooth, wireless control

Poglavje 1

Uvod

Brezžično krmiljenje naprav je dandanes najbolj zaželeno. Praviloma je za to potreben namenski daljinski upravljalnik ali podobna naprava.

Obstaja več različnih protokolov za brezžični prenos podatkov. Odločili smo se, da za upravljanje uporabimo kar mobilni telefon, ki zna komunicirati preko bluetooth povezave. Interakcija z uporabnikom torej poteka preko programa na mobilnem telefonu. Izbrali smo platformo Symbian, ki jo uporabljajo tudi t.i. pametni mobilni telefoni podjetja Nokia. To so mobilni telefoni, namenjeni zahtevnejšim uporabnikom, poslovnežem in drugim, ki potrebujejo in zahtevajo več, kot samo telefoniranje in pošiljanje kratkih sporočil SMS.

Na drugi strani bo bluetooth modul, ki bo povezan z razvojno ploščico Arduino. Le-ta sprejme ukaze, ki jih pošlje mobilni telefon in izvede določene operacije.

Demonstracija krmiljenja z mobilnim telefonom je intervalometer. To je naprava, ki na podani časovni interval proži fotoaparata. Deluje lahko tudi samostojno, brez mobilnega telefona. Intervalna fotografija je področje fotografije, kjer se iz več zaporednih fotografij sestavi video posnetek. Največkrat fotografiramo dogodke ali stvari, ki se odvijajo počasi. Na primer razcvet rože, premikanje polža in podobno.

Mobilni telefon smo uporabili za krmiljenje s sporočili SMS, za kar bi sicer potrebovali GSM modul. Intervalometer torej lahko upravljamo tudi preko SMS sporočil.

Glavni cilj diplomske naloge bo torej brezžično krmiljenje razvojne ploščice Arduino, ki lahko ob minimalni spremembi programa upravlja s poljubno napravo, tako preko uporabniškega vmesnika na mobilnem telefonu kot s sporočili SMS.

Poglavje 2

Mobilni telefon in Symbian

2.1 Splošno

Nokia je velik in pomemben igralec na trgu mobilnih telefonov. Trenutno imajo v navezi s Symbian sistemom največji tržni delež na trgu pametnih mobilnih telefonov.

Pametni mobilni telefoni imenujemo naprave, ki poleg osnovnih funkcij mobilnih telefonov - telefoniranje in pošiljanje kratkih sporočil, omogočajo tudi prenos podatkov, pošiljanje elektronske pošte, imajo funkcijo osebnega organizatorja, nekateri omogočajo tudi navigacijo. Imajo več načinov komuniciranja in povezovanja z drugimi napravami, med drugim tudi bluetooth.

Symbian operacijski sistem [1] teče na procesorjih ARM, okoli katerih so grajeni Nokijini pametni mobilni telefoni. Operacijski sistem je uradno postal odprtokoden februarja 2010. Operacijski sistem Symbian skuša biti čimbolj varčen, saj mobilne telefone napaja baterija. Pomembno je, da je kar se da varčen s prostorom, čeprav lahko uporabnik dokupi večjo pomnilniško kartico. Omogoča tudi večopravilnost. Testni telefon je Nokia E65 (slika 2.1). Na njem teče Symbian OS v9.1. Verzija operacijskega sistema je pomembna pri razvojni platformi, saj imajo novejšje verzije več funkcij in knjižnic.



Slika 2.1: Mobilni telefon Nokia E65.

Aplikacije lahko razvijamo v več programskih jezikih. Prvotno je bil ves razvoj v programskem jeziku C++, trenutno pa so na izbiro:

- C++,
- Qt,
- Python,
- Flash Lite,
- Java ME.

Odločili smo se za Python, saj je dobro dokumentiran in ima knjižnice za vse funkcionalnosti mobilnega telefona, ki jih potrebujemo.

2.2 PyS60 - Python za Symbian

Python je visokonivojski, objektno usmerjen programski jezik, kateremu hitro raste priljubljenost. Je dovolj enostaven za začetnike, hkrati pa nudi dovolj tudi naprednim uporabnikom. Glavna prвина Pythona je preprosta sintaksa, ki nas hkrati prisili, da pišemo pregledno in lahko berljivo kodo. PyS60 [2] je torej izpeljanka programskega jezika Python za operacijski sistem Symbian. Vsebuje splošne Python knjižnice, hkrati pa jih nadgrajuje za delo z mobilnim telefonom. Dobrodošla funkcionalnost je tudi uvoz in uporaba razširitev programskega jezika C ali C++. Tega Flash Lite ter Java ME ne omogočata.

Nekaj lastnosti PyS60:

- uporabniški vmesnik: meni, izbirni seznam, vnosna polja, dialogi, sporočila;
- grafika: različne barve, pisave, prikaz slik, ikon;
- delo s številčnico in ostalimi gumbi;
- komunikacija: TCP/IP, bluetooth;
- sporočila SMS, MMS, elektronska pošta;
- dosotop do spominske kartice, branje/pisanje/ustvarjanje datotek;
- kamera, GPS, telefon, sistemske informacije;
- koledar, kontakti.

Trenutno lahko uporabimo dve končni verziji PyS60, in sicer PyS60 1.4.5 ter PyS60 2.0.0. Prva je namenjena telefonom prejšnje generacije, temelji pa na Pythonu verzije 2.2.2. Ker na našem testnem telefonu verzija PyS60 2.0.0 ni delovala, je bila uporabljena nižja verzija PyS60, in sicer 1.9.6. Tako PyS60 1.9.6 kot PyS60 2.0.0 temeljita na enaki verziji Pythona (verzija 2.5.1).

2.3 Razvoj

Python je na prvem mestu namenjen hitri gradnji aplikacij, ki so lahko neke vrste demonstracijske aplikacije. Kompleksnejše in bolj dodelane aplikacije se še vedno programirajo v programskem jeziku C++. Aplikacijo, ki jo spišemo v programskem jeziku Python, najprej poženemo znotraj Python ukazne lupine. Program ima končnico `py`. Razvojnemu okolju pa je priložena tudi aplikacija, ki takšen Python program zapakira v format SIS. To je poseben format za Symbian, ki vsebuje program in namestitvene datoteke aplikacije. Takšna aplikacija se namesti posebej, torej je ni potrebno poganjati znotraj Pythonove ukazne lupine. Za začetek razvoja aplikacij potrebujemo razvojno okolje [3]. Namestiti moramo Python za Windows ter razvojno okolje za Symbian. Razvojno okolje se izbere glede na verzijo Symbian operacijskega sistema, ki teče na mobilnem telefonu. Za naš testni telefon smo izbrali paket `S60_3rd_FP2_SDK_v1.1_2`. Razvojno okolje vsebuje tudi emulator (slika 2.2), ki je namenjen testiranju aplikacije. Za pretvorbo Python datoteke v namestitveno SIS datoteko

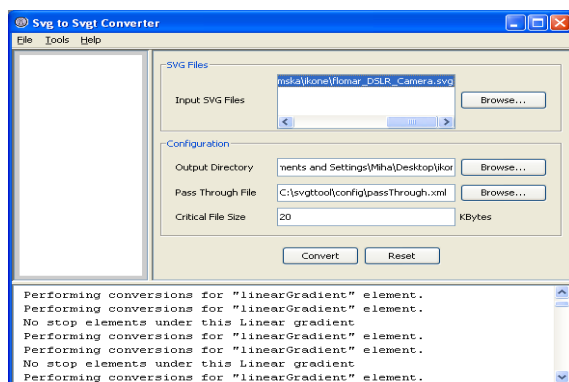


Slika 2.2: Zaslonski posnetek emulatorja za sistem Symbian.



Slika 2.3: Zaslonski posnetek orodja za pretvorbo v format SIS.

je priloženo orodje py2sis (slika 2.3). Če pa želimo svojo aplikacijo opremiti s svojo ikono (namesto privzete Pythonove), pa je na voljo orodje Svg to Svgt converter (slika 2.4). Nokia ima namreč prirejen format ikon, zato je potrebno ikono, ki je praviloma v formatu svg, pretvoriti v format svgt.



Slika 2.4: Zaslonski posnetek orodja za pretvorbo ikon.

Poleg Pythona za okolje Windows je potrebno namestiti tudi Python za Symbian. Tako imamo tudi v emulatorju okolje Python. Za Python aplikacije na mobilnem telefonu pa je potrebno namestiti tudi Python runtime ter Python Script Shell, obe datoteki sta priloženi razvojnemu paketu v formatu SIS. Po namestitvi se je pojavila napaka, ki je nastala zaradi manjkajočega vtičnika Open C/C++ Plugin. Postopek namestitve je tudi podrobno opisan na spletni strani [4]. Potrebno je še omeniti, da Python programov, pisanih za Symbian sistem, ni mogoče pognati v Python Windows okolju. Poslužiti se moramo emulatorja oziroma mobilnega telefona. Po uspešnem zagonu Hello-World programa smo lahko pričeli s programiranjem naše aplikacije. Programiranje je bilo razdeljeno na dva dela: bluetooth komunikacijo in sporočila SMS ter uporabniški vmesnik.

2.3.1 Python in bluetooth

Kot je že bilo omenjeno, ima PyS60 knjižnico za delo z bluetooth modulom [5], ki ne manjka nobenemu pametnemu mobilnemu telefonu. Podpira protokola RFCOMM ter OBEX. V tem primeru je bil uporabljen RFCOMM protokol, ki emulira RS-232 povezavo. Ta protokol nam ustvari navidezni serijski port. Torej bluetooth povezavo uporabimo tako, kot da bi mobilni telefon imeli povezan žično s serijsko povezavo. V Pythonu je potrebno uvoziti knjižnico za delo z bluetooth modulom. V PyS60 se ta knjižnica imenuje socket. Na

začetku programa uvozimo knjižnico z ukazom `import socket`. Tukaj je bilo zaradi nezdružljivosti z različnimi verzijami Pythona potrebno uporabiti trik. Knjižnico smo uvozili z *`import btsocket as socket`*. V novejši verziji se je spremenilo ime, iz `socket` v `btsocket`. Koda za vzpostavitev povezave:

```
def bt_connect():
    global sock
    try:
        arduino_addr='00:06:66:04:13:55'
        #MAC naslov Arduino bluetooth modula
        sock=socket.socket(socket.AF_BT, socket.SOCK_STREAM)
        target=(arduino_addr,1) #vzpostavitev serijske komunikacije
        sock.connect(target)
        return True
    except socket.error:
        return False #ob neuspeli povezavi vrne napako
```

Funkcija `bt_connect` torej vrne vrednost `True`, če je povezava uspešno vzpostavljena. MAC naslov bluetooth modula je vpisan kar ročno, saj gre za namensko aplikacijo, ki se vedno povezuje z istim bluetooth modulom. V primeru, da bi aplikacijo razvijali za splošni namen, bi jo bilo potrebno trivialno prirediti tako, da bi uporabnik sam izbral bluetooth modul, s katerim želi vzpostaviti povezavo. Če pa bluetooth modul ni dosegljiv, bodisi ni vklopljen ali pa je prišlo do kakšne napake, funkcija vrne `False`. V tem primeru se uporabniku izpiše obvestilo o napaki, aplikacija pa se nato zapre. Koda za pošiljanje podatkov preko bluetooth povezave:

```
def bt_send_data(tekst):
    global sock
    sock.send(tekst)
```

Funkcija za pošiljanje podatkov preko povezave bluetooth je lep pokazatelj, kako enostavno in hitro je programiranje v programskem jeziku Python. Programer mora samo pravilno podati besedilo, ki ga želi poslati, ostalo vse naredi knjižnica. Koda za sprejem niza znakov preko povezave bluetooth:

```
def bt_receive_data():
    line=[]
    while 1:
        ch=sock.recv(1)
        if(ch=='\n'):
```

```

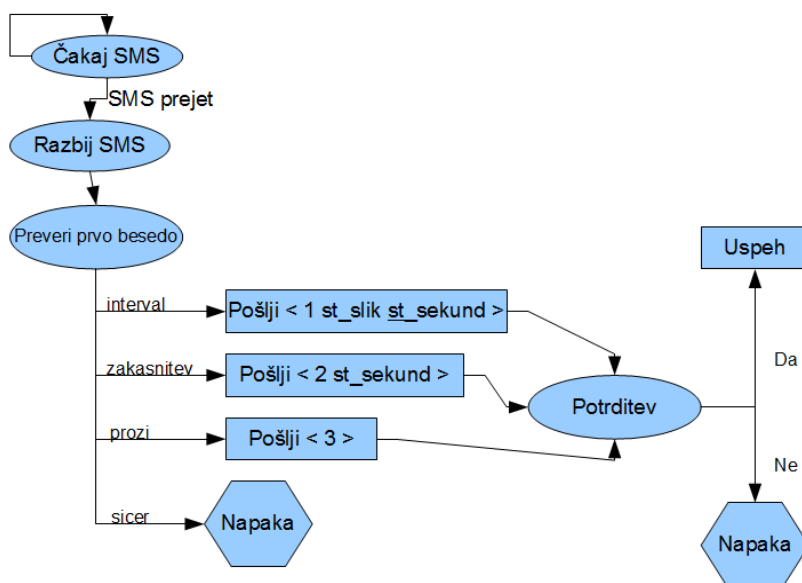
    break
    line.append(ch)
    return ''.join(line)

```

Sprejem niza poteka znak po znak v while zanki, dokler ne dobimo znaka za novo vrstico. Funkcija torej vrne prejeti niz. Uporabljena je za pridobitev povratne informacije od intervalometra, ko le-ta uspešno konča z zadano nalogo.

Ker to poglavje govori o komunikaciji, velja na tem mestu omeniti še krmljenje preko SMS sporočil. V tem primeru mobilni telefon služi kot GSM modul, torej mora biti nameščen nekje na doseg bluetooth povezave. Iz poljubnega mobilnega telefona pošljemo SMS sporočilo na testni telefon. Za koriščenje funkcij nabiralnika sporočil je potrebno uvoziti knjižnico inbox [5].

Ob prejemu SMS sporočila aplikacija najprej preveri ključno besedo in če se ta ujema, pošlje določen ukazni niz preko povezave bluetooth. Algoritem je razviden iz grafa na sliki 2.5.



Slika 2.5: Diagram poteka funkcije za sprejem in obdelavo SMS sporočila.

Intervalometer preko sporočil SMS torej upravljamo s ključnimi besedami, ki so razvidne v tabeli 2.1. Ključne besede niso občutljive na male oziroma velike črke. To pomeni, da če v SMS sporočilu napišemo Proži, proži, prOŽi ali kaj podobnega, se bo interpretiralo kot funkcija za proženje fotoaparata.

Kot je razvidno iz 3. stolpca tabele 2.1, se preko bluetooth povezave pošiljajo samo številke. Arduino ploščica na drugi strani razbere iz prve številke, katero funkcijo smo zahtevali preko SMS sporočila in temu primerno izvede zadane ukaze. Del kode za delo s SMS sporočili:

```
def message_received(msg_id):
    box = inbox.Inbox()
    appuifw.app.body = appuifw.Text(u'SMS prejet.')
    sms_text = box.content(msg_id)
    ...
```

V spremenljivki sms_text je shranjena vsebina prejetega sporočila SMS. Nato ga s funkcijo split() razbijemo na podnize, katerih ločilo je presledek. Tako lahko dalje iščemo po ključni besedi.

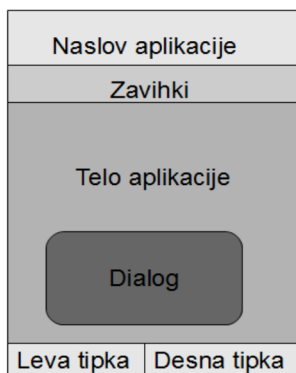
Oblika SMS	Funkcija	Ukaz, poslan preko bluetooth povezave
interval 11 6	intervalno fotografiranje, 11 slik, posamezna slika vsakih 6 sekund	1 11 6
zakasnitev 33	zakasnjeno proženje za 33 sekund	2 33
prozi	proženje fotoaparata	3

Tabela 2.1: Primeri krmiljenja intervalometra s sporočili SMS.

2.3.2 Uporabniški vmesnik

PyS60 je pripravljen tudi za gradnjo uporabniškega vmesnika. Uvoziti je potrebno dve knjižnici, in sicer appuifw ter e32 [5]. Na voljo je precej različic uporabniškega vmesnika. Osnovni gradniki so vidni na sliki 2.6.

Izberemo lahko tri velikosti zaslona za našo aplikacijo: normalen, velik in cel. Odločili smo se za normalnega. Tako so vidni: ime oziroma naslov aplikacije in imeni funkcijskih tipk. Zavihkov ni bilo potrebno uporabiti, saj izbor poteka preko izbirnega seznama. Telo aplikacije je kar tekstovno polje, kamor se izpisuje trenutno stanje oziroma funkcija, ki jo uporabljamo. Tekstovno polje prekriva izbirni seznam, kjer izberemo intervalometer, zakasnitev, proženje ali SMS. Izgled in roko vanje z uporabniškim vmesnikom je podrobneje opisano v poglavju 4. Glavni del programa teče v while zanki, ki jo prekinemo



Slika 2.6: Gradniki uporabniškega vmesnika knjižnice appuifw.

tako, da pritisnemo desno tipko pod zaslonom, kjer piše Cancel. Leva tipka je namenjena potrditvi. Vnos podatkov poteka preko vnosnih polj(slika 2.7a).



Slika 2.7: Primeri gradnikov uporabniškega vmesnika.

Pri funkciji intervalometra sta taki polji dve: za število slik ter vmesno zakasnitev. Primer obvestila uporabniku vidimo na slikah 2.7b in 2.7c. Primer kode za vnosno polje ter obvestilo:

```
interval = appuifw.multi_query(u"Interval [sekund]:", u"Število  
posnetkov:")
appuifw.note(u"Prozenje!", "conf")
```

V spremenljivki interval hranimo dve vrednosti: število posnetkov ter interval. V drugi vrstici zgornjega primera kode pa lahko namesto conf vstavimo tudi info ali error, odvisno od tipa obvestila, ki ga želimo prikazati uporabniku.

Glavni meni je izbirni seznam, lahko pa bi uporabili tudi zavihke. Koda za izbirni seznam:

```
L = [u'Intervalometer', u'Zakasnitev', u'Prozenje', u'SMS']  
index = appuifw.selection_list(choices=L , search_field=0)
```

Ustvariti je potrebno seznam opcij, ki jih želimo ponuditi uporabniku. Vrednost spremenljivke `index` je potrjen izbor, shranjen kot številke 0, 1 ali 2. Spremenljivka `search_field` omogoča tudi iskalno polje, kar pa v tej aplikaciji ni posebej uporabno, zato je postavljena na vrednost 0.

Poglavje 3

Intervalometer

3.1 Vezje

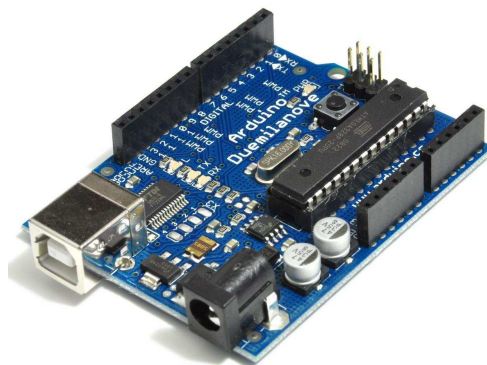
Za krmiljenje intervalometra smo izbrali razvojno ploščico Arduino Duemilanove. Je dovolj zmogljiva in nudi potrebno število vhodov in izhodov. Za interakcijo uporabnika s sistemom je bil izbran dvovrstični LCD zaslon in matrična tipkovnica. Fotoaparata je povezan preko 2.5 mm vtiča, ki je na vezje priklopljen preko optospojnika. Na ohišju intervalometra so tudi tipke za krmiljenje fotoaparata, ki opravljajo funkcijo žičnega prožila.

3.1.1 Arduino

Arduino je zelo razširjena odprtokodna platforma, ki je namenjena širokemu krogu uporabnikov. Primerna je tako za začetnike kot za napredne uporabnike (npr. izdelava prototipov).

Ploščica Arduino Duemilanove [6] (slika 3.1) temelji na mikrokrmilniku ATmega168. Ima 14 digitalnih vhodov oziroma izhodov ter 6 analognih vhodov, ki pa smo jih uporabili kot digitalne vhode. Na ploščici je tudi že USB vtič, preko katerega naložimo program. Tako ne potrebujemo posebnega programatorja. Še ena pomembna lastnost - ploščico lahko direktno napajamo z 9 V baterijo. Tega smo se tudi poslužili, saj je intervalometer samostojna naprava in takšno mora biti tudi napajanje. Programiranje poteka preko posebne razvojne platforme, ki je na voljo na spletni strani <http://www.arduino.cc>. Programira se v prirejenem programskem jeziku, ki je podoben programskemu jeziku C. Arduino skupnost je precej razširjena in aktivna, zato je na voljo veliko knjižnic in vodičev za priklop različnih naprav in senzorjev.

Priklopljeni so bluetooth modul, LCD zaslon ter tipkovnica.



Slika 3.1: Arduino Duemilanove razvojna ploščica.

3.1.2 Tipkovnica

Tipkovnica (slika 3.2) ima 16 tipk, številke od 0 do 9 ter črke od A do D. Tako lahko vpišemo poljubno številko, kar se je izkazalo za boljšo rešitev kot določanje številke s potenciometri. Tipkovnica ima 8 nožic, zato je bilo potrebno z voltmetrom najprej določiti, katere 4 so za vrstice in katere 4 za stolpce. Tudi za matrično tipkovnico je že spisana knjižnica [7], ki smo jo



Slika 3.2: Matrična 4x4 tipkovnica.

morali uvoziti. Potrebno je le pravilno inicializirati nožice za vrstice in stolpce. Koda za inicializacijo tipkovnice:

```
#include <Keypad.h>
const byte ROWS = 4; // 4 vrstice
const byte COLS = 4; // 4 stolpci
char keys[ROWS][COLS] = {
  {'1','2','3','A' },
  {'4','5','6','B' },
```

```

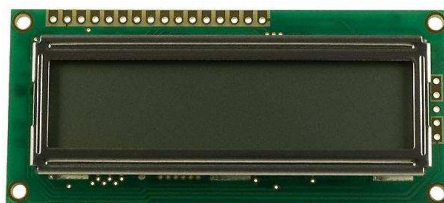
    {'7','8','9','C' },
    {'*','0','#','D'  }};
byte rowPins[ROWS] = { 8, 9, 10, 11 };
byte colPins[COLS] = { 4, 5, 6, 7 };
Keypad kpd = Keypad(makeKeymap(keys), rowPins, colPins, ROWS,
                    COLS);

```

Kot je razvidno iz zgornje kode, so vrstice priklopljene na nožice 8-11, stolpci pa na 4-7. Branje znakov s tipkovnice poteka s funkcijo `keypad.getKey()`.

3.1.3 LCD zaslon

Za intervalometer je bil izbran LCD zaslon (slika 3.3) z dvema vrsticama po 16 znakov. Zgrajen je na krmilniku HD44780, za katerega je na voljo tudi knjižnica za Arduino. Komunikacija z LCD zaslonom poteka preko 8 podatkovnih signalov. Ker pa smo bili na ploščici omejeni z vhodi in izhodi, smo uporabili 4 bitni prenos. To nam omogoča knjižnica `LiquidCrystal` [8]. LCD modul se napaja s 5 V, je tudi osvetljen. Za nastavitve kontrasta za-



Slika 3.3: LCD zaslon.

slona ga povežemo s potenciometrom. Poleg 4 uporabljenih podatkovnih nožic (DB4-DB7 na sliki 3.4) smo povezali tudi nožice signalov RS, RW ter E. S signalom RW krmilnik LCD modula ugotovi, ali mora izpisati znak ali pa gre zgolj za njemu namenjen ukaz. Signal RW se veže kar na maso, saj se na LCD zaslon vedno piše. Signal E pa da LCD kontrolerju vedeti, da je podatek pripravljen za zapis. Signal VO se veže na potenciometer, ki služi nastavitvi kontrasta LCD zaslona. Kot je razvidno iz slike 3.4, smo LCD zaslon povezali kar na analogne vhode/izhode. Zato je bilo potrebno v funkciji `setup()` pri programiranju intervalometra inicializirati nožice na sledeč način:

```

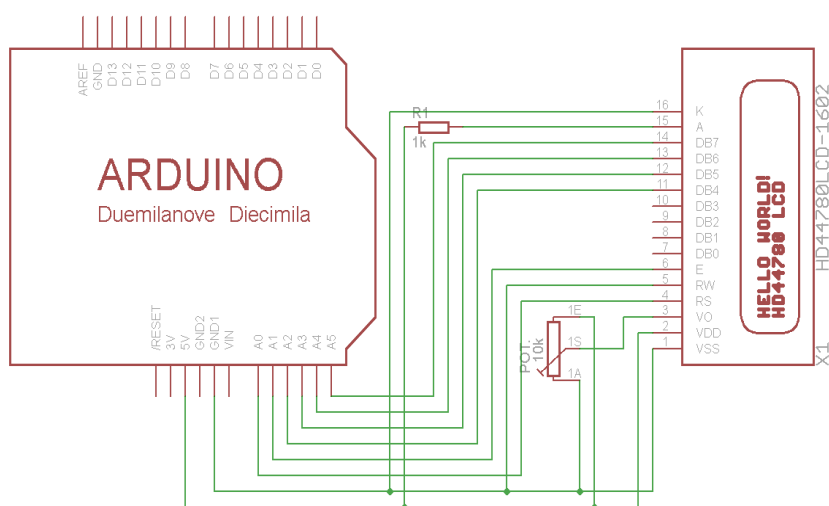
void setup() {
    pinMode(14, OUTPUT);
    pinMode(15, OUTPUT);
}

```

```
pinMode(16, OUTPUT);
pinMode(17, OUTPUT);
pinMode(18, OUTPUT);
pinMode(19, OUTPUT);
...
```

Tako smo lahko analogne vhode/izhode uporabili kot digitalne izhode, saj pri komunikaciji z LCD zaslonom vhodov ne uporabljamo.

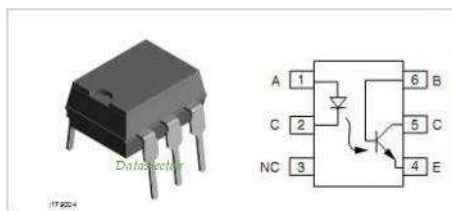
Knjižnica sicer vsebuje tudi ukaze za premik zaslonskega kazalca, določitev vrstice zapisa, brisanje vrstic in pisanje nizov na LCD zaslon.



Slika 3.4: Priklop LCD zaslona na Arduino.

3.1.4 Ostali elementi vezja

Fotoaparat ter intervalometer sta povezana preko 2.5 mm vtiča (slika 3.6) za daljinsko prožilo. To je standardni priključek za več znamk fotoaparatorov, med drugim Pentax in Canon. Za ostale znamke je potrebno uporabiti prirejen vmesnik. Med intervalometer in fotoaparat je vezan optospojnik. Tako se izključi možnost napak in posledično okvare fotoaparata. Optospojnik 4N28, kot je prikazan na sliki 3.5, je v ohišju s 6 nožicami. Iz specifikacij [9] je razvidno, katere nožice moramo priklopiti. Uporabljena sta dva optospojnika, en za autofokus in drug za proženje. Glede na sliko 3.5 in sliko 3.7 je nožica 2 (oziroma 3) s ploščice Arduino priklopljena na nožico 1 optospojnika, masa

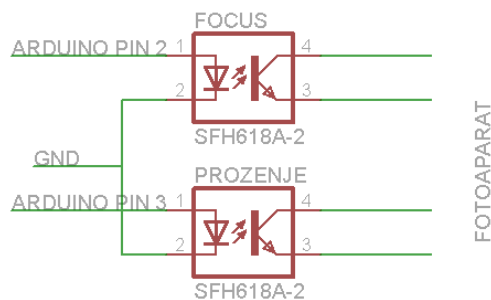


Slika 3.5: Optospojnik 4N28.

pa na nožico 2. Fotoaparāt je povezan z nožico 5 (autofokus ter proženje), nožica 4 pa na maso fotoaparata. Torej sta fotoaparāt in ploščica Arduino tudi galvansko ločena. Nožici 3 in 6 nista povezani. Na sliki 3.7 je shema vezave Arduino ploščice na eni strani in fotoaparata na drugi strani.

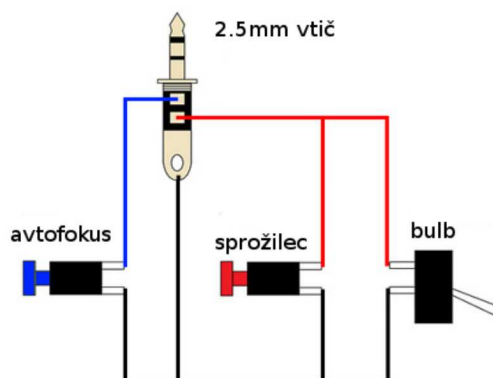


Slika 3.6: 2.5 mm vtič.



Slika 3.7: Vezava optospojnikov.

Da bo intervalometer uporaben tudi kot daljinsko prožilo, so vzporedno vezane tipke na ohišju. Tudi ko je intervalometer izključen, ga še vedno lahko uporabimo kot namensko žično prožilo. Uporabili smo tudi stikalo, ki je namenjeno funkciji bulb. Ta je uporabna za daljše osvetlitvene čase, kjer bi že zaradi manjših tresljajev prišlo do stresenega posnetka. S stikalom povzročimo manj tresljajev kot s konstantnim pritiskanjem tipke. Vezava tipk je vidna na sliki 3.8.



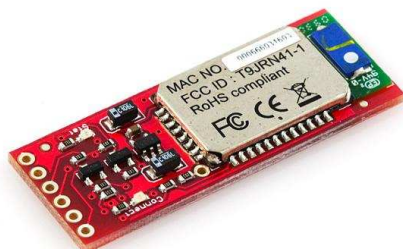
Slika 3.8: Shema vezave tipk na 2.5 mm vtič.

3.2 Bluetooth modul

Uporabljeno bluetooth ploščico izdeluje podjetje Sparkfun pod imenom Bluetooth Mate (slika 3.9). Vsebuje ploščico z nožicami, periferijo in bluetooth modul RN-41 [10]. To je modul razreda 1, kar pomeni, da ima v teoriji na prostem doomet 100 metrov. Proizvajalec tudi zagotavlja neobčutljivost na motnje drugih brezžičnih omrežij. Specifikacije Bluetooth Mate:

- razred 1 bluetooth,
- kriptirana povezava,
- frekvenca delovanja od 2.4 GHz do 2.524 GHz,
- povprečna poraba 25 mA,
- delovna napetost od 3.3 V do 6 V,
- serijska komunikacija od 2400 bps do 115200 bps,
- temperatura delovanja od -40°C do $+70^{\circ}\text{C}$,
- vgrajena antena.

Kot je razvidno iz slike 3.9, ima bluetooth modul 6 zunanjih nožic. Poleg napajanja, ki je v tem primeru 3.3 V, moramo priklopiti še maso, nožici za oddajanje oziroma sprejem (5 V) ter CTS in RTS. CTS (angl. Clear To Send) in RTS (angl. Request To Send) nožici povežemo skupaj in s tem onemogočimo



Slika 3.9: Bluetooth modul Bluetooth Mate.

strojno kontrolo pretoka. Nožici RX ter TX sta povezani z nožicama 0 in 1 na ploščici Arduino, saj sta rezervirani za serijsko komunikacijo. Uporabili bi lahko tudi katerikoli drugi dve nožici, pri tem pa bi si morali pomagati s knjižnico `SoftwareSerial`.

Bluetooth modulu lahko tudi določamo parametre [11], kot so na primer ime modula, PIN koda, hitrost prenosa in drugi. V prvih 60 sekundah po priklopu naprave lahko pridemo v tako imenovano programabilno delovanje. V ta način delovanja pridemo s poslanim nizom \$\$\$\$. Nato z ukazi [11] določimo način gospodar/suženj, hitrost prenosa, ime modula in ostale parametre.

V mobilnem telefonu moramo aktivirati bluetooth povezavo. To storimo v nastavitvah povezljivosti. Ko dodamo novo napravo, bo mobilni telefon preiskal bližnje bluetooth naprave in izpisal imena. Naši napravi smo pustili privzeto ime, to je „FireFly-1355“. Vpišemo še kodo za vzpostavitev povezave, ki je tovarniško prednastavljena na 1234.

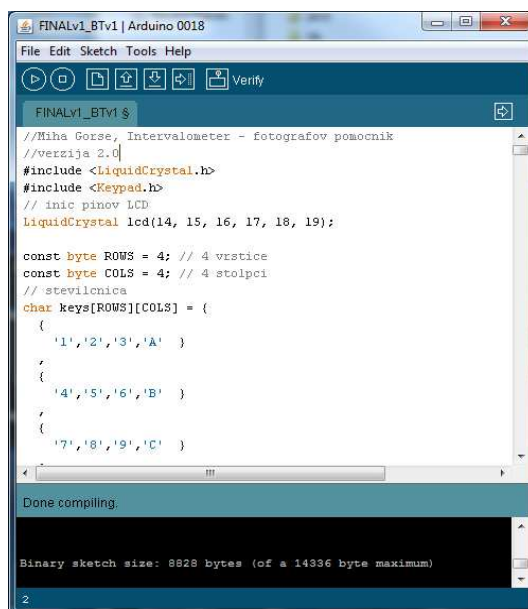
Povezovanje lahko spremljamo tudi z vgrajeno LED diodo. Ko le-ta hitro utripa rdeče, pomeni, da lahko pridemo v programabilno delovanje. Ob počasnejšem utripanju rdeče LED diode modul čaka na povezavo. Ko zasveti in gori zelena LED dioda, pomeni, da je povezava uspešno vzpostavljena.

3.3 Program

Arduino ploščico programiramo s posebnim razvojnim orodjem. Omogoča nam uvoz knjižnic, prevajanje in nalaganje kode. Ima tudi vgrajen terminal za serijsko komunikacijo. Na sliki 3.10 je zaslonski posnetek glavnega okna

razvojnega orodja. Koda se tudi obarva, da je lažje berljiva. Pri razvoju smo uporabljali verzijo 0018.

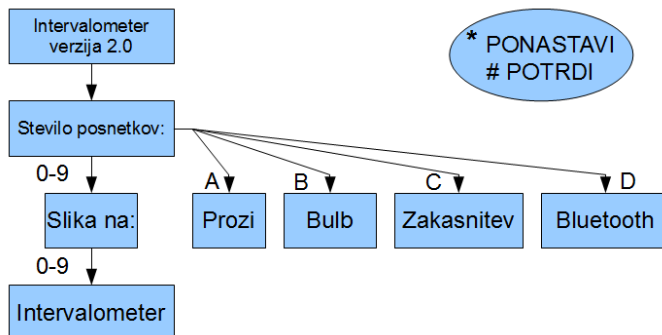
Najprej se uvozi potrebne knjižnice. To storimo z ukazom `import` in imenom knjižnice. Za ta intervalometer smo potrebovali knjižnici za LCD zaslon ter tipkovnico. Kot je že bilo omenjeno, je potrebno tudi inicializirati tipkovnico in zaslon LCD, torej določiti, na katere nožice sta napravi priključeni. Nato je potrebno v funkciji `setup()` določiti nožice kot vhodne oziroma izhodne. Napovemo tudi serijsko komunikacijo s hitrostjo prenosa 115200 bps. Na LCD zaslon izpišemo verzijo programske opreme intervalometra. Funkcija `setup()` se vedno zažene ob vklopu intervalometra.



Slika 3.10: Zasloni posnetek razvojnega okolja za Arduino.

Sledi inicializacija potrebnih spremenljivk za razne števec, znake in pomožne spremenljivke. Glavni del programa se izvaja v funkciji `loop()`. Kot že ime pove, se ta funkcija izvaja v neskončni zanki. Na to smo morali biti pozorni tudi pri načrtovanju izbirnega menija intervalometra. Tu smo si pomagali z različnimi pogoji in zastavicami.

Na sliki 3.11 je diagram izbirnega menija. Po začetnem izpisu verzije programa se na zaslonu izpiše navodilo za vpis števila posnetkov. To pomeni, da deluje intervalometer privzeto na funkciji za intervalno fotografiranje. Če vpišemo številke, ki so število posnetkov, gremo s potrditveno tipko `#` v naslednji nivo, kjer zopet vpišemo številke, tokrat za zakasnitev. Če pa namesto



Slika 3.11: Diagram izbirnega menija v intervalometru.

številke pritisnemo katerokoli izmed tipk A-D, se izvede iz diagrama razvidna funkcija. Tipka * služi kot ponastavitev vrednosti in nas vrne na prvi nivo, torej k vpisu števila posnetkov.

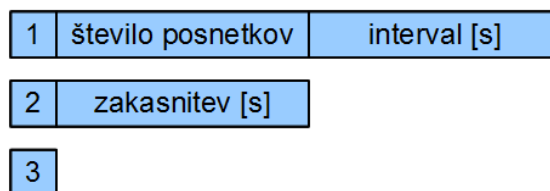
Za branje ukaza preko bluetooth modula je bila napisana funkcija `readSerialString`, ki zaporedno prebrane posamezne znake sestavi v niz znakov. Koda funkcije `readSerialString()`:

```

void readSerialString () {
    memset(serInString, 0, sizeof(char) * 100);
    int sb;
    if(Serial.available()) {
        kk=1;
        while (Serial.available()){
            sb = Serial.read();
            serInString[serInIndx] = sb;
            serInIndx++;
        }
    }
}

```

Na ta način dobimo preko bluetootha poslan ukaz. Tak niz nato razdelimo glede na parametre. Prvi znak je številka 1, 2 ali 3. Če gre za številko 1, je bil poslan zahtevek za interval. Številka 2 pomeni, da gre za zakasnitev, številka 3 pa, da gre za takojšnje proženje fotoaparata. V kolikor je bila prva številka 1, bosta naslednji dve številki pomenili število posnetkov in zakasnitev med njimi. Pri funkciji zakasnitve (številko 2) pa sledi le število sekund, za kolikor zakasnimo proženje fotoaparata. Formati ukazov so predstavljeni tudi na sliki 3.12.



Slika 3.12: Format ukazov, prejetih preko bluetootha.

Seveda je potrebno najprej vklopiti bluetooth s tipko D. Na zaslonu se izpiše bluetooth, kar pomeni, da intervalometer pričakuje ukaze preko bluetootha. Izbirni meni je realiziran s switch stavki, ki imajo pogoj nastavljen glede na pritisnjeno tipko. Vnos večmestnih števil je realiziran z while zanko, ki ima za izhodni pogoj tipko #, kar je tudi neke vrste potrditev. Vrednosti resetiramo in se vrnemo na prvi nivo s tipko *. Če vpišemo preveliko vrednost, se na zaslonu izpiše obvestilo o napaki. Ko vnose potrdimo, lahko na zaslonu spremljamo preostanek posnetkov ter odštevanje sekund. Proženje realiziramo tako, da najprej postavimo nožico, ki je vezana na optospojnik s signalom za autofokus, na logično vrednost 1. Preden na podoben način aktiviramo še nožico za proženje, dodamo zakasnitev 1 sekundo, saj mehanizem za autofokus potrebuje čas za izostritev. Po končanem proženju se na zaslonu izpiše „Uspeh!“. Tudi zakasnitev proženja je bila realizirana z ukazom `delay(1000)`, ki zakasni proženje za 1000 milisekund. Tako torej zakasnimo za želeno število sekund. Podobno je pri intervalu, kjer na enak način odštevamo sekunde med posnetki.

Pri funkciji bulb velja opozoriti na predolge osvetlitvene čase. Že na zaslonu se izpiše obvestilo „max 60 sekund“. Dolgi osvetlitveni časi namreč močno obremenijo senzor fotoaparata, ki posledično lahko odpove. Zato moramo biti pri vnosu daljših časov pazljivi.

Na koncu zapišimo še krajši povzetek implementiranih funkcij:

- *zacetek()* ponastavi vrednosti vseh spremenljivk in postavi sistem v začetno stanje,
- *konec_menu1()* zaključi prvi izbirni meni, kjer smo podali želeno število posnetkov. Izpiše število vnesenih posnetkov in kliče funkcijo za prehod v naslednji nivo izbirnega menija,
- *zacetek_menu2()* na LCD zaslon izpiše izbrani interval med posnetki,
- *konec_menu2()* izpiše vnešene vrednosti in izvede proženje fotoaparata glede na izbran interval in število posnetkov,

- *prozi()* aktivira nožici za autofokus in proženje fotoaparata,
- *konec_zakasnitev()* izvede zakasnitev proženja, ki smo jo vnesli po izbiri funkcije zakasnitve s tipko C,
- *konec_bulb()* izvrši funkcijo bulb,
- *izberi_konec(int)* potrdi konec vnosa in prehod na nov nivo izbirnega menija,
- *brisi_vrednosti()* vse vnešene vrednosti postavi na 0,
- *readSerialString()* prebere znake, prejete preko bluetootha in jih sestavi v niz,
- *razdeli()* razdeli prejeti niz glede na parametre,
- *bluetooth_zakasnitev()* izvede funkcijo zakasnitve glede na parametre, prejete preko povezave bluetooth.

Poglavje 4

Upravljanje intervalometra

4.1 Mobilni telefon

Kot že omenjeno, mobilni telefon služi brezžičnemu daljinskemu upravljanju intervalometra. Na mobilni telefon smo namestili našo aplikacijo Intervalometer (slika 4.1a). Poleg tega je pogoj za delovanje aplikacije prednameščen PyS60. Ko prenesemo in namestimo SIS aplikacijo, jo poiščemo v meniju in zaženemo. Pred tem mora biti intervalometer vklopljen in izbran bluetooth s tipko D. Na zaslonu intervalometra se izpiše bluetooth. Ko zaženemo aplikacijo na mobilnem telefonu, se vzpostavi povezava z intervalometrom (slika 4.1b).



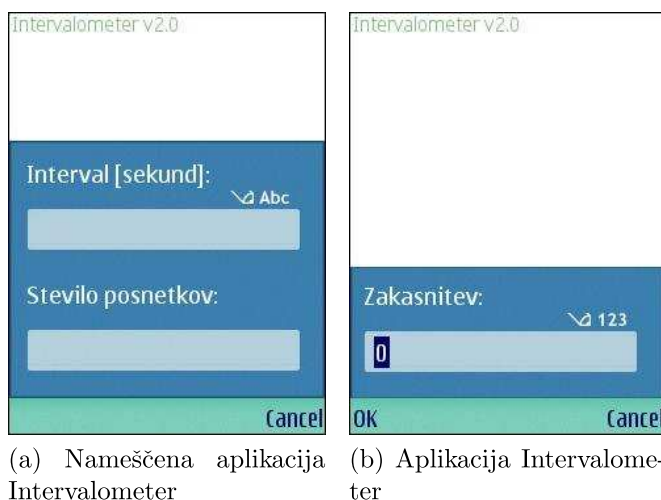
(a) Nameščena aplikacija Intervalometer

(b) Povezovanje z intervalometrom preko bluetootha

(c) Aplikacija Intervalometer

Slika 4.1: Aplikacija Intervalometer na mobilnem telefonu.

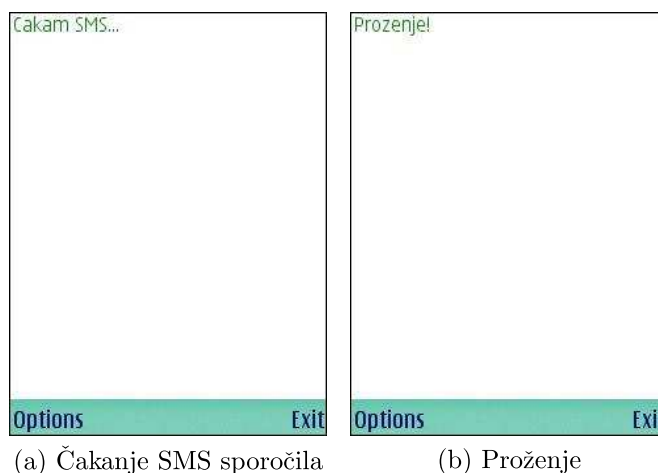
Ker je bil MAC naslov bluetooth modula vpisan že v izvorno kodo, je aplikacija vezana na določen bluetooth modul. Če povezava ne uspe, se izpiše obvestilo o napaki in aplikacija se zapre. Po uspešni povezavi naše aplikacije se prikaže izbirni meni, kot je razvidno iz zaslonskega posnetka aplikacije na sliki 4.1c. Izberemo lahko funkcijo intervalometra, zakasnitve, proženja in upravljanja preko SMS sporočil. Če izberemo funkcijo intervalometra, moramo



Slika 4.2: Zaslonska posnetka vnosa parametrov.

v naslednjem koraku vpisati parametre (slika 4.2a). Seveda moramo vpisati številke, čeprav ima sistem privzet vnos črk. To pomeni, da vnašamo črke in moramo za vnos številke večkrat pritisniti določeno tipko. Do številke lahko pridemo tudi tako, da tipko dlje časa držimo. Na to nastavitve žal nimamo vpliva. Pri posameznem vnosnem polju (slika 4.2b) pa lahko določimo ali gre za črke ali številke. Če vnesemo črke, nas sistem opozori o napaki. Po potrjenem vnosu nas sistem še enkrat povpraša o poslanih vrednostih, nato pa jih preko bluetooth povezave pošlje intervalometru v že omenjeni obliki.

Za uporabo zakasnitve moramo vnesti želeno število sekund, kot je razvidno iz slike 4.2b. Zakasnitev še enkrat potrdimo. V kolikor izberemo proženje, bo ukaz takoj poslan intervalometru in ta bo sprožil fotoaparata. Intervalometer lahko krmilimo tudi preko sporočil SMS s ključnimi besedami iz tabele 2.1. Med čakanjem na sporočilo SMS je to tudi izpisano na zaslonu, kot je razvidno iz zaslonskega posnetka na sliki 4.3a. Takojšnje proženje je razvidno iz slike 4.3b. Aplikacijo zapremo s pritiskom na desno funkcijsko tipko Cancel.



Slika 4.3: Zaslonska posnetka čakanja sporočila SMS in proženja.

4.2 Intervalometer

Kot že omenjeno, lahko intervalometer upravljamo tudi preko tipkovnice in LCD zaslona. Najprej intervalometer priklopimo na vhod za žično prožilo na fotoaparatu. Za nekatere modele zrcalnorefleksnih fotoaparátov je potreben vmesnik. Po vklopu intervalometra se na zaslonu izpiše trenutna verzija naloženega programa, nato pa že število slik. To pomeni, da smo trenutno v funkciji intervalnega fotografiranja, kar je tudi privzeta funkcija same naprave. Če vnesemo število in potrdimo vnos s tipko #, moramo v naslednjem koraku vnesti število sekund med posameznimi posnetki. Vnos zopet potrdimo s tipko #. Nato se intervalno fotografiranje prične. Kadarkoli lahko vrednosti ponastavimo s tipko *. Če želimo uporabiti katero od drugih funkcij, moramo izbrati tipke od A do D. Možne izbire so razvidne tudi iz slike 3.11.

Torej se ob pritisku tipke A takoj izvrši proženje. Upoštevati moramo zakasnitev autofokusa, to je približno 1 sekunda. Priporočljivo pa je na fotoaparatu nastaviti kar ročno ostrenje, tako bo v intervalnih posnetkih fokus vedno na istem mestu.

S tipko B izberemo funkcijo bulb. Ta je primerna za daljše osvetlitvene čase, predvsem za nočne posnetke in posnetke nočnega neba. Zaradi morebitnega pregrevanja senzorja in posledične odpovedi se priporoča uporaba časov, krajših od 1 minute. Za natančnejše podatke se je priporočljivo obrniti na proizvajalca in upoštevati njegova navodila za uporabo. Pri tej funkciji je dejansko nemogoče narediti nestresene posnetke brez žičnega prožila.

V fotoaparatu sta ponavadi le dve zakasnitvi proženja, 2 oziroma 10 sekund.



Slika 4.4: Intervalometer in stranske tipke.

Ta čas se razlikuje od proizvajalca do proizvajalca. Včasih pa bi želeli proženje zakasniti za poljuben čas. Tako lahko izberemo funkcijo zakasnitve s pritiskom na tipko C. Na zaslonu lahko spremljamo števec preostalega časa do proženja. Vnos zopet potrdimo s tipko #. Za vrnitev v izbirni meni pritisnemo tipko *.

Zadnja funkcija je upravljanje preko bluetooth povezave. Izberemo jo s tipko D. Intervalometer bo začel prejemati ukaze, poslane iz mobilnega telefona, ki ima nameščeno aplikacijo Intervalometer. Ukaz, prejet preko povezave bluetooth, lahko tudi spremljamo na zaslonu. Bluetooth komunikacijo prekinemo s ponovnim pritiskom na tipko D, na zaslonu pa se izpiše informacija o stanju bluetooth povezave.

Intervalometer lahko uporabljamo tudi kot žično prožilo. V tem primeru ni potrebno, da je naprava vklopljena. Na levi strani naprave so tipke (slika 4.4), s katerimi vklopimo autofokus oziroma prožimo fotoaparata. S pritiskom na modro tipko vklopimo autofokus. Ko v koku fotoaparata vidimo potrditev fokusa, oziroma slišimo potrditveni zvok autofokusa, lahko fotoaparata prožimo s pritiskom na rdečo tipko. Poleg tipk smo vgradili tudi prekucno stikalo. To nam služi namesto držanja rdeče tipke pri funkciji bulb. Torej lahko namesto držanja rdeče tipke vklopimo stikalo. Kolikor časa je stikalo v aktivnem stanju, tolikšen bo osvetlitveni čas.

Za morebitne napake pri vnosu števil je poskrbljeno že v samem programu. Če je uporabnik vnesel preveliko število posnetkov oziroma prevelik čas med posnetki, naprava izpiše obvestilo o napaki. Za kakršne koli napake, ki bi se pojavile, je vgrajena rumena tipka na levi strani naprave, ki intervalometer ponovno zažene in ponastavi spremenljivke.

Poglavje 5

Sklepne ugotovitve

Gradnje intervalometra je bila zanimiva predvsem zaradi demonstracije komunikacije in upravljanja mikrokrmilnika z mobilnim telefonom preko brezžične povezave bluetooth.

Spoznal sem gradnjo aplikacij za sistem Symbian in programske jezike, ki nam to omogočajo. Odločil sem se za Python in s tem pokazal, da je res namenjen hitri gradnji aplikacij in izdelavi prototipov. Kljub temu omogoča veliko prilagodljivosti, tudi uporabniškega vmesnika. Kar ga loči od izvirnega jezika za sistem Symbian, je predvsem pogoj, da mora biti prednameščen PyS60. Aplikacija deluje stabilno, prav tako povezava bluetooth. Tukaj se je odprlo veliko novih idej in možnosti, saj imajo novejši mobilni telefoni veliko že vgrajenih senzorjev in naprav. Tako bi na primer lahko koristili vgrajene senzorje premika v vseh 3 smereh, digitalni kompas in celo podatke GPS za določanje lokacije. Vse te podatke bi lahko sprocesirali že na samem telefonu, saj procesorji v novejših modelih presegajo 1 GHz, prav tako imajo vgrajeno večjo količino delovnega pomnilnika. Takšne podatke bi lahko pošiljali preko povezave bluetooth podobni ploščici, kot je Arduino. Upravljanje malih motorjev, zaslonov, senzorjev in podobne periferije ob komunikaciji z zmogljivostmi današnjih mobilnih telefonov odpira nešteto novih možnosti. Poleg tega se lahko ob povezavi z mobilnim telefonom izkoristijo tudi vse funkcije telefona, kot je pošiljanje sporočil SMS in klicanje. Ne smemo pozabiti, da imajo vsi današnji telefoni tega razreda vgrajeno tudi kvalitetno kamero.

Na svojo napravo - intervalometer, sem torej priklopil zrcalnorefleksni fotoaparatus. Poskrbeti sem moral za varnost priklopljenega fotoaparata, saj ploščica Arduino deluje na 5 V napetosti, ki bi za fotoaparatus lahko bila usodna. Za ločitev sem poskrbel z optospojnikom. Uporabil bi lahko tudi releje. Tako je naprava, katere vrednost je nekajkrat višja od same ploščice Arduino in osta-

lih komponent, varna. Z malo večjim izzivom sem se srečal tudi pri izdelavi menija za intervalometer. Upoštevati sem moral, da le-ta deluje v neskončni zanki. Temu sem se moral prilagoditi in uporabiti več zastavic za spremljanje trenutnega stanja. Samo izvajanje zakasnitve in proženje je bilo relativno enostavno, saj sem uporabil kar vgrajeno funkcijo delay. Za priklop bluetooth modula sem uporabil dobro dokumentacijo proizvajalca. Lahko bi izbral tudi ploščico ArduinoBT, ki ima že vgrajen bluetooth modul.

Tak koncept povezave in upravljanja bi lahko uporabili tudi namesto namenskega GSM modula. Danes mobilni telefoni hitro zastarijo in izgubijo vrednost. Za prototip izdelka z GSM modulom je bilo praktično ceneje uporabiti povezavo bluetooth in starejši mobilni telefon, kot pa kupiti namenski GSM modul, ki ne omogoča toliko funkcij, kot pametni mobilni telefon. Vsekakor je le-ta močno orodje, ki ima za povrh še vgrajeno baterijo. Večino današnjih naprav, ki jih upravljamo preko omrežja GSM, bi lahko realizirali kot naš intervalometer. Tudi v pametnem domu, kjer preko GSM upravljamo z lučmi, pečjo, alarmom in ostalimi napravami. V bistvu niti ni pomembno, da je mobilni telefon Nokia s sistemom Symbian, saj lahko programe pišemo za poljubne mobilne platforme. Pomembno je le, da ima mobilni telefon povezavo bluetooth. To pa najdemo tudi v mobilnih telefonih nižjega cenovnega razreda.

Znanje, ki sem ga pridobil pri izdelavi tega diplomskega dela, bom s pridom izkoristil pri naslednjih projektih, ki bodo vsekakor temeljili na povezavi bluetooth, mobilnem telefonu in neodvisni napravi, ki jo bom lahko upravljal.

Dodatek A

Priloge

A.1 Slike naprave



Slika A.1: Intervalometer (sprednja stran).



Slika A.2: Intervalometer (leva stran).



Slika A.3: Intervalometer (desna stran).



Slika A.4: Intervalometer in Nokia E65.



Slika A.5: Intervalometer in aplikacija na mobilnem telefonu.

Dodatek B

Priložen CD

- Izvorna koda Python programa za mobilni telefon
- SIS aplikacija za mobilni telefon
- Izvorna koda za Arduino ploščico
- Predstavitveni video posnetek intervalometra s primeri uporabe

Slike

2.1	Mobilni telefon Nokia E65	5
2.2	Zaslonski posnetek emulatorja za sistem Symbian	7
2.3	Zaslonski posnetek orodja za pretvorbo v format SIS	7
2.4	Zaslonski posnetek orodja za pretvorbo ikon	8
2.5	Diagram poteka funkcije sprejema SMS sporočila	10
2.6	Gradniki uporabniškega vmesnika knjižnice appuifw	12
2.7	Primeri gradnikov uporabniškega vmesnika	12
3.1	Arduino Duemilanove razvojna ploščica	15
3.2	Matrična 4x4 tipkovnica	15
3.3	LCD zaslon	16
3.4	Shema vezja za priklop LCD zaslona na Arduino	17
3.5	Optospojnik 4N28	18
3.6	Primer 2.5 mm vtiča	18
3.7	Shema vezave dveh optospojnikov	18
3.8	Shema vezave tipk na 2.5 mm vtič	19
3.9	Uporabljen bluetooth modul	20
3.10	Zaslonski posnetek razvojnega okolja za Arduino	21
3.11	Diagram izbirnega menija v intervalometru	22
3.12	Format ukazov, prejetih preko bluetootha	23
4.1	Aplikacija Intervalometer na mobilnem telefonu	25
4.2	Zaslonska posnetka vnosa parametrov	26
4.3	Zaslonska posnetka čakanja sporočila SMS in proženja	27
4.4	Intervalometer in stranske tipke	28
A.1	Intervalometer (sprednja stran)	31
A.2	Intervalometer (leva stran)	32
A.3	Intervalometer (desna stran)	33
A.4	Intervalometer in Nokia E65	34

A.5 Intervalometer in aplikacija na mobilnem telefonu	35
---	----

Literatura

- [1] (2010) Symbian OS, wikipedia. Dostopno na:
http://en.wikipedia.org/wiki/Symbian_OS
- [2] (2010) Python technical overview. Dostopno na:
http://developer.symbian.org/wiki/index.php/Python_Technical_Overview
- [3] (2010) Python downloads. Dostopno na:
http://developer.symbian.org/wiki/index.php/Python_Downloads
- [4] (2010) Python quick start. Dostopno na:
http://developer.symbian.org/wiki/index.php/Python_Quick_Start
- [5] (2010) Python for Series 60 Platform API reference. Dostopno na:
http://www.mobilenin.com/pys60/resources/API_Reference_for_Python.pdf
- [6] (2010) Arduino board Duemilanove. Dostopno na:
<http://arduino.cc/en/Main/ArduinoBoardDuemilanove>
- [7] (2010) Keypad Library. Dostopno na:
<http://www.arduino.cc/playground/Code/Keypad#Download>
- [8] (2010) Liquid Crystal. Dostopno na:
<http://www.arduino.cc/en/Reference/LiquidCrystal>
- [9] (2010) Specifikacije za optospojnik 4N28. Dostopno na:
<http://www.datasheetcatalog.org/datasheet/fairchild/4N28.pdf>
- [10] (2010) Specifikacije za bluetooth modul RN-41. Dostopno na:
<http://www.rovingnetworks.com/documents/RN-41.pdf>
- [11] (2010) Specifikacije za nastavitev parametrov bluetooth modula RN-41.
Dostopno na:
<http://www.sparkfun.com/datasheets/Wireless/Bluetooth/rn-bluetooth-um.pdf>