

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Andrej Veber

**Nadzor in avtomatizacija funkcij v
sobi**

DIPLOMSKO DELO NA UNIVERZITETNEM ŠTUDIJU

MENTOR: prof. dr. Dušan Kodek

Ljubljana, 2013

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

Besedilo je oblikovano z urejevalnikom besedil \LaTeX .



Št. naloge: 01879/2012

Datum: 05.11.2012

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

Kandidat: **ANDREJ VEBER**

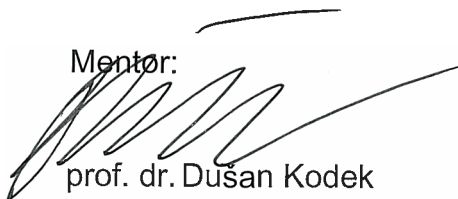
Naslov: **NADZOR IN AVTOMATIZACIJA FUNKCIJ V SOBI**
MONITORING AND AUTOMATION OF ROOM FUNCTIONS

Vrsta naloge: Diplomsko delo univerzitetnega študija

Tematika naloge:

Nadzor in avtomatizacija funkcij v sobi pomenita rešitve, ki naredijo življenjski prostor v sobi bolj lagoden in varen. Te rešitve se dosežejo s senzorji in aktuatorji, ki bodisi avtomatsko ali preko daljinskega nadzora nadzorujejo in spreminjajo stanje v sobi. V ta namen preučite obstoječe rešitve in nato zasnujte sistem za realizacijo nadzora in avtomatizacijo. Izberite ustrezne senzorsko-aktuatorske komponente in primeren mikrokrmilnik z mrežnim vmesnikom. Razvijte in izdelajte vso potrebno strojno in programsko opremo, ki naj vsebuje tudi ustrezno spletno aplikacijo. Pravilnost delovanja preverite in podajte možne načine za izboljšave.

Mentor:



prof. dr. Dušan Kodek

Dekan:



prof. dr. Nikolaj Zimic



IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani Andrej Veber, z vpisno številko **63030075**, sem avtor diplomskega dela z naslovom:

Nadzor in avtomatizacija funkcij v sobi

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 18. marca 2013

Podpis avtorja:

Kazalo

Povzetek	1
Abstract	3
1 Uvod	5
2 Strojna oprema	7
2.1 Arduino Uno	7
2.2 Arduino Ethernet Shield	8
2.3 Rele	9
2.4 Stikalo	10
3 Senzorji	11
3.1 Svetlobni senzor	11
3.2 Temperaturni senzor	12
3.3 Senzor vstopa v prostor	14
3.4 Senzor gibanja v prostoru	16
3.5 Vezava celotnega sistema	19
4 Programska oprema	21
4.1 Razvojno okolje	21
4.2 Program	23
4.3 Upravljanje s krmilnikom	25

KAZALO

5 Spletna aplikacija	27
5.1 Pridobivanje podatkov	28
5.2 Prikaz podatkov	28
6 Sklepne ugotovitve	31
Slike	33
Literatura	35

Seznam uporabljenih kratic

USB	Universal Serial Bus
SRAM	Static Random-Access Memory
EEPROM	Electrically Erasable Programmable Read-Only Memory
IP	Internet Protocol
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
SD	Secure Digital
MDI	Medium Dependent Interface
MDIX	Medium Dependent Interface crossover
SPI	Serial Peripheral Interface
MAC	Media Access Control
NTC	Negative Temperature Coefficient
CD	Compact Disc
IR	Infrared
PIR	Passive Infrared

Povzetek

V tem diplomskem delu je prikazana avtomatizacija prižiganja in ugašanja luči v sobi ter merjenje temperature zunaj prostora in v prostoru. Prižiganje in ugašanje luči se kontrolira s svetlostjo prostora in prisotnostjo osebe v prostoru. Delovanje sistema se lahko preverja in nastavlja preko spleta.

Uporabljena je bila razvojna ploščica Arduino Uno z dodano ploščico Arduino Ethernet Shield. Gibanje v prostoru se zaznava s pasivnim infrardečim senzorjem. Odhod iz prostora se zazna kot prekinitve signala iz infrardeče svetleče diode. Temperaturo se meri s termistorji z negativnim temperaturnim koeficientom povezanimi na analogne vhode.

Na sistem je preko mrežne povezave priklopljen računalnik, ki pobira in hrani podatke. Iz podatkov spletna aplikacija naredi grafe temperature, svetlosti prostora in uporabe luči. Na spletni strani se lahko vidi tudi najvišje, najnižje in povprečne temperature za izbrano obdobje. Možno je spremljati tudi vstop in gibanje v prostoru.

Z ugašanjem luči, ko nas ni v prostoru, privarčujemo pri elektriki. S takim sistemom se lahko spremlja tudi temperaturo in s tem ogrevanje prostora. Tako se lahko analizira porabo energije. Prihranki zaradi tega so pri uporabi samo v eni sobi majhni, če sistem razširimo na celotno hišo so lahko prihranki večji. Na krmilnik se lahko priklopi tudi druge vrste senzorjev in izpopolni program.

Ključne besede:

Arduino, avtomatizacija, gibanje, senzor, splet

Abstract

This thesis describes the automation of turning lights on and off in the room and temperature measurement outside and inside the room. Light in the room is controlled by brightness and the presence of people in the room. Operation of the system can be checked and adjusted via the internet.

The development board Arduino Uno with added module Arduino Ethernet Shield was used. Movement is detected by a passive infrared sensor. Departure from the room is detected as interruption of signal from the infrared light emitting diode. The temperature is measured with a negative temperature coefficient thermistor connected to the analog input.

The system is connected to the computer through a network connection. Computer collects and stores data. From the shared data the web application shows graphs of temperature, brightness and light use. On the website the highest, lowest and average temperature for a selected period can be observed. It is possible to monitor the entry and movement in the room.

By turning off the lights when the room is not used, electricity is saved. This system can monitor temperature and thus heating. This way the energy consumption can be analyzed. When using only one room the savings due to this are small. If the system is extended to the entire house, the savings can be greater. Other types of sensors can be connected to the controller.

Keywords:

Arduino, automation, motion, sensor, web

Poglavje 1

Uvod

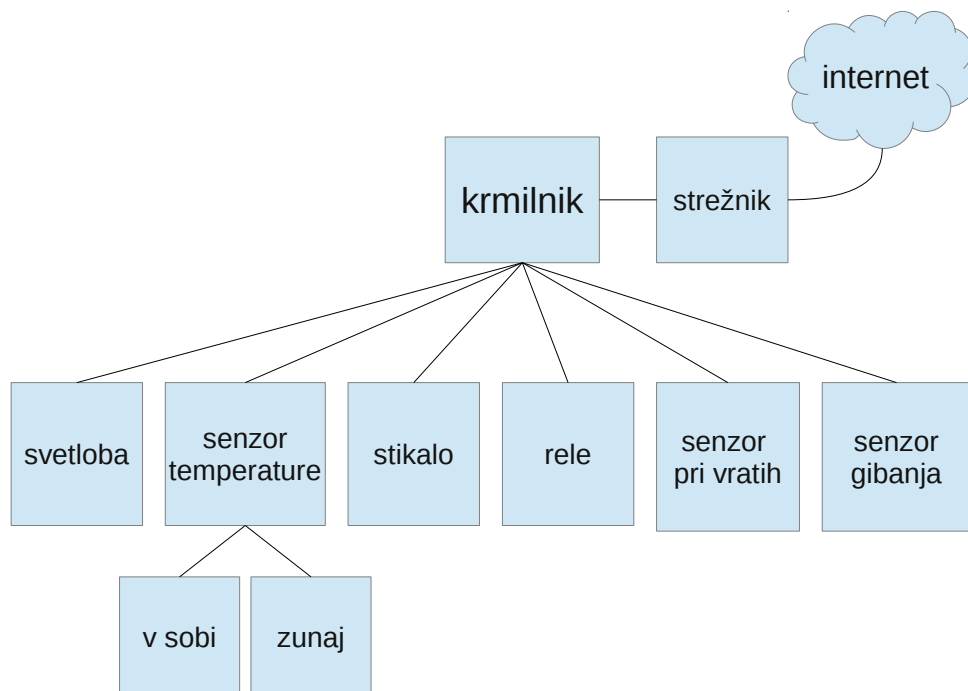
Svetlost zunanje svetlobe se čez dan spreminja. Veliko časa preživimo v prostoru, pri delu pa potrebujemo primerno svetlobo. Ko delamo skrbimo za prižiganje in ugašanje luči v sobi. Če imamo luči vseskozi prižgane se hitro pokvarijo in porabimo veliko elektrike. Želeli pa bi, da se luč prižiga in ugaša sama, takrat ko to potrebujemo.

Pogosto se uporabljajo rešitve, ki prižigajo luči z zaznavo gibanja. To vedno ne zadostuje, pri delu z računalnikom ali branju knjig se ne premikamo dovolj, da bi nas tak sistem zaznal. Če določimo časovno obdobje prižiga, nam luč gori tudi ko nas ni v prostoru. Lahko bi zaznavali vstop v sobo, a pri tem ne vemo koliko ljudi je vstopilo in potem izstopilo.

Za prižiganje in ugašanje luči se lahko uporablja senzor svetlosti prostora z dodatno kontrolo. Poleg premajhne svetlobe v prostoru je za prižiganje luči potrebno zaznati tudi gibanje. Izstop iz prostora se lahko zazna in spremeni stanje na krmilniku. Ko delamo na računalniku se lahko uporablja drugačna kontrola krmilnika. Sam zagon ali ugašanje računalnika ali ohranjevalnika zaslona lahko pošlje ukaz za spremembo stanja na krmilniku.

Sistem (slika 1.1) je sestavljen iz razvojne ploščice Arduino, uporablja se tudi mrežni vmesnik. Na sistem so priklopljeni rele, senzor svetlosti, senzor gibanja, senzor vstopa v prostor, stikalo in senzorji temperature.

Preko spleta je enostavno spremljati podatke in upravljati z napravo.



Slika 1.1: Diagram sistema

Preberemo lahko vrednosti senzorjev in prilagajamo kontrolo. Preko mrežne povezave se lahko priklopi program, ki nabira podatke in jih shranjuje v bazo. Kasneje lahko te podatke uporabimo za analizo in prikaz grafov.

Na ploščico se lahko enostavno priklopi tudi druge senzorje. Spremljamo lahko tudi temperaturo. Prikazana je uporaba senzorjev temperature v sobi in zunaj.

Poglavje 2

Strojna oprema

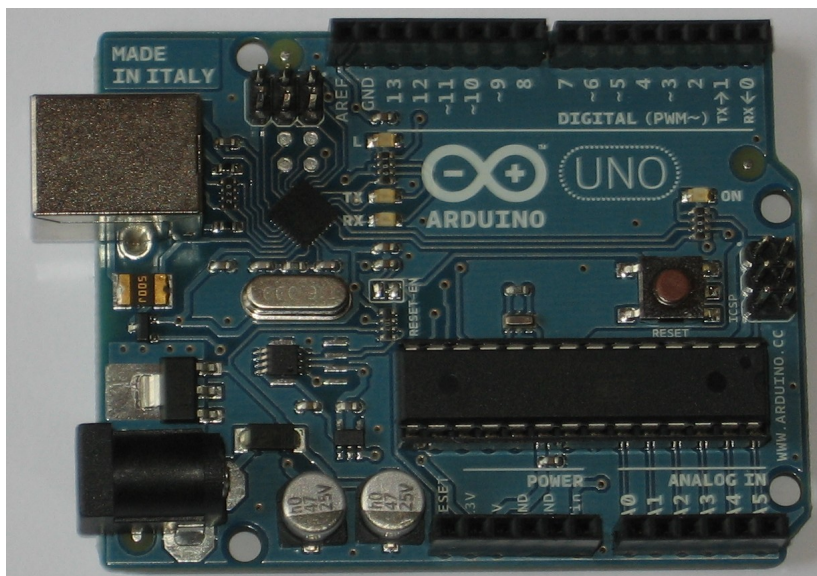
Krmiljenje prižiganja in ugašanja luči in merjenje temperature se izvaja na razvojni ploščici Arduino Uno. Na ploščico je priklopljen še vmesnik za dostop do omrežja Arduino Ethernet Shield.

2.1 Arduino Uno

Arduino Uno (slika 2.1) je razvojna ploščica z mikrokrmilnikom ATmega328. Vsebuje 14 digitalnih in 6 analognih vhodov, ki delujejo tudi kot izhodi. Na ploščici je USB vtič, ki omogoča priklop na računalnik. Nanj je priklopljen krmilnik *ATmega8U2*, ki pretvarja USB v serijski signal povezan z glavnim krmilnikom. Za programe je na voljo 32 KB pomnilnika flash, 2 KB SRAM in 1 KB EEPROM. Več v [1].

Ploščica potrebuje napajanje iz pretvornika ali baterije. Napajamo jo lahko preko povezave USB ali drugega reguliranega vira 5 V. Na ploščici je tudi pretvornik napetosti, ki omogoča napetost od minimalno 6 V do maksimalno 20 V. Priporočena je 7 V do 12 V. Pri višjih napetostih in porabi komponent priklopljenih na ploščico se lahko pretvornik napetosti zelo ogreje. Lahko se segreje do te mere, da uniči ploščico.

Mikrokrmilnik na ploščici ima program, ki omogoča nalaganje programa brez posebnega programatorja. Program se prenese preko povezave USB in



Slika 2.1: Arduino Uno, razvojna ploščica

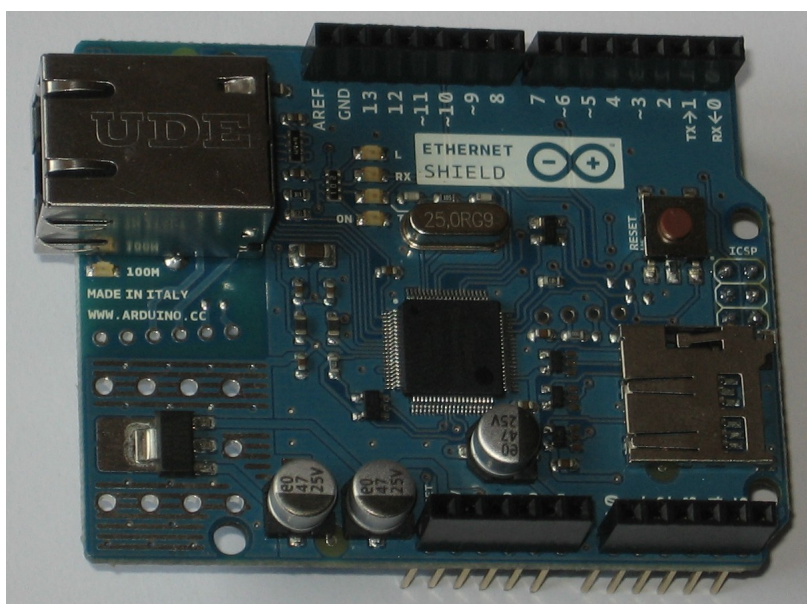
zapiše na čip, ki se tudi ponovno zažene.

Za programiranje je na voljo odprtokodno integrirano razvojno okolje. Okolje vsebuje knjižnice za C++. Knjižnice pomagajo pri uporabi ploščice in nekaterih dodatkov. Vmesnik omogoča tudi pisanje kode, prevajanje in pošiljanje programa na krmilnik.

2.2 Arduino Ethernet Shield

Dodatna kartica za dostop do omrežja (Arduino Ethernet Shield, slika 2.2) vsebuje mrežni čip Wiznet W5100. Ta vsebuje nabor IP, tako TCP kot UDP. Omogoča 4 sočasne povezave. Kartica vsebuje tudi mesto za kartico micro SD. Na omrežje se lahko priklopi z 10 Mbps ali 100 Mbps. Omogoča avtomatsko nastavljanje MDI/MDIX (za povezavo do računalnika, razdelilnika ali stikala ne potrebujemo kabla z navzkrižno vezavo).

Ethernet Shield je z Arduino Uno povezan s pomočjo vodila SPI, ki uporablja digitalne nožice 11, 12 in 13. Oba dela si delita vodilo, zato je lahko izbran samo mrežni ali kartični vmesnik. Nožica 10 je za izbiro W5100, nožica



Slika 2.2: Arduino Ethernet Shield, mrežna ploščica

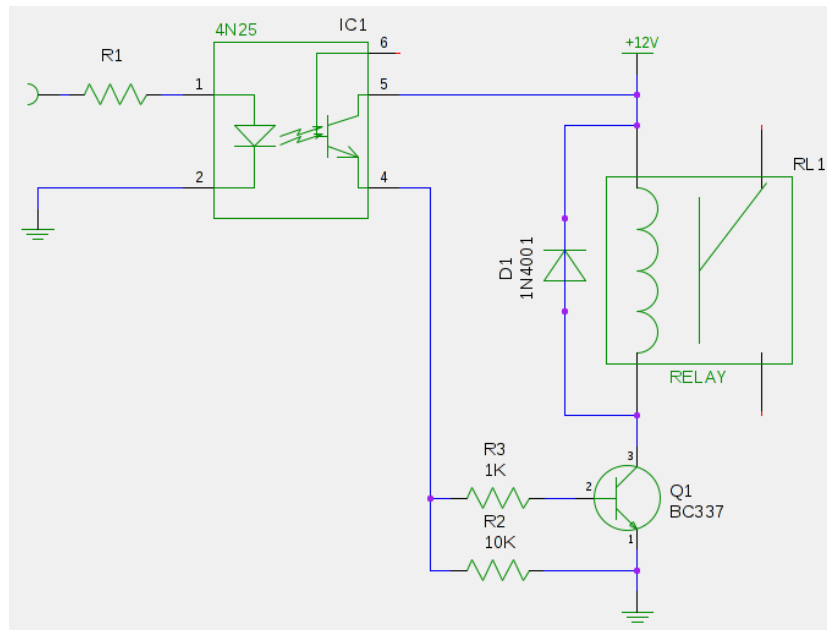
4 pa za izbor SD. Ethernet Shield prejema napajanje preko Arduino ploščice na katero je priključena. Več v [2].

Na voljo so knjižnice za uporabo mrežnega vmesnika in kartic SD. Omogočajo vnos omrežnih nastavitev (naslov MAC, naslov IP, omrežno masko in mrežni prehod) in povezave preko TCP ali UDP. Postavimo lahko tudi strežnik, ki na izbranih vratih čaka na povezave. Ko je vmesnik nastavljen vrača tudi odziv na ping.

2.3 Rele

Za vklop in izklop luči se uporablja na krmilnik priključen rele *JW2SN-DC12V*.

Rele uporablja zunanje napajanje 12 V in porabi 530 mW (dokumentacija releja [16]). Na ploščico Arduino je povezan preko optospojnika (4N28). Krmilnik je torej galvansko ločen od vezja, zato napake pri napajanju ne poškodujejo krmilnika. Na optospojnik je priključen tranzistor (BC337). Tranzistor je potreben, ker optospojnik ne prenaša dovolj toka za delovanje

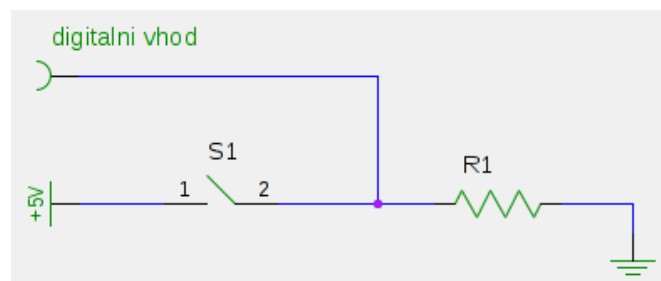


Slika 2.3: Vezava releja

releja. Vezava je na sliki 2.3.

2.4 Stikalo

Da je mogoče prižgati luč tudi ročno, se uporablja stikalo. Stikalo luči je priklopljeno na vhod krmilnika. Signal je povezan z zemljo s spustnim uporom, preko stikala pa je priklopljen na 5 V (slika 2.4).



Slika 2.4: Vezava stikala

Poglavje 3

Senzorji

Za zaznavanje podatkov iz okolja se poleg stikal potrebuje tudi senzorje. V tem poglavju so predstavljeni uporabljeni senzorji in njihove vezave.

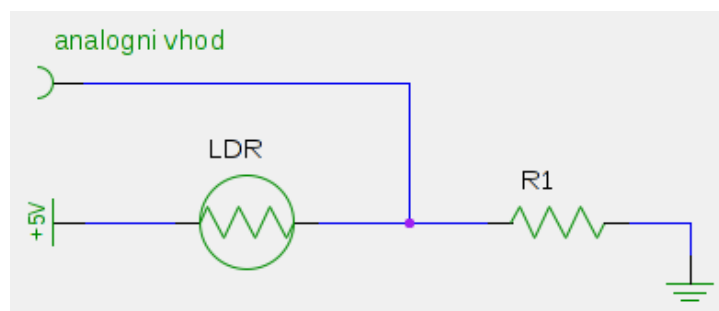
3.1 Svetlobni senzor

Intenzivnost svetlobe je možno meriti s svetlobnim uporom. Pri tem delu se uporablja svetlobni upor *LDR07* (slika 3.1). Svetlobni upor je upor, ki ima veliko upornost v temnem prostoru. Z večanjem svetlosti njegova upornost upada.

Svetlobni upor je narejen iz visoko upornih polprevodnikov. Če na napravo pada svetloba z dovolj visoko frekvenco, potem fotoni, ki jih absorbira polprevodnik dajejo vezanim elektronom dovolj energije, da skočijo v prevo-



Slika 3.1: Svetlobni upor LDR07



Slika 3.2: Vezava svetlobnega upora (LDR)

dni pas. Ostali prosti elektron prevaja elektriko in zato zniža upornost [12].

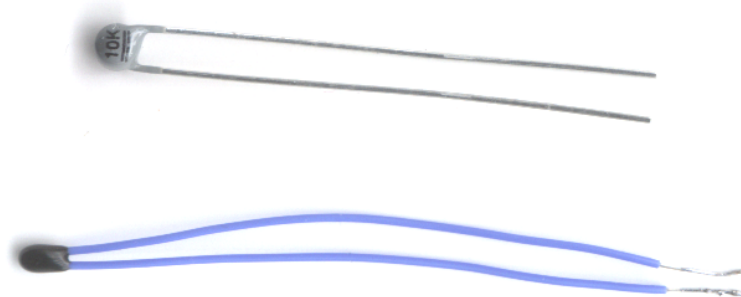
Svetlobni upor se enostavno priklopi na krmilnikov analogni vhod. Ker krmilnik na analognih vhodih meri napetost, se priklopi svetlobni upor s spustnim uporom (kot na sliki 3.2). Krmilnik ima 10 bitni analogno digitalni pretvornik. Tako vrne vrednost od 0 do 1023. Višja vrednost predstavlja svetlejše okolje. Bolj poglobljena razlaga svetlobnih uporov je zapisana v [6].

3.2 Temperaturni senzor

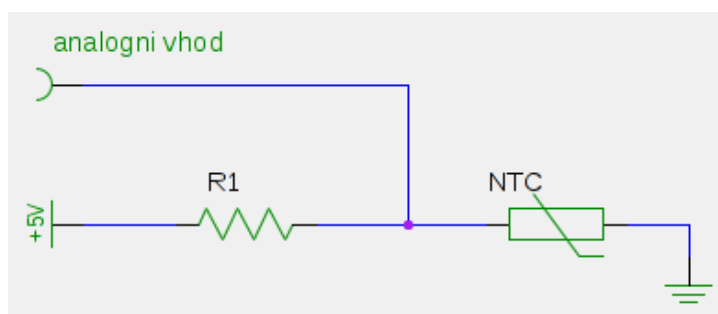
Za merjenje temperature obstaja veliko možnosti. Najenostavneje je z uporabo *termistorja*. Termistor je upor, čigar upornost se zelo spreminja s spremembo temperature. V tem delu se uporablja termistorje z negativnim koeficientom [8] (“negative temperature coefficient”, NTC). Eden meri zunanjo temperaturo, drugi pa notranjo. Negativni koeficient pomeni, da pri naraščajoči temperaturi pada upornost.

Spodnji termistor na sliki 3.3 ima 1% napako, zgornji pa 5% napako, pri 25 °C je upornost obeh 10 kΩ. Tako je napaka pri 25 °C približno $\pm 0,25$ °C in $\pm 1,25$ °C. Termistor se priklopi na zemljo in analogni vhod. Analogni vhod je tudi z uporom dvignjen na napetost (kot na sliki 3.4).

Do temperature se pride preko upornosti termistorja (kot v [15]). Mikrokontroler nima vgrajenega merilca upornosti. Ima pa merilec napetosti znan kot analogno digitalni pretvornik. Zato je potrebno pretvoriti upornost



Slika 3.3: Termistorja z negativnim koeficientom



Slika 3.4: Vezava termistorja z negativnim koeficientom

v napetost. To se naredi z dodatnim uporom, ki je povezan zaporedno (kot na sliki 3.4). Sedaj se izmeri napetost v sredini. Ko se spremeni upornost se spremeni tudi napetost. S poznano upornostjo enega upora se lahko izračuna upornost drugega z napetostno delilno enačbo [14].

V enačbi (3.1) je R_1 upornost dodanega upora, R_2 pa upornost termistorja. Potrebuje se upornost R_2 zato jo izpostavimo in dobimo (3.2). Napetost V_{in} je napajalna napetost, V_{out} pa pride na analogni vhod.

$$V_{out} = \frac{R_2}{R_1 + R_2} V_{in} \quad (3.1)$$

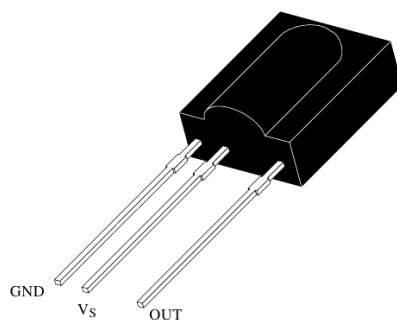
$$R_2 = \frac{R_1}{\frac{V_{in}}{V_{out}} - 1} \quad (3.2)$$

Termistorji spreminjajo upornost. Obstajajo tudi analogni senzorji, ki s spremembo temperature vračajo spreminjajočo napetost. *LM35* [9] vrača temperaturo na izhodu linearno v Celzijevi skali z $10\text{mV}/1^\circ\text{C}$. Ima napako $\pm 0,25^\circ\text{C}$. Potrebuje napajanje 4 V do 20 V pri $60\ \mu\text{A}$. *MCP9700A* [10] potrebuje napajanje 2,3 V do 5,5 V pri $6\ \mu\text{A}$. Izhod je linearen z enako napako in premikom v Celzijevi skali, samo vse vrednosti so dvignjene za 500 mV (pri 0°C je vrednost 500 mV).

Lahko se uporabi tudi digitalni temperaturni senzor *DS18B20* [11], ki ima natančnost $\pm 0,5^\circ\text{C}$. Za napajanje potrebuje 3 V do 5,5 V s tokom do 1,5 mA. Senzor se priklopi preko vodila *1-Wire*. To vodilo potrebuje poleg zemlje samo eno žico. Na vodilo lahko priklopimo več podobnih senzorjev. Senzor se lahko parazitsko napaja preko signalne povezave.

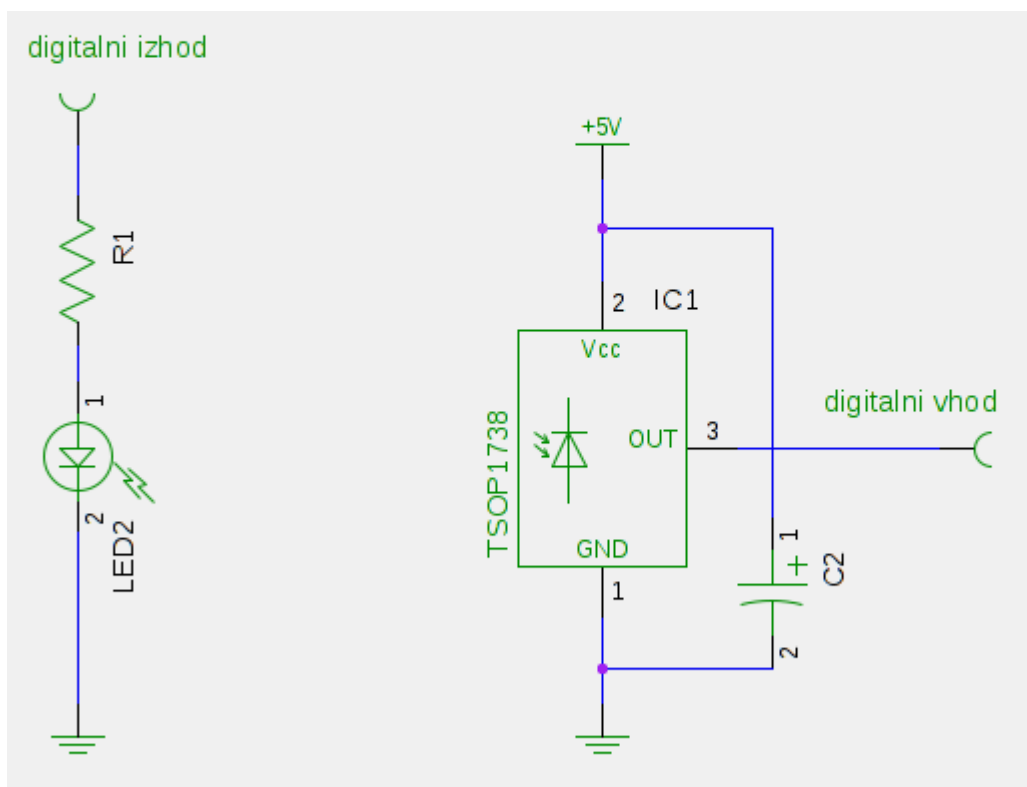
3.3 Senzor vstopa v prostor

Zaznavanje vstopa v prostor je realizirano z infrardečo (IR) svetlečo diodo in IR senzorjem *TSOP1738* (slika 3.5). Na sliki 3.6 je diagram vezave: svetleča IR dioda na levi strani in senzor *TSOP1738* na desni strani. Nameščena sta na različnih straneh vrat. Da tak sistem deluje je potrebno natančno vklapljeti in izklapljeti svetlečo diodo. Razdalja med diodo in senzorjem je lahko kar nekaj metrov.



94 8691

Slika 3.5: Senzor serije TSOP17 (vir: [7])

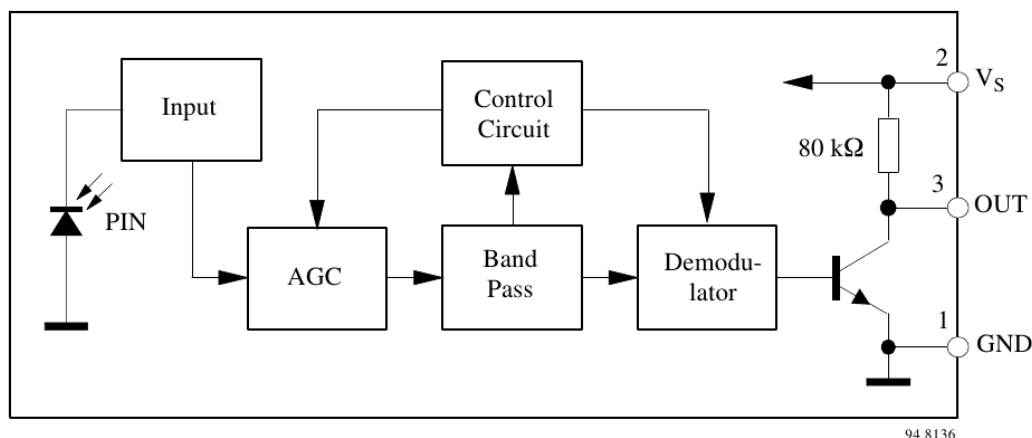


Slika 3.6: Vezava senzorja vstopa: oddajnik (levo) in sprejemnik (desno)

TSOP1738 senzor najbolje zajema infrardeče signale s frekvenco blizu 38 kHz. To je frekvenca, ki jo uporablja večina daljinskih upravljalnikov za televizijske sprejemnike, klimatske naprave, CD predvajalnike in drugo.

Senzor zaznava vstop v prostor ob prekinitvi IR signala. Ko je signal prisoten je izhod nizek, ko signala ni je izhod v visokem stanju. Senzor potrebuje regulirano 5 V napajanje. Zelo je priporočeno, da je blizu senzorja kondenzator. Če napetost prekorači meje, senzor ne zazna signala. Podrobnosti senzorja družine TSOP17 so navedene v [7].

Za signal frekvence 38 kHz je potrebno diodo prižigati in ugašati na $26,3 \mu\text{s}$. Brez uporabe dodatnih komponent je v programu krmilnika tolikšno natančnost težko zagotoviti. Z uporabo časovnika krmilnika ATmega328 je s pravimi nastavitvami to mogoče. ATmega328 vsebuje tri časovnike, ki lahko preklapljujejo šest izhodov, po [3].



Slika 3.7: Diagram senzorja serije TSOP17 (vir: [7])

3.4 Senzor gibanja v prostoru

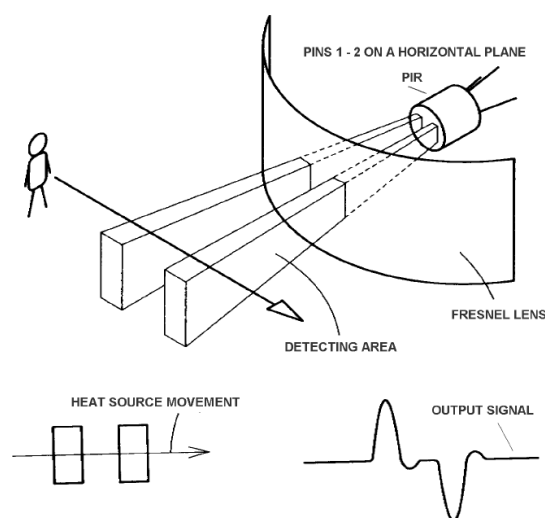
Za senzor gibanja se uporablja pasivni infrardeči senzor (PIR - “Passive Infrared”). To je senzor, ki porabi malo elektrike, je poceni in enostaven za uporabo (po [13]). Senzor se priklopi na napajanje in na digitalni vhod.

3.4.1 Sestava pasivnega infrardečega senzorja

Pasivni infrardeči senzor (RE200B) ima dve reži. Vsaka reža je narejena iz posebnega materiala občutljivega na infrardečo svetlobo. Na sliki 3.8 je prikazan princip delovanja PIR senzorja.

Senzor zazna infrardečo svetlobo dveh območij zaznavanja na določeni razdalji. Ko v prostoru ni gibanja obe reži zaznata enako količino infrardeče svetlobe (vrednost sevanja iz prostora, stene ali iz okolice).

Ko gre mimo toplo telo, to je človek ali žival, to najprej zazna ena polovica senzorja. To povzroči pozitivno razliko med obema stranema. Segrevanje ali ohlajevanje zaznanega območja vpliva na obe strani senzorja, zato se ignorira. Ko toplo telo zapusti območje zaznavanja se zgodi nasprotno. Pri tem senzor naredi negativno diferencialno spremembo. Te spremembe so tisto, kar je zaznano.



Slika 3.8: Princip delovanja PIR senzorja RE200B (vir: [18])

Senzor

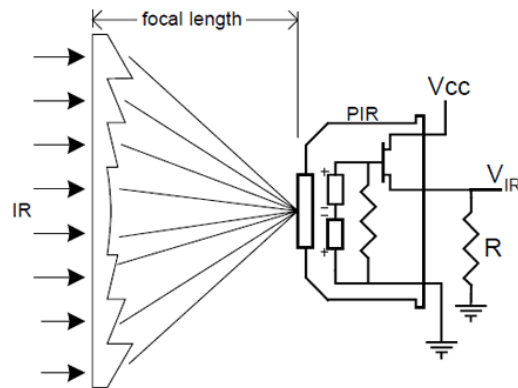
Del IR senzorja, ki zaznava je hermetično zaprt v kovinskem ohišju. To ščiti pred vlago, temperaturo in šumom. V ohišju je okno, ki je narejeno iz materiala, ki prevaja infrardečo svetlobo. Za oknom sta dva uravnovežena senzorja.

Leča

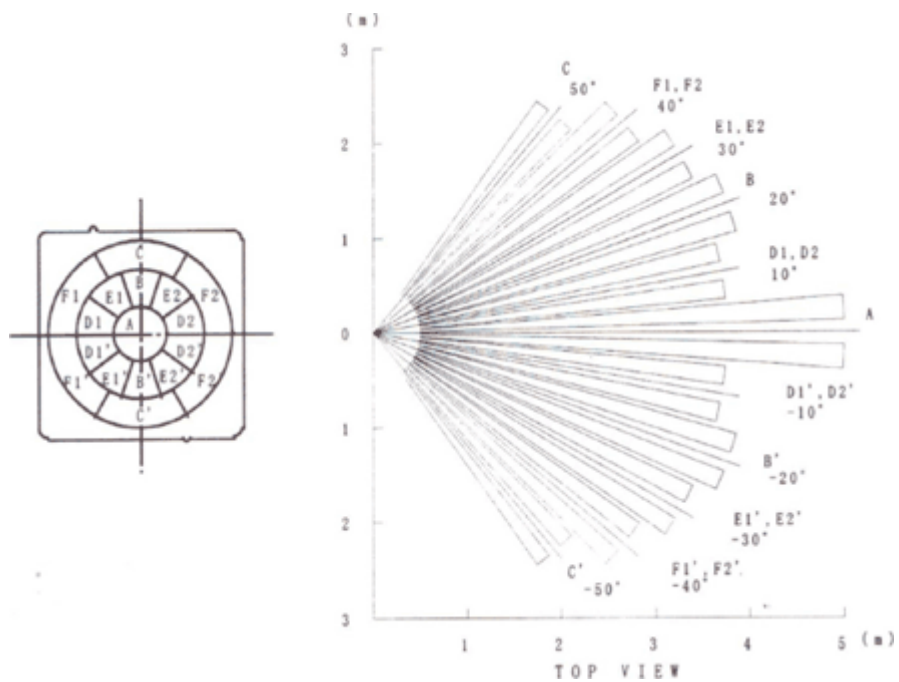
V prejšnji sliki je leča kos plastike, ki zazna samo območje dveh pravokotnikov. Ponavadi se želi zaznati veliko večje območje. Zato se uporabi preprosto lečo, kot v kamerah. Za pasivni infrardeči senzor morajo biti leče majhne in poceni, tudi če je slika zato zamegljena. Zato so senzorji v bistvu Fresnelove leče (slika 3.9).

Z lečo se zajema veliko večje območje. Zaznavanje območja dveh velikih pravokotnikov ni namen tega senzorja. Zajeti se želi razpršena manjša območja. Zato se razdeli lečo v več delov. Vsak del je Fresnelova leča.

Različni deli podleč naredijo zaznana območja prepletena med seboj. Zato so deli nekonsistentni, vsak kaže na drugo polovico senzorja.



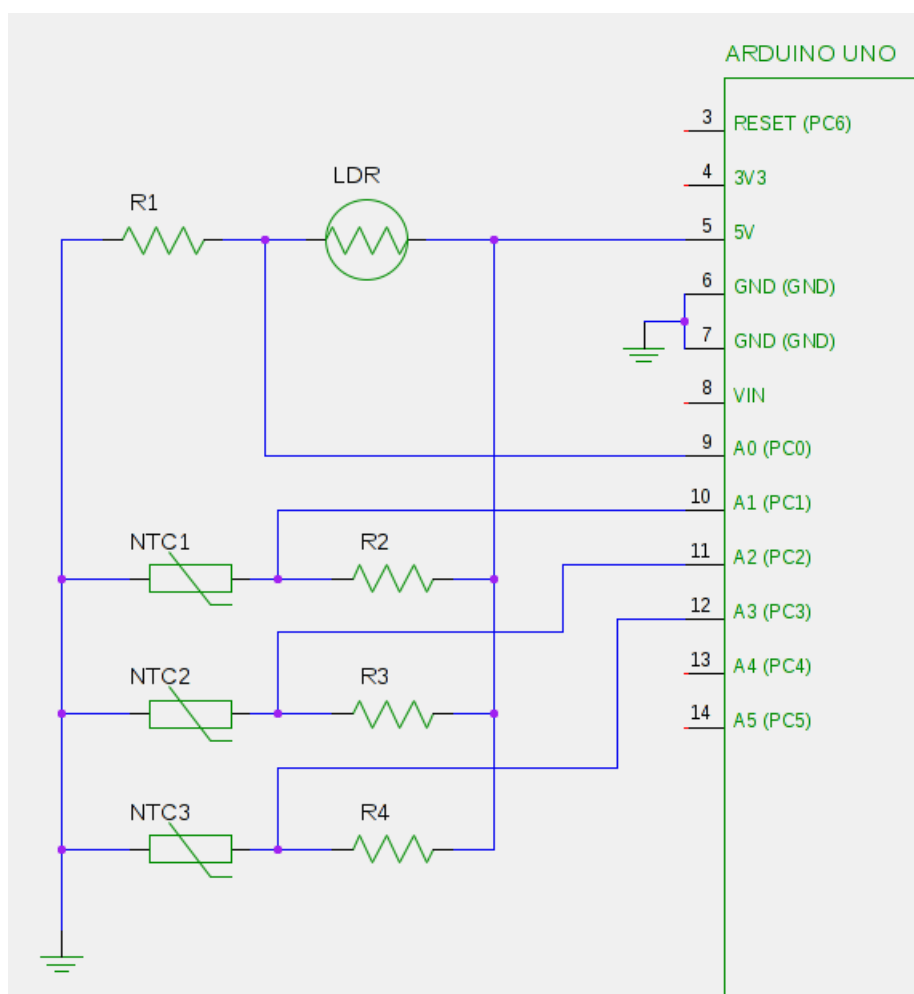
Slika 3.9: Fresnelova leča kondenzira svetlobo in omogoči zaznavanje večjega območja na PIR senzorju RE200B (vir: [13])



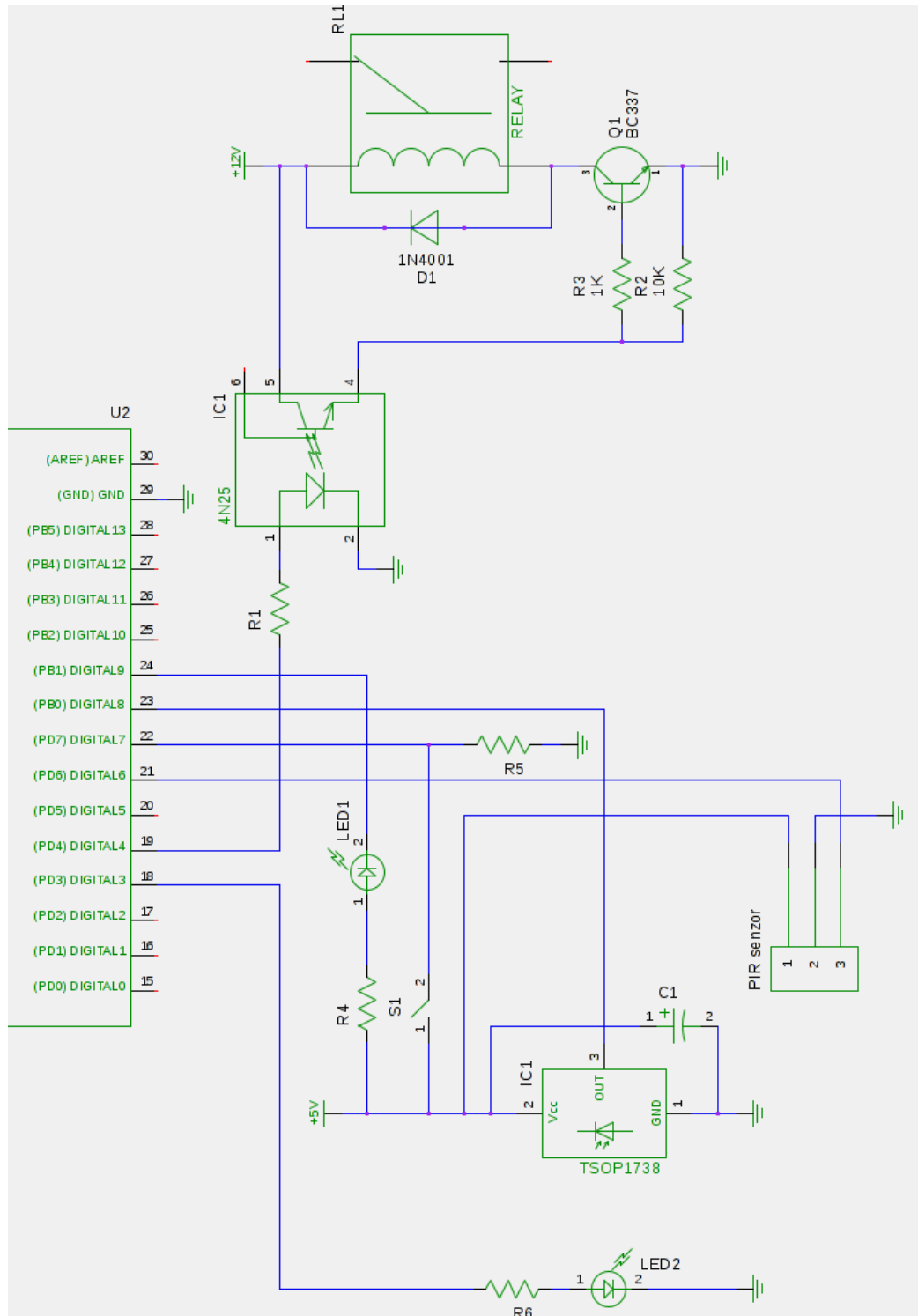
Slika 3.10: Zaznana območja so razdeljena na obe polovici senzorja (vir: [13])

3.5 Vezava celotnega sistema

Celoten sistem je zasnovan po naslednjih dveh shemah. Najprej je predstavljena vezava analognih vhodov (3.11), nato še digitalni del (3.12).



Slika 3.11: Shema sistema povezanega na analogni del



Slika 3.12: Shema sistema povezanega na digitalni del

Poglavje 4

Programska oprema

4.1 Razvojno okolje

Za programiranje krmilnika Arduino je na voljo odprtokodno razvojno okolje. Razvojno okolje vsebuje grafični vmesnik (slika 4.1). Grafični vmesnik je napisan v Javi, deluje v različnih operacijskih sistemih. Omogoča pisanje kode, prevajanje in programiranje ploščice. Vsebuje tudi terminal za serijsko povezavo do ploščice.

Pisanje programov v tem okolju je težko, ker nima funkcij, ki so prisotne v drugih razvojnih vmesnikih in urejevalnikih besedila.

Razvojno okolje vsebuje tudi knjižnice za lažjo uporabo krmilnika in dodatkov za krmilnik. Za uporabo razvojnega okolja se potrebuje še prevajalnika C in C++ in standardne knjižnice za arhitekturo Atmel AVR.

Za zapis programa na sam krmilnik je na njem poseben program za nalaganje, ki omogoča zapis kode preko povezave USB. Uporabi pa se lahko tudi programator za čip, ki naloži program direktno na krmilnik.

Poleg grafičnega vmesnika ima razvojno okolje tudi dodatne nastavitve za prevajanje (Makefile), ki omogočajo enostavno prevajanje kode brez grafičnega vmesnika. V Makefile se napiše samo potrebne knjižnice, kateri procesor se uporablja in nastavitve za programator:

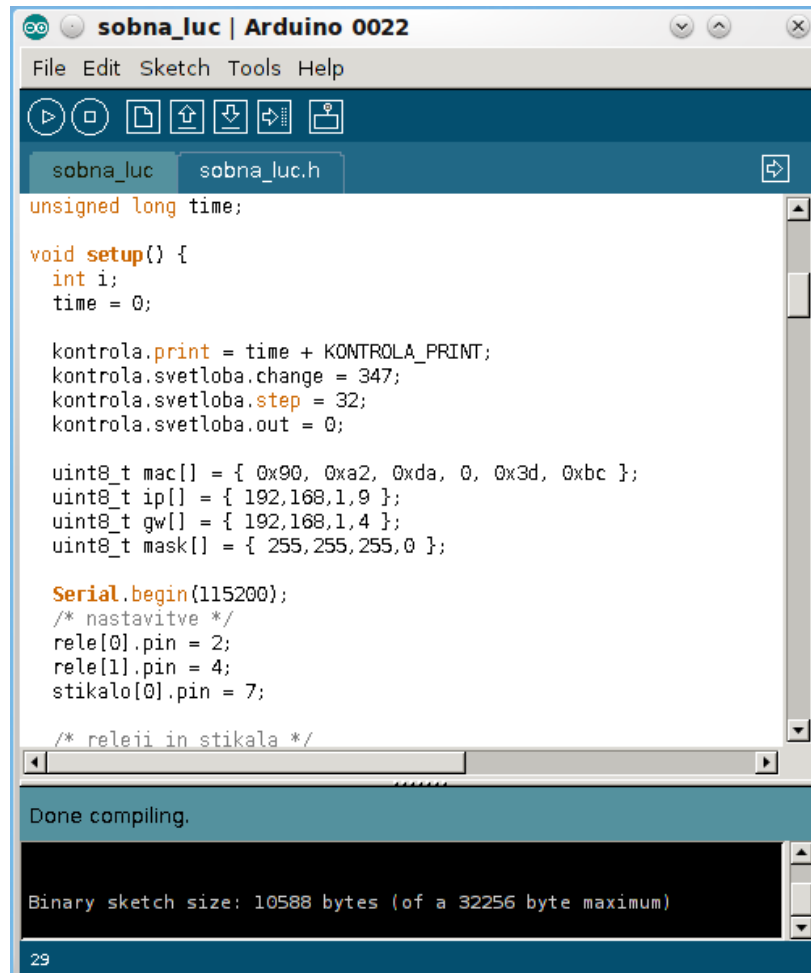
```
ARDUINO_DIR ?= /usr/share/arduino
```

```

ARDMK_DIR      ?= /usr
AVR_TOOLS_DIR?= /usr
TARGET         ?= sobna_luc
ARDUINO_LIBS   ?= SPI Ethernet/utility
BOARD_TAG      = uno
ARDUINO_PORT   ?= /dev/serial/by-id/usb-Arduino*
AVRDUDE_ARD_EXTRAOPTS  ?= -D

include /usr/share/arduino/Arduino.mk
CPPFLAGS := $(subst -w,,$(CPPFLAGS)) # Vrnemo opozorila nazaj

```



Slika 4.1: Grafični vmesnik razvojnega okolja Arduino

Na tak način se prevedejo vse poznane datoteke iz tega direktorija. Povežejo se še s knjižnicami, ki nastavijo krmilnik. Med njimi je deklarirana tudi `main()` funkcija, ki nastavi krmilnik in požene funkcijo `setup()` in v zanki poganja funkcijo `loop()`, ki ju mora napisati uporabnik.

4.2 Program

Program je napisan v jeziku C++. Na začetku je kot globalna spremenljivka nastavljen primerek kontrolnega razreda. V `setup()` se nastavi serijska povezava in požene funkcija v kontrolnem razredu. V `loop()` se prebere čas od zagona programa in požene drugo funkcijo kontrolnega razreda. Kontrolni razred vsebuje objekte za senzor svetlobe, temperature, branje stikal, upravljanje z releji in objekt za prejem ukazov.

4.2.1 Temperatura

Na analognem vhodu je 10 bitna vrednost, ki predstavlja upornost termistorja. Pri temperaturi je zanimiva samo vrednost v stopinjah Celzija, ne upornost termistorja. V specifikaciji termistorja [8] je tabela temperatur in razmerja upornosti pri njih. Tabela vsebuje vrednosti za temperature od -55 °C do 155 °C po 5 °C .

Lažje je izračunati točno temperaturo iz izmerjene upornosti. Temperatura se lahko izračuna (kot pri [17]) z uporabo Steinhart Hartove enačbe (4.1), ki je samo približek. Vrednosti iz tabele so bolj točne, približek pa je dovolj dober.

$$\frac{1}{T} = a + b \ln(R) + c \ln^3(R) \quad (4.1)$$

Za celotno Steinhart Hartovo enačbo ni vseh parametrov. Za termistorje NTC se lahko uporabi poenostavljena enačba (4.1) s parametri $a = (\frac{1}{T_0}) - (\frac{1}{B}) \ln(R_0)$, $b = \frac{1}{B}$ in $c = 0$. Za to enačbo je potreben parameter B , ki se ga dobi v specifikaciji termistorja. V enačbi parametra B (4.2) predstavlja R_0

upornost termistorja pri temperaturi T_0 . Tu se uporablja dva termistorja z $R_0 = 10 \text{ k}\Omega$ in $T_0 = 25 \text{ }^\circ\text{C} = 298.15 \text{ K}$, parametra B pa sta 3950 in 3988.

$$\frac{1}{T} = \frac{1}{T_0} + \frac{1}{B} \ln\left(\frac{R}{R_0}\right) \quad (4.2)$$

4.2.2 Kontrola luči

Prižiganje luči glede na svetlost v prostoru je implementirano v svojem razredu, ki ima tri različna stanja: 0 - svetlost ne vpliva na prižiganje, 1 - poleg svetlosti je potrebno tudi gibanje v prostoru, 2 - za prižiganje je potrebna samo svetlost v prostoru. Ko je luč prižgana oz. ugasnjena s stikalom se stanje postavi na 0.

Svetlost je izmerjena kot napetost na analognem vhodu. Prebrana je v vsakem obhodu `loop()` zanke. Nihanje napetosti je ublaženo z vpisovanjem vrednosti v polje dolžine 16, katerega povprečno vrednost se uporablja. Svetloba vpliva na prižig, ko vrednost pade pod nastavljeno mejo. Da se luč ne prižiga in ugaša pri premajhni spremembi je nastavljena tudi meja. Tako se ugasne, ko je napetost večja za nastavljen interval.

4.2.3 Ukazi

Razred za prejem ukazov na začetku nastavi mrežni vmesnik in prične s sprejemanjem povezav preko nastavljenih vrat. Ob vsakem obhodu se preveri, če je preko mrežne ali preko serijske povezave prejet ukaz. Ko je ukaz prejet v celoti je posredovan kontrolnemu razredu, ki ga izvede. Vsak ukaz se konča z znakom za konec vrstice.

Z ukazom za izpis pomoči se izpiše uporaba ukazov (na sliki 4.2). Obstajajo ukazi za upravljanje z releji, simuliranje preklopa stikala, spreminjanje stanja kontrole luči ter ukazi za spreminjanje in shranjevanje nastavitev.

Pri spreminjanju nastavitev se lahko nastavi mrežno kartico: naslov IP, prehod, vrata na katerih posluša, strojni naslov. Obstajajo ukazi za nastavitve kontrole svetlosti: osvetlitev pri kateri se vklopi luč in meja, vhod senzorja

```

sobna_luc r230
Uporaba ukazov:
  rele[ <0..4> [0|1]]
  t[<0..4>[0|1]]
  auto [0|1]]
  show
  set
    ip <a.dd.re.ss>[:<port>][/<netmask>]
    gw <a.dd.re.ss>[:<port>][/<netmask>]
    port <port>
    mac <HW:a:dd:re:s:s>
    svet <change> <step>
    svet = <pin|A\d+ svet senzorja> <rele luči> <led>
    svet ir <ir stik.> <pir stik.>[!]
    svet fc <ftime> <ctime>
    rele <index> <pin>
    stik <index> <pin>[!] [<rele index>]
    topl <index> <pin|A\d+> <B> <R0> <RC> <T0>
  save <rele|stik|topl|svet|stre>
  quit

```

Slika 4.2: Izpis pomoči programa na krmilniku

svetlobe, indeks releja za preklon luči in diode prikaza statusa, vhoda na katera je priklopljen IR senzor pri vratih in senzor gibanja ter zakasnitev pri prižigu in ugašanju luči pri prehodu skozi vrata, so tudi ukazi za nastavitvev relejev in stikal in za senzorje temperature, kjer je potrebno dodati konstanto in upornost senzorja ter vezanega upora. Vse te nastavitve se lahko tudi zapiše na pomnilnik EEPROM z ukazom za shranjevanje.

4.3 Upravljanje s krmilnikom

Krmilnik se upravlja preko mrežne ali serijske povezave. Ukazi so sestavljeni iz navadnega besedila. Znak za novo vrstico predstavlja konec ukaza. Zato se lahko za dostop uporabi Telnet klient. Enostavna uporaba je s programom `netcat`. Vklon prvega releja se lahko napiše kot:

```
echo 'rele 0 1' | nc 192.168.1.9 23
```

Priklapljanje in izklapljanje avtomatskega upravljanja z lučjo je smiselno klicati pri zagonu in ugašanju računalnika ali pri prijavi in odjavi iz računalnika. Prav pride tudi preklon pri zagonu ohranjevalnika zaslona. Če

se uporablja operacijski sistem GNU/Linux potem je lahko v mapi `~/bin` skript, ki nadomesti ohranjevalnik zaslona:

```
#!/bin/bash

term() {
    kill -TERM %1
}

trap term TERM
trap '' CHLD

/usr/bin/kblankscrn.kss "$@" &
echo "a~2" | nc 192.168.1.9 23 > /dev/null
wait
echo "a|2" | nc 192.168.1.9 23 > /dev/null
```

Tako se vsakič takoj za ohranjevalnikom zaslona pošlje ukaz na krmilnik, ki izklopi stanje 2. Ko se prekine ohranjevalnik zaslona se spet nastavi nazaj na stanje 2. Podobno se lahko spreminja stanje tudi ob drugih dogodkih.

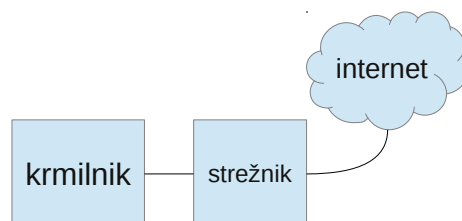
Poglavje 5

Spletna aplikacija

Za prikaz podatkov se uporablja spletna aplikacija. Spletna aplikacija poveča dostopnost podatkov. Tako lahko pogledamo stanje sistema z vsemi napravami s spletnim brskalnikom, na primer z mobilnim telefonom.

Aplikacija ne teče na krmilniku Arduino. Krmilnik na Arduino ploščici je majhen in počasen, zato se za hranjenje in obdelavo podatkov uporablja zunanji računalnik. Zunanji sistem lahko hrani veliko več podatkov in pomaga pri izdelavi grafov. Podatke se lahko uporabi tudi za kasnejšo analizo porabe elektrike in podobno.

Spletna aplikacija ne dostopa direktno do krmilnika, zato napaka v aplikaciji ne omogoči dostopa do sistema. Seveda se lahko omeji tudi dostop do aplikacije z uporabo gesel in kriptirane povezave. To bi bilo na samem krmilniku težko implementirati zaradi omejitev pomnilnika in velikosti programa.



Slika 5.1: Povezava krmilnika s strežnikom in tako internetom

```

$ nc sl 23
T: {0!465-,A2=+3.8,A1=+22.6,A3=+25.1} (4=00)@61106 (2=00) (9=00)@22418 [ 7:0=1]@61106 @868274109
T: {0!464-,A2=+4.2,A1=+22.6,A3=+25.2} (4=00)@25570 (2=00) (9=00)@52418 [ 7:0=1]@25570 @868304109
t: {0!465-,A2=+3.7,A1=+22.5,A3=+25.0} (4=00)@39590 (2=00) (9=00)@902 [ 7:0=1]@39590 @868318129
T: {0!463-,A2=+3.6,A1=+22.6,A3=+25.2} (4=00)@55571 (2=00) (9=00)@16883 [ 7:0=1]@55571 @868334110
T: {0!463-,A2=+3.2,A1=+22.4,A3=+25.1} (4=00)@20035 (2=00) (9=00)@46883 [ 7:0=1]@20035 @868364110
t: {0!462-,A2=+3.1,A1=+22.5,A3=+25.1} (4=00)@34009 (2=00) (9=00)@60857 [ 7:0=1]@34009 @868378084
3: {0!462+,A2=+3.3,A1=+22.4,A3=+25.0} (4=00)@35367 (2=00) (9=00)@62215 [ 7:0=1]@35367 @868379442
3: {1!460-,A2=+3.6,A1=+22.5,A3=+25.0} (4=00)@35801 (2=00) (9=00)@62649 [ 7:0=1]@35801 @868379876
4: {7!461-,A2=+3.4,A1=+22.5,A3=+25.0} (4=00)@37190 (2=00) (9=00)@64038 [ 7:0=1]@37190 @868381265
2: {6!461-,A2=+3.4,A1=+22.4,A3=+25.0} (4=00)@37202 (2=00) (9=00)@64050 [ 7:0=1]@37202 @868381277
3: {2!461+,A2=+3.5,A1=+22.4,A3=+25.0} (4=00)@37214 (2=00) (9=00)@64062 [ 7:0=1]@37214 @868381289
4: {6!460+,A2=+3.6,A1=+22.5,A3=+25.0} (4=00)@38741 (2=00) (9=00)@53 [ 7:0=1]@38741 @868382816
T: {0#458+,A2=+4.1,A1=+22.5,A3=+25.1} (4=00)@3205 (2=00) (9=00)@30053 [ 7:0=1]@3205 @868412816
4: {6#457+,A2=+4.1,A1=+22.5,A3=+25.0} (4=00)@8704 (2=00) (9=00)@35552 [ 7:0=1]@8704 @868418315
t: {6#456+,A2=+4.3,A1=+22.6,A3=+25.0} (4=00)@28430 (2=00) (9=00)@55278 [ 7:0=1]@28430 @868438041
T: {6#455+,A2=+4.4,A1=+22.7,A3=+25.1} (4=00)@38705 (2=00) (9=00)@17 [ 7:0=1]@38705 @868448316

```

Slika 5.2: Izpis stanja krmilnika

5.1 Pridobivanje podatkov

Za pridobivanje podatkov se uporablja program, ki se priklaplja na krmilnik preko mrežne povezave.

Ko je povezava vzpostavljena, krmilnik pošilja podatke približno vsake pol minute. Kadar se spremeni stanje relejev in kadar se temperatura spremeni sunkovito (za več kot 2 °C) krmilnik takoj pošlje sporočilo vsem povezanim klientom.

Program za pridobivanje podatkov skrbi, da redno prejema podatke. Pošlje ukaz za izpis stanja, če ni prejel podatkov v zahtevanem času in preverja stanje povezave. Če je bila povezava prekinjena jo ponovno vzpostavi.

Podatki o stanju, ki jih pošilja krmilnik (kot na sliki 5.2) so pregledni za uporabnika. Program mora iz njih pridobiti primerne podatke za vpis v bazo. Podatki se hranijo v tabeli objektno relacijske baze podatkov. V vsaki vrstici tabele (slika 5.3) so zapisani podatki o svetlosti, temperaturah in stanju relejev. Podatki se zajemajo večkrat na minuto.

5.2 Prikaz podatkov

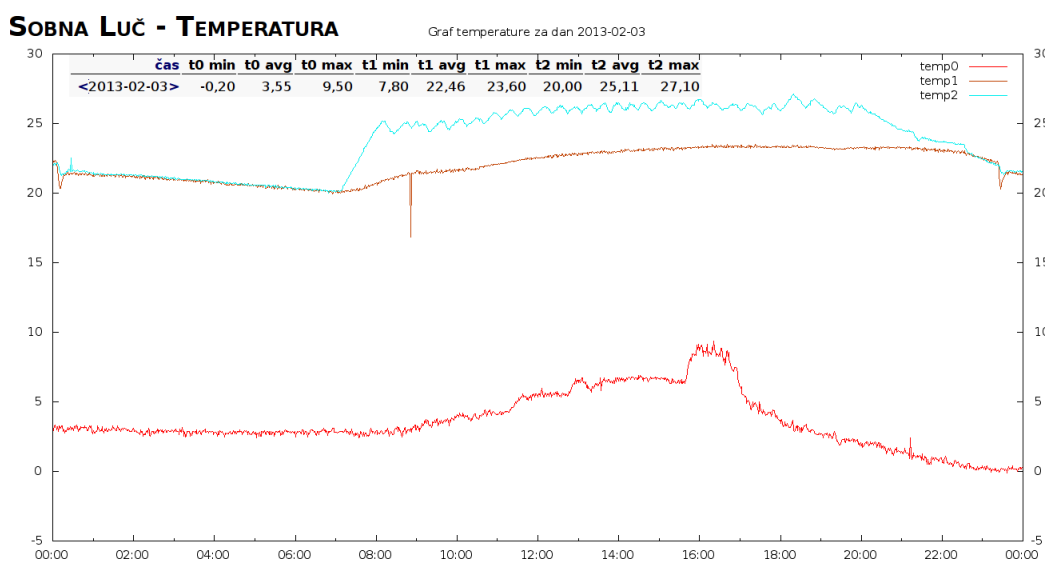
Prikaz vrši spletna aplikacija, ki se priklaplja na bazo podatkov. V tabeli je veliko podatkov, krmilnik jih pošilja večkrat na minuto. Zato se pri prikazih

```

info=# select * from sobna_luc order by time desc limit 10;
      time                | tip | svetlost | temp0 | temp1 | temp2 | temp3 | rele0 | rele1 | rele2
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----
2013-02-04 11:01:11.286172+01 | T   |    482   |   3.9 |  22.4 |  26.4 |         | f     | f     | f
2013-02-04 11:00:41.265694+01 | T   |    482   |   4   |  22.7 |  26.5 |         | f     | f     | f
2013-02-04 11:00:11.243062+01 | T   |    483   |   4   |  22.5 |  26.5 |         | f     | f     | f
2013-02-04 10:59:41.221001+01 | T   |    483   |   3.9 |  22.4 |  26.3 |         | f     | f     | f
2013-02-04 10:59:11.198859+01 | T   |    483   |   4   |  22.6 |  26.3 |         | f     | f     | f
2013-02-04 10:58:41.225924+01 | 2   |    483   |   4   |  22.6 |  26.3 |         | f     | f     | f
2013-02-04 10:58:41.166095+01 | 4   |    483   |   4   |  22.5 |  26.3 |         | f     | f     | f
2013-02-04 10:58:39.585432+01 | 3   |    483   |   4.1 |  22.6 |  26.3 |         | f     | f     | f
2013-02-04 10:58:39.486526+01 | 3   |    483   |   4.1 |  22.5 |  26.3 |         | f     | f     | f
2013-02-04 10:58:26.70069+01  | 4   |    482   |   4.2 |  22.7 |  26.3 |         | f     | f     | f
(10 rows)

```

Slika 5.3: Primer izpisa dela tabele iz relacijske baze



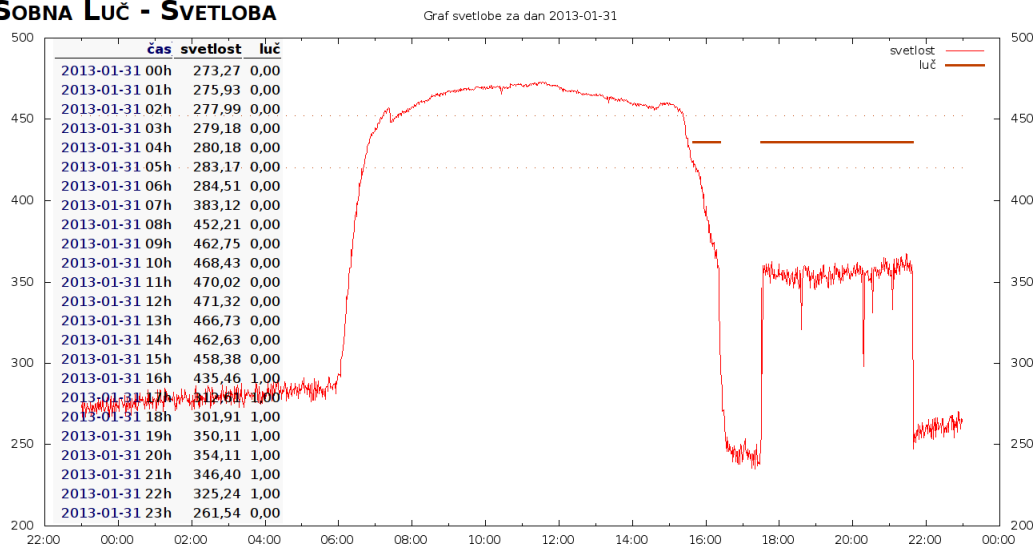
Slika 5.4: Stran za prikaz temperature. Temperatura se meri v treh točkah, kar prikazujejo grafi za temp0, temp1 in temp2.

uporablja povprečna vrednost najmanj vsake minute. Pri večjih pregledih se uporablja povprečje daljšega časovnega razpona.

Na sliki 5.4 je prikaz temperature zadnjih 24 ur. Prikazane so vse izmerjene temperature. Zgoraj je tabela minimalnih, povprečnih in maksimalnih vrednosti.

Pregleda se lahko tudi izmerjeno svetlost v prostoru. Poleg svetlosti je na sliki 5.5 prikazano tudi kdaj je bila prižgana luč. Pikčasti krivulji označujeta območje prižiga in izklopa luči glede na svetlost prostora. Na sliki je tudi tabela povprečnih vrednosti za vsako uro. V zadnjem stolpcu (luč) je vrednost

SOBNA LUČ - SVETLOBA



Slika 5.5: Stran za prikaz svetlosti in uporabe luči

1,00 če je bila v tem obdobju luč prižgana.

Na obeh straneh se da izbirati datum. Strani sprejemata parameter z datumom ('?t=2013-02-04') za izpis samo izbranega dne. Lahko se izbere celoten mesec ('?t=2013-01') oz. leto ('?t=2012'). Če je izbran datum, potem sta v tabeli tudi povezavi za prejšnji ('<') in naslednji ('>') dan (mesec ali leto).

Tabela vrednosti privzeto prikazuje povprečne vrednosti za vsak dan, pri mesečnem pregledu za vsak mesec. Podrobnosti tabele se povečajo s povezavo v glavi tabele levo (kjer piše 'čas'). Pri mesečnem prikazu se lahko razširi tabelo na vrednosti zaokrožene na dan, uro in na deset minut. Pri dnevnem prikazu se lahko izpiše tudi vrednosti zaokrožene na eno minuto.

Poglavje 6

Sklepne ugotovitve

Arduino Uno je bilo izbrano okolje za izdelavo diplomskega dela. Razvojna ploščica olajša uporabo krmilnika. Pisanje programov za ta krmilnik ni tako težko, ker se lahko uporablja programski jezik C++. Enostavno je dodati še mrežni vmesnik. Dostop do mreže odpre veliko možnosti in odpravi veliko omejitev samega krmilnika. Pomanjkanje dostopa do realne ure nas ne ovira, ker lahko računalniki preko mrežne povezave pošiljajo ukaze krmilniku. Beležimo lahko tudi koliko časa so določene naprave vklopljene.

Cilj diplomskega dela je bil izdelati uporaben in poceni sistem kontrole luči. Že od začetka je bila želja uporabe spletnega dostopa. Na začetku se je uporabljalo samo merjenje svetlobe in kontrola iz računalnika. Ta del je bil kmalu uspešno izveden. Malo je motila cena mrežnega vmesnika. Sedaj obstaja Arduino Ethernet, ki na eni kartici vsebuje krmilnik in mrežni vmesnik in je cenejši kot obe komponenti skupaj. Razširitev z uporabo senzorja gibanja in senzorja pri vratih je izboljšalo delovanje sistema. Kasneje je bilo dodano še merjenje temperature, ki je uporaben dodatek.

Z rastjo sistema se je povečeval program na krmilniku. Zaseda skoraj celoten pomnilnik flash, a pri tem ni bilo problema. Problem je nastal, zaradi prekoračitve pomnilnika SRAM, ki ga je samo 2 KB. Pri funkciji za format izpisa `sprintf()` se format, ki je zapisan v programu, skopira tudi v pomnilnik SRAM. Tako zaseda prostor v pomnilniku flash in SRAM. Nekaj

pomnilnika se sprost, če za format izpisa uporabimo kar sestavni del programa iz pomnilnika flash. Za to moramo uporabljati funkcijo `sprintf_P()` in makro `PSTR()`.

Za uporabo sistema je potreben vseskozi prižgan računalnik, ki pobira podatke. Bolje bi bilo skladiščiti več podatkov. Pri pobiranju podatkov bi bil naveden zadnji podatek in prejeti bi bili novejši. Za zbiranje podatkov bi se lahko uporabilo kartico SD, ki se jo lahko priklopi na ploščico mrežnega vmesnika.

Na ploščici Arduino Uno so zasedena že skoraj vsa digitalna vhodno izhodna vrata. Pri dodajanju novih senzorjev bi se lahko uporabilo vodilo 1-Wire, kjer se lahko na ena vrata priklopi več senzorjev.

Obstajajo pa tudi nekatere pomanjkljivosti ploščice Arduino Uno. Za napajanje se lahko uporablja napetost 6 V do 20 V. Ampak ko je priklopljen mrežni dodatek je pri napajanju z 9 V pretvornik napetosti preobremenjen. Pri daljši uporabi temperatura naraste tako visoko, da lahko uniči ploščico!

Nadzor luči, detekcija vstopa, gibanja v prostoru in merjenje temperature ni edina možna uporaba. Lahko bi nadzirali tudi druge senzorje in naprave.

Slike

1.1	Diagram sistema	6
2.1	Arduino Uno, razvojna ploščica	8
2.2	Arduino Ethernet Shield, mrežna ploščica	9
2.3	Vezava releja	10
2.4	Vezava stikala	10
3.1	Svetlobni upor LDR07	11
3.2	Vezava svetlobnega upora (LDR)	12
3.3	Termistorja z negativnim koeficientom	13
3.4	Vezava termistorja z negativnim koeficientom	13
3.5	Senzor serije TSOP17	14
3.6	Vezava senzorja vstopa: oddajnik (levo) in sprejemnik (desno)	15
3.7	Diagram senzorja serije TSOP17	16
3.8	Princip delovanja PIR senzorja RE200B	17
3.9	Fresnelova leča kondenzira svetlobo in omogoči zaznavanje večjega območja na PIR senzorju RE200B	18
3.10	Zaznana območja so razdeljena na obe polovici senzorja	18
3.11	Shema sistema povezanega na analogni del	19
3.12	Shema sistema povezanega na digitalni del	20
4.1	Grafični vmesnik razvojnega okolja Arduino	22
4.2	Izpis pomoči programa na krmilniku	25
5.1	Povezava krmilnika s strežnikom in tako internetom	27

5.2	Izpis stanja krmilnika	28
5.3	Primer izpisa dela tabele iz relacijske baze	29
5.4	Stran za prikaz temperature	29
5.5	Stran za prikaz svetlosti in uporabe luči	30

Literatura

- [1] (2011) Arduino Uno. Dostopno na:
<http://arduino.cc/en/Main/ArduinoBoardUno>
- [2] (2011) Arduino Ethernet Shield. Dostopno na:
<http://arduino.cc/en/Main/ArduinoEthernetShield>
- [3] (2011) 8-bit AVR Microcontroller with 4/8/16/32K Bytes In-System Programmable Flash Dostopno na:
http://www.atmel.com/dyn/resources/prod_documents/doc8161.pdf
- [4] (2011) W5100 Datasheet. Dostopno na:
http://www.wiznet.co.kr/UpLoad_Files/ReferenceFiles/W5100_Datasheet_v1.2.2.pdf
- [5] (2011) AVR Libc. Dostopno na:
<http://www.nongnu.org/avr-libc/user-manual/>
- [6] (2012) Photocells. Dostopno na:
<http://learn.adafruit.com/photocells/>
- [7] (2013) Photo Modules for PCM Remote Control Systems. Dostopno na:
http://www.datasheetcatalog.org/datasheets/208/301092_DS.pdf
- [8] (2012) NTC thermistors for temperature measurement. Dostopno na:
http://www.epcos.com/inf/50/db/ntc_09/LeadedDisks__B57891__M891.pdf

-
- [9] (2012) LM35 Precision Centigrade Temperature Sensors. Dostopno na:
<http://www.ti.com/lit/ds/symlink/lm35.pdf>
- [10] (2013) Microchip MCP9700/9700A MCP9701/9701A. Dostopno na:
<http://ww1.microchip.com/downloads/en/DeviceDoc/21942e.pdf>
- [11] (2013) DS18B20 Programmable Resolution 1-Wire Digital Thermometer. Dostopno na:
<http://datasheets.maximintegrated.com/en/ds/DS18B20.pdf>
- [12] (2013) Photoresistor. Dostopno na:
<http://en.wikipedia.org/wiki/Photoresistor>
- [13] (2012) PIR motion sensors. Dostopno na:
<http://www.ladyada.net/learn/sensors/pir.html>
- [14] (2013) Voltage divider. Dostopno na:
http://en.wikipedia.org/wiki/Voltage_divider
- [15] (2013) Using a Thermistor. Dostopno na:
<http://learn.adafruit.com/thermistor/using-a-thermistor>
- [16] (2013) JW relays. Dostopno na:
http://www.panasonic-electric-works.de/catalogues/downloads/relays/ds.61B08.en_jw.pdf
- [17] (2013) Thermistor. Dostopno na:
<http://en.wikipedia.org/wiki/Thermistor>
- [18] (2013) Infrared Parts Manual, RE200B, FL65, S211FL. Dostopno na:
<http://www.bucek.name/pdf/re200b.pdf>