

SVEUČILIŠTE SJEVER

SVEUČILIŠNI CENTAR VARAŽDIN



ZAVRŠNI RAD br.358/EL/2015

Realizacija daljinskog upravljanja roletom korištenjem Arduino razvojnog sustava

Denis Vaštag

Varaždin, kolovoz 2015.

SVEUČILIŠTE SJEVER
SVEUČILIŠNI CENTAR VARAŽDIN
STUDIJ ELEKTROTEHNIKA



ZAVRŠNI RAD br. 358/EL/2015

**Realizacija daljinskog upravljanja roletom
korištenjem Arduino razvojnog sustava**

Student:

Denis Vaštag, mat.br. 4091/601

Mentor:

Miroslav Horvatić dipl. Ing.

Varaždin, kolovoz 2015.

PREDGOVOR:

Zahvaljujem se mentoru predavaču Miroslavu Horvatiću, dipl. ing. na pomoći prilikom izbora teme završnog rada, na pomoći prilikom odabira literature i materijala, kao i na strpljenju, susretljivosti i želji da mi pomogne prilikom izrade mojeg završnog rada.

Također se zahvaljujem i svim profesorima, kao i stručnim suradnicima u nastavi i vježbama tijekom cijelog studija koji su svoje znanje, volju i motivaciju prenijeli na mene i moj rad.

SAŽETAK:

U ovom radu opisan je sustav za automatsko upravljanje rolo zavjesom pomoću Arduino UNO razvojne platforme. Sustav može raditi u tri moda odnosno može se upravljati zavjesom (motorom) ručno, pomoću fotooptičko osjetila i pomoću vremenskog modula tj. izvršavanje naredbi u zadanom realnom vremenu.

Glavni elementi sustava su Arduino UNO razvojna pločica sa AT mega mikrokontrolerom, optičko osjetilo, koračni motor 28BYJ-48, infracrveni predajnik i prijemnik te Arduino razvojna pločica sa satom realnog vremena (RTC shield).

Detaljno je opisan način spajanja elemenata i programski kod koji to sve objedinjuje u svrsishodan sustav koji na kraju rezultira boljom energetsom učinkovitošću kuće kao i lagodnijim životom u njoj.

Sadržaj:

1. UVOD	1
2. SUSTAV ZA AUTOMATSKO UPRAVLJANJE ROLETAMA	2
3. ARDUINO PLATFORMA	3
3.1 Arduino Uno	3
3.1.1 Serijska komunikacija	4
3.2 Arduino sučelje	6
3.2.1 Arduino programski kod	7
4. IR KOMUNIKACIJA	8
4.1 IR predajnik	9
4.2 IR Prijemnik	12
5. KORAČNI MOTOR	14
5.1 Karakteristike i vrste koračnih motora	15
5.2 Bipolarni i unipolarni koračni motori	17
5.3 Koračni motor 28BYJ-48 i driver ULN2003	18
6. FOTO-OTPORNIK	20
7. RTC SHIELD	22
8. OPIS RADA SUSTAVA	24
8.1 Shema spajanja	24
8.2 Popis elemenata	25
8.3 Ručni mod rada	27
8.4 Automatski mod rada	28
8.5 Vremenski mod rada	30
8.6 Korištene knjižnice	31
8.7 Ispitivanje sustava	32

9. ZAKLJUČAK	34
10. LITERATURA.....	35
11. PRILOZI	38

1.UVOD

Tema ovog završnog rada je: Realizacija daljinskog upravljanja roletom korištenjem Arduino razvojnog sustava.

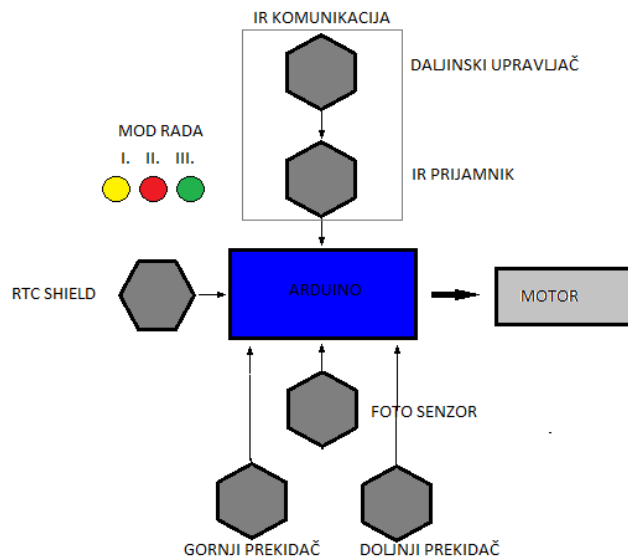
Danas se gotovo u svim segmentima života nastoji olakšati i modernizirati sustavi za rad i život čovjeka. Automatizacija se najprije uvodila u industrijskom i proizvodnom smislu, a danas je posebno zanimljivo kada je koristimo u svome domu.

Ovaj rad odnosno ovaj sustav za daljinsko upravljanje roletom samo je jedan dio automatizacije "inteligentne kuće" no pokazao se kao vrlo važan faktor u poboljšanju energetske učinkovitosti kuće pa čak i zavaravanju provalnika kada ukućana nema duže vrijeme u domu. Pri automatskom načinu rada sustav neće dozvoliti da zrake sunca direktno prodiru u kuću što će rezultirati smanjenjem količine energije potrebne za rashlađivanje kuće dok kasnije pomaže da u prostoriji ne bude prevelika zamračenost i opet smanjuje vrijeme rada rasvjetnih tijela.

Sustavu se može pristupiti na daljinu što omogućuje da ukućani mogu njime jednostavno upravljati.

2. SUSTAV ZA AUTOMATSKO UPRAVLJANJE ROLETAMA

U nastavku je skiciran sustav koji opisuje spajanje elemenata na ulaze Arduina koji potom prema naredbenom kodu upravlja koračnim motorom na izlazu. Motor je spojen na vratilo rolete što rezultira podizanjem ili spuštanjem rolete.



Slika 2.1. Skica sustava za automatsko upravljanje roletom

3. ARDUINO PLATFORMA

Ako danas netko želi započeti sa programiranjem mikrokontrolera gotovo će mu svi preporučiti Arduino i njegovo jednostavno korisničko sučelje.

Arduino platformu je stvorila talijanska tvrtka Smartprojects 2005. godine, a ime je dobio po kafiću u kojem je nastala ideja o njegovom stvaranju. Sastoji se od AVR ATmel mikrokontrolera, ulazno/izlaznih pinova, oscilatora, USB priključka, reset tipke i priključak za napajanje vanjskim istosmjernim (DC) izvorom. Najpoznatija i najzastupljenija verzija je Uno.

3.1 Arduino Uno

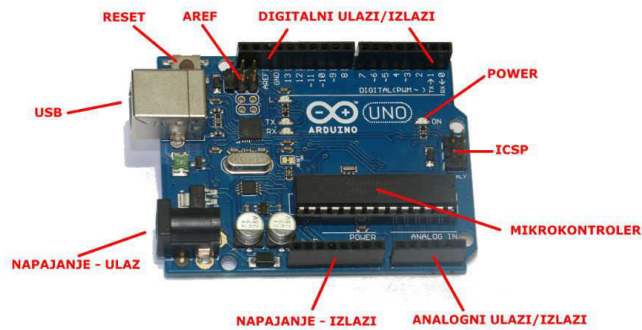
Najpoznatija i najzastupljenija verzija Arduino platforme je Arduino Uno stoga je baš ona korištena u radu.

Pločica sa računalom komunicira preko USB sučelja (serijska komunikacija), te također uzima i napajanje od 5V, ali može se napajati i vanjskim DC izvorom. Mikrokontroler izvodi jedan program koji je zapisan u njegovu Flash memoriju, a u EEPROM-u se čuvaju podaci nakon gašenja uređaja, kao u malom hard disku. Za proširenje Arduina dostupni su brojni shieldovi koji imaju dodatne mogućnosti ili čak svoje vlastite mikrokontrolere [1].

Prikazane su glavne specifikacije osnovnog modela:

- Mikroprocesor: ATmega328
- Frekvencija procesora: 16MHz
- Radni napon: 5V
- Ulazni napon (preporučeno): 7-12V
- Ulazni napon (ograničeno): 6-20V

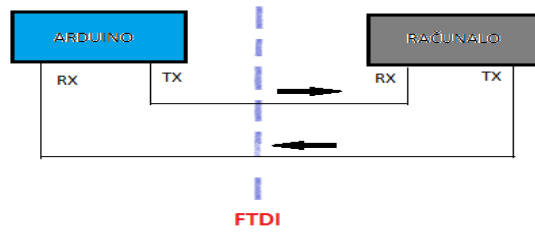
- Digitalni I/O pinovi: 14 (od toga moguće 6 PWM izlaza)
- Analogni ulazni pinovi: 6
- DC struja za I/O pinove: 40mA
- DC struja za 3,3V pin: 50mA
- Flash memorija: 32 KB (0,5KB rezervirano za bootloader)
- SRAM: 2KB
- EEPROM: 1KB



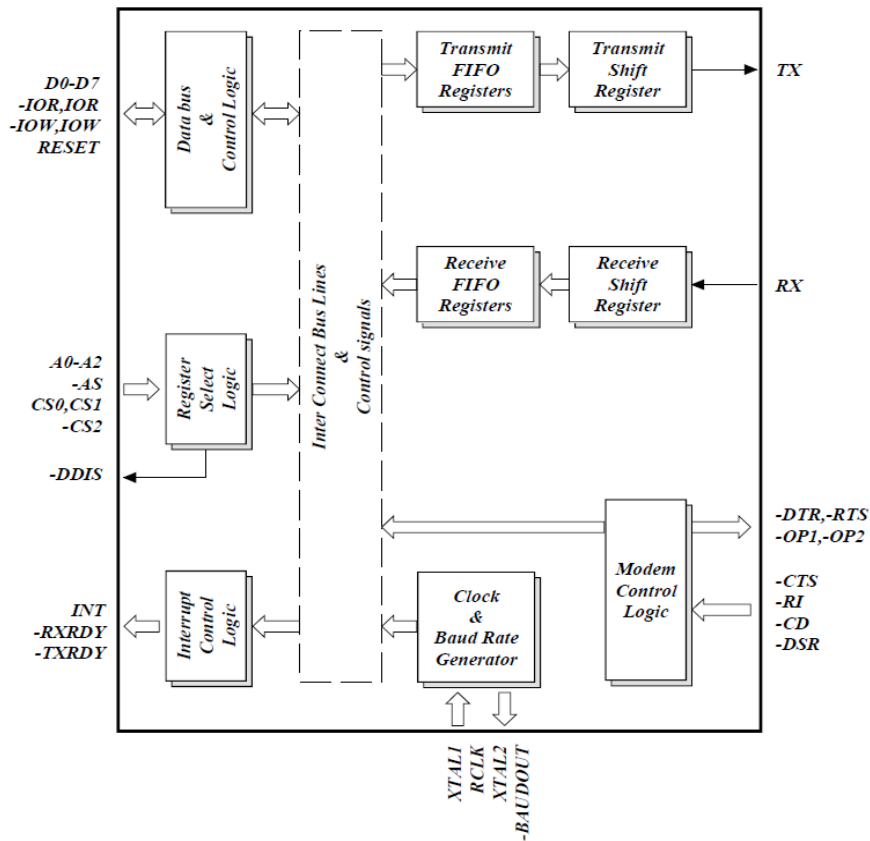
Slika 3.1 Arduino Uno

3.1.1 Serijska komunikacija

Na digitalnim pinovima 0 i 1 nalaze se oznake RX i TX, one služe za međusobnu komunikaciju Arduina i računala na način da TX(Transmit) šalje podatke dok RX(Recive) prima podatke. Budući Arduino radi sa TTL tehnologijom, a ona je za napone od 0 do 5 V, te računalo preko USB-a sa RS-232 (-15 do +15 V) potreban nam je FTDI čip koji omogućuje da se sinkroniziraju. Iz tog razloga ne preporuča se korištenje tih dva pinova osim u iznimnim situacijama.



Slika 3.2 Komunikacija sa računalom

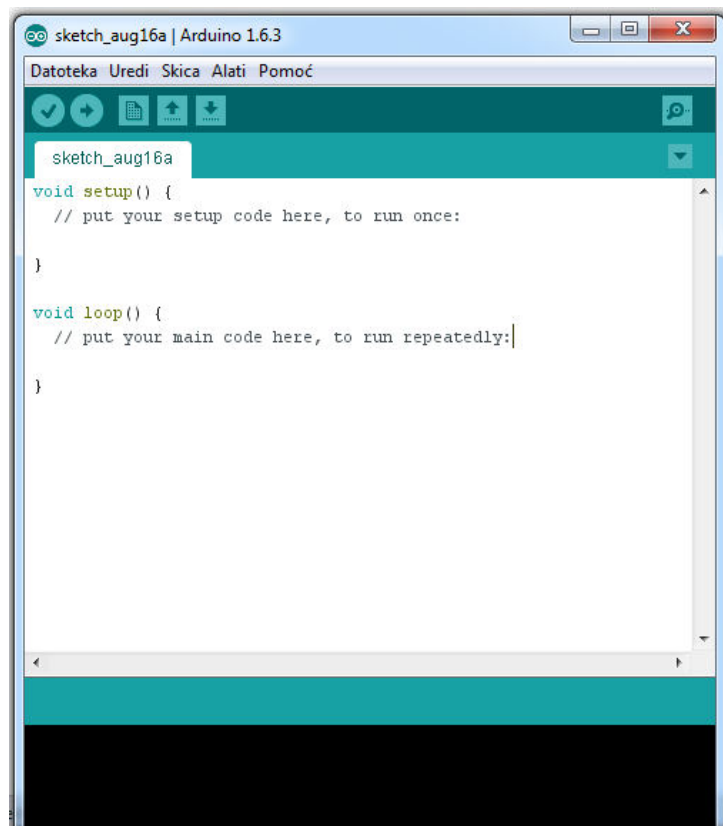


Slika 3.3 Serijska komunikacija [10]

3.2 Arduino sučelje

Arduino programsko sučelje je besplatni program koji nam omogućuje da naše ideje pomoću programskog jezika, koji je kombinacija c/c++ jezika, prenesemo na pločicu i upravljamo hardverom spojenim na Arduino.

U radu je korištena verzija 1.6.3 koja se može preuzeti sa službene stranice Arduino tima. Također se može programirati originalnim AVR alatima.



Slika 3.4 Arduino korisničko sučelje 1.6.3 [1]

3.2.1 Arduino programski kod

Arduino programski kod je ustvari prilagođeni i razumljiviji C/C++ kod te svaki kod mora sadržavati slijedeća dva dijela:

1) void setup(): jednostavno rečeno ovdje definiramo što pinovi rade, da li je pin ulaz ili izlaz odnosno ako je ulaz onda očitava signal, a ako je izlaz onda daje signal tj. struju

Primjer: pinMode (led,OUTPUT); definiramo varijablu led kao izlaznu.

2) void loop(): dio koda koji se izvodi nakon setup dijela i on se izvodi cijelo vrijeme u beskonačnoj petlji. U njemu se nalaze funkcijski dijelovi programa i u tom djelu definiramo što želimo da mikrokontroler radi [2].

Primjer: digitalWrite (led,HIGH); varijabli led šaljem struju(signal)

Vrlo značajna stvar kao i kod drugih programskih alata tako i ovdje jest pozivanje raznih knjižnica (libraries). Knjižnice su skup već predefiniраниh komandi koje nam omogućuju korištenje nekih komponenti. Recimo ako koristimo senzor temperature i vlage (DHT11) nije dovoljno samo ga spojiti na ulaz i očitati stanje, već je potrebno te vrijednosti “provući” kroz zadani niz formula i komandi da bi dobili stvarno i pravilno očitavanje. Budući je postupak uvijek isti on je dostupan u obliku biblioteke na stranicama proizvođača, a nama olakšava i smanjuje duljinu koda te ga pozivamo kada nam je potreban. Uobičajeno je da postoje na desetke knjižnica u samom programu kojima pristupamo na način: datoteka, a zatim primjeri.

U radu su korištene knjižnice za koračni motor, IR komunikaciju, Time shield...

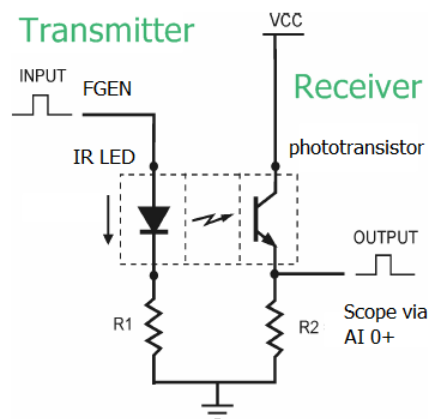
4. IR KOMUNIKACIJA

Infracrvena komunikacija spada među najraširenijim i najjednostavnijim bežičnim komuniciranjem između ljudi i uređaja. Primjenjuje se kod tv, audio, video uređaja, upravljanja klima uređaja, alarmnih sustava itd..

IR svjetlo je vrlo slično vidljivom svjetlu, osim što mu je valna duljina u rasponu od oko 700 do 1 nm, tj. između valnih duljina vidljive svjetlosti i mikrovalova. Infracrveni spektar je nevidljiv ljudskom oku što je idealno za bežičnu komunikaciju. IR komunikacija je jednosmjerna komunikacija kod koje postoje dvije krajnje točke komunikacije (point to point), pri čemu jedna točka predstavlja odašiljač (upravljajući), a druga prijamnik (uređaj). Kako bi komunikacija bila uspješna potrebno je osigurati nesmetan prolaz IR zrakama. IR komunikaciju je moguće ostvariti na udaljenostima od nekoliko metara (5 – 10 m), ovisno o snazi odašiljača.

Za uspješnu komunikaciju između odašiljača i prijemnika zadužen je IR protokol. Postoji nekoliko protokola: SIRC (Sony), RC5 (Phillips), JVC (JVC), NRC17 (Nokia)...

Infracrveno daljinsko upravljanje koristi infracrvene svjetleće diode (IR diode). Zraka se modulira, gasi i pali, da bi se podaci kodirali. Prijemnik koristi silicijevu fotodiodu, da bi pretvorio infracrveno zračenje u električnu struju [3].



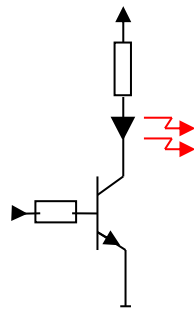
Slika 4.1 IR komunikacija [11]

4.1 IR predajnik

IR predajnik emitira modulirane impulse signala koji predstavljaju odgovarajuće binarne kodove. Predajnik mora trošiti vrlo malo energije kako bi bio što praktičniji u primjeni, stoga najčešće daljinske upravljače napajamo malim baterijama. Kada nije pritisnuta niti jedna tipka, potrošnja struje je toliko mala da je gotovo zanemariva, a procesor će prenijeti odgovarajuću naredbu samo kad je tipka pritisnuta.

Zbog većih vjerojatnosti fizičkih udaraca kvarcni kristali su zamijenjeni keramičkim rezonatorima te unatoč manjom točnošću dobro obavljaju svoju ulogu u radu.

Kao predajnik IR zraka može se koristiti IR LED dioda TSUS5402.



Slika 4.2 Tranzistorski sklop za pogon LED diode [12]

Dimenzije	6.0W x 6.95H x 5.6D
Struja napajanja I_F	100 mA
Napon napajanja	5 V
Temperatura sredine i ambijenta	-40°C do +85°C
Snaga	150 mW
Usmjerenost	45°
Domet prijenosa	2-5 m

Tablica 4.1 Karakteristike diode TSUS5402 [12]

Daljinski upravljač korišten u radu napajan je baterijom od 3V (CR2025) dok struja kroz IR diodu varira od 100mA pa do 1 A, ovisno o daljini koju treba ostvariti.

Signal koji emitira daljinski upravljač koristi pulsno-širinsku modulaciju (PWM). Puls širine 1,2ms predstavlja logičku jedinicu, a puls širine 600 μ s predstavlja logičku nulu. Između slanja logičke nule i jedinice uvijek postoji stanka koja traje 600 μ s. Naredba koju šalje daljinski upravljač sastoji se od 32 bita. Prvih 16 bitova određuje početak nove naredbe, dok zadnjih 16 bitova određuju o kojoj se naredbi radi [5].



Slika 4.3 Daljinski upravljač [13]

Svakom daljinskom upravljaču moguće je očitati njegove kodove pa je moguće jedan daljinski upravljač koristiti za nekoliko uređaja. U slijedećoj tablici dani su kodovi daljinskog upravljača koji je korišten u radu.

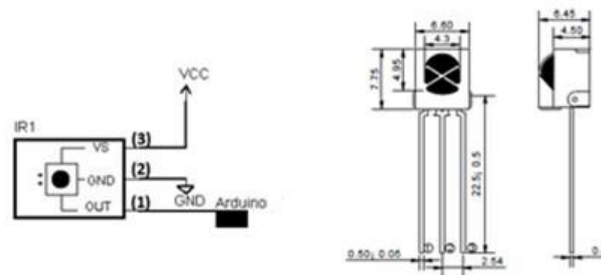
TIPKA	HEKSADECIMALNI ZAPIS	DEKADSKI
CH-	FFA25D	16753245
CH	FF629D	16736925
CH+	FFE21D	16769565
<<PREV	FF22DD	16720605
>> NEXT	FF02FD	16712445
> PLAY/PAUSE	FFC23D	16761405
- VOL-	FFE01F	16769055
+ VOL+	FFA857	16754775
EQ	FF906F	16748655
0	FF6897	16738455
100+	FF9867	16750695
200+	FFB04F	16756815
1	FF30CF	16724175
2	FF18E7	16718055
3	FF7A85	16743045
4	FF10EF	16716015
5	FF38C7	16726215
6	FF5AA5	16734885
7	FF42BD	16728765
8	FF4AB5	16730805
9	FF52AD	16732845

Tablica 4.2 kodovi pojedinih tipki

4.2 IR Prijemnik

Da bi se postigla IR komunikacija odnosno da bi IR predajnik obavio svoju ulogu potreban nam je IR prijemnik. IR prijemnik čita poslane podatke od strane predajnika i šalje ih na određeni ulaz Arduina odnosno bilo kojem sklopu za upravljanje uređajima. Kao IR detektor koristi se foto-tranzistor TSOP2238. Ovi foto-tranzistori su osjetljivi na zračenja u infracrvenom dijelu spektra. Za prijem IR zraka koristi se prijenosna frekvencija od 38kHz. Djelovanje foto-tranzistora slično je djelovanju običnih bipolarnih tranzistora s tim da se struja baze stvara osvjetljavanjem PN spoja baza-kolektor. Foto-tranzistor se može predočiti s pomoću spoja bipolarnog tranzistora i foto-diode spojene između kolektora i baze.

Kad je foto-tranzistor neosvijetljen, njime teče samo vrlo mala „tamna struja“ koju čini preostala struja kolektora, reda veličine od nekoliko nA do nekoliko desetaka nA. Kad se PN spoj baza-kolektor osvijetli, poteče struja baze I_A čija je jakost proporcionalna osvjetljenosti tranzistora pa u kolektorskom krugu poteče struja $I_C = h_{FE} I_A$ [4].



Slika 4.4 IR Prijemnik [14]

Prijenosna frekvencija	38kHz
Dimenzije	6.0W x 6.95H x 5.6D
Napon napajanja	-0.3 do +6 V
Struja napajanja	5 mA
Napon na izlazu	-0.3 V do (Vs + 0,3 V)
Struja na izlazu -	5 mA
Temperatura sredine i ambijenta	-25°C do +85°C
Snaga	10 mW
Usmjerenost	45°
Domet prijenosa	40m
Protokol	RC5

Tablica 4.3 Karakteristika IR prijemne diode TSOP2238 [14]

5.KORAČNI MOTOR

Koračni motori su pretvornici električne energije u mehaničku koji omogućavaju pretvorbu digitalnog podatka u proporcionalni mehanički pomak. Prevladava uporaba rotacijske izvedbe, no proizvode se i koračni motori u translacijskoj izvedbi. Pri uzbudi namotaja po programiranom redosljedu rotor se pomakne u diskretnim koracima, odnosno pokrene u željenom smjeru za predviđeni kut. To su dakle sinkroni motori jer rotor sinkrono prati kretanje statorskog polja zbog sila nastalih međudjelovanjem s poljem rotorskih magneta ili reluktantnih sila. Pozicija rotora se mijenja u koracima jer se statorski namotaji napajaju strujnim impulsima potrebnog rasporeda i polariteta upravljanim pomoću binarnih signala.

Smjer vrtnje se može mijenjati promjenom impulsnog slijeda, brzina rotacije promjenom frekvencije impulsa, a kut (prijeđeni broj koraka) ovisi o ukupnom broju pristiglih impulsa [6].

Rotor se može i držati u željenoj kutnoj poziciji. Koračni motori su nezaobilazni elementi za precizno upravljanje i pozicioniranje pokretnih mehanizama u praktičnim uređajima i sustavima kao što su disketni pogoni, CD/DVD uređaji, printeri, skeneri, fax uređaji, kamere, roboti, elektromotorni ventili, protočne pumpe, CNC strojevi, industrijski ventilatori, te brojni drugi uređaji u domaćinstvu i industriji. Svima njima je zajedničko kontrolirano gibanje mehaničkih pokretnih dijelova. Za široku primjenu zaslužna je njihova prilagodljivost konkretnoj potrebi i relativno jednostavan način njihovog upravljanja [6].

5.1 Karakteristike i vrste koračnih motora

Iz mehaničkog pogleda koračni motor je jednostavna, pouzdana i precizna komponenta koja ne zahtjeva posebno održavanje zahvaljujući konstrukciji bez kolektora i četkica. Uz to karakterizira ih i povoljna cijena, jednostavan dizajn, velika akceleracija i snaga, te veliki raspon brzine vrtnje koja je proporcionalna frekvenciji ulaznih impulsa.

Od nedostataka se može izdvojiti mala korisnost (koračni motori troše značajnu energiju bez obzira na teret), moment znatno opada s povećanjem brzine, sklonost rezonanciji, jako zagrijavanje na zahtjevnim režimima, te niska izlazna snaga s obzirom na veličinu i težinu [5].

Karakteristika	Opis
Rezolucija rotacijskih koračnih motora	broj koraka po okretaju
Rezolucija linearnih koračnih motora	dužina koraka
Odziv jednog koraka	vrijeme potrebno za pomak motora za jedan korak nakon što se napajanje jedne faze isključi, a slijedeće faze uključi
Točnost koračnog motora	stupanj greške zbog konstrukcije
Statički moment	ovisnost uspostavljenog momenta u motoru o pomaku rotora
Dinamički moment	ovisnost srednje vrijednosti momenta u motoru o brzini vrtnje
Start-stop moment	moment tereta s kojim motor može krenuti pri zadanoj frekvenciji koračnih impulsa, a da ne izgubi korak

Tablica 5.1 Karakteristike koračnih motora

Koračni motori se prema svojoj izvedbi mogu podijeliti na više načina, a to su [6]:

→ **Prema načinu stvaranja uzbude**

Elektromagnetska uzbuda

Uzbuda permanentnim magnetima

→ **Prema smještaju uzbude**

Uzbuda na rotoru

Uzbuda na statoru

→ **Prema izvedbi rotora**

Nazubljeni čelični rotor (rotor s varijabilnom reluktancijom)

Rotor sa permanentnim magnetima

Hibridni motori – kombinacija prethodna dva

→ **Prema broju faza – najčešće 2-6 faza**

→ **Prema broju pari polova**

Koračni motori s permanentnim magnetima na rotoru imaju 1 do 4 pari polova

Ostali serijski proizvedeni koračni motori imaju 1 do 90 pari polova

→ **Prema načinu kretanja**

Rotacijski

Translacijski

→ **Prema konstrukciji namotaja odnosno napajanju**

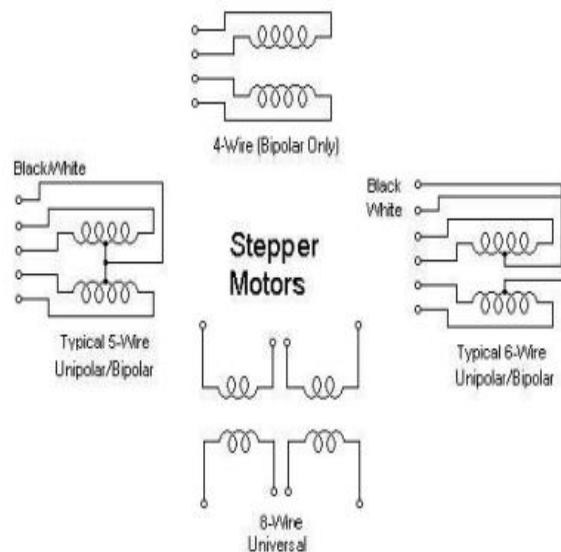
Unipolarni

Bipolarni

5.2 Bipolarni i unipolarni koračni motori

Koračni motor može imati 4, 5, 6 ili 8 priključnih žica. Za razliku od unipolarne izvedbe, statorski namotaji bipolarnog koračnog motora nemaju srednji izvod na pojedinom statorskom namotaju. Umjesto izmjenične uzbude jedne pa druge polovice namotaja za postizanje promjene struje, a time i magnetskog polja, kod bipolarnih je to moguće izvesti samo promjenom vrijednosti napona na oba kraja namotaja.

Koračni motor sa 4 priključne žice je dakle isključivo bipolarni i zahtjeva bipolarni kontroler. Ostali motori se mogu pokretati i unipolarnom i bipolarnom uzbudom jer se srednji statorski izvod može jednostavno zanemariti [5].



Slika 5.1 Različite konstrukcije namotaja koračnih motora [15]

5.3 Koračni motor 28BYJ-48 i driver ULN2003

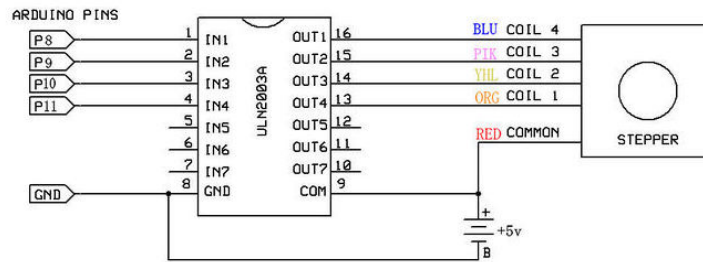
Kod unipolarnih koračnih motora najčešće nalazimo dvije zavojnice sa srednjim izvodom. Svaka zavojnica ima dva krajnja izvoda i srednji izvod. Kod 28BYJ-48 koračnog motora srednji izvodi obje zavojnice spojeni su zajedno pa motor ima 5 izlaznih žica za spajanje kako je prikazano na sljedećoj slici [16]:



Slika 5.2 Motor i zavojnice u motoru [16]

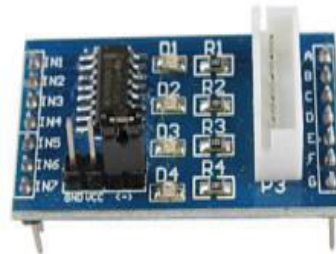
Koračne motore nije moguće spojiti direktno na izvode mikrokontrolera jer je struja potrebna za rad motora znatno veća od one koju može dati jedan izvod mikrokontrolera. Zbog toga se za spajanje mikrokontrolera i koračnog motora koristi integrirani krug ULN2003.

Bilo bi najbolje da je napajanje koračnog motora izdvojeno od napajanja mikrokontrolera, ali obzirom da ovaj koračni motor nema potrebu za velikom strujom pri radu napajati će se sa istog napajanja kao i mikrokontroler pa će se na COM izvod spojiti 5V s Arduino pločice, a na GND izvod ULN2003 integriranog kruga GND izvod s Arduino pločice.



Slika 5.3 Shema spajanja koračnog motora i njegova drivera

Ulazni izvodi 1-7 obilježeni sa IN1, IN2... do IN7 spajaju se na mikrokontroler, dok se izlazni izvodi, obilježeni sa OUT1, OUT2... do OUT7 spajaju na koračni motor. Na izvod GND potrebno je spojiti masu napajanja mikrokontrolera.



Slika 5.4 Pločica sa ULN2003 sklopom [17]

Glavne karakteristike motora su [16]:

- Model : 28BYJ-48
- Napon : 5VDC
- Broj faza : 4
- Brzina varijacije : 1/64
- Omjer kut/korak : 5.625° /64
- Frekvencija : 100Hz
- Otpor : 50Ω±7%(25°C)
- Šum : 35 dB

6.FOTO-OTPORNIK

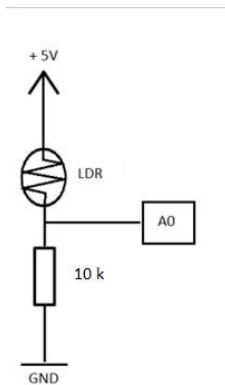
Fotootpornik (eng. photoresistor ili light dependent resistor - LDR) je otpornik čiji se električni otpor smanjuje s povećanjem intenziteta ulazne svjetlosti.

Fotootpornik se izrađuje od poluvodiča sa velikim električnim otporom. Ako svjetlo padne na fotootpornik, sa dovoljno velikom frekvencijom (granična frekvencija), poluvodič će upiti fotone svjetlosti i izbaciti elektrone, koji stvaraju električnu struju, u zatvorenom strujnom krugu.

Poluvodič može biti sa primjesama ili bez. Poluvodiči bez primjesa imaju svoje vlastite nosioce naboja i nisu učinkoviti poluvodiči, kao na primjer silicij. Poluvodiči bez primjesa imaju dostupne elektrone samo u određenom energetskom području i zato fotoni moraju imati dovoljno energije da pobude elektrone u cijelom spektru. Poluvodiči sa primjesama imaju nečistoće u sebi ili primjese, koji povećavaju provodljivost elektrona, tako da fotoni mogu izbaciti elektrone i sa nižom energijom. Siliciju se obično dodaje fosfor, za povećanje električne provodljivosti [7].

Ima mnogo vrsta fotootpornika. Jeftine kadmij sulfid ćelije se mogu naći u puno različitih uređaja, kao što su kamere, ulične svjetiljke, radio satovi, alarmi i vanjski satovi. Olovni sulfid (PbS) i indij antimon (InSb) fotodiode se koriste za srednje infracrveno područje, germanij – bakar (Ge:Cu) fotodiode su najbolje za daleko infracrveno područje i koriste se u infracrvenoj astronomiji i infracrvenoj spektroskopiji.

Foto-otpornici nisu polarizirani, što znači da se mogu spojiti na bilo koju stranu u strujni krug. Foto-otpornik se najčešće spaja serijski sa otpornikom jer takav spoj čini naponsko dijelilo. Sklop se priključuje na istosmjerni napon napajanja iznosa 5V. Izlazni napon U_{iz} dovodi se na analogni ulaz Arduino razvojne platforme. Analogno-digitalnom pretvorbom signala dobiva se brojčana vrijednost u rasponu od 0 do 1023. Ukoliko nema svjetla, foto-otpornik ima vrlo velik otpor, a u mikrokontroleru se dobiva mala brojčana vrijednost. U slučaju da je osvijetljenje najjače, foto-otpornik ima mali otpor pa se u mikrokontroleru dobiva brojčana vrijednost bliska 1023 [8].



Slika 6.1 Naponsko dijelilo

	LDR	R	U
	1 K	10K	0,46
	10K	10K	2,5
	40 K	10K	4

Tablica 6.1 Omjeri otpora nepromjenjivog i fotootpornika te napon izlaza

Otpor foto-otpornika većinom linearno opada s osvjetljenjem, pa se uz stalan napon, može izmjeriti kao linearno povećanje struje kroz otpornik. Pri osvjetljavanju se otpor mijenja nekoliko stotina do nekoliko tisuća puta.

Osjetljivost foto-otpornika varira sa svjetlom valne duljine. Ako je valna duljina izvan određenog raspona, to neće utjecati na otpornost foto-otpornika.

Foto-otpornici imaju manju osjetljivost od foto-dioda i foto-tranzistora. Oni su pasivni elementi kojima nedostaje jedan PN spoj. Ako se intenzitet svjetlosti održava konstantnim, otpor još uvijek može varirati zbog promjene temperature, tako da su foto-otpornici vrlo osjetljivi na temperaturne promjene. Zbog toga oni nisu dovoljno precizni za mjerenje intenziteta svjetlosti.

7. RTC SHIELD

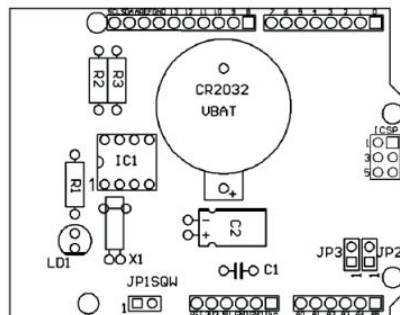
Real time clock shield je jedna od brojnih pločica za nadogradnju Arduina. Arduino ima ugrađen mjeritelj vremena (millis) i timer ugrađen u čip koji može pratiti vrijeme u minutama i danima. Međutim on može pratiti vrijeme samo od njegova pokretanja. Prilikom priključenja napajanja milisekund brojač se postavlja na 0 te nije moguće da Arduino prepozna dane i sate. Zato je potrebno sat odvojiti u zasebnu cjelinu koja ima zadaću mjeriti vrijeme i javljati Arduino točan datum kada je to potrebno.

Time je znatno proširen opseg upotrebe samog Arduina u sustavima za alarme, manjim automatiziranim postrojenjima u kojima je rad ovisan o vremenu i datumu, kao i u sustavima za automatsko upravljanje roletama.

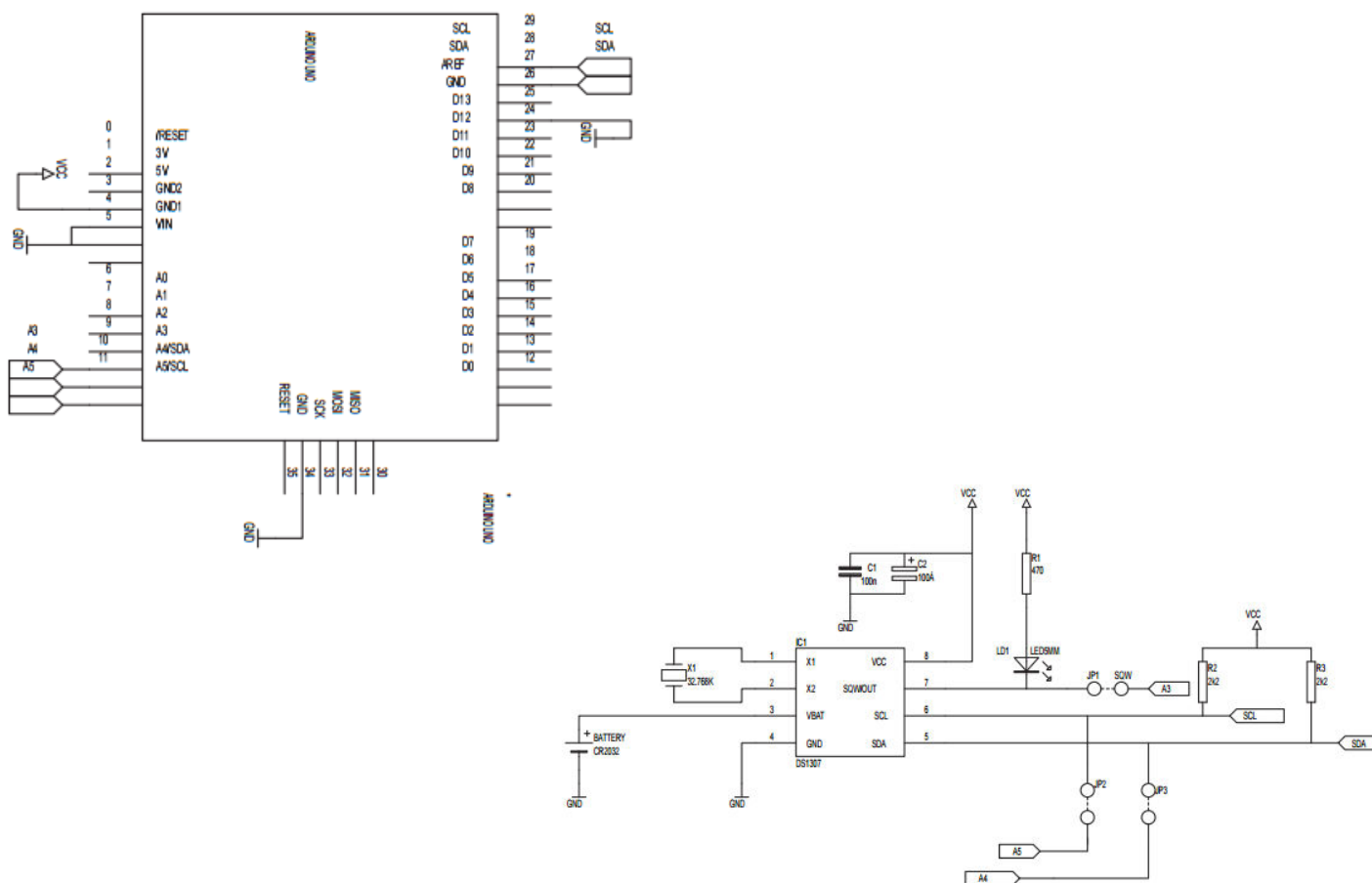
U radu je korišten Velleman-ov KA07 RTC shield. Njegove karakteristike su slijedeće:

- Maxim Dallas DS1307 IC
- brojanje sekundi, sata, dana u tjednu, mjesec, godina i prijestupna godina
- 12 ili 24 satni prikaz
- "prolazni" konektori za mogućnost daljnje nadogradnje shieldovima
- 1.5 mA
- baterija CR2032
- 68x53 mm dimenzije

U prilogu broj 2 se nalazi programski kod za RTC



Slika 7.1 Montažna shema [18]



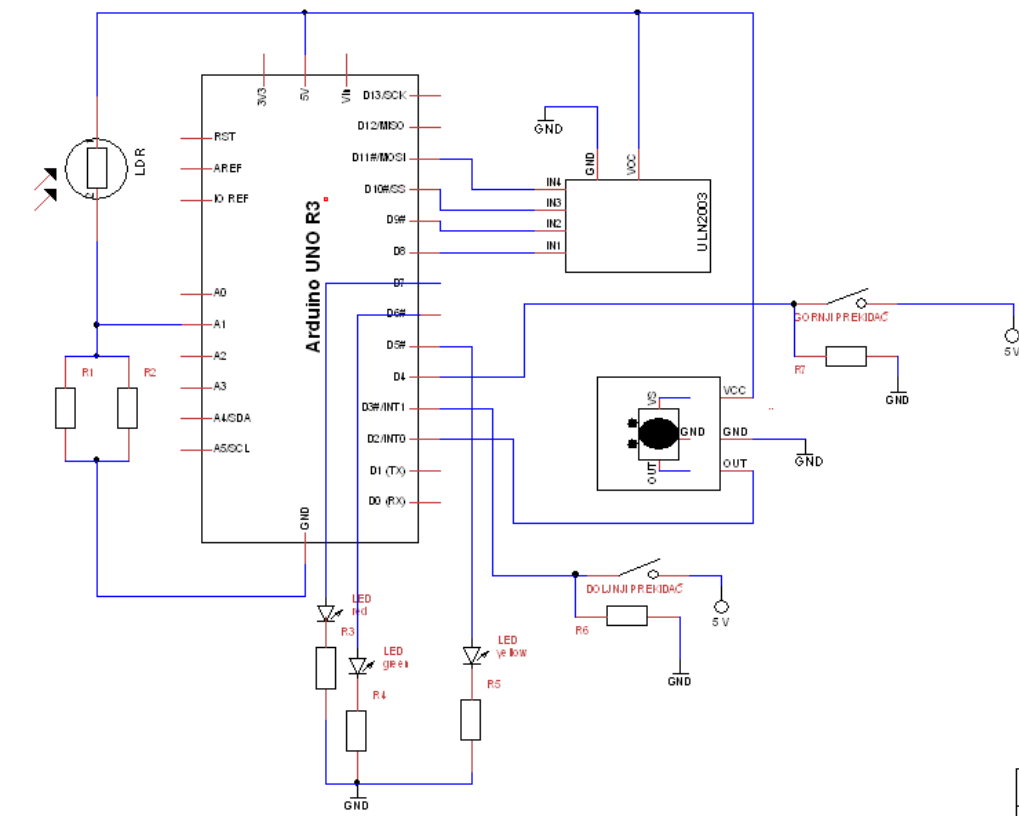
Slika 7.2 Način spajanja RTC na Arduino [18]

8. OPIS RADA SUSTAVA

U nastavku će biti opisan način rada sustava, popis elemenata od kojih se sastoji i programski kod koji ih povezuje.

8.1 Shema spajanja

Radi pouzdanosti i preglednosti sustava elemente je potrebno pričvrstiti te njihove spojeve zalemiti i osigurati na tiskanoj pločici. Kako se radi o vrlo malim iznosima napona i struja, pravilno izvedeni spojevi imaju ključnu ulogu u pouzdanosti sustava. Korištena je pločica dimenzije 16 x 10 cm sa standardno izbušenim rupicama za nožice elemenata.



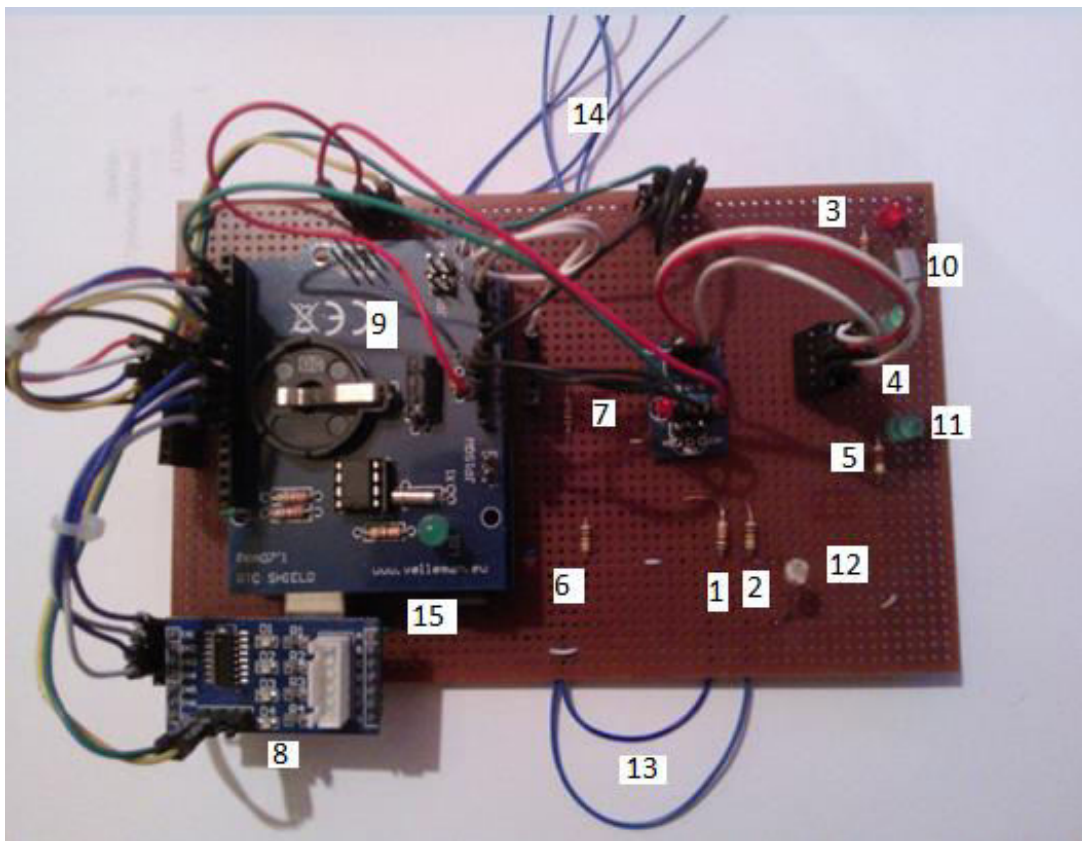
Slika 8.1 Shema tiskane pločice

8.2 Popis elemenata

Na tiskanoj pločici se nalaze slijedeći elementi:

Rb.	Naziv i iznos elementa
1	R1=10 k Ω
2	R2=10 k Ω
3	R3=200 k Ω
4	R4=200 k Ω
5	R5=200 k Ω
6	R6=10 k Ω
7	R7=10 k Ω
8	Driver pločica ULN 2003
9	RTC shield
10	IR prijemnik pločica
11	Svjetleće signalne diode
12	Fotootpornik GL5528
13	Donji mikroprekidač s polugom
14	Gornji mikroprekidač s polugom
15	Arduino UNO R3

Tablica 8.1 Popis elemenata



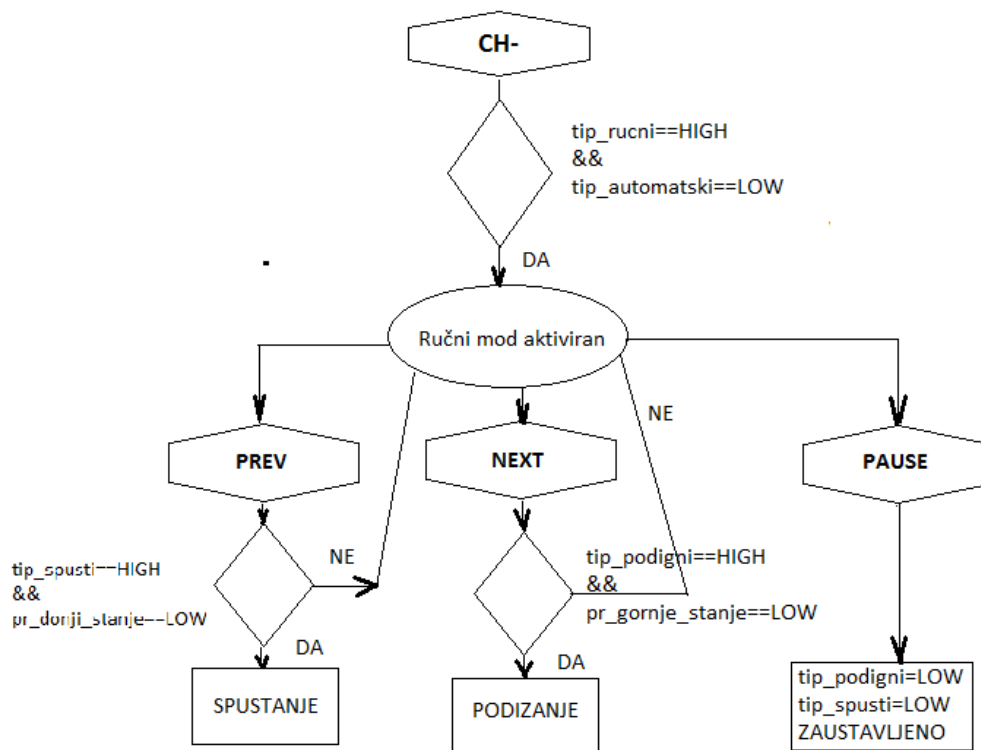
Slika 8.2 Tiskane pločice elektroničkog dijela sustava upravljanja roletama

8.3 Ručni mod rada

Pritiskom tipke CH- na daljinskom upravljaču ulazimo u ručni način rada što i prikazuje uključena signalna LED dioda na pinu 5.

U ovom modu moguće je upravljati roletom u smjeru podizanja ili spuštanja po želji ili do krajnjih prekidača spojenih na pinovima 3 i 4.

Tipkom PREV. Spuštamo roletu sve dok ne pritisnemo tipku PAUSE ili dok roleta ne dođe do krajnje pozicije. Također tipkom NEXT podižemo roletu.



Slika 8.3 Blok dijagram ručnog moda rada

8.4 Automatski mod rada

U ovom načinu rada glavnu ulogu ima fotootpornik spojen na analogni ulaz A1 preko naponskog djelitelja. Vrijednosti od najmanje osvijetljenosti do najveće osvijetljenosti pridružen je brojevima od 0-1023 budući tako Arduino analogne vrijednosti "čita" kao digitalne.

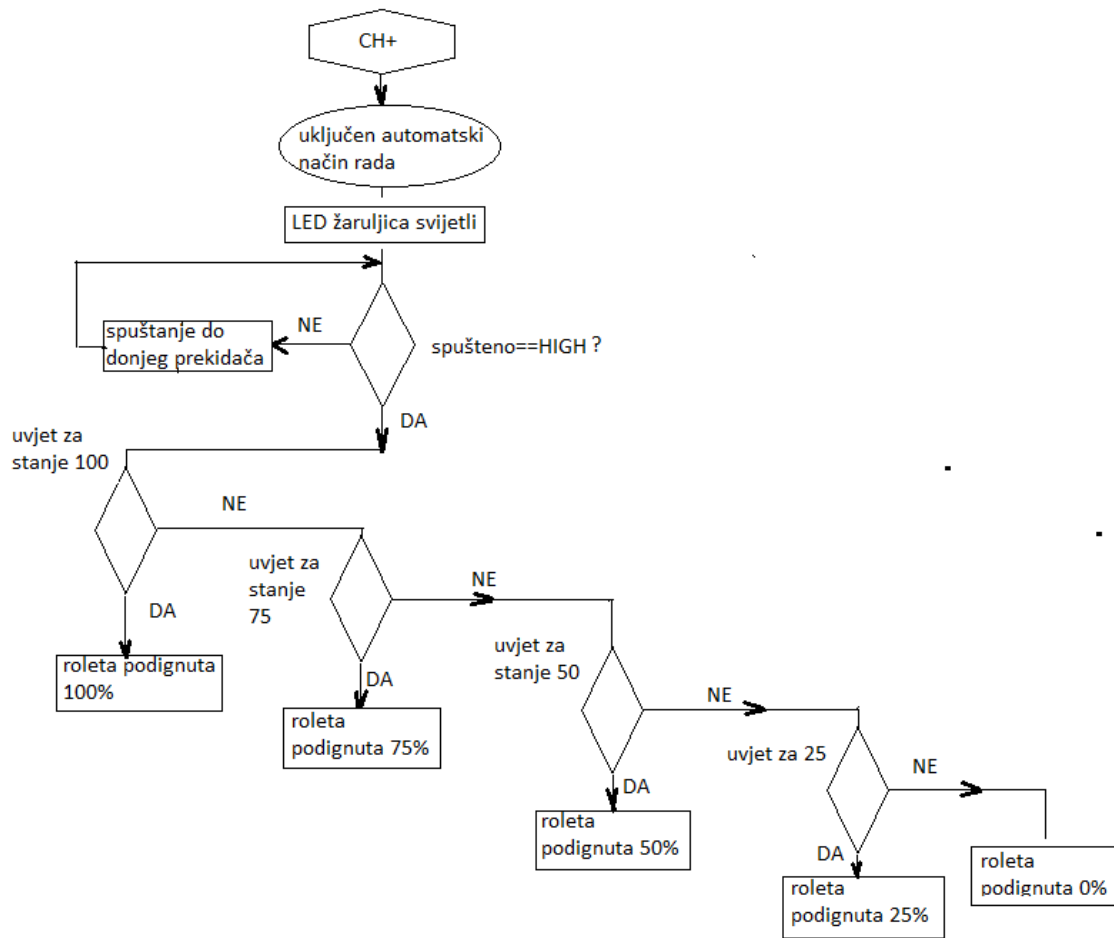
Uključivanje automatskog moda rada postiže se tipkom CH+ na daljinskom upravljaču. Nakon toga ispituje se da li je roleta na najnižoj poziciji, ako nije sustav ju postavlja i za to vrijeme ne može se ništa drugo pritiskati. To je važno jer sustav radi na način da broji korake od nulte pozicije do slijedeće tako da u svakom trenutku zna gdje se roleta nalazi.

Blok shema je pojednostavljena radi veće preglednosti. Stanje 100% otvorenosti rolete dešava se kad je fotosenzor u mraku, njegov otpor je najveći i na analogni ulaz stiže najmanja vrijednost.

Treba voditi računa da se u stanje 100 može doći iz početnog (nultog) stanja, ali i iz stanja 75. U stanje 50 se može doći iz čak 3 prethodna stanja, a to su: početni, stanje 75 i stanje 25.

Primjer uvjeta za stanje 50, dolazak je iz stanja 75:

```
if (aut_stanje == HIGH && (foto_stanje > 400) && (foto_stanje < 600) &&
stanje75 == HIGH && stanje100 == LOW && stanje50 == LOW && stanje25 ==
LOW && stanje0 == LOW)
```



Slika 8.4 Automatski mod rada

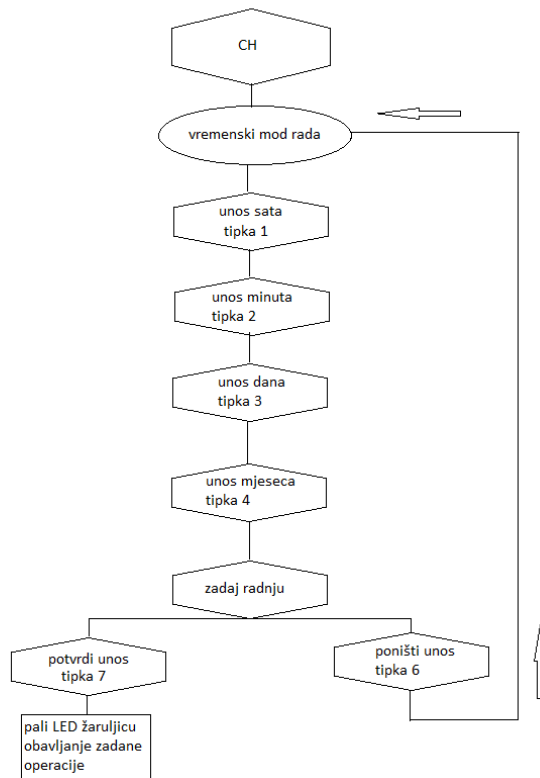
8.5 Vremenski mod rada

Ovaj način rada je vrlo zanimljiv naročito ako nismo u kući, a potrebno je provjetriti ili zatvoriti rolete kada padne noć.

Tipkama od 1 do 4 postavljamo vrijeme u obliku sata: minute: dana: mjeseca te nakon tipke 5 odabiremo radnju za podizanje i spužtanje (prev ili next) zatim tipkom 7 potvrđujemo unos ili tipkom 6 poništavamo i ponovo istim putem unosimo parametre.

Unos funkcionira na način da koliku brojčanu vrijednost trebamo toliko puta pritisnemo odgovarajuću tipku na daljinskom upravljaču.

U radu je samo primjer jednostavnog upravljanja no može se proširiti do vrlo složenih operacija i više dana rada.



Slika 8.5 Vremenski mod rada

8.6 Korištene knjižnice

Kao što je već navedeno biblioteke (knjižnice) koristimo kada radimo sa komponentama koje u svom radu koriste uvijek iste protokole i pravila. Kako ne bismo uvijek pisali programski kod, a vrlo često bi to bilo komplicirano, dovoljno je samo pozvati knjižnicu koja sadrži potrebne komande. Knjižnice uveliko doprinose preglednosti programskog koda i olakšavaju komunikaciju sa komponentama.

Slijede knjižnice koje su korištene u radu:

IRremote.h - biblioteka sa funkcijama za rad sa infracrvenim predajnikom i prijemnikom

- IRrecv irrecv(receivePin)- definira digitalni ulaz mikrokontrolera na koji se priključuje izvod IR prijemnika
- irrecv.enableIRIn() - započinje proces primanja IR signala
- irrecv.decode(&results) - funkcija dekodiranja
- irrecv.resume() - nakon primljenog koda resetira prijemnika

Time.h - biblioteka za rad sa vremenom, ugrađenim ili vanjskim hardverom

- setTime(t) - postavljanje sistemskog vremena
- setTime(hr,min,sec,day,month,yr) - drugi način podešavanja vremena

- `timeStatus()` - označava je li sinkronizirano vrijeme

DS1307RTC.h - biblioteka za rad sa RTC-shieldom

- `RTC.read(tm)` - očitava trenutno vrijeme i datum

Stepper.h - biblioteka za rad sa stepper motorom

- `small_stepper(steps, pin1, pin2, pin3, pin4)` - inicijalizacija step. Motora
- `small_stepper.step(broj)` - pokretanje za točan broj koraka (minus obrnuti smjer)

`Small_stepper.setSpeed(broj)`-postavljanje brzine okretanja

8.7 Ispitivanje sustava

Svaki sustav mora proći testiranje u kojemu će se ispitati ispravnost istog. Najprije se ispita jesu li sve komponente spojene te da na dodir nema "labavih" spojeva.

Nakon toga priključi se napajanje, usb kabelom i provjeri komunicira li IR veza. Budući je baterija nova u daljinskom upravljaču komunikacija je moguća sa udaljenosti veće od 5 metara. Kod ručnog načina rada potrebno je 24 sekunde da se roleta podigne sa najniže u najvišu poziciju te isto toliko u suprotnom smjeru. Pri tome motor napravi 4 puna okreta.

Prilikom zaustavljanja, roleta mirno ostaje u tom položaju te nema proklizavanja.

Kod krajnjih pozicija prekidači vrlo brzo isključuju motor.

Nedostatak je taj što je senzor smješten iznad rolete pa ne detektira u potpunosti svjetlost koja ulazi u prostor.



Slika 8.6 Prototip sustava za automatsko upravljanje roletom

9. ZAKLJUČAK

Ovim završnim radom objedinjen je teorijski i praktični dio s ciljem izrade automatizirane rolete za kućnu upotrebu. Opisani su svi elementi kao i postupci pri izradi sustava. Blok shemama je pobliže opisan princip rada svakog od moguća tri načina rada, a to su ručni, automatski te vremenski način rada.

U ručnom načinu korisnik preko daljinskog upravljača upravlja roletom i podešava ju na određenu visinu. Kod automatskog načina rada korisnik pritiskom gumba uključuje sustav, a zatim sustav sam u ovisnosti o svjetlu podešava roletu na optimalnu visinu. Vremenski način rada funkcionira na način da korisnik podesi vrijeme i datum kada želi podići ili spustiti roletu i zatim može bezbrižno napustiti prostor.

Glavni element sustava je Arduino Uno razvojna pločica koja na svojim izlazima preko drivera upravlja koračnim motorom što rezultira pomicanje rolete u željenim smjerovima.

Sa ekonomskog gledišta ovaj sustav ne zahtjeva velika ulaganja, a poseban značaj ima i energetska učinkovitost koju isti ostvaruje.

10. LITERATURA

- [1] www.arduino.cc (10.06.2015)
- [2] <http://www.pjrc.com> ; Electronic projects (10.06.2015)
- [3] Paolo Zenzerović : Arduino kroz jednostavne primjere
- [4] <http://e-radionica.com/> (10.06.-20.08.2015)
- [5] <http://www.hztk.hr/> (20.06.2015)
- [6] Marin Simčić : Upravljanje koračnim motorima putem Arduino platforme
- [7] IRProtokol:www.arcfn.com/2009/08multi-protocol-infrared-library.html
(10.07.2015)
- [8] Butković Željko : Elektronika 1, Zagreb 2006, skripta FER-a
- [9] Miro Zelić : Sustav za automatsko upravljanje roletama, 2015, FER
- [10] Serijska komunikacija
<https://cdn.sparkfun.com/assets/e/9/7/5/4/50d24680ce395f7172000000.png>
(Dostupno:25.6.2015)
- [11] IR:predajnik:<http://www.next.gr/uploads/58/circuit-configuration.png>
(Dostupno 25.6.2015)
- [12] IR LED : <http://www.vishay.com/docs/81056/tsus5400.pdf>, (Dostupno: 3.7.2015.)
- [13] IRdaljinski upravljač: <http://learning.grobotronics.com/2014/02/ir-control-kit/#>,
(Dostupno: 3.7.2015.)
- [14] IR prijemnik <https://arduino-info.wikispaces.com/IR-RemoteControl>
(Dostupno: 15.06.2015)
- [15] Koračni motor: <http://www.stepperworld.com/Tutorials/pgBipolarTutorial.htm>
(Dostupno:06.06.2015)
- [16] Byj48 <http://www.instructables.com/id/BYJ48-Stepper-Motor/> (Dostupno: 25.06.2015)
- [17] ULN 2003
<http://www.miniinthebox.com/stepper-motor-driver-board-uln2003-for-arduino-5v->

4-phase-5-line_p381455.html (Dostupno: 17.07.2015)

[18] RTC shield

<https://www.guasarelectronics.co.uk/Item/velleman-ka07-rtc-shield-arduino>

(Dostupno 20.07.2015)

POPIS SLIKA:

Slika 2.1 Skica sustava za automatsko upravljanje roletom	2
Slika 3.1 Arduino Uno	4
Slika 3.2 Komunikacija sa računalom	5
Slika 3.3 Serijska komunikacija [10]	5
Slika 3.4 Arduino korisničko sučelje 1.6.3 [1]	6
Slika 4.1 IR komunikacija [11]	8
Slika 4.2 Tranzistorski sklop za pogon LED diode [12]	9
Slika 4.3 Daljinski upravljač [13]	10
Slika 4.4 IR prijemnik [14]	11
Slika 5.1 Različite konstrukcije namotaja koračnih motora [15]	17
Slika 5.2 Motor i zavojnice u motoru [16]	18
Slika 5.3 Shema spajanja koračnog motora i njegova drivera	19
Slika 5.4 Pločica sa ULN2003 sklopom [17]	20
Slika 6.1 Naponsko djelilo	21
Slika 7.1 Montažna shema [18]	22
Slika 7.2 Način spajanja RTC na Arduino [18]	23
Slika 8.1 Shema tiskane pločice	24
Slika 8.2 Tiskane pločice elektroničkog dijela sustava upravljanja roletama. 26	
Slika 8.3 Blok dijagram ručnog moda rada	27
Slika 8.4 Automatski mod rada	29
Slika 8.5 Vremenski mod rada	30
Slika 8.6 Prototip sustava za automatsko upravljanje roletom	33

POPIS TABLICA:

Tablica 4.1 Karakteristike diode TSUS5402 [12].....	10
Tablica 4.2 Kodovi pojedinih tipki.....	11
Tablica 4.3 Karakteristike IR prijemne diode TSOP2238 [14].....	13
Tablica 5.1 Karakteristike koračnih motora.....	15
Tablica 6.1 Omjeri otpora nepromjenjivog i fotootpornika te napon izlaza.....	21
Tablica 8.1 Popis elemenata.....	25

11. PRILOZI

-programski kod
-programski kod RTC Set Time
-cd

PROGRAMSKI KOD:

```
#include <IRremote.h>
#include <IRremoteInt.h>
#include <Stepper.h>
#include <Wire.h>
#include <Time.h>
#include <DS1307RTC.h>
#define STEPPER_KORACI 32

int RECV_PIN = 2;
int tip_spusti = LOW;
int tip_podigni = LOW;
int tip_stani = LOW;
int tip_automatski = LOW;
int tip_rucni = LOW;
```

```
int tip_vremenski = LOW;
const int pr_gornji_Pin = 4;
const int pr_donji_Pin = 3;
const int foto_Pin = A1;
int pr_gornji_stanje = 0;
int pr_donji_stanje = 0;
int foto_stanje = 0;
int aut_stanje = 0;
int stanje100 = 0;
int stanje75 = 0;
int stanje50 = 0;
int stanje25 = 0;
int stanje0 = 0;
int spusteno = 0;
int ir_sati = 0;
int ir_minute = 0;
int ir_dan = 0;
int ir_mjesec = 0;
int ir_potvrdi = 0;

IRrecv irrecv(RECV_PIN);
decode_results results;

Stepper small_stepper(STEPPER_KORACI, 8, 10, 9, 11);
```

```
void setup()
{

    Serial.begin(9600);
    while (!Serial) ; // wait for serial
    delay(200);
    irrecv.enableIRIn();
    small_stepper.setSpeed(500);
    pinMode(pr_gornji_Pin,INPUT);
    pinMode(pr_donji_Pin,INPUT);
    pinMode(foto_Pin,INPUT);
    pinMode(5, OUTPUT);
    pinMode(6, OUTPUT);
    pinMode(7, OUTPUT);
}
```

```
void loop()
{
    if (irrecv.decode(&results))
    {
        translateIR();
        irrecv.resume();
    }
}
```



```

/*    DIO KODA VEZAN UZ RUCNI NACIN RADA
*
*
*/

pr_gornji_stanje = digitalRead(pr_gornji_Pin);
pr_donji_stanje  = digitalRead(pr_donji_Pin);

if (tip_rucni == HIGH && tip_automatski == LOW)
{
    //Serial.println("rucni nacin rada aktiviran");
    if (tip_spusti == HIGH && pr_donji_stanje == LOW)
    {
        Serial.println("spustanje");
        small_stepper.step(32);
    }

    if (tip_podigni == HIGH && pr_gornji_stanje == LOW)
    {
        Serial.println("podizanje");
        small_stepper.step(-32);
    }
}

```

```

/*  DIO KODA VEZAN UZ AUTOMATSKI NACIN RADA
*
*/

    if (spusteno == HIGH){
        aut_stanje = HIGH;
// na pocetku automatskog rada roleta je zatvorena do kraja
    }

    foto_stanje = analogRead(foto_Pin);

/*
*  STANJE 100, MOGUC DOLAZAK IZ 2 STANJA
*/

    if (aut_stanje == HIGH && (foto_stanje < 200) && stanje100 == LOW &&
stanje75 == LOW && stanje50 == LOW && stanje25 == LOW && stanje0 == LOW)
// DOLAZAK IZ POCETNOG STANJA, PODIGNI ZA 1 KRUG ROLETA
OTVORENA DO KRAJA
    {
        Serial.println("ROLETE OTVORENE 100% dolazak iz pocetnog ");
        small_stepper.step(-8384);

        stanje100 = HIGH;
// POSTAVI VARIJABLU TRENUTNOG STANJA U VISOKI POLOZAJ
        delay(100);

```

```

    }

    if (aut_stanje == HIGH && (foto_stanje < 200) && stanje100 == LOW &&
    stanje75 == HIGH && stanje50 == LOW && stanje25 == LOW && stanje0 ==
    LOW)    // DOLAZAK IZ STANJA 75, PODIGNI ZA 25

    {

        Serial.println("ROLETE OTVORENE 100% dolazak iz 75 ");

        small_stepper.step(-2048);

        stanje75 = LOW;
    // VARIJABLU PROSLOG STANJA POSTAVI NA 0

        stanje100 = HIGH;
    // VARIJABLU TRENUTNOG STANJA POSTAVI NA 1

        delay(100);

    }

/*

*   STANJE 75, MOGUC DOLAZAK IZ 3 STANJA

*/

    if (aut_stanje == HIGH && (foto_stanje > 200) && (foto_stanje < 400) &&
    stanje100 == LOW && stanje75 == LOW && stanje50 == LOW && stanje25 ==
    LOW && stanje0 == LOW )    // DOSEL JE IZ POCETNOG STANJA

    {

        Serial.println("ROLETE OTVORENE 75% dolazak iz pocetnog");

        Serial.println(stanje75);

        small_stepper.step(-6336);

        stanje75 = HIGH;
    }

```

```

    delay(100);
}

    if (aut_stanje == HIGH && (foto_stanje > 200) && (foto_stanje < 400) &&
    stanje100 == HIGH && stanje75 == LOW && stanje50 == LOW && stanje25 ==
    LOW && stanje0 == LOW)    // DOSEL JE IZ STANJA 100
    {
        Serial.println("ROLETE OTVORENE 75% dolazak iz 100");
        small_stepper.step(2048);
        stanje75 = HIGH;
        stanje100 = LOW;
        delay(100);
    }

    if (aut_stanje == HIGH && (foto_stanje > 200) && (foto_stanje < 400) &&
    stanje50 == HIGH && stanje100 == LOW && stanje75 == LOW && stanje25 ==
    LOW && stanje0 == LOW)    // DOSEL JE IZ STANJA 50
    {
        Serial.println("ROLETE OTVORENE 75% dolazak iz 50");
        small_stepper.step(-2048);
        stanje75 = HIGH;
        stanje50 = LOW;
        delay(100);
    }

/*
* STANJE 50

```

```
*/
```

```
if (aut_stanje == HIGH && (foto_stanje > 400) && (foto_stanje < 600) &&  
stanje100 == LOW && stanje75 == LOW && stanje50 == LOW && stanje25 ==  
LOW && stanje0 == LOW)
```

```
{
```

```
Serial.println("ROLETE OTVORENE 50% dolazak iz pocetnog ");
```

```
small_stepper.step(-4096);
```

```
stanje50 = HIGH;
```

```
delay(100);
```

```
}
```

```
if (aut_stanje == HIGH && (foto_stanje > 400) && (foto_stanje < 600) &&  
stanje75 == HIGH && stanje100 == LOW && stanje50 == LOW && stanje25 ==  
LOW && stanje0 == LOW)
```

```
{
```

```
Serial.println("ROLETE OTVORENE 50% dolazak iz 75");
```

```
small_stepper.step(2048);
```

```
stanje50 = HIGH;
```

```
stanje75 = LOW;
```

```
delay(100);
```

```
}
```

```
if (aut_stanje == HIGH && (foto_stanje > 400) && (foto_stanje < 600)&&  
stanje25 == HIGH && stanje100 == LOW && stanje75 == LOW && stanje50 ==  
LOW && stanje0 == LOW )
```

```
{
```

```
Serial.println("ROLETE OTVORENE 50% dolazak iz 25 ");
```

```

    small_stepper.step(-2048);

    stanje50 = HIGH;

    stanje25 = LOW;

    delay(100);
}

/*
 * STANJE 25
 */

    if (aut_stanje == HIGH && (foto_stanje > 600) && (foto_stanje < 800) &&
    stanje100 == LOW && stanje75 == LOW && stanje50 == LOW && stanje25 ==
    LOW && stanje0 == LOW)

    {

        small_stepper.step(-2048);

        Serial.println("ROLETE OTVORENE 25% dolazak iz pocetnog ");

        stanje25 = HIGH;

        delay(100);

    }

    if (aut_stanje == HIGH && (foto_stanje > 600) && (foto_stanje < 800) &&
    stanje50 == HIGH && stanje100 == LOW && stanje75 == LOW && stanje25 ==
    LOW && stanje0 == LOW)

    {

        small_stepper.step(2048);

```

```

Serial.println("ROLETE OTVORENE 25% dolazak iz 50");

stanje25 = HIGH;

stanje50 = LOW;

delay(100);

}

if (aut_stanje == HIGH && (foto_stanje > 600) && (foto_stanje < 800) &&
stanje0 == HIGH && stanje100 == LOW && stanje75 == LOW && stanje50 ==
LOW && stanje25 == LOW)

{

small_stepper.step(-2048);

Serial.println("ROLETE OTVORENE 25% dolazak iz 0");

stanje25 = HIGH;

stanje0 = LOW;

delay(100);

}

/*

* STANJE 0

*/

if (aut_stanje == HIGH && (foto_stanje > 800) && stanje100 == LOW &&
stanje75 == LOW && stanje50 == LOW && stanje25 == LOW && stanje0 ==
LOW)

{

small_stepper.step(0);

```

```

Serial.println("ROLETE OTVORENE 0% dolazak iz pocetnog ");

stanje0 = HIGH;

delay(100);

}

if (aut_stanje == HIGH && (foto_stanje > 800) && stanje25 == HIGH &&
stanje100 == LOW && stanje75 == LOW && stanje50 == LOW && stanje0 ==
LOW)

{
    small_stepper.step(2048);

    Serial.println("ROLETE OTVORENE 0% dolazak iz 25 ");

    stanje0 = HIGH;

    stanje25 = LOW;

    delay(100);

}

/*
* DIO PROGRAMA VEZAN UZ VREMENSKI NACIN RADA
*/

if (ir_potvrdi == HIGH){
    tmElements_t tm;

    RTC.read(tm);

    if ( tm.Hour == ir_sati && tm.Minute == ir_minute && tm.Day == ir_dan &&

```



```

tm.Month == ir_mjesecc){
    if (tip_spusti == HIGH && pr_donji_stanje == LOW)
        {
            Serial.println("spustanje");
            small_stepper.step(32);
        }

    if (tip_podigni == HIGH && pr_gornji_stanje == LOW)
        {
            Serial.println("podizanje");
            small_stepper.step(-32);
        }
    }
}
}

int automatski (){

Serial.println("spustanje do doljenjeg prekidaca");

    do{
        small_stepper.step(32);
        pr_donji_stanje = digitalRead(pr_donji_Pin);
    }
    while(pr_donji_stanje == LOW);
    spusteno = HIGH;
}

```

```
Serial.println("roleta spustena do kraja");
```

```
    stanje100 = 0;
```

```
    stanje75 = 0;
```

```
    stanje50 = 0;
```

```
    stanje25 = 0;
```

```
    stanje0 = 0;
```

```
    }
```

```
int sat (){
```

```
    if (tip_vremenski == HIGH){
```

```
        ir_sati++;
```

```
        Serial.println(ir_sati);
```

```
    }
```

```
}
```

```
int minuta (){
```

```
    if (tip_vremenski == HIGH){
```

```
        ir_minute++;
```

```
        Serial.println(ir_minute);
```

```
    }
```

```
}
```

```
int dan (){  
    if (tip_vremenski == HIGH){  
        ir_dan++;  
        Serial.println(ir_dan);  
    }  
}
```

```
int mjesecc (){  
    if (tip_vremenski == HIGH){  
        ir_mjesecc++;  
        Serial.println(ir_mjesecc);  
    }  
}
```

```
int vr_postavke (){  
    if (tip_vremenski == HIGH){  
        Serial.println("ODABRALI STE SAT:");  
        Serial.println(ir_sati);  
        Serial.println("ODABRALI STE MINUTU:");  
        Serial.println(ir_minute);  
        Serial.println("ODABRALI STE DAN:");  
        Serial.println(ir_dan);  
    }  
}
```

```

Serial.println("ODABRALI STE MJESEC:");

Serial.println(ir_mjeseć);

Serial.println("ODABRALI STE NACIN RADA:");

    if (tip_podigni == HIGH && tip_spusti == LOW){
        Serial.println("podizanje");
    }

    if (tip_podigni == LOW && tip_spusti == HIGH){
        Serial.println("spustanje");
    }

}

}

```

```

/* DIO KODA VEZAN UZ PRIMANJE PODATAKA SA IR

```

```

*

```

```

*/

```

```

void translateIR()
{
    switch(results.value)
    {
        case 16753245:

```

```
Serial.println(" RUCNI NACIN RADA ");
tip_rucni = HIGH;
tip_automatski = LOW;
tip_vremenski = LOW;
digitalWrite(5, HIGH);
digitalWrite(6, LOW);
digitalWrite(7, LOW);
break;

case 16736925: +
Serial.println(" VREMENSKI NACIN RADA - UNESITE POSTAVKE ");
tip_vremenski = HIGH;
tip_rucni = LOW;
tip_automatski = LOW;
tip_podigni = LOW;
tip_spusti = LOW;
digitalWrite(5, LOW);
digitalWrite(6, LOW);
break;

case 16769565:
Serial.println(" AUTOMATSKI NACIN RADA ");
tip_automatski = HIGH;
tip_rucni = LOW;
```

```
tip_vremenski = LOW;
tip_podigni = LOW;
tip_spusti = LOW;
automatski();
Serial.println("UKLJUČENO AUTOMATSKO UPRAVLJANJE POMOČU
SVIJETLA!");
digitalWrite(5, LOW);
digitalWrite(6, HIGH );
digitalWrite(7, LOW);
break;

case 16720605:
Serial.println(" SPUSTI ");

tip_spusti = HIGH;
tip_podigni = LOW;
break;

case 16712445:
Serial.println(" PODIGNI ");

tip_podigni = HIGH;
tip_spusti = LOW;
break;
```

```
case 16761405:  
    Serial.println(" STANI ");  
  
    tip_podigni = LOW;  
    tip_spusti = LOW;  
    break;
```

```
case 16724175:  
    Serial.println(" POSTAVI SAT (0-24)");  
    sat();  
    break;
```

```
case 16718055:  
    Serial.println(" POSTAVI MINUTU (0-60)");  
    minuta();  
    break;
```

```
case 16743045:  
    Serial.println(" POSTAVI DAN (1-31) ");  
    dan();  
    break;
```

```
case 16716015:  
    Serial.println(" POSTAVI MJESEC (1-12) ");  
    mjesec();
```

```

break;

case 16726215:
    Serial.println(" POSTAVI NACIN RADA SPUSTI/PODGNI ");
break;

case 16734885:
    Serial.println(" PONISTI VREMENSKE POSTAVKE ");
    ir_sati = 0;
    ir_minute = 0;
    ir_dan = 0;
    ir_mjesec = 0;
break;

case 16728765:
    Serial.println(" POTVRDILI STE VREMENSKE POSTAVKE-UKLJUCEN
VREMENSKI NACIN RADA! ");
    ir_potvrdi = HIGH;
    vr_postavke();
    digitalWrite(7,HIGH);
break;

//    default:

```



```
//      Serial.println(".");  
    }  
    delay(1);  
}
```

RTC SET TIME

```
#include <Wire.h>  
#include <Time.h>  
#include <DS1307RTC.h>  
  
const char *monthName[12] = {  
    "Jan", "Feb", "Mar", "Apr", "May", "Jun",  
    "Jul", "Aug", "Sep", "Oct", "Nov", "Dec"  
};  
  
tmElements_t tm;  
  
void setup() {  
    bool parse=false;  
    bool config=false;
```

```

// get the date and time the compiler was run
if (getDate(__DATE__) && getTime(__TIME__)) {
    parse = true;
    // and configure the RTC with this info
    if (RTC.write(tm)) {
        config = true;
    }
}

Serial.begin(9600);
while (!Serial) ; // wait for Arduino Serial Monitor
delay(200);
if (parse && config) {
    Serial.print("DS1307 configured Time=");
    Serial.print(__TIME__);
    Serial.print(", Date=");
    Serial.println(__DATE__);
} else if (parse) {
    Serial.println("DS1307 Communication Error :-{");
    Serial.println("Please check your circuitry");
} else {
    Serial.print("Could not parse info from the compiler, Time=\");
    Serial.print(__TIME__);
    Serial.print("\", Date=\");

```

```

        Serial.print(__DATE__);
        Serial.println("\n");
    }
}

void loop() {

bool getTime(const char *str)
{
    int Hour, Min, Sec;

    if (sscanf(str, "%d:%d:%d", &Hour, &Min, &Sec) != 3) return false;
    tm.Hour = Hour;
    tm.Minute = Min;
    tm.Second = Sec;
    return true;
}

bool getDate(const char *str)
{
    char Month[12];
    int Day, Year;
    uint8_t monthIndex;

```

```
if (sscanf(str, "%s %d %d", Month, &Day, &Year) != 3) return false;
for (monthIndex = 0; monthIndex < 12; monthIndex++) {
    if (strcmp(Month, monthName[monthIndex]) == 0) break;
}
if (monthIndex >= 12) return false;
tm.Day = Day;
tm.Month = monthIndex + 1;
tm.Year = CalendarYrToTm(Year);
return true;
}
```