Realizacija rotirajućeg LED sata primjenom Arduino mikrokontrolerske platforme

Šeketa, Dino

Undergraduate thesis / Završni rad

2021

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Mechanical Engineering and Naval Architecture / Sveučilište u Zagrebu, Fakultet strojarstva i brodogradnje**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:235:819755

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-04-18**

Repository / Repozitorij:

Repository of Faculty of Mechanical Engineering and Naval Architecture University of Zagreb

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

ZAVRŠNI RAD

Dino Šeketa

Zagreb, 2021.

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

ZAVRŠNI RAD

Mentor: Student:

Prof. dr. sc. Danijel Pavković, dipl. ing. Dino Šeketa

Zagreb, 2021.

Izjavljujem da sam ovaj rad izradio samostalno koristeći znanja stečena tijekom studija i navedenu literaturu.

Zahvaljujem se svojem mentoru, prof. dr. sc. Danijelu Pavkoviću i komentoru dr. sc. Matiji Krznaru na pruženoj pomoći tijekom izrade rada. Također se zahvaljujem svojim roditeljima, prijateljima i djevojci Dori na neizmjernoj podršci tijekom studiranja te kolegi Matiji Hercegu-Rušecu na korisnim savjetima.

Dino Šeketa

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE
Središnje povjerenstvo za završne i diplomske ispite
Povjerenstvo za završne ispite studija strojarstva za smjerove:
proizvodno inženjerstvo, računalno inženjerstvo, industrijsko inženjerstvo i menadžment, inženjerstvo materijala i mehatronika i robotika

Sveučilište u Zagrebu Fakultet strojarstva i brodogradnje	
Datum	Prilog
Klasa: **602 - 04 / 21 - 6 / 1**	
Ur.broj: **15 - 1703 - 21 -**	

ZAVRŠNI ZADATAK

Student: **Dino Šeketa** Mat. br.: 0035213325

Naslov rada na hrvatskom jeziku: **Realizacija rotirajućeg LED sata primjenom Arduino mikrokontrolerske platforme**

Naslov rada na engleskom jeziku: **Realization of a rotating LED clock by means of Arduino microcontroller platform**

Opis zadatka:

Iako je danas digitalna tehnika u većoj mjeri zamijenila analognu, potonja je daleko prirodnija ljudskoj percepciji te analogna rješenja još uvijek pokrivaju specijalizirane niše. Ovo je u posljednje vrijeme vidljivo i u tehnici izrade mjerača vremena (satova), gdje se prikaz vremena pokušava prilagoditi ljudskoj percepciji polazeći od ergonomskih načela. Jedan od takvih primjena je i LED sat, a koji koristi analogni prikaz putem svjetlosnih (LED) dioda uz interno digitalno mjerenje vremena. U radu je potrebno napraviti sljedeće:

1. Proučiti relevantnu dokumentaciju komercijalnih rješenja različitih izvedbi LED satova i napraviti kratki pregled dostupnih pristupa, rješenja i tehnologija.
2. Osmisliti potrebne elektroničke i mehatroničke sustave, te elektromotorne servopogone pomoću kojih se može realizirati analogno sučelje prema korisniku, odnosno kazaljke s LED osvjetljenjem. Također treba osmisliti algoritam upravljanja navedenim komponentama sustava LED sata.
3. Realizirati elektroničko sklopovlje koje uključuje Arduino mikrokontroler i prilagodbene elektroničke sklopove, te mehatronički sustav kazaljki pogonjenih elektromotorima.
4. Realizirati softversko okruženje za upravljanje sustavom LED sata temeljeno na Arduino razvojnom okruženju, te konačno ispitati funkcionalnost navedenih softverskih i hardverskih rješenja.

U radu, također, treba navesti korištenu literaturu i eventualno dobivenu pomoć.

Zadatak zadan:
30. studenoga 2020.

Zadatak zadao:
Danijel Pavković
Izv. prof. dr. sc. Danijel Pavković

Datum predaje rada:
1. rok: 18 veljače 2021.
2. rok (izvanredni): 5. srpnja 2021.
3. rok: 23. rujna 2021.

Predviđeni datumi obrane:
1. rok: 22.2. – 26.2.2021.
2. rok (izvanredni): 9.7.2021.
3. rok: 27.9. – 1.10.2021.

Predsjednik Povjerenstva:
Prof. dr. sc. Branko Bauer

SADRŽAJ

SADRŽAJ I
POPIS SLIKA II
POPIS TABLICA III
POPIS TEHNIČKE DOKUMENTACIJE IV
SAŽETAK V
SUMMARY VI
1. UVOD 1
2. PRINCIP RADA ROTIRAJUĆEG LED SATA 2
3. ODABIR KOMPONENATA 6
3.1. Arduino Nano 6
3.2. Real time clock PCF85063A 8
3.3. Hall efekt senzor A3144 9
3.4. Istosmjerni motor i ispravljač 10
3.5. Odabir ostalih komponenata 11
4. KONSTRUKCIJA SATA, IZRADA I SKLAPANJE 13
4.1. Konstrukcija kućišta s motorom i nosačem PCB-a 13
4.2. Električna shema sata 16
4.3. Izrada PCB pločice 17
4.4. Izrada konstruiranih pozicija 18
4.5. Sklapanje 18
5. PROGRAMSKI KOD 20
5.1. Arduino IDE 20
5.2. Deklaracija varijabli 21
5.3. Setup 22
5.4. Petlja 23
5.5. Očitavanje signala s Hall-ovog senzora 27
6. KORISNIČKO SUČELJE 29
7. TESTIRANJE SATA 31
8. ZAKLJUČAK 32
LITERATURA 33
PRILOZI 34

POPIS SLIKA

Slika 1. Realizacija LED sata korištenjem 7 - segmentnog displeja [8] 2
Slika 2. Realizacija LED sata korištenjem matričnog displeja [9] 3
Slika 3. Analogni zidni sat s kazaljkama [10] 3
Slika 4. Prikaz rasporeda dioda u krugu 4
Slika 5. Zamišljeno rješenje sata 5
Slika 6. Arduino Nano [1] 6
Slika 7. Električna shema Arduino Nano mikrokontrolera [11] 7
Slika 8. Prikaz osnovnih dijelova sklopa – Arduino Nano 7
Slika 9. Prednja strana (lijevo) i stražnja strana (desno) RTC modula [3] 8
Slika 10. A3144 senzor s rasporedom nožica [12] 9
Slika 11. Simbolička shema senzora [2] 9
Slika 12. Karakteristika odabranog motora [5] 10
Slika 13. Odabrani istosmjerni motor [5] 11
Slika14. Korištena LED dioda [13] 11
Slika 15. SL321225 baterija [14] 12
Slika 16. Prekidač [15] 12
Slika 17. Motor s montiranim prstenom 13
Slika 18. Motor s navučenom prirubnicom i produžetkom osovine 14
Slika 19. Valjak za osiguravanje okomitosti nosača PCB-a i baze 14
Slika 20. Model sklopa sata 15
Slika 21. Simbolička shema sklopa LED sata 16
Slika 22. Električna shema LED sata 17
Slika 23. PCB pločica sprijeda 17
Slika 24. PCB pločica straga 18
Slika 25. Sklopljeni sat 19
Slika 26. Arduino IDE sučelje 20
Slika 27. Korisničko sučelje nakon pokretanja 29
Slika 28. Sučelje nakon uspješnog slanja podataka 30
Slika 29. Testiranje sata 31

POPIS TABLICA

Tablica 1. Arduino Nano – tehničke karakteristike..... 6
Tablica 2. Karakteristike RTC modula 8
Tablica 3. Karakteristike RS-365-SA motora pri maksimalnoj iskoristivosti 10

POPIS TEHNIČKE DOKUMENTACIJE

P-0000-00 Rotirajući LED sat

SAŽETAK

Ovaj rad prikazuje princip rada i postupak izrade prototipa rotirajućeg LED (engl. Light Emmiting Diode) sata temeljenog na Arduino platformi korištenjem Arduino Nano mikrokontrolera. Prvo je prikazan odabir korištenih komponenata i njihove glavne karakteristike. Zatim je prikazano konstrukcijsko rješenje kućišta sata i shema spajanja elektroničkih komponenata, postupak njihovog spajanja i način sklapanja sata. Uz sve to napravljen je i detaljan pregled programskog koda s objašnjenjem funkcije svakog zasebnog bloka u kodu. Na kraju, prikazan je postupak izrade korisničkog sučelja koje omogućuje konfiguraciju sata te rezultat testiranja sklopa. U izradi rada korišteni su programski paketi Solidworks, Arduino, Fritzing i Python.

Ključne riječi: LED sat, Arduino, rotirajući sat

SUMMARY

This paper shows the principle of operation and development of rotating LED clock prototype based on Arduino platform using Arduino Nano microcontroller. Firstly, the chosen components and their main characteristics are shown. Second part of the paper shows the design solutions of the clock's casing and scheme of connections of electronic components. Furthermore, it is shown how all components are connected and how everything is assembled together. Complete Arduino code listing is also provided with detailed explanation of every block of code. Creation of user interface for configuration of the clock is shown in the end together with the results of testing of the clock. Solidworks, Arduino, Fritzing and Python softwares were used while working on this project.

Key words: LED clock, Arduino, rotating clock

1. UVOD

Usprkos stotinama tisuća godina evolucije, ljudsko oko i dalje sadrži određene nesavršenosti. Kao i kod analogno – digitalne pretvorbe, ljudski mozak putem senzora (oka) uzorkuje slike iz okoline u otprilike jednakim intervalima. Taj interval iznosi 1/15 do 1/10 sekunde. Tijekom tog intervala, mozak zadržava posljednju uzorkovanu sliku koju zatim pretvara u informaciju kako bi čovjek mogao razumjeti svoju okolinu i tek tada uzorkuje sljedeću sliku. Možemo zaključiti da, ukoliko se nešto odvija u ciklusima koji traju kraće od navedenog intervala, mozak neće moći u potpunosti razumjeti što se događa, već će stvoriti vlastitu percepciju samog događaja koja se najčešće temelji na posljednjoj uzorkovanoj slici. Ako imamo događaj koji se tijekom svojeg izvođenja prekida i ponovo pokreće u intervalima kraćim od 1/15 – 1/10 sekundi, ljudsko oko neće vidjeti te prekide već će se stvoriti efekt kontinuiteta. Ovu nesavršenost oka najbolje je iskoristila filmska industrija čiji su se projektori u samim počecima temeljili na izmjeni sličica tiskanih na filmskim trakama čije brze izmjene gledatelji nisu mogli primijetiti već su vidjeli sliku u pokretu. Također, većina današnjih animacija i video materijala koji se projiciraju na ekranima računala, televizora i mobilnih uređaja koriste isti princip, samo uz pomoć elektroničke i programske potpore. [4]

U ovom radu, taj princip biti će iskorišten u realizaciji rotirajućeg LED sata. Uz pomoć elektromotornog pogona, jednodimenzionalna linija svijetlećih dioda biti će pretvorena u dvodimenzionalni prikaz sata s potpuno funkcionalne sve tri kazaljke. Cijeli sklop temeljiti će se na Arduino platformi.

Ovaj naizgled jednostavan sklop svojim proširenjem može pronaći primjenu u različitim vrstama svjetlosne signalizacije, ali i u reklamnoj industriji.

Obzirom da objedinjuje znanje elektromotornih pogona, elektronike, programiranja i konstruiranja, odličan je projekt za studente mehatronike i robotike kako bi obuhvatio mnoga znanja stečena tijekom studija.

2. PRINCIP RADA ROTIRAJUĆEG LED SATA

Najčešće izvedbe svijetlećih satova podrazumijevaju korištenje 7 – segmentnih ili matričnih prikaznika (engl. display). Brojke na satu sa 7 – segmentnim prikaznikom izvedene su na način da je svaka brojka sastavljena od 7 crtica, a na matričnom displeju je veliki broj dioda posložen u matricu. Raspored uključenih dioda mijenja se svake sekunde, minute ili sata i na taj način se kreiraju brojevi i prikazuje vrijeme. Takva izvedba obično zahtijeva veći broj LED dioda i poprilično je nefleksibilna. Također, izmjena neispravnih dioda može biti relativno kompleksna i česta obzirom na njihov broj.

Unatoč tome, konvencionalna rješenja poprilično su jednostavna, posebno za programiranje, obzirom da svaki prikazani broj je ništa više nego određena kombinacija uključenih dioda. Uz primjenu jednostavne petlje, potrebno je tek nekoliko desetaka linija koda.

Slika 1. Realizacija LED sata korištenjem 7 - segmentnog prikaznika [8]

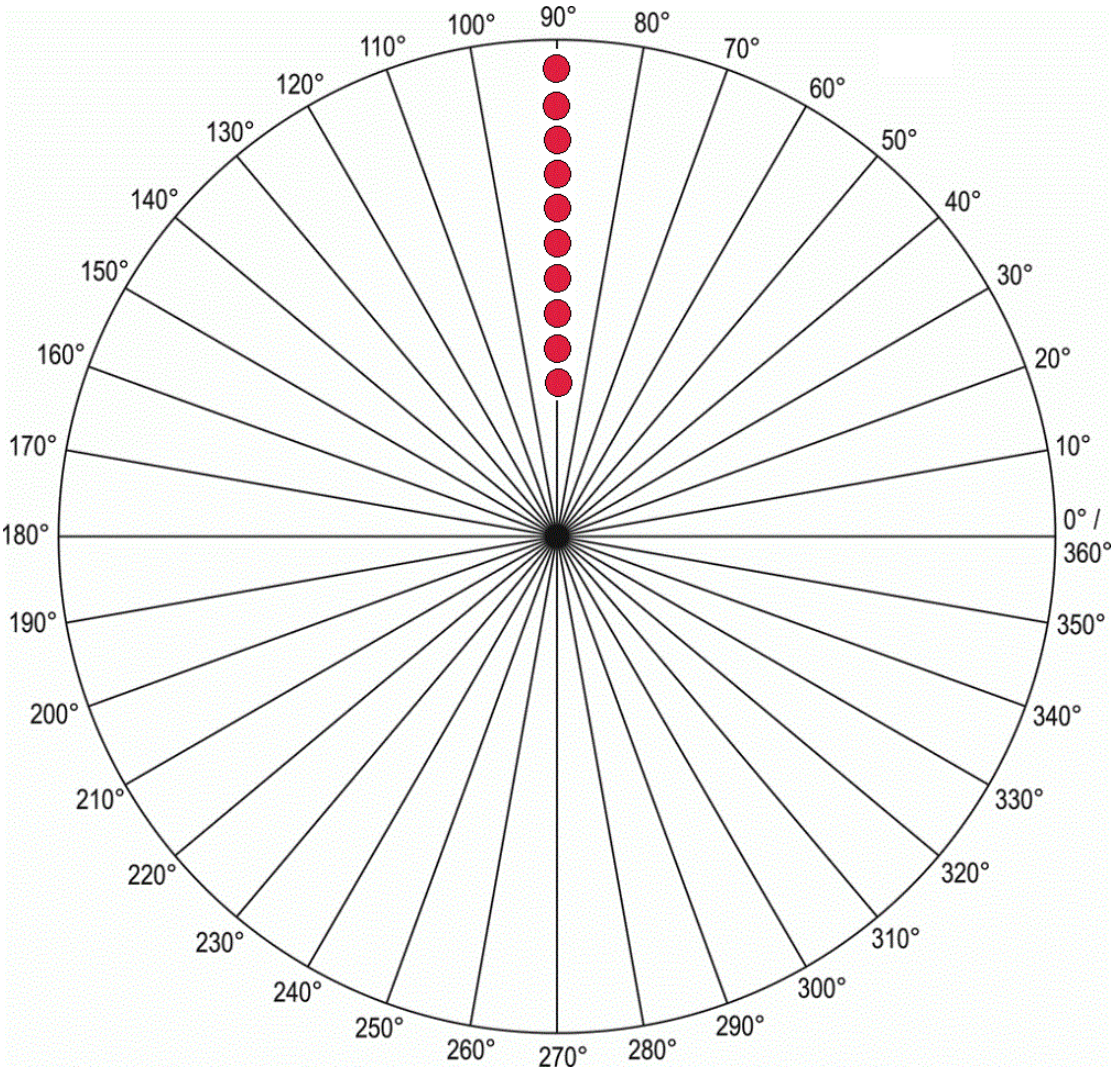
Slika 2. Realizacija LED sata korištenjem matričnog prikaznika [9]

Sat prikazan u ovom radu izveden je na potpuno drugačiji način. Prvenstveno, izvedba podsjeća na konvencionalne analogne satove koji se sastoje od kružnice na kojoj je položeno 12 brojeva s tri kazaljke koje prikazuju sate, minute i sekunde. Na PCB (engl. Printed Circuit Bord) pločicu zalemljeno je niz LED dioda u obliku ravne linije. Uz diode, na pločici se nalaze sve komponente potrebne za rad sata, a koje uključuju mikrokontroler, RTC (engl. Real Time Clock) modul i jednu bateriju za pokretanje mikrokontrolera. Sama pločica spojena je vijcima s nosačem PCB-a kojeg pokreće istosmjerni motor. Istosmjerni motor vijcima je preko prirubnice pričvršćen za nepomičnu bazu i napaja se preko ispravljača iz utičnice.

Slika 3. Analogni zidni sat s kazaljkama [10]

Cilj je, koristeći principe iz prvog poglavlja, uključivati diode u točno određenom dijelu svakog kruga kojeg nosač PCB-a opiše. Kako bi znali u kojem trenutku treba aktivirati određene diode, pozicija nosača u svakom trenutku mora biti poznata. To je realizirano uz pomoć mjerenja vremena potrebnog nosaču da opiše puni krug. Izmjereno vrijeme se zatim dijeli na 360 dijelova što daje točnu informaciju kada će nosač biti u kojoj poziciji unutar punog kruga s preciznošću od jednog stupnja. Teoretski, ako se pomnoži broj dioda s rezolucijom od 360 dijelova, dobit će se broj dioda prikaznika kojeg ovakva izvedba fiktivno predstavlja.

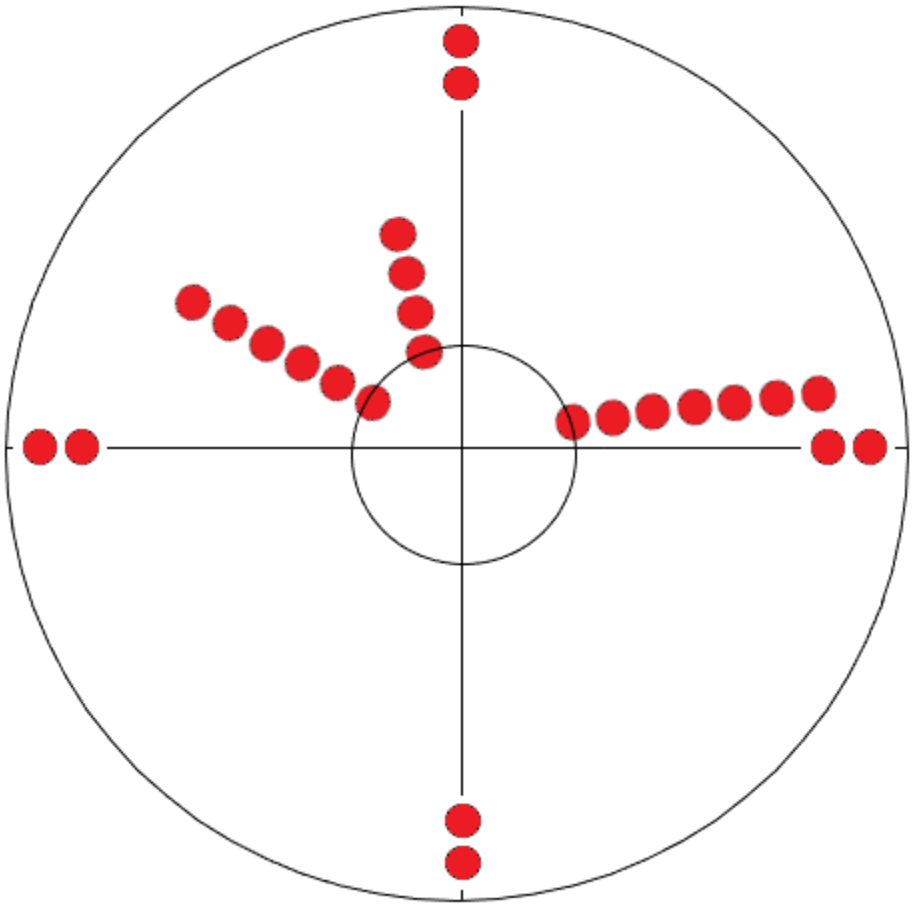
Može se zaključiti da ovakva izvedba prikaznika sa samo 10 dioda aproksimira displej sastavljen od 3600 dioda! Realizacija toga dakako uvelike ovisi o brzini mikroprocesora i brzini vrtnje, ali i o promjeru dioda.



Slika 4. Prikaz rasporeda dioda u krugu

Mjerenje vremena potrebnog da nosač opiše puni krug izvedeno je pomoću Hall-ovog senzora i magneta. Naime, senzor se nalazi na nosaču, a na nepomičnoj bazi se nalazi magnet. Prolaskom senzora uz magnet, na njemu se pojavljuje signal koji zatim biva očitan od strane mikrokontrolera putem prekidne rutine. Programskim kodom određeno je mjerenje vremena koje je proteklo između pojave dva impulsa na senzoru. Ono što je dobro kod ovog rješenja jest da brzina rotacije ne mora biti u potpunosti konstantna zato što se vrijeme potrebno za jedan krug ažurira svakim novim krugom.

Implementiranjem programskog koda cilj je dobiti funkcionalne kazaljke koje pokazuju sate, minute i sekunde. Također, potrebno je dobiti četiri crtice na kružnici koje će predstavljati položaj sata svaka 3 sata, odnosno točke u kojima će sat pokazivati 3, 6, 9 i 12 sati kako bi snalaženje bilo olakšano. Kazaljka sata biti će prikazana s tri, kazaljka minuta sa šest, a kazaljka sekunda sa sedam dioda. Četiri crtice biti će prikazane s po dvije diode. Zamišljeno rješenje prikazano je na slici 3.



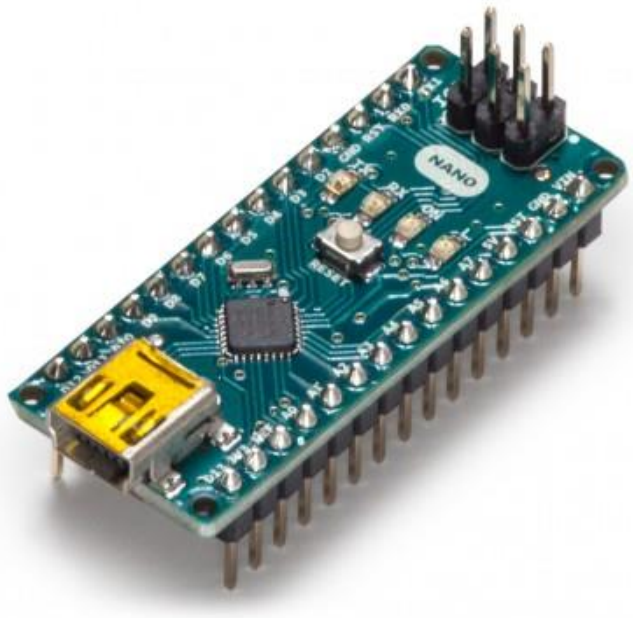
Slika 5. Zamišljeno rješenje sata

Može se primijetiti kako je u zamišljenom rješenju posloženo devet, a ne deset dioda. To je zato što će se posljednja dioda u nizu koristiti za iscrtavanje vanjskog kruga sata.

3. ODABIR KOMPONENATA

3.1. Arduino Nano

Kao osnovna platforma na kojoj će ovaj sat biti razvijen odabran je Arduino mikrokontroler zbog svoje jednostavnosti i dostupnosti, a konkretno upotrijebljeni mikrokontroler je Arduino Nano temeljen na Atmelovom ATmega328 mikroprocesoru.

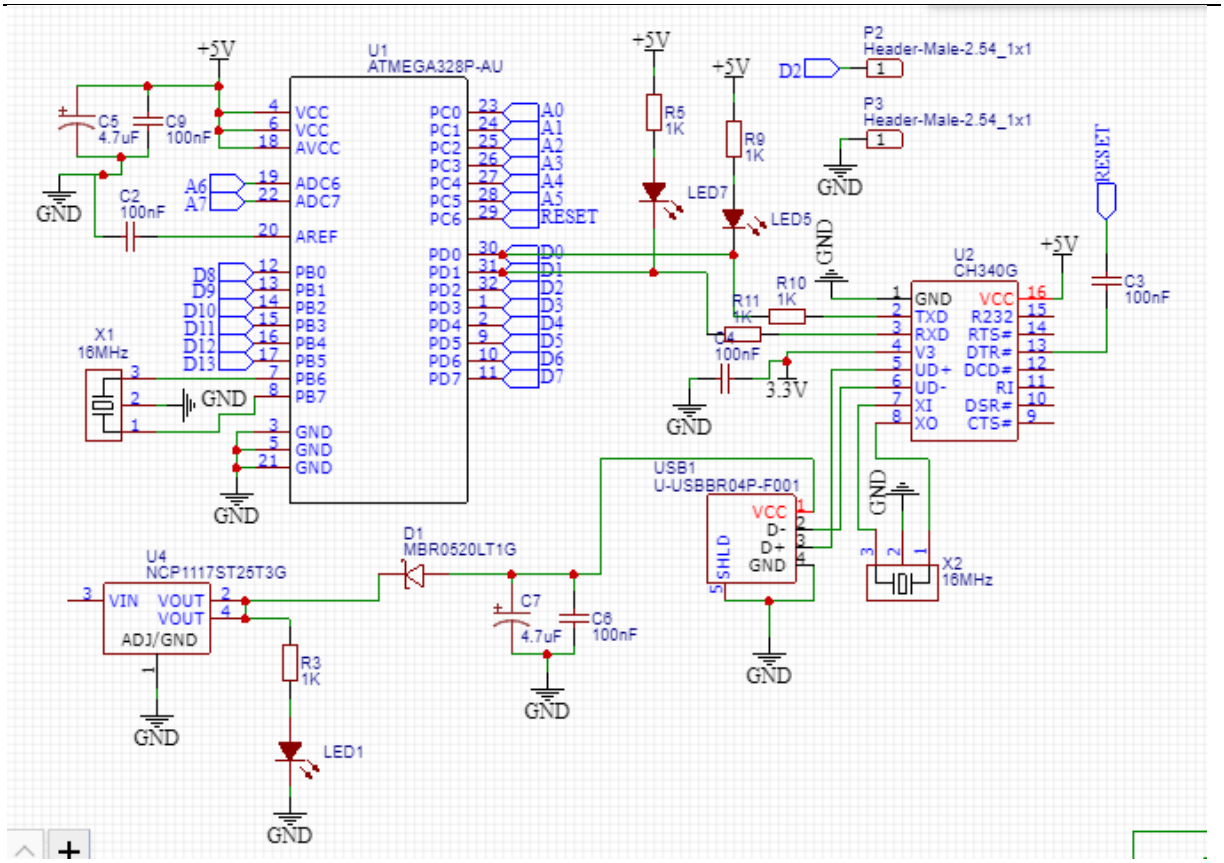


Slika 6. Arduino Nano [1]

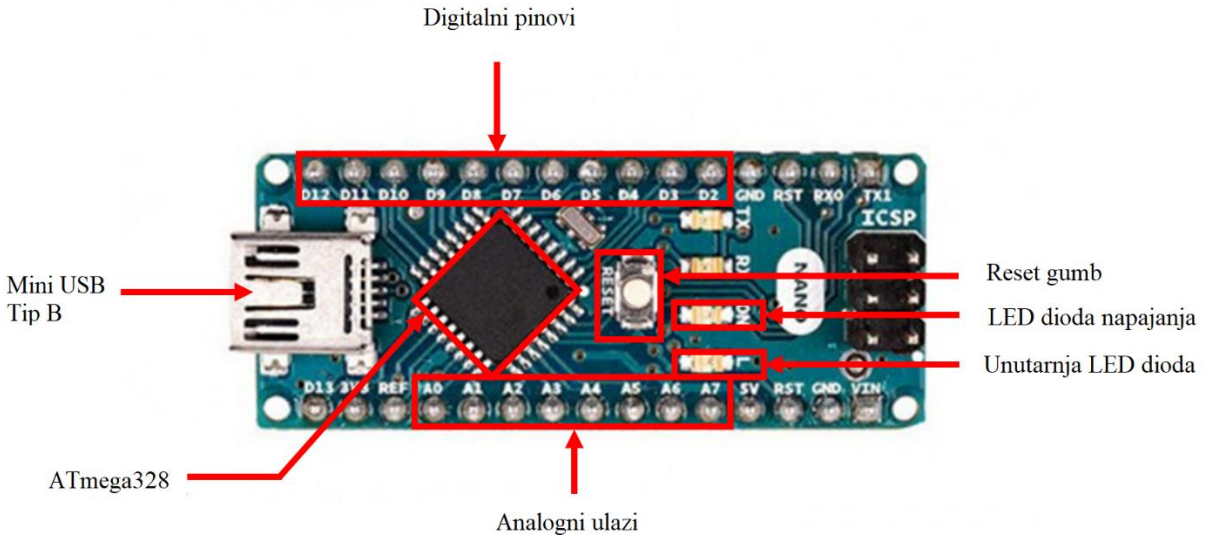
Glavne tehničke karakteristike ovog mikrokontrolera prikazane su u sljedećoj tablici:

Tablica 1. Arduino Nano – tehničke karakteristike [1]

Mikroprocesor	ATmega328
Radni napon	5 V
Flash memorija	32 KB
Radni takt	16 MHz
Broj analognih pinova	8
Broj digitalnih I/O pinova	22
Broj PWM pinova	6
Veličina	18 x 45 mm
Masa	7 g



Slika 7. Električna shema Arduino Nano mikrokontrolera [11]



Slika 8. Prikaz osnovnih dijelova sklopa – Arduino Nano

3.2. Real time clock PCF85063A

Kod uključivanja/isključivanja, odnosno spajanja na računalo, Nano se resetira. To znači da se njegov kod svaki put počinje izvršavati iznova. Stoga je lako zaključiti da ga neće biti moguće koristiti kao brojač vremena jer će svaki put kada se ponovo uključi prikazivati inicijalno postavljeno vrijeme na satu. Iz tog razloga potrebno je implementirati sklop koji će pamtiti stvarno vrijeme i čuvati ga u svojoj memoriji. Na taj način će Nano nakon svakog novog uključivanja pročitati vrijednost vremena iz tog sklopa i moći će prikazivati stvarno vrijeme. Odabrani sklop je Real time clock PCF85063A.

Slika 9. Prednja strana (lijevo) i stražnja strana (desno) RTC modula [3]

PCF85063A naziv je integriranog kruga, dok je sami PCB modula proizveden od strane hrvatske tvrtke E-radionica. Sa stražnje strane modula nalaze se komponente i easyC konektori, dok se s prednje strane modula nalazi utor za CR2032 bateriju. Baterija služi za napajanje sklopa kada je nadređeni uređaj, u ovom slučaju Nano, isključen, odnosno njegovo napajanje je isključeno i ne može podržavati rad RTC uređaja.

Glavne karakteristike ovog modula su prikazane u tablici 2.:

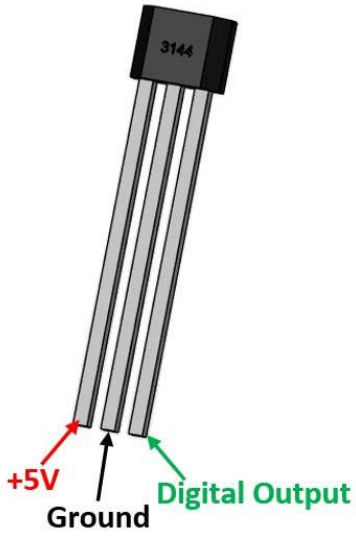
Tablica 2. Karakteristike RTC modula [3]

Radni napon	0,9 – 5,5 V
Komunikacijski protokol	I^2C
Frekvencija oscilatora	32,768 kHz

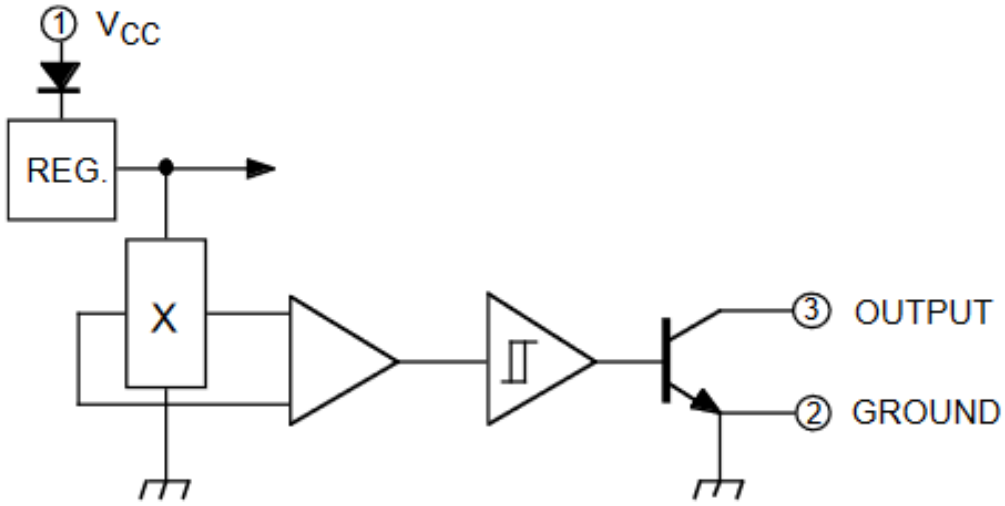
Modul ima i neke dodatne mogućnosti koje u ovom radu neće biti korištene kao što su programabilni alarm, mjerenje dana u tjednu, datuma i godine te vremenski sklop (engl. timer) koji je moguće postaviti od 244 mikrosekunde pa sve do 4 sata i 15 minuta.

3.3. Hall efekt senzor A3144

Kao što je već objašnjeno u drugom poglavlju, ovaj senzor će sudjelovati u mjerenju vremena potrebnog da nosač PCB-a opiše jedan krug. Na izlazu ovog senzora je logička „1“ kada se on ne nalazi u blizini magnetskog polja. Kada se senzor nađe u blizini magnetskog polja dovoljne jačine, njegov izlaz se mijenja u logičku „0“. Ako postavimo ovaj senzor fiksno na obod nosača, a magnet na stacionarni dio kućišta na način da se jednom po krugu senzor i magnet nađu u neposrednoj blizini, možemo putem prekidne rutine vrlo precizno detektirati tu promjenu stanja te pomoću Arduina izračunati koliko je vremena prošlo od posljednje promjene stanja. Na taj način ćemo dobiti vrijeme potrebno nosaču da opiše jedan krug.



Slika 10. A3144 senzor s rasporedom nožica [12]



Slika 11. Simbolička shema senzora [2]

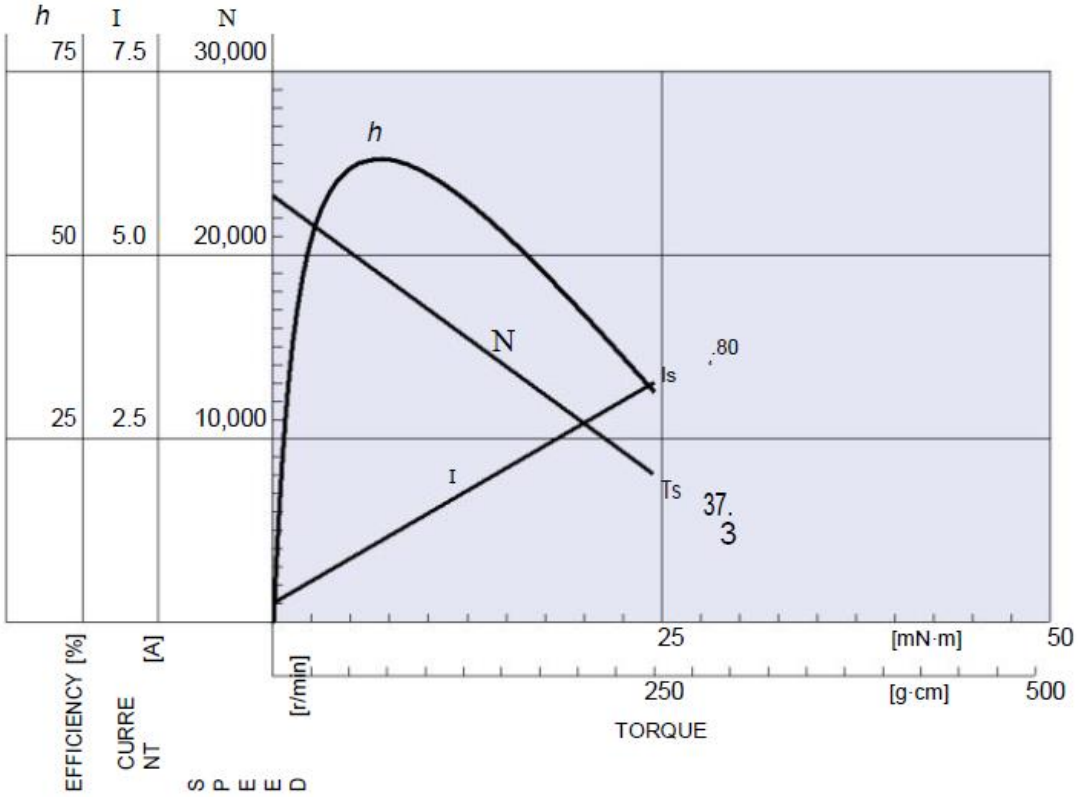
3.4. Istosmjerni motor i ispravljač

Odabrani motor za pogon sata je istosmjerni motor RS-365-SA. Karakteristike motora prikazane su u tablici

Tablica 3. Karakteristike RS-365-SA motora pri maksimalnoj iskoristivosti [5]

Nazivni napon	20 V
Nazivna struja	1,07 A
Nazivni moment	6,82 mNm
Nazivna snaga motora	13,5 W
Nazivna brzina	18960 min^{-1}

Na sljedećoj slici prikazana je karakteristika odabranog motora:



Slika 12. Karakteristika odabranog motora [5]

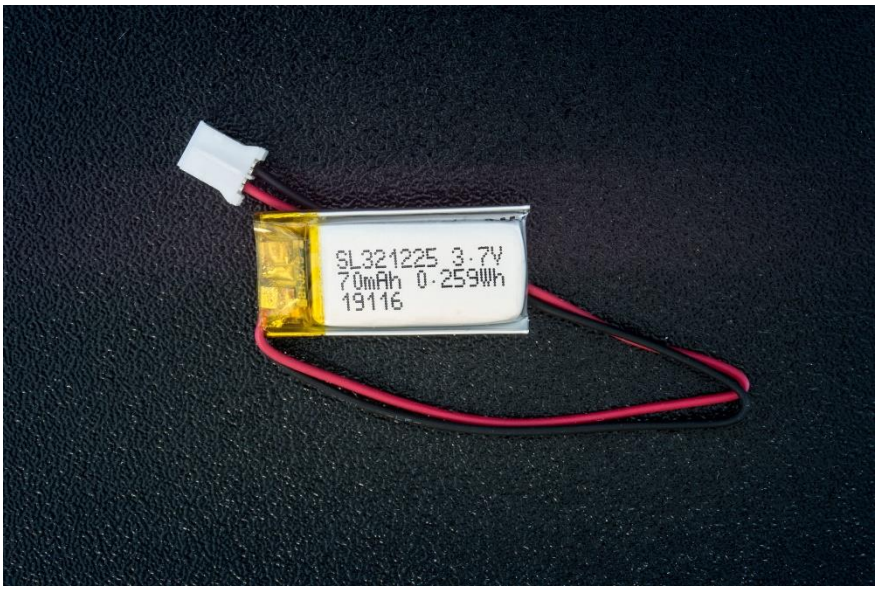
Slika 13. Odabrani istosmjerni motor [5]

3.5. Odabir ostalih komponenata

Već je navedeno da sat koristi 10 LED dioda, no sa svakom diodom u seriju treba spojiti i „pull up“ otpornik, u ovom slučaju nazivnog otpora 1 kΩ.

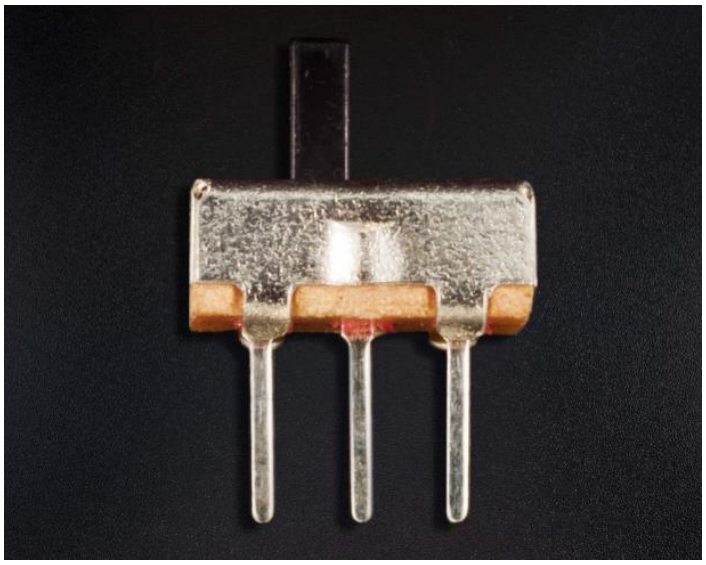
Slika14. Korištena LED dioda [13]

Kako bi cijeli sklop mogao raditi korištena je SL321225 litij – ionska baterija nominalnog kapaciteta 0,259 Wh i nazivnog napona 3,7 V.



Slika 15. SL321225 baterija [14]

Uz sve navedeno, koristi se i prekidač kako Arduino ne bi bespotrebno radio i trošio energiju iz baterije dok je motor isključen.



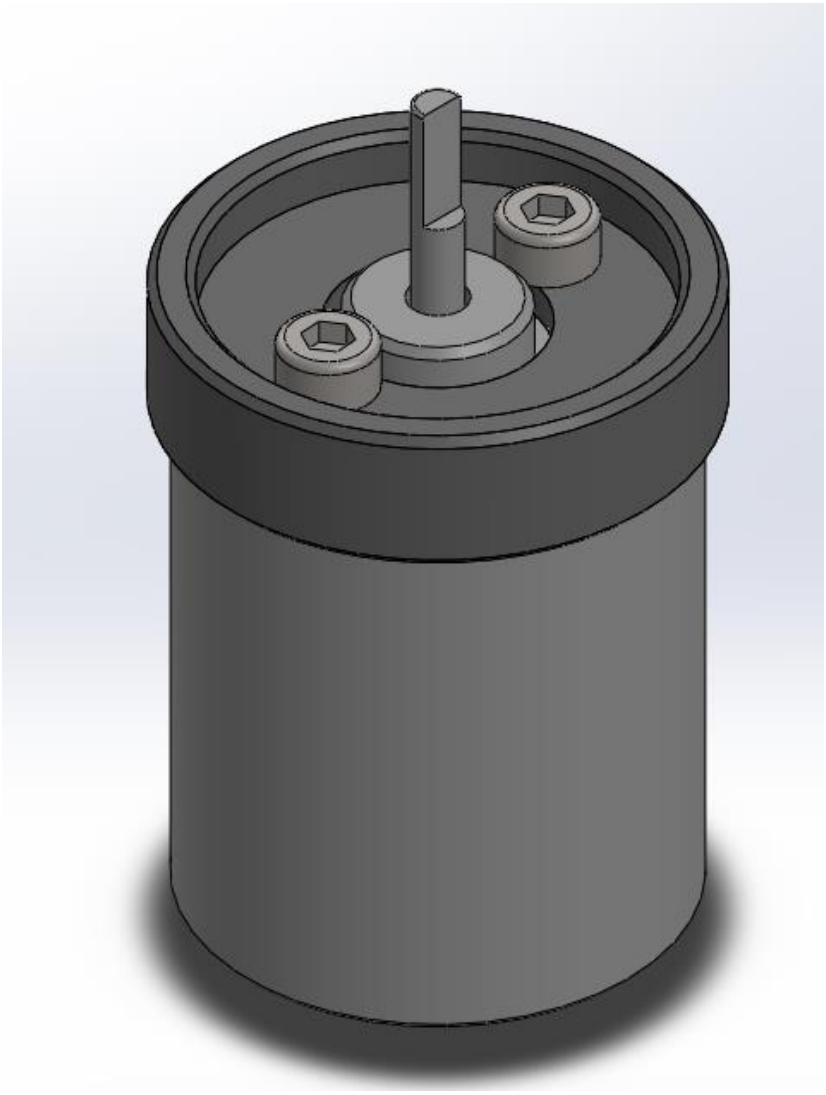
Slika 16. Prekidač [15]

4. KONSTRUKCIJA SATA, IZRADA I SKLAPANJE

4.1. Konstrukcija kućišta s motorom i nosačem PCB-a

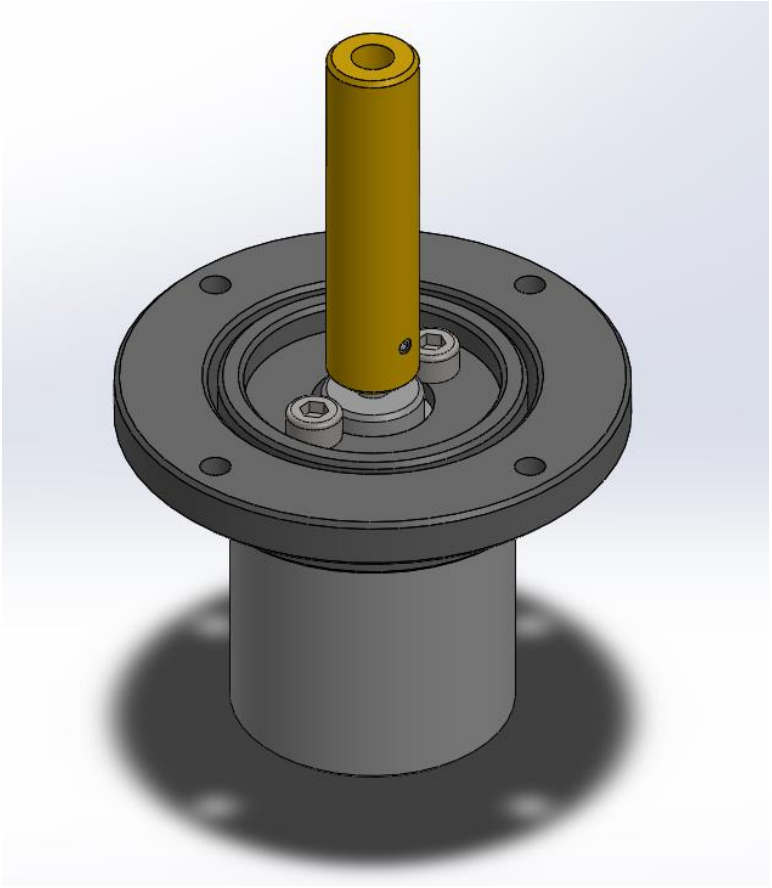
Kao baza motora koristi se drveni disk koji je vijcima pričvršćen za četiri aluminijske nožice. Motor je pomoću dodatnog prstena, prirubnice i samoureznih vijaka pričvršćen za bazu, a na njegovo vratilo pričvršćen je produžetak s proširenjem za lakšu montažu nosača. Na produžetak se montira valjak koji služi za osiguravanje okomitosti nosača i baze te se sami nosač vijkom pričvršćuje kroz njega za vratilo.

Prsten se za motor pričvršćuje s dva vijka, kako je prikazano na slici ispod.

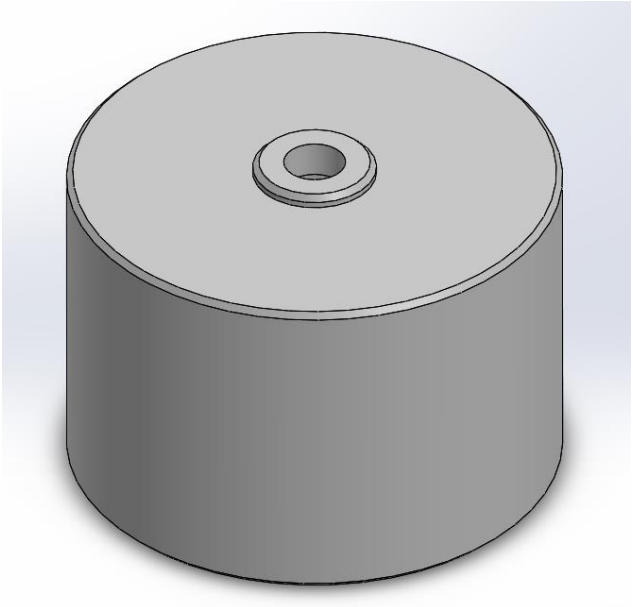


Slika 17. Motor s montiranim prstenom

Prirubnica motora je navučena preko motora i prstena te osigurava motor od ispadanja. Također, izrađena je na način da trenjem sprječava motor da se okrene oko svoje osi u kućištu. Produžetak vratila motora pričvršćen je M2x4 uvrtnim vijkom.

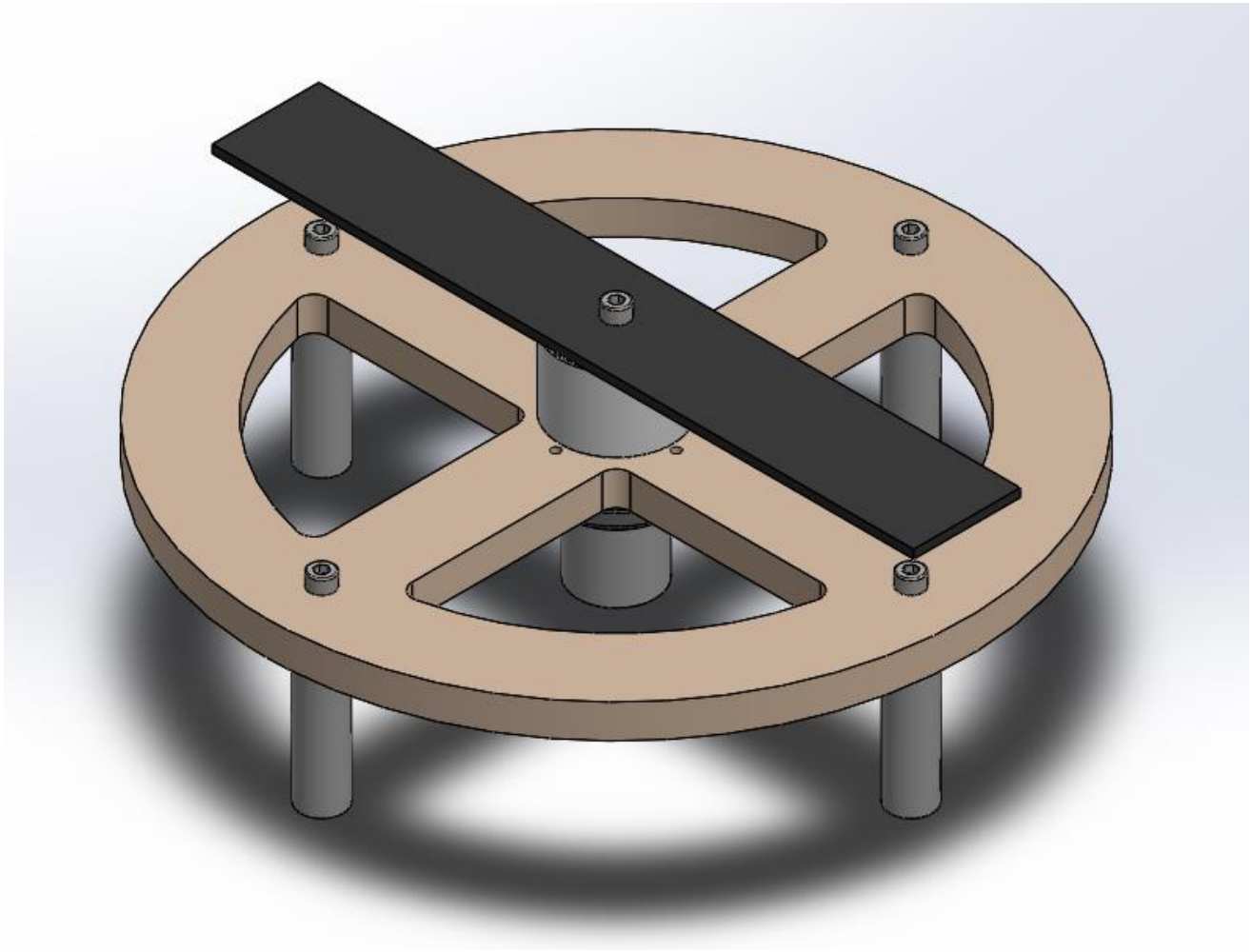


Slika 18. Motor s navučenom prirubnicom i produžetkom osovine



Slika 19. Valjak za osiguravanje okomitosti nosača PCB-a i baze

Nosač PCB-a se kroz valjak pričvršćuje za vratilo pomoću M5 vijka. Navoj vijka je normalan, a smjer vrtnje nosača je udesno, odnosno u smjeru kazaljke na satu. Stoga se može zaključiti da osiguranje od odvrtanja ovdje nije potrebno obzirom da se svakim krugom nosač dodatno učvršćuje.

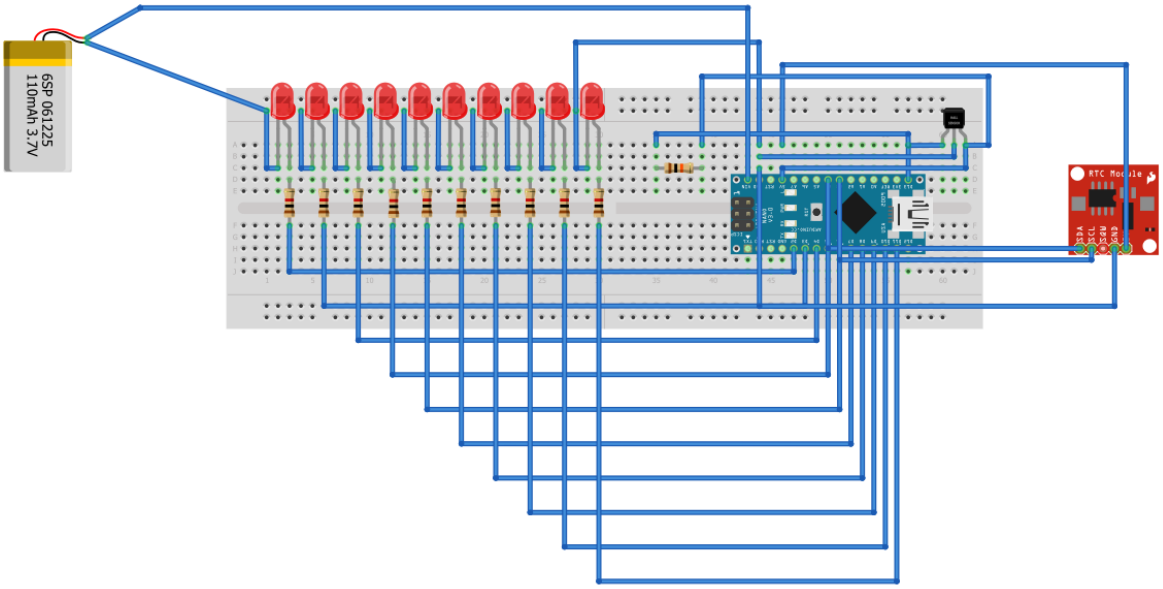


Slika 20. Model sklopa sata

4.2. Električna shema sata

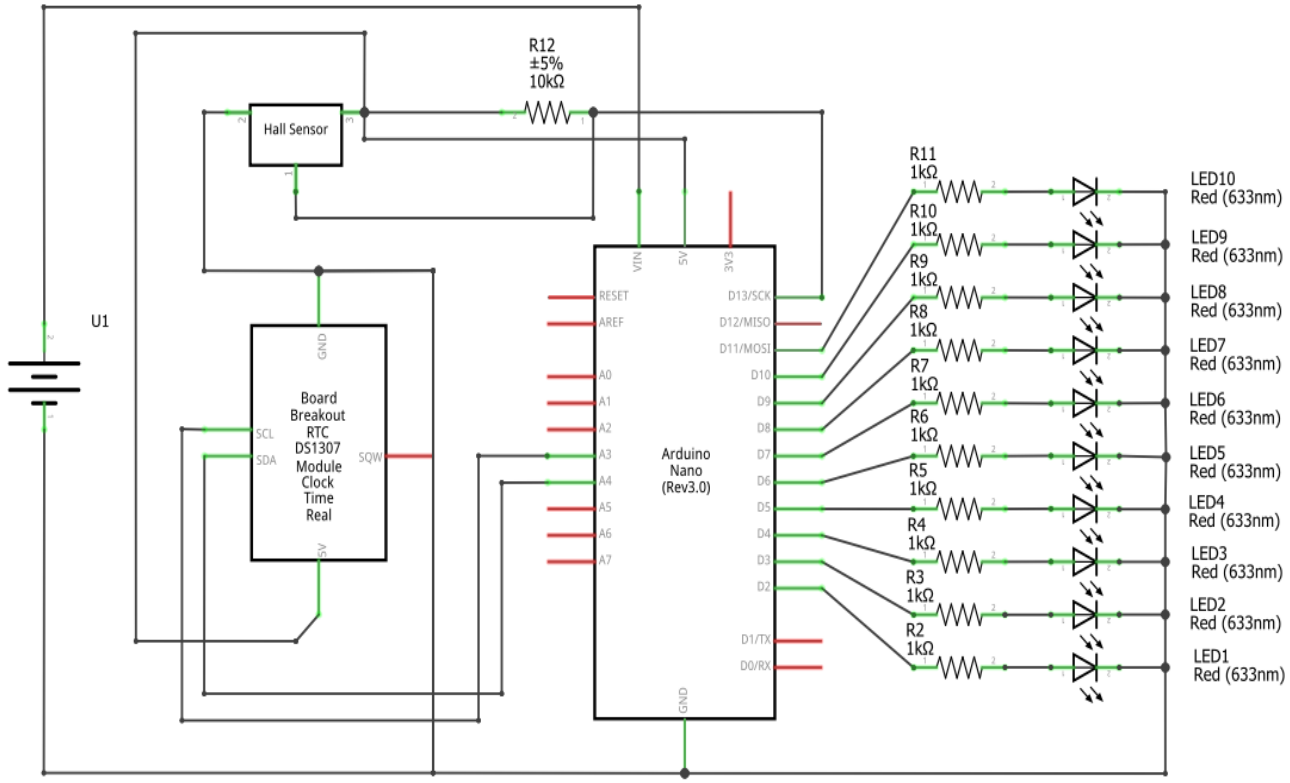
Električna shema nacrtana je u programskom paketu Fritzing. Ovaj program sadrži širok izbor stvarnih komponenata proizvedenih od strane etabliranih proizvođača. Posebno je dobar za primjenu kod projekata temeljenih na Arduinu obzirom da sadrži sve Arduino pločice, ali i veliki broj dodatnih modula. Zbog toga je već prilikom stvaranja sheme sklopa moguće predvidjeti kako će sklop izgledati.

Kako bi nacrtali električnu shemu, prvo je potrebno nacrtati simboličku shemu. Crtanje simboličke sheme poprilično je jednostavno zato što se linije povlače između stvarnih izvoda (pinova) korištenih komponenata.



Slika 21. Simbolička shema sklopa LED sata

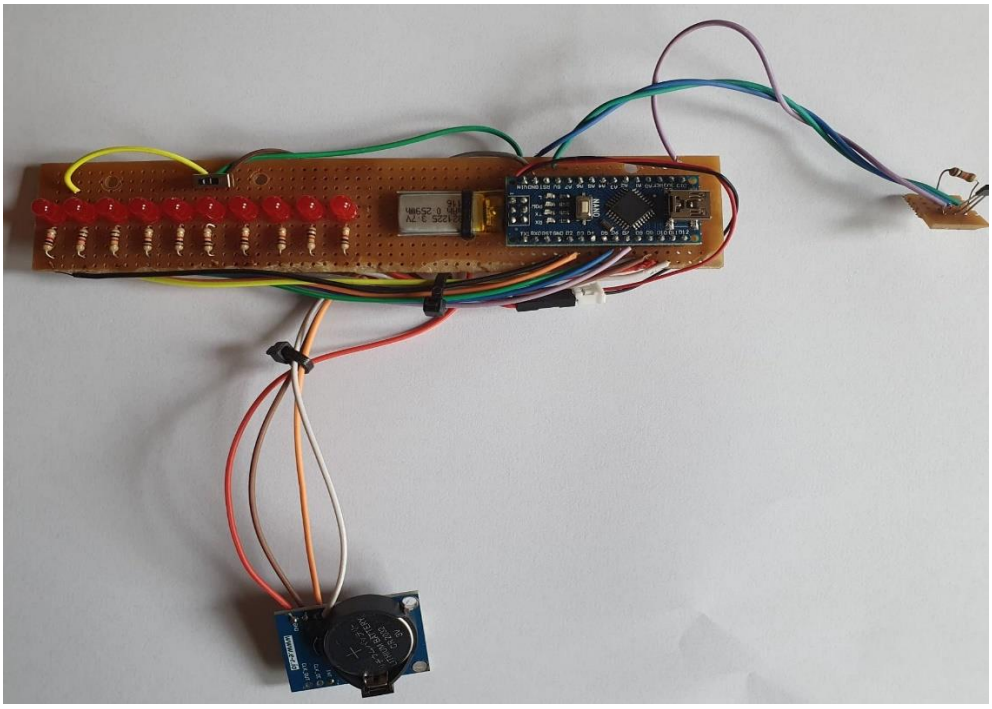
Nakon crtanja simboličke sheme može se nacrtati električna shema. Ovaj program na temelju simboličke sheme sam iscrtava električnu shemu i uz to još nudi opciju automatskog spajanja komponenata koja je, uz male ručne preinake, relativno uspješna.



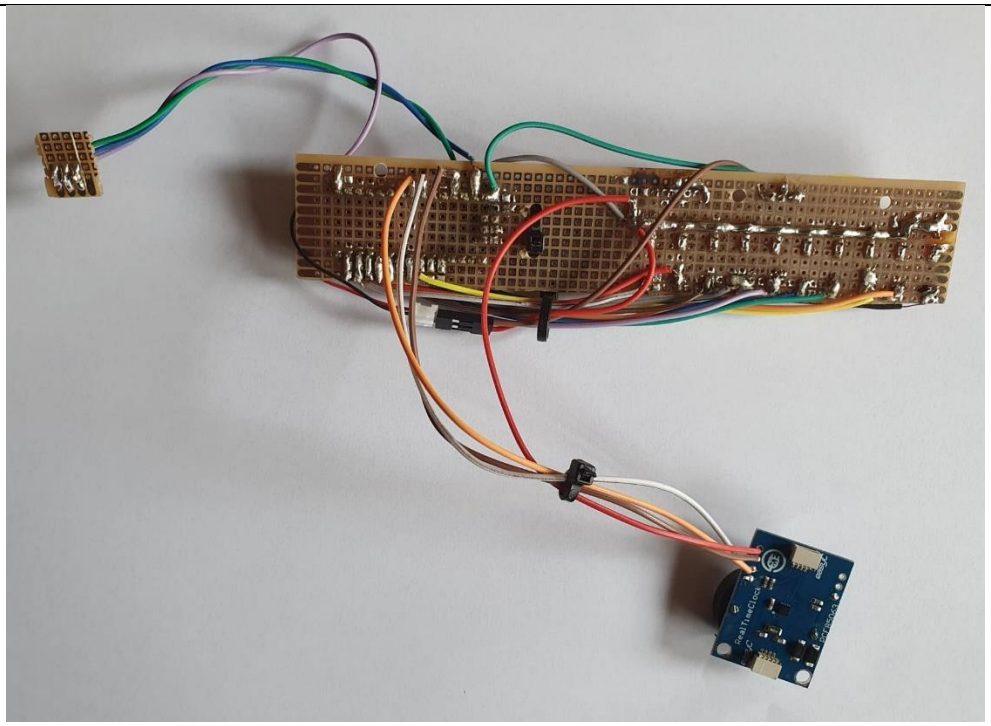
Slika 22. Električna shema LED sata

4.3. Izrada PCB pločice

U izradi PCB pločice korištena je pločica s rupicama. Takav pristup ne zahtijeva izradu tiskane pločice čija cijena izrade u malim serijama, u ovom slučaju jednog komada, može biti visoka.



Slika 23. PCB pločica sprijeda



Slika 24. PCB pločica straga

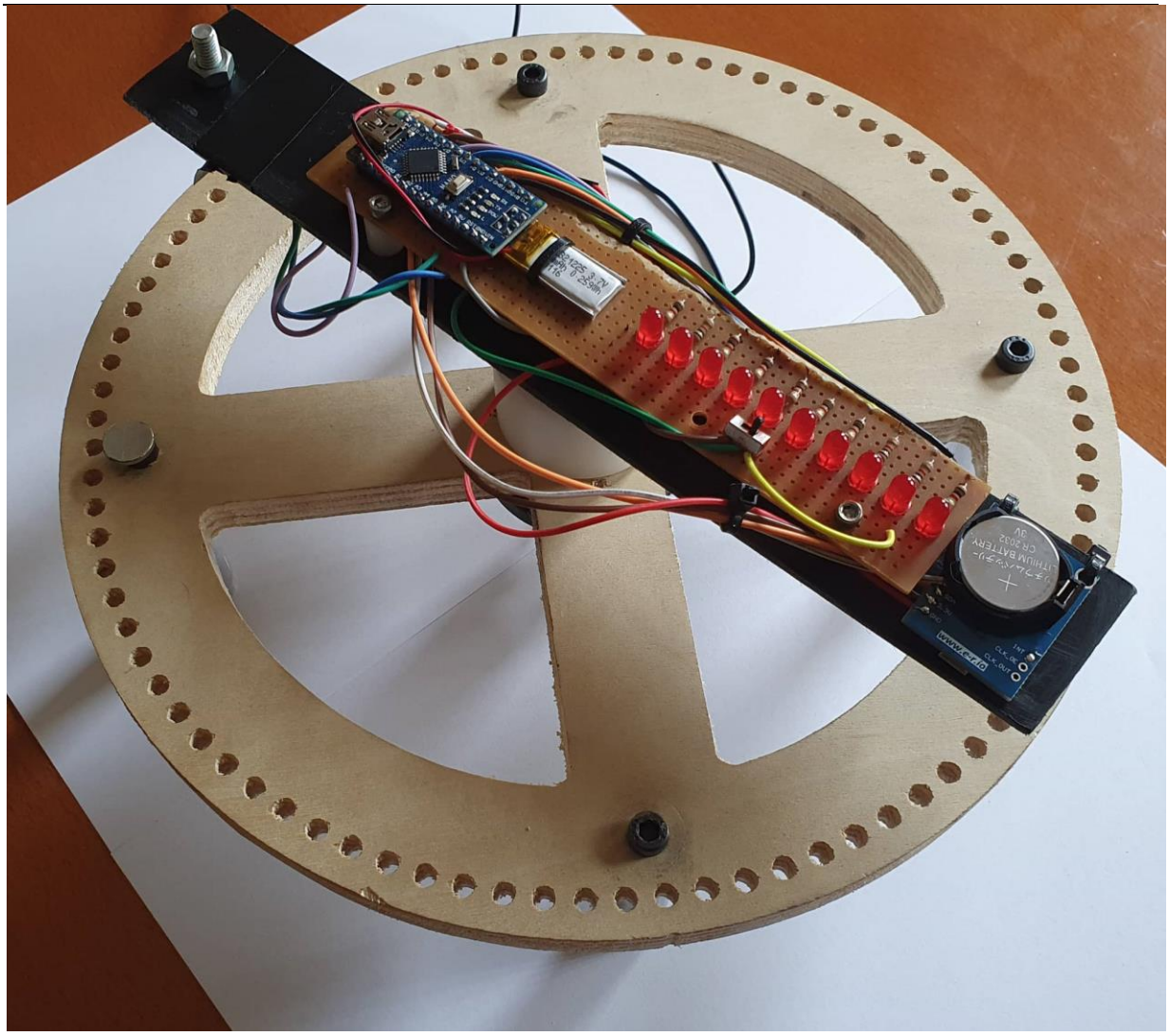
4.4. Izrada konstruiranih pozicija

Konstruirane pozicije navedene u potpoglavlju 4.1 izrađene su postupkom tokarenja.

Prsten za motor i prirubnica izrađene su od poliamida, produžetak vratila od mjedi, a valjak od poliacetala. Nosač PCB-a je izrađen rezanjem ploče crnog polikarbonata.

4.5. Sklapanje

PCB pločice učvršćuju se M2,5 vijcima i maticama za nosač. Hall-ov senzor zaljepljen je s donje strane nosača kako bi mogao prolaziti uz magnet. Magnet je zaljepljen u uspravnom položaju na obodu baze sata. Na slici 19 može se vidjeti kako je na nosač dodan i vijak s maticom. To je učinjeno zato što su nakon sklapanja primijećene vibracije zbog neravnomjernog rasporeda mase po duljini nosača. Stoga je na jedan kraj nosača dodan vijak s maticom kako bi kompenzirao višak mase na drugom kraju nosača.

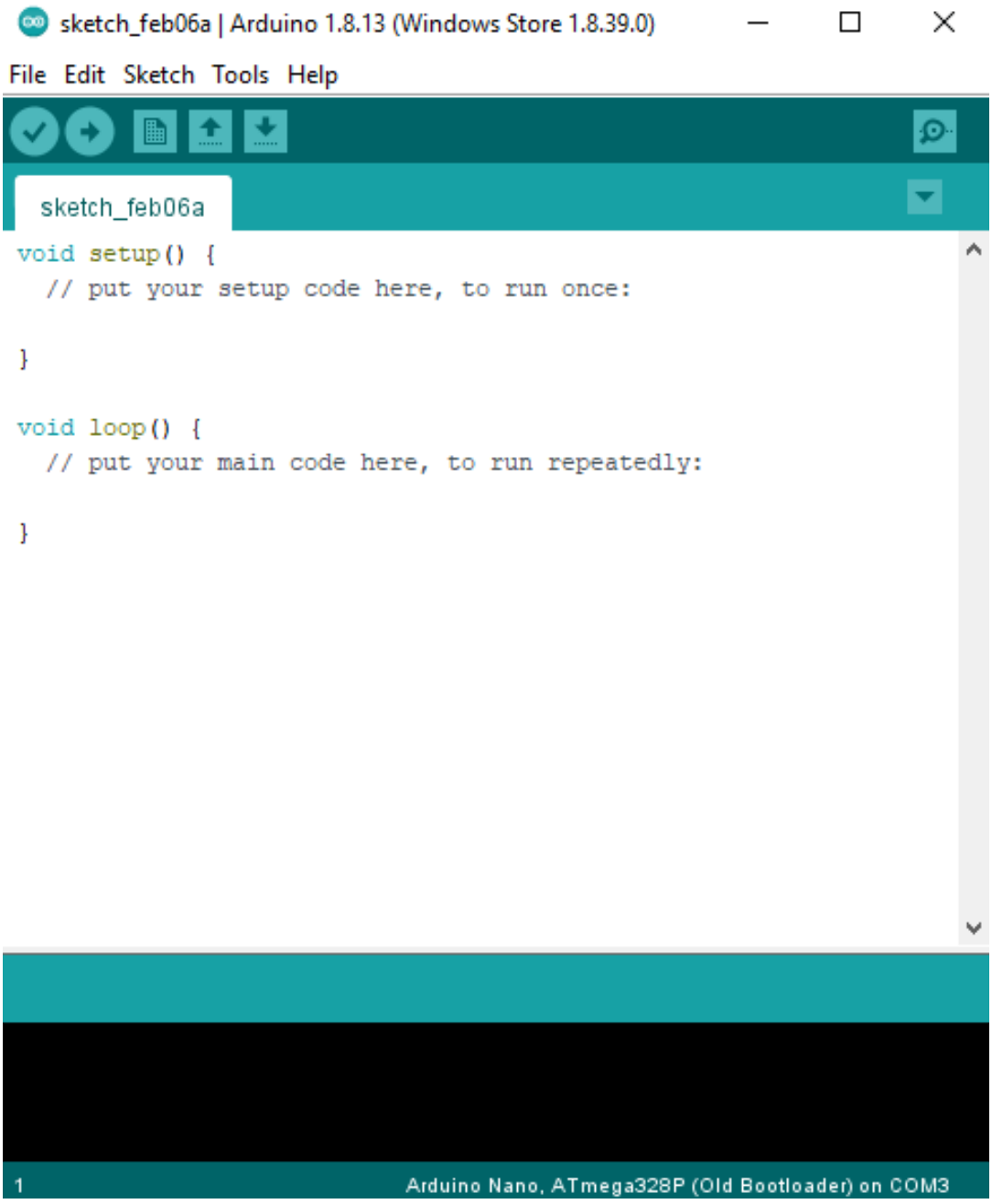


Slika 25. Sklopljeni sat

5. PROGRAMSKI KOD

5.1. Arduino IDE

Programski kod napisan je u Arduino IDE (engl. Integrated Development Enviroment) sučelju. Sučelje se sastoji od prozora za uređivanje teksta, trake s porukama, konzole i alatne trake s nizom izbornika. U njemu se može napisati kod za Arduino pločice, ali uz određene prilagodbe i za neke druge. Programski kod se temelji na C++ programskom jeziku.



Slika 26. Arduino IDE sučelje

Kod napisan u Arduino IDE sučelju sastoji se od 3 dijela.

Prvi dio koda, prije inicijalizacijske rutine (engl. setup), sastoji se najčešće od uključivanja dodanih library-a u kod i deklariranja korištenih varijabli te definiranja nekih osnovnih parametara, ovisno o primjeni.

Drugi dio koda upisuje se unutar void setup() funkcije. Tako napisan kod izvršava se samo jednom prilikom pokretanja Arduina.

Treći dio koda upisuje se unutar void loop() funkcije i izvršava se isponova u petlji cijelo vrijeme dok Arduino radi sve do njegovog gašenja.

5.2. Deklaracija varijabli

```
#include "PCF85063A.h"
PCF85063A rtc;
```

Uvođenje dodatnog library-a za RTC modul.

```
unsigned long Real_time_millis = 0;
unsigned long Previous_real_time_millis = 0;
unsigned long currentMillis, elapsed_loop_counter, previousMillis;
unsigned long counter_1, current_count;
```

Deklaracija varijabli korištenih za realizaciju raznih brojača unutar koda.

```
float SEC = 0;
float MIN = 0;
float HOUR = 0;
float dots_marker = 0;
```

Deklaracija varijabli korištenih za iscrtavanje linija sati, minuta i sekunda te pokazivača 3, 6, 9, i 12 sati.

```
byte last_IN_state;
float one_rot_time=0;
float time_per_deg=0;
String poruka = "";
```

Deklaracija ostalih varijabli, između ostalog i varijable u koju spremamo signal dobiven sa senzora putem prekidne rutine.

U ovom slučaju varijable brojača deklarirane su s „unsigned long“ zato što će oni brojati sve dok sat radi. Takvo deklariranje varijabli omogućava spremanje čak 32 bita.

5.3. Setup

U inicijalizacijskoj (setup) rutini potrebno je definirati što će se sve izvršiti prilikom pokretanja sata. Najprije treba napisati kod koji će omogućiti prijenos podataka putem serijske komunikacije. Taj dio koda služi za povezivanje sata s računalom i definiranje točnog vremena putem korisničkog sučelja o kojem će više riječi biti u sljedećem poglavlju. Zatim treba očitati točno vrijeme iz RTC modula.

```
void setup() {
  Serial.begin(115200);
  Serial.setTimeout(200);
  SEC = rtc.getSecond();
  MIN = rtc.getMinute();
  HOUR = rtc.getHour();
  if (HOUR>=12){
        HOUR=HOUR-12;
      }
```

Označavanje početka serijske komunikacije i primanje podataka za namještanje točnog vremena na satu od RTC modula.

Obzirom da analogni sat pokazuje 12 – satni format vremena, potrebno je učiniti pretvorbu.

Potom treba definirati koji pinovi su ulazni, a koji izlazni te njihove inicijalne vrijednosti. U ovom radu neće biti korištene klasične Arduino varijable za upravljanje ulazno izlaznim signalima poput digitalWrite() i digitalRead() zato što je njihovo izvođenje sporije i gotovo je nemoguće izvesti da se izvode u potpunosti paralelno, a što je u ovom slučaju i više nego potrebno. Upravo zato će se koristiti funkcije za direktno upravljanje registrima pojedinih portova.

ATmega328 sadrži 3 porta:

- Port B – digitalni pinovi 8 do 13
- Port C – analogni ulazni pinovi
- Port D – digitalni pinovi 0 do 7

Lako je zaključiti da će se koristiti port B i port D.

Ovakav način programiranja puno je brži kod izvođenja programa, ali čini kod teže razumljivim što može dovesti do poteškoća kod održavanja i izmjene programskog koda. Također, takvo programiranje nije podržano od strane svih mikrokontrolera.

```
PCICR |= (1 << PCIE0);
PCMSK0 |= (1 << PCINT5);
```

Omogućavanje prekidne rutine i postavljanje pina 13 kao prekidne linije.

```
DDRD |= B11111100;
DDRB |= B00001111;
```

Postavljanje pinova 2-7 i 8-11 u funkciju izlaznih pinova.

```
DDRB &= B11011111;
```

Postavljanje pina 13 u funkciju ulaza.

```
PORTD &= B00000011;
PORTB &= B11110000;
```

Inicijalno postavljanje promjenjivih izlaza u stanje logičke „0“.

```
digitalWrite(2, HIGH);
}
```

Postavljanje izlaza na koji je spojena dioda koja predstavlja vanjsku kružnicu sata u stanje logičke „1“.

5.4. Petlja

Prvi korak u petlji jest omogućiti primanje podataka putem korisničkog sučelja za namještanje točnog vremena na satu.

```
void loop() {

  while (Serial.available() > 0){
    poruka = Serial.readString();
```

Prijenos informacije o točnom vremenu s računala putem serijske komunikacije.

```
rtc.setTime(poruka.substring(0, 2).toInt(), poruka.substring(2,
4).toInt(), poruka.substring(4, 6).toInt());
delay(100);
```

Postavljanje vremena na satu na temelju očitanog podatka. Podatak se očitava kao string od 6 znakova te ga treba podijeliti u sate, minute i sekunde.

```
    SEC = rtc.getSecond();
    MIN = rtc.getMinute();
    HOUR = rtc.getHour();
    if (HOUR>=12){
        HOUR=HOUR-12;}}
```

Očitavanje postavljenog vremena od strane mikrokontrolera.

Pretvorba u 12 – satni format

Sljedeći korak je kreiranje brojača koji će nakon očitavanja vremena iz RTC modula nastaviti brojati vrijeme i spremati ga u varijable koje će se koristiti kod uključivanja dioda. Za brojanje vremena koristit će se funkcija millis(). To je funkcija koja prilikom izvršavanja vraća broj milisekundi koje su prošle od uključivanja Arduina.

```
Real_time_millis = millis();
  if(Real_time_millis - Previous_real_time_millis >= 1000){
    Previous_real_time_millis += 1000;
    SEC=SEC+1;
    if(SEC > 59)
    {
      SEC=0;
      MIN=MIN+1;
    }

    if(MIN > 59)
    {
      MIN = 0;
      HOUR = HOUR+1;
      if (HOUR>11){
      HOUR=0;}}}
```

Dodavanje 1 sekunde u brojač stvarnog vremena ukoliko je prošla 1 sekunda.

Brojanje vremena. Na kraju brojanja sati ponovo dolazi pretvorba u 12 - satni format vremena.

Na red dolazi uključivanje dioda. Prvo treba napisati kod za uključivanje već spomenute četiri crtice, ali prije toga treba napraviti brojač trajanja izvršavanja petlje koji će se koristiti za postavljanje uvjeta za uključivanje dioda:

```
currentMillis = micros();
elapsed_loop_counter = currentMillis - previousMillis;
```

Ispis četiri crtice:

```
if(elapsed_loop_counter   >=     time_per_deg*(dots_marker)   &&
elapsed_loop_counter <  time_per_deg*(dots_marker+3)     )
  {
    PORTD |= B00011000;
  }
```

Ako je brojač petlje veći ili jednak vremenu potrebnom da nosač stigne u poziciju prve crtice i manji od vremena da nosač prođe tu poziciju za 3 stupnja, uključuju se diode 3 i 4.

```
  if(elapsed_loop_counter >= time_per_deg*(dots_marker+3))
  {
    PORTD &= B11100111;
    dots_marker = dots_marker + 90;
    if(dots_marker >= 360)
    {
      dots_marker = 0;}}
```

Ako je brojač veći od vremena potrebnog da nosač prođe poziciju 3 stupnja unutar crtice, diode se isključe i brojaču za crtice se dodaje 90 stupnjeva, što je ujedno i lokacija sljedeće crtice.

Nakon ispisa crtica, moguće je krenuti s ispisom kazaljki. Prva na redu je kazaljka sekundi koja će se ispisati na isti princip kao i četiri crtice. Kazaljka je zamišljena na način da se pomiče isto kao i kazaljka na pravom satu, a za ispis će se koristiti 7 dioda.

```
if(elapsed_loop_counter    >=        time_per_deg*(SEC*6)     &&
elapsed_loop_counter < time_per_deg*((SEC*6)+2) )
  {
    PORTB |= B00001111;
    PORTD |= B11100000;
  }
```

Ukoliko je vrijeme brojača petlje veće ili jednako vremenu potrebnom za pomak nosača za jedan stupanj pomnoženog sa brojem sekundi i brojem 6 (zato što pomak od jedne sekunde na satu iznosi 6 stupnjeva u krugu), i manje od vremena potrebnog za pomak veličine još dva stupnja, diode se uključuju.

```
  if(elapsed_loop_counter >= time_per_deg*((SEC*6)+2))
  {
    PORTB &= B11110000;
    PORTD &= B00011111;
  }
```

Ukoliko je nosač prošao poziciju trenutnog stanja sekundi za 2 ili više stupnja, diode se isključuju.

Na vrlo sličan način ispisuju se i kazaljke minuta i sati, samo s različitim brojem dioda kako bi ih se moglo razlikovati.

```
  if(elapsed_loop_counter    >=        time_per_deg*(MIN*6)     &&
elapsed_loop_counter <  time_per_deg*((MIN*6)+1)     )
  {
    PORTB |= B00001111;
    PORTD |= B11000000;
  }
```

Ukoliko je vrijeme brojača petlje veće ili jednako vremenu potrebnom za pomak nosača za jedan stupanj pomnoženog sa brojem minuta i brojem 6 (zato što pomak od jedne minute na satu iznosi 6 stupnjeva u krugu), i manje od vremena potrebnog za pomak veličine još jednog stupnja, diode se uključuju.

```
  if(elapsed_loop_counter >= time_per_deg*((MIN*6)+1))
  {
    PORTB &= B11110000;
    PORTD &= B00111111;
  }
```

Ukoliko je nosač prošao poziciju trenutnog stanja minuta za 1 ili više stupnjeva, diode se isključuju.

```
  if(elapsed_loop_counter   >=      time_per_deg*(HOUR*30)    &&
elapsed_loop_counter <  time_per_deg*(   (HOUR*30)  +1)   )
  {
    PORTB |= B00001111;
    PORTD |= B00000000;
  }
```

Ukoliko je vrijeme brojača petlje veće ili jednako vremenu potrebnom za pomak nosača za jedan stupanj pomnoženog sa brojem sati i brojem 30 (zato što pomak od jednog sata na satu iznosi 30 stupnjeva u krugu), i manje od vremena potrebnog za pomak veličine još jednog stupnja, diode se uključuju.

```
  if(elapsed_loop_counter >= time_per_deg*(  (HOUR*30)   +1))
  {
    PORTB &= B11110000;
    PORTD &= B11111111;
  } }
```

Ukoliko je nosač prošao poziciju trenutnog stanja sati za 1 ili više stupnjeva, diode se isključuju.

5.5. Očitavanje signala s Hall-ovog senzora

Kao što je već spomenuto, očitavanje signala vrši se putem prekidne rutine. To znači da taj dio programa ima u mikroprocesoru prednost izvršavanja nad svim ostalim dijelovima programa. Rezultat je izrazito brzo očitavanje signala. U ovom slučaju prekid će biti potaknut promjenom stanja na ulazu 13 na kojeg je spojen senzor. Uz pomoć brojača mjerit će se vrijeme između dvije promjene signala i uz pomoć toga izračunati vrijeme potrebno za rotaciju nosača za puni krug.

```
ISR(PCINT0_vect){
```

Uspostavljanje prekidne rutine potaknute promjenom stanja na portu B.

```
current_count = micros();
  if(PINB & B00100000){
    if(last_IN_state == 0){
      last_IN_state = 1;
      counter_1 = current_count;
    } }
```

Ako je stanje na ulazu 13 logička „1“ (magnet nije u blizini senzora), a prethodno stanje je bilo logička „0“, dogodila se promjena stanja i u brojač „counter 1“ sprema se vrijeme proteklo od početka izvršavanja programa u mikrosekundama.

```
else if(last_IN_state == 1){

    last_IN_state = 0;

    one_rot_time = current_count - counter_1;

    time_per_deg = one_rot_time/360.0;

    previousMillis = micros();


  } }
```

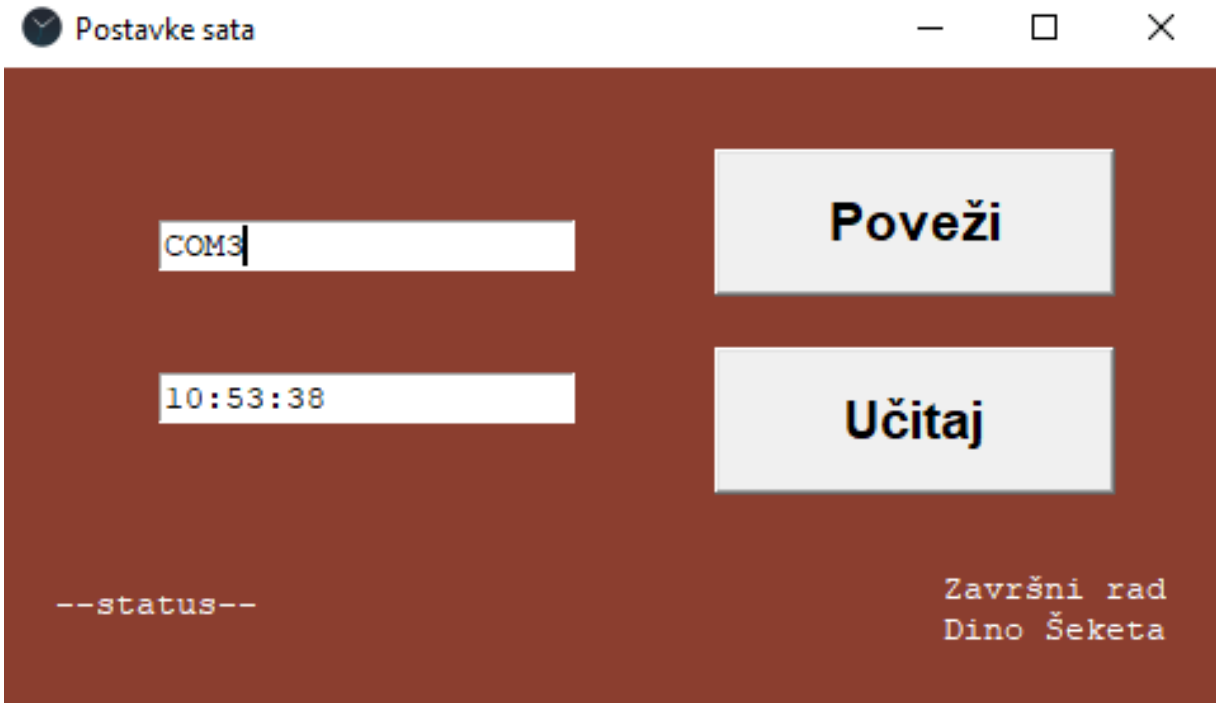
Ukoliko je stanje na ulazu 13 logička „0“ (magnet je u neposrednoj blizini senzora), a prethodno stanje je bila logička „1“, računa se razlika između vremena proteklog od početka izvršavanja programa i vremena u kojem se zadnji put dogodila promjena stanja te se na taj način izračunava vrijeme potrebno nosaču da napravi jedan krug. Dodatno se izračunava i vrijeme potrebno nosaču za pomak od jednog

6. KORISNIČKO SUČELJE

RTC modul kao takav nije dovoljan za održavanje stvarnog vremena na satu. Njegov posao je da nakon inicijalnog postavljanja vremena nastavi brojati, no ne može predvidjeti u kojoj će vremenskoj zoni sat raditi. Također, ne može predvidjeti prelazak sa „zimskog“ na „ljetno“ računanje vremena i obratno. Taj prelazak bi doduše bilo moguće izvesti uz pomoć programskog koda, no to i dalje ne vrijedi u svakom dijelu svijeta. Također, bilo bi vrlo nespretno mijenjati vrijeme direktno u kodu svaki put kada je to potrebno. Iz navedenih razloga, napravljeno je korisničko sučelje za konfiguraciju sata.

Sučelje je izvedeno u Python programskom jeziku, točnije u Tkinter paketu. Tkinter je standardni paket za stvaranje korisničkog sučelja ili GUI-a (engl. Graphical User Interface) u Python programskom jeziku.

Uz samu konfiguraciju sučelja, napravljena je i funkcija za spajanje na Arduino putem serijske komunikacije. Podatak o vremenu šalje se u obliku šesteroznamenkastog stringa koji se zatim u Arduino programskom kodu dijeli na tri dvoznamenkasta stringa i zapisuje u RTC modul.



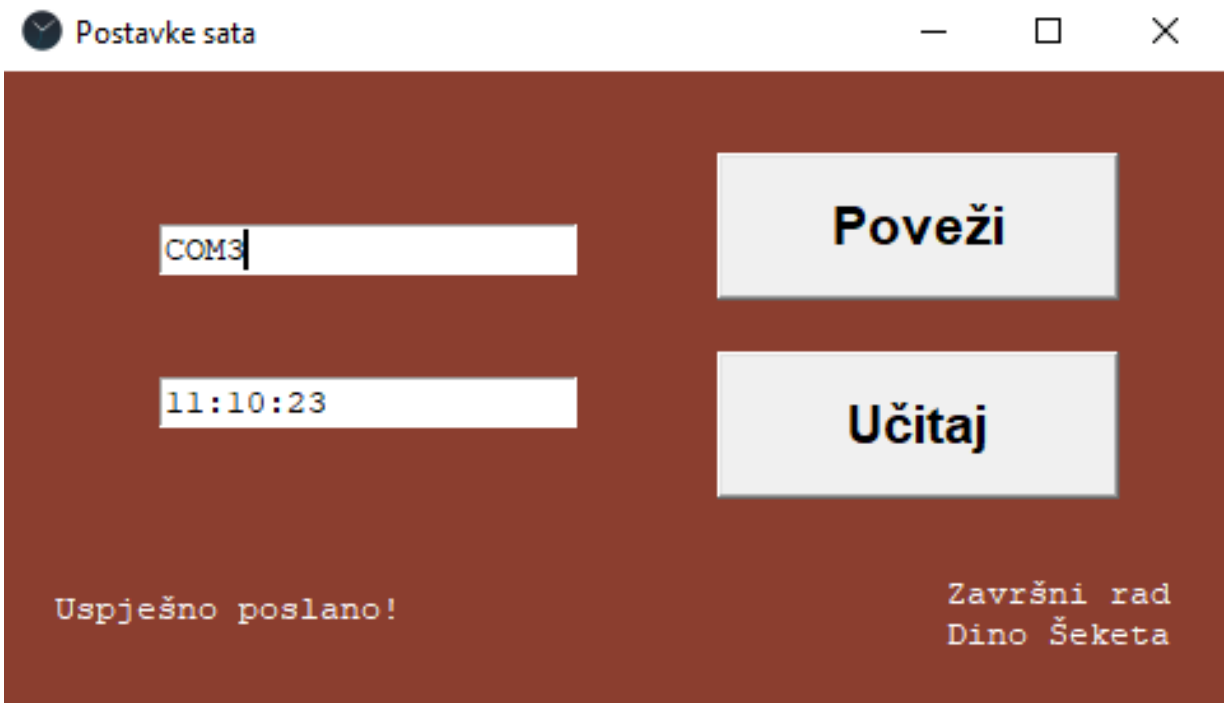
Slika 27. Korisničko sučelje nakon pokretanja

Korisničko sučelje je vrlo jednostavno za koristiti. Sadrži dvije tekstualne linije za upisivanje teksta i dvije tipke. Arduino Nano koji se nalazi na satu spoji se preko USB kabla na računalo. U gornju tekstualnu liniju upisuje se ime porta na koji je spojen Arduino, a što se može doznati direktno iz Arduino IDE sučelja.

Nakon toga potrebno je kliknuti na tipku „Poveži“. Ukoliko je povezivanje uspješno, „--status—“ se mijenja u „Uspješno povezano!“. Ako je kojim slučajem odabran krivi port ili je došlo do neke druge greške u povezivanju, pojaviti će se poruka „Greška spajanja!“.

Nakon toga, potrebno je postaviti vrijeme u donjoj tekstualnoj liniji. Prilikom pokretanja sučelja, u toj liniji će se zapisati točno vrijeme u tom trenutku, no isto je moguće mijenjati prema vlastitom izboru.

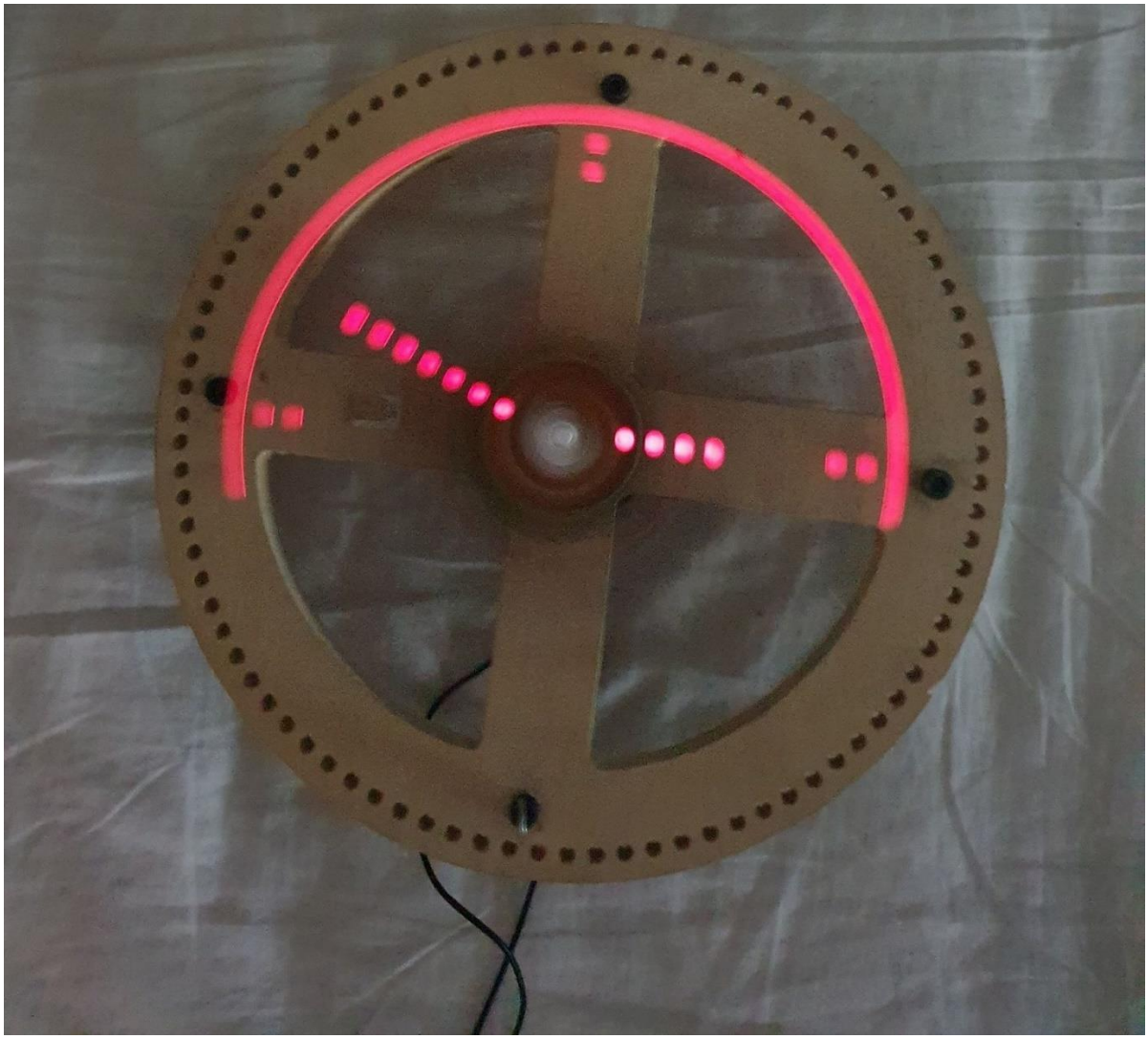
Naposlijetku, potrebno je kliknuti na tipku „Učitaj“ što pokreće slanje vremena u memoriju Arduina. Ukoliko je slanje uspješno, „Uspješno povezano!“ se mijenja u „Uspješno poslano!“. Ukoliko dođe do greške prilikom slanja, pojaviti će se poruka „Greška slanja!“.



Slika 28. Sučelje nakon uspješnog slanja podataka

7. TESTIRANJE SATA

Spajanjem sata na izvor napajanja i uključivanjem utvrđeno je da sat radi kako je i zamišljen. Ubrzanje nosača do konstantne brzine iznosi desetak sekundi, a kazaljke su vidljive već nakon pet sekundi. Mali nedostatak je što pretposljednja i posljednja crtica koja označava sate lagano trepere. Razlog tome je što je prolazak nosača kroz zadane točke brži od izvršavanja petlje te se kod za uključivanje dioda ne uspije izvršiti svaki krug, a rješenje za to bi bilo korištenje mikrokontrolera s bržim mikroprocesorom. Utvrđeno je da sat i kroz nekoliko dana uspješno zadržava točno vrijeme.



Slika 29. Testiranje sata

Na gornjoj slici vidljivo je kako polovica sata nedostaje. Razlog tomu je što je kamera mobitela ipak „brža“ od ljudskog oka i nju ne možemo u potpunosti „prevariti“.

8. ZAKLJUČAK

Ovim radom u potpunosti je prikazan postupak realizacije digitalnog rotirajućeg LED sata temeljenog na jednostavnoj i jeftinoj Arduino platformi. U današnje vrijeme sve je popularnija digitalizacija uređaja koji su do sad bili poznati samo kao analogni. Rezultat toga je pojava sve traženijih i prodavanijih „gadgeta“ na tržištu.

Možda naizgled banalan primjer digitalizacije analognog sata uz neka proširenja može izrasti u ozbiljan i tražen proizvod. Dodavanjem RGB dioda moguće je dobiti sat koji će raditi u više boja, a proširivanjem programskog koda moguće je od sata napraviti displej koji može prikazivati bilo što zamišljeno. Dodavanjem treće dimenzije dioda moguće je dobiti sve traženiji holografski prikaz.

Na ovom radu moguće je napraviti nekoliko poboljšanja. Prvenstveno, može se koristiti brži mikrokontroler kako bi se smanjilo treperenje crtica. Također, moguće je implementirati mikrokontroler s ugrađenom mogućnošću povezivanja putem WiFi ili Bluetooth veze. To bi omogućilo konfiguraciju sata bez potrebe za sučeljem zato što bi se tada moglo direktno očitavati stvarno vrijeme iz nadređenog uređaja. Uz dostupnost boljih postupaka izrade, moguće je izraditi i bolje izbalansirano kućište sata.

Tokom izrade ovog rada moguće je mnogo naučiti iz različitih područja mehatronike i općenito strojarstva, od programiranja i elektronike pa sve do konstruiranja fizičke izvedbe. Sva ova znanja vrlo su korisna za budućeg inženjera mehatronike i robotike.

LITERATURA

[1] Arduino Nano https://store.arduino.cc/arduino-nano

[2] A3144 Datasheet https://components101.com/sites/default/files/component_datasheet/A3144%20Hall%20effect%20Sensor.pdf

[3] RTC modul https://e-radionica.com/hr/blog/2020/05/21/kkm-pcf85063a-rtc/

[4] „Persistence of vision“ https://www.oxfordreference.com/view/10.1093/acref/9780199587261.001.0001/acref-9780199587261-e-0521

[5] RS-365-SA https://3.imimg.com/data3/KB/JX/MY-308084/dc-motors.pdf

[6] Arduino manipulacija portovima https://www.arduino.cc/en/Reference/PortManipulation

[7] Tkinter https://docs.python.org/3/library/tkinter.html

[8] LED sat sa 7-segmentnim displejem https://www.amazon.com/Updated-Version-Plumeet-Adjustable-Brightness/dp/B07Y9HW1XB

[9] LED sat s matričnim displejem https://www.indiamart.com/proddetail/p10-led-based-mono-colour-digital-clock-size-2-x-1-2-ft-14882698230.html

[10] Analogni sat https://www.muji.com/in/products/cmdty/detail/4547315915224

[11] Arduino Nano shema https://forum.arduino.cc/index.php?topic=654630.0

[12] A3144 https://components101.com/a3144-hall-effect-sensor

[13] LED dioda https://zero2maker.org/product/red-led-5mm-through-hole-edd-002-a/

[14] Baterija https://e-radionica.com/hr/li-ion-baterija-70mah-3-7v.html

[15] Prekidač https://e-radionica.com/hr/prekidac-spdt-tht.html

PRILOZI

I. CD-R disc

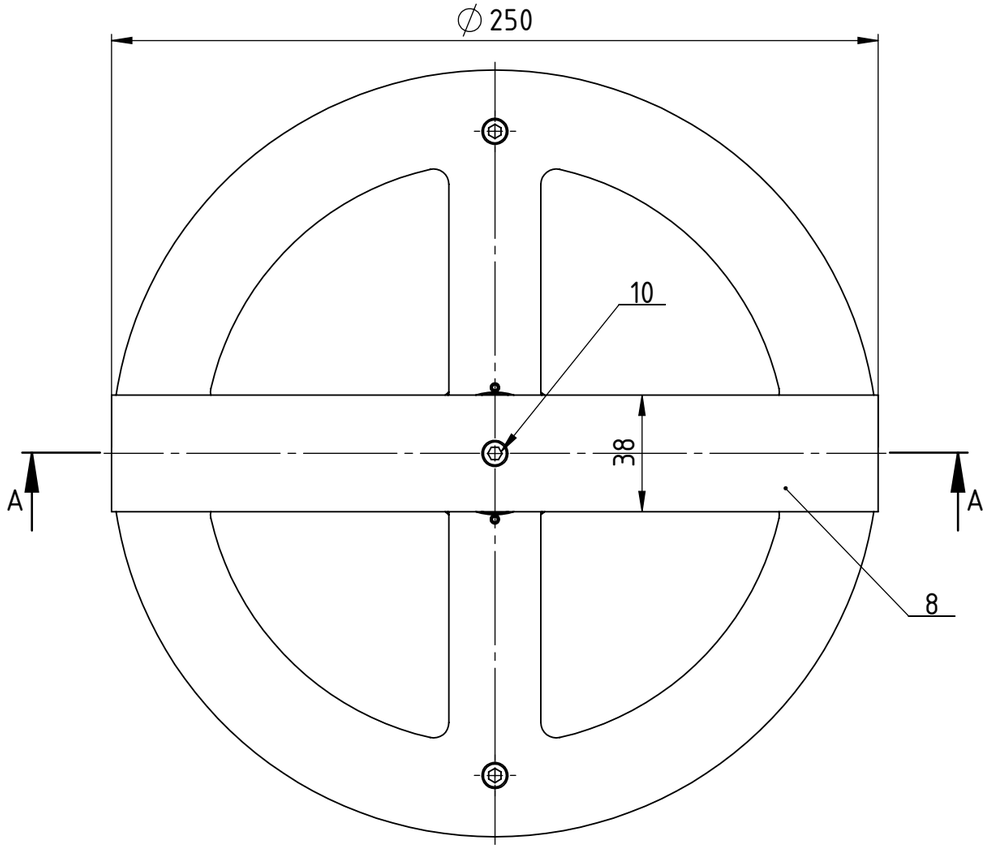
II. Python kod za korisničko sučelje

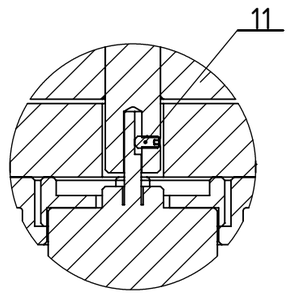
III. Sklopni crtež sata

Python kod za korisničko sučelje:

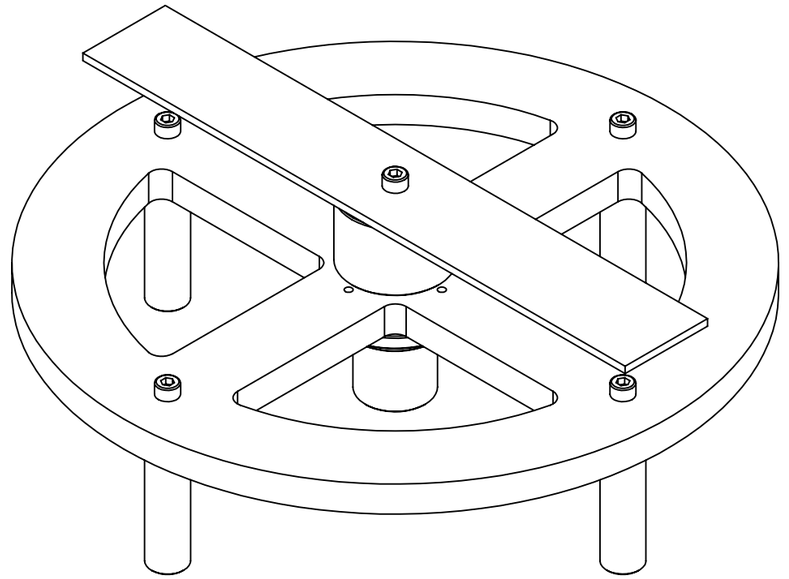
```
from tkinter import *
import serial
import time
import platform
from datetime import datetime


def center(win):
    win.update_idletasks()
    width = win.winfo_width()
    height = win.winfo_height()
    x = (win.winfo_screenwidth() // 2) - (width // 2)
    y = (win.winfo_screenheight() // 2) - (height // 2)
    win.geometry('{}x{}+{}+{}'.format(width, height, x, y))


root = Tk()
root.title("Postavke sata")
root.minsize(480, 250)
root.configure(background='coral4')
if platform.system() == 'Windows':
    root.iconbitmap("./slike/ikona.ico")
center(root)


global arduino
def spoji():
    try:
        global arduino
        port = kucica_com.get("1.0",END)
        port = port[:-1] #brisanje znaka "\n" na kraju stringa
        arduino = serial.Serial(port, 115200)
        status.delete('1.0', 'end')
        status.insert('1.0', "Uspješno spojeno!")
    except:
        status.delete('1.0', 'end')
        status.insert('1.0', "Greška spajanja!")

def posalji():
    try:
        global arduino
        vrijeme = kucica_vrijeme.get("1.0",END)
        vrijeme = vrijeme[:-1] #brisanje znaka "\n" na kraju
stringa
        vrijeme = vrijeme.replace(":", "") #zamjena ":" s
praznim stringom
        time.sleep(1)
        arduino.write(bytes(vrijeme, 'utf8'))
        status.delete('1.0', 'end')
```

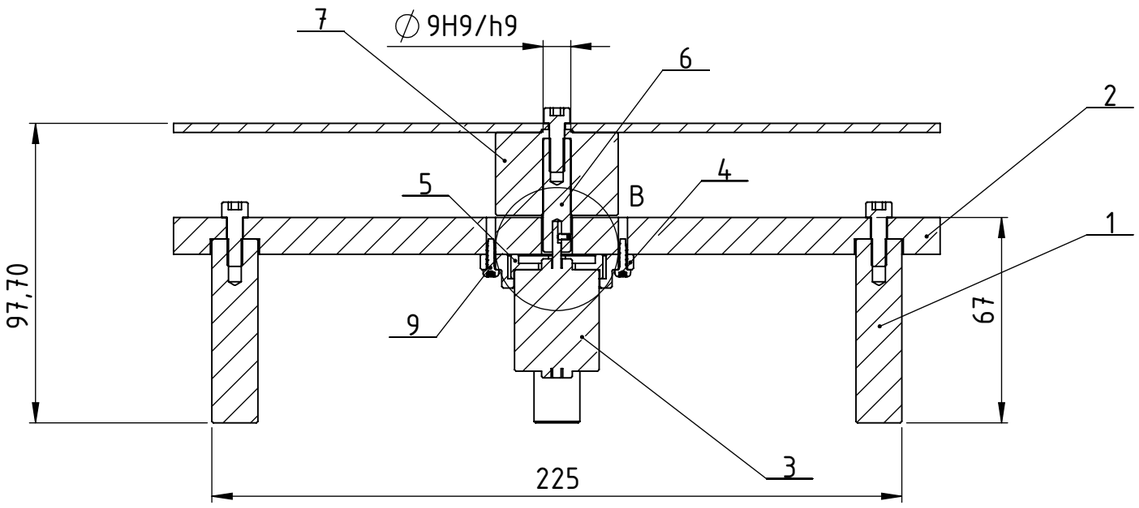
```
        status.insert('1.0', "Uspješno poslano!")
    except:
        status.delete('1.0', 'end')
        status.insert('1.0', "Greška slanja!")


pixel = PhotoImage(width=1, height=1)


kucice = Frame(root)
kucice.configure(background='coral4')
kucice.place(relx=0.30, rely=0.4, anchor=CENTER)

kucica_com = Text(kucice, wrap="word", height=1, width=20)
kucica_vrijeme = Text(kucice, wrap="word", height=1, width=20)
kucica_com.pack(pady=20)
kucica_vrijeme.pack(pady=20)

#Postavljanje pocetnih vrijednosti:
kucica_com.insert('1.0', 'COM4')
datum_i_vrijeme = datetime.now()
trenutno_vrijeme = datum_i_vrijeme.strftime("%H:%M:%S")
kucica_vrijeme.insert('1.0', trenutno_vrijeme)


gumbi = Frame(root)
gumbi.configure(background='coral4')
gumbi.place(relx=0.75, rely=0.4, anchor=CENTER)

button1 = Button(gumbi, text="Poveži", font="Helvetica 16
bold", command=spoji, image=pixel, height=50, width=150,
compound="c")
button2 = Button(gumbi, text="Učitaj", font="Helvetica 16
bold", command=posalji, image=pixel, height=50, width=150,
compound="c")
button1.pack(pady=10)
button2.pack(pady=10)

status = Text(root, height=1, width=20, bg='coral4',
fg='white', bd=0, highlightthickness=0)
status.insert(END, "--status--")
status.tag_add("poravnanje", "1.0", "end")
status.pack(side=LEFT, anchor=SW, padx=20, pady=30)

potpis = Text(root, height=2, width=20, bg='coral4',
fg='white', bd=0, highlightthickness=0)
potpis.tag_configure("poravnanje", justify='right')
potpis.insert(END, "Završni rad\nDino Šeketa")
potpis.tag_add("poravnanje", "1.0", "end")
potpis.pack(side=BOTTOM, anchor=SE, padx=20, pady=20)
root.mainloop()
```





Detalj B
Mjerilo 1:1





Presjek A-A

Poz.	Naziv dijela	Kom.	Crtež broj / Norma	Materijal	Sirove dimenzije / Proizvođač	Masa
11	Vijak M2x4	1	DIN914			
10	Vijak M5x16	5	DIN912			
9	Samourezni vijak M3	4	ISO14585C			
8	Nosač PCB-a	1		Polikarbonat		
7	Valjak	1		Poliacetal		
6	Produžetak vratila	1		CuZn10		
5	Prsten	1		Poliamid		
4	Prirubnica	1		Poliamid		
3	Motor	1	RS-365-SA			
2	Disk	1		Drvo		
1	Nožica	4		Al6061		

Broj naziva - code		Datum	Ime i prezime	Potpis
	Projektirao	07.02.2020.	Dino Šeketa	
	Razradio	07.02.2020.	Dino Šeketa	
	Crtao	07.02.2020.	Dino Šeketa	
	Pregledao			
	Voditelj rada		prof. dr. sc. Danijel Pavković	

FSB Zagreb

ISO - tolerancije		Objekt:	Objekt broj:
∅9 H9/h9	0,036 / -0,036		R. N. broj:
		Napomena:	Smjer: Mehatronika i robotika
		Materijal:	Masa: 0,55 kg
		Mjerilo originala M1:2	Naziv: Rotirajući LED sat
			Pozicija: 00
			Crtež broj: P-0000-00

Kopija

Format: A3

Listova: 1

List: 1

0 10 20 30 40 50 60 70 80 90 100

Design by CADLab