TEHNIČKA ŠKOLA RUĐERA BOŠKOVIĆA  
GETALDIĆEVA 4, ZAGREB

ZAVRŠNI STRUČNI RAD:

Digitalni sat izveden s Arduinom

MENTOR: UČENIK: Kristijan Jagunec

Zoran Mutak RAZRED: 4.G

Zagreb, travanj 2021.

Sadržaj

[1. Uvod 1](#_Toc70358594)

[2. Teorijski dio 2](#_Toc70358595)

[2.1. Povijest satova 2](#_Toc70358596)

[2.1.1 Pješčani sat 2](#_Toc70358597)

[2.1.2. Sunčani sat 3](#_Toc70358598)

[2.1.3. Mehanički sat 3](#_Toc70358599)

[2.1.4 Digitalni sat 4](#_Toc70358600)

[2.2. Arduino 7](#_Toc70358601)

[2.2.1. Arduino IDE 8](#_Toc70358602)

[2.2.2. Arduino Uno 10](#_Toc70358603)

[2.3. TM1637 četveroznamenkasti 7-segmentni LED indikator 12](#_Toc70358604)

[2.4. RTC DS3231 13](#_Toc70358605)

[3. Opis rada 14](#_Toc70358606)

[3.1. Sklopovlje 14](#_Toc70358607)

[3.2. Kod 15](#_Toc70358608)

[3.3. Testiranje sklopa 16](#_Toc70358609)

[4. Zaključak 19](#_Toc70358610)

[Literatura 20](#_Toc70358611)

[Cijeli kod 21](#_Toc70358612)

# 1. Uvod

Čovječanstvo je oduvijek imalo potrebu bilježiti periode vremena. Bilo to dugi periodi poput godišnjih doba do dana i sati. Najlakši način za prikaz vremena je pomoću sata. Kroz vrijeme su se koristili razni satovi; sunčani, pješčani, analogni, mehanički… U današnje vrijeme su ipak najčešći oni digitalni. Možemo ih naći posvuda, od računala i mobilnih uređaja do ručnih i zidnih satova. Upravo ta rasprostranjenost nam pokazuje veliku potrebu za satovima.

Ovaj završni rad prikazuje izradu jednog takvog digitalnog sata s izvedbom pomoću Arduino platforme. Vrijeme se pokazuje na 7-segmentnom LED zaslonu s 4 znamenke. RTC (*Real Time Clock*) broji vrijeme od zadnjeg namještanja i šalje signal mikroupravljaču. Sat se napaja preko USB-a ili baterije od 9V koja se treba zamijeniti nakon potrošnje.

U nastavku će biti prikazana teorija potrebna za shvaćanje rada i zatim prikaz praktične izvedbe rada.

# 2. Teorijski dio

## 2.1. Povijest satova

Sat je naprava za mjerenje vremena. To može biti pomoću bilježenja prirodnih pojava poput pomaka Sunca na nebu i protoka vode. No s napretkom tehnologije, i satovi su napredovali. Ali krenimo redom počevši s pješčanim satom.

### 2.1.1 Pješčani sat

Pješčani sat se koristio do 19. st. tj. dok ga nisu zamijenili precizniji satovi. [1] Ovaj sat se sastoji od dvije staklene posude nasuprot jedna druge dok je između njih tanki prostor. Što je prolaznost tog prostora manja, to će pijesku trebati duže vremena da pređe iz jedne posude u drugu. To potrebno vrijeme je najveće vrijeme koje pješčani sat može mjeriti i ovisi o veličini posuda te količini i prolaznosti pijeska.



*Slika 2.1. Pješčani sat*

*Izvor: https://cijena-lako.cam/1949/images\_Pješčani-sat-pijesak-timer-poboljšati-performanse-6*

### 2.1.2. Sunčani sat

Sunčani satovi potiču iz Mezopotamije i ostalih drevnih civilizacija. Rade na principu pomicanja Sunca na nebu i bilježenja sjene tijekom cijelog dana. Mjesto gdje je sjena padala tijekom zore je početak, a mjesto gdje je sjena bila bi bio kraj. Prostor između se podijeli na 12 dijelova i svaki dio označava jedan sat. Ti satovi nisu uvijek pouzdani jer ovise o dobu godine. [1]



*Slika 2.2. Sunčani sat u Krku*

*Izvor: http://www.celestialscenes.com/sundials*

### 2.1.3. Mehanički sat

Prvi mehanički satovi su bili nepouzdani zbog raznih mehanizama prema kojima su radili (utezi, protjecanje vode itd.). Da bi se popravila pouzdanost, trebao se dodati oscilator. Prvi oscilator koji se koristio za te svrhe je bio mehanički oscilator tj. njihalo, a primijenio ga je Christiaan Huygens 1656. g. [1] Satovi s njihalom su bili izrazito točni i koristili su se sve do pojave elektroničnih satova o kojima ćemo pričati kasnije. Jedina mana takvog sata je što, zbog naravi njihala, nije mogao biti korišten u pokretu. I ta mana se riješila korištenjem opruge koja pokreće sat. Tada su se mogli raditi satovi za korištenje tijekom kretanja i tako su počeli džepni i potom ručni satovi. Satovi poznati po svojoj

točnosti se nazivaju kronometri.Mehanički satovi najčešće imaju dvije kazaljke, jednu veliku koja pokazuje minute i jednu malu koja pokazuje sate. Neki satovi imaju i još jednu tanku kazaljku koja pokazuje minute. Osovina s kazaljkama je u sredini kružnice na kojoj se nalaze brojevi od 1 do 12. Ti brojevi su najčešće arapski, ali mogu biti i rimski. Ako se na satu s rimskim brojevima broj 4 prikazuje s IIII, znamo da je to stariji sat.



*Slika 2.3. Mehanički zidni sat*

*Izvor: https://jysk.hr/kucanstvo/dekoracija/satovi/zidni-sat-tobias-o41cm-tihi-mehanizam*

### 2.1.4 Digitalni sat

Digitalni sat nema kazaljke nego prikazuje vrijeme znamenkama. Potreban mu je izvor napajanja, uglavnom baterija ili se spaja u utičnicu. Mana digitalnih satova bez vlastite baterije je što se moraju ponovno namještati u slučaju nestanka struje. Prikaz vremena je na LCD (*Liquid-Crystal Display*) ili LDE (*Light Emitting Diode*) zaslonu.

Prvi digitalni satovi su bili dijelom mehanički jer je sadržavao oprugu koja je na slijedeći sat pogurala disk na kojem su bile znamenke. Taj mehanizam se nazivao „jump hour“. *Jump hour* satovi su bili česti krajem devetnaestog stoljeća, a početkom dvadesetog su se počeli proizvoditi i ručni satovi s istim mehanizmom. [2]



*Slika 2.4. Jump hour*

*Izvor: https://www.chrono24.com/magazine/jump-hour-watches7916-p\_7916/*

Prvi električni sat je proizveden 1957. godine i proizvela ga je kompanija *The Hamilton Watch*. Taj sat je radio kao i svi prije njega osim malog dodatka koji je bio baterija za napajanje te je tako potreba za navijanjem zauvijek otišla. Problem s tim satom su bili kontakti koji su se brzo potrošili pa sat nije imao dug vijek trajanja. [3] Taj problem je riješen sa satom *Accutron* koji je imao tranzistor za manje trošenje. Sljedeći napredak u digitalnim satovima su otkrili Švicarci s kvarcnim oscilatorom. To su bili izrazito točni satovi s maksimalnim odstupanjem od 0.01 sekunde u danu. Prvi digitalni prikaz vremena je bio 1972. godine. Pritiskom na gumb se pojavilo vrijeme na ručnom satu tako da je za svaku znamenku bilo potrebno sedam LED dioda. Takvi satovi su bili izrazito skupi i tek su 1975. godine cijenom postali dostupni masama. Izradom kućišta od plastike je smanjilo cijenu satova koji bi inače bili od zlata. Otkrićem LCD tehnologije više nije bilo potrebno pritisnuti gumb da bi se vrijeme pokazalo nego je sat prikazivao vrijeme cijelo vrijeme. To je tad bilo moguće zbog manje energetske potrošnje LCD-a. [3]



*Slika 2.5. Digitalni sat s LCD zaslonom*

*Izvor: https://nedis.com/en-us/product/home-living/clocks-alarms/alarms/550674455/digital-desk-alarm-clock-backlight-grey*



*Slika 2.6. Digitalni sat s LDE zaslonom*

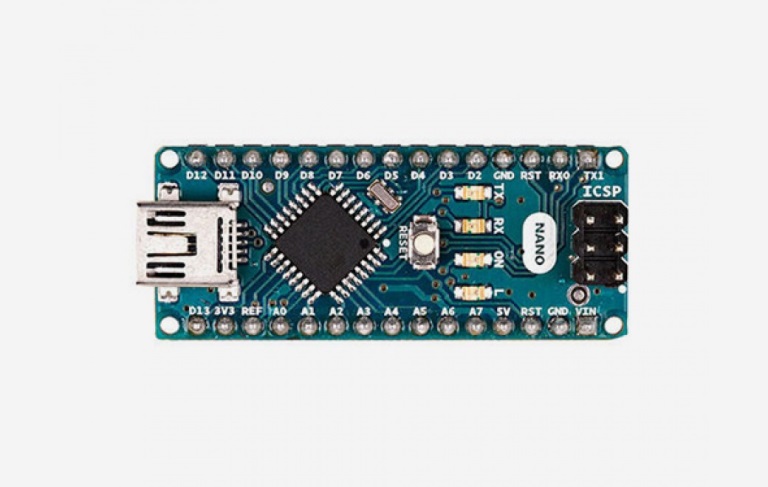
*Izvor: https://www.mobatime.com/product/dc-series/*

## 2.2. Arduino

Arduino je open-source platforma napravljena za lako korištenje u elektroničkim projektima. Mogu je koristiti početnici i učenici zbog svoje jednostavnosti, ali isto tako nudi i mnogo opcija za iskusnije. Podržava Windows, Mac i Linux. Arduino pločica može primati serijske i digitalne signale pomoću ulaznih pinova. Također može i davati signal pa tako aktivirati LED diodu, zaslon, motor itd. Spaja se s računalom preko USB kabela te je to jedan od načina za napajanje Arduina. Software se koristi pomoću Arduino programskog jezika i Arduino IDE (Integrated Development Environment). Programirati se može u C ili C++ jezicima. Postoji više vrsta Arduino pločica koje se razlikuju po svojim svojstvima, dimenzijama i opcijama. Tako imamo veći izbor i možemo lakše naći nešto što nam treba za određeni projekt na kojem radimo. Neki od njih su: Arduino Uno, Arduino Leonardo, Arduino Nano, Arduino Micro… [4]



*Slika 2.7. Arduino logo*



*Slika 2.8. Arduino Nano*

*Izvor: http://arduino.cc/*

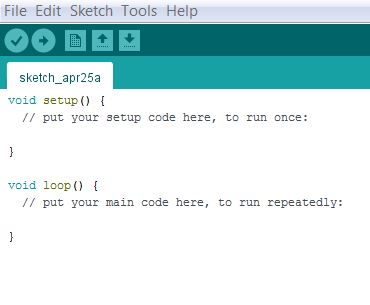
### 2.2.1. Arduino IDE

Arduino sofware za programsko upravljanje s mikroupravljačem. Može se koristiti za sve Arduino verzije. Programski jezik koji se koristi je pojednostavljena verzija C++ jezika. Korištenjem biblioteka (*engl. library*) si možemo olakšati i pojednostaviti programiranje. S obzirom na open-source narav platforme, lako možemo naći *library* koji nam je potreban. Instalacijom Arduino IDE programa su određene biblioteke već instalirane.

Primjer biblioteka:

* LiquidCrystal – za upravljanje LCD zaslonima
* Wire – za korištenje TWI i I2C protokola [5]
* Arduino (uglavnom nije potrebno pisati)
* Servo – za upravljanje servo motora
* RTC – za upravljanje Real Time Clock uređaja

Nakon pokretanja Arduino IDE aplikacije, vidimo sljedeći prikaz:



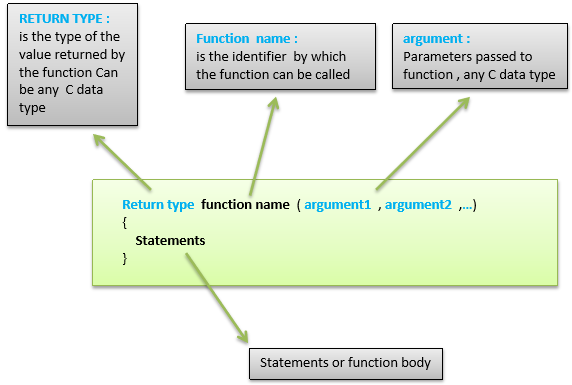
*Slika 2.9. Arduino IDE sučelje*

U void setup() funkciji upisujemo kod koji će se pokretati samo jedanput. Tu upisujemo kod za inicijalizaciju varijabli, definiramo opcije pinova, dodajemo biblioteke itd.

U void loop() funkciju upisujemo kod koji će se ponavljati. Kod koji ovdje upišemo će služiti za stalno kontroliranje mikroupravljača i mijenjanja određenih stanja po želji i potrebi.

Česte funkcije koje koristimo za upravljanje Arduina [6]:

* **digitalRead(**pin**)** – čita vrijednost s digitalnog pina (HIGH ili LOW)
* **digitalWrite(**pin, value**)** – upisuje vrijednost u digitalni pin (HIGH ili LOW)
* **pinMode(**pin, mode**)** – postavlja pin da se ponaša kao ulazni ili izlazni
* **analogRead(**pin**)** – čita vrijednost s analognog pina (od 0 do 1023)
* **analogWrite(**pin, value**)** – upisuje vrijednost u analogni pin
* **delay(**ms**)** – zaustavi program na zadano vrijeme
* **map(**value, fromLow, fromHigh, toLow, toHigh**)** – pretvara vrijednost iz jednog mjernog opsega u drugi
* **random(**min, max**)** – generira nasumični broj između zadanih minimuma i maksimuma
* **Serial** – koristi se za komunikaciju između Arduina i ostalih uređaja preko serijskih portova



*Slika 2.10. Sintaksa funkcije*

*Izvor: https://www.tutorialspoint.com/arduino/arduino\_functions.htm*

### 2.2.2. Arduino Uno

Najčešće korištena verzija Arduino pločice, Arduino Uno se koristi i u ovom projektu. Baziran je na ATmega328p. Na sebi ima 14 digitalnih i analognih pinova, 6 analognih, keramički rezonator od 16 MHz, USB port, port za napajanje, ICSP (*In-circuit serial programming*) port i gumb za resetiranje. [7]



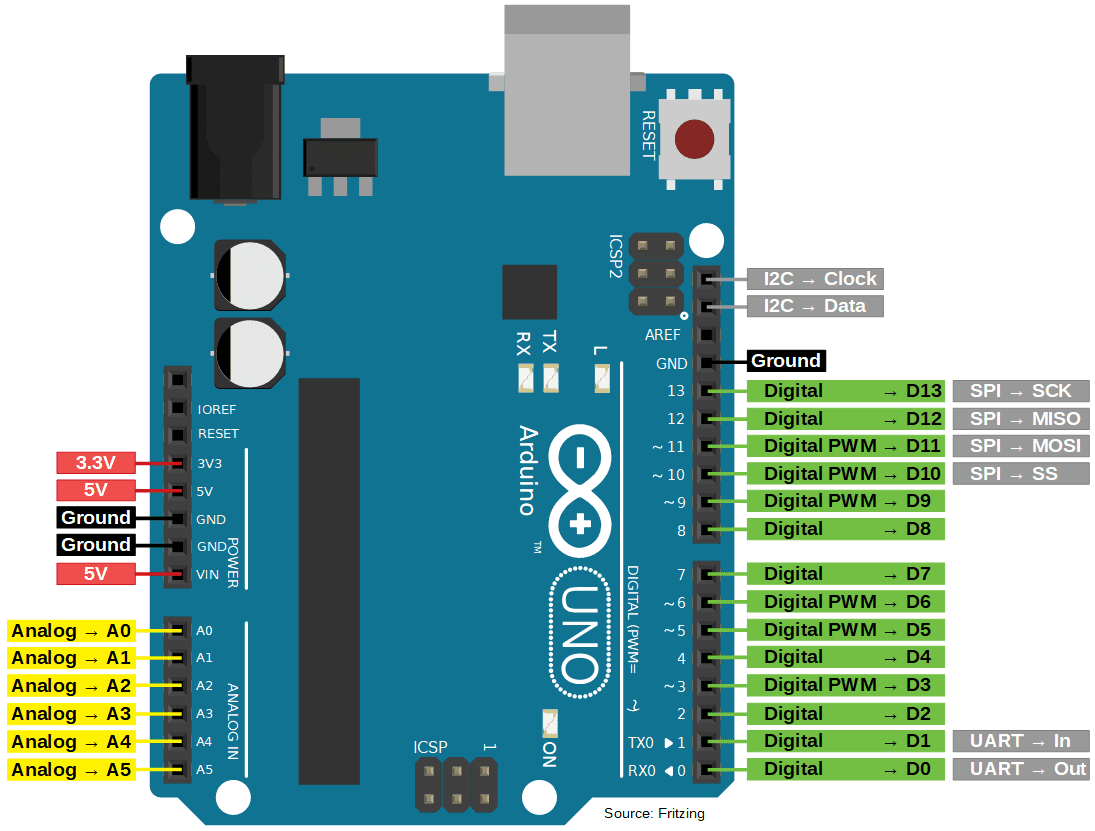
*Slika 2.11. Arduino Uno*

*Izvor: https://store.arduino.cc/arduino-uno-rev3*

|  |  |
| --- | --- |
| Microcontroller | ATmega328P |
| Operating Voltage | 5V |
| Input Voltage  (recommended) | 7-12V |
| Input Voltage (limit) | 6-20V |
| Digital I/O Pins | 14 (of which 6 provide PWM output) |
| PWM Digital I/O Pins | 6 |
| Analog Input Pins | 6 |
| DC Current per I/O Pin | 20 mA |
| DC Current for 3.3V Pin | 50 mA |
| Flash memory | 32 KB (ATmega328P) of which 0.5 KB  used by bootloader |
| SRAM | 2 KB (ATmega328P) |
| EEPROM | 1 KB (ATmega328P) |
| Clock Speed | 16 MHz |
| LED\_BUILTIN | 13 |
| Length | 68.6 mm |
| Width | 53.4 mm |
| Weight | 25 g |

*Slika 2.12. Arduino Uno specifikacije*

*Izvor: https://store.arduino.cc/arduino-uno-rev3*

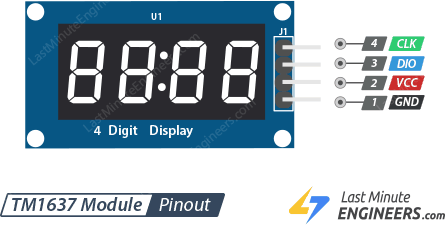


*Slika 2.13. Arduino Uno pinout*

*Izvor: https://diyi0t.com/arduino-uno-tutorial/*

## 2.3. TM1637 četveroznamenkasti 7-segmentni LED indikator

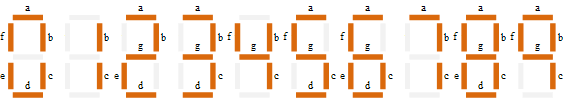
Pomoću ovog LED prikaza možemo prikazati 4 znamenke s opcijom dvotočke između druge i treće znamenke. Dok bi inače trebali 12 žica za spojiti zaslon s Arduinom, uz pomoć TM1637 nam treba samo 4. Zaslon ima 4 pina; ground, vcc, i 2 podatkovna pina CLK i DIO.



*Slika 2.14. TM1637 modul pinout*

*Izvor: https://lastminuteengineers.com/tm1637-arduino-tutorial/*

Svaka znamenka se sastoji od 7 dioda, od A do G. Kombinacijom paljenja pojedinih dioda možemo prikazati željeni broj.



*Slika 2.15. Prikaz 7 segmentnog zaslona*

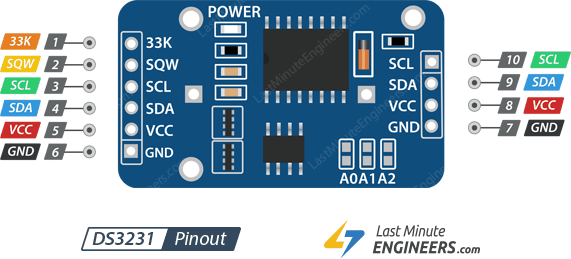
*Izvor: https://www.electronics-tutorials.ws/blog/7-segment-display-tutorial.html*

## 2.4. RTC DS3231

DS3231 je izrazito precizan Real Time Clock s kristalnim oscilatorom. Uređaj bilježi i broji sekunde, minute, sate, dan u tjednu, mjesec i godinu. Baterija služi kao sekundarni izvor napajanja u slučaju kvara s primarnim napajanjem.

DS3231 RTC ima 6 pinova;

* GND
* VCC
* SDA – serijski podatkovni pin za I2C protokol
* SCL – serijski pin za sat za I2C protokol
* SQW – izlazni pin koji šalje pravokutni impuls određene frekvencije
* 32K – izlazni pin koji šalje stabilni signal sata [8]

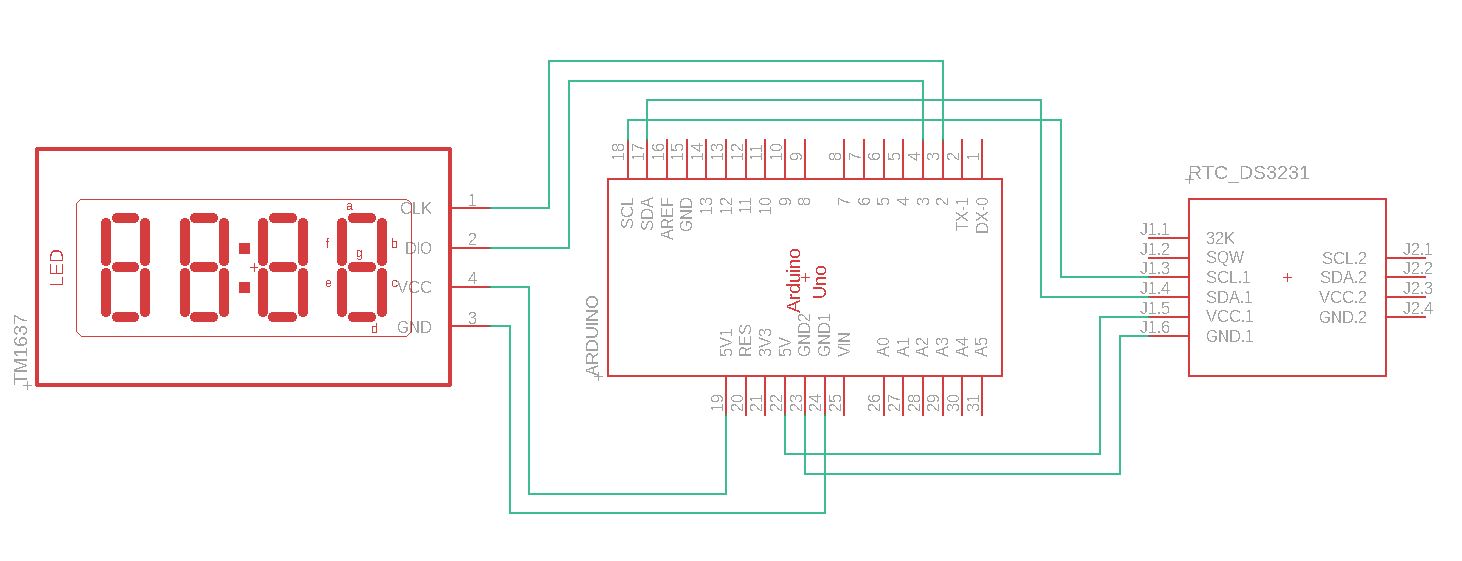


*Slika 2.16. DS3231 RTC pinout*

*Izvor: https://lastminuteengineers.com/ds3231-rtc-arduino-tutorial/*

# 3. Opis rada

## 3.1. Sklopovlje



*3.1. Shema sklopa*

Komponente korištene za realiziranje sklopa (slijeva nadesno):

* TM1637 četveroznamenkasti 7-segmentni LED indikator
* Arduino Uno
* RTC DS3231

RTC DS3231 broji vrijeme od zadnjeg podešavanja. Spojen je na GND i 5V pinove na Arduinu, ali ima i vlastitu bateriju u slučaju prekida napajanja Arduina. Trenutno vrijeme se šalje na SDA i SCL pinove gdje ih mikroupravljač prima. Zatim Arduino obrađuje te podatke i preko digitalnih pinova 2 i 3 šalje informaciju zaslonu o tome koje diode treba upaliti.

## 3.2. Kod

**1:** #include <RTClib.h>  
**2:** #include <TM1637Display.h>

Kod započinje s učitavanjem biblioteka za upravljanje RTC-om i TM1637 prikazom.

**3:** int CLK = 2;  
**4:** int DIO = 3;  
**5:** int prvi = 0;

Zatim definiramo pin 2 i pin 3 kao pinove CLK i DIO. Također je potrebno definirati jednu varijablu koja će nam trebati kasnije.

**6:** TM1637Display display(CLK, DIO);  
**7:** RTC\_DS3231 rtc;

Ovdje definiramo prikaz i nazivamo ga *display* te definiramo pinove za komunikaciju s TM1637. Definiramo RTC i nazivamo ga *rtc*.

**8**: void setup() {  
**9:** rtc.begin();  
**10:** // rtc.adjust (DateTime(2021, 4, 17, 14, 20, 0));  
**11:** display.setBrightness(0);  
**12:** display.clear();  
**13:** }

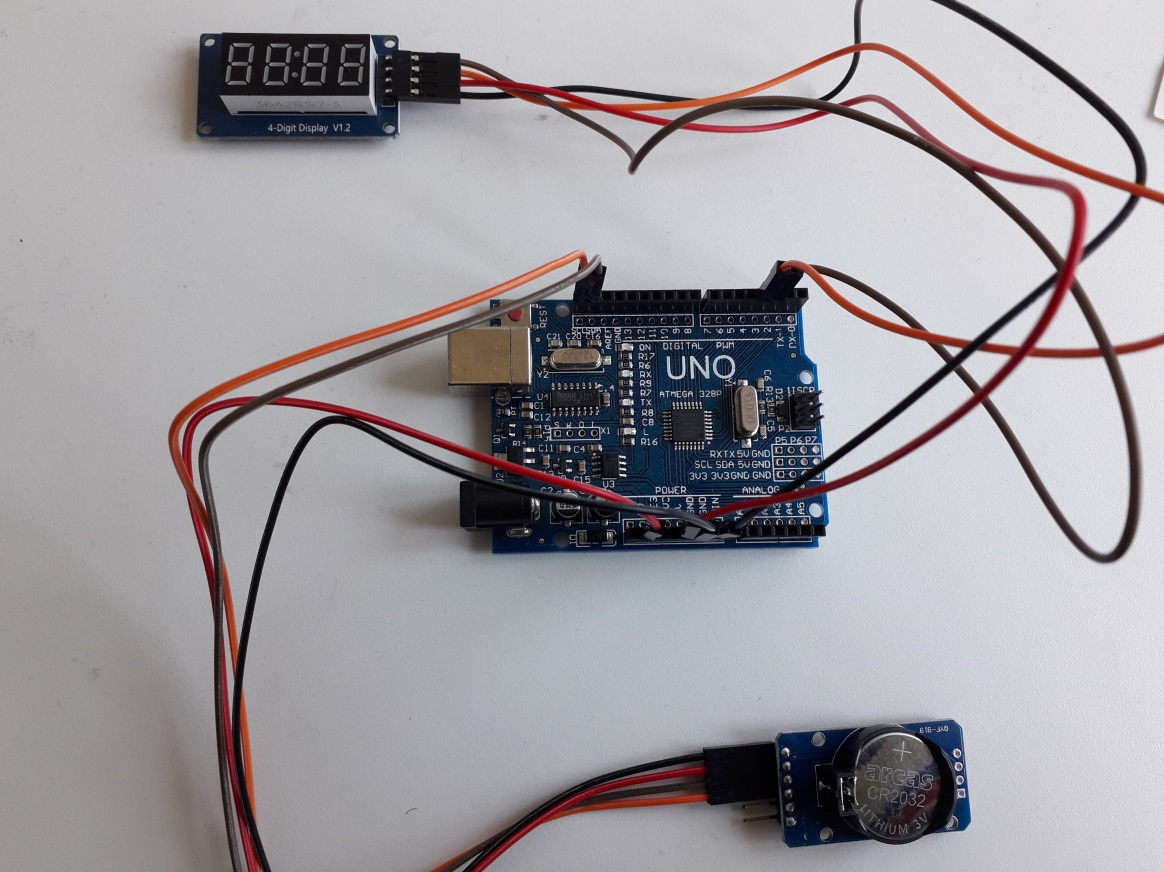
U void setup funkciji prvo inicijaliziramo RTC. Zatim, ako smo RTC spojili prvi put, trebamo namjestiti vrijeme. Tu liniju koda mičemo nakon prvog učitavanja koda. Onda postavljamo svjetlinu LED zaslona na najmanju i brišemo prijašnje postavljanje zaslona (za slučaj da ga je bilo).

**14:** void loop() {  
**15:** DateTime now = rtc.now();  
**16:** if (now.second() == 0 || prvi == 0) {  
**17:** display.showNumberDecEx(now.hour(), 0b01000000, true, 2, 0);  
**18:** display.showNumberDecEx(now.minute(), 0b01000000, true, 2, 2);  
**19:** prvi = 1;  
**20:** }  
**21:** }

U void loop funkciji prvo definiramo varijablu *now* kao varijablu za vrijeme i u spremili povratnu vrijednost funkcije *rtc.now*. Ta funkcija nam govori koliko je trenutno sati. Onda imamo *if* funkciju koja izvodi kod koji je u njoj upisan samo u slučaju kada se promijeni minuta ili kad se sat prvi put pali. Zatim u prvoj liniji koda u *if-u* ispisujemo trenutni sat na nultoj poziciji s dvotočkom. U drugoj liniji ispisujemo minute na drugoj poziciji na zaslonu uz ispis dvotočke. Na kraju postavljamo vrijednost varijable *prvi* u 1 te zatvaramo *if*.

## 3.3. Testiranje sklopa

Za realizaciju sklopa je prvo potrebno spojiti Arduino sa zaslonom i RTC-om.



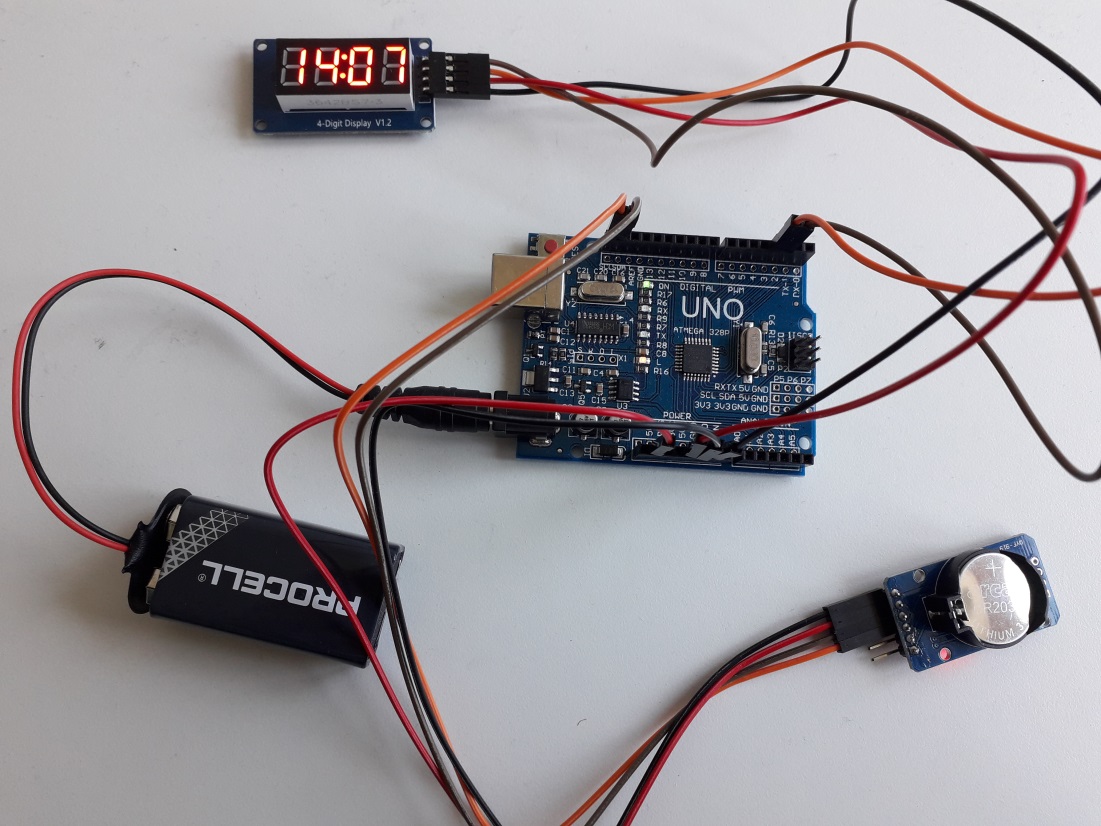
*Slika 3.2. Spoj Arduina sa zaslonom i RTC-om bez baterije*

S obzirom na to da Arduino nije spojen na računalo preko USB-a, potreban mu je vanjski izvor napajanja. Za to će nam poslužiti baterija od 9V. Nju ćemo spojiti s adapterom na utikač od 2.1mm.



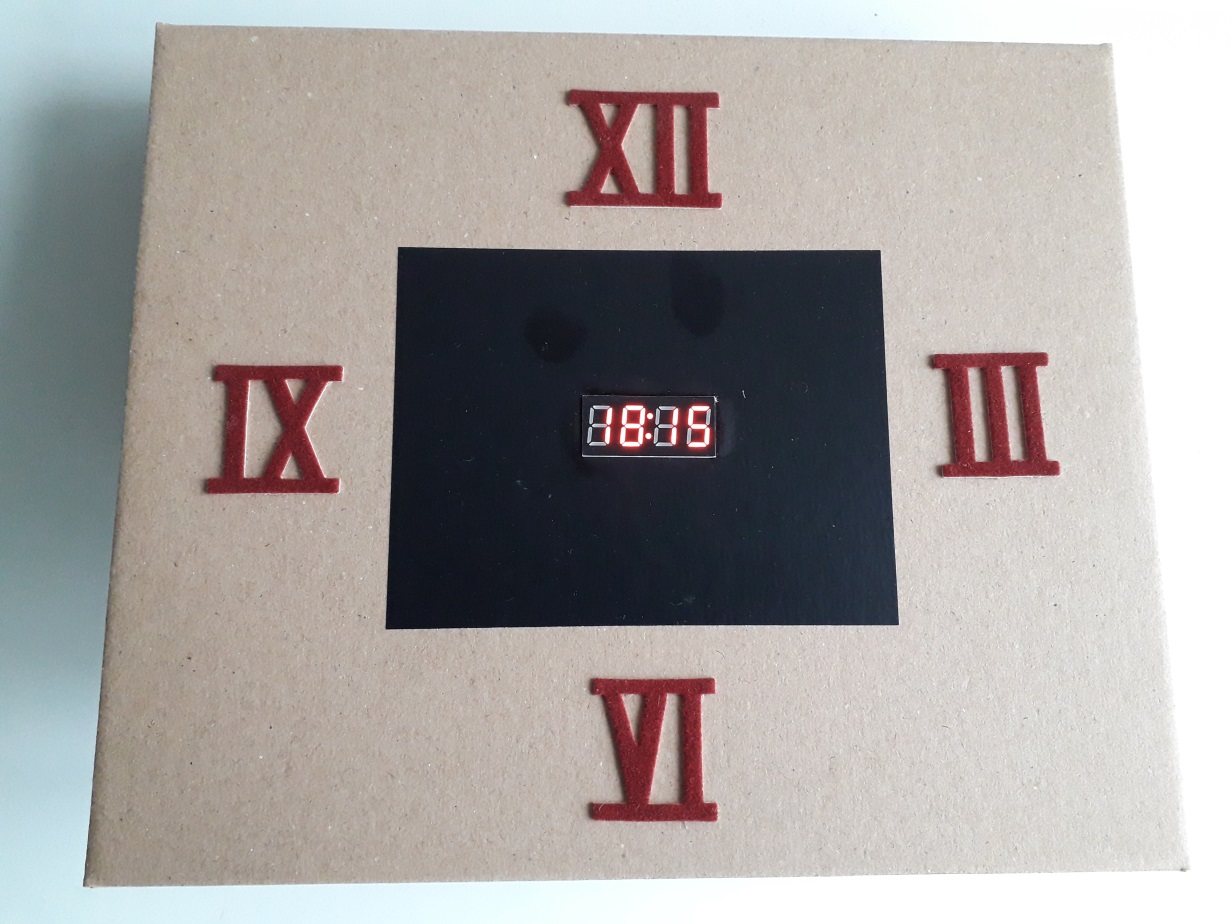
*Slika 3.3. Baterija s adapterom*

Gotov sklop s napajanjem je funkcionalan i preostala je još izrada kućišta.



*Slika 3.4 Spoj s napajanjem*

Kućište je izrađeno od troslojne valovite ljepenke. Taj materijal je odabran zbog niske cijene i lake dostupnosti. Još jedna dobra strana je što se materijal može reciklirati. Komponente su pričvršćene uz dno kutije pomoću vijaka i matica, osim baterije koja je u svom džepu za lakšu promjenu i zaslona koji je pričvršćen za poklopac. Na poklopcu se nalazi pravokutna rupa za zaslon. Kutija je dimenzija 23 mm u širini, 4,5 mm u dužini i 19,2 mm u visini. Sat također ima opciju da se postavi u okomit položaj.



*Slika 3.5. Prikaz završnog rada*

# 4. Zaključak

Cilj ovog rada je bila izrada digitalnog sata pomoću Arduina. Taj cilj je funkcionalno ostvaren. Za izradu sata je bilo potrebno mnogo istraživanja o potrebnim komponentama i isto tako gdje ih naći. Isto tako je trebalo primijeniti nove biblioteke u Arduino IDE programu i shvatiti njihov način korištenja. Nakon svog istraživanja se vidi da sa svim dostupnim opcijama možemo dodati još raznih opcija. Tek smo dotaknuli površinu. Također možemo vidjeti koliko se izrada satova promijenila od starih sunčanih satova, ne tako starih mehaničkih te sada ovakvih digitalnih satova. Funkcionalnost sata ovisi o načinu korištenja i dugotrajnosti komponenata. To će se najvjerojatnije najprije pokazati s kućištem koje je napravljeno od kratkotrajnog materijala. No to se nebi trebalo pokazati kao veliki problem s obzirom na to da je ovaj digitalni sat namijenjen za upotrebu na stolu ili zidu.

# Literatura

[1] sat. Hrvatska enciklopedija, mrežno izdanje. Leksikografski zavod Miroslav Krleža, 2021. (pristupljeno 24. 4. 2021.)  
URL: http://www.enciklopedija.hr/Natuknica.aspx?ID=54679

[2] *Mechanical Digital Watch History.* 2015.   
URL: http://www.uniquewatchguide.com/mechanical-digital-watches.html,  
(pristupljeno: 24. 4. 2021.)

[3] *The History of the Digital Watch*. 29. travanj, 2003.   
URL: https://h2g2.com/edited\_entry/A1006534, (pristupljeno: 24. 4. 2021.)

[4] *What is Arduino?*. 5. veljače, 2018.   
URL: https://www.arduino.cc/en/Guide/Introduction, (pristupljeno: 25. 4. 2021.)

[5] *Libraries*. URL: https://www.arduino.cc/en/reference/libraries,  
(pristupljeno: 25. 4. 2021.)

[6] *Language Reference*. URL: https://www.arduino.cc/reference/en,   
(pristupljeno: 25. 4. 2021.)

[7] *Arduino Uno Rev3*. URL: https://store.arduino.cc/arduino-uno-rev3,  
(pristupljeno: 25. 4. 2021.)

[8] *Interface DS3231 Precision RTC Module with Arduino.*   
URL: https://lastminuteengineers.com/ds3231-rtc-arduino-tutorial,   
(pristupljeno: 25. 4. 2021.)

# Cijeli kod

**1:** #include <RTClib.h>  
**2:** #include <TM1637Display.h>

**3:** int CLK = 2;  
**4:** int DIO = 3;  
**5:** int prvi = 0;

**6:** TM1637Display display(CLK, DIO);  
**7:** RTC\_DS3231 rtc;

**8**: void setup() {  
**9:** rtc.begin();  
**10:** // rtc.adjust (DateTime(2021, 4, 17, 14, 20, 0));  
**11:** display.setBrightness(0);  
**12:** display.clear();  
**13:** }

**14:** void loop() {  
**15:** DateTime now = rtc.now();  
**16:** if (now.second() == 0 || prvi == 0) {  
**17:** display.showNumberDecEx(now.hour(), 0b01000000, true, 2, 0);  
**18:** display.showNumberDecEx(now.minute(), 0b01000000, true, 2, 2);  
**19:** prvi = 1;  
**20:** }  
**21:** }