Praćenje aktivnosti

Uglješić, Stefan

Undergraduate thesis / Završni rad

2022

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:833381

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-05-06**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA

Sveučilišni studij

PRAĆENJE AKTIVNOSTI

Završni rad

Stefan Uglješić

Osijek, 2021

SADRŽAJ

1. UVOD 1

1.1. Zadatak završnog rada 1

2. POSTOJEĆA RJEŠENJA 2

2.1. Važnost praćenja aktivnosti 2

2.2. Primjeri programa za praćenje aktivnosti 2

3. DIZAJN I IZRADA MODELA 4

3.1. Virtualni model uređaja 4

3.2. 3D ispisivanje modela vremenske kocke 6

4. SKLOPOVLJE 9

4.1. Arduino Nano 9

4.2. Litijeva 18650 baterija 9

4.3. TP4056 modul za punjenje litijeve baterije 10

4.4. Indikator kapaciteta litijeve baterije 11

4.5. Pretvarač napona *step-up* (2,5 V – 5 V) 12

4.6. Žiroskop GY-521 12

4.7. *Bluetooth* AT-09 BLE modul 12

4.8. RGB diode 13

5. PROGRAMSKA PODRŠKA UREĐAJA 15

5.1. Arduino programska podrška 15

5.1.1. Određivanje rotacije kocke 16

5.1.2. *Bluetooth* komunikacija 17

5.1.3. Upravljanje RGB diodama 18

5.2. Opisivanje MIT app Android dijela koda 19

5.2.1. Upravljanje *bluetoothom* 20

5.2.2. Upravljanje radom kocke 23
5.2.3. Upravljanje podacima i pohrana 23
5.2.4. Izgled aplikacije 26
6. TESTIRANJE RADA UREĐAJA I MOBILNE APLIKACIJE 28
7. ZAKLJUČAK 30
LITERATURA 32
SAŽETAK 34
ABSTRACT 35

1. UVOD

Važnost praćenja aktivnosti poprima značajan dio svakodnevnog života modernog čovjeka. Razvojem brojnih tehnologija, javila se i potreba za načinom na koji bi se potencijalno moglo pratiti vrijeme utrošeno na izvođenje pojedinih aktivnosti primjerice spavanja, učenja, odmora ili rekreacije. Danas postoje razni računalni programi i mobilne aplikacije koje su namijenjene optimizaciji i uvidu u potrošnju vremena na izvođenje nekih od aktivnosti. Osim praćenja navedenih dnevnih aktivnosti, moguće je pratiti i vrijeme utrošeno na rješavanje određenih problema u sklopu radnog vremena pa se ova rješenja često koriste i u tu svrhu. Neke od poslovnih aktivnosti koje je moguće pratiti su efektivno vrijeme utrošeno na posao, vrijeme za pauzu ili mogućnost organiziranja vremena prilikom rada od kuće.

Između ostalog, razvijena je ideja tzv. vremenske kocke (engl. *time cube*) koja je spoj hardverskih i softverskih rješenja potrebnih za jednostavno praćenje aktivnosti [1], što je ujedno i predmet ovog završnog rada.

Nakon uvodnog poglavlja u kojem je dan uvod u temu završnog rada, u drugom poglavlju opisana su postojeća rješenja i aplikacije za praćenje aktivnosti, treće poglavlje opisuje način na koji je izrađen „skelet“ vremenske kocke te njegova implementacija u realnom prostoru. Detaljniji opis elektroničkih dijelova korištenih za izradu kocke opisan je u četvrtom poglavlju. Sljedeća dva poglavlja (peto i šesto) opisuju softversko rješenje te izradu mobilne aplikacije kojom se upravlja radom hardvera. U posljednjem poglavlju opisan je testiranja kocke i mobilne aplikacije.

1.1. Zadatak završnog rada

Predmet ovog završnog rada je dizajnirati i izraditi vremensku kocku kojom bi se pratilo vrijeme koje je korisnik utrošio na izvođenje neke aktivnosti. Nadalje, cilj je osmisliti kocku što jednostavnijom za uporabu kako bi krajnji korisnik uz što manje potrošenog vremena i resursa dobio što više informacija o svojim aktivnostima.

2. POSTOJEĆA RJEŠENJA

Niz je primjera aplikacija, računalnih programa ili tehnoloških „*gadget*-a“ koji za cilj imaju pomoći korisniku pri praćenju aktivnosti. U tekstu koji slijedi izdvojeni su tek oni koji su postigli zapažen komercijalni rezultat ili prodaju.

2.1. Važnost praćenja aktivnosti

Praćenje aktivnosti ima velik značaj u današnjici kada svi prate sat i gledaju na svaku minutu uloženog vremena. Svima su poznate aplikacije za pametne mobitele koje govore koliko je potrošenog vremena na određenu aktivnost, recimo koliko dugo vremena je korisnik gledao filmove ili koliko dugo je pričao na telefon. Svim tim aplikacijama za cilj je da korisnik ima bolji pregled aktivnosti i s obzirom na to promjeni svoje navike, organizira svoje vrijeme te poboljša svoju efikasnost.

Praćenje aktivnosti u sportu i rekreaciji je vrlo bitna stavka i omogućuje uvid u duljinu i tip vježbi koje su odrađene u nekom razdoblju. Produljenjem trajanja vježbi korisnik može postepeno doći do boljih postignuća.

2.2. Primjeri programa za praćenje aktivnosti

U ovom poglavlju bit će navedeno nekoliko značajnijih primjera sličnih projekata te su prikazane njihove osnovne karakteristike.

Jedan od najboljih primjera je „*Time Tracking Cube*“ [1] kojeg su dizajnirali Brent Rubell i Ruiz Brothers. Uređaj omogućuje praćenje vremena kao što je šetanje psa ili jednostavno ispijanje jutarnje kave. Korisniku pruža pohranjivanje podataka u Microsoft Excel gdje postoji mogućnost pravljenja grafova i raznih vizualnih reprezentacija podataka. U svojem projektu kao osnovu koriste *Adafruit Feather* HUZZAH s ESP8266, cjelovitom WiFi razvojnom pločom, ugrađenim USB-om i punjačem baterije. *Prop-Maker Wing* dodaje *NeoPixel* podršku i zvučne efekte. Model „*Time Tracking Cube*“ napravljen je od osam dijelova i ima mogućnost promjene stranica u ovisnosti o željenim aktivnostima.

Drugi primjer je „*Timeflip 2*“ [2] koji ima oblik dodekaedra i time ima veću ponudu aktivnosti koje korisnik može odabrati u određenom trenutku. Uređaj pokreću dvije AA baterije, ima mogućnost paljenja na dodir, te unutarnju memoriju koja sprema podatke ako veza s drugim uređajem nije

uspostavljena. Komunikacija između uređaja i mobitela zasniva se na *Bluetooth LE* vezi koja za svrhu ima smanjenje potrošnje energije i produljenje trajanja baterija. Aplikacije nude mogućnost spajanja na korisnikov uređaj i pri čemu uređaj šalje sve zabilježene aktivnosti. Korisnik ima detaljan uvid u sve nedavno izvedene aktivnosti i u ovisnosti o tome može bolje organizirati svoje vrijeme.

Tvrtka *Microsoft* u sklopu svog programa za obradu podataka *MS Excel* pruža preuzimanje proširenja za praćenje aktivnosti (eng. *Activity tracker*) kojim je omogućeno praćenje sagorijevanja kalorija izvođenjem pojedinih vježbi. Ovo proširenje daje na uvid vrijeme kada je odrađena neka aktivnosti, njeno trajanje i količina kalorija koja je pri tome utrošena, a sve u sklopu vizualno ugodnih tablica u kojima je to prikazano. Nadalje od ponuđenih predložaka vrijedno za izdvojiti je „*Home gym tracker*“, proširenje kojim se prati vrijeme utrošeno na izvođenje pojedine vježbe u teretani. [3]

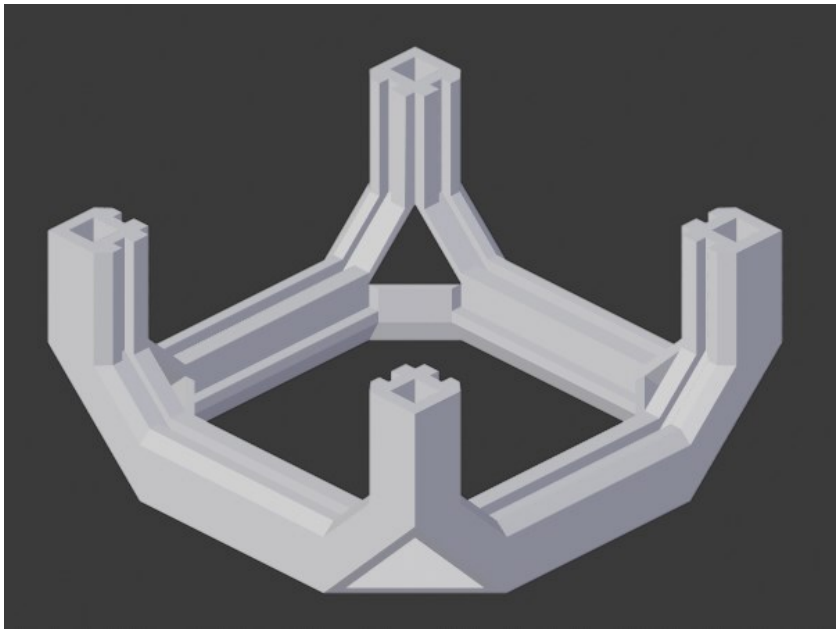
Unazad nekoliko godina veliki proizvođači mobilnih uređaja i tehnike u ponudu su stavili nova rješenja za praćenje aktivnosti. Predstavljeni su pametni satovi i pametne narukvice koje imaju mogućnost povezivanja s pametnim telefonom, zaslon na dodir, prikaz sata i vremena. Uz to korisnik ima mogućnost praćenja vitalnih funkcija kao što su frekvencija srca, saturacija krvi kisikom, EKG-a i sl., pa sve do brojanja pređenih koraka i utrošenih kalorija. Danas je teško zamisliti pametan telefon bez pripadajućeg pametnog sata ili narukvice pa se klasični mehanički satovi sve više napuštaju. [4]

3. DIZAJN I IZRADA MODELA

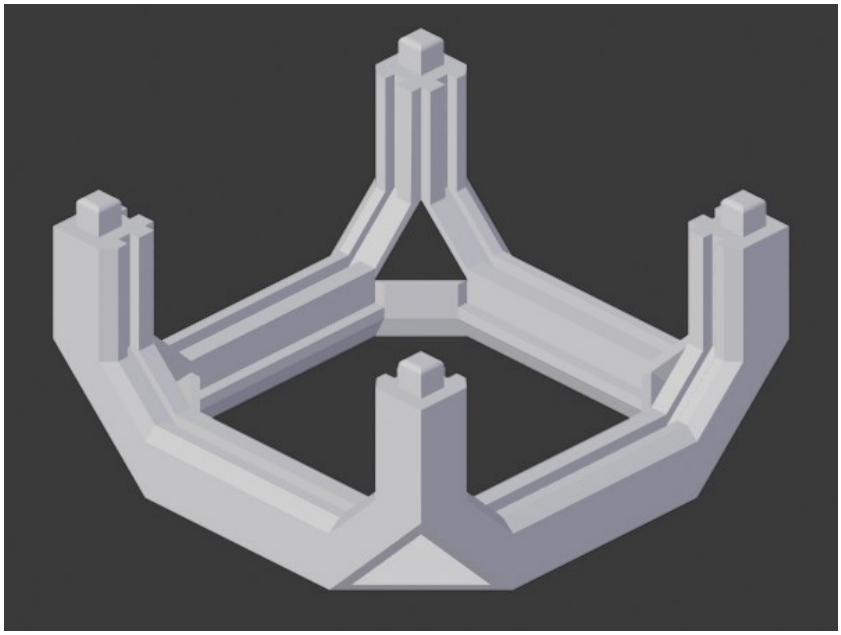
U ovom poglavlju opisan je dizajn kocke izrađen u programu *Blender*, te ispisivanje istog koristeći 3D printer. Model vremenske kocke prilagođen je što lakšoj primjeni za krajnjeg korisnika uz vizualnu i estetsku profinjenost.

3.1. Virtualni model uređaja

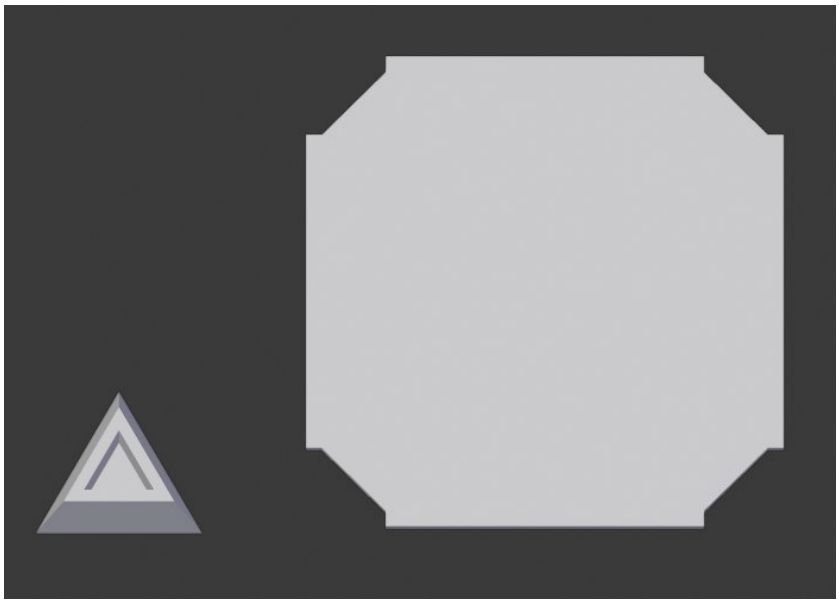
Uređaj se sastoji od dijelova koje čine vanjski (vidljivi dio kocke) i unutarnji dio kocke u kojem se nalaze svi dijelovi potrebni za njenu funkcionalnost. Vidljivi dijelovi kocke dizajnirani su koristeći program *Blender* koji omogućuje preciznu izradu virtualnih 3D modela koji se mogu 3D printati u željenim dimenzijama. Model kocke sastavljen je od: gornjeg dijela skeleta kocke (Slika 3.1.), donjeg dijela skeleta kocke (Slika 3.2.), 6 stranica i 8 trokuta koji zatvaraju kutove skeleta kocke (Slika 3.3.).



Slika 3.1. Model donjeg dijela skeleta kocke.

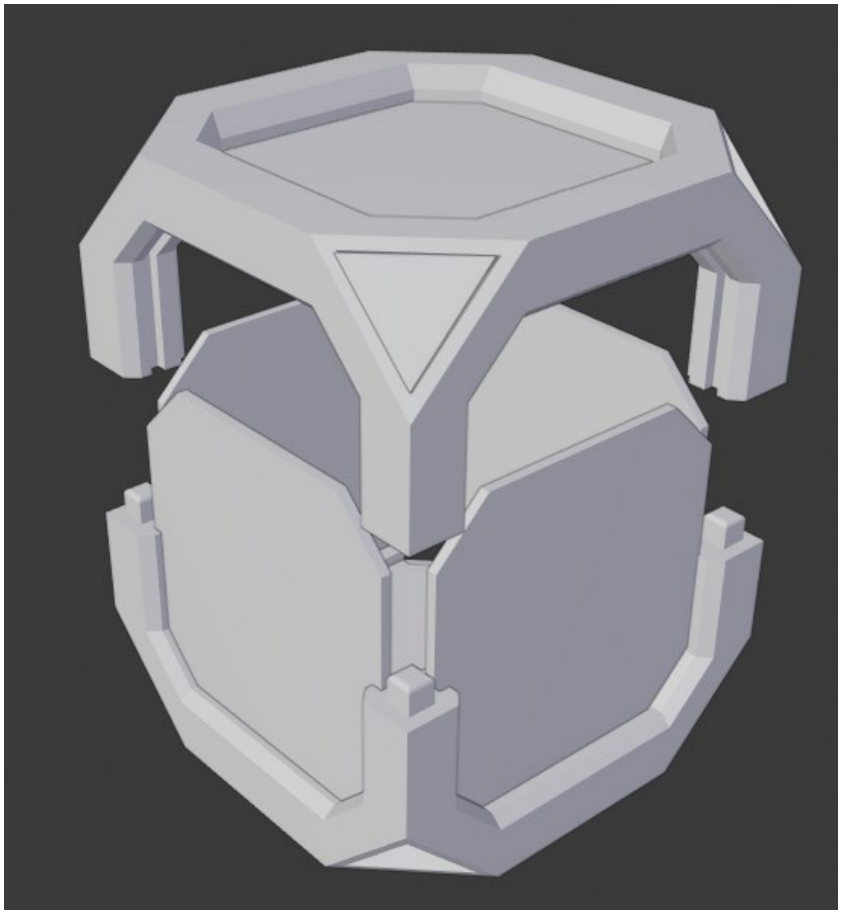


Slika 3.2. Model gornjeg dijela skeleta kocke.



Slika 3.3. Model trokuta i stranice kocke.

Trokut (Slika 3.3.) je dio koji zatvara uglove skeleta kocke pri čemu je izrađen tako da je djelomično transparentan kako bi se omogućio prolazak svjetlosti koju odašilju RGB diode. Stranice kocke (Slika 3.3.) se prema potrebama korisnika mogu mijenjati i prilagoditi individualnim potrebama. Primjerice, na stranicama kocke mogu se nalijepiti sličice koje simboliziraju aktivnosti za koje taj dio naličja kocke ima funkciju (npr. spavanje, učenje, čitanje, vrijeme provedeno na internetu…).



Slika 3.4. Model sastavljanja kocke.

Slika 3.4. prikazuje konačno sklapanje vanjskog dijela kocke gdje je vidljiva mogućnost da korisnik prilagodi raspored stranica svojim potrebama (stranice se lako vade i mijenjaju drugima). Dizajn je osmišljen tako da je sklapanje vrlo jednostavno pri čemu se dijelovi uklapaju u unaprijed urezane rubove a uglovi kocke se popunjavaju trokutastim dijelovima.

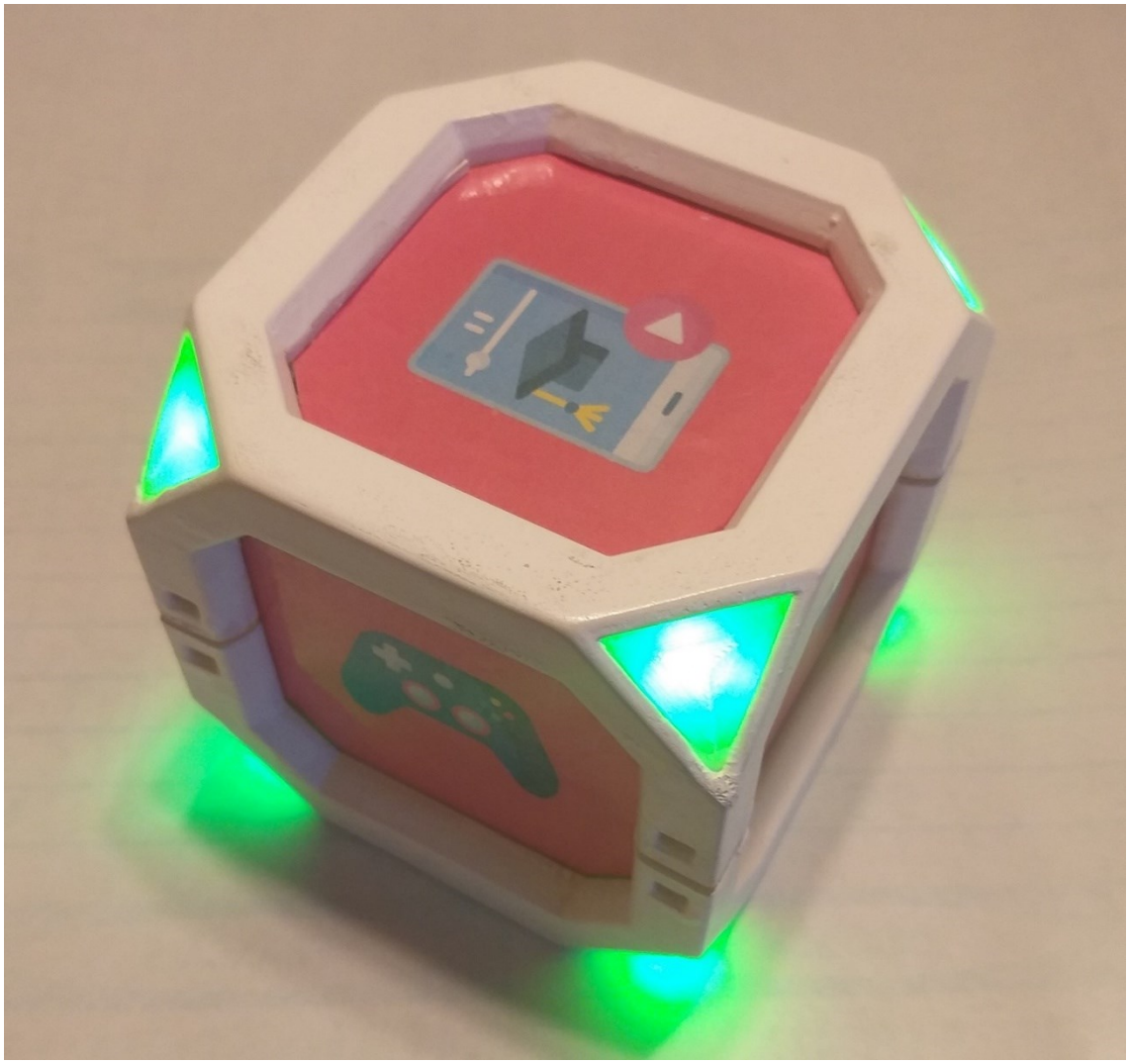
3.2. 3D ispisivanje modela vremenske kocke

Trodimenzionalni ispis (engl. *3D printing*) način je brze izrade prototipa, tako da se direktno iz 3D CAD programa izrađuju fizički predmeti u prostoru. Kako bi se neki model mogao 3D printati potrebna je prvo prilagodba koja se izvodi pomoću proširenja programa *Blender* - *Tool box*. Kako bi se moglo pristupiti ispisu potrebno je provjeriti nekoliko varijabli navedenih u nastavku.

1. „*Solid*“ prikazuje broj jednakih rubova u modelu i ističe te problematične rubove tako da ih se može vidjeti i popraviti. Rubovi koji nisu višestruki definirani su kao rubovi koje dijeli više od dva lica, što može dovesti do stvaranja praznina u modelu.
2. „*Intersections*“ prikazuje i ističe broj lica koja se sijeku u modelu.
3. „*Degeneration*“ prikazuje i ističe svaku deformiranu geometriju u modelu. Također se može podesiti prag degeneracije.
4. „*Distortion*“ prikazuje iskrivljena (ili neravna) lica u modelu. Također se može prilagoditi kut za koji se provjerava.
5. „*Thickness*“ označava broj tankih lica u modelu. Tanka lica mogu učiniti strukturu modela nestabilnom.
6. „*Edge Sharp*“ prikazuje broj oštrih rubova u modelu, što 3D pisačima može biti teško za ispis.
7. „*Overhang*“ označava broj visećih lica u modelu. Kad se ispišu ta lica, trebat će dodatne strukture potpore, pa se mogu ručno popraviti. Također je moguće promijeniti kut koji će izazvati „*overhang*“ [5].

Za pripremu modela prije ispisivanja potrebno je koristiti neki od „*slicer*“ softvera. „*Slicer*“ za 3D ispis - poznat i kao softver za rezanje ili pripremu ispisa - program je koji pretvara 3D model u jezik koji 3D pisač razumije. Softver za „*slicing*“ digitalno reže model na ravne slojeve koje pisač može ispisati jedan po jedan. Za izradu modela vremenske kocke upotrijebljen je *Ultimaker Cura „slicer*“ softver [6].

Za ispisivanje modela korišten je *Ultimaker 3S 3D* printer, a materijal od kojeg je kocka napravljena je PLA plastika žute boje. Nakon što je kocka isprintana, stranice kocke (Slika 3.3.) su obojene crvenom bojom dok je skelet kocke bojan bijelom bojom. Isprintani trokuti koji popunjavaju prostore između stranica i skeleta kocke (Slika 3.3.) obojani su transparentnim *fillerom*. Na svaku stranicu zalijepljena je unaprijed pripremljena naljepnica koja prikazuje određenu aktivnost (čitanje knjige, spavanje, razgovaranje telefonom, slušanje glazbe, treniranje, gledanje filmova). Konačni izgled kocke prikazan je na Slika 3.5.



Slika 3.5. Konačan izgled kocke.

4. SKLOPOVLJE

Cjelokupno sklopovlje smješteno je unutar kućišta kocke. Osnovu hardvera čini Arduino Nano razvojni sustav nadograđen modulima za žiroskop, *bluetooth* komunikaciju, litijevom baterijom kao i modulom za punjenje iste, „*stand up boost converter*“ od 5 V te 8 RGB dioda s pripadajućim sklopovljem za njihovo upravljanje.

4.1. Arduino Nano

Arduino Nano je mali razvojni sustav temeljen na mikroupravljaču ATmega328P. Proizveden je 2008. godine. Ima istu funkcionalnost kao i Arduino Uno ali je puno manji od svog prethodnika. Arduino Nano opremljen je s 30 muških I/O zaglavlja, u konfiguraciji sličnoj DIP30, koja se može programirati pomoću integriranog razvojnog okruženja Arduino Software (IDE), koje je zajedničko za sve Arduino ploče i radi i *online* i *offline*. Ploča se može napajati putem mikro-USB kabela tipa B ili iz 9 V baterije. Duljine je 45 mm, širine 18 mm a teži samo 7 grama [7].

Za izradu vremenske kocke Arduino Nano je iskorišten iz nekoliko razloga. Prvo, zbog svojih malih dimenzija pogodan je za lagano uklapanje unutar kocke, pri čemu ne doprinosi značajno njenoj masi. Drugo, energetska potrošnja Arduino Nano razvojnog sustava je izrazito mala, a dobivene performanse za iskorištenu energiju su značajne (*cost-benefit omjer*) [8]. Nadalje, ovaj razvojni sustav ima velik broj digitalnih pinova, što je bitno za mogućnost izrade ovog projekta (prvenstveno za mogućnost upravljanja RGB diodama). Budući da Arduino Nano nema mogućnost direktnog napajanja litijevom baterijom od 3,7 V, potrebno je ugraditi *boost* modul koji pretvara napon od 2-5 V u 5 V.

4.2. Litijeva 18650 baterija

Litij-ionska baterija (Li-Ion, LIB) pripada skupini punjivih baterija. U ovoj vrsti baterija tijekom pražnjenja litijevi ioni kreću se od negativne elektrode ka pozitivnoj. Suprotan se proces zbiva prigodom punjenja baterije. Ova vrsta baterija služi se inter kaliranim litijevim spojem kao jednom elektrodnom tvari, za razliku od metalnog litija koji se primjenjuje u nepunjivim litijevim baterijama. Stalne sastavnice litij-ionske ćelije su elektrolit koji dopušta kretanje iona te dvije elektrode. Konvencijski katoda je metalni oksid, anoda ugljik, a elektrolit litijeva sol u organskom topivu. Budući da je osnovna tvar vrlo lagana kovina litij, što uzrokuje da je baterija vrlo lagana. Prednost

ovakve vrste baterije je vrlo velika gustoća energije koja je skoro dvostruko veća od obične Ni MH baterije. Sljedeća prednost je tri puta veći nazivni napon od napona koji je u baterijama na bazi nikla (3,6 V prema 1,2 V). Treća velika prednost u odnosu na nikl baterije je što ju nije nužno održavati, jer nije podložna nikakvim štetnim efektima poput memorijskog štetnog efekta kad se ne koristi, a niti kristalizacijskog [9].

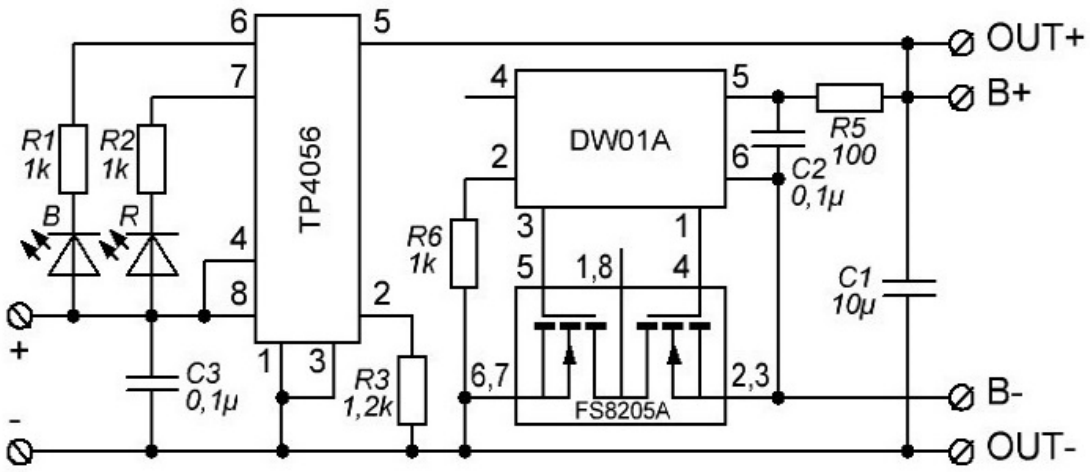
Daleko najpopularniji model cilindrične litijeve baterije je 18650 model (litij -ionska punjiva baterija). Pravo ime baterije je „*18650 cell*". Ćelija 18650 ima napon od 3,7 V i ima između 1800 i 3500 mAh. Navedene baterije mogu imati raspon napona između 2,5 i 4,2 V ili napon punjenja od 4,2 V, no nominalni napon standardnog 18650 iznosi 3,7 V. Odabir ove baterije za izradu vremenske kocke temeljio se na činjenici da je baterija vrlo malih dimenzija, lagana za koristiti, velikog kapacitet (u ovom slučaju 2000 mAh) te relativno povoljna na tržištu. Za punjene litijeve baterije koristi se TP4056 modul detaljnije objašnjen u nastavku rada.

4.3. TP4056 modul za punjenje litijeve baterije

Integrirani sklop TP4056 je litij-ionski punjač baterije za jednoćelijsku bateriju koji štiti ćeliju od prekomjernog napona i trošenja baterije ispod njenog minimuma. Ima dva statusna izlaza koji pokazuju da je punjenje u tijeku i da je punjenje završeno. Također ima programibilnu struju punjenja do 1 A. TP 4056 modul puni se preko USB tip C punjača.

Ovaj modul posjeduje sljedeće sigurnosne značajke:

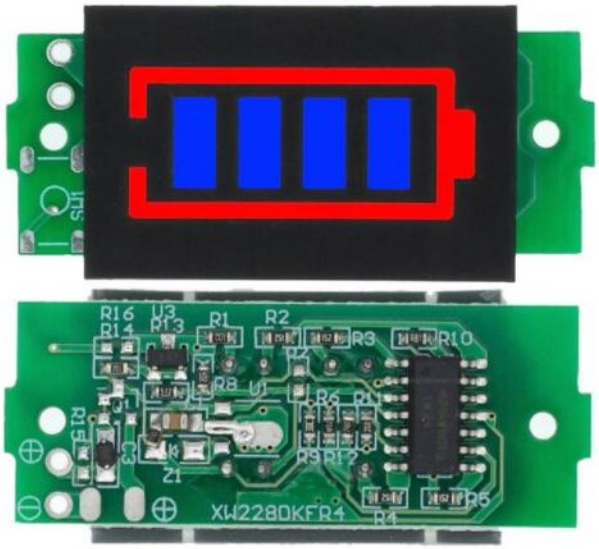
1. Upravlja punjenjem konstantne struje do konstantnog napona povezane litij-ionske baterije
2. Zaštita od prekomjernog pražnjenja - sprječava prekomjerno pražnjenje baterije ispod 2,4 V
3. Ako je spojena baterija ispražnjena ispod 2,4 V, modul će prekinuti izlaznu snagu baterije sve dok se napon baterije ponovno ne napuni iznad 3 V, što će u tom trenutku ponovno omogućiti pražnjenje snage baterije iz priključenog opterećenja. Iako modul isključuje napajanje iz baterije tijekom prekomjernog pražnjenja, ipak dopušta punjenje baterije kroz parazitsku diodu MOSFET-a za kontrolu pražnjenja.
4. Zaštita od prekomjernog punjenja - modul će napuniti bateriju do 4,2 V.
5. Zaštita od prekomjerne struje i kratkog spoja - modul će prekinuti izlaznu snagu baterije ako brzina pražnjenja prelazi 3 A ili ako dođe do kratkog spoja [10].



Slika 4.1. TP4056 modul za punjenje litijeve baterije

4.4. Indikator kapaciteta litijeve baterije

Kako bi se mogao prikazati aktivni kapaciteti litijeve baterije od 3,9 V na istu je spojen indikator statusa. LED zaslon (širine 20 mm i širine 31 mm) indikatora postavljen je s donje strane kocke odmah uz prekidač za njeno paljenje i gašenje, dok se preostali dio modula nalazi unutar kućišta kocke. Zaslon prikazuje status baterije tako da svaka plava crtica označava 25 % njenog kapacitet (npr. ako su prikazane 3 plave crtice, baterija se nalazi na 75 % kapaciteta). U slučaju da je kapacitet baterije kritično nizak, svijetlit će samo okvir baterije crvenom bojom [11].



Slika 4.2. Indikator kapaciteta litijeve baterije.

4.5. Pretvarač napona *step-up* (2,5 V – 5 V)

Pretvarač napona (*step-up* pretvarač) je istosmjerni u istosmjerni pretvarač koji pojačava napon (dok smanjuje struju) od svog ulaza (napajanja) do izlaza (opterećenja). To je klasa napajanja s komutiranim načinom rada (SMPS) koja sadrži najmanje dva poluvodiča (diodu i tranzistor) i najmanje jedan element za pohranu energije: kondenzator, induktor ili dva u kombinaciji. Kako bi se smanjila valovitost napona, filteri izrađeni od kondenzatora (ponekad u kombinaciji s prigušnicima) obično se dodaju na izlaz takvog pretvarača (filter na strani opterećenja) i ulaz (filter na strani napajanja) [8]. U sklopu ovog završnog rada *step-up* konverter služi za pretvaranje napona litijeve baterije 18650 od 3,7 V (odnosno od 3,5 V – 4,2 V) u stabilnih 5 V, a koji je potreban za napajanje Arduino Nano razvojnog sustava. Izlazi pretvarača spojeni su na VIN pin i GND pin Arduino Nano-a.

4.6. Žiroskop GY-521

Žiroskop je uređaj koji sadrži kotačić koji se brzo okreće ili cirkulirajući snop svjetlosti koji se koristi za otkrivanje odstupanja objekta od željene orijentacije. Modul GY-521 za Arduino Nano je probojna ploča za MPU-6050 MEMS (mikro elektromehanički sustavi) koja sadrži troosni žiroskop, troosni akcelerometar, digitalni procesor pokreta (DMP) i temperaturni senzor. Digitalni procesor pokreta može se koristiti za obradu složenih algoritama izravno na ploči. Obično DMP obrađuje algoritme koji sirove vrijednosti pretvaraju iz senzora u podatke o stabilnom položaju. Vrijednosti senzora dohvaćaju se pomoću serijske sabirnice podataka I2C, koja zahtijeva dvije žice (SCL i SDA). U primjeru vremenske kocke žiroskop je poslužio za određivanje orijentacije kocke u prostoru kako bi se mogao pratiti položaj gornje stranice kocke te prema prepoznatom položaju upaliti određene boje na LE diodama [12].

4.7. *Bluetooth* AT-09 BLE modul

Sljedeća komponenta bitna za razmjenu podataka između vremenske kocke i mobilnog uređaja je *bluetooth* modul za Arduino Nano. *Bluetooth* je radio-tehnologija koja koristi frekvenciju i prenosi pakete podataka unutar opsega od 2,4 GHz. Ovi se paketi razmjenjuju putem jednog od 79 određenih *Bluetooth* kanala (od kojih je svaki 1 MHz u propusnosti). *Bluetooth* je zapravo bežična veza između telefona i uređaja s kojim je telefon povezan. S ovom tehnologijom omogućena je razmjena podataka

na kratkom dometu bez potrebe za žicom. Na tržištu se 2011. godine pojavio *Bluetooth* 4.0 ili *Bluetooth Low Energy*. Kada je riječ o *Bluetooth Low Energy* u odnosu na *Bluetooth*, ključna razlika je u tome što *Bluetooth* LE-ima nisku potrošnju energije. Uz nisku potrošnju energije, aplikacije mogu raditi dulje vrijeme uz minimalnu potrošnju baterije. Slično *Bluetoothu*, *Bluetooth LE* radi u opsegu 2,4 GHz. Skrivena razlika je u tome što *Bluetooth* s niskom potrošnjom energije ostaje u stanju mirovanja osim ako se ne pokrene veza. Stvarno vrijeme povezivanja traje nekoliko milisekundi, za razliku od *Bluetootha*, koji se povezuje na nekoliko sekundi ili nekoliko sati u jednoj sesiji [13].

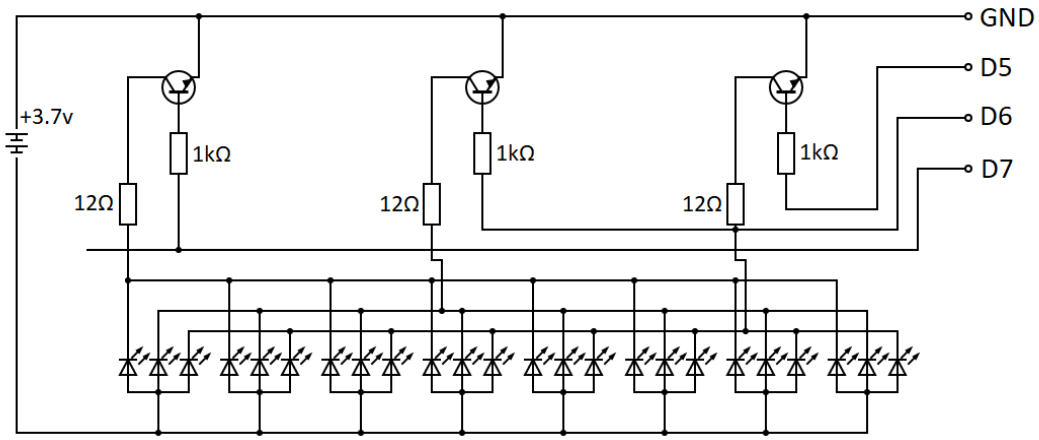
Zbog svoje niske energetske potrošnje i kompatibilnosti za projekt vremenske kocke izabran je AT-09 BLE modul za Arduino Nano koji radi na *Bluetooth LE* principu. Ovaj modul omogućuje obavljanje serijske komunikacije s *BLE* čipom zahvaljujući Rx i Tx pinu. Nadalje ovaj je modul također vrlo sličan modulu HM-10 i također je kompatibilan s njim [14]. Kako bi se Arduinu omogućila komunikaciju s AT-09 BLE modulom, koristi se serijsko sučelje (pinovi TXD i RXD). Arduino omogućuje pretvaranje bilo kojeg para digitalnih pinova u serijsko sučelje pomoću biblioteke *SoftwareSerial* uključene u Arduino IDE. Nakon uspostave serijske veze, može se početi s prijenosom podataka na modul AT-09 BLE.

4.8. RGB diode

Svjetlosna dioda je poluvodički izvor svjetlosti koji odašilje svjetlost prilikom protjecanja struje. Elektroni u poluvodiču se ponovno spajaju s elektronskim rupama, oslobađajući energiju u obliku fotona. Boja svjetlosti (koja odgovara energiji fotona) određena je energijom potrebnom da elektroni pređu pojasni pojas poluvodiča. Bijelo svjetlo dobiva se upotrebom više poluvodiča ili sloja svjetlosnog fosfora na poluvodičkom uređaju. Miješanjem intenziteta crvene, zelene i plave svjetlosti dobivaju se sve ostale kombinacije boja. Budući da LE diode imaju neznatno različite obrasce emisije, ravnoteža boja može se promijeniti ovisno o kutu gledanja, čak i ako su RGB izvori u jednom paketu, pa se RGB diode rijetko koriste za proizvodnju bijele svjetlosti Unatoč tome, ova metoda ima mnoge primjene zbog fleksibilnosti miješanja različitih boja, i u načelu, ovaj mehanizam također ima veću kvantnu učinkovitost u stvaranju bijele svjetlosti. Postoji nekoliko vrsta višebojnih bijelih LE dioda: dvo-, tro- i tetrakromatske bijele LE diode. Nekoliko ključnih čimbenika koji utječu na ove različite metode uključuju stabilnost boje, sposobnost iscrtavanja boje i svjetlosnu učinkovitost. Često veća učinkovitost znači nižu reprodukciju boja, predstavljajući kompromis između svjetlosne učinkovitosti

i prikazivanja boje. Na primjer, dikromatske bijele LE diode imaju najbolju svjetlosnu učinkovitost (120 lm/W), ali najmanju sposobnost prikazivanja boja [15].

Za izradu vremenske kocke iskorištene su 5mm RGB diode koje sadrže tri različita LE diode u jednom kućištu (zajednička anoda). Svaki odašiljač spojen je na zasebni vod tako da se njima može neovisno upravljati. U ovom slučaju upotrijebljeno je 8 RGB dioda tako da je svaka raspoređena u jedan kut kocke te povezana s Arduino Nano razvojnim sustavom. Svaka RGB dioda odašilje svjetlost koja joj je zadana i pali se s obzirom na orijentaciju kocke. Iz Slika 4.3. vidi se da su baze tranzistora spojene s digitalnim pinovima 5, 6 i 7 tako da oni preuzimaju funkciju prekidača (releja). Svaki tranzistor je odgovoran za jednu boju na svim LE diodama.



Slika 4.3. Shematski prikaz sklopke za upravljanje diodama.

5. PROGRAMSKA PODRŠKA UREĐAJA

U nastavku rada prikazan je programski dio projekta izrađen u Arduino i MIT programskom jeziku. Korištenjem Arduino programskog jezika izrađen je dio hardverskog koda koji upravlja mikroupravljačem i *bluetooth* dijelom sklopovlja, dok je u MIT programu napravljena Android aplikacija.

5.1. Arduino programska podrška

U početnom dijelu programa za mikroupravljač uključene su potrebne biblioteke „*Wire.h*“ i „*SoftwareSerial.h*“, gdje “*Wire.h*“ biblioteka omogućuje komunikaciju s I2C / TWI uređajima. „*SoftwareSerial.h*“ biblioteka razvijena je kako bi omogućila serijsku komunikaciju na drugim digitalnim pinovima Arduina, koristeći softver za repliciranje funkcionalnosti (otuda i naziv "*SoftwareSerial*"). Moguće je imati više softverskih serijskih portova s brzinama do 115200 bps.

Nadalje korištenjem biblioteke „*SoftwareSerial*“ napravljena je inicijalizacija naredbe „*SoftwareSerial BluetoothSerial(3, 4);*“ za pinove 3 (RX) i 4 (TX) koji služe za razmjenu podatkovnog prometa. Varijabla *orjentacija* tipa *int* definirana je kao dvodimenzionalna matrica s kojom je određeno 6 mogućih orijentacija kocke (u stupnjevima) kada se svaka od stranica kocke nađe u gornjem položaju, kako je prikazano na Slika 5.1.

Funkcija *setup ()*, (Slika 5.2) poziva se pri pokretanju programa. Koristi se za inicijalizaciju varijabli, te komunikacije korištenjem ranije navedenih biblioteka. Funkcija *setup ()* pokreće se samo jednom, nakon svakog uključivanja ili resetiranja Arduino ploče. Nadalje definirani su izlazi za pinove koji predstavljaju pojedinačne boje (*red, blue , green*), a komunikacija se započinje naredbom *Wire.begin*.

Slika 5.3. prikazuje dio koda koji je odgovoran za prijenos na I2C podređeni (eng. *slave*) uređaj s danom adresom. *Wire.requestFrom(MPU_addr,14,true)* koristi se za traženje bajtova s podređenog uređaja. Bajtovi se tada mogu dohvatiti s *read ()* funkcijom.

```
12                                  //rx1        rx2          ry1          ry2       stranica
13 int orjentacija [ 23 ][ 5 ] = {{  240,        300,           0,         120,         6},
14                                  {   60,        120,         270,         340,         5},
15                                  {    0,        360,         330,         360,         1},
16                                  {    0,        360,           0,          30,         1},
17                                  {    0,        360,         240,         300,         2},
18                                  {    0,        360,         150,         210,         3},
19                                  {    0,        360,          60,         120,         4}};
```

Slika 5.1. Prikaz inicijalizacijskog dijela koda.

```
20 void setup()
21 {
22   pinMode (red, OUTPUT);
23   pinMode (blue, OUTPUT);
24   pinMode (green, OUTPUT);
25   BluetoothSerial.begin(9600);
26   Wire.begin();
27   Wire.beginTransmission(MPU_addr);
28   Wire.write(0x6B);
29   Wire.write(0);
30   Wire.endTransmission(true);
31   Serial.begin(9600);
32 }
```

Slika 5.2. Definiranje pinova.

```
36   Wire.beginTransmission(MPU_addr);
37   Wire.write(0x3B);
38   Wire.endTransmission(false);
39   Wire.requestFrom(MPU_addr,14,true);
```

Slika 5.3. Dio komunikacije koji se nalazi unutar *loop ()* petlje.

5.1.1. Određivanje rotacije kocke

Kako bi mikroupravljač odredio orijentaciju kocke u prostoru potrebno je očitati podatke sa žiroskopa. Podaci koji se primaju od žiroskopa moraju biti pretvoreni u poznate jedinice radi lakšeg rukovanja. Pomoću funkcije *Wire.read()* (Slika 5.4.) omogućeno je očitavanje koordinata po svakoj osi (varijable *AcX*, *AcY*, *AcZ*). Očitane koordinate mapiraju u zadane intervale te se njihove nove

vrijednosti prebacuju iz radijana u stupnjeve u varijable *x*, *y* koje su ranije inicijalizirane. Navedenim postupkom se iz orijentacije kocke izostavlja z os koja u ovom slučaju nije potrebna.

```
41  AcX=Wire.read()<<8|Wire.read();
42  AcY=Wire.read()<<8|Wire.read();
43  AcZ=Wire.read()<<8|Wire.read();
44
45  int xAng = map(AcX,minVal,maxVal,-90,90);
46  int yAng = map(AcY,minVal,maxVal,-90,90);
47  int zAng = map(AcZ,minVal,maxVal,-90,90);
48
49  x= RAD_TO_DEG * (atan2(-yAng, -zAng)+PI);
50  y= RAD_TO_DEG * (atan2(-xAng, -zAng)+PI);
```

Slika 5.4. Dio koda za učitavanje i određivanja orijentacije kocke.

```
52  for(int i=0;i<10;i++){
53    if(x>orjentacija[i][0]&&x<orjentacija[i][1] && y>orjentacija[i][2]&&y<orjentacija[i][3]){
54      prethodna=stranica;
55      stranica=orjentacija[i][4];
56      break;
57    }
58  }
```

Slika 5.5. Petlja za određivanje stranice koja je trenutno okrenuta prema gore.

Kako bi program provjerio trenutnu orijentaciju kocke potreban je dio koda (Slika 5.5.) koji prolazi kroz matricu i provjerava jel trenutna orijentacija u rasponu zadanih koordinata u unaprijed definiranoj matrici *orjentacija*. U slučaju da je orijentacija u rasponu zadanih koordinata, varijabla „*stranica*“ poprima vrijednost koja je pohranjena u matrici i izlazi iz petlje. Također u varijabli „*prethodna*“ ostaje pohranjena stvarna prethodna stranica.

5.1.2. *Bluetooth* komunikacija

BluetoothSerial je varijabla tipa *SoftwareSerial* koja omogućuje pozivanje naredbi *.print()* (linija 70 na Slika 5.7.) i *.available()* (linija 88 na Slika 5.6.)

Nadalje Slika 5.6. prikazuje funkciju koja se poziva u *loop ()* petlji i izvodi čitanje ulaza na bluetooth modulu te provjeru o kakvom se podatku radi. U slučaju da se na ulazu pojavi broj „1“ postavlja se indikator „upaljen“ na „*true*“ te se omogućuje nastavak programa. Ako se na ulazu pojavi „E“ to znači

da je došlo do prekida veze između *bluetooth*-a i mobilnog uređaja te se RGB diode moraju ugasiti i varijable postaviti na početna stanja.

```
86 void provjeriPrimljenePodatke()
87 {
88     if (BluetoothSerial.available())
89     {
90         c = BluetoothSerial.read();
91         if ( c== '1') { upaljen=true; }
92         if ( c== 'E') { upaljen=false;rgbColor(0, 0, 0);brojac=0; posalji=true; }
93     }
94 }
```

Slika 5.6. Primanje podataka s *bluetootha*.

5.1.3. Upravljanje RGB diodama

U kodu se također nalazi funkcija *rgbColor* koja za argumente prima tri boje (crvena, zelena i plava) u rasponu od 0 do 255 svaka, pri čemu 0 znači da dioda ne svijetli a 255 da svijetli s najjačim intenzitetom. Kombinacijama različitih intenziteta između ove tri boje dobiju se svjetlosti dodatnih boja spektra (ljubičasta, žuta, bijela). Pozivanje ove funkcije može se vidjeti u 67. liniji koda na Slika 5.7.

Ako korisnik pokrene praćenje vremena s mobilnog uređaja (pritiskom na tipku start, Slika 5.16.), tada RGB diode unutar kocke počnu treptati u boji definiranoj za trenutno aktivnu gornju stranicu. Nakon treperenja koje traje 5 s, kocka odašilje kontinuiranu svjetlost iz RGB dioda u trajanju od 20 s, što označava da je započeto praćenje željene aktivnosti (Slika 5.7.).

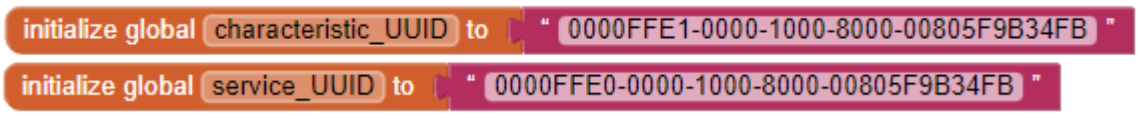
```
59  if(upaljen){
60  if(prethodna!=stranica){
61    posalji=true;
62    brojac=0;
63  }
64    if(brojac<=30*2){
65      brojac++;
66      if(brojac<5*2){
67        rgbColor(0, 0, 0);
68      }
69      else{
70        if(posalji){BluetoothSerial.print(stranica); posalji=false;}
71      }
72      if(brojac%2==1){
73        ukljuciStranicu(stranica);
74      }
75    }else{rgbColor(0, 0, 0);}
76  }
```

Slika 5.7. Dio koda koji upravlja paljenjem RGB dioda i slanjem broja stranice.

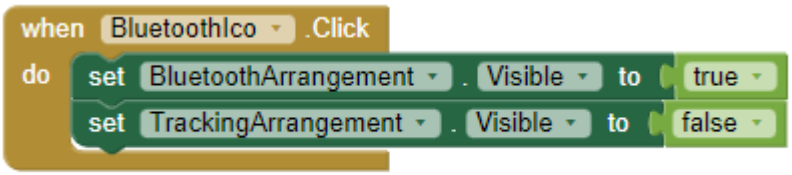
5.2. Opisivanje MIT app Android dijela koda

Android aplikacija izrađena je pomoću *MIT app inventor* alata u kojemu se kod generira korištenjem predefiniranih blokova čime se postiže željena funkcija. Pomoću navedenog programa moguće je u kratkom roku i vrlo jednostavno isprogramirati Android aplikaciju. U nastavku ovog odlomka su prikazani važniji dijelovi koda.

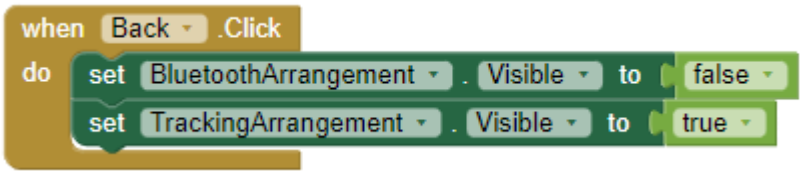
Slika 5.8. prikazuje inicijalizaciju varijabli koje predstavljaju *UUID* kratice. To je jedinstven broj koji se koristi za identifikaciju usluga, karakteristika i deskriptora. Navedene varijable kasnije se koriste u dijelovima programskog koda prikazanim na Slika 5.14., Slika 5.15. i Slika 5.16.



Slika 5.8. Postavljanje *UUID*-a.



Slika 5.9. Dio koda odgovoran za prebacivanje na *bluetooth* zaslon aplikacije.

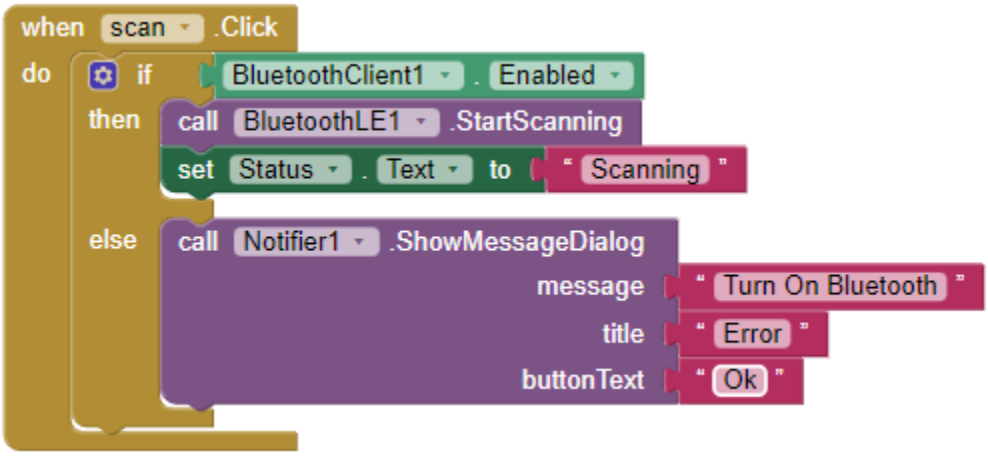


Slika 5.10. Dio koda odgovoran za vraćanje na početni zaslon aplikacije.

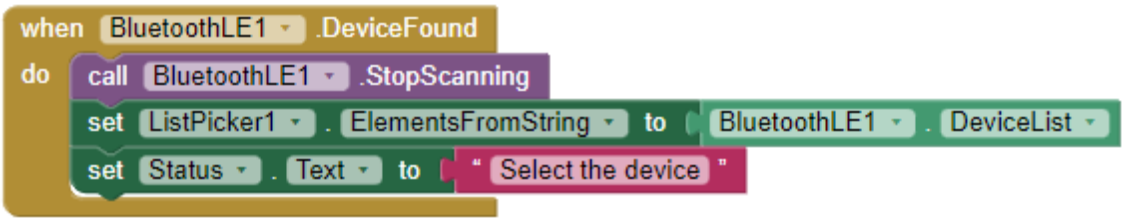
Na Slika 5.9. i Slika 5.10. prikazani su blokovi kojima se generira programski kod za prebacivanje između početnog zaslona i *bluetooth* zaslona unutar aplikacije tako da se pritiskom tipke *BluetoothIcon* ulazi u *bluetooth* dio aplikacije a pritiskom na *Back* vraća se na početni zaslon.

5.2.1. Upravljanje *bluetoothom*

Funkcija *scan* prikazana na Slika 5.11. omogućuje pretraživanje okolnih *bluetooth* uređaja pritiskom na istoimenu tipku unutar aplikacije. Svi pronađeni *bluetooth* uređaji popunjavaju listu uređaja kojoj se pristupa tipkom *Select*, a funkcija koja izvodi navedeno popunjavanje prikazana je na Slika 5.12.

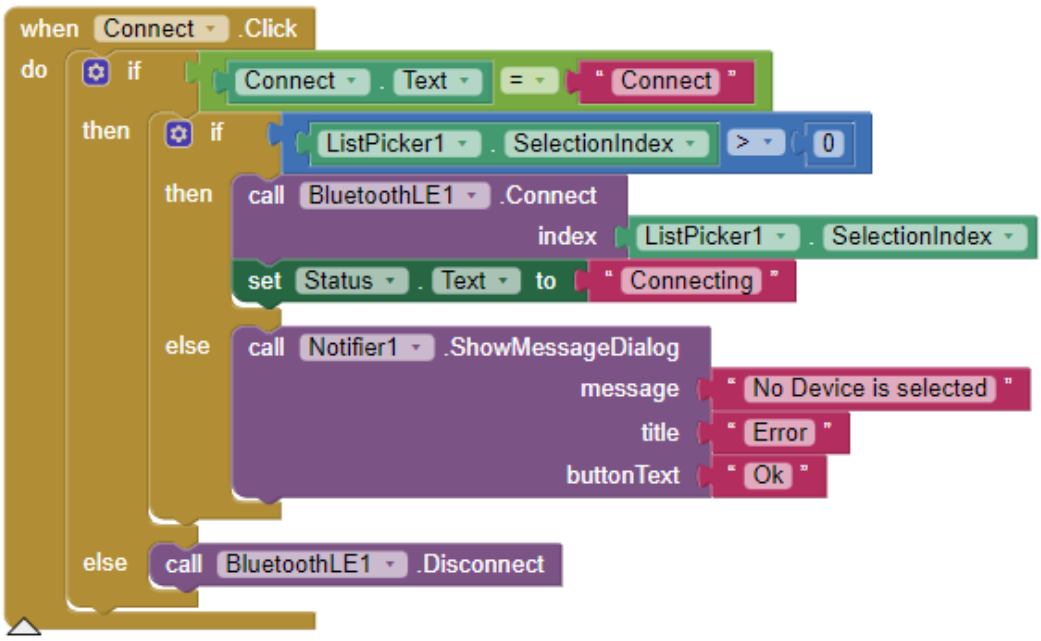


Slika 5.11. Pretraživanje *bluetooth* uređaja.



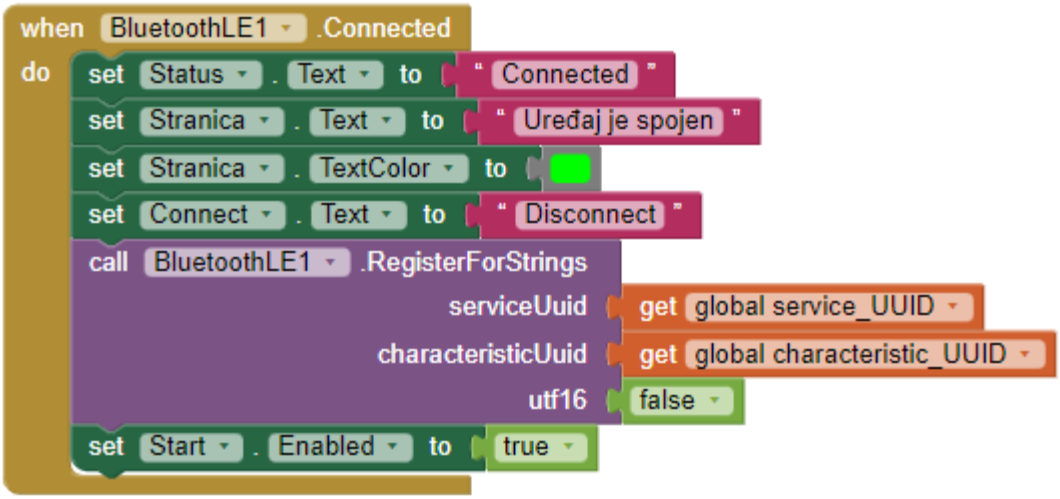
Slika 5.12. Popunjavanje liste *bluetooth* uređaja.

Funkcija *connect*, koja se poziva pritiskom istoimene tipke zadužene za spajanje Android aplikacije s ranije odabranim *bluetooth* uređajem, prikazana je na Slika 5.13.



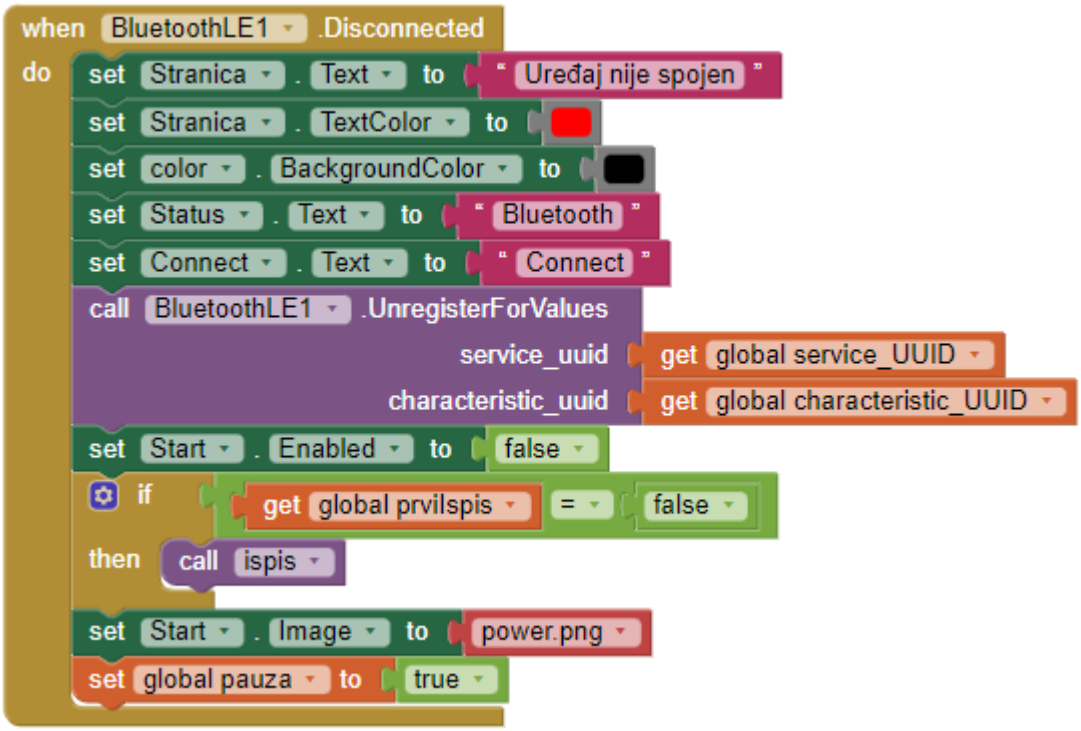
Slika 5.13. Spajanje na ranije odabran *bluetooth* uređaj.

Na Slika 5.14. prikazan je dio koda koji se izvodi nakon uspješnog povezivanja aplikacije s *bluetooth* uređajem pri čemu se funkcijom *Registerforstrings* uspostavlja tekstualni prijenos podataka između aplikacije i mikroupravljača unutar kocke.



Slika 5.14. Dio koda koji se pokreće prilikom uspješne konekcije s *bluetooth* uređajem.

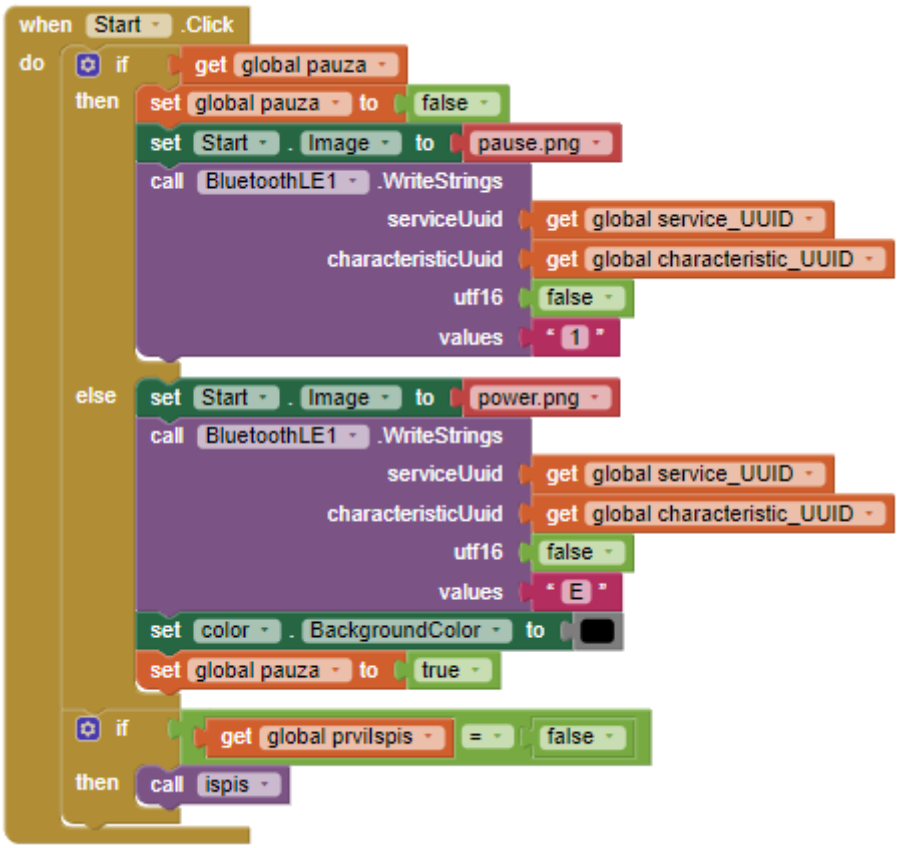
Dio koda koji se pokreće kada je *bluetooth* uređaj odspojen prikazan je na Slika 5.15. Iz prikazanog je vidljivo da se funkcijom *UnregisterForValues* aplikacija odspaja s *bluetooth* uređaja za ranije definirane *UUID* varijable. Nakon odspajanja aktivira se funkcija *ispis* objašnjena u nastavku teksta.



Slika 5.15. Dio koda koji se pokreće kada je *bluetooth* uređaj odspojen.

5.2.2. Upravljanje radom kocke

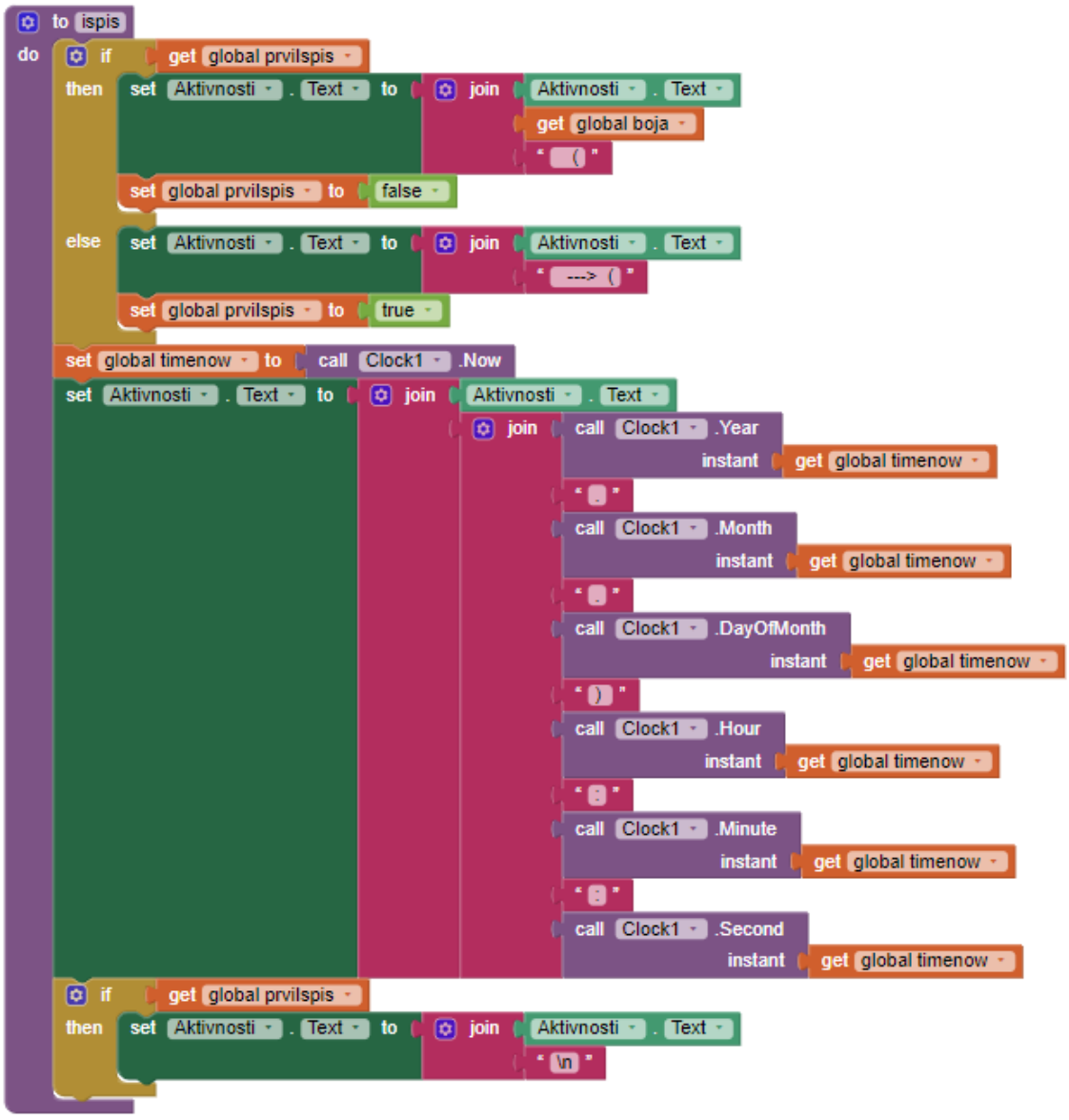
Slika 5.16. prikazuje funkciju *Start* koja upravlja paljenjem i gašenjem praćenja aktivnosti, a što se postiže pritiskom tipke „*on/off*“ unutar aplikacije. Ako korisnik odluči započeti praćenje aktivnosti pritiskom na tipku aktivira navedenu funkciju za koju se preko *bluetooth* modula šalje vrijednost „1“. Ukoliko dođe do gašenja praćenja aktivnosti istoj se pridružuje vrijednost „E“. Dio Arduino koda koji služi za prepoznavanje vrijednosti („1“ i „E“) prikazan je ranije na Slika 5.6.



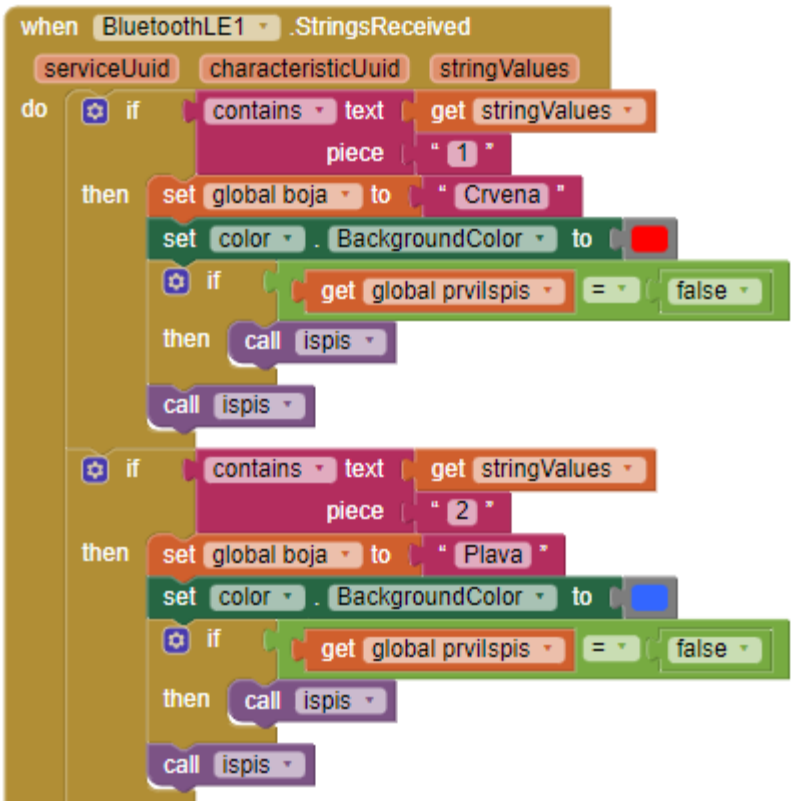
Slika 5.16. Dio koda koji upravlja početkom i krajem praćenja aktivnosti.

5.2.3. Upravljanje podacima i pohrana

Funkcija *ispis* dohvaća trenutno vrijeme s mobilnog uređaja (godina, mjesec, dan, sat, minuta, sekunda) te ga zapisuje na zaslon aplikacije tako da korisnik ima uvid u vrijeme izvršenih aktivnosti. Nadalje ova funkcija omogućuje povezivanje aktivnosti s vremenom kada je ista započeta i završena (Slika 5.17.).

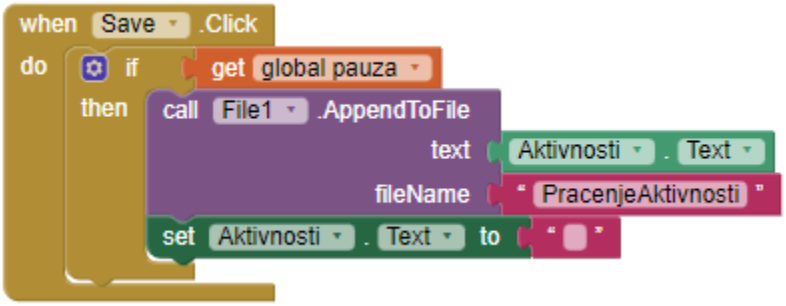


Slika 5.17. Funkcija *ispis* koja organizira podatke o vremenu promjene aktivnosti.



Slika 5.18. Dio koda koji određuje boju gornje stranice kocke ovisno o ulaznom podatku.

Funkcija „*StringsRecieved* (Slika 5.18.) pokreće se prilikom zaprimanja podatka s *bluetooth* uređaja tj. vremenske kocke. Varijabla *boja* poprima odgovarajuću boju u ovisnosti o kojoj se vrijednosti radi (npr. 1- crvena, 2- plava itd.). Ova funkcija također poziva ranije opisanu funkciju *ispis* (Slika 5.17.).



Slika 5.19. Dio koda odgovoran za pohranu prikupljenih podataka.

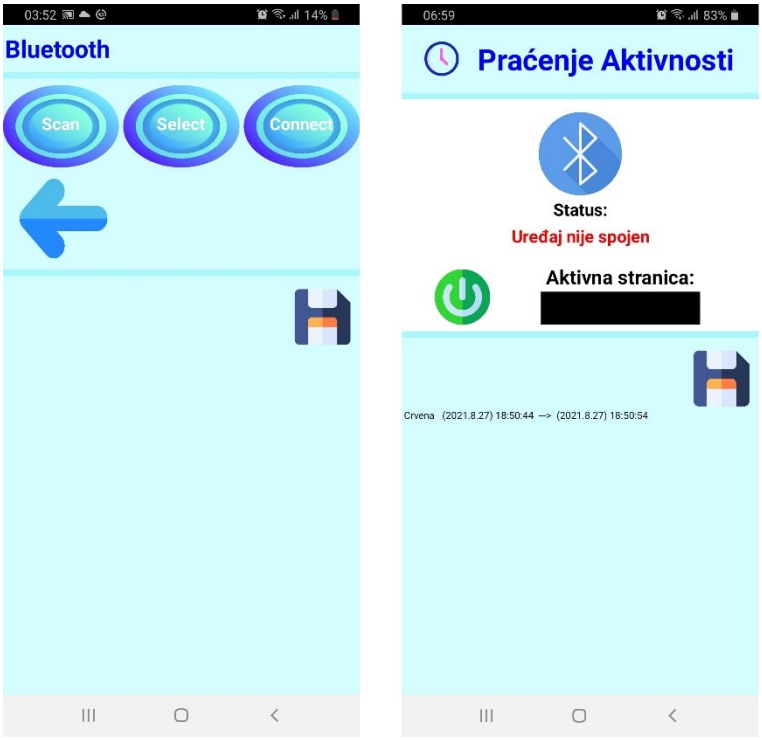
Slika 5.19. prikazuje dio koda koji se izvodi pritiskom tipke *Save* namijenjenom za pohranu podataka o ranije praćenim aktivnostima u koje korisnik ima uvid na zaslonu aplikacije.

5.2.4. Izgled aplikacije

Na Slika 5.20. prikazan je izgled zaslona mobilne aplikacije koji je izrađen korištenjem programa *MIT app Inventor* Program omogućuje lagano oblikovanje zaslona, uređivanje veličine pojedinih slika, tipki i tekstualnih dijelova koristeći *horizontal* i *vertical view*. Također ovakvo uređivanje omogućuje da je aplikacija prilagođena svim dimenzijama ekrana na kojima se može koristiti.

Slika 5.20. Oblikovanje sučelja android aplikacije.

Rezultat navedenog uređivanja zaslona prikazan je na Slika 5.21., gdje je a) dio sučelja odgovoran za spajanje na *bluetooth* uređaj, a b) početni zaslon Android aplikacije na kojem se prikazuju informacije o praćenju aktivnosti te se omogućuju razne opcije opisane ranije u ovom poglavlju.



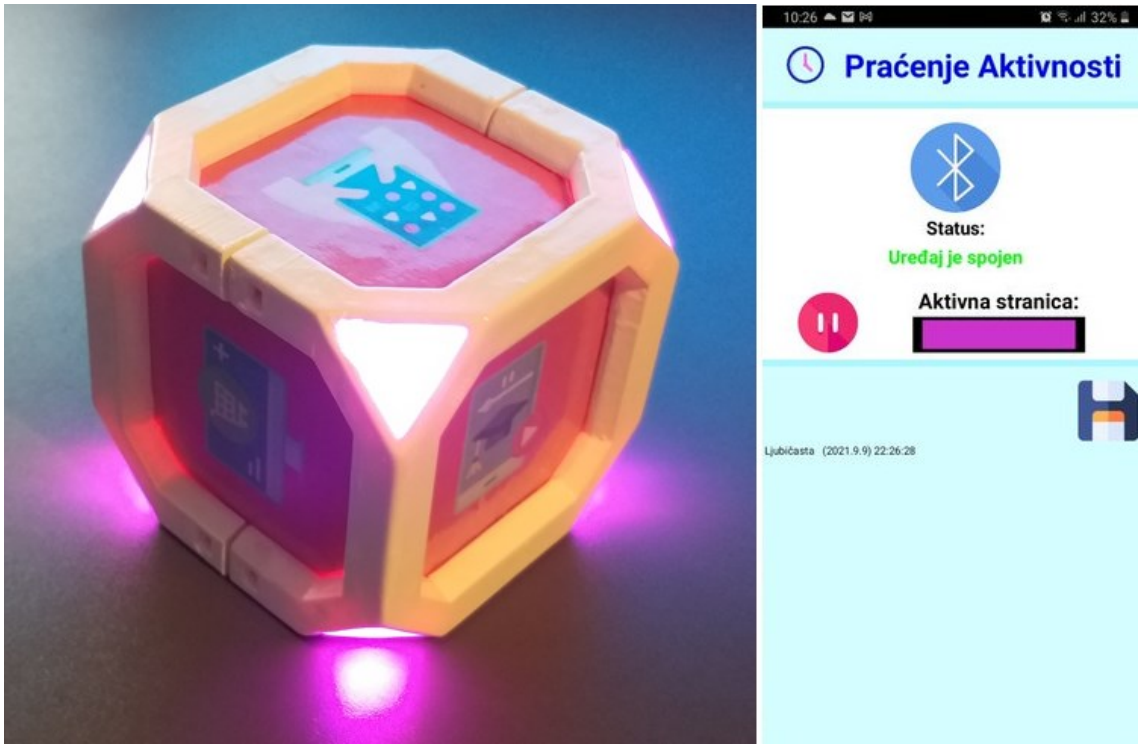
a) b)

Slika 5.21. Zaslon mobilne aplikacije: a) zaslon za odabir *bluetooth* uređaja, b) početni zaslon.

Na Slika 5.21. a) vidljive su tipke: *Scan*, *Select*, *Connect* i tipka za vraćanje nazad, kojima je funkcija opisana ranije u ovom poglavlju (Slika 5.10., Slika 5.11., Slika 5.12. i Slika 5.13.). Na Slika 5.20. b) nalaze se tipke u obliku sličica koje predstavljaju funkcije „*on/off*“, „*bluetooth*“ i „*save*“ čije funkcije su opisana ranije (Slika 5.9., Slika 5.16., Slika 5.19.).

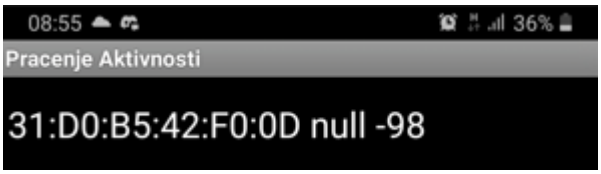
6. TESTIRANJE RADA UREĐAJA I MOBILNE APLIKACIJE

Konačan izgled vremenske kocke prikazan je na Slika 3.5. i Slika 6.1. Početak praćenja aktivnosti ostvaruje se paljenjem kocke na vanjski prekidač. Nakon uključivanja potrebno je spojiti aplikaciju s *bluetooth* modulom kocke što se postiže pritiskom tipki *scan – select – connect* . Tek nakon pritiska tipke „*on/off*“ u aplikaciji, RGB diode uređaja započinju treperiti određenom bojom u trajanju od 5 sekundi s obzirom na trenutnu orijentaciju kocke. Prestanak treperenja i paljenje kontinuirane svjetlosti signalizira početak praćenja aktivnosti, a nakon 20 sekundi trajanja kontinuiranog svjetla isto se gasi čime se postiže smanjenje potrošnje baterije. Nakon prekida praćenja aktivnosti ili njene promjene, prethodna ostaje pohranjena u bazi podataka mobilne aplikacije kako bi se postigla preglednost i kasnija analiza utrošenog vremena. Za normalan rad aplikacije, na mobilnom uređaju potrebna je uspostavljena *bluetooth* veza i lokacija mobitela. Kada je kocka kontinuirano spojena s mobilnim uređajem, trajanje baterije je prosječno 2 dana, što je utvrđeno eksperimentalnom analizom. Punjenje kocke u prosjeku traje 2 h i ne sprečava praćenje aktivnosti.



Slika 6.1. Testiranje točnosti kocke.

Na Slika 6.1. prikazano je testiranje uređaja iz kojeg je vidljivo da se kocka nalazi u položaju koji predstavlja praćenje vremena utrošenog na korištenje mobitela (gornja stranica kocke). Pri tome RGB diode odašilju svjetlost ljubičaste boje što se poklapa s bojom koja je prikazana na mobilnoj aplikaciji u za to namijenjenom okviru. Zbog manjkavosti modula žiroskopa, mali nagibi u jednu stranu mogu dovesti do promjene stranice, ali sve dok je kocka na ravnoj površini taj problem nije izražen. Testiranje točnosti detekcije gornje stranice kocke napravljeno je za 15 nasumično odabranih položaja pri čemu je svih 15 položaja ispravno detektirano.



Slika 6.2. *Bluetooth* greška.

Prilikom paljenja uređaja na vanjski prekidač kocke, na mobilnoj aplikaciji može se pojaviti greška „null“ (Slika 6.2.) gdje *bluetooth* modul nije dobro prepoznat. Pri tome je potrebno određeno vrijeme sačekati na ponovnu uspostavu veze ili ponovno pokrenuti uređaj. Navedeno je posljedica lošije kvalitete *bluetooth* modula.

7. ZAKLJUČAK

Ideja rada bila je osmisliti i izraditi model vremenske kocke po uzoru na postojeće projekte dostupne na tržištu. Cilj je bio da konačni izgled i funkcionalnost kocke budu u potpunosti prilagođeni krajnjem korisniku, a namijenjeni praćenju njegovih željenih aktivnosti

Vanjski (vidljivi) dio kocke čine 3D isprintani modeli donjeg i gornjeg skelet kocke između kojih se uklapaju stranice i trokuti kocke, prethodno dizajnirani u programu „*Blender*“. Pri tome stranice predstavljaju zamjenjivi dio kocke tako da se u određenom trenutku njihov raspored može mijenjati prema potrebama korisnika, u svrhu praćenja različitih aktivnosti koje su prikazane na slikama.

Hardverski dio kocke smješten je unutar njenog kućišta pri čemu njegovu osnovu čini razvojni sustav Arduino Nano koji se napaja preko punjive litijeve 18650 baterije na koju je spojen indikator kapaciteta vidljiv na vanjskoj stranici kocke. Litijeva baterija puni se pomoću TP4056 modula (USB c ulaz), a *step-up* pretvarač napona omogućuje pretvorbu od 3,7 V do stabilnih 5 V potrebnih za rad Arduino Nano-a. Na razvojnu su pločicu spojeni moduli koji omogućavaju praćenje položaja (žiroskop) i *bluetooth* povezanost (BLE modul) kocke, a 8 RGB dioda smještenih u uglove kocke omogućuju njenu osvijetljenost.

Za uspostavljanje softverske kontrole korišten je Arduino programski jezik kojim je programiran Arduino Nano razvojni sustav, *bluetooth* povezanost s modulom, određivanje orijentacije kocke pomoću žiroskopa, upravljanje paljenjem RGB dioda te dodjeljivanje boje određenom položaju kocke. *MIT App Inventor* program poslužio je za programiranje i oblikovanje sučelja mobilne aplikacije. Navedenim je isprogramirano traženje i povezivanje *bluetooth* uređaja s aplikacijom, paljenje RGB dioda u različitim bojama te mjerenje vremena izvođenja aktivnosti. Izgled zaslona aplikacije izrađen je na jednostavan način sa svega nekoliko tipki potrebnih za izvođenje procesa.

Gore navedenim postupcima u potpunosti je omogućen rad vremenske kocke. Prva potrebna radnja za njeno pokretanje je traženje *bluetooth* modula te povezivanje s istim. Nakon toga korisnik stavlja kocku u željeni položaj s obzirom na aktivnost koju izvodi te se nakon 5 s treptanja kocke pali kontinuirano svjetlo u boji povezanoj s trenutnom orijentacijom. Nakon promjene aktivnosti, prethodna se pamti i tako prati vrijeme utrošeno na izvođenje iste.

U ovom radu prikazan je potencijalni način izrade uređaja koji bi pratio aktivnosti (dnevne i poslovne) korisnika na jednostavan i vizualno privlačan način, a pri tome uz malu potrošnju energije. Budući projekti koji bi proizašli iz modela vremenske kocke predstavljali bi inovaciju u području praćenja aktivnosti, uz mogućnost daljnje nadogradnje i unaprjeđenja postojeće osnove.

LITERATURA

[1] R. B. , Brent Rubell, Adafruit IO Time Tracking Cube [online], Adafruit, New York, 2019., dostupno na: https://learn.adafruit.com/time-tracking-cube [26.08.2021.]

[2] Timeflip, Timeflip 2: Human friendly time tracker [online], Timeflip, EU, 2019., dostupno na: https://timeflip.io/ [26.08.2021.]

[3] Microsoft, MS Office: Activity tracker [online], Microsoft, SAD, 2021., dostupno na: https://templates.office.com/en-us/activity-tracker-tm00000027 [25.08.2021.]

[4] O. S. , Olamid, Smartwatch vs smartband: which should you buy? [online], Ignited, SAD, 2020., dostupno na: https://www.dignited.com/49707/smartwatch-vs-smartband-which-should-you-buy/. [25.08.2021.]

[5] Blender, Blender 2.93 Manual: 3D Blender Toolbox [online], Blender inc, Nizozemska, 2021., dostupno na: https://docs.blender.org/manual/en/latest/addons/mesh/3d_print_toolbox.html [26.08.2021.]

[6] D. , Robinson, What software to use for 3D slicing? [online], Ultimaker, Nizozemska, 2021, dostupno na: https://ultimaker.com/learn/what-software-to-use-for-3d-printing. [26.08.2021.]

[7] Wikipedia, Arduino nano wikipedia [online], Wikipedia, SAD, 2021., dostupno na: https://en.wikipedia.org/wiki/Arduino_Nano [26.08.2021.]

[8] Wikipedia, Step-up converter [online], Wikipedia, SAD, 2021., dostupno na: https://en.wikipedia.org/wiki/Boost_converter [27.08.2021.]

[9] Wikipedia, Lithium-ion battery [online], Wikipedia, SAD, 2021. , dostupno na: https://en.wikipedia.org/wiki/Lithium-ion_battery [26.08.2021.]

[10] Roboelements, TP4056 modul for lithium battery charging, [online], Roboelements, India, 2017., dostupno na: https://www.roboelements.com/product/tp4056-1a-5v-li-ion-battery-charging-module-with-current-protection/ [26.08.2021.]

[11] Aliexpress, Indicator for lithium battery [online], Aliexpress, Kina, 2021, dostupno na: https://www.aliexpress.com/item/1005001524966003.html?spm=a2g0s.9042311.0.0.429a4c4dx0lt5L&fbclid=IwAR3_Z7eMXJfGqYV40Kd_YoM9tDdFTpPNDj5uLOz6PNr7Kn6388Z6zOEFY [26.08.2021.]

[12] M. , Scoeffler, GY - 521 gyroscope modul, [online], Michael Scoeffler, SAD, 2017., dostupno na: https://mschoeffler.com/2017/10/05/tutorial-how-to-use-the-gy-521-module-mpu-6050-breakout-board-with-the-arduino-uno/ [26.08.2021.]

[13] Y. , Benabbas, Bluetooth At-09 BLE modul [online], Yassine Benabbas, SAD, 2017. , dostupno na: https://medium.com/@yostane/using-the-at-09-ble-module-with-the-arduino-3bc7d5cb0ac2 [26.08.2021.]

[14] B. , Ray, Bluetooth v bluetooth LE: Whats the difference? [online], Link-labs, SAD, 2021. , dostupno na: https://www.link-labs.com/blog/bluetooth-vs-bluetooth-low-energy [26.08.2021.]

[15] Wikipedia, Light emitting diod [online], Wikipedia, SAD, 2021. , dostupno na: https://en.wikipedia.org/wiki/Light-emitting_diode [25.08.2021.]

SAŽETAK

Cilj rada je dizajnirati i izraditi vremensku kocku za praćenje aktivnosti korisnika. Model vanjskog dijela kocke dizajniran je programom *Blender*, a potom isprintan korištenjem 3D printera. Hardversku osnovu čini Arduino Nano razvojni sustav uz pripadajuće module. Mobilna aplikacija razvijena je putem *MIT App Inventor* programa te pruža povezivanje i upravljanje radom kocke. Konačan izgled kocke omogućuje praćenje aktivnosti pomoću različitih boja koje se pale kod određene orijentacije kocke dok aplikacija za to vrijeme prati trajanje njenog izvođenja uz mogućnost pohranjivanje i kasnije analize.

Ključne riječi: Arduino, *bluetooth* komunikacija, praćenje aktivnosti, vremenska kocka, 3D ispis

ABSTRACT

Title: Activity tracking

The aim of the work is to design and build a time cube to monitor user activities. The model of the outer part of the cube was designed with the *Blender* program, and then printed using a 3D printer. The hardware is based on the Arduino Nano development system with the corresponding modules. The mobile application was developed through the MIT App Inventor program, and enabled the connection and management of the cube. The final appearance of the cube allows tracking activity using different colors that light up at a particular orientation of the cube, while the application during this time monitors the duration of its execution with the possibility of storage and later analysis.

Keywords: Arduino, bluetooth communication, activity tracking, time cube, 3D printing