

# Pametna kasica - sustav za pohranu i sortiranje kovanica

---

Laco, Renato

Undergraduate thesis / Završni rad

2023

*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://um.nsk.hr/um:nbn:hr:200:616238>

*Rights / Prava:* [In copyright](#)/[Zaštićeno autorskim pravom.](#)

*Download date / Datum preuzimanja:* **2025-03-17**

*Repository / Repozitorij:*

[Faculty of Electrical Engineering, Computer Science and Information Technology Osijek](#)



**SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU**

**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I**

**INFORMACIJSKIH TEHNOLOGIJA OSIJEK**

**Stručni studij**

**PAMETNA KASICA – SUSTAV ZA POHRANU I**

**SORTIRANJE KOVANICA**

**Završni rad**

**Renato Laco**

**Osijek, 2023.**

## Sadržaj

<b>1. UVOD</b> .....	1
<b>1.1. Zadatak i struktura rada</b> .....	2
<b>2. TEORIJSKI OSVRT NA SUSTAV PAMETNE KASICE PRASICE</b> .....	3
<b>2.1. Pregled postojećeg stanja i teorijski osvrt na problem i rješenje zadatka</b> .....	3
<b>2.2. Prijedlog sklopovskog rješenja</b> .....	5
<b>2.3. Prijedlog programskog rješenja</b> .....	7
<b>3. REALIZACIJA "Pametne kasice"</b> .....	10
<b>3.1. Korišteni alati i programska okruženja</b> .....	10
<b>3.2. Realizacija sklopovskog rješenja</b> .....	12
<b>3.3. Realizacija programskog rješenja</b> .....	18
<b>4. TESTIRANJE I REZULTAT</b> .....	22
<b>4.1. Metodologija testiranja</b> .....	22
<b>4.2. Rezultati testiranja</b> .....	22
<b>5. ZAKLJUČAK</b> .....	29
<b>LITERATURA</b> .....	30
<b>SAŽETAK</b> .....	32
<b>ABSTRACT</b> .....	33
<b>PRILOZI I DODACI</b> .....	34

## 1. UVOD

U današnjem svijetu, potreba za automatizacijom je sveprisutna i postaje ključni dio našeg svakodnevnog života. U mnogim područjima, uključujući financije i štednju novca, automatizacija donosi brojne prednosti koje čine upravljanje novcem jednostavnijim i učinkovitijim te nam tako olakšava život.

U radu je opisana izrada i testiranja pametne kasice prasice, s fokusom na automatizaciju sortiranja novčića, pohranjivanjem trenutne vrijednosti i prikaz na izrađenoj mobilnoj aplikaciji kao i na pokazivaču na samoj kasici. Danas u svijetu postoji raznolikost kasica prasica, od mehaničkih do potpuno automatiziranih. Jedan od problema klasičnih, također i mehaničkih kasica prasica je osjetljivost na zaglavljivanje ili pogrešno prepoznavanje novčića, što smanjuje preciznost brojanja i otežava točno praćenje pohranjenog iznosa. Nepraktičan pristup novcu također stvara velik problem, ovisno o dizajnu kasice može biti potrebno otvaranje ili rastavljanje dijelova kako bi se pristupilo pohranjenom novcu. Glavni nedostatak je funkcionalnost praćenja stanja kasice pa korisnici ne mogu pratiti svoj napredak. Automatizirane kasice prasice rješavaju ove probleme uvodeći elektroničke komponente koje poboljšavaju funkcionalnost i preciznost upravljanja pohranjenim novcem. Kada je riječ o realizaciji pametne kasice, postoji širok spektar mogućnosti kao što su dizajn kasice, način sortiranja novčića, način pohrane i praćenja pohranjenih iznosa, kao i interakcija s mobilnom aplikacijom, a sve to može se prilagoditi različitim potrebama i zahtjevima. Odluka o odabiru određene metode izrade ovisi o specifičnim uvjetima u kojima će se kasica koristiti, s posebnim naglaskom na preciznost, sigurnost, ali i brzinu sortiranja.

Zbog svoje jednostavne upotrebe u kućnom okruženju, odlučeno je za izradu kasice koja kombinira mehaničke i elektroničke komponente. Odnosno spoj dijela kasice koji automatski sortira kovanice po veličini, temeljen na utorima koji odgovaraju promjeru različitih kovanica, uz dodatak elektroničkih komponenti poput senzora blizine, čini sklop koji poboljšava funkcionalnost kasice. U ovom slučaju, preciznost brojanja novca nije presudna kao što je u složenijim sustavima, ali je i dalje na fokusu. Kombiniranjem mehaničke kasice s elektroničkim komponentama postiže se praktičnost i lakša upotreba bez potrebe za složenijim i skupljim sustavima. S obzirom da se time smanjuje broj elektroničkih komponentata, imamo i manje potencijalnih točaka kvara i pojednostavljenje prilikom održavanja pametne kasice, što je poželjno u kućnom okruženju gdje se traži jednostavnost i dugoročna upotreba.

U daljnjem radu, po poglavljima će biti opisan tijek izrade pametne kasice na sljedeći način. U drugom poglavlju proći će se kroz teorijski osvrt na problem realizacije pametne kasice, to jest biti

će objašnjen pristup rješavanja tih problema sa rezultatom davanja prijedloga realizacije programskog kao i sklopovskog rješenja. Bit će predložen dijagram toka, za programski i za sklopovski sustava koji će kasnije služiti kao podloga pri cjelokupnoj izradi. U trećem poglavlju kratko će se opisati korištena programska okruženja. Bit će i objašnjeno kako je sve realizirano temeljem danih prijedloga iz drugog poglavlja. U četvrtom poglavlju proved će se i prikazati rezultati testiranja, kao što su na primjer preciznost sortiranja umetnutih kovanica i preciznost detektiranja točnih vrijednosti. Te će se u zaključku kratko opisati sva problematika i rješenja ovog rada i osvrnuti na prednosti, ali i mane koje bi se trebale doraditi kako bi sustav bio još funkcionalniji.

### **1.1. Zadatak i struktura rada**

Zadatak završnog rada je izraditi, projektirati i testirati sustav za automatizirano prebrojavanje i sortiranje kovane monete, odnosno kovanica. Sustav pametne kasice treba imat mogućnost detekcije vrijednosti različitih kovanica i sortiranja na mjesto kovanica koje je priloženo. Trenutno stanje kasice treba biti prikazano na pokazivaču, pružajući trenutni uvid u ukupnu količinu pohranjenih kovanica. Osim toga kasica treba biti povezana s mobilnim uređajem koji će putem aplikacije omogućiti pregled trenutnog stanja. Cilj rada je na što jednostavniji način postići gore navedeno uz što veću efikasnost i kvalitetu same kasice.

## **2. TEORIJSKI OSVRT NA SUSTAV PAMETNE KASICE PRASICE**

### **2.1. Pregled postojećeg stanja i teorijski osvrt na problem i rješenje zadatka**

Danas postoji širok spektar metoda razvrstavanja kovanica, od jednostavnijih poput mehaničkog sortiranja sa što manje elektronike, [1], pa do kompliciranijih poput optičkog sortiranja gdje se koriste kamere ili senzori za prepoznavanje vizualnih karakteristika kovanica [2]. Prema [1] možemo vidjeti pristup mehaničkog sortiranja kovanica bez elektroničkih komponenti, dok prema [2] vidimo napredniji način detekcije kovanica koji to izvršava temeljem kamere. Odvojenost problema detekcije kovanica i sortiranja kovanica još dodaje u prilog da postoji mnogo raznih postojećih sustava koji mogu činiti različite kombinacije kako bi cijeli sustav funkcionirao. U daljnjem tekstu nabrojat ćemo neka postojeća rješenja koji se već negdje koriste. Jedan od najstarijih metoda temelji se na težinskom sortiranju. Prema [3] možemo vidjeti primjer gdje kovanica pada prema dolje pod blagim nagibom i prolazi kroz razne prepreke. Zbog razlike u veličini i težini kovanica, svaka će prilikom nailaska na prepreku drukčije reagirati i promijeniti putanju. Ako postoji više kovanica koje idu istom putanjom, dodaje se još prepreka dok se kovanice potpuno ne podijele. Jedna od najbržih metoda sortiranja kovanica je rotaciono sortiranje. Ova metoda koristi rotirajući disk s različitim otvorima, koji prilikom svog rotiranja uzrokuju da kovanice padaju kroz otvore njima namijenjenima što je opisano prema [4]. Prema [5] možemo vidjeti pristup ovakvog sortiranja gdje se disk okreće uz pomoć ljudske snage, što je kasnije nadograđeno dodavanjem motora. Ovom tehnikom moguće je ubaciti velik broj kovanica u bubanj u isto vrijeme, pa ga čini jednim od najbržih. Još jedna od čestih metoda je klizanje kovanice niz površinu koja je položena pod blagim nagibom prikazano prema [6]. Ovom metodom se na površinu postavljaju otvori različitih dimenzija koje odgovaraju promjerima kovanica, pa će tako kovanica ako odgovara promjeru upasti u otvor, a ako ne odgovara, proći preko njega. Na sve od ovih načina sortiranja, moguće je na mjesta gdje su kovanice već sortirane, integrirati senzore blizine koji bi zabilježili detekciju kovanica kako bi imali pregled stanja. Detekcija kovanica negdje je izvedena pomoću ultrazvučnog senzora što je prikazano prema [7]. Kada kovanica dođe na određenu poziciju, pomoću ultrazvučnog senzora prima se vrijednost udaljenosti do te kovanice. S obzirom na to da se zna vrijednost udaljenosti od ultrazvučnog senzora do mjesta mjerenja u trenutku bez kovanice, lako se izračuna visina i tako sortiraju sve kovanice. Još jedna od metoda je i težinsko sortiranje gdje se kovanica postavlja na vagu ili senzor za težinu kao prema [8]. Svaka kovanica ima svoju specifičnu vrijednost pa jednostavnom usporedbom vrijednosti lako možemo zaključiti o kojoj se kovanici radi. Ovo su samo neke od poznatih metoda koje se

uglavnom koriste u manje složenijim sustavima. Slika 2.1 prikazuje moderni uređaj za sortiranje i prebrojavanje kovanica koji koristi rotirajući disk, [4].



Slika 2.1 Uređaj za sortiranje kovanica s rotirajućim diskom, [4].

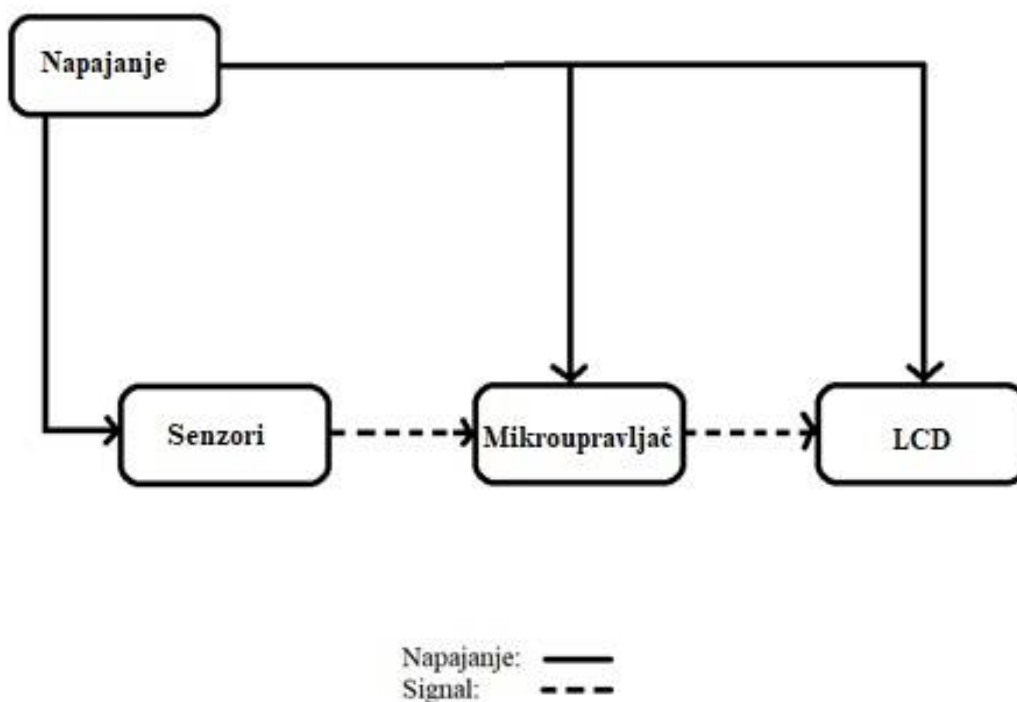
Promatrano s teorijske strane, funkcioniranje ovog sustava je vrlo jednostavno. Problem zadatka je podijeljen na četiri temeljna problema. To su razvrstavanje 8 različitih kovanica eura, njihova detekcija, zbrajanje vrijednosti, prikaz na pokazivaču i u mobilnoj aplikaciji. Za razvrstavanje kovanica od 1,2,4,10,20,50 centi te 1 i 2 eura bit će korištena ukošena pločica sa 8 otvora različitih dimenzija. Otvori će biti postavljeni od najmanjeg do najvećeg, tako da će svaka od kovanica moći upasti samo u svoj odgovarajući otvor. Kako bi se zabilježio ulazak kovanice u kasicu, ispod svakog otvora bit će postavljen senzor blizine koji će slanjem signala obavijestiti mikroupavljač o nastaloj promjeni, s obzirom da će se slati samo digitalni signal bit će potrebno definirati koji se točno senzor nalazi ispod kojeg otvora da se zna o kojoj dodanoj vrijednosti je riječ, što će kasnije biti riješeno u programu mikroupavljača. Još jedan od problema je taj gdje će se komponente poput pokazivača, mikroupavljača i dodatnih komponenti nalaziti unutar kasice, a da ne zauzimaju previše dodatnog prostora. Rješenje toga je postavljanje tih komponenti unutar kasice ispod dijela samog početka podloge zadužene za sortiranje kovanica. Taj početni dio podloge služiti će kako bi kovanice uz pomoću gravitacije postigle dovoljnu brzinu za klizanje prema dolje i postigle stabilnost, tako da u tom dijelu još uvijek nema otvora pa samim time ispod toga ostaje slobodan prostor u kojemu će te komponente biti postavljene. Problem izrade mobilne aplikacije zbog naizgled same kompliciranosti izrade bit će riješen uz pomoć platforme MIT App Inventor, [9]. Ova platforma omogućuje jednostavno blokovsko programiranje i integraciju već gotovih rješenja.

S obzirom na potrebu za bluetooth komunikacijom radi slanja trenutnog stanja kasice na mobilni uređaj, odabir ove platforme je izuzetno koristan.

## 2.2. Prijedlog sklopovskog rješenja

U ovom poglavlju blokovski je prikazan prijedlog funkcionalnog blok dijagrama sklopova planiranih za korištenje, odnosno sustav koji će se koristiti, te njihovo međusobno djelovanje. Napojni kabel će od jednog izvora napajanja biti provučen prema svim komponentama. Dok će signalni kabel povezivati senzore sa mikroupravljačem, isto tako mikroupravljač sa pokazivačem.

Prema slici 2.1 puna crta označava tok napajanja, dok isprekidana crta označava tok signala. Ti signali su prvenstveno digitalni signali koje senzori šalju mikroupravljaču o detekciji kovanice, te analogni signal koji mikroupravljač šalje pokazivaču, kako bi se prikazalo trenutno stanje vrijednosti novaca u kasici.

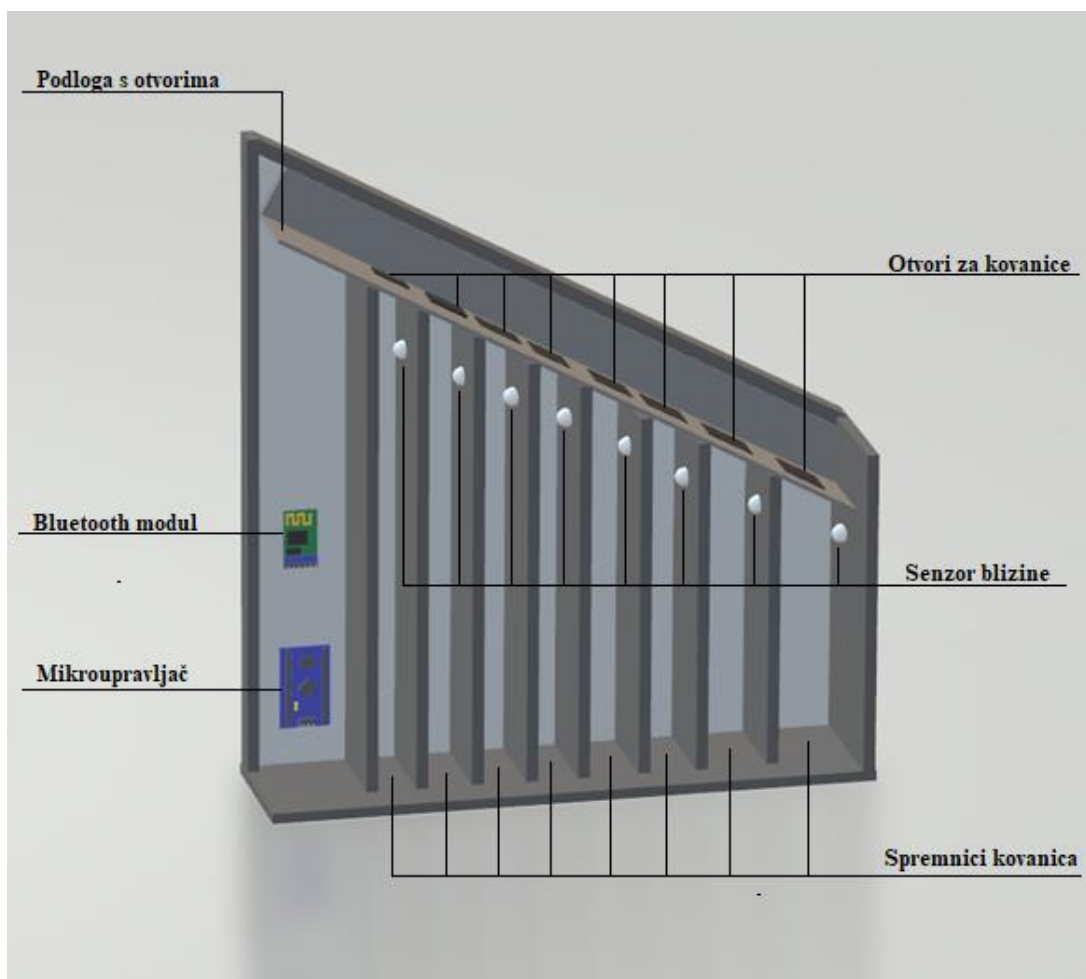


**Slika 2.1.** Blokovski prikaz prijedloga sklopovskog rješenja.

Prema slici 2.2 prikazan je prijedlog sklopovskog rješenja. Plan je izrezati otvore na površini neke podloge koji bi odgovarali promjeru kovanica, a redosljed otvora bi išao od najmanjeg promjera do najvećeg. Tu podlogu postaviti na vrh kasice pod blagim nagibom i zakrivljenijem, pri čemu je najviši dio podloge neizravno povezan s otvorom na vanjskom dijelu kasice. Tako da kad se kovanica umetne u kasicu, dolazi točno do tog najvišeg dijela podloge i polako klizi prema dolje.



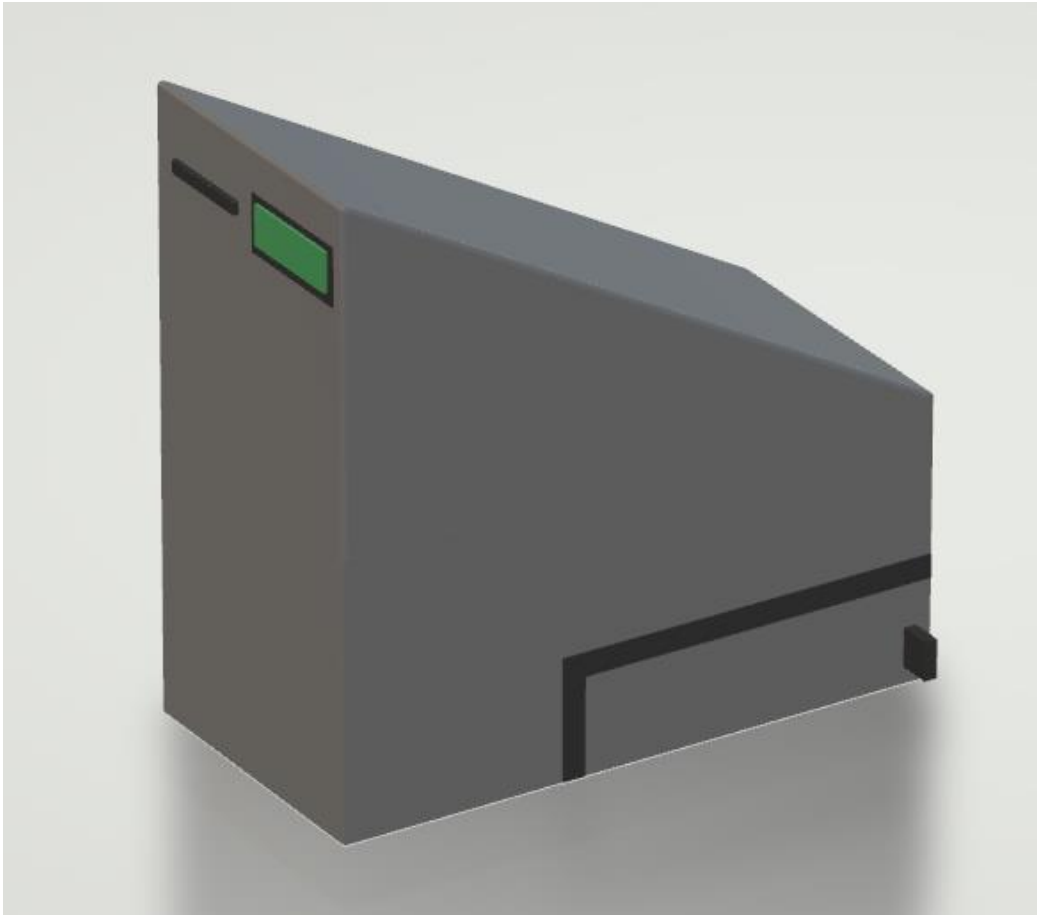
Tako će kovanica, ako odgovara promjeru otvora, upasti u nju, dok će, ako ne odgovara, proći preko i nastaviti prema sljedećem otvoru. Ispod svakog otvora bit će postavljeni senzori blizine koji će detektirati prolazak kovanice i zatim poslati signal mikroupravljaču koji će zabilježiti tu promjenu. Kako bi svaka kovanica imala svoje odgovarajuće mjesto potrebno je staviti i pregradu između svakog otvora, koja će dijeliti donji dio kasice na 8 spremnika za kovanice. Pokazivač bi se trebao nalaziti na vanjskom dijelu kasice, neposredno pokraj utora za kovanice. Modul za povezivanje s mobilnim uređajem kao i mikroupravljač nalazit će se u slobodnom prostoru unutar kasice, u dijelu koji će služiti kao centralni dio.



**Slika 2.2.** Prijedlog sklopovskog rješenja s poprečnim presjekom.

Prema slici 2.3 prikazan je prijedlog vanjskog izgleda kasice, prema kojem možemo vidjeti položaj utora za kovanice na lijevom boku. Rješenje za položaj pokazivača je da se smjesti na lijevoj strani prednjeg dijela kasice ili na lijevom boku neposredno pokraj otvora za kovanice.

Zbog praktičnosti i samog dizajna kasice, bolji prijedlog je staviti pokazivač na lijevi bok pokraj otvora.

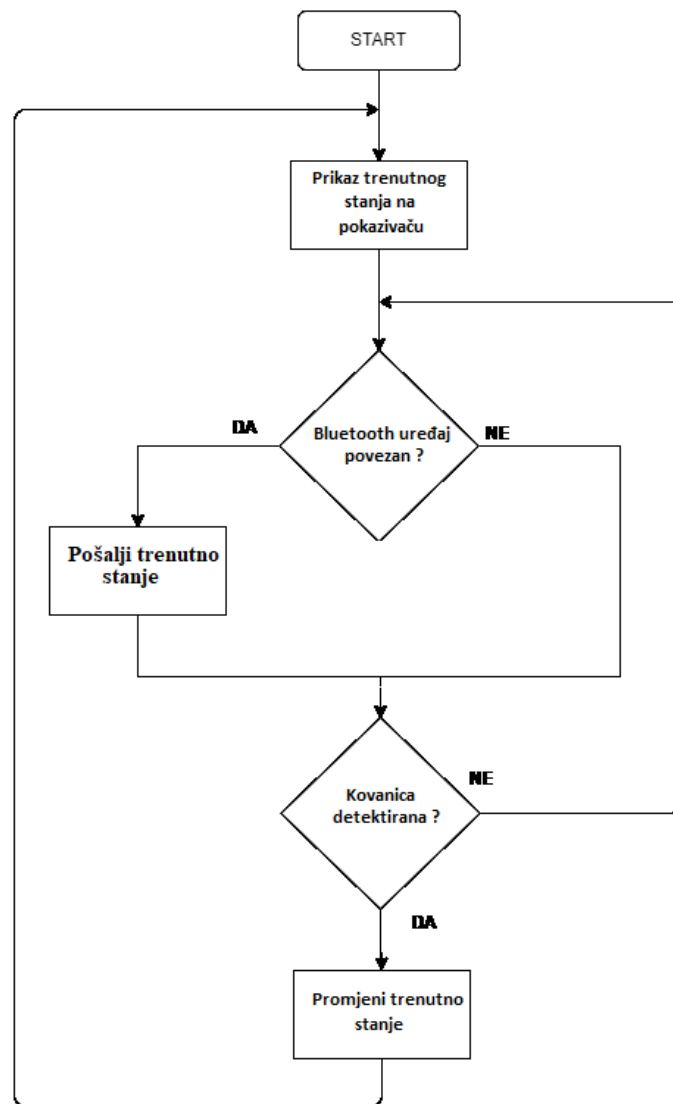


**Slika 2.3.** Prijedlog sklopovskog rješenja.

### **2.3. Prijedlog programskog rješenja**

Zadatak programskog dijela mikroupravljača izveden je na jednostavan način s obzirom da sama logika programa ne zahtijeva rad u više stanja jer nema više međusobno vremenski ovisnih zadataka, što znači da će se uvijek odvijati samo isti zadatci koji se zbog toga nalaze u jednoj petlji. Pri pokretanju programa, na pokazivaču se ispisuje trenutno stanje kasice i vrijednost prethodno umetnute kovanice, u isto vrijeme se pomoću bluetooth modula te vrijednosti šalju na povezani

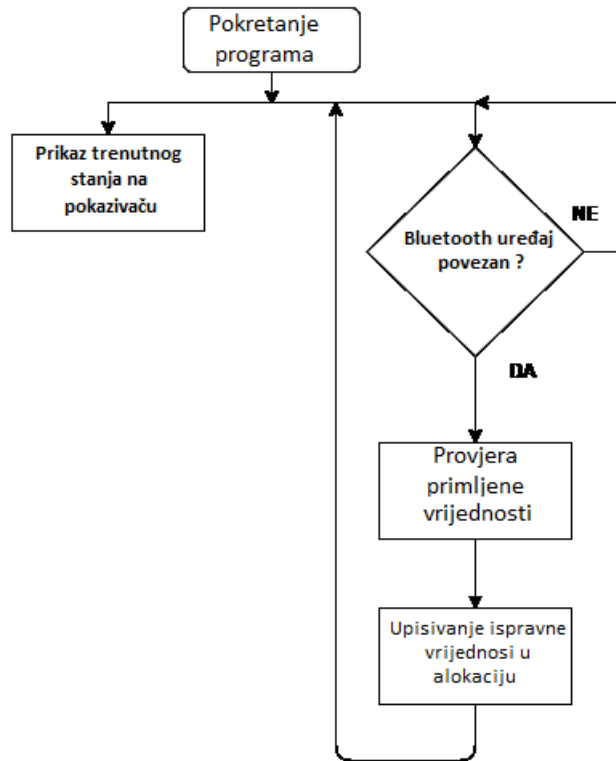
uređaj. Konstantno se provjerava postoje li primljeni signali o aktivnosti senzora blizine. Svaki senzor će biti zadužen za samo jednu vrstu kovanica, pa će se prilikom detekcije pozitivnog signala vrijednost ukupnog stanja povećati ovisno o kovanici koja je detektirana. Prema slici 2.4 možemo vidjeti jednostavan dijagram logike programskog dijela mikroupravljača koji objašnjava tok programa.



**Slika 2.4.** Pojednostavljeni prikaz dijagrama toka mikroupravljača.

Zadatak programskog dijela mobilne aplikacije je koncipiran na način da se pri pokretanju aplikacije prvo uspostavi povezivanje s bluetooth modulom unutar pametne kasice. Nakon uspješnog povezivanja mobilnog uređaja s modulom, provjeravaju se primljene vrijednosti i prikazuju na pozadini mobilnog uređaja. Radi dodatnog unosa informacija i bolje korisničke

interakcije, u aplikaciju je dodan pregled prethodno ubačene kovanice. Prema slici 2.5 možemo vidjeti dijagram toka mobilne aplikacije ,dok prema slici 2.6 možemo vidjeti predloženi izgled aplikacije.



Slika 2.5. Pojednostavljeni prikaz dijagrama toka mobilne aplikacije.

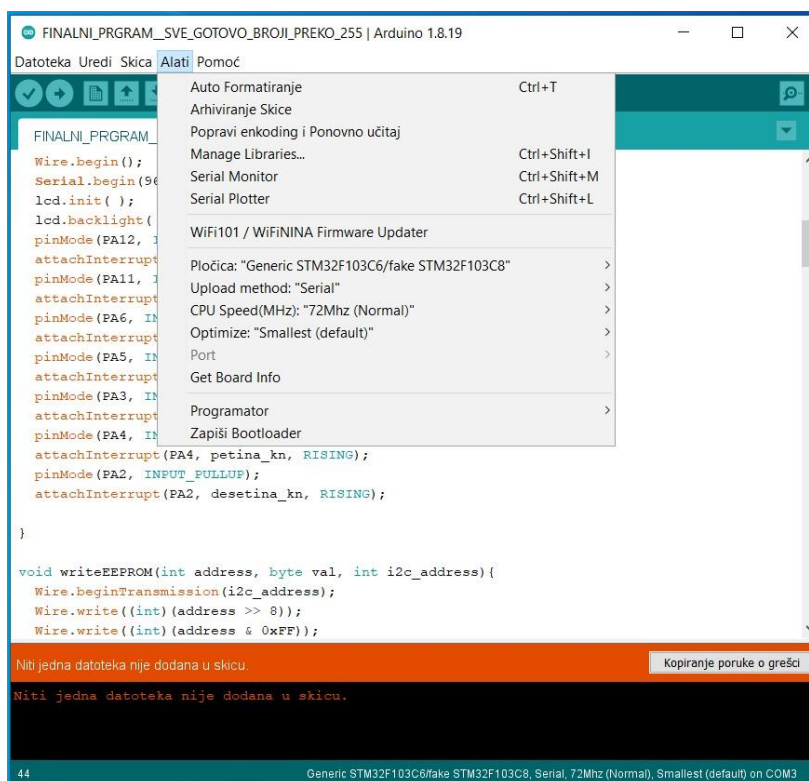


Slika 2.6. Vizualni prijedlog programa mobilne aplikacije.

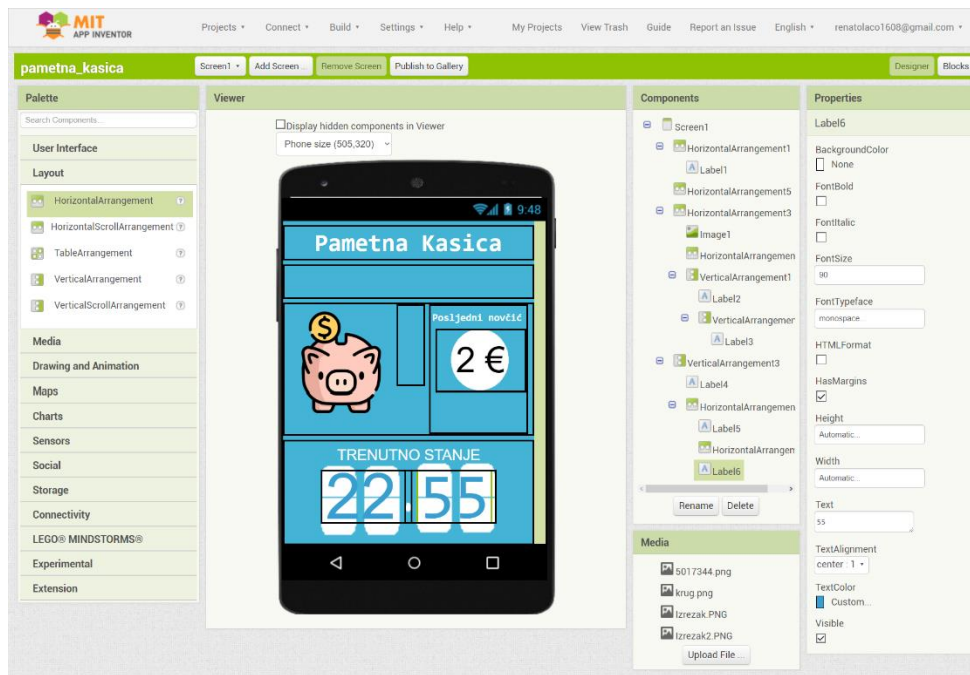
### 3. REALIZACIJA "Pametne kasice"

#### 3.1. Korišteni alati i programska okruženja

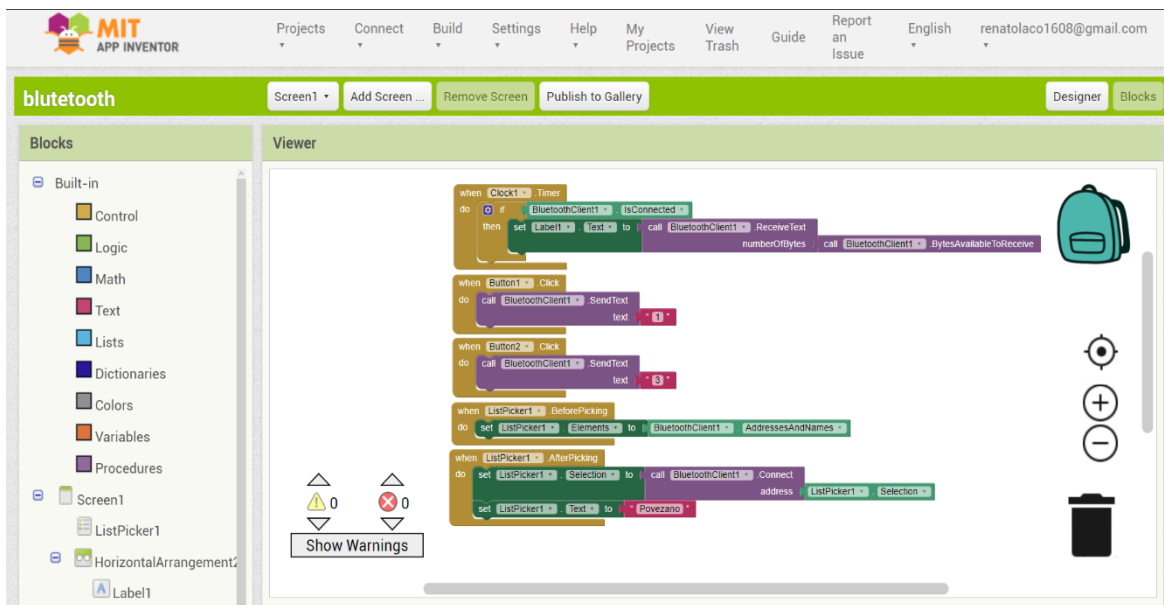
Prilikom izrade pametne kasice korišten je besplatan program Arduino *IDE* (engl. Integrated development environment, hrv. Integrirano razvojno okruženje, u daljnjem tekstu korišteno kao '*IDE*'), [10]. Besplatno razvojno okruženje koje se koristi za programiranje mikroupravljača. U njemu je realizirana cijela logika pametne kasice. Slika 3.1 prikazuje prilagođivanje potrebnih parametara kako bi se omogućilo programiranje STM32 mikroupravljača [11]. Osim programa Arduino *IDE* korištena je platforma MIT App Inventor, [9]. MIT App Inventor je platforma koja omogućuje jednostavno dizajniranje i programiranje mobilnih aplikacija bez potrebe za dubokim poznavanjem programiranja. Prema slici 3.2 možemo vidjeti način dizajniranja mobilne aplikacije, gdje se jednostavno postavljajući na zaslon različite komponente, vizualno dizajnira sučelje aplikacije. Nakon što se sučelje aplikacije definira, programiranje funkcionalnosti izvodi se u blokovskom editoru prikazanom prema slici 3.3. Vizualnim pristupom kodiranju, spajanjem blokova različitih funkcionalnosti definiramo sve akcije i reakcije koje će se događati u aplikaciji.



**Slika 3.1.** Potrebne postavke za omogućavanje serijske komunikacije Arduino *IDE* s STM32, [11].



Slika 3.2. Dizajniranje mobilne aplikacije unutar Mit App Inventor, [9].



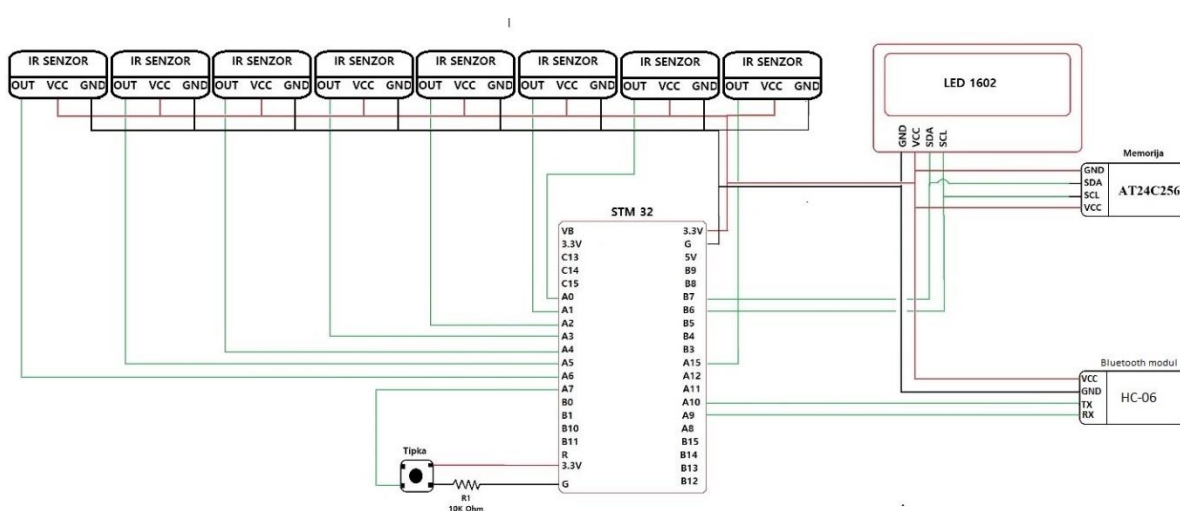
Slika 3.3. Programiranje mobilne aplikacije unutar Mit App Inventor, [9].

### 3.2. Realizacija sklopovskog rješenja

Temeljne komponente koje se koriste u ovom radu su mikroupravljač STM32F103C8TH, osam infracrvenih senzora TCRT5000 za detekciju kovanica, AT24C256 EEPROM memorija za pamćenje prethodnih stanja, LCD (eng. Liquid crystal display, hrv. Zaslona sa tekućim kristalima, u daljnjem tekstu korišten kao 'LCD') na kojemu je ispisano trenutno stanje kovanica, HC-06 bluetooth modul za komunikaciju s pametnim uređajem, te jedno tipkalo za resetiranje svih stanja.

S obzirom na to da mikroupravljač STM32 koristi *UART* (engl. Universal Asynchronous Receiver / Transmitter, hrv. Univerzalni asinkroni prijemnik / odašiljač, u daljnjem tekstu korišten kao '*UART*') serijsku komunikaciju, programiranje je izvršeno pomoću FT232RL odnosno *USB* (engl. Universal serial bus, hrv. Univerzalna serijska sabirnica, u daljnjem tekstu korišten kao '*USB*') prema *UART* čip, koji omogućuje komunikaciju uređaja koji koriste *UART* serijsku komunikaciju i računala putem *USB* porta. Preko njega je mikroupravljač također bio napajan u svim početnim testnim fazama. Pri završetku napajanje je prebačeno na punjač od 5 V, što se dalje ispostavilo kao najjednostavnija metoda.

Slika ispod prikazuje način na koji su te komponente međusobno povezane kako bi ovaj sustav ispravno radio.



Slika 3.4. Električna shema.

Predložena konstrukcijska schema iz poglavlja 2.2 iako vrlo slična, na kraju se dosta razlikuje od stvarne realizacije pametne kasice. Svi infracrveni senzori pozicionirani su na drukčiji način, jer je u slučaju s predložene scheme kovanica mogla padati s licem prema dolje, što rezultira tome da se emitirana infracrvena svjetlost ne reflektira u dovoljnoj količini kako bi senzor to i zabilježio. Povodom toga ispod svih otvora postavljena je još jedna drvena pločica koja će sve kovanice u početnoj fazi padanja rotirati, tako da padaju pod kutom od oko 90 stupnjeva, a senzor je postavljen tako da gleda prema prednjoj strani kasice, točno na sredini prolaza kovanice, kako niti jedna ne bi bila promaknuta. *LCD* je dobio novu poziciju, na prednjoj strani kasice. Razlog tome su veća jednostavnost u spajanju na takav način, ali i zbog boljeg vizualnog izgleda. Neposredno pored *LCD-a*, dodana je i tipka koja je zadužena za resetiranje trenutnog stanja kasice. Od manje bitnih elemenata koje ne mijenjaju funkciju pametne kasice, promijenjen je i mehanizam otvaranja vrata, koji je sada prema gore.

Jedan od temeljnih problema sustava pametne kasice je upravo detekcija kovanica. Navedeni problem zahtjeva tehničko rješenje te će njegova realizacija biti objašnjena u ovom poglavlju. Kako je prethodno spomenuto, za razvrstavanje kovanica od 1, 2, 5, 10, 20, 50 centi i 1 i 2 eura, korištena je ukošena metalna pločica s 8 rupa različitih dimenzija, točnije svaka rupa jednaka je dimenziji jedne kovanice. Rupe su poredane od manje prema većoj, te će tako svaka kovanica moći upasti samo u svoju rupu. Najveći problem je vrlo mala razlika u dimenzijama kovanica, pa je tako naprimjer razlika u promjeru kovanice od 50 centi i 1. eura samo jedan milimetar. Zbog toga pločicu nije bilo moguće izraditi ručno, već je izrezana laserskim rezanjem pomoću *CNC* (engl. Computer numerical control, hrv. računalno numeričko računanje , u daljnjem tekstu korišten kao '*CNC*' ) stroja. Zbog sigurnosti da će kovanice slobodno klizati po pločici, a u isto vrijeme i biti kvalitetan i pouzdan, bilo je potrebno pomno odabrati materijal od kojeg će ona biti napravljena. Iz tog razloga se nehrđajući čelik već pripremljen za rezanje pokazao kao odličan materijal za ovu svrhu zbog njegove, glatkosti, debljine i čvrstoće. Slika ispod pokazuje dva metalna dijela koji su kasnijim spajanjem izgledali kao T profil.





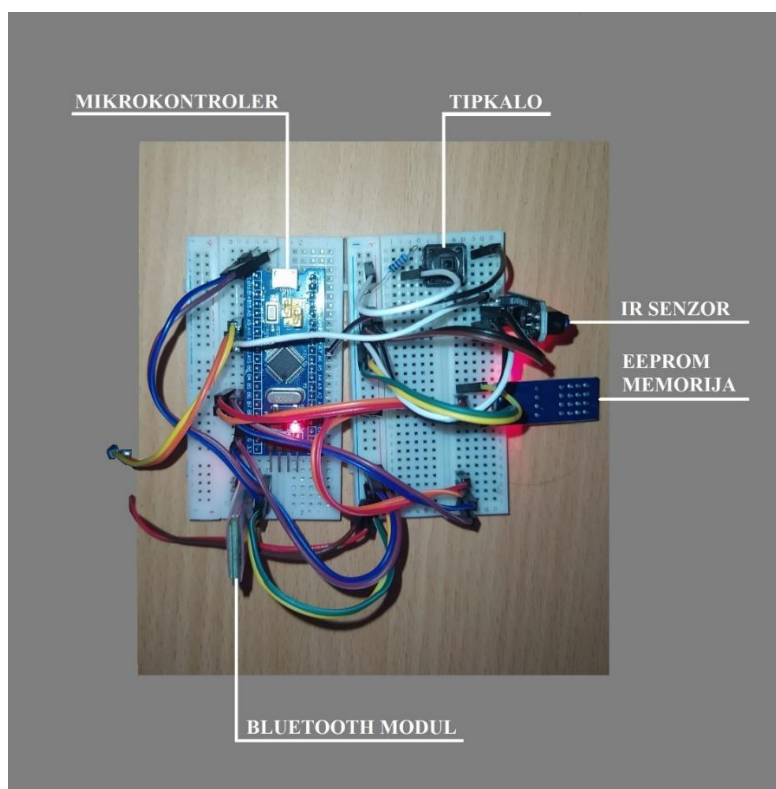
**Slika 3.5.** Metalne pločice izrezane *CNC* strojem.

Problem razvrstavanja kovanica uspješno je riješen pomoću prethodno prikazane pločice. Kada se kovanica stavi na početak, gravitacija ju povlači prema dolje te uspješno klizi uz lijevi brid. Ostavljeno je prostora između brida i otvora kako bi se kovanice nesmetano mogle kretati do otvora u koji treba upasti. Kovanica upada u svoj otvor zato što je udaljenost između lijevog brida i desne strane otvora malo veća od promjera kovanice, ali opet ne više od dimenzija kovanica čiji je promjer veći od te kovanice. Tako je kovanica oslonjena samo na lijevo strani metalne pločice između brida i rupe, te ju gravitacija povlači prema dolje. U skroz lijevu rupu sa slike upasti će kovanica od 1 centa, jer ima najmanji promjer, zatim otvori idu po veličinama prema 2 centa, 10 centi, 5 centi, 20 centi, 1 euro, 50 centi i na kraju 2 eura. Nakon prikupljenih svih potrebnih dijelove i potvrde da pločica ispravno radi, započeto je sastavljanje kasice.



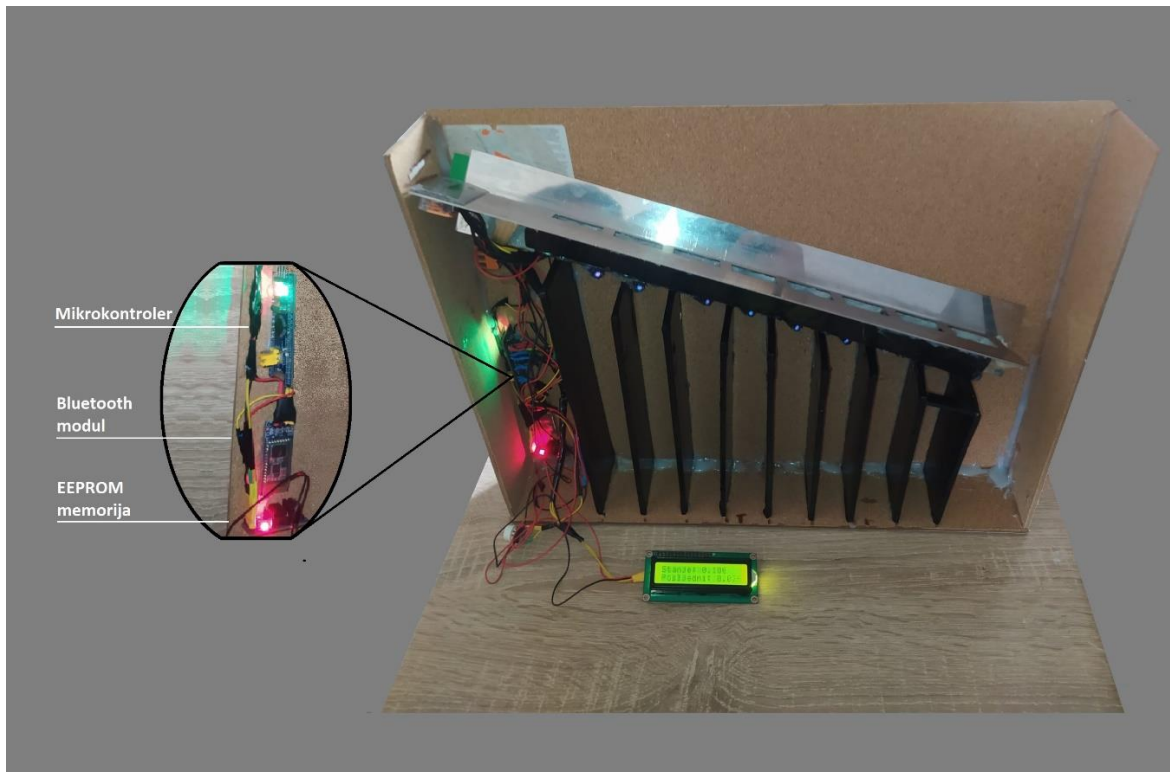
**Slika 3.6.** Metalna pločica.

Kako bi se sve komponente mogle početi sastavljati, u prvom koraku je najpotrebnije bilo odrediti točan položaj same pločice u kasici. Bilo je potrebno pozicionirati ju pod točnim kutom i nagibom kako bi kovanice mogle kliziti prema dolje uz brid, a da pri tom ne postignu ili preveliku brzinu ili premalu, da ne bi stale ili prebrzo prešle preko svog otvora, te ne bi uspješno upale. Nakon puno eksperimentiranja određen je pogodan nagib i kut te je pločica zalijepljena na zadnji dio kasice.



**Slika 3.7.** Testiranje komponenata.

Prije početka slaganja svih komponenata na samo kućište kasice, provedeno je ispitivanje na eksperimentalnoj pločici. Cilj ovoga bio je osigurati da svaka komponenta radi ispravno i međusobno se povezuju u funkcionalnu cijelinu. Nakon što je potvrđeno da sve komponente rade ispravno započelo je slaganje tih komponenti na kućište pametne kasice.



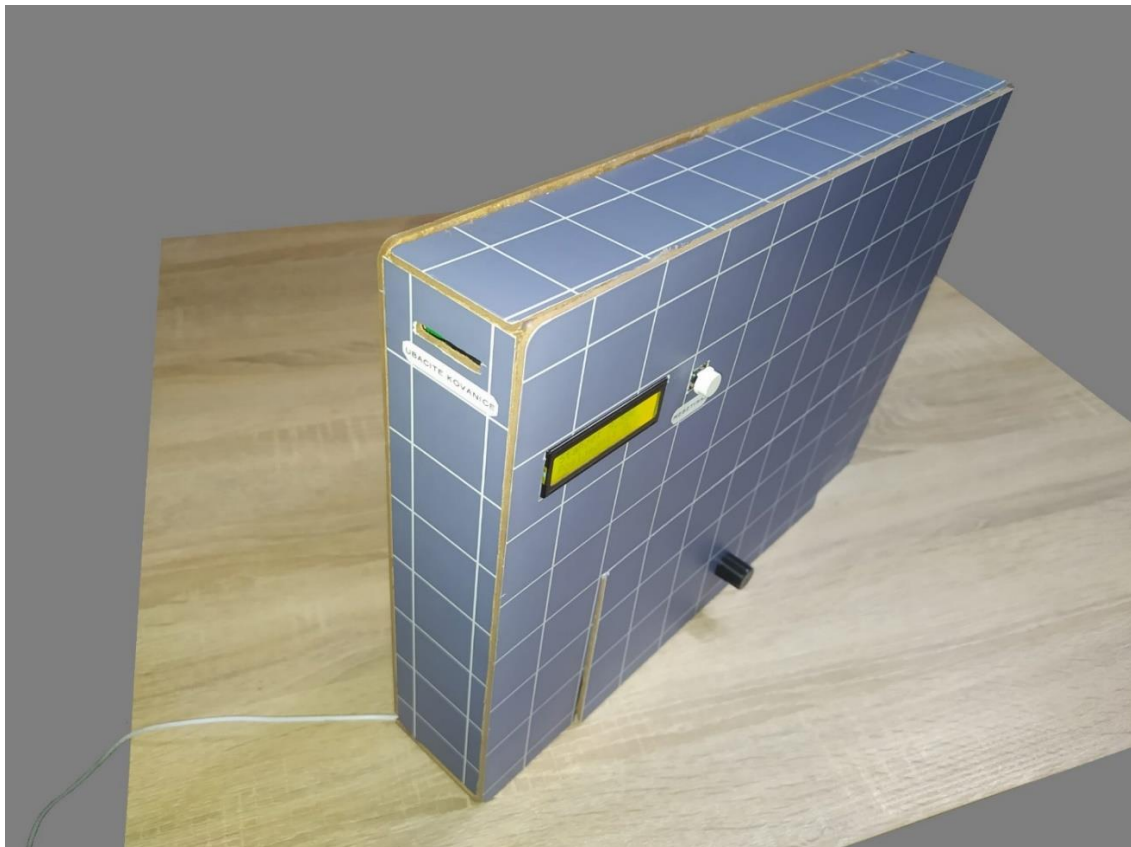
**Slika 3.8.** Unutarnji izgled kasice.

Nakon stavljanja senzora uočen je problem da zbog male udaljenosti senzora blizine i svih ostalih komponenata, dolazi do smetnji, pa bi se tako senzor kad se upali, ostao upaljen zauvijek. To je riješeno postavljanjem crne izolir trake oko bližnjih komponenata kako bi se svjetlost što manje odbijala. Prema slici 3.7 možemo vidjeti samu unutrašnjost pametne kasice. S obzirom na to da metalna pločica na početnom dijelu nema nikakve otvore kako bi kovanice mogle postići brzinu, ispod je ostao slobodan prostor pa su tu pozicionirani mikroupravljač, EEPROM memorija, i bluetooth modul. Ispod svih rupa su postavljeni senzori i pregrade kako bi svaka kovanica imala svoj odvojeni dio za skladištenje. Prema slici 3.8 možemo vidjeti unutrašnji dio kasice koji je dostupan korisnicima kao skladište kovanica koje su sortirane prema promjeru od najmanje kovanice prema najvećoj. Spremniku se pristupa podizanjem vrata koja se nalaze na prednjoj strani kasice. Vrata, radi bolje čvrstoće prilikom upadanja kovanica u spremnike, na sebi imaju dvije metalne pločice koje se nakon zatvaranja drže za magnete postavljene na pregrade.



**Slika 3.9.** Prednja strana kasice.

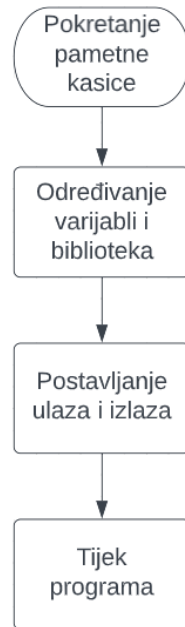
Prema slici 3.10 možemo vidjeti konačan izgled pametne kasice i poziciju otvora na bočnoj strani kasice kroz kojeg prolaze kovanice i dolaze na vrh metalne pločice, poziciju *LCD*-a i tipkala na prednjoj strani kasice.



**Slika 3.10.** Vanjski izgled pametne kasice.

### 3.3. Realizacija programskog rješenja

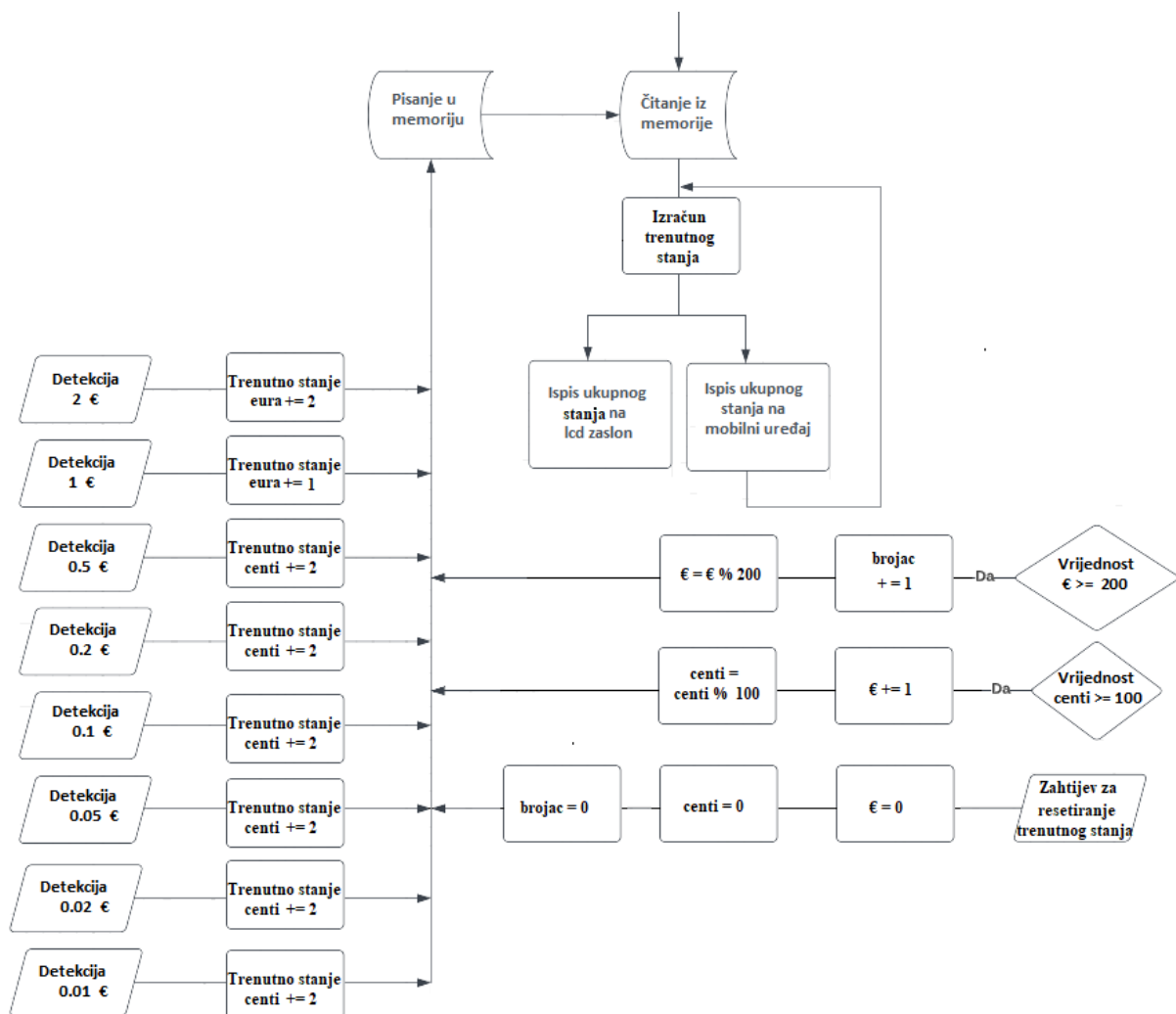
S obzirom na to da za program pametne kasice nije potrebno više stanja, sama logika realizirana je na vrlo jednostavan način gdje se apsolutno sve konstantno obavlja u jednoj petlji. Dijagram ispod pokazuje dijelove programa sve do dijela same petlje.



**Slika 3.11.** Početni dio dijagrama toka, inicijalizacija i postavljanje elemenata.

U prvom dijelu programa definirane su sve globalne varijable koje se koriste u programu. Tu su i uključene sve biblioteke potrebne za normalan rad programa. Nakon toga slijedi 'setup' dio gdje su definirani svi ulazi i izlazi programa. Sama detekcija kovanica odrađena je preko dijela koda koji poziva funkciju za zbrajanje kovanica na detekciju pozitivnog brida, ovisno o tome koji je senzor aktiviran, na koji je digitalni ulaz na mikroupravljaču spojen, pozvat će se određena funkcija definirana točno za tu kovanicu. Osim za detekciju kovanica, funkcija za resetiranje trenutnog stanja se isto tako pali na detekciju pozitivnog brida koji se događa kad se pritisne tipka. Još dvije funkcije zaslužne su za čitanje i pisanje vrijednosti u EEPROM. Program također konstantno ispisuje na 'serial monitor' trenutno stanje i vrijednost posljednje ubačene kovanice, ta se vrijednost ako je bluetooth uređaj povezan, na njega šalje. Isto tako se preko serial monitora provjerava postoje li ikakve primljene poruke, ako da trenutno stanje i vrijednost posljednje kovanice postaviti će se na 0.

Dijagram tijeka programa mikroupravljača prikazan na slici 3.12 pokazuje detaljan tijek izvršavanja zadatak i rezultate koji nastaju prilikom određenih uvijeta i zahtijeva. Brojač postoji jer naša memorija može primiti maksimalnu vrijednost 255, zato kad vrijednost eura pređe 200, u brojač se zbroji plus 1, a euri postaju ostatak pri dijeljenju 200. U daljnjem dijelu programa izračunava se stvarno trenutno stanje eura tako da se zbroji trenutni iznos varijable eura i iznos varijable brojača pomnožen sa 200. Ovom metodom povećan je kapacitet maksimalne količine eura sa 255 na 51199.

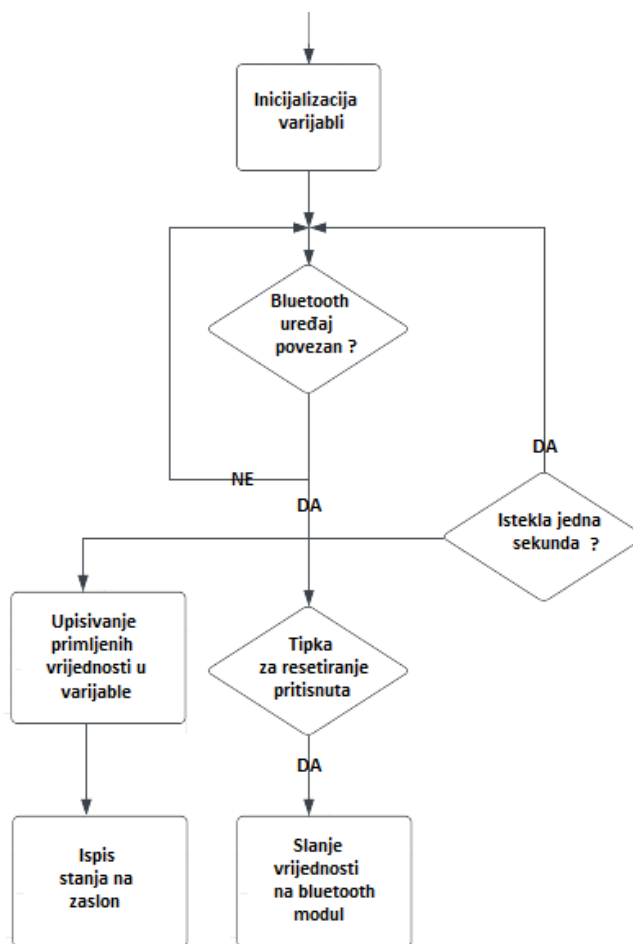


Slika 3.12. Detaljni dijagram tijeka programa mikroupravljača.

Struktura programa mobilne aplikacije zbog svoje jednostavnosti posla, a kojemu je najbitnije samo ispisati trenutno stanje pametne kasice, izvedena je na vrlo jednostavan način i ne zahtijeva nikakvu algoritmičku složenost jer se radi o jednostavnim programskim zadacima. Od same inicijalizacije varijabli do kraja cijelog programa postoji još samo nekoliko zadataka. Programska

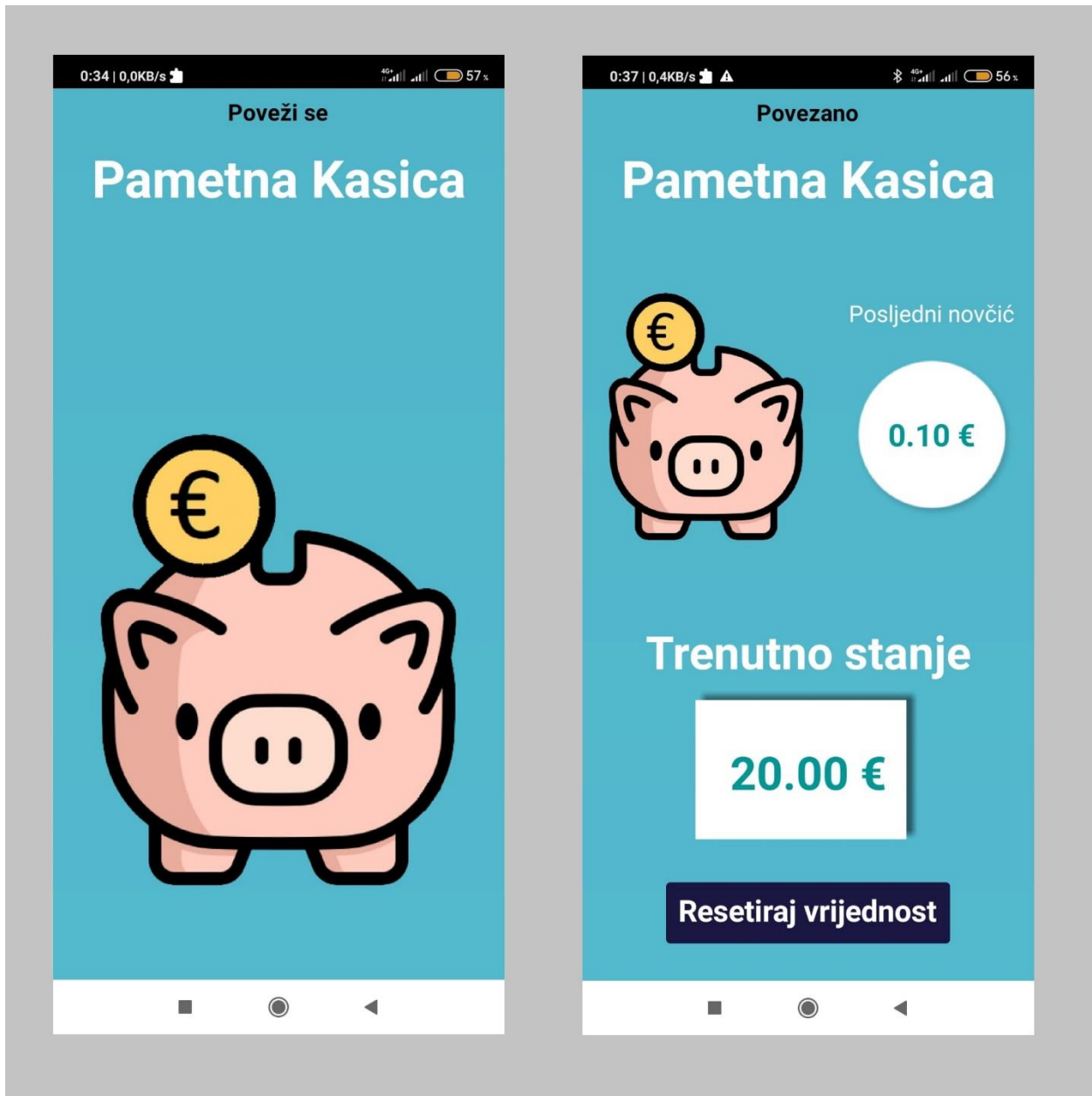
petlja zamišljena je tako da svake sekunde provjerava je li bluetooth uređaj povezan, u ovom slučaju gledamo na bluetooth modul implementiran unutar pametne kasice, povezivanje nekog drugog uređaja osim našeg neće imati nikakav rezultat, jer je u programu definirano da prima točno specificirane podatke koje se šalju preko našeg mikroupravljača. Osvježenje od jedne sekunde sasvim je dovoljno kako bi program bio što manje zasićen podacima, ali da u isto vrijeme zabilježi sve nastale promjene na samoj pametnoj kasici. Nakon primljenih vrijednosti o trenutnom stanju i posljednjoj kovanici, ispisuju se na početni zaslon aplikacije. Program također konstantno provjerava je li pritisnuta tipka za resetiranje trenutnog stanja, ako je tipka pritisnuta, šalje se poruka na bluetooth modul.

Prema slici ispod 3.12 možemo vidjeti cijeli dijagram toka mobilne aplikacije pametne kasice.



**Slika 3.13.** Dijagram toka programa mobilne aplikacije.

Na slici 3.14 prikazan je vizualni izgled programa mobilne aplikacije, gdje na lijevoj strani vidimo prikaz početnog zaslona, tek nakon što je program pokrenut. Na pritisak tipke 'Poveži se' otvara se lista s dostupnim bluetooth uređajima u blizini. Nakon što se uspješno povežemo, na zaslon se prikazuje trenutno stanje, posljednji novčić, i tipka za resetiranje vrijednosti, kao što vidimo na desnoj slici.



Slika 3.14. Vizualni izgled mobilne aplikacije.



## 4. TESTIRANJE I REZULTAT

### 4.1. Metodologija testiranja

Nakon što je spojeno sklopovlje i napisana cijela programska logika, bilo je potrebno testirati proces i njegove dijelove te se uvjeriti da sve radi kako bi trebalo.

Metodologiju testiranja rada „Pametne kasice“ možemo podijeliti na 4 dijela, to su:

- Testiranje rada odvojenih komponenti sklopovlja
- Testiranje upadanja kovanice u predviđen otvor
- Testiranje detekcije svake vrste kovanice
- Ostale smetnje prilikom testiranja

### 4.2. Rezultati testiranja

#### 1. Testiranje rada komponenti sklopovlja.

Svi infracrveni senzori ispitani su radi ispravne detekcije predmeta. Cilj testiranja je vidjeti koja je minimalna udaljenost na koju kovanica mora doći da bi bila detektirana. S obzirom da se najmanje svjetlosti reflektira okrenemo li rub kovanice prema senzoru, testiranje je tako i provedeno jer znamo da će senzor u svakom drugom položaju kovanice dati puno bolje rezultate. Detekcija kovanice je provedena 20 puta, a rezultati testiranja nalaze se u tablici 4.1. gdje se iznad nalazi redni broj provedenog testa, a ispod njegov rezultat.

1.	2.	3.	4.	5.	6.	7.	8.	9.	10.
1.3 cm	1.4 cm	1.4 cm	1.5 cm	1.6 cm	1.1 cm	1.3 cm	1.3 cm	1.2 cm	1.2 cm
11	12.	13.	14.	15.	16.	17.	18.	19.	20.
1.1 cm	1.5 cm	1.4 cm	1.5 cm	1.4 cm	1.5 cm	1.6 cm	1.3 cm	1.4 cm	1.2 cm

**Tablica 4.1.** Mjerenja detekcije infracrvenih senzora.

Srednju vrijednost ćemo dobiti pomoću sljedeće formule :

$$lsr = \frac{1}{n} \sum_{i=1}^n li$$

$$lsr = 1.36 \text{ cm}$$

Srednje kvadratno odstupanje pojedinih mjerenja i srednje kvadratno odstupanje aritmetičke sredine su:

$$m = \sqrt{\frac{\sum_{i=1}^n (li - lsr)^2}{n - 1}}$$

$$m = 0,225 \text{ cm}$$

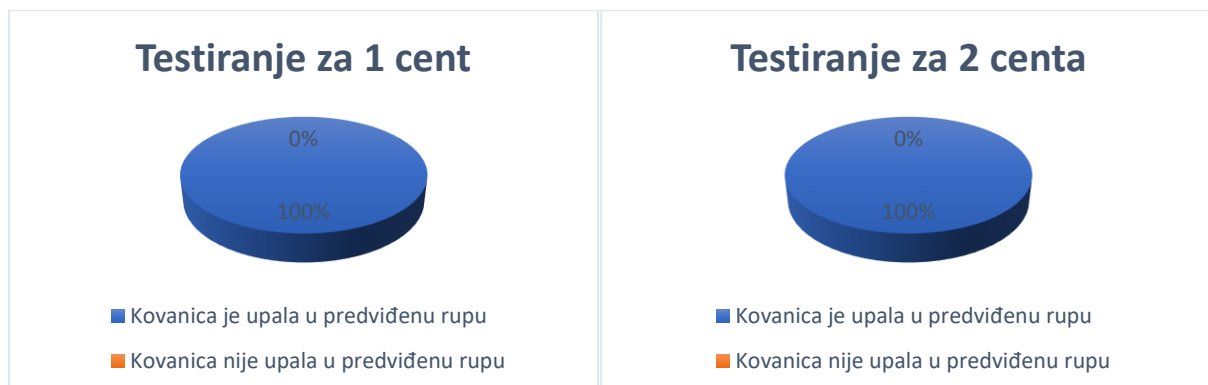
$$M = \frac{m}{\sqrt{n}}$$

$$M = 0,05 \text{ cm}$$

Prema rezultatima provedenih testiranja možemo vidjeti kako je najmanja udaljenost na kojoj senzor detektira objekat 1.1 cm, što nam je vrlo važna informacija za ovaj zadatak, zbog toga što saznajemo na kojoj optimalnoj maksimalnoj udaljenosti mora biti kovanica kako bi ju senzor uspio detektirati.

## 2. Testiranje upadanja kovanica u predviđen otvor

S obzirom na to da je temeljni mehanizam razvrstavanja kovanica, sama pločica s otvorima promjera kovanica, što dalje vodi ka detekciji, vrlo je bitno da radi savršeno precizno. Ovaj test je proveden nakon što je kasicica bila potpuno gotova sa svojom izradom. Test je proveden tako da je 10 različitih kovanica jedne vrste ubačeno u kasicu, zatim izvađeno iz kasicice i tako ponovljeno još 9 puta, što ukupno dovodi do 100 testova gdje je 10 različitih kovanica ubačeno 10 puta. Na isti princip provedena su testiranja na svih 8 vrsti kovanica. Rezultati testiranja prikazani su pomoću tortnog dijagrama.





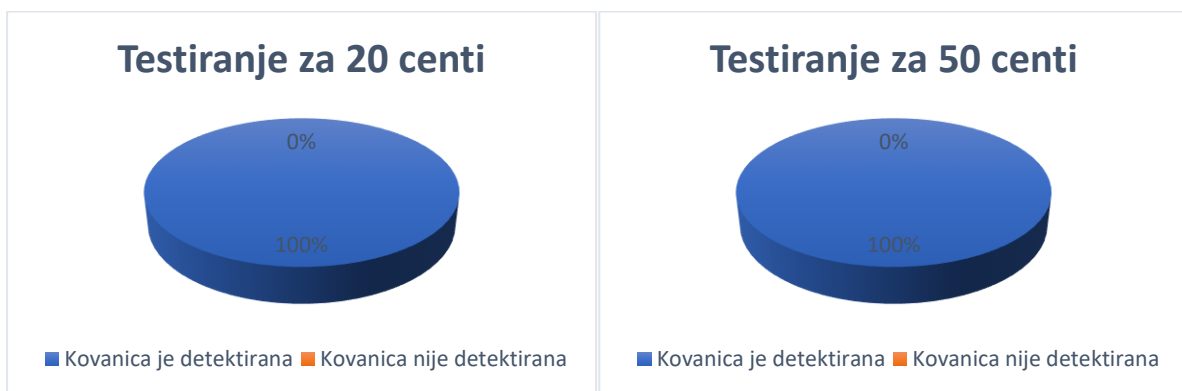
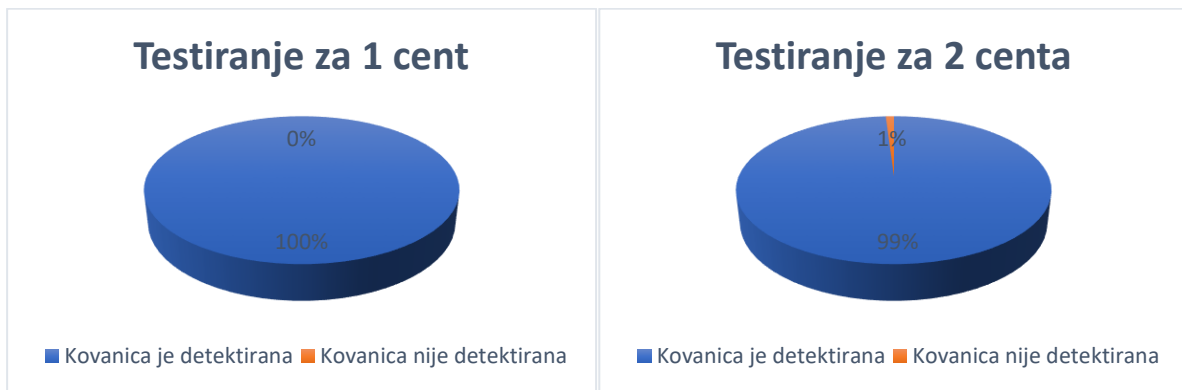
Iz provedenih testiranja možemo vidjeti kako je samo kovanica od 20 centi jednom upala u krivi otvor, što nam daje 99 točnih upadanja i 1. krivo, sve ostale kovanice savršeno su upale u svoj spremnik.

### 3. Testiranje detekcije svake vrste kovanice

Kako je sam mehanizam razvrstavanja kovanica bitan i za samu detekciju, bilo je potrebno provesti i ta testiranja. Od velikog značaja ovom testiranju bio je pregled prethodno ubačene kovanice sa *LCD-a*.

Testiranje upadanja kovanice u predviđen otvor i testiranje detekcije te kovanice predviđenim senzorom provedeno je u isto vrijeme, pa tako uzimamo u obzir 100 obavljenih testiranja za sve vrste kovanica, osim kovanice od 20 centi koja je u svoj spremnik upala 99 puta.

Sljedeći dijagrami prikazuju rezultate testiranja detekcije kovanica .

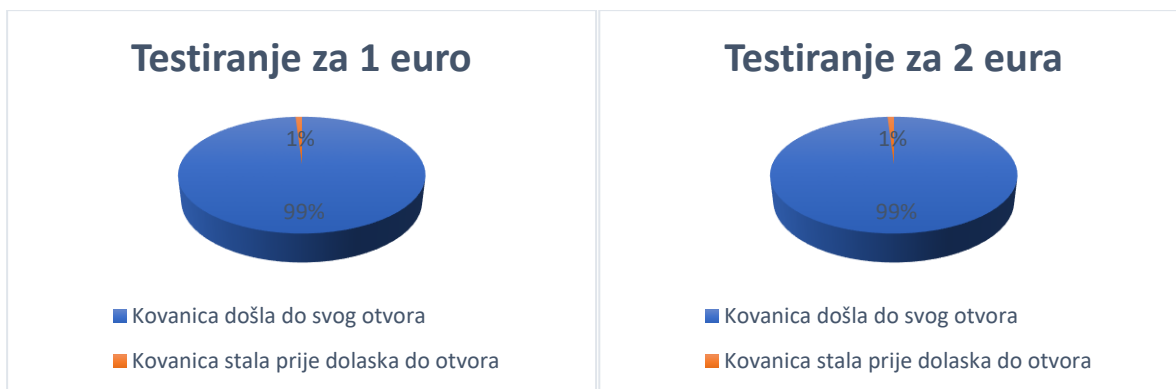


Temeljem provedenih rezultata možemo vidjeti kako samo senzor od 2 centa nije uspio jednom detektirati kovanicu. Svi ostali senzori na prolazak kovanice uspješno su reagirali.

#### 4. Ostale smetnje nastale prilikom testiranja

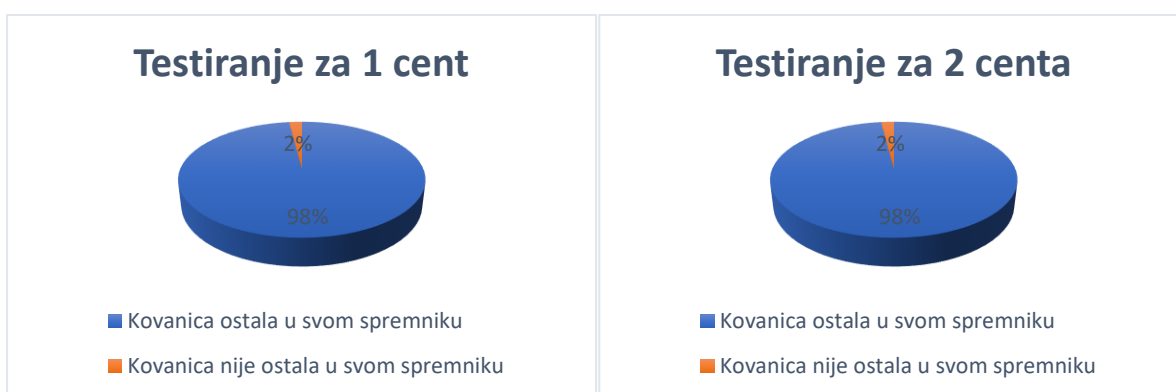
Temeljem provedenog testiranja bitno je napomenuti i ostale probleme pametne kasice. Jedan od njih je da kovanica prilikom pada do svog otvora, zbog ne postizanja dovoljne brzine jednostavno stane na mjestu. Te smetnje uzete su u obzir provedenim testiranjem iznad, pa isto tako možemo gledati da se to testiranje provelo od 100 puta.



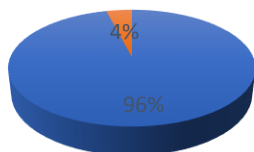


Temeljem prikazanih dijagrama iznad možemo vidjeti kako je najveći broj puta zastala kovanica od 5 centi, i to 4 puta. Kovanice od 1 i 2 eura zastale su po jedan put. Ti problemi riješeni su podizanjem lijeve strane kasice prema gore, tako da se postigne još veći kut metalne pločice, pa bi kovanice nastavile kliziti prema dolje. Prilikom nastanka ovih smetnji i rješavanjem ovim pristupom, svaki put od 6 stajanja kovanice su uspješno došle do svog otvora.

Još jedna smetnja koja je nastajala je ta da bi kovanica nakon što uspješno prođe kroz svoj otvor, zatim uspješno bude detektirana i upadne u svoj spremnik, nakon pada na tlo, se otkotrlja u spremnike pored, što nastaje jer rub spremnika i vrata dijeli magnet, pa tako imamo otprilike 0.3 cm slobodnog prostora kroz koje kovanice mogu proći lijevo ili desno. Nastale smetnje prikazane su dijagramima ispod.

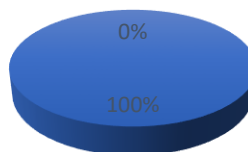


### Testiranje za 5 centi



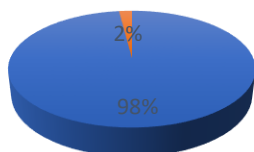
- Kovanica ostala u svom spremniku
- Kovanica nije ostala u svom spremniku

### Testiranje za 10 centi



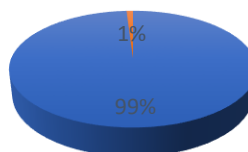
- Kovanica ostala u svom spremniku
- Kovanica nije ostala u svom spremniku

### Testiranje za 20 cent



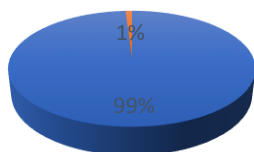
- Kovanica ostala u svom spremniku
- Kovanica nije ostala u svom spremniku

### Testiranje za 50 centi



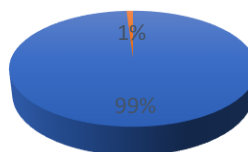
- Kovanica ostala u svom spremniku
- Kovanica nije ostala u svom spremniku

### Testiranje za 1 euro



- Kovanica ostala u svom spremniku
- Kovanica nije ostala u svom spremniku

### Testiranje za 2 eura



- Kovanica ostala u svom spremniku
- Kovanica nije ostala u svom spremniku

## 5. ZAKLJUČAK

Zadatak ovog završnog rada bio je izraditi sustav za automatizirano prebrojavanje i sortiranje kovanica te prikazom tih vrijednosti na pokazivač, te na mobilnoj aplikaciji. Rad je izvršen koristeći razne resurse iz otvorenih izvora i primjenom vlastitih metoda, te korištenje iskustva na izradi već sličnog projekta. S obzirom na ne poznavanje izrade mobilnih aplikacija, to se pokazalo kao jedan od najvećih izazova. Međutim temeljnim istraživanjem otkriven je jednostavan sustav Mit App Inventor, koji je dizajniran za početnike i one bez iskustva u programiranju, što je uveliko pomoglo u izradi cijele aplikacije.

Sam dizajn i način izrade cijele kasice temeljio se na već izrađenom projektnom zadatku, pa možemo reći da je ovo nadogradnja na već postojeći sustav gdje su ispravljani neki nedostaci poput korištenja pločice s otvorima od metala, radi postizanja bolje preciznosti detekcije samih kovanica, te izrade cijelog sustava od drveta radi bolje izdržljivosti i kvalitete. Nagib i kut pločice s otvorima utjecali su na promjenu brzine ubačenih kovanica, pa tako i kut senzora te orijentiranost kovanice prilikom pada u svoj spremnik, radi njezine detekcije. Zbog velike osjetljivosti na bilo kakve promjene koje su nastale, sam sustav zahtijevao je testiranje nakon svake nadogradnje. Pa je tako i dolazilo do nekoliko izmjena koje su na kraju rezultirale s gotovo savršenom preciznošću rada kasice.

Glavna mana ovog sustava je ta što kasica treba biti položena na ravnu površinu, svaka promjena kuta mogla bi utjecati na krivu detekciju i sortiranje. U vidu poboljšanja cijelog sustava mogla se dodati još jedna tipka odnosno prekidač za uključivanje/isključivanje koji bi se dodao prije napajanja mikroupravljača kako se prilikom svakog rada kasica ne bi morala iznova uključivati u utičnicu. Još jedna mana je nedostatak korisničkog pristupa unutar aplikacije pametne kasice, ili eventualno kreiranje zaporke koja bi se trebala upisati prilikom svakog korištenja. Isto tako temeljem provedenih testiranja možemo vidjeti gdje i kako su smetnje nastajale pa bi i poboljšanju sortiranja kovanica pomoglo da između vrata i rubova spremnika nema ni malo prostora kako kovanice ne bi mogle prolaziti iz jednog spremnika u drugi.



## LITERATURA

- [1] *Mehaničko sortiranje kovanica*, <https://kloppcoin.com/products/coin-sorters/klopp-model-sm-manual-coin-sorter/>, Pristup:29.10.2023.
- [2] *Detekcija kovanica pomoću kamere*, [https://courses.ece.cornell.edu/ece5990/ECE5725\\_Spring2018\\_Projects/sl2954\\_gc535/website/s-ample\\_cool\\_website/index.html](https://courses.ece.cornell.edu/ece5990/ECE5725_Spring2018_Projects/sl2954_gc535/website/s-ample_cool_website/index.html), Pristup: 29.10.2023.
- [3] *Uređaj Nadex za sortiranje kovanica*, <https://www.worthpoint.com/worthopedia/vintage-nadex-coin-sorter-original-491924214>, Pristup: 29.10.2023.
- [4] *Sortiranje kovanica koristeći rotirajući disk*, <https://www.ribaostore.com/products/7-pockets-coin-sorter-cs-600b>, Pristup: 30.10.2023.
- [5] *Stari mehanički uređaj za sortiranje kovanica*, <https://www.liveauctioneers.com/price-result/early-standard-johnson-co-coin-counter>, Pristup: 30.10.2023.
- [6] *Sortiranje kovanica klizanjem niz podlogu sa otvorima*, <https://www.electromaker.io/project/view/arduino-based-coin-sorting-and-counting-machine>  
Pristup: 1.10.2023.
- [7] *Detekcija kovanice temeljem ultrazvučnog senzora*, <https://lets.gethacking.com/activities/coin-sorter-with-micro-bit>, Pristup: 1.10.2023.
- [8] *Detekcija kovanica temeljem digitalne vage*, <https://www.fraudfighter.com/products/counteasy>, Pristup: 1.10.2023.
- [9] *Sustav Mit App Inventor*, <https://ai2.appinventor.mit.edu>, Pristup: 15.09
- [10] *Program Arduino IDE*, <https://www.arduino.cc/en/software> , Pristup: 5.8.2023
- [11] *Programiranje STM32 putem Arduino IDE*, [https://www.sgbotic.com/index.php?dispatch=pages.view&page\\_id=48](https://www.sgbotic.com/index.php?dispatch=pages.view&page_id=48), Pristup : 5.8.2023.
- [12] *Prikaz pinova STM32 mikroupravljača*, <https://components101.com/microcontrollers/stm32f103c8t8-blue-pill-development-board>,  
Pristup: 2.9.2023.

[13] *USB prema UART serijski prilagodnik*, <https://how2electronics.com/stm32-bootloader-programming-usb/>, Pristup: 2.9.2023.

[14] *Programiranje EEPROM memorije pomoću STM32 mikroupravljača*, <https://www.instructables.com/Reading-and-Writing-Data-to-External-EEPROM-Using-/>, Pristup: 2.9.2023.

[15] *Korištenje LCD 16x2 zaslona sa Arduino IDE*, <https://lastminuteengineers.com/i2c-lcd-arduino-tutorial/>, Pristup: 2.9.2023.

[16] *HC-06 Bluetooth modul*, <https://components101.com/wireless/hc-06-bluetooth-module-pinout-datasheet>, Pristup: 2.9.2023

## SAŽETAK

Zadatak završnog rada bio je projektirati, izraditi i testirati sustav za automatizirano prebrojavanje i sortiranje kovanica, odnosno izraditi pametnu kasicu. Sustav treba imati mogućnost prikaza trenutnog stanja pametne kasice na pokazivač i na mobilnoj aplikaciji. Nadalje, u radu su objašnjeni prednosti i nedostaci prijašnjih metoda sortiranja i detekcije kovanica, prema čemu je odlučeno za metodu koja kombinira mehaničke i elektroničke komponente. Objašnjen je način spajanja svih komponenti i alata koji su se koristili kako bi sustav funkcionirao. Cijela logika pametne kasice izvedena je pomoću *STM32* mikroupravljača koji je programiran pomoću besplatnog razvojnog okruženja Arduino. Mobilna aplikacija za pregled trenutnog stanja izrađena je pomoću platforme MIT App Inventor.

Ključne riječi: Android aplikacija, pametna kasicu, mikroupravljač, Arduino, STM32, IoT.

## **ABSTRACT**

### **MONEY BOX – COIN STORAGE AND SORTING SYSTEM**

The task of the final paper was to design, create and test a system for automated counting and sorting of coins, that is to make a smart money box. The system should have the ability to show the current state of the money box on the display and the mobile application. Furthermore, the paper explains the advantages and disadvantages of previous methods that were used for sorting and detecting coins, which is why it was decided to use the method that combines mechanical and electronic components. It is explained how to connect all of the components and tools that were used in order for the system to function. The entire logic of the money box is derived using an *STM32* microcontroller programmed using the free integrated development environment Arduino. The mobile application for viewing the current state was created using the Mit App Invertor platform.

Keywords: Android application, money box, Arduino, microcontroller, STM32, IoT.

## PRILOZI I DODACI

U ovom dijelu objasniti će se sve komponente korištene u izradi pametne kasice, struktura rada programa mikroupravljača a kao i programa mobilne aplikacije, te detaljnije prikazati dimenzije pločice od nehrđajućeg čelika sa otvorima zaduženima za detekciju kovanica.

Popis svih sklopovskih komponenti korištenih u ovom radu:

STM32F103C8TH mikroupravljač,

8x IR-HW-006 V1.3 senzora,

1x 16x2 LCD,

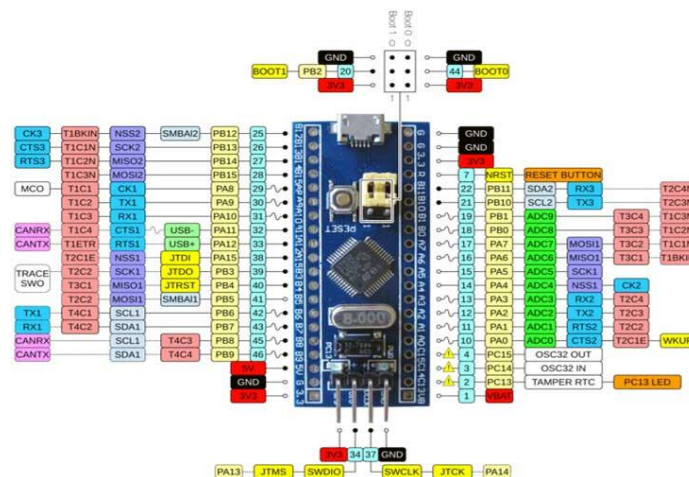
1x HC-06 bluetooth modul,

1x tipkalo,

1x ATC24C256 EEPROM memorija,

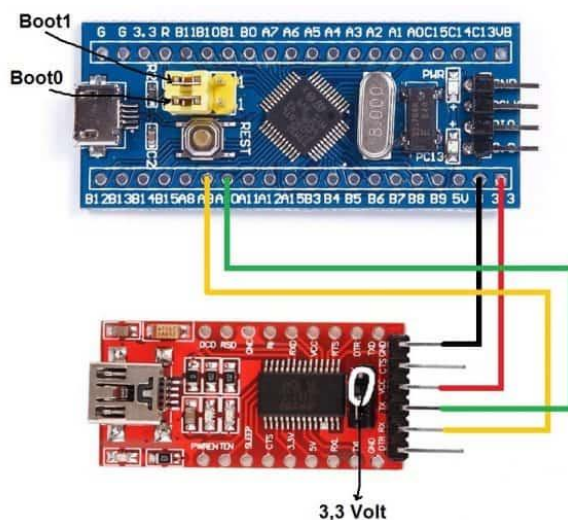
USB prema *UART* prilagodnik za programiranje mikroupravljača.

STM32F103C8T6 zaslužen je za upravljanje i napajanje svih nabrojenih komponenti u ovom radu. Posjeduje 15 analognih i 10 digitalnih pinova koji mogu služiti kao ulazi i izlazi, što je prikazano prema [12]. U ovom radu iskorišteno je 13 ulaza od kojih 8 koriste infracrveni senzori, 1 tipkalo, 2 hc-06 modula, i po 2 ista ulaza za LCD i EEPROM memoriju koji koriste *SDA* (engl. Serial Data, hrv. serijski podaci, u daljnjem tekstu korišten kao '*SDA*') i *SCL* (engl. Serial Clock, hrv. Serijski sat, u daljnjem tekstu korišten kao '*SCL*') pinove za *I2C* (engl. Inter-Integrated Circuit, hrv, Unutarnji-Integrirani krug, u daljnjem tekstu korišten kao '*I2C*') komunikaciju.



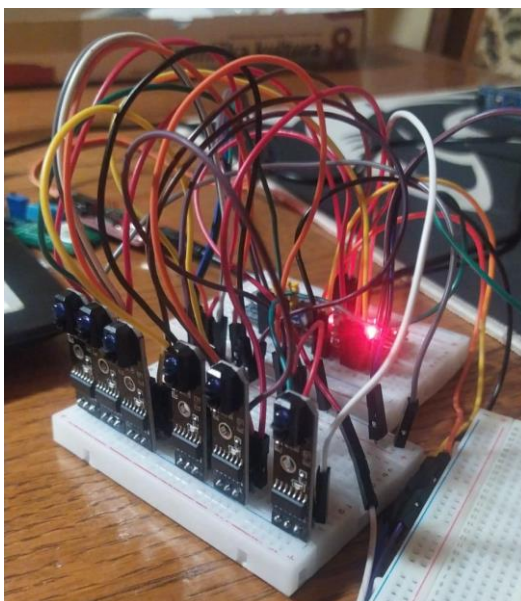
Slika 7.1. Raspored pinova mikroupravljača, [12].

Komunikacija između mikroupravljača i računala izvedena je putem *USB* prema *UART* serijskom prilagodniku gdje su pinovi Rx spojeni na PA09 i Tx na PA10 pinove mikroupravljača, što je objašnjeno prema [13]. Uređaj je bio zadužen za programiranje i praćenje serial monitora mikroupravljača.



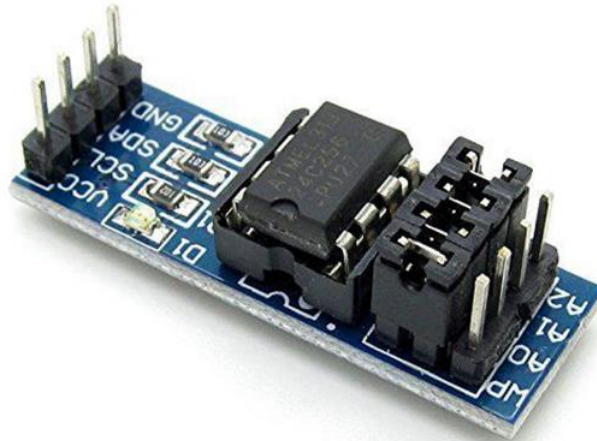
**Slika 7.2.** T232RL FTDI USB to TTL serial converter, [13].

Infracrveni senzori TCRT5000 poslužili su kao detekcija kovanice. Sam senzor posjeduje tri pina, od kojeg je jedan signal da se nešto nalazi ispred senzora, dok ostala dva služe kao napajanje. Senzor na sebi posjeduje emiter koji emitira infracrveno svjetlo, te čeka povratnu informaciju, ako se svjetlost reflektira od objekta koji se nalazi ispred njega on će to zabilježiti te poslati signal prema mikroupravljaču.



### Slika 7.3. Rad na sklopovlju.

Kao memorija za pohranu trenutnog stanja korištena je AT24C256 EEPROM koji ima kapacitet od 256 kilobita, odnosno 32 kilobajta za pohranu podataka. Koristi *I2C* komunikacijski protokol za komunikaciju s mikroupravljačem pa je iz tog razloga spojen na sda i scl pinove, kao što je opisano prema [14].



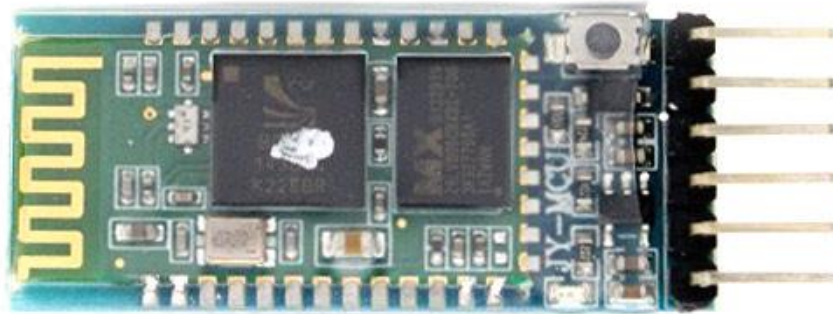
Slika 7.4. AT24C256 EEPROM memorija, [14]

*LCD* je zadužen za prikaz trenutnog stanja ubačenih kovanica, i prikaz posljednje ubačene kovanice. Sam zaslon sastoji se od 16 pinova, koji je kasnijim povezivanjem *I2C* komunikacijskim modulom, taj broj smanjen na 4 koja su povezana na mikroupravljač. Pin modula *SDA* spojen je na PB7 pin, a *SCL* je spojen na PB6 mikroupravljača, [15].



Slika 7.5. LCD 16x02, [15].

Bluetooth komunikacija uspostavljena s mobilnim uređaja omogućena je s HC-06 modulom. Koristi bluetooth 2.0 komunikacijski protokol i može se ponašati jedino kao *slave* uređaj. Radi na frekvencijskom opsegu od 2.402 GHz do 2.480 GHz te je pogodan za komunikaciju do 10 metara, što je opisano u [16].



**Slika 7.6.** HC-06 Bluetooth modul, [16].

U daljnjem dijelu prikazan je cijeli program pametne kasice, program je napisan i kompajliran u programu Arduino *IDE*, [9].

```
#include "Wire.h"
#include "EEPROM.h"
#define EEPROM_I2C_ADDRESS 0x50
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16 ,2);

int address = 0;
byte val= 0;
byte val_lipe = 0;
byte reSet = 0;
bool stanje_val;
int readVal=0;
int readVal_lipe = 0;
float posljedni_novcic;
char Incoming_value;
bool first_scan;
const unsigned long activationInterval = 1000;
unsigned long lastActivationTime = 0;
const unsigned long activationDelay = 5000;
volatile bool buttonPressed = false;
volatile unsigned long buttonPressStartTime = 0;
volatile bool functionActivated = false;
```

**Slika 7.7.** Prikaz biblioteka i korištenih varijabli.



U prvom dijelu programa uključene su biblioteke za *LCD* i EEPROM memoriju. Biblioteka 'Wire.h' potrebna je za *I2C* komunikaciju s EEPROM memorijom i *LCD*-om. U donjem dijelu koda nalaze se globalne varijable korištene u programu.

```
int euri;
int euri2;
int centi;
int centi2;
int euri_old = 1;
int centi_old = 1;
int brojac;
int brojac_old;
float euri_and_cent;
char s;
int i;
bool reload = 0;

byte customChar[8] = {
    0b00111,
    0b01000,
    0b10000,
    0b11111,
    0b10000,
    0b01000,
    0b00111,
    0b00000
};
```

**Slika 7.8.** Varijable stanja i ispisa znaka €.

S obzirom na to da znak '€' nije standardni *ASCII* (engl. American Standard Code for Information Interchange, hrv. Američki standardni znakovnik za razmijenu informacija) znak, sam ispis njega na zaslon dat će potpuno krivi rezultat. Stoga je stvorena varijabla bajt vrijednosti imena 'customChar' koja svojim ispisom čini znak '€'.

```

void setup() {
  Wire.begin();
  Serial.begin(9600);
  lcd.init();
  lcd.backlight();
  lcd.createChar(0, customChar); // create a new custom character

  pinMode(LED_BUILTIN, OUTPUT);
  pinMode(PA1, INPUT_PULLUP);
  attachInterrupt(PA1, dva_aura, RISING);
  pinMode(PA2, INPUT_PULLUP);
  attachInterrupt(PA2, jedan_euro, RISING);
  pinMode(PA3, INPUT_PULLUP);
  attachInterrupt(PA3, pedeset_cent, RISING);
  pinMode(PA4, INPUT_PULLUP);
  attachInterrupt(PA4, dvadeset_cent, RISING);
  pinMode(PA5, INPUT_PULLUP);
  attachInterrupt(PA5, deset_cent, RISING);
  pinMode(PA6, INPUT_PULLUP);
  attachInterrupt(PA6, pet_cent, RISING);
  pinMode(PA7, INPUT_PULLUP);
  pinMode(PA15, INPUT_PULLUP);
  attachInterrupt(PA15, jedan_cent, RISING);
  pinMode(PA0, INPUT_PULLUP);
  attachInterrupt(PA0, dva_centa, RISING);

}

```

**Slika 7.9.** void setup() funkcija.

U funkciji void setup() deklarirani su svi ulazi i izlazi u programu. Te također pokretanje *I2C* komunikacije i *LCD-a*. Pomoću naredbe 'attachInterrupt' se na svaki pozitivni brid, odnosno iz prelaska 0 u 1 poziva navedena funkcija. Na primjer ako je na ulaz mikroupravljača PA1 doveden signal, pozvat će se funkcija dva\_aura. Naredba 'Serial.begin()' potrebna je za slanje na bluetooth modul i njegovo određivanje brzine prijenosa kompatibilnog s hc-06 modulom.

```

void dva_eura(){
    unsigned long currentTime = millis();
    if(currentTime - lastActivationTime >= activationInterval){
        euri = euri + 2;
        writeEEPROM(4144,200,EEPROM_I2C_ADDRESS);
        delay(500);
        reload = false;
        lastActivationTime = currentTime;
    }
}

void jedan_euro(){
    unsigned long currentTime = millis();
    if(currentTime - lastActivationTime >= activationInterval){
        euri = euri + 1;
        writeEEPROM(4144,100,EEPROM_I2C_ADDRESS);
        delay(500);
        reload = false;
        lastActivationTime = currentTime;
    }
}

void pedeset_centri(){
    unsigned long currentTime = millis();
    if(currentTime - lastActivationTime >= activationInterval){
        centi = centi + 50;
        writeEEPROM(4144,50,EEPROM_I2C_ADDRESS);
        delay(500);
        reload = false;
        lastActivationTime = currentTime;
    }
}

```

**Slika 7.10.** Funkcije detekcije kovanica.

Prema slikama 7.10 , 7.11, 7.12 prikazane su funkcije koje se pozivaju na dolazak signala na pinove deklarirane u setup-u. U svakoj od funkcija se provjerava je li prošla jedna sekunda od prošlog pozivanja funkcije, ako da tek onda će se nova vrijednost zbrojiti u trenutno stanje kasice. Razlog provjere jedne sekunde je taj da se senzor ne bi aktivirao dva puta prilikom pada kovanice, što bi rezultiralo krivom vrijednošću.

```

void dva_centa(){
    unsigned long currentTime = millis();
    if(currentTime - lastActivationTime >= activationInterval){
        centi = centi + 2;
        writeEEPROM(4144,2,EEPROM_I2C_ADDRESS);
        delay(500);
        reload = false;
        lastActivationTime = currentTime;
    }
}

void jedan_cent(){
    unsigned long currentTime = millis();
    if(currentTime - lastActivationTime >= activationInterval){
        centi = centi + 1;
        posljedni_novcic = 0.01;
        writeEEPROM(4144,1,EEPROM_I2C_ADDRESS);
        delay(1000);
        reload = false;
        lastActivationTime = currentTime;
    }
}

```

**Slika 7.11.** Funkcije detekcije kovanica

```

void dvadeset_centi(){
    unsigned long currentTime = millis();
    if(currentTime - lastActivationTime >= activationInterval){
        centi = centi + 20;
        writeEEPROM(4144,20,EEPROM_I2C_ADDRESS);
        delay(500);
        reload = false;
        lastActivationTime = currentTime;
    }
}

void deset_centi(){
    unsigned long currentTime = millis();
    if(currentTime - lastActivationTime >= activationInterval){
        centi = centi + 10;
        writeEEPROM(4144,10,EEPROM_I2C_ADDRESS);
        delay(500);
        reload = false;
        lastActivationTime = currentTime;
    }
}

void pet_centi(){
    unsigned long currentTime = millis();
    if(currentTime - lastActivationTime >= activationInterval){
        centi = centi + 5;
        writeEEPROM(4144,5,EEPROM_I2C_ADDRESS);
        delay(500);
        reload = false;
        lastActivationTime = currentTime;
    }
}

```

**Slika 7.12.** Funkcije detekcije kovanica

```

void writeEEPROM(int address, byte val, int i2c_address){
    Wire.beginTransmission(i2c_address);
    Wire.write((int)(address >>8));
    Wire.write((int)(address & 0xFF));
    Wire.write(val);
    Wire.endTransmission();
    delay(200);
}

byte readEEPROM(int address, int i2c_address){
    byte rcvData = 0xFF;
    Wire.beginTransmission(i2c_address);
    Wire.write((int)(address >>8));
    Wire.write((int)(address & 0xFF));
    Wire.endTransmission();
    Wire.requestFrom(i2c_address, 1);
    rcvData = Wire.read();
    return rcvData;
    delay(200);
}

```

**Slika 7.13.** Funkcije pisanja i čitanja iz EEPROM memorije, [14].

Sljedeće dvije funkcije prikazane na slici 7.13 zadužene su za pisanje i čitanje iz EEPROM memorije. Zahtijevaju adresu na koju će pisati unutar EEPROM memorije, bajt vrijednost koja će se upisati te adresu EEPROM-a koja mora biti unikatna, u našem slučaju različita od adrese *LCD*-a.

```

void loop() {
  if (first_scan == false){
    euri_old = euri;
    centi_old = centi;
    brojac_old = brojac;
    first_scan = true;
  }

  if (centi != centi_old){
    if (centi > 99){
      centi2 = centi -100;
      centi = centi2;
      euri = euri +1;
      delay(25);
    }

    writeEEPROM(4142,centi,EEPROM_I2C_ADDRESS);
    centi_old = centi;
    reload = false;
  }

  if (euri != euri_old){
    if ( euri > 200) {
      euri2 = euri - 200;
      brojac = brojac + 1;
      writeEEPROM(4149,brojac,EEPROM_I2C_ADDRESS);
      euri = euri2;
      delay(25);
    }

    writeEEPROM(4141,euri,EEPROM_I2C_ADDRESS);
    euri_old = euri;
    reload = false;
    delay(25);
  }

  if(brojac != brojac_old) {
    writeEEPROM(4149,brojac,EEPROM_I2C_ADDRESS);
    delay(30);
    brojac_old = brojac;
    reload = false;
  }
}

```

**Slika 7.14.** Prikaz petlje programa.

U daljnjem dijelu ulazimo u 'loop' funkciju u kojoj se vrti cijeli program.

Na samom početku se provjerava je li program tek započeo, ako je, u varijable sa završetkom 'old' u imenu se upisuju vrijednosti zapisane iz memorije, a koje će se dalje koristiti kao provjera dali je došlo do promjene trenutnih vrijednosti. Kako bi program bio što manje zasićen, ali i zbog

ograničenosti broja pisanja u memoriju, upisivanje se vrši tek kad je došlo do neke promjene, što je vidljivo prema 'if' petljama.

```
if (reload == false){
    euri = byte(readEEPROM(4141, EEPROM_I2C_ADDRESS));
    centi = byte(readEEPROM(4142, EEPROM_I2C_ADDRESS));
    brojac = byte(readEEPROM(4149, EEPROM_I2C_ADDRESS));
    euri_and_centni = float(euri) + float(float(centi)/100) + (brojac * 200);
    float posljedni_novcic2 = byte(readEEPROM(4144, EEPROM_I2C_ADDRESS));
    posljedni_novcic = posljedni_novcic2 / 100;
    lcd.clear();
    lcd.setCursor( 0 , 0 );
    lcd.print("Stanje: ");
    lcd.print(euri_and_centni);
    lcd.write((byte)0);
    lcd.setCursor( 0 , 1 );
    lcd.print("Posljedni: ");
    lcd.print(posljedni_novcic);
    lcd.write((byte)0);
    reload = true;
    delay(100);
}
```

**Slika 7.15.** Čitanje iz memorije.

Vrijednost reload varijable postaje negativna samo kad dođe do neke promjene vrijednosti. Ta se vrijednost upisuje u memoriju. Ovom petljom te se vrijednosti odmah i čitaju iz memorije kako bi program bio konstantno ažuriran. U isto vrijeme se odmah vrijednost ispisuje na *LCD*. Primjer programiranja *LCD-a* možemo vidjeti prema [15]. Sljedeće čitanje iz memorije i ispisivanje na *LCD* dogodit će se tek kad varijabla reload opet bude pozitivna, odnosno kad dođe do neke promjene vrijednosti.

```

if (Serial.available()>0){
  Incoming_value = Serial.read();
  Serial.println(Incoming_value);
  writeEEPROM(4141,0,EEPROM_I2C_ADDRESS);
  writeEEPROM(4142,0,EEPROM_I2C_ADDRESS);
  writeEEPROM(4149,0,EEPROM_I2C_ADDRESS);
  writeEEPROM(4144,0,EEPROM_I2C_ADDRESS);
  reload = false;
}

if(digitalRead(PA7)){
  if(buttonPressStartTime == 999999){
    buttonPressStartTime = millis();
  }
  buttonPressed = true;
if (buttonPressed && millis() - buttonPressStartTime >= activationDelay ){
  brojac = 0;
  euri = 0;
  centi = 0;
  writeEEPROM(4144,0,EEPROM_I2C_ADDRESS);
  delay(30);
  buttonPressStartTime = millis();
}

}else {
  buttonPressed = false;
  buttonPressStartTime = 999999;
}
}

```

**Slika 7.16.** Provjera zahtjeva resetiranja trenutnog stanja .

U ovom dijelu koda se provjeravaju zahtjevi za resetiranjem trenutnog stanja kasice. Prva petlja provjerava dolaze li koje vrijednosti s povezanog mobilnog uređaja na 'Serial Monitor', dok druga petlja provjerava je li tipka za resetiranje trenutnog stanja na pametnoj kasici pritisnuta duže od 5 sekundi. Obje petlje za pozitivan rezultat daju resetiranje svih vrijednosti na 0.

```

Serial.print(euri_and_cent);
Serial.print(" €");
Serial.print("");
Serial.print(posljedni_novcic);
Serial.println(" €");
delay(100);

```

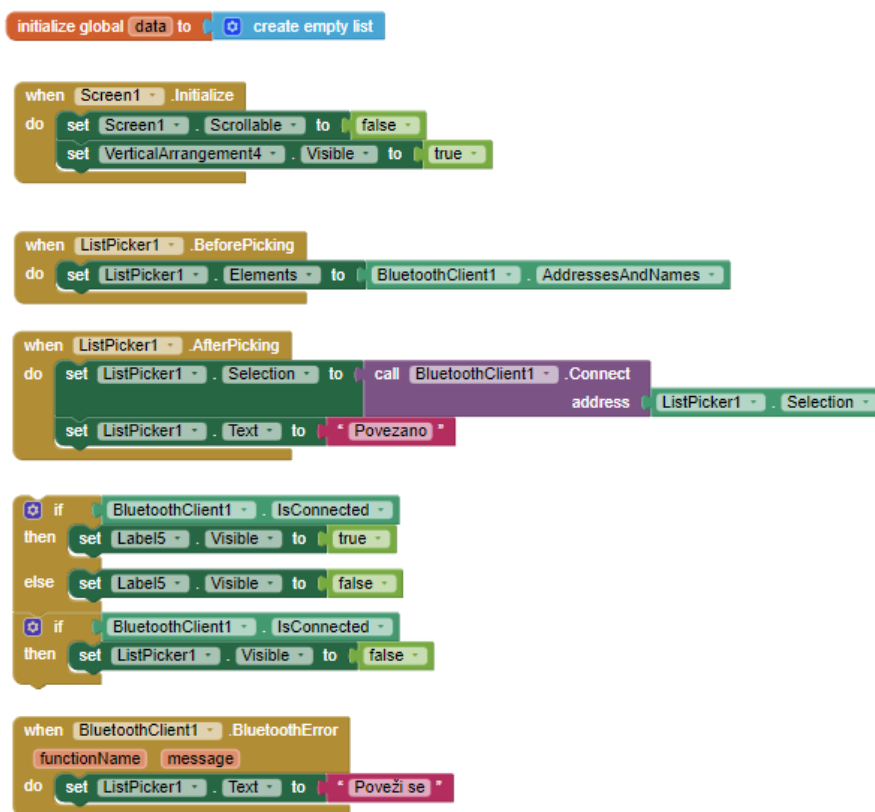
**Slika 7.17.** Slanje vrijednosti na bluetooth modul.



Zadnji dio koda zadužen je za ispis vrijednosti trenutnog stanja i vrijednosti posljednjeg novčića na 'Serial Monitor', ta se vrijednost šalje na mobilni uređaj i dalje obrađuje.

Dizajniranje i programiranje mobilne aplikacije izvršeno je preko sustava Mit App Inventor koji omogućava jednostavno stvaranje Android aplikacija bez potrebe za razumijevanjem programiranja, [9]. Cijeli vizualni izgled dizajnira se povlačenjem već gotovih modela na zaslone aplikacije. Nakon dodavanja svih potrebnih elemenata prelazi se na blokovsko programiranje, gdje umjesto koda, slaganjem blokova određujemo što će se događati u aplikaciji.

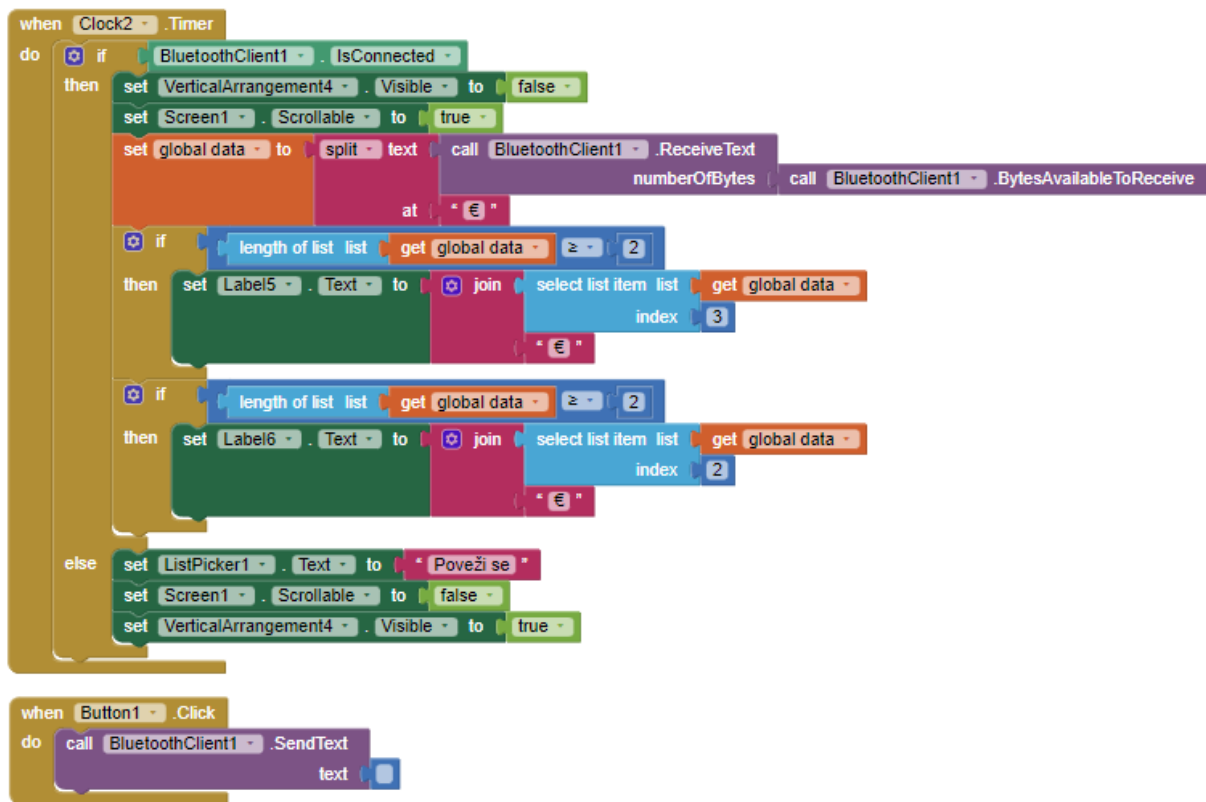
Prema slici 7.18 možemo vidjeti programsku logiku mobilne aplikacije.



Slika 7.18. Inicijalizacija varijabli i bluetooth klijenta [9].

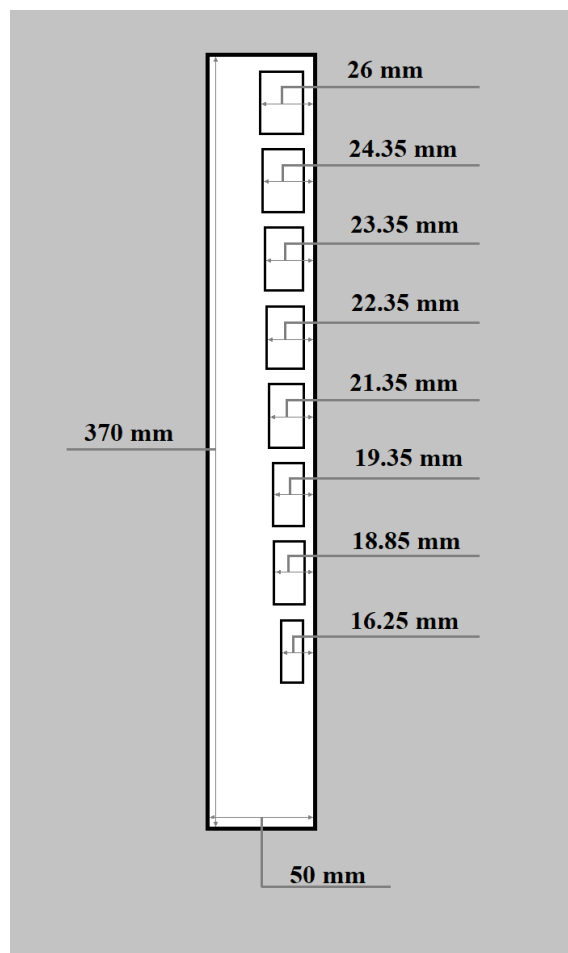
U početnom dijelu stvara se varijabla 'data' u koju su kasnije upisane vrijednosti primljene preko bluetooth modula. Elementi kao što su 'Scrollable' i 'Visible' utječu na izgled početnog zaslona, pa će tako ako bluetooth uređaj nije povezan, na početnom zaslonu biti prikazana samo temeljna slika pametne kasice i tipka 'poveži se'. Tek kad se uređaj uspješno poveže, slika postaje nevidljiva i na

pozadinu na prvo mjesto dolazi ispis posljednje ubačene kovanice i trenutnog stanja što možemo vidjeti prema slici 3.12



Slika 7.19. Primanje podataka s bluetooth modula, [9].

Glavnu petlju izvršava vremenska varijabla 'Clock2' koja ima početno vrijeme od jedne sekunde. Na svakom završetku odbrojavanja jedne sekunde provjerava se je li bluetooth uređaj povezan. Ako je bluetooth uređaj povezan, primljeni tekst dijeli se na dvije vrijednosti i upisuje u globalnu varijablu 'data' gdje prva vrijednost čini trenutno stanje, a druga posljednju ubačenu kovanicu. Ova metoda primanja i podjele podataka nastala je međusobnom povezanošću i kompatibilnosti programa mikroupravljača. Na samom kraju se provjerava je li tipka 'Resetiraj vrijednost' pritisnuta. Ako je pritisnuta, šalje se prazna vrijednost na bluetooth modul. Razlog slanja prazne vrijednosti je taj što prilikom slanja neke druge vrijednosti, nastaju smetnje prilikom primanja naših vrijednosti, pa se dogodi da ta vrijednost koja se poslala na pritisak tipke, bude ispisana na početnom zaslonu aplikacije na poziciji trenutnog stanja ili posljednje kovanice. Struktura programa mikroupravljača zato je napravljena da provjerava je li poslana bilo kakva vrijednost, stoga nije potrebno provjeravati koja je točno vrijednost poslana.



**Slika 7.20.** Nacrt pločice sa otvorima

Prema slici 7.20 možemo detaljnije vidjeti dimenzije pločice od nehrđajućeg čelika. Schema iznad služila je kao podloga prema kojoj je kasnije pomoću *CNC* stroja pločica i izrezana. Širina pločice iznosi 5 cm, a dužina 37 cm. Od početka desne strane pločice do svakog otvora dijeli 0.5 cm po kojemu su kovanice klizile prema dolje. Prema tablici 7.1 možemo vidjeti usporedbe promjera kovanica i udaljenosti od početka desnog brida do kraja otvora pojedine kovanice.

Kovanica	0.01 €	0.02 €	0.05 €	0.10 €	0.20 €	0.50 €	2.00 €	2.00 €
Promjer kovanice (mm)	16.25	18.75	21.25	19.75	22.25	24.25	23.25	25.75
Udaljenost od desnog brida do kraja otvora (mm)	16.25	18.85	19.35	21.35	22.35	23.35	24.35	26

**Tablica 7.1.** Usporedba dimenzija kovanica i otvora pločice.