

# Pametna kanta za smeće

---

**Kunce, Daniel**

**Undergraduate thesis / Završni rad**

**2021**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **Istrian University of applied sciences / Istarsko veleučilište - Università Istriana di scienze applicate**

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:212:119242>

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

*Download date / Datum preuzimanja:* **2024-05-19**



*Repository / Repozitorij:*

[Digital repository of Istrian University of applied sciences](#)



image not found or type unknown

ISTARSKO VELEUČILIŠTE –  
UNIVERSITÁ ISTRIANA DI SCIENZE APPLICATE

Daniel Kunce

## **Pametna kanta za smeće**

Završni rad

Pula, 2021

ISTARSKO VELEUČILIŠTE –  
UNIVERSITÁ ISTRIANA DI SCIENZE APPLICATE

Daniel Kunce

**Pametna kanta za smeće**

Završni rad

JMBAG: 0233006990, redoviti student

Studijski smjer: Preddiplomski stručni studij Mehatronike

Predmet: Završni rad

Mentor: Marko Turk

Sumentor: ---

Pula, 2021



## IZJAVA O AKADEMSKOJ ČESTITOSTI

Ja, dolje potpisani Daniel Kunce, kandidat za prvostupnika mehatronike ovime izjavljujem da je ovaj Završni rad rezultat isključivo mogega vlastitog rada, da se temelji na mojim istraživanjima te da se oslanja na objavljenu literaturu kao što to pokazuju korištene bilješke i bibliografija. Izjavljujem da niti jedan dio Završnog rada nije napisan na nedozvoljen način, odnosno da je prepisan iz kojega necitiranog rada, te da ikoji dio rada krši bilo čija autorska prava. Izjavljujem, također, da nijedan dio rada nije iskorišten za koji drugi rad pri bilo kojoj drugoj visokoškolskoj, znanstvenoj ili radnoj ustanovi.

U Puli, 2021. godine

Student

Daniel Kunce



IZJAVA  
o korištenju autorskog djela

Ja, Daniel Kunce dajem odobrenje Istarskom veleučilištu –  
Università Istriana di scienze applicate, kao nositelju prava iskorištavanja, da moj završni rad  
pod nazivom

"Pametna kanta za smeće"

koristi na način da gore navedeno autorsko djelo, kao cjeloviti tekst trajno objavi u javnoj  
internetskoj bazi Sveučilišne knjižnice u Puli te kopira u javnu internetsku bazu završnih  
radova Nacionalne i sveučilišne knjižnice (stavljanje na raspolaganje javnosti), sve u skladu s  
Zakonom o autorskom pravu i drugim srodnim pravima i dobrom akademskom praksom, a  
radi promicanja otvorenoga, slobodnoga pristupa znanstvenim informacijama.

Za korištenje autorskog djela na gore navedeni način ne potražujem naknadu.

U Puli, 2021. godine

Potpis

Daniel Kunce

## SADRŽAJ

1. Uvod.....	1
2. Upravljanje otpadom i budućnost razvoja gospodarenja otpadom .....	3
2.1. Generiranje otpada i suvremeno gospodarenje otpadom .....	3
2.2. Implementacija IoT i pametne tehnologije.....	6
2.3. Pametna kanta za smeće.....	9
2.3.1. Ideja i koncept.....	9
2.3.2. Mikrokontroler .....	10
2.3.3. Senzori i aktuatori .....	10
2.3.4. Komunikacija .....	11
3. Komponente i hardver .....	12
3.1. Grafička ilustracija i shematski prikaz sklopa .....	13
3.2. Arduino.....	15
3.3. Ultrazvučni senzori.....	16
3.4. Senzor opterećenja i analogno digitalni pretvornik HX711 .....	18
3.5. Radio-frekvencijski čitač kartica .....	20
3.6. Servo motor .....	22
3.7. LCD Zaslona .....	23
3.8. Ostale komponente .....	24
3.8.1. Piezoelektrična zupčica .....	24
3.8.2. Otpornik i potencijometar.....	24
3.8.3. Vodiči i eksperimentalna kartica .....	24
4. Programsko rješenje zadatka .....	25
4.1. Softverski paketi.....	25
4.1.1. TinkerCAD .....	25
4.1.2. Fritzing.....	25
4.1.3. Arduino .....	26

4.2. Postupci stvaranja softverskog rješenja .....	27
4.3. Pomoćni program za HX711 .....	38
4.4. Pomoćni program za MFRC-522 .....	40
5. 3D model i izrada prototipa .....	41
6. Zaključak .....	45
7. Popis Literature .....	46
7.1. Knjige i članci .....	46
7.2. Programski paketi .....	46
7.3. Internet izvori .....	46

Popis Tablica

Popis Slika

DODATAK

## 1. UVOD

U suvremenome društvu mnogo pojedinaca, rijetko se osvrće i priupita se na koje načine sve utječu na prirodu i okoliš koji nas okružuje. Naviknuti smo na načine života gdje je sve brzo i dostupno, nadohvat klika miša ili pritiskom na ekran.

Ovaj završni rad okrenuti će se ka drugome smjeru, osvrnuti će se na probleme ekologije u urbanim regijama, te u već poznata rješenja donijeti razinu automatizacije i autonomije.

Kroz sljedeće dijelove rada, promotrit će se suvremeni načini upravljanja otpadom, načini na koje se može djelovati i doprinijeti daljnjem razvoju.

Cilj rada je stvoriti koncept sustava pametne kante za smeće koja bi bila pozicionirana u urbanim središtima te pružila ljubaznost korisnicima dok bi istovremeno organizacijama zaduženima za zbrinjavanje okoliša, olakšala proces sakupljanja i sortiranja otpadnih materijala unutar urbanih regija.

Pametna kanta za smeće bazira se na modelu „*zelenih otoka*“ koji već čini dijelove naših gradova i urbanih regija. Zeleni otoci<sup>1</sup> su prema definiciji, mjesto predodređeno za posebne spremnike namijenjene za sakupljanje zasebnih kategorija otpada. Njima upravlja komunalna tvrtka kojoj je djelatnost gospodarenje i upravljanje otpadom na razini lokalne ili regionalne samouprave.

Model zelenih otoka prikazao se kao dobar način sortiranja otpada te je vrijedan izvor sekundarnih sirovina u reciklažnoj industriji.

Sam model zelenih otoka nije savršen, te postoji mjesto za nadogradnju modela. Mjesto za proširenje nalazimo u IoT (engl. *Internet of Things* – *Internet Stvari*) tehnologijama koje su sve dominantnije u našim okruženjima. Spajanjem tih dvaju koncepta ciljalo bi se na poboljšanje logistike pojedinih organizacija za gospodarenje otpadom, podižući njihovu efikasnost i omogućuje organizaciji prikupljanje raznih podataka o sakupljenom otpadu, npr. količini i volumenu otpada na pojedinoj lokaciji, vrsti otpada koju korisnik generira i sl.

---

<sup>1</sup> Branimir Fuk - Oprema i infrastruktura za održivo gospodarenje otpadom u jedinicama lokalne/regionalne samouprave (I.dio)



Model pametne kante za smeće trebao bi objediniti ideju o recikliranju, poboljšanoj logistici, lakom korištenju i u konačnici boljim gospodarenjem otpadom.

Osim teorijskog dijela gdje će se govoriti o dosadašnjem razvoju i idealnom konceptu pametne kante za smeće i njezinoj implementaciji s IoT tehnologijama, rad će imati fizički koncept pametne kante za smeće koji će biti osmišljen za samo mjerenje i evaluaciju pohranjenog otpada i interakciju s korisnikom te neće zadirati u fizičku implementaciju IoT tehnologije koja se spominje u daljnjoj razradi.

Pametna kanta za smeće realizirat će se u fizičkom dijelu rada, te ona u konačnici stvara samo dio IoT sustava za gospodarenjem otpadom.

Unaprjeđivanje metoda gospodarenja otpadom postala je potreba, kako bismo očuvali našu prirodu kojoj prijete sve brže i rastuće konzumerističko društvo.

*„We cannot solve our problems with the same thinking we used when we created them“ – Albert Einstein*

## **2. UPRAVLJANJE OTPADOM I BUDUĆNOST RAZVOJA GOSPODARENJA OTPADOM**

### **2.1. Generiranje otpada i suvremeno gospodarenje otpadom**

Uvidom u statističke podatke Republike Hrvatske prema kojima je donesen plan za gospodarenje otpadom<sup>2</sup>, 2017. godine, vidljivo je da je količina komunalnog i proizvodnog otpada porasla za 10,5% u odnosu na prethodne godine. Tu količinu otpada čini 97% neopasnog otpada dok je ostalih 3% opasan otpad.

Najveći udio otpada nastaje u kućanstvima (31%), te taj otpad čine razne vrste otpada koji proizvode građani. Ostali dio otpada nastaje u raznim sektorima. Sektor uslužnih, tj. tercijarnih djelatnosti (trgovina, ugostiteljstvo, turizam) zajedno sa sektorom građevinarstva (pretežito građevinski otpad) stvaraju udio od 17%. Sektor prerađivačke industrije generira otpad u udjelu od 12%, te kao treći najveći čimbenik s udjelom od 11% čine djelatnosti sakupljanja, zbrinjavanja otpada, obrade otpada i ponovne uporabe materijala.

Ostatak gospodarske djelatnosti imaju udio od 12% i manje. Valja napomenuti da mnogo sektora raspolaže s podosta malim brojem ili netočnim brojem podataka (ponajviše sektor poljoprivrede i građevinarstva).

Porastom razine industrijalizacije, veliki dio organizacija usvaja niz normi ISO 14000<sup>3</sup> koje se odnose na standarde za poboljšanje upravljanja okolišem i umanjivanje negativnog utjecaja industrije na okolinu. Samim time se obvezuju na očuvanje okoliša i održivi razvoj. Time se može uvidjeti da je problem gospodarenje otpada u industriji i sektorima privrede poprilično uređen, dok se isto ne može reći za gospodarenjem komunalnim otpadom generiranim od strane građana.

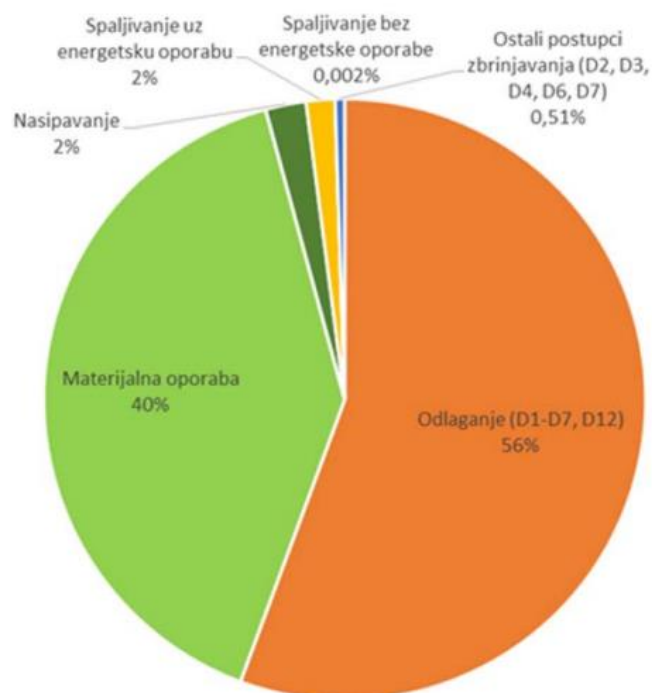
U planu gospodarenjem otpadom prikazane su metode s kojima se otpad proizveden u RH zbrinjava ili ponovno upotrebljuje, vidljivo na slici 1.

Kako bi se stopa materijalne uporabe povećala, a odlaganje smanjilo, važno je prepoznati na koje načine možemo djelovati. Miješani komunalni otpad u RH čine razne vrste otpada, uključujući staklo, plastiku, kuhinjski otpad i ostalo, vidljivo u tablici 1.

---

<sup>2</sup> Plan Gospodarenja Otpadom Republike Hrvatske

<sup>3</sup> ISO 14000

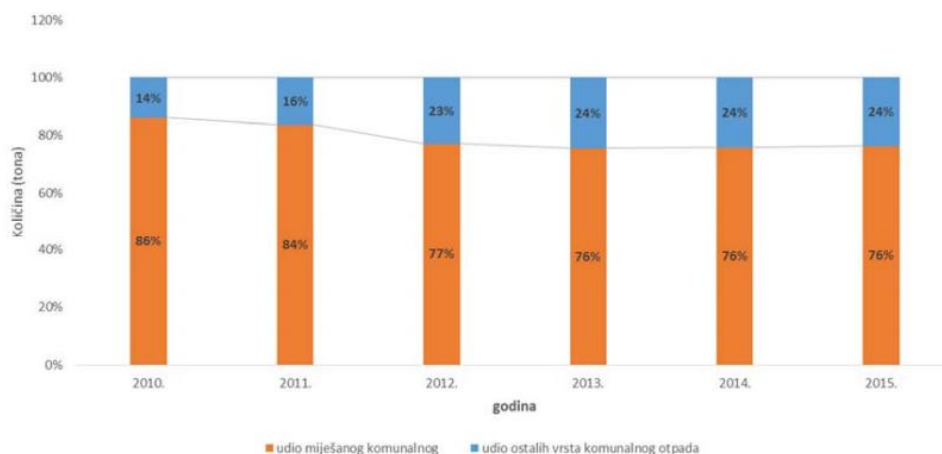


Slika 1. Postupci korišteni za oporabu i zbrinjavanje ukupnog otpada u području RH prema podacima iz 2014. godine (HAOP, 2016.)

Sastavnica	Udio (%)
Metal	2,1
Drvo	1,0
Tekstil/odjeća	3,7
Papir i karton	23,2
Staklo	3,7
Plastika	22,9
Guma	0,2
Koža/kosti	0,5
Kuhinjski otpad	30,9
Vrtni otpad	5,7
Ostali otpad (zemlja, prašina, pijesak, nedefinirano)	6,3
<b>Ukupno</b>	<b>100</b>

Tablica 1. Miješani komunalni otpad u RH prema podacima iz 2015. godine

Na miješani komunalni otpad da se djelovati na načine da se potakne razdvajanje pojedinih materijala poput papira, plastike, stakla, metala i slično. Takav pristup prikazao je rezultate te je od ukupne količine komunalnog otpada 2012. ostalo 86% miješanog otpada. Uz poticanje razvrstavanja, do 2015. godine, razina miješanog otpada opala je na 76%, vidljivo na slici 2.



Slika 2. Prikaz udjela odvojenog otpada i miješanog sakupljenog komunalnog otpada u RH u razdoblju od 2010. do 2015. (HAOP, 2016.)

Na količine otpada koji je sakupljenog odvojeno od miješanog komunalnog otpada veliki utjecaj imaju zeleni otoci i reciklažna dvorišta. Prema definiciji to prostori su ograđeni i nadzirani te im je namjena prikupljanje otpada, razvrstavanje i privremeno skladištenje određene količina zaprimljenog otpada sve dok taj razvrstani i pohranjeni otpad ne bude transportiran ka nekom regionalnom ili lokalnom centru za gospodarenje otpadom, na daljnju obradu ili zbrinjavanje<sup>4</sup>.

Također, odvajanje manjih količina otpada provodi se kroz odvajanje na komunalnom pragu tako da komunalna tvrtka koja vodi operacije u toj lokaciji, korisnicima dodjeljuje spremnike za odvajanje otpada kako bi se otpad odvajao odmah u trenutku kad je proizveden. Prethodno spomenuti način gospodarenja otpadom djelotvoran je, ali nije pogodan za urbane regije, već mu je najbolja namjena u manjim naseljima gdje su obiteljske kuće ili zgrade s manjim brojem stanovnika.

U urbanim sredinama, gradovima i gusto naseljenim kvartovima, komunalne tvrtke se određuju za instalaciju i izgradnju zelenih otoka. Korak kako bi se sustav zelenih otoka

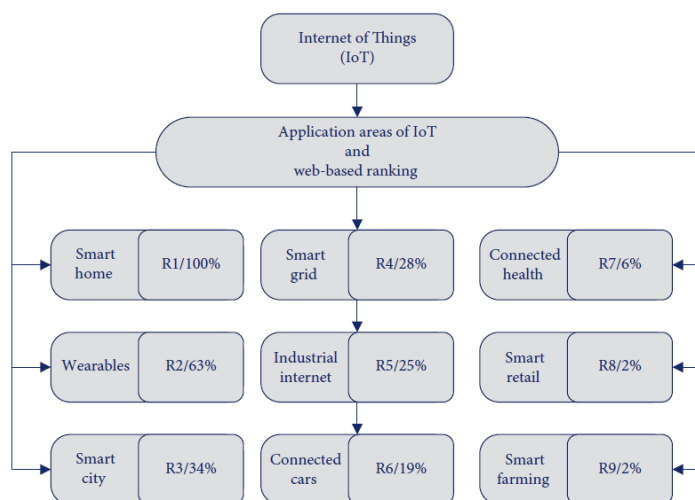
<sup>4</sup> Branimir Fuk - Oprema i infrastruktura za održivo gospodarenje otpadom u jedinicama lokalne/regionalne samouprave (I.dio)

unaprijedio i komunalnim tvrtkama pružio veću količinu informacija je upravo implementacija IoT tehnologija koja bi taj sustav sakupljanja učinio „*Pametnim*“.

## 2.2. Implementacija IoT i pametne tehnologije

Prethodno spomenuto u prošleme poglavlju, današnji gradovi suočavaju se s problemima gospodarenjem otpadom, parkirnim prostorima i općom komunikacijom među određenim podsustavima. Sve to u konačnici rezultira na ljudsku rutinu i živote ljudi u afektiranoj okolini. Kako bi se pružilo bolje rješenje, pojavio se koncept pametnog grada<sup>5</sup> koji bi inkorporirao IoT tehnologiju.

Zadnjih nekoliko godina, istraživači i razvojni timovi fokusiraju se na razvoj i primjenu IoT tehnologija kako bi se u konačnici razvio koncept pametnog grada (eng. *Smart City*). Takav grad bi posjedovao pametnu infrastrukturu koja bi imala mogućnost komunikacije i interakcije sa svakim međusobnim elementom. Takav grad bi vodio do razvoja pametnog očuvanja okoliša, pametnog zdravstva, pametnog parkinga, ekonomije, administracije itd.. Na slici 3 prikazana su dosadašnja korištenja IoT tehnologija prema internet istraživanju.



Slika 3. Polja korištenja IoT tehnologije u konceptu pametnog grada (Aniqa Bano, Ikram Ud Din, Asma A. Al-Huqauil, 2020.)

<sup>5</sup> AioT-Based Smart Bin for Real-Time Monitoring and Management of Solid Waste - Aniqa Bano, Ikram Ud Din, Asma A. Al-Huqauil, Department of Information Technology, The University of Haripur, Pakistan, 2020

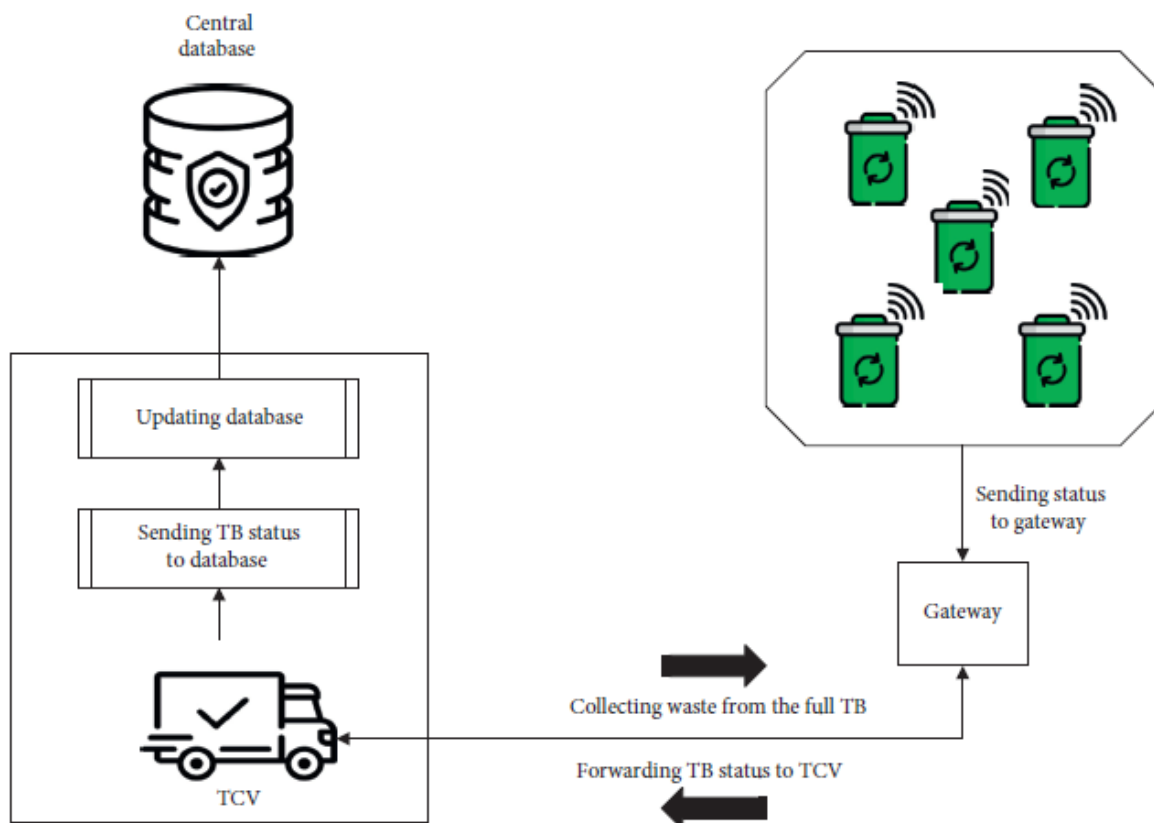
Ovdje u radu bavimo se samo dijelom koncepta pametnog grada, a to je pametno upravljanje otpadom Internet stvarima<sup>6</sup> (eng. *IoT based solid waste management*). Propozicija implementiranja IoT sustava u ovaj sektor bazira se na razvoju komunikacije između pojedinih dijelova sustava za gospodarenje otpadom (slika 4). Cilj bi bio povezati vozila za odvoz otpada i omogućiti im planiranu i stratešku navigaciju prema točno određenim spremnicima. Spremnici bi također bili umreženi u tu cjelinu te sustavu za gospodarenje otpadom davali informacije o stanju pojedinih spremnika. Tako bi se u samome startu povećala efikasnost sakupljanja otpada uz korištenje planiranih ruta za vozila za sakupljanje otpada. Spremnici za pohranu otpada koji bi se nalazili kod korisnika ili u gradovima u obliku zelenih otoka, također bi trebali posjedovati određenu razinu automatizacije, mikro upravljač, komunikaciju i senzore za mjerenje.



Slika 4. Arhitektura IoT sustava za gospodarenje otpadom (Aniqa Bano, Ikram Ud Din, Asma A. Al-Huqaquil , 2020.)

<sup>6</sup> AioT-Based Smart Bin for Real-Time Monitoring and Management of Solid Waste - Aniqa Bano, Ikram Ud Din, Asma A. Al-Huqaquil, Departmet of Information Technology, The University of Haripur, Pakistan, 2020.

Sustav bi trebao biti centraliziran, imati svoju bazu podataka i zadane parametre prema kojima će se dobavljeni podaci procesuirati (slika 5). Komunikacija između centralnog sustava i periferija mogao bi biti ostvaren na više načina, uporabom bežičnih tehnologija, GSM, WI-FI, LoRaWan ili žičanom komunikacijom(Ethernet protokolom), npr. PROFINET, preko koje bi taj uređaj kroz siguran komunikacijski kanal imao komunikaciju s centralnim sustavom.

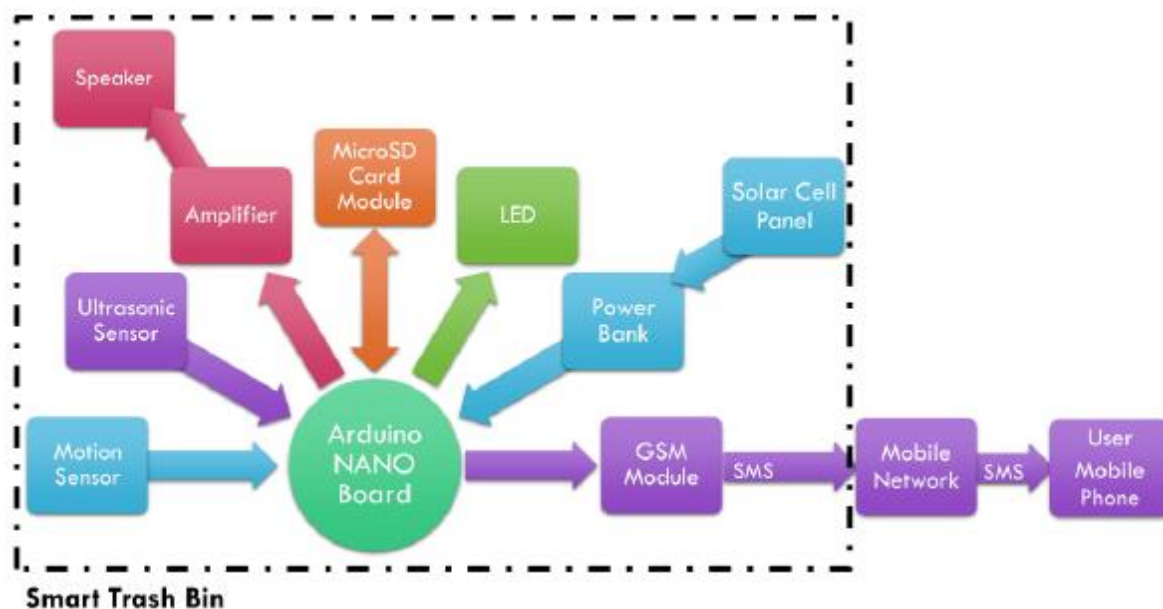


Slika 5. Primjer komunikacije između pametnih kanti i centralnog sustava (Aniqa Bano, Ikram Ud Din, Asma A. Al-Huqail , 2020.)

## 2.3. Pametna kanta za smeće

### 2.3.1. Ideja i koncept

Princip pametne kante za smeće bazira se umrežavanju svih manjih potrebnih podsustava u jednu cjelinu s programabilnim mikro upravljačem dajemo svrhu i zadatak tome uređaju. Pametna kanta za smeće kao upravljačku osnovu ima mikro upravljač, koji može biti napajan obnovljivim izvorima energije ili standardnim metodama napajanja. Na sam upravljač se zatim spajaju ulazne komponente koje omogućuju upravljaču da ima „pogled“ u svoje okruženje. Spajaju se aktuatori koji služe za komunikaciju s korisnikom i vršenje ostalih radnji, ovisno o programu i dizajnu te neku vrstu komunikacije koja je predodređena za taj sustav<sup>7</sup> (slika 6).



Slika 6: Blok dijagram osnovnih dijelova i podsustava pametne kante za smeće (Fady Samann, 2017)

Takav uređaj upravlja procesom gdje bi korisnik prišao uređaju, očitao svoj jedinstveni identifikacijski kôd koji bi sustav prepoznao. Shodno tome sustav bi otvorio pretinac gdje bi korisnik imao određeni vremenski period naznačen na zaslonu kako bi pohranio otpad. Kroz par trenutaka pretinac bi se zatvarao te bi pohrambeni prostor sustava bio mjeran i prikazivao stanje na svojem zaslonu, a također bi mrežnim tehnologijama podatke slao u operativni centar. Otvaranje i zatvaranja spremnika bili bi popraćeni zvučnim i vizualnim signalima. Ovisno o stanju pohrambenog prostora stanje bi bilo dostupno na uvid korisniku a više informacija bi bile dostupne

<sup>7</sup> *The Design and Implementation of Smart Trash Bin* - Fady Samann, Duhok Polytechnic University, Iraq, Duhok, 2017.



administratorima i operaterima sustava. Ukoliko bi pohrambeni prostor sustava bio popunjen po volumenu ili previše opterećen masom, potrebno je spriječiti daljnju mogućnost da korisnik/korisnici koristi sustav te naznačiti da je spremnik pun. Više o samim komponentama i spajanju fizičkih dijelova u jednu cjelinu u poglavlju 3.

### 2.3.2. Mikrokontroler

Pri odabiru mikrokontrolera za ovakav uređaj valja obratiti pozornost na okruženje u kojem se nalazi te odabrati sustav koji je robustan i pouzdan. Uobičajeno se u industrijskom okruženju koriste PLC uređaji koji su namijenjeni za rad u industriji te su upravo stvoreni za pouzdan i robustan način rada. PLC bi bio najpouzdaniji uređaj, ali ne nužno i optimalan zbog svoje cijene i održavanja. U samome nastavku rada, u poglavlju 3, za dokazivanje koncepta koristit će se Arduino mikrokontroler radi lakše izvedbe koncepta i cijene. Ovisno o dizajnu i potrebama uređaja u krajnjem rješenju bilo bi poželjno odabrati mikrokontroler koji bi odgovarao namjeni i uvjetima.

### 2.3.3. Senzori i aktuatori

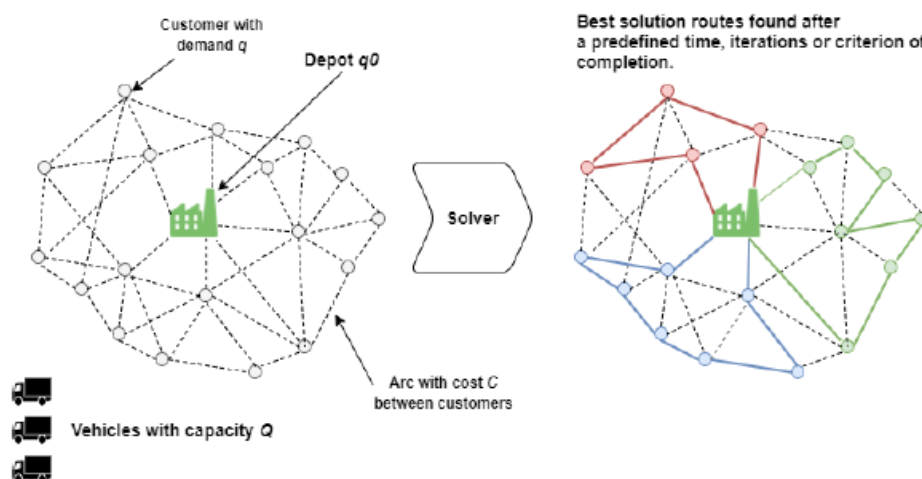
Senzori pri ovakvom sustavu mogu varirati, ovisno o vrsti podatka i informacije u koji želimo imati uvid. Ultrazvučni senzori se koriste za mjerenje udaljenosti te samim time i mjerenje volumena zapremine spremnika. Infracrveni senzori pokreta mogu detektirati korisnika pri dolasku uređaju te uređaj probuditi iz „sna“. Senzori za mjerenje težine dodatno se mogu koristiti u kombinaciji s ultrazvučnim senzorima kako bi se doznala i masa samog otpada u spremniku. U visoko automatiziranim sustavima koristili bi se i spektrometri koji bi uvidom u materijal mogli omogućiti daljnje automatsko selektiranje otpada. Dodaje se i sučelje za komunikaciju s korisnikom.

Kod odabira aktuatora, bitno je da je korisnik obaviješten o samome sustavu te se u današnje sustave najčešće ugrađuje LED zaslon. Korisnik dobiva audio-vizualne signale o stanju i pripravnosti sustava za rad, tj. da li je spremnik pun ili prazan, da li je pretinac otvoren ili nije, te pri unosu u komunikacijsko sučelje, ako ono postoji.

#### 2.3.4. Komunikacija

Komunikacija s mikrokontrolerom također bi trebala biti veoma pouzdana, robusna i sigurna. Centralni sustav koji upravlja pojedinim podsustavom, prethodno prikazano, mora znati u svakome trenutku, stanje i lokaciju pojedinog elementa sustava. Važno je uzeti u obzira da u sustavu prikupljanja otpada ima mnogo pokretnih dijelova te je jako teško stvoriti čvrste konekcije koristeći neke od metoda koje se koriste u industriji, poput PROFINET-a, PROFIBUS-a ili nekog drugog oblika s vodičima. U ovome slučaju idealno bi bilo koristiti neku vrstu bežične komunikacije.

Sustav koji se prikazao konceptualno kao idealan za ovaj problem je sustav po imenu LoRaWan<sup>8</sup>. LoRaWan sustav je mješavina LPWAN(*eng. Low Power Wide Area Network*) i LoRa(*eng. Long Range*) radio komunikacije. Koristeći male radio frekvencijske module koji mogu biti umreženi međusobno i komunicirati s centralnom jedinicom sustavom bila bi osnova ka optimizaciji ruta sakupljanja i odvoza otpada (slika 7).



Slika 7: Princip komunikacije LoRaWan sustava s centralnom jedinicom sustava (Alvaro Lozano, Javier Caridad, Juan Francisco De Paz, Gabriel Villarrubia, Javier Bajo, 2018.)

Pridobiveni podaci iz LoRaWan sustava obrađivali bi se u centralnoj jedinici sustava te bi se automatski mogle planirati rute sakupljanja za spremnike koji su označeni kao puni.

<sup>8</sup> *Smart Waste Collection System with Low Consumption LoRaWAN Nodes and Route Optimization* – Alvaro Lozano, Javier Caridad, Juan Francisco De Paz, Gabriel Villarrubia, Javier Bajo, Faculty of Science, University of Salamanca, Spain, 2018.

### 3. KOMPONENTE I HARDVER

Kao osnovu za rješavanje odabranog zadatka, koristit će se Arduino hardverske komponente zbog svoje dostupnosti i jednostavnosti. Ukoliko bi se projekt u potpunosti razvijao, svakako bi bilo bolje razviti pripradajući ugrađeni računalni sustav s mikroprocesorom i popratnim komponentama.

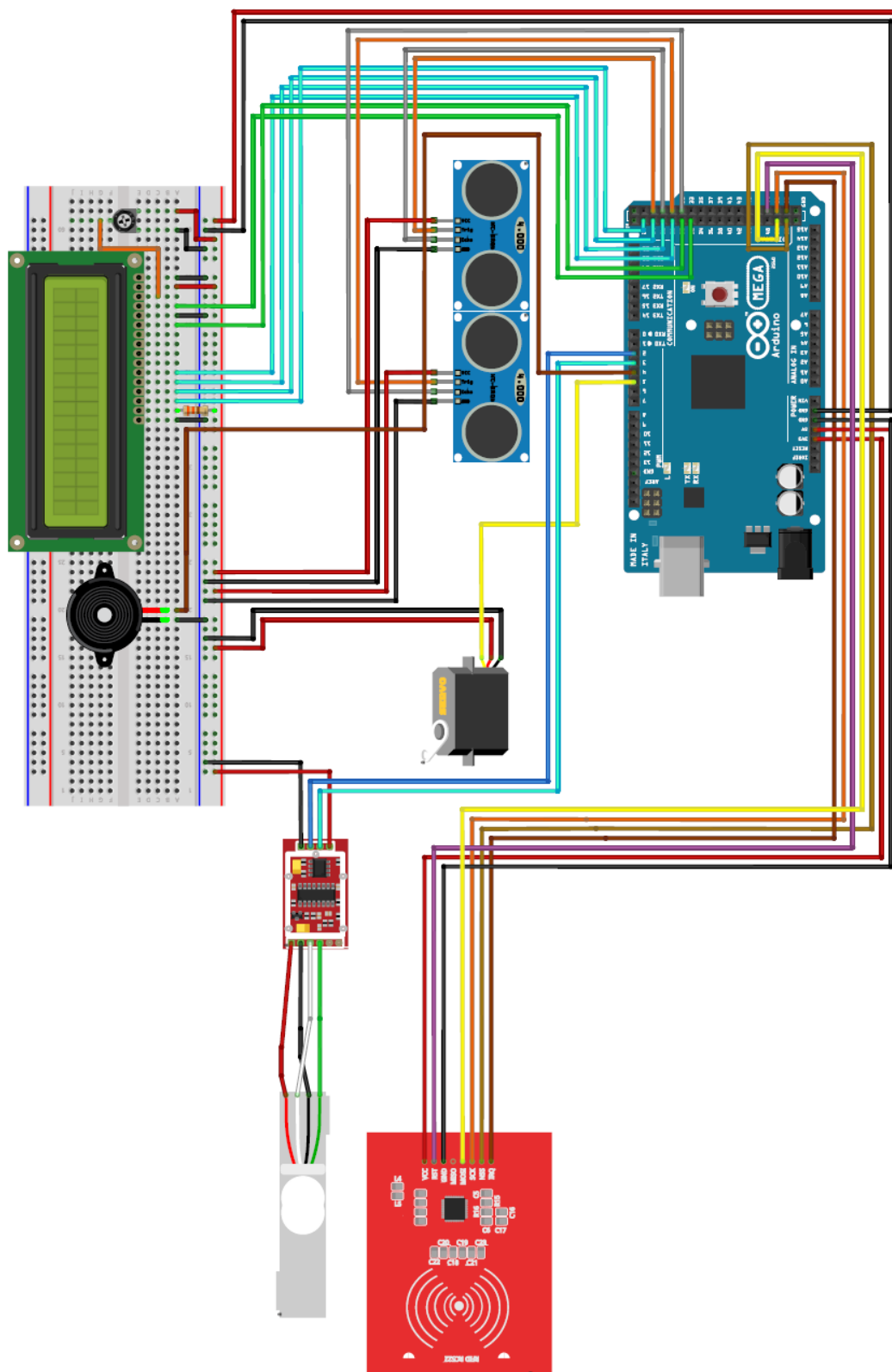
Kroz ovo poglavlje prikazat će se i pojasniti pojedine komponente potrebne za izradu sustava.

Komponente koje će biti potrebne za izvedbu ovog rada su sljedeće:

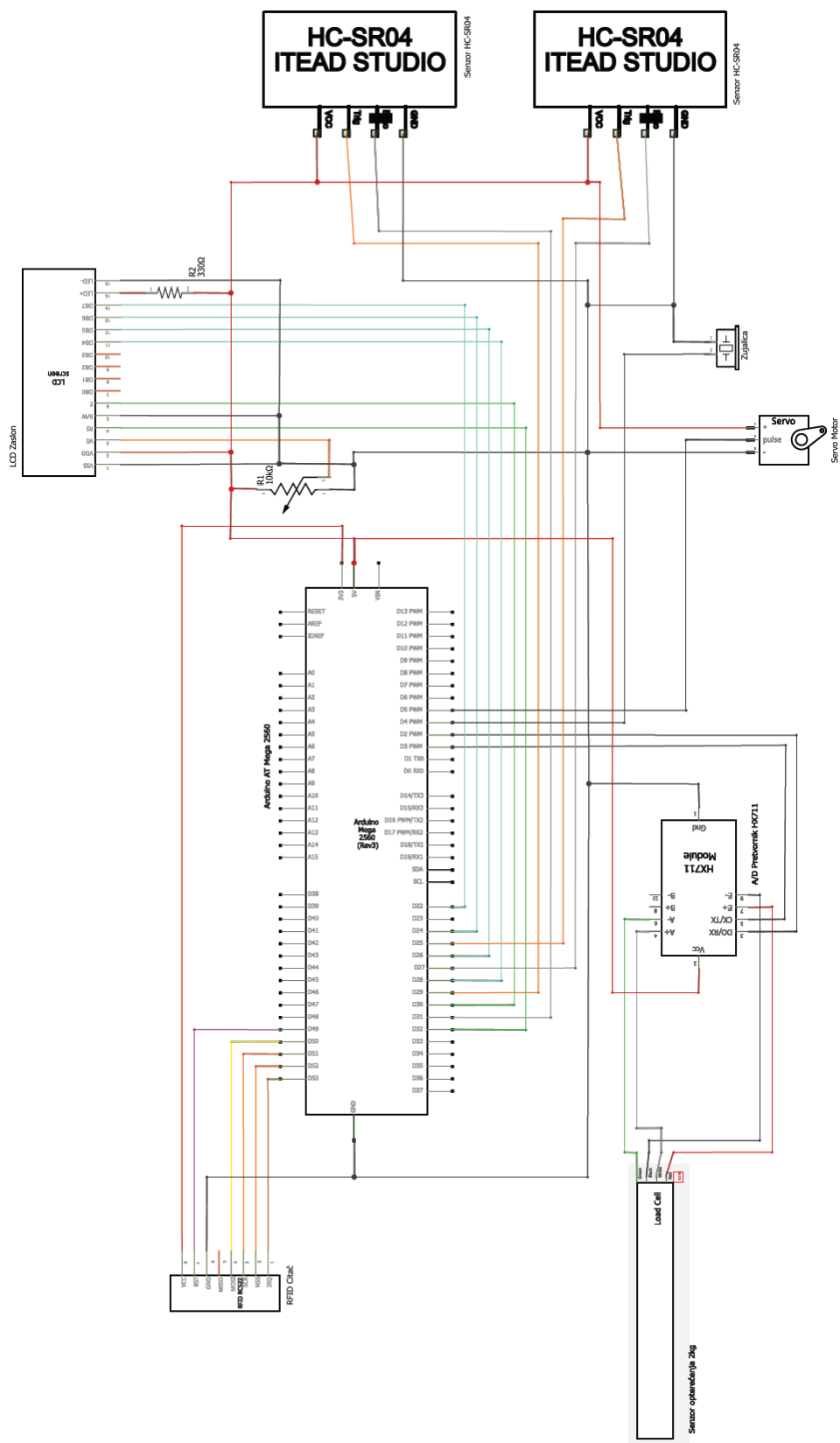
- Arduino AT Mega 2560 mikrokontroler
- RFID čitač, kartice i privjesci
- Senzor opterećenja i pripadajući A/D pretvornik
- Servo motor
- Ultrazvučni senzor HC-SR04 (2 komada)
- LCD zaslon
- Piezo zujalica
- Potenciometar 10k  $\Omega$
- Otpornik 330 $\Omega$
- Vodiči

Kako bi idejno rješenje bilo moguće realizirati, komponente je potrebno spojiti kao što je prikazano na ilustraciji (slika 8) ili na shematskom prikazu (slika 9).

### 3.1. Grafička ilustracija i shematski prikaz sklopa



Slika 8. Grafička ilustracija sklopa (Izradio Autor)

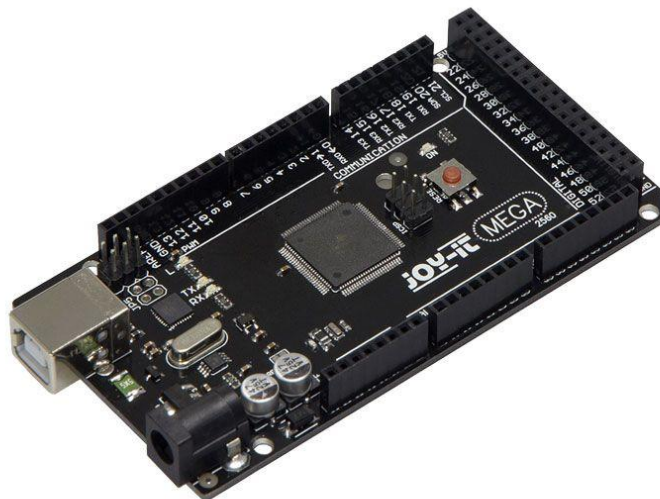


Slika 9. Shematski prikaz sklopa (Izradio Autor)

### 3.2. Arduino

Arduino Mega 2560 (Slika 10.) je elektronička ploča s Atmelovim mikroprocesorom ATmega2560. Sadrži 54 digitalna ulaza/izlaza od kojih 15 mogu biti korišteni za pulsno širinsku modulaciju signala (PWM). Ima 16 analognih ulaza, 4 UARTs serijska sučelja. Sam procesor ima kristalni oscilator sa 16 Mhz taktom. Napajan može biti preko USB priključka uz koji komunicira s računalom ili uz vanjsko napajanje od 5V.

Ova Arduino komponenta, za razliku od ostalih, odabrana je za ovaj rad zato što zadovoljava broj ulaza/izlaza bez da su potrebne dodatne komponente koje bi proširivale njegove ulaze/izlaze.



Slika 10. JoyIT Arduino Mega 2560.

### 3.3. Ultrazvučni senzori

Ultrazvučni senzori u pametnoj kanti za smeće služe kako bi se mjerila udaljenost od vrha spremnika do dna spremnika i uz pomoć zadanih parametara spremnika, izračunala preostala zapremina spremnika za pohranu otpada. Na ovaj način operater pametne kante spoznaje njezino stanje zapremine.

Kao ultrazvučni senzor koristi se HC-SR04 (Slika 11.) koji je kompatibilan s Arduinoom, zbog svojeg operativnog napona i parametara udaljenosti koje zadovoljava. (Tablica 2.)



Slika 11. Ultrazvučni senzor HC-SR04.

Ultrazvučni senzor HC-SR04 (Slika 11.) na sebi ima 4 konektora te su redom prikazani:

- 5V - Napajanje (eng. *5V Supply*)
- Ulaz – puls okidanja (eng. *Trigger Pulse Input*)
- Izlaz – puls jeke (eng. *Echo Pulse Output*)
- 0V - Uzemljenje (eng. *0V Ground*)

Električni parametri ultrazvučnog senzora HC-SR04 prikazani su u tablici 2.

Nazivni napon	DC 5V
Nazivna struja	15 mA
Radna frekvencija	40 kHz
Maksimalan raspon	4 m
Minimalan raspon	2 cm
Kut mjerenja	15 stupnjeva
Ulazni signal okidanja (eng. <i>Trigger Signal Input</i> )	10µs TTL(time-to-live) impulsa
Izlazni signal jeke (eng. <i>Echo Signal Output</i> )	Izlazni signal je proporcionalan ulaznom TTL(time-to-live) impulsu i mjerenoj udaljenosti
Dimenzije	45*20*15 mm

Tablica 2. Električni parametri ultrazvučnog senzora HC-SR04.

Pri samoj izradi rada koriste se dva ultrazvučna senzora koja mjere zasebne dijelove spremnika koji je manjih dimenzija te se njihova mjerenja uzimaju kao srednja vrijednost.

Pri korištenju s većim spremnikom koristilo bi se više senzora te bi se mapiralo zasebno područje kako bi se dobilo što preciznije mjerenje udaljenosti te u konačnici slobodnog volumena spremnika.



### 3.4. Senzor opterećenja i analogno digitalni pretvornik HX711

Kako bi se izmjerila masa otpada koji će se nalaziti unutar spremnika, potrebno je konstruirati platformu na kojoj bi se taj spremnik nalazio. Ta platforma mora imati sposobnost vršenja funkcije mjerenja opterećenja koje se na njoj nalazi.

U ovome radu koristit će se jednostavan modul senzora opterećenja (*eng. Load cell*) koji ima mogućnost mjerenja do 2kg (slika 12).



Slika 12. Senzor opterećenja – 2 kg

Senzor opterećenja obavlja svoju zadaću kao promjenjivi otpornik. Vršenjem pritiska na pričvršćen senzor na njegovom izlazu dobiva se mjerena vrijednost. U ovom slučaju, senzor opterećenja je sastavljen i radi kao Wheatstoneov most<sup>9</sup>, te kako se jednom paru otpornika otpor smanjuje, drugome se povećava, te je potrebno samo izmjeriti promjenu otpora.

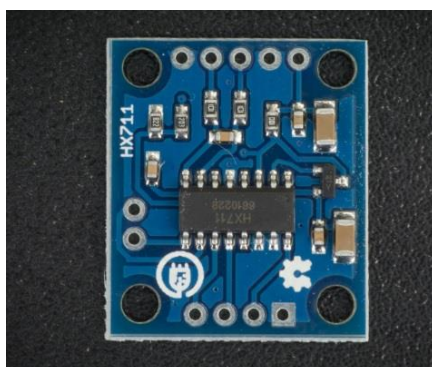
Senzor opterećenja ima 4 vodiča koji su standardizirani po boji, crvenu i crnu za napajanje (crvena E+, crna E-), te dvije žice za mjerenje koje se spajaju na analogno digitalni pretvarač (bijela A+, zelena A-).

---

<sup>9</sup> Wheatstoneov most – spoj otpornika poznatih vrijednosti i jednog promjenjivog otpornika. Vršiti se mjerenje ravnoteže dvaju čvorova za dobivanje vrijednosti promjenjivog otpornika uz pomoć Kirchhoffovih pravila.

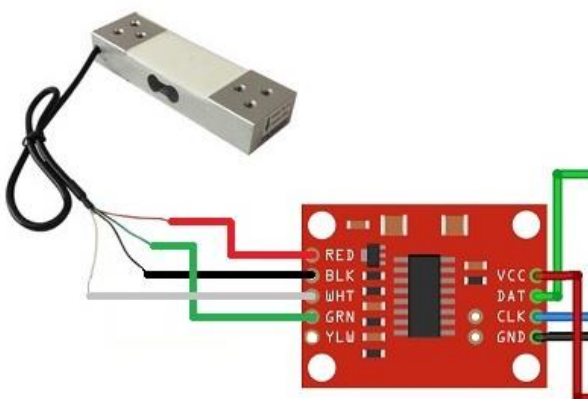
Kako bi se ovakav uređaj mogao umrežiti u Arduino sučelje, potrebno je spojiti međukomponentu koja će spojiti dva sustava. Povezivanje se vrši uz pomoć analognog digitalnog pretvornika (slika 13.) (u daljnjem tekstu A/D pretvornik).

A/D pretvornik služi kako bi se analogni signal dobiven mjerenjem uz pomoć senzora opterećenja, pretvorio u digitalni oblik i pojačao, te kasnije koristio u Arduino sučelju kao digitalna varijabla.



Slika 13. A/D pretvarač HX711

Senzor opterećenja i A/D pretvarač potrebno je spojiti prema shemi na slici 14. te kasnije pridružiti u glavnu shemu (slika 8 i slika 9). VCC priključak spaja se na napajanje od 5V, a GND priključak na uzemljenje, koje snabdijeva sam Arduino. DAT i CLK priključci su podatkovni te se povezuju na digitalne ulaze Arduina. Više o samim podatkovnim ulazima u programskom dijelu rada.

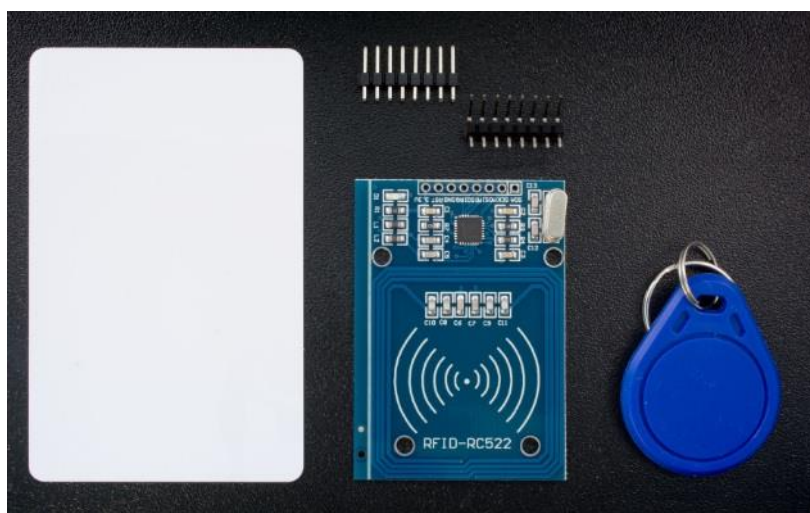


Slika 14. Shema spajanja senzora opterećenja i A/D pretvarača HX711

### 3.5. Radio-frekvencijski čitač kartica

Kako bi se omogućilo da svaki korisnik ima jedinstveni kôd i/ili uzorak za identificiranje pojedinog korisnika, u radu se koristi radio-frekvencijski čitač kartica MFRC522 (slika 15).

Prednosti korištenja ovakvog sustava bila bi mogućnost jedinstvene identifikacije pojedinog korisnika koja se može koristiti za daljnje vršenje naplate, vođenje statistika i slično.



Slika 15. Radio-frekvencijski čitač MFRC522 s karticom i privjeskom

Pri izradi rada koristi se radio-frekvencijski čitač MFRC522 koji u setu dolazi s priloženom karticom i privjeskom koji svaki nose svoj jedinstveni identifikator. Popis priključnih kontakata i njihovi opisi mogu se pronaći u tablici 3.

VCC	Napajanje 3.3V
RST	Priključak za resetiranje
GND	Uzemljenje
MISO	Priključak SPI sučelja (Master In Slave Out)
MOSI	Priključak SPI sučelja (Master Out Slave In)
SCK	Priključak serijskog takta (Serial Clock)
NSS	NSS/SS priključak ulaznog signala za komunikaciju sa SPI sučeljem
IRQ	Priključak za buđenje modula kada se uređaj približi (Interrupt Pin)

Tablica 3. Opis priključaka na MFRC522 radio-frekvencijskom čitaču

Kako bi radio-frekvencijski modul komunicirao s Arduinoom, proizvođač modula predviđa dvije vrste komunikacije. SPI (eng. *Serial Peripheral Interface*) komunikacijsko sučelje ili I2C (eng. *Inter-Integrated Circuit*) sučelje. Radi lakše izvedbe rada i mogućnosti Arduino AT Mega 2560 mikrokontrolera odabrana je metoda komunikacije sa SPI sučeljem.

Spajanje radio-frekvencijskog čitača MFRC522 s Arduino AT Mega 2560 potrebno je vršiti preko predviđenih SPI komunikacijskih priključaka koji su predviđeni na samom mikrokontroleru. Priključci na mikrokontroleru koji podržavaju SPI su sljedeći: 50-MISO, 51-MOSI, 52-SCK i 53-SS, prikazani na slici 16.

Prikaz potpuno spojenog modula vidljiv je i na prethodno prikazanom grafičkom prikazu (slika 7) i shematskom prikazu (slika 8).



Slika 16. SPI priključci na Arduino AT Mega 2560

Komunikacija i prepoznavanje jedinstvenog identifikatora dalje je pojašnjena u programskom djelu rada.

### 3.6. Servo motor

Kao glavni aktuator u ovome radu korišten je servo motor kako bi predstavljao otvaranje i zatvaranje cilindričnog pretinca u kojeg bi korisnik pohranio otpad koji bi se rotacijom cilindričnog spremnika potom istresao u glavni pohrambeni prostor.

Servo motor koji se koristi u radu je model TowerPro SG90 (slika 17) sa svojim specifikacija i dimenzijama opisanim u tablici 4, te priključcima prikazanim u tablici 5.

Ovakav servo motor je odabran zbog jednostavnosti i sposobnosti zakretanja 180 stupnjeva, tj. 90 stupnjeva u svakome smjeru.



Slika 17. Servo motor TowerPro SG90

Masa	9g
Dimenzije	22.2*11.8*31mm
Maksimalni okretni moment	2.5kg/cm
Operativna brzina	0.1s/60°
Nazivni napon	4.8V(~5V)
Širina mrtvog pojasa	10μs
Radna temperatura	0°C-55°C

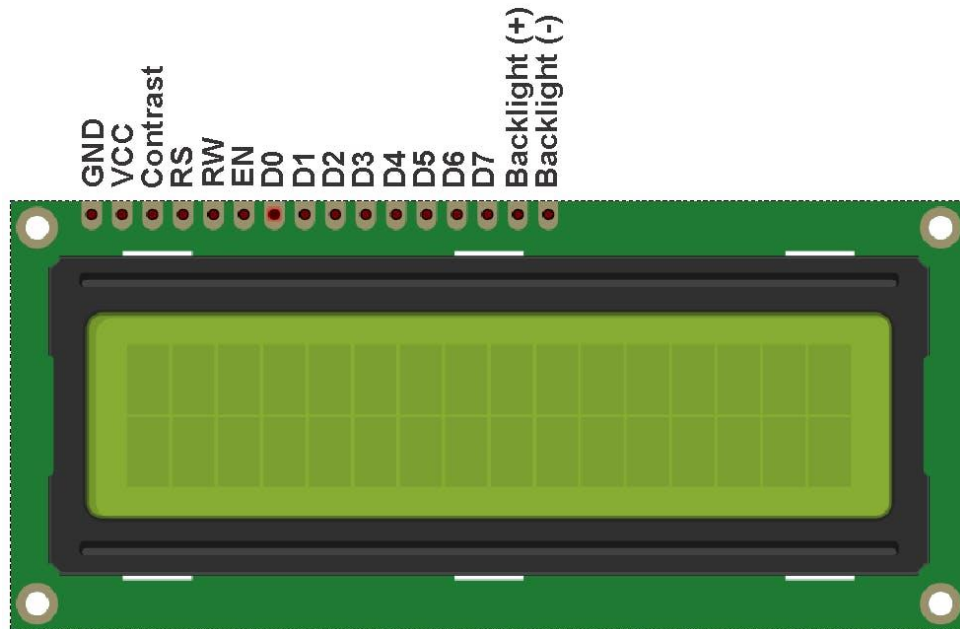
Tablica 4. Specifikacije i dimenzije servo motora TowerPro SG90

Smeđa	Uzemljenje
Crvena	Napajanje +5V
Narančasta	Priključak za upravljanje (PWM)

Tablica 5. Priključni kontakti za servo motor TowerPro SG90

### 3.7. LCD Zaslon

Kako bi se uspostavila vizualna komunikacija s korisnikom koristi se mali 16x2 LCD zaslon koji je kompatibilan s Arduino kontrolerom (slika 18). Na slici su vidljivi svi pripadajući konektori na zaslonu.



Slika 18. LCD zaslon 16x2

Opis pojedinih konektora prikazan je u tablici 6. Važno je napomenuti kako je zaslon korišten u 4-bitnom modu kako bi se smanjilo broj potrebnih spojeva s kontrolerom i zbog jednostavnosti te su korišteni samo podatkovni ulazi D4-D7.

GND	Uzemljenje
VCC	Napajanje 5V
Contrast	Pin za kontroliranje kontrasta, koristi se uz kombinaciju potencijometra
RS	Pin registra
RW	Pin za odabir čitanja iz registra ili zapis
EN	<i>Enable</i> pin odobrava slanje podataka podatkovnim pinovima DB0-DB7
D0-D7	8 podatkovnih pinova
Backlight (+/-)	Pinovi za napajanje pozadinskog osvjetljenja ekrana

Tablica 6. Opis konektora na LCD zaslonu

### 3.8. Ostale komponente

Ostatak komponenti veoma je jednostavan ali nipošto manje važan pri izradi ovog rada.

#### 3.8.1. Piezoelektrična zujalica

Zujalica (slika 19) služi kako bi korisniku dala zvučni signal pri promjenama stanja ili aktivnostima sustava te upozorila korisnika na akcije koje slijede. Zujalica korištena u ovome radu je piezoelektrična zujalica koja radi pri nazivnom naponu od 5V.



Slika 19. Piezoelektrična zujalica

#### 3.8.2. Otpornik i potencijometar

Otpornik korišten u ovom radu je veličine  $330\Omega$  te u sklopu prikazanom na shemi (slika 8 i slika 9) ima ulogu spriječiti preveliku struju, kako bi se zaštitio LCD zaslon.

Potencijometar korišten u ovome radu, veličine  $10k\ \Omega$  prikazanom na shemi (slika 8 i slika 9) ima zadaću regulacije kontrasta zaslona radi prilagodbe vidljivosti prilikom ovog rada.

#### 3.8.3. Vodiči i eksperimentalna kartica

Povezivanje komponenti vršilo se uz pomoć vodiča koji imaju priključne mogućnosti, muško-muški priključak, muško-ženski priključak te žensko-ženski priključak.

Uz same vodiče korištena je eksperimentalna kartica radi jednostavnog povezivanja komponenti, te kako bi se u testnim okruženjima izbjeglo kreiranje tiskanih pločica koje bi vršile zadanu funkcionalnost.

## **4. PROGRAMSKO RJEŠENJE ZADATKA**

### **4.1. Softverski paketi**

Kako bi se zadatak realizirao, koristio se niz manjih softvera kako bi se krenulo u izradu rada, od kojih su sljedeći: TinkerCAD, Fritzing i Arduino.

#### **4.1.1. TinkerCAD**

Platforma TinkerCAD je besplatan AutoDeskov softverski proizvod koji omogućuje izradu jednostavnih elektroničkih krugova u svojem sučelju te simulaciju istih za zadani kôd. Kao dodatne mogućnosti ima još opcije izrade jednostavnih 3D modela i dodatne slične pogodnosti.

U ovome projektu TinkerCAD se koristio kako bi se prije pribave samih komponenti izradili manji dijelovi cjelokupne sheme sklopa te stvorili manji dijelove kôda, potom provjerili osnovnu funkcionalnost u simulacijskom sučelju koje omogućuje programiranje Arduina. Zatim se stvorila osnovna shema komponenti i sklopa u cjelini.

#### **4.1.2. Fritzing**

Fritzing je otvoreni računalski softverski paket kreiran od strane profesora, alumnijsa i studenata iz Potsdama, Njemačka. Stvoren je kako bi pružio lakše dizajniranje i projektiranje elektroničkih sklopova, te lakše stvaranje njihove dokumentacije. Program se najviše koristi za izradu električkih shema i priprema za izradu tiskanih pločica i slično.

Za ovaj projekt Fritzing je korišten kako bi stvorio bolji i temeljitiji shematski prikaz, za razliku od sučelja TinkerCAD.



#### 4.1.3. Arduino

Arduino je otvorena računalna i softverska platforma koju je stvorila talijanska tvrtka SmartProjects 2005. godine. Koristi se kako bi se programirali istoimeni mikrokontroleri s Atmel-ovim mikroprocesorima. Takvi uređaji se koriste za manje projekte ili za stvaranje prototipova, zbog toga Arduino ima svoju popularnost među hobi kreatorima i mladim inovatorima. U ovome projektu Arduino MEGA mikrokontroler je programiran koristeći Arduino sučelje kako bi obavio prezentirane zadatke.

## 4.2. Postupci stvaranja softverskog rješenja

U poglavlju ovoga rada, postepeno će se proći kroz pojedine dijelove Arduino kôda te objasniti ključne segmente. Kroz ovo poglavlje objašnjavat će se pojedini dio kôda te će isti biti prikazan uz samo pojašnjenje, dok se potpuni Arduino kôd nalazi kao dodatak samome radu.

Pri samom početku kôda potrebno je definirati potrebne programske biblioteke (eng. *Libraries*) koje će omogućavati da program radi u potpunosti. U redovima 1-4 definiramo programske biblioteke za modul AD pretvarač HX711, radio frekvencijski modul MFRC-522 i dodajemo još programsku biblioteku za SPI spajanje kako se ultrazvučni modul mogao spojiti kroz to sučelje, kao što je navedeno u poglavlju 3.5.

```
1. #include "HX711.h"
2. #include "SPI.h"
3. #include "MFRC522.h"
4.
```

Kroz programske redove 5-10 definira se vaga koristeći programsku biblioteku HX711. Definira se dodatna varijabla za faktor kalibracije vage koja koristi kako bi se signal dobiven od AD pretvarača, u daljnjem kôdu, pretvorio u točnu vrijednost mase. Faktor kalibracije se ovim putem može direktno mijenjati po potrebi, ali je prvotno dobiven koristeći jedan manji pomoćni, pojednostavnjen pred program uz korištenje istog modula ćelije za mjerenje mase. Više o pomoćnom programu u poglavlju 4.3.

Sljedeće su definirani podatkovni pinovi koji s AD pretvarača dolaze na sam kontroler te su im dodijeljeni ulazi. Pin DAT(DOUT) je podatkovni preko kojeg podatak mjerenja dolazi na Arduino kontroler, dok pin SCK(CLK) upravlja s generatorom takta koji generira puls koji omogućuje valjanu pretvorbu signala.

```
5. //HX711 Pinovi
6. HX711 vaga;
7. float faktorKalibracije = -825000;
8. #define DAT 3 //DOUT
9. #define SCK 2 //CLK
10.
```

U sljedećim linijama (11-17) definirani su pinovi za SPI komunikaciju s radio frekvencijskim modulom RFID MFRC-522. Pinovi su definirani prema opisu u poglavlju 3.5., te njihovo spajanje ovisi o Arduino SPI pinovima koji su također navedeni.

```
11. //RFID Pinovi
12. #define RFID_MISO 50
13. #define RFID_MOSI 51
14. #define RFID_SCK 52
15. #define RFID_SDA 53 //Set State
16. #define RFID_RST 49 //Reset pin
17.
```

Kako bi se omogućilo da radio frekvencijski čitač radi unutar Arduino sučelja, koristi se navedena programska biblioteka MFRC522 te se uz pomoć nje definira nova varijabla i dodjeljuju pinovi za postavljanje i resetiranje stanja.

```
18. MFRC522 mfrc522(RFID_SDA, RFID_RST);
19. String read_rfid;
20. String rfid_card = "d5543e11"; //kod naše kartice, dobiveno prethodno
21. String rfid_tag = "3b2ead"; //kod našeg taga, dobiveno prethodno
22.
23.
```

Definirane su varijable za vrijednost koja je očitana na čitaču, te dodatne dvije varijable koje su naši prethodno definirani i dobiveni RFID oznaka i kartica. Kako se sazna identifikator pojedinačne kartice i oznaka koje se koristi, prikazano je u pomoćnom programu u poglavlju 4.4.

U linijama kôda 24-27 definirana je programska biblioteka za korištenje servo motora, definiran je servo motor te je istoimeno nazvan te je definiran pin na koji će biti spojen s Arduino mikrokontrolerom.

```
24. #include <Servo.h>
25. Servo servo;
26. #define servoPin 5 //Servo PWM
27.
```

Kako bi se omogućilo korištenje LCD zaslona, potrebno je definirati programsku biblioteku za njegovo korištenje te definirati sve potrebne pinove koji moraju biti u komunikaciji s kontrolerom. Definiranje LCD zaslona prikazano je u linijama kôda 28-36.

```

28. #include <LiquidCrystal.h>
29. //Register Select Pin = 32
30. //Enable Pin = 30
31. //DB4 Pin = 28
32. //DB5 Pin = 26
33. //DB6 Pin = 24
34. //DB7 Pin = 22
35. LiquidCrystal lcd(32, 30, 28, 26, 24, 22);
36.

```

Prikazano u linijama kôda 37-50, nalaze se definicije oba ultrazvučna senzora. Definirana su dva priključna pin senzora koja komuniciraju s Arduino kontrolerom i predaju mu informaciju. Kako bismo kasnije podatak senzora pretvorili u vrijednu informaciju definiraju se varijable vremena, varijabla vrijednosti očitana od strane senzora, te varijabla realne vrijednost koja ovisi i o samoj poziciji i senzora i geometriji pohrabenog prostora koja je kasnije definirana.

```

37. //Senzor 1
38. #define trigPin1 25
39. #define echoPin1 27
40. long vrijeme1;
41. float senzorUdaljenost1;
42. float realnaUdaljenost1;
43.
44. //Senzor 2
45. #define trigPin2 29
46. #define echoPin2 31
47. long vrijeme2;
48. float senzorUdaljenost2;
49. float realnaUdaljenost2;
50.

```

U kasnijem dijelu projekta, u poglavlju 5, odabrane su dimenzije spremnika unutar kante za smeće te su definirane sljedećim linijama kôda (51-55). U slučaju da se sustav primjenjuje za drukčije dimenzije spremnika, ovdje je moguće promijeniti parametre. Poznavanjem dimenzija, moguće je izračunati volumen spremnika u koji će se pohranjivati otpad.

```

51. int a_strana = 26;    //Kanta, dimenzije u centimetrima
52. int b_strana = 15;
53. float visinaKante = 12; //cm
54. float volumenKante = a_strana*b_strana*visinaKante;
55.

```

Kao sljedeće varijable potrebno je definirati volumen i masu smeća koja će se računati i mjeriti. Definirani su u linijama 56-58.

```
56. float volumen;  
57. float masa;  
58.
```

Pri izradi prototipa promjerena udaljenost senzora do spremnika unutar kante iznosi 3 cm te je i to potrebno uzeti u obzir pri računanju. Definirana je vrijednost srednje udaljenosti zbog prisustva više senzora te se definira varijabla postotka kako bi se kasnije mogla vrijednost prikazati na LCD zaslonu. Prikazano na linijama kôd 59-63.

```
59. float razmakKantaSenzor = 3; //cm  
60. float srednjaUdaljenost;  
61. int postotak;  
62. //int površina = a_strana*b_strana;  
63.
```

U linijama kôda 64-66 definirana je piezoelektrična zujalica i njezin priključni pin, te je definirana i varijabla prethodnog stanja, koja će se koristiti kako bi se manipuliralo radom pametne kante, znajući kada je bila otvorena ili zatvorena.

```
64. #define zujalica 4 //Buzzer Pin  
65. int pretStanje;  
66.
```

Kao zadnju fazu definiranja, u linijama kôda 67-73, definira se varijabla vremena koje će se koristiti za upravljanje vremenom dok je kanta u otvorenom stanju a definirano je i vrijeme reakcije servo motora, koje će se koristiti za precizan, stupnjeviti pomak servo motora. Dodatno je definirana i varijabla servo pozicije.

```
67. int otvorenoVrijeme;  
68. int servoVrijeme = 1000;  
69. int servoPozicija;  
70.  
71. //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////  
72. //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////  
73.
```

Linijama kôda 74-83 započinje postavljanje svih podsustava od kojih je građena cjelina. Pinovima senzorima 1 i 2 pridodana je namjena ulaza i izlaza. Pinu zujalice također je pridodana izlazna namjena. Servu je pridodan njegov unaprijed definirani pin te je pomaknut u početnu poziciju.

```
74. void setup()
75. {
76.   pinMode(trigPin1, OUTPUT); //Senzor 1
77.   pinMode(echoPin1, INPUT);
78.   pinMode(trigPin2, OUTPUT); //Senzor 2
79.   pinMode(echoPin2, INPUT);
80.   pinMode(zujalica, OUTPUT);
81.   servo.attach(servoPin); //Servo 1
82.   servo.write(0);
83.
```

LCD zaslon započinje svoj rad (linije kôda 84-87) te je definirano njegovo polje rada 16x2. Početak rada zaslona kreće s praznim zaslonom na kojem se ispisiše „Učitavanje...“, sve dok kanta ne dođe u radno stanje.

```
84. lcd.begin(16, 2); //LCD
85. lcd.clear();
86. lcd.print("Ucitavanje...");
87.
```

U linijama kôda 88-96, serijski monitor počinje s radom, te on služi kao nadzor pojedinih varijabli, iz sustava nadzora. Vaga započinje s radom te je sa spremnikom koji se nalazi u kanti, vaga nulirana pri početku rada. Više o samom postavljanju vage u poglavlju 4.3.

```
88. Serial.begin(9600);
89.
90. vaga.begin(DAT, SCK);
91. vaga.set_scale();
92. vaga.tare(); //Reset the scale to 0
93. long zero_factor = vaga.read_average(); // Osnovno očitavanje
94. //Serial.print("Zero factor: "); // Komentar ako je potrebno očitati stanja u serial monitoru
95. //Serial.println(zero_factor); // Komentar ako je potrebno očitati stanja u serial monitoru
96.
```

U zadnjim linijama pokretanja programa (linije kôda 97-106) s radom započinje SPI sučelje i inicira se rad radio frekvencijskog čitača kartica. Varijabla prethodnog stanja postavlja se na 1, što prikazuje da je sustav prethodno bio otvaran ili tek započeo s radom.

```
97. while (!Serial);
98. SPI.begin();
99. mfrc522.PCD_Init();
100.
101. pretStanje=1;
102. }
103.
104. //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
105. //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
106.
```

Sljedeće linije kôda (linije kôda 107-117) koriste dio naredbi iz programske biblioteke MFRC-522 koja omogućuje očitavanje RFID vrijednosti na čitaču te pohranjuje očitane vrijednosti. Više o MFRC-522 u opisu pomoćnog programa u poglavlju 4.4.

```
107. void dump_byte_array(byte *buffer, byte bufferSize)
108. {
109.   read_rfid="";
110.   for (byte i = 0; i < bufferSize; i++) {
111.     read_rfid += String(buffer[i], HEX);
112.   }
113. }
114.
115. //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
116. //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
117.
```

Zadnji dio u programu svodi se na petlju (linija kôda 275) koja provjerava ulaze sustava. Ovisno o ulazu i stanju sustava aktivira se pojedini segment. U linijama 118-134 prikazan je segment koji se aktivira kad radnik priđe uređaju sa svojim RFID oznakom te s time ima uvid u dodatne informacije o masi i volumenu.

```
118. void dodatneInformacije()
119. {
120.   lcd.clear();
121.   lcd.print("Masa: ");
122.   lcd.print(masa);
```

```

123. lcd.print(" kgs");
124. lcd.setCursor(0, 1);
125. lcd.print("Vol: ");
126. lcd.print(volumen);
127. lcd.print(" cm3");
128. delay(10000);
129. pretStanje=1;
130. }
131.
132. //////////////////////////////////////
133. //////////////////////////////////////
134.

```

Sljedeći segment kôda aktivira se kada korisnik priđe pametnoj kanti za smeće te prisloni svoju RFID karticu. Uređaj daje zvučni i vizualni signal korisniku te se potom spremnik za pohranu. Otvor ostaje otvoren 15 sekundi, te se jasno ispisiuje preostalo otvoreno vrijeme na LCD zaslonu. Nakon toga servo motor pomalo zatvara otvor pametne kante za smeće uz davanje vizualnog i zvučnog signala. Pri zatvaranju sustava, varijabla prethodnog stanja se mijenja u 1 što označava da je sustav u ovome ciklusu bio otvoren.

```

135. void visoko()
136. {
137.   tone(zujalica,1000, 1000);
138.   delay(100);
139.   tone(zujalica,500, 500);
140.   delay(100);
141.   noTone(zujalica);
142.
143.   lcd.clear();
144.   lcd.print("Otvaranje...");
145.   delay(2000);
146.
147.   for(servoPozicija=0; servoPozicija<=140; servoPozicija++){ //Stupnjevito otvaranje servo
motora
148.     servo.write(servoPozicija);
149.     delay(servoVrijeme/140);
150.   }
151.
152.   for(otvorenoVrijeme=15; otvorenoVrijeme>=0; otvorenoVrijeme--){
153.     lcd.clear();
154.     lcd.print("Otvoreno");
155.     lcd.setCursor(0, 1);
156.     lcd.print(otvorenoVrijeme);
157.     delay(1000);
158.   }
159.
160.   lcd.clear();
161.   lcd.print("Zatvoreno");
162.   lcd.setCursor(0, 1);
163.   lcd.print("Stanje: ");
164.   lcd.print(postotak);
165.   lcd.print("%");
166.

```



```

167. lcd.clear();
168. lcd.print("Zatvaranje...");
169.
170. tone(zujalica,2000, 100);
171. delay(100);
172. tone(zujalica,2000, 100);
173. delay(100);
174. noTone(zujalica);
175. delay(1000);
176.
177. for(servoPozicija=140; servoPozicija>=0; servoPozicija--){ //Stupnjevito zatvaranja servo
    motora
178. servo.write(servoPozicija);
179. delay(servoVrijeme/140);
180. }
181.
182. delay(500);
183. lcd.clear();
184. lcd.print("Zatvoreno");
185. pretStanje=1;
186. }
187.
188. //////////////////////////////////////
189. //////////////////////////////////////
190.

```

Segment kôda koji radi pri niskom stanju, kad je sustav zatvoren započinje s mjerenjem senzora udaljenosti. Poznavajući brzinu zvuka i vrijeme, dobiva se realna udaljenost koja se potom umanjuje za razmak senzora od spremnika. Potom se računa srednja udaljenost ovisna o broju senzora.

Imajući podatke o udaljenosti i veličinama spremnika, može se proračunati postotak popunjenosti volumena prostora spremnika.

Na LCD zaslonu spremnika ispisuje se stanje „Zatvoreno“ te postotak popunjenosti.

Modul vage potom vrši unutarnje očitavanje mase, koja isto ima ulogu u nadzoru, te se prikazuje stanje na serijskom monitoru, za unutarnji nadzor. Informacija je isto dostupna radniku u prikazu za dodatne informacije koristeći RFID oznaku.

Prethodno stanje se postavlja na 0, te znači da je prethodno stanje sustava bilo zatvoreno i mjereno.

```

191. void nisko()
192. {
193. //Nakon sto je sustav zatvoren ide procjena zapremnine
194. // Udaljenost // zvuk (cm/us) //dijelimo s 2, dupla udaljenost
195.
196. //////////////////////////////////////
197.
198. digitalWrite(trigPin1, LOW); //Senzor1

```

```

199. delayMicroseconds(2);
200. digitalWrite(trigPin1, HIGH);
201. delayMicroseconds(10);
202. digitalWrite(trigPin1, LOW);
203. vrijeme1 = pulseIn(echoPin1, HIGH);
204. senzorUdaljenost1= vrijeme1*0.034/2;
205. realnaUdaljenost1=senzorUdaljenost1-razmakKantaSenzor;
206.
207. delay(100);
208.
209. ///////////////////////////////////////////////////////////////////
210.
211. digitalWrite(trigPin2, LOW); //Senzor2
212. delayMicroseconds(2);
213. digitalWrite(trigPin2, HIGH);
214. delayMicroseconds(10);
215. digitalWrite(trigPin2, LOW);
216. vrijeme2 = pulseIn(echoPin2, HIGH);
217. senzorUdaljenost2= vrijeme2*0.034/2;
218. realnaUdaljenost2=senzorUdaljenost2-razmakKantaSenzor;
219.
220. ///////////////////////////////////////////////////////////////////
221.
222. srednjaUdaljenost=(realnaUdaljenost1+realnaUdaljenost2)/2;
223. postotak=((visinaKante-srednjaUdaljenost)/visinaKante)*100;
224.
225. lcd.clear();
226. lcd.print("Zatvoreno");
227. lcd.setCursor(0, 1);
228. lcd.print("Stanje: ");
229. lcd.print(postotak);
230. lcd.print("%");
231.
232. vaga.set_scale(faktorKalibracije); //Prilagoditi faktor kalibracije po potrebi
233. masa=vaga.get_units(),3;
234. Serial.print("Ocitanje: ");
235. Serial.print(masa);
236. Serial.print(" kgs");
237. Serial.print(" faktor kalibracije: ");
238. Serial.print(faktorKalibracije);
239. Serial.println();
240.
241. Serial.print("Udaljenost: ");
242. Serial.print(srednjaUdaljenost);
243. Serial.println();
244. Serial.print("Postotak: ");
245. Serial.print(postotak);
246. Serial.println();
247. Serial.print("Senzor 1: ");
248. Serial.print(senzorUdaljenost1);
249. Serial.println();
250. Serial.print("Senzor 2: ");
251. Serial.print(senzorUdaljenost2);
252. Serial.println();
253.
254. volumen=(a_strana*b_strana*srednjaUdaljenost);
255.
256. pretStanje=0;
257. }
258.

```

```

259. //////////////////////////////////////////////////
260.
261. //////////////////////////////////////////////////

```

Pretposljednji segment kôda je predviđen za stanje sustava kada je sustav pun, tj. ako je masa veća od zadane (u ovome slučaju 2 kg) ili ako je spremnik za pohranu otpada popunjen. Ake je spremnik pun, spremnik ispisuje „Zatvoreno“ i „Stanje:PUNO“. Prethodno stanje sustava se postavlja u 2 te se time sprječava daljnje otvaranje i diranje sustava sve dok radnik ili nekakav autonomni uređaj pristupe i isprazne sadržaj spremnika za otpad te vrate sustav u prvobitno stanje.

```

262. void puno()
263. {
264.   lcd.clear();
265.   lcd.print("Zatvoreno");
266.   lcd.setCursor(0, 1);
267.   lcd.print("Stanje: PUNO");
268.   pretStanje=2;
269. }
270.
271.
272. //////////////////////////////////////////////////
273. //////////////////////////////////////////////////
274.

```

Zadnji dio programa čini petlja koja upravlja uvjetima i odlučuje koji segment kôda će se aktivirati ovisno o zadovoljenim uvjetima i prethodnim stanjima. Ake je sustav bio prethodno otvoren, petlja pokreće mjerenje i pohranu. Ako je masa sustava ili volumen veći od dopuštenog, sustav se zatvara, u način rada „Stanje: Puno“. Ako na RFID čitač priđe RFID identifikator koji nije poznat sustavu, sustav samo nastavlja s daljnjim radom.

Petlja prepoznaje kad joj priđe poznata RFID kartica korisnika ili RFID oznaka radnika, te sustav potom čini sljedeće vezane akcije.

```

275. void loop()
276. {
277.
278.   if (pretStanje==1){
279.     niski();
280.   }
281.
282.   if(masa>=2){
283.     puno();
284.   }

```

```

285.
286. if(volumen<(volumenKante*0,05)){
287.   puno();
288. }
289.
290. if ( ! mfrc522.PICC_IsNewCardPresent()){
291.   return;
292. }
293.
294. if ( ! mfrc522.PICC_ReadCardSerial()){
295.   return;
296. }
297.
298. dump_byte_array(mfrc522.uid.uidByte, mfrc522.uid.size);
299.
300. Serial.println(read_rfid);
301.
302. if (pretStanje==1 && read_rfid == rfid_card){
303.   visoko();
304. }
305.
306. if (pretStanje==0 && read_rfid == rfid_card){
307.   visoko();
308. }
309.
310. if (read_rfid == rfid_tag){
311.   dodatneInformacije();
312. }
313. }

```

Završetkom kôda završava ovo poglavlje gdje se pokušalo približiti programski kôd čitatelju. Dodatni pomoćni programi koji su se koristili za postavljanje RFID kartica i oznaka i postavljanje vage slijede u sljedećim poglavljima.

### 4.3. Pomoćni program za HX711

Pomoćni program za HX711 je program koji se koristi u pripremi uređaja za rad. Svrha ovoga kôda je da se dokuči točni faktor kalibracije vage. U slučaju ovoga rada, radi se sa senzorom opterećenja koji ima mogućnost mjerenja do 2 kg. Kako bi izlazni signal na serijskom monitoru i kasnije u samom proračunu bio točan, valja provesti kalibraciju. Kalibracija se odrađuje uz uporabu 100 gramskog utega koji je postavljan na senzor i njegovu platformu te se podešavao faktor kalibracije dok vrijednosti nisu bile ujednačene. Provjera se vršila s usporedbama raznih predmeta na vagi, kako bi se potvrdila točnost.

```
1. #include "HX711.h"
2.
3. #define DOUT 3
4. #define CLK 2
5.
6. HX711 vaga;
7.
8. float faktorKalibracije = -825000;
9.
10. void setup() {
11.   Serial.begin(9600);
12.   Serial.println("HX711 kalibracijski test");
13.   Serial.println("Ukloniti sve s vage");
14.   Serial.println("Kad započne mjerenje, staviti predmet");
15.   Serial.println(" + ili a za povećanje faktora kalibracije");
16.   Serial.println(" - ili d to smanjivanje faktora kalibracije");
17.
18.   vaga.begin(DOUT, CLK);
19.   vaga.set_scale();
20.   vaga.tare(); //postavljanje vage na 0
21.
22.   long zero_factor = vaga.read_average(); //Osnovno očitavanje
23.   Serial.print("Zero factor: "); //Može se koristiti za maknuti opciju tare vage, te se može koristiti
   u trajnijem projektu
24.   Serial.println(zero_factor);
25. }
26.
```

Vaga je definirana uz pomoć programske biblioteke HX711.h te su na pinove 2 i 3 spojeni izvodi s HX711 AD pretvarača. Serijski monitor započinje s radom te se uz pomoć petlje, pritiskom na tipke +/- ili a/d povećava i smanjuje faktor kalibracije za 10.

```
27. void loop() {
28.
29.   vaga.set_scale(faktorKalibracije);
30.   Serial.print("Masa: ");
31.   Serial.print(vaga.get_units(), 3);
32.   Serial.print(" kgs");
33.   Serial.print(" Faktor Kalibracije: ");
34.   Serial.print(faktorKalibracije);
```

```
35. Serial.println();
36.
37. if(Serial.available())
38. {
39. char temp = Serial.read();
40. if(temp == '+' || temp == 'a')
    a. faktorKalibracije += 10;
41. else if(temp == '-' || temp == 'd')
    a. faktorKalibracije -= 10;
42. }
43. }
```

Kada rezultat mjerenja bude jednak mjerenom predmetu, te s raznim predmetima jednake mase, utvrđen je faktor kalibracije za slučaj ovog senzora opterećenja, tj. vage. Faktor kalibracije se prenosi u glavi dio kôda.

#### 4.4. Pomoćni program za MFRC-522

Prije same uporabe RFID čitača, kartica i oznaka, bilo je potrebno dobiti identifikatore istih. Kako bi se prepoznali identifikatori kartica i oznaka, korišten je sljedeći pomoćni program koji ima jednostavan cilj.

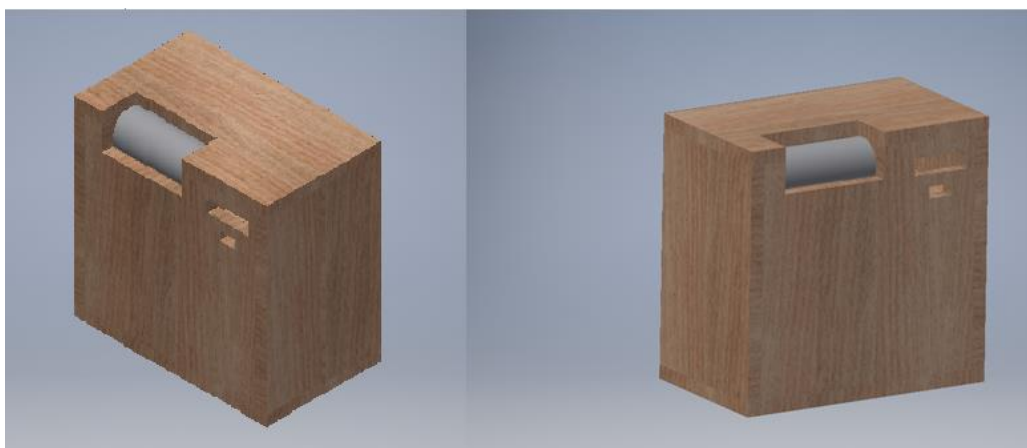
Program vrši funkciju tako što su RFID podatkovni pinovi spojeni s Arduino kontrolerom koristeći SPI sučelje. Definiran je MFRC522 uređaj uz pomoć svoje programske biblioteke te serijski monitor počinje s radom. Petlja izvršava funkciju provjera novih kartica i oznaka, te ako je detektirala nove, očitava ih i ispisuje njihove identifikatore na serijskom monitoru. Identifikatori poznatih kartica i oznaka korišteni su u glavnom programu kao poznate vrijednosti s kojima se radi.

```
1. #include <SPI.h>
2. #include <MFRC522.h>
3.
4. #define RST_PIN    5
5. #define SS_PIN     53
6.
7. MFRC522 mfrc522(SS_PIN, RST_PIN); // Definiranje MFRC522
8.
9. void setup() {
10.   Serial.begin(9600);
11.   while (!Serial);
12.   SPI.begin(); // Init SPI bus
13.   mfrc522.PCD_Init(); // Init MFRC522
14.   delay(4); // Opcionalno kašnjenje
15.   mfrc522.PCD_DumpVersionToSerial(); // Detalji RFID čitača
16.   Serial.println(F("Scan PICC to see UID, SAK, type, and data blocks..."));
17. }
18.
19. void loop() {
20.   // Povratak petlje ako nije detektirana nova kartica ili tag
21.   if ( ! mfrc522.PICC_IsNewCardPresent()) {
22.     return;
23.   }
24.   // Nova kartica ili tag biva očitano
25.   if ( ! mfrc522.PICC_ReadCardSerial()) {
26.     return;
27.   }
28.
29.   // Izbacivanje podataka u serijski monitor
30.   mfrc522.PICC_DumpToSerial(&(mfrc522.uid));
31. }
```

## 5. 3D MODEL I IZRADA PROTOTIPA

Najbolji način za prikazati projekt je da se ukratko isplanira i realizira nekakva vrsta prototipa. Prototip pametne kante za smeće bit će realiziran s namjenom da se prikaže funkcionalnost samog sustava i softvera, dok se za samu estetiku i ostale vizualne dijelove uvijek može vratiti i dotjerati po potrebama i željama. Sve dimenzije prototipa su minimizirane, dok bi se u realnosti projekt radio u veličinama adekvatnima za pohranu otpada.

Kako bi se najbolje i najjednostavnije stvorilo rješenje 3D modela, koristio se AutoDeskov Inventor u kojem su stvorene sheme pojedini dijelova koji su potrebni za sklapanje samog modela, te su na posljetku složeni u cjelinu (slika 20).



Slika 20. 3D model pametne kante za smeće (Izradio Autor)

Nakon što je stvoren model, odlučeno je za izradu samo prototipa. Pošto se izrada prototipa vrši u kućnoj radionici kao materijal odabrano je drvo, dok će pokretni cilindrični dio biti izrađen od metalne cijevi s plastičnim priključkom za servo motor.

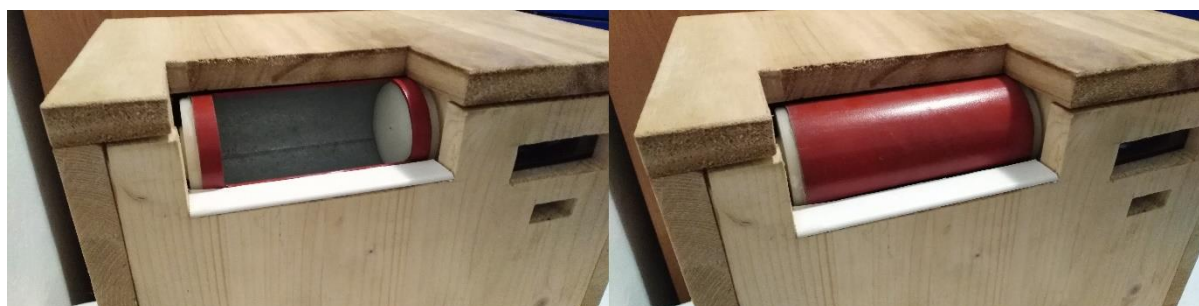
Kad je izrada u kućnoj radionici završila, dobio se model sličan zamišljenom u AutoDesk Inventoru. Prototip je prikazan na slici 21.





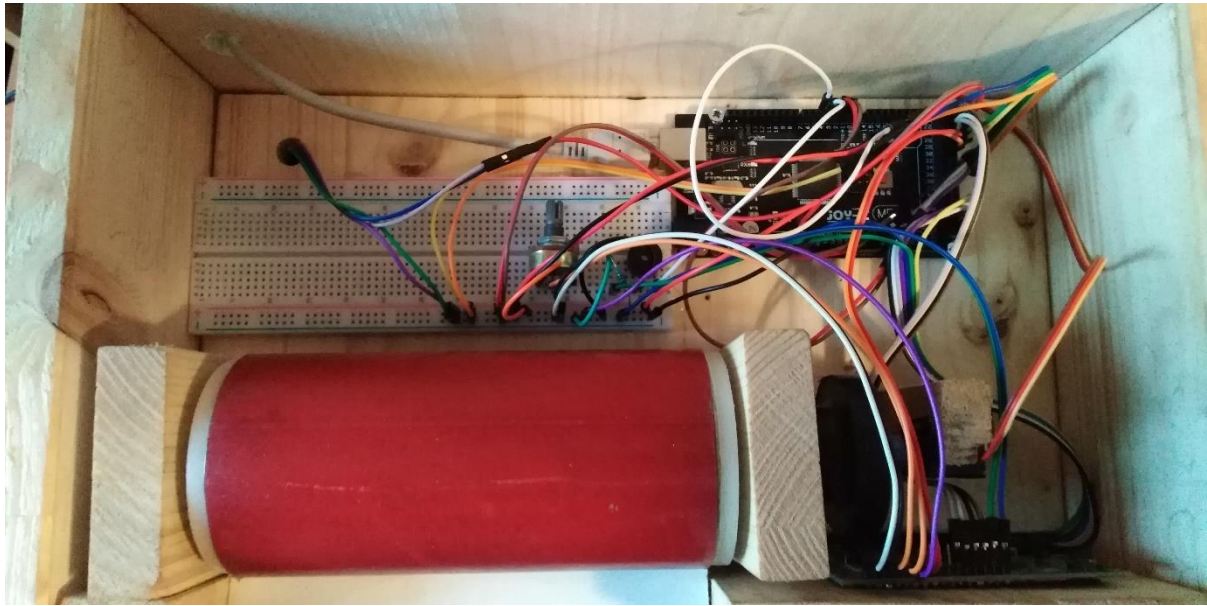
Slika 21. Kućni prototip pametne kante za smeće (Izradio Autor)

Na slici 22 prikazan je pretinac za ubacivanje otpada u oba stanja, u otvorenom stanju i zatvorenom stanju.



Slika 22. Pretinac za ubacivanje otpada (Izradio Autor)

Prilikom otvaranja gornjeg poklopca, vidljiva je polica s komponentama, na slici 23. Na polici se nalazi Arduino mikrokontroler s eksperimentalnom karticom koja je korištena kako bi se umrežile i spojile sve komponente. Većinu prostora zauzima pokretni dio, pretinac za ubacivanje otpada. Na strani bližoj korisniku, na stranicu kante, pričvršćen je LCD zaslon i RFID čitač.



Slika 23. Polica s komponentama (Izradio Autor)

Sa stražnje strane nalazi se otvor gdje se nalazi spremnik za otpad, koji je odmah ispod prethodnog za ubacivanje otpada. Unutar donjeg dijela kante nalaze se dva ključna dijela sklopa, a to su ultrazvučni senzori (slika 24) i vaga (slika 25).



Slika 24. Ultrazvučni senzori (Izradio Autor)

Spremnik za pohranu otpada nalazi se na drvenom postolju ispod kojeg se nalazi pričvršćeni senzor mjerenje mase, kao što je prikazano na slici 25. Uz stijenku kante za smeće, pričvršćen je HX711 AD pretvarač koji čini komunikaciju senzora i mikrokontrolera mogućom.



Slika 25. Vaga (Izradio Autor)

## 6. ZAKLJUČAK

Ovaj rad služi kao buduća ideja za efikasniju i bolje organizirano upravljanje otpadom u urbanim zonama. Snaga i cilj ovog uređaja nije u tome da je jedan i jedinstven, već da se na više područja nalazi ovakav uređaj, te da su uređaji međusobno umreženi, te s operativnim centrom za zbrinjavanje otpada razmjenjuju informacije o svojoj zapremini i stanju. Samim time, logistika odvoza i sortiranja otpada bi se poboljšala.

Na ovakav koncept veoma je lako nadograditi direktan sustave naplate samome klijentu koristeći jedinstvene identifikatore, te je klijentu moguća naplata usluge, proporcionalna količini otpada koju je korisnik pohranio kao i pregled količina odloženog otpada u realnom vremenu.

Kao i mnogi slični sustavi, i ovaj ima iste mane, a ta mana je da se još uvijek bazira na povjerenju korisnika. Pod povjerenje korisnika podrazumijeva se na to da ako postoji više različitih kanti za različite otpade, (npr. papir, staklo, plastika, komunalni otpad), još uvijek ovisimo o korisniku da će prepoznati u koji spremnik odložiti točan otpad, te da korisnik odvaja sam otpad što se može unaprijediti stalnim edukacijama ili kampanjama.

Kao daljnju nadogradnju urbanih kanti za smeće, bilo bi vješto smisliti sustav koji u minijaturnoj verziji, samostalno, na bazi inteligentnog sustava imao moć prepoznavanja vrste otpada koristeći spektrometre ili nekakve slične senzore, te nakon toga razvrstavao otpad uz sve dosadašnje mogućnosti koje bi i ovakav sustav imao, tj. umreženost, dijagnostika i bolja logistika.

## 7. POPIS LITERATURE

### 7.1. Knjige i članci

- *Oprema i infrastruktura za održivo gospodarenje otpadom u jedinicama lokalne/regionalne samouprave (I.dio)* – dr.sc. Branimir Fuk, dipl. ing. Rud., Ministarstvo zaštite okoliša i energetike, Zagreb *SIGURNOST* 60 (2) 183 - 188 (2018)
- Norme ISO 14000 – Normizacija sustava upravljanja zaštitom okoliša
- Znanstveni članak: *AioT-Based Smart Bin for Real-Time Monitoring and Management of Solid Waste* – Anika Bano, Ikram Ud Din, Asma A. Al-Huqail, Department of Information Technology, The University of Haripur, Pakistan
- Akademski članak: *The Design and Implementation of Smart Trash Bin* - Fady Samann, Duhok Polytechnic University, Iraq, Duhok, 2020.
- *Arduino: Advanced Methods and Strategies of Using Arduino* – Ethan Thorpe
- Akademski članak: *Smart Waste Collection System with Low Consumption LoRaWAN Nodes and Route Optimization* – Alvaro Lozano, Javier Caridad, Juan Francisco De Paz, Gabriel Villarrubia, Javier Bajo, Faculty of Science, University of Salamanca, Spain, 2018.

### 7.2. Programski paketi

- Fritzing (16.02.2021.) <https://fritzing.org/>
- Arduino (16.02.2021.) <https://www.arduino.cc/>
- TinkerCAD (19.5.2021.) <https://www.tinkercad.com/>
- AutoDesk Inventor (23.5.2021.)  
<https://www.autodesk.com/products/inventor/overview?term=1-YEAR>
- 18: Vaga – Izvor: Izradio autor

### 7.3. Internet izvori

- Plan Gospodarenja Otpadom Republike Hrvatske, usvojen 2017 godine  
[http://www.haop.hr/sites/default/files/uploads/dokumenti/021\\_otpad/Izvjescia\\_OTP\\_Pregled%20podataka%20o%20gospodarenju\\_otpadom%20u%20RH\\_iz\\_PGO.pdf](http://www.haop.hr/sites/default/files/uploads/dokumenti/021_otpad/Izvjescia_OTP_Pregled%20podataka%20o%20gospodarenju_otpadom%20u%20RH_iz_PGO.pdf)
- Arduino Mega 2560 (11.02.2021.) <https://store.arduino.cc/arduino-mega-2560-rev3>

- Ultrazvučni senzor HC-SR04 e-radionica tutorijal (11.02.2021.) <https://e-radionica.com/hr/blog/2015/08/19/kkm-ultrazvucni-modul-hc-sr04/>
- Ultrazvučni senzor HC-SR04 Datasheet (11.02.2021.) <https://e-radionica.com/productdata/HCSR04.pdf>
- Senzor opterećenja i pretvarač HX711 e-radionica tutorijal (11.02.2021.) <https://e-radionica.com/hr/blog/2019/04/08/kkm-hx711-load-cell/>
- AC/DC pretvarač HX711 Datasheet (11.02.2021.) [https://e-radionica.com/productdata/hx711\\_english.pdf](https://e-radionica.com/productdata/hx711_english.pdf)
- RFID Čitač MFRC522 e-radionica tutorijal (16.02.2021.) <https://e-radionica.com/hr/blog/2018/07/26/kkm-rfid-reader-mfrc-522/>
- RFID MFRC522 Čitač Datasheet (16.02.2021.) <https://e-radionica.com/productdata/MFRC522.pdf>
- Servo Motor TowerPro SG90 Datasheet (16.02.2021.) <https://www.e-radionica.com/productdata/SG90Servo.pdf>
- Piezoelektrična zujalica Datasheet (16.02.2021.) <https://www.e-radionica.com/productdata/1956696.pdf>
- HX711 Arduino Library (21.5.2021.) <https://github.com/bogde/HX711>
- MFRC-522 Arduino Library (21.5.2021.) <https://github.com/miguelbalboa/rfid>
- Arduino LCD Zaslon 16x2 (21.5.2021.) <https://create.arduino.cc/projecthub/akshayjoseph666/interface-16x2-lcd-parallel-interface-with-arduino-uno-2e87e2>



## POPIS TABLICA

- Tablica 1. Miješani komunalni otpad u RH prema podacima iz 2015. godine – Izvor: HAOP, projekt: „Izrada jedinstvene metodologije za analize sastava komunalnog otpada, određivanje prosječnog sastava komunalnog otpada u Republici Hrvatskoj i projekcija količina komunalnog otpada“
- Tablica 2. Električni parametri ultrazvučnog senzora HC-SR04 – Izvor: E-radionica Datasheet <https://e-radionica.com/productdata/HCSR04.pdf>
- Tablica 3. Opis priključaka na MFRC522 radio-frekvencijskom čitaču – Izvor: E-radionica Datasheet <https://e-radionica.com/productdata/MFRC522.pdf> (Autor prilagodio)
- Tablica 4. Specifikacije i dimenzije servo motora TowerPro SG90 – Izvor: E-radionica Datasheet <https://www.e-radionica.com/productdata/SG90Servo.pdf> (Autor prilagodio)
- Tablica 5. Priključni kontakti za servo motor TowerPro SG90 - Izvor: E-radionica Datasheet <https://www.e-radionica.com/productdata/SG90Servo.pdf> (Autor prilagodio)
- Tablica 6. Opis konektora na LCD zaslonu – Izvor: Componenst 101 Datasheet <https://components101.com/displays/16x2-lcd-pinout-datasheet> (Autor prilagodio)

## POPIS SLIKA

- Slika 1: Postupci korišteni za oporabu i zbrinjavanje ukupnog otpada u području RH prema podacima iz 2014. godine – Izvor: Plan Gospodarenja Otpadom Republike Hrvatske
- Slika 2: Prikaz udijela odvojenog otpada i mješanog sakupljenog komunalnog otpada u RH u razdoblju od 2010. do 2015. – Izvor: HAOP, projekt: „Izrada jedinstvene metodologije za analize sastava komunalnog otpada, određivanje prosječnog sastava komunalnog otpada u Republici Hrvatskoj i projekcija količina komunalnog otpada“
- Slika 3: Polja korištenja IoT tehnologije u konceptu pametnog grada – Izvor: AioT-Based Smart Bin for Real-Time Monitoring and Management of Solid Waste - Aniqa Bano, Ikram Ud Din, Asma A. Al-Huqail, Departmet of Information Technology, The University of Haripur, Pakistan, 2020.

- Slika 4: Arhitektura IoT sustava za gospodarenje otpadom – Izvor: AioT-Based Smart Bin for Real-Time Monitoring and Management of Solid Waste - Aniqā Bano, Ikram Ud Din, Asma A. Al-Huqail, Departmet of Information Technology, The University of Haripur, Pakistan, 2020.
- Slika 5: Primjer komunikacije između pametnih kanti i centralnog sustava – Izvor: AioT-Based Smart Bin for Real-Time Monitoring and Management of Solid Waste - Aniqā Bano, Ikram Ud Din, Asma A. Al-Huqail, Departmet of Information Technology, The University of Haripur, Pakistan, 2020.
- Slika 6: Blok dijagram osnovnih dijelova i podsustava pametne kante za smeće – Izvor: *The Design and Implementation of Smart Trash Bin* - Fady Samann, Duhok Polytechnic University, Iraq, Duhok, 2017.
- Slika 7: Princip komunikacije LoRaWan sustava s centralnom jedinicom sustava – Izvor: *Smart Waste Collection System with Low Consumption LoRaWAN Nodes and Route Optimization* – Alvaro Lozano, Javier Caridad, Juan Francisco De Paz, Gabriel Villarrubia, Javier Bajo, Faculty of Science, University of Salamanca, Spain, 2018.
- Slika 8: Grafička ilustracija sklopa – Izvor: Izradio autor
- Slika 9: Shematski prikaz sklopa – Izvor: Izradio autor
- Slika 10: JoyIT Arduino Mega 2560 – Izvor: Joy-It web stranica (08.06.2021.) <https://joy-it.net/en/products/ARD-Mega2560R3>
- Slika 11: Ultrazvučni senzor HC-SR04 – Izvor: E-radionica web stranica (08.06.2021.) <https://e-radionica.com/hr/ultrazvucni-modul-hc-sr04.html>
- Slika 12: Senzor opterećenja – 2 kg – Izvor: E-radionica web stranica (08.06.2021.) <https://e-radionica.com/hr/loadcell-senzor-2kg.html>
- Slika 13: A/D pretvarač HX711 – Izvor: E-radionica web stranica (08.06.2021.) <https://e-radionica.com/hr/adc-24-bit-s-pojacalom-hx711-made-by-e-radionica.html>
- Slika 14: Shema spajanja senzora opterećenja i A/D pretvarača HX711 - Izvor: E-radionica web stranica (08.06.2021.) <https://e-radionica.com/hr/blog/2019/04/08/kkm-hx711-load-cell/>



- Slika 15: Radio-frekvencijski čitač MFRC522 s karticom i privjeskom - Izvor: E-radionica web stranica (08.06.2021.) <https://e-radionica.com/hr/rfid-citac-mfrc-522-s-rfid-karticom.html>
- Slika 16: SPI priključci na Arduino AT Mega 2560 – Izvor: Nick Gammon (08.06.2021.) <http://www.gammon.com.au/spi>
- Slika 17: Servo motor TowerPro SG90 – Izvor: E-radionica web stranica (08.06.2021.) <https://e-radionica.com/hr/servo-motor-towerpro-sg90.html>
- Slika 18: LCD zaslon 16x2 – Izvor: RoboElements web stranica (08.06.2021.) <https://www.roboelements.com/product/%EF%BB%BF16x2-lcd-display-module-green-backlight/>
- Slika 19: Piezoelektrična zujalica – Izvor: PCBoard web stranica (08.06.2021.) <https://www.pcboard.ca/minipiezo-buzzer>
- Slika 20: 3D model pametne kante za smeće – Izvor: Izradio autor
- Slika 21: Kućni prototip pametne kante za smeće – Izvor: Izradio autor
- Slika 22: Pretinac za ubacivanje otpada – Izvor: Izradio autor
- Slika 23: Polica s komponentama – Izvor: Izradio autor
- Slika 24: Ultrazvučni senzori – Izvor: Izradio autor
- Slika 25: Vaga – Izvor: Izradio autor

## **DODATAK**

Dodatak rada čini potpuni primjerak kôda iz Arduino sučelja.

```
1. #include "HX711.h"
2. #include "SPI.h"
3. #include "MFRC522.h"
4.
5. //HX711 Pinovi
6. HX711 vaga;
7. float faktorKalibracije = -825000;
8. #define DAT 3 //DOUT
9. #define SCK 2 //CLK
10.
11. //RFID Pinovi
12. #define RFID_MISO 50
13. #define RFID_MOSI 51
14. #define RFID_SCK 52
15. #define RFID_SDA 53 //Set State
16. #define RFID_RST 49 //Reset pin
17.
18. MFRC522 mfrc522(RFID_SDA, RFID_RST);
19. String read_rfid;
20. String rfid_card = "d5543e11"; //kod naše kartice, dobiveno prethodno
21. String rfid_tag = "3b2ead"; //kod našeg taga, dobiveno prethodno
22.
23.
24. #include <Servo.h>
25. Servo servo;
26. #define servoPin 5 //Servo PWM
27.
28. #include <LiquidCrystal.h>
29. //Register Select Pin = 32
30. //Enable Pin = 30
31. //DB4 Pin = 28
32. //DB5 Pin = 26
33. //DB6 Pin = 24
34. //DB7 Pin = 22
35. LiquidCrystal lcd(32, 30, 28, 26, 24, 22);
36.
37. //Senzor 1
38. #define trigPin1 25
39. #define echoPin1 27
40. long vrijeme1;
41. float senzorUdaljenost1;
42. float realnaUdaljenost1;
43.
44. //Senzor 2
45. #define trigPin2 29
46. #define echoPin2 31
47. long vrijeme2;
48. float senzorUdaljenost2;
49. float realnaUdaljenost2;
50.
51. int a_strana = 26; //Kanta, dimenzije u centimetrima
52. int b_strana = 15;
53. float visinaKante = 12; //cm
54. float volumenKante = a_strana*b_strana*visinaKante;
```

```

55.
56. float volumen;
57. float masa;
58.
59. float razmakKantaSenzor = 3; //cm
60. float srednjaUdaljenost;
61. int postotak;
62. //int površina = a_strana*b_strana;
63.
64. #define zujalica 4 //Buzzer Pin
65. int pretStanje;
66.
67. int otvorenoVrijeme;
68. int servoVrijeme = 1000;
69. int servoPozicija;
70.
71. //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
72. //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
73.
74. void setup()
75. {
76.   pinMode(trigPin1, OUTPUT); //Senzor 1
77.   pinMode(echoPin1, INPUT);
78.   pinMode(trigPin2, OUTPUT); //Senzor 2
79.   pinMode(echoPin2, INPUT);
80.   pinMode(zujalica, OUTPUT);
81.   servo.attach(servoPin); //Servo 1
82.   servo.write(0);
83.
84.   lcd.begin(16, 2); //LCD
85.   lcd.clear();
86.   lcd.print("Ucitavanje...");
87.
88.   Serial.begin(9600);
89.
90.   vaga.begin(DAT, SCK);
91.   vaga.set_scale();
92.   vaga.tare(); //Reset the scale to 0
93.   long zero_factor = vaga.read_average(); //Osnovno očitavanje
94.   //Serial.print("Zero factor: "); // Komentar ako je potrebno očitati stanja u serial monitoru
95.   //Serial.println(zero_factor); // Komentar ako je potrebno očitati stanja u serial monitoru
96.
97.   while (!Serial);
98.   SPI.begin();
99.   mfrc522.PCD_Init();
100.
101.   pretStanje=1;
102. }
103.
104. //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
105. //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
106.
107. void dump_byte_array(byte *buffer, byte bufferSize)
108. {
109.   read_rfid="";
110.   for (byte i = 0; i < bufferSize; i++) {
111.     read_rfid += String(buffer[i], HEX);
112.   }
113. }

```

```

114.
115. //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
116. //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
117.
118. void dodatneInformacije()
119. {
120.   lcd.clear();
121.   lcd.print("Masa: ");
122.   lcd.print(masa);
123.   lcd.print(" kgs");
124.   lcd.setCursor(0, 1);
125.   lcd.print("Vol: ");
126.   lcd.print(volumen);
127.   lcd.print(" cm3");
128.   delay(10000);
129.   pretStanje=1;
130. }
131.
132. //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
133. //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
134.
135. void visoko()
136. {
137.   tone(zujalica,1000, 1000);
138.   delay(100);
139.   tone(zujalica,500, 500);
140.   delay(100);
141.   noTone(zujalica);
142.
143.   lcd.clear();
144.   lcd.print("Otvaranje...");
145.   delay(2000);
146.
147.   for(servoPozicija=0; servoPozicija<=140; servoPozicija++){ //Stupnjevito otvaranje servo
motora
148.     servo.write(servoPozicija);
149.     delay(servoVrijeme/140);
150.   }
151.
152.   for(otvorenoVrijeme=15; otvorenoVrijeme>=0; otvorenoVrijeme--){
153.     lcd.clear();
154.     lcd.print("Otvoreno");
155.     lcd.setCursor(0, 1);
156.     lcd.print(otvorenoVrijeme);
157.     delay(1000);
158.   }
159.
160.   lcd.clear();
161.   lcd.print("Zatvoreno");
162.   lcd.setCursor(0, 1);
163.   lcd.print("Stanje: ");
164.   lcd.print(postotak);
165.   lcd.print("%");
166.
167.   lcd.clear();
168.   lcd.print("Zatvaranje...");
169.
170.   tone(zujalica,2000, 100);
171.   delay(100);
172.   tone(zujalica,2000, 100

```

```

173. delay(100);
174. noTone(zujalica);
175. delay(1000);
176.
177. for(servoPozicija=140; servoPozicija>=0; servoPozicija--){ //Stupnjevito zatvaranja servo
    motora
178. servo.write(servoPozicija);
179. delay(servoVrijeme/140);
180. }
181.
182. delay(500);
183. lcd.clear();
184. lcd.print("Zatvoreno");
185. pretStanje=1;
186. }
187.
188. //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
189. //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
190.
191. void nisko()
192. {
193. //Nakon sto je sustav zatvoren ide procjena zapremnine
194. // Udaljenost // zvuk (cm/us) //dijelim 2, dupla udaljenost
195.
196. //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
197.
198. digitalWrite(trigPin1, LOW); //Senzor1
199. delayMicroseconds(2);
200. digitalWrite(trigPin1, HIGH);
201. delayMicroseconds(10);
202. digitalWrite(trigPin1, LOW);
203. vrijeme1 = pulseIn(echoPin1, HIGH);
204. senzorUdaljenost1= vrijeme1*0.034/2;
205. realnaUdaljenost1=senzorUdaljenost1-razmakKantaSenzor;
206.
207. delay(100);
208.
209. //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
210.
211. digitalWrite(trigPin2, LOW); //Senzor2
212. delayMicroseconds(2);
213. digitalWrite(trigPin2, HIGH);
214. delayMicroseconds(10);
215. digitalWrite(trigPin2, LOW);
216. vrijeme2 = pulseIn(echoPin2, HIGH);
217. senzorUdaljenost2= vrijeme2*0.034/2;
218. realnaUdaljenost2=senzorUdaljenost2-razmakKantaSenzor;
219.
220. //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
221.
222. srednjaUdaljenost=(realnaUdaljenost1+realnaUdaljenost2)/2;
223. postotak=((visinaKante-srednjaUdaljenost)/visinaKante)*100;
224.
225. lcd.clear();
226. lcd.print("Zatvoreno");
227. lcd.setCursor(0, 1);
228. lcd.print("Stanje: ");
229. lcd.print(postotak);
230. lcd.print("%");

```

```

231.
232. vaga.set_scale(faktorKalibracije); //Prilagoditi faktor kalibracije po potrebi
233. masa=vaga.get_units(),3;
234. Serial.print("Ocitanje: ");
235. Serial.print(masa);
236. Serial.print(" kgs");
237. Serial.print(" faktor kalibracije: ");
238. Serial.print(faktorKalibracije);
239. Serial.println();
240.
241. Serial.print("Udaljenost: ");
242. Serial.print(srednjaUdaljenost);
243. Serial.println();
244. Serial.print("Postotak: ");
245. Serial.print(postotak);
246. Serial.println();
247. Serial.print("Senzor 1: ");
248. Serial.print(senzorUdaljenost1);
249. Serial.println();
250. Serial.print("Senzor 2: ");
251. Serial.print(senzorUdaljenost2);
252. Serial.println();
253.
254. volumen=(a_strana*b_strana*srednjaUdaljenost);
255.
256. pretStanje=0;
257. }
258.
259. //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
260. //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
261.
262. void puno()
263. {
264. lcd.clear();
265. lcd.print("Zatvoreno");
266. lcd.setCursor(0, 1);
267. lcd.print("Stanje: PUNO");
268. pretStanje=2;
269.
270. }
271.
272. //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
273. //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
274.
275. void loop()
276. {
277.
278. if (pretStanje==1){
279. nisko();
280. }
281.
282. if(masa>=2){
283. puno();
284. }
285.
286. if(volumen<(volumenKante*0,05)){
287. puno();
288. }

```

```
289.
290. if ( ! mfrc522.PICC_IsNewCardPresent()){
291. return;
292. }
293.
294. if ( ! mfrc522.PICC_ReadCardSerial()){
295. return;
296. }
297.
298. dump_byte_array(mfrc522.uid.uidByte, mfrc522.uid.size);
299.
300. Serial.println(read_rfid);
301.
302. if (pretStanje==1 && read_rfid == rfid_card){
303. visoko();
304. }
305.
306. if (pretStanje==0 && read_rfid == rfid_card){
307. visoko();
308. }
309.
310. if (read_rfid == rfid_tag){
311. dodatneInformacije();
312. }
313. }
```



## **SAŽETAK**

Gospodarenje otpadom postaje sve veći problem modernog društva. Problem gospodarenja otpadom započinje od čovjeka koji stvara otpad svojim svakodnevnim potrebama i sve rastućom industrijom. Zaostale i stare metode koje se koriste za gospodarenje otpadom nalaze se u sve većem koštaku kako bi savladale potrebe društva.

Istraživanje čini dio ovog rada te se bazira se promatranju suvremenih metoda gospodarenja otpadom, pronalaskom njihovih mana i zapreka. Kroz samo istraživanje traže se ideje na koje sve načine približiti postojeće sustave u jedan cjeloviti IoT sustav koji bi imao potpunu kontrolu informacija za gospodarenje otpadom.

Sam takav sustav sastojao bi se od upravljačkog centra te zatim niza manje kompleksnih ugrađenih sustava koji imaju svoje zadaće, poput mjerenja količine otpada, GPS lokacija spremnika i vozila, te komunikacija između svih sustava međusobno.

Analizom podataka dobivenih istraživanje omogućuje stvoriti koncept i prototip podsustava zvan „Pametna kanta za smeće“ koja je jedan u nizu dijelova potpunog sustava gospodarenja otpadom. Takva kanta za smeće imala bi jednostavnu identifikaciju korisnika i interaktivno sučelje koje je lako i intuitivno za korištenje, uz to bi centralnom sustavu pružala sve informacije o pohranjenom otpadu, lokaciji i te korisniku koji je koristi.

**Ključne riječi:** Pametna kanta za smeće, ugrađeni sustavi, gospodarenje otpadom, mikro upravljač, ultrazvučni senzori, senzor opterećenja, servo motor

## **ABSTRACT**

Waste management is becoming an increasing problem of modern society. The problem of waste management starts with the man who generates waste for his daily needs and the growing industry. Backward and old methods used for waste management are increasingly struggling to meet the needs of society.

The research is part of this paper and is based on the observation of modern methods of waste management, finding their shortcomings and obstacles. Through the research itself, ideas are sought in all ways to bring existing systems closer into one comprehensive IoT system that would have complete control of information for waste management.

Such a system itself would consist of a control center and then a series of less complex embedded systems that have their own tasks, such as measuring the amount of waste, GPS locations of tanks and vehicles, and communication between all systems with each other.

By analyzing the data obtained, the research enables the creation of a concept and prototype of a subsystem called "Smart Garbage Bin", which is one in a series of parts of a complete waste management system. Such a garbage bin would have a simple user identification and an interactive interface that is easy and intuitive to use, in addition to providing the central system with all the information about the stored waste, the location and the user who uses it.

**Keywords:** Smart garbage bin, embedded systems, waste management, micro controller, ultrasonic sensors, load sensor, servo motor