

**Elektronsko poslovanje**

**“Pametna kancelarija-Pametna kancelarijska kanta”**

Neskovic Stefan 225/2016

Sadrzaj

[Opis proizvoda 3](#_Toc534142677)

[Izgled proizvoda 4](#_Toc534142678)

[Osnovni elementi proizvoda 5](#_Toc534142679)

[Arduino Uno 5](#_Toc534142680)

[Senzor za detekciju razdaljine (ultra sonic) 5](#_Toc534142681)

[Servo motor 5](#_Toc534142682)

[Motor 360 o 6](#_Toc534142683)

[Dodatni delovi 6](#_Toc534142684)

[Impementacija u programskom jeziku C 7](#_Toc534142685)

[Literatura 9](#_Toc534142686)

# Opis proizvoda

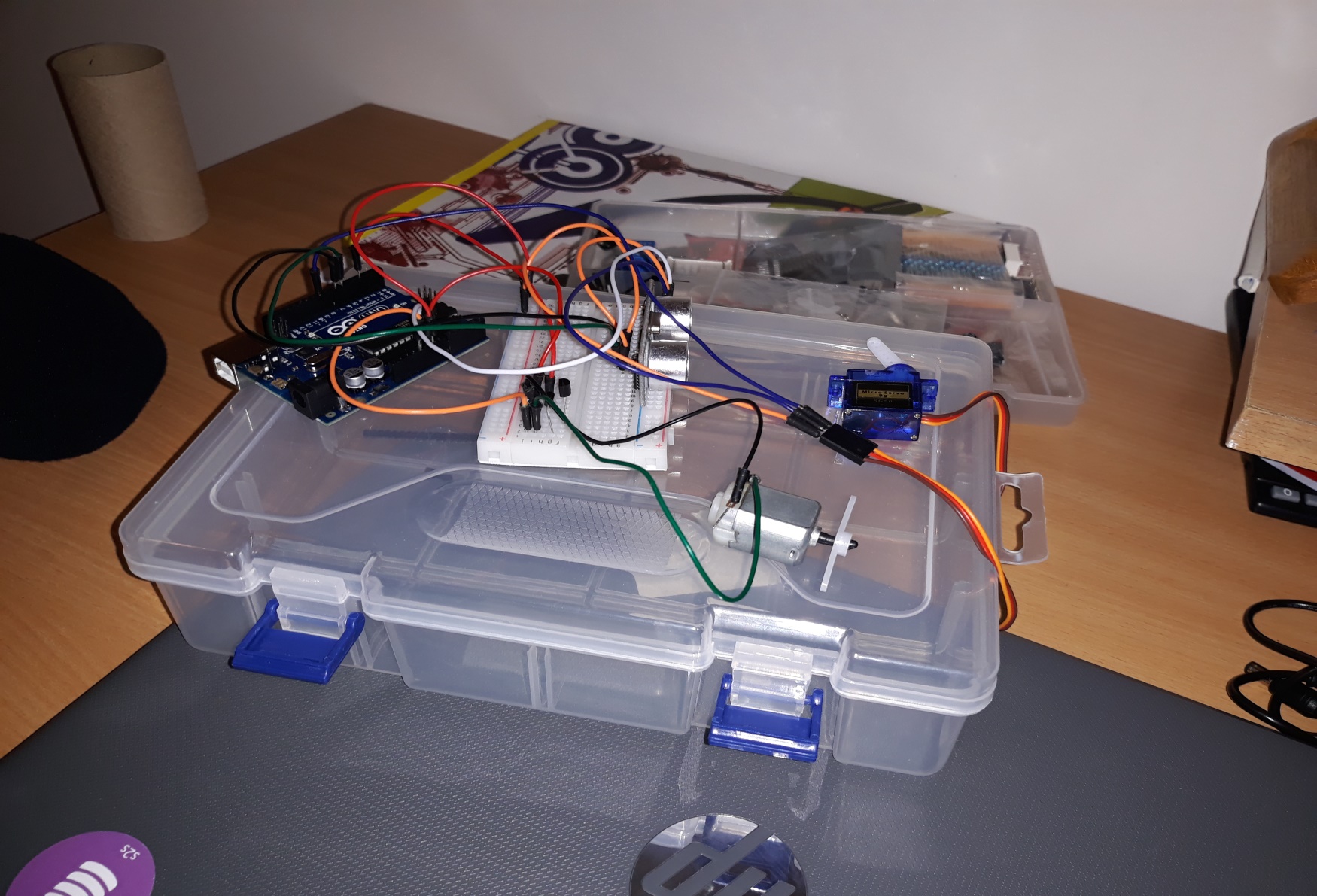
U ovom projektu trudio sam se da na sto inovativniji nacin uptrebim, danas jednu od najzanimljivijih oblasti IT, a to je IoT ( Internet of Things ). Rec je o veoma zanimljivoj grani racunarske industrije kojom je omoguceno olaksavanje zivota I rada za vecinu ljudi na nasoj planeti. Neretko se srecu problem sadasnjice koji se na zanimljiv nacin mogu softverski iterpretirati. Ovaj mali izum je jedanod njih.

Naime, u ovom projektu povezao sam stvar automatike, ekologije I bezbednosti u jednom. Takva mesavina naucnih disciplina je kljuc uspeha u svetu IoT. Kako bih Vam na sto verniji nacin prikazao rad I strukturu projekta prikazujem sliku u nastavku teksta. Na njoj mozemo videti tri komponente :

* Senzor za detekciju blizine
* Servo motor za otvaranje kante
* Motor za pokretanje seciva

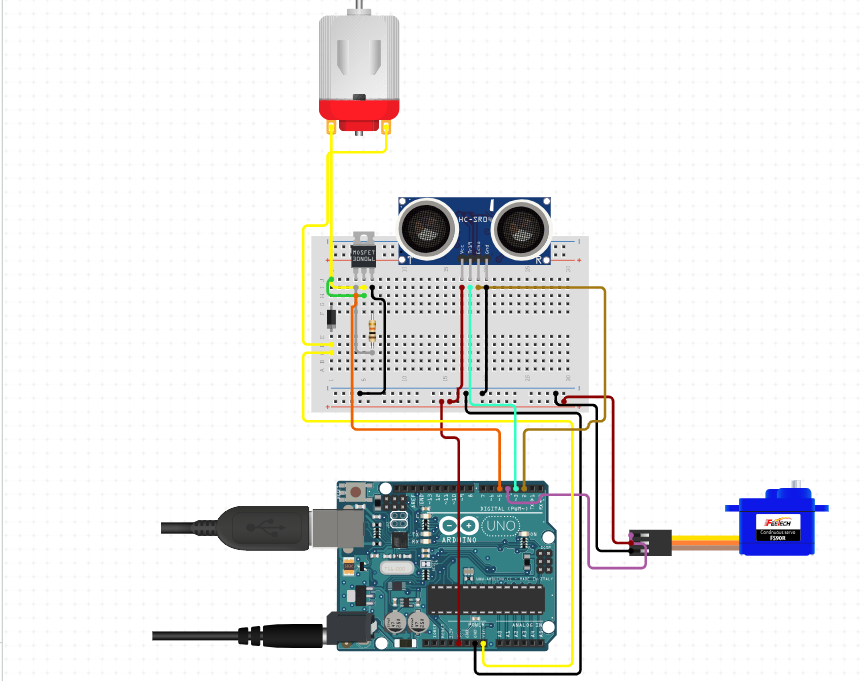
Ceo process rada zasniva se na tome da sensor za blizinu odredi razdaljinu predmeta I ukoliko se on nalazi na razdaljini kracoj od dozvoljene pokrece rad servo motora za podizanje poklopca. Ovime smo postigli automatizaciju dva procesa, a sto je najvaznije omogucili smo ljudima da ne prljaju ruke I ne naprezu se kako bi otklonili nepotrebni kancelarijski materijal. Ovaj process radi neprestano dokle god se ne iskljuci. Medjutim na svakih pola sata aktivira se motor za rotiranje seciva kojima se sece I presuje sav sadrzaj kante. Kao to se na slici moze primetiti, tu postoji I jos jedan deo ispod seciva a to je odeljak za skupljanje tog otpada koji se na adekvatan I pre svega lagan nacin moze odstraniti. Ovaj process traje svega trideset sekundi sto se ispostavilo kao veoma dovoljno da se sav material isece. Kao vrhunac ovog proizvoda treba napomenuti da dokle god se seciva okrecu mehanizam za otvaranje poklopca ostaje zamrznut I onemogucen da na negativan nacin ugrzi bezbednost nasih korisnika.

# Izgled proizvoda



Kao sto se sa slike moze zakljuciti ovo je samo sturi prikaz structure samog modela uredjaja koji sam projektovao. Na njemu mozemo uociti vise razlicitih komponenti koje su medjusobno povezane I cine jedno veliko strujno kolo. U okviru ovog kola razlikuju se dva manja, zasebna ali programski veoma dobro povezana. Rec je o :

* Kolo za prkupljanje podataka sa glavnog sensor
* Kolo za pokretanje motora za otvaranje kante
* Kolo za pokretanje nozeva namenjenih za secenje papirnog sadrzaja

Pocnimo od prvog, mozda I najznacajnijeg za nas. Rec je osenzoru za merenje udaljenosti predmeta od naseg uredjaja. Njegova funkcija je da prepozna osobe I time signalizira raunaru da je vreme za pokretanje drugog po redu strujnog kola. Nacin rada I strukturu ovog senzora objasnicu u narednimm odeljcima.

Naredni deo je strujno kolo u kome je definisan rad servo motora koji ima namenu da omoguci otvaranje poklopca pametne kante I time omoguci osobi da ne prlja ruke I da na veoma lak nacin odstrani visak koji je planirao da baci.

Na kraju mozda I jedan od najzanimljivijih delova ovog uredjaja a to je secko, motor koji pokrece nozeve za secenje nepotrebnog materijala iz kante. On je povezan na Arduino Uno plocu kao zasebno strujno kolo koje se pokrece u zavisnosti od vremenskog perioda koji mu je programmer podesio. U ovom sucaju to sam uradio ja podesivsi mu vreme odziva na trideset minu I rad od nekih trideset sekundi.

# Osnovni elementi proizvoda

## Arduino Uno

Platforma kojom sam izradjivao ovaj uredjaj bazirao se na konceptima Arduino Uno generacije, gde je kod bio zapisan u C programskom jeziku u Arduino IDE okruzenju. Za ovaj vid programiranja sam se odlucio iskljucivo zbog njene jednostavnosti u primeni, kao I mogucnosti implementacije u razlicitim programskim jezicima.

## Senzor za detekciju razdaljine (ultra sonic)

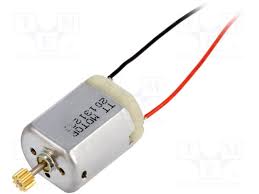
Kao sto im i samo ime govori, ultrazvucni senzori daljine se koriste za merenje rastojanja između senzora i objekta koji se nalazi ispred njega. Ovi senzori nude dobru preciznost i jednostavni su za upotrebu. Ovde sam koristio senzor **HC-SR04**. Rezolucija senzora je 0,3cm, a opseg detektovanja objekta od 2 do 460cm. Senzor pokriva horizontalni ugao do 15o. Snaga njegovog napajanja je 5V.

## Servo motor



Ovi motori rade po principu DC sto im omogucava da im odredjujete rad ispravkom svog koda napisanom u Arduino okruzenju. Ovaj motor ima opseg rada od 180o sto mu omogucava da radi neku iteraciju iskljucivo u tom opsegu. U ovom primeru to je sazeto na ugao od 70o i time omogucio rad mehanizma za otvaranje poklopca. Sam uredjaj radi na naponu od 5V.

## Motor 360 o



Pisanje koda za ovaj uredjaj je nesto lakse nego za prethodni motor jer je njemu predefinisano da radi jedan process okretanja elise, bez definisnja ugla pod kojim rani. Naime, njemu je omoguceno da radi u dva smera u zavisnosti kako je on povezan na strujno kolo I to sve u beskonacnost dokle god se na kolo ne utice da prestane napajanje. Za napajanje ovog motora neophodno je obezbediti napon od ne vise od 5V I ne manje od 3,3V.

## Dodatni delovi

Od dodatni delova koje sam koristio su neke od uobicajenih delova a to su :

* Ubodna ploca
* Kablovi
* Tranzistori
* Diode

# Impementacija u programskom jeziku C

Kao sto sam vec naveo svoj program sam radio u okruzenju Arduino IDE koji podrzava programski jezik C, pa sam se odlucio ipak za ovaj programski jezik, mada sam sve ovo mogao implementirati I u drugim jezicima. U ovom odeljku bice detaljno objasnjen svaki korak kako bi I svaki pocetnik mogao da razume moj koncept rada .

Za pocetak cemo uvesti biblioteke koje ce nam biti neophodne za rad mada pri prvom pokretanju naseg programa bice automatski referencirani ova dva reda naseg koda, ali mi cemo t uradi postupno.

#include <Servo.h> Servo myservo;

int trigPin = 6;

int echoPin = 7; ***Ovim fragmentom koda definisemo globalne promenljive***

int motorPin= 3; ***koje cemo kasnije koristiti u nasem programu***

int servoPin=9;

long cm; ***Uvedimo sada promenljive koje ce nam biti potrebne***

int poz; ***u ostatku programa***

U svakom kodu koji je implementiran za Arduino uredaje je neophodno definisati dve osnovne metode I to setup I loop. U narednom odeljku predstavicu nas setup kod u kome definisemo nase izlazne I ulazne pinove sa kojima kontrolisemo nasu Arduino plocu. Kao sto vidimo to su pinovi koje smoranije definisali kao globalne promenljive, pa necemo zalaziti sada u detalje koji su to brojevi. Funkcijom **setup** definisemo sve one ulazno- izlazne pinove sa kojima mozemo da kontrolisemo nasu plocu.

**void setup() {**

**Serial.begin (9600);**

**myservo.attach(servoPin);**

**pinMode(trigPin, OUTPUT);**

**pinMode(echoPin, INPUT);**

**pinMode(motorPin,OUTPUT);**

**}**

Kao sto smo napisali metodu setup isto cemo uraditii za metodu **loop**. Ova funkcija ce se koristiti za definisanje onoga sto cenas program izvrsavati. Pa da krenemo sa pisanjem.

void loop() {

int j=0; ***Uvedimo za pocetak beskonacnu petlju kojom***

***omogucavamo da nas program radi zauvek***

for(j=0;j>=0;j++){

if(j%20==0){  ***Definisimo if petljom da nam se dva procesa*** ***odvajaju za 20 iteracija***

analogWrite(motorPin,1000);

delay(5000); ***Kod za pokretanje seciva***

analogWrite(motorPin,0);

}else{

digitalWrite(trigPin, LOW);

delayMicroseconds(5);

digitalWrite(trigPin, HIGH); ***Prikupimo podatke sa naseg senzora ovim***

delayMicroseconds(10); ***fragmentom***

digitalWrite(trigPin, LOW);

pinMode(echoPin, INPUT);

duration = pulseIn(echoPin, HIGH); ***Sve prikupljene podatke cemo cuvati u promenljive***

cm = (duration/2) / 29.1; ***ali prevodimo ih u cm ovom formulom***

if(cm<30){ ***Proveravamo razdaljinu od uredjaja, ukoliko je manja od 30 cm pokrecemo motor za poklopac***

for (poz = 15; poz <= 90; poz += 1) {

myservo.write(poz); ***Ovim fragmentom definisemo ugao I brzinu***

delay(15);  ***poretanja motora, u ovom slucaju je to:***

} ***1) Od 15 do 90 stepena I nazad***

delay(1000); ***2) Svaka iteracija ima pauzu od***

for (poz = 90; poz >= 15; poz -= 1) { ***15 milisekundi***

myservo.write(poz);

delay(15);

}

}else{

continue; ***Ukoliko nije ispunjen uslov samo preskacemo***

}

}

delay(1000); ***Svaka iteracija glavne petlje trajace 1000 milisek.***

}

delay(250); ***Pauza pri pokretanju programa je 250 milisek.***

}

Ovim kodom omogucavamo poretanje I kontrolu naseg gore navedenog proizvoda. Naime, u ovom kodu ja sa podesio kao testiranu metodu sa nesto kracim vremenskim periodom koji Vi mozete veoma lako promeni u period koji smo ranije ustanovili, ili u neki po vasem izboru.

# Literatura

Arduino official site [***www.arduino.cc***](http://www.arduino.cc)

Arduino PDF knjige:[***www.pdfknjige.net***](http://www.pdfknjige.net)

[Arduino uno: ***45 projekata za početnike i stručnjake***](http://www.mikroknjiga.rs/store/prikaz.php?ref=978-86-80134-07-9)