

Upravljanje mikrokontrolerom pomoću aplikacije izrađene u MIT App Inventor razvojnom okruženju

Puškaric, Marko

Undergraduate thesis / Završni rad

2023

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Virovitica University of Applied Sciences / Veleučilište u Virovitici**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:165:994869>

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

Download date / Datum preuzimanja: **2024-09-08**

Repository / Repozitorij:



[Virovitica University of Applied Sciences Repository - Virovitica University of Applied Sciences Academic Repository](#)



VELEUČILIŠTE U VIROVITICI

Stručni prijediplomski stručni studij Elektrotehnika

MARKO PUŠKARIĆ

UPRAVLJANJE MIKROKONTROLEROM POMOĆU APLIKACIJE
IZRAĐENE U MIT APP INVENTOR RAZVOJNOM OKRUŽENJU

ZAVRŠNI RAD

VIROVITICA, 2023

VELEUČILIŠTE U VIROVITICI

Stručni prijediplomski stručni studij Elektrotehnika

UPRAVLJANJE MIKROKONTROLEROM POMOĆU APLIKACIJE
IZRAĐENE U MIT APP INVENTOR RAZVOJNOM OKRUŽENJU

ZAVRŠNI RAD

Predmet: Digitalna elektronika

Mentor:

Ivan Heđi, dipl. ing., v. pred.

Student:

Marko Puškarić

VIROVITICA, 2023



OBRAZAC 1b

ZADATAK ZAVRŠNOG RADA

Student/ica: MARKO PUŠKARIĆ JMBAG: 0307017431

Imenovani mentor: Ivan Hedi, dipl. ing., v. pred.

Imenovani komentor: -

Naslov rada:

Upravljanje mikrokontrolerom pomoću aplikacije izrađene u MIT App Inventor razvojnom okruženju

Puni tekst zadatka završnog rada:

Koristeći današnje dostupne tehnologije moguće je automatizirati svakodnevne procese. Vaš zadatak je istražiti mogućnosti korištenja jedne IoT platforme koja će omogućiti otvaranje i zatvaranje ograde i garažnih vrata koristeći mobilni uređaj. Kao mikrokontroler koristite ESP8266. Za izradu mobilne aplikacije koristite MIT App Inventor. Predložene tehnologije verificirati uz pomoć makete.

Datum uručenja zadatka studentu/ici: 31.07.2023.

Rok za predaju gotovog rada: 08.09.2023.

Mentor:

Ivan Hedi, dipl. ing., v. pred.

Dostaviti:

1. Studentu/ici
2. Povjerenstvu za završni rad - tajniku

**UPRAVLJANJE MIKROKONTROLEROM POMOĆU APLIKACIJE
IZRAĐENE U MIT APP INVENTOR RAZVOJNOM OKRUŽENJU**

Sažetak

U poglavljima ovog rada navedene su i opisane sve tehnologije i elektroničke komponente za izradu mobilne aplikacije koristeći MIT App Inventor razvojnog okruženja i makete autonomnog sustava kojim će mobilna aplikacija upravljati. Na početku rada upoznat će se sa tehnologijama i elektroničkim komponentama koje su korištene za realizaciju samog autonomnog sustava. Nadalje govorit će se o samom okruženju za izradu android mobilne aplikacije za upravljanje ogradom i garažnim vratima. Na praktičnom dijelu zadatka biti će naveden postupak izrade električne sheme sustava te postupak izrade makete od akrilnog stakla. Na samom kraju biti će opisani koraci izrade i sastavljanja tiskane pločice na kojoj se nalazi ESP8266 NodeMCU mikrokontroler i objašnjenje najvažnijih dijelova programskog koda.

Ključne riječi: *ESP8266 NodeMCU mikrokontroler, Firebase, Infracrveni senzor Iduino ST1081, MIT App Inventor, Servo motor TowerPro MG995*

**CONTROLLING THE MICROCONTROLLER USING APPLICATION
CREATED IN THE MIT APP INVENTOR DEVELOPMENT ENVIRONMENT**

Abstract

In chapters of this Bachelor's Thesis are listed all technologies and electronic components for making mobile application using MIT App Inventor development environment and physical model of autonomous system that mobile application will control. At the very beginning of Bachelor's Thesis, we are introduced with technologies and electronic component that were used to realize autonomous system. We will also talk about the creation of android mobile application for controlling fence and garage door. Practical part of this task will include a procedure for creating an electric diagram of the autonomous system and steps for creating a physical model made from acrylic glass. At the very end, we will describe steps for creating and assembling the printed circuit board on which ESP8266 NodeMCU microcontroller is mounted and describe the most important parts of program code.

Keywords: *ESP8266 NodeMCU Microcontroller, Firebase, Iduino ST1081 Infrared Sensor, MIT App Inventor, TowerPro MG995 Servo Motor*

Sadržaj

1.	Uvod.....	1
2.	Primijenjene tehnologije u radu.....	2
2.1.	ESP8266 NodeMCU mikrokontroler	2
2.2.	Firestore baza podataka	4
2.3.	Arduino IDE razvojno okruženje.....	5
2.4.	MIT App Inventor	6
2.4.1.	Uređivač dizajna.....	7
2.4.2.	Uređivač blokova.....	8
2.5.	Infracrveni senzor Iduino ST1081.....	9
2.6.	Servo motor TowerPro MG995, 360 stupnjeva	10
2.7.	Funkcijski dijagram sustava.....	11
2.8.	Električna shema sustava	12
2.9.	Realizacija funkcionalne makete.....	13
2.9.1.	Provjera rada elektroničkih komponenti.....	13
2.9.2.	Izrada programskog koda	14
2.9.3.	Testiranje rada programskog koda i elektroničkih komponenti.....	16
2.9.4.	Tiskana pločica	16
2.9.5.	Maketa.....	16
2.9.6.	Mobilna aplikacija za upravljanje.....	17
2.9.7.	Verificiranje makete	19
3.	Zaključak.....	20
	Popis literature.....	21
	Popis slika.....	23

1. Uvod

U današnje vrijeme korištenjem različitih dostupnih tehnologija moguće je automatizirati svakodnevne procese ljudskog života. Jedan od takvih primjera je mogućnost korištenja različitih Internet stvari (engl. *Internet of Things*) platformi za otvaranje i zatvaranje ograde i dvorišnih vrata koristeći mobilnu aplikaciju. Stoga će se u nadolazećim poglavljima rada uz pomoć danas nama dostupnih tehnologija automatizirati proces otvaranja i zatvaranja dvorišnih i garažnih vrata pomoću mobilne aplikacije i senzora kako bi olakšali ljudski život i umanjili nepotrebni ljudski naponi za obavljanje tog svakodnevnog procesa.

Proces automatizacije izvršavat će se pomoću ESP8266 NodeMCU mikrokontrolera koji pomoću infracrvenih senzora prikuplja podatke o tome nalazi li se objekt ispred ograde i garažnih vrata. Također će ESP8266 NodeMCU mikrokontroler čitati vrijednosti iz *Firebase* baze podataka u koju će se spremati podatci iz mobilne aplikacije za otvaranje i zatvaranje dvorišnih i garažnih vrata. Na temelju tih prikupljenih podataka i podataka koji su zapisani u *Firebase* bazi podataka izvršavat će se akcije otvaranja i zatvaranja dvorišnih i garažnih vrata po potrebi.

U radu će biti navedene sve tehnologije koje su korištene za izradu same aplikacije i za skladištenje podataka same aplikacije, kao i same elektroničke komponente koje se koriste za prikupljanje i obradu podataka te pokretanje određenih akcija.

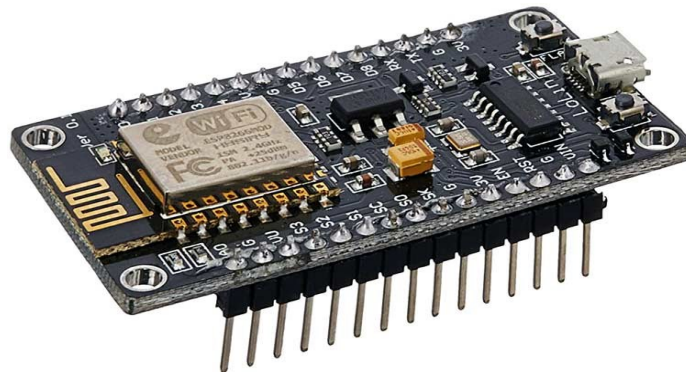
2. Primijenjene tehnologije u radu

U nadolazećim poglavljima biti će navedene razne tehnologije i elektroničke komponente koje su korištene u samom procesu automatizacije otvaranja i zatvaranja dvorišnih i garažnih vrata. U nastavku će biti spomenuti i objašnjen rad elektroničkih komponenti ESP8266 NodeMCU mikrokontrolera, infracrvenog senzora Iduino ST1081, servo motora TowerPro MG995 te tehnologije kao što su *Firestore* baza podataka i MIT (engl. *Massachusetts Institute of Technology*) *App Inventor*.

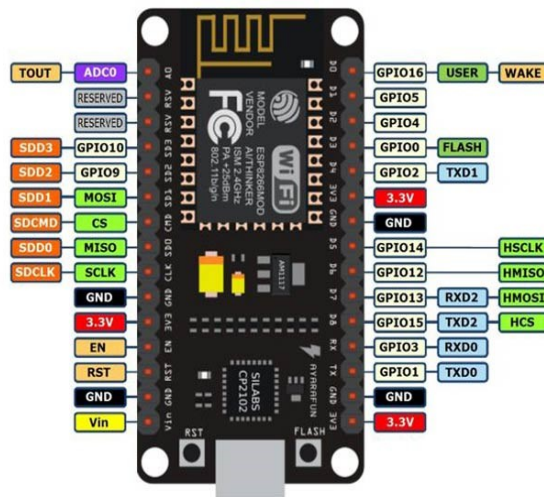
2.1. ESP8266 NodeMCU mikrokontroler

ESP8266 NodeMCU je 32-bitni mikrokontroler kojeg je razvila tvrtka *Espressif Systems*, koji je zbog svoje niske cijene, niske potrošnje električne energije i sposobnosti povezivanja na Wi-Fi mrežu postao nezaobilazan u mnogim Internet stvari projektima kao što su automatizacija domova, pametni uređaji, vremenske stanice i tako dalje. Pristup internetu ima pomoću svog ESP8266 Wi-Fi mikročipa, koji ima u sebi ugrađeni TCP/IP mrežni softver pomoću kojeg može pristupiti mrežnim uslugama da bi slao HTTP zahtjeve ili za spremanje podataka u oblak ili za komunikaciju sa drugim uređajima i serverima. Na slici 1 se može vidjeti izgled mikrokontrolera ESP8266 NodeMCU. Mikrokontroler je također postao popularan jer je kompatibilan sa Arduinom, odnosno sa Arduino IDE softverom pomoću ESP8266 paketa pločica čime postaje pristupačan velikom broju ljudi. Zbog svoje velike popularnosti moguće je pronaći detaljnu i opsežnu dokumentaciju, projekte ostalih korisnika te veliki broj knjiga sa raznim primjerima. Mikrokontroler raspolaže s velikim brojem ulazno/izlaznih nožica opće namjene (engl. *General Purpose Input/Output*) ili skraćeno GPIO koje služe za povezivanje mikrokontrolera s raznim sensorima i mnogim drugim uređajima te s plavom LED (engl. *Light-emitting diode*) lampicom koja je interno spojena na GPIO02 što ga čini veoma praktičnim za praćenje tijeka samog koda ili za otkrivanje pogrešaka u radu. Raspored GPIO nožica, kao i raspored drugih nožica mikrokontrolera, se može vidjeti na slici 2. Mikrokontroler na sebi ima i dva gumba sa natpisima RST i FLASH, koji služe za ponovno pokretanje mikrokontrolera ili za postavljanje mikrokontrolera u engl. *flashing mode* tijekom kojeg se *firmware* mikrokontrolera može ažurirati ili u potpunosti promijeniti. Mikrokontroler ima ulazni napon od 4.4 V do 10 V te također na sebi ima stabilizator napajanja od 3.3 V koji se može koristiti za napajanje drugih elektroničkih komponenti kao što su senzori. Prijenos koda na pločicu je veoma jednostavan jer sama pločica ima ugrađen USB (engl. *Universal serial*

bus)-na-serijski (engl. *USB-to-serial*) pretvarač, čime se smanjuje potreba za FTDI (engl. *Future Technology Devices International Limited*) programom ili dodatnim strujnim krugovima. Jedna od najvećih mana ESP8266 NodeMCU mikrokontrolera je njegova mala količina memorije i njegova brzina procesiranja. Mikrokontroler raspolaže sa svega 80 *KiB* memorije za korisničke podatke i s 32 *KiB* memorije za instrukcije.



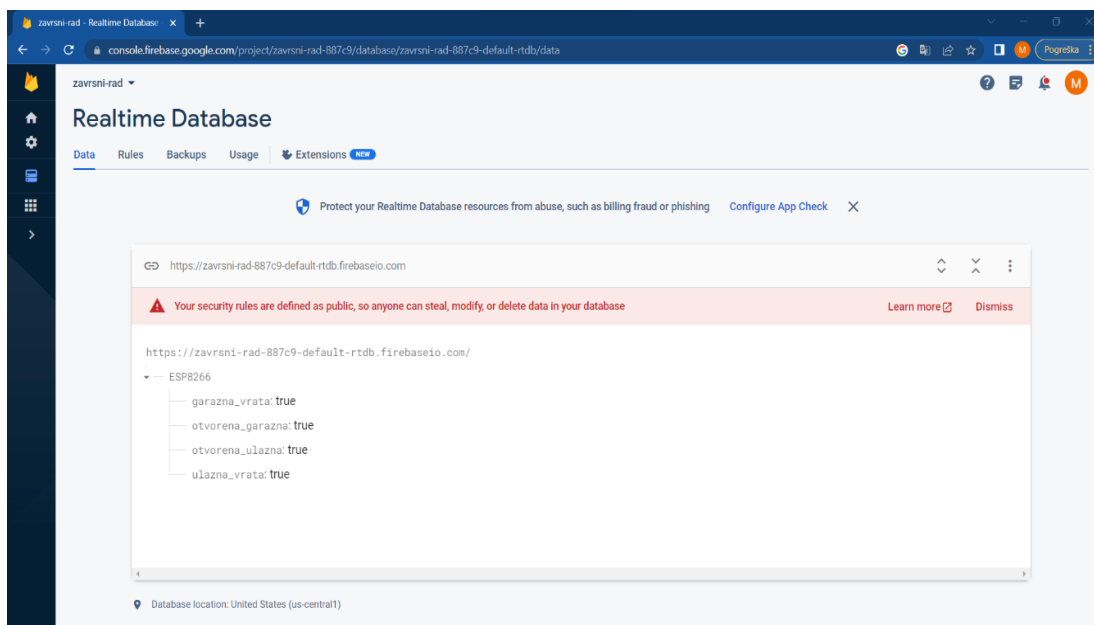
Slika 1. ESP8266 NodeMCU mikrokontroler [1]



Slika 2. ESP 8266 NodeMCU izvodi [2]

2.2. Firebase baza podataka

Firebase je platforma koja omogućuje programerima brzi razvoj mobilnih i web aplikacija. Nastala je 2011. godine kao razvojni projekt dvaju prijatelja, a 2014. godine ima značajan rast kada je kupljen od strane Google-a. *Firebase* nudi veliki broj alata i usluga za izradu, praćenje, izvještavanje, popravljavanje i korištenje aplikacije, a neki od alata su sustav za notifikacije, poruke, autentifikaciju i mnoštvo drugih. Usluga koja je najvažnija za mobilnu aplikaciju i mikrokontroler je *Firebase Realtime Database*, odnosno baza podataka u oblaku (engl. *Cloud-hosted*) koja se mijenja u stvarnom vremenu što je čini idealnom za aplikacije koje zahtijevaju stvarne podatke u vremenu. Na slici 3 se može vidjeti izgled sučelja *Firebase Realtime Database* baze podataka. *Firebase Realtime Database* baza podataka spada u NoSQL (engl. *no SQL, not only SQL*) baze podataka, što znači da kod nje ne postoje relacije između podataka. *Firebase Realtime Database* baza podataka ima JavaScript objektu notaciju (engl. *JavaScript Object Notation*), skraćeno JSON strukturu podataka za organiziranje i pohranu podataka. Struktura podataka sastoji se od parova ključ-vrijednost, gdje svaki ključ ima jedinstveni identifikator i vrijednost za taj ključ. Vrijednosti mogu biti različitog tipa kao što su brojevi, znakovi, objekti i ostalo. *Firebase Realtime Database* baza podataka omogućuje i definiranje sigurnosnih pravila za upravljanje bazom podataka i tako se osigurava da samo željeni korisnici mogu mijenjati sam izgled baze podataka, kao i vrijednosti unutar nje. Sigurnosna pravila su zapisana u sintaksi sličnoj JSON datoteci i lako ih je izmijeniti. Svaka *Firebase Realtime Database* baza podataka ima i svoj token. Token služi korisniku da se može spojiti na bazu podataka. Token i URL (engl. *Uniform Resource Locator*) dati će se ESP8266 NodeMCU mikrokontroleru kako bi se mogao spojiti na bazu podataka. Za jednu bazu podataka moguće je stvoriti više tokena, te tako pratiti koji se korisnik spaja na bazu podataka. Da bi korisnik dobio token i mogao ga koristiti prvo mora stvoriti *Firebase Realtime Database* bazu podataka.

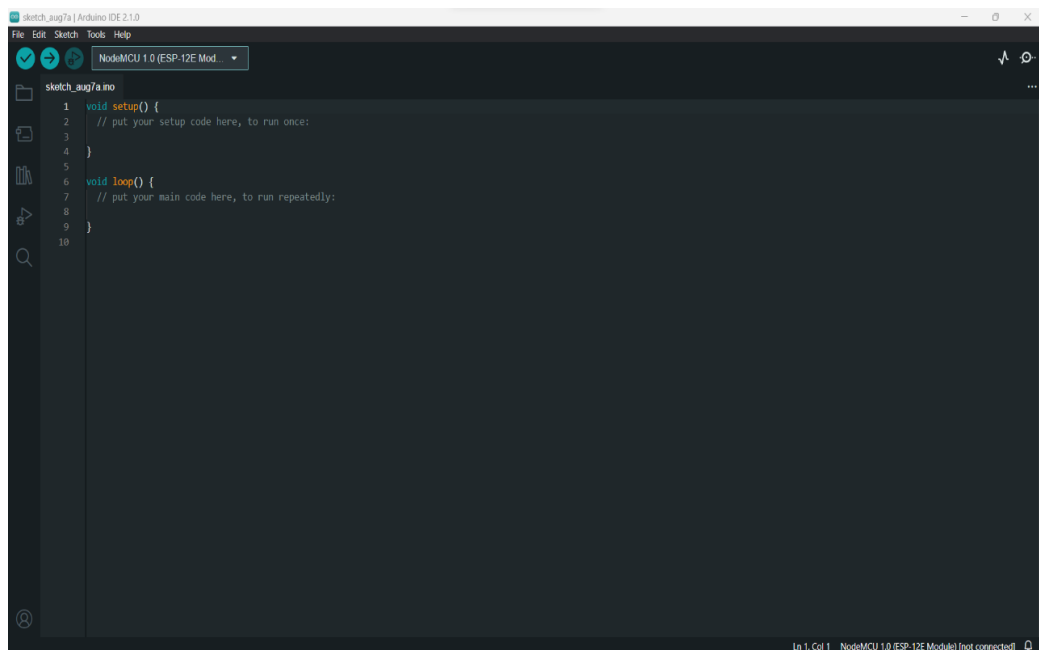


Slika 3. Firebase Realtime Database korisničko sučelje

2.3. Arduino IDE razvojno okruženje

Arduino IDE (engl. *Integrated Development Enviroment*) je *open-source* Arduino softver koji se koristi za programiranje ESP8266 NodeMCU mikrokontrolera i bilo kojih drugih Arduino pločica. Arduino IDE je kompatibilan sa raznim računalnim sustavima kao što su Windows, macOS i Linux, čime je dostupniji većem broju korisnika. Arduino IDE se sastoji od uređivača teksta u koji se piše kod i alatnih traka koje služe za odabir pločice, instaliranje potrebnih programskih knjižnica za pisanje koda, primjera koda za te knjižnice i mnogih drugih mogućnosti. Na slici 4 se može vidjeti kako izgleda Arduino IDE softver kada ga korisnik prvi put pokrene. Za pisanje programskog koda se koristi jezik koji je zasnovan na C i C++ jeziku i ima mnoštvo ugrađenih funkcija i biblioteka koje korisnicima olakšavaju pisanje koda. Arduino IDE u sebi ima upravitelja knjižnica (engl. *Library manager*) koji korisniku omogućuje lagano instaliranje, upravljanje i ažuriranje potrebnih knjižnica za upravljanje senzorima, motorima i drugim komponentama i uslugama. Arduino IDE za svaku svoju knjižnicu ima predložak koda koji korisniku pomaže da shvati kako pojedina knjižnica ili elektronička komponenta radi. Kako bi korisnik mogao pratiti rad koda radi eventualnog otklanjanja pogrešaka ili radi nadzora Arduino IDE unutar sebe ima serijski monitor (engl. *Serial Monitor*) koji korisniku omogućuje primanje i slanje podataka preko

serijskog porta. Prije početka pisanja programskog koda potrebno je izabrati tip pločice koja se koristi. To je moguće na nekoliko načina. Prvi je da u izborniku "Alati" (engl. *Tools*) pod dijelom "Pločice" (engl. *Boards*) se izabere ručno tip pločice koja se koristi. Drugi način je da se pločica pomoću USB kabela spoji na računalo te pomoću padajućeg izbornika koji se nalazi pokraj gumba za prijenos koda na pločicu izabere pločica. Također u izborniku "Alati" korisnik sam može konfigurirati postavke pločice. Nakon što smo napisali kod potrebno je provjeriti postoje li moguće pogreške. To će se provjeriti pritiskom na odgovarajući gumb koje se nalazi iznad uređivača teksta. Ako provjera ne pronađe nikakve pogreške, prelazi se na proces kompiliranja koda i njegov prijenos na mikrokontroler putem USB veze.



Slika 4. Arduino IDE programsko sučelje

2.4. MIT App Inventor

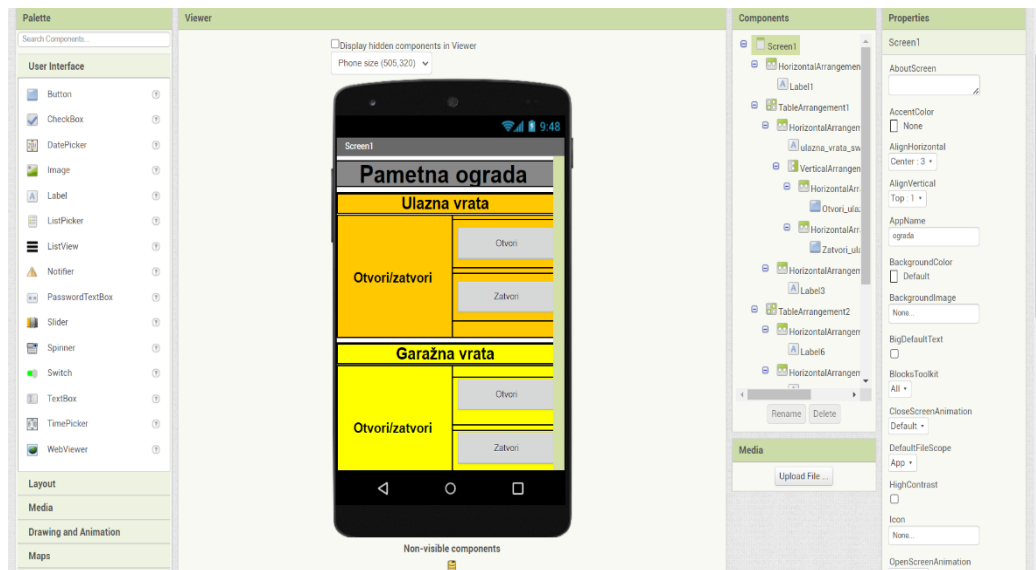
MIT *App Inventor* je online alat za izradu aplikacija za telefone ili tablete unutar web preglednika kojeg je razvio MIT. Alat je jako popularan jer za njegovu upotrebu nije potrebno nikakvo iskustvo ni predznanje u izradi Android mobilnih aplikacija.

MIT *App Inventor* korisničko sučelje se sastoji od dva glavna dijela, a to su uređivač dizajna i uređivač blokova. U uređivaču dizajna osoba koja radi aplikaciju koristi razne

elemente kao što su gumbi, naljepnice, padajući izbornik i mnoge druge, i slaže ih da bi dobio željeno korisničko sučelje aplikacije. Korisnik također može koristiti senzore i hardver svog vlastitog mobitela kao što su akcelerometar, GPS (engl. *Global Positioning System*) za aplikacije koje interagiraju sa stvarnim svijetom. Nakon uspješnog dizajniranja korisničkog sučelja aplikacije, u uređivaču blokova se može svakom elementu korisničkog sučelja dodati posebna funkcija pomoću jednog ili više blokova. Funkcije pojedinog elementa pokrenut će se kada korisnik bude u interakciji s njima, odnosno kada se dogodi određeni događaj. Korisnik koji piše aplikaciju odredit će što se događa klikom na određeni gumb, određenu opciju padajućeg izbornika i ostalo. Nakon što korisnik dizajnira aplikaciju i dodijeli potrebne funkcije pojedinom elementu aplikaciju MIT *App Inventor* nudi mogućnost testiranja aplikacije. Može se koristiti AI (engl. *Artificial Intelligence*) *Companion* za testiranje aplikacije na stvarnom mobilnom uređaju koristeći QR kod i MIT AI2 *Companion* mobilne aplikacije. Ako korisnik nema pristupa mobilnom uređaju MIT *App Inventor* omogućuje simulaciju mobilnog uređaja na računalu pomoću *aiStarter* programa. Ukoliko korisniku nije dopušteni da dva uređaja komuniciraju na bežičnom mreži aplikaciju je moguće testirati na stvarnom uređaju koristeći USB vezu i program *aiStarter*.

2.4.1. Uređivač dizajna

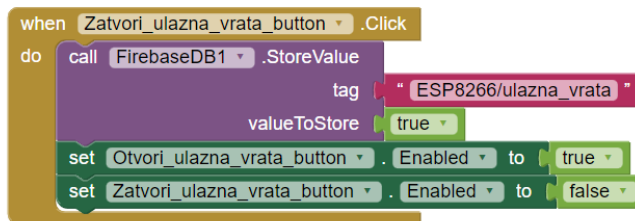
Dio MIT *App Inventor* alata koji služi za izradu vizualnog dizajna korisničkog sučelja, koristeći povlačenje i ispuštanje (engl. *drag and drop*) metodu koja korisnicima omogućuje stvaranje izgleda aplikacije bez da piše ikakav programski kod. Prvo što će korisnik vidjeti kada pokrene uređivač dizajna je prazno platno u obliku mobilnog telefona na koje će korisnik ispuštati elemente kao što su naljepnice, gumbi, padajući izbornici, tekstualni okviri, slike, horizontalno poravnanje, vertikalno poravnanje i mnogi drugi da bi postigao željeni izgled aplikacije kao što se može vidjeti na slici 5. Svaki element ima svoju određenu svrhu i može ga se podesiti prema izgledu aplikacije, to jest može ga se uvećati, smanjiti, povećati ili smanjiti font, promijeniti boju i ostalo. Svakom elementu se može promijeniti ime tako da kad korisnik bude slagao blokove kasnije zna koji je to element. Moguće je dodati više ekrana i načine kretanja između njih, čime se postiže aplikacija sa više zaslona kako bi se osiguralo optimalno korisničko iskustvo. Korisnik može učitati i slike sa svog računala koje mogu poslužiti kao ikona za aplikaciju, pozadina aplikacije ili vizualni izgled pojedinog elementa.



Slika 5. Izgled korisničkog sučelja uređivača dizajna

2.4.2. Uređivač blokova

Dio MIT *App Inventor* alata koji služi za izradu programskog dijela aplikacije koristeći vizualne programske blokove koji definiraju ponašanje i logiku mobilne aplikacije čime se izbjegava pisanje bilo kakvog programskog koda. Svaki blok u MIT *App Inventor*-u predstavlja određenu operaciju, događaj ili funkciju. Svaka vrsta blokova označena je drukčijom bojom i kategorizirana je u odnosu na svoju namjenu kao što su logičke operacije, matematičke operacije, usporedbe, upravljanje tokom koda, kontrole korisničkog sučelja i ostalo. Sama logika aplikacije se postiže tako da se željeni blok povlači i ispušta na platno na kojem se blokovi se mogu lako uređivati, povezivati, slagati ili mogu biti samostalni kako bi se postiglo željeno ponašanje mobilne aplikacije za svaki njen element, odnosno svaki element korisničkog sučelja.



Slika 6. Izgled MIT App Inventor blokova

Blokovi se često koriste u kombinaciji sa upraviteljima događaja. Na slici 6 se može vidjeti jedan od primjera programiranja događaja, koji se pokreće kada korisnik pritisne gumb pod imenom `Zatvori_ulazna_vrata_button`, on će pokrenuti dio koda koji će unutar Firebase Realtime Database spremi u varijablu `ulazna_vrata` vrijednost `true` te će potom još gumb pod imenom `Otvori_ulazna_vrata_button` postaviti u stanje u kojem ga je moguće pritisnuti, a gumb `Zatvori_ulazna_vrata_button` će se postaviti u stanje u kojem ga je nemoguće pritisnuti.

2.5. Infracrveni senzor Iduino ST1081

Infracrveni senzor Iduino ST1081 je senzor prepreka koji se najčešće koristi na robotima koji pužu ili su na kotačima. Na maketi senzor će se koristiti kako bismo otkrili nalazi li se neka prepreka pred ulaznim vratima ili garažnim vratima. Senzor je prilagodljiv i visoke preciznosti. Senzor ima infracrveni odašiljač i infracrveni prijemnik koje se može vidjeti na slici 7. Infracrveni odašiljač emitira infracrveno zračenje određene frekvencije, koje će se kada dođe do prepreke reflektirati natrag te će prijemnik detektirati reflektirano infracrveno zračenje i u tom će se trenutku upaliti i crvena LED dioda koja se nalazi na samom senzoru kao potvrda da senzor radi pravilno. Efektivna udaljenost na kojoj će senzor otkrivati prepreku može se podesiti pomoću potencijometra koji se nalazi na samoj pločici senzora te ta efektivna udaljenost može iznositi od 2 do 40 centimetara. Radni napon senzora je od 3.3 do 5 V, radna struja senzora je manja ili jednaka 20 mA, a efektivni kut iznosi 35 stupnjeva. Ulazno-izlazno sučelje se sastoji od 4 žice: +, GND (engl. *Ground*), *Out* i EN. + sučelje se spaja na pozitivnu stranu izvora napajanja, GND sučelje na GND konektor mikrokontrolera i negativnu stranu napajanja, *out* sučelje je signalni pin i spaja se na GPIO

nožice mikrokontrolera, EN sučelje je sučelje koje aktivira rad niskog statusa, odnosno kad senzor detektira prepreku, uobičajeno je onemogućen.



Slika 7. Infracrveni senzor Iduino ST1081 [3]

2.6. Servo motor TowerPro MG995, 360 stupnjeva

Za razliku od normalnih servo motora kojima se može odrediti točan kut zakretanja do 180 stupnjeva, TowerPro MG995 ima sposobnost kontinuirane rotacije od 360 stupnjeva, bez mogućnosti postavljanja u određeni položaj. Na slici 8 se može vidjeti izgled servo motora, kao i njegove kablove. Servo motor TowerPro MG995 se može okretati u oba smjera. U navedenom slučaju se servo motoru ne zadaje kut za koji će se zakrenuti, nego mu se zadaje smjer i brzina u kojem će se rotor servo motora zakretati te koliko će dugo trajati zakretanje rotora motora. Servo motor ima tri izlazna kabela. Prvi je +, crvene boje, koja se spaja na pozitivnu stranu napajanja od 4.8 do 7.5 V. Drugi je GND, crne boje, koji se spaja na GND nožicu mikrokontrolera i na negativnu stranu napajanja. Treći je PWM (engl. *Pulse Width Modulation*), narančaste boje, koji služi za kontrolu brzine i smjera vrtnje motora, a spaja se na GPIO nožicu mikrokontrolera pomoću koje će mikrokontroler upravljati servo motorom tako što će mu generirati PWM signal. Pomoću tog signala servo motor će znati za koliko se dugo rotor motora zakretati, u kojem smjeru i kojom brzinom.

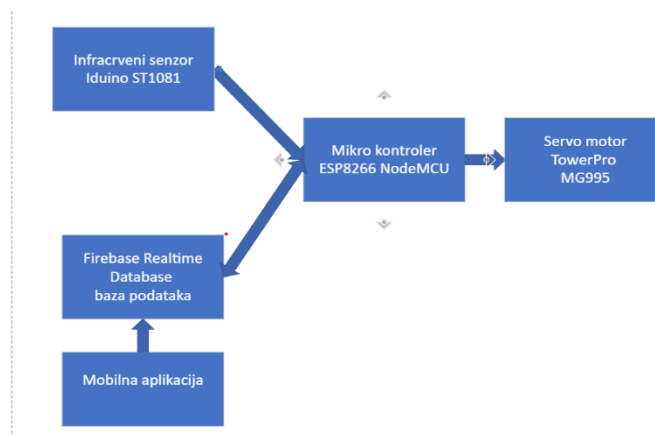


Slika 8. Servo motor [4]

2.7. Funkcijski dijagram sustava

Sustav za upravljanje ogradom i garažnim vratima zasniva se na jednostavnom principu. U centru cijelog sustava je mikrokontroler ESP8266 NodeMCU koji očitava vrijednosti infracrvenih senzora i vrijednosti pojedinih varijabli u *Firestore Realtime Database* bazi podataka. Na temelju tih vrijednosti mikrokontroler izvršava određene funkcije i mijenja vrijednosti određenih kontrolnih varijabli koje se nalaze unutar *Firestore Realtime Database* baze podataka. Mobilna aplikacija također komunicira sa *Firestore Realtime Database* bazom podataka te pomoću nje može zadati naredbe za otvaranje i zatvaranje ograde i garažnih vrata. To je predstavljeno pomoću funkcijskog dijagrama, na slici 9., nacrtanog u programu Microsoft Visio kojeg je moguće koristiti kao dio paketa Microsoft Office 365. Aplikacija se u radu koristi za crtanje dijagrama, a može se koristiti i za razne druge primjene kao što su vektorska grafika, stvaranje vizualnih prikaza, tlocrte i još mnoštvo toga. Visio na raspolaganju ima mnoštvo predložaka i oblika koji korisniku služe za brzo i efikasno razvijanje dijagrama. Tim oblicima može se manipulirati i spajati sa ostalim elementima dijagrama. U našem dijagramu postoje tri vrste oblika: pravokutni, jednosmjerne strelce i dvosmjerne strelce. Pravokutni oblici nam prikazuju elemente sustava i elektroničke komponente, kao što su ESP8266 NodeMCU mikrokontroler, *Firestore Realtime Database* baza podataka i ostalo. Strelce nam prikazuju mogućnost komunikacije između određenih elemenata sustava. Jednosmjerne strelce pokazuju da je moguća komunikacija samo u jednom smjeru, odnosno to znači da će sustav imati jednosmjernu komunikaciju između svojih komponenti. Dvosmjerne strelce pokazuju da je moguća

komunikacija u oba dva smjera, odnosno to znači da će sustav imati dvosmjernu komunikaciju između svojih komponenti.

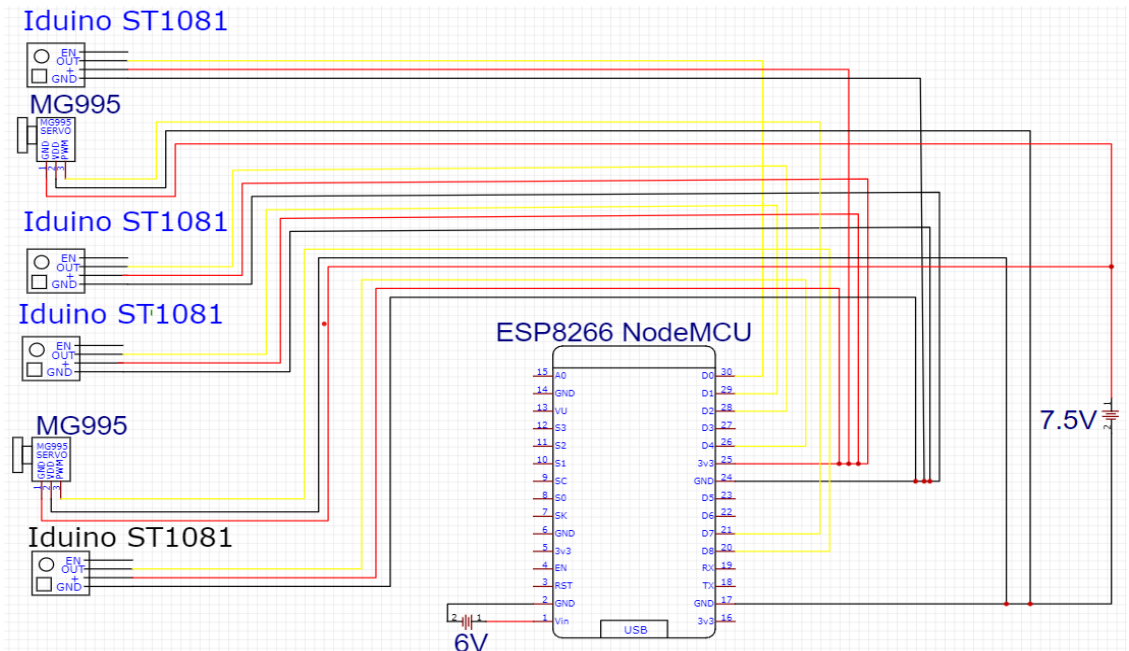


Slika 9. Funkcijski dijagram sustava.

2.8. Električna shema sustava

Električna shema sustava nam služi za prikaz električnih strujnih krugova te način povezivanja i međusobnih odnosa pojedinih elemenata strujnog kruga, kao što je prikazano na slici 10. Električna shema je izrađena pomoću web aplikacije EasyEda, za čije korištenje nije potrebno ni napraviti korisnički račun, već se korisnik može prijaviti pomoću svog vlastitog Google računa. Web aplikacija ima veliku bazu raspoloživih komponenti, a ako ne postoji pojedini element korisnik ga može sam nacrtati i kasnije, ako želi, može ga objaviti na korištenje drugim korisnicima te tako i sam povećati bazu dostupnih komponenti unutar web aplikacije. Alat ima dosta funkcija za manipulaciju kako samim elektroničkim komponentama, tako i ostalim komponentama elektroničke sheme kao što su kablovi. Za elektroničke komponente moguće je urediti veličini fonta, vrstu fonta, boju fonta, same izvode elektroničke komponente, natpise i mnoštvo drugih. U električnoj shemi će biti prisutne tri boje kablova: žutu, crvenu i crnu. Žuti kablovi nam predstavljaju kontrolne kablove, kod servo motora služe za upravljanje smjerom vrtnje rotora, brzinom vrtnje rotora i vremenom koliko će se dugo rotor vrtjeti, a kod senzora pokreta kabel nam služi da prenese informaciju mikrokontroleru da li je otkrivena prepreka. Crveni kablovi nam predstavljaju pozitivnu stranu napajanja. Crni kablovi nam predstavljaju negativnu stranu napajanja, i one

se još moraju spojiti na jedan od GND izvoda mikrokontrolera, koji će biti zajednički za tu vrstu elektroničke komponente.



Slika 10. Električna shema sustava

2.9. Realizacija funkcionalne makete

Maketa sustava se izrađuje od pleksiglasa i cijeli je sustav podijeljen na manje podsustave koji rade zajedno, kao što su upravljački podsustav kojeg čini mikrokontroler ESP8266 NodeMCU, podsustav za ogradu i podsustav za garažna vrata. Podsustav za ogradu i podsustav za garažna vrata komuniciraju sa upravljačkim sustavom kojeg čini mikrokontroler ESP8266 NodeMCU i on izravno izdaje naredbe servo motorima podsustava za ogradu i podsustava za garažnu ogradu i time se postiže funkcionalnost cijelog sustava.

2.9.1. Provjera rada elektroničkih komponenti

Prije same izrade makete, potrebno je testirati električne komponente i kod. Ovo se može postići uz pomoć električne sheme, kojom će se izraditi prototip autonomnog sustava

na eksperimentalnoj pločici (engl. *breadboard*). Eksperimentalna pločica je pravokutni komad plastike na kojem se nalazi mnoštvo malih rupa. Te rupe nam dopuštaju da u njih spojimo kablove i ostale elektroničke komponente kao mikrokontroler ESP8266 NodeMCU kako bi provjerili elektronički strujni krug i rad cijelog sustava.

2.9.2. Izrada programskog koda

Nakon uspješnog spajanja svih elektroničkih komponenti prateći električnu shemu i prilagodbe osjetljivosti pojedinog senzora se može krenuti u izradu programskog koda koji će biti implementiran na ESP8266 NodeMCU mikrokontroler. Prije pisanja programskog koda potrebno je osigurati da sam mikrokontroler bude prepoznat od strane računala. Nakon toga odabire se odgovarajući tip ESP8266 pločice i instalira ju se potrebne knjižnice koje omogućuju mikrokontroleru spajanje na Internet i *Firestore Realtime Database* bazu podataka te knjižnicu za upravljanje servo motorima.

Nakon odrađenih koraka pripreme slijedi pisanje programskog koda. Programski kod se piše unutar dvije glavne petlje: "*void setup*" i "*void loop*". Iznad "*void setup*" petlje se uključuju potrebne knjižnice i inicijaliziraju varijable kojima se može pristupiti u bilo kojem dijelu programskog koda.

Unutar "*void setup*" petlje se piše dio koda koji se izvršava jedanput prilikom pokretanja programskog koda sustava ili prilikom ponovnog pokretanja sustava, na primjer postupak spajanja mikrokontrolera na Internet i *Firestore Realtime Database* bazu podataka.

Unutar "*void loop*" petlje se piše programski kod koji se neprestano ponavlja dok mikrokontroler radi. Ovdje se implementiraju funkcionalnosti kao što su čitanje podataka i ažuriranje kontrolnih varijabli u *Firestore Realtime Database* bazi podataka, ažuriranje vrijednosti varijabli senzora pokreta i slično.

U primjerima ispod nalaze se najvažniji dijelovi programskog koda koji su kratko objašnjeni. Tu su dijelovi programskog koda za spajanje na WiFi mrežu i *Firestore Realtime Database* bazu podataka, ažuriranje podataka senzora pokreta i dohvaćanje ažuriranih podataka sa *Firestore Realtime Database* baze podataka u koju aplikacija sprema svoje podatke.

Isječak programskog koda 1: Programski kod za spajanje na WiFi mrežu i bazu podataka

```
1. void setup() {
2.   // put your setup code here, to run once:
3.   Serial.begin(9600);
4.   //pokušaj spajanja na Wi-Fi mrežu
5.   //-----
6.   WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
7.   Serial.println("Connecting to Wi-Fi");
8.   while (WiFi.status() != WL_CONNECTED)
9.   {
10.    Serial.print(".");
11.    delay(300);
12.  }
13.  Serial.println();
14.  Serial.println("Connected with IP: ");
15.  Serial.println(WiFi.localIP());
16.  //-----
17.  Serial.println();
18.  //spajanje na Firebase
19.  //-----
20.  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
21.  Firebase.reconnectWiFi(true);
22.  //-----
23.  //postavljanje pinova ESP8266 NodeMCU mikro kontrolera na koji
24.  //su spojeni senzori u INPUT mod
25.  //-----
26.  pinMode(D0, INPUT);
27.  pinMode(D2, INPUT);
28.  pinMode(D1, INPUT);
29.  pinMode(D3, INPUT);
30.  //-----
31. }
32.
```

Isječak programskog koda 2: Programski kod za ažuriranje podatka senzora pokreta

```
1. //funkcija za osvježavanje vrijednosti senzora
2. void IRUpdate(){
3.   delay(1000);
4.   //očitanje podataka pojedinog senzora i spremanje u pojedinu varijablu
5.   senzor1=digitalRead(D0);
6.   senzor2=digitalRead(D2);
7.   senzor3=digitalRead(D1);
8.   senzor4=digitalRead(D3);
9.   delay(1000);
10. }
11.
```

Isječak programskog koda 3: Programski kod za ažuriranje podatka iz baze podataka

```
1. //funkcija za isčitavanje vrijednosti spremljenih u Firebase Realtime Database
2. //bazu podatak i spremanje u pojedinu pripadajuću varijabli
3. void Update_firebase_podataka(){
4.   otvorena_dvorisna=Firebase.getBool(otvorena_dvorisna_fd, "/ESP8266/otvorena_ulazna
");
5.   otvorena_garazna=Firebase.getBool(otvorena_garazna_fd, "/ESP8266/otvorena_garazna"
);
);
```

```
6.   ulazna=Firebase.getBool(aplikacija_ulazna,"/ESP8266/ulazna_vrata");
7.   garazna=Firebase.getBool(aplikacija_garazna,"/ESP8266/garazna_vrata");
8. }
9.
```

2.9.3. Testiranje rada programskog koda i elektroničkih komponenti

Nakon što je napisan programski kod i spojene elektroničke komponente na eksperimentalnu pločicu (engl. *breadboard*) potrebno je testirati rad cijelog sustava da se korisnik može uvjeriti radi li programski kod kako je željeno i u slučaju da postoje neke pogreške programskom kodu ili elektroničkoj shemi ta da ih odmah korisnik može ispraviti. Za testiranje sustava potrebno je dovesti napajanje od 6 i 7.5 V. Izvor napajanja od 6 V će se koristiti za napajanje mikrokontrolera. Obično se za napajanje mikrokontrolera ESP8266 NodeMCU koristi izvor napajanja od 3-3.3 V, ali zato što će se koristiti jedan od 3.3 V regulatora za napajanje senzora prepreka, potrebno je dovesti veći izvor napajanja za mikrokontroler. U slučaju da se koristi izvor napajanja od 3-3.3 V mikrokontroler bi radio, kao i senzori prepreka, ali senzori prepreka ne bi imali željeni domet te bi to odmah primijetili pomoću kontrolne LED lampice na senzoru koja ne bi svijetlila punim intenzitetom. Izvor napajanja od 7.5 V će se koristiti za napajanje servo motora jer je potrebna veća količina napona i električne struje za pokretanje samog motora.

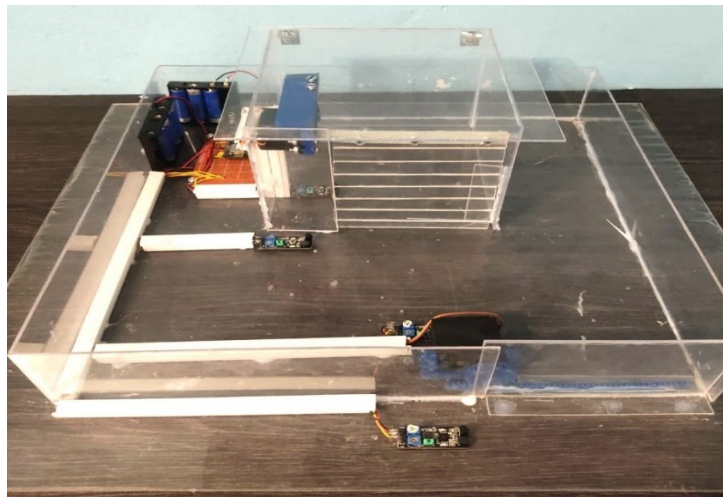
2.9.4. Tiskana pločica

Nakon uspješno provedenog testiranja programskog koda može se prijeći na izradu tiskane pločice. Tiskana pločica je sredstvo za povezivanje elektroničkih komponenti koja se sastoji od izolatorskog materijala na kojem se različitim postupcima oblikuje vodič. Za potrebe ovog rada će se koristiti univerzalna tiskana pločica s bakrenim kružićima koji su obloženi lemnom legurom. Kod takve pločice se postavljaju elementi unutar kružića i zatim se leme uz pomoć lemne legure na pločicu.

2.9.5. Maketa

Maketa dvorišnih i garažnih vrata, koja se može vidjeti na slici 11., je izrađena od pleksiglasa koji je spajan vrućim ljepilom. Stijenka je debljine 2.5 mm što osigurava dovoljnu stabilnost za navedenu maketu. Na maketi je napravljena ograda na kojoj je dio stacionaran, a dio pokretan, i garaža na kojoj se nalaze garažna vrata. Pokretni dio ograde je napravljen pomoću linearnog pokretača koji je zalijepljen na pokretni dio ograde te zupčanik

koji se nalazi na servo motoru. Zupčanik se kreće po linearnom pokretaču i tako se omogućuje otvaranje i zatvaranje ograde. Sam servo motor je pričvršćen na držač motora pomoću šarafa, a držač motora je pričvršćen na podlogu makete. Na krovu garaže se također nalazi držač motora koji je pričvršćen šarafima za krov garaže. Na rotoru servo motora garaže nalazi se crijevo te se pomoću njega omogućuje otvaranje i zatvaranje garažnih vrata. Garažna vrata se sastoje od više dijelova pleksiglasa koji su isječeni i s jedne strane zalijepljeni ljepljivom trakom. To služi kako bi se postiglo da, kada se garažna vrata otvaraju, ne udare u krov, nego da se smotaju oko crijeva.



Slika 11. Izgled završene makete sustava

2.9.6. Mobilna aplikacija za upravljanje

Kako bi sustav radio bez potrebe senzora razvijena je aplikacija za mobilni telefon s Android operativnim sustavom. Pomoću aplikacije je moguće otvarati i zatvarati dvorišna i garažna vrata bilo gdje na svijetu gdje korisnik ima pristup internetu. Za izradu mobilne aplikacije je korišten spomenuti besplatni alat MIT *App Inventor* koji ne zahtjeva predznanje programiranja jer se cijeli proces programiranja mobilne aplikacije svodi na povuci i ispusti metodi kojom se izbjegava bilo kakvo pisanje programskog koda od strane korisnika. Korisnik sam slaže vizualni dio aplikacije unutar vizualnog prozora koji predstavlja korisničko sučelje. Prvo što korisnik vidi kad pokrene MIT *App Inventor* je prazno platno u obliku mobilnog telefona na koje će korisnik ispuštati razne elemente kao što su gumbi,

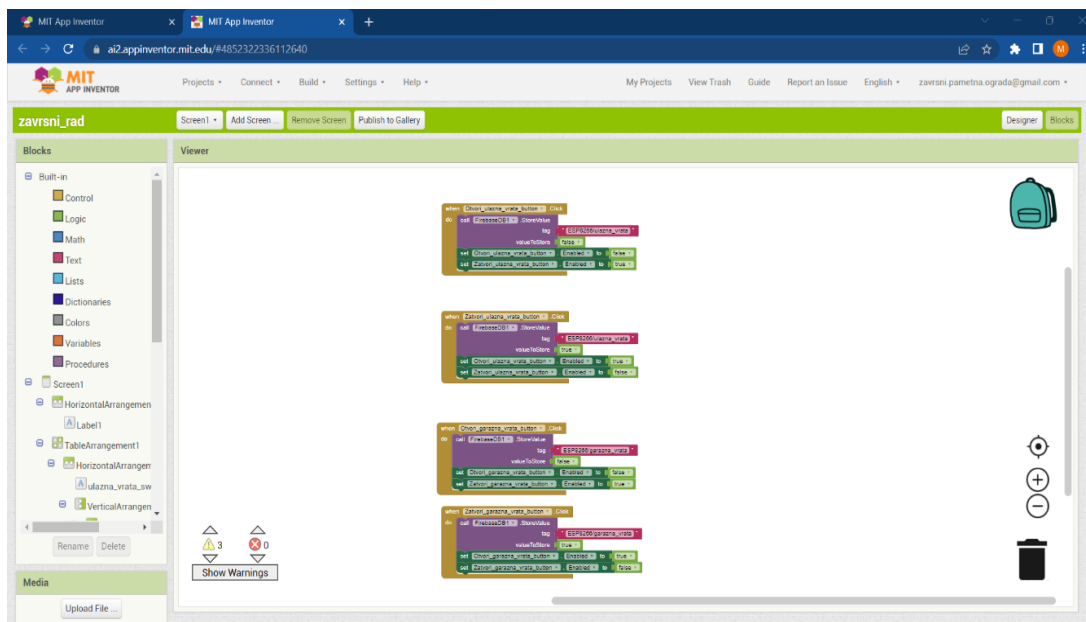
tekstualni okviri, padajući izbornici, slike, horizontalno poravnanje, vertikalno poravnanje i mnoge druge, kao što se može vidjeti na slici 12., kako bi se postigao željeni izgled mobilne aplikacije, odnosno korisničkog sučelja. Svaki element korisničkog sučelja ima određenu ulogu i moguće ga je podesiti da ne narušava izgled aplikacije tako šta ga se može smanjiti, uvećati, promijeniti veličina, boja i stil fonta i tako dalje. Nakon što se uredi korisničko sučelje kako je zamišljeno, moguće je svakom elementu promijeniti ime kako bi se korisnik kasnije lakše snašao kada bude spajao blokove. Također je moguće dodati još zaslona kako bi se poboljšalo korisničko iskustvo ako to korisnik želi.



Slika 12. Izgled korisničkog sučelja za uređivanje dizajna aplikacije

Nakon što je korisnik dizajnirao korisničko sučelje prelazi se na samo programiranje programskog dijela aplikacije koristeći vizualne programske blokove pomoću kojih se svakom elementu korisničkog sučelja dodaje određena funkcionalnost. Svakom elementu korisničkog sučelja može se dodati događaj koji će pokrenuti kod za taj element, na primjer za gumb može se reći da je klik na njega događaj koji pokreće njegovu funkciju. Svaki blok unutar MIT APP Inventor-a predstavlja određenu funkciju, operaciju ili događaj te je svaki blok označen drukčijom bojom i karakteriziran u odnosu na njegovu namjenu. Neke od kategorija su matematičke operacije, upravljanje tokom koda, logičke operacije, usporedbe i tako dalje. Programiranje pojedinih elemenata aplikacije, odnosno logika iza elemenata korisničkog sučelja, postiže se povlačenjem željenog bloka na platno gdje se blokovi mogu

uređivati, povezivati, slagati ili biti samostalni kako bi korisnik postigao željeno ponašanje mobilne aplikacije. Primjer takvog sastavljanja koda pomoću blokova može se vidjeti na slici 13.



Slika 13. Izgled blokovskog sučelja za dodjeljivanje funkcionalnosti aplikacije

Nakon završetka izrade aplikacije moguće je testirati radi li kako je zamišljeno. Ako aplikacija zadovolji sve potrebe, moguće ju je instalirati na mobilni telefon.

2.9.7. Verificiranje makete

Nakon uspješne izrade mobilne aplikacije, makete i programskog koda potrebno je testirati maketu prije trajnog puštanja u rad. Potrebno je osigurati da senzori ne detektiraju stijenke ograde, ali i da domet senzora nije premalen jer tako neće detektirati nikakve prepreke. Ako bi se dogodilo da senzor detektira stijenke ograde mikrokontroler bi stalno pokretao kod za otvaranje i zatvaranje ograde ili garažnih vrata. Nakon toga potrebno je osigurati da mikrokontroler pokreće odgovarajući dio koda ovisno o tome koji je senzor detektirao prepreku ili nije. Na kraju je potrebno i uvjeriti se da mobilna aplikacija još radi kako je zamišljeno i ukoliko je potrebno izvršiti promjene.

3. Zaključak

Živimo u vremenu kada se brzim napretkom tehnologije sve više svakodnevnih procesa ljudskog života pokušava automatizirati i svesti potrebu ljudske interakcije na minimum. Stoga nije ni čudno da se povećava i broj firmi koje nude proces automatizacije otvaranja i zatvaranja dvorišnih, odnosno garažnih vrata. Zadatak ovog rada bio je prikazati upravo jedan takav proces automatizacije sve od početne ideje do izrade makete koja prikazuje taj proces. Na samom početku bilo je potrebno odrediti koje će se elektroničke komponente koristiti. Kod elektroničkih komponenti prvo je trebalo donijeti odluku hoće li se koristiti senzor pokreta ili senzor prepreka. Nakon testiranja donesena je odluka da je bolje koristiti senzor prepreka. Da bi se proces otvaranja i zatvaranja ograde, odnosno garažnih vrata ostvario, bilo je potrebno naći odgovarajući tip motora. Nakon istraživanja donesena je odluka da se koristi servo motor s mogućnošću kontinuirane rotacije. Za potrebe otvaranja i zatvaranja ograde, odnosno garažnih vrata izrađena je i mobilna aplikacija, kojom korisnik može sam pokrenuti određeni proces pritiskom na gumb. Također je postignuto da, ukoliko korisnik ne želi koristiti aplikaciju, cijeli proces može pokrenuti tako što će stati ispred senzora. Cijeli sustav bi se mogao podići na još jednu razinu više tako što bi se izgradio vlastiti izvor napajanja sustava, odnosno električne energije, a to se može postići pomoću izgradnje fotonaponskog sustava s baterijama u koje bi se spremala električna energija koja bi napajala cijeli sustav u vremenu kad nema Sunca. Tijekom cijelog rada došlo se do spoznaje koliko je vremena moguće uštediti i poboljšati kvaliteta ljudskog života automatizacijom svakodnevnih procesa.

Popis literature

- [1] ESP8266 NodeMCU mikrokontroler. Pristupljeno : 2. kolovoza 2023. [Online]. Dostupno na : <https://components101.com/sites/default/files/components/ESP8266-NodeMCU.jpg>
- [2] ESP 8266 NodeMCU izvodi . Pristupljeno : 2. kolovoza 2023. [Online]. Dostupno na: https://components101.com/sites/default/files/component_pin/NodeMCU-ESP8266-Pinout.jpg
- [3] Infracrveni senzor Iduino ST1081 . Pristupljeno : 2. kolovoza 2023. [Online]. Dostupno na: https://asset.conrad.com/media10/isa/160267/c1/hr/1485307_LB_00_FB/image.jpg?x=400&y=400
- [4] Servo motor . Pristupljeno : 2. kolovoza 2023. [Online]. Dostupno na: https://soldered.com/productdata/2017/01/DSC_3019.jpg
- [5] Schwartz,M.(2016) : Internet of Things with ESP8266
- [6] Kong,S., Abelson,H. (2019) : Computational Thinking Education
- [7] ESP8266 NodeMCU: Getting Started with Firebase (Realtime Database). Pristupljeno : 5. kolovoza 2023. [Online]. Dostupno na : <https://randomnerdtutorials.com/esp8266-nodemcu-firebase-realtime-database/>
- [8] MG995 High Speed Metal Gear Dual Ball Bearing Servo. Pristupljeno : 6. kolovoza 2023.[Online]. Dostupno na : https://soldered.com/productdata/2017/01/Soldered_MG995_datasheet.pdf
- [9] Obstacle Avoidance Sensor (ST1081). Pristupljeno : 6. kolovoza 2023. [Online]. Dostupno na : <https://asset.conrad.com/media10/add/160267/c1/-/en/001485307DS01/datasheet-1485307-iduino-1485307-infrared-sensor-suitable-for-single-board-pcs-arduino-1-pcs.pdf>
- [10] Firebase. Pristupljeno: 12. kolovoza 2023. [Online]. Dostupno na : <https://en.wikipedia.org/wiki/Firebase>
- [11] Banzi,M.,Shiloh,M (2014) Getting Started with Arduino, 3rd Edition

[12] How to Use a Breadboard for Electronics and Circuits. Pristupljeno : 25. kolovoza 2023. [Online] Dostupno na : <https://www.sciencebuddies.org/science-fair-projects/references/how-to-use-a-breadboard#what-breadboard>

[13] MIT APP Inventor web-stranica. Pristupljeno : 26. kolovoza 2023. [Online] Dostupno na : <https://appinventor.mit.edu/>

[14] 8 Reasons Why ESP8266 has Been So Popular Pristupljeno : 26. kolovoza 2023. [Online] Dostupno na : <https://blog.techdesign.com/8-reasons-esp8266-popular/>

Popis slika

Slika 1. ESP8266 NodeMCU mikrokontroler.....	3
Slika 2. ESP 8266 NodeMCU izvodi.....	3
Slika 3. Firebase Realtime Database korisničko sučelje.....	5
Slika 4. Arduino IDE programsko sučelje.....	6
Slika 5. Izgled korisničkog sučelja uređivača dizajna.....	8
Slika 6. Izgled MIT App Inventor blokova.....	9
Slika 7. Infracrveni senzor Iduino ST1081.....	10
Slika 8. Servo motor.....	11
Slika 9. Funkcijski dijagram sustava.....	12
Slika 10. Električna shema sustava.....	13
Slika 11. Izgled završene makete sustava.....	17
Slika 12. Izgled korisničkog sučelja za uređivanje dizajna aplikacije.....	18
Slika 13. Izgled blokovskog sučelja za dodjeljivanje funkcionalnosti aplikacije.....	19

OBRAZAC 5

IZJAVA O AUTORSTVU

Ja, Marko Puškarić

izjavljujem da sam autor/ica završnog/diplomskog rada pod nazivom

Upravljanje mikrokontrolerom pomoću aplikacije
izrađene u MIT App Inventor razvojnom
okruženju

Svojim vlastoručnim potpisom jamčim sljedeće:

- da je predani završni/diplomski rad isključivo rezultat mog vlastitog rada koji se temelji na mojim istraživanjima i oslanja se na objavljenu literaturu, a što pokazuju korištene bilješke i bibliografija,
- da su radovi i mišljenja drugih autora/ica, koje sam u svom radu koristio/la, jasno navedeni i označeni u tekstu te u popisu literature,
- da sam u radu poštivao/la pravila znanstvenog i akademskog rada.

Potpis studenta/ice

Marko Puškarić



OBRAZAC 6

**ODOBRENJE ZA OBJAVLJIVANJE ZAVRŠNOG/DIPLOMSKOG RADA U
DIGITALNOM REPOZITORIJU**

Ja Marko Puškarić

dajem odobrenje za objavljivanje mog autorskog završnog/diplomskog rada u javno dostupnom digitalnom repozitoriju Veleučilišta u Virovitici sadržanom u Dabar (Digitalni akademski arhivi i repozitoriji) te u javnoj internetskoj bazi završnih radova Nacionalne i sveučilišne knjižnice bez vremenskog ograničenja i novčane nadoknade, a u skladu s odredbama članka 58. stavka 5., odnosno članka 59. stavka 4. Zakona o visokom obrazovanju i znanstvenoj djelatnosti (NN 119/22).

Potvrđujem da je za pohranu dostavljena završna verzija obranjenog i dovršenog završnog/diplomskog rada. Ovom izjavom, kao autor navedenog rada dajem odobrenje i da se moj rad, bez naknade, trajno javno objavi i besplatno učini dostupnim na sljedeći način:

- a) Rad u otvorenom pristupu
- b) Rad dostupan nakon: _____ (upisati datum nakon kojeg želite da rad bude dostupan)
- c) Pristup svim korisnicima iz sustava znanosti i visokog obrazovanja RH
- d) Pristup korisnicima matične ustanove
- e) Rad nije dostupan (u slučaju potrebe dodatnog ograničavanja pristupa Vašem završnom/diplomskom radu, podnosi se pisani obrazloženi zahtjev).

Potpis studenta/ice

Marko Puškarić

U Virovitici, 07.09.2023.