Edukativna Arduino kutija

Zrnić, Mario

Undergraduate thesis / Završni rad

2022

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

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:016345

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-04-19**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

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

Sveučilišni studij

Edukativna Arduino kutija

Završni rad

Mario Zrnić

Osijek, 2021.

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Obrazac Z1P - Obrazac za ocjenu završnog rada na preddiplomskom sveučilišnom studiju

Osijek, 21.09.2021.

Odboru za završne i diplomske ispite

Prijedlog ocjene završnog rada na preddiplomskom sveučilišnom studiju

Ime i prezime Pristupnika:	Mario Zrnić
Studij, smjer:	Preddiplomski sveučilišni studij Elektrotehnika i informacijska
Mat. br. Pristupnika, godina upisa:	4601, 24.07.2018.
OIB Pristupnika:	54927199846
Mentor:	Izv. prof. dr. sc. Ivan Aleksi
Sumentor:	,
Sumentor iz tvrtke:	
Naslov završnog rada:	Edukativna Arduino kutija
Znanstvena grana rada:	**Arhitektura računalnih sustava (zn. polje računarstvo)**
Zadatak završnog rad:	U ovom završnom radu je potrebno napraviti kutiju s puno pokaznik aktuatora i senzora. Neki od dijelova edukativne kutije su tipke, sklopke, pokaznik, LE diode, mikroupravljač, servo motor, ventilator, senzor temperature, IR daljinski...
Prijedlog ocjene završnog rada:	Vrlo dobar (4)
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: 3 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 1 bod/boda Jasnoća pismenog izražavanja: 3 bod/boda Razina samostalnosti: 2 razina
Datum prijedloga ocjene od strane mentora:	21.09.2021.
Datum potvrde ocjene od strane Odbora:	25.09.2021.

Potvrda mentora o predaji konačne verzije rada:	Mentor elektronički potpisao predaju konačne verzije.
	Datum:

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

IZJAVA O ORIGINALNOSTI RADA

Osijek, 03.10.2022.

Ime i prezime studenta:	Mario Zrnić
Studij:	Preddiplomski sveučilišni studij Elektrotehnika i informacijska tehnologija
Mat. br. studenta, godina upisa:	4601, 24.07.2018.
Turnitin podudaranje [%]:	6

Ovom izjavom izjavljujem da je rad pod nazivom**: Edukativna Arduino kutija**

izrađen pod vodstvom mentora Izv. prof. dr. sc. Ivan Aleksi

i sumentora ,

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija.
Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:

SADRŽAJ

1. UVOD 5

1.1. Zadatak završnog rada 5

2. PREGLED POSTOJEĆIH RJEŠENJA 6

2.1. Arduino komplet za početnike – IN2 Grupa 6

2.2. Smart Home Kit for Arduino – Keyestudio 6

3. DIJELOVI EDUKATIVNE ARUDINO KUTIJE 7

3.1. Treptajuće svijetlo 7

3.2. Tipkalom upravljana LE dioda 8

3.3. Servo motor 9

3.4. Ventilator 10

3.5. Pasivni buzzer 12

3.6. Photocell sensor 13

3.7. Senzor vodene pare 15

3.8. Senzor pokreta 16

3.9. Senzor detekcije plina 17

3.10. Senzor vlažnosti tla 19

3.11. Senzor temperature 20

3.12. LCD zaslon 22

3.13. OLED zaslon 24

4. ZAKLJUČAK 27

LITERATURA 28

SAŽETAK 29

ABSTRACT 30

ŽIVOTOPIS 31

PRILOZI 1

1. UVOD

U ovom radu su opisane neke od osnovnih komponenata kompatibilnih sa Arduino razvojnom pločicom. Arduino je platforma otvorenog koda koja na jednostavan način spaja softver sa hardverom. Nudi svoj Arduino programski jezik baziran na C i C++ programskim jezicima sa dodatkom posebnih metoda i funkcija. Razvojna pločica preko pinova može čitati ulazne signale te ih pretvarati u izlazne signale. Ulazni signali su uglavnom očitanja raznih senzora, a izlazni signali su naredbe koje određuju ponašanje komponenti spojenih na razvojnu pločicu. Sve izvedbe Arduino pločica imaju analogne i digitalne pinove te je na određenim digitalnim pinovima moguće imitirati izlazni analogni signal pomoću modulacije širine impulsa (PWM). Za potrebe ovog rada je korištena Arduino Uno R3 razvojna pločica sa ATmega328 mikrokontrolerom. Komponente koje se spajaju na Arduino pločicu su LE diode, tipkala, motori, zasloni te senzori temperature, vlažnosti i dima. Za neke komponente nužno je spajanje dodatnih otpornika i/ili tranzistora za regulaciju struje kako bi ispravno radile što je prikazano na dijagramima spajanja komponenti sa Arduino pločicom.

1.1. Zadatak završnog rada

Edukativna Arduino kutija je zamišljena kao priručnik za sve osobe koje žele savladati Arduino osnove kako bi stečeno znanje kasnije primijenili na vlastitim projektima. Fokus rada je jednostavnost spajanja komponenti i njihov programski kod, odnosno u radu je je prikazano spajanje komponenti na najjednostavniji mogući način te što jednostavniji i razumljiviji programski kod za prikazane komponente. Dodatno u prilogu rada se nalaze kompleksniji dijagrami spajanja više različitih komponenti sa jednom Arduino razvojnom pločicom.

2. PREGLED POSTOJEĆIH RJEŠENJA

Budući da je Arduino platforma najpopularnija platforma upravljanja mikrokontrolerima na tržištu je dostupno mnoštvo Arduino kompleta za početnike. Nedostatak tih kompleta je u tome što je čest slučaj da proizvođač stavlja svoje gotove module za pojedine komponente, umjesto samih komponenti. Stoga se i dijagrami spajanja mogu razlikovati u odnosu na uobičajene komponente kada se kupuju pojedinačno. Također, moguća je i razlika u programskom kodu upravo zbog specifičnosti modula koji se koristi u odnosnu na samostalnu komponentu. U ovom poglavlju su opisana dva kompleta od kojih je jedan ne koristi vlastite gotove module, dok u drugom su sve komponente specifični moduli proizvođača.

2.1. Arduino komplet za početnike – IN2 Grupa

Ovaj komplet za početnike koji je sastavila IN2 Grupa se sastoji od 30 različitih komponenti, 30 različitih senzora te priručnik za spajanje komponenti.. Cilj priručnika je prikazati najčešće metode pri izradi vlastitih projekata. Sastoji se od 16 lekcija koje objašnjavaju spajanje komponenti i programski kod potreban za izvođenje projekata. Osim 16 lekcija priručnik sadrži dodatnih 8 zadataka u kojima je potrebno proširiti objašnjene lekcije. Bitno je spomenuti kako ovaj komplet ne zadrži module posebne izrade od strane tvrtke, nego su sve komponente standardne koje se mogu kupiti pojedinačno na tržištu.

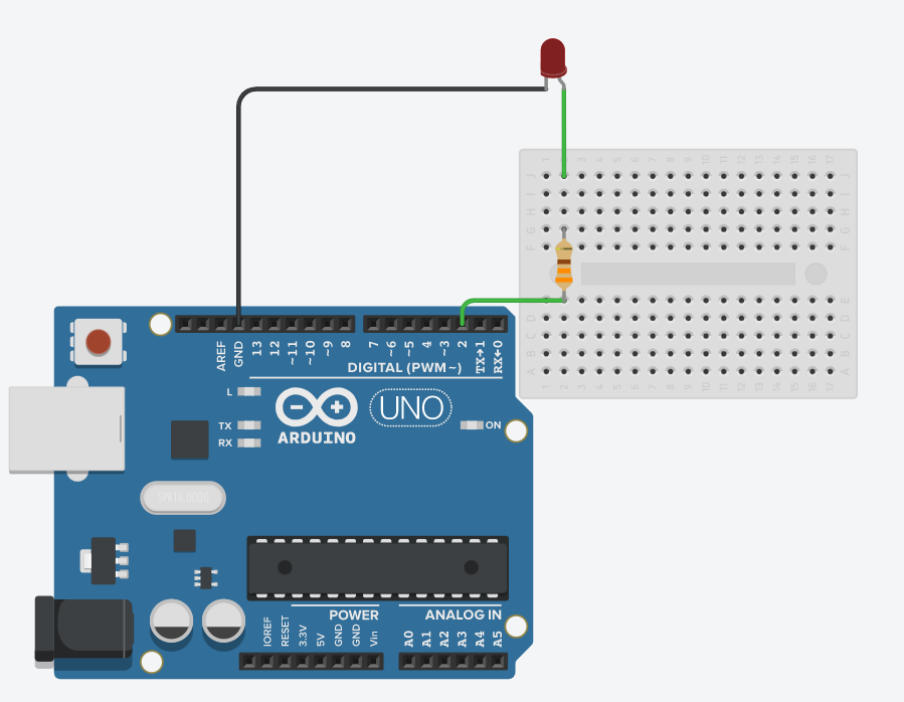
2.2. Smart Home Kit for Arduino – Keyestudio

Smart Home Kit for Arduino je jedan od niza kompleta koje je stvorila tvrtka Keyestudio. Ovaj komplet sadrži preko 30 različitih komponenti i senzora, a priručnik za spajanje i pisanje programskog koda je moguće pronaći na internetu. Sastoji se od 14 projekata za sve pojedine komponente koje se nalaze u kompletu te posljednji 15.ti projekt u kojem je objašnjeno kako sve zajedno spojiti u maketu pametne kuće. Sve komponente u ovom kompletu su vlastiti moduli tvrtke, stoga je spajanje komponenti u nekim slučajevima drugačije od uobičajenog.

3. DIJELOVI EDUKATIVNE ARUDINO KUTIJE

3.1. Treptajuće svijetlo

Standardna LE (Light Emmiting) dioda se sastoji od dvije nožice, a to su anoda (duža) i katoda (kraća). Kako bi dioda emitirala svijetlost na anodi se mora nalaziti veći napon nego na katodi, a obično se radi o naponima od 3.3V i 0V. Također, svaka dioda ima svoj takozvani napon praga koji mora biti osiguran kako bi dioda ispravno radila. To se postiže preko Ohmovog zakona dodavanjem dodatnog otpora u strujni krug kako kroz diodu ne bi potekla prevelika struja. Katoda se spaja na uzemljenje Arduino pločice (0V ili GND), a anoda preko otpornika na digitalni pin sa kojim se upravlja diodom. Shematski prikaz opisanog strujnog kruga se nalazi na slici 1. Treptajuće svijetlo.



Sl. 1. Treptajuće svijetlo

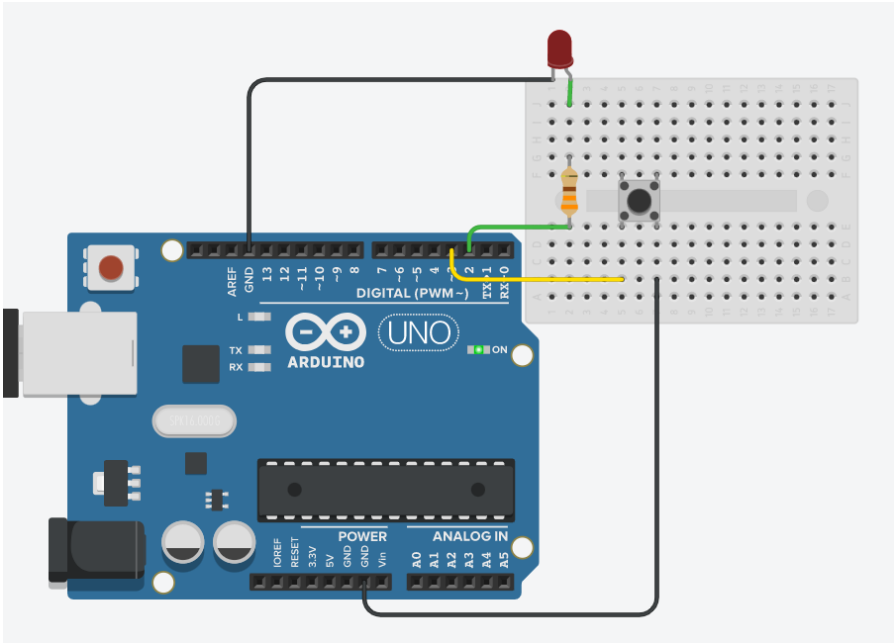
Za ispravan rad diode sa slike 1. Treptajuće svijetlo potrebno je unutar `setup()` dijela programa navesti naredbu `pinMode(2,OUTPUT);` koja inicijalizira digitalni pin 2 Arduino pločice kao pin izlaznog signala. Zatim je moguće unutar `loop()` dijela programa pomoću naredbi `digitalWrite(2,HIGH);`, `digitalWrite(2,LOW);` i `delay(milisekunde);` upravljati paljenjem i gašenjem diode preko digitalnog pina 2 u određenim vremenskim intervalima. Cijeli kod je vidljiv u tablici 1. Treptajuće svijetlo.

Tablica 1. Treptajuće svijetlo

Linija	***Kod***
1:	`void setup(){`
2:	`pinMode(2, OUTPUT);`
3:	`}`
4	`void loop(){`
5:	`digitalWrite(2, HIGH);`
6:	`delay(1000);`
7:	`digitalWrite(2, LOW);`
7:	`delay(1000);`
8:	`}`

3.2. Tipkalom upravljana LE dioda

Klasično tipkalo (eng. button) u elektronici ima četiri nožice od kojih su po dvije spojene. Kada se pritisne tipkalo dolazi do zatvaranja ili prekida strujnog kruga u ovisnosti o izvedbi tipkala.



Sl. 2. Tipkalom upravljana LE dioda

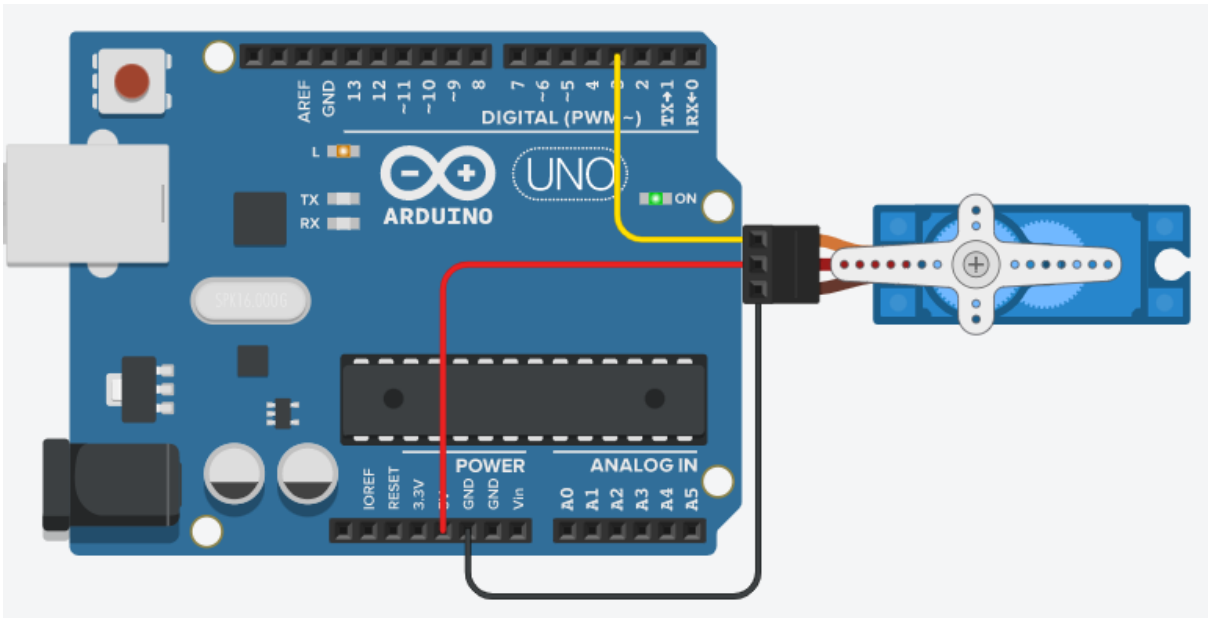
Tipkalo se spaja na uzemljenje (GND pin) i na digitalni pin na Arduino pločici. Uz to se koristi LE dioda kao u prethodnom dijagramnu sa slike 1. Treptajuće svijetlo. Uz tipkalo se spaja otpornik između digitalnog pina 3 i nožice tipkala. Međutim, ukoliko se u kodu koristi `INPUT_PULLUP` parametar prilikom inicijalizacije digitalnog pina 3 umjesto `INPUT` otpornik nije potreban. `INPUT_PULLUP` koristi unutarnji otpornik Arduino pločice između u vrijednosti od 20kΩ do 50kΩ koji se postavlja između digitalnog pina i napona od 5V te time osigurava da stanje na digitalnom pinu bude `HIGH` sve dok se ne pritisnemo tipkalo kada će stanje postati `LOW`. Detaljniji prikaz koda u kojem se LE dioda pali na 1 sekundu pritiskom tipkala je prikazan na tablici 2. Tipkalom upravljana LE dioda.

Tablica 2. Tipkalom upravljana LE dioda

Linija	***Kod***
1:	`void setup(){`
2:	`pinMode(2, OUTPUT);`
3:	`pinMode(3, INPUT_PULLUP);`
4:	`}`
5:	`void loop(){`
6:	`int btn_stanje=digitalRead(3);`
7:	`if(btn_stanje==LOW){`
8:	`digitalWrite(2, HIGH);`
9:	`delay(1000);`
10	`}`
11:	`else{`
12:	`digitalWrite(2,LOW);`
13:	`}`
14:	`}`

3.3. Servo motor

Servo motori su motori kojima je moguće jako precizno postaviti kut, odnosno rotirati točno koliko želimo. Imaju ugrađen servo mehanizam pomoću kojeg uvijek raspoznaju u kojem se kutu nalaze te je potrebno sa postaviti krajnji kut koji mi želimo da bude. Tri žice servo motora se spajaju na Arduino pločicu tako da smeđa predstavlja upravljački pin sa Arduino pločice, crvena pozitivan napon 5V i crna uzemljenje. Za programiranje servo motora potrebna je Servo biblioteka koju je moguće instalirati kroz Arduino IDE. Inicijalizacija motora na digitalni pin se vrši naredbom `myservo.attach()` koja kao argument prima redni broj pina na koji se postavlja. Nulta pozicija (kut) se postiže naredbom `myservo.write(0)`. Na slici 3. Servo motor je prikazan dijagram spajanja servo motora.



Sl. 3. Servo motor

Kako bi navedene naredbe mogle raditi, osim uključivanja Servo biblioteke potrebno je stvoriti objekt servo motora, a se radi na samom početku koda iznad `setup()` dijela navođenjem biblioteke i zatim imena objekta `Servo myservo`. Detaljan prikaz koda gdje će se servo pozicionirati na nultu poziciju prikazan je u tablici 3. Servo motor.

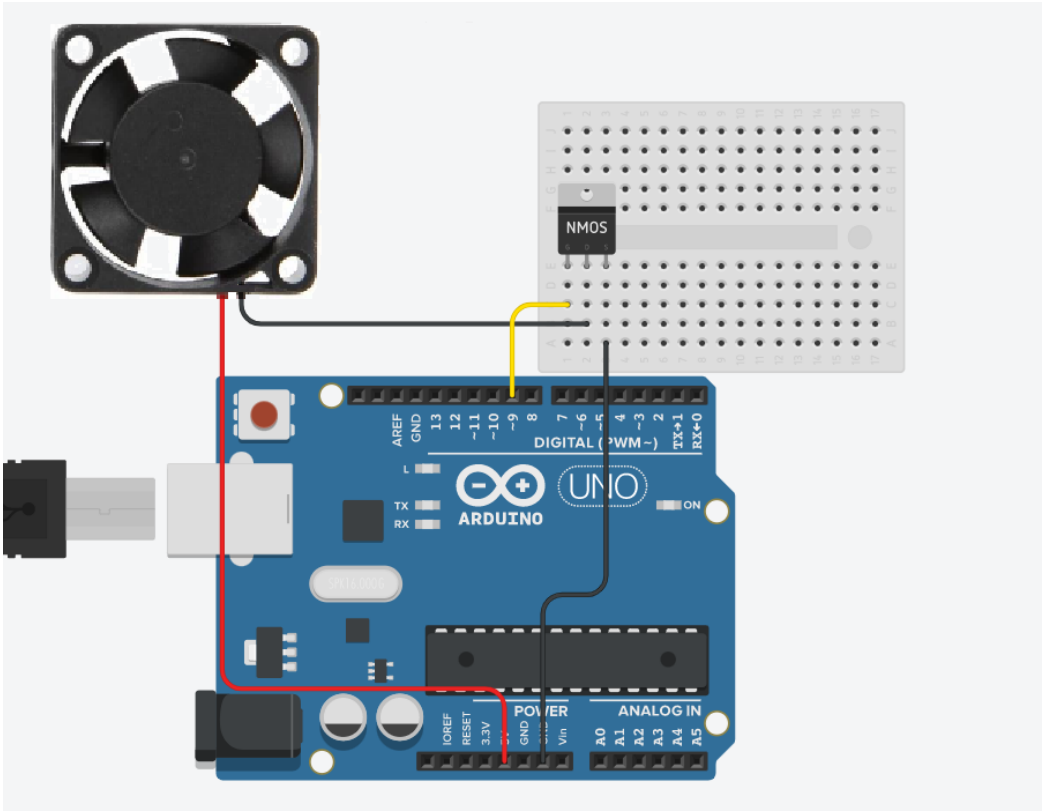
Tablica 3. Servo motor

Linija	***Kod***
1:	`#include <Servo.h>`
2:	`Servo myservo;`
3:	`void setup(){`
4:	`myservo.attach(3);`
5:	`}`
6:	`void loop(){`
7:	`myservo.write(0);`
8:	`}`

3.4. Ventilator

Ventilator je istosmjerni (DC) motor. Najprikladniji za prikaz dijagrama spajanja je motor od 5V, jer na njega se ne mora spajati dodatni izvor napajanja. Zbog ograničenja struje, ventilator je potrebno spojiti preko tranzistora na Arduino pločicu, a prikazani tranzistor je N-kanalni

MOSFET. Na njemu se nalaze tri nožice sa oznakama G (gate) D (drain) i S (source). Nožica S se spaja na uzemljenje (GND pin), D na negativni pol ventilatora, G je upravljački pin te se spaja na digitalni pin Arduino pločice, a pozitivni pol ventilatora na izvor napajanja (5V pin). Na slici 4. Ventilator je prikazan dijagram spajanja ventilatora preko tranzistora.



Sl. 4. Ventilator

Prikazani upravljački pin 9 spojen na G nožicu tranzistora je odabran jer ima oznaku ~ pored brojčane vrijednosti pina. To označava da je pin sposoban za PWM (Pulse-width modulation) ili modulaciju širine impulsa. Takvom modulacijom moguće je digitalni signal maskirati u analogni promjenom širine impulsa signala, te u ovisnosti o širini impulsa moguće je postići različite vrijednosti napona od 0V do 5V. Konkretno u jednom periodu signal dio vremena ima vrijednost `HIGH`, dok dio vremena ima vrijednost `LOW`. Naprimjer, signal koji pola perioda ima vrijednost `HIGH`, a pola `LOW` ima napon 2.5V od mogućih 5V. Ovako se može upravljati ventilatorom preko više brzina kontrolirajući iznos napona na digitalnom pinu Arduino pločice. Iznos napona se kontrolira sa vrijednostima od 0 do 255, jer je vrijednost pina u 8 bitnom zapisu stoga ukupno može biti 256 različitih vrijednosti. Funkcija `analogWrite(9,255)` u `loop()` dijelu koda

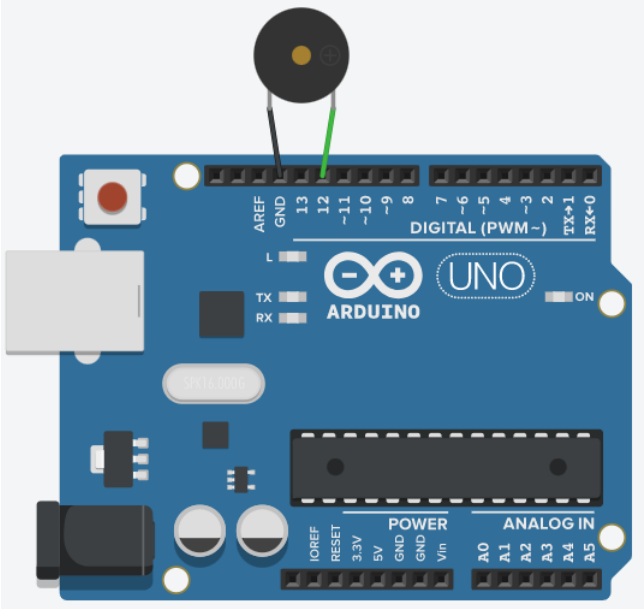
podešava ventilator na najveću moguću brzinu. Cijeli kod za ventilator koji ima dvije brzine vrtnje i stop vrtnje je prikazan u tablici 5. Ventilator.

Tablica 4. Ventilator

Linija	*Kod*
1:	`void setup(){}`
2:	`void loop(){`
3:	`analogWrite(9,0);`
4:	`delay(2000);`
5:	`analogWrite(9,130);`
6:	`delay(2000);`
7:	`analogWrite(9,255);`
8:	`delay(2000);`
9:	`}`

3.5. Pasivni buzzer

Buzzeri su male zujalice kojima je moguće podešavati frekvenciju kako bi proizvodili određen ton te frekvencije. Postoje aktivni i pasivni Piezzo buzzeri. Aktivni imaju ugrađen oscilator koji generira konstantnu frekvenciju te se aktiviraju postavljanjem stanja `HIGH` na pinove, dok su pasivni nešto jeftiniji jer nemaju ugrađen osciloskop te se kontroliraju dovođenjem izmjeničnog signala na pinove. Raspoznaju se po tome što pasivni imaju oznaku HX na kućištu, dok aktivni nemaju. U pokusu se koristi pasivni buzzer koji ima dvije nožice. Pozitivna nožica sa oznakom +



Sl. 5. Pasivni buzzer

na kućištu se spaja na digitalni pin 12 Arduino pločice, a negativna nožica na uzemljenje pin (GND pin). Dijagram spajanja je prikazan na slici 5. Pasivni Buzzer.

Zvuk se generira pomoću funkcije `tone()` koja mora primiti parametre rednog broja pina i frekvencije zvuka kojeg treba proizvesti, odnosno note. Dodatno može primiti treći parametar koji govori koliko milisekundi će se izvoditi određena nota. Funkcija `noTone()` prima samo oznaku broja pina te prekida izvođenje zvuka na zadanom pinu. U tablici 5. Pasivni buzzer je prikazan kod jednostavnog programa koji svira ton note C4 frekvencije 262Hz tokom jedne sekunde, a zatim slijedi sekunda pauze bez ikakvog tona.

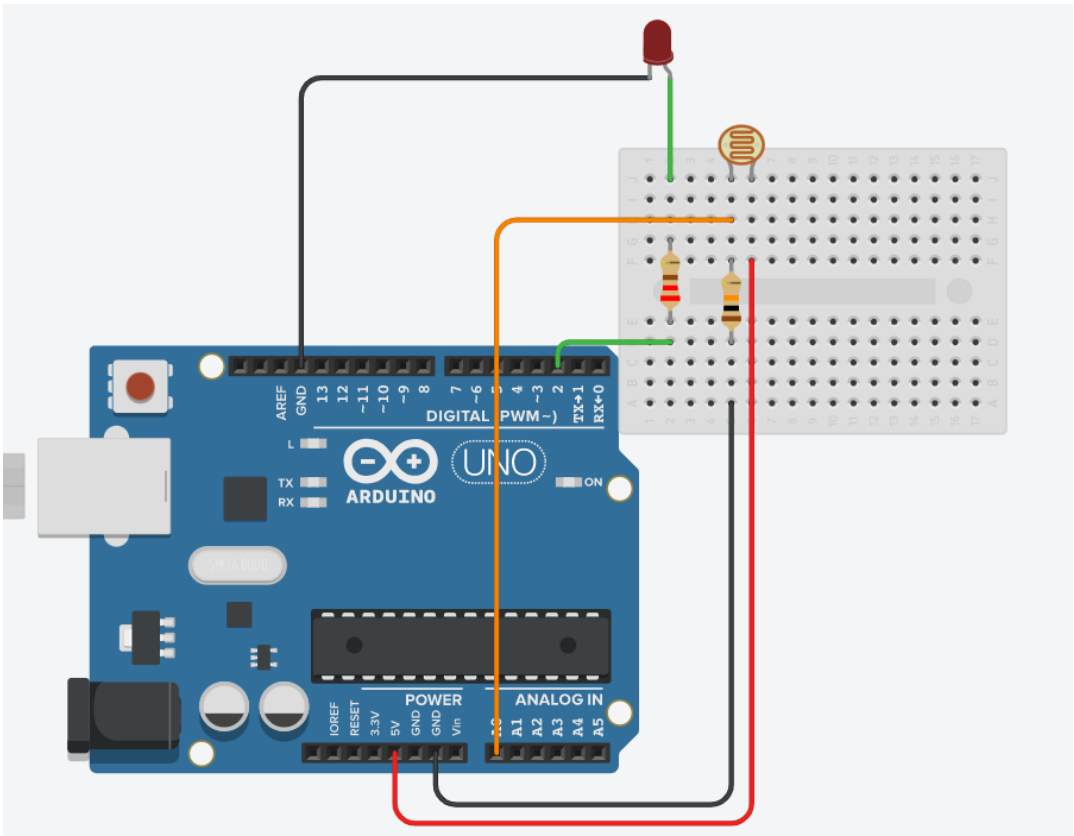
Tablica 5. Pasivni buzzer

Linija	***Kod***
1:	`void setup() {`
2:	`tone(12, 262, 1000);`
3:	`delay(2000);`
4:	`noTone(12);`
5:	`}`
6:	`void loop() {}`

Budući da se funkcije `tone()` i `noTone()` nalaze u `setup()` dijelu programa zvuk će se čuti samo jednu sekundu. Kako bi se ponavljao zvuk svake sekunde funkcije `tone()` i `noTone()` se navode unutar `loop()` dijela programa umjesto `setup()`. Frekvencije za sve note je moguće pronaći u datoteci pitches.h na internetu koja se može dodati u projekt i inicijalizirati u kodu pozivom naredbe `#include<pitches.h>`.

3.6. Photocell sensor

Photocell sensor ili photoresistor su engleski nazivi za svjetlosni otpornik kojem otpornost ovisi o intenzitetu svijetla. Porastom intenziteta svijetla otpornost otpornika se smanjuje od maksimalne vrijednosti 40kΩ do minimalne vrijednosti10kΩ, a isto vrijedi i obratno. Koristi se kao naponsko dijelilo, stoga jedna nožica se spoja na analogni pin, na što se u paralelu dodaje otpornik od 10kΩ spojen na uzemljenje. Druga nožica se spaja na izvor napona od 5V (5V pin), a radi prikaza rezultata može se koristiti LE dioda spojena kao na slici 1. Treptajuće svijetlo. Cijeli dijagram spajanja strujnog kruga je vidljiv na slici 6. Photocell sensor.



Sl. 6. Photocell sensor

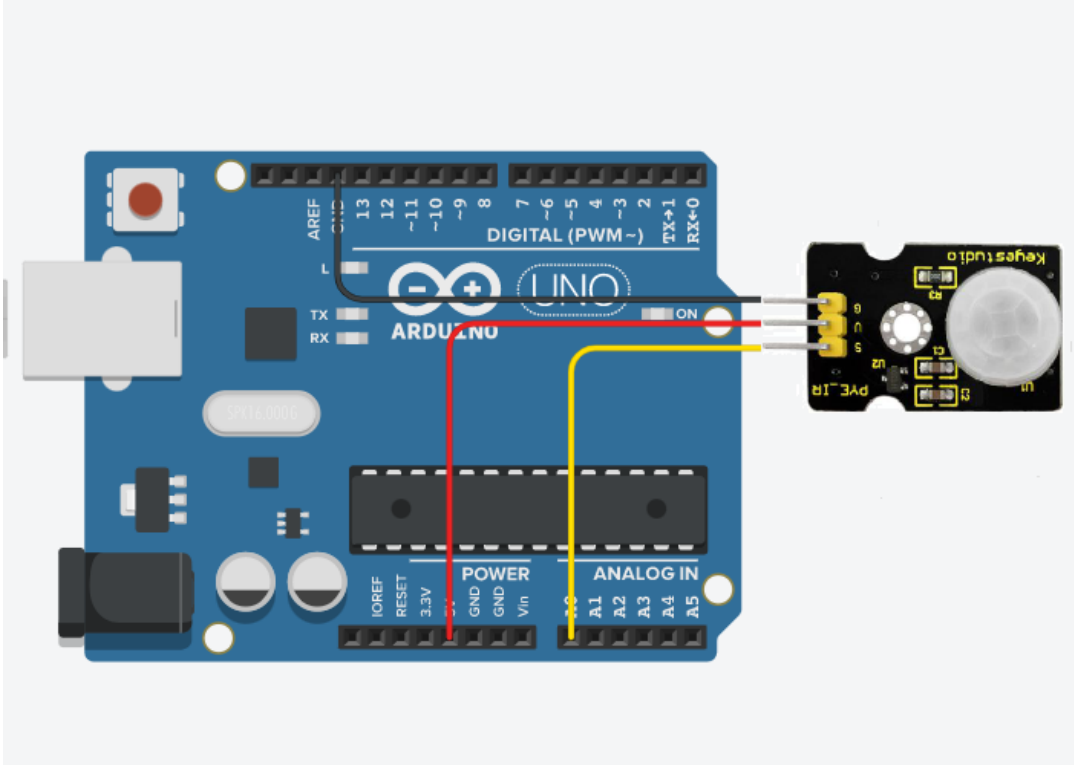
Kako bi dioda zasvijetlila potrebno je dovesti svijetlo na svjetlosni otpornik, a u ovisnosti intenziteta svijeta zavisit će jačina svijetlosti diode. Naredbom `analogRead(A0)` se očitava analogna vrijednost napona u rasponu od 0 do 1023 sa digitalnog pina A0, zato što je vrijednost očitanja zapisana u 10 bitnom zapisu što podrazumijeva maksimalan zapis 1024 vrijednosti. Funkcijom `map(sensorValue, 0, 1023, 0, 255);` mapira se raspon vrijednosti iz 10 bitnog zapisa u 8 bitni, odnosno umjesto 1024 različite vrijednosti može se dobiti 256 različitih vrijednosti. To je potrebno napraviti jer digitalni izlazi Arduino pločice rade sa zapisom vrijednosti u 8 bita. Cijeli kod očitanja vrijednosti sa svjetlosnog otpornika i postavljanja vrijednosti jačine svijetla diode je prikazan u tablici 6. Photocell sensor.

Tablica 6. Photocell sensor

Linija	***Kod***
1:	`void setup() {}`
2:	`void loop() {`
3:	`int sensorValue = analogRead(A0);`
4:	`sensorValue = map(sensorValue, 0, 1023, 0, 255);`
5:	`analogWrite(2, sensorValue);`
6:	`}`

3.7. Senzor vodene pare

Steam sensor ili senzor vodene pare se temelji na principu detekcije količine vode na paralelnim ispisanim linijama strujnog kruga na pločici senzora. Što ima više vode ili vodene pare, više je linija spojeno te raste vodljivost površine, a time raste i iznos izlaznog napona. Iznos izlaznog napona možemo očitati analognim pinom. Potrebno je spojiti nožicu senzora sa oznakom S na analogni pin Arduino pločice. Zatim se nožica s oznakom V (VCC) spaja na pozitivni napon od 5V (5V pin), a nožica sa oznakom G se spaja na uzemljenje. Dijagram spajanja senzora vodene pare je prikazan na slici 7. Senzor pare.



Sl. 7. Senzor pare

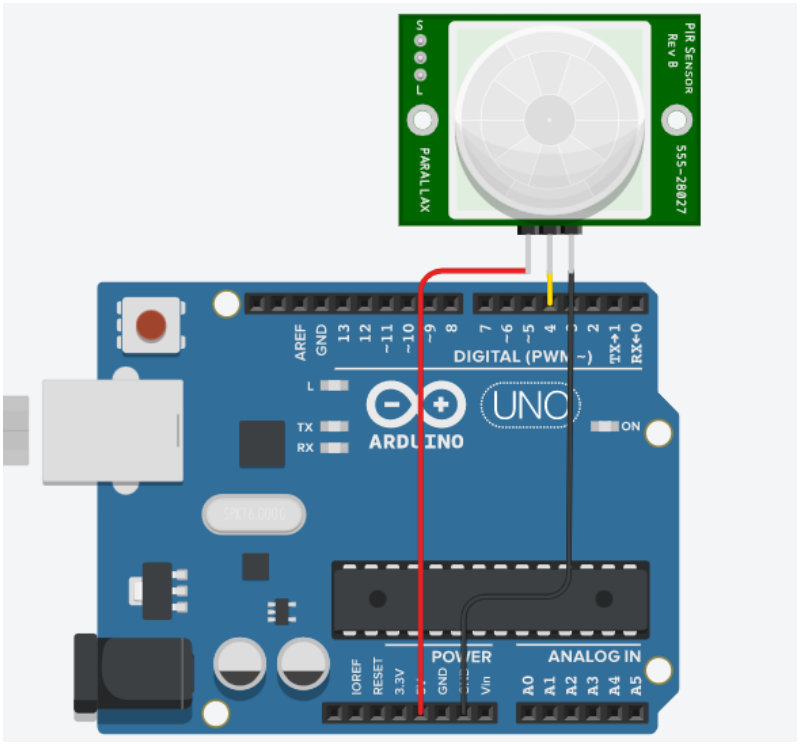
Za očitavanje vrijednosti sa analognog pina opet se može koristiti funkcija `analogRead();`, a da bi se prikazali rezultati može se koristiti serijska komunikacija između računala i Arduino pločice koju treba inicijalizirati naredbom `Serial.begin(9600);` gdje broj 9600 označava brzinu prijenosa u baudima, odnosno količinu prenesenih bita u jednoj sekundi. Funkcijom `Serial.println(sensorValue);` se na ekranu računala pomoću serijskog monitora ispisuju vrijednosti količine vodene pare u rasponu od 0 do 1023. Cijeli kod u kojem se vrijednost sa senzora očitava analognim pinom A0 te potom ispisuje na zaslon računala svakih 200 milisekundi je prikazan u tablici 7. Senzor pare.

Tablica 7. Senzor pare

Linija	*Kod*
1:	`void setup() {`
2:	`Serial.begin(9600);`
3:	`}`
4:	`void loop(){`
5:	`int sensorValue = analogRead(0);`
6:	`Serial.println(sensorValue);`
7:	`delay(200);`
8:	`}`

3.8. Senzor pokreta

Pasivni infracrveni senzor (PIR senzor) predstavlja modul sa HC-SR501 senzorom koji može detektirati kretanje objekata koji zrače infracrveno svijetlo poput ljudskog tijela, zbog toga je upotreba ovog senzora veoma učestala. Početno postavljanje i kalibracija senzora traje oko 10 do 60 sekundi. Senzor ima tri nožice, od kojih jedna predstavlja izlazni signal senzora detekcije pokreta. Ukoliko senzor ne detektira pokret signal će imati vrijednost nula, ali ukoliko signal ima bilo koju drugu vrijednost znači da je pokret detektiran. Lijeva nožica senzora se spaja na pozitivni izvor napona od 5V, desna nožica senzora se spaja na uzemljenje, dok srednja nožica senzora



Sl. 8. Senzor pokreta

predstavlja signalnu nožicu koja šalje očitane vrijednosti sa senzora na Arduino pločicu. Na slici 8. Senzor pokreta je prikazan jednostavni dijagram spajanja senzora pokreta sa Arduino pločicom.

Signalni pin na slici 8. Senzor pokreta je digitalni pin 4 Arduino pločice te je pomoću njega moguće detektirati vrijednost kretanja sa senzora. Naredbom `pinMode(4,INPUT);` postavljamo digitalni pin 4 kao ulaz signala sa senzora, a funkcijom `Serial.println();` se u redovima ispisuje na zaslonu računala je li pokret detektiran ili nije. Dodatno se koristi `delay();` funkcija na početku svakog mjerenja kako bi se preciznije očitala vrijednost sa senzora i ispisivala na zaslonu računala u određenom vremenskom intervalu koji se zadaje kao parametar `delay();` funkcije u milisekundama. Cijeli kod programa za detekciju pokreta je ispisan u tablici 8. Senzor pokreta.

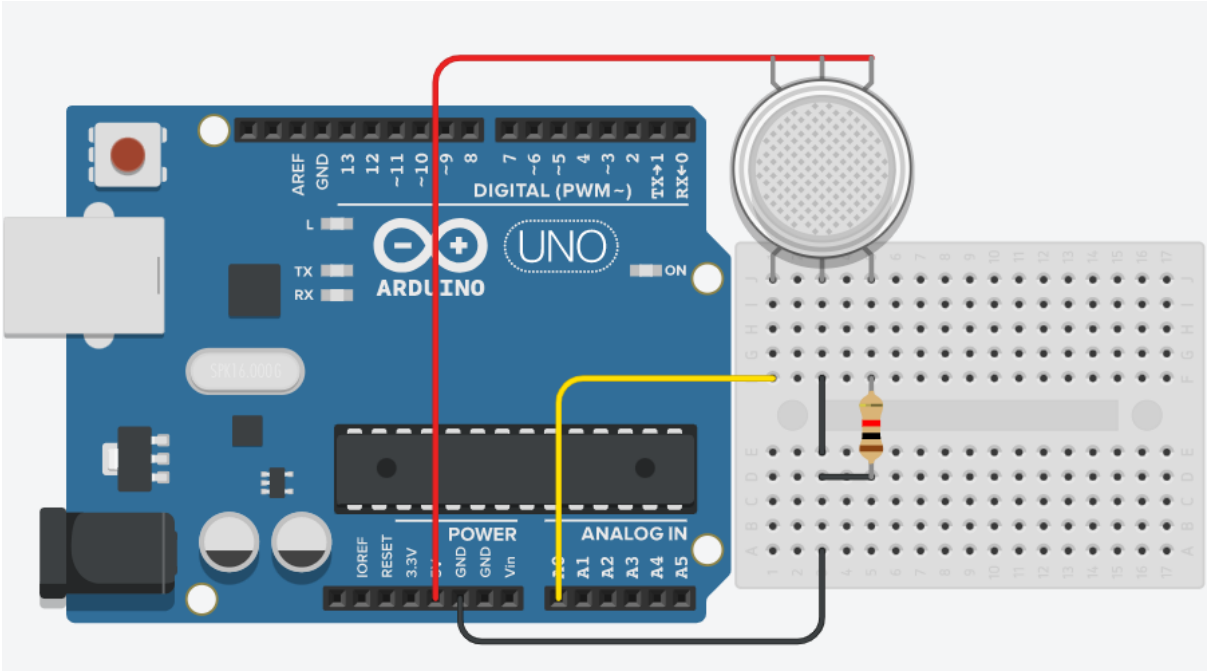
Tablica 8. Senzor pokreta

Linija	***Kod***
1:	`void setup() {`
2:	`Serial.begin(9600);`
3:	`pinMode(2,INPUT);`
4:	`}`
5:	`void loop(){`
6:	`delay(500);`
7:	`if (digitalRead(2) == HIGH) {`
8:	`Serial.println("Kretanje zabilježeno");}`
9:	`else{Serial.println("Nema kretanja");}`
10:	`}`

3.9. Senzor detekcije plina

Najčešće korišten senzor za detekciju plina je MQ-2 senzor. Ovaj senzor plina koristi se za alarme za curenje plina u kućanstvu, industrijske alarme zapaljivih plinova i prijenosne instrumente za detekciju plina. Pogodan je za detekciju ukapljenog plina, benzena, alkana, alkohola, vodika te se široko koristi u raznim sustavima za dojavu požara. Može otkriti koncentraciju zapaljivog plina i dima u rasponu od 300 do 10000 ppm i ima visoku osjetljivost na prirodni plin, ukapljeni naftni plin i drugi dim, osobito na dim alkana. Ima prednosti visoke osjetljivosti, brzog odziva, dobre stabilnosti, dugog vijeka trajanja i jednostavnog pogonskog kruga. Za potrebe ovog projekta korišten je detektor dima dostupan na Tinkercad web aplikaciji za simulaciju Arduino sklopovlja. On posjeduje 6 nožica, od kojih se gornje tri spajaju zajedno na izvor napona od 5V. Donja lijeva nožica predstavlja signalni pin pomoću kojeg se može očitati vrijednost napona od 0 do 5V sa senzora na Arduino pločici te se spaja na analogni pin. Srednja donja nožica se spaja na uzemljenje,

a na nju se u paraleli spaja desna donja nožica sa otpornikom od 1kΩ u seriji. Detaljan prikaz dijagrama spajanja senzora detekcije plina je prikazan na slici 9. Senzor detekcije plina.



Sl. 9. Senzor detekcije plina

Signalni pin senzora na slici 9. Senzor detekcije plina je spojen na analogni pin A0 na Arduino pločici. Funkcijom `analogRead();` se može očitati vrijednost signala sa senzora u rasponu od 0 do 1023, odnosno od 0V do 5V te funkcijom `map(vrijednost,0,1023,0,255);` mapiramo 10 bitni zapis vrijednosti signala u 8 bitni zapis pogodan za daljnju obradu preko Arduino pinova. Koristeći serijski monitor moguće je na računalu ispisati vrijednosti očitanja senzora. Ukoliko 10 bitna vrijednost otprilike prelazi 450 to predstavlja da se radi o velikoj koncentraciji plina, a to odgovara 8 bitnoj vrijednosti od 112. U tablici 9. Senzor detekcije plina je prikazan kod očitanja

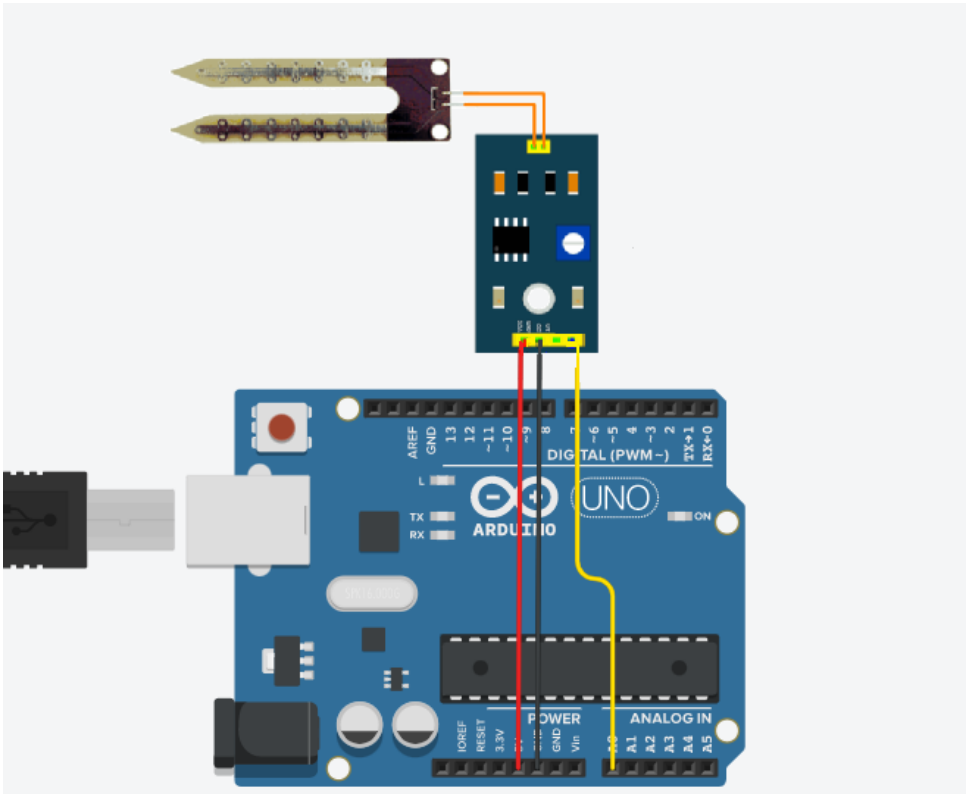
Tablica 9. Senzor detekcije plina

Linija	***Kod***
1:	`void setup() {`
2:	`Serial.begin(9600);`
3:	`}`
4:	`void loop(){`
5:	`delay(1000);`
6:	`int vrijednost=analogRead(A0);`
7:	`vrijednost=map(vrijednost,0,1023,0,255);`
8:	`Serial.println(vrijednost);`
9:	`}`

vrijednosti sa senzora detekcije plina te se potom očitane vrijednosti ispisuju na računalu putem serijskog monitora.

3.10. Senzor vlažnosti tla

Za mjerenje vlažnosti tla najčešće korišten senzor je FC-28 senzor vlažnosti tla. Ako tlu nedostaje vode, analogna vrijednost signala koji senzor šalje će biti velika i obratno. Senzor se sastoji od dvije sonde koje se koriste za mjerenje volumetrijskog sadržaja vode. Dvije sonde propuštaju struju kroz tlo, što mijenja vrijednost otpora u ovisnosti o količini vlage koja se nalazi u tlu. Suho tlo slabo provodi struju, što daje velike vrijednosti otpora, dok vlažno tlo dobro provodi struju te će vrijednosti otpora biti niža. U projektu se koristi analogna metoda očitanja vrijednosti sa senzora, odnosno manjak vode u tlu će davati veću količinu otpora te će i očitana vrijednost na analognom pinu biti veća. Za jako vlažno tlo, otpornost će se smanjiti te će se i očitana vrijednost analognog pina biti manjeg iznosa. Modul senzora vlažnosti tla se sastoji od dva dijela. Jedan dio su dvije sonde koje se zabadaju u tlo, a drugi dio je strujni krug na pločici senzora sa potenciometrom koji služi za podešavanja osjetljivosti senzora prilikom korištenja digitalne metode očitanja vrijednosti vlažnosti tla. Modul senzora vlažnosti tla ima četiri nožice od koji se tri koriste prilikom analognog očitanja vrijednosti tla. VCC nožica se spaja na pozitivni izvor



Sl. 10. Senzor vlažnosti tla

napona od 5V, GND nožica se spaja na uzemljenje, a A0 nožica se spaja na analogni pin Arduino pločice. Na slici 10. Senzor vlažnosti tla je prikazan dijagram spajanja senzora na Arduino pločicu, gdje je A0 nožica spojena na analogni pin A0.

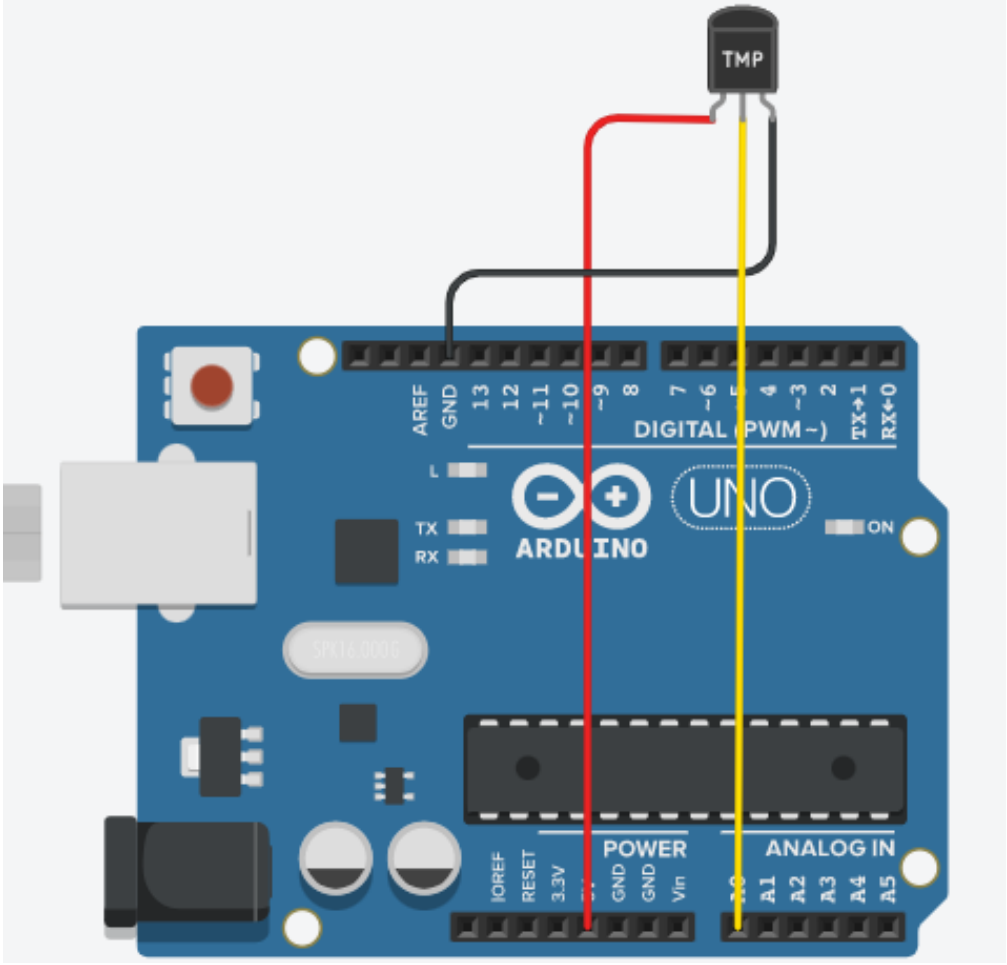
Funkcijom `analogRead(A0);` se očitava vrijednost signala na analognom pinu A0. Vrijednost je zapisana u 10 bitnom zapisu, a kako bi se saznala prava vrijednost vlažnosti u postotcima potrebno je mapirati očitanu 10 bitnu vrijednost u postotke. Za to se koristi funkcija `map(output_value,1023,0,0,100);` iz koje je vidljivo da će vrijednost biti 0% ukoliko signal na A0 pinu ima vrijednost 1023, te 100% za vrijednost 0. Dalje se može koristiti serijski monitor za ispis očitanih vrijednosti tla. U tablici 10. Senzor vlažnosti tla je prikazan primjer koda očitavanja i ispisa vrijednosti vlažnosti tla na serijskom monitoru analognom metodom gdje će se svake sekunde ispisivati vrijednost vlažnosti tla u postotcima.

Tablica 10. Senzor vlažnosti tla

Linija	***Kod***
1:	`void setup() {`
2:	`Serial.begin(9600);`
3:	`}`
4:	`void loop(){`
5:	`delay(1000);`
6:	`int vrijednost=analogRead(A0);`
7:	`vrijednost=map(vrijednost,1023,0,0,100);`
8:	`Serial.print("Važnost tla iznosi: ");`
9:	`Serial.print(vrijednost);`
10:	`Serial.println("%");`
11:	`}`

3.11. Senzor temperature

Za mjerenje temperature u projektu će se koristiti TMP36 senzor temperature. On može mjeriti prilično širok raspon temperature (-50 ° C do 125 ° C), veoma je precizan (razlučivost 0,1 ° C), jako je pristupačan (niska cijena) i ne zahtijeva nikakve dodatne komponente. No, nije otporan na vremenske uvijete stoga ga je za vanjsku upotrebu potrebno dodatno zaštiti. Sastoji se od 3 nožice, od kojih se prva (lijeva) sa oznakom VCC spaja na izvor napona od 5V na Arduino pločici. Druga (srednja) nožica predstavlja signalnu nožicu senzora koja daje vrijednost napona za određenu temperaturu te se spaja na analogni pin Arduino pločice, a treća (desna) nožica sa oznakom GND se spaja na uzemljenje. Dijagram spajanja TMP36 senzora sa Arduino pločicom je prikazan na slici 11. Senzor temperature gdje je signalni pin senzora spojen na analogni pin A0, VCC na 5V pin i GND na GND pin pločice.



Sl. 11. Senzor temperature

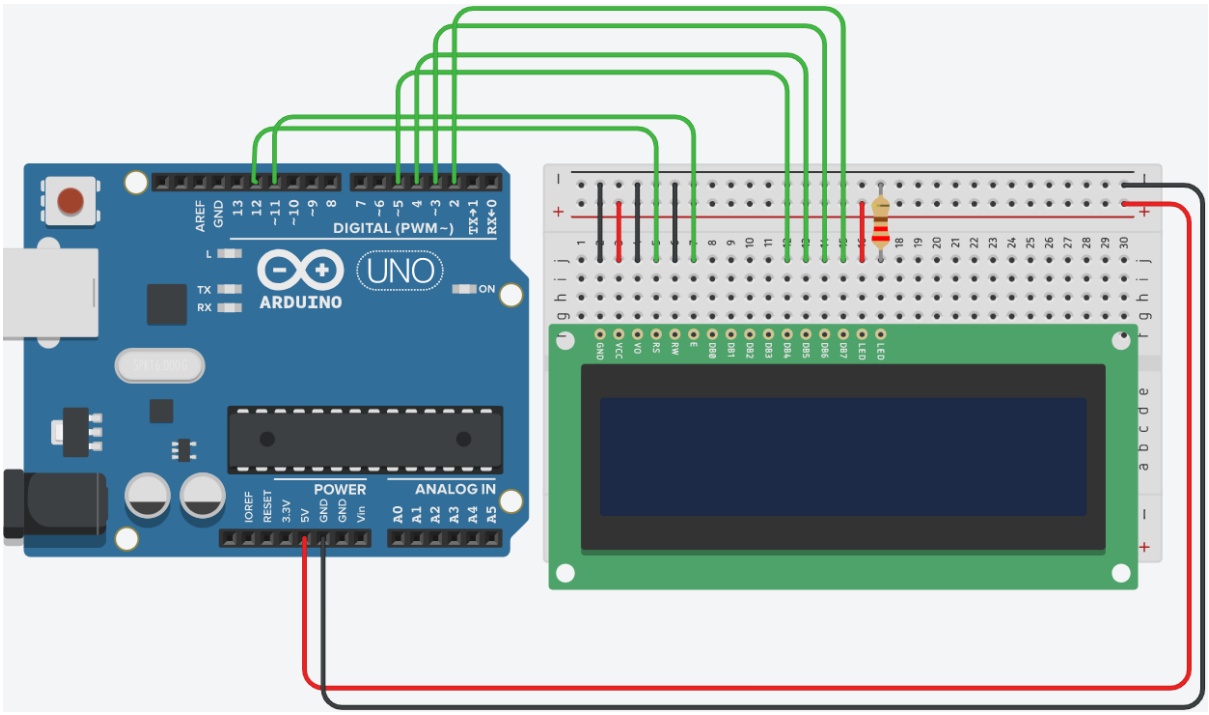
Za razliku od termistora, TMP36 nema otpornik osjetljiv na temperaturu. Umjesto toga ovaj senzor koristi svojstvo dioda, kako dioda mijenja temperaturu, napon se s njom mijenja poznatom brzinom. Senzor mjeri male promjene i na temelju toga daje analogni napon između 0V i 1,75V (DC). Očitana vrijednost se množi sa brojem 5 (ukoliko je Arduino pločica spojena na računalo putem USB-a potrebno je pomnožiti sa 4.68), zbog izvora napona Arduino pločice od 5V kako bismo dobili iznos napona u voltima, zatim se dijeli sa 1024 kako bi očitana vrijednost bila izražena u procentima te se oduzima vrijednost pomaka 0.5 i u konačnici množi sa 100 da bi se izvršila pretvorba u stupnjeve C. Funkcijom `analogRead(A0);` je moguće očitati početnu vrijednost signala, a krajnji rezultat je moguće ispisati na računalu uz pomoć serijskog monitora. Cijeli kod očitavanja vrijednosti TMP36 senzora preko analognog pina A0 na Arduino pločici i ispisa rezultata preko serijskog monitora svake sekunde je prikazan u tablici 11. Senzor temperature.

Tablica 11. Senzor temperature

Linija	*Kod*
1:	`void setup() {`
2:	`Serial.begin(9600);`
3:	`}`
4:	`void loop(){`
5:	`delay(1000);`
6:	`int vrijednost=analogRead(A0);`
7:	`float voltage = vrijednost * 5;`
8:	`voltage /= 1024.0;`
9:	`float temperatureC = (voltage - 0.5) * 100;`
10:	`Serial.print(temperatureC);`
11:	`Serial.println(" stupnjeva C");`
12:	`}`

3.12. LCD zaslon

LCD (Liquid crystal display) zasloni se proizvode u raznim dimenzijama, a najpopularniji je zaslon sa po 16 mogućih znakova u dva reda. Njihova primjena je široka, od kalkulatora i starijih modela digitalnih satova pa do modernih LED zaslona koji su zapravo LCD zasloni. Za upotrebu je potrebno koristiti Liquid Crystal biblioteku koja se inicijalizira naredbom `#include <LiquidCrystal.h>`. Modul LCD zaslona ima 16 pinova, a prvi slijeva nadesno je uzemljenje. Drugi pin je VCC koji se povezuje sa pinom od 5V na Arduino pločici. Slijedi Vo pin na koji se može pričvrstiti potenciometar za kontrolu kontrasta zaslona. Zatim se RS pin ili pin za odabir registra koristi za odabir hoćemo li poslati naredbe ili podatke na LCD. Naprimjer, ako je RS pin postavljen na `LOW` stanje ili nula volti, tada se na LCD šalju naredbe poput: postavite pokazivač na određeno mjesto, izbrišite zaslon, isključite zaslon itd., a kad je RS pin postavljen na stanje `HIGH` ili 5V, šalju se podaci ili znakovi na LCD zaslon. Nakon RS pina slijedi R / W pin koji odabire način rada hoćemo li čitati ili pisati na LCD zaslon, odnosno koristi se za pisanje ili slanje naredbi i podataka na LCD zaslon. Slijedi E pin koji omogućuje pisanje u registre, odnosno sljedećih 8 pinova od D0 do D7 kroz koje je moguće poslati 8 bitne podatke prilikom pisanja u registre. Naprimjer, ako želimo vidjeti potonje veliko A na ekranu, potrebno je poslati 0100 0001 u registre prema ASCII tablici. Posljednja dva pina A i K ili anoda i katoda služe za LED pozadinsko osvjetljenje. Dijagram spajanja LCD zaslona sa Arduino pločicom je prikazan na slici 12. LCD zaslon.



Sl. 12. LCD zaslon

U primjeru koda za ovaj projekt će se koristiti 6 digitalnih ulaznih pinova s Arduino pločice. Registri LCD zaslona od D4 do D7 bit će spojeni na digitalne pinove od 2 do 5 Arduino pločice. Pin E se treba spojiti na pin 11, a RS pin na pin broj 12 na Arduino pločici. R/W i Vo pin se spajaju na uzemljenje, a VCC i anoda sa LCD zaslona se spajaju na pozitivni izvor napona 5V na Arduino pločici. Također, radi postizanja odgovarajućeg kontrasta zaslona potrebno je spojiti katodu LCD zaslona u seriji sa otpornikom od 220 na uzemljenje. Naredbom `LiquidCrystal lcd(12, 11, 5, 4, 3, 2);` postavljamo spojene pinove sa dijagrama u programski kod, a naredbom `lcd.begin(16, 2);` se inicijalizira LCD 1602 zaslon. Funkcijom `lcd.setCursor(0,0);` se postavlja početno mjesto ispisa znakova na LCD zaslonu, dok će se naredbom `lcd.print("FERIT!");` na zaslonu ispisati „FERIT!“. Cijeli primjer koda za LCD zaslon koji će u jednom redu ispisati „FERIT!“, a u drugom „Arduino“ sa dva prazna znaka ispred je prikazan u tablici 12. LCD zaslon.

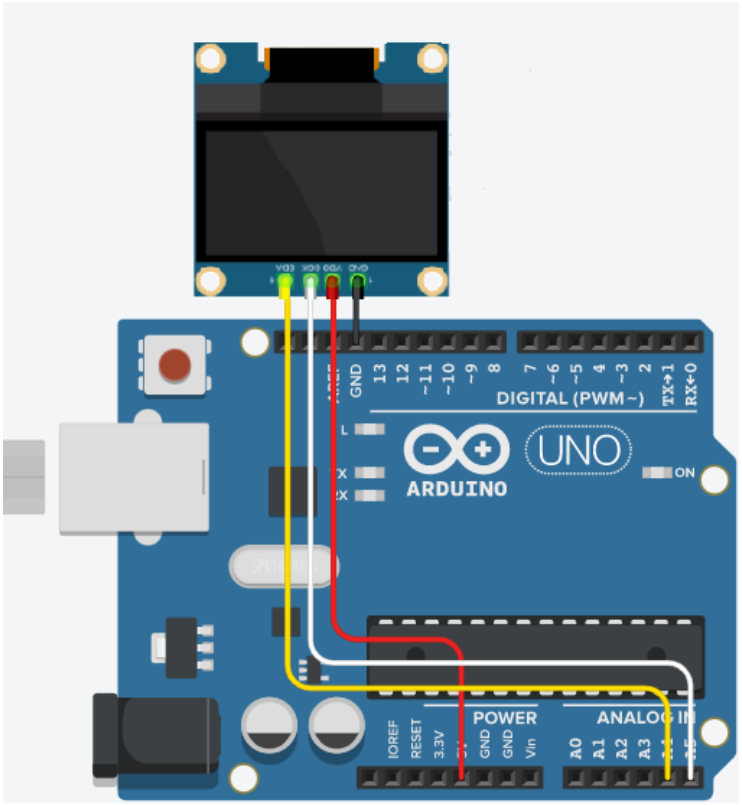
Tablica 12. LCD zaslon

Linija ***Kod***

```
1:      #include<LiquidCrystal.h>
2:      LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
3:      void setup(){
4:      lcd.begin(16, 2);
5:      }
6:      void loop(){
7:      lcd.setCursor(0,0);
8:      lcd.print("FERIT!");
9:      lcd.setCursor(2,1);
10:     lcd.print("Arduino");
11:     }
```

3.13. OLED zaslon

OLED zasloni su organske LE diode, a naziv dolazi od principa rada zaslona. Svijetlost emitiraju poluvodiči organske prirode koji su otkriveni 1970-ih godina. Sastoje se od katode, emiterskog sloja, provodnog sloja i anode. Protjecanjem struje organski (emiterski) sloj proizvodi svijetlost. Prednost im je što su tanji i lakši od LCD zaslona, a također emitiraju jaču količinu svijetla koristeći slabiju struju. Imaju veliki kut vidljivosti do 170 stupnjeva te se proizvode u različitim dimenzijama. Komunikacija između OLED zaslona i Arduino pločice se odvija putem I2C sabirnice, koristeći SDA i SCL pinove zaslona i pinova za serijsku komunikaciju Arduino pločice. Dodatna prednost je što se ovim tipom komunikacije mogu slati podaci na više različitih uređaja, a bitno je samo da svaki uređaj ima svoju adresu zapisanu u heksadecimalnom zapisu. Adresa OLED zaslona je 0xC3. Također, za upravljanje OLED zaslonom potrebno je koristiti Adafruit_SSD1306 i Adafruit_GFX biblioteke. Uvođenjem tih biblioteka moguće je koristiti naredbe poput `display.clearDispay();` za gašenje svih piksela zaslona, `display.setTextSize();` za odabir veličine fonta (od 1 do 8), `display.setCursor(x,y);` za postavljanje koordinata početka teksta, `display.print();` za ispis teksta te `display.display();` za osvježavanja zaslona novim podacima koji se unose navedenim funkcijama. Dijagram spajanja OLED zaslona je prikazan na slici 13. OLED zaslon.



Sl. 13. OLED zaslon

Nakon definiranja potrebnih biblioteka u programskom kodu za OLED zaslon potrebno je stvoriti objekt zaslona pozivajući naredbu `Adafruit_SSD1306 display(128, 64);` koja govori da se radi o zaslonu duljine 128 piksela i širine 64 piksela. Zatim je potrebno inicijalizirati zaslon naredbom `display.begin(SSD1306_SWITCHCAPVCC, 0x3C);` kojoj je u parametrima naveden stvoreni objekt i heksadecimalna adresa zaslona. Pomoću ranije opisanih funkcija moguće je na zaslonu ispisati bilo kakav tekst ili čak stvarati različite oblike. Detaljniji prikaz koda programiranja OLED zaslona, gdje se na zaslonu ispisuje riječ „FERIT“ je prikazan u tablici 13. OLED zaslon.

Tablica 13. OLED zaslon

Linija ***Kod***

```
1:      #include <Wire.h>
2:      #include <Adafruit_SSD1306.h>
3:      #include <Adafruit_GFX.h>
4:      Adafruit_SSD1306 display(128, 64);
5:      void setup() {
6:      display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
7:      }
8:      void loop() {
9:      display.clearDisplay();
10:     display.setTextSize(2);
11:     display.setTextColor(WHITE);
12:     display.setCursor(0, 0);
13:     display.println("FERIT");
14:     display.display();
15:     delay(5000);
16:     }
```

4. ZAKLJUČAK

Cilj ovog rada je bio prikazati spajanje i programski kod Arduino komponenti na jednostavan i razumljiv način. To je postignuto jednostavnim prikazom spajanja komponenti i kratkim programskim kodovima dostatnim za ispravan rad komponenti. Glavna prednost rada je upravo u tome što nisu prikazani specifični moduli pojedinih proizvođača, nego standardne komponente kako bi svatko na osnovu rada mogao samostalno napraviti pojedinačne projekte. U prilogu se nalaze dodatni dijagrami spajanja komponenti za tri kompleksnija projekta što je odličan način za testiranje znanja u samostalnoj primjeni. U radu se prvo opisuju najjednostavnije i najčešće korištene komponente, a zatim se dolazi do nešto složenijih u smislu načina rada i broja žica koje je potrebno spajati. Smatram da je rad ostvario postavljeni cilj iz razloga što je programski kod sažet i razumljiv te ostavlja dosta prostora za nadogradnju. Osim što je opisan način spajanja komponenti dodatno se nalaze dijagrami spajanja iz kojih je moguće vidjeti kako spoj komponente i Arduino razvojne pločice treba izgledati.

LITERATURA

1. "Hello World". Dostupno na: https://www.arduino.cc/en/Tutorial/LibraryExamples/HelloWorld. (2021-08-10)
2. Blink. Dostupno na: https://www.arduino.cc/en/Tutorial/BuiltInExamples/Blink. (2021-08-10)
3. ElectroPeak. Complete Guide to Use Soil Moisture Sensor w/Examples, 2019. Dostuno na: https://create.arduino.cc/projecthub/electropeak/complete-guide-to-use-soil-moisture-sensor-w-examples-756b1f. (2021-08-17)
4. ElectroPeak. PIR Motion Sensor: How to Use PIRs w/Arduino & Raspberry Pi, 2019. Dostupno na: https://create.arduino.cc/projecthub/electropeak/pir-motion-sensor-how-to-use-pirs-w-arduino-raspberry-pi-18d7fa. (2021-08-19)
5. Guide for I2C OLED Display with Arduino. Dostupno na: https://randomnerdtutorials.com/guide-for-oled-display-with-arduino/. (2021-09-05)
6. How to use Servo Motors with Arduino, 2018. Dostupno na: https://create.arduino.cc/projecthub/arcaegecengiz/how-to-use-servo-motors-with-arduino-b02bed. (2021-08-17)
7. KS0085 Keyestudio Smart Home Kit for Arduino. Dostupno na: https://wiki.keyestudio.com/KS0085_Keyestudio_Smart_Home_Kit_for_Arduino. (2021-08-09)
8. Mukherjee, Aritro. Smoke Detection using MQ-2 Gas Sensor, 2016. Dostupno na https://create.arduino.cc/projecthub/Aritro/smoke-detection-using-mq-2-gas-sensor-79c54a. (2021-08-19)
9. Photocells: Arduino Code, 2021. Dostupno na: https://learn.adafruit.com/photocells/arduino-code. (2021-08-17)
10. Pushbutton. Dostupno na: https://www.arduino.cc/en/tutorial/pushbutton. (2021-08-10)
11. Steam Sensor: Introduction. Dostupno na: https://wiki.dfrobot.com/Steam_Sensor__SKU_SEN0121_ . (2021-08-18)
12. Suryateja, Sai. Use a Buzzer Module (Piezo Speaker) using Arduino Uno, 2018. Dostupno na: https://create.arduino.cc/projecthub/SURYATEJA/use-a-buzzer-module-piezo-speaker-using-arduino-uno-89df45. (2021-08-18)
13. Using A TMP36 Temperature Sensor With Arduino. Dostupno na: https://bc-robotics.com/tutorials/using-a-tmp36-temperature-sensor-with-arduino/. (2021-08-18)

SAŽETAK

Ovaj rad se sastoji od 13 projekata koji opisuju korištenje različitih komponenti uz Arduino razvojnu pločicu. Svaki projekt prvo ukratko opisuje komponentu, zatim je opisan i prikazan dijagram spajanja komponente sa Arduino razvojnom pločicom te naposljetku je objašnjen i napisan programski kod potreban za rad komponente. Na početku rada je pojašnjeno upravljanje jedne LE diode, zatim je na nju dodatno tipkalo. Nakon LE dioda rad se osvrće na DC motore, a to su servo motor i ventilator. Poslije motora je opisan pasivni buzzer kao jednostavni zvučnik nakon čega slijedi opis raznih senzora. Opisan je senzor jačine svijetlosti, vodene pare, pokreta, detekcije plina i temperature. Kraj rada opisuje dva najčešće korištena zaslona uz Arduino razvojnu pločicu. To su LCD zaslon i OLED zaslon.

Ključne riječi: Arduino, LE dioda, DC motori, pasivni buzzer, senzori, zasloni

ABSTRACT

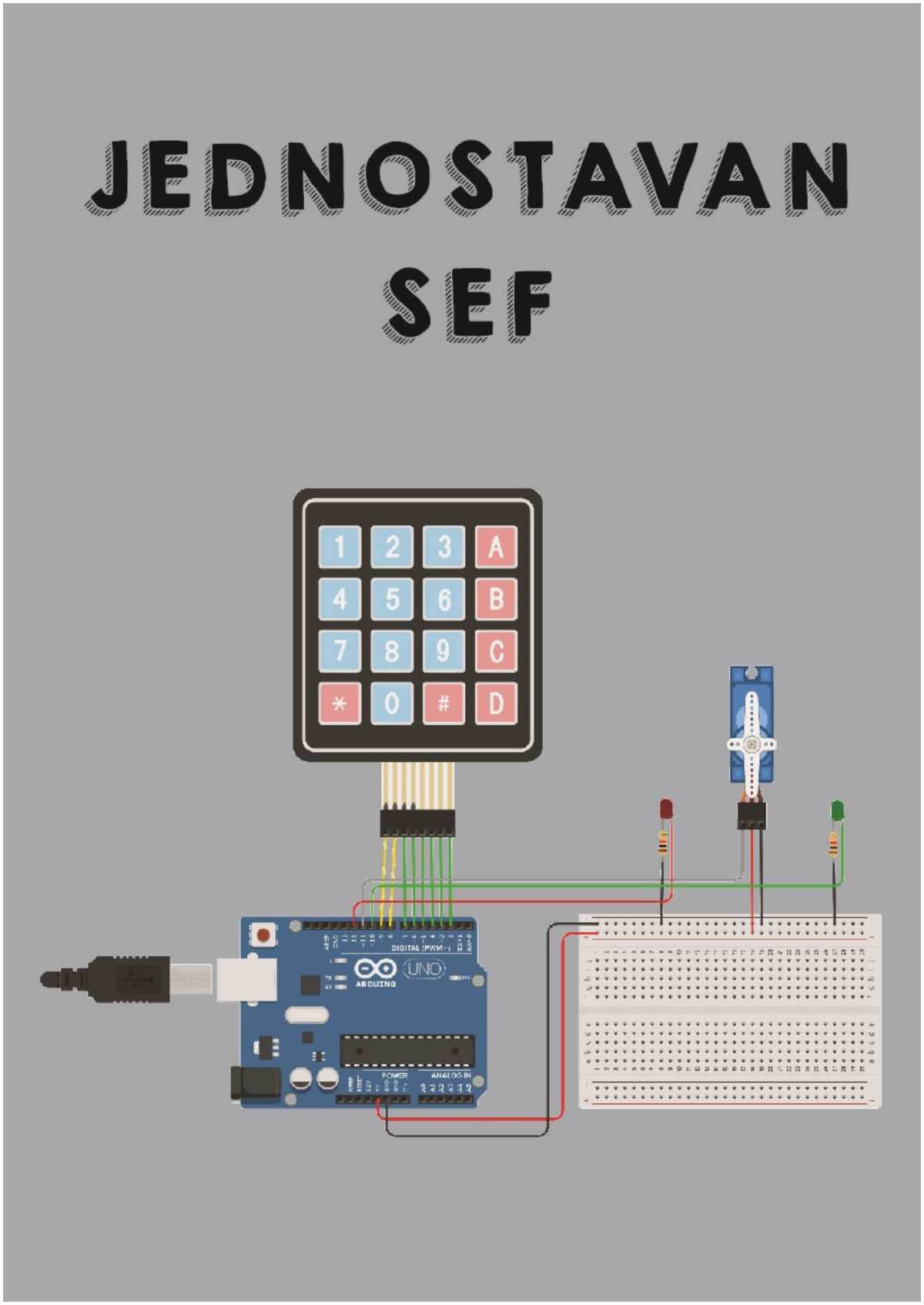
This paper consists of 13 projects which describe the use of different components along with the Arduino development board. In each project are firstly described the components. Secondly, a diagram of the connection of components with Arduino development board is described and displayed. Finally, the setup and the program code required for the components to work is explained and written down. At the beginning of the paper, the control of one LE diode is clarified, and later it was added a pushbutton. After the LE diode, this paper discusses the DC motors, which are the servo motor and the fan. After the motors, the passive buzzer is described as a simple speaker followed by a description of the various sensors which are the sensor of light intensity, water vapor, motion, gas detection and temperature. The end of the paper discusses the two most commonly used screens along with the Arduino development board. Those are the LCD screen and the OLED screen.

Keywords: Arduino, LE diode, DC motors, passive buzzer, sensors, screens

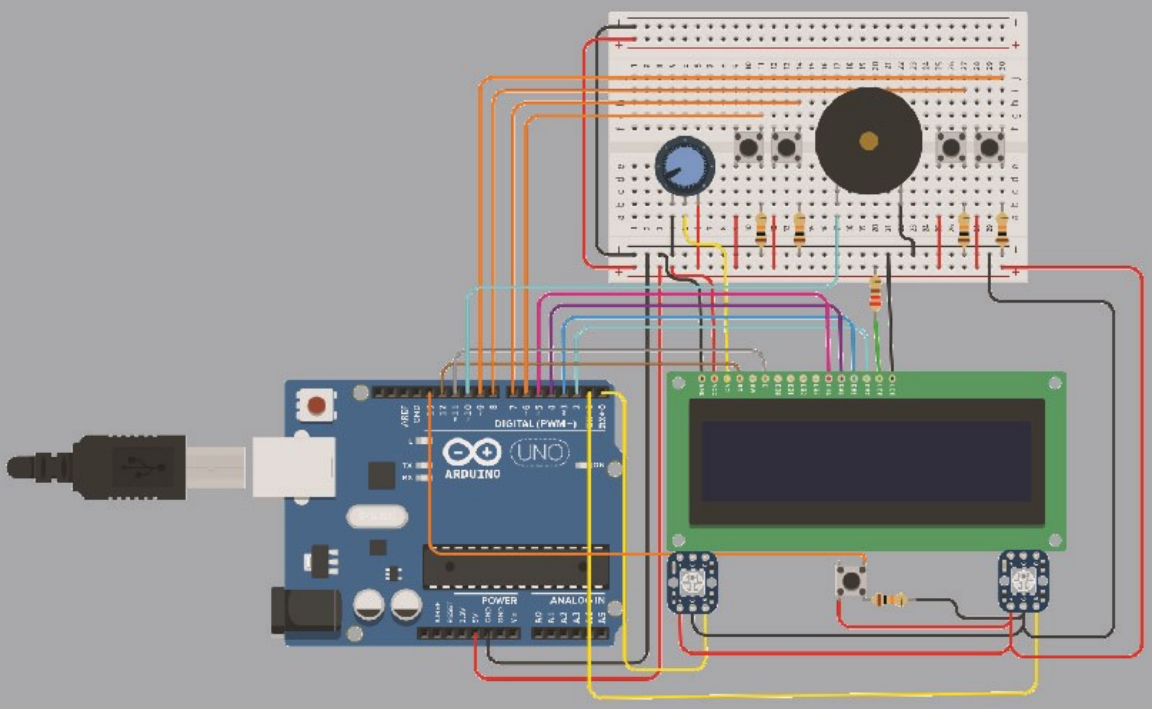
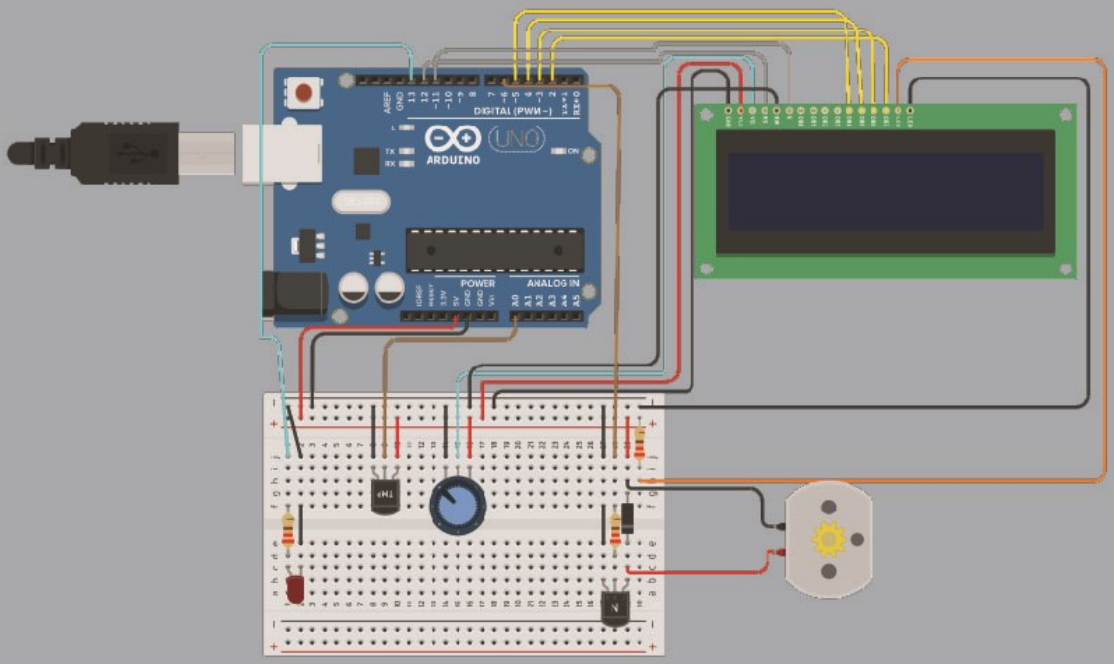
ŽIVOTOPIS

Mario Zrnić rođen je 18.11.1994. u Varaždinu. Pohađao je Prvu osnovnu školu Brčko distrikt u Brčkom, nakon toga upisuje opću gimnaziju Vaso Pelagić u Brčkom. Po završetku srednje škole prvotno upisuje studij Fizike u Osijeku, ali se nakon godinu dana ispisuje te počinje raditi u Hrvatskom Telekomu. Nakon tri godine rada u telekomu 2018. godine upisuje preddiplomski studij elektrotehnike izborni blok Komunikacije i informatika na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija u Osijeku.

PRILOZI



PING PONG IGRICA



AUTOMATSKI REGULATOR TEMPERATURE