

UNIVERZITET U TUZLI  
FAKULTET ELEKTROTEHNIKE TUZLA  
TELEKOMUNIKACIJE  
SATELITSKE TELEKOMUNIKACIJE



*PROJEKAT*

***Obavještenje o detekciji dima***

**Profesor:**

dr. sci. Alma Šećerbegović, docent

**Studenti:**

Adnan Ljeskovic  
Armin Korajlić

Tuzla, Jun 2023

# Uvod

Ovaj projekat koristi Arduino mikrokontroler, senzor za detekciju dima, buzzer za obavještenje i dvije diode kako bi vas upozorio na prisustvo dima cigarete.

Glavni cilj ovog projekta je da pruži pouzdanu detekciju dima cigarete i obavijesti vas o potencijalnoj opasnosti slanjem upozorenja na e-poštu. Kroz jednostavnu i efikasnu implementaciju, sistem koristi senzor za detekciju dima koji kontinuirano nadgleda okolinu.

Kada senzor detektuje dim, Arduino mikrokontroler reaguje aktivirajući crvenu diodu kako bi vizuelno signalizirao prisustvo dima. Arduino ne samo da aktivira crvenu diodu već šalje i e-poštu na unaprijed definisanu e-mail adresu. E-pošta sadrži obavještenje o aktivaciji senzora što vam omogućava da odmah budete informisani o potencijalnoj opasnosti, čak i ako niste u neposrednoj blizini. Osim toga, buzzer se aktivira kako bi pružio dodatno audio upozorenje. U slučaju kada senzor nije aktiviran, zelena diode će biti uključena, ukazujući na sigurnu situaciju bez prisustva dima.

Ovaj projekat ima široku primjenu i može biti koristan u različitim okruženjima kao što su domovi, uredi ili javne institucije. S jednostavnom montažom i lako razumljivim obavještenjem, ovaj sistem pruža jednostavan način da budete informisani o potencijalnoj opasnosti od dima cigarete.

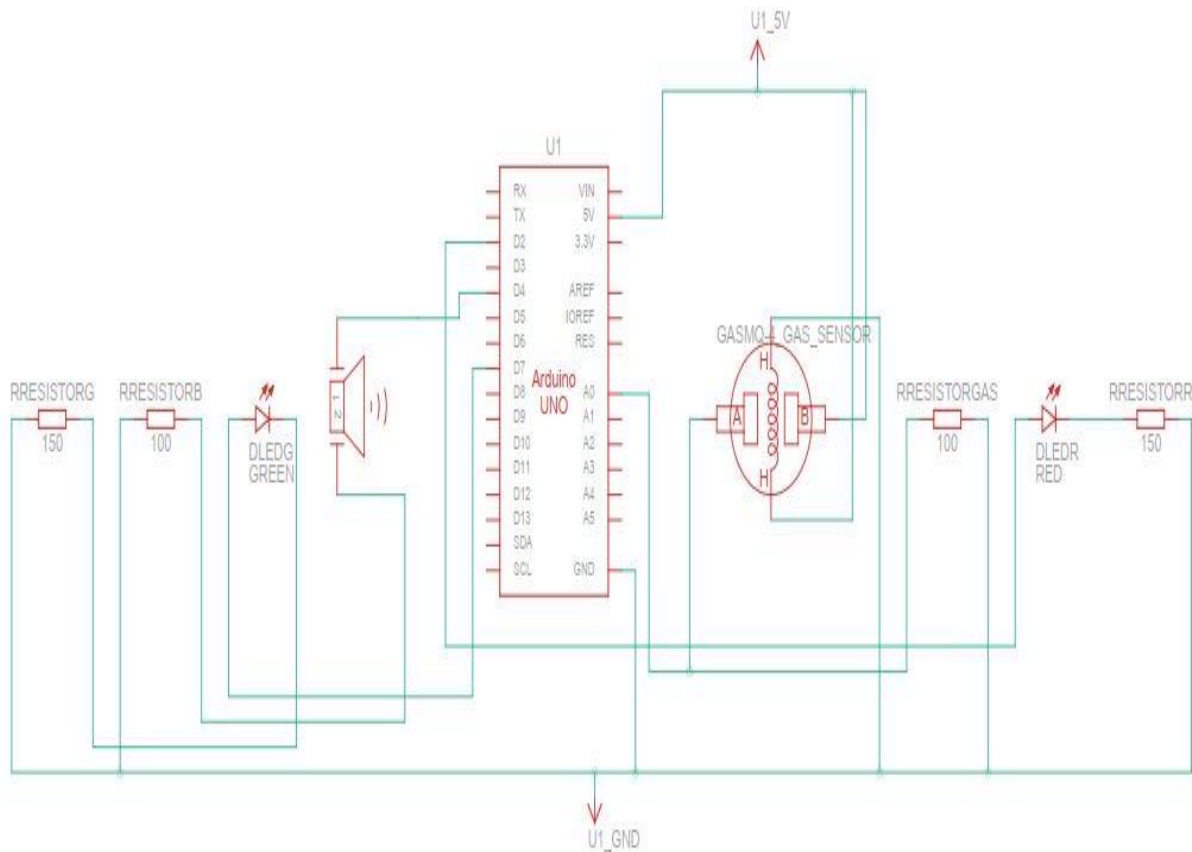
# Realizacija projekta

Za realizaciju projekta potrebno je razmotriti nekoliko elemenata kao što su: komponente, dijagram kola, sami kod i testiranje. U nastavku ćemo vidjeti koje smo komponente koristili pri realizaciji projekta, kako izgleda samo kolo te njegova šema, i na kraju sami kod koji smo koristili da bi sve funkcionisalo kako smo zamislili. Na kraju ćemo pokazati „Live“ demonstraciju.

## Komponente

- Arduino UNO -> ploča mikrokontrolera zasnovana na ATmega328P
- MQ2 Gas Senzor -> Koji detektuje nivo dima
- Buzzer -> Uređaj koji proizvodi zvuk upozorenja
- 5V LED -> Diode za emitovanje svjetlosti
- Otpornici
- Prespojni provodnici -> Za povezivanje komponenata u dijagramu kola

## Dijagram kola



*Dijagram kola*

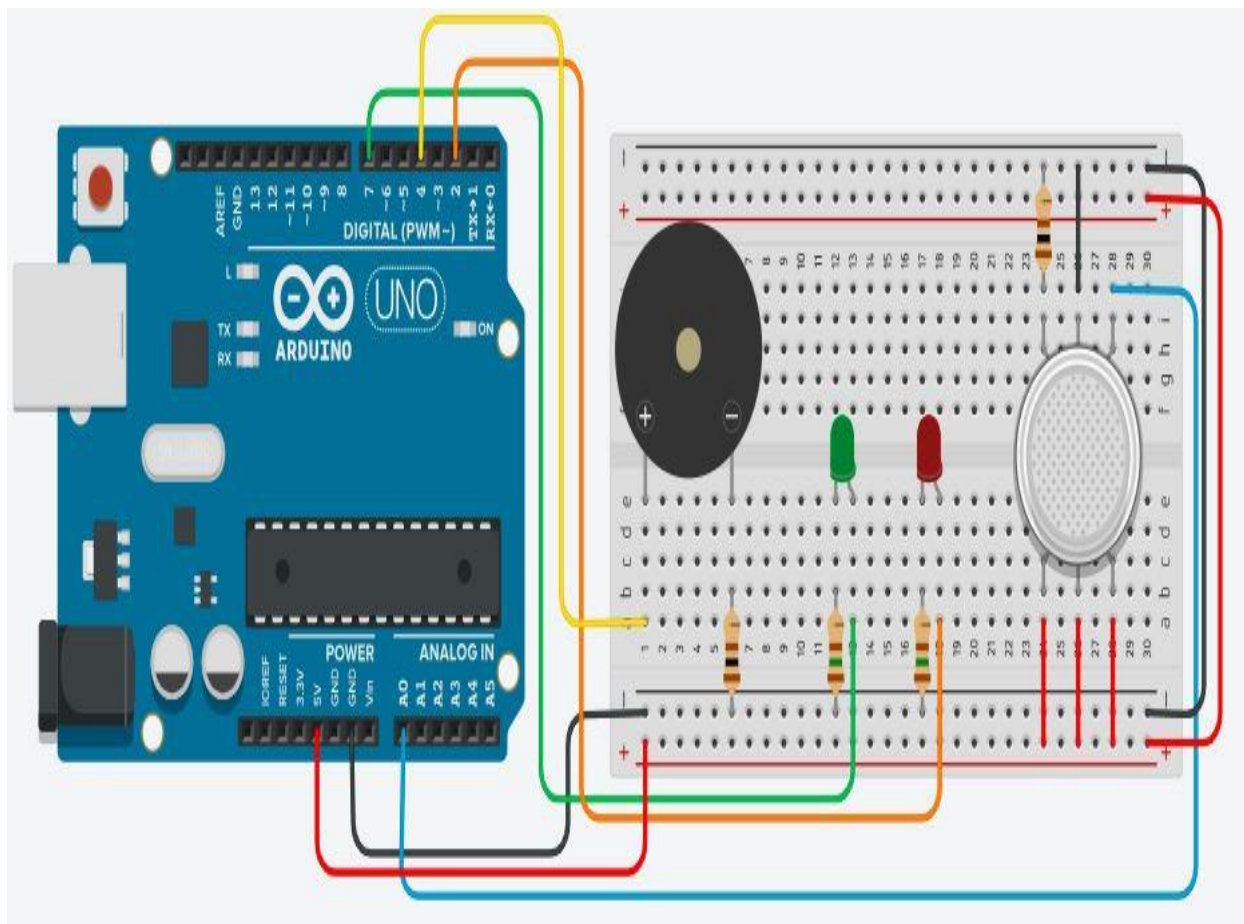
Dijagram kola za projekat "Obavještenje o detekciji dima" prikazuje kako su komponente povezane i kako informacije teku unutar sistema. Ovaj dijagram pruža jasan pregled rasporeda i veza između komponenata projekta. Arduino mikrokontroler je povezan s dvije diode - crvenom diodom i zelenom diodom.

Crvena dioda se aktivira kada senzor za detekciju dima registruje dim cigarete, pružajući vizuelno upozorenje.

Zelena dioda svijetli kada senzor nije aktiviran, signalizirajući sigurno okruženje bez dima. Osim toga, Arduino mikrokontroler je povezan s buzzerom koji generira zvučno upozorenje kada se senzor aktivira.

Buzzer omogućava dodatnu dimenziju upozorenja kako bi korisnici bili obaviješteni o prisustvu dima.

Na kraju, Arduino preko Python skripte kada se senzor za detekciju dima aktivira šalje obavještenje putem e-pošte na unaprijed definiranu adresu, pružajući korisnicima trenutnu informaciju o detekciji dima.



*Šema kola*

Dok šema prkazuje kako su komponente povezane i raspoređene na fizičkoj razini.

## Opis koda

```
8 import smtplib
9 import datetime
10 import serial
11
12 # Email
13 smtp_server = 'smtp.gmail.com'
14 smtp_port = 587
15 smtp_username = 'Ludabudala.budalaaa@gmail.com'
16 smtp_password = 'kdygnthsgcnkklji'
17 email_sender = 'Ludabudala.budalaaa@gmail.com'
18 email_recipient = 'armin.korajlic@fet.ba'
19 email_subject = 'Gas Alert'
20 email_message = 'Senzor se aktivirao, prisustvo gasa detektovano !'
21
22 #import serial - bio je ovdje
23 ser = serial.Serial('COM3', 9600) #Port com3 na windows.
24 email_counter = 0 #Counter za brojanje koliko mailova saljemo
25
26 while email_counter < 1: #Vrti beskonacno ovu petlju dok se ne posalje mail.
27     line = ser.readline().strip().decode('utf-8')
28     if line == 'SendEmail':
29         time_stamp = datetime.datetime.now().strftime("%Y-%m-%d %H:%M:%S")
30         email_message = f'Senzor se aktivirao, prisustvo gasa detektovano !\n\nVrijeme detekcije: {time_stamp}'
31
32     # Send
33     try:
34         server = smtplib.SMTP(smtp_server, smtp_port)
35         server.starttls()
36         server.login(smtp_username, smtp_password)
37         message = f"Subject: {email_subject}\n\n{email_message}"
38         server.sendmail(email_sender, email_recipient, message)
39         server.quit()
40         print(f"Email je poslan uspjesno!\nVrijeme detekcije:{time_stamp}")
41         email_counter += 1
42     except Exception as e:
43         print(f"Failed to send email: {e}")
44     ser.close() # close serial
45     #Moze se jednostavno napraviti i da vrti petlju beskonacno, sa malom izmjenom
46     #koda, ali zbog jednostavnosti nastimao sam da salje samo jednom.
47     #Ako hocu da salje vise puta samo cu prvu petlju zavrtiti na while 0, da
48     #vrti beskonacno, i onda unutar samo if uslov, poslije
49     #if line == 'sendemail', koji ce glasiti if emailcounter < 1 i da se onda
50     #odradi ovo ispod try:
51     #Te treba line postaviti na neku trash vrijednost, da ne aktivira se opet,
52     #ili imati counter od 10s, te ako i dalje gori detektor da opet salje mail
53     #i tako sve dok ne ugasi se.
```

### Python kod

Prilikom izrade ovog projekta, glavni problem je bio razumjeti kako funkcioniše biblioteka 'smtplib' u Python programskom jeziku. Drugi problem je bio kako spojiti senzor, buzzer sa ostalim dijelovima koji uključuju ovaj projekat, nismo bili sigurni sto posto kako i gdje šta ide, te smo tu tražili pomoć asistenta, jer komponente nisu naše, te je postojala mogućnost da uništimo jedan od elemenata, što bi koštalo. Prilikom razumijevanja kako funkcioniše biblioteka, morali smo naći na internetu primjere za istu, te razumijeti sintaksu.

Vidimo u kodu da smo koristili varijable da bi dodijelili potrebne podatke funkcijama kao sto su SMTP(), .starttls(), .login(), .sendmail(), .quit(). Pristupili smo Google-ovom serveru preko smtp.gmail.com, te preko standardnog porta '587', ovo je za Windows!

“smtp\_username” nam predstavlja naš mail sa kojeg šaljemo podatke koje primimo preko serijskog porta ‘COM3’, ovaj port je nešto slično što smo koristili na predmetu ‘Mikroprocesorski sistemi u Telekomunikacijama’, samo što smo tamo koristili drugo ime porta, jer smo koristili Linux OS.

“smtp\_password” je šifra koja je posebno generirana od strane Google-a da bi pristupili na ovaj način emailu, prije je moglo i sa običnom šifrom, ali se to promijenilo te smo morali na ovaj način.

“email\_recipient” je email na koji šaljemo, te je to moguće promijeniti.

“email\_subject” je naziv maila koji šaljemo.

“email\_message” je poruka koju želimo poslati, ali smo je preimenovali dalje u kodu, jer željeli smo poslati trenutno vrijeme kada se desila detekcija dima. Kada port registruje da je poslat signal preko Arduino mikrokontrolera, Python skripta to detektuje te se aktivira dio koda ‘line == ‘SendEmail’’.

Nakon toga učitavamo trenutno vrijeme te to stavljamo u poruku. Nakon toga pravimo varijablu server, te startamo TLS. Pristupamo sa username, i šifrom serveru te šaljemo mail sa potrebnim podacima, i nakon toga izlazimo sa servera.

Postavili sam da se email šalje samo jednom, jer inače bi slalo mail u petlji, možemo napraviti da se mail šalje svakih 10s npr, te da se ugasi kada senzor se ugasi.

Prije svega ovoga potrebna je detekcija dima. Kada se dim detektuje na senzoru, šalje se signal preko TX linije na port laptopa, te ovo očitava Python skripta, koja ‘odsluškuje’ taj port. Nakon ovoga se aktivira prethodno.

Naš kod je bio Python kod, te dio Arduino koda. Nasli smo dio na internetu, ali ga nismo mogli koristiti jer se preklapala šema sa TX i RX pinovima. Ovo je problem jer koristimo te pinove da bi slali/primili signal sa porta, te kada bi se desila detekcija dima, nije bilo moguće ugasiti detektor, tj buzzer, jer bi TX slao signal, i dolazilo je do problema i preklapanja signala.

Stoga smo iskoristili ostale pinove. Šemu i spajanje smo pronašli na internetu i to smo iskoristili da bi postigli što smo željeli, a to je da se pošalje email, prilikom detektovanja dima. Probali bi sami isto uraditi, ali nismo željeli rizikovati da uništimo pločicu.

Dio koda koji smo koristili, te način spajanja je bio sa sljedeće stranice:  
<https://www.sensingthecity.com/configuring-an-mq-4-sensor-to-detect-methane-gas/>

Dio koda u Arduino je vrlo jednostavan za shvatiti. Napravili smo 4 varijable kojima smo dodijelili vrijednosti. Te vrijednosti su zapravo konekcije na Arduino mikrokontroleru, a threshold je prag detekcije.

Koristili smo setup() da bi započeli transmisiju podataka na 9600 baudrate, te postavili pinove na 'INPUT' ili 'OUTPUT'. U petlji bi čitali svaku iteraciju vrijednost sa senzora. Ako bi ta vrijednost prešla 'threshold' onda bi se aktivirao alarm i poslao bi se email. U suprotnom bi čitavo kolo bilo u 'normalnom' stanju. To normalno stanje je da nema tona u buzzeru, te da zelena dioda svijetli, dok crvena ne svijetli. U našem slučaju crvena je žuta, nismo imali crvenu diodu. Kada bi se aktivirao senzor pokrenula bi se funkcija alarmState() i aktivirala buzzer, te isključila zelenu diodu i upalila crvenu. Te aktivirala bi se i funkcija triggerEmail() koja bi na port emitovala poruku "SendEmail", a upravo tu poruku čeka Python kod, koji se onda aktivira i šalje mail na željenu adresu.



```

const int gasSensorPin = A0;
const int buzzerPin = 4;
const int ledGreenPin = 7;
const int ledRedPin = 2;
const int threshold = 650;

void setup() {
  Serial.begin(9600);
  pinMode(gasSensorPin, INPUT);
  pinMode(buzzerPin, OUTPUT);
  pinMode(ledGreenPin, OUTPUT);
  pinMode(ledRedPin, OUTPUT);
}

void loop() {
  int gas_value = analogRead(gasSensorPin);

  if (gas_value > threshold) {
    alarmState();
    triggerEmail();
  } else {
    normalState();
  }

  delay(200);
}

void alarmState() {
  tone(buzzerPin, 1000, 500);
  digitalWrite(ledRedPin, HIGH);
  digitalWrite(ledGreenPin, LOW);
}

void normalState() {
  noTone(buzzerPin);
  digitalWrite(ledGreenPin, HIGH);
  digitalWrite(ledRedPin, LOW);
}

void triggerEmail() {
  Serial.println("SendEmail");
}

```

*Arduino kod*

# Zaključak

Na kraju projekta možemo zaključiti da se Arduino Uno koristi u raznim tehnologijama zbog svoje jednostavnosti korištenja, velike dostupnosti komponenti, fleksibilnosti i pristupačnosti koje su posebno korisne za projekte detekcije dima.

Projekat pruža efikasno i pouzdano rješenje za nadzor i obavješćavanje o prisustvu dima cigarete. Njegove prednosti uključuju lokalno upozorenje putem dioda i buzzer-a, kao i slanje e-pošte za trenutno obavješćavanje. Ovaj projekat doprinosi sigurnosti i pruža korisnicima potrebne informacije za brzu reakciju u slučaju detekcije dima.

Tokom realizacije projekta kolege Armin Korajlić i Adnan Ljeskovica nisu imali jasno definisane raspodjele zaduženja zbog nedostatka prethodnog iskustva sa ovakvim projektima, članovi su bili uključeni u sve faze projekta. Pristup projektu je bio u duhu timskog rada i obojica smo bili angažovani u istraživanju i razumijevanju potrebnih komponenti, kao i njihovih međusobnih veza. Zajedno smo istraživali i proučavali Arduino platformu kako bi stekli potrebno znanje za implementaciju projekta. Svako je doprinio svojim idejama i sugestijama, a zajedno smo pravili plan povezivanja komponenata pri čemu je veći doprinos imao kolega Adnan Ljeskovica kako bi se postigla željena funkcionalnost projekta. Kada je došlo do pisanja koda, tim je također surađivao ali uz veći doprinos kolege Armina Korajlića, jer svaki član je dao svoj doprinos u skladu sa svojim znanjem i sposobnostima, stvarajući timski radni okvir u kojem su svi radili zajedno ka uspješnoj realizaciji projekta.