

# Upotreba mikrokontrolera u nastavi fizike

---

Gašparić, Mirko

Master's thesis / Diplomski rad

2023

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Science / Sveučilište u Zagrebu, Prirodoslovno-matematički fakultet**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:217:915647>

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

Download date / Datum preuzimanja: **2024-08-27**



Repository / Repozitorij:

[Repository of the Faculty of Science - University of Zagreb](#)



SVEUČILIŠTE U ZAGREBU  
PRIRODOSLOVNO-MATEMATIČKI FAKULTET  
FIZIČKI ODSJEK

Mirko Gašparić

UPOTREBA MIKROKONTROLERA U  
NASTAVI FIZIKE

Diplomski rad

Zagreb, 2023.

SVEUČILIŠTE U ZAGREBU  
PRIRODOSLOVNO-MATEMATIČKI FAKULTET  
FIZIČKI ODSJEK

INTEGRIRANI PREDDIPLOMSKI I DIPLOMSKI SVEUČILIŠNI STUDIJ  
FIZIKA; SMJER: NASTAVNIČKI

**Mirko Gašparić**

Diplomski rad

**Upotreba mikrokontrolera u nastavi fizike**

Voditelj diplomskog rada: doc. dr. sc. Dalibor Paar

Ocjena diplomskog rada: \_\_\_\_\_

Povjerenstvo: 1. \_\_\_\_\_

2. \_\_\_\_\_

3. \_\_\_\_\_

Datum polaganja: \_\_\_\_\_

Zagreb, 2023.

## Sažetak

Suvremena nastava fizike nalazi se pred novim izazovima. Fizika je među ključnim STEM područjima koja postaju sve važnija u kontekstu suvremenih zanimanja i potreba u okviru svakodnevnog života. U okvirima postojećeg kurikuluma, postavlja se pitanje o metodama kojima bi se učenike više zainteresiralo za fiziku i upoznalo ih s različitim aspektima znanstvene metode i puno puta provelo kroz istraživački postupak. Fizikalni pokus je nešto što nije moguće zamijeniti digitalnim sadržajem jer daje stvarni, interaktivni i istraživački uvid u prirodne pojave i procese. Temeljna uloga mikroračunala u nastavi fizike je konstrukcija mjernog instrumenta. Brojni dostupni vanjski senzori otvaraju mogućnosti konstrukcije mjernih instrumenata. Poanta nije da učenik dobije složen mjerni instrument već da ga samostalno konstruira spajanjem potrebnih komponenti na mikroračunalo i doradom programa za mjerenje. Na primjeru Arduino platforme pokazat ćemo da je ovakav pristup moguće relativno brzo uvesti u nastavni proces.

Ključne riječi: suvremena nastava fizike, fizikalni pokus, mikrokontroleri, Arduino

# **THE USE OF MICROCONTROLLERS IN THE TEACHING OF PHYSICS**

## **Abstract**

Modern physics teaching is facing new challenges. Physics is among the key STEM fields that are becoming increasingly important in the context of modern occupations and needs in everyday life. Within the framework of the existing curriculum, the question arises about the methods that would make students more interested in physics and introduce them to different aspects of the scientific method and lead them many times through the research process. A physical experiment is something that cannot be replaced by digital content because it provides a real, interactive and exploratory insight into natural phenomena and processes. The fundamental role of the microcomputer in teaching physics is the construction of a measuring instrument. Numerous available external sensors open possibilities for the construction of measuring instruments. The point is not for the student to receive a complex measuring instrument, but to construct it independently by connecting the necessary components to a microcomputer and refining the measurement program. Using the example of the Arduino platform, we will show that this approach can be introduced quickly into the teaching process.

Keywords: modern physics teaching, physics experiment, microcontrollers, Arduino

# Sadržaj

1	Uvod.....	1
2	Upotreba Arduina u nastavi fizike. ....	3
2.1	Arduino mikrokontroler.....	3
2.2	Vanjski senzori .....	8
	Senzor temperature LM35 .....	8
	Senzor jakosti zvuka s mikrofonom KS0035 .....	9
	Senzor svjetlosne jakosti TEMT6000.....	9
2.3	Pokus: Svojstva izvora svjetlosti .....	10
2.3.1	Eksperimentalni postav .....	10
2.3.2	Uputa za nastavnika.....	19
2.3.3	Rezultati i diskusija .....	22
2.4	Pokus: Energija u okolišu: Svjetlost, toplina i zvuk.....	29
2.4.1	Eksperimentalni postav .....	29
2.4.2	Uputa za nastavnika.....	38
2.4.3	Rezultati i diskusija .....	41
2.5	Zaključak .....	44
2.6	Literatura.....	45

# 1 Uvod

Suvremena nastava fizike nalazi se pred novim izazovima. Fizika je među ključnim STEM područjima koja postaju sve važnija u kontekstu suvremenih zanimanja i potreba u okviru svakodnevnog života. Zato je važno da svi učenici savladaju temeljne fizikalne koncepte. U okvirima postojećeg kurikulumu, postavlja se pitanje o metodama kojima bi se učenike više zainteresiralo za fiziku, upoznalo ih s različitim aspektima znanstvene metode i puno puta provelo kroz istraživački postupak. Pored toga ističe se važnost međupredmetnog povezivanja i primjene znanja na probleme iz realnog života.

Danas su učenici prekomjerno izloženi digitalnim sadržajima, a novi alati poput ChatGPT-ja predstavljaju izazov obrazovanju svojim sve većim mogućnostima putem kojih učenici mogu dobiti gotova rješenja zadataka koje bi trebali samostalno rješavati. Fizikalni pokus je nešto što nije moguće zamijeniti digitalnim sadržajem jer daje stvarni, interaktivni i istraživački uvid u prirodne pojave i procese. Fizika je kvantitativna znanost te nije dovoljno samo promatrati određenu pojavu već je potrebno izmjeriti, kvantitativno iskazati vrijednosti varijabli koje se mijenjaju.

Mikroračunala, su u širu primjenu u obrazovanju ušla u prošlom desetljeću modelom BBC micro:bit koji je posebno dizajniran za tu namjenu, Ubrzo je prepoznat potencijal micro:bita u nastavi fizike o čemu je izrađen niz diplomskih radova na Fizičkom odsjeku [11-17]. Temeljna uloga mikroračunala u nastavi fizike je konstrukcija mjernog instrumenta. Brojni dostupni vanjski senzori otvaraju mogućnosti konstrukcije mjernih instrumenata. Poanta nije da učenik dobije složen mjerni instrument već da ga samostalno konstruira spajanjem potrebnih komponenti na mikroračunalo i doradom programa za mjerenje.

Za složenije primjene, uključujući i profesionalne, micro:bit nije pogodan zbog skromnih hardverskih mogućnosti. Za te namjene na raspolaganju su kompleksniji mikrokontroleri. Jedan od njih koji je je pronašao širu primjenu ne samo u obrazovanju je otvorena elektronička i softverska platforma Arduino. Ideja ovog rada je da pokažemo da se unatoč složenijoj platformi Arduino može relativno brzo implementirati u nastavu fizike. Stoga je ovaj rad zamišljen i kao uputa nastavniku kako ga uključiti u nastavni proces.

Arduino je inovativni projekt započet 2005. godine na Interaction Design Institute Ivrea – dvogodišnjem diplomskom studiju u području dizajna interakcija u Ivreu, Italija. Tim talentiranih pojedinaca, uključujući Massima Banzija, Davida Cuartiellesa, Toma Igoea, Gianlucu Martina i Davida Mellisa, udružio se s ciljem stvaranja pristupačne i jednostavne elektroničke platforme.

Njihova vizija bila je stvoriti uređaj koji može komunicirati s okolinom, obrađivati ulazne signale i kontrolirati različite uređaje. Otvorenost projekta očituje se u otvorenom hardveru i softveru Arduino platforme. Nacrti i upute za izgradnju Arduino pločice dostupni su svima bez naknade. Stručnjaci mogu samostalno izraditi Arduino pločicu koristeći „Bord“ i osnovne elektroničke komponente. No, ono što čini Arduino posebnim je njegova dostupnost širokom krugu ljudi, bez obzira na predznanje iz elektronike ili inženjerstva. Dio globalne Arduino zajednice može se postati putem Arduino Starter Kita, koji se može nabaviti po pristupačnoj cijeni. Uz bogatu kolekciju videa za edukaciju na Arduino stranici te platformama kao što je Youtube, primjera i projekata koji su već razvijeni i dijeljeni od strane drugih korisnika, svatko može naučiti i stvoriti svoje vlastite inovativne projekte [8].



## 2 Upotreba Arduina u nastavi fizike

Praktičan rad ima izuzetno važnu ulogu u učenju fizike jer razvija znanstvene sposobnosti učenika, omogućuje razumijevanje procesa znanstvenog istraživanja i povezuje stvarni fizički svijet s teorijom. Izazov u nastavi fizike je kako omogućiti učenicima da učinkovito i sigurno dožive laboratorijski rad. Postoje različite alternative koje se mogu primijeniti kako bi se prevladao taj izazov, uključujući korištenje računalnih simulacija i prijenosnih eksperimentalnih setova. Računalne simulacije mogu uključiti učenike u znanstvene procese kao što su manipulacija varijablama, promatranje fenomena i analiza odnosa između varijabli. Međutim, one ne mogu pružiti autentično iskustvo laboratorijskog rada. Učenici mogu fizički raditi s predmetima, vježbati sigurnu uporabu instrumenata, biti svjesni mjernih nesigurnosti, promatrati stvarne fenomene i stvarati konceptualne modele fizičkog svijeta [6].

U ovom radu ćemo pokazati kako Arduino pretvoriti u mjerni instrument (*data logger*). Za to ćemo koristiti vanjske senzore i dodatak za bilježenje podataka. Na raspolaganju je veći broj senzora već prilagođenih elektroničkim svojstvima za Arduino platformu. U ovom radu smo odabrali tri senzora: temperature, zvuka i svjetlosne jakosti. Detaljno ćemo prikazati način njihovog spajanja na mikrokontroler i prikupljanja izmjerenih podataka

### 2.1 Arduino mikrokontroler

Arduino je otvorena platforma za razvoj mikrokontrolera koja se sastoji od hardverskih i softverskih komponenti. Arduino pločice sadrže mikrokontroler s nizom ulaza i izlaza koje se mogu koristiti za kontrolu drugih uređaja ili senzora. Arduino se može povezati s računalom preko USB veze kako bi ga se programiralo i komuniciralo sa softverom na računalu.



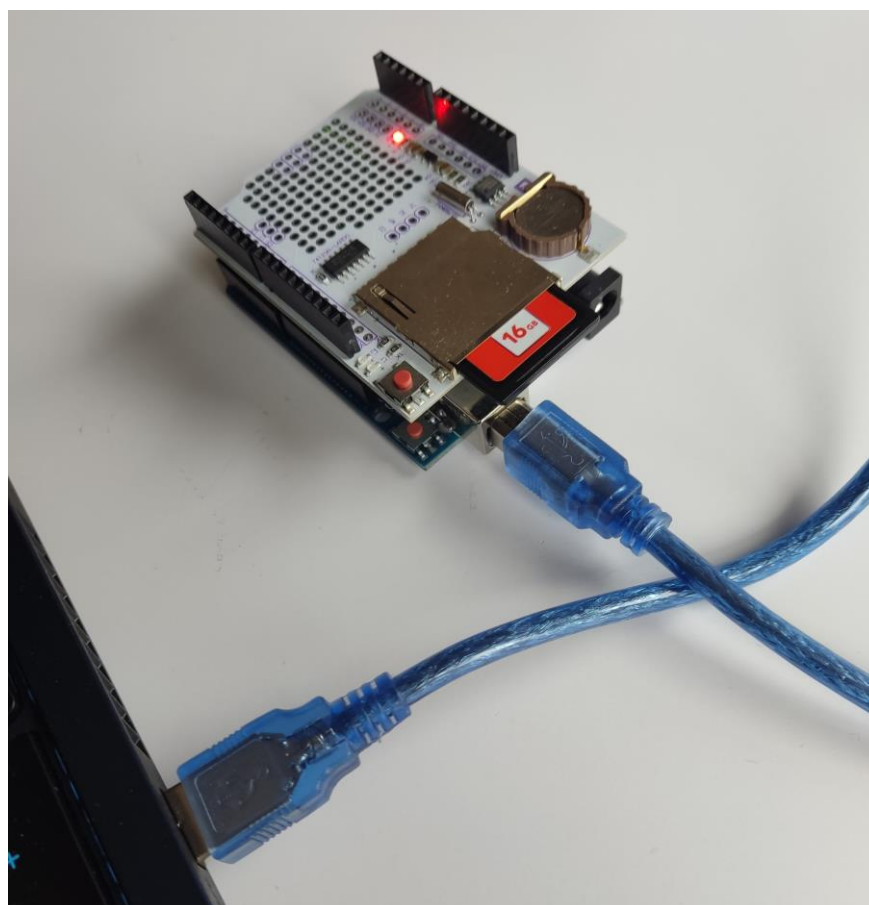
**Slika 1** Arduino mikrokontroler

Arduino softverska platforma na osobnom računalu sastoji se od Arduino IDE (Integrated Development Environment) softvera koji se koristi za pisanje i prevođenje koda, kao i za prijenos koda na mikrokontroler. Arduino IDE omogućava programerima da lako razviju kontrolne aplikacije za svoje projekte.

Spajanjem senzora na Arduino želimo konstruirati mjerni instrument, izvoditi mjerenja te prikupljati podatke. Kako Arduino nema odgovarajuću mogućnost pohrane podataka, za nju smo upotrijebili dodatnu karticu (nadogradnju naziva *Data Logging Shield*), koja na sebi ima utor za SD karticu i produžetke kao nastavak na pinove za originalnu Arduino pločicu. S ovim dodatkom uz različite spojene senzore Arduino postaje tzv. *data logger* kojim se u vremenu periodički mogu prikupljati podaci.



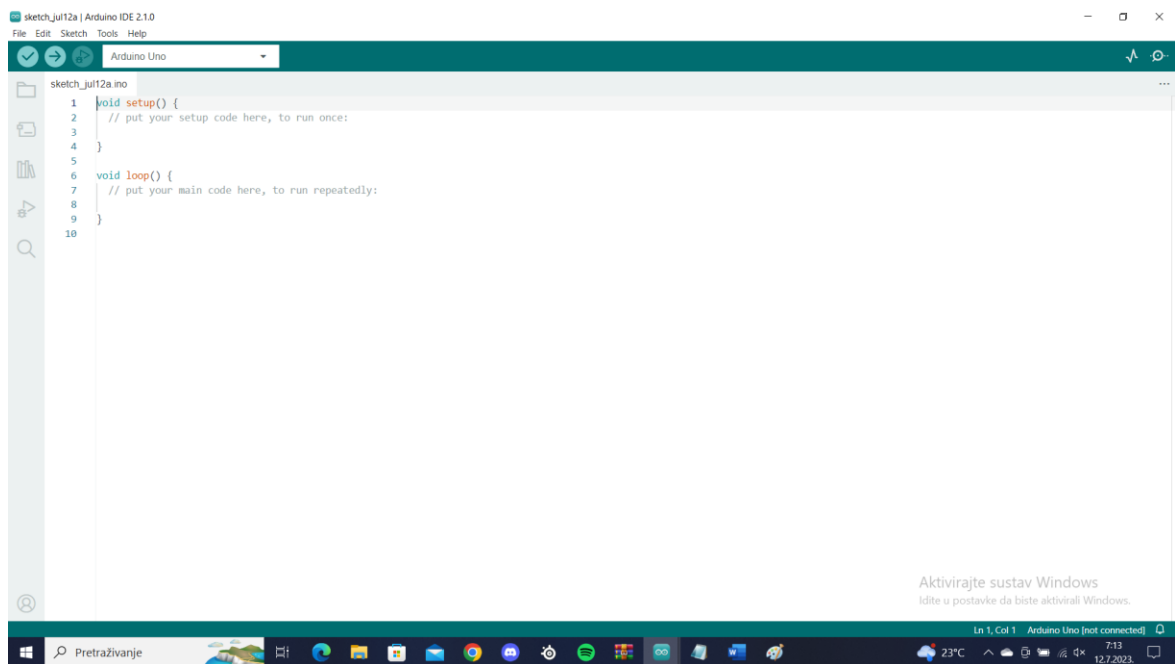
**Slika 2** SD kartica i *Data Logging Shield* za bilježenje izmjerenih podataka



**Slika 3** *Data Logging Shield* sa SD karticom spojenom na mikrokontrolerom i povezanom s računalom putem USB kabela

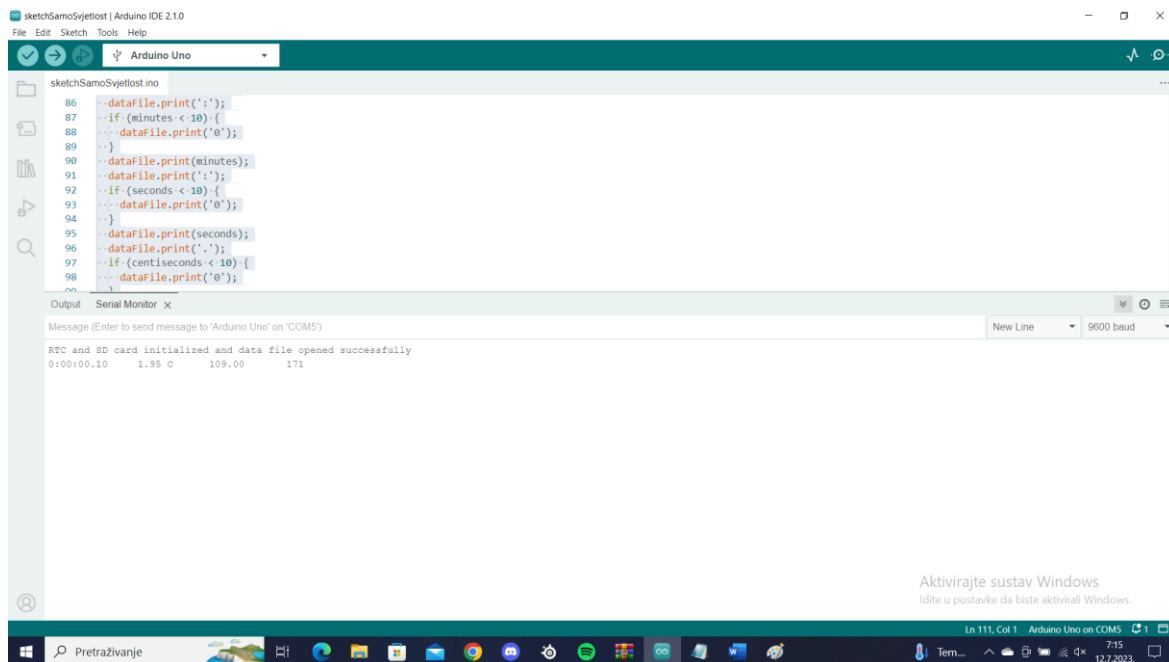
Nakon pokretanja Arduino IDE programa, on prepozna Arduino Uno pločicu, no prije prebacivanja koda dobro je provjeriti da je odabrana dobra pločica. To je vidljivo na alatnoj traci pod „Board“.

Ovisno o sklopu, potrebne će biti biblioteke za određene komponente (senzore u ovom slučaju) kako bi Arduino IDE prepoznao o čemu je riječ. Također, u gornjem lijevom vrhu se nalaze gumbi koje najviše koristimo u radu s Arduino IDE programom, a to su „Verify“ i „Upload“. „Verify“ nam služi za provjeru koda. „Upload“ nam služi za programiranje pločice nakon što smo kod provjerili naredbom „Verify“. Upload programira pločicu i započinje mjerenje u trenutku kada je program na pločici spreman.



**Slika 4** Arduino IDE sučelje prilikom prvog paljenja

Jedna od praktičnih komponenti provjere funkcionalnosti sklopa i koda je „Serial Monitor“, čiji se gumb nalazi u gornjem desnom kutu prozora Arduino IDE programa. Klikom na „Serial Monitor“, otvara se prozorčić na dnu programa gdje možemo vidjeti ispis koji nam kod omogućuje nakon pokretanja.



**Slika 5 Serial monitor pregled ispisa nakon instaliranja koda na mikrokontroler**

Arduino nudi nekoliko prednosti u odnosu na tradicionalnu nastavu kada se koristi za mjerenja. Neki od glavnih razloga uključuju [9][10]:

- Brži i precizniji rezultati: mjerni instrumenti konstruirani pomoću Arduina mogu prikupiti i obraditi podatke brže i preciznije od tradicionalnih metoda mjerenja.
- Veća fleksibilnost i prilagodljivost: Arduino omogućuje učenicima da lako mijenjaju postavke za mjerenja i senzore, što omogućuje veću fleksibilnost u pokusima.
- Povezivanje s drugim uređajima i tehnologijama: Arduino se može povezati s drugim uređajima i tehnologijama, kao što su pametni telefoni i računala, što omogućuje daljinsko praćenje i nadzor pokusa.
- Mogućnost programiranja: Programiranje Arduina omogućuje učenicima da prilagode i razvijaju programe koji će podržati njihova istraživanja i istraživačke projekte.
- Veća uključenost učenika: Korištenje Arduino tehnologije u nastavi i istraživanju može potaknuti učenike na aktivno sudjelovanje u procesu učenja, te potaknuti njihovu znatiželju i interes za STEM.

Sve ove prednosti Arduino tehnologije omogućuju učenicima da lakše i učinkovitije razumiju ključne koncepte iz fizike, informatike i drugih STEM područja te da budu uključeni u dinamična i zanimljiva mjerenja i pokuse.

## 2.2 Vanjski senzori

Za konstrukciju mjernog instrumenta za potrebe nastave fizike važna je dostupnost vanjskih senzora koji omogućuju detekciju i kvantitativno određivanje vrijednosti pojedinih fizikalnih varijabli. Dostupnost kompleta koji sadrže brojne senzore i druge elektroničke elemente prilagođene za upotrebu s mikrokontrolerom znatno olakšava osmišljavanje i konstrukciju mjernog instrumenta za potrebe nastave fizike. U učeničkom istraživačkom radu, u okviru projektne nastave ili izvannastavnih aktivnosti, učenici mogu isprobavati različite senzore koji nisu skupi te su dostupni preko elektroničkih dućana. Kao primjer u ovom radu odabrali smo tri senzora.

Senzor temperature LM35



**Slika 6** Senzor temperature LM35

Senzor temperature LM35 (kao i preostala dva senzora) dio je obrazovnog paketa „37 in 1 Starter Kit for BBC micro:bit“ proizvođača Keyestudio koji sadrži čak 37 različitih senzora i drugih elektroničkih elemenata prilagođenih za jednostavno spajanje na Arduino. Ostatak potrebne opreme (žice, postolje i mikrokontroler) je iz Arduino Allnet Starter Kit seta. Način spajanja je vrlo jednostavan budući da su pinovi na sensorima i Arduinu jasno označeni. LM35 ima 3 pina (ulaza). V ulaz spajamo s 5V na mikrokontroleru, a G na GND na mikrokontroleru. Time smo senzoru spojili napajanje, a potrebno je još spojiti podatkovnu liniju. To je pin S na senzoru kojeg ćemo spojiti na ulaz A0 na mikrokontroleru. Pinove spajamo kablom s odgovarajućim konektorom.

Kako bi očitavanje napona dalo zapis temperature, potrebno je u kodu definirati ulazne pinove, u ovom slučaju A0, te podijeliti vrijednost napona u milivoltima sa 170 kako bi dobili temperaturu u Celzijevim stupnjevima (to su postavke za ovaj specifični senzor).





**Slika 8 Senzor svjetlosne jakosti TEMT6000**

Senzor svjetlosne jakosti G i V spajamo na isti način kao kod prethodnih, a S na konektor A1. U kodu senzor očitavamo funkcijom za očitavanje jakosti zvuka `analogRead` s varijablom `LIGHTSENSORPIN`.

Senzor TEMT6000 ima široki kut polovice osjetljivosti  $\varphi = \pm 60^\circ$ . Prilagođen je za ulazni napon od 3.3 V do 5 V, te daje analogni izlaz. Raspon valnih duljina je od 440 nm do 800 nm, dok su minimum i maksimum intenziteta od 10 lux do 1000 lux.

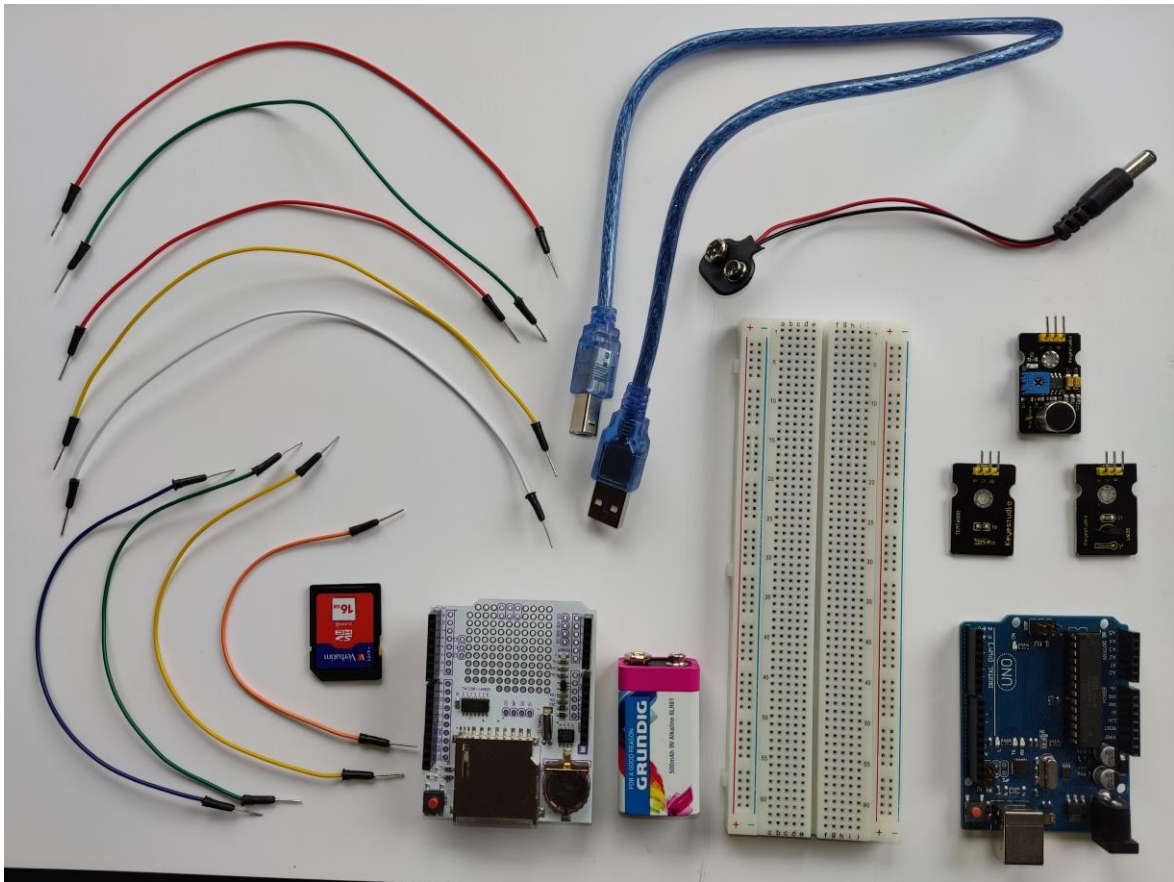
## **2.3 Pokus: Svojstva izvora svjetlosti**

### **2.3.1 Eksperimentalni postav**

Upotrebu senzora u proučavanju fizikalnih pojava demonstrirati ćemo prvo kroz pokus u kome je je zadatak napraviti mjerenje svjetlosne jakosti i temperature za ekvidistantne razmake senzora od izvora svjetlosti (svijeće). Zanima nas kako se ponašaju svjetlosna jakost i temperatura u ovisnosti o udaljenosti od izvora.

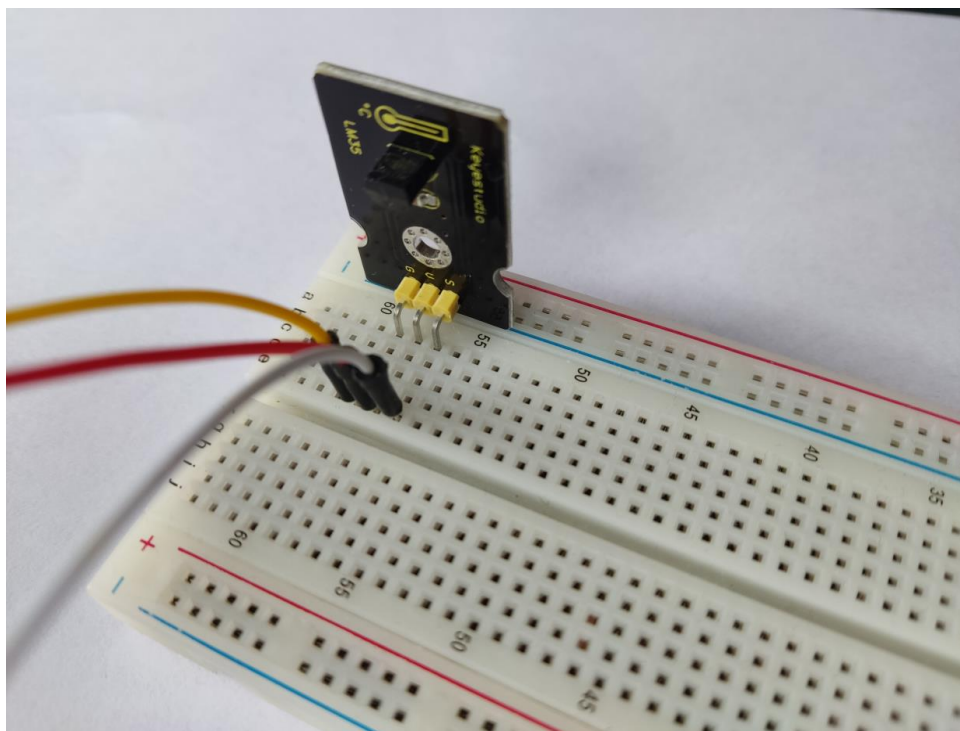
Mjerenja u pokusu smo izveli u dva dijela. Prvi je promjena svjetlosne jakosti u ovisnosti o udaljenosti od svijeće, a drugi promjena temperature u ovisnosti o udaljenosti. No prije samog sastavljanja sklopa, potrebno je proučiti same komponente koje ga sastavljaju. Na slici ispod su prikazane komponente za sastavljanja prvog pokusa. Koristimo Arduino mikrokontroler s dodatkom za prikupljanje podataka na kojeg spajamo SD karticu. Za spajanje senzora koristimo odgovarajuće žice s konektorima iz paketa s mikrokontrolerom, a za spajanje na računalo USB kabel. Za napajanje kada Arduino nije spojen na računalo koristimo bateriju.



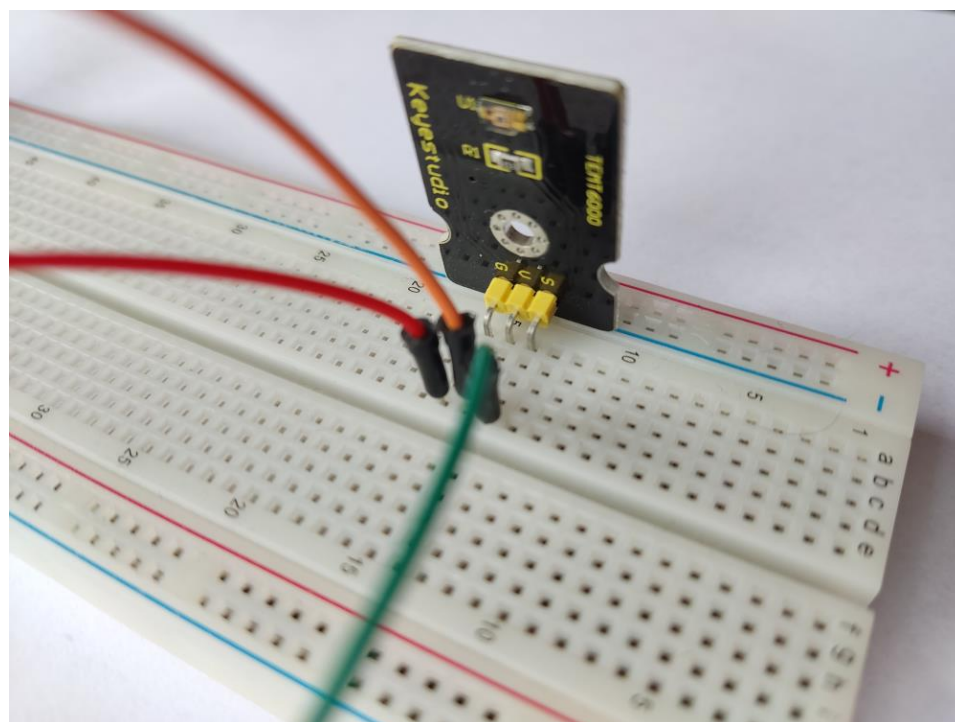


**Slika 9** Skup komponenti potrebnih za provedbu eksperimenta

Kako bi pripremili mikrokontroler za ovaj pokus, koristimo dva senzora, TEMT6000 senzor svjetlosne jakosti i LM35 senzor temperature. Napon 5V i GND sa senzora spajamo na odgovarajuće pinove na mikrokontroleru.

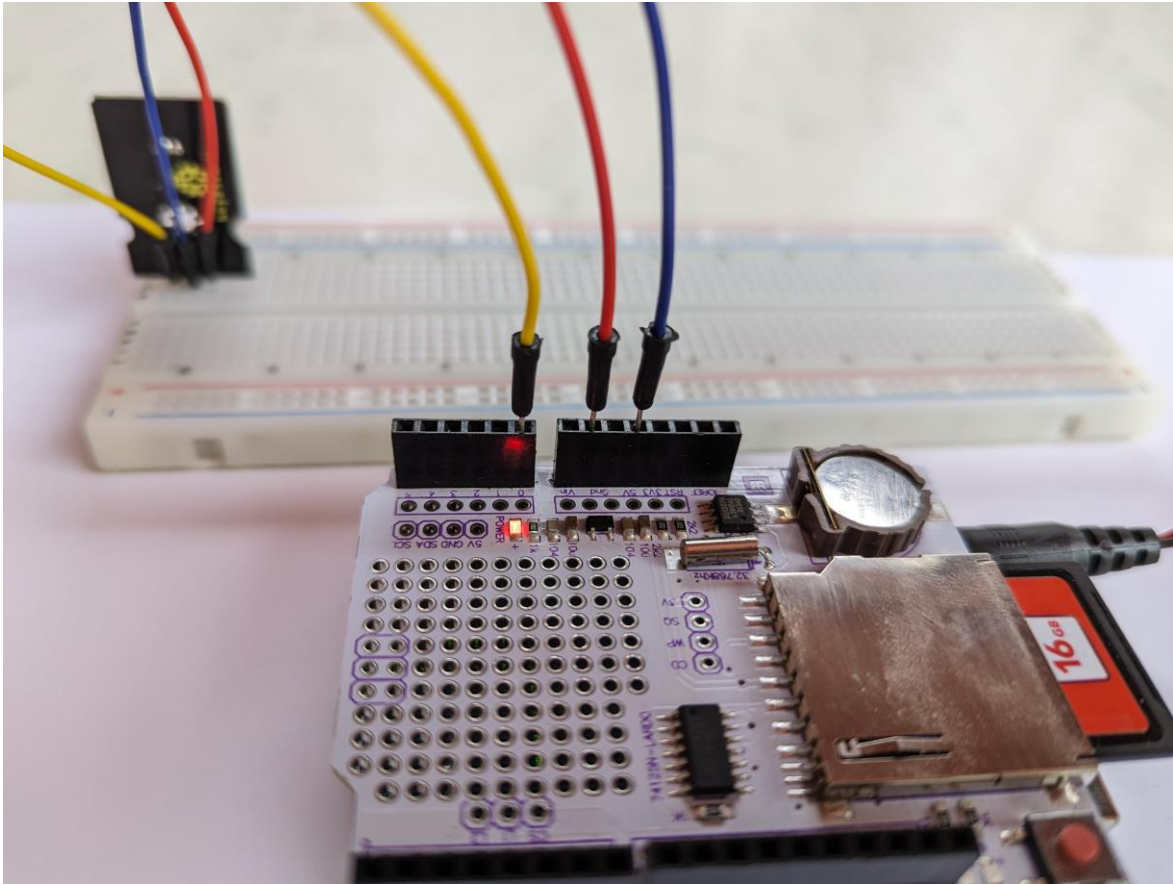


**Slika 10** Senzor temperature LM35 spojen na ploču za spajanje komponenti



**Slika 11** Senzor svjetlosne jakosti TEMT6000 spojen na ploču za spajanje komponenti

Zadnji pin na oba senzora je S i on nam daje izlazni signal, odnosno mjerne rezultate koje mikrokontroler poslije pretvara u temperaturu odnosno svjetlosnu jakost. Taj S pin spajamo na pinove A0 i A1 na mikrokontroleru, gdje u kodu definiramo koji senzor je spojen na koji pin.



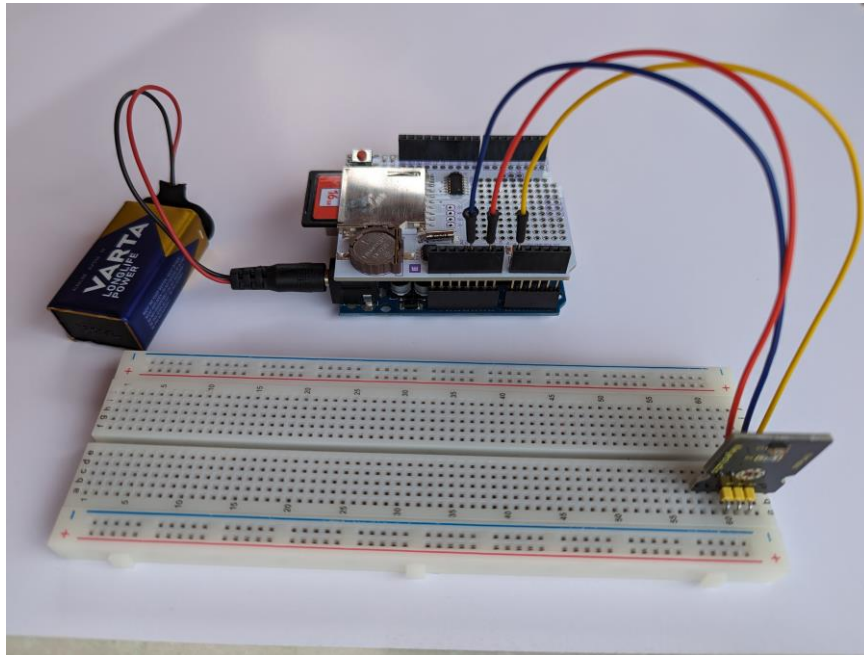
**Slika 12** Prikaz spremnog postava za mjerenje na mikrokontroleru

Kod spajanja sklopa koristimo ploču za spajanje komponenti koja se nalazi u kutiji sa Arduinoom. Ploča za spajanje komponenti poznata i kao "breadboard" ili "prototyping board", funkcionira kao poligon za testiranje elektroničkih komponenti i testiranje elektroničkih sklopova. Ova ploča ima rupice koje su razmaknute na standardnoj udaljenosti od 2,54 mm i povezane su metalnim trakama ispod ploče. One omogućuju da lako umetnete i spojite elektroničke komponente bez potrebe za lemljenjem.

Nakon spajanja komponenti u ploču, odgovarajući pinovi se spoje žicama. Kontakti na ploči su organizirani u redove i stupce s odgovarajućim oznakama, što olakšava organizaciju komponenti i žica. Arduino ploča sa kontaktima omogućuje brzo testiranje sklopova, što je korisno u razvoju projekata.

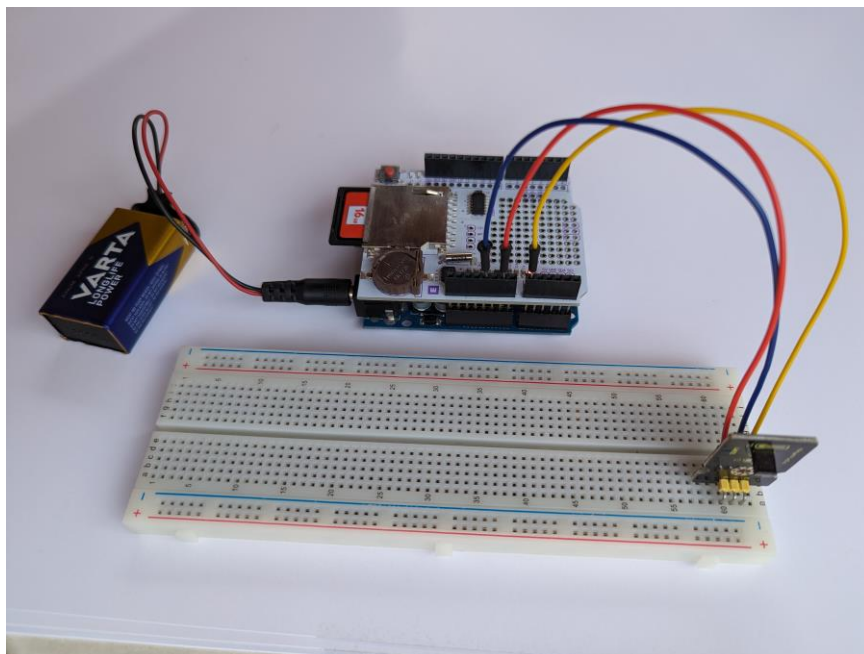


Svaki red ploče je zasebno kratko spojen i tako redom od prvog do šezdeset i trećeg. Sklop za mjerenje promjene svjetlosne jakosti u ovisnosti o udaljenosti od svijeće prikazan je na slici.



**Slika 13 Slika spremnog sklopa za mjerenje svjetlosne jakosti**

Sklop za mjerenje promjene temperature u ovisnosti o udaljenosti prikazan je na Slici 14.



**Slika 14 Slika spremnog sklopa za mjerenje temperature**

Kako bi sklop mogao raditi bez računala, spajamo bateriju preko adaptera. Sljedeći korak je kodirati mikrokontroler tako da se mjerenje pohranjuje na SD karticu. Počnimo s kodiranjem pločice za pokus sa mjerenjem svjetlosne jakosti.

```
#include <SD.h>
#include <Wire.h>
#include "RTClib.h"
```

„SD.h“ je biblioteka koja omogućuje Arduino kontrolu nad SD karticama. „Wire.h“ je biblioteka za komunikaciju preko I2C sabirnice (engl. I2C bus). „RTClib.h“ je biblioteka koja omogućuje rad s Real-Time Clock modulima.

```
const int chipSelect = 10;
RTC_DS1307 rtc;

File dataFile;

void setup() {
  Serial.begin(9600);

  // Initialize RTC
  if (!rtc.begin()) {
    Serial.println("RTC initialization failed!");
    return;
  }
}
```

"Initialize RTC" znači inicijalizirati RTC (Real-Time Clock) modul, što obično podrazumijeva uspostavljanje veze između mikrokontrolera i RTC-a te postavljanje parametara poput datuma i vremena. U ovom konkretnom primjeru, RTC modul je DS1307, a biblioteka "RTClib.h" se koristi za upravljanje modulom. Funkcija "rtc.begin()" se poziva za inicijalizaciju RTC modula.

```
// Check if RTC is running and set the time if not
if (!rtc.isrunning()) {
  Serial.println("RTC is not running. Setting time...");
  rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
}
```

Ovaj dio koda provjerava da li RTC („Real Time Clock“) modul radi, odnosno da li se dobiva točno vrijeme. Ako se ne dobiva točno vrijeme, tada će se RTC modul postaviti na trenutno vrijeme. To je važno jer se u ovom kodu mjeri vrijeme i točno vrijeme je potrebno za precizno mjerenje.

```
// Initialize SD card
if (!SD.begin(chipSelect)) {
```

```

Serial.println("SD card initialization failed!");
return;
}

```

Ovaj dio koda je priprema za korištenje SD kartice kao medij za spremanje podataka. U ovom dijelu se koristi funkcija „SD.begin(chipSelect)“ koja inicijalizira SD karticu i definira koji pin na Arduino pločici će biti korišten kao „chip select“ pin. Ovo je važno jer se pomoću „chip select“ pina bira koja će se komponenta na SPI magistrali (u ovom slučaju SD kartica) komunicirati s Arduino pločicom.

```

// Open the data file for writing
dataFile = SD.open("data.txt", FILE_WRITE);
if (!dataFile) {
Serial.println("Could not open data.txt");
return;
}

```

U ovom slučaju se otvara datoteka "data.txt" na SD kartici, koja će se koristiti za spremanje podataka o vremenu, temperaturi i svjetlosnoj jakosti koje se očitavaju u petlji "loop" koda. Ako se datoteka ne uspije otvoriti, program će ispisati poruku "Could not open data.txt" i prekinuti daljnje izvođenje koda.

```

// Write header to data file
dataFile.println("Time          Light (Lux)");

Serial.println("RTC and SD card initialized and data file opened
successfully");
}

```

Ovaj dio koda otvara datoteku "data.txt" i ispisuje zaglavlje u nju, koje se sastoji od naziva stupaca ("Time", "Light") koji će se koristiti za bilježenje podataka. Dakle, funkcija „dataFile.println("Time Light");“ ispisuje zaglavlje datoteke.

```

void loop() {

// Get elapsed time since start of measurement
uint32_t elapsedTime = millis();

// Convert elapsed time to hours, minutes, seconds, and centiseconds
uint32_t hours = elapsedTime / 3600000UL;
elapsedTime -= hours * 3600000UL;
uint32_t minutes = elapsedTime / 60000UL;
elapsedTime -= minutes * 60000UL;
uint32_t seconds = elapsedTime / 1000UL;
elapsedTime -= seconds * 1000UL;

```

```
uint32_t centiseconds = elapsedTime / 10UL;
```

Ovaj dio koda konvertira proteklo vrijeme od početka mjerenja (u milisekundama) u sate, minute, sekunde i centisekunde kako bi se moglo prikazati u kasnijem ispisu podataka na serijski monitor i u zapisu u datoteku. Konverzija se radi tako da se vrijeme podijeli s potrebnim brojem milisekundi u jedinici vremena, npr. 3600000 za sat, a zatim se koristi operacija „modulo“ da bi se izračunao ostatak vremena koji se prebacuje u sljedeću jedinicu vremena.

```
// Convert analog value to light intensity (Lux)
float lightIntensity = map(lightValue, 0, 1023, 0, 1000); // Adjust
the mapping range according to your light sensor specifications
```

U ovom dijelu, funkcija „map()“ se koristi za pretvaranje analognih vrijednosti sa senzora svjetlosti „(lightValue)“ u odgovarajuću vrijednost svjetlosne jakosti „(lightIntensity)“. Funkcija „map()“ skalira i pretvara „lightValue“ iz ulaznog raspona u odgovarajuću vrijednost izlaznog raspona, koja se zatim dodjeljuje varijabli „lightIntensity“.

```
// Print time and light intensity to serial monitor
Serial.print(hours);
Serial.print(':');
if (minutes < 10) {
Serial.print('0');
}
Serial.print(minutes);
Serial.print(':');
if (seconds < 10) {
Serial.print('0');
}
Serial.print(seconds);
Serial.print('.');
if (centiseconds < 10) {
Serial.print('0');
}
Serial.print(centiseconds);
Serial.print(" ");
Serial.print(lightIntensity); // Print light intensity
Serial.println(" Lux");
```

Ovaj dio koda ispisuje vrijeme, temperaturu i svjetlosnu jakost na serijski monitor. Koristi funkciju „Serial.print()“ za ispisivanje vrijednosti varijabli „hours“, „minutes“, „seconds“, „centiseconds“ i „light“ na serijski monitor.

```
// Write data to data file
dataFile.print(hours);
```

```

dataFile.print(':');
if (minutes < 10) {
dataFile.print('0');
}
dataFile.print(minutes);
dataFile.print(':');
if (seconds < 10) {
dataFile.print('0');
}
dataFile.print(seconds);
dataFile.print('.');
if (centiseconds < 10) {
dataFile.print('0');
}
dataFile.print(centiseconds);
dataFile.print("    ");
dataFile.print(String(light)); // print light value
dataFile.println();

```

Ovaj dio koda upisuje vrijednosti vremena i svjetlosne jakosti u datoteku. Konkretno, koristi se „dataFile“ objekt koji je stvoren u funkciji „setup()“ i otvara datoteku s imenom "datalog.txt" u načinu dodavanja. Nakon toga, vrijednosti vremena i svjetlosne jakosti su formatirane i upisane u datoteku koristeći „dataFile.print()“ i „dataFile.println()“ funkcije. Nakon upisa podataka, funkcija „dataFile.flush()“ osigurava da su podaci zapisani na SD karticu. U ovom slučaju se može podesiti broj decimalnih mjesta koje želimo da nam se zapisuju u „data.txt“ datoteku, međutim, ni sam senzor nema mogućnost određivanja velike preciznosti.

```

// Flush the data file to ensure data is written to the SD card
dataFile.flush();

```

Ovaj dio koda se odnosi na upisivanje podataka (vremena, temperature i svjetlosne jakosti) u datoteku na SD karticu. Kada podaci budu upisani u datoteku, treba osigurati da se oni stvarno zapisu na karticu, a to se postiže s „flush()“ metodom. To je vrlo važno jer se u slučaju neočekivanog prekida napajanja podaci mogu izgubiti ako nisu pravilno zapisani na SD karticu.

```

// Wait for 100 milliseconds before taking another measurement
(500);
}

```

Završni dio koda služi da si sami postavimo učestalost očitavanja mjerenja. Trenutna vrijednost je postavljena na 500 milisekundi.

U drugom slučaju, kod mjerenja temperature, kod se razlikuje u sljedećim stavkama:



- `// Write header to data file`  
`dataFile.println("Time                    Temperature (C)");`

Sada definiramo ispis temperature u početku tablice koju ispisujemo mjerenjem.

- Umjesto skaliranja vrijednosti svjetlosne jakosti, ovdje imamo korekciju temperature

```
// Read temperature sensor
float temperature = analogRead(A0) * 0.00488;
temperature = temperature * 100.0;
```

Ovaj dio koda čita vrijednost temperature sa senzora i pretvara je u stvarnu temperaturu u Celzijusima. Funkcija „analogRead(A0)“ čita analognu vrijednost sa pina A0. „\* 0.00488“ množi analognu vrijednost s faktorom 0.00488 kako bi se pretvorila u stvarni napon. Ovaj faktor se koristi kada je referentni napon postavljen na 5 volti. „temperature = temperature \* 100.0“ množi pretvoreni napon s faktorom 100.0 kako bi se pretvorio u temperaturu.

Ovaj faktor ovisi o karakteristikama specifičnog senzora.

Na kraju izvršavanja ovog dijela koda, varijabla temperature će sadržavati stvarnu temperaturu u Celzijusima koja je očitana s analognog pina A0.

- U zapisu podataka u „data.txt“ mijenjamo varijablu iz svjetlosne jakosti u temperaturu, gdje dodajem 2 kako bi se zapisivala vrijednost temperature sa dvije decimalne točke.  
`Serial.print(String(temperature, 2)); // print temperature with two decimal points`

### 2.3.2 Uputa za nastavnika

Odgojno-obrazovni ishodi (OOI) su:

OOI1.1 - Razumjeti i primijeniti pojmove veličine, mjere i mjernih jedinica:

- Razumjeti koncepte veličine, mjerne jedinice, točnost i preciznost mjerenja, te neizvjesnost mjerenja
- Razumjeti kako se koriste osnovne mjere u fizici: duljina, vrijeme i masa
- Primijeniti osnovne mjere za mjerenje svjetlosne jakosti i temperature te interpretirati rezultate mjerenja

OOI1.4 - Razumjeti, primijeniti i interpretirati zakone i formule:

- Razumjeti pojam zakona u fizici i primijeniti ih u mjerenju svjetlosne jakosti i temperature
- Primijeniti zakone i formule za računanje osnovnih veličina u fizici, poput svjetlosne jakosti i temperature, te interpretirati rezultate mjerenja

OOI2.1 - Razumjeti i primijeniti zakon očuvanja energije:

- Razumjeti koncepte energije i njezine različite oblike, poput toplinske energije
- Primijeniti zakon očuvanja energije u pokusu mjerenja temperature i interpretirati rezultate

OOI3.1 - Razumjeti i primijeniti osnovne koncepte elektromagnetizma:

- Razumjeti koncept elektromagnetizma i primijeniti ga u mjerenju svjetlosne jakosti
- Razumjeti kako djeluje senzor svjetlosne jakosti i kako se povezuje s Arduinoom

OOI3.3 - Razumjeti i primijeniti osnovne koncepte termodinamike:

- Razumjeti osnovne koncepte termodinamike, poput topline, temperature, termodinamičkog rada i procesa
- Primijeniti osnovne koncepte termodinamike u mjerenju temperature i interpretirati rezultate

Uvod u temu:

Objasnite učenicima što je Arduino, kako se koristi u mjerenju i koje su prednosti upotrebe Arduina u mjerenju svojstva izvora svjetlosti.

Istraživačka pitanja:

Postavite učenicima sljedeća istraživačka pitanja koja će ih potaknuti na razmišljanje o temi:

1. Kako Arduino mjeri svjetlosnu jakost?
2. Kako Arduino mjeri temperaturu?
3. Koja je svrha mjerenja svojstava izvora svjetlosti?
4. Kako se izmjerena svjetlosna jakost može primijeniti u praksi?
5. Koje su prednosti i nedostaci korištenja Arduina u mjerenju svojstva izvora svjetlosti?

Ciljevi:

- Učenje o senzorima svjetlosne jakosti i temperature i njihovoj upotrebi u kombinaciji s Arduinoom
- Razvijanje sposobnosti u istraživanju, analiziranju i interpretiranju podataka
- Poboljšanje vještina u programiranju i upotrebi različitih komponenti

Potrebni materijali:

- Arduino UNO ploča
- Senzor svjetlosne jakosti
- Senzor temperature
- Računalo s instaliranim Arduino IDE softverom
- USB kabel za povezivanje Arduina s računalom
- Kabel za povezivanje senzora s Arduinoom
- SD kartica i pripadni adapter
- Baterija i pripadni adapter

Organizacija:

Uvod u senzore svjetlosne jakosti i temperature i njihovu upotrebu u kombinaciji s Arduino pločom. Objasniti kako senzori rade i kako se njihova vrijednost može prikazati i interpretirati pomoću Arduina. Učenici će sastaviti sklop koji će se sastojati od senzora svjetlosne jakosti i senzora temperature koji su spojeni na Arduino ploču. Koristeći Arduino IDE softver, učenici će napisati program koji će omogućiti čitanje i prikaz vrijednosti senzora na serijskom monitoru na računalu. Učenici će provesti eksperimente s različitim izvorima svjetlosti, izmjeriti njegovo svojstvo pomoću senzora svjetlosne jakosti, te temperaturu okoline pomoću senzora temperature. Učenici će interpretirati podatke dobivene iz eksperimenata i napraviti zaključke o utjecaju različitih izvora svjetlosti na temperaturu okoline.

Primjer radnog listića:

1. Sastavite sklop koji se sastoji od Arduina, senzora svjetlosne jakosti, a zatim sklop koji se sastoji od Arduina i senzora temperature.
2. Učitajte primjer koda za čitanje vrijednosti senzora u Arduino IDE softver.

3. Provjerite je li sklop ispravno spojen i pokrenite program.
4. Prikupite podatke o svjetlosnoj jakosti i temperaturi okoline za odabrani izvor svjetlosti (svijeća, LED svjetiljka). Druga ideja pokusa je da promatramo svjetlost i temperaturu kako se približavamo svjetiljci ili svijeći u ekvidistantnim razmacima.
5. Analizirajte podatke i napravite graf koji prikazuje vrijednosti temperature i svjetlosne jakosti za ekvidistantne razmake.
6. Interpretirajte podatke i napišite zaključke o utjecaju različitih izvora svjetlosti na temperaturu okoline.

Analiza podataka:

Učenici mogu usporediti svoje rezultate s rezultatima drugih grupa i raspravljati o razlikama u izmjerenoj vrijednosti senzora svjetlosne jakosti. Također, učenici mogu raspravljati o prednostima i nedostacima korištenja Arduina u mjerenju svojstva izvora svjetlosti i kako bi se tehnologija mogla primijeniti u praksi.

Zaključak:

Na kraju, učenici mogu izraditi prezentacije u kojima će predstaviti svoje rezultate i zaključke o korištenju Arduina u mjerenju svojstva izvora svjetlosti. Također, mogu raspravljati o primjeni Arduina u drugim područjima i predložiti ideje.

### **2.3.3 Rezultati i diskusija**

Nakon instalacije koda na mikrokontroler, spremni smo za mjerenje. Pokus izvodimo tako da mračnoj prostoriji zapalimo svijeću, te pomičemo senzor u ekvidistantnim razmacima prema svijeći kako bi izmjerili ovisnost temperature i svjetlosne jakosti o udaljenosti od izvora svjetlosti. Na slici je prikazana izvedba prvog pokusa.



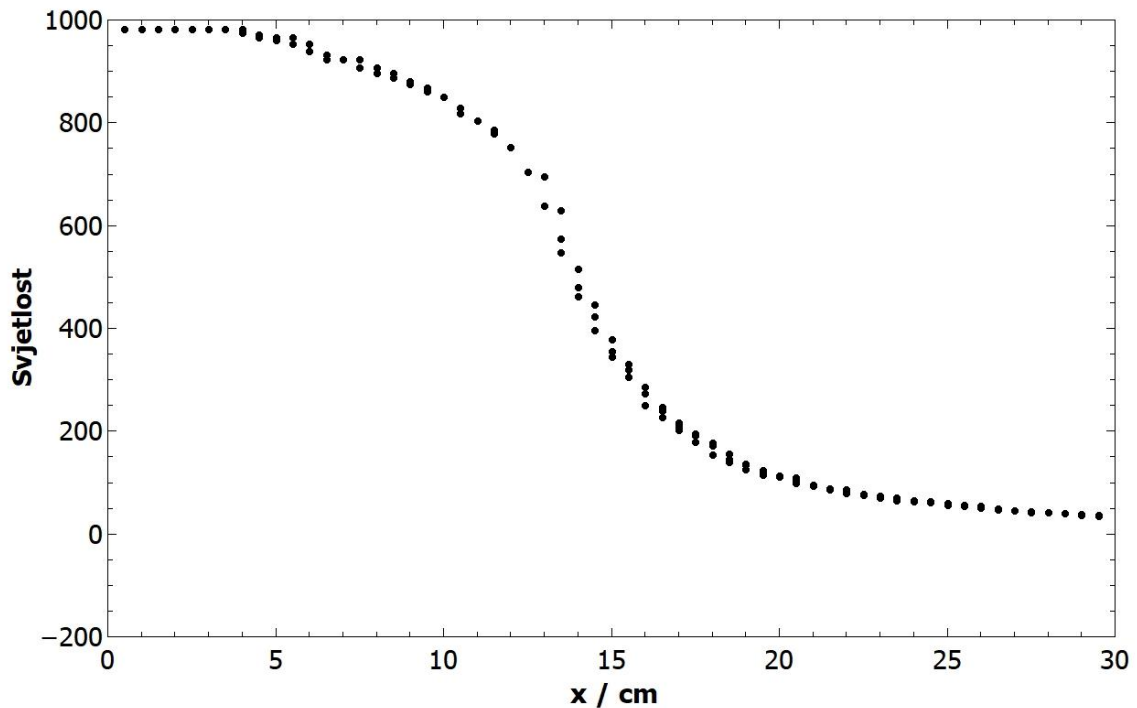
**Slika 15 Izvedba pokusa Svojstva izvora svjetlosti**

Rezultati mjerenja svjetlosne jakosti u ovisnosti o udaljenosti izgledaju ovako:

		0:00:10.51	41.00	0:00:22.47	64.00
Time	Light (Lux)	0:00:11.03	42.00	0:00:22.99	62.00
		0:00:11.55	42.00	0:00:23.51	63.00
0:00:00.11	62.00	0:00:12.07	43.00	0:00:24.03	66.00
0:00:00.65	62.00	0:00:12.59	43.00	0:00:24.55	66.00
0:00:01.17	61.00	0:00:13.11	43.00	0:00:25.06	66.00
0:00:01.70	61.00	0:00:13.63	44.00	0:00:25.58	68.00
0:00:02.22	60.00	0:00:14.15	45.00	0:00:26.10	70.00
0:00:02.73	60.00	0:00:14.67	46.00	0:00:26.62	71.00
0:00:03.25	60.00	0:00:15.20	46.00	0:00:27.14	72.00
0:00:03.77	60.00	0:00:15.71	48.00	0:00:27.66	74.00
0:00:04.29	10.00	0:00:16.23	49.00	0:00:28.18	76.00
0:00:04.81	9.00	0:00:16.75	50.00	0:00:28.69	78.00
0:00:05.33	17.00	0:00:17.27	51.00	0:00:29.21	78.00
0:00:05.84	38.00	0:00:17.79	53.00	0:00:29.73	80.00
0:00:06.36	35.00	0:00:18.31	54.00	0:00:30.25	82.00
0:00:06.88	37.00	0:00:18.82	56.00	0:00:30.78	86.00
0:00:07.40	36.00	0:00:19.34	55.00	0:00:31.30	86.00
0:00:07.92	37.00	0:00:19.86	57.00	0:00:31.82	89.00
0:00:08.44	38.00	0:00:20.38	56.00	0:00:32.34	89.00
0:00:08.96	39.00	0:00:20.91	60.00	0:00:32.86	94.00
0:00:09.47	40.00	0:00:21.43	60.00	0:00:33.37	95.00
0:00:09.99	40.00	0:00:21.95	63.00	0:00:33.89	96.00

0:00:34.42	100.00	0:00:44.79	195.00	0:00:55.20	480.00
0:00:34.94	104.00	0:00:45.31	202.00	0:00:55.71	516.00
0:00:35.46	109.00	0:00:45.83	210.00	0:00:56.23	548.00
0:00:35.98	112.00	0:00:46.37	217.00	0:00:56.75	574.00
0:00:36.50	113.00	0:00:46.88	228.00	0:00:57.27	630.00
0:00:37.01	113.00	0:00:47.40	247.00	0:00:57.79	638.00
0:00:37.53	115.00	0:00:47.92	239.00	0:00:58.31	695.00
0:00:38.05	118.00	0:00:48.44	250.00	0:00:58.83	804.00
0:00:38.57	124.00	0:00:48.97	273.00	0:00:59.35	862.00
0:00:39.09	125.00	0:00:49.49	286.00	0:00:59.87	923.00
0:00:39.61	134.00	0:00:50.01	306.00	0:01:00.39	956.00
0:00:40.13	136.00	0:00:50.53	320.00	0:01:00.91	962.00
0:00:40.64	140.00	0:00:51.05	330.00	0:01:01.43	966.00
0:00:41.16	145.00	0:00:51.57	345.00	0:01:01.95	970.00
0:00:41.68	156.00	0:00:52.08	355.00	0:01:02.47	973.00
0:00:42.20	155.00	0:00:52.60	378.00	0:01:02.99	975.00
0:00:42.72	172.00	0:00:53.12	396.00	0:01:03.51	978.00
0:00:43.24	177.00	0:00:53.64	424.00	0:01:04.03	980.00
0:00:43.76	180.00	0:00:54.16	447.00	0:01:04.55	981.00
0:00:44.27	191.00	0:00:54.68	462.00	0:01:05.07	982.00

Na grafu ispod je prikazana ovisnost svjetlosne jakosti o udaljenosti senzora od svijeće.

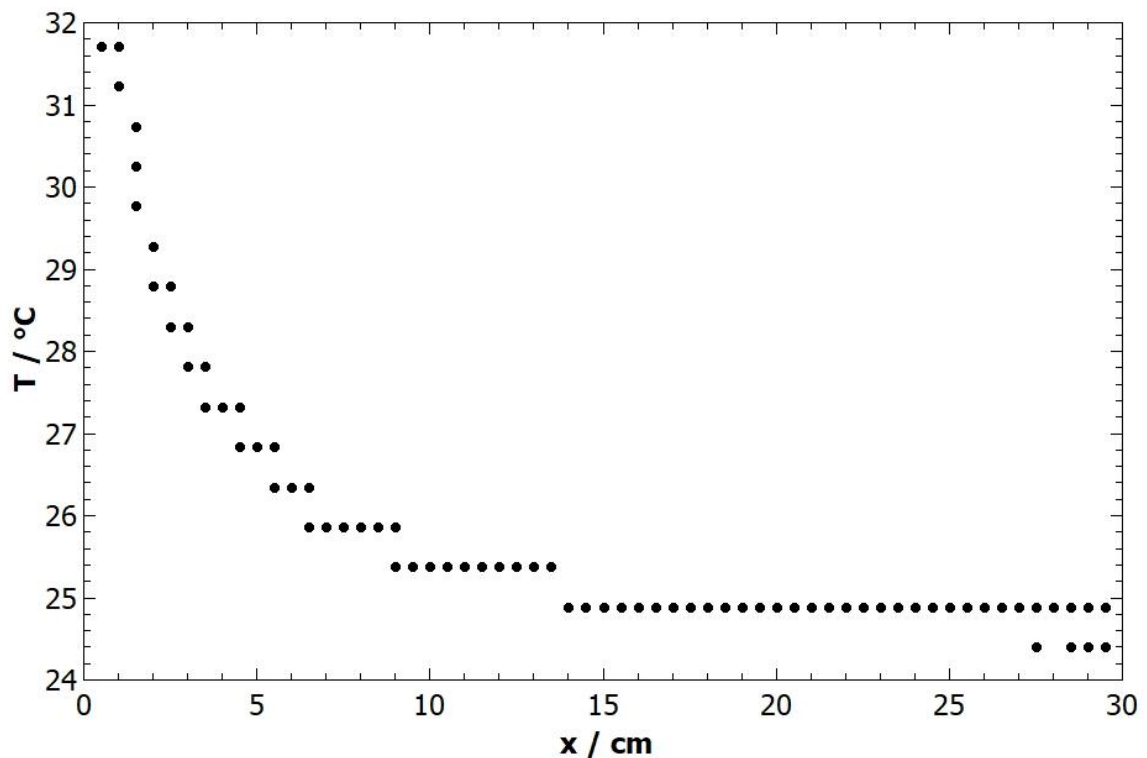


Slika 16 Graf ovisnosti svjetlosne jakosti o udaljenosti od izvora svjetlosti

Vidljivo je da se svjetlost i temperatura mijenjaju potpuno različito. Kod svjetlosti imamo najizraženiju promjenu između 10 i 20 centimetara razmaka, kod čega je promjena najizraženije oko 14 centimetara. Prvih 5 centimetara i zadnjih 5 je gotovo konstanta. Rezultati mjerenja temperature u ovisnosti o udaljenosti za prvih 30 sekundi izgledaju ovako:

Time	Temperature (C)					
		0:00:20.36	24.89 C		0:00:41.11	24.89 C
0:00:00.11	24.40 C	0:00:20.88	24.89 C		0:00:41.63	24.89 C
0:00:00.65	24.89 C	0:00:21.40	24.89 C		0:00:42.15	24.89 C
0:00:01.17	24.40 C	0:00:21.92	24.89 C		0:00:42.67	24.89 C
0:00:01.69	24.40 C	0:00:22.44	24.89 C		0:00:43.20	24.89 C
0:00:02.21	24.40 C	0:00:22.96	24.89 C		0:00:43.72	24.89 C
0:00:02.73	24.40 C	0:00:23.48	24.89 C		0:00:44.23	24.89 C
0:00:03.24	24.89 C	0:00:23.99	24.89 C		0:00:44.75	24.89 C
0:00:03.76	24.40 C	0:00:24.52	24.89 C		0:00:45.28	24.89 C
0:00:04.28	24.89 C	0:00:25.04	24.89 C		0:00:45.80	24.89 C
0:00:04.80	24.89 C	0:00:25.55	24.89 C		0:00:46.32	24.89 C
0:00:05.32	24.40 C	0:00:26.07	24.89 C		0:00:46.83	24.89 C
0:00:05.84	24.40 C	0:00:26.59	24.89 C		0:00:47.35	24.89 C
0:00:06.36	24.89 C	0:00:27.11	24.89 C		0:00:47.87	24.89 C
0:00:06.87	24.89 C	0:00:27.63	24.89 C		0:00:48.39	24.89 C
0:00:07.39	24.40 C	0:00:28.15	24.89 C		0:00:48.90	24.89 C
0:00:07.91	24.89 C	0:00:28.67	24.89 C		0:00:49.42	24.89 C
0:00:08.43	24.40 C	0:00:29.19	24.89 C		0:00:49.94	24.89 C
0:00:08.95	24.40 C	0:00:29.71	24.89 C		0:00:50.46	24.89 C
0:00:09.47	24.89 C	0:00:30.22	24.89 C		0:00:50.98	24.89 C
0:00:09.99	24.89 C	0:00:30.74	24.89 C		0:00:51.49	24.89 C
0:00:10.51	24.89 C	0:00:31.26	24.89 C		0:00:52.01	24.89 C
0:00:11.02	24.89 C	0:00:31.78	24.89 C		0:00:52.53	24.89 C
0:00:11.54	24.89 C	0:00:32.30	24.89 C		0:00:53.05	24.89 C
0:00:12.07	24.89 C	0:00:32.81	24.89 C		0:00:53.57	24.89 C
0:00:12.59	24.89 C	0:00:33.33	24.89 C		0:00:54.08	24.89 C
0:00:13.11	24.40 C	0:00:33.85	24.89 C		0:00:54.60	24.89 C
0:00:13.62	24.89 C	0:00:34.37	24.89 C		0:00:55.12	24.89 C
0:00:14.14	24.89 C	0:00:34.88	24.89 C		0:00:55.64	25.38 C
0:00:14.66	24.89 C	0:00:35.41	24.89 C		0:00:56.16	25.38 C
0:00:15.17	24.89 C	0:00:35.93	24.89 C		0:00:56.68	25.38 C
0:00:15.70	24.89 C	0:00:36.44	24.89 C		0:00:57.20	25.38 C
0:00:16.22	24.89 C	0:00:36.96	24.89 C		0:00:57.72	25.38 C
0:00:16.74	24.89 C	0:00:37.48	24.89 C		0:00:58.24	25.38 C
0:00:17.26	24.89 C	0:00:38.00	24.89 C		0:00:58.76	25.38 C
0:00:17.77	24.89 C	0:00:38.52	24.89 C		0:00:59.28	25.38 C
0:00:18.29	24.89 C	0:00:39.04	24.89 C		0:00:59.80	25.38 C
0:00:18.81	24.89 C	0:00:39.56	24.89 C		0:01:00.32	25.38 C
0:00:19.33	24.89 C	0:00:40.07	24.89 C		0:01:00.84	25.38 C
0:00:19.85	24.89 C	0:00:40.59	24.89 C			

Na grafu ispod je prikazana ovisnost temperature o udaljenosti od svijeće.



**Slika 17 Graf ovisnosti temperature o udaljenosti od izvora svjetlosti**

Prva stvar koja upada u oči je preciznost senzora temperature. Vidljivo je iz grafa i rezultata da sam senzor nema sposobnost mjerenja između vrijednosti 24.89 Celzija i 25.38 Celzija, već su vrijednosti ograničene. Vidimo da nema značajne promjene temperature dok sa senzorom ne dođemo do 15 centimetara, tada dobivamo blagi porast temperature. S daljnjim približavanjem očituje se eksponencijalni rast temperature.

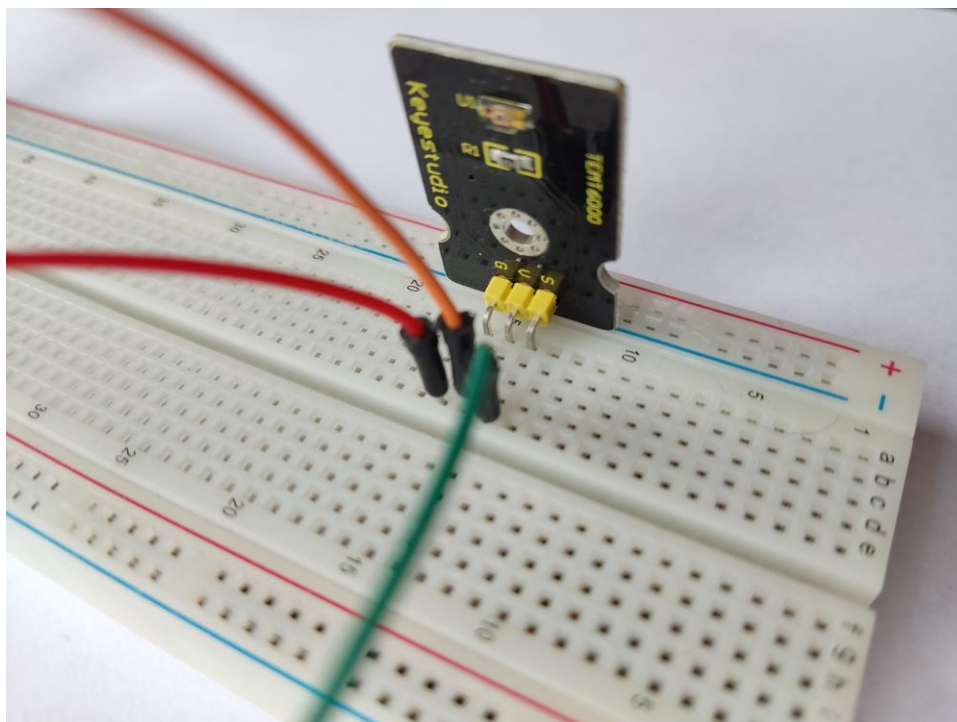
## **2.4 Pokus: Energija u okolišu: Svjetlost, toplina i zvuk**

### **2.4.1 Eksperimentalni postav**

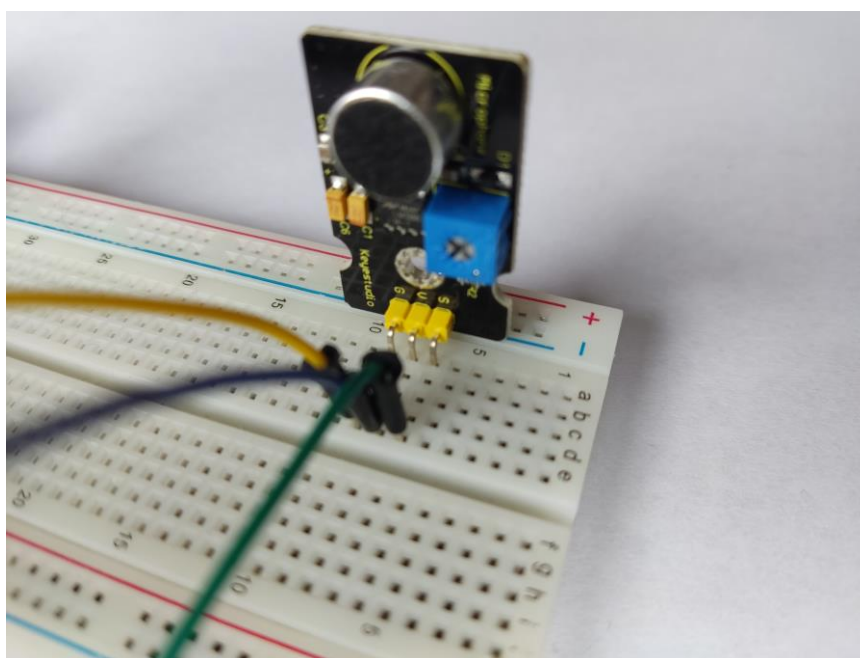
Kako bi pripremili mikrokontroler za ovaj pokus, koristimo tri senzora, TEMT6000 senzor svjetlosne jakosti, LM35 senzor temperature i senzor jakosti zvuka KS0035. Sklop spajamo tako da na V i GND priključke sva tri senzora spojimo na 5V i GND na mikrokontroleru. Senzor temperature LM35 spajamo kao na Slici 18.





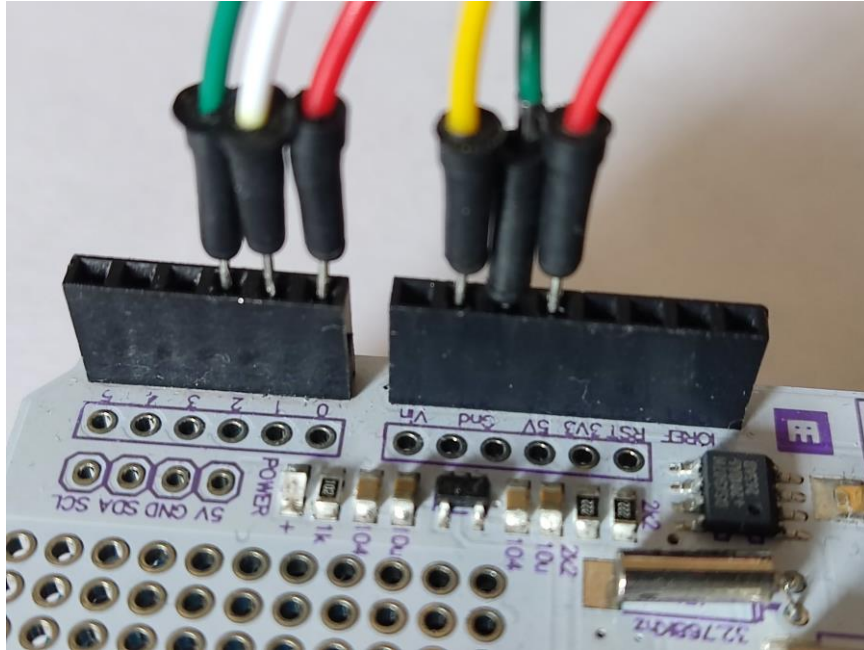


**Slika 19** Senzor jakosti svjetlosti TEMT6000 spojen na ploču za spajanje komponenti



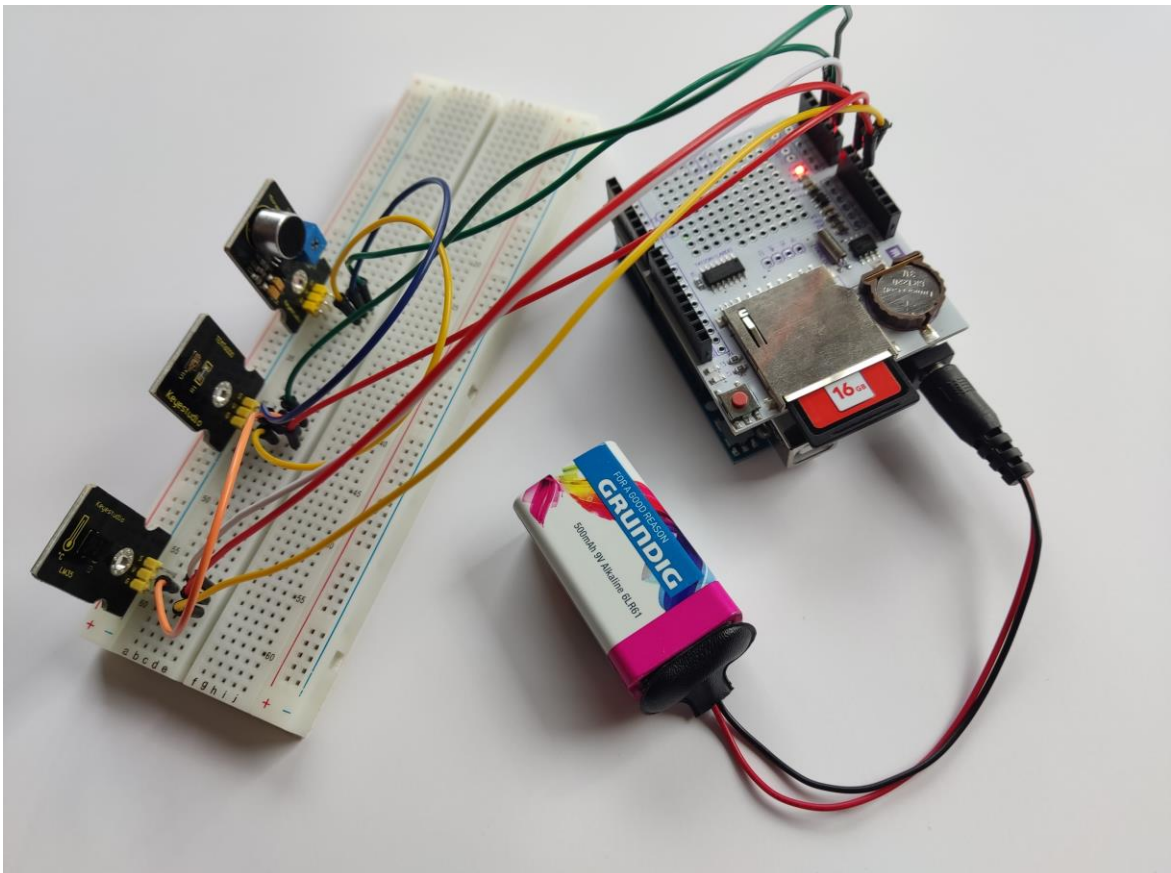
**Slika 20** Senzor zvuka KS0035 spojen na ploču za spajanje komponenti

Nakon spajanja, trebali bismo imati spojene senzore redom na A1, A2 i A0 pinove, napajanje spojeno preko pina 5 V, te GND pin koji vodi do „G“ kontakta na svakom senzoru. Na Slici 21 vidimo kako izgledaju kontakti na mikrokontroleru nakon spajanja sklopa.



Slika 21 Svi pinovi spojeni na Arduino mikrokontroler

Nakon organizacije žica i zgodnije razmještaja komponenata na ploči, sklop izgleda kao na Slici 22.



Slika 22 Slika spremnog sklopa za mjerenje svjetlosne jakosti, temperature i zvuka

Sljedeći korak našeg pokusa je programiranje mikrokontrolera.

```
#include <SD.h>
#include <Wire.h>
#include "RTClib.h"
```

„SD.h“ je biblioteka koja omogućuje Arduino kontrolu nad SD karticama. „Wire.h“ je biblioteka za komunikaciju preko I2C sabirnice (engl. I2C bus). „RTClib.h“ je biblioteka koja omogućuje rad s Real-Time Clock modulima.

```
const int chipSelect = 10;
RTC_DS1307 rtc;

File dataFile;

void setup() {
  Serial.begin(9600);

  // Initialize RTC
  if (!rtc.begin()) {
    Serial.println("RTC initialization failed!");
    return;
  }
```

"Initialize RTC" znači inicijalizirati RTC (Real-Time Clock) modul, što obično podrazumijeva uspostavljanje veze između mikrokontrolera i RTC-a te postavljanje parametara poput datuma i vremena. U ovom konkretnom primjeru, RTC modul je DS1307, a biblioteka "RTClib.h" se koristi za upravljanje modulom. Funkcija "rtc.begin()" se poziva za inicijalizaciju RTC modula.

```
// Check if RTC is running and set the time if not
if (!rtc.isrunning()) {
  Serial.println("RTC is not running. Setting time...");
  rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
}
```

Ako RTC modul nije pokrenut (metoda isrunning() vraća false), ispisuje se poruka "RTC is not running. Setting time..." putem serijskog monitora. Nakon toga se postavlja vrijeme na RTC modulu pomoću metode adjust(). Vrijeme se postavlja na trenutno datum i vrijeme preuzeto iz makroa \_\_DATE\_\_ i \_\_TIME\_\_.

```
// Initialize SD card
if (!SD.begin(chipSelect)) {
  Serial.println("SD card initialization failed!");
  return;
}
```



```
}
```

Ovaj dio koda služi za inicijalizaciju SD kartice i provjeru uspješnosti inicijalizacije. Ako inicijalizacija SD kartice na odabranom CS („Chip Select“) pinu nije uspješna, ispisuje se poruka "SD card initialization failed!" putem serijskog monitora. Nakon toga, izvršavanje programa se prekida korištenjem „return“, čime se izlazi iz „setup()“ funkcije i prelazi na loop() funkciju. Ova provjera osigurava da se program ne nastavi izvršavati s neispravno inicijaliziranom SD karticom, što može rezultirati greškama pri pisanju ili čitanju podataka s kartice.

```
// Open the data file for writing
dataFile = SD.open("data.txt", FILE_WRITE);
if (!dataFile) {
  Serial.println("Could not open data.txt");
  return;
}
```

Ovaj dio koda otvara datoteku "data.txt" na SD kartici za pisanje podataka. Ako otvaranje nije uspješno, ispisuje se poruka "Could not open data.txt" i program prekida izvršavanje. Otvorena datoteka se pohranjuje u objekt „dataFile“.

```
// Write header to data file
dataFile.println("Time      Temperature (C)  Light  Sound");

Serial.println("RTC and SD card initialized and data file opened
successfully");
}
```

Ovaj dio koda piše zaglavlje u datoteku. Zaglavlje sadrži tekstualne oznake za vrijeme, temperaturu, svjetlosne jakosti i jakost zvuka. Nakon pisanja zaglavlja, ispisuje se poruka "RTC and SD card initialized and data file opened successfully" putem serijskog monitora kako bi se potvrdilo uspješno inicijaliziranje RTC modula, SD kartice i otvaranje datoteke.

```
void loop() {
  // Get elapsed time since start of measurement
  uint32_t elapsedTime = millis();
```

Ovaj dio koda u petlji „loop()“ koristi funkciju „millis()“ za dobivanje proteklog vremena (u milisekundama) od početka mjerenja. Vrijednost proteklog vremena sprema se u varijablu „elapsedTime“. Ova varijabla omogućuje praćenje proteklog vremena određenog dijela koda unutar petlje „loop()“.

```
// Convert elapsed time to hours, minutes, seconds, and centiseconds
uint32_t hours = elapsedTime / 3600000UL;
```

```

elapsedTime -= hours * 3600000UL;
uint32_t minutes = elapsedTime / 60000UL;
elapsedTime -= minutes * 60000UL;
uint32_t seconds = elapsedTime / 1000UL;
elapsedTime -= seconds * 1000UL;
uint32_t centiseconds = elapsedTime / 10UL;

```

Ovaj dio koda konvertira proteklo vrijeme izraženo u milisekundama u sate, minute, sekunde i stotinke sekunde. Varijabla „hours“ dobiva broj sati, „minutes“ dobiva broj minuta, „seconds“ dobiva broj sekundi, a „centiseconds“ dobiva broj stotinki sekunde. Korištenjem aritmetičkih operacija dijeljenja i oduzimanja određuje se svaki dio vremena na temelju preostalih milisekundi. Ovaj postupak omogućuje prikazivanje proteklog vremena u željenom formatu.

```

// Read temperature sensor
float temperature = analogRead(A0) * 0.00488;
temperature = temperature * 100.0;

```

Ovaj dio koda čita analognu vrijednost s temperaturnog senzora spojenog na pin A0. Dobivena vrijednost se skalira kako bi se pretvorila u temperaturu, množenjem s faktorom 0.00488 i množenjem s 100.0.

```

// Read light sensor
float light = analogRead(A1);

// Read sound sensor
int sound = analogRead(A2);

```

Ovaj dio koda čita vrijednosti s analognih senzora svjetlosne jakosti i jakosti zvuka spojenih na odgovarajuće analogne pinove A1 i A2. Vrijednost svjetlosti se sprema u varijablu „light“ tipa „float“, dok se vrijednost jakosti zvuka sprema u varijablu „sound“ tipa „int“. Ove vrijednosti se koriste za daljnju obradu i prikazivanje podataka.

```

// Print time, temperature, light, and sound to serial monitor
Serial.print(hours);
Serial.print(':');
if (minutes < 10) {
Serial.print('0');
}
Serial.print(minutes);
Serial.print(':');
if (seconds < 10) {
Serial.print('0');
}
Serial.print(seconds);
Serial.print('.');

```

```

if (centiseconds < 10) {
Serial.print('0');
}
Serial.print(centiseconds);
Serial.print("    ");
Serial.print(String(temperature, 2)); // print temperature with two
decimal points
Serial.print(" C    ");
Serial.print(String(light)); // print light value
Serial.print("    ");
Serial.print(sound); // print sound value
Serial.println();

```

U ovom dijelu koda se vrijeme, temperatura, svjetlosna jakost i zvuk ispisuju na serijskom monitoru. Vrijeme se prikazuje u formatu sati:minuta:sekunda.stotinka. Temperatura se ispisuje s dvije decimale i pridodaje se oznaka za stupnjeve Celzijusa. Vrijednosti svjetlosne jakosti i jakosti zvuka se ispisuju bez dodatnih modifikacija. Na kraju svakog ispisivanja se prelazi u novi redak.

```

// Write data to data file
dataFile.print(hours);
dataFile.print(':');
if (minutes < 10) {
dataFile.print('0');
}
dataFile.print(minutes);
dataFile.print(':');
if (seconds < 10) {
dataFile.print('0');
}
dataFile.print(seconds);
dataFile.print('.');
if (centiseconds < 10) {
dataFile.print('0');
}
dataFile.print(centiseconds);
dataFile.print("    ");
dataFile.print(String(temperature, 2)); // print temperature with two
decimal points
dataFile.print(" C    ");
dataFile.print(String(light)); // print light value
dataFile.print("    ");
dataFile.print(sound); // print sound value
dataFile.println();

```

U ovom dijelu koda se podaci zapisuju u datoteku. Vrijeme se zapisuje u formatu sati:minuta:sekunda.stotinka, s dodavanjem vodećih nula ako je potrebno. Nakon toga slijedi

razmak i zapisuje se temperatura s dvije decimale i oznakom za stupnjeve Celzijusa. Zatim se zapisuje vrijednost svjetlosne jakosti i jakosti zvuka. Sve vrijednosti se odvajaju razmacima. Nakon što su podaci zapisani, prelazi se u novi redak u datoteci.

```
// Write data to data file
dataFile.print(hours);
dataFile.print(':');
if (minutes < 10) {
dataFile.print('0');
}
dataFile.print(minutes);
dataFile.print(':');
if (seconds < 10) {
dataFile.print('0');
}
dataFile.print(seconds);
dataFile.print('.');
if (centiseconds < 10) {
dataFile.print('0');
}
dataFile.print(centiseconds);
dataFile.print("    ");
dataFile.print(String(temperature, 2)); // print temperature with two
decimal points
dataFile.print(" C    ");
dataFile.print(String(light)); // print light value
dataFile.print("    ");
dataFile.print(sound); // print sound value
dataFile.println();
```

U ovom dijelu koda se podaci zapisuju u datoteku. Vrijeme se zapisuje u formatu sati:minuta:sekunda.stotinka, s dodavanjem vodećih nula ako je potrebno. Nakon toga slijedi razmak i zapisuje se temperatura s dvije decimale i oznakom za stupnjeve Celzijusa. Zatim se zapisuje vrijednost svjetlosne jakosti i jakosti zvuka. Sve vrijednosti se odvajaju razmacima. Nakon što su podaci zapisani, prelazi se u novi redak u datoteci.

```
// Flush the data file to ensure data is written to the SD card
dataFile.flush();
```

Ovaj dio koda osigurava da podaci budu zapisani na SD karticu. Metoda „flush()“ koristi se za osiguranje da svi podaci koji su napisani u privremenu memoriju datoteke budu stvarno zapisani na fizičku SD karticu. To osigurava da nema gubitka podataka i da su sve informacije sigurno spremljene na SD karticu.



```
// Wait for 5 minutes before taking another measurement
delay(300000);
```

Ovaj dio koda čeka 5 minuta prije nego što se izvrši nova mjerenja. Funkcija „delay()“ blokira izvršavanje programa na određeno vrijeme, a vrijednost argumenta izražena je u milisekundama. U ovom slučaju, program će čekati 300000 milisekundi, što je jednakih 5 minuta, prije nego što nastavi s izvršavanjem sljedeće iteracije petlje „loop()“. Ovo osigurava periodičnost mjerenja s vremenskim razmakom od 5 minuta.

## 2.4.2 Uputa za nastavnika

Odgojno obrazovni ishodi su:

OOI2.1 - Razumjeti i primijeniti zakon očuvanja energije:

- Analizirati i objasniti kako se energija očuvana u sustavu svjetlosti, topline i zvuka.
- Primijeniti zakon očuvanja energije u kontekstu mjerenja i promjene svjetlosne jakosti, topline i jakosti zvuka.
- Prepoznati primjere pretvorbe energije između različitih oblika (npr. pretvorba svjetlosti u toplinu, zvuk u električnu energiju).

OOI2.2 - Primijeniti koncepte svjetlosti u različitim kontekstima:

- Istražiti i objasniti kako svjetlost putuje, reflektira se i lomi u različitim medijima.
- Primijeniti znanje o svjetlosti za rješavanje praktičnih problema (npr. oblikovanje optičkih sustava, prepoznavanje oblika i boja).

OOI2.3 - Razumjeti i primijeniti koncepte topline:

- Analizirati termičke pojave i objasniti kako toplina putuje i prenosi se različitim mehanizmima.
- Primijeniti koncepte topline za objašnjavanje i rješavanje problema (npr. termalna ravnoteža, promjena agregatnih stanja).

OOI2.4 - Razumjeti i primijeniti koncepte zvuka:

- Istražiti i objasniti kako zvuk putuje, reflektira se i prenosi kroz različite medije.

- Primijeniti znanje o zvuku za analizu i oblikovanje zvučnih sustava (npr. akustika, audio tehnologija).

Ciljevi:

- Razumjeti važnost mjerenja svjetlosne jakosti, temperature i jakosti zvuka u kontekstu promatranja energije u okolišu.
- Upoznati se s osnovnim konceptima Arduina i njegovom upotrebom u mjerenju i prikupljanju podataka.
- Izraditi senzore za mjerenje svjetlosne jakosti, temperature i jakosti zvuka pomoću Arduino platforme.
- Provoditi eksperimente i analizirati prikupljene podatke u svrhu istraživanja energije u okolišu.
- Razviti istraživačke vještine, kritičko razmišljanje i timski rad.

Nastavni sat 1:

Uvod (5 min):

Predstavite temu i ciljeve nastavne jedinice. Objasnite učenicima važnost mjerenja svjetlosne jakosti, temperature i jakosti zvuka za razumijevanje energije u okolišu.

Teorijska pozadina (10 min):

Uvod u osnovne koncepte svjetlosti, temperature i zvuka te njihovu vezu s energijom. Objasnite kako se Arduino može koristiti za mjerenje i prikupljanje podataka u realnom vremenu.

Demonstracija (10 min):

Prikazivanje Arduino ploče i senzora za svjetlosnu jakost, temperaturu i zvuk. Pokažite kako povezati senzore s Arduinoom i programirati ih za prikupljanje podataka.

Istraživačka pitanja i zadaci (15 min):

Postavite istraživačka pitanja kao poticaj za daljnje istraživanje. Dajte zadatke učenicima za izradu senzora, provođenje eksperimenata i prikupljanje podataka. Neka učenici rade u parovima ili manjim grupama.

Radni listić (5 min):

Podijelite radni listić s učenicima koji će ih voditi kroz istraživanje. Objasnite im zadatke i upute za bilježenje rezultata.

Nastavni sat 2:

Ponavljjanje prethodnog sata (5 min):

Brzi pregled ključnih pojmova, istraživačkih pitanja i postignutih rezultata.

Rad na istraživanju (30 min):

Učenici provode eksperimente i prikupljaju podatke koristeći svoje Arduino senzore. Potaknite ih da bilježe podatke, provode usporedbe i analiziraju rezultate.

Rasprava i dijeljenje rezultata (5 min):

Učenici razmjenjuju rezultate, grafikone i zaključke s ostalim timovima. Potičite ih na kritičko razmišljanje i postavljanje pitanja jedni drugima.

Zaključak (5 min):

Osigurajte vrijeme za zaključivanje i raspravu o postignutim rezultatima. Istaknite ishode koje su učenici ostvarili kroz istraživanje.

Primjer radnog listića za učenike:

Izrada senzora:

Izradite senzor za mjerenje svjetlosne jakosti pomoću Arduino platforme. Izradite senzor za mjerenje temperature i senzor za mjerenje jakosti zvuka.

Eksperimenti:

Promatrajte kako se mjerene veličine mijenjaju u različitim uvjetima (npr. vani, u učionici, pod umjetnim svjetlom). Izmjerite temperaturu, svjetlost i zvuk u različitim dijelovima škole i usporedite rezultate. Dobra ideja je napraviti mjerenje u jednoj učionici ili prostoriji kroz dvadeset i četiri sata, te nakon toga komentirati dobivene vrijednosti.

Analiza podataka:

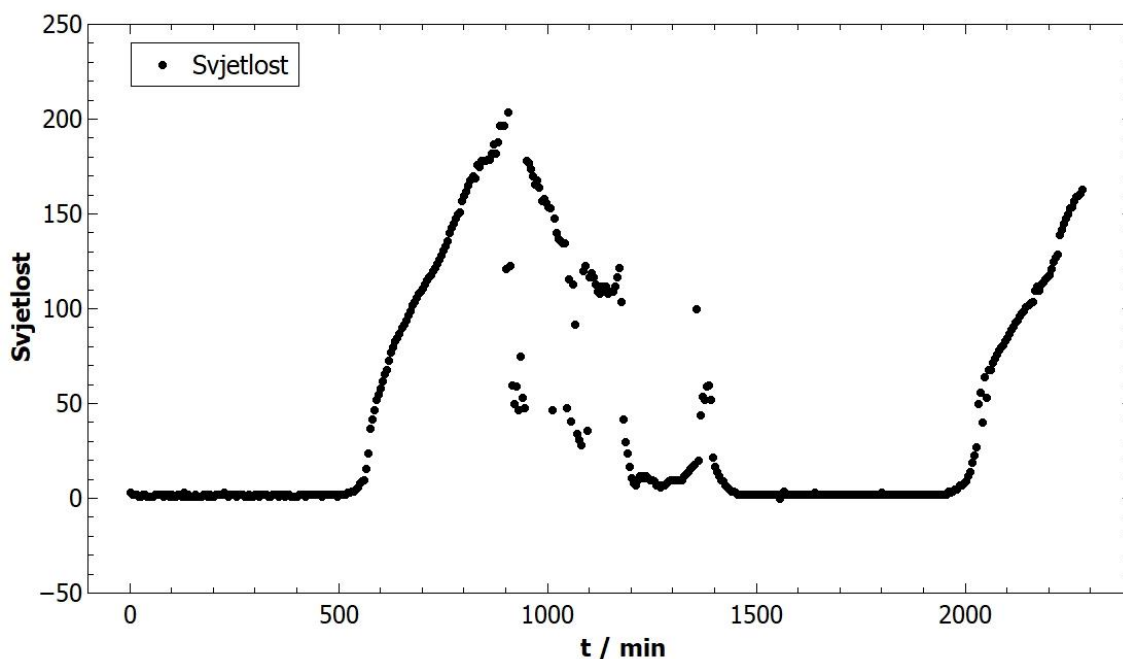
Grafički prikažite rezultate svojih mjerenja. Usporedite podatke iz različitih eksperimenata i izvedite zaključke o kretanju energije u okolišu. Kod prikaza podataka, idealno je prikazati tri mjerene vrijednosti grafom u ovisnosti o vremenu, tako bi učenici najlakše vidjeli promjene mjerenih veličina.

### 2.4.3 Rezultati i diskusija

Mjerenje temperature, jakosti zvuka i svjetlosne jakosti je trajalo 36 sati, ovako izgledaju rezultati za prvih dva sata mjerenja.

Time	Temperature (C)	Light	Sound				
				1:00:00.41	23.42 C	2.00	11
0:00:00.10	23.91 C	3.00	6	1:05:00.42	23.42 C	2.00	7
0:05:00.16	23.91 C	2.00	0	1:10:00.45	23.42 C	2.00	4
0:10:00.18	23.91 C	2.00	8	1:15:00.47	23.42 C	2.00	0
0:15:00.20	23.91 C	2.00	1	1:20:00.49	23.42 C	1.00	2
0:20:00.23	24.40 C	1.00	9	1:25:00.51	23.42 C	2.00	11
0:25:00.25	23.91 C	1.00	0	1:30:00.53	23.42 C	2.00	4
0:30:00.27	23.91 C	2.00	2	1:35:00.55	23.42 C	1.00	0
0:35:00.28	23.91 C	2.00	12	1:40:00.57	23.42 C	2.00	7
0:40:00.30	23.91 C	1.00	1	1:45:00.59	23.42 C	1.00	22
0:45:00.33	23.91 C	1.00	10	1:50:00.60	23.42 C	1.00	0
0:50:00.37	23.91 C	1.00	4	1:55:00.62	23.42 C	2.00	3
0:55:00.39	23.91 C	1.00	15	2:00:00.64	23.42 C	2.00	8

Kako senzori nisu kalibrirani, nema mjerne jedinice za svjetlosnu jakost i zvuk. Ispod je prikazan graf ovisnosti svjetlosne jakosti o vremenu. Sklop je bio ostavljen duže od jednog dana u sobi koja se povremeno koristi par puta dnevno za odlaganje nekih stvari, međutim sklop je bio smješten na prozorsku dasku tako da do njega ne dopiru direktno sunčeve zrake.

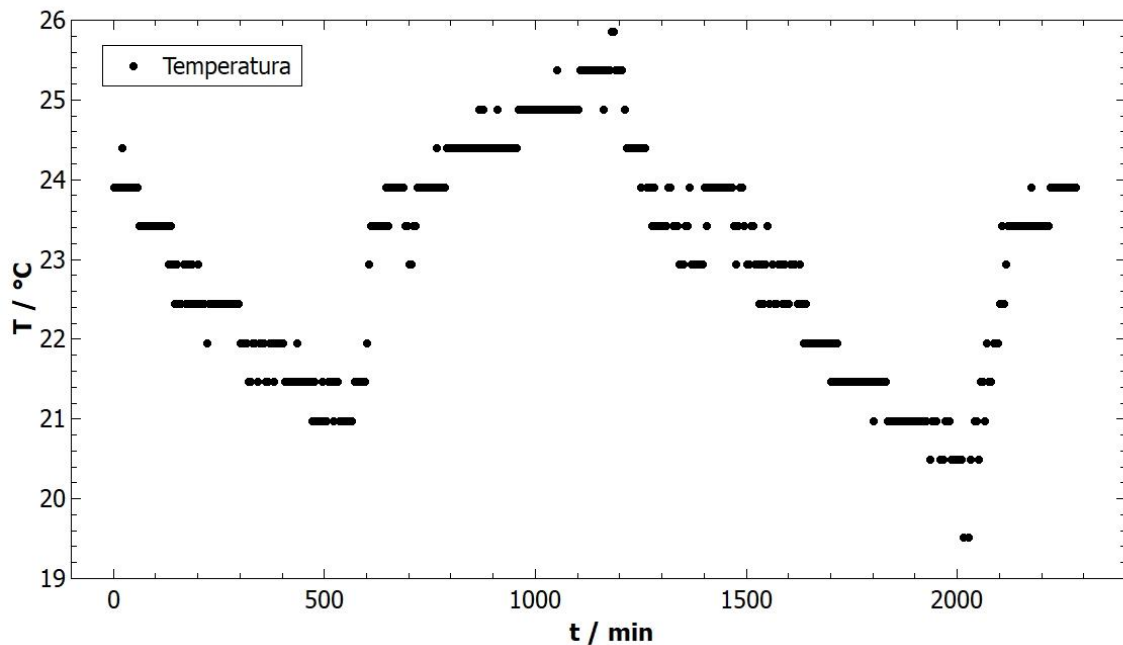


Slika 23 Graf ovisnosti svjetlosne jakosti o vremenu

Na 500 minuta na x osi je približno pet sati ujutro te nakon toga vidimo kako svjetlosna jakost raste. Porast traje sve do vrhunca koji doseže oko podneva i nakon toga vidimo nesrazmjere u

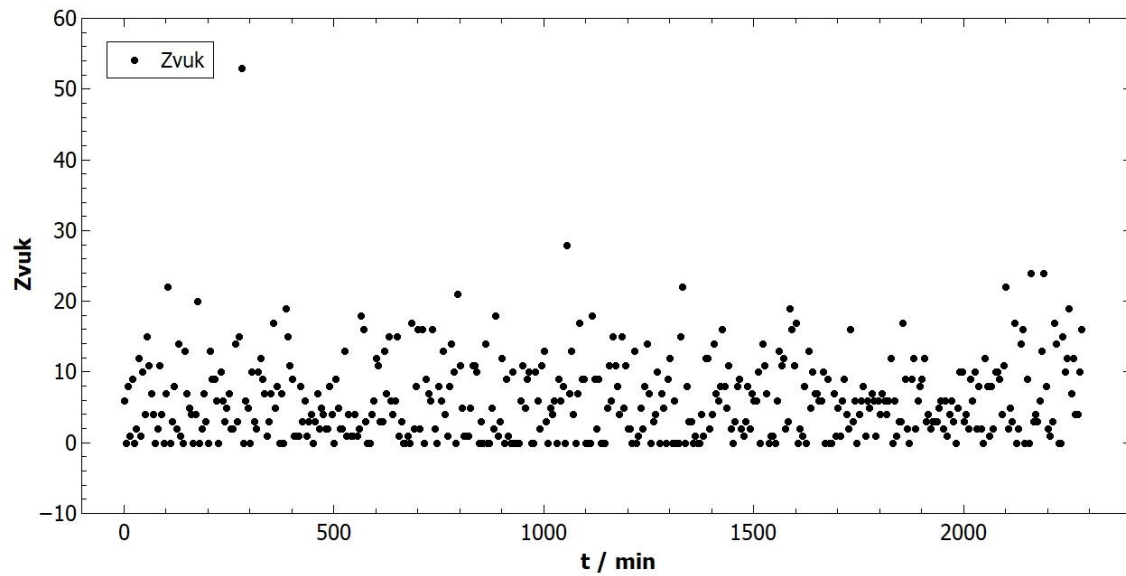
mjerenju koji su rezultat naoblacenja i ulaska i izlaska ukućana iz sobe i u sobu. Postavljeni intervali od pet minuta između svakog vremena dovoljni su kako bi na ovako dug vremenski period mogli dobiti lijepu karakteristiku. Vidimo kako u mraku nemamo potpuno ravnu liniju na grafu, što može biti rezultat paljenja svjetla na hodniku isto kako i svjetlucanje lampice na samom Arduinu kada je pod napajanjem.

Jednako kao i u prvom pokusu, na grafu je vidljiva nepreciznost senzora temperature, to je vidljivo kod rubova nekog „stanja“ kako po jedna točka mjerenja preskakuje jer je na rubu dvije vrijednosti. Međutim, iz grafa vidimo da najniža temperatura nastupa tik prije zore, prije nego što sunce počne dizati istu. Ova zapažanja lako možemo provjeriti na vremenskoj prognozi koja nam iz sata u sat daje podatke na internetu.



Slika 24 Graf ovisnosti temperature o vremenu

Ovdje nam puno više utječu na mjerenje vanjski čimbenici kao što su otvaranje i zatvaranje vrata, kao i paljenje i gašenje svjetla. Makar je senzor neprecizan, lijepo se vidi da je sljedeće jutro bilo hladnije od prvog.



**Slika 25 Graf ovisnosti zvuka o vremenu**

Budući da se moja kuća nalazi pokraj željezničke pruge, možda nije najadekvatnije podneblje za mjerenje zvuka u ovisnosti o vremenu. Može se vidjeti da su točke mjerenja malo zgusnutije za vrijeme noći, no taj podatak nije trivijalan.

## 2.5 Zaključak

Neki senzori dostupni za izradu ovog rada nisu odgovarajuće baždareni i precizni. Na primjer, kod mjerenja temperature vidimo da je senzor LM35 sposoban mjeriti u koracima od gotovo 0.5 stupnjeva Celzija što nije idealno za neka detaljnija mjerenja. Međutim i takvi senzori su prikladni za određene primjene u nastavi i za učenje osnova upravljanja mikroračunalom i konstrukcije mjernog instrumenta. Za naprednije primjene mogu se nabaviti puno precizniji senzori čime Arduino može postati ozbiljan mjerni instrument.

Samo sučelje Arduino IDE programa nije prilagođeno učenicima kao što je slučaj kod Micro:bita, no opet jednostavnije od ostalih alata za programiranje kao što su Visual Studio. Arduino IDE sam prepoznaje da je Arduino priključen na računalo. U samom procesu traženja biblioteka potrebnih za kodiranje, IDE pronalazi biblioteke na internetu ako u tražilicu upišemo prvih par slova biblioteke.

Dostupnost izvornih kodova ili uputa na internetu olakšava učenje i primjenu u različitim fizikalnim problemima. Arduino je jedan od projekata koji mladima može pokazati brojne mogućnosti interakcije digitalnog i realnog svijeta povezujući fiziku, informatiku, robotiku i druga područja. Prikladan je za uporabu u srednjoj školi i za naprednije učenike u osnovnoj školi koji su savladali sve mogućnosti micro:bita.

## 2.6 Literatura

- [1] Senzor temperature LM35 <https://arduinogetstarted.com/tutorials/arduino-lm35-temperature-sensor>
- [2] Senzor svjetlosti TEMENT6000 <https://learn.sparkfun.com/tutorials/temt6000-ambient-light-sensor-hookup-guide/all>
- [3] Senzor jakosti zvuka KS0035 [https://wiki.dfrobot.com/Gravity\\_Water\\_Pressure\\_Sensor\\_SKU\\_SEN0257](https://wiki.dfrobot.com/Gravity_Water_Pressure_Sensor_SKU_SEN0257)
- [4] ALLNET Arduino set <https://www.allnet.de/en/allnet-brand/produkte/arduino-sets/p/allnet-4duino-starter-kit-light-uno-r3/>
- [5] Arduino compatible data logging shield <https://whadda.com/product/arduino-compatible-data-logging-shield-wpsh202/>
- [6] Elisabeth Pratidhina, Dadan Rosana, Heru Kuswanto and Wipsar Sunu Brams Dwandaru Using Arduino and online block-structured programming language for physics practical work Elisabeth Pratidhina, Dadan Rosana, Heru Kuswanto and Wipsar Sunu Brams Dwandaru
- [7] Jelovica L., Erceg N. Application of the Arduino platform in education - determination of the specific heat capacity of water
- [8] Scaradozzi D., Guasti L., Di Stasio M., Miotti B., Moneriù A. Makers at School, Educational Robotics and Innovative Learning Environments
- [9] Arduino platforma i senzor udaljenosti za mjerenje brzine svjetlosti u tekućini A. Svedružić, D. Ptičar
- [10] Arduino Education, <https://www.arduino.cc/education>
- [11] Sara Vidošević (Diplomski, 2022). Klimatske promjene kao tema u nastavi fizike. Fizički odsjek, PMF, Sveučilište u Zagrebu
- [12] Viktorija Milec (Diplomski, 2021). Konstrukcija mjernih instrumenata baziranih na micro:bitu za potrebe nastave fizike. Fizički odsjek, PMF, Sveučilište u Zagrebu
- [13] Filip Kuna (Diplomski, 2022). Konstrukcija vremenskih vrata i primjena u nastavi fizike. Fizički odsjek, PMF, Sveučilište u Zagrebu
- [14] Nastassia Sekulić (Diplomski, 2018). Upotreba micro:bit uređaja u dizajnu demonstracijskih pokusa. Fizički odsjek, PMF, Sveučilište u Zagrebu
- [15] Petar Kalafatić (Diplomski, 2018). Upotreba micro:bit uređaja u dizajnu demonstracijskih pokusa u osnovnoj školi. Fizički odsjek, PMF, Sveučilište u Zagrebu
- [16] Josip Bakula (Diplomski, 2020). Upotreba micro:bit uređaja u prikupljanju fizikalnih podataka. Fizički odsjek, PMF, Sveučilište u Zagrebu
- [17] Tomislav Višnić (Diplomski, 2022). Upotreba micro:bita u razumijevanju valova u fizici. Fizički odsjek, PMF, Sveučilište u Zagrebu