

Міністерство освіти і науки України

Національний технічний університет України „КПІ

імені Ігоря Сікорського ”

Факультет інформатики та обчислювальної техніки

Кафедра інформаційних систем та технологій

**ЗВІТ**

лабораторної роботи №3

курсу «Проектування мікропроцесорних систем на базі Arduino та Raspberry»

Тема: «Програмування мікропроцесорних систем. Робота з аналоговими датчиками»

Перевірив: Виконав:

Доц. Голубєв Л. П. Гр. ІТ-04

Чапча Святослав

Київ 2023

ЗМІСТ

[1. Постановка задачі 3](#_Toc146628348)

[2. Виконання 4](#_Toc146628349)

[3. Контрольні питання 8](#_Toc146628350)

[4. Висновок 11](#_Toc146628351)

[5. Додатки 11](#_Toc146628352)

# Постановка задачі

**Мета:** ознайомити студентів методам перетворення аналогових величин в цифрові значення, які можуть бути проаналізовані мікроконтролером Arduino.

**Завдання до роботи:**

**В кожній з робіт потрібно розробити схеми та заставити її працювати за правилами, що викладені в задачах 1-4. До кожної із задач у зошиті повинні бути намальовані відповідні схеми! Всюди 93 – номер студента за списком.**

**Завдання 1. Робота з потенціометром**.

Створити проект (електронну схему і написати програмний код) в якому при повороті ручки потенціометра, якщо його значення знаходиться в діапазоні від n\*20 до n\*40 загорявся б червоний світлодіод. Значення потенціометра має постійно видаватися в монітор послідовного порту. При введенні в моніторі послідовного порту числа n повинен включатися режим миготіння світлодіоду в цьому діапазоні.

**Завдання 2. Робота з датчиком температури.**

Створити проект (електронну схему і написати програмний код) електронного термометра, який видавав би значення температури в град. Цельсія і Фарінгейта в монітор послідовного порту через проміжок часу рівний n\*50 (задавши його за допомогою функції millis ()).

**Завдання 3. Робота з фоторезистором.**

Створити проект (електронну схему та написати програмний код) світильника, що складається з 3-х білих світлодіодів. Він повинен працювати у двох режимах: 1- ступеневий та 2 -плавний (режим задається у моніторі послідовного порту).

1. Світильник працює на трьох рівнях (low=(350-n\*2), middle, high=(700+n\*4)). Якщо освітленість менше low - горять усі 3 світлодіоди, якщо від low до middle - 2, якщо від middle до high – 1, якщо більше high – світлодіоди вимкнено.

2. Світильник працює плавно: при зменшенні освітленості збільшується яскравість усіх світлодіодів, а при збільшенні - зменшується (за допомогою ШІМ).

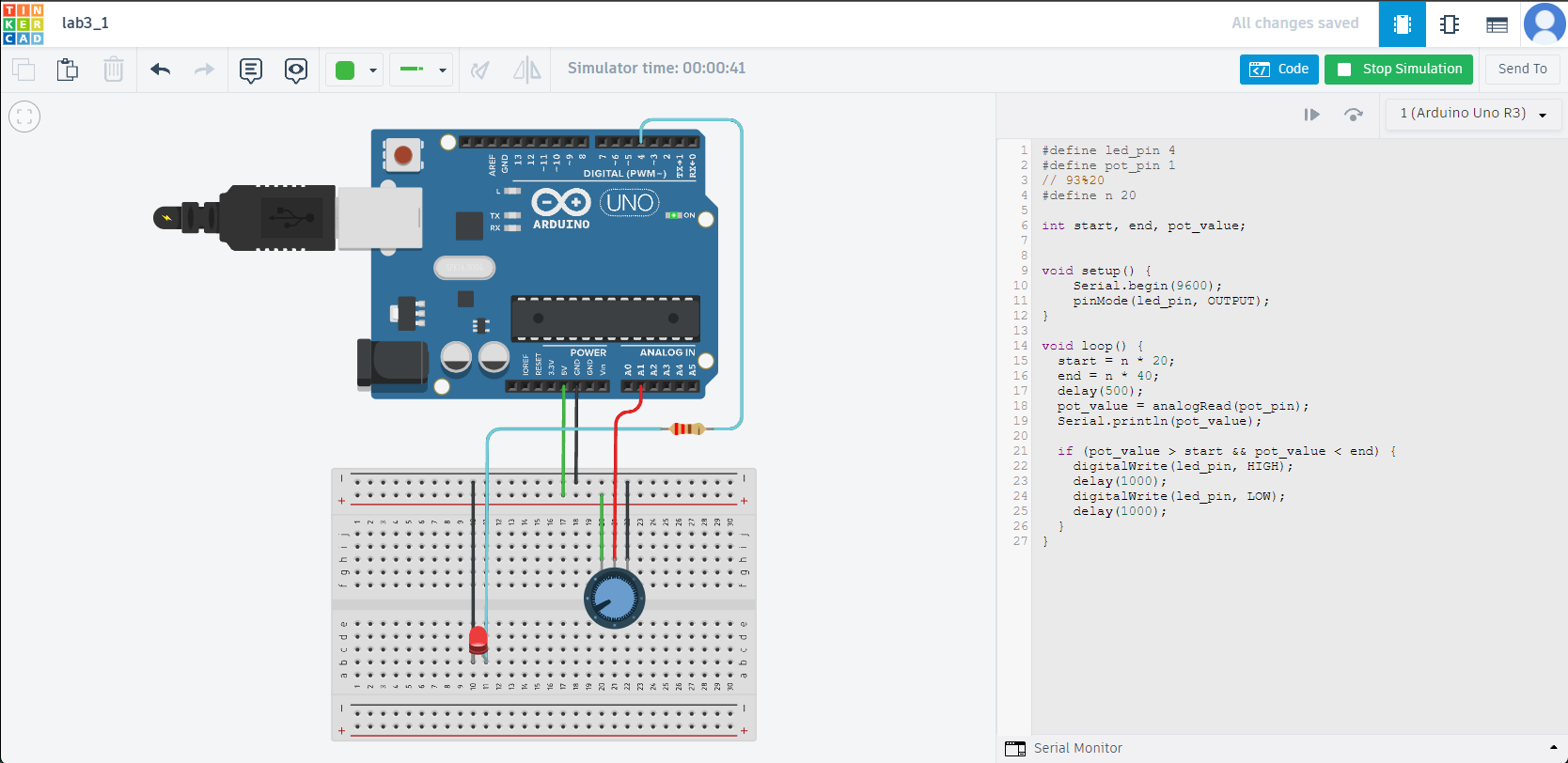
Використайте функції **map()**, та **constrain().**

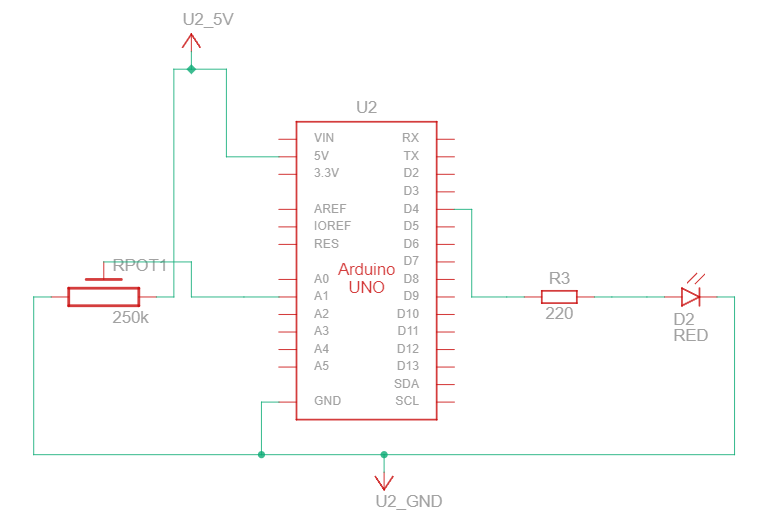
Кнопка керує світильником (вкл./викл.).

Інформація про освітленість видається у моніторі послідовного порту.

# Виконання

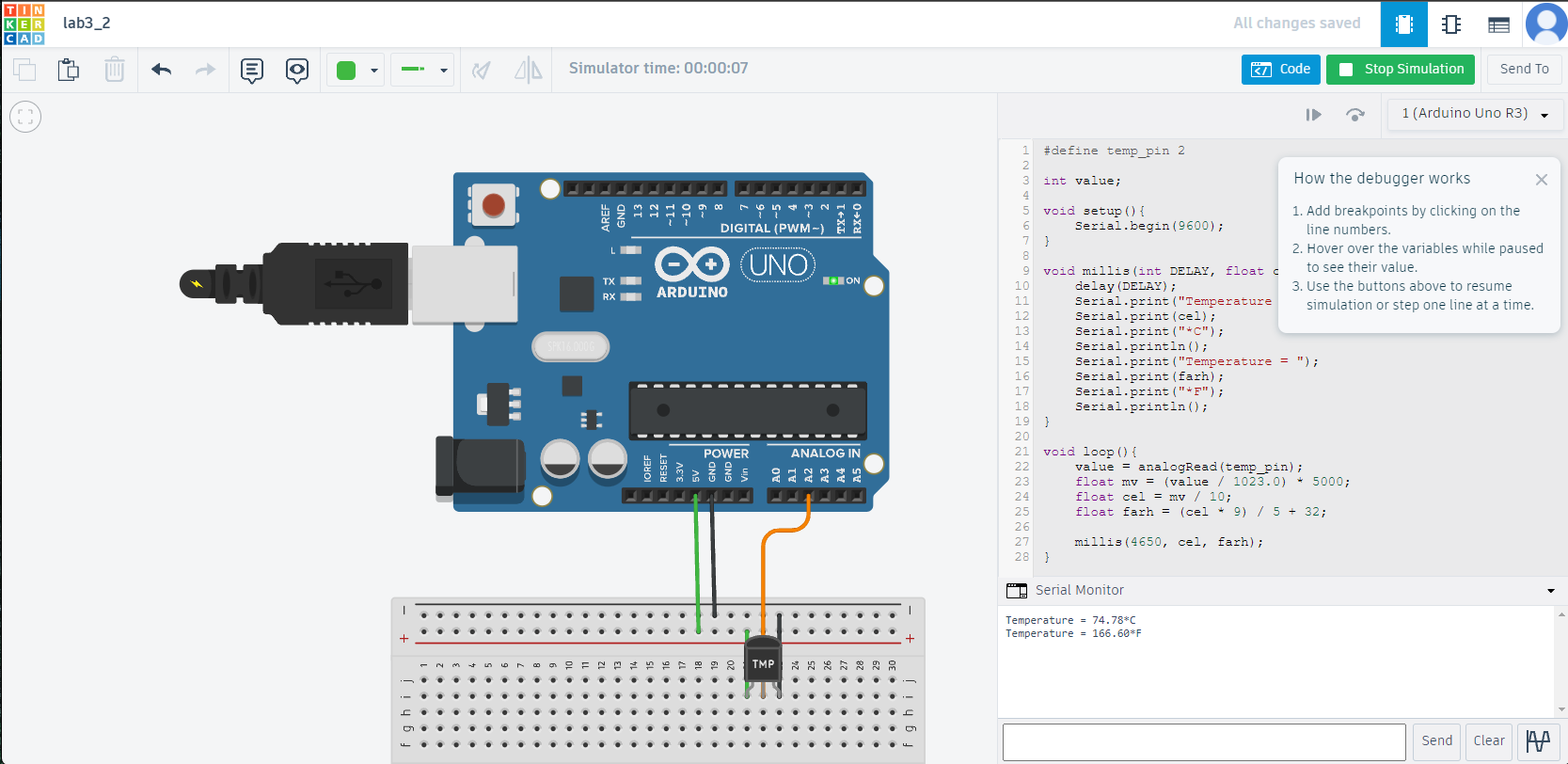
Створимо перший проект «Potens», суть якого полягає в розробці коду в якому при повороті ручки потенціометра, якщо його значення знаходиться в діапазоні загорявся б червоний світлодіод, код та знімки екрану наведені нижче:

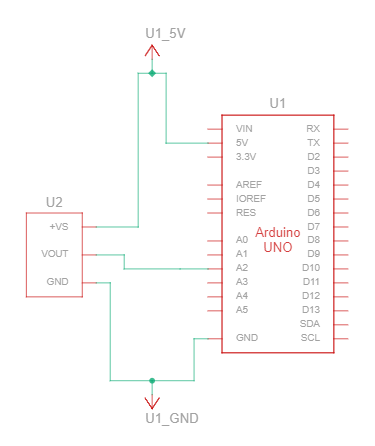




Як бачимо, усе працює, отже можна переходити до створення другого проекту «Temp».

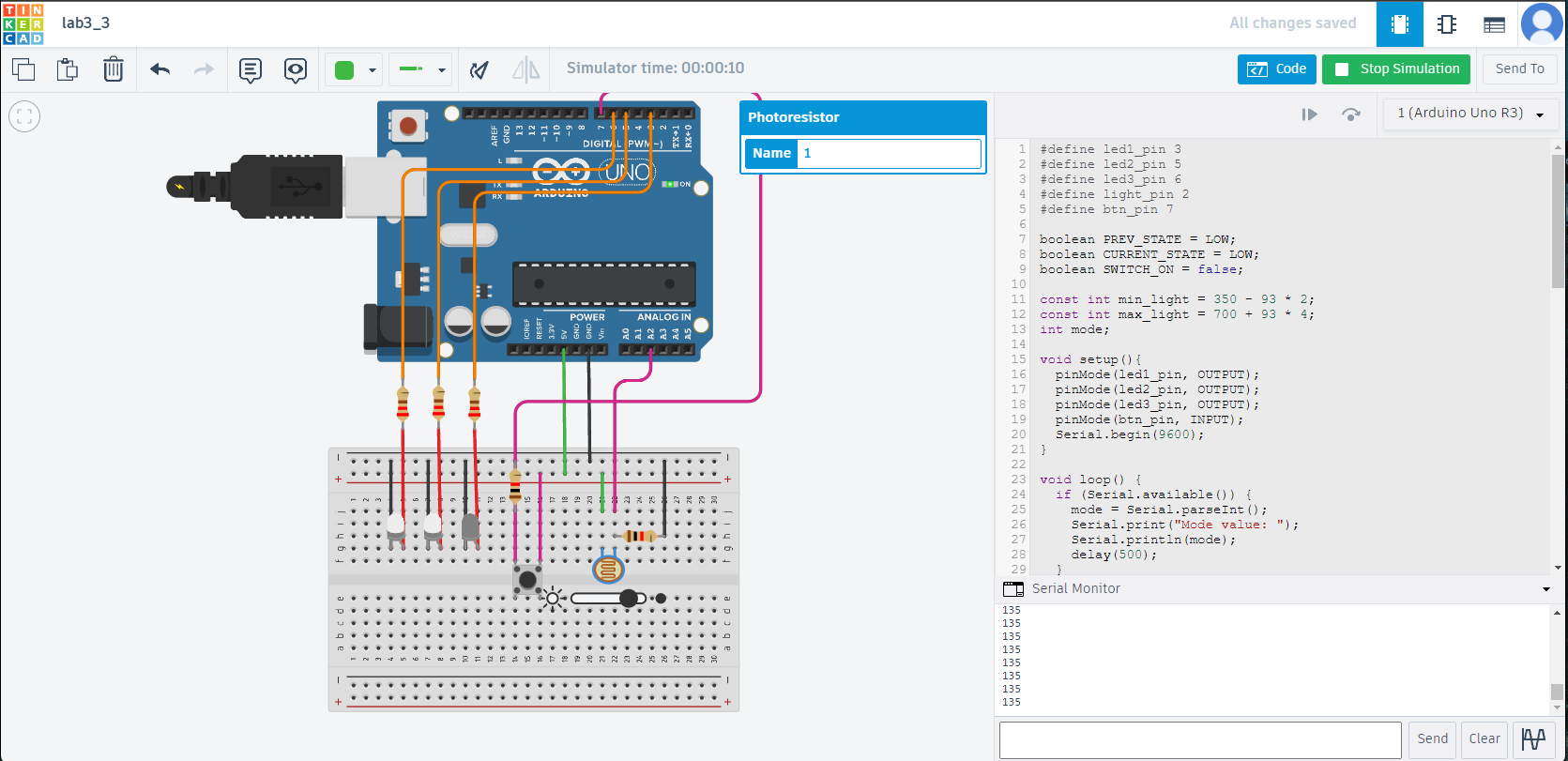
Для виконання потрібно розробити проект електронного термометра, який видавав би значення температури в град. Цельсія і Фарінгейта в монітор послідовного порту через проміжок часу:

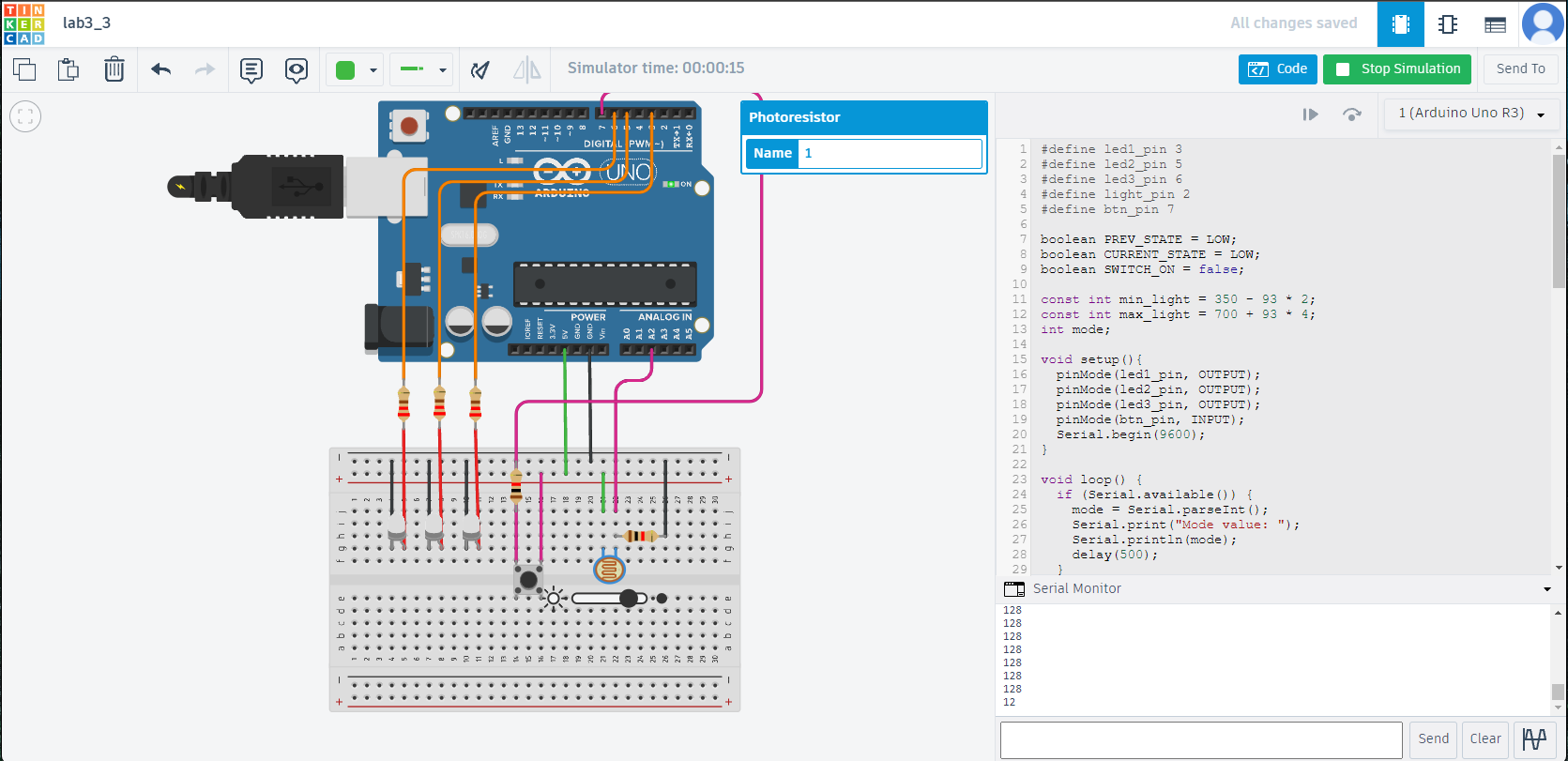


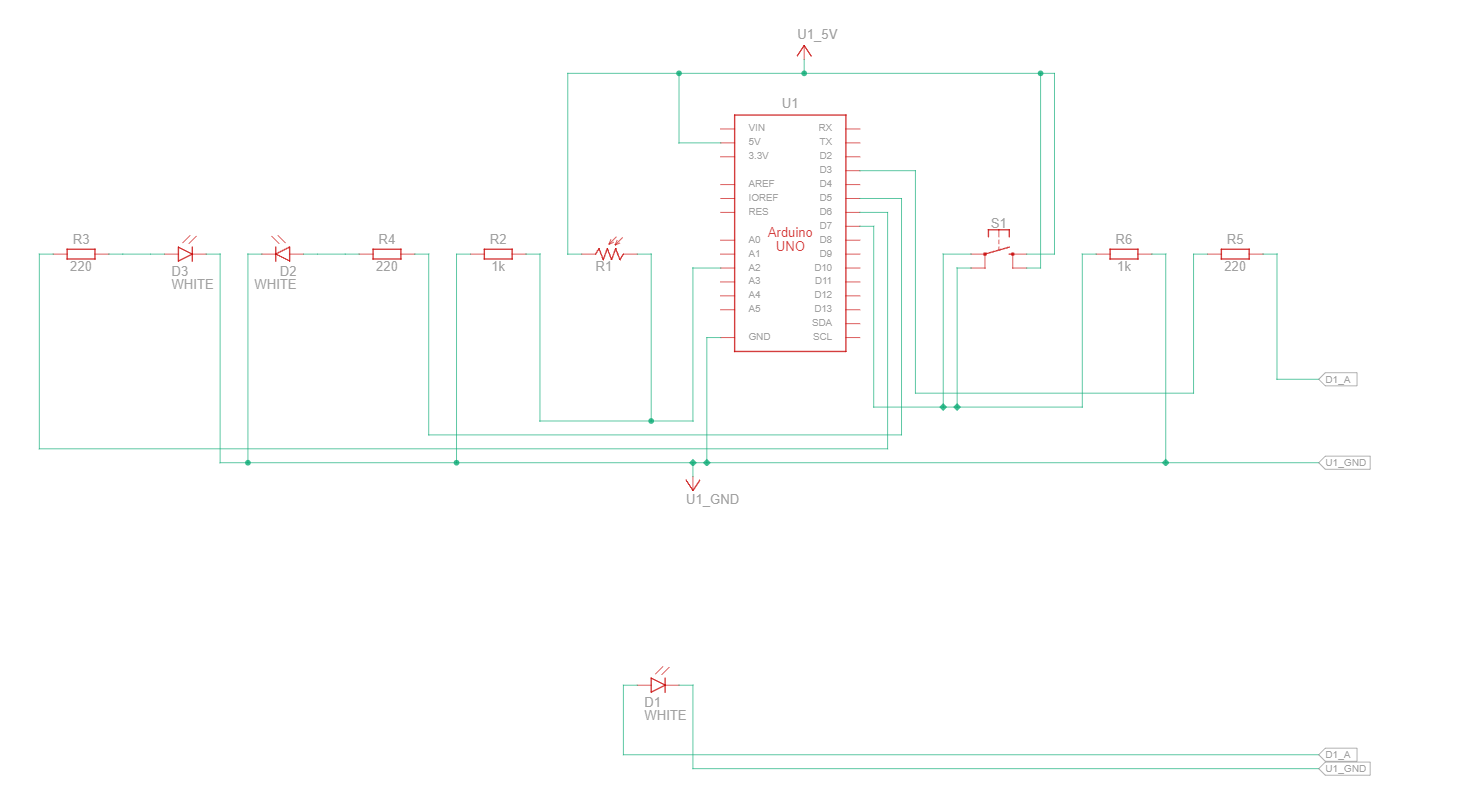


Як бачимо, усе працює, отже можна переходити до створення третього проекту «Photores».

Суть полягає у створенні проекту світильника, що складається з 3-х білих світлодіодів. Він повинен працювати у двох режимах:







# Контрольні питання

1. Чим відрізняються аналогові та цифрові сигнали?

Аналогові сигнали представляють собою неперервні значення, які можуть приймати будь-які значення у певному діапазоні (наприклад, напруга від 0 до 5 В). Цифрові сигнали, натомість, можуть приймати тільки два стани - HIGH (високий) або LOW (низький), що відповідає 1 і 0 відповідно.

1. Що означає термін «оцифровування» аналогового сигналу?

Оцифровування аналогового сигналу - це процес перетворення неперервного аналогового сигналу на цифровий сигнал, який складається з обмеженого набору дискретних значень.

1. З якими числами «спілкуються» аналогові входи Arduino?

Аналогові входи Arduino спілкуються з аналоговими сигналами, які представлені як числа від 0 до 1023. Arduino використовує 10-бітний аналого-цифровий перетворювач (ADC), тому може розпізнавати 1024 різних рівнів напруги.

1. Чи є в Arduino аналогові виходи? Як імітується в Arduino аналоговий вихід?

Так, в Arduino є аналогові виходи. Аналоговий вихід імітує аналоговий сигнал шляхом встановлення напруги на певному піні, що може змінюватися від 0 до 5 В. Імітація аналогового виходу в Arduino виконується за допомогою функції analogWrite(), яка встановлює широтно-імпульсну модуляцію (ШІМ).

1. Що таке потенціометр? Як підключити потенціометр в Arduino ?

Потенціометр - це регульований резистор, який можна використовувати для зміни опору і, таким чином, напруги. Для підключення потенціометра до Arduino, один з кінців підключається до напруги (5 В або 3.3 В), інший кінець - до землі (GND), а середній вивід (середня точка) - до аналогового входу.

1. Як програма на Arduino спілкується з комп’ютером через послідовний порт?

Для спілкування з комп'ютером через послідовний порт, Arduino використовує протокол UART (Універсальний Асинхронний Приймач-Передавач). Він передає дані у вигляді байтів через RX і TX піни на платі Arduino.

1. Для чого потрібна функція Serial.begin()? Що за аргумент вона використовує?

Функція Serial.begin() встановлює швидкість обміну даними (бод) для послідовного порту. Аргумент цієї функції вказує на бажану швидкість обміну даними. Наприклад, Serial.begin(9600) встановить швидкість 9600 бод.

1. Що таке бод?

"Бод" (біти на секунду) - це одиниця вимірювання швидкості обміну даними через послідовний порт.

1. Для чого потрібна функція Serial.print()?Що за аргумент вона використовує?

Функція Serial.print() використовується для виведення даних в послідовний порт. Вона може приймати аргументи різних типів даних і виводити їх у текстовому форматі.

1. Яка різниця у функціях Serial.print() та Serial.println()?

Різниця між Serial.print() і Serial.println() полягає в тому, що Serial.print() виводить дані без переведення курсора на новий рядок, тоді як Serial.println() додає символ нового рядка після виводу даних, що робить вивід наступним рядком.

1. Що таке резистивний дільник напруги?

Резистивний дільник напруги - це схема, що використовує два резистори для поділу напруги. Вона використовується для зменшення напруги із збереженням пропорційності.

1. Як використовується резистивний дільник напруги при роботі із фоторезистором?

При роботі з фоторезистором, резистивний дільник напруги може використовуватися для конвертації змінного опору фоторезистора в змінну напругу, яку можна зчитувати аналоговим входом Arduino.

1. Навіщо потрібна функція map()? Як вона працює?

Функція map() використовується для перетворення значення з одного діапазону в інший. Вона має чотири аргументи: вхідне значення, початок діапазону вхідного значення, кінець діапазону вхідного значення і новий діапазон вихідного значення.

1. Навіщо потрібна функція constrain ()? Як вона працює?

Функція constrain() використовується для обмеження значення змінної до певного діапазону. Вона має три аргументи: значення, мінімальне обмеження і максимальне обмеження, і повертає значення, яке знаходиться між мінімальним і максимальним обмеженнями.

# Висновок

У даній лабораторній роботі я познайомився з методами перетворення аналогових величин в цифрові значення, які можуть бути проаналізовані мікроконтролером Arduino. Під час ознайомлення було розроблено три проекти. Усі результати наведені на знімках екрану вище, код програми та посилання на сам проект

# Додатки

Проект Potens:

<https://www.tinkercad.com/things/j8G7jLspcvc?sharecode=BDsNlw1IoaA0iLPEYNE5tihtwX1blz4t7GOjcresb38>

Проект Temp:

<https://www.tinkercad.com/things/g9xl3wBT6AC?sharecode=kpyst9WdOo_pC4SlFIf_zZOG5j7EUleK5lLAgj28wEA>

Проект Photores:

<https://www.tinkercad.com/things/fIGxlnohgI5?sharecode=gzZG7L9G-r_nvouvgwQZMStxAXDSMezXWr2k-3-6_Jg>

Potens.cpp

#define led\_pin 4

#define pot\_pin 1

// 93%20

#define n 20

int start, end, pot\_value;

void setup() {

  Serial.begin(9600);

    pinMode(led\_pin, OUTPUT);

}

void loop() {

  start = n \* 20;

  end = n \* 40;

  delay(500);

  pot\_value = analogRead(pot\_pin);

  Serial.println(pot\_value);

  if (pot\_value > start && pot\_value < end) {

    digitalWrite(led\_pin, HIGH);

    delay(1000);

    digitalWrite(led\_pin, LOW);

    delay(1000);

  }

}

Temp.cpp

#define temp\_pin 2

int value;

void setup(){

    Serial.begin(9600);

}

void millis(int DELAY, float cel, float farh){

    delay(DELAY);

    Serial.print("Temperature = ");

    Serial.print(cel);

    Serial.print("\*C");

    Serial.println();

    Serial.print("Temperature = ");

    Serial.print(farh);

    Serial.print("\*F");

    Serial.println();

}

void loop(){

    value = analogRead(temp\_pin);

    float mv = (value / 1023.0) \* 5000;

    float cel = mv / 10;

    float farh = (cel \* 9) / 5 + 32;

    millis(4650, cel, farh);

}

Photores.cpp

#define led 3

const int MIN\_VALUE = 93;

const int MAX\_VALUE = 255 - 93;

const int DELAY = 10;

void setup()

{

  Serial.begin(9600);

  pinMode(led, OUTPUT);

}

void loop()

{

  for (int i = MIN\_VALUE; i < MAX\_VALUE; i++) {

    analogWrite(led, i);

    Serial.println(i);

    delay(DELAY);

  }

  for (int i = MAX\_VALUE; i >= MIN\_VALUE; i--) {

    analogWrite(led, i);

    Serial.println(i);

    delay(DELAY);

  }

}

Multibtn.cpp

#define led1\_pin 3

#define led2\_pin 5

#define led3\_pin 6

#define light\_pin 2

#define btn\_pin 7

boolean PREV\_STATE = LOW;

boolean CURRENT\_STATE = LOW;

boolean SWITCH\_ON = false;

const int min\_light = 350 - 93 \* 2;

const int max\_light = 700 + 93 \* 4;

int mode;

void setup(){

  pinMode(led1\_pin, OUTPUT);

  pinMode(led2\_pin, OUTPUT);

  pinMode(led3\_pin, OUTPUT);

  pinMode(btn\_pin, INPUT);

  Serial.begin(9600);

}

void loop() {

  if (Serial.available()) {

    mode = Serial.parseInt();

    Serial.print("Mode value: ");

    Serial.println(mode);

    delay(500);

  }

  CURRENT\_STATE = debounce(PREV\_STATE);

  if (PREV\_STATE == LOW && CURRENT\_STATE == HIGH) {

    SWITCH\_ON = !SWITCH\_ON;

    Serial.print("Access mode: ");

    Serial.println(SWITCH\_ON);

  }

  PREV\_STATE = CURRENT\_STATE;

  if (SWITCH\_ON) {

    int value = analogRead(light\_pin);

    value = map(value, 6, 679, 255, 0);

    value = constrain(value, 0, 255);

    int max = map(min\_light, 200, 900, 255, 0);

    max = constrain(max, 0, 255);

    int min = map(max\_light, 200, 900, 255, 0);

    min = constrain(min, 0, 255);

    int middle = (max + min) / 2;

    Serial.println(value);

    if (mode == 1) {

      if (value >= max) {

        digitalWrite(led1\_pin, HIGH);

        digitalWrite(led2\_pin, HIGH);

        digitalWrite(led3\_pin, HIGH);

      }

      if (value >= middle && value < max) {

        digitalWrite(led1\_pin, LOW);

        digitalWrite(led2\_pin, HIGH);

        digitalWrite(led3\_pin, HIGH);

      }

      if (value > min && value < middle) {

        digitalWrite(led1\_pin, LOW);

        digitalWrite(led2\_pin, LOW);

        digitalWrite(led3\_pin, HIGH);

      }

      if (value <= min) {

        digitalWrite(led1\_pin, LOW);

        digitalWrite(led2\_pin, LOW);

        digitalWrite(led3\_pin, LOW);

      }

    }

    if (mode == 2) {

      analogWrite(led1\_pin, value);

      analogWrite(led2\_pin, value);

      analogWrite(led3\_pin, value);

    }

  }

}

boolean debounce(boolean previous){

  boolean current = digitalRead(btn\_pin);

  if (previous != current) delay(3);

  current = digitalRead(btn\_pin);

  return current;

}