Дніпровський ліцей інформаційних технологій

при Дніпровському національному університеті

імені Олеся Гончара

Курсова робота

на тему:

**Світлодіодна динамічна анімація** Виконавець:

ліцеїст 10-Г класу

Устінов Іван

Науковий керівник:

Чашка Ю.М.\_\_\_\_\_\_\_\_

Зміст

[Вступ 3](#_Toc37880499)

[1 Тема роботи: 3](#_Toc37880500)

[2 Мета роботи: 3](#_Toc37880501)

[Основна частина 4](#_Toc37880502)

[1 Теоретична частина з фізики: 4](#_Toc37880503)

[2 Теоретична частина з інформатики 5](#_Toc37880504)

[3 Фрагмент програмного коду 5](#_Toc37880505)

[3.1 Розрахунок періоду переривання таймера та кадру анімації 5](#_Toc37880506)

[3.2 Зчитування кольорів та вмикання світлодіодів. 6](#_Toc37880507)

[4 Опис роботи 6](#_Toc37880508)

[5 Посібник користувача 6](#_Toc37880509)

[6 Напрямки, де можна використовувати роботу: 7](#_Toc37880510)

[7 Комплектація програми 7](#_Toc37880511)

[8 Використані програмні засоби 7](#_Toc37880512)

[Висновки 7](#_Toc37880513)

[Додатки 9](#_Toc37880514)

[1 Додаток №1 9](#_Toc37880515)

[2 Додаток №2 10](#_Toc37880516)

# Вступ

## Тема роботи:

Світлодіодна динамічна ілюмінація.

## Мета роботи:

Поглиблене вивчення двовимірних масивів кольорів, розрахунок положення стрічки в певний момент часу.

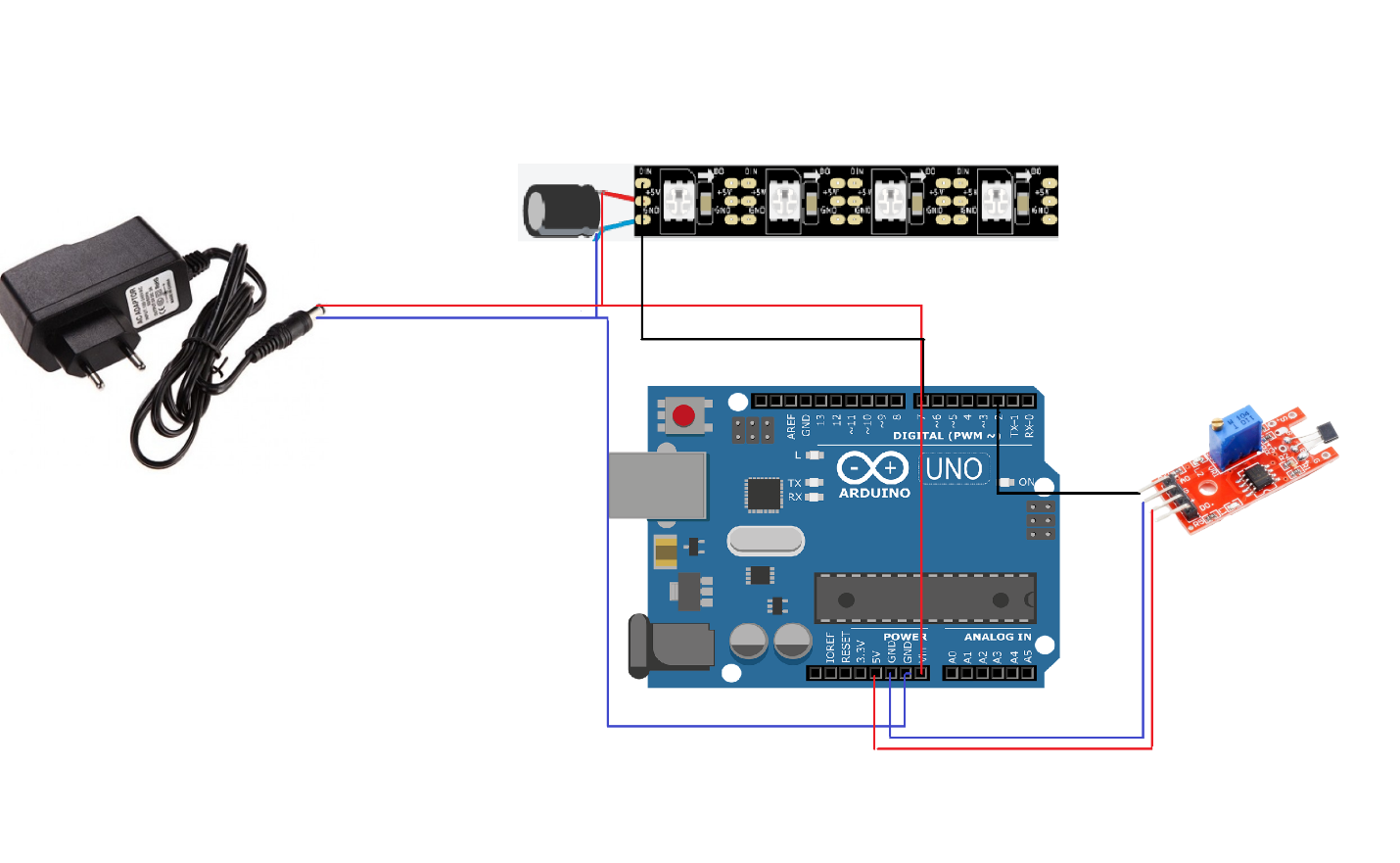
# Основна частина

## Теоретична частина з фізики:

Метою моєї роботи була розробка струнного дисплею. Він являє собою стрічку світлодіодів, що обертається. Виведення зображення відбувається за рахунок швидкого перемикання світлодіодів на адресній RGB світлодіодній стрічці, а саме WS2812b. Зображення будується по аналогії з телевізійним кадром. Відмінність в тому, що використовується не декартова система координат, а полярна. У нашому випадку, коли стрічка розташовується вздовж діаметру, симетрично відносно центру кола, кадр складається з кута обертання двигуна на 180°. Мітка початку кадру формується як сигнал від датчика Холла. Датчик Холла кріпиться з правого боку від центру обертання стрічки, на діаметрі міститься магніт. У нашому випадку сигнал синхронізації формується один раз на 360° обертання світлодіодної стрічки, тобто один раз на два кадри.

Дана стрічка має 3 контакти: +5V, ground та логічний цифровий вхід. Для передачі живлення та керуючого сигналу на стрічку, яка обертається я використав аудіо-штекер 6.3мм, який має 3 контакти.

Ефект зображення виникає за рахунок вмикання певних світлодіодів стрічки при відхиленні стрічки на певний градус від датчика. Розберемо стрічку WS2812b: відправка сигналу на 1 діод займає 30 мкс, на 72 – 2160мкс. Отже оновлювати стрічку можна кожні 2160 мкс. Для оновлення одного оберту кожний градус необхідно 2160мкс×360≈778000мкс.Отже максимальна частота обертання оберти в секунду, а частота оновлення зображення 2.6 Гц, оскільки за 1 оберт кадр оновлюється 2 рази. При такій швидкості ефекту цільного зображення не буде, тому я вирішив оновлювати стрічку кожні 10 градусів, тобто максимальна можлива швидкість = 13 обертів/с.

Для визначення положення пластини зі стрічкою я використав датчик магнітного поля (Датчик Холла).

## Теоретична частина з інформатики

Для розкладання зображення на масив кольорів я використав програму «Pixel.exe». Дані про кольори світлодіодів зберігаються в бібліотеках. Для зберігання великого обсягу даних, які не змінюються, використовуємо програмну, а не оперативну пам’ять, тобто PROGMEM.

Програма працює на двох перериваннях: магнітного датчика і таймера. При підвищенні сигналу з магнітного датчика, викликається функція, яка розраховує період обертання стрічки і період зміщення стрічки на 10 градусів. При перериванні таймера, програма отримує значення кольорів для світлодіодів і відправляє на стрічку.

## Фрагмент програмного коду

### Розрахунок періоду переривання таймера та кадру анімації

[Додаток №1](#_Додаток_№1)

При перериванні датчика виконується функція hall(). Для фільтрації помилкових даних використовуємо if. Період оберту = час з моменту запуска – час з моменту запуска при попередньому проходженні магнітом датчика. Для згладжування зображення використовуємо коефіцієнти k і b.

### Зчитування кольорів та вмикання світлодіодів.

[Додаток №2](#_Додаток_№2)

При перериванні таймера викликається функція **Set\_led.** Num1 та Num2­ – показники положення першої та другої половини стрічки відповідно. У циклах функції Set\_led викликається функція **Led,** у якій зчитуються дані про колір світлодіода і передаються на стрічку.

## Опис роботи

Після запуску програми необхідно обрати один з декількох представлених малюнків або додати свій. Після цього необхідно загрузити скетч на плату Arduino. Програма почне свою роботу після вмикання плати.

Для живлення стрічки та плати необхідно 5В, для живлення мотора – 12В.

## Посібник користувача

Для додавання своєї картинки користувач використовує програму Pixel.Exe.Параметри картинки:

1)У форматі bmp

2)Розмір – 250\*250 рх

У програмі необхідно задати:

1. Кількість сегментів – кількість світлодіодів на половині стрічки (36).
2. Скільки градусів – через скільки градусів оновлювати стрічку(10)
3. Розмір сегменту – не має значення.
4. Кількість відрахунків в градусі – необхідно задати 1.
5. Початковий кут: 0

Зберігати файл необхідно в папку з програмою для Arduino. У скетчі необхідно підключити нову бібліотеку командою **#include "назва файлу.h"**

## Напрямки, де можна використовувати роботу:

Голографічні дисплеї набирають популярність у рекламній індустрії, оскільки дають можливість виводити будь-яке зображення, а в виключеному стані прозорі. Це дозволяє закривати зображенням вітрини, а в робочий час вітрини будуть відкриті.

Також такий дисплей використовується в театрах для демонстрації певної інформації, а під час вистави не заважає глядачам бачити сцену.

## Комплектація програми

* Програма led\_display
* Програма Pixel.exe
* Бібліотеки для роботи з таймером і стрічкою.
* Картинки
* Бібліотеки, в яких зберігаються дані про кольори стрічки.

## Використані програмні засоби

Arduino.Ide

Pixel.exe

# Висновки

В курсовій роботі було зроблено струнний дисплей. Програма Pixel.exe дозволяє кодувати графічні зображення в масив кольорів світлодіодів стрічки відповідно до куту відхилення. Проведено розрахунки які забезпечують формування зображення потрібної якості. Максимальна частота оновлення 26Hz. Це обумовлено характеристиками стрічки, а саме часом відправки сигналу на світлодіоди. При використанні світлодіодів з кращими показниками можливе проектування відео.

Також чіткість зображення залежить від кількості світлодіодів на стрічці. Мій дисплей має 72 світлодіоди, а подібні магазинні аналоги – 320-350. Це дозволяє проектувати чітке зображення, подібне до зображення на нерухомих дисплеях.

# Додатки

## Додаток №1

attachInterrupt(0, hall, RISING); //переривання датчика

void hall()

{

if (micros() - timer > 25000) //фільтруємо випадкові сигнали

{

new\_period = micros() - timer; //розрахунок періоду обертання

timer = micros();

hall\_bool = true;

}

}

void loop() {

if (hall\_bool)

{

hall\_bool = false;

num1 = 0 + start\_pos; //розрахунок стартової позиції першої

num2 = n\_segm / 2 + start\_pos; //та другої половини стрічки

if (num2 > n\_segm) num2 = num2 - n\_segm;

count\_frame++; //розраховуємо кількість проходжень стрічкою положення рівноваги для зміни кадру через N проходжень

if (count\_frame >= number\_frame) {

count\_frame = 0;

this\_frame = (this\_frame + 1) % NUM\_FRAMES; //Вибір кадру слайдшоу

}

period = (float)new\_period \* k + (float)period \* ((float)1 - k); //період повного оберту, коефіцієнт використовуємо для згладжування при зміні швидкості обертання

Timer1.setPeriod(period / n\_segm \* b); //задаємо період переривання таймера (кожні N градусів).

Timer1.restart();

}

}

## Додаток №2

void timer\_interrupt()

{

timer\_bool = true;

}

void set\_led() {

strip.clear();

byte this\_pix = 0;

for (int i = leds\_numb / 2 - 1; i >= 0; i--) //Працюємо з першою половиною стрічки (0-35 світлодіод). Стрічку оновлюємо від центру до краю.

{

led(i, num1, this\_pix);//викликається функція, у якій встановлюються кольори світлодіодів. i-номер світлодіода; num1 – номер строки масиву, тобто номер сегменту, у якому знаходиться дана половина стрічки; this\_pix-номер елементу строки масиву кольорів, тобто номер світлодіода рахуючи від центру обертання.

this\_pix++;

}

this\_pix = 0;

for (int i = leds\_numb / 2 ; i < leds\_numb; i++) //робота з другою половиною стрічки (36-71 світлодіод). Оновлення стрічки відбувається аналогічно до першої половини.

{

led(i, num2, this\_pix);

this\_pix++;

}

num1++;

num2++;

if (num2 > n\_segm) num2 = 0; //При досяганні певною половиною стрічки вертикального положення, номер сегменту = 0;

if (num1 > n\_segm) num1 = 0;

strip.show();

}

void led(byte i, int counter, int this\_pix) {

switch (this\_frame) {

case 0: strip.setPixelColor(i,pgm\_read\_dword(&(frame1[counter][this\_pix]))); // з бібліотеки з масивами кольорів зчитуємо необхідні значення кольорів та відправляємо на стрічку. Frame 1, frame 2, frame3 – назви кадрів анімації.

break;

case 1: strip.setPixelColor(i,pgm\_read\_dword(&(frame2[counter][this\_pix])));

break;

case 2: strip.setPixelColor(i,pgm\_read\_dword(&(frame3[counter][this\_pix])));

break;

}

}

void loop() {

if (timer\_bool)

{

timer\_bool = false;

set\_led();

}

}