

# Spis treści

<b>1</b>	<b>Wstęp</b>	<b>3</b>
1.1	Uzasadnienie wyboru tematu . . . . .	3
1.2	Problematyka i zakres pracy . . . . .	4
1.3	Cele pracy . . . . .	4
1.4	Metoda badawcza . . . . .	4
1.5	Przegląd literatury w dziedzinie . . . . .	4
1.6	Układ pracy . . . . .	4
<b>2</b>	<b>Zagadnienia teoretyczne</b>	<b>5</b>
<b>3</b>	<b>Analiza istniejących rozwiązań</b>	<b>6</b>
3.1	Kryteria analizy . . . . .	6
3.2	Porównanie istniejących rozwiązań . . . . .	6
<b>4</b>	<b>System oświetlenia Lightning</b>	<b>7</b>
4.1	Komponenty systemu . . . . .	7
4.2	Moduły oprogramowania . . . . .	8
4.2.1	Oprogramowanie przeznaczone na mikrokontroler . . . . .	8
4.2.2	Aplikacja przeznaczona na komputer . . . . .	8
4.3	Analiza wymagań . . . . .	8
4.3.1	Wymagania funkcjonalne . . . . .	8
4.3.2	Wymagania нефункционалне . . . . .	9
4.4	Projekt . . . . .	9
4.4.1	System kontroli wersji . . . . .	9
4.4.2	Środowisko programistyczne . . . . .	10
4.4.3	Diagram klas . . . . .	10
4.4.4	Wzorce projektowe . . . . .	10
4.4.5	Interfejs użytkownika . . . . .	10
4.5	Implementacja . . . . .	10

<i>SPIS TREŚCI</i>	2
4.6 Podręcznik użytkownika . . . . .	10
4.7 Przykładowa implementacja animacji . . . . .	10
4.8 Analiza projektu . . . . .	10
4.9 Perspektywy rozwoju projektu . . . . .	10
<b>5 Podsumowanie</b>	<b>12</b>
5.1 Dyskusja wyników . . . . .	12
5.2 Perspektywy rozwoju pracy . . . . .	12
<b>Bibliografia</b>	<b>12</b>
<b>Spis rysunków</b>	<b>13</b>
<b>Spis tabel</b>	<b>14</b>
<b>Spis listingów</b>	<b>15</b>

# Rozdział 1

## Wstęp

### 1.1 Uzasadnienie wyboru tematu

Wraz z upływem czasu postęp technologiczny ma wpływ na życia co raz szerszej rzeszy ludzi na całym świecie. Niezliczone ilości urządzeń zagościły na stałe w domach i mało kto wyobraża sobie bez nich swoje życie. Zaczynając od artykułów gospodarstwa domowego, a kończąc na elektronice użytkowej do której zaliczają się komputery, telewizory czy też smartfony<sup>1</sup>. Wszystkie te urządzenia mają na celu ułatwianie życia swoim użytkownikom.

W parze z licznymi zaletami urządzeń elektronicznych idą jednak pewne wady. Jedną z istotniejszych jest wpływ czasu spędzanego przed różnego rodzaju wyświetlaczami na zdrowie. Badania przeprowadzone na bazie danych Nielsen Audience Measurement pokazują, że przeciętny Polak spędza dziennie średnio 4,5 godziny przed ekranem telewizora<sup>2</sup>. Nie oznacza to jednak, że przez cały ten czas ogląda on telewizję. Oglądanie filmów z dysku komputera, za pomocą serwisów VOD<sup>3</sup> czy granie na konsoli także są wliczone w ten czas. Gdyby jednak dodać do tego czas spędzony przed ekranem smartfona czy też komputera wynik byłby zapewne dwukrotnie większy.

Pogorszenie wzroku czy też wysychanie gałki ocznej są wymieniane jako najczęstsze skutki zbyt dużej ilości czasu spędzanego przed ekranem. Poza próbą jego ograniczenia, jedną z częstszych porad jest próba zmniejszenia kontrastu pomiędzy ekranem a jego otoczeniem.

---

<sup>1</sup> Przenośne urządzenia łączące w sobie zalety telefonów komórkowych oraz przenośnych komputerów (z ang. smartphone).

<sup>2</sup> Badania zostały przeprowadzone z uwzględnieniem osób powyżej 4 roku życia w okresie od stycznia do czerwca 2015 roku [1].

<sup>3</sup> Wideo na życzenie (z ang. video on demand).

Do celów niniejszej pracy należy złożenie i oprogramowanie systemu oświetlenia, który ma rozszerzać obraz widziany na ekranie na jego otoczenie. Poza zmniejszeniem kontrastu, a więc aspektem zdrowotnym, system ma także na celu zwiększyć wrażenia wizualne dostarczane przez oglądany obraz.

System oświetlenia składa się z taśmy diod elektroluminescencyjnych<sup>4</sup> podłączonych do mikrokontrolera Arduino Uno, który z kolei ma współpracować z komputerem z zainstalowanym systemem operacyjnym MacOS. Oprogramowanie mikrokontrolera, którego zadaniem jest sterowanie diodami zostało przygotowane w języku Arduino, natomiast aplikacja kontrolująca cały system przeznaczona na komputer z systemem MacOS została napisana w języku Swift. Wybór języków jest ściśle związany z koniecznością uzyskania jak najlepszej wydajności oraz użyciem najnowszych technologii.

Podobne systemy oświetlenia dostępne są już od pewnego czasu na rynku, jednak to właśnie nowoczesne technologie, prostota wykonania i niskie koszty powinny uczynić z Lightning, bo taką nazwę otrzymał projekt, pełnowartościowego konkurenta.

## **1.2 Problematyka i zakres pracy**

## **1.3 Cele pracy**

## **1.4 Metoda badawcza**

## **1.5 Przegląd literatury w dziedzinie**

## **1.6 Układ pracy**

---

<sup>4</sup> LED (z ang. light-emitting diode).

## **Rozdział 2**

### **Zagadnienia teoretyczne**

## **Rozdział 3**

### **Analiza istniejących rozwiązań**

#### **3.1 Kryteria analizy**

#### **3.2 Porównanie istniejących rozwiązań**

## Rozdział 4

# System oświetlenia Lightning

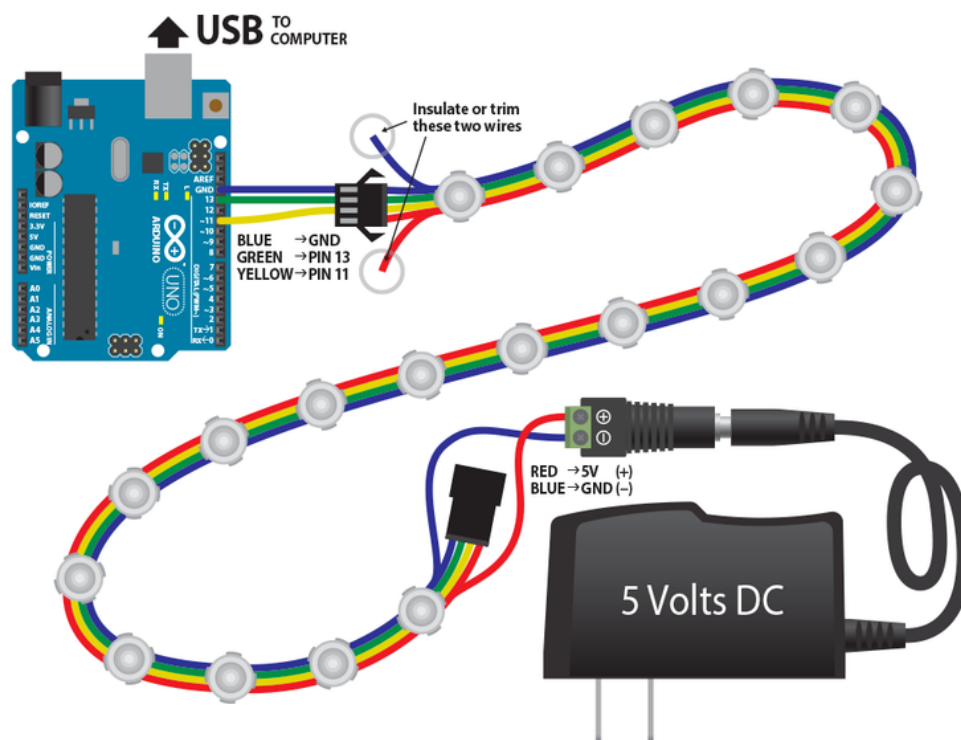
### 4.1 Komponenty systemu

Do konstrukcji systemu wykorzystane zostały następujące komponenty:

- Cyfrowo adresowany łańcuch składający się z 25 diod elektroluminescencyjnych o średnicy 12 mm ze sterownikiem WS2801. Dostępny w sklepie internetowym Botland w cenie 195 zł [2].
- Stabilizowany zasilacz sieciowy o napięciu wyjściowym 5 V. Dostępny w sklepie internetowym Botland w cenie 19,90 zł [3].
- Wtyk ze złączem pozwalającym połączyć łańcuch z zasilaczem. Dostępny w sklepie Botland w cenie 1,90 zł [4].
- Mikrokontroler Arduino Uno w wersji 3. Dostępny w sklepie Botland w cenie 95 zł [5].
- Przewód USB. Dostępny w sklepie Botland w cenie 4,90 zł [6].

Łańcuch z diodami został podłączony od jednej strony za pomocą wymienionego wcześniej wtyku z zasilaczem, z drugiej strony natomiast z mikrokontrolerem. Ten z kolei komunikuje się z komputerem z systemem macOS za pomocą przewodu USB. Dokładny schemat połączenia komponentów przedstawia rysunek 4.2.

Koszt związany ze skompletowaniem wszystkich komponentów wyniósł łącznie 316,70 złotych. Cena ta mogłaby ulec znacznemu zmniejszeniu w przypadku użycia kompatybilnych zamienników dla najdroższych elementów, czyli dla łańcucha diod oraz Arduino Uno w najnowszej wersji.



Rysunek 4.1: Schemat podłączenia komponentów systemu [7]

## 4.2 Moduły oprogramowania

### 4.2.1 Oprogramowanie przeznaczone na mikrokontroler

### 4.2.2 Aplikacja przeznaczona na komputer

## 4.3 Analiza wymagań

### 4.3.1 Wymagania funkcjonalne

W celu stworzenia jak najatrakcyjniejszego system oświetlenia podczas projektowania Lightning przyjęto poniższe wymagania funkcjonalne:

- System musi udostępniać tryb przechwytywania rozszerzający obraz wyświetlany na ekranie na jego otoczenie za pomocą diod elektroluminescencyjnych.



- System musi posiadać tryb animacji za pomocą diod elektroluminescencyjnych. Powinien być on łatwy do rozszerzenia o kolejne animacje.
- System musi być konfigurowalny z poziomu aplikacji sterującej. Konfiguracji podlegać musi co najmniej liczba i rozmieszczenie diod za ekranem, a także wykorzystywany port szeregowy.
- Projekt musi być odpowiednio udokumentowany. Dokumentacja powinna ułatwiać szybkie skonfigurowanie oraz uruchomienie systemu.

### 4.3.2 Wymagania niefunkcjonalne

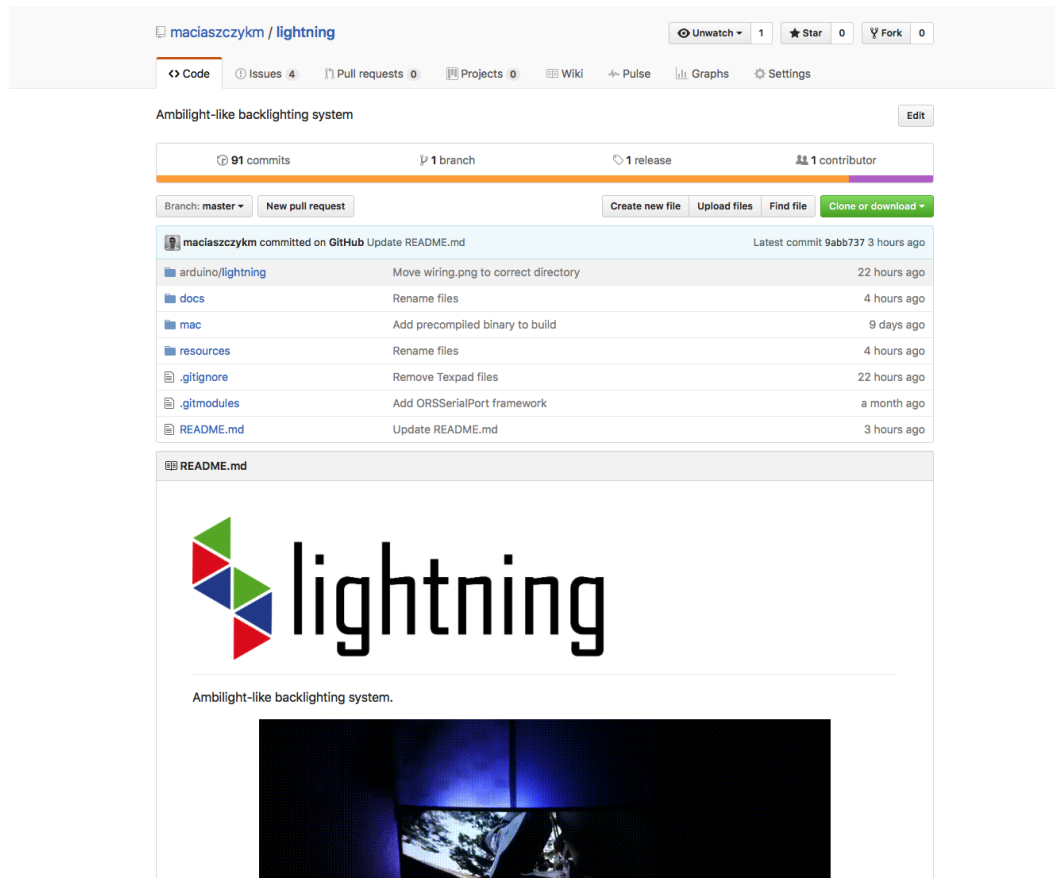
Do wymagań niefunkcjonalnych postawionych przed projektowanym systemem należą:

- System musi działać płynnie, a więc liczba osiągniętych klatek na sekundę powinna być jak największa nawet przy dużych rozdzielczościach przechwytywanego ekranu.
- Korzystanie z aplikacji sterującej powinno być jak najbardziej intuicyjne, a użytkownik powinien mieć do wskazówek dotyczących jej interfejsu.
- Oprogramowanie mikrokontrolera powinno oferować jak najprostszy interfejs i zawierać jak najmniej logiki sterującej.
- Pamięć na obu urządzeniach sterujących powinna być odpowiednio zarządzana, niedopuszczalne są wycieki pamięci czy też zapętlenia programu.

## 4.4 Projekt

### 4.4.1 System kontroli wersji

Podczas pracy nad projektami programistycznymi często wymagana jest współpraca kilku programistów, cofanie pomyłkowo wprowadzonych zmian czy też dziennik zadań do wykonania. Wspomniane funkcjonalności udostępniają systemy kontroli wersji. Do najpopularniejszych należy Git, którego funkcjonalność darmowo udostępnia serwis GitHub, gdzie z kolei znajduje się repozytorium projektu [8].



Rysunek 4.2: Repozytorium projektu w serwisie GitHub

#### 4.4.2 Środowisko programistyczne

#### 4.4.3 Diagram klas

#### 4.4.4 Wzorce projektowe

#### 4.4.5 Interfejs użytkownika

### 4.5 Implementacja

### 4.6 Podręcznik użytkownika

### 4.7 Przykładowa implementacja animacji

### 4.8 Analiza projektu

### 4.9 Perspektywy rozwoju projektu



Rysunek 4.3: Logo projektu

## **Rozdział 5**

### **Podsumowanie**

#### **5.1 Dyskusja wyników**

#### **5.2 Perspektywy rozwoju pracy**

# Bibliografia

- [1] <http://www.wirtualnemedi.pl/arttykul/coraz-dluzej-ogladamy-telewizje-najwiecej-czasu-przed-szklanym-ekranem-spedzaja-seniorzy-raport>.  
Data dostępu – 15.01.2017.
- [2] <https://botland.com.pl/lancuchy-i-matryce-led/2443-lacuch-led-rgb-12-mm-ws2801-cyfrowy-adresowany-25-szt.html>.  
Data dostępu – 17.01.2017.
- [3] <https://botland.com.pl/zasilacze-sieciowe-5-v/1364-zasilacz-impulsowy-5v-25a-wtyk-dc-55-21-mm.html>.  
Data dostępu – 17.01.2017.
- [4] <https://botland.com.pl/szybkoszlacza/1590-wtyk-dc-55-x-21-mm-z-szybkoszlaczem.html>.  
Data dostępu – 17.01.2017.
- [5] <https://botland.com.pl/arduino-moduly-glowne/1060-arduino-uno-r3.html>.  
Data dostępu – 17.01.2017.
- [6] <https://botland.com.pl/przewody-usb-a-b-20/5313-przewod-usb-a-b-tracer-18m.html>.  
Data dostępu – 17.01.2017.
- [7] [https://cdn-learn.adafruit.com/assets/assets/000/001/484/-medium800/led\\_pixels\\_wiring-diagram.png?1396773276](https://cdn-learn.adafruit.com/assets/assets/000/001/484/-medium800/led_pixels_wiring-diagram.png?1396773276).  
Data dostępu – 17.01.2017.
- [8] <https://github.com/maciaszczykm/lightning>.  
Data dostępu – 17.01.2017.

## Spis rysunków

4.1	Schemat podłączenia komponentów systemu [7] . . . . .	8
4.2	Repozytorium projektu w serwisie GitHub . . . . .	10
4.3	Logo projektu . . . . .	11

## **Spis tablic**

## **Spis listingów**