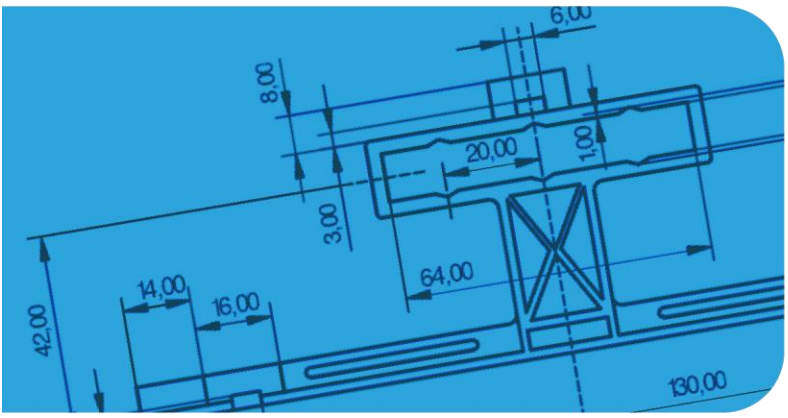
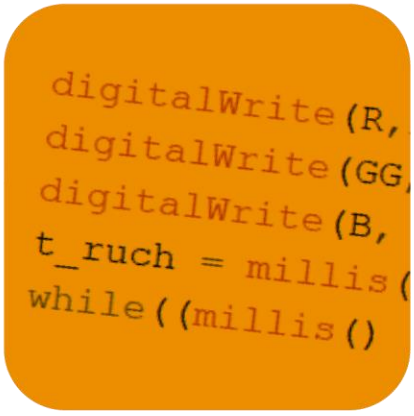




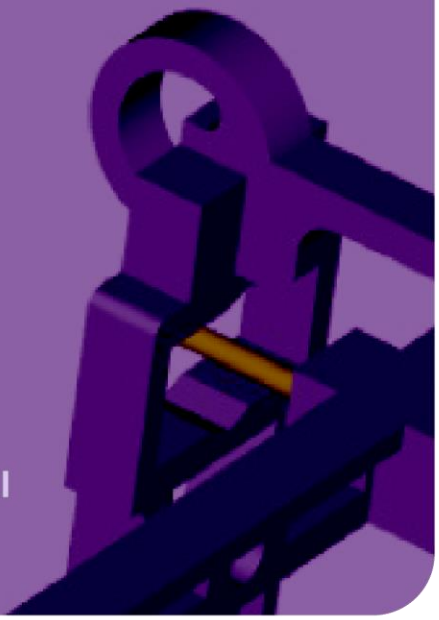
Wojciech Kolarz

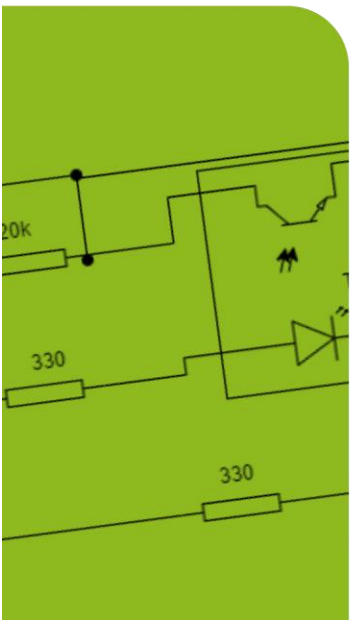
PROJEKT

MAGIA KOLORÓW

DIY

MODELOWANIE 3D
PROGRAMOWANIE
PODSTAWY ELEKTRONIKI





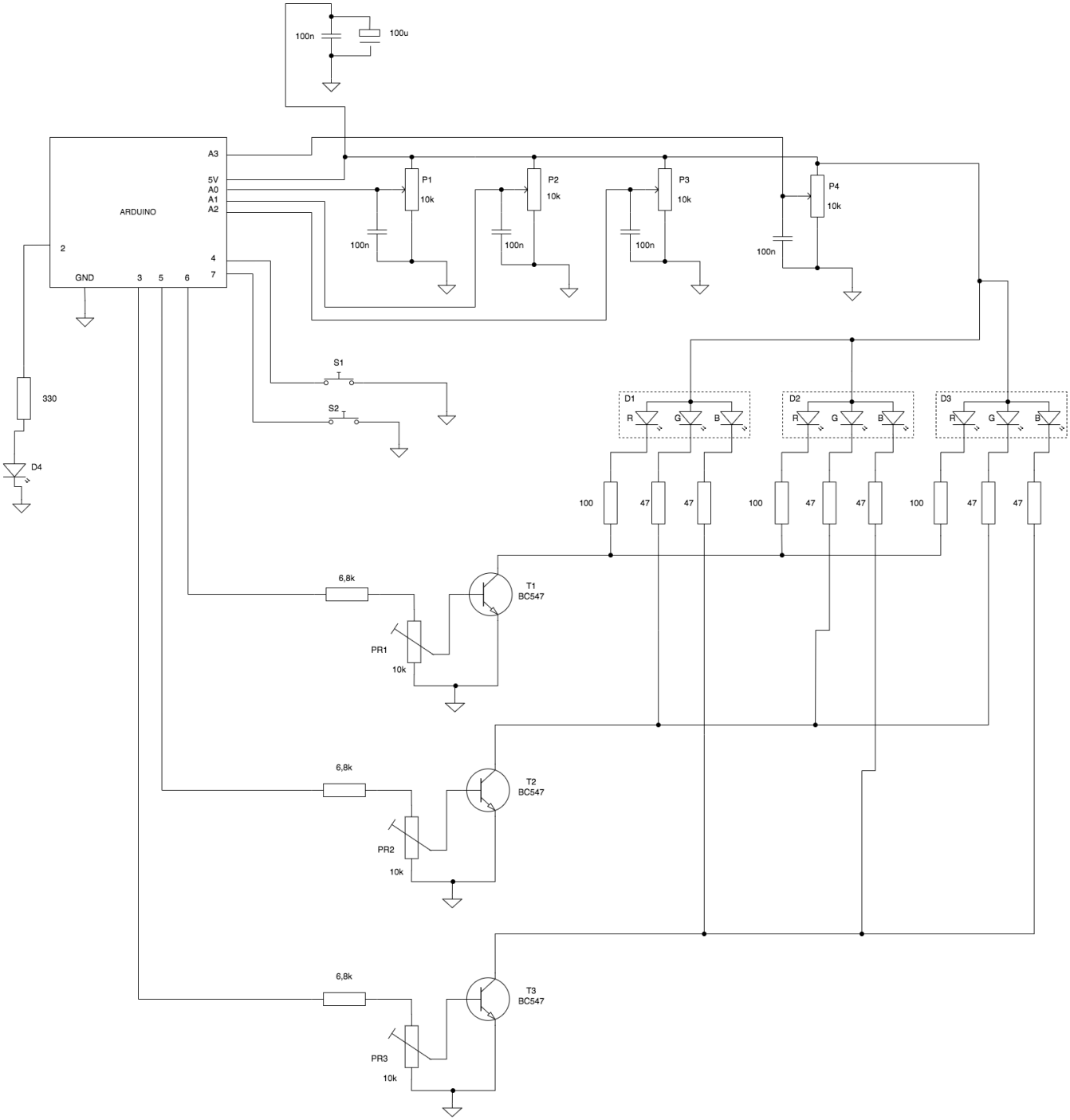
CENTRUM KISS

Informacje uzupełniające do materiału filmowego

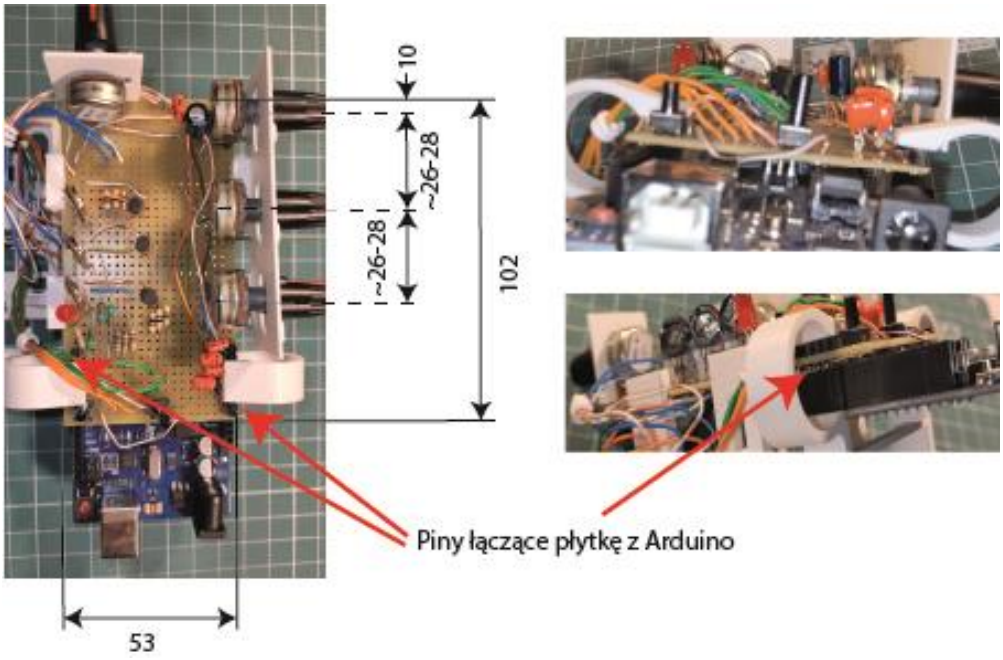
Zastosowane materiały:

- Konstrukcja: filament PLA, wypełnienie dowolne (w prezentowanym modelu 10%), standardowe ustawienia dla dyszy 0,4mm.
- Potencjometr:
  - 10 kΩ logarytmiczny, obrotowy, średnica zewnętrzna 17 mm – 3 szt.
  - 10 kΩ liniowy, obrotowy, średnica zewnętrzna 17 mm – 1 szt.
- Rezystory:
  - 47 Ω - 6 szt.
  - 100 Ω - 3 szt.
  - 300 Ω - 1 szt.
  - 6,8 kΩ - 3 szt.
- Potencjometry montażowe: 10k – 3 szt.
- Włącznik monostabilny typu tact switch – 2 szt.
- Kondensatory:
  - 100nF – 5 szt.
  - Elektrolityczny 100µF/16V – 1 szt.
- Diody LED:
  - Typowa, czerwona 5mm – 1 szt.
  - RGB o wspólnej katodzie, 5mm, prąd maksymalny 20mA, strumień światła >600 mcd – 3 szt.
- Tranzystory: BC 547 lub podobny – 3 szt.
- Arduino UNO R3 – 1 szt.
- Uniwersalne płytki drukowane (raster 2,54m):
  - przycięta do wymiarów 2,1” (53mm) x 4” (102mm) (minimalna długość płytki: 97mm) – 1 szt.
  - przycięta do wymiarów 55mm x 14mm – 1 szt.
- Przewody.
- Wtyki typu gold pin, gniazdo proste 1 x 22piny raster 2,54 lub podobne.
- Śruby M4, M3.
- Pokrętła do potencjometrów (można wydrukować – pliki .stl dostępne w internecie).
- Farby modelarskie (czarna, srebrna).
- Opcjonalnie: folia aluminiowa, klej cyjanoakrylowy.

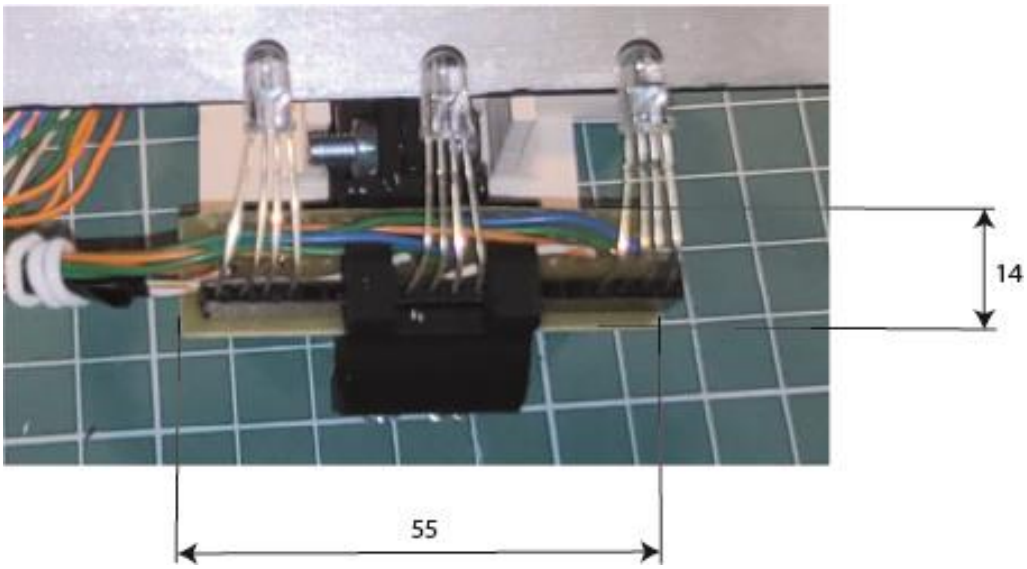
Elementy elektroniczne zlutowane są na uniwersalnej płytce drukowanej. Rozmieszczenie elementów na płytce jest dowolne, ważne jest jedynie usytuowanie pionów łączących płytkę z Arduino, oraz usytuowanie potencjometrów. Diody RGB wpięte są do gniazda, wlutowanego do małej, uniwersalnej płytki drukowanej. W prezentowanym rozwiązaniu użyte jest jedno długie gniazdo, zamiast tego można do płytki wlutować trzy gniazda po 4 piny. Włączniki monostabilne S1 i S2, służą do zmiany trybu pracy oraz do zapamiętania koloru lub kasowania wcześniej zapamiętanej sekwencji. Dioda D4 sygnalizuje tryb pracy urządzenia oraz sygnalizuje zapamiętanie i kasowanie sekwencji kolorów. Zmiana intensywności poszczególnych kolorów realizowana jest poprzez trzy potencjometry (P1, P2, P3). Potencjometr P4 służy do regulacji częstotliwości z jaką odtwarzane będą zapamiętane kombinacje ustawień. Sterowanie diod D1, D2, D3 jest realizowane poprzez zwieranie katod do masy za pomocą tranzystorów T1, T2, T3. Rezystory 6,8kΩ wraz z potencjometrami montażowymi zapewniają odpowiedni punkt pracy tranzystorów.



Rys. 1. Schemat układu.

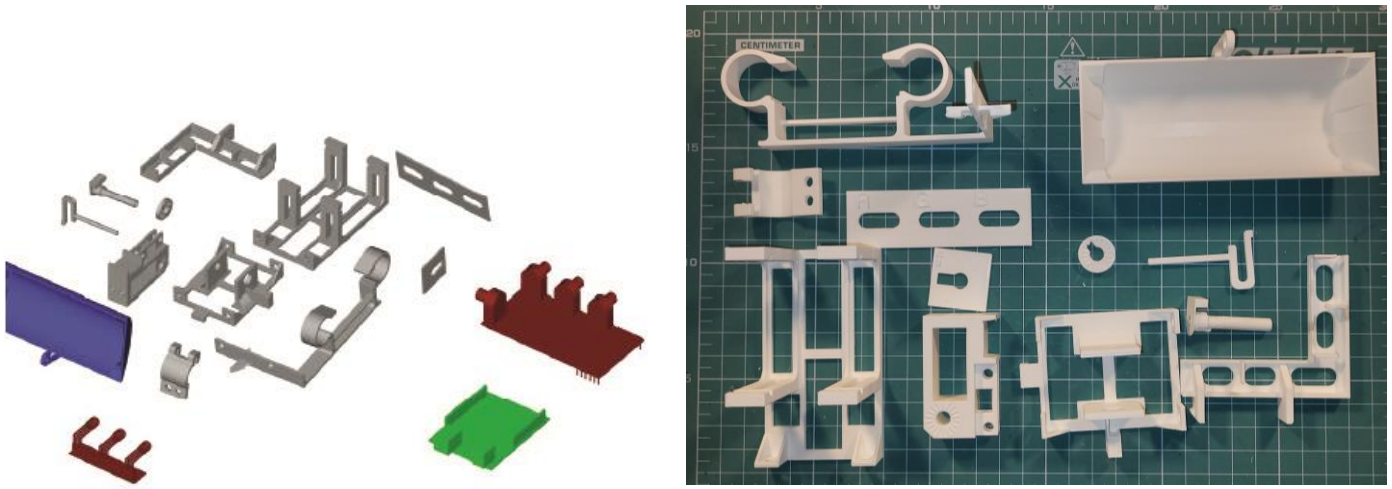


Rys. 2. Wymiary płytki, rozmieszczenie elementów, połączenie z Arduino.

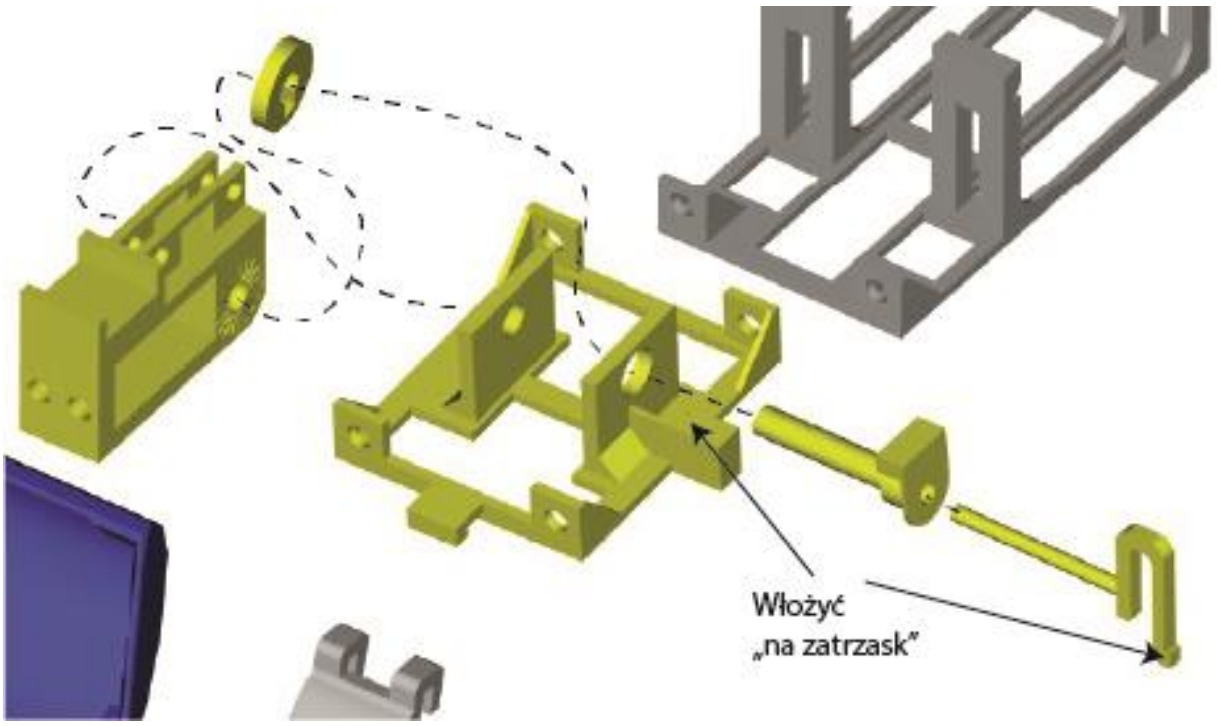


Rys. 3. Płytka z diodami RGB.

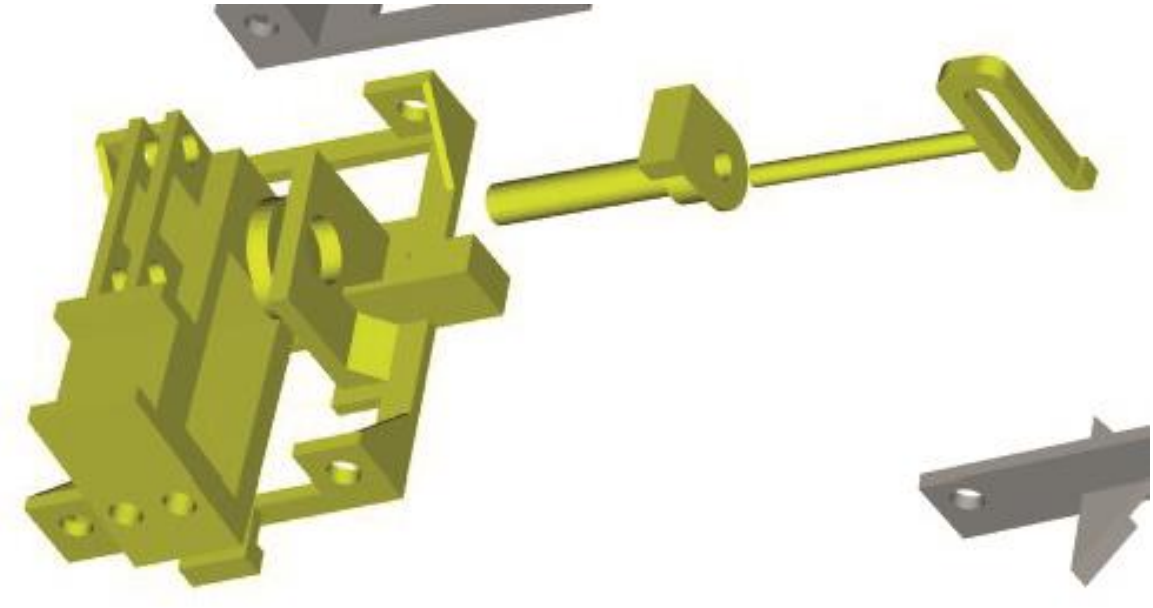
Wydrukowane elementy należy złożyć w sposób pokazany na rysunkach (rys. 4. – rys. 12.). Przez złożeniem, reflektor pomalować farbami modelarskimi (wnętrze – kolor srebrny, powierzchnie zewnętrzne – kolor czarny), wnętrze reflektora można również wykleić folią aluminiową (stroną matową folii – na zewnątrz).



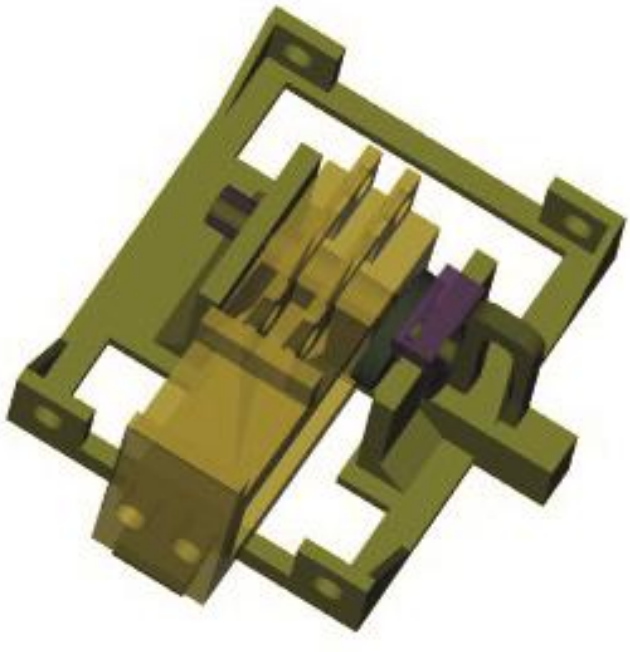
Rys. 4. Elementy przed złożeniem.



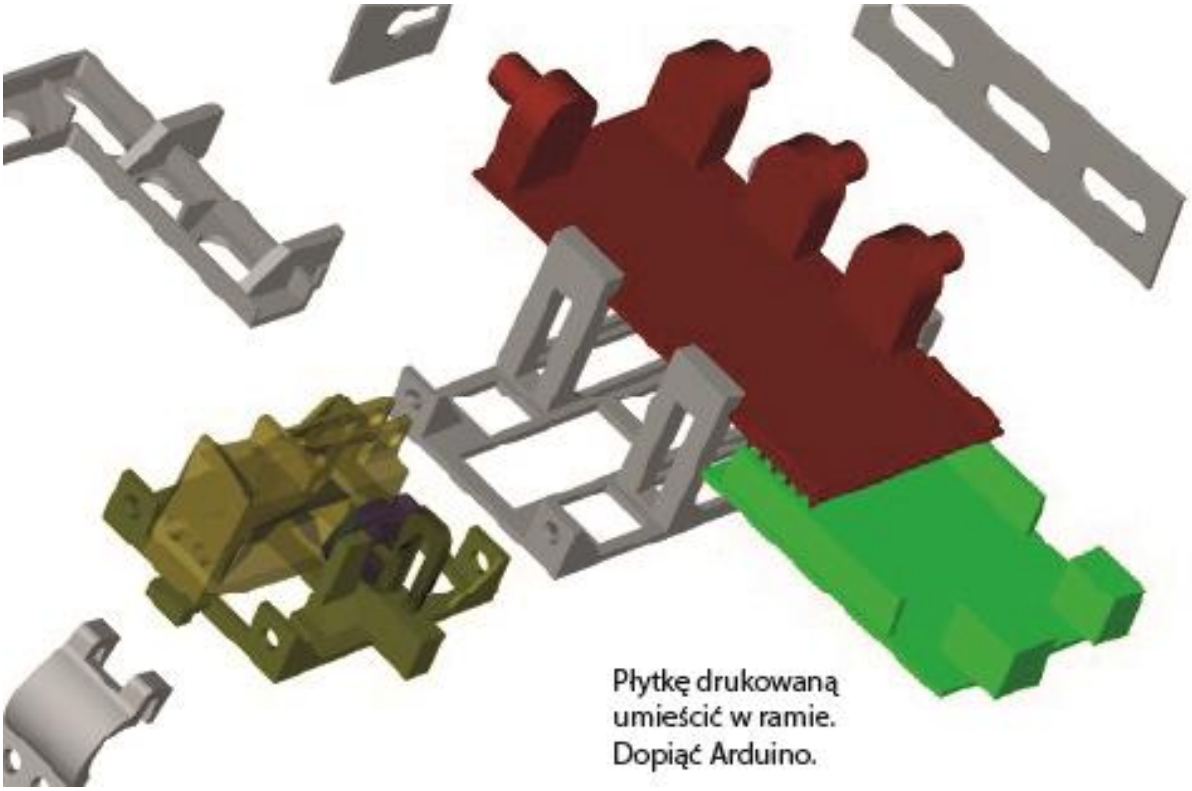
Rys. 5. Złożenie elementów mocowania reflektora (1).



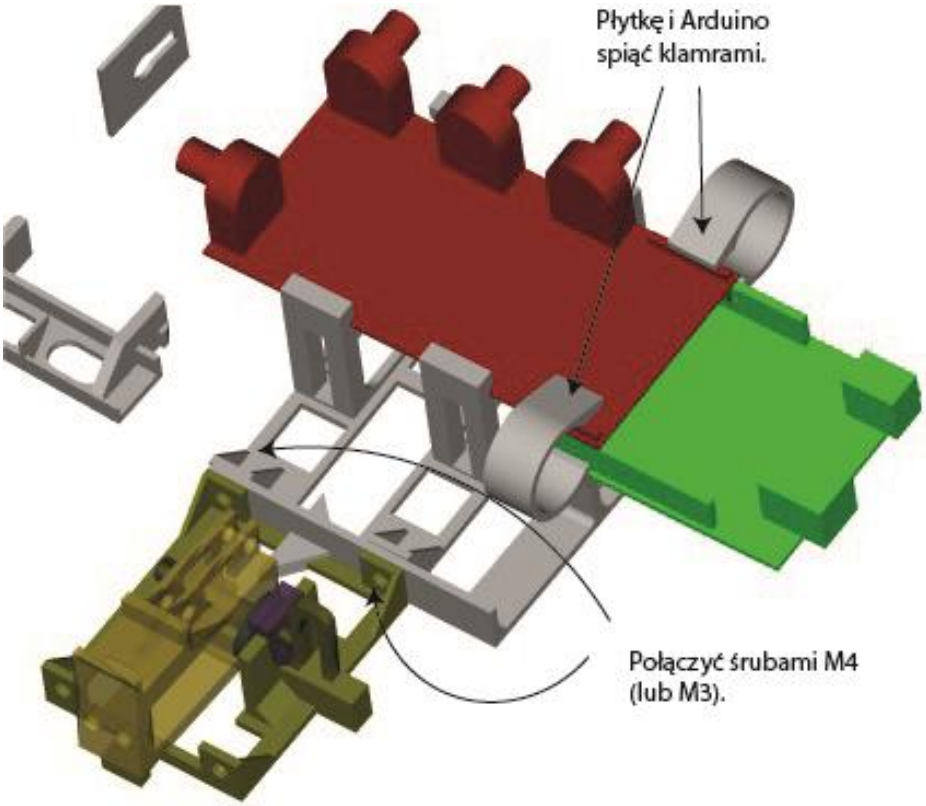
Rys. 6. Złożenie elementów mocowania reflektora (2).



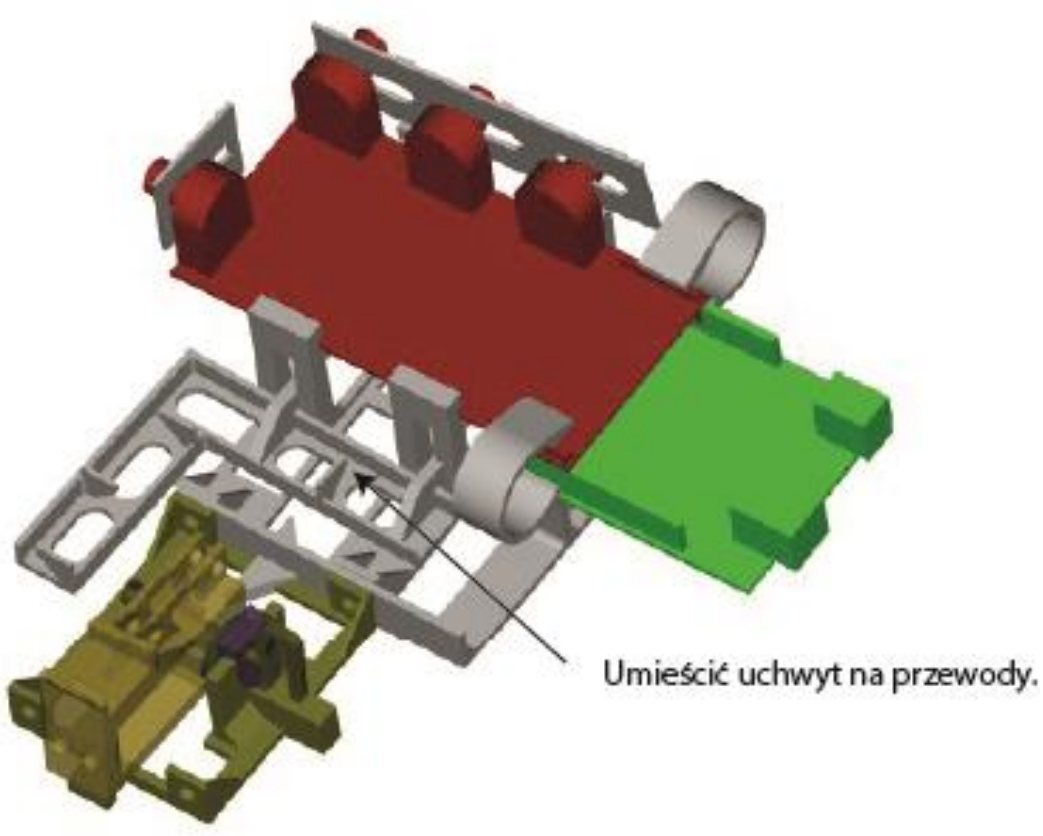
Rys. 7. Złożenie elementów mocowania reflektora (3).



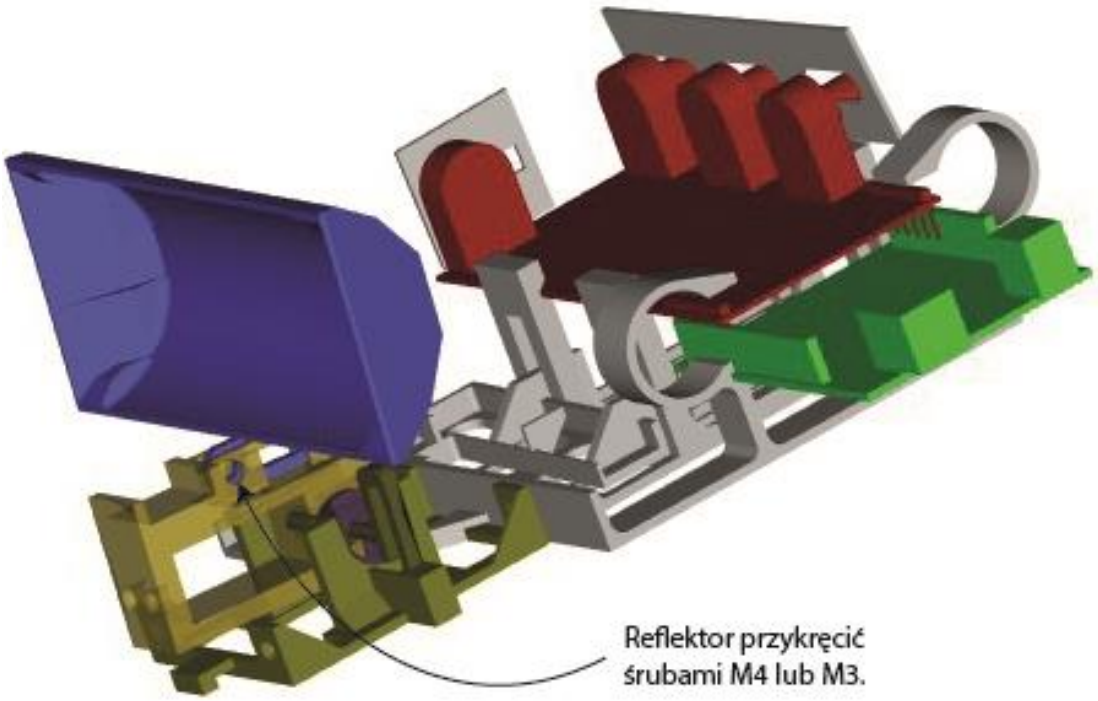
Rys. 8. Zamocowanie płytki i Arduino (1).



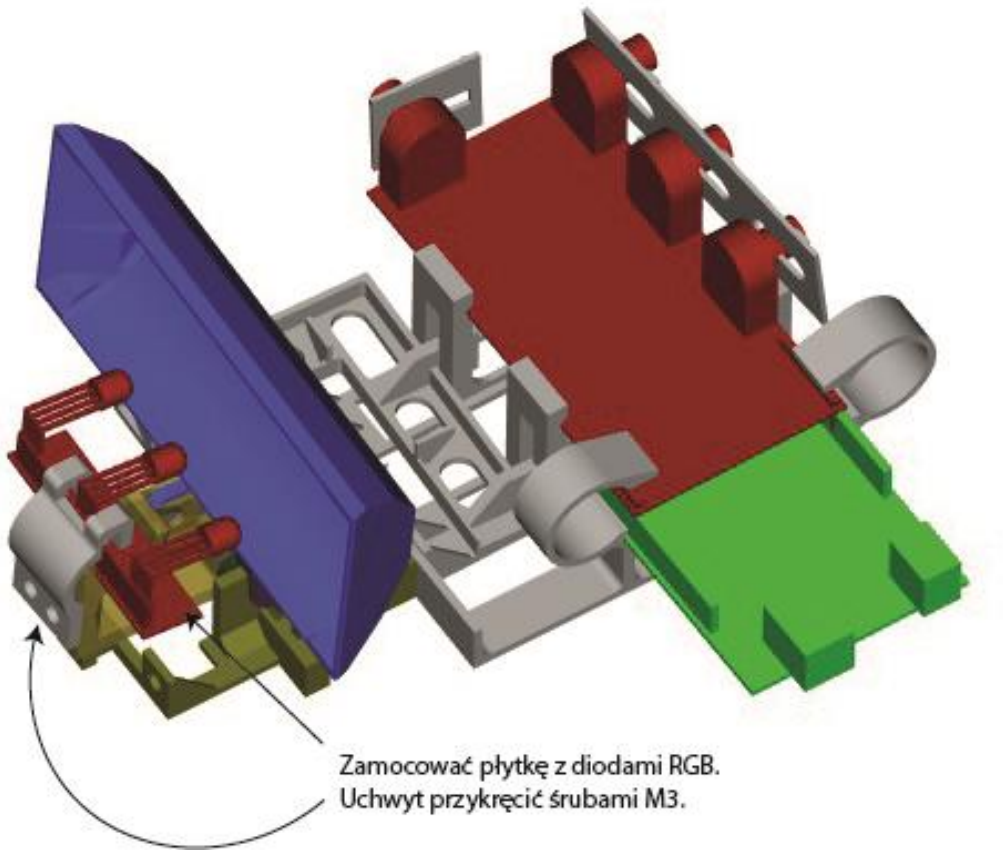
Rys. 9. Zamocowanie płytki i Arduino (2).



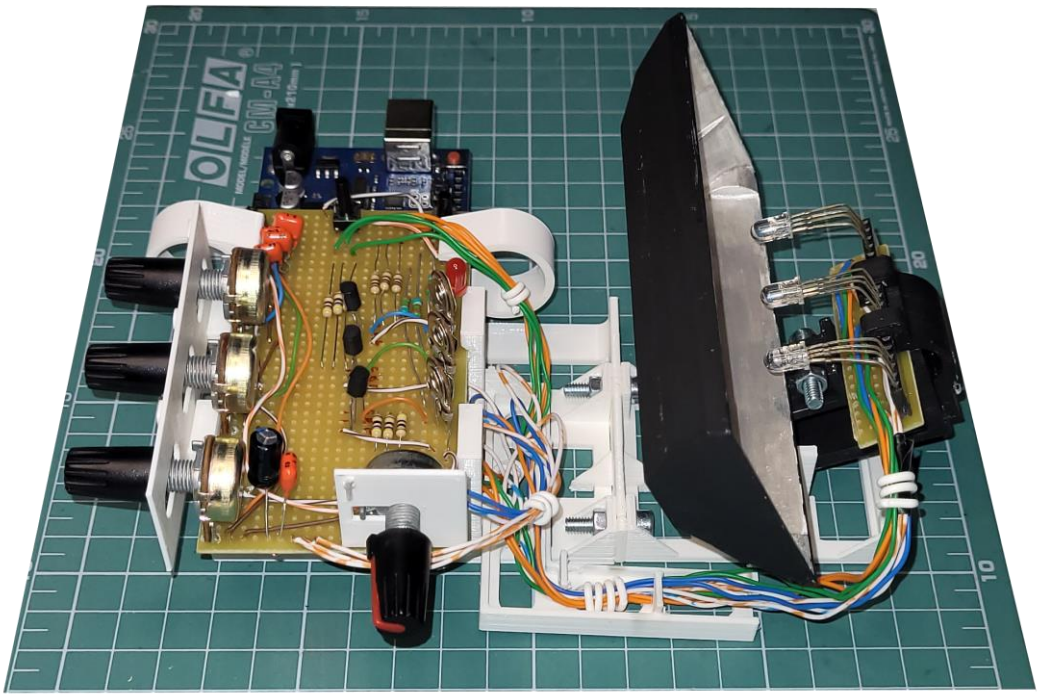
Rys. 10. Montaż uchwytu (rynienki) na przewody.



Rys. 11. Montaż reflektora.



Rys. 12. Montaż diod RGB.



Rys. 13. Zmontowane urządzenie.

Program.

```
#define D1 3 //R
#define D2 5 //G
#define D3 6 //B
#define D4 2 //kontrola
#define S1 4
#define S2 7
int r, g, b;
int pwm_r, pwm_g, pwm_b;
boolean tryb;
int kolory [3][50];
int nr_kolor;
int t;
void setup()
{
  pinMode(D1, OUTPUT);
  pinMode(D2, OUTPUT);
  pinMode(D3, OUTPUT);
  pinMode(D4, OUTPUT);
  pinMode(S1, INPUT_PULLUP);
  pinMode(S2, INPUT_PULLUP);
  nr_kolor = 0;
}
```

Rys. 14. Program (1).

Zaprezentowany program umożliwiać będzie wybór koloru (mieszanie barw podstawowych). Wybrany kolor (kombinacja kolorów podstawowych) będzie mógł być zapamiętany. Program umożliwia cykliczne odtwarzanie zapamiętanych kolorów z częstotliwością regulowaną potencjometrem.

Realizując przedstawiony program, urządzenie pracuje w dwóch trybach:

- **Tryb 1.** umożliwiający ustawienie dowolnego koloru (regulacja kolorów podstawowych potencjometrami P1, P2, P3), oraz zapamiętanie ustawionego koloru (poprzez przyciśnięcie włącznika S2),
- **Tryb 2.** cyklicznego wyświetlania zapamiętanych kolorów z częstotliwością regulowaną potencjometrem P4.

W programie zdefiniowano, że tranzystory sterujące diodami RGB podłączone są do pinów 3,5,6 (kolejność pinów odpowiada kolejności kolorów). Nie jest ważna kolejność podłączenia poszczególnych kolorów (np. inna kolejność na schemacie), kolejność będzie zależna od rozmieszczenia elementów na płytce drukowanej.

W dalszej części opisu D1, D2, D3 będą oznaczały poszczególne kanały kolorów w kolejności R, G, B. Przyjęto takie oznaczenia z uwagi na zgodność z oznaczeniami pinów zastosowane w programie.

Do zmiennych **r, g, b** zapisywane będą wartości odczytane z wejść analogowych Arduino, wartości te będą zależne od ustawienia potencjometrów. Do zmiennych **pwm_r, pwm_g, pwm_r** zapisywane będą obliczone wartości **pwm** dla poszczególnych kolorów.

W tablica **kolory** przechowywane będą zapamiętane kombinacje kolorów.

Zmienna **nr_kolor** będzie wykorzystywana w procesie zapamiętywania kombinacji kolorów i ich odtwarzania.

Zmienna logiczna **tryb** ustawiana będzie na wartość *true* dla trybu 1. (wartość *false* dla trybu 2.).

Do zmiennej **t** zapisywane będą wartości odczytane z wejścia analogowego Arduino, które zależeć będą od nastawienia potencjometru ustawiającego częstotliwość odtwarzania zapamiętanych kolorów.

```
23 void loop()
24 {
25   tryb = true;
26   for(int i=0; i<3; i++)
27   {
28     digitalWrite(D4,HIGH);
29     delay(100);
30     digitalWrite(D4,LOW);
31     delay(500);
32   }
33   while(tryb)
34   {
35     r = analogRead(A0);
36     g = analogRead(A1);
37     b = analogRead(A2);
38     pwm_r = map(r,0,1023,0,255);
39     pwm_g = map(g,0,1023,0,255);
40     pwm_b = map(b,0,1023,0,255);
41     analogWrite(D1,pwm_r);
42     analogWrite(D2,pwm_g);
43     analogWrite(D3,pwm_b);
```

Rys. 15. Program (2).

Pętla **loop** rozpoczyna się sekwencją poleceń powodujących trzykrotne migniecie diody D4 – sygnał rozpoczęcia pracy urządzenia w trybie 1.

W pierwszej pętli **while** odczytywane będą wartości na wejściach analogowych (regulacja natężenia kolorów podstawowych), odczytane wartości mapowane będą na zakres 0 – 255 (zakres pwm). Na piny, do których podłączone są tranzystory sterujące diodami podawany będzie odpowiedni sygnał PWM.

```
44    if(!digitalRead(S2))
45    {
46      kolory[0][nr_kolor] = pwm_r;
47      kolory[1][nr_kolor] = pwm_g;
48      kolory[2][nr_kolor] = pwm_b;
49      if(nr_kolor<40) nr_kolor++;
50      digitalWrite(D4,HIGH);
51      delay(1500);
52      digitalWrite(D4,LOW);
53      if(!digitalRead(S2))
54      {
55        nr_kolor = 0;
56        for(int i = 0; i<4; i++)
57         {
58           digitalWrite(D4,HIGH);
59           delay(100);
60           digitalWrite(D4,LOW);
61           delay(300);
62         }
63      }
64     }
65     tryb = digitalRead(S1);
66   }
```

Rys. 16. Program (3).

W przypadku przyciśnięcia włącznika S2 zapamiętana zostanie aktualna kombinacja kolorów podstawowych, czyli wybrany aktualnie kolor.

Zapamiętanie wybranego koloru sygnalizowane będzie zaświeceniem się diody D4.

Jeżeli program wykryje wciśnięcie włącznika S1, zmienna **tryb** przyjmie wartość *false* (wartość *false* jest tożsama z wartością *LOW* odczytaną z wejścia Arduino).

```
67    tryb = true;
68    for(int i=0; i<3; i++)
69    {
70      digitalWrite(D4,HIGH);
71      delay(100);
72      digitalWrite(D4,LOW);
73      delay(100);
74      digitalWrite(D4,HIGH);
75      delay(100);
76      digitalWrite(D4,LOW);
77      delay(500);
78    }
79    while(tryb)
80    {
81      for(int i = 0; i<nr_kolor; i++)
82      {
83        analogWrite(D1,kolory[0][i]);
84        analogWrite(D2,kolory[1][i]);
85        analogWrite(D3,kolory[2][i]);
86        t = analogRead(A3);
87        t = map(t,0,1023,5,500);
88        delay(t);
89        if(!digitalRead(S1)) break;
90       }
91     tryb = digitalRead(S1);
92    }
93  }
```

Rys. 17. Program (4).

Zmiana trybu pracy sygnalizowana będzie poprzez mignięcie diody D4.

Druga pętla **while** wykonywana będzie do momentu wciśnięcia włącznika S1 (zmiana trybu pracy). W tej pętli **while** znajduje się pętla **for**, w której sekwencyjnie odtwarzane będą zapamiętane wcześniej kolory (kombinacje kolorów podstawowych).

Przedstawiony program, nie jest prostym rozwiązaniem realizującym funkcjonalności urządzenia. Program można rozbudować np. poprzez wprowadzenie wyświetlania zapamiętanych kolorów „tam i z powrotem”. Ciekawym wyzwaniem może być również taka modyfikacja programu, aby podczas wyświetlania zapamiętanych kolorów zmiana pomiędzy kolejnymi kolorami była płynna.

Pierwsze uruchomieni.

Podczas pierwszego uruchomienia urządzenia należy ustawić maksymalną jasność świecenia diod, czyli PWM z wypełnieniem 100%, następnie za pomocą potencjometrów montażowych należy wyregulować optymalne natężenie światła dla poszczególnych kolorów. W tym celu najlepiej skierować światło na białą kartkę i zmieniając nastawy potencjometrów montażowych doprowadzić do tego, aby natężenie światła było możliwe jak największe, lecz równocześnie światło to powinno być światłem białym, bez dominacji któregokolwiek koloru.