**Przeróbka stomatologicznej lampy polimeryzacyjnej halogenowej na lampę diodową**

W stomatologii podczas leczenia ubytków zębowych możemy skorzystać z różnych materiałów do wypełniania. Jednymi z nich są plomby z materiału światło utwardzalnego. Charakteryzują się lepszym efektem kosmetycznym i trwałością w porównaniu z chemoutwardzalnymi oraz większą łatwością zakładania. W związku z tym wypierają te pierwsze z użycia .

Aby doszło do uwarzenia kompozytu, musi on zostać poddany działaniu światła o określonej długości fali zakresie 430 nm-450nm oraz intensywności świetlnej.

Lampa poddana przeróbce to Vivadent Astralis 5 wyprodukowana w 1998r.

Przeróbka została zastosowana ze względu na uszkodzenie lampy (problem z procesorem, włączanie i wyłączanie się lampy, buzzer szaleje). Dodatkowo o żarówki halogenowe do lampy było coraz trudniej (i są one drogie).

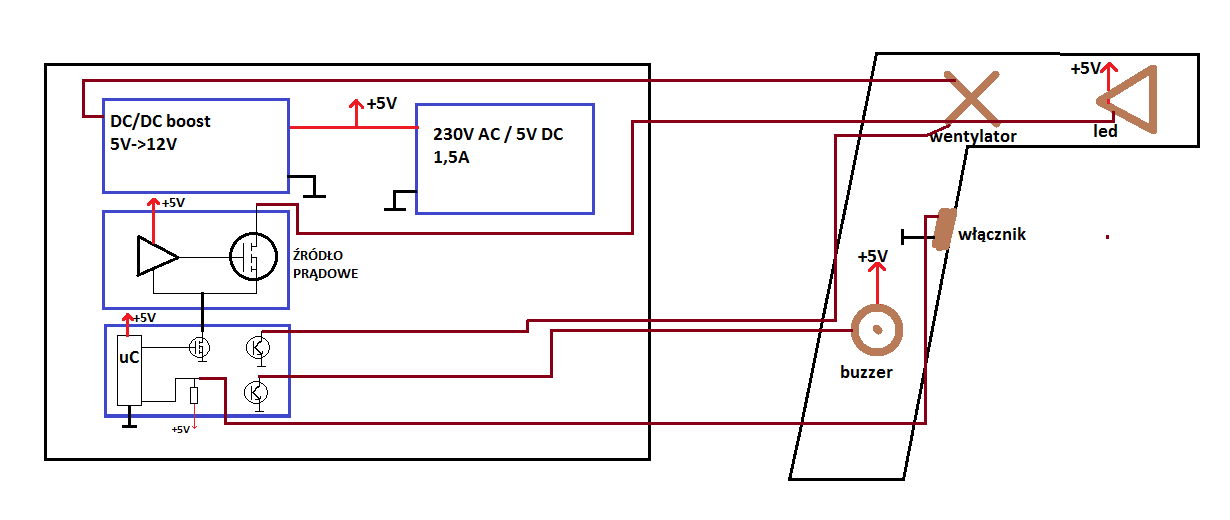
Dlatego lampa została przerobiona na diodową.



**Poddana przeróbce lampa to Vivadent Astralis 5**

Aby zrozumieć istotę przeróbki został przegotowany schemat blokowy. Znajduje się na nim:

* przetwornica impulsowa 230V AC → 5V/1,5A DC
* przetwornica DC/DC podwyższająca napięcie 5V → 12V służąca do zasilania wentylatora
* źródło prądowe z napięciem odniesienie
* mikroprocesor Attiny13A sterujący źródłem prądowym, buzzerem oraz wentylatorem



**Schemat blokowy połączeń modułów w lampie**

1. **Płytka mikroprocesora**

Do analizy układu, wykorzystany został zamieszczony poniżej schemat układu. Użyty mikroprocesor to Attiny13A. Na PCB znajduje się złącze do programatora (ISP). Cały układ zasilany jest bezpośrednio z przetwornicy impulsowej AC/DC (5V).

- PB0: sterowanie tranzystorem N-MOSFET (typu logic lever, małe Rdson przy małym Vgs). Tranzystor podaje masę na źródło prądowe.

- PB1: przycisk (zwieranie do masy, włączenie) rezystor pull-up R2

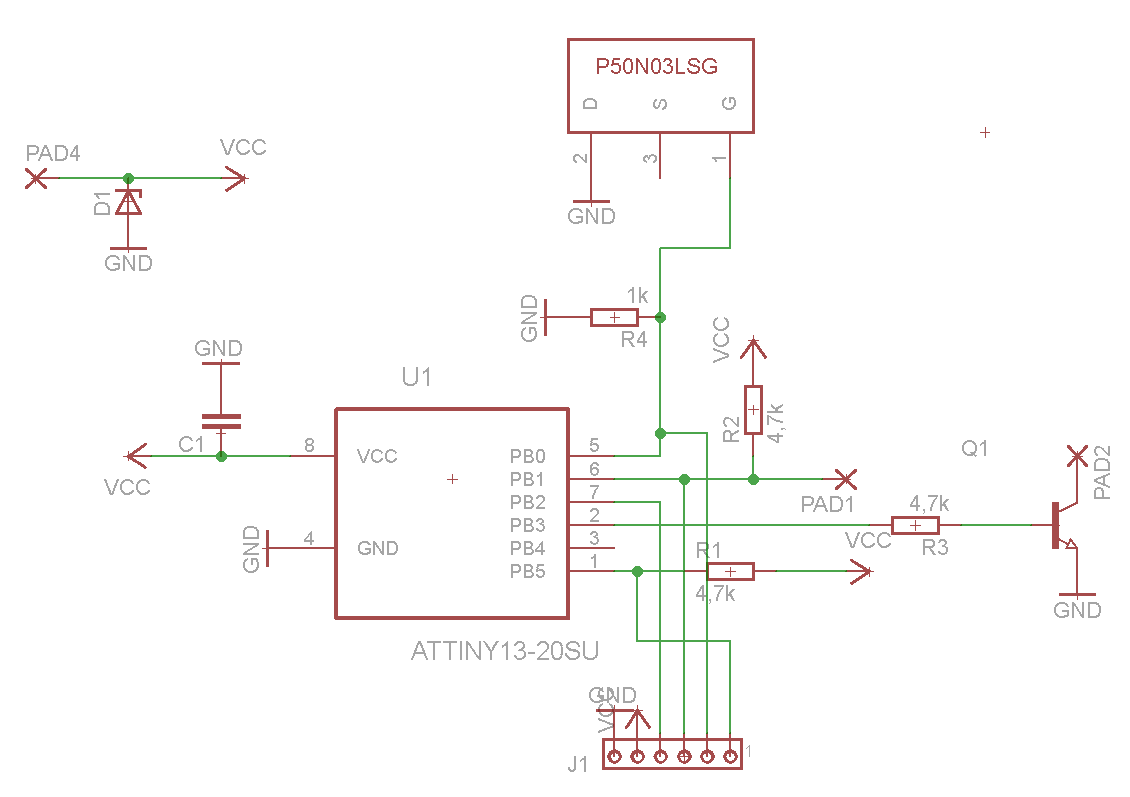
- PB2: wentylator (na schemacie brak tranzystora NPN BC547 służącego do załączania oraz rezystora bazowego połączonego do wyjścia PB2, jest dolutowany później widoczny w zdjęciach konstrukcji)

- PB3: wyjście na tranzystor sterujący buzzerem

- PB4: N/C

- PB5: reset z rezystorem podciągającym.

Dioda zenera D1 (5V1) służy jako zabezpieczenie przed odwrotnym podaniem napięcia oraz zbyt jego wysokim poziomem.



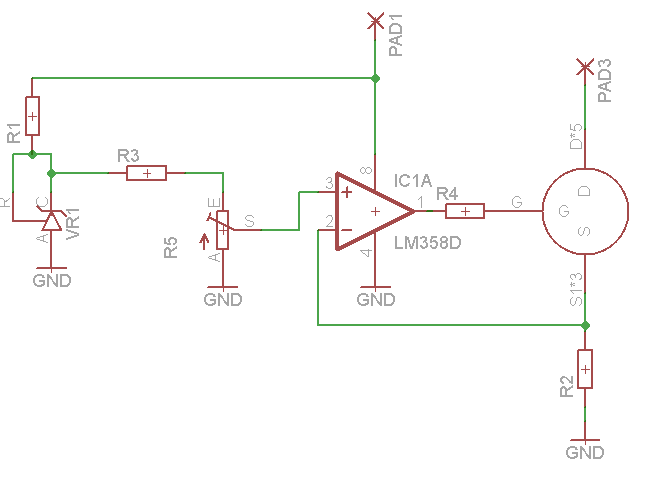
**Schemat układu sterującego – mikroprocesor Attiny13**

1. **Źródło prądowe**

Użyta aplikacja to kanoniczna wręcz postać źródła prądowego ze wzm. operacyjnym i tranzystorem. Ze względu na to że napięcie zasilania wynosi 5V natomiast dioda pracuje w optymalnych warunkach na ok. 3.2V zastosowany został tranzystor MOSFET. W przypadku zastosowania bipolarnego spadek napięcia byłby zbyt duży. Jako napięcie odniesienia zastosowany został popularny TL431 (2,5V).

Rezystor R1=2k, R3=20k. Potencjometrem R7=1k ustawione zostało napięcie które odpowiada prądowi przepływającemu przed diodę. Rezystor R2=0,1R jest rezystorem pomiarowym.

Układ jest teoretycznie OK ale wzmacniacz operacyjny nie lubi obciążeń pojemnościowych i źle może współpracować z pojemnością bramki tranzystora MOSFET. Prawie na pewno znajdą się takie warunki, przy których rezystancja wyjściowa wzmacniacza połączona z pojemnością wejściową MOSFETa dadzą przesunięcie fazy 180 stopni. Drugie 180 dołoży wzmacniacz (od wejścia odwracającego) i otrzymujemy genenerator (układ będzie się wzbudzał). Dlatego został dołożony opornik szeregowo z bramką (100R).



**Schemat źródła prądowego – warto zauważyć że masa GND nie jest połączona bezpośrednio do masy układu sterującego lecz podawana jest przez MOSFET-a gdy chcemy właczyć diodę.**

1. **Dioda LED**

Ponieważ do utwardzenia niezbędne jest promieniowanie UV o określonej długości fali, została użyta specjalna dioda LED. Można by zastosować diodę LED o świetle normalnym ponieważ lampa zawiera filtr który na wyjście dopuści tylko światło o określonej długości, ale wolałem zastosować specjalistyczną diodę której cała moc będzie o określonej barwie światła.

|  |  |
| --- | --- |
| Kolor | UV- ultrafiolet |
| Moc | 3W |
| Chip | 45mil x 45mil Taiwan |
| Napięcie pracy | 3.3V - 3.6V |
| Max. prąd przewodzenia | 700mA |
| Długość Fali | 430 - 435 nm |
| Kąt świecenia | 140 stopni |

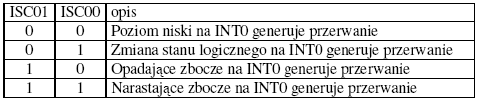
1. **Analiza programu**

W tej części omówię ważniejsze części programu. Resztę opisu znajduje się w komentarzach w programie.

**- ustawienie przerwania od przycisku**

|  |
| --- |
| MCUCR |= (1<<ISC01)|(1<<ISC00);  GIMSK |= (1<<INT0);  GIFR|=(1<INTF0); |

Rejestr MCUCR - rejestr kontroli mikrokontrolera, zawiera między innymi bity kontrolne do wyboru sposobu wykrywania przerwania INT0 i INT1.



Gdy określimy sposób generowania przerwana, musimy uaktywnić przerwanie przez ustawienie na 1 bitu **INT0** w rejestrze **GIMSK** (ang. General Interrupt Mask Register – rejestr maskowania przerwań zewnętrznych)

Z przerwaniami zewnętrznymi związany jest jeszcze rejestr znaczników przerwań GIFR**.** Ma on podobną budowę do rejestru GIMSK, który maskuje (tzn. uaktywnia lub blokuje) przerwania zewnętrzne na tych samych bitach. Bity INTF0 i PCIF zostają ustawione na 1, gdy wystąpi warunek przerwania INT lub PCINT. Jeśli zostanie przy tym wywołana procedura ISR danego przerwania, to odpowiedni bit rejestru znaczników przerwań zostanie automatycznie wyzerowany.

-**ustawienie przerwania od timera**

Timer pracuje w trybie CTC, licznik zlicza od 0 do OCR0A i potem znów od zera. Generuje przerwanie w wektorze ISR(TIM0\_COMPA\_vect) co około 0,1s.

|  |
| --- |
| OCR0A = 96; //rejest porównania timera - max 255 (dla 94 – 0,1s 100 Hz)  TCCR0A |= (1 << WGM01);//Tryb CTC  TCCR0B |= (1 << CS02)|(1 << CS00);//clock select - clk/1024  TIMSK0 |= (1 << OCIE0A);//włączenie timera |

**- przerwanie od przycisku**

Obsługa przerwania następuje w wektorze ISR(INT0\_vect).

Instrukcja GIFR|=(0<INTF0); czyli wyłączenie przerwania, została dodana ponieważ następowało niepożądane wyzwalanie przerwania podczas jego działania co objawiało się naprzemiennym włączaniem i wyłączaniem lampy.

|  |
| --- |
| ISR(INT0\_vect)  {  GIFR|=(0<INTF0);  if (flaga==0)  {  led\_on;  buzz\_on;  \_delay\_ms(20);  buzz\_off;  flaga=1;  } else if (flaga>0)  {  led\_off;  flaga=0;  pm1=0;  buzz\_on;  \_delay\_ms(20);  buzz\_off;  pomocnicza\_fan1=0;  }  \_delay\_ms(50);    GIFR|=(1<INTF0);  } |

Lampa czeka na naciśnięcie przycisku. W momencie naciśnięcia program wchodzi w przerwanie zewnętrzne. Włącza led, piknie buzerem na włączenie (przez czas 20ms) i ustawi „flaga=1”.

W przypadku ponownego naciśnięcia przycisku, program znów wejdzie w to przerwanie z tym że przejdzie do drugiej części czyli wyłączy led, piknie buzerem na wyłączenie (przez czas 20ms) oraz ustawi „flaga=0” oraz inne pomocnicze wykorzystywane w pętli gównej.

**- pętla główna**

W pętli gównej odbywa się sterowanie buzerem oraz wentylatorem. Po naciśnięciu przycisku włączenia w obsłudze przerwania zewnętrzenego zemini się wartość stanu zmiennej „flaga=1”. Program wyzeruje zmienne zliczania czasu buzera (timer) i wentylatora (fan\_timer1, fan\_timer2). Zmienna pm1 wprowadzona została aby nie zerować timerów w jednym cyklu świecenia.

|  |
| --- |
| if((flaga>0 && pm1==0)==1)  {  timer=0;  fan\_timer2=0;  fan\_timer1=0;  pm1=1; |

W następnej kolejności, progam cały czas sprawdza w pętli czy przekroczone zostały wartości zmiennych timer oraz fan\_timer1. Narastają one cały czas w wektorze przerwania timera. Jeśli timer>120 (dobrane dla czasu 10s) nastąpi uruchomienie buzera na 30ms (piknięcie).

Gdy od początku włączenia minie ok. 15sek. nastąpi włączenie wentylatora. Wentylator nie ma sprzężenia zwrotnego w postaci temperatury radiatora i uruchamiany jest gdy lampa chodzi przez ten określony czas (dobrane eksperymentalnie, radiator zaczyna się nagrzewać po chwili). Zmienna „pomocnicza\_fan1” została wprowadzona aby po uruchomieniu wentylatora, zatrzasnąć stan jego włączenia, tzn. gdyby w tym czasie naciśnięty został przycisk wentylator nie zostanie wyłączony tylko będzie dalej chłodził radiator.

Zerowana jest zmienna „fan\_timer2=0” która odpowiada na naliczanie czasu do wyłączenia wentylatora

|  |
| --- |
| else if((flaga>0 && pm1>0)==1)  {  if(timer>120)  {  buzz\_on;  \_delay\_ms(30);  buzz\_off;  timer=0;  }  if((fan\_timer1>170 && pomocnicza\_fan1==0)==1)  {  fan\_on;  fan\_timer2=0;  pomocnicza\_fan1=1;  }  } |

Wyłączenie wentylatora następuje po czasie ok. 15s od włączenia.

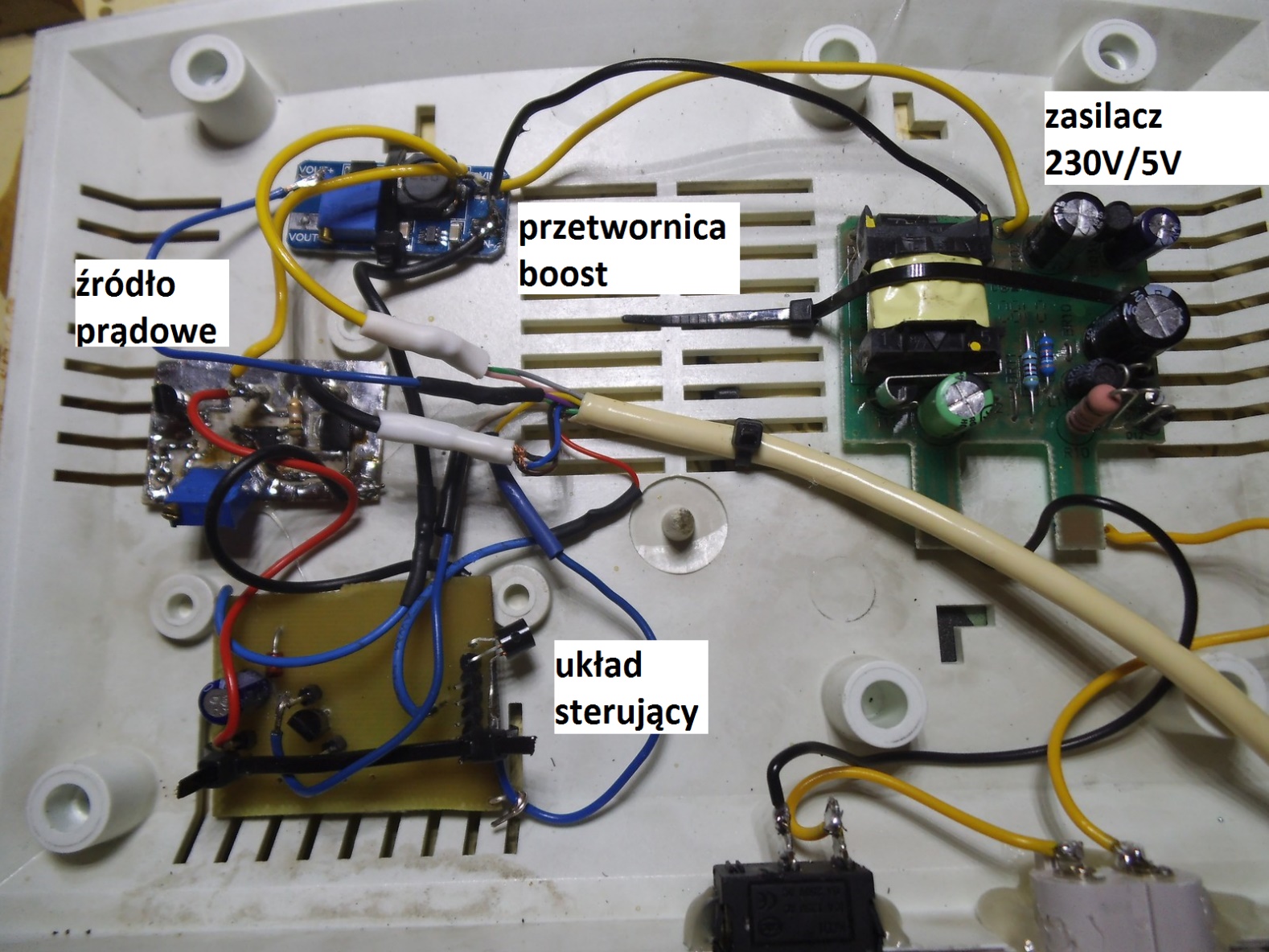
|  |
| --- |
| if(fan\_timer2>170)  {  fan\_off;  } |

**- przerwanie od timera**

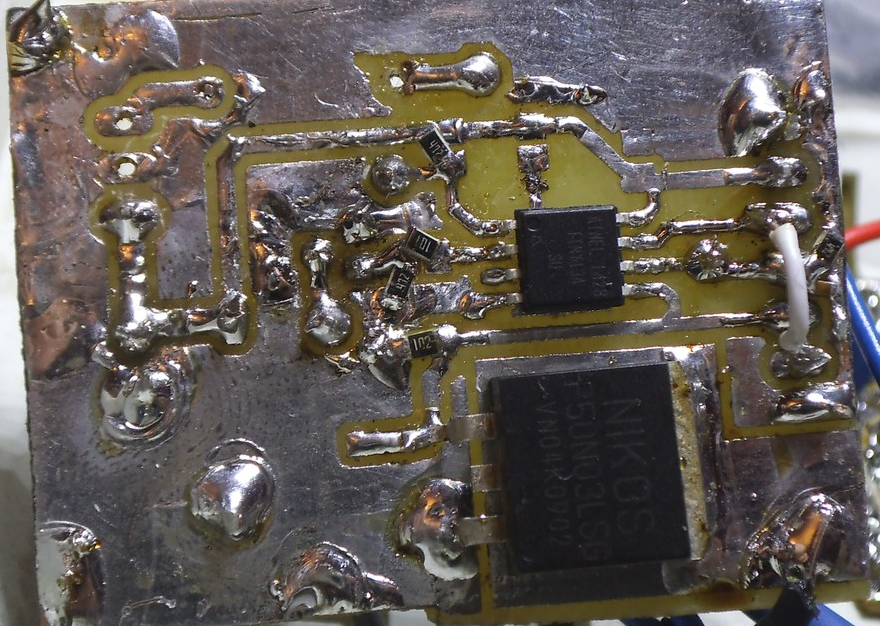
Przerwanie realizowane jest w wektorze ISR(TIM0\_COMPA\_vect).

|  |
| --- |
| ISR(TIM0\_COMPA\_vect)  {  timer++;  fan\_timer1++;  fan\_timer2++;  } |

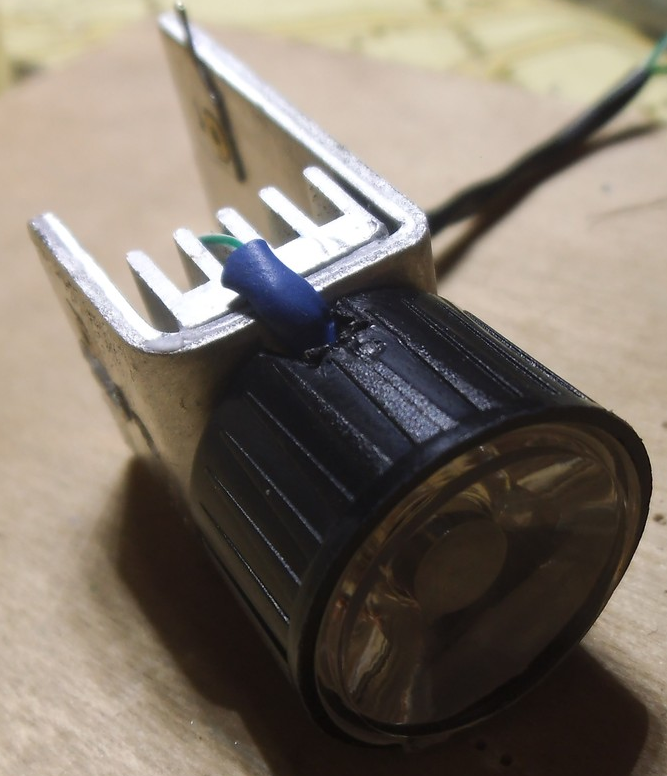
1. **Dokumentacja fotograficzna**

****

**Urządzenie w środku, rozmieszczenie elementów**



**PCB źródła prądowego**



**Zastosowana dioda LED wraz z soczewką skupiającą oraz radiatorem**