

Spis treści

1	Wstęp	2
2	Komunikacja z urządzeniami pomiarowymi na platformie Linux	4
2.1	Programowane urządzenia pomiarowe	4
2.2	Komunikacja	4
2.3	SCPI — standard komend do komunikacji z urządzeniami	5
2.4	Przykładowe programy do sterowania sprzętem	6
3	Wykorzystania oprogramowania open source w nauce.	7
3.1	Oprogramowanie open source	7
3.2	Python — idealne narzędzie dla fizyka	7
4	Py3LabDeviceIO — moduł do sterowania sprzętem laboratoryjnym	8
4.1	Opisz programu	8
4.2	Używanie programu	9
5	Program okienkowy do przeprowadzania pomiarów	10
5.1	Opisz programu	10
6	Eksperyment	11
6.1	Teoria	11
6.1.1	Prąd progowy	11
6.1.2	Sprawność	13
6.1.3	Wpływ temperatury chłodnicy lasera na jego parametry	13
6.2	Badanie lasera	14
6.2.1	Laser 980 nm	14
6.2.2	Laser VCSEL 850 nm	15
6.2.3	Laser krawędziowy 635 nm	15

Rozdział 1

Wstęp

Niniejsza praca dotyczy zakresu inżynierii oprogramowania sprzętu pomiarowego w celu wykorzystania go w badaniu charakterystyk laserów półprzewodnikowych w laboratorium fotoniki Politechniki Łódzkiej.

Głównym celem pracy jest przedstawienie wykorzystania systemu Linux oraz oprogramowania open source w badaniach naukowych na przykładzie stworzenia interfejsu pomiarowego w laboratorium fononiki do badania charakterystyk laserów półprzewodnikowych.

W ostatnich latach obserwuje się gwałtowny rozwój wykorzystania oprogramowania open source w codziennej pracy naukowej. Coraz większą popularność zdobywa język Python. Od dawna podstawowym systemem operacyjnym używanym przez naukowców są różne odmiany systemu Unix. Jest to spowodowane dostępnością wielu narzędzi (C, Python, Gnuplot) których naturalnym środowiskiem jest środowisko Linux, ułatwiającym pracę naukową. Inną zaletą środowiska Unix jest możliwością korzystanie z linii poleceń, która ułatwia wiele zadań. Szukając informacji o wykorzystaniu języka Python do komunikacji ze sprzętem pomiarowym można zauważyć pewną lukę, którą moja praca ma cel wypełnić. Korzystając z strony oraz dokumentacji firmy Thorlabs, której sprzęt jest używany w laboratorium fononiki, należy zauważyć brak programu do komunikacji ze sprzętem na platformie Linux. Dostępne są jedynie wysokopoziomowe API do systemu Windows oraz możliwość użycia LabVIEW. Minusów środowiska Windows nie sposób wymienić w kilku zdaniach. Program LabVIEW jest programem płatnym. Rozwiązaniem wszystkich problemów jest użycie środowiska Linux, gdzie wszystko jest plikiem, także sprzęt połączony przez usb z komputerem, dzięki czemu możemy się z nim komunikować używając standardu komend SCPI przez wykorzystanie wywołań systemowych. Dzięki temu mamy możliwość dostępu do wszystkich możliwych funkcji sprzętu pomiarowego bez ponoszenia kosztów. Umożliwia nam to sterowanie sprzętu za pomocą komputera oraz wizualizację i analizę danych w sposób, jaki potrzebujemy. A wszystko to dzięki połączeniu możliwości środowiska Linux oraz języka Python

Głównymi celami mojej pracy jest przedstawienie wykorzystania oprogramowania open source takiego jak Python, C/C++ oraz systemu Linux do stworzenia stanowiska pomiarowego w celu badania laserów półprzewodnikowych. Korzystając z tych technologii mam zamiar stworzyć interfejs pomiarowy na platformę Ubuntu w laboratorium

fotoniki.

Dzięki mojej pracy możliwe będzie wykonywanie w szybki sposób charakterystyk laserów półprzewodnikowych. Charakterystyki te dają nam ważne informacje o laserze, dzięki nim możliwe jest określenie prądu progowego dla laserów krawędziowych, określenie ich sprawności. Za pomocą mojego stanowiska pomiarowego możliwe także będzie badanie ile uzyskuje się mocy z lasera przy danej mocy aplikowanej.

Praca jest podzielona na dwie części: jedna składa się z opisu przygotowania eksperymentu, komunikacji oraz sterowaniem urządzeniami laboratoryjnymi za pomocą programu napisanego w języku Python. Druga część pracy opisuje badanie laserów półprzewodnikowych na podstawie danych uzyskanych za pomocą programu przedstawionego w pierwszej części programu. Do wykreślenia charakterystyk wyjściowych oraz wyznaczenie sprawności badanych laserów używam skryptów napisanych w języku Python.

Rozdział 2

Komunikacja z urządzeniami pomiarowymi na platformie Linux

2.1 Programowane urządzenia pomiarowe

Przez programowane urządzenia pomiarowe rozumiemy sprzęt mogący dokonywać pomiarów wielkości elektrycznych i nieelektrycznych, który wyposażony jest w interfejs umożliwiający sterowanie nimi przy pomocy komputera. Przykładami takich urządzeń, którymi zajmuje się w swojej pracy są:

- Zasilacza diód laserowych firmy Thorlabs model LDC4005.
- Miernik mocy firmy Thorlabs model PM100.

Z wyżej wymienionymi urządzeniami możliwa jest fizyczna komunikacja za pomocą interfejsu USB przy pomocy standardu komend SCPI, który zostanie opisany w dalszej części rozdziału.

2.2 Komunikacja

W systemach Unix z którego dziedziczy system Linux, wszystko jest plikiem. Linuksowy sterownik znakowy (ang. *char driver*) pozwala na reprezentowanie urządzenia za pomocą specjalnych plików wirtualnych, które znajdują się w przestrzeni użytkownika w katalogu `/dev/ < nazwa >`. Obsługa tych plików możliwa jest za pomocą wywołań systemowych (ang. *system call*), które stanowią API za pomocą którego użytkownik może sterować sprzętem. Podstawowe wywołania systemowe pozwalające na sterowanie sprzętem to:

- `open` — służy do połączenia z urządzeniem, zwraca deskryptor pliku.
- `write` — funkcja służąca do wysyłania komend do urządzenia .
- `read` — funkcja służąca do odczytywania buffora urządzenia.
- `close` — funkcja zamykająca połączenie.

Funckje te mają swoją implementację w języku C w bibliotece `<fcntl.h>`, oraz w języku Python w bibliotece `os`.

2.3 SCPI — standard komend do komunikacji z urządzeniami

SCPI (ang. *Standard Commands for Programmable Instruments*) jest tekstowym interfejsem ASCII do programowanych urządzeń pomiarowych mający na celu standaryzację poleceń używanych w systemach pomiarowych. Zdefiniowany został 1990 roku, wedle specyfikacji IEEE 488.2. (Institute of Electrical and Electronics Engineers, międzynarodowa organizacja stowarzyszeń inżynierów elektryków i elektroników. Dzięki temu możliwa jest obsługa tych urządzeń przy wykorzystaniu komputera. Polecenia SCPI są to ciągi tekstowe ASCII, które są wysyłane do urządzenia. Polecenia są serią jednego lub więcej słów, przy czym wiele z nich używa dodatkowych parametrów. Odpowiedzi do zapytania polecenia są zazwyczaj ciągami ASCII. W przypadku danych masowych mogą być używane także formaty binarne.

Cechą poleceń wspólnych (ang. common) jest ich implementacja przez każde urządzenie. Czyli na przykład to samo polecenie będzie działać na każdym oscyloskopie bez względu na producenta. Można wyróżnić dwie grupy poleceń:

- Polecenia dla każdego urządzenia pomiarowego nie zależnie od jego przeznaczenia. Takimi komendami są m.in.
 - `*idn?` — odczytuje identyfikator urządzenia.
 - `*rst` — powoduje przywrócenie ustawień początkowych urządzenia.
 - `*cls` — powoduje wyzerowanie informacji o błędach.
 - `*opc?` — (ang. operation complete) zapytanie o zakończenie wykonania poprzedzających poleceń.
W odpowiedzi na zapytanie po zakończeniu wykonywania poprzedzających poleceń urządzenie prześle wartość 1.
 - `*wai` — (ang. wait) oczekiwanie na zakończenie wykonania poprzedzających poleceń.
- Polecenia charakterystyczne dla danego urządzenia pomiarowego zgodnie z jego przeznaczeniem. Przykładowe polecenie które będzie działać na każdym zasilaczu korzystającym z standardu SCPI:
 - Służące do ustawienia wartości prądu na 0.01 A
`SOURce : CURRent : LEVel : AMPLitude 0.01`

Fizyczne łącze komunikacyjne nie jest zdefiniowane przez SCPI. Stworzony standard IEEE-488 był dla GPIB, ale może być również używany z interfejsem RS-232, Ethernet, USB, VXIbus.

2.4 Przykładowe programy do sterowania sprzętem

Przykładowy program w języku Python do zapytania sprzętu o jego nazwę.

```
1 import os
2
3
4 class IODevice:
5     def __init__(self, path_to_device):
6         self.path_to_device = path_to_device
7         self.file_descriptor = os.open(path_to_device, os.O_RDWR | O_NOCTTY
8     )
9
10    def write(self, command):
11        os.write(self.file_descriptor, command)
12
13    def read(self, length=4000):
14        return os.read(self.file_descriptor, length)
15
16    def close(self):
17        os.close(self.file_descriptor)
18
19 device = IODevice("/dev/usbtlmc0")
20 device.write("*IDN?")
21 print(device.read())
```

Przykładowy program w języku C do zapytania sprzętu o jego nazwę.

```
1 #include <errno.h>
2 #include <fcntl.h>
3 #include <unistd.h>
4 #include <stdio.h>
5
6 int main()
7 {
8     int fd = open("/dev/usbtlmc0", O_RDWR | O_NOCTTY);
9     if (fd == -1) {
10         perror("open");
11         exit(EXIT_FAILURE);
12     } else {
13         write(fd, "*IDN?", 100);
14         char buffer[128];
15         read(fd, buffer, 128);
16         printf(buffer);
17     }
18 }
```

Rozdział 3

Wykorzystania oprogramowania open source w nauce.

3.1 Oprogramowanie open source

Otwarte oprogramowanie (ang. *open source*) jest odłamem ruchu darmowego oprogramowania (ang. *free software*), którego głównym celem jest możliwość swobodnego dostępu do oprogramowania dla wszystkich. Jawny musi być kod źródłowy. Przykładami takiego oprogramowania które używam w swojej pracy jest:

- System Linux
- Język programowania Python

3.2 Python — idealne narzędzie dla fizyka

Rozdział 4

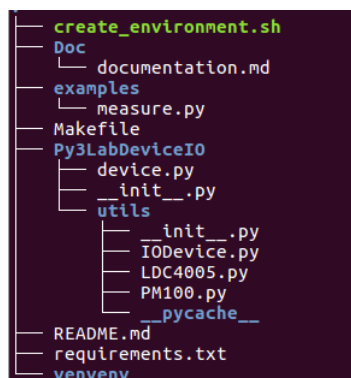
Py3LabDeviceIO — moduł do sterowania sprzętem laboratoryjnym

4.1 Opis programu

PyLabDeviceIO jest modulem napisanym w języku Python umożliwiającym sterowanie zasilaczem do lasera oraz miernikiem mocy. Program opiera się na wykorzystaniu wywołań systemowych oraz na komunikacji przez komendy SCPI.

Program składa się z czterech klas:

- `device.py` — główna klasa, zawiera funkcje: do sprawdzania dostępnych urządzeń, zwraca instancje danego urządzenia, co umożliwia korzystanie z jego funkcji.
- `IODevice.py` — klasa do operacji wejścia-wyjścia na programowalnych urządzeniach pomiarowych.
- `LDC4005.py` — klasa zawierająca funkcje od obsługi zasilacza diód laserowych Thorlabs 4005.
- `PM100.py` — klasa zawierająca funkcje do obsługi detektora mocy Thorlabs PM100.



Rysunek 4.1: Struktura programu.

4.2 Używanie programu

W celu dokonania pomiarów za pomocą programu Py3LabDeviceIO możliwe są dwie operacje:

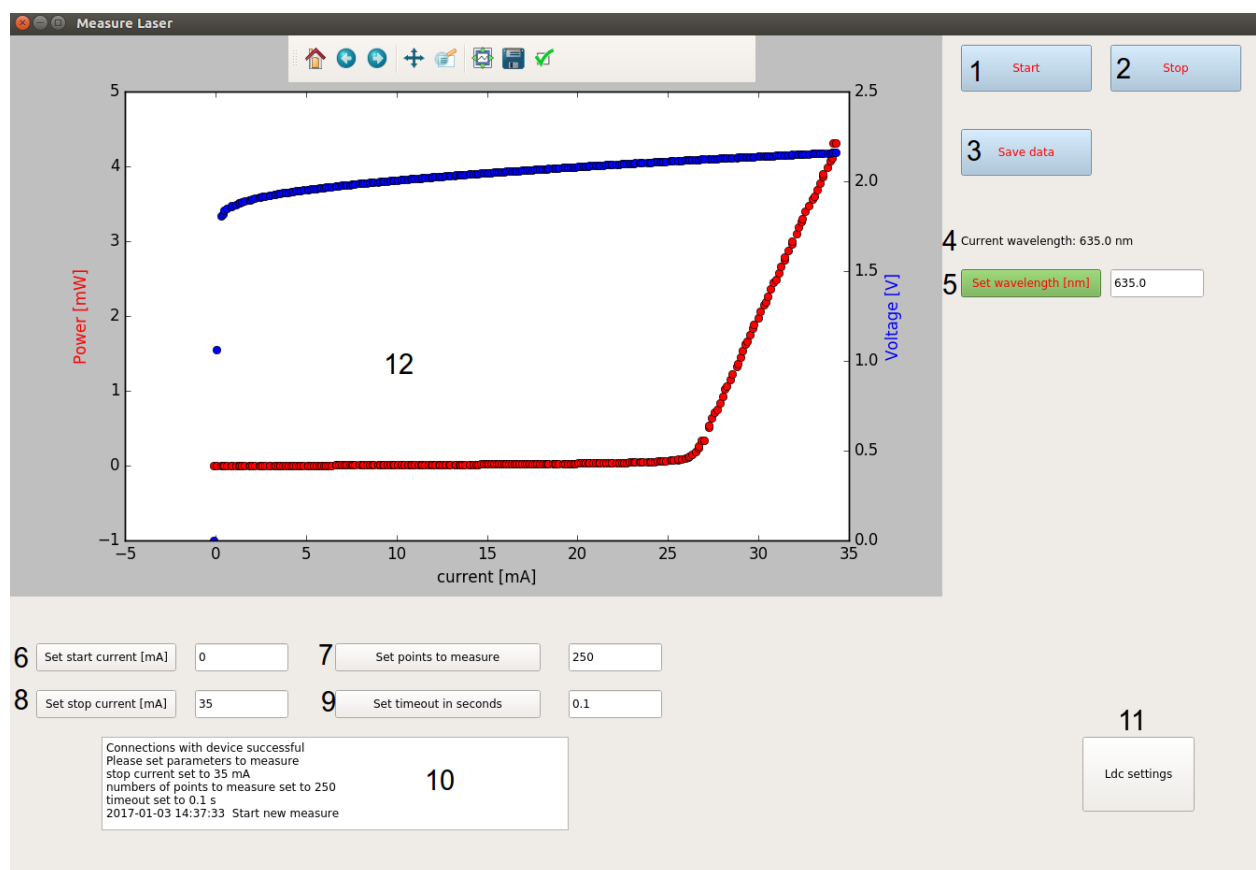
- wywołanie skryptu z linii poleceń z odpowiednimi parametrami.
- uruchomienie programu okienkowego.

Obie wymagają uprawnień użytkownika uprzywilejowanego (sudo).

Rozdział 5

Program okienkowy do przeprowadzania pomiarów

5.1 Opis programu



Rysunek 5.1: 1 — rozpoczęcie pomiarów, 2 — zatrzymanie pomiarów, 3 — zapisanie danych pomiarowych, 4 — pokazuje długość fali detektora, 5 — zmiana długości fali detektora.

Rozdział 6

Eksperyment

6.1 Teoria

6.1.1 Prąd progowy

Wśród laserów półprzewodnikowych możemy wyróżnić lasery krawędziowe oraz lasery o emisji powierzchniowej z pionową wnęką rezonansową tzw. Lasery VCSEL (ang. *Vertical Cavity Surface Emitting Laser*) będące obiektem moich badań. Aby scharakteryzować lasery, można wykonać ich charakterystyki, które przedstawiają, jak zmienia się moc wyjściowa oraz napięcie lasera w funkcji zadanego prądu.

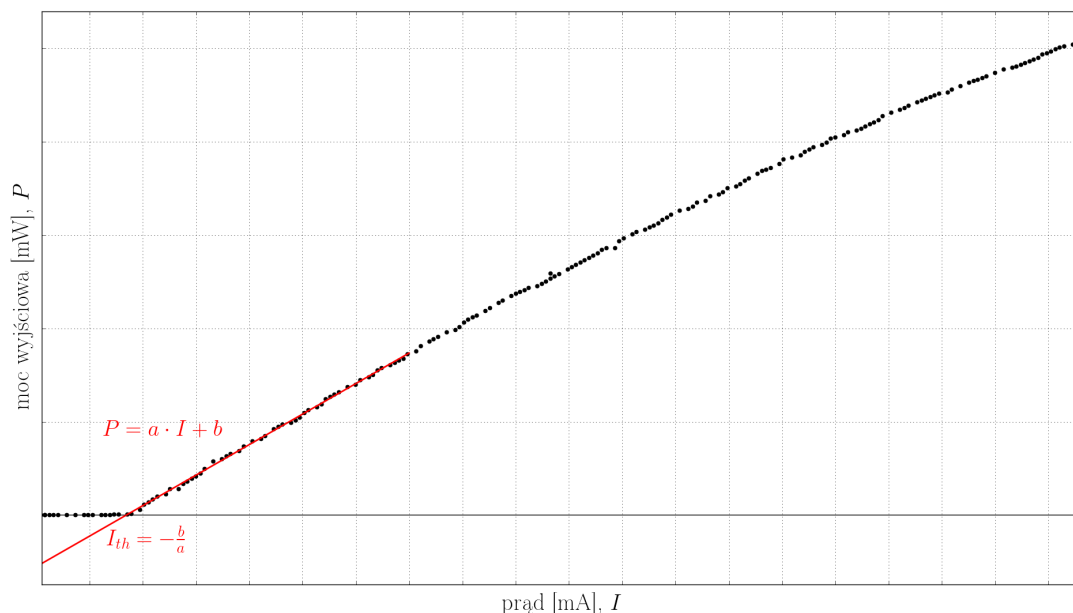
Ważnym parametrem laserów półprzewodnikowych jest prąd progowy (z ang. *threshold current*) który określa wartość prądu, przy którym zaczyna zachodzić akcja laserowa, czyli rośnie gwałtownie natężenie promieniowania i maleje szerokość linii emisyjnej. W celu wyznaczenia prądu progowego należy sporządzić wykres zależności mocy wyjściowej lasera od prądu zasilającego. Następnie dla prądu gdzie zaczyna się akcja laserowa dla odcinka liniowego należy metodą najmniejszych kwadratów przy użyciu wielomianu pierwszego stopnia znaleźć parametry krzywej. Dla wyznaczonej krzywej należy znaleźć miejsce zerowe, które będzie wyznaczonym prądem progowym.

$$P = a \cdot I + b \quad (6.1)$$

$$I_{th} = -\frac{b}{a} \quad (6.2)$$

$$\Delta I_{th} = \left| \frac{\partial I_{th}}{\partial a} \right| \cdot \Delta a + \left| \frac{\partial I_{th}}{\partial b} \right| \cdot \Delta b \quad (6.3)$$

$$\Delta I_{th} = \left| -\frac{b}{a^2} \right| \cdot \Delta a + \left| -\frac{1}{a} \right| \cdot \Delta b \quad (6.4)$$



Rysunek 6.1: Charakterystyki wyjściowe lasera krawędziowego 980 nm w różnych temperaturach.

Dla laserów krawędziowych zależności prądu progowego I_{th} od temperatury T możemy wyrazić za pomocą równania:

$$I_{th} = I_0 \exp\left(\frac{T}{T_0}\right) \quad (6.5)$$

Wartości parametrów I_0 oraz T_0 możemy wyznaczyć na podstawie charakterystyk emisyjnych lasera w różnych temperaturach T .

Przez zlogarytmowanie wartości prądu oraz podstawienie otrzymujemy:

$$\ln(I_{th}) = \frac{T}{T_0} + \ln(I_0) \quad (6.6)$$

Mając wartości prądu progowego w danej temperaturze można do nich dopasować funkcję liniową w postaci:

$$y = a \cdot T + b \quad (6.7)$$

Gdzie:

$$y = \ln(I_{th}) \quad (6.8)$$

$$a = \frac{1}{T_0} \quad (6.9)$$

$$b = \ln(I_0) \quad (6.10)$$

Na tej podstawie możemy znaleźć poszukiwane parametry I_0 oraz T_0 :

$$I_0 = e^b \quad (6.11)$$

$$T_0 = \frac{1}{a} \quad (6.12)$$

Korzystając z różniczki zupełnej można obliczyć wartości błędów wyznaczonych wartości:

$$\Delta I_0 = \left| \frac{\partial I_0}{\partial b} \right| \cdot \Delta b = |e^b| \cdot \Delta b \quad (6.13)$$

$$\Delta T_0 = \left| \frac{\partial T_0}{\partial a} \right| \cdot \Delta a = \left| -\frac{1}{a^2} \right| \cdot \Delta a \quad (6.14)$$

6.1.2 Sprawność

Innym ważnym parametrem, którym możemy scharakteryzować lasery półprzewodnikowe jest ich sprawność. Można wyróżnić następujące sprawności:

- Sprawność różniczkowa (ang. *slope efficiency*) — jest zdefiniowana jako nachylenie krzywej uzyskanej przez wykreślenie zależności mocy wyjściowej z lasera versus energii dostarczonej do lasera (natężenie prądu lub moc dostarczona).
- Sprawność całkowita (ang. *Wall-plug-efficiency*) — jest zdefiniowana jako stosunek mocy wyjściowej do całkowitej mocy wejściowej lasera.

6.1.3 Wpływ temperatury chłodnicy lasera na jego parametry

6.2 Badanie lasera

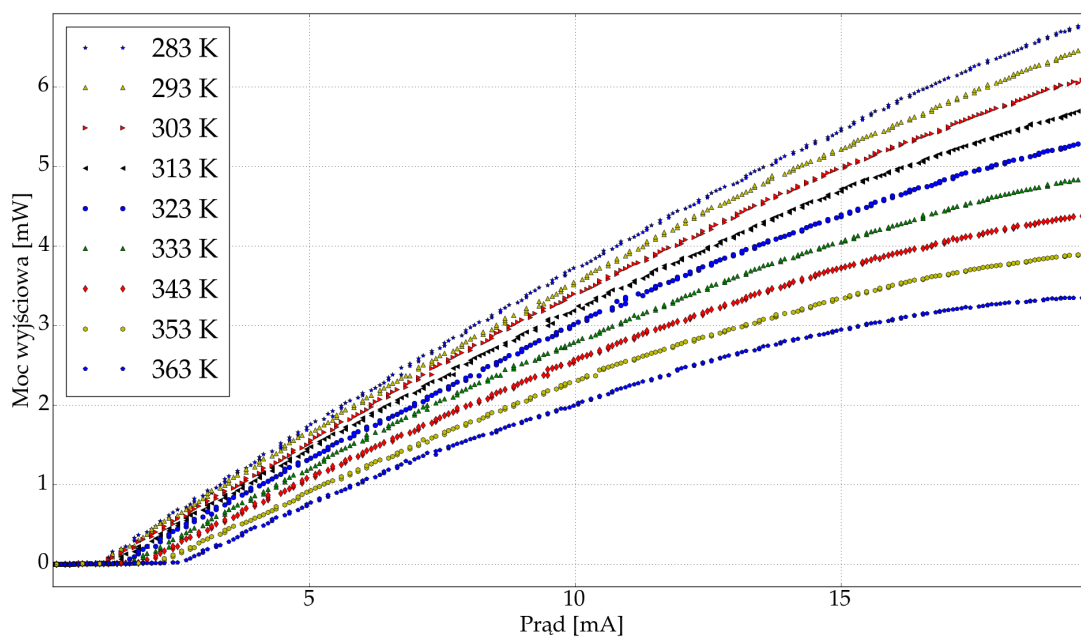
6.2.1 Laser 980 nm

Podpunkt ten zawiera wyniki pomiarów dla lasera VCSEL o emitowanej fali długości 980 nm. Pomiar polegał w pierwszej części na wykonaniu charakterystyk prądowo-napięciowych oraz prądowo-oświetleniowych lasera w temperaturze lasera od 283 K do 363 K krokiem co 10 K. Na podstawie otrzymanych charakterystyk wyznaczyłem wartość prądu progowego w danej temperaturze. Następnie korzystając z wyznaczonych wartości na podstawie wzorów (5.7) i (5.8) znalazłem parametry $I_0 = (89.9 \pm 0.1)$ mA oraz $T_0 = (0.04 \pm 0.01)$ K.

Tabela 1 zawiera wyznaczone wartości prądu progowego I_0 dla tego lasera. Rysunek 5.1 przedstawia charakterystykę wszystkich badanych laserów. Rysunki 5.2-5.29 przedstawiają charakterystyki lasera wraz z obliczonymi parametrami które służą do charakterystyki lasera. Rysunek 5.13 przedstawia wykres prądu progowego w funkcji temperatury.

Tabela 6.1: Wyznaczone wartości prądu progowego I_0 w różnych temperaturach T dla lasera krawędziowego 980 nm.

T [K]	I_{th} [mA]
283	0.92 ± 0.03
288	0.94 ± 0.03
293	0.98 ± 0.03
298	1.05 ± 0.04
303	1.10 ± 0.03
308	1.18 ± 0.03
313	1.23 ± 0.03
318	1.25 ± 0.03
323	1.36 ± 0.04
328	1.47 ± 0.03
333	1.59 ± 0.04
338	1.63 ± 0.04
343	1.76 ± 0.04
348	1.86 ± 0.06
353	2.07 ± 0.05
358	2.25 ± 0.06
363	2.43 ± 0.06

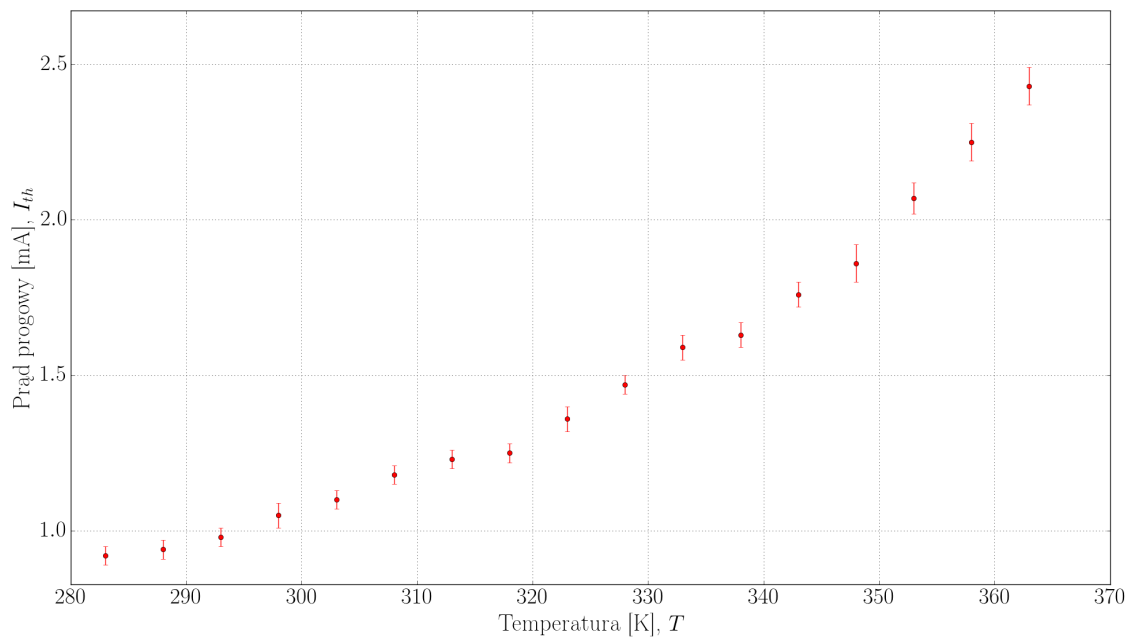


Rysunek 6.2: Charakterystyki wyjściowe lasera VCSEL 980 nm w różnych temperaturach.

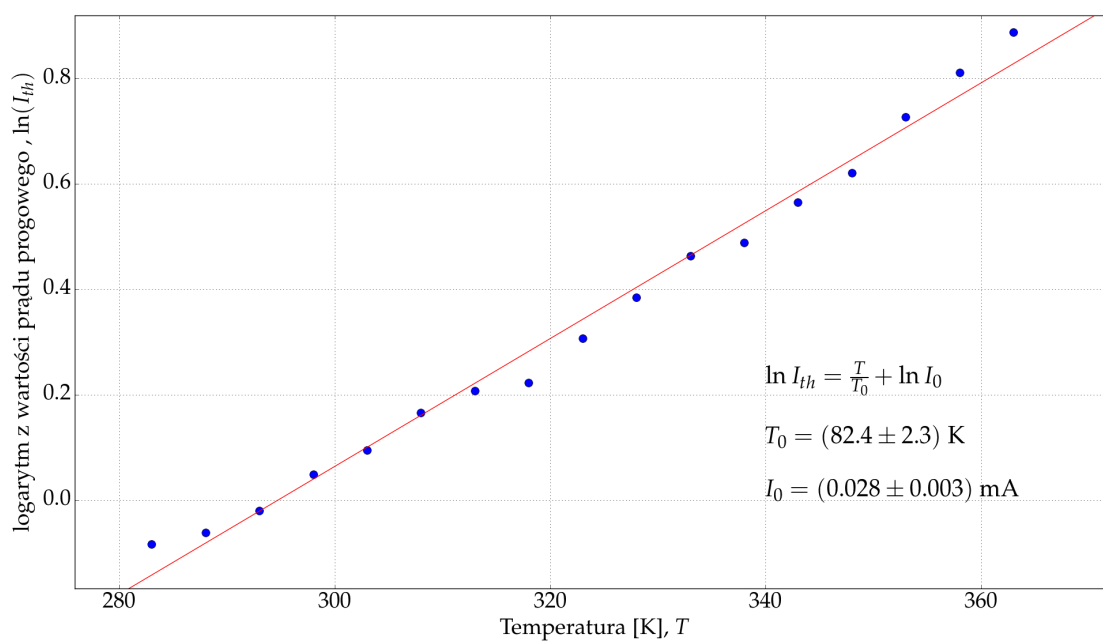
6.2.2 Laser VCSEL 850 nm

Laser 850

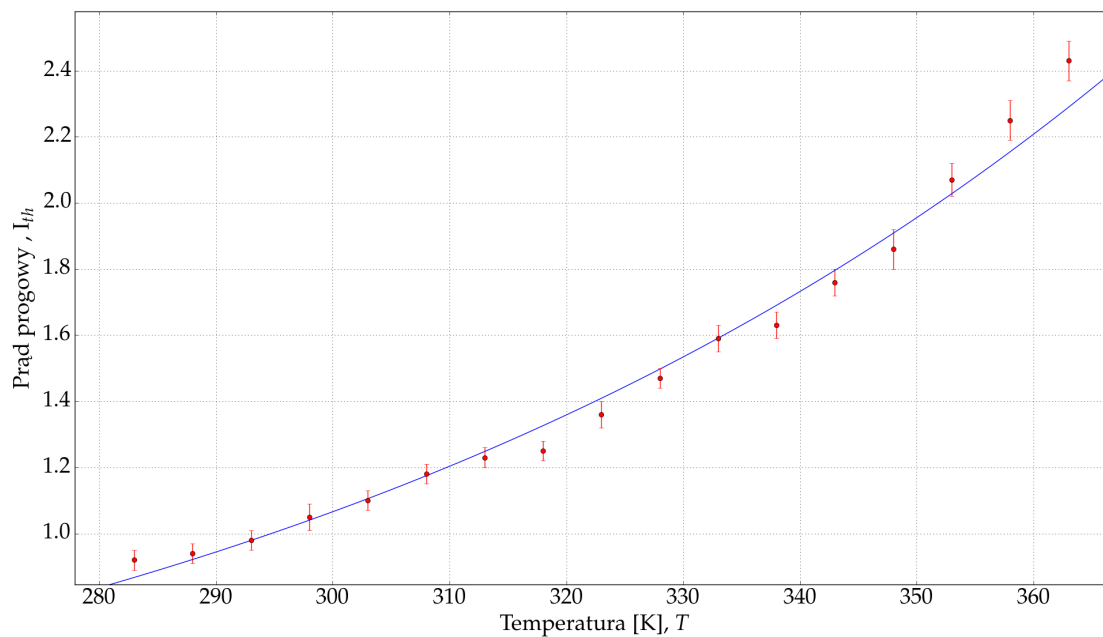
6.2.3 Laser krawędziowy 635 nm



Rysunek 6.3: Wykres zależności prądu progowego od temperatury dla lasera VCSEL 980 nm.



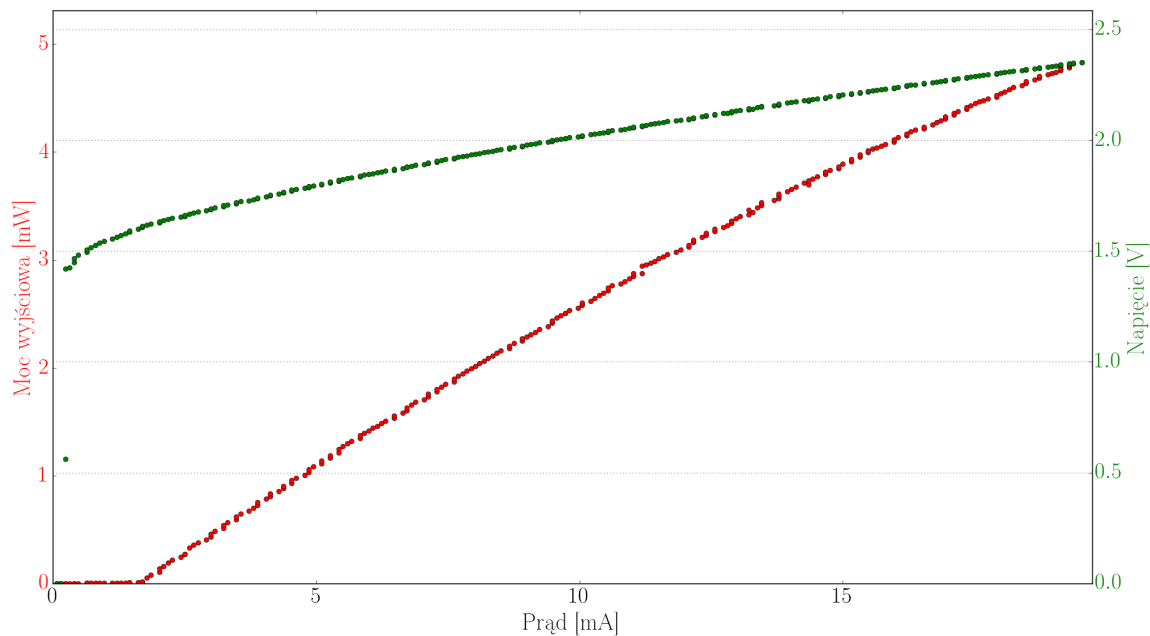
Rysunek 6.4: Wykres zależności prądu progowego od temperatury dla lasera VCSEL 980 nm.



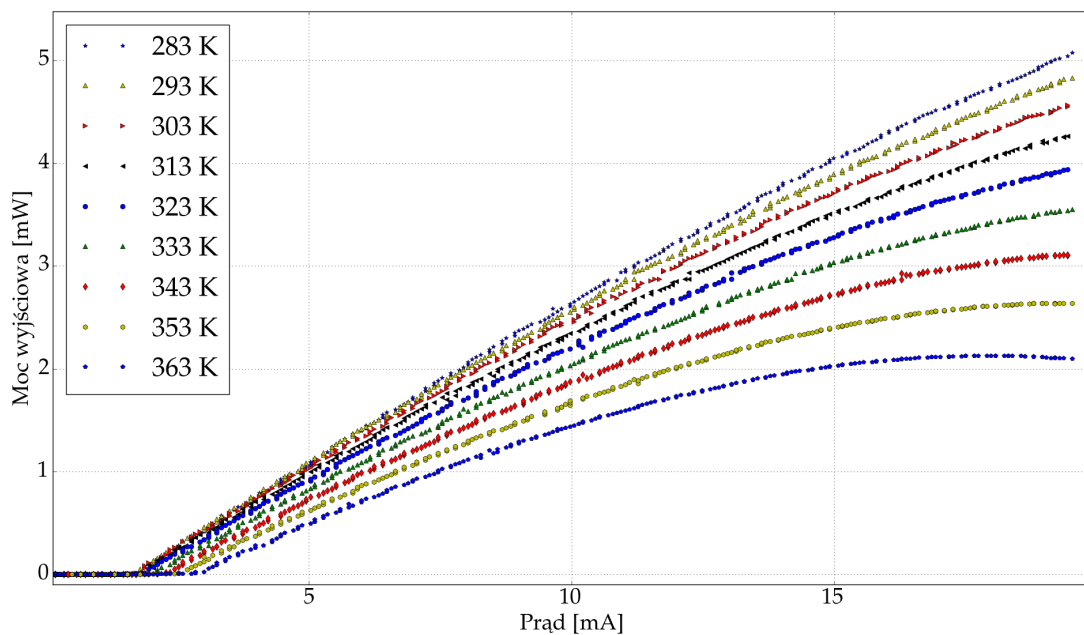
Rysunek 6.5: Wykres zależności prądu progowego od temperatury dla lasera VCSEL 980 nm. Wartości wyznaczone eksperymentalnie wraz z dopasowaną krzywą.

Tabela 6.2: Wyznaczone wartości prądu progowego I_0 w różnych temperaturach T dla lasera VCSEL 850 nm.

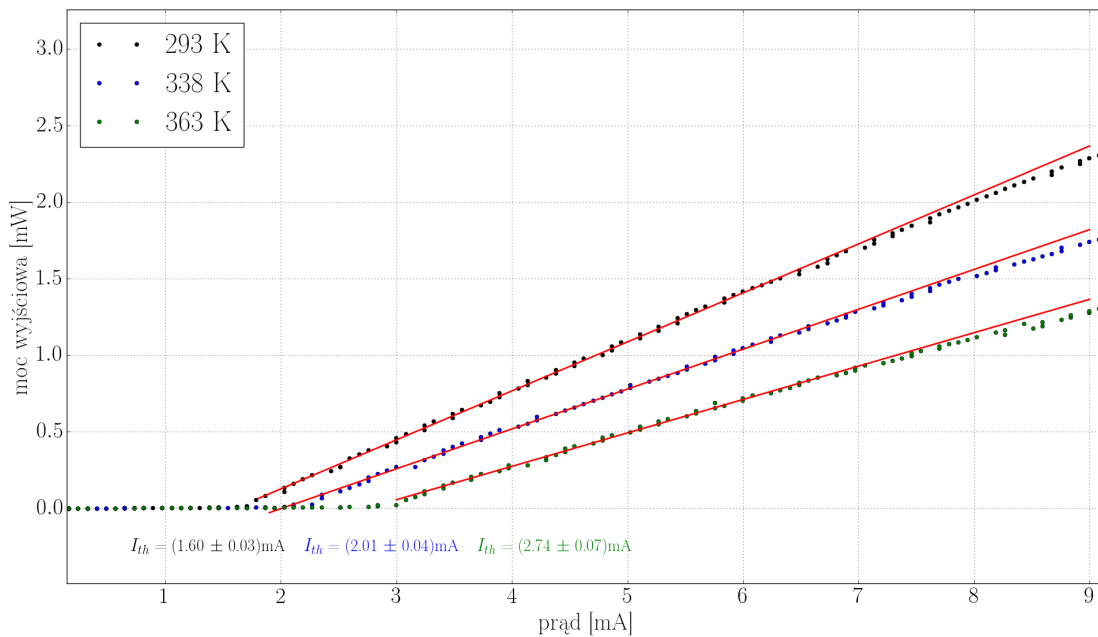
T [K]	I_{th} [mA]
283	1.70 ± 0.03
288	1.67 ± 0.03
293	1.60 ± 0.03
298	1.55 ± 0.04
303	1.59 ± 0.03
308	1.63 ± 0.03
313	1.65 ± 0.03
318	1.68 ± 0.04
323	1.73 ± 0.04
328	1.83 ± 0.04
333	1.89 ± 0.04
338	2.01 ± 0.04
343	2.14 ± 0.04
348	2.24 ± 0.05
353	2.38 ± 0.05
358	2.57 ± 0.05
363	2.74 ± 0.07



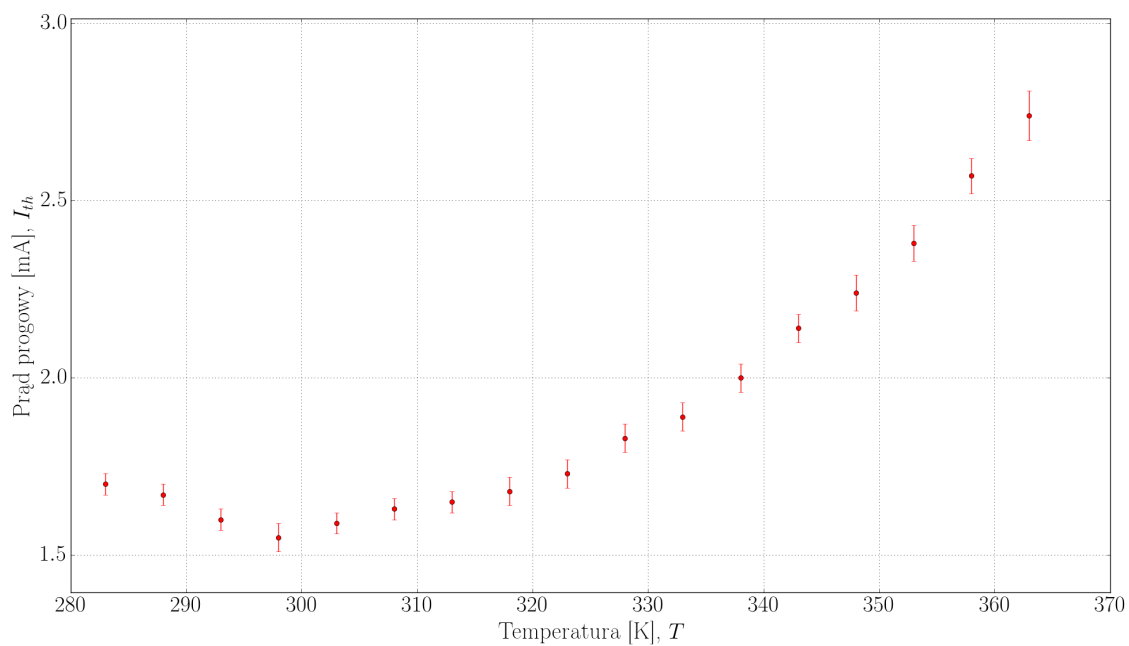
Rysunek 6.6: Charakterystyki wyjściowe lasera VCSEL 850 nm w temperaturze 293 K.



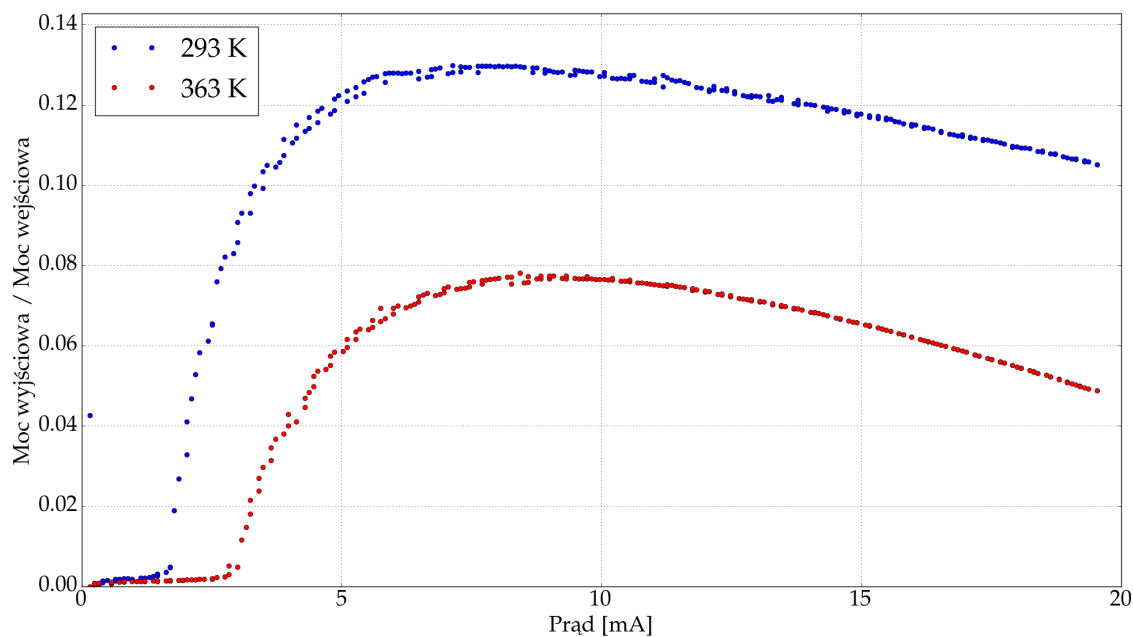
Rysunek 6.7: Charakterystyki wyjściowe lasera VCSEL 850 nm w różnych temperaturach.



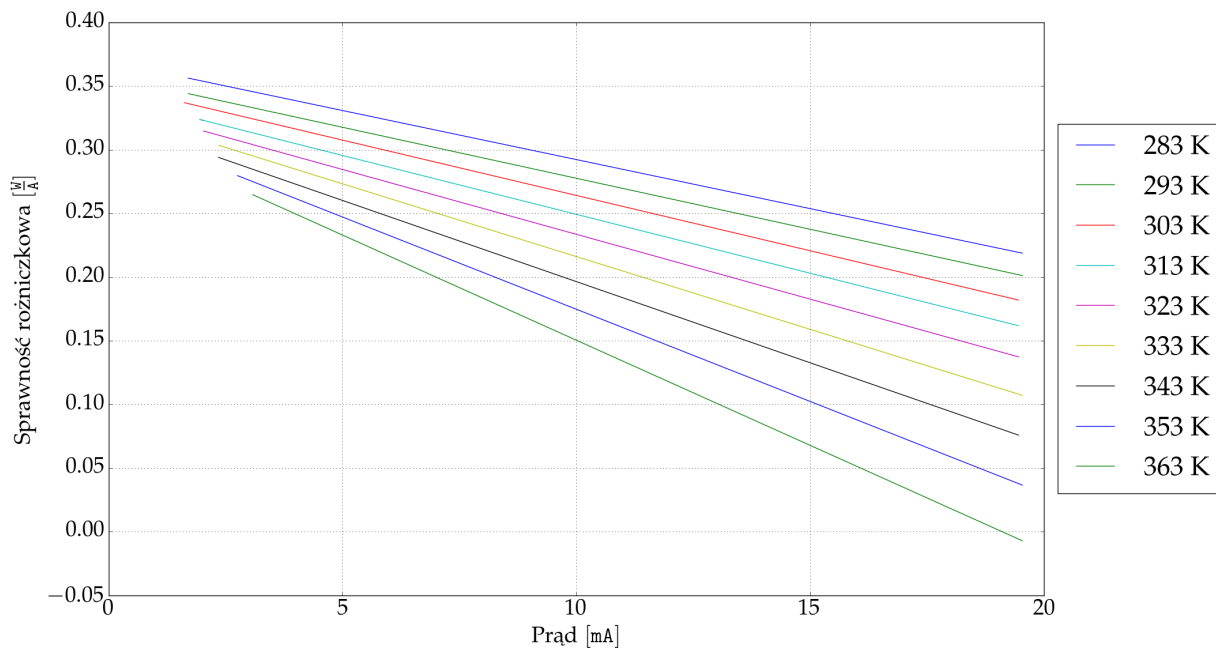
Rysunek 6.8: Charakterystyki wyjściowe lasera VCSEL 850 nm z wyznaczonym prądem progowym I_{th} w różnych temperaturach.



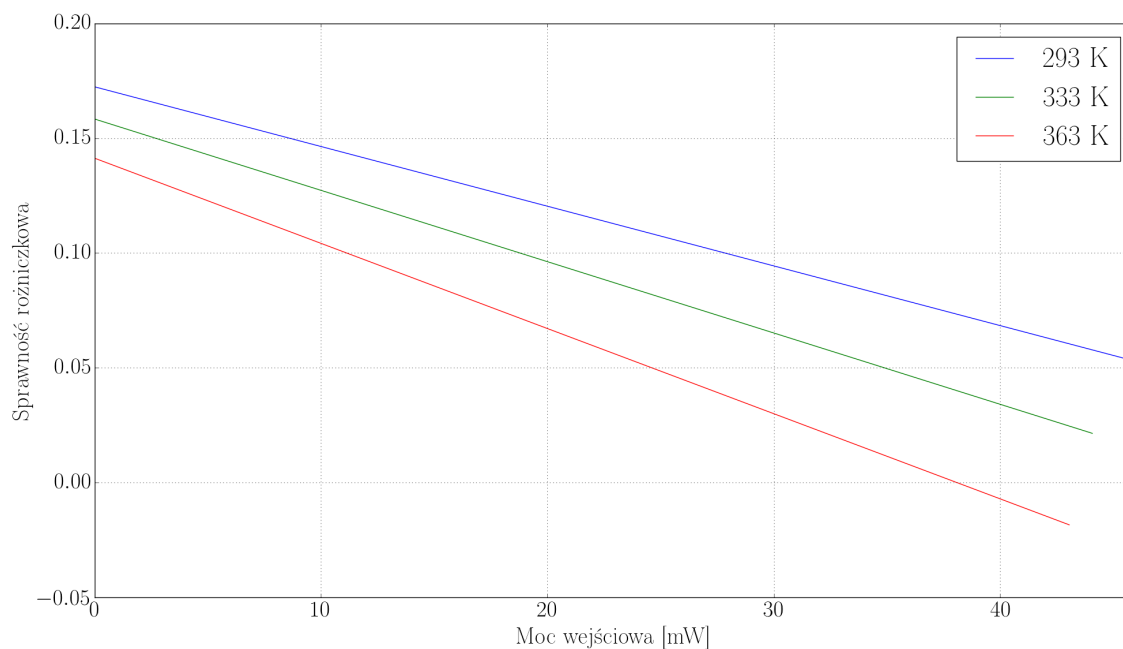
Rysunek 6.9: Wykres prądu progowego w zależności od temperatury dla lasera VCSEL 850 nm.



Rysunek 6.10: Wykres sprawności lasera VCSEL 850 nm.



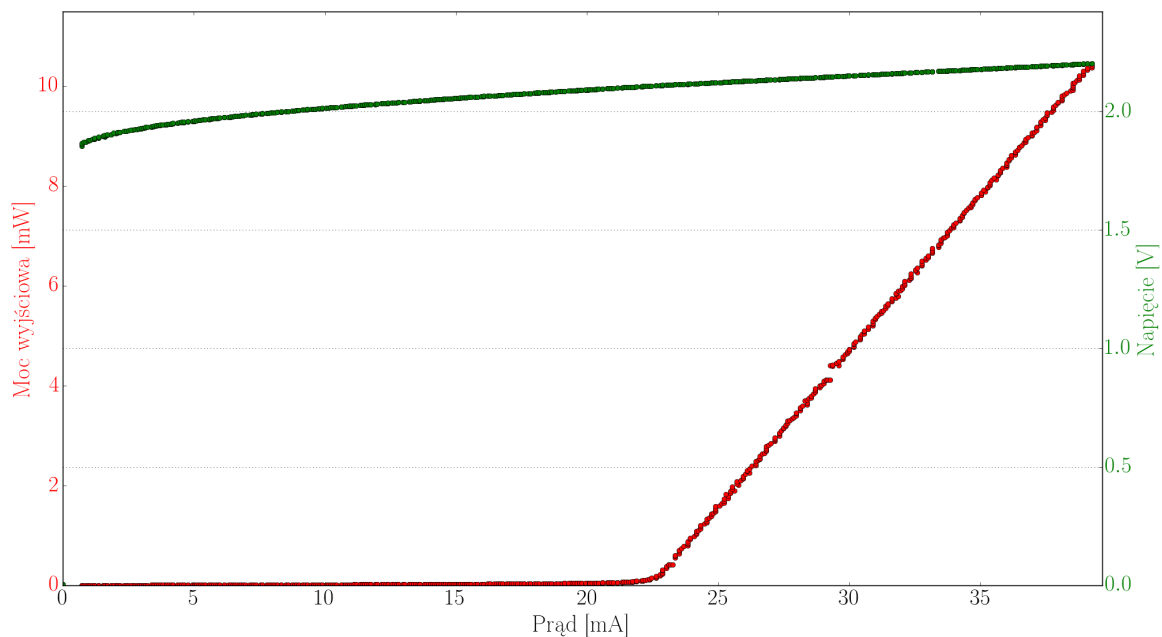
Rysunek 6.11: Wykres sprawności różniczkowej lasera VCSEL 850 nm w funkcji prądu.



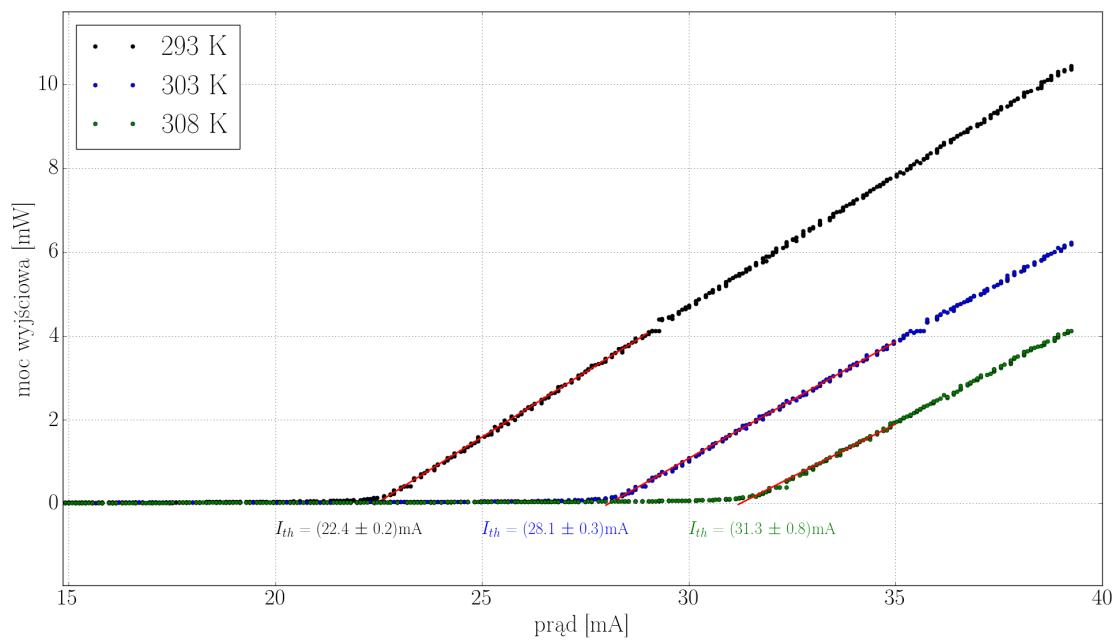
Rysunek 6.12: Wykres sprawności różniczkowej lasera VCSEL 850 nm w funkcja mocy wejściowej.

Tabela 6.3: Wyznaczone wartości prądu progowego I_0 w różnych temperaturach T dla lasera krawędziowego 635 nm.

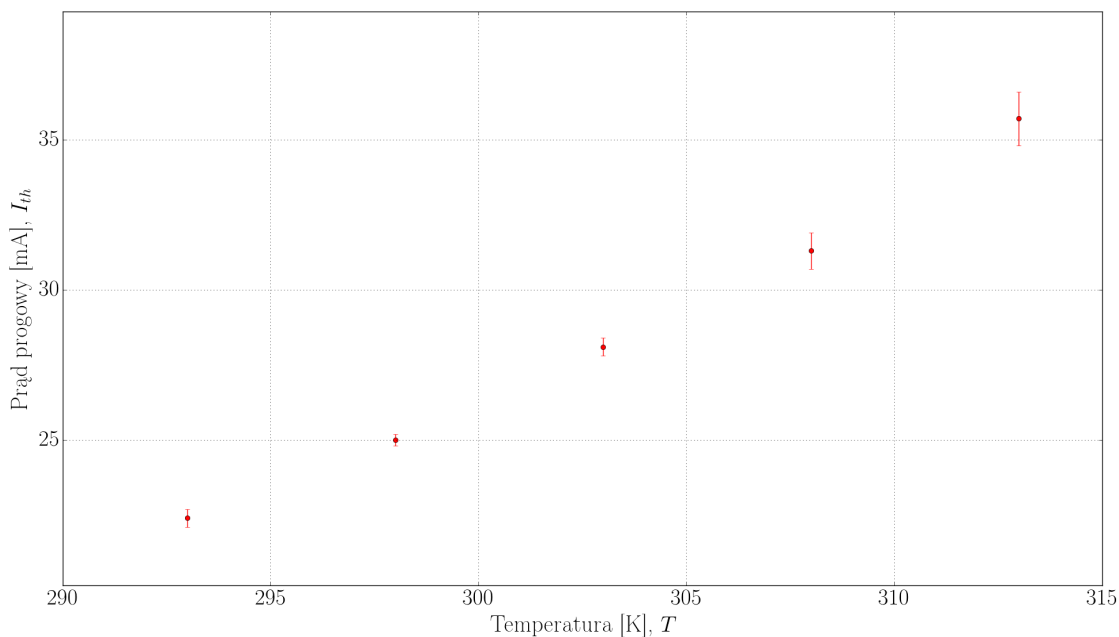
T [K]	I_{th} [mA]
293	22.4 ± 0.3
298	25.0 ± 0.2
303	28.1 ± 0.3
308	31.3 ± 0.6
313	35.7 ± 0.9



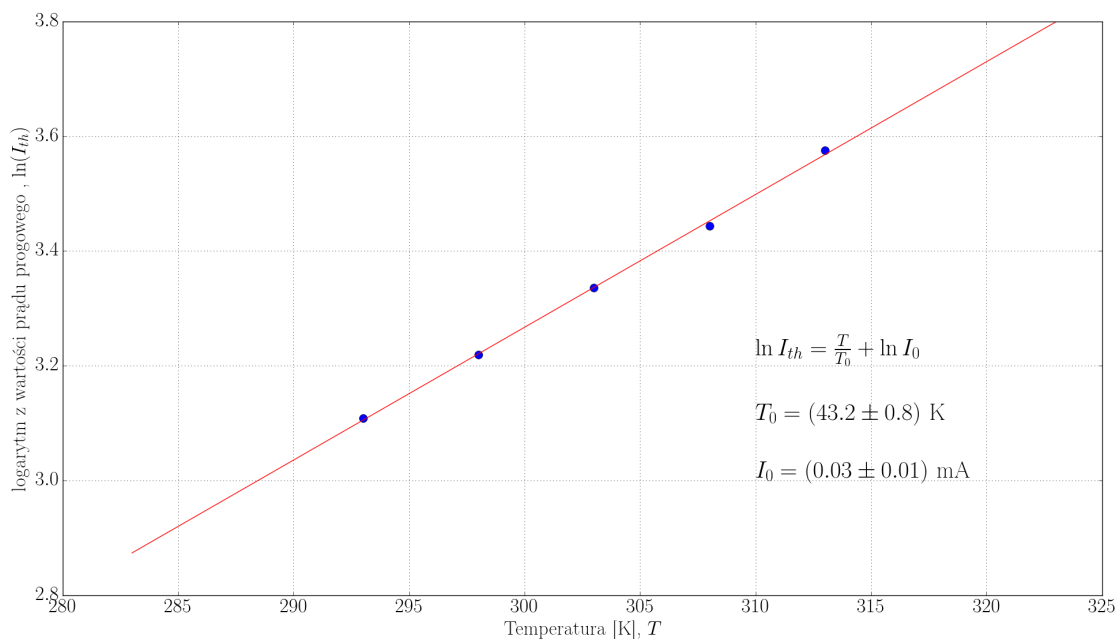
Rysunek 6.13: Charakterystyka wyjściowa lasera krawędziowego 635 nm.



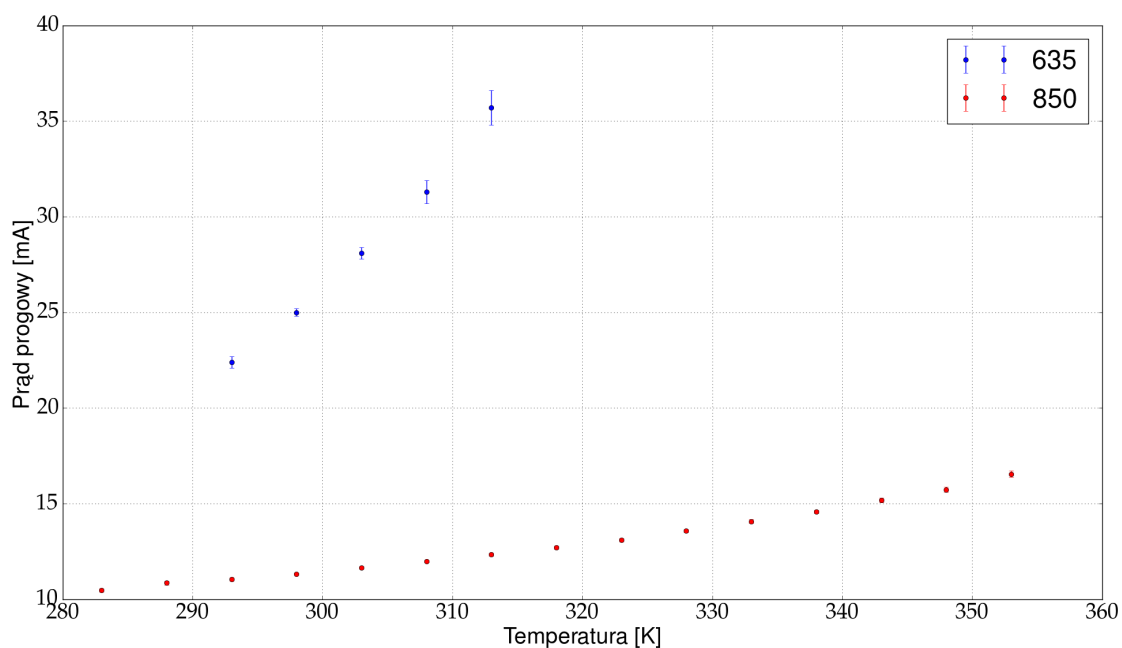
Rysunek 6.14: Charakterystyka wyjściowej lasera krawędziowego 635 nm wraz z wyznaczonym prądem progowym dla różnych temperatur.



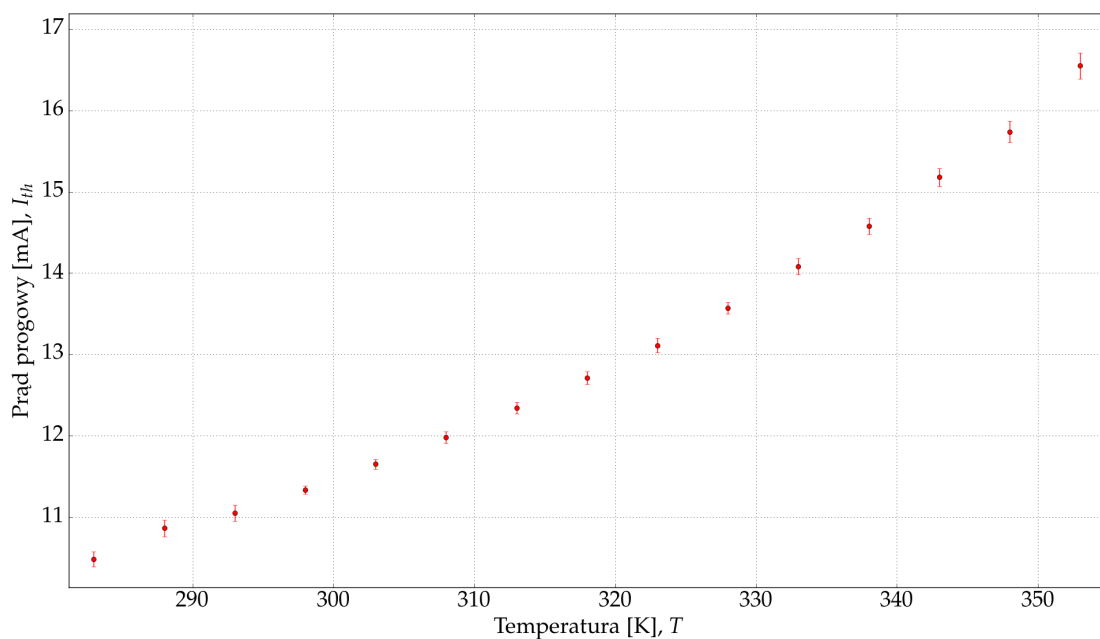
Rysunek 6.15: Wykres logarytmu z prądu progowego w zależności od temperatury chłodnicy wraz z wyznaczonymi parametrami I_0 i T_0 dla lasera krawędziowego 635 nm.



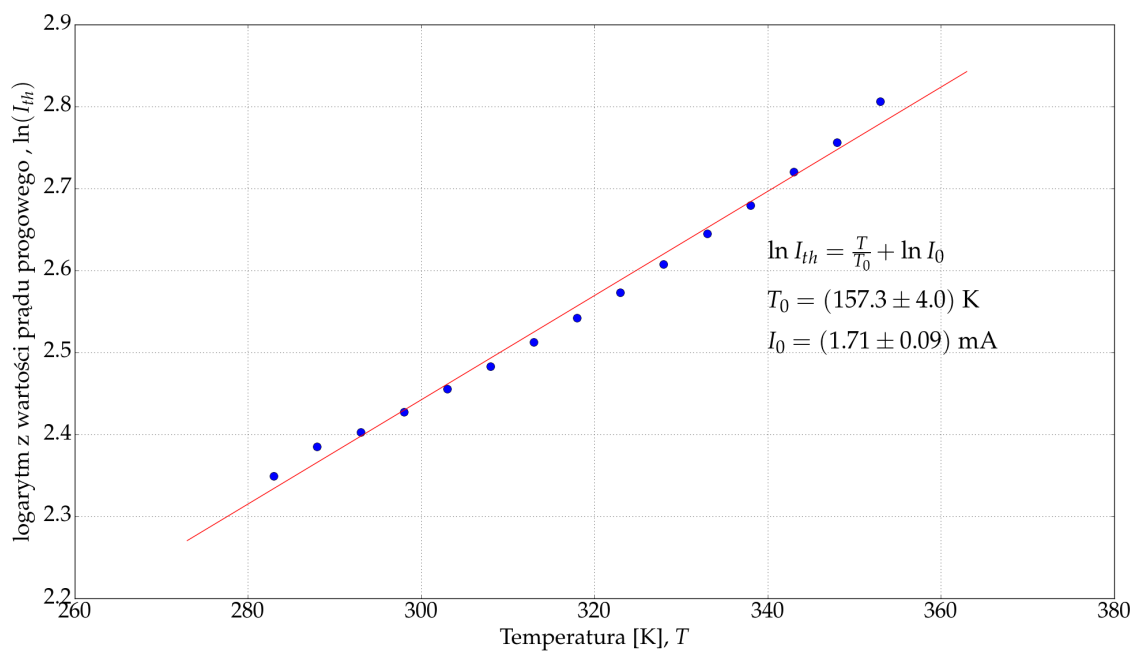
Rysunek 6.16: Wykres logarytmu z prądu progowego w zależności od temperatury chłodnicy wraz z wyznaczonymi parametrami I_0 i T_0 dla lasera krawędziowego 635 nm.



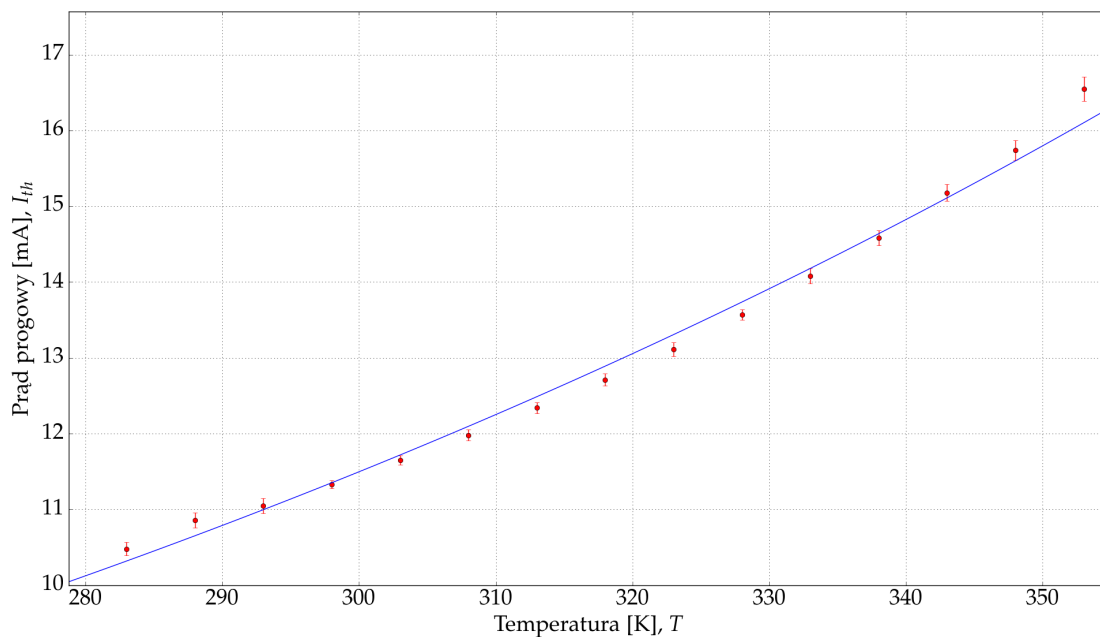
Rysunek 6.17: Wykres logarytmu z prądu progowego w zależności od temperatury wraz z wyznaczonymi parametrami I_0 i T_0 dla laserów krawędziowych.



Rysunek 6.18: Wykres zależności prądu progowego od temperatury dla lasera krawędziowego 850 nm.



Rysunek 6.19: Wykres logarytmu z prądu progowego w zależności od temperatury wraz z wyznaczonymi parametrami I_0 i T_0 dla lasera krawędziowego 850 nm.



Rysunek 6.20: Wykres prądu progowego od temperatury.