Spis treści

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 dawana 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 ponoszenie kosztów. Umożliwia nam to sterowania sprzętu za pomocą komputera oraz wizualizacje i analizę danych w sposób, jaki potrzebujemy. A wszystko to dzięki połączeniu możliwości środowiska Linux oraz języka Python

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

Dzięki mojej pracy możliwe będzie wykonywanie w szybki sposób charakterystyk laserów półprzewnodnikowcyh. 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.

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 Thorlas firmy Thorlabs model PM100.

Z wyżej wymienionymi urządzeniami możliwa jest fizyczna komunikacja za pomocą interfejsu USB przy pomocy standardu komend SCPI, który zostanie opiszany w dalszej cześci 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 użytkowniki może sterować sprzętem. Podstawowe wywołaniami systemowymi pozwalające na sterowanie sprzętem to:

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

Funckje te mają swoją implementacje w języku C w bibliotecze $<$ `fcntl.h` $>$, oraz w języku Python w bibliotecze `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 standaryzacje polecenie używanych w systemach pomiarowym. 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 użytawane także formaty binarne.

Cechą poleceń wspólnych (ang. common) jest ich implementacja przez każde urządzenie. Czyli naprzykład 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 przeznacznia. Takimi komendami są m.in.
  - `*idn?` — odczytuje identyfikator urządzenia.
  - `*rst` — powduje przywrócenie ustawień początkowych urządzenia.
  - `*cls` — powduje 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:
  - Służacę do ustawienie 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 nazwe.

```python
import os


class IODevice:
    def __init__(self, path_to_device):
        self.path_to_device = path_to_device
        self.file_descriptor = os.open(path_to_device, os.O_RDWR | O_NOCITY
    )

    def write(self, command):
        os.write(self.file_descriptor, command)

    def read(self, length=4000):
        return os.read(self.file_descriptor, length)

    def close(self):
        os.close(self.file_descriptor)

device = IODevice("/dev/usbtmc0")
device.write("*IDN?")
print(device.read())
```

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

```c
#include<errno.h>
#include<fcnl.h>
#include<unistd.h>
#include<stdio.h>

int main()
{
    int fd = open("/dev/usbtmc0", O_RDWR | O_NOCTTY);
    if(fd == -1) {
        perror("open");
        exit(EXIT_FAILURE);
    } else {
        write(fd, "*IDN?", 100);
        char buffor[128];
        read(fd, buffor, 128);
        printf(buffor);
    }
}
```

Rozdział 3

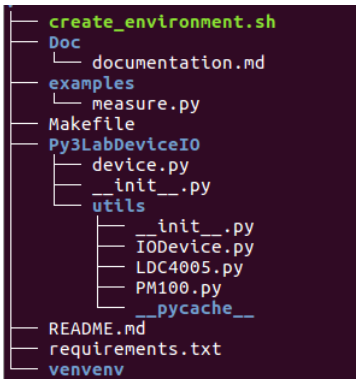
Py3LabDeviceIO — moduł do sterowania sprzetem laboratoryjnym

3.1 Opisz programu

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

Program składa sie z czterech klas:

- `device.py` — główna klasa, zawiera funkcje: do sprawdzania dostępnych urządzeń, zwraca instancje danego urządzenia, co umażliwa 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 3.1: Strukura programu.

3.2 Używanie programu

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

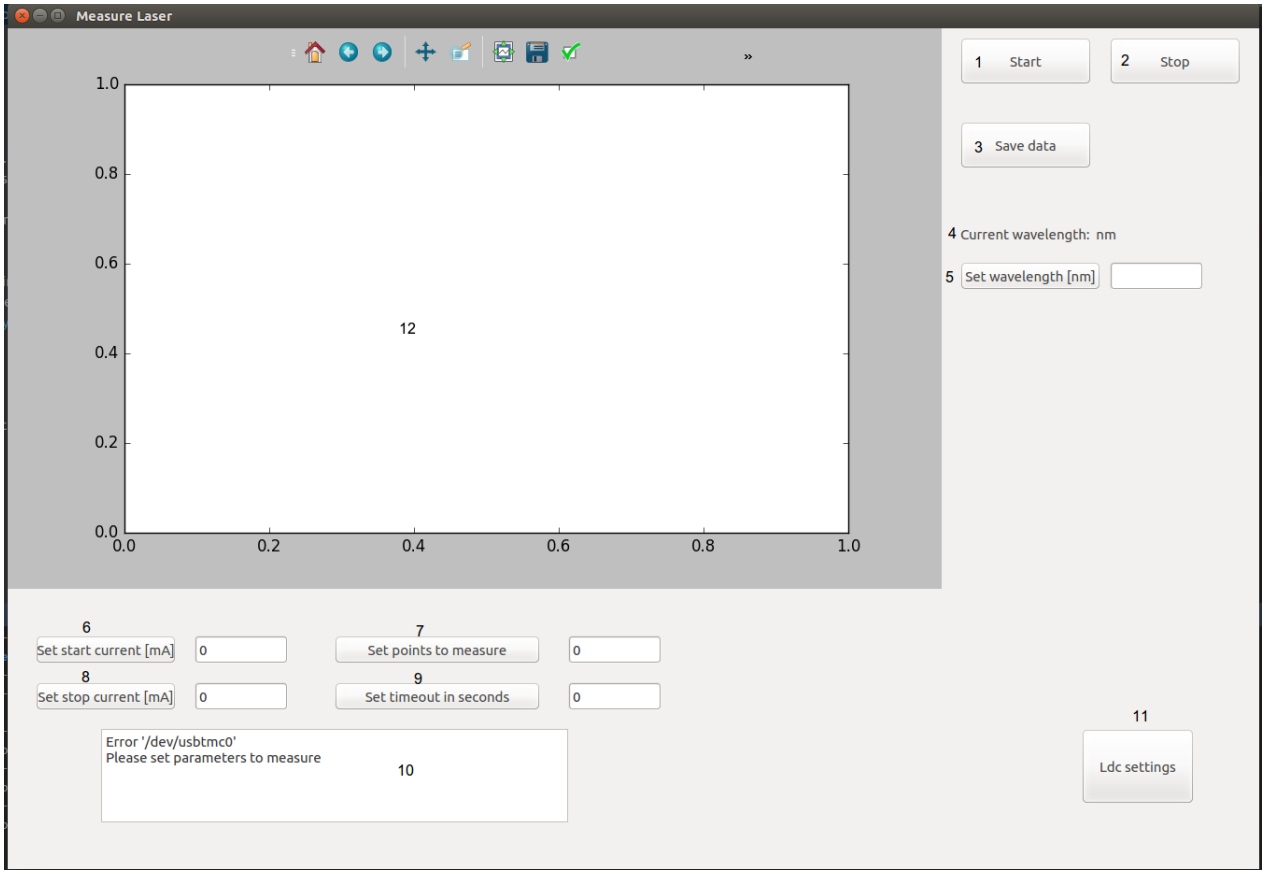
- wywołanie skryptu z linii poleceń z odpowienimi parametrami.
- uruchomienie programu okienkowego.

Obie wymagają uprawnień użytkownika uprzywijowanego (sudo).

Rozdział 4

Gui

4.1 Gui



Rysunek 4.1: 1 — rozpoczecie pomiarów, 2 — zatrzymanie pomiarów, 3 — zapisanie danych pomiarowych, 4 — pokazuje długość fali detektora, 5 — zmiana długości fali detektora.

Rozdział 5

Eksperyment

5.1 Teoria

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 krawędziowych 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. Zależności prądu progowego I_{th} od temperatury T możemy wyrazić za pomocą równania:

$$I_{th} = I_0 \exp\left(\frac{T}{T_0}\right) \tag{5.1}$$

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:

$$\ln(I_{th}) = \frac{T}{T_0} + \ln(I_0) \tag{5.2}$$

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

$$y = a \cdot T + b \tag{5.3}$$

Gdzie:

$$y = \ln(I_{th}) \tag{5.4}$$

$$a = \frac{1}{T_0} \tag{5.5}$$

$$b = \ln(I_0) \tag{5.6}$$

Na tej podstawie możemy znaleźć poszukiwane parametry I_0 oraz T_0:

$$I_0 = \mathrm{e}^b \tag{5.7}$$

$$T_0 = \frac{1}{a} \tag{5.8}$$

Korzystając z różniczki zupełnej można obliczyć wartości błędów wyznaczonych wartości:

$$\Delta I_0 = \left|\frac{\partial I_0}{\partial b}\right| \cdot \Delta b = |b\mathrm{e}^b| \cdot \Delta b \tag{5.9}$$

$$\Delta T_0 = \left|\frac{\partial T_0}{\partial a}\right| \cdot \Delta a = \left|-\frac{1}{a^2}\right| \cdot \Delta a \tag{5.10}$$

5.2 Badanie lasera

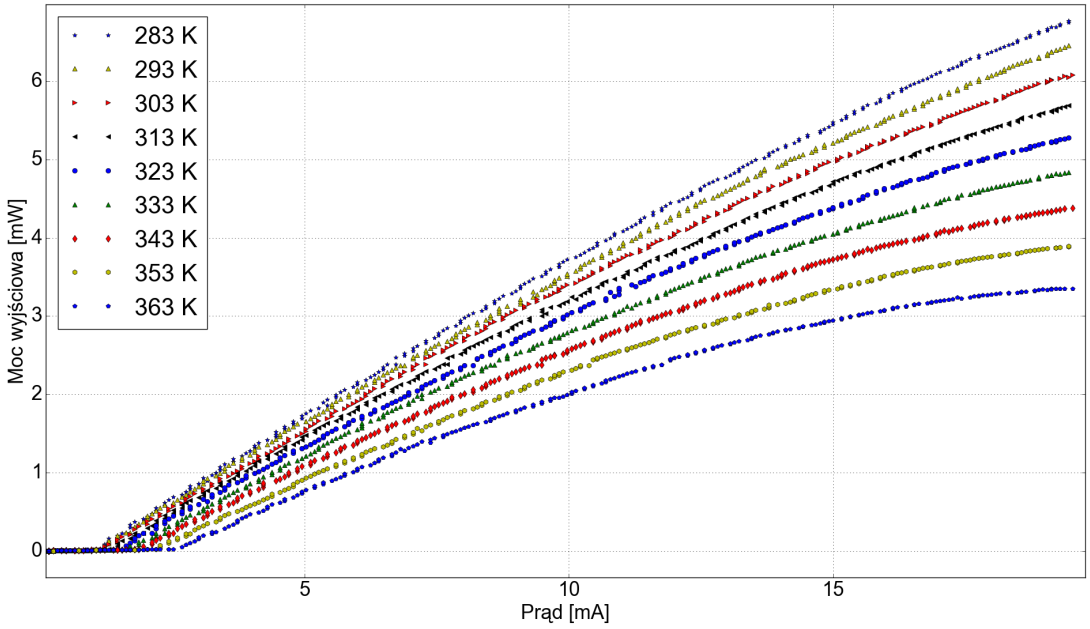
5.2.1 Laser 980 nm

Podpunkt ten zawiera wyniki pomiarów dla lasera krawędziowego 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 wartość 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 wyznazone 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 oblicoznymi paramenrami które słuzą do charakterystyki lasera. Rysunek 5.13 przedstawia wykres prądu progowego w funkcji temperatury.

Tabela 5.1: Wyznaczone wartośc prądu progowego I_0 w różnych temperaturach T dla lasera krawędziowego 980 nm.

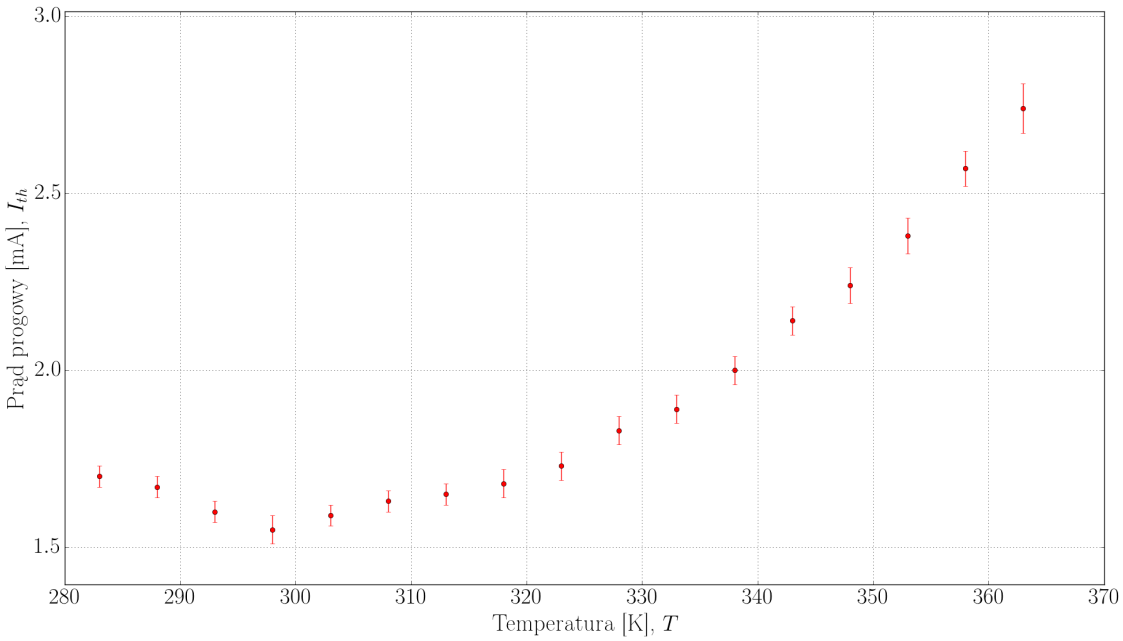
T [K]	I_{th} [mA]
283	1.0 ± 0.1
293	1.0 ± 0.1
303	1.1 ± 0.1
313	1.2 ± 0.1
323	1.3 ± 0.1
333	1.5 ± 0.1
343	1.7 ± 0.1
353	2.0 ± 0.1
363	2.4 ± 0.1



Rysunek 5.1: Charakterystyki wyjściowe lasera krawędziowego 980 nm w różnych temperaturach.

5.2.2 Laser VCSEL 850 nm

Laser 850

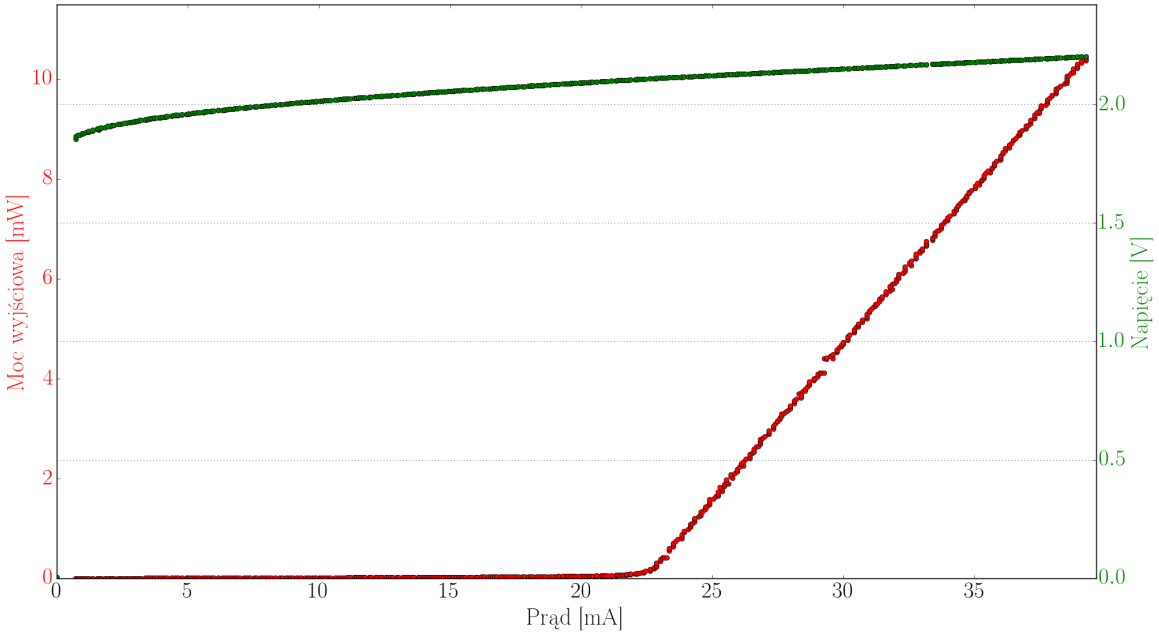


Rysunek 5.2: Wykres prądu progowego w zależności od temperatury dla lasera VCSEL 850 nm.

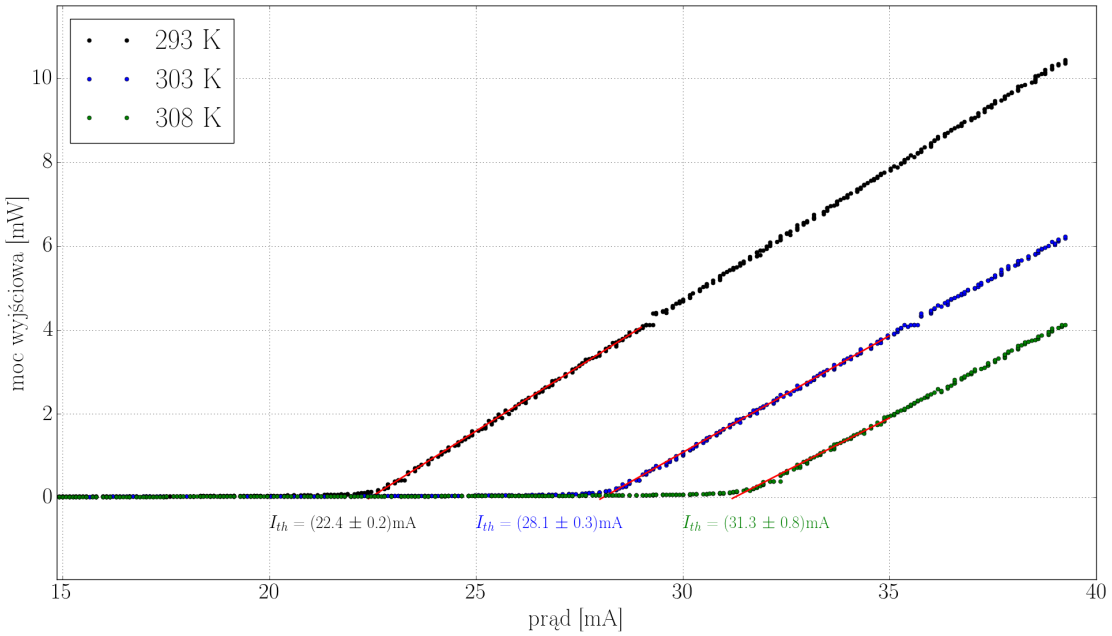
Tabela 5.2: Wyznaczone wartośc prądu progowego I_0 w różnych temperaturach T dla lasera krawędziowego 635 nm.

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

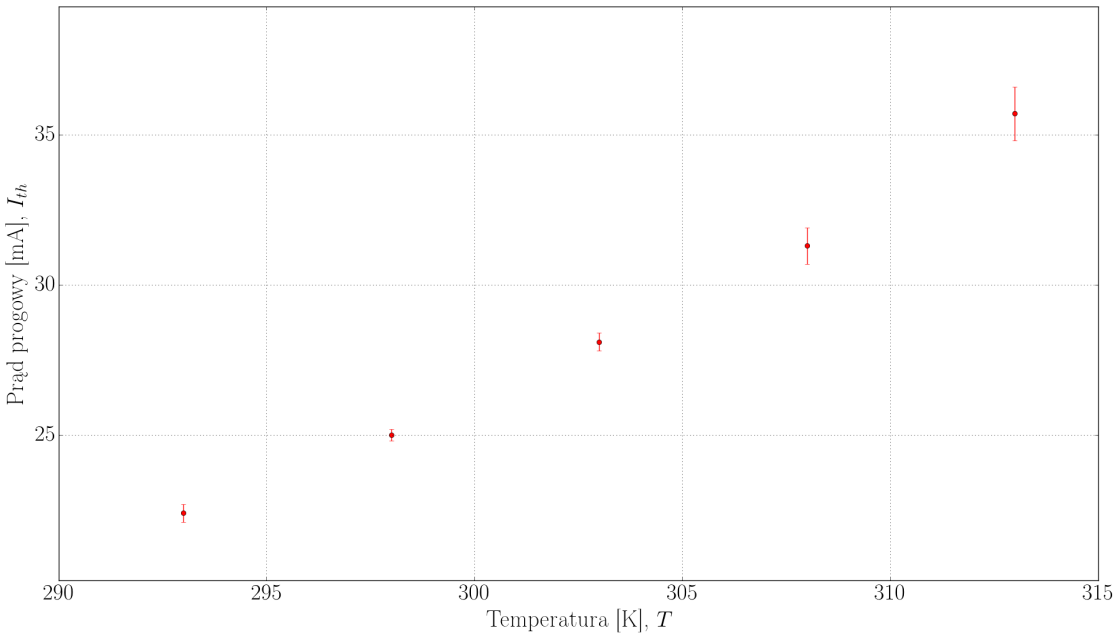
5.2.3 Laser krawędziowy 635 nm



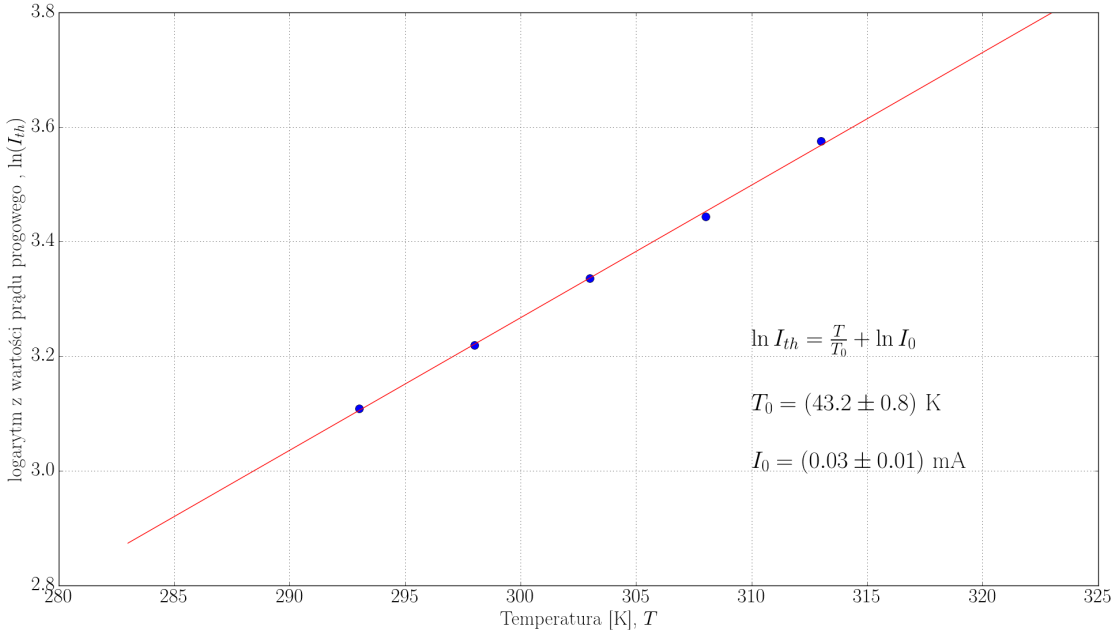
Rysunek 5.3: Charakterystyka wyjściowa lasera krawędziowego 635 nm.



Rysunek 5.4: Charakterystyka wyjściowej lasera krawędziowego 635 nm wraz z wyznaczonym prądem progowym dla różnych temperatur.



Rysunek 5.5: 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 5.6: Wykres logarytmu z prądu progowego w zależności od temperatury wraz z wyznaczonymi parametrami I_0 i T_0 dla lasera krawędziowego 635 nm.