МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

НОВОСИБИРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Факультет информационных технологий

Кафедра параллельных вычислений

ОТЧЕТ

О ВЫПОЛНЕНИИ ПРАКТИЧЕСКОЙ РАБОТЫ

«Высокоуровневая и низкоуровневая работа с периферийными устройствами»

студента 2 курса, группы 22206

***Тропина Никиты Васильевича***

Направление 09.03.01 – «Информатика и вычислительная техника»

Преподаватель:

доцент, к.т.н.

А.Ю.Власенко

Новосибирск 2023

# СОДЕРЖАНИЕ

[ЦЕЛИ 3](#_Toc153406068)

[ЗАДАНИЕ 3](#_Toc153406069)

[ОПИСАНИЕ РАБОТЫ 4](#_Toc153406070)

[ЗАКЛЮЧЕНИЕ 6](#_Toc153406071)

[ПРИЛОЖЕНИЕ 1. Полный листинг программы 1. 7](#_Toc153406072)

[ПРИЛОЖЕНИЕ 2. Полный листинг программы 2. 9](#_Toc153406073)

# ЦЕЛИ

Ознакомиться с программированием периферийных устройств на примере ввода данных с Web-камеры с использованием библиотеки OpenCV.

Ознакомиться с началами низкоуровневого программирования периферийных устройств на примере получения информации о доступных USB-устройствах с помощью библиотеки libusb.

# ЗАДАНИЕ

Реализовать программу №1 с использованием OpenCV, которая получает поток видеоданных с камеры и выводит его на экран.

Выполнить произвольное преобразование изображения (**кроме** указанных в computerlab5.pdf сглаживания и установки значений цветовых каналов в константу).

Измерить количество кадров, обрабатываемое программой в секунду.

Оценить долю времени, затрачиваемого процессором на обработку (ввод, преобразование, показ) видеоданных, получаемых с камеры.

Реализовать программу №2, получающую список всех подключенных к машине USB устройств с использованием libusb. Для каждого найденного устройства напечатать его класс, идентификатор производителя, идентификатор изделия и серийный номер.

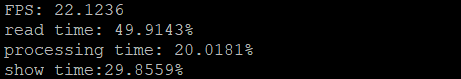
# ОПИСАНИЕ РАБОТЫ

**Описание высокоуровневой работы с периферийными устройствами**

В ходе работы была реализована программа с использованием OpenCV, которая получает поток видеоданных с камеры, обрабатывает и выводит его на экран. Изображение преобразуется с помощью функции processImage: меняет цветовые канала G и R.

Были определены скорость обработки видео и доля времени, затраченного процессором на ввод, обработку и показ видеоданных.



**Описание низкоуровневой работы с периферийными устройствами**

Была реализована программа для получения информации о USB-устройствах.



Расшифровка полученных значений:

Расшифровка кодов классов:

00 – Unclassified device

09 – Input device controller

ef - “устройство класса различные устройства”

Расшифровка самих устройств:

046d 0825 - Webcam C270

1d6b 0002 - 2.0 root hub

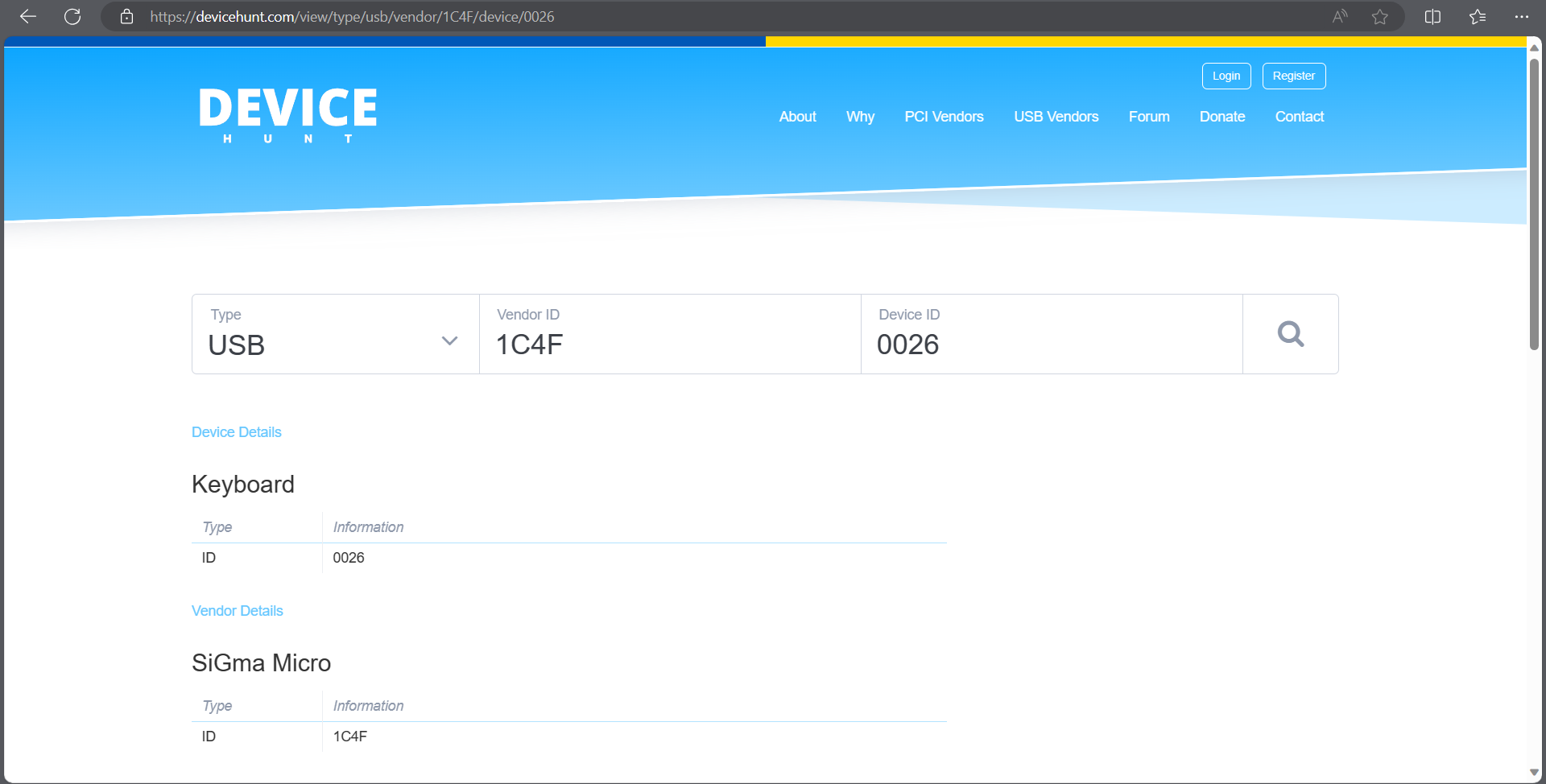
1d6b 0001 - 1.1 root hub

1c4f 0026 - Keyboard

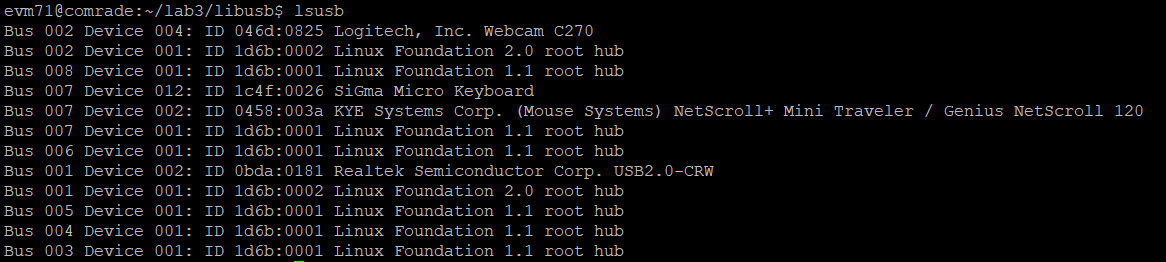
0458 003a - NetScroll+ Mini Traveler / Genius NetScroll 120 (мышка)

0bda 0181 - Realtek Semiconductor Corp. (карт-ридер)

Расшифровка устройств была получена с помощью сайта devicehunt.com



Чтобы проверить корректность данных, используем команду lsusb:



# ЗАКЛЮЧЕНИЕ

В данной работе мы познакомились с программированием периферийных устройств на примере обработки данных с Web-камеры с использованием библиотеки OpenCV, а также с началами низкоуровневого программирования периферийных устройств на примере получения информации о подключенных USB-устройствах с помощью библиотеки libusb.

# ПРИЛОЖЕНИЕ 1. Полный листинг программы 1.

#include <iostream>

#include <string>

#include <iomanip>

#include <sstream>

#include <opencv2/core.hpp>

#include <opencv2/imgproc.hpp>

#include <opencv2/videoio.hpp>

#include <opencv2/highgui.hpp>

#include <opencv2/imgcodecs.hpp>

using namespace std;

using namespace cv;

void processImage(Mat frame) {

int fromTo[] = { 0,0,1,2,2,1};

mixChannels(&frame, 1, &frame, 1, fromTo, 3);

}

int main(int argc, char\* argv[]) {

clock\_t startTime;

VideoCapture cap(0);

int framesCounter = 0;

double readTime = 0, showTime = 0, processingTime = 0, frameStart = 0, frameTime = 0;

Mat img;

startTime = clock();

while (true) {

frameStart = clock();

framesCounter++;

clock\_t start = clock();

cap.read(img);

readTime = clock() - start;

start = clock();

processImage(img);

processingTime = clock() - start;

start = clock();

imshow("Image", img);

showTime = clock() - start;

frameTime = clock() - frameStart;

char c = waitKey(1);

if (c == 'z') break;

}

double fullTime = clock() - startTime;

cout << "FPS: " << (framesCounter / (fullTime / CLOCKS\_PER\_SEC)) << endl;

cout << "read time: " << 100 \* (readTime / frameTime) << "%" << endl;

cout << "processing time: " << 100 \* (processingTime / frameTime) << "%" << endl;

cout << "show time:" << 100 \* (showTime / frameTime) << "%" << endl;

return 0;

}

# ПРИЛОЖЕНИЕ 2. Полный листинг программы 2.

#include <iostream>

#include <libusb.h>

#include <stdio.h>

using namespace std;

void printdev(libusb\_device\* dev) {

libusb\_device\_descriptor desc;

libusb\_device\_handle \*handle = nullptr;

unsigned char str[256]; //для хранения серийного номера

int r = libusb\_get\_device\_descriptor(dev, &desc);

if (r < 0) {

fprintf(stderr, "Ошибка: дескриптор устройства не получен, код : %d.\n", r);

return;

}

printf(" %.2x %.4x %.4x ",

(int) desc.bDeviceClass,

desc.idVendor,

desc.idProduct

);

libusb\_open(dev, &handle);

if (handle && desc.iSerialNumber)

{

r = libusb\_get\_string\_descriptor\_ascii(handle, desc.iSerialNumber, str, sizeof(str)); //печать поля iSerialNumber дескриптора устройства

if (r > 0) {

printf("%s\n", str);

}

else {

printf("empty\n");

}

}

else {

printf("null\n");

}

libusb\_close(handle);

}

int main() {

libusb\_device\*\* devs;

libusb\_context\* ctx = NULL;

int r;

ssize\_t countUsbDevices;

ssize\_t i;

r =libusb\_init(&ctx);

if (r < 0) {

fprintf(stderr, "Ошибка: инициализация не выполнена, код :%d.\n", r);

return 1;

}

libusb\_set\_debug(ctx, 3);

countUsbDevices = libusb\_get\_device\_list(ctx, &devs);

if (countUsbDevices < 0) {

fprintf(stderr, "Ошибка: список USB устройств не получен. %d\n", r);

return 1;

}

printf("найдено устройств: %ld\n", countUsbDevices);

printf("===========================================================\n");

printf("|\* класс устройства\n");

printf("| | \* идентификатор производителя\n");

printf("| | | \* идентификатор устройства\n");

printf("| | | | \* серийный номер\n");

printf("+--+----+----+--------------------+\n");

for (ssize\_t i = 0; i < countUsbDevices; i++){

printdev(devs[i]);

}

printf("===========================================================\n");

libusb\_free\_device\_list(devs, 1);

libusb\_exit(ctx);

return 0;

}