

ЦЕЛЬ

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

ОПИСАНИЕ РАБОТЫ

С помощью листинга, приведенного в задании к лабораторной работе, реализуем программу, которая получает класс, идентификатор производителя и идентификатор изделия, подключенных через usb.

Листинг

```
#include <iostream>
#include <libusb.h>
#include <stdio.h>

using namespace std;

void print_device_info(libusb_device *device) {
    libusb_device_descriptor desc;
    libusb_config_descriptor *config;
    const libusb_interface *inter;
    const libusb_interface_descriptor *interdesc;
    int result = libusb_get_device_descriptor(device, &desc);
    if (result < 0) {
        cerr << "ОШИБКА: не получен дескриптор устройства. Код ошибки: " <<
result << endl;
        return;
    }
    libusb_get_config_descriptor(device, 0, &config);
    inter = &config->interface[0];
    cout << "Класс: " << static_cast<int>(desc.bDeviceClass) << endl;
    cout << "Идентификатор производителя: " << desc.idVendor << endl;
    cout << "Идентификатор устройства: " << desc.idProduct << endl;
    cout << "-----" << endl;
}

int main() {
    libusb_device **devices;
    libusb_context *ctx = nullptr;
    int result;
    ssize_t device_count;
    result = libusb_init(&ctx);
    if (result < 0) {
        cerr << "ОШИБКА: инициализация не выполнена. Код ошибки: " << result
<< endl;
        return 1;
    }
    device_count = libusb_get_device_list(ctx, &devices);
    if (device_count < 0) {
        cerr << "ОШИБКА: список USB устройств не получен." << endl;
        return 1;
    }
    cout << "Число найденных устройств " << device_count << endl;
    for (ssize_t i = 0; i < device_count; i++) {
        print_device_info(devices[i]);
    }
    libusb_free_device_list(devices, 1);
    libusb_exit(ctx);
    return 0;
}
```

Печать серийного номера

Получившуюся программу необходимо дополнить печатью серийного номера каждого устройства. Для этого в `print_device_info` добавим

```
libusb_device_handle *handle;
result = libusb_open(device, &handle);
if (result == 0) {
    unsigned char buffer[256];
    if (desc.iSerialNumber > 0) {
        libusb_get_string_descriptor_ascii(handle, desc.iSerialNumber,
        buffer, sizeof(buffer));
        cout << "Серийный номер: " << buffer << endl;
    }
    libusb_close(handle);
}
```

Теперь, если имеем доступ к серийному номеру устройства, сможем его напечатать.

В результате выполнения программы получим:

Число найденных устройств 12

Класс: 239
Идентификатор производителя: 1133
Идентификатор устройства: 2085
Серийный номер: 95410D90

Класс: 9
Идентификатор производителя: 7531
Идентификатор устройства: 2
Серийный номер: 0000:00:1d.7

Класс: 9
Идентификатор производителя: 7531
Идентификатор устройства: 1
Серийный номер: 0000:00:1d.2

Класс: 0
Идентификатор производителя: 7247
Идентификатор устройства: 38

Класс: 0
Идентификатор производителя: 1112
Идентификатор устройства: 58

Класс: 9
Идентификатор производителя: 7531
Идентификатор устройства: 1
Серийный номер: 0000:00:1d.1

Класс: 9
Идентификатор производителя: 7531
Идентификатор устройства: 1
Серийный номер: 0000:00:1d.0

Класс: 0
Идентификатор производителя: 3034
Идентификатор устройства: 385
Серийный номер: 20060413092100000

Класс: 9
Идентификатор производителя: 7531
Идентификатор устройства: 2
Серийный номер: 0000:00:1a.7

Класс: 9
Идентификатор производителя: 7531
Идентификатор устройства: 1
Серийный номер: 0000:00:1a.2

Класс: 9
Идентификатор производителя: 7531
Идентификатор устройства: 1
Серийный номер: 0000:00:1a.1

Класс: 9
Идентификатор производителя: 7531
Идентификатор устройства: 1
Серийный номер: 0000:00:1a.0

Первое устройство имеет класс 239 (в шестнадцатиричной системе EF), что означает “различные устройства”. Остальные относятся к классу 0 (информацию о классе нужно получать в дескрипторе интерфейса) или 9 (концентратор)

Чтобы получить дополнительную информацию об устройстве добавим в код:

```

if (desc.iProduct > 0) {
    libusb_get_string_descriptor_ascii(handle, desc.iProduct, buffer,
sizeof(buffer));
    cout << "Название устройства: " << buffer << endl;
}

```

В итоге получим

Число найденных устройств 12

Класс: 239

Идентификатор производителя: 1133

Идентификатор устройства: 2085

Серийный номер: 95410D90

Класс: 9

Идентификатор производителя: 7531

Идентификатор устройства: 2

Серийный номер: 0000:00:1d.7

Название устройства: EHCI Host Controller

Класс: 9

Идентификатор производителя: 7531

Идентификатор устройства: 1

Серийный номер: 0000:00:1d.2

Название устройства: UHCI Host Controller

Класс: 0

Идентификатор производителя: 7247

Идентификатор устройства: 38

Класс: 0

Идентификатор производителя: 1112

Идентификатор устройства: 58

Название устройства: USB Optical Mouse

Класс: 9

Идентификатор производителя: 7531

Идентификатор устройства: 1

Серийный номер: 0000:00:1d.1

Название устройства: UHCI Host Controller

Класс: 9

Идентификатор производителя: 7531

Идентификатор устройства: 1

Серийный номер: 0000:00:1d.0

Название устройства: UHCI Host Controller

Класс: 0

Идентификатор производителя: 3034

Идентификатор устройства: 385

Серийный номер: 20060413092100000

Название устройства: USB2.0-CRW

Класс: 9

Идентификатор производителя: 7531

Идентификатор устройства: 2

Серийный номер: 0000:00:1a.7

Название устройства: EHCI Host Controller

Класс: 9

Идентификатор производителя: 7531

Идентификатор устройства: 1

Серийный номер: 0000:00:1a.2

Название устройства: UHCI Host Controller

Класс: 9

Идентификатор производителя: 7531

Идентификатор устройства: 1

Серийный номер: 0000:00:1a.1

Название устройства: UHCI Host Controller

Класс: 9

Идентификатор производителя: 7531

Идентификатор устройства: 1

Серийный номер: 0000:00:1a.0

Название устройства: UHCI Host Controller

Теперь знаем название каждого устройства.

Однако не для всех видим идентификатор и серийный номер, так как нам не хватает прав для их просмотра.

ЗАКЛЮЧЕНИЕ

Я научилась пользоваться библиотекой `libusb` и с помощью нее смогла написать программу и узнать часть доступной мне информации о подключенных с помощью `usb` устройствах.

Приложение 1. Листинг программы

```
#include <iostream>
#include <libusb.h>
#include <stdio.h>

using namespace std;

void print_device_info(libusb_device *device) {
    libusb_device_descriptor desc;

    int result = libusb_get_device_descriptor(device, &desc);
    if (result < 0) {
        cerr << "ОШИБКА: не получен дескриптор устройства. Код ошибки: " <<
result << endl;
        return;
    }

    cout << "Класс: " << static_cast<int>(desc.bDeviceClass) << endl;
    cout << "Идентификатор производителя: " << desc.idVendor << endl;
    cout << "Идентификатор устройства: " << desc.idProduct << endl;

    libusb_device_handle *handle;
    result = libusb_open(device, &handle);

    if (result == 0) {
        unsigned char buffer[256];
        if (desc.iSerialNumber > 0) {
            libusb_get_string_descriptor_ascii(handle, desc.iSerialNumber,
buffer, sizeof(buffer));
            cout << "Серийный номер: " << buffer << endl;
        }
        if (desc.iProduct > 0) {
            libusb_get_string_descriptor_ascii(handle, desc.iProduct, buffer,
sizeof(buffer));
            cout << "Название устройства: " << buffer << endl;
        }
        libusb_close(handle);
    }
    cout << "-----" << endl;
}

int main() {
    libusb_device **devices;
    libusb_context *ctx = nullptr;
    int result;
    ssize_t device_count;
    result = libusb_init(&ctx);

    if (result < 0) {
        cerr << "ОШИБКА: инициализация не выполнена. Код ошибки: " << result
<< endl;
        return 1;
    }

    device_count = libusb_get_device_list(ctx, &devices);

    if (device_count < 0) {
        cerr << "ОШИБКА: список USB устройств не получен." << endl;
        return 1;
    }
    cout << "Число найденных устройств " << device_count << endl;

    for (ssize_t i = 0; i < device_count; i++) {
```



```
        print_device_info(devices[i]);  
    }  
  
    libusb_free_device_list(devices, 1);  
    libusb_exit(ctx);  
  
    return 0;  
}
```