

УДК 004.4

## АВТОМАТИЧЕСКАЯ ВЫГРУЗКА ПОКАЗАНИЙ СПЕКТРОМЕТРОВ OPTOSKY ДЛЯ ЗАДАЧИ ОБРАБОТКИ БИОЛОГИЧЕСКИХ ДАННЫХ

Тимофеев Е.М. (ИГУ)

Научный руководитель – Зав. Лабораторией Роботехники и программирования Веснин А.М. (ИГУ)

**Введение.** В НИИ биологии Иркутского государственного университета реализуется проект по разработке технологии отслеживания биохимических показателей обитателей озера Байкал - эндемичных рачков-амфипод. Технология предполагает использование флуоресцентных имплантов, размещённых в теле рачка и меняющих свои спектральные характеристики в зависимости от физиологического состояния организма. Для анализа этих характеристик была собрана установка, обрабатывающая сигнал имплантов, состоящая из флуоресцентного микроскопа, интегрированного со спектрометром и цифровой камерой. Однако, при реализации проекта пользователи столкнулись с тем, что имеющееся в наличии оборудование (макроскоп Z16 Apo, Leica, ФРГ; спектрометр АТР6500, Optosky, КНР; камера EOS 2000D, Canon, Тайвань) обладает недостатками не позволяющими реализовать проект.

**Основная часть.** Одной из основных проблем собранной установки являлось программное обеспечение для получения данных со спектрометра, предоставляемое компанией производителем Optosky. В частности, графический интерфейс программного обеспечения (ПО) для операционной системы (ОС) Windows не обеспечивает возможность быстрого сохранения спектров, что крайне важно при работе с активно двигающимися животными, и ряд других традиционных функций для обработки спектров. Для ОС Linux производителем поставляется ПО, предусматривающее исключительно ручную выгрузку данных через консоль. В то же время, выбор именно ОС Linux предпочтителен для простой интеграции цифровой камеры через библиотеку libgphoto2. Поэтому было принято решение по созданию специального ПО, которое будет, в том числе, выгружать, анализировать и сохранять данные со спектрометра АТР6500 в необходимом формате, предоставляя пользователям все эти функции в удобном графическом интерфейсе.

В целом, разрабатываемый комплекс представляет собой решение, которое обеспечит пользователю множественный функционал — автоматический сбор, анализ и визуализацию спектров, фото и видеофиксацию образца, а также возможность снимать ряды данных при фокусировке на различные глубины в тканях организма. Ключевым требованием к комплексу является ПО и широкий функционал для автоматизации работы. Комплекс включает модуль связи со спектрометром, обеспечивающий автоматизированную выгрузку данных и их представление в удобной для анализа форме. Это позволяет существенно ускорить обработку информации и минимизировать вероятность ошибок, допускаемых при ручной работе. Также имеется возможность управления серводвигателем (через систему Maestro, Pololu, США), подключённым к микроскопу, что даёт пользователю инструмент для точной настройки высоты и фокуса, а также обеспечивает возможность съёмки фотографий и спектров через фиксированные шаги по вертикальной оси. Наконец, предусмотрен модуль захвата изображений, использующий свободную библиотеку libgphoto2. Комбинирование двух последних модулей позволяет фиксировать снимки на разных высотах, которые затем обрабатываются алгоритмом стэкинга изображений. Алгоритм объединяет несколько снимков с разным фокусом на объекте, что значительно улучшает итоговое качество изображения и повышает его информативность. Программный комплекс реализован на языке Python, так как он оптимально подходит для работы с фотоматериалами и динамическими графиками в реальном времени. Помимо Python, для выгрузки данных со спектрометра был использован скрипт на языке C.

Большая часть комплекса ранее была нами реализована в виде прототипа, работающего с искусственно созданными данными. Однако, при запуске с реальными показателями, возникла серьёзная проблема, связанная с ПО, предоставляемым компанией-производителем

спектрометров Optosky. При этом вся документация оригинального ПО предоставляется только на китайском языке.

Для решения задачи была сделана попытка реализовать выгрузку данных стандартным способом — подключаясь к скрипту и обмениваясь данными через `os.popen()`. В результате программа заходила в бесконечный цикл, постоянно отправляя приветственное сообщение с информацией о функционале спектрометра. После нескольких неудачных попыток было принято решение использовать `subprocess.Popen()`, а после `subprocess.run()`, однако с помощью этих библиотек проблема решена не была. При этом само подключение происходило успешно, но программа также уходила в бесконечный цикл. В ходе изучения скрипта выгрузки было обнаружено, что при получении любого сообщения, не входящего в указанный в документации список команд, программа начинает бесконечно пытаться его обработать. На основе этого наблюдения было сделано предположение, что `subprocess` при подключении может отправлять невидимый пользователю текст (например, для проверки соединения), что и провоцирует заикливание. Попытка отключить буферизацию вывода и проверка вывода с `strace` и `tee` для анализа скрытых запросов, не дали результата. После повторного изучения скрипта выгрузки и перебора вариантов подключения, было установлено, что проблема заключается в том, что ПО для выгрузки данных работает исключительно в режиме интерактивного ввода команд, а `subprocess` не рассчитан на работу с таким вариантом ввода, что и приводит к проблемам при подключении:

1. **`subprocess.Popen()` не всегда корректно читает `stdout.readline()`**

Если программа ждёт ввода, `stdout.readline()` может зависнуть и ничего не вернуть. Даже если добавить `flush=True` в `print()` внутри скрипта для выгрузки данных, `subprocess` не всегда распознаёт новый вывод.

2. **`subprocess.communicate()` завершает процесс после выполнения**

Если вызвать `communicate()`, программа получит ввод и сразу закроется. В то время как необходимо держать процесс открытым и отправлять команды.

3. **Проблемы с буферизацией**

`subprocess` использует буферы для чтения/записи, из-за чего часть данных может передаваться не сразу.

Исходя из понимания механизма проблемы была опробована другая узкоспециализированная библиотека `rexpect`, разработанная конкретно под интерактивный ввод. Её использование решило проблему автоматической выгрузки показателей спектрометра и позволили запустить аналитический комплекс в требуемом автоматическом режиме.

**Выводы.** Применение стандартного подхода подключения к спектрометру через библиотеки `subprocess` или `os` не позволяют решить проблему автоматической выгрузки данных, для решения этой задачи необходимо использовать библиотеки, ориентированные на интерактивный ввод команд, например `rexpect`.

#### Список использованных источников:

1. Документация по библиотеке `rexpect` для языка `python` <https://rexpect.readthedocs.io/en/stable/> (10.02.2025)
2. Документация по библиотеке `subprocess` для языка `python` <https://docs.python.org/3/library/subprocess.html> (23.01.2025)
3. Документация по библиотеке `os` для языка `python` <https://docs.python.org/3/library/os.html> (17.01.2025)