

Реализовать подключение шаговых двигателей(28BYJ-48) к Raspberry Pi и написание веб-приложения для контроля столика микроскопа, подробнее ниже по пунктам.

Задача:

Автоматизировать микроскопическое исследование паразитов.

Для этого уже готово:

- моторизированный предметный столик для микроскопа
- Нейронная сеть для анализа изображений

Usercase:

1. Лаборант подготавливает образец(предметное стекло) для диагностики.
2. Фиксирует приметное стекло на моторизованном предметном столике.
3. Выбирает увеличение микроскопа.
4. Через вебинтерфейс устанавливает:
  - размеры предметного стекла;
  - увеличение микроскопа;
  - стратегию анализа предметного стекла ,то есть делать пропуски или анализировать каждое поле зрения;
  - стартовую позицию - левый верхний угол или центр.
  - путь куда сохранять фотографии/результаты анализа.
  - флаг, прерывать ли анализ при наступлении события. Событие это детекция паразита в образце
5. Запускает анализ по нажатию кнопки в вебприложении
6. В течение анализа получает статистику через вебприложение, сколько проанализировано, сколько осталось и пр...

1. Разработать скрипт управления моторами, схему подключения, веб-интерфейс.
2. В веб-интерфейсе реализовать: выбор увеличения микроскопа и размера предметного столика, выбор стратегии анализа предметного стекла(то есть делать пропуски или анализировать каждое поле зрения), выбор пути куда сохранять фотографии/результаты анализа, флаг - прерывать ли анализ при наступлении события. (Событие это детекция паразита в образце), кнопки старт/стоп анализ, вывод статистики сколько проанализировано/сколько осталось, в админ интерфейсе указывать параметры шестеренок или другие параметры, которые влияют на движение осей, параметры сети и прочие данные, которые могут понадобиться
4. Реализовать функции: съемки предметного столика, сохранения изображения, автофокусировки, отправки изображения в бэкэнд для обработки, прерывания анализа при получении информации с бэкэнда о детекции паразита(сделать функцию, которая может получать True от бэкэнда и останавливать процесс)

