

# ПРОЕКТИРОВАНИЕ АРХИТЕКТУРЫ ДИСТАНЦИОННОЙ СИСТЕМЫ УПРАВЛЕНИЯ РОБОТОМ

*Слоневский Степан Васильевич*

УО «Брестский государственный университет имени А.С. Пушкина»

В процессе создания автономной мобильной платформы (Робота) сформулирована задача создания дистанционной системы управления. Для решения поставленной задачи мы воспользовались известной в веб программировании архитектурой фронтенд-бэкенд. Сущность данной архитектуры [1] состоит в том, что она включает два компонента: фронтэнд (front-end) и бэкенд (back-end). Существенное значение имеет то, что такое построение архитектуры позволяет разделить внешнее представление и внутреннюю реализацию приложения, которое в нашем случае является веб-приложением.

В нашем случае Бэкенд реализован на базе программно-аппаратной платформы на базе микрокомпьютера Raspberry Pi 3. Тогда как фронтэнд представлен клиентской стороной пользовательского интерфейса. Микроконтроллер Arduino выступает в качестве программно-аппаратного устройства непосредственного управления роботом. Бэкенд позволяет принимать и отправлять данные на arduino, а также реализует API (Application Programming Interface [2]) для взаимодействия с фронтендом.

Для управления роботом необходимо сформировать систему команд, которые требуется кодировать, передавать Arduino, и декодировать, а также выполнять на Шилдах [3] Arduino. Arduino может принимать команды только от микрокомпьютера к которому он присоединен. Заметим, что микрокомпьютер и микроконтроллер размещены непосредственно на мобильной платформе. Самое очевидное решение данной проблемы это настроить в качестве посредника для передачи команд устройствам управления Arduino микрокомпьютер Raspberry. Микрокомпьютер должен принимать команды от клиента (внешнего вычислительного устройства, например, ноутбука или мобильного телефона) с использованием локальной вычислительной сети. Для построения сети на микрокомпьютере должен быть установлен веб сервер, например, Apache [4]. Обращение к приложению со стороны клиента управления мобильной платформой происходит с использованием веб-браузера. При обращении к URL микрокомпьютера в адресной строке

браузера с указанием порта приложения сервер выдает клиенту html страницу веб-приложения (фронтэнд) с реализацией скриптов (для связи с бэкэндом). Таким образом веб-приложение реализует архитектуру управления роботом на основе предложенного подхода. На html странице размещены необходимые элементы для управления роботом, окна контроля состояния робота, отражающие показания датчиков, графики их динамики. В отдельном окне в режиме реального времени выводится видеоряд, полученный с использованием установленных приборов компьютерного зрения робота (видеокамер). Для связи сервера и клиента использован скрипт на основе протокола websocket [5]. Скрипт использован для подключения клиента (веб-приложение) к серверу (Raspberry Pi), благодаря чему имеется возможность передавать данные в обе стороны.

В качестве альтернативной реализации фронтэнда может выступать приложение на платформах Android, IOS, Windows, Linux, с заранее заданным дизайном отображения.

#### Список первоисточников

1. Front end и back end. [Электронный ресурс]. Дата доступа: 1.02.2019. Режим доступа: [https://ru.wikipedia.org/wiki/Front\\_end\\_и\\_back\\_end](https://ru.wikipedia.org/wiki/Front_end_и_back_end)
2. Что такое API. [Электронный ресурс]. Дата доступа: 2.02.2019. Режим доступа: <https://habr.com/ru/sandbox/52599/>
3. Arduino Shields – платы расширения для ардуино. [Электронный ресурс]. Дата доступа: 2.02.2019. Режим доступа: <https://arduinomaster.ru/platy-arduino/shildy-i-platy-rasshireniya-arduino/>
4. Apache Directory. Официальный сайт. [Электронный ресурс] Дата доступа: 2.02.2019. Режим доступа: <http://directory.apache.org>
5. WebSockets — полноценный асинхронный веб. [Электронный ресурс]. Дата доступа: 2.02.2019. Режим доступа: <https://habr.com/ru/post/79038/>