

подключится к наиболее близкому роботу с учетом реализованного протокола общения между устройствами.

Подводя итог, отметим достоинство групповой робототехники как децентрализованной системы, хорошо масштабируемой и работающей на больших площадях, но при этом не являющейся достаточно универсальной и требующей отдельной реализации для решения частных задач.

Список использованных источников:

1. Hamann, H.: Swarm Robotics: A Formal Approach. Springer International Publishing AG 2018

ОДНОПЛАТНЫЕ КОМПЬЮТЕРЫ В СИСТЕМАХ ПОЛУЧЕНИЯ ПОТОВОКОВЫХ АУДИОДАННЫХ

Ермолович И.А., Пилинко Н.А., Фадеева Е.Е

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Леванцевич В.А. – м.т.н., старший преподаватель

Описана реализация автономного приемника потоковых аудиоданных для воспроизведения передач с интернет радиостанций на базе одноплатного компьютера Raspberry Pi и операционной системы RaspbianOS.

В настоящее время широко распространено вещание радиостанций в УКВ диапазоне с весьма хорошим качеством. Однако у эфирных радиостанций есть несколько недостатков: во-первых, это неуверенный прием сигнала в некоторых районах, во-вторых ограниченный выбор радиостанций.

Целью проекта стало создание автономного приемника потоковых аудиоданных, способного воспроизводить вещание с интернет радиостанций on-line [1]. Для реализации проекта был выбран полярный микрокомпьютер Raspberry Pi, причем можно использовать любую модель: B, B+, 2 или Zero [2]. Подключение к сети Интернет может быть как проводным, так и по WiFi. В качестве операционной системы выбрана RaspbianOS как наиболее оптимизированная система для данного семейства микрокомпьютеров, позволяющая достичь максимальной производительности.

Система имеет клиент-серверную архитектуру. Сервер построен на связке LAMP (Linux, Apache, MySQL, PHP), клиент использует Python и Bash для управления и MPD (Music Player Daemon) для получения мультивещательного потока данных, а также имеет MPC (Media Player Classic) для упрощения управления MPD системы (Рис 1).

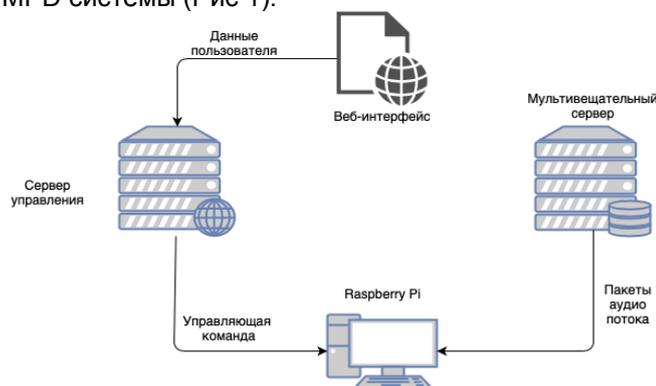


Рисунок 1 - Структура клиент-серверной системы

Программа на устройстве использует системные функции и не требует дополнительных библиотек для своей работы. Установка сделана на основе Bash-скрипта и самих файлов программы. По запуску Bash-скрипт помещает все необходимые файлы в загрузочные сектора системы и производит настройку программы для автозапуска при включении системы. Для работы графической части используется веб-страница, запускаемая в веб-браузере Chromium на устройстве. Данная связка позволяет производить улучшения системы без необходимости обновления ПО на самом устройстве. Также HTML+CSS предоставляет лучшие инструменты для проектирования пользовательского интерфейса, чем библиотеки поставляемые для Python. Программа на устройстве делает запрос на сервер управления каждые 300мс. Сервер сообщает устройству необходимые команды, которые оно впоследствии выполняет. Используется безопасный протокол HTTPS, поэтому дополнительного шифрования трафика не производится.

Список использованных источников

1. Как работает Интернет-радио <https://radio-tochka.com>
2. Пети, В. Микрокомпьютеры Raspberry Pi. Практическое руководство/ В.А. Петин – БХВ-Петербург, 2017.-240с

ВЕБ-ПРИЛОЖЕНИЕ СОЗДАНИЯ И ПРОВЕДЕНИЯ ОПРОСОВ С ИСПОЛЬЗОВАНИЕМ ТЕХНОЛОГИЙ REACT, REDUX, ASP.NET WEBAPI 2

Ермошин М.А.

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Болтак С.В. – ассистент

В современном мире многим компаниям для того, чтобы оставаться конкурентоспособными необходимо собирать отзывы своих клиентов и быстро реагировать на их изменения. Для этого есть много способов. К ним относятся такие как эксперимент, наблюдение, изучение различных документов. Но один из лучших способов сбора этой информации – проведение опроса.

Опрос – это чаще всего небольшой перечень вопросов, способных дать необходимую информацию. Для улучшения качества результатов вопросы могут разбиваться по темам и вводиться вопросы-фильтры для отсеивания анкет, заполненных ненадлежащим образом.

Вот некоторые преимущества онлайн-опросов:

- экономия ресурсов (денег, времени и трудозатрат);
- широта охвата (преодоление границ и расстояний, доступ к различным социальным группам и сообществам);
- оставляют возможность выбора удобного времени и места участия, могут быть завершены в любое удобное для респондента время;
- релевантность (самостоятельность) коммуникации, т. е. более низкий уровень влияния интервьюера на респондента, возможность давать более развёрнутые ответы;
- широта охвата предметных полей (возможность изучать деликатные и закрытые для публичного обсуждения темы);
- возможность автоматической проверки анкет.

Однако пока что в онлайн-опросах так же присутствуют некоторые недостатки, которые являются критическими для некоторых методологий проведения опроса. Главный недостаток онлайн-опросов связан с проблемой обеспечения репрезентативности выборки. Во-первых, это отсутствие основы выборки. Эта проблема может быть успешно решена в исследованиях организаций с широкими сетевыми базами, а также при построении выборки по результатам оффлайн опроса. Во-вторых, проблема охвата, то есть неспособность используемой выборочной процедуры охватить реальную генеральную совокупность (то есть задать известную ненулевую вероятность попадания в выборку для каждой единицы совокупности). И, в-третьих, это не-ответы либо отказы от участия. Обычно успешно удаётся решить первые две [1].

В разрабатываемом приложении учтено большинство из преимуществ онлайн-опросов. Процедура создания опроса проста и интуитивно понятна. Создать опрос можно на любую интересующую создателя опроса тему. Сразу же после создания опрос публикуется, и респонденты-пользователи сайта смогут проходить опрос. Для приглашения человека поучаствовать в опросе необходимо любым способом (*email*, почта, звонок, факс и т. д.) переслать ему ссылку на опрос.

Стек технологий для приложения был выбран неслучайно. Для клиентской стороны приложения были выбраны такие библиотеки как *React* и *Redux*. Они легковесные, легко дополняемые другими модулями и просты в использовании. Так же их использование уменьшит количество работы, исполняемой на серверной части, что позволит снизить требования к аппаратному обеспечению.

React используется для генерации пользовательского интерфейса. В основе данной библиотеки лежит компонентный подход, т. е. вся разметка разбивается на много компонентов различного уровня. Каждый компонент отвечает только за одну определённую цель.

Клиентское приложение требует хранения его состояния (идентификатор пользователя, вошедшего в систему, промежуточные данные при создании опроса и т. д.). Так как в *React* нету необходимых компонент для хранения состояния, выбор пал на *Redux*. Эта библиотека предоставляет единственное хранилище состояния приложения, которое и является «единственным источником правды» для всего приложения.