

ТЕХНИЧЕСКИЕ НАУКИ

Факультет математики и технологий программирования

В. В. Громыко

Науч. рук. Е. И. Сукач,

канд. техн. наук, доцент

СОЗДАНИЕ КОММУНИКАЦИОННОЙ ВЕБ-ПЛАТФОРМЫ ДЛЯ АВТОМАТИЗАЦИИ ПЕРЕДАЧИ И ПОЛУЧЕНИЯ СООБЩЕНИЙ С ПОДДЕРЖКОЙ LATEX-ВЫРАЖЕНИЙ

Основным требованием, предъявляемым к современным платформам для коммуникации, является обеспечение бесперебойной и быстрой работы. Программное обеспечение такого вида становится всё более необходимым для организации научного и учебного общения. Современные решения на рынке позволяют передавать большой объем текстовой и бинарной информации. Однако несмотря на все технологические преимущества крупных решений, их объединяет общая проблема: отсутствие поддержки функций для общения ученых и студентов, которые хотят обмениваться формулами LaTeX-документов. Если необходимо отправить кому-то часть подобного документа, то возможна либо отправка всего документа с выражениями как файл, либо отправка выражения отдельно в виде программного кода, который невозможно прочитать, предварительно не вставив их в нужную программу. Это, в свою очередь, может доставлять множество неудобств и трудностей на дистанции. До сих пор на рынке нет популярного и простого решения этой проблемы, которое бы удовлетворяло все потребности пользователей. Поэтому задача по созданию собственного продукта, которое бы полностью поддерживало как отpravку всего математического документа, так и отдельного выражения сразу, является актуальной и по сей день.

В результате было создано полноценное веб-приложение с удобным и простым пользовательским интерфейсом, которое позволяет отправлять как обычный текст, так и LaTeX-выражения в простом для просмотра виде. Сервис, состоящий из серверной и клиентской частей, был создан благодаря множеству технологий, среди которых ЯП TypeScript, JS-библиотеки React и Next.js, веб-фреймворк Nest.js, СУБД PostgreSQL и многие другие.

Для осуществления приватности никакие данные пользователей не передаются третьим лицам. Также для безопасности в приложении был реализован «Анонимный чат», который позволяет общаться с другими пользователями в общей комнате под специально сгенерированным именем пользователя, с помощью которого трудно вычислить реальные данные собеседника.

Представленная в проекте идея является уникальной и необычной для сферы мессенджеров, что говорит о перспективности данного направления и отличного опыта использования, как для обычных пользователей, так и для более продвинутых представителей научной сферы.

Р. Ю. Громыко

Науч. рук. Е. И. Сукач,

канд. техн. наук, доцент

РАБОТА С ДАТЧИКАМИ И ФАЙЛАМИ НА МИКРОКОНТРОЛЛЕРЕ ESP-32

Во все времена точная информация являлась гарантией надежных расчетов и правильных выводов. Благодаря своевременному контролю и обработке данных можно предсказывать погодные явления, контролировать работы автономных систем, отслеживать состояния людей и животных, и это только малая часть всех примеров. Многие из вышеперечисленного можно

реализовать с помощью микроконтроллеров Arduino, например, на плате ESP-32. Микроконтроллеры ESP-32 достаточно гибкие и легко могут подстроиться под нужды пользователя благодаря большому количеству модулей и датчиков. Также платы достаточно дешевы, их цена является низкой, что позволяет делать большие системы, состоящие из множества датчиков.

В докладе описывается, как создать небольшой скрипт для обработки данных, поступающих с датчиков, и записи их в файл на SD-карту, подключённую через модуль к плате. Для реализации скрипта было использовано приложение Arduino IDE. Логика работы датчика предполагается цикличной. Сама по себе плата ESP-32 оснащена встроенным датчиком Холла, получать данные с которого можно с `hallRead()`. Помимо встроенного датчика Холла к микропроцессору ESP-32 подключено 3 датчика и модуль для вставки SD-карты. Работа с ними осуществлялась с помощью специальных одноимённых библиотек.

Первый датчик – это датчик освещения BH1750. Чтобы работать с этим датчиком, в скрипт была добавлена библиотека “BH1750.h”. Следующий датчик – это датчик атмосферного давления и температуры BMP280. Чтобы работать с этим датчиком, в скрипт была добавлена библиотека “Adafruit_BMP280.h”. Последний датчик – это датчик влажности и температуры AHT20. Чтобы работать с этим датчиком, в скрипт была добавлена библиотека “Adafruit_AHTX0.h”.

В каждый тысячный цикл метода микроконтроллера все данные с датчиков записываются в строку, а после и в файл. Данные с датчиков записываются на SD-карту. Для этого подключаются 2 библиотеки “FS.h” для работы с файлами и “SD.h” для работы с SD-картой. На SD-карте находятся 2 файла, в один из которых раз в некоторое время записываются данные с датчиков, а в другой при отправке и удалении данных из первого файла копируется информация.

А. А. Зданович

Науч. рук. Л. Н. Марченко,

канд. техн. наук, доцент

РАЗРАБОТКА ПРИЛОЖЕНИЯ РАСПОЗНАВАНИЯ ПОЖАРОВ НА ИЗОБРАЖЕНИЯХ НА ЯЗЫКЕ ПРОГРАММИРОВАНИЯ PYTHON

В последнее время области применения машинного обучения быстро расширяются, охватывая все больше различных сфер жизнедеятельности. На сегодняшний день для реализации систем машинного зрения в основном применяются сверточные нейронные сети, которые являются одним из видов искусственных нейронных сетей. Именно благодаря способности к обучению искусственные нейронные сети используются для решения сложных задач классификации и регрессионного анализа. Актуальность данной работы обуславливается тем, что применение методов распознавания объектов на изображении может помочь уменьшить роль человеческого фактора в тех случаях, где первоначально важным является быстрое реагирование на конкретные происшествия. Одним из таких случаев является возникновение пожаров.

В работе разработана и обучена сверточная нейронная сеть для распознавания пожаров на изображениях с использованием библиотеки TensorFlow. Модель обучалась на 1880 изображениях свободного доступа. Изображения были разделены на два класса: пожар и его отсутствие. Обучение проводилось в течение 10 эпох, и функция потерь MSE показала точность 74,46 %. Для повышения эффективности использовались функции активации ReLU в скрытых слоях и Softmax в выходном слое для нормализации вероятностей классов [1, 2].

В качестве метода оптимизации использовался стохастический градиентный спуск, который способствовал снижению функции потерь и повышению точности предсказаний [1].

Тестирование показало, что модель успешно классифицирует большинство изображений, но испытывает сложности при различении пожаров и закатов из-за схожих цветовых оттенков, что требует расширения тренировочного набора данных более разнообразными примерами [2].