

Сервис также соблюдает все стандарты по безопасности, поэтому все сообщения передаются по защищенному соединению с сервером, что предотвращает возможность перехвата информации третьими лицами с помощью атаки Man-in-the-Middle.

И. Н. Громько

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **Е. А. Ружицкая**, канд. физ.-мат. наук, доцент

РЕАЛИЗАЦИЯ КАТАЛОГА WEB-ПРИЛОЖЕНИЯ ДЛЯ ОНЛАЙН-МАГАЗИНА КОСМЕТИКИ И ПАРФЮМЕРИИ

При разработке web-приложения магазина одним из основных компонентов является каталог. Каталог должен выполнять следующие функции: просмотр содержимого каталога и отдельных товаров, фильтрация и сортировка товаров, возможность их добавления в списки «Корзина» и «Избранное». Для реализации этого списка web-приложение требует наличия такой технологии, которая не только выполнит эти действия с достаточной плавностью и скоростью, но и сделает это с сохранением текущей позиции пользователя на сайте.

Для этого можно использовать технологию AJAX языка JavaScript. AJAX не только динамически обращается к серверу, но и позволяет обновлять не всю страницу, а только определенную ее часть.

Для обмена данными между сайтом и сервером можно использовать объект XMLHttpRequest. Чтобы отправить HTTP-запрос, после создания XMLHttpRequest отправляется POST-запрос. Серверная часть приложения обрабатывает входные данные и на их основе генерирует ответ в различных форматах (в конкретном случае на выход будет получен json-файл) или изменяет какую-либо информацию на сервере, например, данные в таблице БД. Для малых объемов входных данных можно использовать GET-запрос.

Серверная же часть приложения написана на PHP.

Поскольку технология AJAX использует асинхронную передачу данных, пользователю не придется ждать выполнения запроса, чтобы выполнять другие действия. Однако следует каким-либо образом оповестить пользователя, что сервер обрабатывает информацию, иначе работа запроса может произвести на пользователя ошибочное впечатление, что приложение «зависло».

Асинхронность функции нужно учитывать и при написании функций с использованием технологии AJAX. Так, для использования информации из базы данных приложения нужно сначала дождаться выполнения запроса, а только потом обработать ответ.

Р. Ю. Громько

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **Е. И. Сукач**, канд. техн. наук, доцент

АВТОМАТИЗАЦИЯ ОБМЕНА ДАННЫМИ С ПЛАТОЙ ESP-32 ПО BLUETOOTH

Одним из решений для создания беспроводной связи малого радиуса действия с возможностью передачи и получения данных можно считать один из самых популярных стандартов передачи данных между устройствами Bluetooth. С его помощью можно слушать музыку в беспроводных наушниках, управлять роботом-пылесосом или включить чайник на кухне из другой комнаты. Он находит достаточно широкое применение в радиоэлектронных проектах.

С другой стороны, микроконтроллеры ESP32, обладающие встроенными Wi-Fi и Bluetooth, контактами для аналого-цифрового преобразователя и цифро-аналогового преобразователя, поддержкой аудио устройств, спящего режима и многими другими возможностями, смотрятся весьма неплохо для конструирования различных проектов тематики интернета вещей (Internet of Things, IoT).

В докладе рассматриваются вопросы разработки мобильного приложения, которое позволяет обнаруживать и подключаться к платам ESP-32 через Bluetooth, а также обмениваться данными (отправка и приём) с выбранной платой.

Создан канал обмена данными посредством Bluetooth между микроконтроллером ESP-32-WROOM-DA и смартфоном на платформе Android. Поскольку устройства не похожи друг на друга, то для каждого из них были написаны отдельные скрипты.

Для смартфона было реализовано приложение в среде Android Studio на языке Kotlin. В нём передача данных реализована с помощью библиотеки android.bluetooth. С помощью BluetoothAdapter создан BluetoothDevice со стандартным для Bluetooth UUID адресом (00001101-0000-1000-8000-00805F9B34FB). По сути BluetoothDevice – это устройство и через него уже можно передавать данные.

Сама передача осуществляется через BluetoothSocket. Он создаётся при связывании по Bluetooth платы с телефоном. Чтение данных происходит через InputStream, а отправка через OutputStream. Обмен данными происходит байтами, которые расшифровываются и шифруются, соответственно.

BluetoothSocket является потокобезопасным. В частности, close() всегда будет немедленно прерывать текущие операции и закрывать сокет.

InputStream и OutputStream работают с байтами и для передачи данных с их использованием данные преобразовываются в байты. В реализованном приложении идёт работа с данными типа String.

Для поиска новых Bluetooth соединений или просмотра старых Bluetooth соединений в приложение реализован необходимый функционал с помощью класса BluetoothAdapter (рисунок 1). Для поиска новых устройств используется метод startDiscovery(). А для получения информации об старых соединениях используется метод bondedDevices, который имеет возвращаемый тип данных Set<BluetoothDevice>.

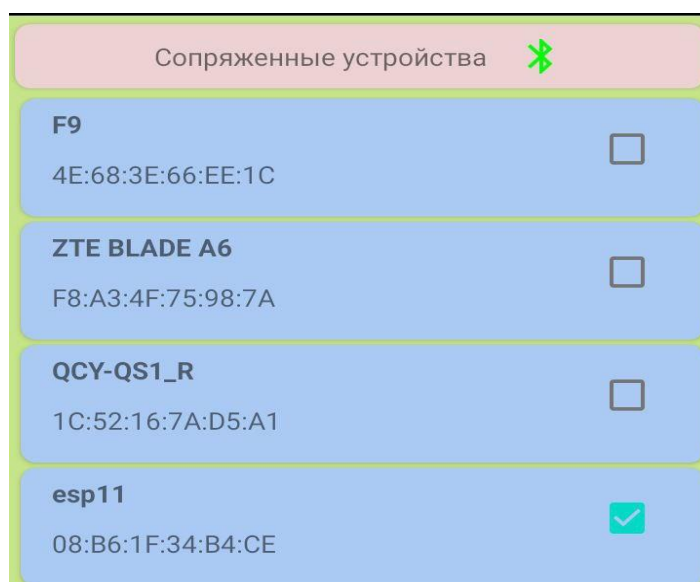


Рисунок 1 – Список сопряжённых устройств

Для платы ESP-32 было использовано приложение Arduino IDE. Работа с Bluetooth реализована с помощью библиотеки BluetoothSerial.h. Класс BluetoothSerial

позволяет создать источник сигнала, а также осуществлять обмен данными. С помощью метода `begin` происходит создание источника данных. А именно, в строке кода `SerialBT.begin("esp32");`, `SerialBT` – это название объекта класса `BluetoothSerial`, а `esp32` – название устройства в сети Bluetooth. Метод `print` позволяет отправлять данные, а `readString` – принимать.

Так как передача данных должна происходить не одновременно, то в программе присутствует бесконечный цикл, в котором оператор `if` фиксирует состояние передачи приёма данных по Bluetooth с помощью условия `SerialBT.available()`, которое срабатывает при передаче данных по Bluetooth. Для отправления данных, полученных с платы ESP-32 проверяется условие `Serial.available()`, что в отличие от `SerialBT` фиксирует наличие данных, подготовленных для отправки с платы. Чтобы не допустить чрезмерного количества проверок состояний, в цикле присутствует задержка на 20 миллисекунд, выполненная с помощью команды `delay(20)`.

Плата ESP32 оснащена встроенным датчиком Холла, который обнаруживает изменения магнитного поля в его окружении (рисунок 2). В данном проекте предусмотрена фиксация данных с этого датчика. Фиксация происходит в том же цикле обмена данных программы. Данные от датчика получаются с помощью команды `hallRead()`.

28/11/2023 10:34:04: 59
28/11/2023 10:34:08: 48
28/11/2023 10:34:12: 63
28/11/2023 10:34:16: 36
28/11/2023 10:34:20: 67
28/11/2023 10:34:24: 49

Рисунок 2 – Просмотр данных датчика Холла, полученных с ESP32

Разработанное приложение позволяет контролировать платы ESP-32 в зоне действия Bluetooth (зависит от устройства). Применение плат с контролем позволит через телефон дистанционно контролировать различные технические датчики из различных областей жизни, начиная от включения света дома, заканчивая просмотром состояния больного.

А. Д. Губанова

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **В. С. Смородин**, д-р техн. наук, профессор

О РАЗРАБОТКЕ КОНТРОЛИРУЮЩЕЙ СИСТЕМЫ

Работа посвящена разработке системы для контроля знаний студентов и иных пользователей по различным дисциплинам.

При открытии сайта отображается главная страница (рисунок 1).