

УДК 004.624, 004.77

**АВТОМАТИЗАЦИЯ ФИЗИЧЕСКОГО ЭКСПЕРИМЕНТА
ФОРМИРОВАНИЕМ И ПЕРЕДАЧЕЙ СЕРВЕРУ ЗАПРОСОВ HTTP
С ПОМОЩЬЮ МИКРОКОНТРОЛЛЕРА ESP8266****Т. П. ГУЛЕВИЧ***(Представлено: канд. физ.-мат. наук, доц. С. А. ВАБИЩЕВИЧ)*

Работа посвящена технологии беспроводной сети Wi-Fi, тесно связанной с устройствами интернета вещей (IoT), и используемой также в сфере автоматизации различных экспериментов, в том числе и физических. В данной статье рассматриваются вопросы использования HTTP запросов для обеспечения беспроводного физического эксперимента с помощью микроконтроллера ESP8266 на базе платформы NodeMcu v3. Внимание уделено написанию кода для отправки HTTP запроса на сервер с помощью вышеупомянутого микроконтроллера, а также программированию микроконтроллера.

Введение. По мере развития IoT количество устройств, подключенных к сети интернет, стремительно растет. Реализация подключения удаленного доступа стала возможной благодаря развитию множества технологий, в том числе и протокола передачи гипертекста (Hypertext Transfer Protocol, HTTP). Данный протокол является самым популярным выходом в интернет. Целью данной работы является разработка прототипа для оптимальной отправки и получение данных с микроконтроллера ESP8266 на локальный сервер.

Теоретическая часть. HTTP – это протокол, с помощью которого веб-браузер запрашивает и получает данные веб-страниц с сервера. Данный протокол является протоколом вопроса и ответа. Браузер формирует запрос соблюдая синтаксис HTTP, а сервер отвечает на него, передавая на веб-страницу, текст и прочее содержимое.

Протокол HTTP имеет несколько версий: HTTP/1, HTTP/2, HTTPS. Версия HTTP/1 является самой первой версией и изначально подразумевалась только для передачи текста, но позже с помощью данной версии начали передавать и изображения, аудио, видео и т.д. Так же в качестве текста можно понимать файлы HTML, PHP, CSS, JS и т.д. Версия HTTP/2 – это новая, продвинутая версия HTTP/1, которая появилась совсем не давно и в настоящее время все плавно переходят на неё. HTTPS, отличается тем, что в отличии от двух предыдущих она имеет сквозное шифрование то есть данные, шифруются отправителем и дешифруются получателем. Также стоит отметить, что сайты, работающие по протоколу HTTPS, имеют перенаправления – это значит, что если попытаться попасть на данный сайт по протоколу HTTP, то пользователь будет перенаправлен на версию HTTPS. Так же важным отличиям между HTTP и HTTPS являются порты TCP. TCP – это протокол соединения через IP адреса. Стандартными портами для HTTPS и HTTP являются 443 и 80, соответственно. Однако могут быть использованы и не стандартные. Далее, после установления соединения для передачи данных необходимо сформировать GET-запрос. Самым простым примером формирование GET запроса является набор в адресную строку URL-адреса сайта. Современные браузеры расставляют нужный индекс сами, так что для обычного пользователя совсем не обязательно знать индексы GET запросов [1].

Практическая часть. В практической части данной статьи будет использован локальный сервер, в связи с чем, для выполнения задач достаточно использования первой версии HTTP и 80 порта TCP. В локальном сервере нет необходимости шифровать данные, поскольку данные не выходят за пределы локальной сети и не попадают в глобальную сеть. В данной статье будет использоваться собранный прототип измерительного устройства, ранее представленный в работе [2].

Для начала нужно создать локальный сервер и расположить на нем файл, который будет содержать в себе веб-страницу, отображающую полученную информации. В данной работе для это использовалась бесплатная программная среда разработки Open Server Panel. На сервере находился файл index.php. Затем идет программирование микроконтроллера в программной среде Arduino IDE.

```
#include <ESP8266WiFi.h>
```

С помощью директивы препроцессора include подключаем нужную библиотеку для работы с технологией Wi-Fi.

```
const char* ssid = "*****";  
const char* password = "*****";  
const char* host = "192.168.***.***";  
const uint16_t port = 80;
```

Затем объявляем константные значения для имени Wi-Fi, к которому подключаемся, пароль – это ssid и password соответственно. Константе host задаем IP адрес сервера, а для константы port, номер порта по которому устанавливается соединение.

```
WiFi.begin(host, password);
```

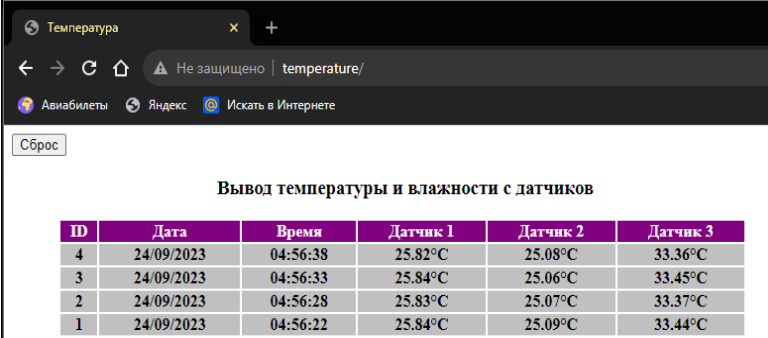
Дальше в функции setup (функция, которая выполняется только при запуске микроконтроллера), классу Wi-Fi, передаем константы логина и пароли сети Wi-Fi.

```
WiFiClient client;
if (client.connect(host, port)){
  client.print("GET /insert.php?");
  client.print("temp1=");
  client.print(temp1);
  client.print("&temp2=");
  client.print(temp2);
  client.print("&temp3=");
  client.print(temp3);
  client.println(" HTTP/1.0");
  client.print("Host: ");
  client.println(host);
  client.println("Connection: close");
  client.println();
  client.println();

  while (client.connected())
  {
    if (client.available()){ String line = client.readStringUntil('\n'); }
  }
  client.stop();
}
```

Далее в функции loop (которая выполняется бесконечное число раз) создаем класс WiFiClient с именем client и прописываем условия, которое гласит, что пока client подключен к сети, создается запрос, который выглядит следующим образом: *GET /insert.php? temp1= 'значение с первого датчика' &temp2= 'значение со второго датчика' &temp3= 'значение с третьего датчика' HTTP/1.0 Host: host Connection: close*. Данный запрос означает что мы вставляем в файл под названием 'insert.php' на место переменной temp1, temp2 и temp3 значения с первого, второго и третьего датчика соответственно, затем указываем версию протокола HTTP, хост и закрываем соединение. Далее идет цикл while, который работает пока клиент подключен, он принимает строку GET запроса. После останавливаем клиент. В конце ставим задержку чтобы снимать данные с датчиков через определенные промежутки времени. Задержку можно поставить, например с функции delay: *delay(5000)*, где 5000 – число миллисекунд.

Для демонстрации отправки и получения данных на сервер микроконтроллером ESP8266 использовалась веб-страница, включающая в себя таблицу (рис.). Количество строк которой постоянно увеличивается. Столбец ID – указывает порядковый номер измерения. Таблица отсортирована так, что самые старые значения будут указываться в конце таблицы, соответственно самые новые в начале (рисунок).



ID	Дата	Время	Датчик 1	Датчик 2	Датчик 3
4	24/09/2023	04:56:38	25.82°C	25.08°C	33.36°C
3	24/09/2023	04:56:33	25.84°C	25.06°C	33.45°C
2	24/09/2023	04:56:28	25.83°C	25.07°C	33.37°C
1	24/09/2023	04:56:22	25.84°C	25.09°C	33.44°C

Рисунок. – Пример веб-страницы, на которой изображены значения датчиков с микроконтроллеров

Заключение Рассмотренный метод отправки данных на сервер в виде HTTP запросов, с помощью микроконтроллера ESP8266. Может использоваться для создания различных измерительных приборов и установок на основе технологий IoT, что существенно облегчит проведение физических экспериментов.

ЛИТЕРАТУРА

1. Поллард Б. HTTP/2 в действии / пер. с англ. П. М. Бомбаковой. – М.: ДМК Пресс, 2021. – 424 с.: ил.
2. Использование беспроводной сети Wi-Fi в автоматизации физического эксперимента Т.П.Гулевич; науч. рук. С. А. Вабищевич // Актуальные вопросы физики и техники: материалы XII Респ. науч.-практ. конф. аспирантов, магистрантов и студентов (Гомель, 20 апр. 2023 г.) / Учреждение образования "Гомельский государственный университет имени Франциска Скорины". – Гомель: ГГУ им. Франциска Скорины, 2023. – С. 344-347.