

WEB-SERVER НА ARDUINO С АВТОРИЗАЦИЕЙ И ГРАФИЧЕСКИМ ПРЕДСТАВЛЕНИЕМ ИНФОРМАЦИИ С ДАТЧИКОВ

В работе на базе Arduino Mega и контроллера W5100 построен web-сервер для графического отображения данных удаленного клиента, полученных с датчиков температуры, давления, влажности. Программа сервера написана в среде разработки Arduino IDE. Откорректирована библиотека Ethernet для W5100, которая для Arduino IDE ver. 1.0.3, 1.0.5-r2 приводила к зависанию сервера. Показана возможность использования библиотеки Dugraphs для графической визуализации данных, полученных с помощью датчиков. Написаны скрипты для прорисовки графиков для спроектированного web-сервера. Проанализирована скорость передачи данных с web-сервера Arduino для различных сетевых контроллеров ENC28J60, W5100, W5500 для среды программирования Arduino IDE и библиотек UIPEthernet, Ethernet, Ethernet2. Показано, что с наименьшей скоростью данные передаются web-сервером с контроллером ENC28J60 - 3.3КБайт/с, с наибольшим контроллером W5500 - 23.4Кбайт/с. Отмечено, что эти сервера не поддерживают многопоточную работу. Поэтому они не могут быть использованы для создания миниатюрных универсальных web-серверов для обработки нескольких запросов одновременно. Рассмотренный в работе сервер может обслуживать только один запрос от одного удаленного клиента. Проанализированы скоростные параметры передачи данных для сервера на микроконтроллере ATmega328p (Arduino UNO) с контроллером сети ENC28J60. Программа сервера реализована на языке Си в среде программирования AVR Studio. Отмечена высокая скорость передачи данных - 140КБайт/с и возможность многопоточной работы. Установлено, что при одновременной передаче трех файлов разным клиентам суммарная скорость передачи достигает 120-130КБайт/с, а для каждого клиента 40-50КБайт/с. Показано, что использование такого сервера для решения задачи графического представления данных с датчиков затруднено вследствие сложности переноса программного обеспечения на другие микроконтроллеры и ограниченностью библиотеки для работы с картой microSD. Эксплуатация разработанного здесь сервера в течении трех лет показала высокую надежность его работы.

В работе рассмотрено создание web-сервера на Arduino, который имеет модернизированную HTTP basic authentication. Модернизация состоит в том, что для авторизации используется пароль из списка паролей, который выбирается пользователем на основании ключа, пересылаемого сервером. При каждом новом входе на сервер предыдущий пароль становится недействительным. Представлен практический пример web-сервера на Arduino Mega, контроллерами Ethernet: enc28j60 и w5500.

Ключевые слова: микроконтроллер, ENC28J60, W5100, W5500, Arduino Mega, Arduino UNO, ATmega2560, библиотека Dugraphs, AVR Studio, base64-encoded, basic authentication.

Постановка проблемы. Для управления удаленными объектами часто требуется не только получать детальную информацию с множества датчиков, но и отслеживать текущее состояние системы, динамику изменения состояния системы в течении заданного промежутка времени. Для решения такой задачи удобно использовать графическое представление величин, снимаемых с различных датчиков. Поскольку датчики находятся на удаленных объектах, наиболее дешевым решением является использование web-сервера на Arduino, к которому подключены необходимые датчики и который имеет доступ к сети Интернет. Со стороны клиента удобно использование браузера, который формирует графическое представление массива полученных данных с удаленного датчика. Для формирования графиков с сервера требуется пересылать достаточно большой объем информации. Т.е. кроме массива данных, накопленных с датчиков, пересылается программное обеспечение для браузера, которое позволяет браузеру формировать графики. Это требует, чтобы сервер Ардуино мог пересылать через Интернет не один пакет данных, а множество с приемлемой скоростью. Например, для формирования месячного графика температуры с интервалом 5 минут необходимо передать браузеру массив

данных размером примерно 200КБайт. Программа для формирования графика имеет размер примерно 125КБайт, которая должна быть передана с массивом данных. Таким образом для построения одного месячного графика web-серверу Ардуино требуется передать примерно 325КБайт информации.

С появлением технологий удаленного управления оборудованием через сеть Интернет с помощью микроконтроллеров (Ардуино) актуальной также стала задача авторизации на управляющих серверах. Однако в отличие от современных компьютеров Ардуино имеют скромные вычислительные ресурсы в связи с чем не в состоянии поддерживать полноценные сетевые протоколы, не говоря уже о программном обеспечении, обеспечивающих шифрование данных в пакетах. Несмотря на это рассмотрим возможность упрощенной удаленной авторизации на сервере Arduino. Для этого воспользуемся технологией NTTP authentication, описанной в стандартах NTTP 1.0/1.1.

Анализ последних исследований и публикаций. Решению задачи построения web-сервера на микроконтроллерах уделялось много внимания особенно с появлением первого аппаратного контроллера сети Ethernet 10Base-T - микросхемы enc28j60 [1-9]. Первая практическая работа по созданию такого сервера представлена в работе [8]. Однако этот сервер мог пересылать только один пакет, выполняя удаленное включение, выключение нескольких устройств. В этом проекте использовались микроконтроллеры avr - atmega88, atmega168 со скромными ресурсами. В 2011г. появилась работа [2] в которой был представлен модернизированный протокол tcp/ip, позволяющий с использованием микросхемы enc28j60 и микроконтроллера atmega328r создать web-сервер с многопоточным режимом работы, пересылающий множество пакетов с высокой скоростью (до 140кбайт/с). Подключение карты microSD позволяло использовать этот микроконтроллер для создания web-сервера, содержащего достаточно большой объем статических html страниц с фотографиями. Дальнейшее развитие получил web-сервер на базе микроконтроллера atmega2560 и контроллера сети w5100(w5500) [10-14]. Однако эти сервера до настоящего времени не в состоянии были отображать графическое представление показаний датчиков. С появлением библиотеки Dugraphs[3,7], эффективных библиотек Ethernet.h, Ethernet2.h[10-12] среды Arduino IDE позволили создать web-сервер на базе atmega2560 для формирования графического представления. Однако малые ресурсы микроконтроллеров не позволяют создать авторизированный вход на web-сервер. В работе [16] рассматривалась попытка создания доступа по паролю с использованием GET, POST запросов. А в работе [15-19] - авторизация с использованием схемы basic. Однако использование нешифрованного протокола NTTP позволяло легко перехватить используемые логин и пароль с помощью программ - снифферов.

Цель работы - разработка аппаратных и программных средств, реализующих web-сервер на базе микроконтроллеров avr с авторизацией доступа по циклически меняющимся паролям и графическим представлением информации с датчиков через сеть Интернет.

Основной материал исследований. Предположим, что канал Интернет имеет пропускную способность 4Мбит/с - 500КБайт/с (например, самая низкая скорость подключенного к телефонной линии цифрового модема). Рассмотрим скорость передачи данных web-серверами Ардуино, построенные на наиболее распространенных и доступных сетевых контроллерах и библиотеках к ним.

На рисунке 1 представлены фото контроллеров Ethernet с микросхемами: ENC28J60, W5100, W5500.



Рисунок 1 – Контроллеры Ethernet ENC28J60, W5100, W5500

На рис. 2 представлены стенды исследуемых web-серверов Arduino Mega с установленными, показанными выше, контроллерами сети.



Рисунок 2 – Web-сервера Arduino Mega с сетевыми контроллерами ENC28J60, W5100, W5500

Для определения скоростей передачи данных использовалась программа универсального web-сервера, представленная в источнике [1]. Эта программа по запросу браузера считывает html документы с карты microSD и передает их клиенту. Причем сервер корректно работает как с текстовыми документами, так и с изображениями и исполняемыми файлами. На рисунке 3 показаны скоростные диаграммы работы серверов. Для работы с сетевыми контроллерами в среде Arduino IDE использовались популярные библиотеки UIPEthernet (ENC28J60), Ethernet (W5100) и Ethernet2 (W5500).

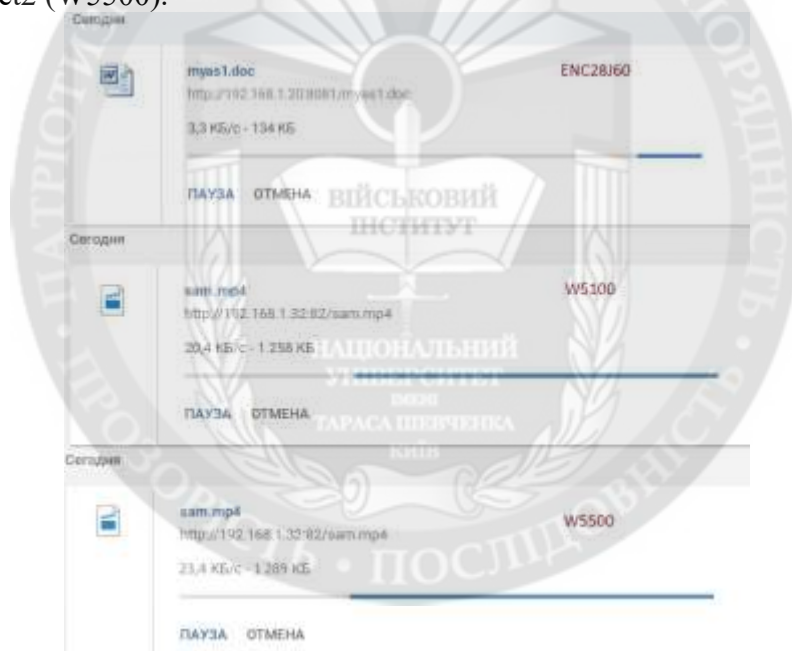


Рисунок 3 – Скоростные диаграммы работы серверов для разных контроллеров сети

Из рис. 3 видно, что максимальная скорость передачи данных 23.4Кбайт/с соответствует контроллеру W5500, минимальная 3.3Кбайт/с - контроллеру ENC28J60. Сервер Arduino с контроллером W5500 на передачу данных для формирования месячного графика потратит $325/23.4 = 13.9$ секунд. Недостатком серверов с указанными библиотеками является то, что они работают только с одним потоком данных. Т.е. одновременно копировать с них два файла (например, текст и рисунок) не получится. Второй файл будет ожидать очереди, пока первый полностью не будет скопирован. Также опыт показал, что библиотека Ethernet для W5100, поставляемая с Arduino IDE ver. 1.0.5-r2 работает некорректно, сервер зависает, поэтому в работе ниже будет дана корректировка этой библиотеки. В последних версиях Arduino IDE (например, ver. 1.8.6.) ошибка исправлена и сервер работает устойчиво, но также не поддерживает многопоточной работы, а скорость передачи данных несколько ниже, чем в исправленной библиотеке Ethernet.

Рассмотрим еще один проект web-сервера на Arduino. Он реализован в среде программирования AVR Studio на языке C. Здесь используется микроконтроллер ATmega328p (Arduino UNO, Arduino Nano) и контроллер сети ENC28J60. Технология его работы и программное обеспечение подробно описаны в работе [2]. На рисунке 4 показана фотография воспроизведенного из источника [2] сервера, который использует контроллер Arduino UNO.

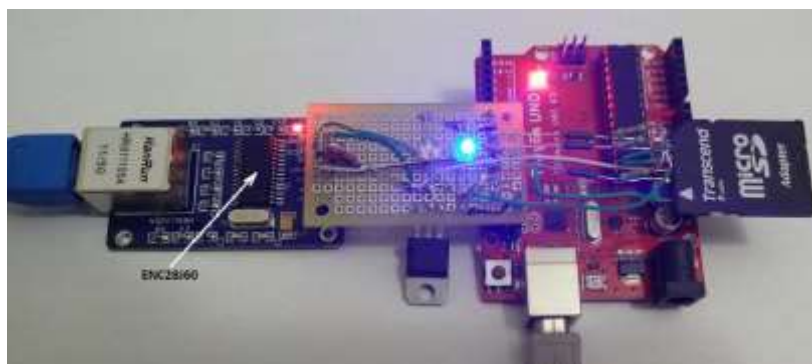


Рисунок 4 – Web - сервер на базе Arduino UNO и ENC28J60

На рис. 5 показаны скоростные диаграммы работы этого сервера. Видно, что он допускает многопоточную обработку, а скорость одного потока достигает 140Кбайт/с. Причем, максимальное количество потоков устанавливается программным путем. Поэтому этот сервер является наиболее подходящим для решения задачи графического представления массива данных, накопленного на карте microSD. Может также быть использован в качестве универсального миниатюрного web-сервера для пересылки html документов с карты microSD и управления оборудованием.



Рисунок 5 – Скоростные диаграммы работы многопоточного сервера Arduino UNO

В данной работе для построения сервера для графического представления данных будут использоваться Arduino Mega с контроллером ATmega2560, контроллер сети W5100, среда разработки Arduino IDE ver. 1.0.3, 1.0.5-r2 с модернизированной библиотекой Ethernet для W5100. Такой сервер позволит формировать недельные графики с временем ожидания примерно 15-20 секунд. Более предпочтительным является сервер, представленный на рисунке 4, однако программирование его для задач с дополнительными серверными функциями является более сложной задачей, чем создание сервера на основе библиотек, включенных в состав среды программирования Arduino IDE. Дополнительно к этому сервер на рис. 4 написан под конкретный микроконтроллер и переход к другой модели требует изменения кода, а библиотека для работы с картами microSD поддерживает только формат карт SD (до 2Гбайт), которые в настоящее время не выпускаются. Например, microSD карты формата SDHC, имеющие емкость 8 и более Гбайт, не поддерживаются.

Использование сервера с контроллером сети W5100 связана с тем, что используется простая и отработанная универсальная среда разработки Arduino IDE, встроенная библиотека для работы с картами microSD (поддерживает формат SDHC), откорректированная устойчиво работающая библиотека Ethernet, относительно небольшой объем пересылаемых данных. В случае необходимости пересылки большого объема информации целесообразно использовать сервер, представленный на рис. 4 и описанный в работе [2].

На рис. 6 представлен экспериментальный стенд работающего сервера, который получает данные с датчиков температуры DS18B20, давления и температуры MS5611, влажности и температуры DHT11, записывает данные на карту microSD каждые 300 сек. и пересылает все записанные данные при запросе конкретного массива данных браузеру клиента вместе с программой dygraph (java-скрипт), которая запускается браузером и формирует график.

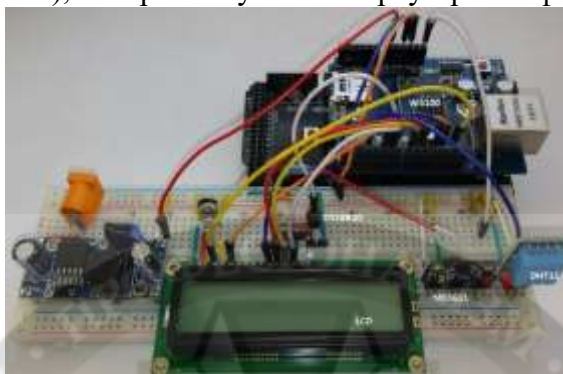


Рисунок 6 – Экспериментальный стенд web-сервера для формирования графиков

Рассмотренный сервер использует библиотеку dygraph [3], и формирует графики температуры вне помещения (DS18B20), внутри помещения (MS5611), атмосферного давления (MS5611) и влажности воздуха (DHT11). Web - server позволяет формировать графики в течение 3-х суток. Для этого существует страничка "текущие показания датчиков, настройки" в которой устанавливается время и 3-и последовательные даты, для которых требуются графики. После 3-х суток в файлы данных информация с датчиков заноситься не будет. Так как данные с датчиков добавляются в файлы, то после перезагрузки сервера можно внести следующие последовательные 3-и даты. Таким образом на графиках будет отображено еще дополнительно трое суток. Для построения новых графиков необходимо карту microSD переставить на компьютер и удалить данные с файлов datat.txt, datata.txt, datar.txt, datah.txt, содержимое которых будет рассмотрено ниже. На рис. 7 представлены интерфейсы главной страницы сервера, график температуры вне помещения и текущие показания датчиков с установкой времени и даты.

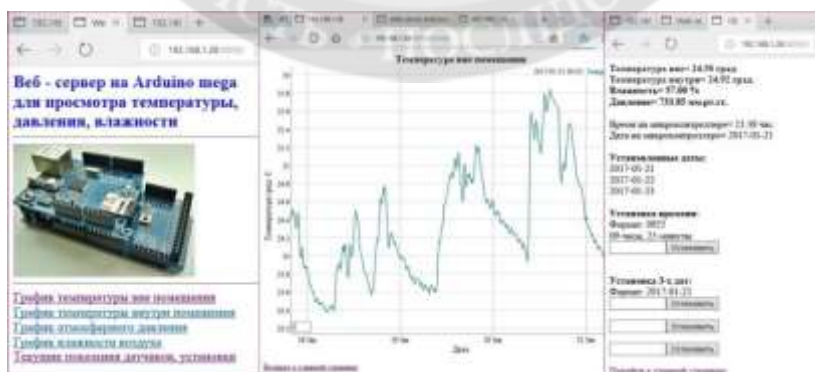


Рисунок 7 – Интерфейсы главной страницы сервера, график температуры вне помещения и текущие показания датчиков с установкой времени и даты

Рассмотрим принцип работы сервера, выполняющего визуализацию графиков. При запросе клиента по адресу сервера (<http://192.168.1.38:9000>), сервер обращается к microSD карте и считывает файл index.htm, который пересылается браузеру. При переходе по ссылке, например <http://192.168.1.38:9000/in0.htm>, сервер считывает файл in0.htm с microSD и пересылает

этот html документ браузеру, который его исполняет. Исключение составляет переход по ссылке <http://192.168.1.38:9000/temp>. В этом случае открывается страничка, которая представляет текущие показания датчиков с установкой времени и даты (рис.7). Данные с microSD для этой ссылки не считываются.

Dygraphs - это библиотека, подключаемая к любой программе (скрипту), написанной на языке JavaScript. Таким образом HTML-документ превращается в инструмент графической визуализации. Для этого необходимо подключить библиотеку к скрипту и создать объект Dygraph в соответствии с правилами языка. Каждый такой объект в общем случае представляет собой диаграмму, обладающую соответствующими признаками – легендой, наименованиями осей и т.д. Скрипт с библиотекой, которые находятся на браузере, запрашивают набор данных в виде файла в формате CSV с сервера при помощи запроса XHR. Обозначение параметров на графике выбирается со скрипта. Если прорисовывается несколько кривых на графике, то цвет линий выбирается библиотекой автоматически так, чтобы визуально кривые отличались друг от друга. Обозначения по оси X будут меняться при масштабировании - например, если по умолчанию по оси аргументов идут месяцы года, то при увеличении диаграммы будут появляться сначала недели месяца, а затем дни и потом часы. Для правильного функционирования библиотеки Dygraphs на карте microSD сервера должны присутствовать следующие файлы:

1. [index.htm](#)

```
<HTML>
<HEAD>
  <meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
  <TITLE>Web server Arduino w5100</TITLE>
</HEAD>
<BODY>
  <B><FONT color=blue size="5">Веб - сервер на Arduino mega<br> для просмотра температуры,<br> давления, влажности</FONT></B>
  <hr>
  <a href="web_gr.jpg"></a>
  <hr>
  <FONT size="4"><A href="in0.htm">График температуры вне помещения</A></FONT>
  <br><FONT size="4"><A href="in1.htm">График температуры внутри помеще-
  ния</A></FONT>
  <br><FONT size="4"><A href="in2.htm">График атмосферного давления</A></FONT>
  <br><FONT size="4"><A href="in3.htm">График влажности воздуха</A></FONT>
  <br><FONT size="4"><A href="temp">Текущие показания датчиков, уста-
  новки</A></FONT>
</body>
</html>
```

2. [in0.htm](#)

```
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
  <script type="text/javascript"
  src="dygm.js"></script>
```

```

<link rel="stylesheet" href="dygraph.css">
</head>
<body>
<div id="graphdiv2"
  style="width:800px; height:600px;"></div>
<script type="text/javascript">
  g2 = new Dygraph(
    document.getElementById("graphdiv2"),
    "DATAT.TXT", // path to CSV file
    {
title: 'Температура вне помещения',
xlabel: 'Дата',
ylabel: 'Температура град. С',
showRoller: true,
rollPeriod: 5,
//color: "#0000ff",
labels: [ "Date", "Temp" ],
strokeWidth: 1.5
}
  );
</script>
<br>
<A href="">Возврат к главной странице</A>
</body>
</html>

```

3. [in1.htm](#) - скрипт для формирования графика температуры внутри помещения, аналогичный in0.htm

4. [in2.htm](#) - скрипт для формирования графика атмосферного давления, аналогичный in0.htm

5. [in3.htm](#) - скрипт для формирования графика относительной влажности воздуха, аналогичный in0.htm

6. [dygraph.min.js](#) - файл библиотеки dygraphs (переименован в dygm.js, так как имя файла должно быть не более 8 символов, расширение - не более 3-х символов в файловой системе FAT)

7. [dygraph.css](#) - файл библиотеки dygraphs

8. Файлы данных показаний датчиков, представленных в таблице 1. Эти файлы при первом включении сервера создаются автоматически. В них могут дописываться данные при повторном включении сервера, поэтому график будет продолжаться. Например, если сервер возобновит работу через 10 дней, то эти дни будут отображены графиком в виде прямой линии, соединяющей конец предыдущей кривой с началом следующей.

При включении сервера и последующей его работе каждые 5 минут в файлы `datat.txt`, `datata.txt`, `datar.txt`, `datah.txt` записываются построчно данные, как представлено в таблице 1.

Таблица 1

DATAT.TXT	DATATA.TXT	DATAP.TXT	DATAH.TXT
2017-01-17 20: 4, 24.56	2017-01-17 20: 4, 24.92	2017-01-17 20: 5, 722.4	2017-01-17 20: 5, 40
2017-01-17 20: 9, 24.56	2017-01-17 20: 9, 24.89	2017-01-17 20: 9, 722.4	2017-01-17 20:10, 40
2017-01-17 20:14, 24.44	2017-01-17 20:14, 24.84	2017-01-17 20:14, 722.3	2017-01-17 20:15, 40
2017-01-17 20:19, 24.50	2017-01-17 20:19, 24.87	2017-01-17 20:19, 722.3	2017-01-17 20:20, 41
2017-01-17 20:24, 24.44	2017-01-17 20:24, 24.86	2017-01-17 20:24, 722.3	2017-01-17 20:25, 41
2017-01-17 20:29, 24.44	2017-01-17 20:29, 24.82	2017-01-17 20:29, 722.3	2017-01-17 20:30, 41
...

В эти файлы пишутся дата вдоль оси X, температура вне помещения(datat.txt), температура внутри помещения(datata.txt), атмосферное давление в мм.рт.ст.(datap.txt), относительная влажность в %(datah.txt) по оси Y.

Программа сервера с подключенным к нему LCD индикатором для отображения текущего состояния датчиков, представлена в источнике [4].

Программа работает следующим образом.

1. Каждую секунду идет обращение к функции прерывания ISR (TIMER5_COMPA_vect), реализующую работу часов. Она вычисляет минуты, часы, дни и рассчитывает дополнительные параметры, которые позволяют корректно обрабатывать некоторые временные фрагменты программы.

2. В цикле loop() выполняются следующие действия:

2.1. Выполняется запись данных на карту microSD. Например для датчика температуры вне помещения запись на microSD выполняется следующими операторами:

if(time==299 && endn==0 && endgraf==0) // При совпадении времени пишем в файл каждые 300 сек. и так до достижения 24 часов 3-го дня

{ endn=1; // Чтобы была только 1 запись при time=299

myFile1 = SD.open("datat.txt", FILE_WRITE);

if(myFile1) { sensors.requestTemperatures(); cel=sensors.getTempCByIndex(0);

dtostrf(cel,6,2,celbuf); dtostrf(ho,2,0,hobuf); dtostrf(mi,2,0,mibuf);

if(day==0) myFile1.print(fd0); else if(day==1) myFile1.print(fd1);

else if(day==2) myFile1.print(fd2); myFile1.print(" ");

myFile1.print(hobuf); myFile1.print(":"); myFile1.print(mibuf);

myFile1.print(","); myFile1.println(celbuf);

myFile1.close(); } ... }

2.2. Открывается сетевое соединение (socket) и прослушивается. Если соединение установлено (ESTABLISHMENT), определяется длина поля данных пакета TCP. Если длина равна нулю и так 25 раз подряд, срочно принудительно на сервере закрывается соединение, иначе контроллер W5100 "зависнет" (вероятно аппаратная ошибка). После этого опять открывается соединение и так до тех пор, пока длина поля данных не будет больше нуля. Далее идет считывание заголовка, посылаемого браузером до достижения пустой строки, которая определяет конец заголовка при запросе GET. Если первой строкой заголовка является строка

GET / HTTP/1.1

то данные считываются с карты microSD с файла index.htm и пересылаются браузеру в виде html документа.

Если первой строкой являются строки

GET /temp?c= HTTP/1.1, GET /temp?d= HTTP/1.1

то выполняется распознавание времени и 1-й даты при их вводе в форму следующими операторами:

if(strstr(clientline, "GET /temp?c=") !=0) { // Распознаем время

fh[0]=clientline[12]; fh[1]=clientline[13]; fh[2]=0;

fm[0]=clientline[14]; fm[1]=clientline[15]; fm[2]=0; ho=atoi(fh); mi=atoi(fm);

}

if(strstr(clientline, "GET /temp?d=") !=0) { // Распознаем 1-ю дату

fd0[0]=clientline[12]; fd0[1]=clientline[13]; fd0[2]=clientline[14]; fd0[3]=clientline[15];


```
fd0[4]=clientline[16]; fd0[5]=clientline[17]; fd0[6]=clientline[18];fd0[7]=clientline[19];  
fd0[8]=clientline[20]; fd0[9]=clientline[21];fd0[10]=0; }
```

Если первой строкой является строка

```
GET /temp HTTP/1.1
```

то на в окне браузера распечатываются текущие показания датчиков, формы ввода времени и даты (рис. 7). Программа содержит также функцию `meter()`, которая распечатывает на экране LCD сервера текущие показания датчиков, обновляемые каждые 18 секунд.

Для предотвращения "зависания" сервера стандартная библиотека Ethernet в Arduino IDE ver. 1.0.3, 1.0.5-r2 была откорректирована. В источнике [5] подробно рассматривается содержимое измененных файлов:

```
socket.cpp, EthernetClient.cpp, EthernetClient.h.
```

Внесены изменение в функцию `void EthernetClient::stop()` - (`client.stop()`) для файла `EthernetClient.c`.

Рассмотрим HTTP authentication. Применительно к web-сайтам он работает следующим образом:

1. Сервер, при обращении неавторизованного клиента к защищенному ресурсу, отправляет HTTP статус "401 Unauthorized" и добавляет заголовок "WWW-Authenticate" с указанием схемы и параметров аутентификации.

2. Браузер, при получении такого ответа, автоматически показывает диалог ввода username и password. Пользователь вводит детали своей учетной записи.

3. Сервер аутентифицирует пользователя по данным из этого заголовка.

Весь процесс стандартизирован и поддерживается всеми браузерами и web-серверами. Существует несколько схем аутентификации. Для случая сервера на Ардуино рассмотрим Basic. Здесь username и password пользователя передаются в заголовке Authorization в незашифрованном виде (base64-encoded).

Рассмотрим схему аутентификации basic, и впоследствии модифицируем ее для веб-сервера Arduino. Механизм авторизации выглядит следующим образом. Сначала при запросе на подключение к серверу Arduino браузер посылает заголовок без запроса авторизации:

```
GET / HTTP/1.1
```

```
Host: 192.168.1.18:81
```

```
Connection: keep-alive
```

```
Cache-Control: max-age=0
```

```
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
```

```
Upgrade-Insecure-Requests: 1
```

```
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko)
```

```
Chrome/44.4.2403.3 Amigo/44.4.2403.3 MRCHROME SOC Safari/537.36
```

```
Accept-Encoding: gzip, deflate, sdch
```

```
Accept-Language: ru-RU,ru;q=0.8,en-US;q=0.6,en;q=0.4
```

Сервер Ардуино отвечает браузеру заголовком на необходимость запроса с авторизацией:

```
HTTP/1.0 401 Unauthorized
```

```
WWW-Authenticate: Basic realm="Arduino - HOME"
```

Браузер Amigo посылает серверу запрос но уже с авторизацией:

```
GET / HTTP/1.1
```

```
Host: 192.168.1.18:81
```

```
Connection: keep-alive
```

```
Cache-Control: max-age=0
```

```
Authorization: Basic YWwleDoxMzY=
```

```
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
```

Upgrade-Insecure-Requests: 1

User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko)

Chrome/44.4.2403.3 Amigo/44.4.2403.3 MRCHROME SOC Safari/537.36

Accept-Encoding: gzip, deflate, sdch

Accept-Language: ru-RU,ru;q=0.8,en-US;q=0.6,en;q=0.4

Сервер Arduino посылает ответ и после пустой строки html документ

HTTP/1.0 200 OK

Content-Type: text/html

<Здесь идет передача браузеру html документа>

YWxleDoxMzY= -это закодированный в Base64 набор символов alex:136 (alex - это login, 136 - password, которые вводятся в выскакивающем у браузере окне авторизации). На рисунке 8 представлены интерфейсы ввода логина и пароля и ответа сервера Arduino. Программа сервера Ардуино с описанной авторизацией представлена в источнике [3]. В этой программе работа сервера по представленной схеме реализуется с помощью операторов:

```
if (readString.lastIndexOf("YWxleDoxMzY=")>-1) {  
  if (readString.lastIndexOf("GET /favicon.ico")>-1) {  
    client.println("HTTP/1.0 404 Not Found");  
  } else html_doc(client); }  
else { client.println("HTTP/1.0 401 Unauthorized");  
  client.println("WWW-Authenticate: Basic realm=\"Arduino - HOME\");}
```

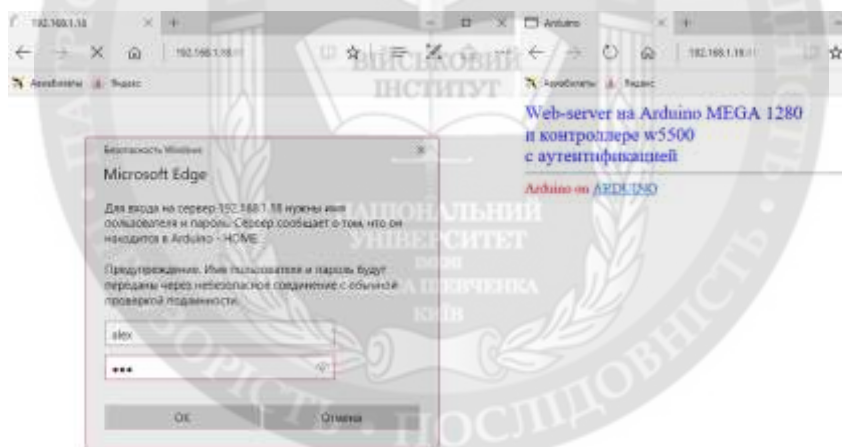


Рисунок 8 – Интерфейса ввода логина и пароля и ответа сервера Arduino

Однако безопасность при этой схеме весьма низкая. Пакеты по Интернет передаются в открытом виде, поэтому закодированный логин и пароль (YWxleDoxMzY=) можно легко выловить с помощью программ sniffеров и раскодировать с помощью Base64 decode. Модернизируем программу, чтобы в большей степени защитить сервер Ардуино для предотвращения несанкционированного управления им. Для этого создадим список паролей, которые известны пользователю сервера. Причем при каждом новом входе на сервер пароль доступа должен меняться на следующий по списку. Поэтому, если предыдущий пароль будет расшифрован, то он становится недействительный и злоумышленник не сможет зайти по нему. В источнике [14] представлена программа web-сервера Ардуино, на котором установлены три светодиода, имитирующие включение-выключение 3-х силовых источников питания(например электро-розеток), датчик температуры DS18B20(температура улицы) и датчик влажности и температуры DHT 11 (делает измерения в помещении) или датчик давления и температуры BMP280. Программа использует 3 пары логин: пароль:

1. alex:136;
2. alex:138;
3. alex:140

При каждом новом обращении к серверу пароль циклически меняется на следующий в списке. Для того, чтобы знать следующий номер пароля в браузере выводится параметр index. Его пользователь должен запомнить или определить с помощью открытого входа по адресу <http://192.168.1.18:81/index>. Таким образом, чем больше паролей в списке, тем более защищенным должен выглядеть сервер. На рисунке 9 представлен интерфейс управления сервером на Arduino Mega.

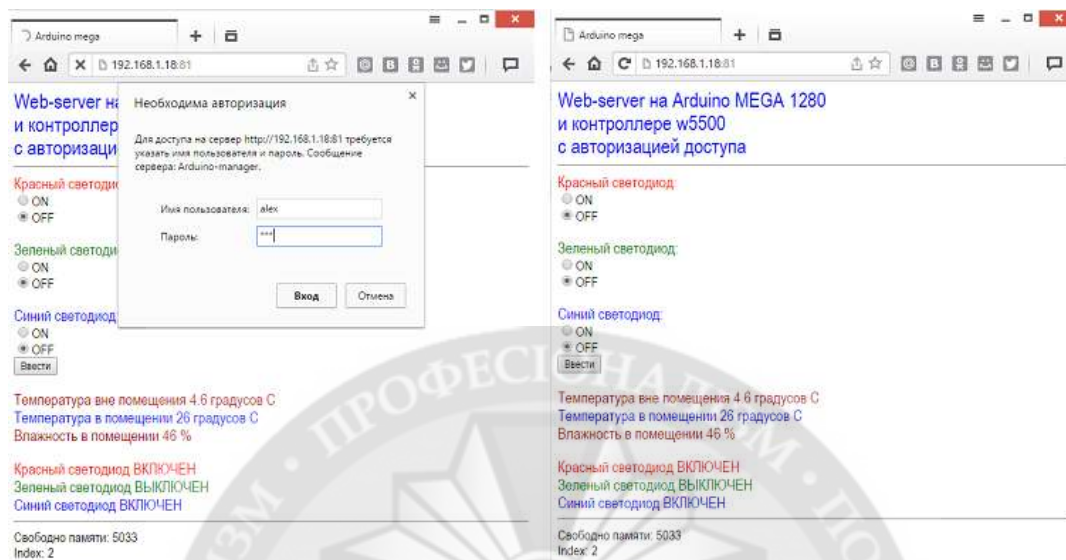


Рисунок 9 – Интерфейс управления web-сервером на Arduino Mega

В представленном интерфейсе Index:2 обозначает то, что при следующем входе на сервер необходимо для авторизации использовать пару alex:138.

Выводы

1. Проанализирована скорость передачи с web-сервера Arduino для различных сетевых контроллеров ENC28J60, W5100, W5500 для среды программирования Arduino IDE[6] и библиотек UIPEthernet, Ethernet, Ethernet2. С наименьшей скоростью данные передаются сервером с контроллером ENC28J60 - 3.3КБайт/с, с наибольшей контроллером W5500 - 23.4Кбайт/с. Отмечено, что эти сервера не поддерживают многопоточную работу.

2. Проанализированы скоростные параметры передачи данных для сервера на микроконтроллере ATmega328p(Arduino UNO) с контроллером сети ENC28J60. Программа сервера реализована на языке Си в среде программирования AVR Studio[2]. Отмечена высокая скорость передачи данных - 140КБайт/с и возможность многопоточной работы.

3. Показана возможность использования библиотеки Dugraphs[3] для графической визуализации данных, полученных с помощью датчиков. Написаны скрипты для прорисовки графиков для спроектированного сервера.

4. На базе Arduino Mega и контроллере W5100 построен web-сервер для графического отображения данных удаленного клиента, полученных с датчиков температуры, давления, влажности. Программа сервера написана в среде разработки Arduino IDE на языке Wiring (C++).

5. Откорректирована библиотека Ethernet для W5100, которая для Arduino IDE ver. 1.0.3, 1.0.5-r2 приводила к зависанию сервера.

6. Показано, что HTTP авторизация Basic передает username и password пользователя в незашифрованном виде (base64-encoded). И если не используется протокол HTTPS (HTTP over SSL), является относительно небезопасной.

7. Представленная в работе модернизация схемы авторизации Basic позволяет относительно безопасно управлять сервером Ардуино, но требует использования нескольких паролей, каждый из которых сообщается сервером пользователю через специальный параметр index.

ЛИТЕРАТУРА:

1. Мясичев А.А. Универсальный web - сервер на Arduino. [Electronic resource]. - Mode of access: http://webstm32.sytes.net/web_1.htm, 2013.
2. Подключение микроконтроллера к локальной сети. [Electronic resource]. - Mode of access: <http://we.easyelectronics.ru/electro-and-pc/podklyuchenie-mikrokontrollera-k-lokalnoy-seti.html>, 2011.
3. Dan Vanderkam. Dygraphs library. [Electronic resource]. - Mode of access: <http://dygraphs.com>, 2017.
4. Мясичев А.А. Web-сервер на Arduino для представления графиков температуры, давления, влажности. [Electronic resource]. - Mode of access: https://sites.google.com/site/webstm32/web_graph, 2018.
5. Мясичев А.А. Управление через Интернет голосовыми командами Web - сервером на Arduino Mega и контроллере W5100 с файлами на SD. [Electronic resource]. - Mode of access: https://sites.google.com/site/webstm32/web_server_speech, 2015.
6. Arduino. Официальный сайт. [Electronic resource]. - Mode of access: <https://www.arduino.cc/>, 2018.
7. Филипова Е. Инструменты визуализации данных: Dygraphs. [Electronic resource]. - Mode of access: <http://datareview.info/article/instrumentyi-vizualizatsii-dannyih-dygraphs>, 2014.
8. Мясичев А.А. Создание надежного http-сервера для удаленного управления по TCP/IP сети на основе ATmega1280 и Wiznet W5100 //Збірник наукових праць Військового інституту Київського національного університету імені Тараса Шевченка. - К. ВІКНУ, 2014. - Вип. №46 - С. 59 - 70.
9. ENC28J60 Stand-Alone Ethernet Controller with SPI Interface. [Electronic resource]. - Mode of access: <http://ww1.microchip.com/downloads/en/DeviceDoc/39662e.pdf>, 2012.
10. W5100 Datasheet. [Electronic resource]. - Mode of access: http://www.hobbytronics.co.uk/datasheets/W5100_Datasheet_v1_1_6.pdf, 2008.
11. W5500 Datasheet Version 1.0.7. [Electronic resource]. - Mode of access: http://wizwiki.net/wiki/lib/exe/fetch.php?media=products:w5500:w5500_ds_v108e.pdf, 2013.
12. Web Server. [Electronic resource]. - Mode of access: <https://www.arduino.cc/en/Tutorial/Web-Server>, 2015.
13. Arduino SD Card Web Server. [Electronic resource]. - Mode of access: <https://startingelectronics.org/tutorials/arduino/ethernet-shield-web-server-tutorial/SD-card-web-server/>, 2013.
14. Мясичев А.А. Web-сервер на Arduino MEGA и контроллере сети w5500 с авторизацией доступа для удаленного управления. [Electronic resource]. - Mode of access: <https://sites.google.com/site/webstm32/web-server-avtorizaciej>, 2018.
15. Анисимов В.В. Криптографические методы защиты информации. [Electronic resource]. - Mode of access: <https://sites.google.com/site/anisimovkhv/learning/kripto/lecture>, 2012.
16. Мясичев А.А. Авторизация на web-сервере Arduino с помощью GET и POST запросов. Materials of the XIII International scientific and practical Conference Modern scientific potential - 2018 , February 28 - March 7 , 2018 Construction and architecture. Mathematics. Modern information technology. Technical science. Physics. : Sheffield. Science and education LTD с. 36-42
17. Мясичев А.А. WEB-SERVER НА ARDUINO ДЛЯ ГРАФИЧЕСКОГО ПРЕДСТАВЛЕНИЯ ДАННЫХ С ДАТЧИКОВ ЧЕРЕЗ ИНТЕРНЕТ. [Electronic resource]. - Mode of access: https://www.researchgate.net/publication/332257299_WEB-SERVER_NA_ARDUINO_DLA_GRAFICESKOGO_PRED-STAVLENIA_DANNYH_S_DATCIKOV_CEREZ_INTERNET, 2019.
18. HTTP/TCP with an atmega88 microcontroller (AVR web server). Guido Socher. [Electronic resource]. - Mode of access: <http://www.tuxgraphics.org/electronics/200611/article06111.shtml>, 2006
19. Kitsum. Авторизация на Web сервере микроконтроллера. [Electronic resource]. - Mode of access: <https://it4it.club/topic/13-авторизация-на-web-сервере-микроконтроллера/>, 2015

REFERENCES:

1. Myasishchev A.A. Universal'nyj web - server na Arduino. [Electronic resource]. - Mode of access: http://webstm32.sytes.net/web_1.htm, 2013.
2. Podklyuchenie mikrokontrollera k lokal'noj seti. [Electronic resource]. - Mode of access: <http://we.easyelectronics.ru/electro-and-pc/podklyuchenie-mikrokontrollera-k-lokalnoy-seti.html>, 2011.
3. Dan Vanderkam. Dygraphs library. [Electronic resource]. - Mode of access: <http://dygraphs.com>, 2017.

4. Myasishchev A.A. Web-server na Arduino dlya predstavleniya grafikov temperatury, davleniya, vlazhnosti. [Electronic resource]. - Mode of access: https://sites.google.com/site/webstm32/web_graph, 2018.
5. Myasishchev A.A. Upravlenie cherez Internet golosovymi komandami Web - serverom na Arduino Mega i kontrollere W5100 s fajlami na SD. [Electronic resource]. - Mode of access: https://sites.google.com/site/webstm32/web_server_speech, 2015.
6. Arduino. Oficial'nyj sajt. [Electronic resource]. - Mode of access: <https://www.arduino.cc/>, 2018.
7. Fillipova E. Instrumenty vizualizacii dannyh: Dygraphs. [Electronic resource]. - Mode of access: <http://datareview.info/article/instrumentyi-vizualizatsii-dannyih-dygraphs>, 2014.
8. Myasishchev A.A. Sozdanie nadezhnogo http-servera dlya udalennogo upravleniya po TCP/IP seti na osnove ATmega1280 i Wiznet W5100 //Zbirnik naukovih prac' Vijs'kovogo institutu Kiïvs'kogo nacional'nogo universitetu imeni Tarasa SHEvchenka. - K. VIKNU, 2014. - Vip. №46 - S. 59 - 70.
9. ENC28J60 Stand-Alone Ethernet Controller with SPI Interface. [Electronic resource]. - Mode of access: <http://ww1.microchip.com/downloads/en/DeviceDoc/39662e.pdf>, 2012.
10. W5100 Datasheet. [Electronic resource]. - Mode of access: http://www.hobbytronics.co.uk/datasheets/W5100_Datasheet_v1_1_6.pdf, 2008.
11. W5500 Datasheet Version 1.0.7. [Electronic resource]. - Mode of access: http://wizwiki.net/wiki/lib/exe/fetch.php?media=products:w5500:w5500_ds_v108e.pdf, 2013.
12. Web Server. [Electronic resource]. - Mode of access: <https://www.arduino.cc/en/Tutorial/Web-Server>, 2015.
13. Arduino SD Card Web Server. [Electronic resource]. - Mode of access: <https://startingelectronics.org/tutorials/arduino/ethernet-shield-web-server-tutorial/SD-card-web-server/>, 2013.
14. Myasishchev A.A. Web-server na Arduino MEGA i kontrollere seti w5500 s avtorizatsiey dostupa dlya udalennogo upravleniya. [Electronic resource]. - Mode of access: <https://sites.google.com/site/webstm32/web-server-avtorizaciej>, 2018.
15. Anisimov V.V. Kriptograficheskie metodyi zaschityi informatsii. [Electronic resource]. - Mode of access: <https://sites.google.com/site/anisimovkhv/learning/kripto/lecture>, 2012.
16. Myasishchev A.A. Avtorizatsiya na web-servere Arduino s pomoschyu GET i POST zaprosov. Materials of the XIII International scientific and practical Conference Modern scientific potential - 2018 , February 28 - March 7 , 2018 Construction and architecture. Mathematics. Modern information technology. Technical science. Physics. : Sheffield. Science and education LTD s. 36-42
17. Myasishchev A.A. WEB-server na ARDUINO dlya graficheskogo predstavleniya dannyh s datchikov cherez internet. [Electronic resource]. - Mode of access: https://www.researchgate.net/publication/332257299_WEB-server_na_ARDUINO_dla_graficeskogo_predstavlenia_dannyh_s_datcikov_cherez_internet, 2019.
18. HTTP/TCP with an atmega88 microcontroller (AVR web server). Guido Socher. [Electronic resource]. - Mode of access: <http://www.tuxgraphics.org/electronics/200611/article06111.shtml>, 2006
19. Kitsum. Avtorizatsiya na Web servere mikrokontrollera. [Electronic resource]. - Mode of access: <https://it4it.club/topic/13-avtorizatsiya-na-web-servere-mikrokontrollera/>, 2015

**д.т.н., проф. Мясішев О.А., д.т.н., с.н.с. Комарова Л.О., Грицький Р.В., Кулік К.В.
WEB-SERVER НА ARDUINO З АВТОРИЗАЦІЄЮ І ГРАФІЧНИМ ПРЕДСТАВЛЕННЯМ
ІНФОРМАЦІЇ З ДАТЧИКІВ**

У роботі на базі Arduino Mega і контролера W5100 побудований web-сервер для графічного відображення даних віддаленого клієнта, отриманих з датчиків температури, тиску, вологості. Програма сервера написана в середовищі розробки Arduino IDE. Відкоригована бібліотека Ethernet для W5100, яка для Arduino IDE ver. 1.0.3, 1.0.5-r2 приводила до зависання сервера. Показана можливість використання бібліотеки Dugraphs для графічної візуалізації даних, отриманих за допомогою датчиків. Написані скрипти для промальовування графіків для спроектованого web-сервера. Проаналізовано швидкість передачі даних з web-сервера Arduino для різних мережевих контролерів ENC28J60, W5100, W5500 для середовища програмування Arduino IDE і бібліотек UIPEthernet, Ethernet, Ethernet2. Показано, що з найменшою швидкістю дані передаються web-сервером з контролером ENC28J60 - 3.3КБайт/с, з найбільшою контролером W5500 - 23.4Кбайт / с. Відзначено, що ці сервера не підтримують багатопотокову роботу. Тому вони не можуть бути використані для створення мініатюрних універсальних web-серверів для обробки декількох запитів одночасно. Розглянутий в роботі сервер може обслуговувати тільки один запит від одного віддаленого клієнта. Проаналізовано швидкісні параметри передачі даних для сервера на мікроконтролері

ATmega328p (Arduino UNO) з контролером мережі ENC28J60. Програма сервера реалізована на мові Сі в середовищі програмування AVR Studio. Відзначено високу швидкість передачі даних - 140КБайт/с і можливість багатопотокового режиму роботи. Встановлено, що при одночасній передачі трьох файлів різним клієнтам сумарна швидкість передачі досягає 120-130КБайт / с, а для кожного клієнта 40-50КБайт/с. Показано, що використання такого сервера для вирішення завдання графічного представлення даних з датчиків ускладнено внаслідок складності перенесення програмного забезпечення на інші мікроконтролери і обмеженістю бібліотеки для роботи з картою microSD. Експлуатація розробленого тут сервера в перебігу трьох років показала високу надійність його роботи.

В роботі розглянуто створення web-сервера на Arduino, який має модернізовану HTTP basic authentication. Модернізація полягає в тому, що для авторизації використовується пароль зі списку паролів, який вибирається користувачем на підставі ключа, який пересилається сервером. При кожному новому вході на сервер попередній пароль стає недейсним. Представлений практичний приклад web-сервера на Arduino Mega, контролерами Ethernet: enc28j60 і w5500.

Ключові слова: мікроконтролер, ENC28J60, W5100, W5500, Arduino Mega, Arduino UNO, ATmega2560, бібліотека Dygraphs, AVR Studio, base64-encoded, basic authentication.

**Prof. Myasishev A.A., Prof Komarova L.O., Gritsky R.V., Kulik K.V.
WEB SERVER ON ARDUINO WITH AUTHORIZATION AND GRAPHIC REPRESENTATION OF INFORMATION FROM SENSORS**

In the work on the basis of Arduino Mega and the W5100 controller, a web server is built for graphical display of remote client data obtained from temperature, pressure, humidity sensors. The server program is written in the Arduino IDE development environment. The Ethernet library for the W5100 has been updated, which for Arduino IDE ver. 1.0.3, 1.0.5-r2 caused the server to hang. The possibility of using the Dygraphs library for graphical visualization of data obtained with sensors is shown. Written scripts for drawing graphics for the projected web server. The data transfer rate from the Arduino web server for various network controllers ENC28J60, W5100, W5500 for the Arduino IDE programming environment and UIPEthernet, Ethernet, Ethernet2 libraries was analyzed. It is shown that with the lowest speed the data is transmitted by the web server with the ENC28J60 controller - 3.3Kb/s, with the largest controller W5500 - 23.4Kbytes/s. It is noted that these servers do not support multithreaded work. Therefore, they can not be used to create miniature universal web servers for processing multiple requests simultaneously. The server considered in the work can serve only one request from one remote client. The speed parameters of data transfer for the server on the ATmega328p microcontroller (Arduino UNO) with the network controller ENC28J60 are analyzed. The server program is implemented in C language in the AVR Studio programming environment. A high data transfer rate of 140Kb/s and a possibility of multithreading are noted. It is established that with simultaneous transmission of three files to different clients, the total transmission speed reaches 120-130Kb/s, and for each client 40-50Kb/s. It is shown that the use of such a server to solve the problem of graphical representation of data from sensors is difficult due to the complexity of transferring software to other microcontrollers and the limited library for working with a microSD card. The operation of the server developed here for three years has shown high reliability of its operation.

The paper discusses the creation of a web server on the Arduino, which is a modernized HTTP basic authentication. The upgrade consists in the fact that for authorization a password is used from the list of passwords that is selected by the user on the basis of the key sent by the server. With each new login to the server, the previous password becomes invalid. Presents a practical example of a web server on Arduino Mega, Ethernet controllers: enc28j60 and w5500.

Keywords: microcontroller, ENC28J60, W5100, W5500, Arduino Mega, Arduino UNO, ATmega2560, Dygraphs library, AVR Studio, base64-encoded, basic authentication.