

УДК 004.451.83

doi:10.20998/2413-4295.2017.32.10

## МЕТОДОЛОГІЯ ПРОЕКТУВАННЯ ARDUINO В ЯКОСТІ WEB-CLIENT ТА WEB-SERVER З ВИКОРИСТАННЯМ ДАТЧИКА DHT11 ТА ЇХ ПОРІВНЯЛЬНА ХАРАКТЕРИСТИКА

Д. В. ЛУБКО, Ю. О. ЛИТВИН\*

Кафедра комп'ютерних наук, Таврійський державний агротехнологічний університет, м. Мелітополь, УКРАЇНА  
email: litwin.yu@yandex.ua

**АНОТАЦІЯ** У статті проаналізовано можливість використання мікроконтролеру Arduino в двох режимах: веб-сервер та веб-клієнт, які слугують основою для розробки автономних інтерактивних приладів або може працювати під управлінням програмного забезпечення, встановленого на з'єднаному з ним комп'ютері. Розроблений програмний засіб має такі комплектуючі, як: Arduino Uno, Ethernet shield, DHT11, резистор та проводи типу «мама-папа», «папа-папа». Представлений код програми дає змогу порівняти два режими роботи Arduino та виявити їх певні переваги та недоліки в плані передачі, зберігання та відображення певних даних для користувача.

**Ключові слова:** Arduino; Ethernet shield; DHT11; web-server; web-client; датчик.

## METHODOLOGY OF ARDUINO DESIGNING AS A WEB-CLIENT AND WEB-SERVER WITH THE USE OF THE DHT11 SENSOR AND THEIR COMPARATIVE CHARACTERISTICS

D. LUBKO<sup>1</sup>, Y. LYTVYN<sup>2</sup>

Department of Computer Science, Tavria State Agrotechnological University, t. Melitopol, UKRAINE

**ABSTRACT** Currently, there is a significant increase in interest in Smart House technology. The issue of availability and ease of monitoring of the management of a smart house is of crucial importance. An owner or a user of a smart house should have access to all systems, be able to manage them remotely and receive information about the status of each of them on a 24-hour basis. Therefore, the question of implementation of monitoring of the received data from sensors arises. This paper discusses the choice between using Arduino as a server or a client. The purpose of this work is to provide a program for processing temperature and humidity data using Arduino as a web server and a web client and conducting their comparative characteristics in terms of speed and quality of transmission and data display. In the context of this work, the following functions should be performed: processing the received temperature data, humidity of the room from the sensor DHT 11 and displaying data in the browser. As a result of the work, the basic principles of Arduino's work as a web server and web client are analyzed. The "Server" class creates a server that can transfer data and receive data from connected clients (users). The "Client" class creates clients that can connect to servers and exchange data that connects to the server by the IP address and port specified in the parameters. In the aspect of monitoring, according to the results obtained on the web page, we can conclude that using Arduino as a web client is more appropriate because it allows the developer to create data display in a more user-friendly form for the user. In the aspect of data storage it is better to use Arduino as a web client, since the resulting data will be stored in the developed database, in our case phpMyAdmin. All this allows monitoring of the received data from any "Smart House" equipment.

**Keywords:** Arduino; Ethernet shield; DHT11; web-server; web-client; sensor.

### Вступ

В даний час спостерігається значний ріст інтересу до технології «Розумний дім». Термін «розумний будинок» (або «інтелектуальний будинок») використовується для позначення сучасних будинків і будівель, в яких інженерні, інформаційні системи і системи безпеки об'єднані в єдину і організовану комплексну інтелектуальну систему. Дана інтелектуальна система покликана забезпечити більшу безпеку, а також найкращий комфорт мешканцям будинку. Як правило, основна причина установки систем розумного будинку полягає в підвищенні домашнього комфорту шляхом автоматизації рутинних завдань, таких як керування

освітленням, клімат-контролем, системами мультимедіа і т.д. Все гостріше постає питання доступності і легкості моніторингу та управління розумним будинком. Власник, або користувач, розумного будинку повинен мати доступ до всіх систем, мати змогу керувати ними на відстані та отримувати інформацію про стан кожної з них цілодобово. Тому постає питання - як саме реалізувати моніторинг отриманих даних з датчиків. В рамках цієї роботи ми розглянемо питання вибору між використанням Arduino (Ардуіно) в якості серверу чи клієнту.

Сьогодні на Україні, а також у світі є тенденція значного зростання інтересу до технологій, пов'язаних

з мікроконтролерами Arduino. Наприклад, задачі управління пристроями за допомогою певних команд.

Останнім часом з'явилася можливість підключати різноманітні датчики, робототехніку та керувати ними за допомогою веб-сайтів. Мікроконтролер Arduino має можливість виступати в якості веб-клієнта чи веб-сервера.

### Мета роботи

Мета даної роботи полягає в представленні програми для обробки даних температури та вологості, використовуючи Arduino в якості веб-сервера та веб-клієнту, а також проведення їх порівняльної характеристики в плані швидкості, якості передачі та відображення даних.

В рамках цієї роботи повинні виконуватися такі функції: обробка отриманих даних температури та вологості приміщення з датчика DHT11; відображення потрібних користувачу даних у браузері.

### Основний матеріал

Для виконання даної роботи ми використали датчик температури та вологості DHT11. Принцип роботи датчику температури заснований на вимірюванні опору при вимірюванні температури, що приводить до вимірювання вихідної напруги, яка слугує для вимірювання температури. Датчик вологості заснований на зміні концентрації електроліту, що покриває собою будь-електроізоляційний матеріал.

*Розглянемо докладно та покроково методологію проектування Ардуіно в якості веб-сервера.*

#### 1 крок.

Для того, щоб усі комплектуючі могли коректно працювати, треба спочатку імпортувати певні бібліотеки, а саме такі: SPI.h, Ethernet.h, Wire.h та DHT.h.

#### 2 крок.

Ініціюємо константи та номери контактів датчику температури та вологості. Введемо наступний код:

```
#define DHTPIN 2
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);
```

#### 3 крок.

Введемо mac-address контролера. Код:  
byte mac[] = {0x10, 0xBF, 0x48, 0xDD, 0xB5, 0xDA};

#### 4 крок.

Введемо IP-адрес Ethernet shield.

*Примітка:* Обов'язково треба перевірити, щоб IP-адрес Ethernet shield не співпадав з ip адресою іншого пристрою в мережі. При необхідності треба змінити останній байт. Код:

```
byte ip[] = {192, 168, 1, 100};
byte subnet[] = {255, 255, 255, 0};
```

```
EthernetServer server(80);
EthernetClient client;
String readString;
```

Для того, щоб написати цикл програми, який буде приймати вхідних веб-клієнтів та відображати http сторінку з даними, які будуть відновлюватися кожні 5 секунд без оновлення веб-сторінки браузера, треба використати в коді програми такі функції, як:

- available () - отримує посилення на підключеного до сервера клієнта для обміну даними. Возвращаемое значення: посилення на підключений до сервера клієнт (EthernetClient client = server.available ());

- connected () - визначає, підключений клієнт чи ні. Клієнт вважається підключеним, якщо з'єднання закрито, але є непрочитані дані;

- println () - відправляє дані на сервер з підключеного клієнта. Аналогічна функції print () з додаванням в відправку символу переходу на новий рядок;

- read () - отримує наступний (після останнього, отриманого командою read ()) байт від сервера або -1, якщо даних більше немає;

- client.println ("Content-Type: text/html"); - відображення веб-сторінки;

- client.println ("<metahttp-equiv=\"refresh\" content=\"5\">"); - відновлення веб-сторінки кожні 5 секунд;

- write () - відправляє дані всім клієнтам, підключеним до сервера;

- print () - відправляє дані всім клієнтам, підключеним до сервера. При цьому число відправляється як послідовність цифр даного числа.

- Ethernet.begin () - ініціалізує бібліотеку і мережеві настройки.

Синтаксис функції: Ethernet.begin (mac); Ethernet.begin (mac, ip); Ethernet.begin (mac, ip, gateway); Ethernet.begin (mac, ip, gateway, subnet).

*Параметри:* mac - mac-адресу (Media Access Control) пристрою (масив з 6 байтів). Це апаратний адрес адаптера вашого пристрою. Нові версії Arduino Ethernet shield мають наклейку з MAC-адресою пристрою. Для старих пристроїв mac потрібно знайти самостійно.

ip – це IP-адреса пристрою (масив з 4 байтів); gateway - IP-адреса шлюзу (масив з 4 байтів).

Опціонально - за замовчуванням IP-адрес пристрою з останнього октету встановлений в значення 1; subnet - маска підмережі мережі (масив з 4 байтів). Опціонально, за замовчуванням, такий - 255.255.255.0;

#### 5 крок.

Отримання даних з датчику DHT11 розрахунку та відображення результатів на екрані:

```
client.print ("Temperature (C): ");
client.println ((float) DHT11.temperature, 1);
client.println ("<br />");
client.print ("Temperature (F): ");
```

```

client.println(Fahrenheit(DHT11.temperature), 1);
client.println("<br />");
client.print("Humidity (%): ");
client.println((float)DHT11.humidity, 0);
client.println("<br />");
client.print("Temperature (K): ");
client.println(Kelvin(DHT11.temperature), 1);
client.println("<br />");
client.print("DewPoint (C): ");
client.println(dewPoint(DHT11.temperature, DHT11.humidity));
client.println("<br />");
client.print("DewPointFast (C): ");
client.println(dewPointFast(DHT11.temperature, DHT11.humidity));
client.println("<br />");

```

**6 крок.**

Закінчення роботи циклу обробки даних. Код:

```

client.println("</html>");
break;
if (c == '\n')
{ currentLineIsBlank = true;}
else if (c != '\r')

```

**7 крок.**

Отримання даних в поточному рядку. Код:

```
currentLineIsBlank = false;
```

**8 крок.**

Надання в браузер даних з Arduino. Код:

```

delay(1);
client.stop();
Serial.println("clientdisconnected");

```

**9 крок.**

Перерахуємо данні з градусів Цельсія до Фаренгейта. Код:

```

doubleFahrenheit(doublecelsius)
{ return 1.8 * celsius + 32; }

```

**10 крок.**

Перерахуємо данні з Цельсія до Кельвіна:

```

doubleKelvin(doublecelsius)
{ returncelsius + 273.15;}

```

**2 крок.**

Вводимо мак-адресу та IP-address серверу.

**3 крок.**

Використовуємо функцію delay (), яка зупиняє виконання програми на задане в параметрах значення - кількість мілісекунд (1000 мілісекунд в 1 секунд).

**4 крок.**

Для того, щоб використати Ардуіно в якості веб-клієнта потрібно ще використати такі функції:

- client () - створює клієнта, який підключається до сервера по вказаному в параметрах IP-адресою і порту. Client (ip, port), де ip - IP-адреса підключення клієнта (масив з 4 байтів); port - порт підключення клієнта.

- EthernetClient () - створює клієнта, який підключається до сервера. параметрів підключення (IP-адреса і порт) визначаються в функції client.connect ();

- write () - відправляє на сервер дані з підключеного клієнта;

- print () - відправляє дані на сервер з підключеного клієнта. При цьому число відправляється як послідовність цифр даного числа;

- println () - відправляє дані на сервер з підключеного клієнта. аналогічна функції print () з додаванням в відправку символу переходу на новий рядок;

- available () - повертає число байтів, доступних для читання (тобто обсяг даних, який був відправлений сервером для клієнта);

- read () - отримує наступний (після останнього, отриманого командою read ()) байт від сервера або -1, якщо даних більше немає;

- flush () - скидає отримані від сервера, але ще не прочитані дані;

- stop () - відключає клієнта від сервера.



Рис. 1 – Данні температури та вологості

**5 крок.**

Проектуємо в програмі цикл, який дозволяє отримати данні з датчику вологості та температури.

Код:

```

floathumidity = dht.readHumidity();
floattemperature = dht.readTemperature();
float f = dht.readTemperature(true);
Serial.print("Temperature: ");
Serial.println(temperature);
Serial.print("Humidity: ");
Serial.println(humidity);

```

**1 крок.**

Для того, щоб використати Ардуіно в якості веб-клієнта, ми використали бібліотеки SPI.h, Ethernet.h, Wire.h, DHT.h та ініціалізували датчик температури та вологості.

```
if (isnan(humidity) || isnan(temperature)
|| isnan(f)) {
Serial.println("Failed to read from DHT
sensor");
return; }
sendTemperature(temperature);
sendHumidity(humidity);
delay(3000);
```

Об'являємо змінну та вносимо данні температури та вологості. Код:  

```
$sql_insert = "insert into
sensor_temp_hum (humedad, time) values
('$humedad', time())";
```

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Дополнительно	Действие
1	id	int(11)			Нет	Нет	AUTO_INCREMENT	Изменить Удалить Обзор уникальных значений Первичный
2	temperatura	int(3)			Нет	Нет		Изменить Удалить Обзор уникальных значений Первичный
3	humedad	int(2)			Нет	Нет		Изменить Удалить Обзор уникальных значений Первичный
4	time	datetime			Нет	Нет		Изменить Удалить Обзор уникальных значений Первичный

Рис. 2 – Структура таблиці «Sensor\_temp\_hum» в базі даних

**6 крок.**

**Примітка:** Головною відмінною веб-серверу від веб-клієнту являється те, що данні ми передаємо GET-запитом.

Нами був розроблений код для відправки даних за допомогою GET-запиту на сервер. Код:

```
Void sendTemperature(float temperature) {
if (client.connect("31.170.165.229 ",80)) {
Stringrequest = "GET
/write_temp.php?temp=";
request += String(temperature);
request += " HTTP/1.0";
client.println(request);
client.println("Host: 31.170.165.229 ");
client.println("Content-Type:
application/x-www-form-urlencoded");
client.print("Content-Length: ");
client.println(data.length());
client.println();
client.print(data); }
if (client.connected()){
client.stop(); } }
Void sendHumidity(float humidity){
if (client.connect("31.170.165.229",80)){
Stringrequest = "GET
/write_hum.php?humidity=";
request += String(humidity);
request += " HTTP/1.0";
client.println(request);
Serial.println("Данні надіслані");
client.println("Host: 31.170.165.229 ");
client.println("Content-Type:
application/x-www-form-urlencoded");
client.print("Content-Length: ");
client.println(data.length());
client.println();
client.print(data); }
if (client.connected()) {
client.stop(); } }
```

**7 крок.**

Далі данні з датчику DHT11 ми помістили в базу даних phpMyAdmin (рис. 2) за допомогою файлів write\_hum.php та write\_temp.php, які підключаються до бази даних.

**8 крок.**

**Примітка:** Попередньо ці файли були розміщенні в головній папці серверу.

**9 крок.**

Використавши мови програмування PHP, CSS, HTML та AJAX створюємо сторінку відображення даних температури та вологості на графіку (рис. 3).

**Обговорення результатів**

В результаті проектування коду на мікроконтролері Arduino, з датчику DHT11 за допомоги веб-запиту та файлів write\_hum.php та write\_temp.php дані передаються в базу даних phpMyAdmin. Після чого туди вносяться дані температури та вологості. Далі використовуємо ці дані для побудови та відображення на графіку, використавши мови програмування PHP, CSS, HTML та AJAX.

**Висновки**

У результаті виконаної роботи були проаналізовані основні принципи роботи Ардуіно в якості веб-сервера та веб-клієнта. Клас «Server» створює сервер, який може передавати данні та отримувати данні від підключених клієнтів (користувачів). Клас «Client» створює клієнтів, які можуть підключатися до серверів та обмінюватися даними, який підключається до сервера по вказаному в параметрах IP-адресою та портом.

В аспекті моніторингу, згідно отриманих результатів на веб-сторінці, можемо зробити висновок, що використовувати Ардуіно в якості веб-клієнту доцільніше, бо це дозволяє розробнику створити відображення даних у більш зручній формі для користувача. А саме - зробити графік зданими температури та вологості, отримувати данні про дату, час отриманих даних та виводити їх статистику, тощо, що буде влаштовувати розробника подібної метеостанції.

В аспекті збереження даних - краще використовувати Ардуіно в якості веб-клієнту, так як

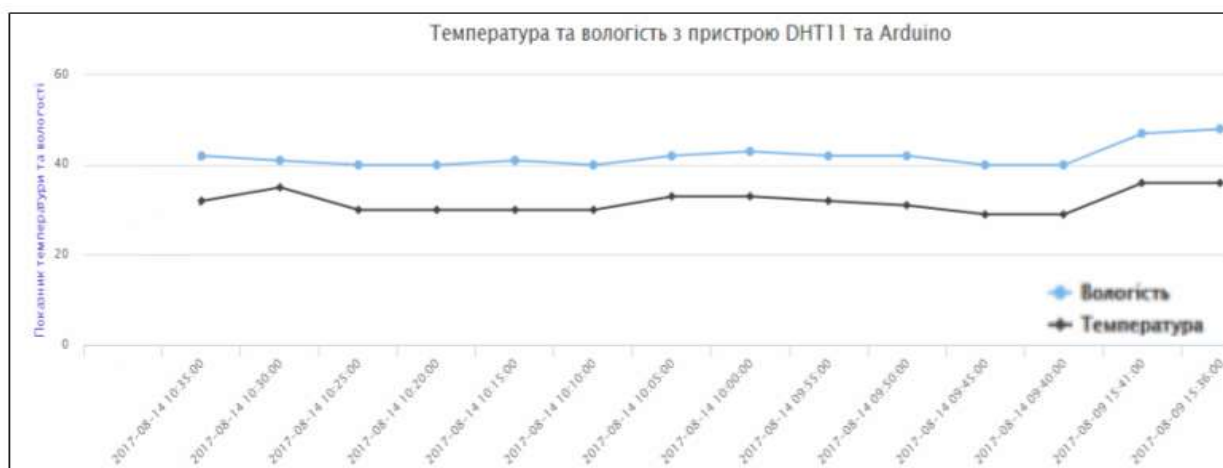


Рис. 3 – Графік відображення даних температури та вологості

отримані данні будуть зберігатися в розробленій базі даних, в нашому випадку у phpMyAdmin.

Все вищенаведене і дає змогу аналізувати та проводити моніторинг отриманих даних з будь-якого устаткування «Розумного будинку».

У перспективі ми розглянемо питання керування таким устаткуванням «Розумного будинку» як веб-камера (наприклад модель OV7670 300КР з VGA-модулем) через веб-сервер.

А це допоможе значно розширити можливості нашого проекту. Що в свою чергу дасть змогу підвищити ефективність проектування та роботи подібних систем, і в цілому, принесе значну користь, зручність та комфорт у будинки людей.

#### Список літератури

1. **Боос, Г. О.** Следящая система механизма подачи станка с ЧПУ с использованием "Arduino Uno" / **Г. О. Боос, А. С. Гордеев, Р. А. Просоедов** // *Технические науки - от теории к практике: сб. ст. по матер. XLV междунар. науч.-практ. конф. № 4(41)*. – Новосибирск: СибАК, 2015.
2. **Петин, В.** Проекты с использованием контроллера Arduino / **В. Петин**. Санкт-Петербург: БХВ-Петербург, 2014. – 401 с.
3. **Момот, М.** Мобильные роботы на базе Arduino / **М. Момот**. – Санкт-Петербург: БХВ-Петербург, 2017. – 288 с.
4. **Massimo, B.** Getting Started with Arduino, 2nd Edition / **B. Massimo**. - MakerMedia, 2011. - 130 P.
5. **Марголис, М.** Arduino Cookbook / **М. Марголис**. - O'ReillyMedia, 2011. - 648 P.
6. **Монк, С.** Arduino Projects for the Evil Genius / **С. Монк**. - McGraw-Hill Book Company, Inc., 2010. - 208 P.
7. **Монк, С.** Programming Arduino: Getting Started with Sketches / **С. Монк**. – СПб.: Питер, 2017. - 452 P.
8. **McRoberts, M.** Beginning Arduino / **М. McRoberts**. – Apress, 2010. - 460 P.
9. **Agus, K.** Arduino and Genuino MKR1000 Development Workshop (+code) / **К. Agus**. - PE Press, 2016. - 125 P.
10. **Oxer, J.** PracticalArduino: Cool Projects for Open Source Hardware. / **J. Oxer, H. Blemings**. – Apress, 2009. - 445 P.
11. **Jones, D.** Arduino: Simple and Effective Strate giesto Learn Arduino Programming / **D. Jones**. – Amazon Digital Services LLC, 2017. - 81 P.
12. **Leong, K. C.** Arduino Web-Control system for Broiler house / **K. C. Leong** // Submitted to the Faculty of Electrical Engineering in partial fulfillment of the requirement for the degree of Bachelor in Electrical Engineering. - *Faculty of Electrical Engineering. Universiti Teknologi Malaysia*, 2014. - 70 P.

#### Bibliography (transliterated)

1. **Boos, G. O.** Sledyaschaya sistema mehanizma podachi stanka s ChPU s ispolzovaniem "Arduino Uno". *Tehnicheskie nauki - ot teorii k praktike: sb. st. po mater. XLV mezhdunar. nauch.-prakt. konf. 4(41)*. – Novosibirsk: SibAK, 2015.
2. **Petin, V.** Proektyi s ispolzovaniem kontrollera Arduino. Sankt-Peterburg: BHV-Peterburg, 2014.
3. **Momot, M.** Mobilnyie roboty na baze Arduino. - Sankt-Peterburg: BHV-Peterburg, 2017.
4. **Massimo, B.** Getting Started with Arduino, 2nd Edition. - MakerMedia, 2011.
5. **Margolis, M.** Arduino Cookbook. - O'Reilly-Media, 2011.
6. **Monk, S.** Arduino Projects for the Evil Genius. - McGraw-Hill Book Company, Inc., 2010.
7. **Monk, S.** Programming Arduino: Getting Started with Sketches. – SPb.: Piter, 2017.
8. **McRoberts, M.** Beginning Arduino. – Apress, 2010.
9. **Agus, K.** Arduino and Genuino MKR1000 Development Workshop (+code). - PE Press, 2016.
10. **Oxer, J.** PracticalArduino: Cool Projects for Open Source Hardware. – Apress, 2009.
11. **Jones, D.** Arduino: Simple and Effective Strate giesto Learn Arduino Programming. – Amazon Digital Services LLC, 2017.
12. **Leong, K. C.** Arduino Web-Control system for Broiler house. Submitted to the Faculty of Electrical Engineering in partial fulfillment of the requirement for the degree of Bachelor in Electrical Engineering, *Faculty of Electrical Engineering. Universiti Teknologi Malaysia*, 2014, 70.

### Відомості про авторів (About authors)

**Лубко Дмитро Вікторович** – кандидат технічних наук, доцент, Таврійський державний агротехнологічний університет, доцент кафедри комп'ютерних наук; м. Мелітополь, Україна, e-mail: di75ma@mail.ru.

**Dmitro Lubko** - Candidate of Engineering Sciences, Associate professor, Tavria State Agrotechnological University, Associate professor of the Department of Computer Science; Melitopol, Ukraine, e-mail: di75ma@mail.ru.

**Литвин Юлія Олександрівна** - асистент, Таврійський державний агротехнологічний університет, асистент кафедри комп'ютерних наук; м. Мелітополь, Україна, e-mail: litwin.yu1994@gmail.com.

**Yulia Lytvyn** - assistant, Tavria State Agrotechnological University, assistant of the Department of Computer Science; Melitopol, Ukraine, e-mail: litwin.yu1994@gmail.com.

*Будь ласка, посилайтесь на цю статтю наступним чином:*

**Лубко, Д. В.** Методологія проектування Arduino в якості Web-client та Web-server з використанням датчика DHT11 та їх порівняльна характеристика / **Д. В. Лубко, Ю. О. Литвин** // *Вісник НТУ «ХПІ»*, Серія: *Нові рішення в сучасних технологіях*. – Харків: НТУ «ХПІ». – 2017. – № 32 (1254). – С. 62-67. – doi:10.20998/2413-4295.2017.32.10.

*Please cite this article as:*

**Lubko, D., Lytvyn, Y.** Methodology of Arduino designing as a Web-client and Web-server with the use of the DHT11 sensor and their comparative characteristics. *Bulletin of NTU "KhPI". Series: New solutions in modern technologies*. – Kharkiv: NTU «KhPI», 2017, **32** (1254). – С. 62-67. – doi:10.20998/2413-4295.2017.32.10.

*Пожалуйста, ссылайтесь на эту статью следующим образом:*

**Лубко, Д. В.** Методология проектирования Arduino в качестве Web-client и Web-server с использованием датчика DHT11 и их сравнительная характеристика / **Д. В. Лубко, Ю. А. Литвин** // *Вестник НТУ «ХПИ»*, Серія: *Новые решения в современных технологиях*. – Харьков: НТУ «ХПИ». – 2017. – № 32 (1254). – С. 62-67. – doi:10.20998/2413-4295.2017.32.10.

**АННОТАЦИЯ** В статье проанализирована возможность использования микроконтроллера Arduino в двух режимах: веб-сервер и веб-клиент, которые служат основой для разработки автономных интерактивных приборов либо может работать под управлением программного обеспечения, установленного на соединенном с ним компьютером. Разработанное программное средство имеет такие комплектующие, как: Arduino Uno, Ethernet shield, DHT11, резистор и провода типа «мама-папа», «папа-папа». Представленный код программы дает возможность сравнить два режима работы Ардуино и выявить их определенные преимущества и недостатки в плане передачи, хранения и отображения определенных данных для пользователя.

**Ключевые слова:** Arduino; Ethernet shield; DHT11; web-server; web-client; датчик.

*Поступила (received) 10.09.2017*