

issue №6, December 2018

брой №6, декември 2018г.

САХ

САХ ТЕХНОЛОГИИ

технологии

списание

technologies

journal

ISSN 1314-9628

CAx technologies

Computer Aided Technologies

- PLM
- CAD/CAM/CAE
- Intelligent Systems (incl. Web Intelligence)
- Modeling and Simulation
- E-Engineering with Internet Applications

- E-Learning
- Digital Humanities
- Business Management Systems
- CAx in Medicine
- Other



GA Calculator

Calculations :-)

```

$fitnesspro=$weight[$c1-1][$c2-1]+$wei
[$c3-1]+$weight[$c3-1][$c4-1]+$weight[
[$c5-1]+$weight[$c5-1][$c6-1]+$weight[
[$c7-1]+$weight[$c7-1][$c8-1]+$weight[
[$c9-1]+$weight[$c9-1][$c10-1]+$weight
else {$fitnesspro=1000000000;}

```

N	Name	Type	From	To	Period
1	c1	0	1	1	0
2	c2	0	2	10	0
3	c3	0	2	10	0
4	c4	0	2	10	0
5	c5	0	2	10	0
6	c6	0	2	10	0
7	c7	0	2	10	0
8	c8	0	2	10	0
9	c9	0	2	10	0
10	c10	0	2	10	0

Add Gene Input Gene[] Delete [X] Read Project Save Project

EDITORIAL BOARD

Honorary Chairman:

BOYANOV Kiril, Acad., DSc., *Bulgarian Academy of Sciences, Bulgaria*

Chairman:

TUDJAROV Boris, Prof., Dr., *Technical University of Sofia, Bulgaria*

Members:

ANDONOVA Anna, Assoc. Prof., Dr., *Technical University of Sofia, Bulgaria*

ANISIC Zoran, Prof., Dr., *University of Novi Sad, Serbia*

AOMURA Shigeru, Prof., Dr., *Tokyo Metropolitan University, Japan*

BUBENCHIKOV Aleksey, Prof., DSc., *Tomsk State University, Russia*

BOTZHEIM Janos, Assoc. Prof., Dr., *Tokyo Metropolitan University, Japan*

DURIDANOV Ludmil, Dr., *New Bulgarian University, Bulgaria*

GOTSEVA Daniela, Prof., Dr., *Technical University of Sofia, Bulgaria*

HRISTOV Valentin, Assoc. Prof., Dr., *Technical University of Sofia, Bulgaria*

IVANOV Stanislav, Assoc. Prof., Dr., *New Bulgarian University, Bulgaria*

KUBOTA Naoyuki, Prof., Dr., *Tokyo Metropolitan University, Japan*

LAZAROVA Milena, Prof., Dr., *Technical University of Sofia, Bulgaria*

LOO Chu Kiong, Prof., Dr., *University of Malaya, Malaysia*

MAHDY Hatam, Dr., *Future University, Sudan*

MITREV Rosen, Assoc. Prof., Dr., *Technical University of Sofia, Bulgaria*

MLADENOV Valeri, Prof., Dr., *Technical University of Sofia, Bulgaria*

NAKOV Ognyan, Prof., D.Sc., *Technical University of Sofia, Bulgaria*

NIKOLOVA Irena, Assoc. Prof., Dr., *Technical University of Sofia, Bulgaria*

PAPAKOSTAS Dimitrios, Prof., Dr., *Alexander Technological & Educational
Institute of Thessaloniki, Greece*

PENCHEV Vasil, Assoc. Prof., Dr., *Technical University of Sofia, Bulgaria*

PETKOVA Radka, Assoc. Prof., Dr., *Technical University of Sofia, Bulgaria*

SAVOV Simeon, Dr., *University of Mining and Geology "St. Ivan Rilski", Bulgaria*

SHTRAKOV Stanko, Assoc. Prof., Dr., *South-West University, Bulgaria*

TRIFONOV Roumen, Assoc. Prof., Dr., *Technical University of Sofia, Bulgaria*

TASHEV Tasho, Prof., Dr., *Technical University of Sofia, Bulgaria*

TATENO Toshitake, Assoc. Prof., Dr., *Meiji University, Japan*

YU Hui, Dr., *University of Portsmouth, United Kingdom*

The papers are reviewed by editorial board members and/or external reviewers.

Publisher: Technical University of Sofia, ISSN 1314-9628

Publisher Address: Bulgaria, Sofia 1000, Kliment Ohridski blvd. №8, Technical University of Sofia



Публикациите се рецензират от членове на редакцията колегия и/или външни рецензенти.

Издателство на Технически университет – София, ISSN 1314-9628
България, София 1000, бул. “Климент Охридски” №8, Технически университет – София



CONTENTS

1. Evolution of programming languages	5
<i>Daniela Gotseva, Yavor Tomov, Petko Danov</i>	
2. Mathematical modelling of a construction manipulator performing horizontal rectilinear motion	11
<i>Miroslava Ivanova</i>	
3. Digital Humanities in the Age of Visual Immediacy	27
<i>Ludmil Duridanov</i>	
4. Linearized mathematical model of a construction manipulator performing a horizontal rectilinear motion	33
<i>Rosen Mitrev, Miroslava Ivanova, Ivan Ivanov</i>	
5. Design of system for remote control of electrical appliances over the Internet	43
<i>Aleksandar Hristov, Rumen Trifonov</i>	
6. Experimental determination of the mechanical characteristics of laboratory jaw crusher with directly drive of the movable jaw	49
<i>Simeon Savov</i>	
7. Random generation of four-bar linkages under Grashof's conditions for the needs of evolutionary optimization algorithms	59
<i>Boris Tudjarov</i>	
8. Mathematical modelling of the dynamical behavior of a heavy manipulator with freely suspended payload	65
<i>Rosen Mitrev</i>	
9. Windows calculator of Markov chain characteristics	75
<i>Valentin Hristov, Rosen Mitrev, Boris Tudjarov</i>	
10. MySQL scripts evaluation with Moodle and Virtual Programming Lab	83
<i>Ivaylo Chenchev</i>	

Design of system for remote control of electrical appliances over the Internet

Aleksandar Hristov

Technical University of Sofia, Sofia, Bulgaria
E-mail: ahristov@tu-sofia.bg

Rumen Trifonov

Technical University of Sofia, Sofia, Bulgaria
E-mail: r_trifonov@tu-sofia.bg

Abstract. *The purpose of this paper is to present designing and prototyping a system for remote control of electrical appliances over the Internet using Arduino. Some technologies for remote control of electrical appliances are analyzed. Arduino is open source hardware which makes it suitable for designing a variety of devices, ranging from simple thermostat to aircrafts. Based on this analysis is determined that the client-server architecture for designing and prototyping an Arduino based system is the most suitable. Also, in this paper are proposed microcontroller program in C language and website for the system.*

Keywords: *Arduino, C Programming, PHP Programming, Website, MySQL*

Проектиране на система за дистанционно управление на електроуреди през Интернет

Александър Христов

Технически Университет - София, София, България
E-mail: ahristov@tu-sofia.bg

Румен Трифонов

Технически Университет - София, София, България
E-mail: r_trifonov@tu-sofia.bg

Резюме. *Целта на тази статия е да представи възможностите за използване на Arduino за прототипиране на система за дистанционно управление на електроуреди през Интернет. Разгледани са съвременните технологии за дистанционно управление на електрически уреди. Arduino е open source хардуер, което го прави подходящ за проектиране на различни устройства, вариращи от обикновен термостат до самолети. Въз основа на този анализ е избрана архитектура клиент-сървър за проектиране и прототипиране на система върху Arduino. Също така, в тази статия е дадена управляваща програма за микроконтролера на езика C и уебсайт за управление на системата.*

Ключови думи: *Arduino, Програмиране на C, Програмиране на PHP, уебсайт, MySQL*

1. Увод

В последните няколко години се появи и започна да придобива все по-масов характер направлението „Интернет на нещата“ (Internet of Things- IoT). При него основната идея е свързване на различни видове електроуреди (телефони, телевизори, хладилници и т.н.) към Интернет и осъществяване на комуникация помежду им [5]. Така се появяват т.нар. „умни“ уреди (smart devices).

Свързването на уредите към Интернет спомага за увеличаване на техните функционални възможности и за постигане на по-голямо удобство при работата с тях. Известно е, че към момента 99 % от устройствата във физическия свят не са свързани, което е сериозно техническо предизвикателство. Проблемът трябва да се разреши в следващите няколко години. Бъдещият стандарт 5G New Radio стъпвайки

на 2G/3G/4G ще увеличи пропускателната способност до над 10 Gb/s; ще увеличи броя устройства до 100 млн. на 1 км²; ще намали и гарантира времезакъснения в мрежата до 1 ms. Ясно е, че това ще доведе до революция във високоскоростното предаване на огромни обеми от данни с невероятно ниски времезакъснения за огромния брой свързани устройства, което ще доведе до по-нататъшно развитие на IoT и промишления интернет (IIoT). Очаква се към 2020 г Интернет да свързва 50 милиарда устройства. Тези петдесет милиарда устройства ще обменят трилиони гигабайта данни, за да работят заедно за подобряване на живота ни и нашия бизнес.

IoT свързва smart устройства, традиционни компютърни устройства и нетрадиционни устройства към Интернет. В рамките на IoT комуникацията е от вида машина-към-машина (Machine-to-Machine - M2M), която дава възможност за обмен на данни между компютърни устройства без човешка намеса. Съществуват множество решения за отдалечено управление на електроуреди.

Дистанционно управление през GSM мрежата

Използва SMS съобщения и комуникатор. Налице са решения, които вместо SMS използват предаване на съобщения през Интернет като например Facebook [9], Telegram Messenger и други. По-долу са дадени някои недостатъци на тези подходи:

- Управление от ограничен брой клиентски устройства (GSM номера);
- Ограничен брой независими релейни изходи и/или нормално отворени или нормално затворени контакти;
- Труден монтаж и конфигуриране на SMS комуникатора;
- Ограничено покритие на голяма част от GSM операторите;
- Висока инсталационна и оперативна цена.

Дистанционно управление през Интернет

Основно може да бъдат разделени в две направления – управление чрез веб-приложения (веб-интерфейс) и на базата на клиент-сървърни разпределени приложения. При управлението посредством веб-приложения отдалеченият потребител се свързва със системата чрез веб-браузър. Интерфейсът за управление се изгражда в прозореца на браузъра. По този начин системата става платформено независима – потребителят може да използва всякакъв браузър, инсталиран на всякаква

операционна система. Подходът е универсален, въпреки че в известен смисъл се ограничават функционалните възможности на отдалеченото управление (управлението е от браузъра, т.е. обработката е по-бавна, тъй като кодът се интерпретира вместо да се използва компилиран такъв).

При подхода, базиран на инсталирано приложение, отдалеченият потребител използва специална клиентска програма за достъп до системата. За конкретната система за управление на електроуреди не е целесъобразно разработването на клиентска програма, която трябва да бъде предварително инсталирана на потребителския компютър. Подходът е платформено зависим – за всяка операционна система трябва да бъде разработвана отделна програма.

Протоколът MQTT

Друг подход е вместо традиционната клиент-сървър технология да се използва Message Queue Telemetry Transport – MQTT. Този протокол е от типа publish/subscribe за комуникация между машини, изграден върху TCP/IP в реално време. Протоколът използва MQTT брокер, който разпределя съобщенията между MQTT клиентите. Съществува от 1999г., а като стандарт - от 2013г. Съществуват множество Open source библиотеки на много програмни езици. Например, Mosquitto е пример за MQTT брокер с отворен код, написан на програмния език C. Наличен е за всички популярни GNU/Linux дистрибуции: Debian, Ubuntu, Fedora, RedHat Enterprise Linux, openSUSE, CentOS, ArchLinux, Slackware и също за Windows, FreeBSD и Mac. Paho предоставя имплементации на MQTT клиенти на C/C++, Java, JavaScript, Python и C#. Съществува MQTT библиотека и за Arduino [4].

MQTT клиентите са софтуерни приложения, които се свързват към MQTT брокер. Клиентите могат да изпращат и да получават съобщения. MQTT брокерът е софтуер, който се изпълнява като демон и се грижи за разпределянето на съобщенията между клиентите. Сигурността на комуникациите се осигурява чрез криптиране на предаваните данни с TLS/SSL, автентикация с потребителско име и парола и ауторизация с Access Control Lists. При този подход за решаване на конкретната инженерна задача недостатък е наличието на много клиенти от тип subscriber и по-трудното му

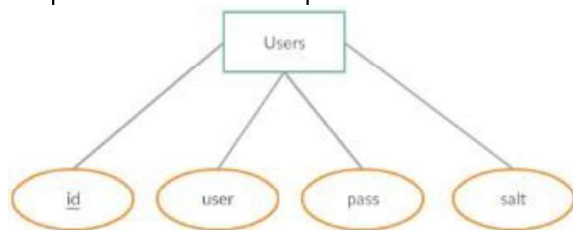
имплементиране върху ограничения като хардуер Arduino контролер. Целта на настоящата работа е да се разработи система за дистанционно управление на електроуреди през Интернет.

2. СИСТЕМА ЗА ДИСТАНЦИОННО УПРАВЛЕНИЕ

На базата на така направения анализ по-горе при разработването на системата за дистанционно управление на електроуреди е избрана архитектура клиент-сървър с уеб интерфейс и MySQL база данни, тъй като по-добре отговаря на нуждите на системата. Клиентската част от системата е независима от операционната система и брауъра на потребителя, което позволява отдалеченото управление да се извършва по всяко време от всяка точка и от всяко устройство. Разработваната система се нуждае от данни за автентикация на потребители, които трябва да бъдат добре защитени от достъп на хакери.

2.1. База данни за ауторизация

За ауторизацията на клиентите в разработваната системата за дистанционно управление на множество електроуреди е предпочетено да се използва база данни предвид възможностите за сигурност, бъдещо развитие и по-точно възможността за внедряване на системата при множество клиенти. На фиг.1. е показана ER диаграмата на базата данни, използвана за управление достъпа до системата чрез автентикация с потребителско име и парола.



Фиг. 1 ER диаграмата на базата данни

Създава се база данни с име loginDB. Таблицата users има първичен ключ id и полета за потребителско име, парола и salt.

```

CREATE TABLE users (
  id INT NOT NULL PRIMARY KEY,
  user VARCHAR(255) NOT NULL UNIQUE,
  pass BINARY(32) NOT NULL,
  salt BINARY(32) NOT NULL
);
  
```

Овкусяването със salt се използва за по-голяма сигурност. Ако в таблицата бъдат

записани двама потребителя, които имат една и съща парола, то след добавянето на salt резултатите в колоната pass ще са различни: За целите на разработваната система уеб сървърът има нужда от права само за SELECT заявки, тъй като се прави проверка дали даден потребител съществува и дали паролата му съвпада с въведената (няма нужда да променя данните). Евентуалната промяна на данните от хакер се ограничава като се дават права за изпълнение единствено и само на SELECT заявки върху таблицата users от базата данни loginDB. Например, за да се зададе достъп само от уеб сървъра се задават потребителско име loginform и парола 'p@s5w0rd', а хостът е localhost, тъй като MySQL сървърът и уеб сървърът са на един и същи компютър:

```
GRANT SELECT ON loginDB.users
```

```
TO loginform@localhost
```

```
IDENTIFIED BY 'p@s5w0rd';
```

Накрая таблицата users се попълва с данните на потребителите. Колоната user се попълва с потребителското име в четима форма, тъй като само чрез потребителско име е невъзможно да се премине страницата за автентикация и управление на достъпа до системата. За попълване на колоната salt се използва произволно число, което се изтегля чрез RAND(). След това се хешира чрез алгоритъма SHA2. Използва се функцията SHA2(string, hash length), чиито аргументи са стрингът за хеширане и дължината на резултатът в битове. Колкото по-голяма е дължината, толкова по-трудно е да се намери първоначалният текст. Накрая се използва функцията UNHEX(), която интерпретира всяка двойка шестнадесетични цифри в аргумента като число, преобразува я в знак и връща двоичен стринг. За попълване на колоната pass се използва паролата, към която се конкатенира salt. След това резултатът преминава през хеширане и превръщане в двоичен формат (за да се подобри сигурността). Така полученият двоичен стринг се записва в полето pass.

2.2. Разработване на web страница

Началната страница на системата за дистанционно управление съдържа една форма, в която има две полета: за потребителско име и за парола, както и бутон за влизане в системата. Когато потребителят попълни полетата на формата и кликне на бутона Submit, данните от формата се изпращат за обработка към PHP скрипт (файл

с име „login.php“). чрез HTTP POST метода. В страницата за обработка на въведените от потребителя данни първо се прави проверка дали са въведени стойности и дали въведените данни са от текстов тип. В случай, че въведените данни са некоректни, потребителят се връща в началната страница. Прави се проверка на потребителското име, за да се избегне възможността от SQL Injection атаки. Въведеното потребителско име трябва да е с дължина от 4 до 64 символа и трябва да съдържа само букви, цифри или символът underscore. За проверката се използва функцията preg_match(), на която като параметри се подават шаблон и променливата за проверка. За връзка с базата данни, където се съхраняват потребителските имена и паролите на всички потребители се използва функцията mysqli_connect(). Подадените параметри са адрес на хоста, потребителско име за MySQL базата данни, парола за базата данни и име на базата данни. При настъпване на грешка и неуспешна връзка с базата данни на екрана се извежда подходящо съобщение и работата на потребителя се прекратява.

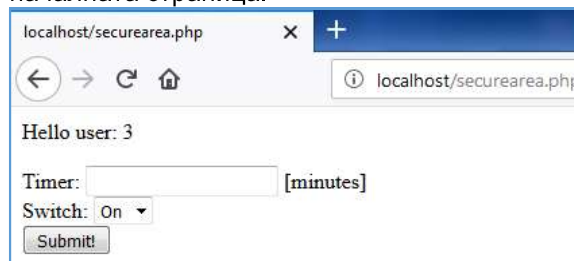
Заявката към базата данни е SELECT, тъй като данните трябва само да бъдат прочетени, след което сравнени за съпадение. Създава се променлива, в която текстът на заявката да бъде съхраняван. Заявката извлича id на потребителя в случай, че въведените потребителско име и парола съвпадат с определен ред от таблицата users.

Заявката към базата данни се изпълнява чрез функцията mysqli_query(), като подадените параметри са връзка с базата данни и текстът на заявката. Резултатът се записва в променлива. Ако променливата е празна, то при изпълнението на заявката е настъпила грешка и на екрана се извежда подходящо съобщение и работата на потребителя се прекратява.

Създава се променлива, в която се съхранява върнатия от базата данни ред. Извличането на реда става чрез функцията mysqli_fetch_assoc(). Ако полето id е на променливата row е празно, то в базата данни не се съдържа ред с подадените от потребителя потребителско име и парола, т.е. са въведени некоректни данни. Тогава на екрана се извежда подходящо съобщение и работата на потребителя се прекратява.

Ако всички проверки за валидност на въведените данни и за наличието им в базата данни са успешни, то потребителят е с

потвърдена автентичност и му се предоставя достъп до страницата securearea.php, чрез която се управлява електроуредът. За да е сигурно, че до securearea.php достъп имат само авторизирани потребители се използва т. нар. сесиен подход. Създава се нова сесия чрез функцията session_start(). Текущия сесиен идентификатор се заменя с нов, запазвайки текущата информация в сесията чрез функцията session_regenerate_id(). Чрез страницата securearea.php потребителят управлява собствения си Arduino контролер, с който се управлява и електроуредът. За целта securearea.php разполага с форма, която се състои от текстово поле за въвеждане на продължителност на работата на електроуредът, контрола select за включване и изключване и бутон за записване на данните, въведени от потребителя. На фиг. 2 е показан работен екран на страницата securearea.php. При натискането на бутона Submit (фиг. 2) данните се изпращат към saveToFile.php чрез HTTP POST метода. С цел сигурност в страницата saveToFile.php отново се прави проверка дали потребителят е авторизиран. В противен случай потребителят се връща в началната страница.



Фиг. 2 Работен екран на securearea.php

Ако потребителят е авторизиран данните се валидират (невалидните данни включват празни полета или нечислови данни). При въведени невалидни данни потребителят се връща в страницата securearea.php за повторно въвеждане.

При авторизиран потребител и коректни данни, стойностите на последните се записват в променливи. След това се отваря файлов поток чрез функцията fopen(). Подадените параметри на fopen() са име на файла и mode, който определя типа достъп до файла. Името на файла е съставено от текст fileArduino, към който се конкатенира идентификационния номер от реда в базата данни на съответния потребител. Накрая към името се конкатенира и разширението на файла (.html). За mode е избран „w“, тъй като

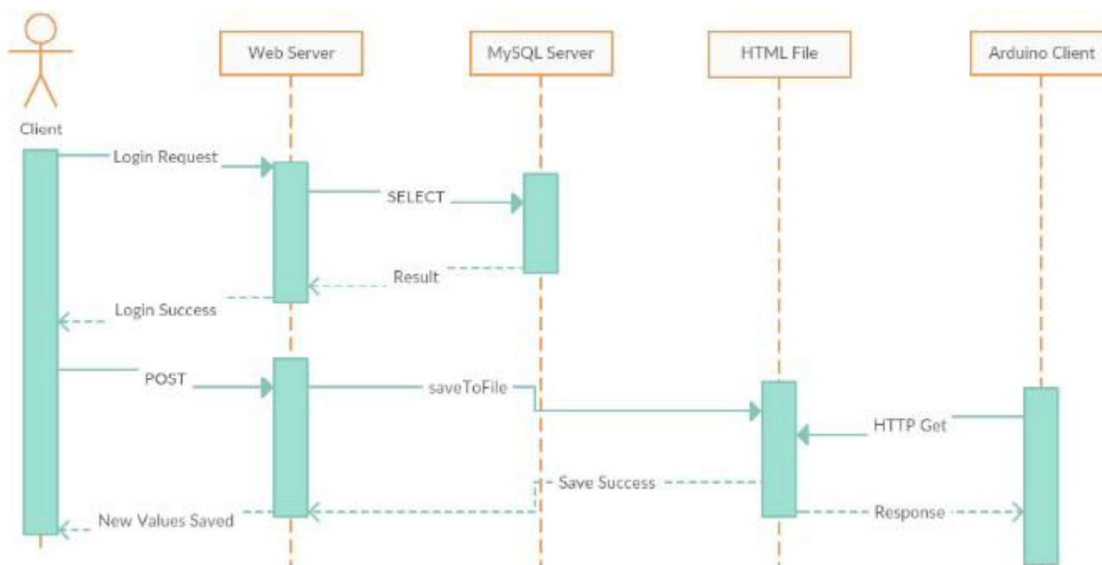
се използва само за запис на данни, а при наличие на стари данни във файла те биват изтривани. Всеки Arduino контролер чете данните от файла на съответния потребител. За запис на съдържанието на потребителските данни във файловия поток се използва функцията `fwrite()`. Подаваните параметри са файловият поток и променливата, която трябва да бъде записана. Данните, които са въведени от потребителя (Вж. по-горе страницата `securearea.php`) се записват във файла. Данните са продължителност на работа в минути и код на команда за включване/изключване.

Форматът на примерна команда е:

XXXXY, където
XXXX е числов string определящ продължителността на работа;
Y е код на командата.

Както е видно от формата на командата към XXXX е конкатениран код Y. Стойността на Y е 1 при команда за изключване (OFF) и 2 при команда за включване (ON). Стойностите от 3 до 9 са запазени за бъдещо развитие на системата.

На фиг. 3. е показана Sequence диаграмата на разработената система.



Фиг. 3 Sequence диаграма

Последователността на работа на системата е следната:

- Потребителят стартира уеб браузър и въвежда своите потребителско име и парола. Той се свързва чрез Интернет със сървъра, който осъществява процедурата по автентикация;
- Уеб сървърът подготвя и изпраща SELECT заявка, за да получи необходимите за автентикация потребителско име и парола, записани в базата данни;
- Уеб сървърът получава потребителско име, парола и идентификационен номер от базата данни и ги сравнява с въведените от потребителя;
- При успех (ауторизация) разрешава на клиента достъп до страницата за въвеждане на команди;
- Потребителят изпраща команда към контролера чрез въвеждане на продължителността на работа на електроуред и режим (включване или изключване);
- Сървърът получава командата и я записва във файл след като я валидира и верифицира.
- Контролерът за управление на електроуред изпраща HTTP GET заявка, за да прочете този файл;
- Контролерът за управление получава от уеб сървъра отговор и изпълнява съответната команда.

2.3. Прототипиране върху Arduino на системата за дистанционно управление

За реализирането на контролера са избрани: микроконтролерна платка Arduino Uno [6],

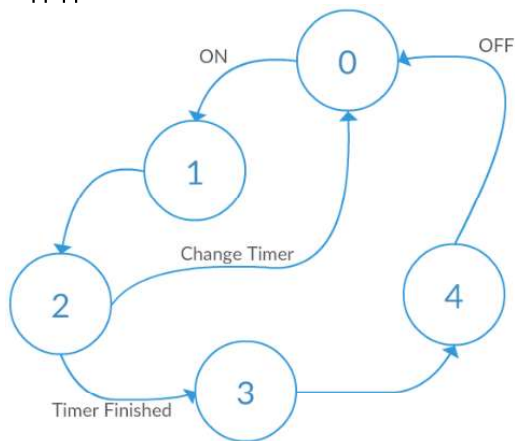
мрежовия интерфейс Arduino Ethernet Shield W5500 [5] и разширителна платка с релета Arduino 4 Relays Shield [7].

Реализирането на инициализиращата и управляващата програма на контролера е аналогично на предишна авторска публикация [1].

Инициализиращата част на програмата за Arduino прототипа включва задаването като изходи на портовете, към които са свързани релета, инициализиране на Ethernet контролера (Ethernet Shield) за достъп до Интернет [2] и установяване на настройките на серийния порт. Инициализирането на Ethernet става чрез функцията Ethernet.begin(mac, ip).

Работната част на програмата е реализирана като краен автомат - КА.

На фиг. 4. е показан автоматният граф, по който работи контролерът за дистанционно управление на електроуред. Началното състояние на този граф е състояние „0“, в което уредът е изключен и се чете файла за този контролер от уеб сървъра. При неуспешно прочитане на данни от файла се прави нов опит след изтичане на програмно зададен timeout.



Фиг. 4 Автоматен граф

При прочитане на команда „ON“ крайният автомат преминава в състояние „1“, в което контролерът записва в променливата startTime момента на включване на уреда и едновременно с това задейства реле за включване. След това КА преминава в състояние „2“, в което се изчислява (в милисекунди) времето за работа на електроуреда и ако то е достигнало зададеното време се преминава в състояние „3“. Освен това в състояние „2“ се чете файла за този контролер от уеб сървъра за получена

нова команда. Ако командата е за промяна във времето за което трябва да е включен уредът, в състояние „2“ на КА се преизчислява времето за работа. Ако пък командата е за спиране на електроуреда (последната цифра от командата е 1) се задейства реле за изключване и КА преминава в състояние „0“. В състояние „3“ на КА релето за изключване се задейства за 5 секунди, тоест електроуредът се изключва и се преминава в състояние „4“. В състояние „4“ КА изчаква до получаването на команда OFF от потребителя. По такъв начин потребителят може многократно да повтаря едно и също време за работа на електроуреда, като контролерът всеки път разпознава командата като нова.

Програмния код на езика C е даден в [3].

3. ЗАКЛЮЧЕНИЕ

В настоящата работа са представени етапите от разработването на система за дистанционно управление на електроуред/и през Интернет като е направен литературен обзор върху Internet of Things- IoT. Разработена е web страница за дистанционно управление, която е сървърната част на системата. Уеб страницата е с възможности за авторизация и управление на достъпа до системата чрез автентикация с потребителско име и парола, като последната е криптирана съгласно съвременните изисквания за сигурност. Проектиран е контролер базиран на Arduino хардуер.

Форматът на командите, чрез които потребителя управлява електроуредът позволява да се добавят и нови команди например такива за циклично повторение на действия в денонощие. Друга възможност за бъдещо развитие на системата е използването на бази данни вместо файл за запазване на командите на съответния потребител.

ЛИТЕРАТУРА:

- Hristov, A., V. Vakadinova, T. Trifonov, Using Arduino for prototyping an alarm system, CAx Technologies Journal, issue No 4, December 2016, ISSN 1314-9628, pp. 27-31
- <https://www.arduino.cc/en/Reference/Ethernet> (23.11.18).
- https://github.com/sashkinaaa/Arduino_IoT
- <http://pubsubclient.knolleary.net/> (23.11.18)
- <https://store.arduino.cc/arduino-ethernet-shield-2> (23.11.18)
- Arduino Site - URL: <https://www.arduino.cc/> (23.11.18)
- <https://store.arduino.cc/arduino-4-relays-shield> (23.11.18)
- Oxer J., Blemings H, Practical Arduino: Cool Projects for Open Source Hardware, Apress, 2009, ISBN: 978-1-4302-2477-8.
- Панкратъев Д. Електроприборами управляет Facebook-Радио, 2018, № 9, с. 44-47