IJITCE

ISSN 2347- 3657

International Journal of
Information Technology & Computer Engineering

Email : ijitce.editor@gmail.com or editor@ijitce.com

International Journal of Information Technology & Computer Engineering

ISSN 2347–3657

Volume 12, Issue 4, 2024

BRIGHTNET IOT: NEXT-GEN INTELLIGENT LIGHTING SOLUTION

SURKA SRILATHA, Mr. M Syaam Babu

UG Student, Department of Electronics and Computer Engineering, JBIET, India.

Assistant professor, Department of Electronics and Computer Engineering, JBIET, India.

ABSTRACT

The aim of this thesis was to use various web development, Internet of Things (IoT) and software development technologies to create a smart home lighting system prototype. System's prototype development using more affordable hardware and software appliances to controlS home 12 Volt lighting devices without abundance of features, while maintaining the aspect of continuous common communication between devices of IoT systems. The thesis is structured into three main content parts: Part 2, background research for hardware and software components, programming languages as well as Web system and IoT system security issues, other technologies used in development of such management systems. Part 3, system's hardware and software subsystem specification. Part 4, system's and its components design model, results and inspection of implemented prototype. The development of the project showed that IoT systems use a vast variety of different technologies, including many variables when generating use cases for wanted results. The created smart home lighting system prototype was a successful, cheaper and simpler alternative to the current smart home management system consumer market, however, it lacks the plug-and-play nature of such expensive and less accessible systems like Apple Homekit or Fibaro.

INTRODUCTION

The contents of the chapter resolve around the relevance of the topic, research problem, work's aim and objective as well as workflow.

Relevance of the topic. Internet of Things enables various objects and devices that are used in daily life, automating them to transmit data and information over the network without requiring human supervision, thus facilitating the daily tasks of a modern-day man while reducing the resource use and providing him with valuable information while saving valuable time. IoT can be met almost anywhere, whether walking down the street through a smart pedestrian crossing, looking for a place to park a car while looking at a smartphone with an app that provides information on the status of a smart parking lot vacancies; these are some of the fields that help our daily lives. One of those fields is home automation. Smart home appliance systems allow the user to relax from their worries by connecting devices and objects to a network, controlling and managing them.

RESEARCH

Research chapter focuses on the task analysis, IoT technology implementation and definition for home appliance systems, various web development technology element analysis, hardware and software comparison and research used in both large and small scale IoT projects as well as commonly occurring web system security breach analysis

Task analysis

This thesis describes the development of a home lighting management system prototype using IoT technologies. The system requires the use of a variety of software and hardware technologies and resources to make it easy to use. The control system will be available to the user using a web browser on a computer or smartphone. The system prototype would allow the user to control unique, system-designed devices that can be controlled digitally and see their activity status in a graphical environment.

The prototype will use a variety of programming and scripting languages like: HTML5 hypertext markup language, CSS cascading template language, PHP hypertext dynamic interpretable programming language, Python3 interactive, open source programming language and various libraries for it, Arduino IDE microcontroller programming environment.

The system will be hosted on a physical single-board-computer, which will have a configurable web server. The connected user will be able to control the agreed and prepared lighting devices in the house. The system will be accessible online through a private network, however in order to protect the system from external malware, it is necessary to protect the database from SQL injections, eliminate unwanted connections and ensure data encryption. The microcontrollers of the managed and under development devices will be connected to single-board computer with a USB cable for data transfer.

Internet of Things technology

Internet of Things (IoT) is an information technology structure that encompasses a network of interconnected objects that enables the transmission of information over the network without the need for human-computer interaction. A person with a heart monitoring implant or cattle with an injected ID chip are a great example of IoT in our lives (Alexander S. Gillis 2019).

In the consumer market for home automation, IoT technologies are reflected in smart home systems like Fibaro or Apple HomeKit, including devices such as:

- Thermostats
- Luminaires
- Security cameras
- Smart wall plugs
- Humidifiers

These devices in the home appliance management system can be controlled by other compatible devices on the network, such as Apple smartphones with the iOS operating system. However, the above-mentioned home systems are expensive due to the abundance of integrated functions, which are difficult to adapt for simpler home management.

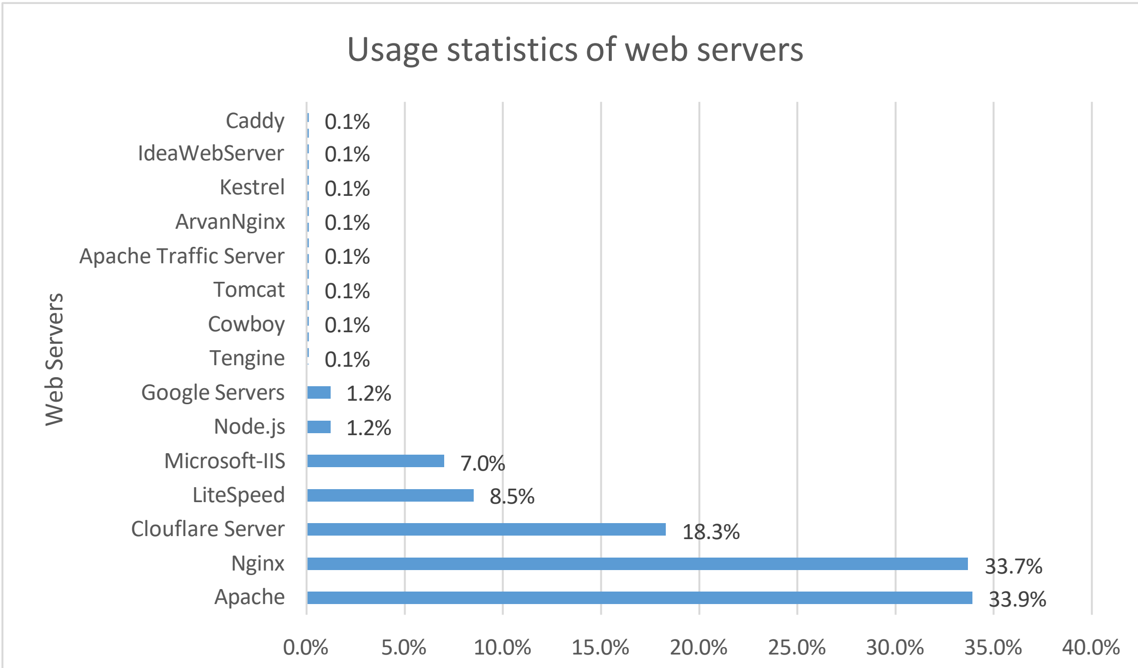

Figure 1 Web Server software usage statistics (W3techs, 2021)

1 SPECIFICATION

Chapter specifies hardware and software requirements for the development of the system as well as project's features.

1.1 Projected object

The purpose of the designed prototype is the control of a 12 V lighting unit using a web browser. Log in availability to the management system using a web browser. After logging into the system website, the user of the prototype – an authorized person whose house is equipped with the select lighting unit will be able to control its brightness and light up delay parameters, see its operational statistic information about it. The lighting unit connected to the microcontroller and designed will automatically illuminate the steps of the stairs of the house using motion sensors. Information between the microcontroller and the server will be transmitted via a USB serial port.

Projected object functions

Requirements for the functions of the developed system prototype:

1. Controlling brightness and light up delay time of LED strips through a graphical user interface implemented on a web server.

2. Rendering of information from a database server on a Web server site.
3. Accumulation of the lighting device's operation information into the database and its rendering on the website of the Web server.
4. Lighting systems prototype user authorization using logins.
5. Storage of information in a database.

1.2 Requirements for subsystems of the designed object

1.2.1 Requirements for hardware subsystem

Requirements for the technical characteristics of single-board computer:

- CPU: 1.4ghz 64bit
- Micro SD memory card: 16GB (at least)
- RAM: 1GB (at least)
- Ethernet port or Wi-Fi module

To take system's continuous and persistent operation and smooth user's experience into account, the SBC's hardware requirements are appropriately selected.

Requirements for the technical characteristics of microcontroller:

- High digital I/O port number
- USB serial connector support
- 128 KB flash memory (minimum)
- 5 V operating voltage

The amount of I/O ports ensures that the system could be scaled up if needed, more devices or sensors could be connected. Larger and more complex operational scripts could be implemented, thus, a bigger flash memory capacity is taken into account.

1.2.2 Requirements for the software subsystem and user interface

- Raspbian buster operating system with graphical user interface
- Chromium, open source, low-resource web browser
- Apache Web Server
- MySQL type database server
- Text editor, for programming tasks

For smoother developing and maintenance experience, the software subsystem requirements are appointed, ensuring low resource usage and more efficient workflow.

PROJECT DESIGN

This chapter focuses on home lighting management system's software and hardware subsystem component design.

Smart home lighting system's prototype structure and its components

Based on the carried out background research and the specification, the following key components are chosen for the development of the smart home lighting system's prototype:

- Raspberry Pi 3b 2GB single-board computer with 16GB memory micro SD card.
  Technical characteristics are suitable for use in small systems, the resources available in this SBC are sufficient to support the project web server and perform information transmission tasks, knowing that the system will be used by a single user.
- Arduino Mega 2560 R3 single-board microcontroller.
  The selected microcontroller has 54 digital I/O connections, so the designed system will be able to install new managed devices if necessary, and the relatively large Flash 256KB memory of this microcontroller will ensure that the programmable script code will not run out of space. Due to the USB connection used in this microcontroller, it will not require a separate power supply, as power will be provided via USB. Arduino devices provide a great opportunity to develop various IoT technology projects at a low cost (Monk, 2016).
- Apache2 Web Server.
  Apache web server was selected due to extremely detailed, official usage documentation (Httpd.apache, 2021), and large user community. Although not as efficient as Nginx, the Apache web server uses a generally small amount of resources and is highly configurable.
- MariaDB database server.
  This database is based on the MySQL database. It is supported and updated to strengthen it against security breaches. The MariaDB database is supported by Raspbian Buster operating systems, which uses a small amount of system resources, making it perfect for use on small systems while maintaining stability.

All Web server front-end and back-end processes are written using HTML, CSS and PHP programming languages. All Raspberry Pi single-board computer and Arduino microcontroller data communication is scripted using Python and its libraries. All tasks are written with "**GNU NANO**" text editor used in Raspbian Buster and other UNIX type operating systems. This text editor is not modern by today's standards, but is readily available for use in the Raspberry Pi command line while depleting minimal amount of resources, therefore there are no interferences during programming.

Arduino microcontroller tasks are scripted using Arduino IDE, which merges functions from both "C" and "C++" programming languages.

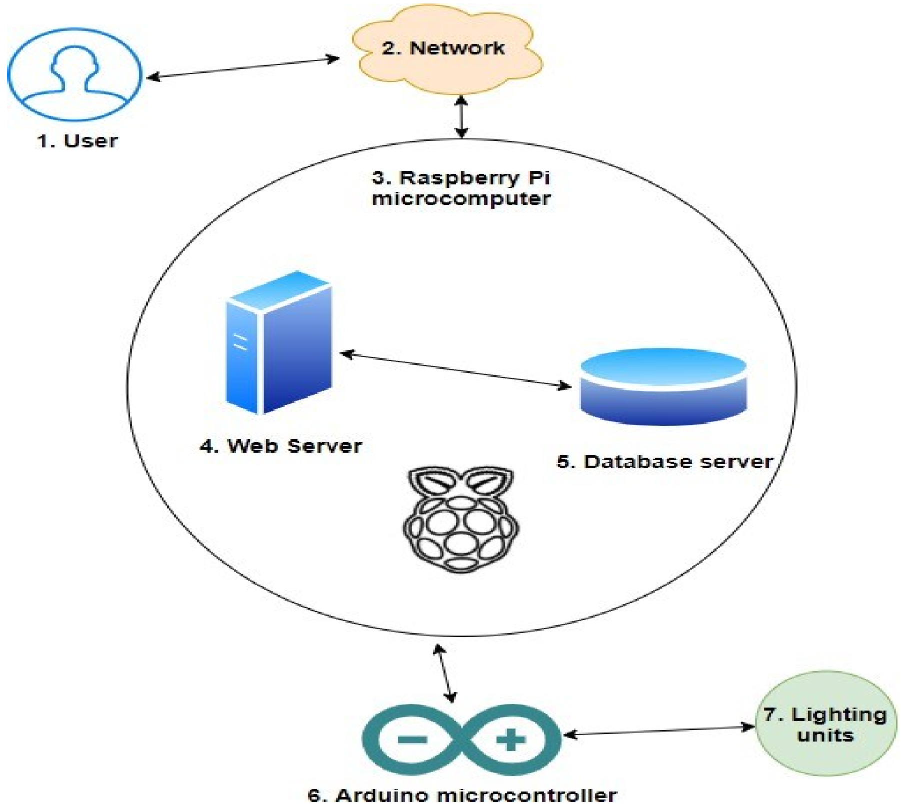

Figure 4 System's principle scheme

According to Figure 4, in the designed smart home lighting system, each component will transmit information to each other using various technologies for common communication. The user will need a smart device that could browse the web to access the system. Raspberry Pi will act as a host to both Web and Database servers. It will both collect and send digital data over the USB serial cable to the Arduino microcontroller.

Principle scheme components:

1. **User**: authorized system user who controls available parameters for the lighting unit.
2. **Network**: local area network where system prototype is implemented.
3. **Raspberry Pi single-board computer**: device on which software is installed and programmed: web server, database, software for communication between the single-board computer itself and the microcontroller in the system. A USB port is used for physical communication between the controller and computer.
4. **Web Server:** software to support the system and graphical user interface.
5. **Database server**: software to store information about the system, manageable lighting device parameters, user login data.

ISSN 2347–3657
Volume 12, Issue 4, 2024

6. **Arduino microcontroller**: SBM used to read inputs and turn them into output signals.
7. **Lighting units**: Various devices for lighting, in this case, 12 V LED strips.

Raspberry Pi 3 B+ configuration and software installation

For the implementation of the project, the single-board computer requires the installation of the Raspbian buster operating system, which is based on Linux Debian. Therefore, the list of commands running on the terminal corresponds to other distributions of Debian operating systems. Using the mentioned terminal further software installation and configuration work is performed (Figure 5).

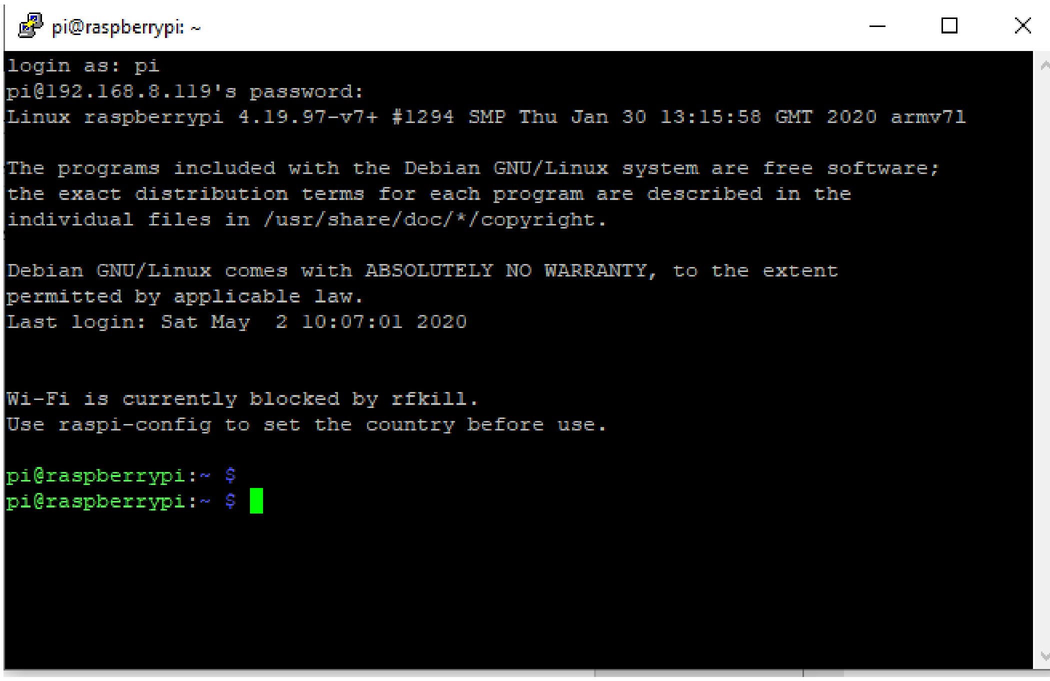

Figure 5 Raspberry Pi's Raspbian OS Terminal

Advanced Package Tool (APT) is used inside the terminal in order to download and install needed software packages. The followed command is typed in the command line: **sudo apt-get install** *selected package name.*

The following software packages will be installed on the Raspberry:

- Arduino IDE – official Arduino microcontroller programming environment.
- Apache2 – Web Server.
- MariaDB – Database server.
- Php – Php programming language compiler.
- Python-mysql.connect – python programming language library for MySQL database connection

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 12, Issue 4, 2024

initialization.

- phpMyAdmin – database graphical user interface.

Once the select packages are installed, a data model can be projected for our mariaDB database server using the phpMyAdmin graphical user interface on a web browser.

1.3 Database data model

In order to implement common communication between systems components, it is necessary to ensure that the database server is designated to:

1. Store user login data that will allow them to log in to the system using a web browser.
2. Store information for changing the settings of a controlled lighting unit. This data is updated when the user sends the query using the graphical interface.
3. Store data intended to provide the user with information on the frequency of use of the controlled lighting unit in the graphical interface.

When using a database, efforts are made to avoid duplication of data (data redundancy). However, all tables in the database and their entities are not dependent on each other. The purpose of the tables is to implement a platform where various data could be transmitted between the system components and rendered on a graphical user interface for the user.

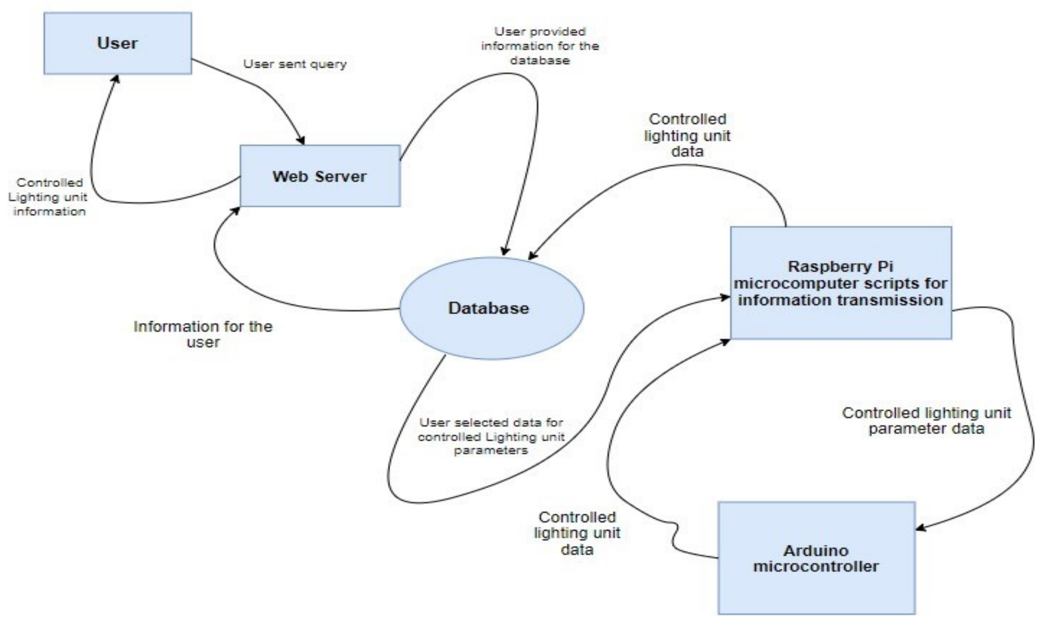

Figure 6 System's DFD-0 diagram of data flows

Database normalization tables and their structure

Database tables are created accordingly to the data flow diagram (*Figure 6*).

ISSN 2347–3657
Volume 12, Issue 4, 2024

Table “Logins”

The "Logins" table used in the database is designed to store data for a user's login to the system. This will ensure the use of the authorized person system. Fields that make up the table (Figure 7):

1. “userID“ – Primary key
2. “Username“– unique key, users name
3. “Password“ – pre-generated password for the user to use while logging into the system

#	Pavadinimas	Tipas	Palyginimas	Atributai	Null	Nutylint	Komentarai	Papildomai	Veiksmas
1	userID	int(11)			Ne	*Jokio*		AUTO_INCREMENT	Redaguoti Šalinti Pirminis Unikalus Indeksas Spatial Daugiau
2	Username	varchar(111)	utf8mb4_general_ci		Ne	*Jokio*			Redaguoti Šalinti Pirminis Unikalus Indeksas Spatial Daugiau
3	Password	longtext	utf8mb4_general_ci		Ne	*Jokio*			Redaguoti Šalinti Pirminis Unikalus Indeksas Spatial Daugiau

Figure 7 Database table „Logins“ structure

Table “JudesioData”

The purpose of the table is to store information about the operation of the controlled lighting unit. Fields that make up the table (Figure 8):

1. “judesioID“ – primary key
2. “Aptikas“ – operation description field
3. “Laikas“ – automatic timestamp of the record.

#	Pavadinimas	Tipas	Palyginimas	Atributai	Null	Nutylint	Komentarai	Papildomai	Veiksmas
1	judesioID	int(11)			Ne	*Jokio*		AUTO_INCREMENT	Redaguoti Šalinti Pirminis Daugiau
2	Aptiktas	varchar(111)	utf8mb4_general_ci		Ne	*Jokio*			Redaguoti Šalinti Pirminis Daugiau
3	Laikas	timestamp			Ne	current_timestamp()		ON UPDATE CURRENT_TIMESTAMP()	Redaguoti Šalinti Pirminis Daugiau

Figure 8 Database table „JudesioData“ structure

Table “ledBrightness”

The purpose of the table is to store information for the settings of the controlled lighting device that the user will be able to change when logged into the system. Fields that make up the table (Figure 9):

1. “BrightID“– primary key
2. “Brange“– light intensity parameter of the controlled lighting unit
3. “Drange“– parameter of the controlled lighting unit light up time delay.

International Journal of Information Technology & Computer Engineering

ISSN 2347–3657

Volume 12, Issue 4, 2024

	#	Pavadinimas	Tipas	Palyginimas	Atributai	Null	Nutylint	Komentarai	Papildomai	Veiksmas
	1	**BrightID**	int(11)			Ne	*Jokio*		AUTO_INCREMENT	Redaguoti Šalinti Daugiau
	2	**Brange**	varchar(111)	utf8mb4_general_ci		Ne	*Jokio*			Redaguoti Šalinti Daugiau
	3	**Drange**	varchar(111)	utf8mb4_general_ci		Ne	*Jokio*			Redaguoti Šalinti Daugiau

Figure 9 Database Table „ledBrightness" structure

All database tables are created using the graphical user interface software package *phpMyAdmin*.

1.4 Lighting unit and Arduino microcontroller design

For the implementation of this work, it was chosen to design and create a 12 step stair lighting device, which, after integrating PIR motion sensors, would sequentially light up the 12-volt LED strips in the stair steps, at the exact moment when the movement directed to the stairs is detected. Two LED parameters could be changed by the user, the LED brightness and sequential light up delay time.

The total count of stair steps is used to generate a rough estimate of calories burned while using the stairs.

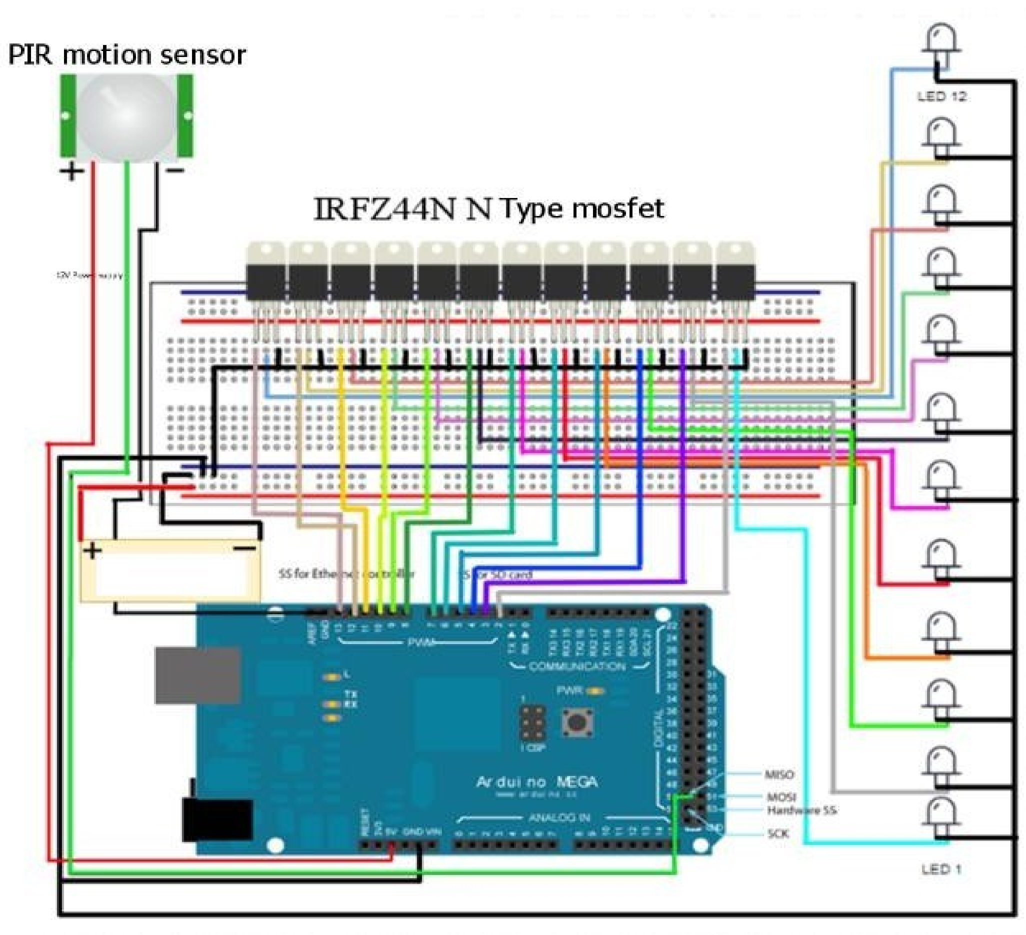

Figure 10 Lighting unit and Arduino microcontroller connection scheme

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 12, Issue 4, 2024

The diagram of the design stair lighting device shows the connection of all components used (Figure 10). The components included:

1. Twelve IRFZ44N N type MOSFETS
2. Twelve 12 Volt LED Strips
3. Arduino Mega 2560 R3 microcontroller
4. 12 Volt power supply
5. PIR motion sensor

A 12V power supply must be used to feed the 12 volt appliances. However, the working voltage of the Arduino mega 2560 R3 microcontroller is only 5 volts, therefore, in order to control 12 volt devices, semiconductor devices – transistors or relays – need to be inserted into the electrical circuit (Seedstudio, 2020). For this work, the use of type N conductive channel metal, oxide and semiconductor field transistors (MOSFET) IRFZ44N has been selected, which has three control electrodes – G "*Gate*", D "*Drain*" and S "*Source*" (Figure 11).

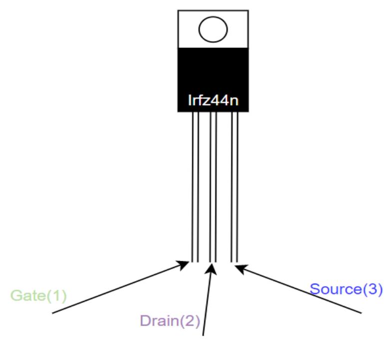

Figure 11 IRFZ44N MOSFET electrodes

LED strips and IRFZ44N MOSFETS

The poles of the negative electric current source of all LED strips are separately connected to the MOSFET IRFZ44N Drain electrodes. The positive electric current poles of the LED strips are commonly connected to the 12-volt power supply positive pole. The MOSFET source electros are commonly connected to the negative pole of the power supply, and the remaining gate electrodes of the MOSFETS are connected in series with the Arduino mega 2560 R3 microcontroller PWM (Pulse-width modulation) ports labeled 2–13, which, depending on the transmitted signal complete the circuit between LED strips and the 12V power supply. IRFZ44N MOSFET will act as a switch, controlled by Arduino microcontroller, whose GND (ground) port is also connected to the 12V power supply negative pole in order to complete a common electric circuit.

HC-SR501 PIR motion sensor and Arduino Mega 2560 R3

The purpose of the motion sensor is to ensure that led strips light up automatically, when user comes close to the stairs. After detecting movement,

ISSN 2347–3657
Volume 12, Issue 4, 2024

the sensor will transmit a HIGH signal to the Arduino microcontroller, which will initiate a sequential LED light-up of the stairs. The HC-SR501 motion sensor has a working electric voltage of 4.5V to 12V, so it can be fed using the 5V output port of the Arduino. The PIR motion sensor used has three pins GND - Ground, HIGH/LOW Dout - digital signal transmission pin and Vcc - positive electrical current connector (). When using the OUT pin, depending on whether the sensor has detected movement, a HIGH or LOW signal will be transmitted to the controller.

Table 4 PIR motion sensor and Arduino connection table

HC-SR501 PIR pins	**Arduino mega 2560 R3 ports**
Out(Output)	Digital 50 (Input)
GND	GND (Output)
Vcc	5V (Output)

Table 4 represents the motion sensor and Arduino microcontroller pin connections correspondingly.

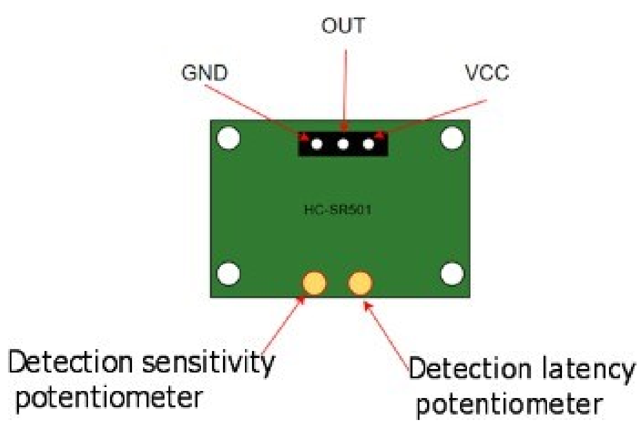

Figure 12 HC-SR501 PIR motion sensor

The sensor detection sensitivity and latency are adjusted physically using the potentiometers.

Arduino microcontroller code

To program the logical work and parameter management scenario of the stair lighting device, Arduino IDE software is used in the Raspberry Pi SBC.

Arduino board is connected to the Raspberry Pi 3 B+ using a USB serial cable, which provides the needed 5 V power as well as ability to send and receive data from the computer.

Variable and pin declaration

Based on the connection diagram, the program initially defines the connections of all IRFZ44N MOSFET gate electrodes used, PIR motion sensor's Dout to microcontroller ports, and other variables that are assigned as output in the void setup() function. Serial line bandwidth equal to 9600 bits per second (Figure 13).

```
int ledPin1 = 13;
int ledPin2 = 12;
int ledPin3 = 11;
int ledPin4 = 10;
int ledPin5 = 9;
int ledPin3 = 8;
int ledPin1 = 7;
int ledPin2 = 6;
int ledPin3 = 5;
int ledPin1 = 4;
int ledPin2 = 3;
int ledPin3 = 2;
int PIRpin = 50;
int pirState = LOW;
int val = 0;
int X;
int brightness;
              void setup() {
                pinMode (ledPin1, OUTPUT);
                pinMode (ledPin2, OUTPUT);
                pinMode (ledPin3, OUTPUT);
                pinMode (ledpin...);
                pinMode(PIRPin, INPUT);

                Serial.begin(9600);
              }
```

Figure 13 Pin declaration

Serial line read and variable assignment using ASCII encoding and decoding

Arduino uses ASCII character encoding for electronic communication over the serial line, meaning that various encoded data could be either sent or received in order to control various signal outputs for the connected electronic appliances, in this case - 12 V LED strips and PIR motion sensor. Using function "*void loop()*" ensures that the information is transmitted or read in a real-time cycle. Depending on the information received, ASCII values are assigned to corresponding variables inside the code (Table 5).

ISSN 2347–3657
Volume 12, Issue 4, 2024

Table 5 ASCII value to variable assignment table

ASCII values	Equivalent variable
118	brightness = 20;
108	brightness = 80;
109	brightness = 120;
107	brightness = 255;
104	X = 500;
111	X = 1000;
112	X = 1500;
116	X = 2000;

```
void loop(){
  if (Serial.available()){
  switch(Serial.read()){
    case 118:
    brightness = 20;

    break;
    case 108:
    brightness = 80;

    break;
    case 109:
    brightness = 120;

    break;
    case 107:
    brightness = 255;
```

Figure 14 ASCII value assignment to variables

Switch statement is used to help assign each ASCII value read from the serial line to the corresponding variables.

2 CONCLUSIONS

1. Inspected the current market for smart home control systems Fibaro HomeCenter and Apple HomeKit, their identified advantages: systems are complete, have many functions; and disadvantages: extremely high prices, compatible only with specific devices.
2. Web development and IoT project programming languages and common security breaches of web systems and applications have been researched and inspected. Equipment chosen to achieve the work goal: Raspberry Pi 3b, Arduino Mega 2560 R3, Apache2 web server, mariaDB database, Php internal programming language, HTML programming language, Python programming language.

3. The selected Web server and database are setup and configured for storing and sharing information between devices and home lighting unit system components on the local network.
4. A 12 Volt LED strip stair lighting unit has been designed and developed to be controlled by the arduino microcontroller with the help of a PIR motion sensor and IRFZ44N N type MOSFETS. Written Python software scripts for single-board computer and controller.
5. A graphical user interface was designed and implemented onto the configured apache2 Web server, user is able to control two parameters of the controlled lighting unit by connecting to the system via a web browser, as well as review it‘s operational information.
6. System`s prototype task operation was inspected.

The home lighting system has been researched and designed, which automates lighting devices in user‘s home, thus providing additional comforts and the ability to monitor the operation of the lighting units in a web browser. An experimental prototype using IoT technologies, single-board computers and microcontrollers, computer network technologies, programming technologies has been developed. A prototype system has been designed to allow the installation of new controlled appliances if necessary.

REFERENCES

Mohan Krishnamurthy, Eric S. Seagren, Raven Alder, Aaron W. Bayles, Josh Burke, Skip Carter, & Eli Faskha 2008. Chapter 11 - Apache Web Server Hardening. In How to Cheat at Securing Linux (pp. 383-401). Elsevier.

Ramoška, A. 2012. Apsaugos nuo SQL injekcijų el.verslo svetainėse metodikos sudarymas ir tyrimas: Magistro darbas. Kaunas: Kauno technologijos universitetas. Prieiga per eLABa – nacionalinė Lietuvos akademinė elektroninė biblioteka.

Dow, Colin 2018. Internet of things programming projects: Build modern IoT solutions with the Raspberry Pi 3 and Python. Birmingham: Packt Publishing.

Monk, Simon 2016. Programming Arduino: Getting started with Sketches (2nd ed.). New York, N.Y.: McGraw-Hill Education.

Devguide.python.org 2021. Python developer‘s guide. Available at:https://devguide.python.org/ [Accessed 2021-02-05].

Httpd.apache.org 2021. Apache2 official Web Server documentation. Available at: https://httpd.apache.org/docs/2.4/ [Accessed 2021-02-06].

Dev.mysql.com 2021. Official MySQL documentation. Available at: https://dev.mysql.com/doc/ [Accessed 2021-02-07].

Dev.mysql.com 2021. Python and MySQL connection. Connecting to MySQL Using Connector/Python. Available at: https://dev.mysql.com/doc/connector- python/en/connector-python-example-connecting.html [Accessed 2021-02-15].

ISSN 2347–3657
Volume 12, Issue 4, 2024

W3schools.com 2021. Php and other programming language online learning environment. PHP Tutorial w3schools.com. Available at: https://www.w3schools.com/php/ [Accessed 2021-02-20].

Seeedstudio.com *2020*. Arduino High voltage device control. Available at: https://www.seeedstudio.com/blog/2020/01/03/arduino-tutorial-control-high- voltage-devices-with-relay-modules/ [Accessed 2021-03-01].

W3Techs.com 2021. W3Techs provides survey information about the usage of technologies on the web. W3techs.com Usage statistics of web server. Available at:https://w3techs.com/technologies/overview/web_server [Accessed 2020-03-28].

Blog.ptsecurity.com 2017. *Most common Web system attacks*. *Positive Technologies – learn and secure*. Available at: http://blog.ptsecurity.com/2017/09/web-application-attack-statistics-q2.html [Accessed 2021-03-05]

Alexander S. Gillis 2019. Definition of Internet of Things (IoT). Available at: https://internetofthingsagenda.techtarget.com/definition/Internet-of-Things-IoT [Accessed 2021-02-29].

Gómez, A., Cuiñas, D., Catalá, P., Xin, L., Li, W., Conway, S., & Lack, D. (2015). Use of single board computers as smart sensors in the manufacturing industry. Procedia engineering, 132, 153-159.

Güven, Yılmaz, et al. "Understanding the concept of microcontroller based systems to choose the best hardware for applications." (2017).