REVERSE ENGINEERING IN IOT AND CPS

USING A BLE RGB LED LAMP AS AN EXAMPLE

HAMK
HÄMEEN AMMATTIKORKEAKOULU
HÄME UNIVERSITY OF APPLIED SCIENCES

Bachelor's thesis

Automation Engineering
Valkeakoski

Spring, 2018

Roman Lutai

HÄMEEN AMMATTIKORKEAKOULU
HÄME UNIVERSITY OF APPLIED SCIENCES

Automation Engineering
Valkeakoski

Author Roman Lutai **Year** 2018

Subject Reverse engineering in IoT and CPS using a BLE RGB LED lamp as an example

Supervisor(s) Jukka Pulkkinen

ABSTRACT

The objectives of this thesis project are to study industrial revolutions, to foresee further turns of industrial revolution, to suggest necessary skills for that and apply these skills into practice. These objectives were achieved by researching the history of industrial revolutions, following the news of the modern history and inducting current trends to the logical conclusions and applying scientific techniques into practice. As a result was discovered an exponential growth of complexity in technologies both on industrial fields and in everyday tasks. The current technological trend is Cybernation. However, cyber-physical objects are often closed for third-party applications. It may decrease the demand on Cyber Physical Systems. By chance, this challenge can be eliminated by means of reverse engineering. Its methods were studied and applied into practice in this project. Consequently, a light-control application was made. Its main advantage to the original program in the code, which is open source and can be used for new solutions.

Keywords Industry revolutions, Cybernation, IoT, CPS, reverse engineering, BLE, RGB LED, software developing.

Pages 20 pages including appendices 47 pages

CONTENTS

1 INTRODUCTION

Industry has been evolving in parallel with the society. As much as yet another revolutionary discovery was changing industry, as well it was affecting people's routines. The biggest four industrial revolutions have taken place during the last several centuries. In the time span from the 18th to the 21st century the industry has emerged from the first mechanical industrial machine on steam power to automated lines of mass-production and smart factories with the utilization of IoT (Internet of Things) and CPS (Cyber Physical Systems) and it still keeps emerging. Therefore, to see the steps, which industry has been taking during each industrial revolution, its history has been studied in the current thesis work.

The current industry trend is cybernation. Cybernation is the fusion of the physical, digital and biological worlds with the utilization of CPS, IoT, Big Data and many other currently emerging technologies. However, CPS objects are often integrated to their closed applications, so they are not applicable to other solutions. This problem can be solved with reverse engineering. Different processes of reverse engineering were studied in this thesis project, because it can be used for designing interoperable products, but for illegal copying of intellectual property as well.

The availability of interoperable solutions is important for the product, as this creates a wide environment for the product's applications and can give for a developer a freedom of choice. Creating an interoperable solution is presented in the practical CPS application of this thesis.

2 HISTORY OF INDUSTRY

The world we are living in has experienced four main industrial revolutions. The last one is taking place right now. Each of these revolutions has brought important inventions to technology and work processes. These revolutions can be characterized as follows.

2.1 First Industrial Revolution – revolution of mechanization

The First Industrial Revolution took place from the end of the 18th century to the beginning of the 19th century. Nevertheless, it begun in Great Britain and its government tried to prevent spreading their innovations; Industrial Revolution also gained great success in whole Europe and American States. (Industrial revolution, 2010)

Most of innovations that time were starting from the textile industry. It was a leading industry with large labor power involved. Therefore, textiles were the leaders of modernization back then. (Landes, 1969)

Invention of the steam engine provided the mechanical energy wherever it was necessary, without relying on the weather conditions, unlike the water and wind power sources used before. (Temin, 1966)

Mechanized spinning and weaving tools powered by the central steam engines had moved the major labor from hand production methods on manufactures to the machines on the factories. (Landes, 1969)

The Industrial Revolution marks a major turning point in history. For the first time in history, almost every aspect of daily life began to exhibit remarkable consistent growth. However, at the dawn of the First Industrial Revolution, an economic recession has occurred. The next significant growth appeared with the inventions of the Second Industrial Revolution. (Lucas, 2002)

2.2 **Second Industrial Revolution – revolution of mass production**

The Second Industrial Revolution spans from the middle of the 19th century to the middle of 20th century. It was the time of different inventions, such as electricity, telegraph, petroleum, internal combustion engine, cars and even planes, which have changed our world once again. (Engelman, 2015)

The Bessemer process of yielding steel was a great advance in steel making. The new process was developed by Sir Henry Bessemer and patented in 1856 by him. The process allowed expanding the scale and speed of steel production, and reducing its labor requirements. Such enhance of technology led to mass-production of this vital material, expanding related railroad, and construction industries. (Muntone, 2012)

Another remarkable event, which signed the beginning of industrial revolution, was utilizing electrical power, which started from Michael Faraday discovery in years of 1831-1832. It led to series of inventions as telegraph, telephone and radio, which in combination with railroad network expansion, allowed exceptional migration of people and exchange of ideas. (Second Industrial Revolution, 2017)

Another combination of electricity with freshly discovered power source, petroleum, led to the invention of internal combustion engine with following inventions of a car and even a plane powered by it.

Electricity had great success in any field of industry and normal life. The process of providing electricity to a specific place is called electrification. National Academy of Engineering has called electrification as "the most

important engineering achievement of the 20th century". (Constable, Somerville, 2003)

Electrification resulted in the significant improvements of production as assembly line and mass production. (Ford, 2008)

2.3 Third Industrial Revolution – the Digital Revolution

The Third Industrial Revolution appeared in the 20th century and it is ongoing. This revolution, also known as Digital Revolution, is characterized by the discovery of transistor, which led to the replacement of analog electronics and devices with the digital information and digital operating devices. (Schoenherr, 2004)

Main advantages of the digital information over analogue are in its precision and ability to replicate identical copies of whether data or products made by digital operated machines. This advantage took to the new level techniques of mass production, discovered in the end of the previous industrial revolution.

Fundamental research started in USA in the late 1960s by Defense Advanced Research Projects Agency (DARPA) has led to developing protocols of digital data exchange TCP/IP. These protocols became a ground for the modern Internet. The day, when protocols were permanently activated was 1st of January 1983. (Bryant, 2011)

The same time computing machines were also emerging and profiling from the digital integration circuits to PLCs, routers and PCs on microprocessors, thereby increasing productivity and changing business and production traditions. (Schoenherr, 2004)

Achieving PC, Internet, digital cellular phone and 2G connection to the public use led to the digitalization of social systems, transforming them into the modern information society. (Perez, 2011)

2.4 Fourth Industrial Revolution – Industry 4.0

The current trend of a fusion between the physical, digital and biological worlds indicates the next industrial revolution, also known as Industry 4.0. (Marr, 2016)

Each industrial revolution was based on the previous one, as does the Fourth Industrial Revolution, as can be seen in Figure 1.

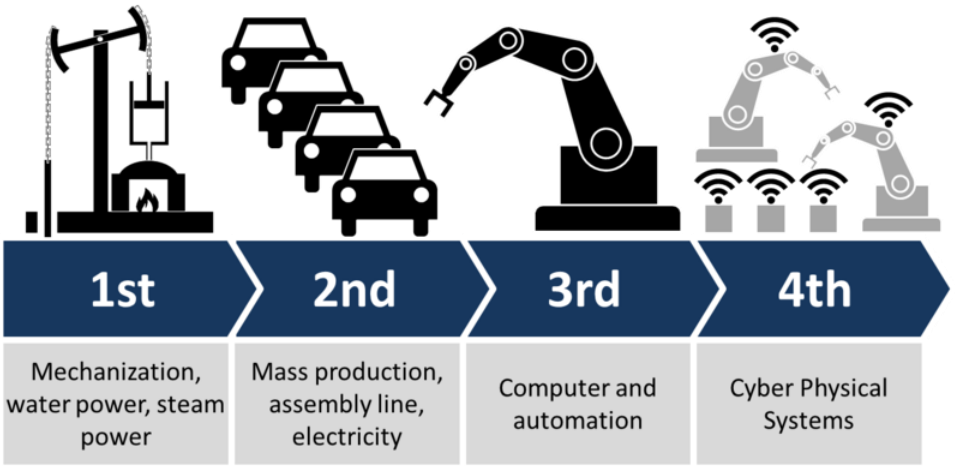

Figure 1. Sequence of industrial revolutions leading to new inventions (Roser, 2015)

Sensors, PLCs, robots and digital telecommunications - are cybernetic elements, which are performing growth in complexity nowadays. Their active utilization transforms physical systems into cyber-physical systems. Connecting these systems to the global web creates the Internet of Things (IoT) and the Internet of Systems. (Marr, 2016)

Moreover, big data analysis, based on results from sensors of cyber-physical systems, potentially allows increasing the efficiency of factories making them “smart”. It leads to better asset management, which would reverse the damage caused to the nature by previous revolutions. (Schwab, 2017)

These and a range of other breakthroughs in different fields also influence on our society and often appear in our life today:

- smart houses improve the efficiency of services;
- new devices are augmenting the physical world with digital information and are accessible from any shop for an ordinary user;
- a wide utilization of artificial intelligence even challenges the definition of consciousness.

(Schwab, 2017)

Industry 4.0 may result in smart healthcare, smart electrical grids and even smart cities. These and many other CPS solutions would not be running well without proper interoperability of their units. (Zhong, Xu, Klotz, & Newman, 2017)

3 REVERSE ENGINEERING FOR CYBER-PHYSICAL SYSTEMS

Current trends of the industrial evolution are showing rapid emergence of cyber physical systems. Nevertheless, cybernetic objects are often having a closed connection with a certain application, which makes it almost impossible to integrate these objects into a new solution. However, capacity for interoperability is crucial for emerging smart systems. (Zhong, Xu, Klotz, & Newman, 2017)

Therefore, it is vital to be able to recognize functions and communication protocols of a product to utilize it in a particular solution, different from the original one. The process of this product analysis is called reverse engineering, or back engineering.

Reverse engineering is a process of examination of an existing product to achieve necessary specifications about the object. This information can be used for the replication or proper utilization of chosen object. (Eilam & Chikofsky, 2005)

Methodology of reverse engineering mainly replies to questions how an object was created and how it operates. Reverse engineering in mechanics usually means creating a documentation after analyzing a disassembled object. However, back engineering can be applied for other fields, beyond mechanical assemblies. Similar methods applicable in chemistry, biology, software and electrical development.

Usually, the main intention of people applying reverse engineering is to replicate a product, which, in combination with ownership legislation, has brought this field to the fine line of legality. However, violating owners' rights is not a part of the process definition. Reverse engineering contains set of tools, which can be used for good or evil, depending on the person applying them. Whereas a negative application of reverse engineering is widely known, it also has a positive utilization, some cases of which listed below:

- Documentation of an under development product: usually people, who developed a feature and who documented it for an automatization, are different persons.
- Legacy product – is a product, which is no longer produced and does not have drawings to reproduce it. Similar definition relates to the obsolete product, which is not supported by Original Equipment Manufacturer (OEM).
- Partial investigation for repairing or servicing products described above also relates to reverse engineering.
- Implementing improvements to the product does also require reverse engineering in case of missed documentation.

- Crime prevention, like reverse engineering of malware, or a product investigation for vulnerabilities to reinforce weak sides.
- Unhandled errors and undocumented failures are also requiring investigation by reverse engineering.

(Eilam & Chikofsky, 2005)

The definition of reverse engineering may look the same as for scientific research. The only difference between reverse engineering and scientific research is in the object of research: whereas scientific research usually aims for understanding natural conditions, though the object of reverse engineering is an artificial product. (Eilam & Chikofsky, 2005)

4 PRACTICAL CPS APPLICATION

With the emergence of cyber-physical systems, the border between embedded engineering and full-stack software developing has completely blurred out. Since the moment when embedded devices became accessible from PC and smartphones, the full user experience is also depending on a proper application for smart devices.

Nevertheless, cybernetic objects are often designed for specific solution and cannot be used for different applications. Thus, reverse engineering is an important skill for engineer of recognition functions of the product for successful it integration into the own application.

For a proper demonstration, the cybernetic object of choice has to have several characteristics:

- Its solution has to be closed for the specific application.
- It has to be able to connect to smart devices (Smartphones, PCs) with the wireless connection (Bluetooth, Wi-Fi).
- The application has to be clear and available for the broad masses.

To fit the criteria listed above for this thesis project a smart light bulb from the Chinese company Magic Blue was chosen. Its first look can be seen in Figure 2.

Figure 2. Smart lamp from Magic Blue (Magic blue LED bulb, n.d.)

4.1 **Smart lamp from Magic Blue**

The smart lamp from Magic Blue is a set of LED lights with a Bluetooth controller. It has a shape of a light bulb and it can be installed instead of a regular light bulb. Therefore, it can be considered as both a lightbulb because of a shape and a lamp because of a complexity of its components. However, considering that this is a cyber physical object, it is called a smart lamp in this thesis work. Figure 3 shows its main components:

1. Acrylic cover, which protects the controller from dust and physical impacts, also diffuses light for softer illumination;
2. Light controller, which consists of LED lights and a Bluetooth controller;
3. Aluminum body in form of a light bulb;
4. Light bulb base E27, the standard base for most of the lamps.

(Magic blue LED bulb, n.d.)

Figure 3. Smart lamp components (Magic blue LED bulb, n.d.)

Detailed information about a smart lamp is listed in Table 1:

Table 1. Detailed characteristics of a smart lamp from Magic Blue (Magic blue LED bulb, n.d.)

Model Number:	ZJ-BTBD-RGBW
Type:	Bulb
Base Type:	E27
Light Source:	LED
LED Light Source:	SMD2835(24) 5050(8)
Input Voltage:	100 V … 240 V
Lamp Power:	10 W
Lamp Luminous Flux:	800 lm
Color Temperature(CCT):	RGBW
Color temperature:	2700 K … 6500 K
Working Temperature:	-20°C … 55°C
Working Lifetime:	50000 h
Size:	130mm X 64mm

Lamp Body Material:	Aluminum
Certification:	CE, FCC, RoHS, UL
Waterproof grade:	IP25
Place of Origin:	Guangdong, China (Mainland)
Features:	waterproof shockproof sun-proof
Control type:	Android, IOS iPhone iPad
Output channels::	3/4 CHNS

LED lights are connected in two series to the light controller as shown in Figure 4:

1. Outer circle of 24 white LEDs 2835 is responsible for bright white light with adjustable brightness.
2. Inner circle of eight RGB LEDs 5050 is responsible for the color mode in the lamp.

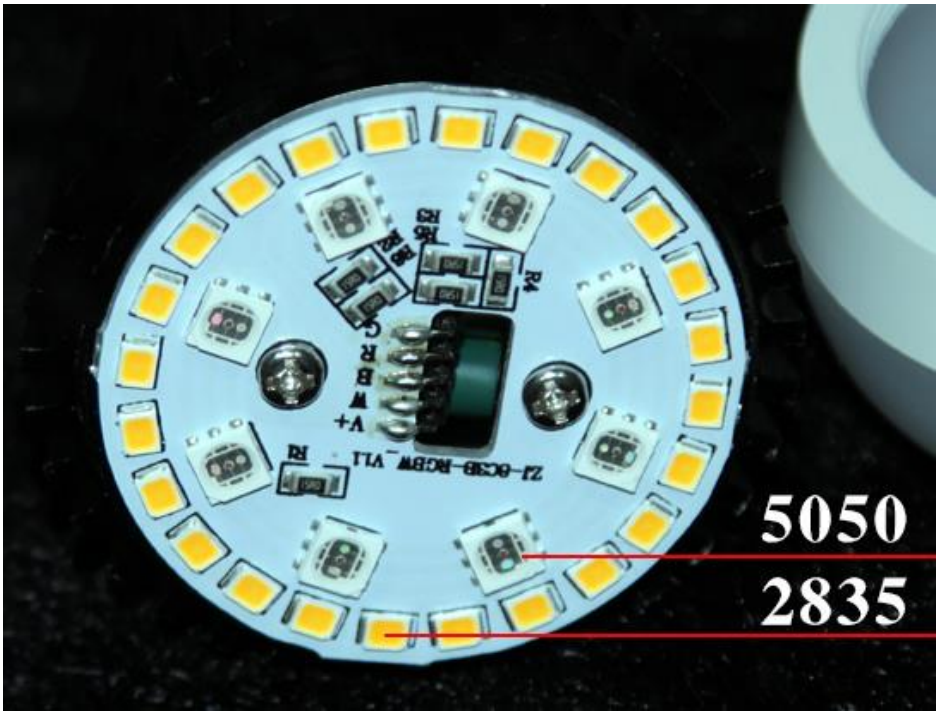

Figure 4. Light controller with sets of LEDs and the Bluetooth module in the middle (Magic blue LED bulb, n.d.)

Figure 4 illustrates the main electronics of the Bluetooth module in the middle of a picture. The smart lamp from Magic Blue uses a novel wireless Bluetooth connection called Bluetooth Low Energy connection, or shortly BLE.

Nevertheless, the lamp can only change its colors from dark- to bright-white or in the RGB specter; its original software extends its application with a wide number of functions.

4.2 The original Magic Blue application

The Smart lamp is supported with the original application for smartphones from Magic Blue. Its appearance can be seen in Figure 5.

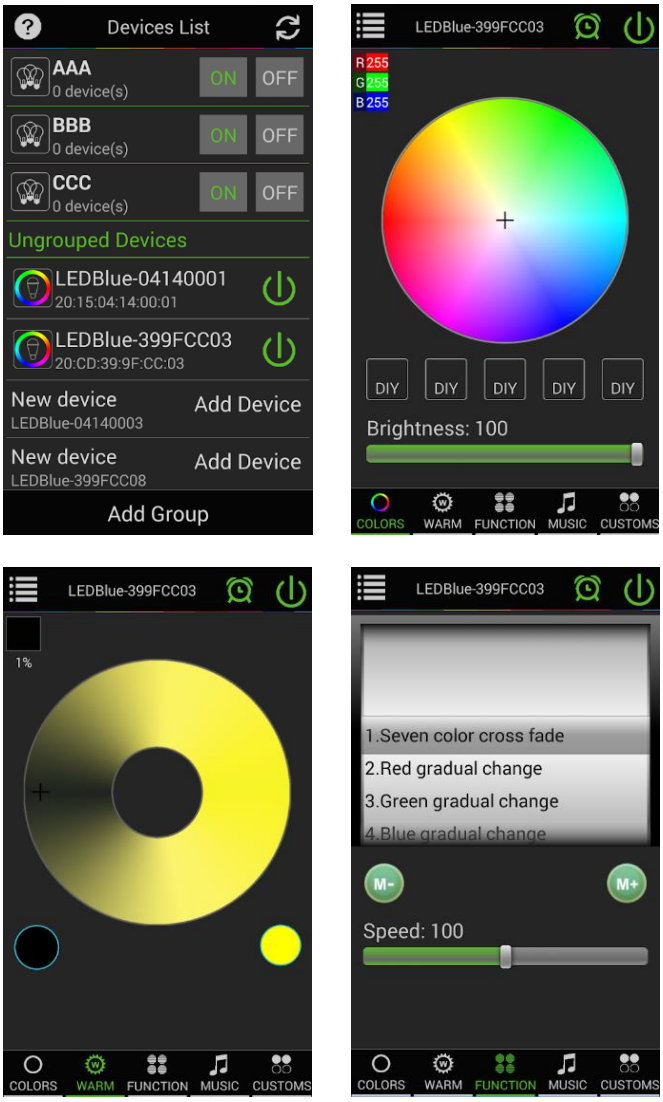

Figure 5. Screenshots of several layers of a smartphone application from Magic Blue (LED Magic Blue, n.d.)

The application is available for Android and iOS and provides the following functions:

- Controlling separate lamps from Magic Blue, or groups of them;
- Turning the light On and Off;
- Scheduling the state of lamp;
- Changing the colors of the lamp in the “Colors” mode;
- Changing the brightness of the lamp in the “Warm” mode;
- Sending the sequence of changing states to the lamp in the “Function” mode;

- Adjusting the state of the lamp according to the music or smartphone's microphone in the "Music" and "Disco" mode respectively;
- Creating own sequences of changing states in the "Customs" mode;
- Choosing the color for the lamp from a picture on the phone or its camera.

(LED Magic Blue, n.d.)

Hence, the wide functionality of the smartphone application increases the number of possible ways to utilize the smart lamp.

Yet, the solution from Magic Blue cannot ever cover all possible applications, but its closed solution limits development potential. Besides, as long as it is a closed solution, there is no documentation or open-source code to create new applications based on this lamp. As a result, anyone, who would like to integrate the lamp into a new solution, will need to discover its protocols, in other words, to reverse engineer it.

4.3 **Bluetooth protocols of the lamp**

The chosen smart lamp uses a Bluetooth Low Energy (BLE) connection. BLE is a novel wireless Bluetooth connection. It differs from the Bluetooth 4.0 and it necessary to know communication protocols of the object to establish a proper connection with the smart device.

4.3.1 Comparison of BLE and Bluetooth 4.0

Bluetooth Low Energy, previously known as Bluetooth Smart, is a wireless network, which comply with standards of Wireless Personal Area Network (WPAN) as like as Classic Bluetooth. It is created and promoted by the Bluetooth Special Interest Group. (Ray, 2015)

Classic Bluetooth serves the purpose of point-to-point connection between smart devices for file transferring or audio streaming. Unlike from it, BLE is aimed at IoT applications with wireless sensors, controllers and other cybernetic objects. Cybernetic objects are usually dealing with short messages and they have to operate autonomously on one battery life as long as possible. Therefore, reduced power consumption is more important for CPS solutions than transferring big packages of data. (Ray, 2015)

Table 2 provides a wide comparison of BLE and classic Bluetooth characteristics:

Table 2. Detailed comparison of Classic Bluetooth and BLE characteristics (Bluetooth vs BLE, n.d.)

Specifications	Bluetooth	BLE
Network/Topology and Nodes/Active Slaves	Scatternet (7 devices)	Point-to-Point (unlimited), Broadcast (one-to-many), Mesh (many-to-many)
Peak current consumption	30 mA	15 mA in transfer mode; 1 µA in sleep mode
Speed	700 Kbps	1 Mbps
Message size	max 358 bytes	8 … 47 bytes
Application throughput	0.7 to 2.1 Mbps	less than 0.3 Mbps
Latency in data transfer between two devices	Approx. 100 ms	Approx. 3 ms
Range	<100 m	100 m, 150 m in open field
RF Frequency band	2 400 GHz	2 400 GHz
Frequency Channels	79 channels from 2 400 GHz to 2 483,5 GHz with 1 MHz spacing	40 channels (includes 3 advertising and 37 data channels) from 2 402 GHz to 2 480 GHz with 2 MHz spacing

In a contrast to the connection oriented Classic Bluetooth, to reach its aims, protocols of BLE are designed for burst connection. They keep the component in a sleep mode before establishing the connection. (Low Energy: Point-to-Point, n.d.)

With a high speed transferring and a small size of data packages, BLE connection usually takes about 3ms, when Classic Bluetooth keeps the connection for about 100ms.

Even though, BLE is incompatible with Bluetooth devices of previous generations, both BLE and Classic Bluetooth are operating in the same 2.4 GHz band, which makes it possible to connect to them through the same receiver on a smart device.

All popular mobile and computer operating systems natively support BLE connection. The Bluetooth Special Interest Group believes that majority of latest smart devices will also support BLE protocols.

4.3.2 BLE profile

BLE protocols introduce the Generic Attribute Profile (GATT), which is built on top of the generic Attribute Protocol (ATT). As it shown in Figure 6, it is a hierarchy of identifiers, which are used to describe the way that BLE devices transfer standard messages.

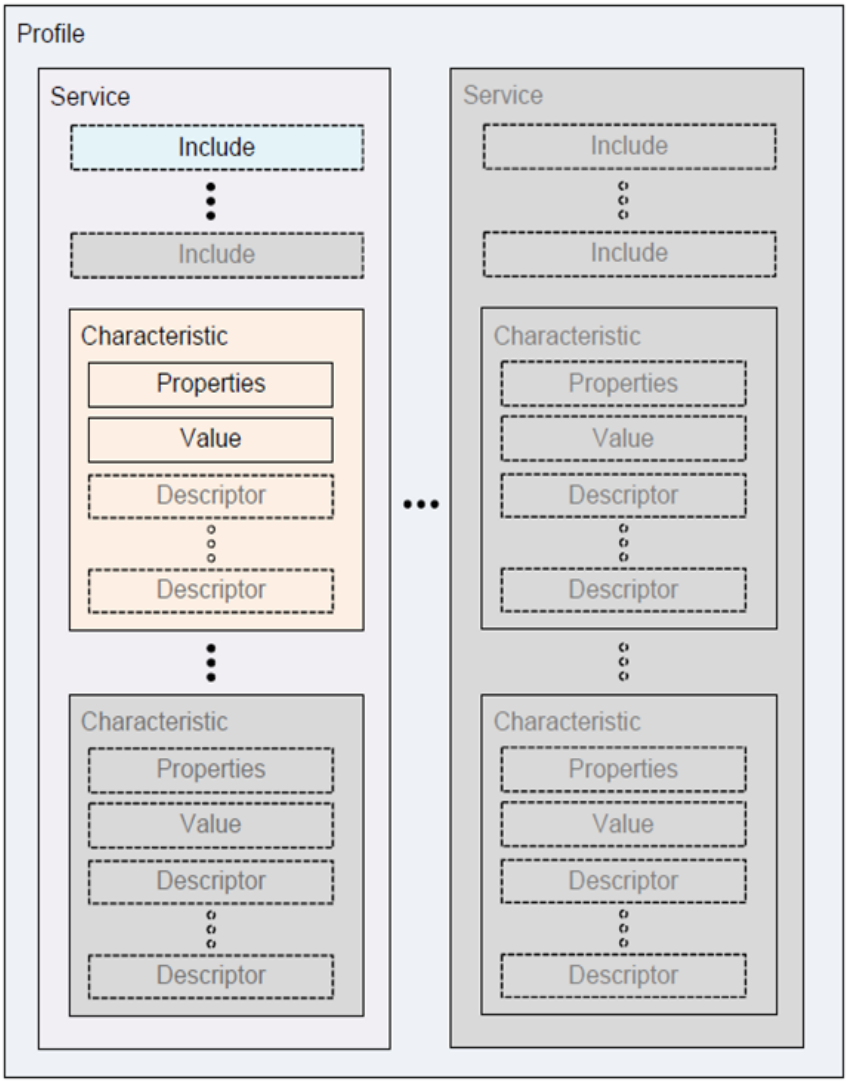

Figure 6. GATT profile hierarchy (GATT overview, n.d.)

The hierarchy of identifiers from the lowest to the highest attribute is listed as follows:

- Characteristic is a defined attribute type that contain a single logical value;
- Descriptor is a defined attribute that describe a characteristic value;
- Service is a collection of characteristics and relationships to other services that encapsulate the behavior of part of a device;
- Profile is a collection of all services that encapsulate the behavior of a whole device.

(GATT overview, n.d.)

In addition, BLE uses terms of a client and a server to define devices that initiate and receive GATT commands respectively.

4.3.3 Reverse engineering the smart lamp

Establishing a communication between an application and a BLE device requires utilization of proper BLE identifiers. Considering the lack of documentation for the smart lamp, reverse engineering may be a solution here.

In this case reverse engineering involves discovering Bluetooth protocols of the original application by reading logs collected by a phone.

Tools used for this task are as follows:

- “nRF Connect” Android app by Nordic;
- “Developer Options” setting on Android smartphone;
- “Wireshark” software for PC.

First, nRF Connect should be used to discover the address of the Smart lamp and to explore its possible identifiers as shown in Figure 7.

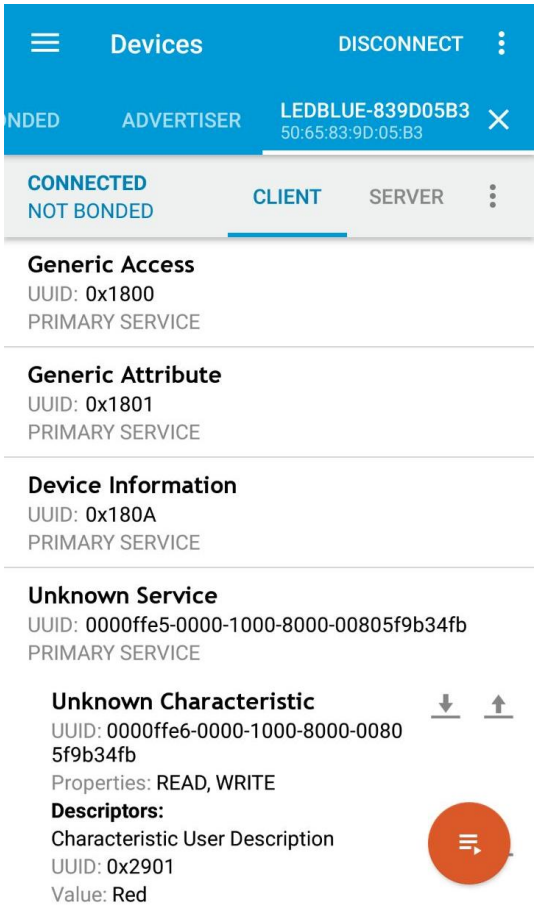

Figure 7. Identifiers of the Smart lamp discovered by the nRF Connect

Yet, bare identifiers are not providing sufficient information to establish proper communication with the BLE device. Communication also depends on proper messages that are sent to exact properties.

The correlation between messages and properties of the BLE device can be discovered through the Bluetooth HCI snoop log. It is a part of the standard Android developer toolset, which is available on any Android smartphone. An option “Bluetooth HCI snoop log” can be activated simply from the smartphone’s settings like in Figure 8. However, this option is hidden, unless the developer mode is enabled on the phone.

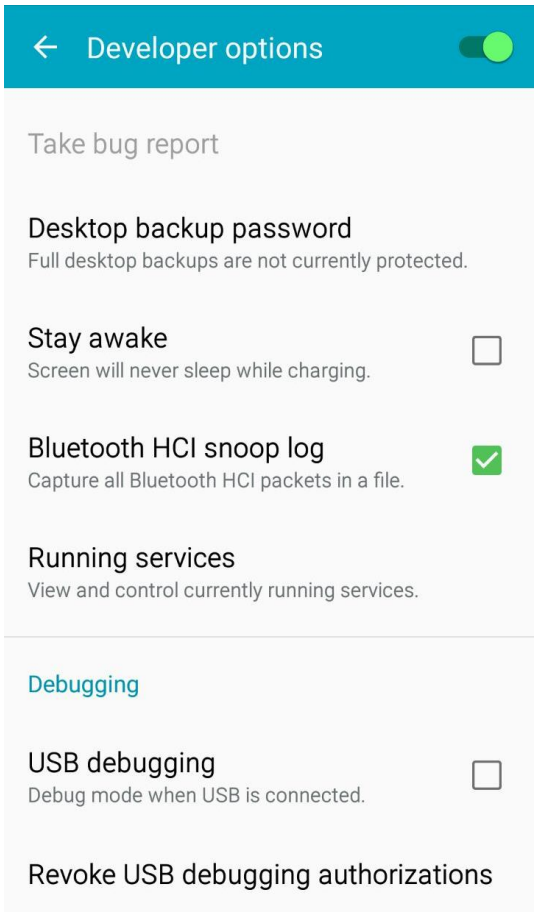

Figure 8. Bluetooth HCI snoop log option in developer options on a typical Android smartphone.

After an option “Bluetooth HCI snoop log” was enabled, it would start collecting data sent and received by the smartphone, including data sent by the original application to the smart lamp. After sending several commands to the smart lamp, the Bluetooth HCI snoop log will contain all the necessary information on Bluetooth activity during this time.

The log file is stored in the root folder of the smartphone’s file system under the name “btsnoop_hci.log”. The file contains flood packets of data, so necessary information can be easily lost between irrelevant packets.

The Wireshark software for Windows may assist in discovering necessary information by filtering and reading through the logs as shown in Figure 9.

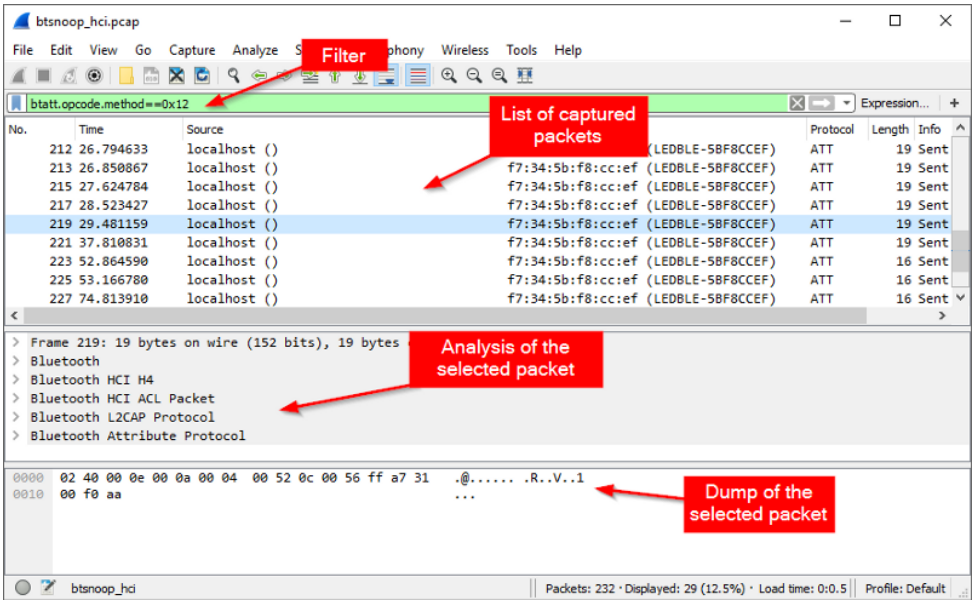

Figure 9. Interface of Wireshark software and convenient search with filters of necessary information through captured packets

After analyzing packets, necessary identifiers were discovered. It was the service UUID “0000ffe5-0000-1000-8000-00805f9b34fb” and the characteristic UUID “0000FFE9-0000-1000-8000-00805F9B34FB”. The standard message to the characteristic contained 7 bytes and looked as follows:

RGB color should be encoded as “56-**RR**-**GG**-**BB**-00-f0-aa”, where:

1 byte - “56” never changes its value;
2 byte - RR – the byte is responsible for the red color. It can take values form 0 to 255;
3 byte - GG – the byte is responsible for the green color. It can take values form 0 to 255;
4 byte - BB – the byte is responsible for the blue color. It can take values form 0 to 255;
5 byte - “00” – the byte is responsible for the white color. It has to keep “00” value in RGB mode; otherwise, the message would not be accepted. It can take values form 0 to 255;
6 byte - “f0” – the byte is responsible for the current mode. It takes values either “f0” for the RGB mode, or “0f” for the White mode;
7 byte - “aa” never changes its value.

In white mode, the message looks accordingly: “56-00-00-00-**WW**-0f-aa”. Bytes “RR”, “GG” and “BB” have to be “00” in the White mode. Otherwise, the message would not be accepted.

4.4 **Software application**

Collected data was enough to write own simple application for the smart lamp from Magic Blue. Usually, software development is a complicated process, which requires programming on Java and sometimes C programming languages and compiling all work files into one application.

By chance, complicated process of software development can be automated by third-party applications like PhoneGap.

PhoneGap is the developer application, which uses Cordova Node.js modules to translate a front-end solution on programming language JavaScript into the final application. It simultaneously compiles applications for smartphones on iOS, Android and Windows Phone OS.

PhoneGap accepts solutions to compile from the version control web page GitHub. It is a web-portal created to storage different versions of application and committing stable changes to the code.

4.4.1 Interface of custom application

Custom application can establish the connection with smart lamps from Magic Blue. It can send commands to the lamp to change colors or brightness of the lamp. Custom application has four work layers, which are shown in Figure 10.

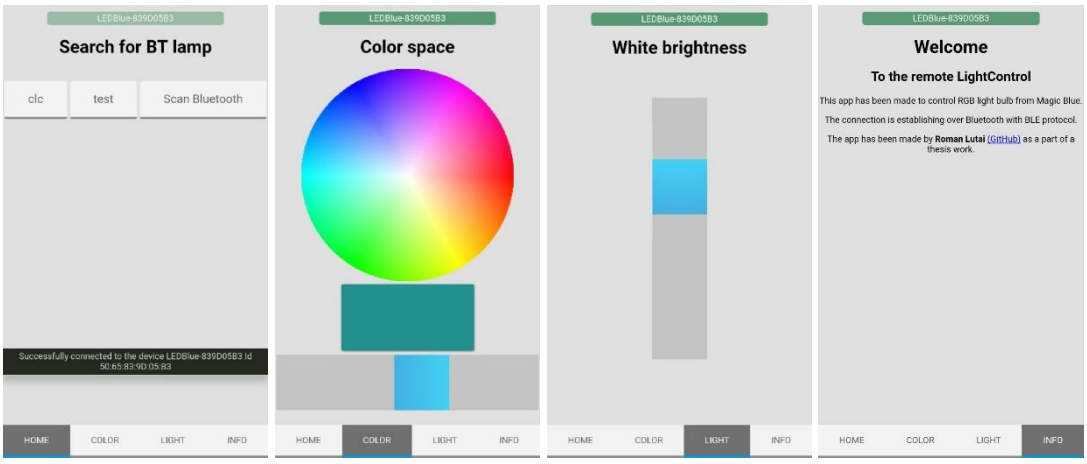

Figure 10. Interface of the custom application

Each layer is responsible for the following:

1. “Home” layer is responsible for monitoring Bluetooth band for smart lamps and establishing connection with the chosen lamp.
2. “Color” layer contains controls to change colors of the smart lamp.
3. “Light” layer contains the input control to change the brightness of the smart lamp in the “White” mode.

4. “Info” layer contains short summary of the app.

Even though custom application looks simple, it contains sophisticated logic in code.

4.4.2 Introduction to source code files

Thanks to PhoneGap software, the project can be fully written as a web-application, or a web page. Therefore, the logic of the app is written in JavaScript with the use of modules from Cordova.js and can be accessed on GitHub (Light-control, n.d.). The code is stored in three files:

1. “index.js” is responsible for general logic and actions in the app;
2. “bt.js” is responsible for the communication with BLE Smart lamp;
3. “color-space.js” is responsible for the proper color changing actions in app and on the Smart lamp.

The content of the app, like order of buttons, text and control units, is described by HTML code. This code is stored in the file “index.html”.

Appearance of the content, like size, position, color and reaction on standard actions, is described by Cascading Style Sheets (CSS). Styles are stored in three files:

1. “index.css” is describing appearance of general elements, as so wrappers, blocks, buttons and inputs;
2. “input-range.css” is describing the appearance of RGB control elements;
3. “footer-nav.css” is describing the appearance of the navigation menu of the app.

All code files can be found in Appendixes 1-7. Nevertheless, the logic of lamp control is worth to be described more detailed.

4.4.3 Lamp control logic

The lamp control logic can be divided in two big chapters: maintaining Bluetooth Low Energy connection and manipulating with the color to send proper message.

Bluetooth functions and objects are stored in the file “bt.js”, which can be found in Appendix 4. The file stores three main objects:

1. Object “ListHTML” (lines 0…13) is responsible for proper mapping discovered Bluetooth devices, dealing with the duplicates and refreshing the list of discovered devices in the contents of app;

2. Object “bt” (lines 42..101) is based on “cordova-ble-central” module and it is responsible for discovering new BLE devices and connecting to them;
3. Object “activeBLE” (lines 104…174) is also based on “cordova-ble-central” module and it is responsible for proper messaging with the Smart lamp after established connection.

Color control functions and objects are stored in the file “color-space.js”, which can be found in Appendix 5. The file stores such objects and functions as:

1. Object “ColorPicker” (lines 0…89) is responsible for rendering the color wheel of all RGB colors with the help of HTML5 element “canvas”;
2. Functions “rgbToHSV”, “HSVtoRGB” and “rgbToHEX” (lines 95…118, 124…147 and 149…153 respectively) are responsible for transforming from and to appropriate color code standard: RGB (Red, Green, Blue), HSV (Hue, Saturation, Value) and HEX (hexadecimal coding);
3. Object “activeColor” (lines 156…204) is responsible for manipulating with the color sent to lamp according to the actions with the color control elements in the app;
4. Functions “mouseRGB” and “touchRGB” (lines 208…223 and 225…230 respectively) using functions “getEventLocation” and “getElementPosition” (lines 232…239 and 241…251 respectively);
5. Function “getColorPicker” (lines 255…263) starts rendering color controls on the display with the object “ColorPicker” when the layer “Color” was called for the first time;

These objects are representing main logic of the custom application for light control of smart lamps from Magic Blue. It would not be possible to create without discovering the communication standards of the smart lamp. This lamp is an object of the cyber-physical system and it was made for use at home. The utilization of CPS objects in our homes indicates the upcoming Industry 4.0.

5 SUMMARY

Aims of this thesis project were as follows:

- To examine the evolution of industry from the first industrial revolution to our days;
- To prove, based on research, the importance of the Internet of Things (IoT) and Cyber Physical Systems (CPS) for further evolution of industry;
- To study the role of reverse engineering for CPS;
- To apply reverse engineering on an instance of CPS.

The aims of the thesis project were successfully achieved and led to the following conclusions:

Industry has been evolving during the history of humankind. Every revolutionary discovery in industry was affecting people's routines as well. Four biggest industry revolutions have taken the time during the last several centuries. In the time stamp from the 18th to the 21st century the industry has emerged from the first industrial mechanical machine on steam power to automated lines of mass-production and smart factories with the utilization of IoT and CPS and it still keeps emerging.

The current industry trend is cybernation. It is a fusion of physical, digital and biological worlds with the utilization of CPS, IoT, Big Data and many other currently emerging technologies. Cybernation is just taking off from factory applications to change our routine life.

The main distinct feature of CPS objects is the ability to communicate through the different networks with each other and computing devices. However, often CPS objects are integrated to their closed application, so they are not applicable to new solutions. To deal with this problem, the communication methods of the object have to be examined. This and many other types of gathering information about the product are called reverse engineering. Reverse engineering is just a tool with the bad reputation of intellectual rights violator. However, it has also positive applications, like security examination, or designing interoperable products.

The availability of interoperable solutions is important for the product, as this creates a wide environment for the product's applications. The process of creating an interoperable solution is presented in the practical CPS application chapter. Therefore, the Smart lamp from Magic Blue was chosen to apply reverse engineering methods to discover product's communication methods and use them to create own application for the same product.

As a result, the Light control application for smartphones had been made. It contains the basic features. It can connect to the smart lamp and control its color. Despite its simplicity, the main advantage of the custom app that its code is available in the internet for further development.

REFERENCES

Bluetooth vs BLE-difference between Bluetooth and BLE (n.d.). Retrieved April 2, 2018, from http://www.rfwireless-world.com/Terminology/Bluetooth-vs-BLE.html

Bryant, M. (2011, August 6). *20 years ago today, the World Wide Web was born - TNW Insider*. Retrieved May 17, 2018, from http://thenextweb.com/insider/2011/08/06/20-years-ago-today-the-world-wide-web-opened-to-the-public/

Constable, G. Somerville, B. (2003). *Electrification History 1 - Early Years*. Retrieved April 29, 2018, from http://www.greatachievements.org/?id=2988

Eilam, E. & Chikofsky, E. (2005). *Reversing: secrets of reverse engineering. Indianapolis*, IN: Wiley.

Engelman, R. (2015, April 10). *The Second Industrial Revolution*, 1870-1914. Retrieved March 22, 2018, from http://ushistoryscene.com/article/second-industrial-revolution/

Ford, H. (2008). *My life and work: an autobiography of Henry Ford*. Place of publication not identified: BN Publishing.

Freeman, C. & Louçã, F. (2002). *As Time Goes By: From the Industrial Revolutions to the Information Revolution*. New York, NY: Oxford University Press.

GATT Overview (n.d.). Retrieved April 2, 2018, from https://www.bluetooth.com/specifications/gatt/generic-attributes-overview

Industrial revolution (2009). Retrieved March 21, 2018, from https://www.history.com/topics/industrial-revolution

Landes, D. (1969). *The unbound Prometheus: technological change and industrial development in Western Europe from 1750 to the present*. London: Cambridge U.P.

LED Magic Blue (n.d.). Retrieved March 29, 2018, from https://play.google.com/store/apps/details?id=com.Zengge.BluetoothLigthDark&hl=ru

Light-control (n.d.). Retrieved April 14, 2018, from https://github.com/romanlutai/light-control

Low Energy: Point-to-Point (n.d.). Retrieved April 2, 2018, from https://www.bluetooth.com/bluetooth-technology/le-p2p

Lucas, R. (2002). *Lectures on economic growth*. Cambridge, Mass: Harvard University Press.

Magic blue LED bulb (n.d.). Retrieved March 28, 2018, from https://www.alibaba.com/product-detail/4-5w-white-Magic-blue-LED_60461038866.html?spm=a2700.7724857.main07.216.49665d5cb2xA3i

Marr, B. (2016, April 5). *Why Everyone Must Get Ready For The 4th Industrial Revolution*. Retrieved March 26, 2018, from https://www.forbes.com/sites/bernardmarr/2016/04/05/why-everyone-must-get-ready-for-4th-industrial-revolution/#5af877b33f90

Muntone, S. (2012, February 4). *Technological Revolutions*. Retrieved April 29, 2018, from https://web.archive.org/web/20131022224325/http://www.education.com/study-help/article/us-history-glided-age-technological-revolution/

Perez, C. (2011, August 13). *Structural change and assimilation of new technologies in the economic and social systems*. Retrieved May 17, 2018, from http://www.carlotaperez.org/papers/scass_v04.pdf

Ray, B. (2015, November 1). *Bluetooth Vs. Bluetooth Low Energy: What's The Difference?* Retrieved April 2, 2018, from https://www.link-labs.com/blog/bluetooth-vs-bluetooth-low-energy

Roser, C. (2015, November 24). *Illustration of Industry 4.0, showing the four industrial revolutions with a brief English description*. [Online image]. Retrieved March 26, 2018, from https://www.allaboutlean.com/industry-4-0-potentials/

Schoenherr, S. (2004, May 5). *The Digital Revolution*. Retrieved March 26, 2018, from http://history.sandiego.edu/gen/recording/digital.html

Schwab, K. (2017, January 3). *The Fourth Industrial Revolution*. New York, NY: Crown Business.

Second Industrial Revolution: The Technological Revolution (2017, January 15). Retrieved March 22, 2018, from http://richmondvale.org/second-industrial-revolution/

Shaked, U. (2016, August 3). *Reverse Engineering a Bluetooth Lightbulb*. Retrieved March 27, 2018, from https://medium.com/@urish/reverse-engineering-a-bluetooth-lightbulb-56580fcb7546

Temin, P (1966, June). *Steam and Waterpower in the Early Nineteenth Century*. Journal of Economic History. London: Cambridge U.P.

Thayer, K. (2017, December 8). *How Does Reverse Engineering Work?* Retrieved March 27, 2018, from https://insights.globalspec.com/article/7367/how-does-reverse-engineering-work

Zhong, R. Y., Xu, X., Klotz, E., & Newman, T. S. (2017). *Intelligent Manufacturing in the Context of Industry 4.0*. Retrieved May 03, 2018, from https://www.sciencedirect.com/science/article/pii/S2095809917307130

Appendix 1

Content of index.html

Below is showed the code from a file “index.html” which is responsible for the content in the app:

```
0  <html>
1
2  <head>
3      <meta charset="utf-8" />
4      <meta name="format-detection" content="telephone=no" />
5      <meta name="msapplication-tap-highlight" content="no" />
6      <meta name="viewport" content="user-scalable=no, initial-scale=1,
       maximum-scale=1, minimum-scale=1, width=device-width" />
7
8      <link rel="stylesheet" type="text/css" href="css/footer-nav.css" />
9      <link rel="stylesheet" type="text/css" href="css/input-range.css" />
10     <link rel="stylesheet" type="text/css" href="css/index.css" />
11     <title>Light Control</title>
12 </head>
13
14 <body>
15     <div id="deviceready" class="blink">
16         <p class="event listening">Connecting to Device</p>
17         <p class="event received" id="green">Choose the Bluetooth</p>
18     </div>
19
20     <div class="app show" id="layerHome">
21         <h1>Search for BT lamp</h1>
22
23         </br>
24         <div style="display: flex">
25             <button style="flex-grow: 1" onclick="semiLog.clc()">clc</button>
26             <button style="flex-grow: 1" onclick="test()">test</button>
27             <button style="flex-grow: 2" onclick="bt.check()">Scan
       Bluetooth</button>
28         </div>
29
30         <div class="comments bright">
31             <ul class="hide" id="bt_list">
32             </ul>
33         </div>
34
35         <div id="status"></div>
36     </div>
37
```

```
38    <div class="app" id="layerRGB">
39        <h1>Color space</h1>
40        <div id="colorSpace"></div>
41        <div id="currentRGB"></div>
42        <div class="slidecontainer">
43            <input type="range" min="0" max="1" step="0.01" class="range"
id="slider_HSV_v">
44        </div>
45    </div>
46
47    <div class="app" id="layerWhite">
48        <h1>White brightness</h1>
49        <div class="slidecontainer vertical">
50            <input type="range" min="0" max="100" step="1" class="range vertical-
highest-first" id="slider_White">
51        </div>
52    </div>
53
54    <div class="app" id="layerInfo">
55        <h1>Welcome</h1>
56        <h2>To the remote LightControl</h2>
57        <p> This app has been made to control RGB light bulb from Magic
Blue.</p>
58        <p> The connection is establishing over Bluetooth with BLE protocol.</p>
59        <p> The app has been made by <strong>Roman Lutai</strong> <a
href="https://github.com/romanlutai">(GitHub)</a> as a part of a thesis
work.</p>
60        <!-- <div class="loader"></div> -->
61    </div>
62
63    <div class="comments dark" id="semiLog"><div class="logline"> Log is clear
</div></div>
64
65    <ul class="footer-nav">
66      <li><a class="active"
onclick="showlayer(this,'layerHome')">HOME</a></li>
67      <li><a
onclick="showlayer(this,'layerRGB',true);getColorPicker();hideLoader()">COLO
R</a></li>
68      <li><a
onclick="showlayer(this,'layerWhite');defineWhiteRange('slider_White')">LIGH
T</a></li>
69      <li><a onclick="showlayer(this,'layerInfo')">INFO</a></li>
70    </ul>
71
72    <div id="loaderWrapper">
73        <div class="loader-window">
```

```
74              <p> Loading. Please, wait. </p>
75              <div class="loader"></div>
76          </div>
77      </div>
78
79
80      <script type="text/javascript" src="cordova.js"></script>
81      <script type="text/javascript" src="js/index.js"></script>
82      <script type="text/javascript" src="js/bt.js"></script>
83      <script type="text/javascript" src="js/menu.js"></script>
84      <script type="text/javascript" src="js/color-space.js"></script>
85      <script type="text/javascript">
86          app.initialize();
87      </script>
88  </body>
89
90  </html>
```

Appendix 2

Content of “index.css”

Below is showed the code from a file “index.css” which is responsible for the appearance of the content in the app:

```
0  * {
1      -webkit-tap-highlight-color: rgba(0,0,0,0); /* make transparent link selection, adjust last value opacity 0 to 1.0 */
2  }
3
4  body {
5      -webkit-touch-callout: none;                /* prevent callout to copy image, etc when tap to hold */
6      -webkit-text-size-adjust: none;             /* prevent webkit from resizing text to fit */
7      -webkit-user-select: none;                  /* prevent copy paste, to allow, change 'none' to 'text' */
8      background-color:#E0E0E0;
9      background-attachment:fixed;
10     font-family:'HelveticaNeue-Light', 'HelveticaNeue', Helvetica, Arial, sans-serif;
11     font-size:12px;
12     height:80%;
13     margin:0px;
14     padding:0px;
15     /*text-transform:uppercase;*/
16     width:100%;
17     margin: auto;
18     text-align:center;
19 }
20
21 button {
22     position: relative;
23     background-color: #f3f3f3;
24     border: none;
25     color: #666;
26     margin: 2px;
27     padding: 15px 32px;
28     text-align: center;
29     text-decoration: none;
30     display: inline-block;
31     font-size: 16px;
32     cursor: pointer;
33     overflow: hidden;
34     float: left;
```

```
35      border-radius: 3px;
36      border-bottom: 3px solid #8e8e8e;
37      -webkit-transition-duration: 0.4s; /* Safari */
38      transition-duration: 0.4s;
39  }
40  button:hover {
41      color: white;
42      background-color: #707070;
43      border-bottom: 3px solid #039be5;
44  }
45  button:after {
46      content: "";
47      background: #f1f1f1;
48      display: block;
49      position: absolute;
50      padding-top: 300%;
51      padding-left: 350%;
52      margin-left: -32px !important;
53      margin-top: -120%;
54      opacity: 0;
55      transition: all 0.8s
56  }
57  button:active:after {
58      padding: 0;
59      margin: 0;
60      opacity: 1;
61      transition: 0s
62  }
63
64  .app {
65      width:100%;
66      min-height: 85% ;
67      display:none;
68  }
69  .show {
70      display:block;
71  }
72
73  .event {
74      border-radius:4px;
75      -webkit-border-radius:4px;
76      color:#FFFFFF;
77      font-size:12px;
78      margin:10px 60px;
79      padding:2px 0px;
80  }
81
```

```
.event.listening {
    background-color:#333333;
    display:block;
}

.event.received {
    background-color:#4B946A;
    display:none;
}

@keyframes fade {
    from { opacity: 1.0; }
    50% { opacity: 0.4; }
    to { opacity: 1.0; }
}

@-webkit-keyframes fade {
    from { opacity: 1.0; }
    50% { opacity: 0.4; }
    to { opacity: 1.0; }
}

.blink {
    animation:fade 3000ms infinite;
    -webkit-animation:fade 3000ms infinite;
}

.comments {
    min-width: 160px;
    max-height: 20%;
    box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);
    margin: 4px 0;
    padding: 0;
    overflow-y: scroll;
}
.comments::-webkit-scrollbar {
  display: none;
}
.dark {
    background-color: #272822;
    color: #ccc ;
}
.bright {
    background-color: #f9f9f9;
    color: black ;
}
.logline {
```

```
    border-bottom:1px solid #ccc;
    padding: 4px 2px;
}

#bt_list {
    list-style:none;
    padding:0;
}
#bt_list li {
    border-bottom:1px solid #ccc;
    padding: 4px 2px;
}
.hide {
    display:none;
}

#colorSpace {
    margin: 4px auto;
    width: 80%;
    left: 10%;
}
#currentRGB {
    display: block;
    background: black;
    margin: 4px auto;
    width: 50%;
    padding-top: 25%;
    box-shadow: 1px 1px 2px rgba(0,0,0,.5);
    border-radius: 2px;
}

#loaderWrapper {
    display: none;
    position: fixed;
    width: 100%;
    height: 100%;
    padding: 25%;
    background-color: rgba(0,0,0,.5);
}

.loader-window {
    background-color: white;
    border-radius: 2px;
}

.loader {
  border: 8px solid #f3f3f3;
```

```
176     border-radius: 50%;
177     border-top: 8px solid #3498db;
178     width: 15px;
179     height: 15px;
180     -webkit-animation: spin 2s linear infinite; /* Safari */
181     animation: spin 2s linear infinite;
182   }
183
184   /* Safari */
185   @-webkit-keyframes spin {
186     0% { -webkit-transform: rotate(0deg); }
187     100% { -webkit-transform: rotate(360deg); }
188   }
189
190   @keyframes spin {
191     0% { transform: rotate(0deg); }
192     100% { transform: rotate(360deg); }
193   }
```

Appendix 3

Content of “index.js”

Below is showed the code from a file “index.js” which is responsible for the logic and actions in the app:

```
var app = {
  initialize: function() {
    this.bindEvents();
  },
  bindEvents: function() {
    document.addEventListener('deviceready', this.onDeviceReady, false);
  },
  onDeviceReady: function() {
    app.receivedEvent('deviceready');
  },
  receivedEvent: function(id) {
    var parentElement = document.getElementById(id);
    var listeningElement = parentElement.querySelector('.listening');
    var receivedElement = parentElement.querySelector('.received');

    listeningElement.setAttribute('style', 'display:none;');
    receivedElement.setAttribute('style', 'display:block;');

    console.log('Received Event: ' + id);
  }
};

var showlayer = (el,id,loader=false) => {
  document.getElementsByClassName("show")[0].classList.remove("show");
  document.getElementsByClassName("footer-nav")[0].getElementsByClassName("active")[0].classList.remove("active");
  document.getElementById(id).classList.add("show");
  el.classList.add("active");
  if (loader) {
    document.getElementById("loaderWrapper").classList.add("show");
  }
};

var hideLoader = () => {
  // document.getElementById("loaderWrapper").classList.remove("show");
}

var defineWhiteRange = (id) => { if (!this.used) {
    element = document.getElementById(id);
    element.addEventListener( "input", function(){
```

```
39        activeBLE.sendRGBW(0,0,0,element.value);
40      });
41      this.used = true;
42    }
43  };
44
45
46
47  var semiLog = {
48    logWindow: document.getElementById('semiLog'),
49    log: function(message) {
50      console.log(message);
51      this.logWindow.insertAdjacentHTML('beforeend','<div
    class="logline">'+message+'</div>');
52      this.logWindow.scrollTop = this.logWindow.scrollHeight;
53    },
54    clc: function() {
55      this.logWindow.innerHTML = '';
56    }
57  };
```

Appendix 4

Content of “bt.js”

Below is showed the code from a file “bt.js” which is responsible for the actions related to Bluetooth in the app:

```
0  function ListHTML (id) {
1    this.element = document.getElementById(id);
2    this.list = {"50:65:83:9D:05:B3":"MyLamp"};
3
4    this.addBT = (devices) => {
5      var self = this;
6      if (devices instanceof Array) {
7        devices.forEach(function(device){
8          self.list[String(device.id)] = device.name;
9        });
10     } else {
11       self.list[String(devices.id)] = devices.name;
12     }
13   };
14
15   this.update = () => {
16     var self = this;
17     this.element.innerHTML = "";
18     for (id in this.list){
19       console.log(id);
20       var name = this.list[id];
21       self.element.insertAdjacentHTML('beforeend',`<li
   onclick="bt.pair('${name}','${id}')">${name}</li>`);
22     };
23   };
24
25   this.hide = () => {
26     if (!this.element.classList.contains('hide')) {
27       this.element.classList.add('hide');
28     }
29   };
30
31   this.show = () => {
32     if (this.element.classList.contains('hide')) {
33       this.element.classList.remove('hide');
34     }
35   };
36 }
37
38 var btListHTML = new ListHTML('bt_list');
39
```

```
40  var greenStatus = document.getElementById('green');
41
42  var bt = {
43    check: function() {
44      btListHTML.show();
45      semiLog.log('start');
46      var self = this;
47      ble.isEnabled(
48        function() {
49          console.log("Bluetooth already ON");
50          semiLog.log("Bluetooth is enabled");
51          self.scan();
52        },
53        function() {
54          console.log("Bluetooth is OFF");
55          semiLog.log("Please, enable Bluetooth.");
56          if(device.platform == 'Android') self.activate();
57        }
58      );
59    },
60    activate: function() {  // For Android Only
61      var self = this;
62      semiLog.log("Android OS");
63      ble.enable(
64        function() {
65          console.log("You have enabled Bluetooth");
66          semiLog.log("You have enabled Bluetooth");
67          self.scan();
68        },
69        function() {
70          console.log("The user did *not* enable Bluetooth");
71          semiLog.log("You have not enabled Bluetooth. App cannot continue
    without it.");
72        }
73      );
74    },
75    scan: function() { // For Android Only
76      semiLog.log('Scanning for BLE devices');
77      btListHTML.update();
78      ble.startScanWithOptions([],{}, function (device) {
79        semiLog.log(device.id);
80        btListHTML.addBT(device);
81        btListHTML.update();
82      }, function(failure){semiLog.log(failure);} );
83    },
84    pair: function(deviceName,deviceId) {
85      ble.stopScan(
```

```
86      function(success){semiLog.log("Scan complete")},
87      function(failure){semiLog.log("stopScan failed: "+failure)}
88    );
89    semiLog.log(`Connecting to device ${deviceName}`);
90    activeBLE.connected(deviceName,deviceId);
91    ble.connect(deviceId, function(success){
92      btListHTML.hide();
93      semiLog.clc();
94      semiLog.log(`Successfully connected to the device ${deviceName} Id
      ${deviceId}`);
95      greenStatus.innerHTML = deviceName;
96      activeBLE.connected(deviceName,deviceId);
97    }, function (failure) {
98      semiLog.log(`Connection failed: ${failure}`);
99    });
100   }
101 }
102
103
104 var activeBLE = {
105   connected: function(name, id){
106     this.name = name;
107     this.id = id;
108     this.active = true;
109   },
110
111   active: false,
112   write_service_UUID: "0000ffe5-0000-1000-8000-00805f9b34fb",
113   write_charachteristic_UUID: "0000FFE9-0000-1000-8000-00805F9B34FB",
114   readUUID : "0000FFE4-0000-1000-8000-00805F9B34FB",
115   infoUUID : "2A00",
116
117   jono: 0,
118
119   sendRGBW: function(r,g,b,w = 0){
120     if ( this.jono < 3 ){
121       this.jono++;
122       var self = this;
123       ble.write(
124         this.id,
125         this.write_service_UUID,
126         this.write_charachteristic_UUID,
127         this.messageRGBW(r,g,b,w),
128         function(success){
129           self.jono--;
130         },
131         function(failure){
```

```
132           semiLog.log("Sending data failed: "+failure);
133         }
134       );
135     }
136   },
137   messageRGBW:function(r,g,b,w){
138     if (r > 255 || g > 255 || b > 255)
139       throw "Invalid color component";
140     var BYTEf = 0xF0;
141     if (w!=0) BYTEf = 0x0F;
142     return this.encodeBytes([0x56,r,g,b,w,BYTEf,0xAA]);
143   },
144
145   sendSwitch: function(isOff){
146     ble.write(
147       this.id,
148       this.write_service_UUID,
149       this.write_charachteristic_UUID,
150       this.messageSwitch(isOff),
151       function(success){},
152       function(failure){
153         alert("Sending data failed: "+failure);
154       }
155     );
156   },
157   messageSwitch: function(isOff){
158     return this.encodeBytes([0xCC,0x23+off,0x33]);
159   },
160
161   encodeBytes:function(arr){
162     var message = new Uint8Array(arr.length);
163     for (var i=0 ; i<arr.length ; i++){
164       message[i] = arr[i];
165     }
166     return message.buffer;
167   }
168 }
169
170 function test() {
171   var t = new Date();
172   semiLog.log( "Time in ms: "+t.getTime() );
173   ble.disconnect(activeBLE.id,function(success){semiLog.log('Successfully
    disconnect')},function(failure){semiLog.log(failure)});
174 }
```

Appendix 5

Content of “color-space.js”

Below is showed the code from a file “color-space.js” which is responsible for the actions related to color changing of a lamp in the app:

```
0  function ColorPicker(element) {
1    this.element = element;
2
3    this.init = function() {
4      var diameter = this.element.offsetWidth;
5
6      var canvas = document.createElement('canvas');
7      canvas.height = diameter;
8      canvas.width = diameter,
9      this.canvas = canvas;
10
11     this.renderColorMap();
12
13     element.appendChild(canvas);
14
15     this.setupBindings();
16   };
17
18   this.renderColorMap = function() {
19     var canvas = this.canvas;
20     var ctx = canvas.getContext('2d');
21
22     var radius = canvas.width / 2;
23     var toRad = (2 * Math.PI) / 360;
24     var step = 1 / radius;
25
26     ctx.clearRect(0, 0, canvas.width, canvas.height);
27
28     var cx = cy = radius;
29     for(var i = 0; i < 360; i += step) {
30       var rad = i * toRad;
31       var x = radius * Math.cos(rad),
32         y = radius * Math.sin(rad);
33
34       ctx.strokeStyle = 'hsl(' + i + ', 100%, 50%)';
35
36       ctx.beginPath();
37       ctx.moveTo(radius, radius);
38       ctx.lineTo(cx + x, cy + y);
39       ctx.stroke();
```

```
40      }
41
42      // draw saturation gradient
43      var grd = ctx.createRadialGradient(cx,cy,0,cx,cx,radius);
44      grd.addColorStop(0,"white");
45      grd.addColorStop(1,'rgba(255, 255, 255, 0)');
46      ctx.fillStyle = grd;
47      ctx.beginPath();
48      ctx.arc(cx, cy, radius, 0, Math.PI * 2, true);
49      ctx.closePath();
50      ctx.fill();
51
52      // render the rainbow box here
53    };
54
55    this.renderMouseCircle = function(x, y) {
56      var canvas = this.canvas;
57      var ctx = canvas.getContext('2d');
58
59      ctx.strokeStyle = 'rgb(255, 255, 255)';
60      ctx.fillStyle = 'rgba(0, 0, 0, 0.5)'
61      ctx.lineWidth = '3';
62      ctx.beginPath();
63      ctx.arc(x, y, 10, 0, Math.PI * 2, true);
64      ctx.closePath();
65      ctx.fill();
66      ctx.stroke();
67    };
68
69    this.setupBindings = function() {
70      var canvas = this.canvas;
71      var ctx = canvas.getContext('2d');
72      var self = this;
73    };
74
75    this.plotRgb = function(r, g, b) {
76      var canvas = this.canvas;
77      var ctx = canvas.getContext('2d');
78
79      var [h, s, v] = rgbToHsv(r, g, b);
80      var theta = h * 2 * Math.PI;
81      var maxRadius = canvas.width / 2;
82      var r = s * maxRadius;
83      var x = r * Math.cos(theta) + maxRadius,
84        y = r * Math.sin(theta) + maxRadius;
85      this.renderMouseCircle(x, y);
86    }
```

```
87
88    this.init();
89  }
90  /* accepts parameters
91   * r  Object = {r:x, g:y, b:z}
92   * OR
93   * r, g, b
94   */
95  function rgbToHsv(r, g, b){
96    if (arguments.length === 1) {
97      b = r.b, g = r.g, r = r.r;
98    }
99    r = r/255, g = g/255, b = b/255;
100   var max = Math.max(r, g, b), min = Math.min(r, g, b);
101   var h, s, v = max;
102
103   var d = max - min;
104   s = max == 0 ? 0 : d / max;
105
106   if(max == min){
107     h = 0; // achromatic
108   }else{
109     switch(max){
110       case r: h = (g - b) / d + (g < b ? 6 : 0); break;
111       case g: h = (b - r) / d + 2; break;
112       case b: h = (r - g) / d + 4; break;
113     }
114     h /= 6;
115   }
116
117   return [h, s, v];
118 }
119 /* accepts parameters
120  * h  Object = {h:x, s:y, v:z}
121  * OR
122  * h, s, v
123  */
124 function HSVtoRGB(h, s, v) {
125   var r, g, b, i, f, p, q, t;
126   if (arguments.length === 1) {
127     s = h.s, v = h.v, h = h.h;
128   }
129   i = Math.floor(h * 6);
130   f = h * 6 - i;
131   p = v * (1 - s);
132   q = v * (1 - f * s);
133   t = v * (1 - (1 - f) * s);
```

```
  switch (i % 6) {
    case 0: r = v, g = t, b = p; break;
    case 1: r = q, g = v, b = p; break;
    case 2: r = p, g = v, b = t; break;
    case 3: r = p, g = q, b = v; break;
    case 4: r = t, g = p, b = v; break;
    case 5: r = v, g = p, b = q; break;
  }
  return [
    Math.round(r * 255),
    Math.round(g * 255),
    Math.round(b * 255)
  ];
}

function rgbToHex(r, g, b) {
  if (r > 255 || g > 255 || b > 255)
    throw "Invalid color component";
  return ((r << 16) | (g << 8) | b).toString(16);
}


function activeColor (id,sliders) {
  this.init = () =>{
    this.element = document.getElementById(id);
    [this.r, this.g, this.b] = [ 254, 254, 254 ];
    [this.h, this.s, this.v] = rgbToHsv(this.r,this.g,this.b);
    if (arguments.length > 1){
      var self = this;
      for (key in sliders) {
        document.getElementById(sliders[key]).addEventListener( "input", function(){self.changed( key, sliders[key] ) });
      }
    }
    this.updateElementsRGB(this.r,this.g,this.b);
  }

  this.updateElementsRGB = (r,g,b) => {
    var hex = "#" + ( "000000" + rgbToHex(r,g,b) ).slice(-6);
    this.element.style.backgroundColor = hex;
    var self = this;
    for (var key in sliders) {
      document.getElementById(sliders[key]).value = self[key];
    }
    if (activeBLE.active) activeBLE.sendRGBW(this.r,this.g,this.b);
  };

```

```
180   this.updateRGB = (r,g,b,v_change) => {
181     [this.r, this.g, this.b] = [ r, g, b ];
182     if (arguments.length === 3) v_change = false;
183     if (v_change) {
184       [this.h, this.s, this.v] = rgbToHsv(this.r,this.g,this.b);
185     } else {
186       [this.h, this.s] = rgbToHsv(this.r,this.g,this.b);
187       [this.r, this.g, this.b] = HSVtoRGB(this.h, this.s, this.v);
188     }
189     this.updateElementsRGB(this.r,this.g,this.b);
190   };
191
192   this.updateHSV = (h,s,v) => {
193     [this.h, this.s, this.v] = [h,s,v];
194     [this.r, this.g, this.b] = HSVtoRGB(this.h, this.s, this.v);
195     this.updateElementsRGB(this.r,this.g,this.b);
196   };
197
198   this.changed = (key,id) => {
199     var el = document.getElementById(id);
200     if (key = "v") this.updateHSV(this.h, this.s, el.value);
201   }
202
203   this.init();
204 }
205
206 var colorInfo = new activeColor("currentRGB",{v:"slider_HSV_v"});
207
208 function mouseRGB(element,e){
209   var eventLocation = getEventLocation(element,e);
210   var coord = "x=" + eventLocation.x + ", y=" + eventLocation.y;
211
212   // Get the data of the pixel according to the location generate by the
      getEventLocation function
213   var context = element.getContext('2d');
214   var pixelData = context.getImageData(eventLocation.x, eventLocation.y, 1,
      1).data;
215
216   // If transparency on the image
217   if((pixelData[0] == 0) && (pixelData[1] == 0) && (pixelData[2] == 0) &&
      (pixelData[3] == 0)){
218     coord += " (Transparent color detected, cannot be converted to HEX)";
219   }
220   else {
221     colorInfo.updateRGB(pixelData[0], pixelData[1], pixelData[2],false);
222   }
223 }
```

```
224
225 function touchRGB(element,e){
226   e.pageX = e.changedTouches[0].pageX;
227   e.pageY = e.changedTouches[0].pageY;
228   e.preventDefault();
229   mouseRGB(element,e);
230 }
231
232 function getEventLocation(element,event){
233   var pos = getElementPosition(element);
234
235   return {
236     x: (event.pageX - pos.x),
237     y: (event.pageY - pos.y)
238   };
239 }
240
241 function getElementPosition(obj) {
242   var curleft = 0, curtop = 0;
243   if (obj.offsetParent) {
244     do {
245       curleft += obj.offsetLeft;
246       curtop += obj.offsetTop;
247     } while (obj = obj.offsetParent);
248     return { x: curleft, y: curtop };
249   }
250   return undefined;
251 }
252
253 var pick;
254 var canvas;
255 function getColorPicker(){
256   var element = document.querySelector('#colorSpace');
257   if (!element.contains(canvas))  {
258     pick = new ColorPicker(element);
259     canvas = pick.element.querySelector('canvas');
260     canvas.addEventListener("mousemove",(e)                    =>
   {mouseRGB(canvas,e)},false);
261     canvas.addEventListener("touchmove",(e) => {touchRGB(canvas,e)},false);
262   }
263 }
```

Appendix 6

Content of “input-range.css”

Below is showed the code from a file “input-range.css” which is responsible for the appearance of the range sliders in the app:

```
.slidecontainer {
    width: 99%;
}

.slidecontainer .vertical{
    height: 100%;
}

.slider {
    -webkit-appearance: none;
    width: 100%;
    height: 25px;
    background: #d3d3d3;
    outline: none;
    opacity: 0.7;
    -webkit-transition: .2s;
    transition: opacity .2s;
}

.slider:hover {
    opacity: 1;
}

.slider::-webkit-slider-thumb {
    -webkit-appearance: none;
    appearance: none;
    width: 25px;
    height: 25px;
    background: #4CAF50;
    cursor: pointer;
}

.slider::-moz-range-thumb {
    width: 25px;
    height: 25px;
    background: #4CAF50;
    cursor: pointer;
}

/*customised range*/ input[type="range"].range
```

```
{
    cursor: pointer;
    width: 100% !important;
    -webkit-appearance: none;
    z-index: 200;
    width:50px;
    outline: none;
    opacity: 0.7;
    -webkit-transition: .2s;
    transition: opacity .2s;
    background: #b8b8b8;
    border-radius: 2px;
}

input[type="range"].range:hover {
    opacity: 1;
}

/*customised range when focusing on input */ input[type="range"].range:focus
{
    border: 0 !imporant;
    outline: none !important;
}

/*customised range slider icon*/ input[type="range"].range::-webkit-slider-thumb
{
    -webkit-appearance: none;
    width: 75px;
    height: 75px;
    background-color: #555;
    background-image: -webkit-gradient(linear, 0% 0%, 0% 100%, from(#00CCFF), to(#039be5));
    background-image: -webkit-linear-gradient(right, #00CCFF, #039be5);
    background-image: -moz-linear-gradient(right, #00CCFF, #039be5);
    background-image: -ms-linear-gradient(right, #00CCFF, #039be5);
    background-image: -o-linear-gradient(right, #00CCFF, #039be5);
    border-radius: 2px;
}

/*setting round corners to the range */ input[type="range"].round {
    -webkit-border-radius: 35px;
    -moz-border-radius: 35px;
    border-radius: 35px;
}

```

```
84  /*setting round corners to the range slider icon*/ input[type="range"].round::-webkit-slider-thumb {
85      -webkit-border-radius: 37px;
86      -moz-border-radius: 37px;
87      -o-border-radius: 37px;
88  }
89
90  /* set range from 0 - 1 horizontal as by default */ .horizontal-lowest-first
91  {
92      -webkit-transform:rotate(0deg);
93      -moz-transform:rotate(0deg);
94      -o-transform:rotate(0deg);
95      -ms-transform:rotate(0deg);
96      transform:rotate(0deg);
97  }
98
99  /* set range from 1 - 0 horizontal (highest first) */ .horizontal-highest-first
100 {
101     -webkit-transform:rotate(180deg);
102     -moz-transform:rotate(180deg);
103     -o-transform:rotate(180deg);
104     -ms-transform:rotate(180deg);
105     transform:rotate(180deg);
106 }
107
108 /* set range from 0 - 1 vertically (lowest on top) */ .vertical-lowest-first
109 {
110     top: 40%;
111     -webkit-transform:rotate(90deg);
112     -moz-transform:rotate(90deg);
113     -o-transform:rotate(90deg);
114     -ms-transform:rotate(90deg);
115     transform:rotate(90deg);
116 }
117
118 /* set range from 1 - 0 vertically (highest on top) */ .vertical-highest-first
119 {
120     margin-top: 50%;
121     -webkit-transform:rotate(270deg);
122     -moz-transform:rotate(270deg);
123     -o-transform:rotate(270deg);
124     -ms-transform:rotate(270deg);
125     transform:rotate(270deg);
126 }
```

Appendix 7

Content of “footer-nav.css”

Below is showed the code from a file “footer-nav.css” which is responsible for the appearance of the navigation pane in the app:

```
.footer-nav {
    list-style-type: none;
    margin: 0;
    padding: 0;
    overflow: hidden;
    background-color: #f3f3f3;
    box-shadow: 16px 0px 0px 0px rgba(0,0,0,0.2);
    position: fixed;
    bottom: 0;
    width: 100%;

    display: flex;
}

.footer-nav li {
    float: left;
    flex-grow: 1;
}

.footer-nav li a {
    display: block;
    color: #666;
    text-align: center;
    padding: 14px 16px;
    text-decoration: none;
    border-bottom: 3px solid #8e8e8e;
}

.footer-nav li a:hover:not(.active) {
    background-color: #ddd;
}

.footer-nav li a.active {
    color: white;
    background-color: #707070;
    border-bottom: 3px solid #039be5;
}
```