

Leveraging Raspberry Pi as a server for the integration of the NETCONF protocol within IoT systems based on YANG

Dalibor Đumić¹, Slobodan Lubura²

¹ International Burch University, Sarajevo, Bosnia and Herzegovina

² University of East Sarajevo, East Sarajevo, Bosnia and Herzegovina

dalibor.dumic@stu.ibu.edu.ba

slobodan.lubura@ets.ues.rs.ba

Abstract – Herein the idea of leveraging Raspberry Pi as a server for the integration of an incipient network management protocol, the Network Configuration Protocol (NETCONF), within IoT systems based on YANG is presented. The practical realization of this idea requires the implementation of the NETCONF protocol together with REpresentational State Transfer web services (RESTful). Such an interesting and innovative practical realization like this opens new additional possibilities in domotics systems and these possibilities will be discussed in this paper.

Keywords – Django, domotics, Internet of Things, NETCONF, Raspberry Pi, RESTful web services, YANG

1. Introduction

In each home network there are always heterogeneous devices that are expected to be connected. All of these devices are different if compared because they can be based on different hardware platforms, the controller services can be of a different nature, and also the software components for enabling the network access can vary [1]. For example, when we compare wearable technology based on the IoT like a smartwatch or wristband with smart home devices such as a washing machine or air conditioner, we will notice different capabilities in terms of memory usage, processing speed, and power consumption [2]. Because of that, the IoT devices can be generally classified based on their key characteristics:

- communication flows in the system,
- memory management,
- data manipulation and processing,
- power control and consumption.

For example, a smart coffee machine is not always powered on because it performs certain tasks when required, but only when a user turns on it via a user interface such as a mobile application whenever the user wants to drink a coffee or when the user is on the way to home and wants to have already prepared coffee. These kinds of devices consume less power for communication. There are many actuators in home automation systems that must be managed by systems connected to the Internet via network protocols [3].

The focus of this paper is on the practical implementation of the methodology proposed in [4] and this methodology was carried out by the empirical study of the NETCONF protocol that will be used as a network protocol for enabling the connection of the gateway with the Internet. The gateway will perform effective management of sensors and devices in a home network and it will be based on the RESTful technologies.

The paper is organized into five sections. Section 1 introduces us to the IoT systems and the purpose of this paper. In Section 2, the NETCONF protocol and its features are introduced. The proposed integration of the NETCONF protocol in the IoT is detailed in Section 3. The results of the proposed integration of the NETCONF protocol are noted in Section 4. The benefits of the proposed integration and the main conclusions are discussed in Section 5.

2. The Network Configuration Protocol (NETCONF) and its features

A. NETCONF

The Network Configuration Protocol (NETCONF) is a network management protocol with great features such as installing, manipulating, and deleting the configuration of the devices in the network. Its purpose is managing network devices, retrieving its configuration data, and uploading or manipulating new configuration data of the network devices [5]. That means devices on the network can take different states according to their configuration.

To switch between the device's states, the configuration datastores are used. By definition, a configuration datastore contains a set of information that is needed for the configuration, and thereby that configuration is required to change the state of a device to chosen operational state from its initial default state. NETCONF currently supports event notification features and the following multiple configuration datastores [6]:

- "running" – this configuration is always present and it is used as the currently active configuration
- "startup" – this configuration is used in the next startup
- "candidate" - this configuration that can be used instead of currently running configuration through an explicit commit.

By using NETCONF operations, it is possible to manipulate device configuration. The NETCONF operations are invoked as Remote Procedure Calls (RPCs) from the client to the server. Some minor operations are [6]:

- "commit" - commits the "candidate" configuration to "running",
- "copy-config" - copy one configuration datastore to another,
- "edit-config" - changes the contents of a configuration database,
- "get-config" - retrieves configuration datastore,
- "lock" - prevent changes to a datastore from another party, and
- "unlock" - releases lock on a datastore.

Configuration data stored on devices and the protocol messages between devices are encoded in Extensible Markup Language (XML) on both client and server side. Any script or application can be the client in order to be run as part of a network manager. The server is typically a network device. There is a rule that a device on the network must support at least one NETCONF session. The main NETCONF message exchange between client and server in a single NETCONF session [7] is illustrated in Figure 1. At the start, the device and controller create a NETCONF session and share their list of their own capabilities by sending <hello> messages. A capability describes a supported data model. After the session has started,, the NETCONF executes exchanges <rpc> and <rpc-reply> messages. The <rpc> message consists of an enclosed NETCONF command which is sent from the controller to the device. The <get> command in the <rpc> message is used to get the running configuration and state information of the device (3). The <edit-config> request is used to write a specific configuration on the device (5). The <rpc-reply> message is sent from the device to the controller in response to a <rpc> message. The response data for the given method invoked is encoded as one or more child elements enclosed in the <rpc-reply> message.

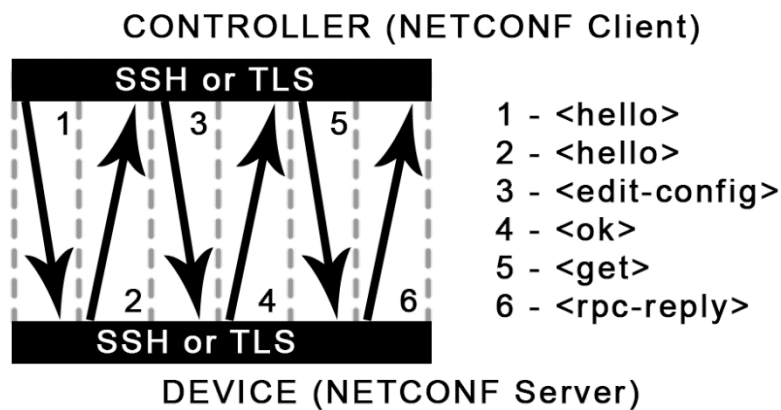


Figure 1. NETCONF messages

The information that a client retrieves from the server consists of two parts: configuration data and state data [6]. The purpose of the configuration data is to give a description of actions that will change a system from its previous state into the state described in the configuration data, while the purpose of the state data is to provide information such as read-only status data and collected statistics. For specifying NETCONF data models and operations, the YANG data modeling language is used.

A. YANG

To perform the NETCONF operations, a YANG module has to be defined as a hierarchy of data such as configuration data, state data, RPCs, and notifications. By defining the YANG module, a description of all data sent between both NETCONF client-side and server-side becomes completed. Each YANG module is consisting of statements and some of the statements are previewed in Table 1 [8].

Table 1. YANG statements

<i>Statements</i>	<i>Description</i>
augment	Extends existing data hierarchies
choice	Defines mutually

	exclusive alternatives
container	Defines mutually exclusive alternatives
extension	Allows new statements to be added to YANG
feature	Indicates parts of the model are optional
grouping	Groups data definitions into reusable sets
key	Defines the key leafs for lists
leaf	Defines a leaf node in the data hierarchy
leaf-list	A leaf node that can appear multiple times
list	A hierarchy that can appear multiple times
notification	Defines notification
rpc	Defines input and output parameters for an RPC
typedef	Defines a new type
uses	Incorporates the contents of a "grouping"

With the help of XML parsers and XSLT scripts, a translation of the YANG module into an equivalent XML syntact becomes possible. Every YANG module consists of a set of built-in types and has a type mechanism through which additional types may be defined. The modeler of the YANG module can add constraints to the model to prevent impossible or illogical data. The purpose of these constraints is to provide information about the data being sent from the server and help a client to understand the data that the server will accept in order to avoid sending incorrect data from the client to the server. Table 2 briefly describes some other common YANG constraints [9]

Table 2. YANG constraints

<i>Statements</i>	<i>Description</i>
length	Limits the length of string

mandatory	Requires the node appear
max-elements	Limits the number of instances in list
min-elements	Limits the number of instances in list
must	XPath expression must be true
pattern	Regular expression must be satisfied
range	Value must appear in range
reference	Value must appear elsewhere in the data
unique	Value must be unique within the data
when	Node is only present when XPath expression is true

Generally said, the YANG module is a single data model that contains three types of statements:

- module-header statements – they describe the module and provide the information about the module
- revision statements – they provide information about the history of the module
- definition statements – they are the body of the module where the YANG module is defined.

In order to use the YANG module, it firstly has to be defined or modeled to the specific problem domain. After that, the YANG module can be loaded, compiled, or coded into the server. In the end, the NETCONF server may implement any number of the YANG modules [10].

3. Proposed Methodology

After the empirical study of the NETCONF protocol and retrieving its features, an implementation of the proposed integration was divided into two parts: server-side and client-side, as it is shown in Figure 2.



Figure 2. Both client and server sides are communicating over the Internet [4]

A. Server-side

To implement the proposed integration, the following requirements are defined:

- small physical dimensions, because it has to be hidden in home installation and not visible;
- able to boot Linux Operating System, since the Linux OS is open-source;
- has General Purpose Input Output (GPIO) pins for interfacing with the sensors and devices,
- has Ethernet port and/or WiFi module, and
- CPU based on ARM for fast computing.

A great match for the single board with the following characteristics is Raspberry Pi 3 B+, which is based on a 1.4GHz 64-bit quad-core ARM Cortex-A53 processor. The good thing about Raspberry Pi is that it has the GPIO module which can be used through several programming languages such as C, C#, Python, Java, etc. The fact is that the integration will be implemented by using Python programming language and it makes Raspberry Pi a perfect match [11]. A server would be connected via appropriate connection lines to these rooms as it is shown in the Figure 3.

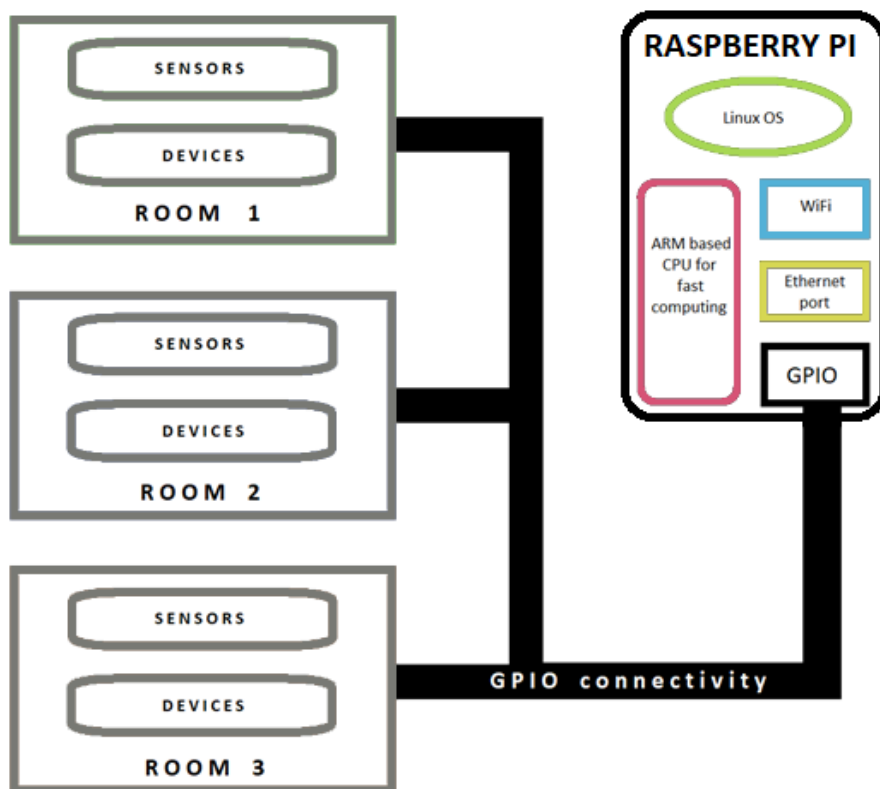


Figure 3. Raspberry Pi as server connected to sensors and devices in each room via GPIO line [4]

In order to build a server, the Netopeer2, a set of tools implementing network configuration based on the NETCONF protocol, is installed [12][13]. Each room in a home has sensors and relays for controlling devices. For each room, a custom YANG module is created, and each custom YANG module manipulates with data such as temperature, humidity, open or closed status, turned off or turned on status, etc. Thanks to custom YANG modules, the server can easily manage the information related to the sensors and relays in the home. The structure of the simplest custom YANG module for a room is shown in the section “Appendix”.

B. Client-side

On the client-side, any device which supports the NETCONF protocol can communicate with the server. However, the challenge is to develop an application by means of RESTful services. It should send the RPC commands such as “edit-config” or “get-config” directly to the NETCONF server in order to retrieve information about rooms in the user’s home. Finally, its interface must be user-friendly and rich with data charts, data graphs, toggle buttons, etc.

The very first step is to develop a script that shall “talk” with the NETCONF server. Thanks to the enormous possibilities of the Python programming language, it is possible to communicate with the server via the NETCONF protocol by using ncclient library. The ncclient library enables an easy way of the client-side scripting around the NETCONF protocol, and as well as the possibility of the application development [14].

The next step was to develop a web application and merge it with the script based on ncclient library. There are many high-level Python web frameworks and one of them is Django. Django is specific because it encourages rapid development and clean, pragmatic design [15]. By combining Django and ncclient, a powerful user-friendly web application is created, and it will fulfill its main purpose – to collect all information about the conditions such as temperature and humidity in the rooms of the user’s home and to control devices in the rooms of the user’s home, all of it over the NETCONF protocol.

4. Results

On the client-side we have an application based on both front-end and back-end development in the Django framework and merging its back-end with the ncclient module for interfacing with the server as shown in Figure 4.

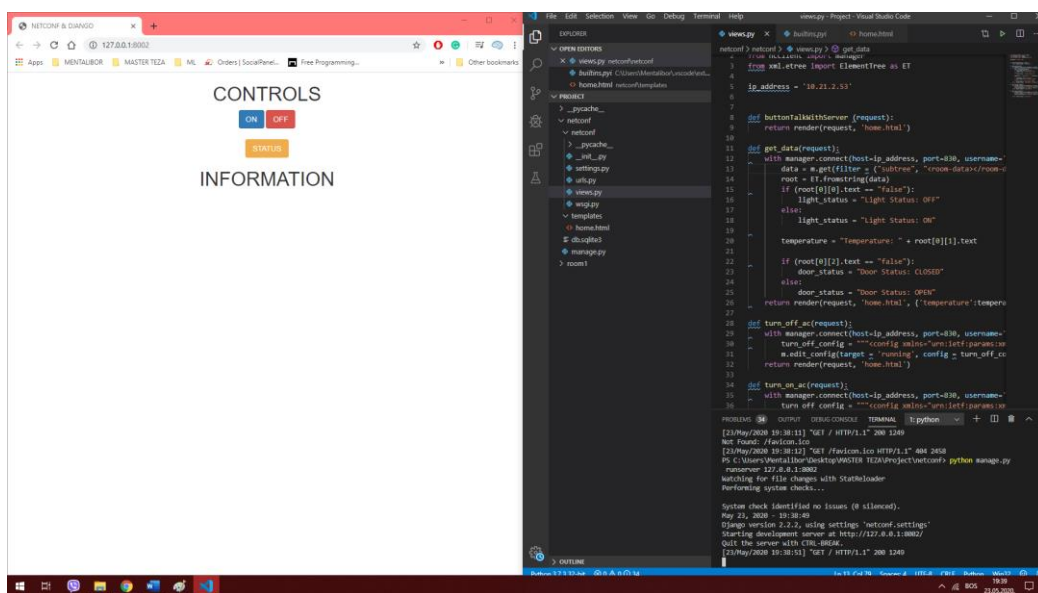


Figure 4. Developed client application

On the server-side we have Raspberry Pi computer booting Linux OS which runs Netopeer2 and sysrepo modules for enabling the NETCONF protocol and interfacing the data through YANG modules. The Raspberry Pi is connected to several sensors and actuators, as shown in the Figure 5:

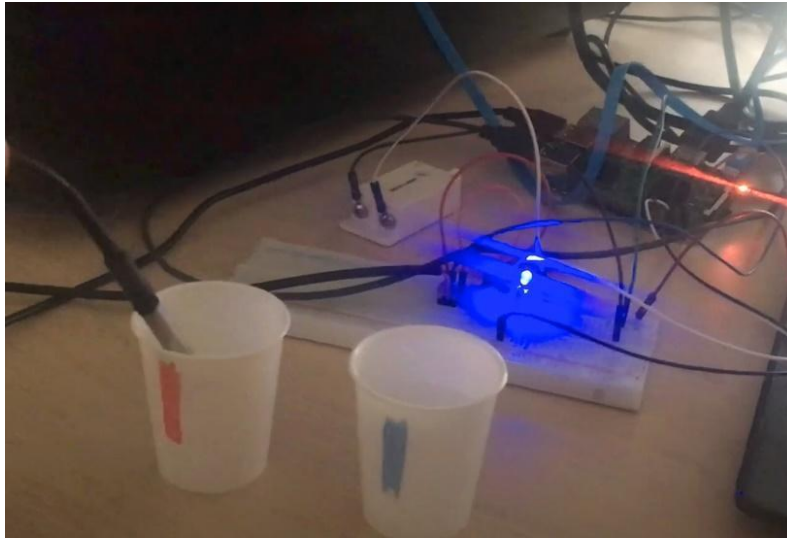


Figure 5. Raspberry Pi running as the NETCONF server

The URL of the recorded video of the methodology proposed in this paper can be found below in the reference section [16]. A clip from the recorded video is shown in Figure 6 and it can be seen that two processes are running parallelly: sysrepo and netopeer2.



Figure 6. Testing the proposed methodology

An overview of both client and server sides is shown in Figure 7.

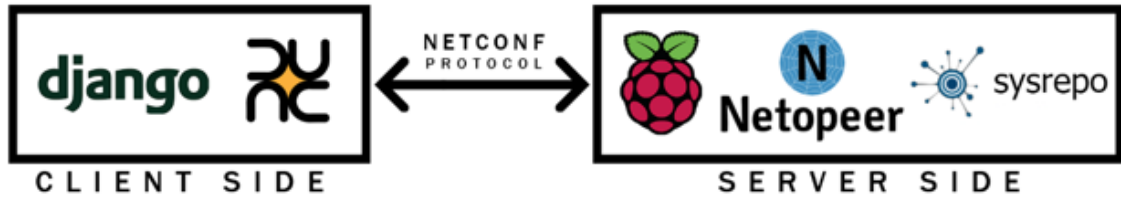


Figure 7. Used technologies on both client and server sides

The complete overview of the proposed integration is shown in Figure 8.

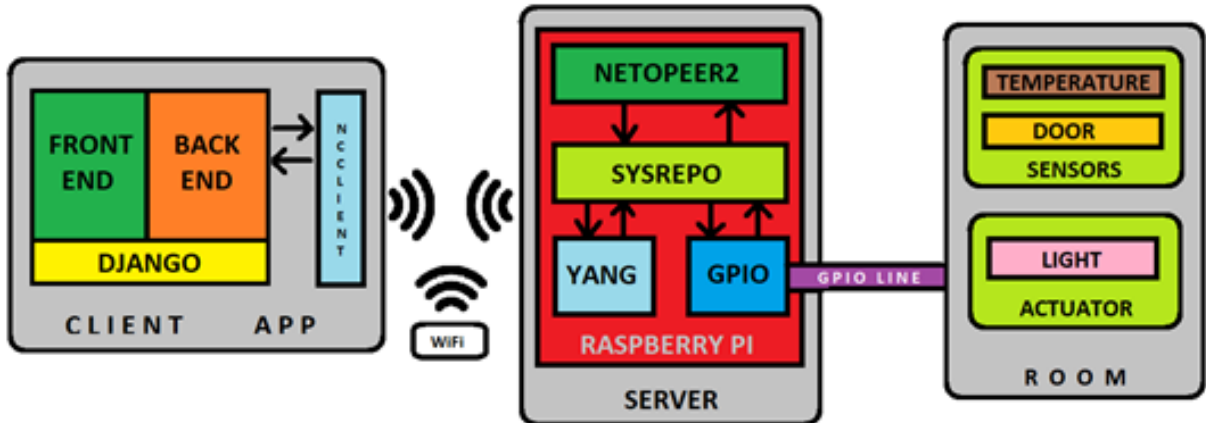


Figure 8. Overview of the complete integrated system

4. Conclusion

Through the empirical study of the NETCONF protocol, great capabilities of the NETCONF protocol are discovered. The NETCONF protocol allows us to have an unlimited number of YANG modules with different structures of the data. This characteristic of the NETCONF protocol is of crucial importance for using it in the home automation system and similar systems. The proposed integration is not a challenge anymore. Thanks to the powerful Python Web framework and ncclient Python library, it is possible to develop a rich web application that can be outperformed on many devices such as single board computers, desktop computers, notebooks, and even tablets.

APPENDIX

Implemented module for a room in the YANG language:

```

module room1 {
    namespace "urn:sysrepo:room1";
    prefix r1;
    description "The room yang module.";
    revision 2019-09-14 {
        description "Initial revision.";
    }

    container room-data {
        description "Room 1 info.";
        leaf temperature {

```

```

        description "Actual temperature inside the room.";
        type uint8 {
            range "0..125";
        }
    }

    leaf humidity {
        description "Actual humidity inside the room.";
        type uint8 {
            range "0..100";
        }
    }

    leaf ac-status {
        description "Informs whether the AC is switched on or off.";
        type boolean;
    }
}
}

```

ACKNOWLEDGMENT

Many thanks to the experts from the RT-RK Institute for Computer Based Systems in Banja Luka who contributed and influenced so much to the development of this research from the early stages of the project.

REFERENCES

- [1] M. Tooba, A. Muhammad and A. M. Martinez-Enriquez, "Smart Solution for Heterogeneous Device Interoperability in IoT," 2018 Seventeenth Mexican International Conference on Artificial Intelligence (MICAI), Guadalajara, Mexico, 2018, pp. 70-75,
- [2] Van den Abeele, F., Hoebeke, J., Moerman, I., & Demeester, P. (2015). Integration of Heterogeneous Devices and Communication Models via the Cloud in the Constrained Internet of Things. *International Journal of Distributed Sensor Networks*.
- [3] Vijay S., Banga M.K. (2018) Management of IoT Devices in Home Network via Intelligent Home Gateway Using NETCONF. In: Kumar N., Thakre A. (eds) *Ubiquitous Communications and Network Computing. UBIUNET 2017. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, vol 218. Springer, Cham
- [4] D. Đumić, S. Došlić, M. Antić, B. Milić, "Integration of the NETCONF Protocol in the Internet of Things by means of RESTful Web Services", 6th International Conference on Electrical, Electronic and Computing Engineering IcETRAN, pp. 983 - 987, ETRAN Society, June 2019
- [5] R. Enns, M. Brojklund, J. Schoenwaelder and A. Bierman, "Network Configuration Protocol (NETCONF)", Internet Engineering Task Force (IETF), ISSN: 2070-1721, June 2011. [Online]. Available: <https://tools.ietf.org/html/rfc6241>
- [6] H. Ji, B. Zhang, G. Li, X. Gao and Y. Li, "Challenges to the New Network Management Protocol: NETCONF," *2009 First International Workshop on Education Technology and Computer Science*, Wuhan, Hubei, 2009, pp. 832-836, doi: 10.1109/ETCS.2009.189.

- [7] M. Dallaglio, N. Sambo, F. Cugini and P. Castoldi, "Management of sliceable transponder with NETCONF and YANG," 2016 International Conference on Optical Network Design and Modeling (ONDM), Cartagena, 2016, pp. 1-6
- [8] M. Dallaglio, N. Sambo, F. Cugini, P. Castoldi, "Management of sliceable transponder with NETCONF and YANG", International Conference on Optical Network Design and Modeling, pp. 1 – 6, IEEE, May 2016
- [9] P. Shafer, "An Architecture for Network Management using NETCONF and YANG", Internet Engineering Task Force (IETF), ISSN: 2070-1721, June 2011, [Online]. Available: <https://tools.ietf.org/id/draft-ietf-netmod-arch-07.html>
- [10] M. Brojklund, "YANG – A Data Modeling Language for the Network Configuration Protocol (NETCONF)", Internet Engineering Task Force (IETF), ISSN: 2070-1721, October 2010. [Online]. Available: <https://tools.ietf.org/html/rfc6020>
- [11] The Raspberry Pi Foundation. "Raspberry Pi 3 Model B+", [Online], Available: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/>
- [12] Czech Educational and Research Network (CESNET), "Netopeer2 – The NETCONF Toolset", [Online], Available: <https://github.com/CESNET/Netopeer2>
- [13] sysrepo - YANG-based datastore for Unix/Linux application, [Online], Available: http://www.sysrepo.org/static/doc/html/start_page.html
- [14] S. Bhushan, L. Pouloupols, Python library for NETCONF clients, [Online], Available: <http://ncclient.readthedocs.org/>
- [15] Django Software Foundation, [Online], Available: <https://docs.djangoproject.com/en/3.0/>
- [16] NETCONF Protocol + Raspberry Pi + Django = Home Automation || Yugoscientiz © 2019, [Online], Available: <https://www.youtube.com/watch?v=ZoiYGt2NbCA>