Design of CoAP Based Model for Monitoring and Controlling Physical Parameters

Vishnu Kanthan Rathina Raj(✉) and Meena Belwal

Department of Computer Science and Engineering, Amrita School of Engineering, Amrita Vishwa Vidyapeetham, Bengaluru, India
vishnukanthan.r@microchip.com, b_meena@blr.amrita.edu

Abstract. With rapid development and increasing demand of Internet of Things (IoT) applications, effective client server system is necessary for fast communication without data loss. IoT applications generally have sensors to provide various physical parameters and actuators to operate as per sensor input. A specialized protocol that enables this machine-to-machine communication is Constrained Application Protocol (CoAP). The CoAP shares features with HTTP such as REST model but personalized for constrained devices like embedded micro controllers and networks in IoT. Industries at present use commercial small computers such as Raspberry Pi for developing IoT applications to speed up the process. This has certain overheads like cost and fixed set of libraries. The purpose of this paper is to present an economical and scalable solution for IoT application development. The working prototype model comprises a bare metal microchip pic 8-bit micro controller, Ethernet controller, infrared sensor, browser display for sensor data, actuator to on/off the motor and integrated TCP/IP CoAP stack libraries. This paper demonstrates the function of the proposed model and how scalability can be achieved by connecting multiple sensors. Infrared sensor can be replaced with any device that generates data such as blood pressure, temperature etc.

Keywords: Internet of Things (IoT) · Constrained Application Protocol (CoAP) · HTTP · REST model · Micro controller · Sensor · Actuator

1 Introduction

IoT applications grow exponentially with availability of sensors, actuators and cost-effective low-power processors. It is estimated that over 17 billion devices connected worldwide in IoT, are collecting and sharing data [1]. From home automation to business to consumers, more and more resource constrained devices are getting connected to network controlled and monitored from remote.

IoT implementation using single board small devices has trade-off with cost and scalability [2]. This research is focused on developing IoT system using peripheral interface controller (PIC) 18F87K222, a bare metal 8-bit micro controller, and achieving complete monitoring and controlling of IoT endpoints using integrated TCP/IP CoAP libraries that are exclusively developed for this work. It receives sensor data and sends

N. Kumar et al. (Eds.): UBICNET 2021, LNICST 383, pp. 105–115, 2021.
https://doi.org/10.1007/978-3-030-79276-3_9

to clients across the network. It costs below 4 USD with features such as high-speed data transfer of 10 Mbps. The proposed solution is tested with different CoAP clients to ensure interoperability.

This model would be useful in numerous real-time applications. Following is a typical application example. Getting alert notification from a remote server when CCTV camera captures image of an intruder. In this case CCTV camera is the sensor, and this model can send alert signal to a remote client upon capturing images of intruders. Instructions can also be sent from remote to control motor.

Bare metal provides convenience of direct programming of hardware registers and eliminates fixed set of libraries. Also, complete flash memory is available for programming and other important resources like CPU, memory and storage units can be utilized at their full capacity and hence it provides maximum performance and speed.

Keeping in mind low overhead, limited RAM/ROM of constrained environments, CoAP web protocol is designed with additional feature that it can communicate to HTTP as well.

The rest of the paper is organized in the following order. Section 2 presents relevant work published by various authors. Section 3 explains overview and stages of IoT and comparison between IoT and web stacks. Design and development of the proposed model is explained in Sect. 4. Section 5 presents the execution results.

2 Related Work

Louis COETZEE [3] et al. found through experiments that CoAP is an efficient transport in low signal strength environments. Dejana Ugrenovic [4] et al. analyzed and simulated the IoT healthcare system that monitors patient's health condition. Gyuhong Choi et al. [5] presented an efficient communication scheme for streaming services in IoT. Kavitha. B. C and Vallikannu. R [6] proposed Raspberry Pi based IoT for monitoring parameters and send the data to cloud. Mrs. Vaishali Puranik et al. [7] et al. discussed automation of agricultural processes such as maintenance, pest control etc. using IoT. G. Tanganelli, C. Vallati, E. Mingozzi [8] have introduced an open source CoAP library called CoAPthon, which can be used to develop IoT applications based on CoAP. This library has been used for this research work. Er. Pooja Yadav [9] et al. discussed general challenges in IoT applications in India. Likith bhushan H N and Niranjan K R [10] analysed the use of wireless sensor networks based IoT in health care applications. Dweepayan Mishra et al. [11] proposed Arudino kit based IoT programmed water system for irrigation. Madhusudan Singh et al. [12] described the infrastructure of IoT for Blockchain network based IoT. Dr.Digvijaysinh Rathod [13] et al. demonstrated that CoAP proxy is vulnerable and succeeded to show that the information is transiting in clear text format which is susceptible for attacks or manipulation of data. Ahmad Zainudin et al. [14] compared CoAP and MQTT protocols and observed that CoAP performance is better in terms of delay. Ravi Kishore Kodali et al. [15] used ESP32 and ESP8266 to implement CoAP based home automation. Jiahao Li, Gengyu Wei [16] suggested blockchain based security for CoAP nodes. Girum Ketema Teklemariam et al. [17] illustrated RESTful interface tool called RESTlets in IoT applications with sensors and actuators. Syed Roohullah Jan et al. [18] explained and compared CoAP with HTTP. Lucas R. B. Brasilino, Martin

Swany [19] presented various security threats in IoT devices. Alok Kumar Gupta et al. [20] discussed an energy saving electrical device Surveillance and Control system based on IOT. A. Paventhan et al. [21] presented an approach to using CoAP for agriculture monitoring. Matthias Kovatsch [22] proposed a system architecture for CoAP based IoT cloud services.

3 CoAP Protocol

As depicted in Fig. 1, Overview of IoT has three stages Sensors that capture data, a platform or bridge to communicate data using protocols and networks, and applications that use data.

Fig. 1. Stages of IoT

Transmitting sensor data across devices indeed is a complex task. TCP/IP stack deals with connectivity to the Internet. CoAP protocol in Application layer handles communication among devices as shown in Fig. 2.

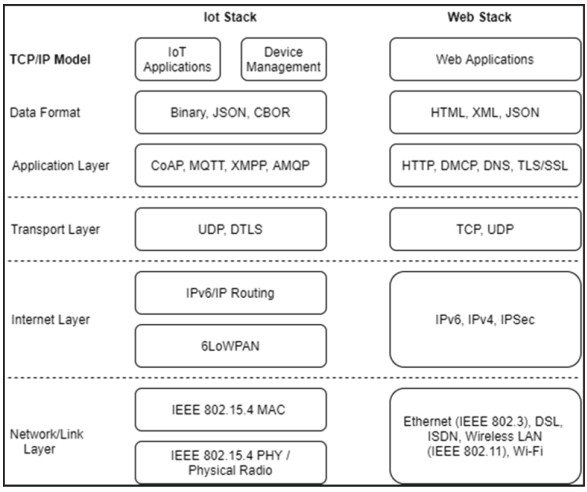

Fig. 2. Comparison of IoT & web stacks

CoAP has lightweight connectionless User Datagram Protocol (UDP) as its transport layer protocol whereas Transmission Control Protocol (TCP) is used by HTTP.

4 Methodology

4.1 High-Level Design

The high-level design of proposed model is shown in Fig. 3. Sensors and actuators are hosted as CoAP resources in the MCU. The MCU is interfaced with the ethernet controller through SPI and acts as the CoAP server. A PC/laptop acts a CoAP client that is running a client software such as CoAPthon3 or Copper. The client and the server are both connected via a conventional twisted pair ethernet cable. The client can retrieve data from the sensor and also control the state of the actuator through the Request-Response model.

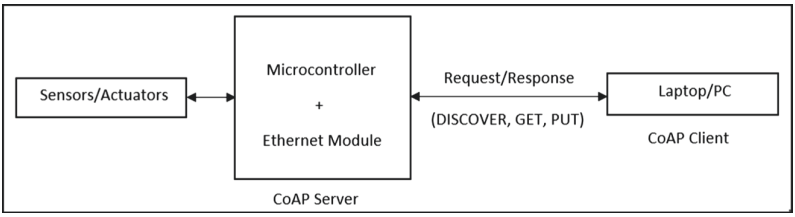

Fig. 3. High-level design

4.2 Low-Level Design

The Fig. 4 shows low-level design and Fig. 5 illustrates the actual implementation of the proposed model. The project is created in MPLAB X IDE for the PIC18F87K22 MCU. Microchip Code Configurator (MCC) is used for the peripheral & library configurations. The system clock of PIC18F87K22 is configured to run at 16 MHz by choosing the Internal RC-Oscillator and the GPIO pins are configured to interface with IR sensor and the Motor control.

The DOUT pin from the IR sensor is connected to the MCU's RD4 that is, pin.no 4 of PORTD. The pins RD0 & RD1 are set as output pins and are connected to the L293D driver. The configurations are shown in Fig. 6. MCC generates various functions that can be readily used when a pin is configured as an input or output. Some of functions used in the project are:

- Pin.no_GetValue()- Gets the value (high/low) of the pin
- Pin.no_SetHigh() and Pin.no_SetLow()- To set a pin high or low
- Pin.no_Toggle()- Toggles the current state of the pin (from high to low or low to high).

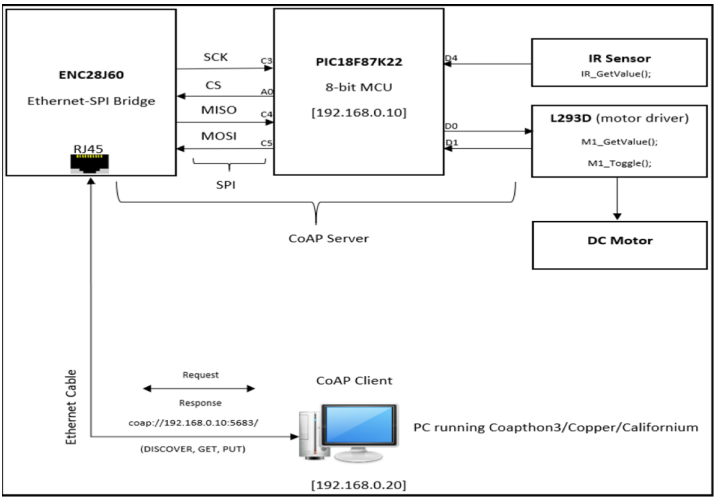

Fig. 4. Low-level design

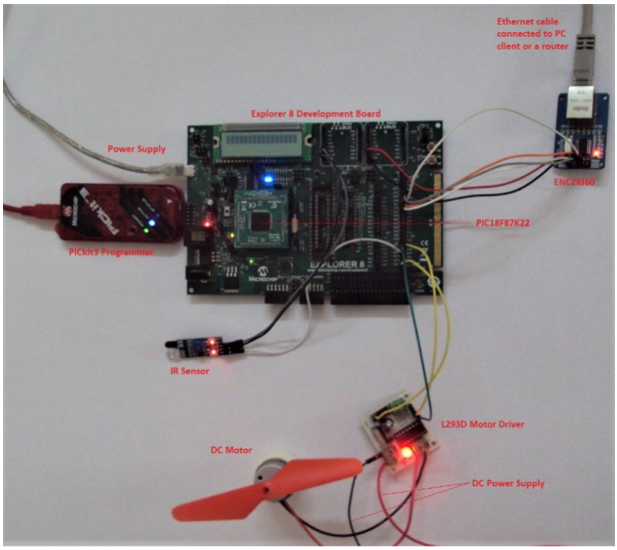

Fig. 5. Actual implementation

4.3 Configuring TCP/IP Lite Stack

The TCP/IP lite stack version 2.2.12 (latest) has been used for implementation. The protocols UDP and ICMP are checked. The rest of them are left unchecked as it is not required for this work and can be added for future extension. But adding too many protocols like the TCP, NTP and DNS may not work as expected due to the embedded

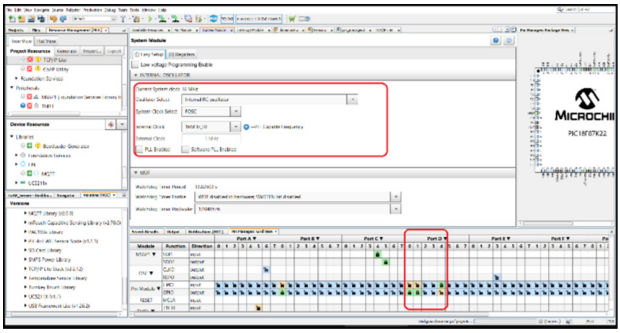

Fig. 6. Initial setup

constraints as only 8-bit MCU is used. The local IPV4 address is set as “192.168.0.10”. This is the server’s IP. The configurations are as shown in Fig. 7.

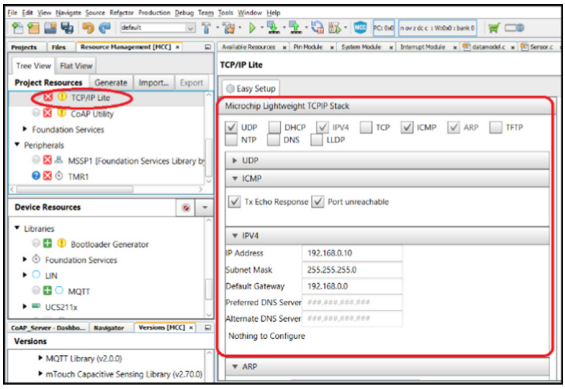

Fig. 7. TCP/IP lite configuration

4.4 Configuring CoAP Library

As the IR sensor and the DC motor are added as simple GPIO interfaces, even though the actual hardware is connected, they needed to be added as CoAP Resources. From the list of libraries, the CoAP Utility library is included to the project. The resources are added and renamed and included under a root called “Node” as shown in Fig. 8.

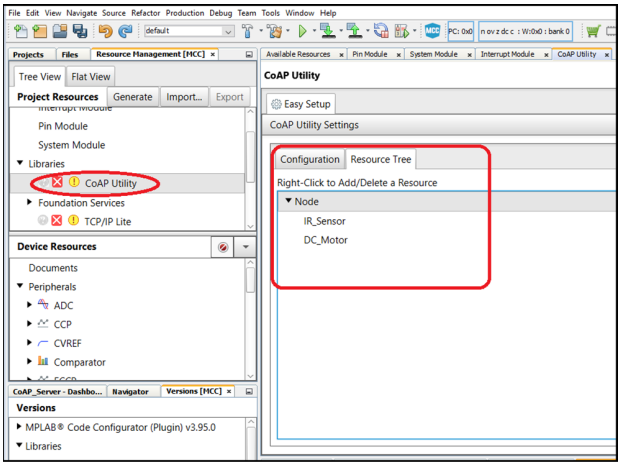

Fig. 8. CoAP utility configuration

4.5 Development of the Firmware

The main.c is as shown below Fig. 9:

```
#include "mcc_generated_files/mcc.h"
void main(void)
{
    SYSTEM_Initialize();
    INTERRUPT_GlobalInterruptEnable();
    INTERRUPT_PeripheralInterruptEnable();
    IO_RB0_SetLow();//RB0 connected to push button(open drain) which is also connected to an LED, set it low
    //To turn the motor OFF initially
    M1_SetHigh();//RD0 motor pin is set high
    M2_SetHigh();//RD1 motor pin is set high

while(1)
    {
        Network_Manage();
    }
}
```

Fig. 9. main.c

The SYSTEM_Initialize() method initializes the interrupt registers, pin manager, the system clock oscillator, timer, initiates the stack and also the CoAP resources. The Network_Manage() is called inside an infinite loop so that when the system is powered ON, the stack keeps running. It is inside this method, where all the above layers TCP/IP and the CoAP stack are managed. When GET request is performed, inside the MCU, it hits the below function SensorGetter(idx). As each resource is assigned a unique identifier, the value of the sensor data is retrieved from the configured input pin of the MCU. The working is similar for the PUT request as well, for which the command is sent to the output pins at which the actuators are interfaced.

5 Results and Discussion

5.1 PING

The ping command comes under the ICMP protocol. It is used to test if a node is able to reach out to another node in a network. The ping command is typed at the command

prompt window of client laptop along with the server's IP address. As seen from the Fig. 10 the average time to hit the server and back is just 6ms.

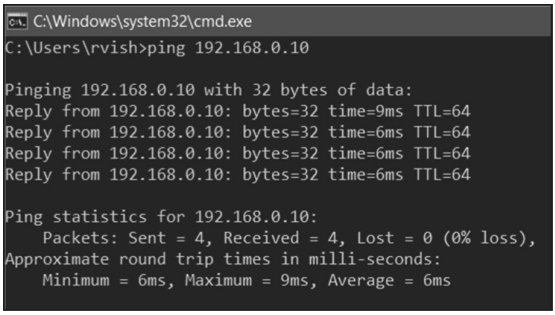

Fig. 10. PING command

5.2 DISCOVER

The Copper client is opened in the Chrome browser and the server's URL is entered. The default port number is 5683. After entering the URL, the server's IP is selected and the DISCOVER button on the top bar is clicked which retrieves information about the CoAP resources hosted by the server as shown in the Fig. 11.

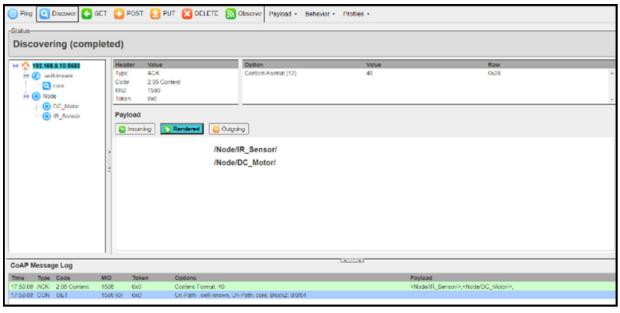

Fig. 11. DISCOVER CoAP resources

5.3 Monitoring the IR Sensor

To monitor the sensor data, the resource IR Sensor under node is selected and then the GET button is clicked. Then an obstacle is placed in front of the IR sensor and then the GET button is clicked again to verify if an obstacle detected message is received. The results are shown in Fig. 12.

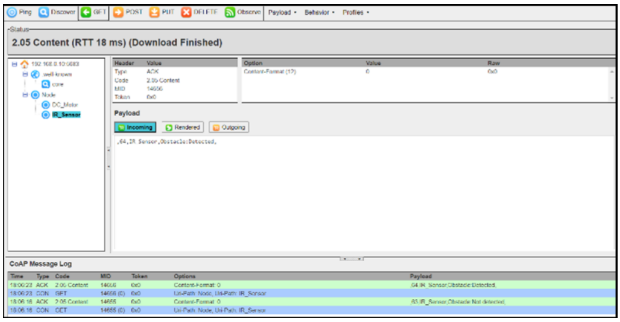

Fig. 12. Monitoring Sensor data

5.4 Monitoring and Controlling the DC Motor

In order to test out the motor, the resource DC Motor is selected under Node and GET button is clicked. Initially the "Motor is OFF" message is received. When PUT button is clicked, it toggles the RD0 pin that is connected as one of the two control signals that are both set as high initially, turns on the motor. When GET button is clicked, the message "Motor is running" is received. The results are shown in Fig. 13.

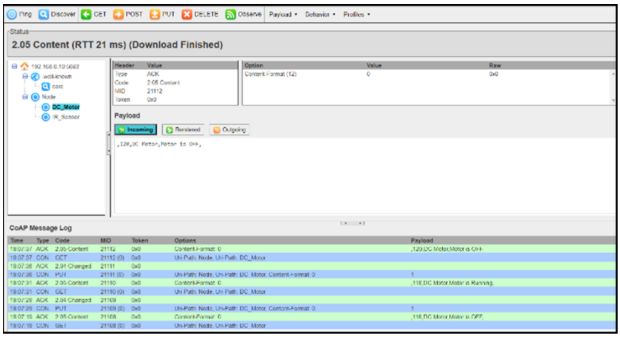

Fig. 13. Monitoring & controlling DC motor

6 Conclusion and Future Work

This paper discussed the development of an IP-address based working prototype model for Server-Client communication architecture using TCP/IP CoAP integrated libraries to monitor and control physical parameters over a local network. The client can identify the resources that are hosted in the server. This work can be further extended to cover security related issues since CoAP is susceptible to DDoS attacks. Hence, implementation of security layer such as IP-Sec or DTLS can be the next objective by taking into consideration the available flash memory and encryption-decryption overheads. Several such developed servers can also be connected to a router or a switch to form a network

and deployed in an environment that needs monitoring. A client connected to such a network can easily identify a server/node with its assigned IP address.

References

1. Statista Research Department, 27 November 2016. https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/
2. Mishra, A., Reichherzer, T., Kalaimannan, E., Wilde, N., Ramirez, R.: Trade-offs involved in the choice of cloud service configurations when building secure, scalable, and efficient Internet-of-Things networks. Int. J. Distrib. Sens. Netw. **16**(2), 1–13 (2020). https://doi.org/10.1177/1550147720908199
3. Louis Coetzee: An Analysis of CoAP as Transport in an Internet of Things Environment. In: Gaberone, Coetzee, B.L., Oosthuizen, D., Mkhize, B. (eds.) IST-Africa 2018, pp. 1–7 (2018)
4. Ugrenovic, D., Gardasevic, G.: CoAP protocol for web-based monitoring in IoT healthcare applications. In: 2015. 23rd Telecommunications Forum (TELFOR), Belgrade, pp. 79–82 (2015)
5. Choi, G., Kim, D., Yeom, I.: Efficient streaming over CoAP. In: 2016 International Conference on Information Networking (ICOIN), Kota Kinabalu, pp. 476–478 (2016). https://doi.org/10.1109/ICOIN.2016.7427163
6. Kavitha, B.C., Vallikannu, R.: IoT based intelligent industry monitoring system. In: 2019 6th International Conference on Signal Processing and Integrated Networks (SPIN), Noida, India, pp. 63–65 (2019). https://doi.org/10.1109/SPIN.2019.8711597
7. Puranik,V., Ranjan, S.A., Kumari, A.: Automation in Agriculture and IoT. In: 2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU), Ghaziabad, India, pp. 1–6 (2019). https://doi.org/10.1109/IoT-SIU.2019.8777619.
8. Tanganelli, G., Vallati, C., Mingozzi, E.: CoAPthon: Easy development of CoAP-based IoT applications with Python. In: 2015 IEEE 2nd World Forum on Internet of Things (WF-IoT), Milan, pp. 63–68 (2015).https://doi.org/10.1109/WF-IoT.2015.7389028
9. Yadav, E.P., Mittal, E.A., Yadav, H.: IoT: challenges and issues in Indian perspective. In: 2018 3rd International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU), Bhimtal, pp. 1–5 (2018). https://doi.org/10.1109/IoT-SIU.2018.8519869
10. Likith Bhushan, H.N., Niranjan, K.R.: An IP Based Patient Monitoring Smart System in Hospitals. Paper ID: IJSRDV4I120073, vol. 4, no. 12, pp. 154–156, 1 March 2017
11. Mishra, D., Khan, A., Tiwari, R., Upadhay, S.: Automated irrigation system-IoT based approach. In: 3rd International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU), Bhimtal, pp. 1–4 (2018). https://doi.org/10.1109/IoT-SIU.2018.8519886
12. Singh, M., Singh, A., Kim, S.: Blockchain: agame changer for securing IoT data. In: 2018 IEEE 4th World Forum on Internet of Things (WF-IoT), Singapore, pp. 51–55 (2018).https://doi.org/10.1109/WF-IoT.2018.8355182
13. Digvijaysinh, R.: Security analysis of constrained application protocol (CoAP): IoT protocol **6**(8), 37 (2017)
14. Zainudin, A., Syaifudin, M.F., Syahroni, N.: Design and implementation of node gateway with MQTT and CoAP protocol for IoT applications. In: ICITISEE, Yogyakarta, Indonesia, pp. 155–159 (2019). https://doi.org/10.1109/ICITISEE48480.2019.9003734
15. Kodali, R.K., Yatish Krishna Yogi, B., Sharan Sai, G.N., Honey Domma, J.: Implementation of home automation using CoAP. In: TENCON 2018 - 2018 IEEE Region 10 Conference, Jeju, Korea (South), pp. 1214–1218 (2018). https://doi.org/10.1109/TENCON.2018.8650135

16. Li, J., Wei, G.: Research on CoAP resource directory based on blockchain. In: 2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), Chongqing, China, pp. 1730–1735 (2020). https://doi.org/10.1109/ITNEC48623.2020.9084996
17. Teklemariam, G.K., Hoebeke, J., Van den Abeele, F., Moerman, I., Demeester, P.: Simple RESTful sensor application development model using CoAP. In: 39th Annual IEEE Conference on Local Computer Networks Workshops, Edmonton, AB, 2014, pp. 552–556 (2014). https://doi.org/10.1109/LCNW.2014.6927702
18. Roohullah, S., Fazlullah, K., Farman, U, Nazia, A, Muhammad, T.: USING CoAp protocol for resource observation in IOT. Int. J. Emerg. Technol. Comput. Sci. Electron. (IJETCSE) **21**(2), 385–388 (2016). ISSN: 0976-1353
19. Brasilino, L.R.B., Swany, M.: Mitigating DDoS Flooding attacks against IoT using custom hardware modules. In: 2019 Sixth International Conference on Internet of Things: Systems, Management and Security (IOTSMS), Granada, Spain, pp. 58–64 (2019). https://doi.org/10.1109/IOTSMS48152.2019.8939176
20. Gupta, A.K., Johari, R.: IOT based electrical device surveillance and control system. In: 2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU), Ghaziabad, India, pp. 1–5 (2019). https://doi.org/10.1109/IoT-SIU.2019.8777342
21. Lavanya, P., Sudha, R.: A study on WSN Based IoT application in agriculture. In: 2018 3rd International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, pp. 1046–1054 (2018). https://doi.org/10.1109/CESYS.2018.8724020
22. Kovatsch, M., Lanter, M., Shelby, Z.: Californium: scalable cloud services for the Internet of Things with CoAP. In: 2014 International Conference on the Internet of Things (IOT), Cambridge, MA, pp. 1–6 (2014). https://doi.org/10.1109/IOT.2014.7030106