Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 109C (2017) 568–575

Procedia Computer Science

www.elsevier.com/locate/procedia

The 8th International Conference on Ambient Systems, Networks and Technologies (ANT 2017)

Design and implementation of an IoT gateway to create smart environments

André Glória[a,*], Francisco Cercas[a,b], Nuno Souto[a,b]

[a] ISCTE-IUL, Av. das Forças Armadas, Lisbon, Portugal
[b] Instituto de Telecomunicações, Av. Rovisco Pais, 1, Lisbon, Portugal

Abstract

The paper presents a proposal of a practical implementation for an IoT gateway dedicated to real-time monitoring and remote control of a swimming pool. Based on a Raspberry Pi, the gateway allows bidirectional communication and data exchange between the user and the sensor network implemented on the environment using an Arduino.

Peer-review under responsibility of the Conference Program Chairs.

Keywords: Internet of Things; Sensor Networks; Smart Environments

1. Introduction

Nowadays the urge to connect everything to the Internet is growing, not just to send information to servers for processing and storage but also to provide full control of physical devices over the web.

While humans will continue to connect their devices to the Web in greater numbers, by 2020 more than 200 billion smart devices are expected to be connected to the Internet [1], making Machine-to-Machine (M2M) communications up to 45% of the whole Internet traffic [1,2,3].

Examples such as Smart Homes, where users can control their thermostats or lights with a smartphone, are the basis for Internet of Things (IoT). IoT was designed to play a great role improving our quality of live and its applications are present in many of our day to day experience such as transportation, health care and industrial automation.

IoT has the ability to transform a simple physical device into a smart one, using the embedded technology and computational power. Using the sensors and actuators available to guarantee the features of the device, it is possible to share that information between devices and put them to work together to improve the user experience. This will contribute to a bigger explosion coming from things connected to the Web that were not connected before, did not exist, or now use their connection as a core feature.

* Corresponding author.
E-mail address: afxga@iscte-iul.pt

Peer-review under responsibility of the Conference Program Chairs.
10.1016/j.procs.2017.05.343

The Internet is one of the most important developments of man kind and IoT will represent the next evolution of the Internet[2,3]. With the capability of gathering, analyzing and distributing the data, IoT consists in the connection between the Internet and a range of devices and sensors.

IoT, as shown in Fig. 1, can be divided into six elements[4] that help us understand its real meaning and functionality, i.e, *identification*, *sensing*, *communication*, *computation*, *services* and *semantics*.

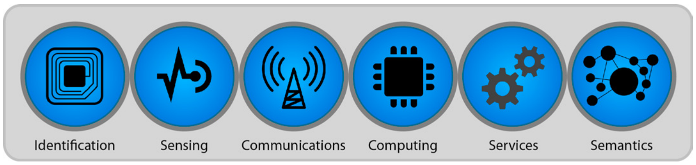

Fig. 1. IoT elements

IoT projects have the ability to do more than just connect the device to the Internet, they can be a big part of improving the efficiency or even adding new features such as Artificial Intelligence, transforming every common objects into connected one.

In this paper the authors describe the designing/developing of a system that can be applied to any object or environment with little or no modifications and easily used by any person. The system will provide full remote and secure control and monitoring of sensor networks, via an online platform, that can be applied to any non-smart object or environment allowing them to be connected to the Internet and to the user. With the possibility to add a set of rules and Artificial Intelligence it is possible to improve the efficiency leading to potential gains, such as energy or water savings. Although the main goal is to create a low cost system that is flexible in the sense that it can be easily adapted to any specification, in this paper the practical functionality will be tested and evaluated in a swimming pool, with an application capable of monitoring the water temperature and level, the environmental temperature, relative humidity, air pollution and luminosity and remotely control the water pumps and pool lights.

2. Related Work

IoT gateways[5] are dedicated hardware applications used to connect the user to the network, allowing the conversion of data between the short distance communication protocols to the traditional communication network. The gateway is supposed to support different types of sensor nodes, multiple communication protocols, both wireless or wired, and provide a set of unified information for the application or user, making these only responsible for data processing.

The main challenge on creating an IoT gateway is the lack of standards, being that each sensor node can communicate with a different protocol that is not compatible for others. This makes the development of a general purpose gateway a complicated task, which explains why it is common to find gateways developed for specific applications. Nevertheless, all have the same key requirements: low-cost hardware, easy implementation and extensibility and an application layer support.

R. Gerstendorf in his article[6] examines ten platforms for creating smart homes, from ZigBee to the Apple HomeKit. All ten are market solutions available to consumers as a ready-to-use product in order to control smart devices through a gateway. Each one has a communication protocol, proprietary characteristics and a range of pros and cons.

In the literature it is possible to find several proposals[7,8,9] for IoT gateways implemented using low-cost hardware devices, such as Arduino and Raspberry Pi. Most of them use these devices to support the web server, which difficults its access from outside the network. Other solutions were found[10,11], that use wireless communication protocols for specific applications and little to none IoT gateways were designed to use wired protocols. Only one of these[12] did it, where the authors used the gateway with the RS485 protocol to control end-devices from the Internet. This makes all of these solutions limited in flexibility and adaptation to other environments.

The literature presented a similar concept for a multi-communication protocol IoT gateway. In this[13], the authors present a heterogeneous IoT gateway capable of using the same board to communicate with multiple wireless protocols as well as support for a large amount of communication buses, in a modularity basis.

3. System Architecture

To maintain the normal requirements for IoT gateways and in order to be able to use multiple communication protocols, adapting them to user requirements, the authors of this paper created a gateway based on common requirements and capable of using every sensor and communication protocol, both wired and wireless, that is available. The system relies on a group of devices connected over a network that is controlled by an application using a given communication protocol. Fig. 2 presents a high level system architecture and, as shown, the system is composed by hardware, software and the implemented communication features. The proposed system was conceived using hardware capable of supporting multiple communication protocols and flexible enough so as to run adaptable software, as described in the following sections.

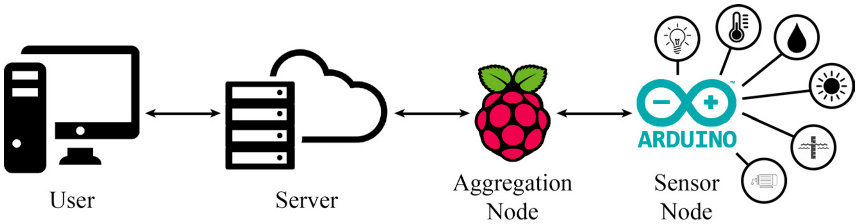

Fig. 2. System Architecture

3.1. Hardware

As displayed in Fig. 2 and 3, the hardware components adopted for our implementation consist in a Raspberry Pi 3 [14], a low-cost credit card sized computer with enough processing power and memory, that support programmable I/O ports and the use of standard peripherals, as an aggregation node or gateway; an Arduino Uno [15], an open-source electronics single board based on a simple input/output capable of interfacing different peripherals, sensor and wireless communication devices, as a node sensor that collects information from temperature, humidity, luminosity and water level. Also a set of relays are used to control numerous actuators such as lights and motors.

The aggregation node are not responsible for reading sensors, they just provide a gateway between the user and the Sensor Network and also perform some data analyses. The sensor node, the lowest level of a Sensor Network, is responsible for gather information from sensors, perform user actions and use communication mechanisms to send data to the aggregation node.

3.2. Software

As presented in Figure 2 there are several pieces of software that are essential to the proposed architecture. They are divided into two sections that are mentioned below.

3.2.1. Web Platform

The web platform consists on a display for the information gathered from the sensor network. As can be seen in Fig. 4 in the platform the user can choose which sensors to display, as well as the current values and past data retrieved, turn on and off all the actuators and switch between functionality modes. The web platform also provides the user a bridge to the sensor network using the MQTT protocol, with the Eclipse Paho JavaScript Client [16]. It is hosted on a private server, containing a private MySQL database for storage and was built using HTML, PHP, Javascript, CSS and AJAX for both control and communication.

3.2.2. Python Scripts

Present in the aggregation node, this scripts are responsible for receiving the data from the sensor node and send them to the server using the MQTT protocol. They are also responsible for receiving commands sent from the user through the web platform and pass them to the sensor node. These scripts use the Eclipse Paho Python Client [16]. They

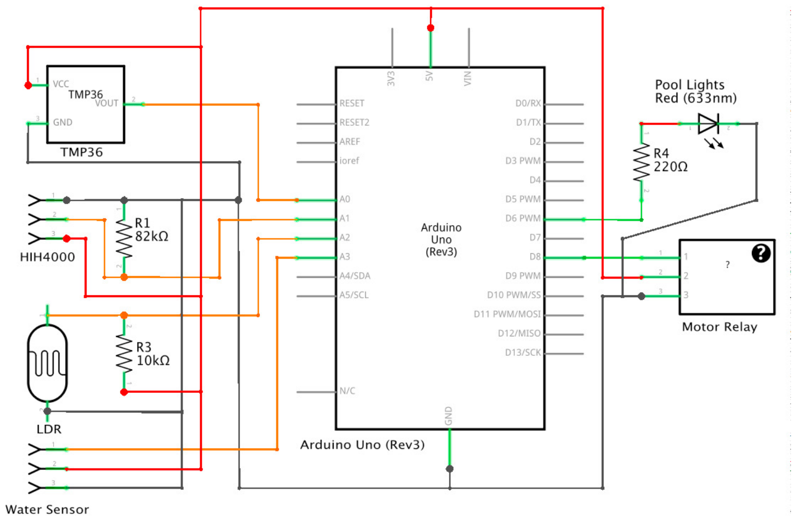

Fig. 3. Hardware connections

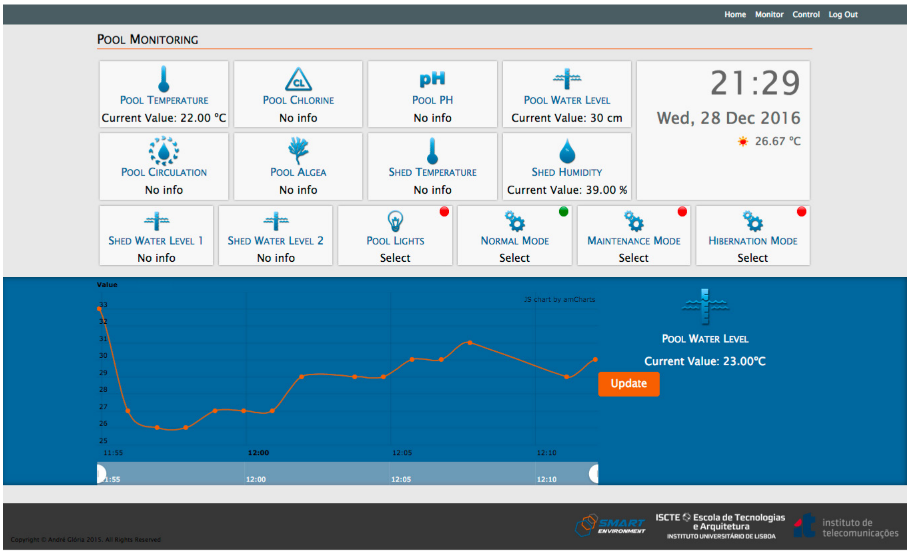

Fig. 4. Screenshot of the web platform

represent as well the MQTT broker, in which the web platform and the client scripts will connect, using the Mosquitto MQTT broker[17].

In Fig. 5 and 6 it is possible to see the Activity Diagram with the process needed to initiate the system, with the authentication of both network and user, including the initial message exchange for the online platform configuration as well as the data and action exchange process in order to send information from the Sensor Network to the user and actions from the user to the Sensor Network.

3.3. Communications protocols

As mentioned before, communication is one of the main elements of IoT. In the system there are two key communication points, between the server and the aggregation node and from the this last to the sensor node. In the first, the need was to have an open-source, low bandwidth consumption and good performance application layer Internet

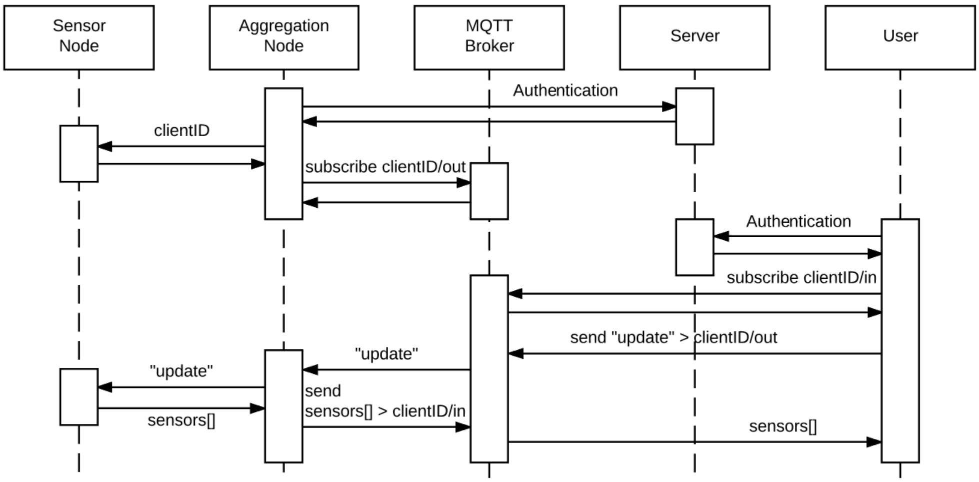

Fig. 5. Network and User authentication process and initial process

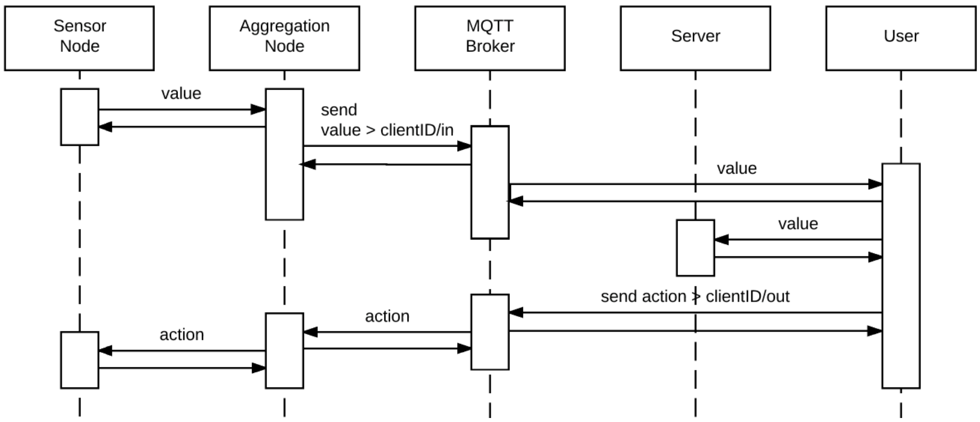

Fig. 6. Data and Actions exchange

protocol that can also connect a JavaScript Client to a Python Client, so the communications are done using MQTT, a protocol design for IoT projects. In the second, there is a range of options with both wired or wireless capabilities. In here we search for a low-cost, low-power, high range, multi-node communication protocol with low hardware complexity. The protocols that adapt to the specification range from both environments, like I2C, RS485, ZigBee or LoRaWAN but to ensure a bigger reliability and an easy connection the communications are done using Serial Communication in the form of an USB cable.

3.3.1. Universal Serial Bus

USB was developed in order to facilitate the connection of peripheral devices to a computer. The propose of USB was to give the ability to devices to be plug and play with a single interface and automatic configuration. Consists in a communication bus used for connecting and power devices, such as micro-controllers from other interfaces. USB operates at 5 volts using 4 different lines: Power, Ground and a twisted pair of data lines using NRZI encoding[18,19]. Providing a fast Master/Slave interface capable of supporting up to 127 devices with up to 6 hubs, USB uses an enumeration process where each slave is assigned with a unique address and give the master information about which speed will be used and what type of data will be transferring.[19].

3.3.2. Message Queuing Telemetry Transport

MQTT is a messaging protocol that aims at connecting embedded devices and networks with applications and middleware[4,20,21]. Built on top of the TCP protocol, uses a publish/subscribe pattern, with a routing mechanism (one-to-one, one-to-many, many-to-many), to provide flexibility and simplicity transition making MQTT an optimal connection protocol for the IoT and M2M being suitable for small, cheap, low power and low memory devices with low bandwidth networks. Compared to HTTP, MQTT is designed to have a lower protocol overhead.

MQTT consist of three components: subscriber, publisher and broker. A device registers as a subscriber for specific topics of interest in order to get the information publish to that topic. The publisher acts as a generator of data for some topic, transmitting that information to the subscriber through the broker.

4. Results

The developed gateway can be seen in Fig. 7, 8, 9 and 10 with every hardware feature and communication protocol used.

In Fig. 7 we see the whole setup, with the online platform on the laptop, showing the data retrieved from the sensor network. It is possible to see that the data is shown at the respective place, that the user has the possibility to turn the lights on and of and change between modes (Normal Mode turned on, green on the platform and first light on the sensor node).

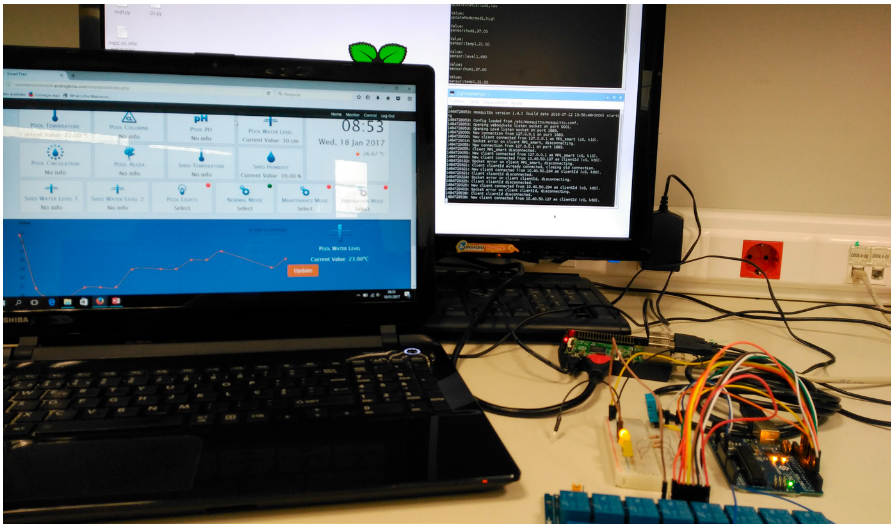

Fig. 7. Functional prototype setup

In Fig. 8 it is possible to see the MQTT broker log showing the client and the network connecting and disconnecting. Without these results the communication over MQTT is not possible, since both parts need to be connected to the broker in order to receive the messages from the topics they subscribed previously. Fig. 9 shows the script that controls the aggregation node with the response to the *update* message, data transmission and actions received. When the aggregation node receives the *update* message it responds with all the sensors and actuators present in the network and after the successful connection it receives the user chosen mode and light command. Also it is possible to see that the data retrieved from the sensor node over USB are successfully received in the aggregation node and then sent to the platform.

Fig. 10 demonstrates the hardware setup, with the nodes connected with USB, the sensors connected to the sensor node and the actuator working with user specifications. As said before the lights (red led) are off and the normal mode lights (first yellow led) are on, both selected by the user on the platform. The sensor values retrieved are accordingly with the expected ones, with the temperature at 22°C and humidity at 39% in a room at 25°C and low humidity. With these results we conclude that the sensor node is capable of obtaining the values, send them to the aggregation node and also perform the action that the user send using the online platform.

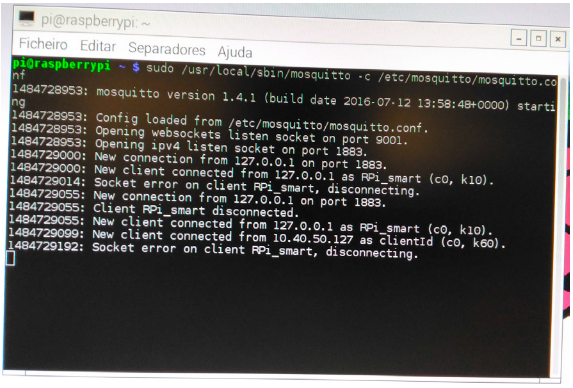

Fig. 8. MQTT Broker script

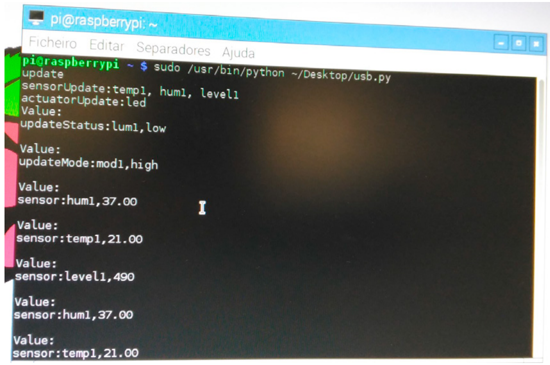

Fig. 9. Aggregation node script

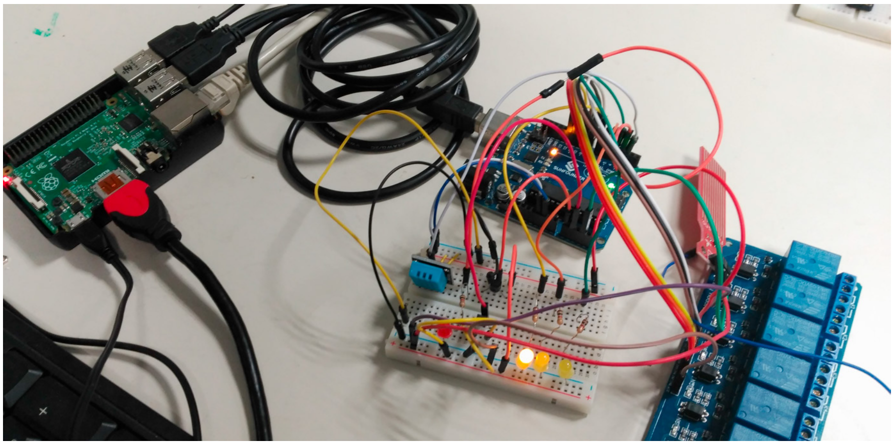

Fig. 10. Functional prototype hardware setup

5. Conclusions

In this paper, an IoT gateway for creating a smart swimming pool was designed, consisting in a Raspberry Pi acting as an aggregation node, an Arduino with a set of sensors as a sensor node and a web platform for monitor and control the network. The developed system so far, accomplishes the proposal features and purpose, giving the user the ability to control the environment remotely. Nevertheless more tests to the system in multiple situations can be done in order to prove the efficiency of the developed architecture.

In the future, the authors of this publication are planning to compare multiple communications protocols for each point of communication, or even offer a solution supporting multiple communication schemes. Furthermore, additional functionalities will be considered for the system, such as the possibility to create sets of rules directly from the platform, and add Artificial Intelligence or Machine Learning. At last the focus will be towards a more secure system, with the implementation of encrypted communications and a SSL protocol and certificate on web server side.

Acknowledgments

The authors would like to thank Professor Octavian Postolache for providing the sensors. This work was partially supported by the FCT - Fundação para a Ciência e Tecnologia and Instituto de Telecomunicações under project UID/EEA/50008/2013.

References

1. J. Gantz, D. Reinsel, THE DIGITAL UNIVERSE IN 2020: Big Data, Bigger Digital Shadows, and Biggest Growth in the Far East, IDC iView: IDC Anal. Future 2007 (2012) 1–16.
2. D. Evans, The Internet of Things How the Next Evolution of the Internet Is Changing Everything, CISCO, San Jose, CA, USA, White Paper.
3. S. Taylor, The Next Generation of the Internet Revolutionizing the Way We Work, Live, Play, and Learn, CISCO, San Francisco, CA, USA, CISCO Point of View.
4. A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, M. Ayyash, Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications, IEEE Communications Surveys and Tutorialsdoi:10.1109/COMST.2015.2444095.
5. C. L. Zhong, Z. Zhu, R. G. Huang, Study on the IOT architecture and gateway technology, in: Proceedings - 14th International Symposium on Distributed Computing and Applications for Business, Engineering and Science, DCABES 2015, 2016. doi:10.1109/DCABES.2015.56.
6. R. Gerstendorf, Ten platforms examined (9 2016).
7. A. Grygoruk, J. Legierski, IoT gateway – implementation proposal based on Arduino board, Federated Conference on Computer Science and Information Systems (FedCSIS)doi:10.15439/2016F283.
8. S. M. Kim, H. S. Choi, W. S. Rhee, IoT home gateway for auto-configuration and management of MQTT devices, in: 2015 IEEE Conference on Wireless Sensors, ICWiSE 2015, 2015. doi:10.1109/ICWISE.2015.7380346.
9. R. Balakrishnan, IoT based Monitoring and Control System for Home Automation, Global Conference on Communication Technologies (GCCT).
10. T. Zachariah, N. Klugman, B. Campbell, J. Adkins, N. Jackson, P. Dutta, The Internet of Things Has a Gateway Problemdoi:10.1145/2699343.2699344.
URL `http://dx.doi.org/10.1145/2699343.2699344.`
11. S. Guoqiang, C. Yanming, Z. Chao, Z. Yanxu, Design and implementation of a smart IoT gateway, in: Proceedings - 2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing, GreenCom-iThings-CPSCom 2013, 2013. doi:10.1109/GreenCom-iThings-CPSCom.2013.130.
12. M. Dong, X. Zeng, S. Bi, S. Guo, Design and implementation of the multi-channel RS485 IOT gateway, in: Proceedings - 2012 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems, CYBER 2012, 2012. doi:10.1109/CYBER.2012.6392581.
13. E. Gioia, P. Passaro, M. Petracca, AMBER: an advanced gateway solution to support heterogeneous IoT technologies, 24th International Conference on Software, Telecommunications and Computer Networks (SoftCOM).
14. C. Bell, Beginning Sensor Networks with Arduino and Raspberry Pi, 1st Edition, Apress, 2013.
15. M. Banzi, Geting started with Arduino, 2nd Edition, Make:Books, 2011.
16. Eclipse Foundation Inc., Paho - Open Source messaging for M2M (2016).
URL `https://eclipse.org/paho/`
17. Eclipse Foundation Inc., Mosquitto - An Open Source MQTT broker (2016).
URL `https://mosquitto.org/`
18. M. Sharma, N. Agarwal, S. Reddy, Design and Development of Daughter Board for USB-UART Communication between Raspberry Pi and PC, International Conference on Computing, Communication & Automation.
19. Computer-solutions.co.uk, USB - a brief tutorial for embedded engineers (2015).
URL `http://www.computer- solutions.co.uk/info/Embedded_tutorials/usb_tutorial.htm`
20. D. Locke, MQ Telemetry Transport (MQTT) v3.1 protocol specification (2010).
URL `https://www.ibm.com/developerworks/webservices/library/ws-mqtt/index.html`
21. N. De Caro, W. Colitti, K. Steenhaut, G. Mangino, G. Reali, Comparison of two lightweight protocols for smartphone-based sensing, in: IEEE SCVT 2013 - Proceedings of 20th IEEE Symposium on Communications and Vehicular Technology in the BeNeLux, 2013. doi:10.1109/SCVT.2013.6735994.