Ministerul Educației Naționale
ROMÂNIA
UNIVERSITATEA DE MEDICINĂ, FARMACIE, ȘTIINȚE ȘI TEHNOLOGIE DIN TÂRGU MUREȘ

Scientific Bulletin of the „Petru Maior" University of Târgu Mureș
Vol. 15 (XXXII) no. 2, 2018
ISSN-L 1841-9267 (Print), ISSN 2285-438X (Online), ISSN 2286-3184 (CD-ROM)

INTERNET OF THINGS BASED ON HTTP

Traian TURC
University of Medicine, Pharmacy, Sciences and Technology of Tîrgu Mureș
Gheorghe Marinescu Street, no. 38, 540139 Tîrgu Mureș, Romania
traian.turc@ing.upm.ro

Abstract

This paper proposes using HTTP in the context of IoT, as a liaison between an ever-growing number of devices, standards and requirements. IoT devices are often resource-limited and do not support complex protocols. This study was made on an IoT device that implements a minimal subset of HTTP, namely connection to a URL and reading the data received from the URL. IoT devices do not communicate directly among each other; instead, they talk to a web application with database access, which stores the data they collect.

Key words: Internet of Things, HTTP, programming, web sites

.

1. Introduction

IOT is increasingly being recognized by researchers and analyst as one of the most sophisticate technology that has the potential of innovation. It consists of sensors, actuators, computing device and data communication capabilities [1].

Following the vision of an Internet of Things (IoT) real world objects are integrated into the Internet to provide data as sensors and to manipulate the real world as actors. While current IoT approaches focus on the integration of things based on service technologies, scenarios in domains like smart cities, automotive or crisis management require service platforms involving real world objects, backed-systems and mobile devices [2].

Things in IoT will interact with each other and with human beings through a myriad of communication technologies, often wireless, and almost always subject to interference, corruption, eaves dropping and all kinds of cybernetic attacks [3].

Today the main focus of researchers is to meet the growing communication needs of modern-day society. Reliable data transmission with improved efficiency is among one of the most challenging goals of today's era. [4]

Recent advances in IoT technology paves the path to developing smart systems that can automatically track goods/objects and people [5].

In today's Information Technology driven society, automation systems are making people's life easier and comfortable than ever before. Internet of Things is one such technology that is widely being applied for automating several routine activities. The smart home system enabled by IOT is one of the trends that is gaining momentum. Internet of Things connects living and non living things through internet. IOT model views everything as a smart object and enables communication between them. [6]

With recent trends and developments in the area of communication and information technologies and the emergence of the Internet of Things, one can find a variety of small Internet connected devices [7].

Most IoT devices of today are simple devices such as sensors and actuators that are dependent on controllers in the cloud for analysis of recorded data, logical reasoning, and management. This may cause potential latency issues where the Internet connections of devices could become a bottleneck for performance and cause the devices to be sensitive to connectivity problems [8].

Internet of Things (IOT) mainly involves in integrating the data generating objects called as sensors which show continuous stream of data and are capable of using Internet as a main communication for data processing which acts as a database as a service [9].

Internet of things (IoT) is a connecting a physical smart device and providing a information and also connecting a human to human relationships. In today's world of disconnectedness, people are becoming accustomed to easy access to information [10].

The paper contains a description of the minimal structure of an IoT for case study, the minimum set of HTTP methods implemented in IoT, the description of the server application, the client application and the database structure.

In the first part, the IoT device manages a single value acquired from a photovoltaic sensor. A series of

IoT inputs and outputs are then added. Within the client application, two complex objects for managing multiple inputs and outputs are created and used.

2. IoT

The IoT (Internet of Things) device is based on an Arduino UNO board, to which we attached an 8266 Wi-Fi module (Figure 1)

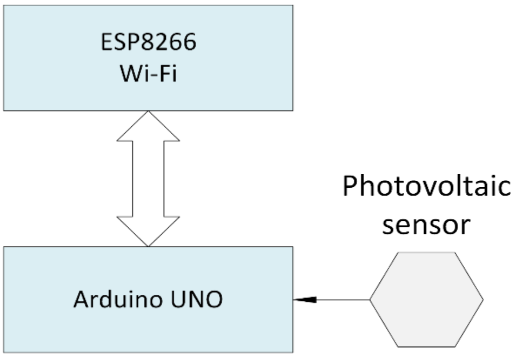

Fig. 1: IoT device

For the case study, a minimal structure was made from an Arduino Uno controller and a Wi-Fi module and a photovoltaic sensor (Figure 2).

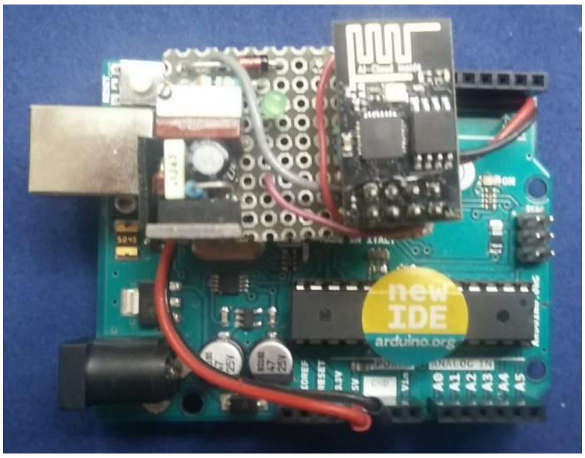

Fig. 2: Case study board

3. Using HTTP

We upload an application implementing a minimal HTTP (Hyper Text transfer protocol) API (Application Programming Interface) onto the Arduino board. The most notable method is GET.

The electric voltage coming from the photovoltaic sensor is read from an analog input by the Arduino controller and sent to the server via the Wi-Fi module using the HTTP protocol.

The connection to a WAP (Wireless Access Point) requires the name of the Wi-Fi network (“Access_point_name”) and a password (“passwd”).

```
sendData(“Access_point_name”, “passwd”)
```

Once the board is connected to WiFi, it can then connect to the server:

```
sendData(“http://server_name.dom”)
```

4. Sending values to the server application

Once a connection to the server is established, the next step is to invoke the GET method to send the value measured by the sensor (“val_senz”) to the server application (“server_apl”):

```
sendData(“GET/url/server_apl.php
id=cod_senz&a0=val_senz”)
```

As shown above, the value read by the sensor ("val_senz") is sent via a URL parameter. The server application ("server_apl") identifies the sensor based on its ID ("cod_senz") and stores the value in a database to be queried later by the client.

For several attached sensors, to transfer values val_a0, val_a1, val_a2, val_a3, val_a4, val_digi, the GET method will take the form:

```
sendData(“GET/url/server_apl.php?id=cod_senz
&a0=val_a0&a1=val_a1&a2=val_a2&a3=val_a3&a
=val_a4 &digi=val_digi ";
```

Analogical values val_a0, val_a1, val_a2, val_a3, val_a4, come from five potentiometers simulating 5 analogical inputs, “val_digi” is a value representing the binary code of the four switches and “val_com” is a value representing the binary code for 8 digital commands. (Figure 3).

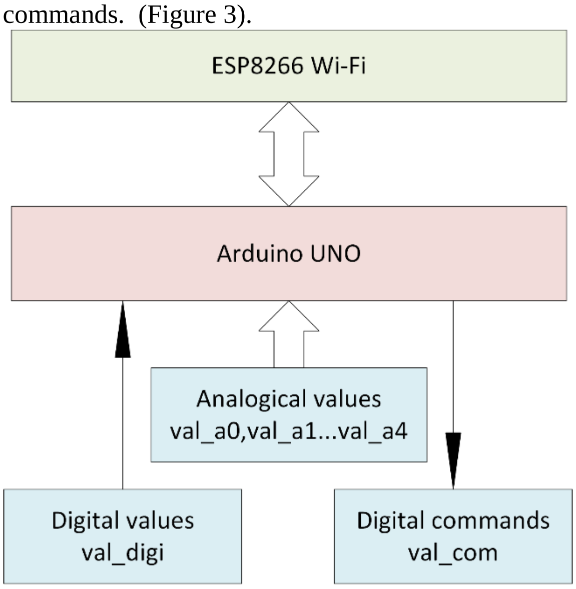

Fig. 3: IoT device multi i/o

5. Data base structure.

Data collected from the IoT device is transferred to the server application using the GET method and stored in the database. The database contains multiple tables, including "sensors"; its structure is shown in Figure 4.

This table contains information like the sensor

ID (id_mas), the sample number (nr_mas), the date of the same acquisition (dat_mas), the measured value of the sample (val_mas), and the command code (val_cda).

The command code sent by the application to the IoT device is decoded and executed locally. The result is returned back to the application, to confirm the successful execution.

#	Nume	Tip
1	id_mas	int(11)
2	nr_mas	bigint(11)
3	dat_mas	datetime
4	val_mas	int(11)
5	val_cda	int(20)
6	exec_cda	int(11)

Fig. 4: Sensors table

6. Client application

The IoT performance is largely ensured by the client application. The speed running of the client application significantly influences the reaction time of the IoT.

The client application must always interact with IoT both for reading the values acquired by it and for transferring orders to IoT

We experimented with various setups for the client applications and found that the most efficient solution combines AJAX and JavaScript.

For the efficient implementation of the client application, a number of classes have been developed and by instantiating them, the necessary interacting objects with IoT are obtained.

For example, the "digi_com" object is used to read and transfer data to and from IoT. It has a number of methods that make it easy to interact with IoT.

The object "digi_com" (Figure 6) stores the binary values specified by the user. After being converted to decimal values ("val_cda"), they are stored in the database. The "digi_com" object calls a server application to access the database (Figure 5).

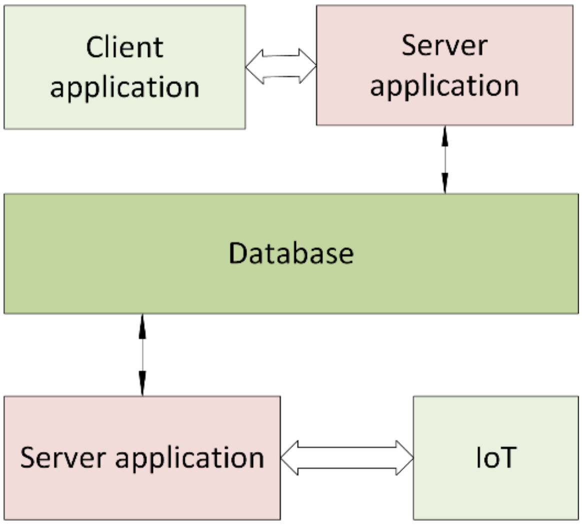

Fig. 5: Communication between client application and IoT

The IoT device reads the "val_cda" value from the database, using the server application; this value will be decoded and executed. The IoT device sends feedback to the server via "exec_cda". By comparing it against "val_cda", the client application can figure out whether execution was successful (Figure 6).

The client application must be executed in a loop, so that the data displayed is constantly up-to-date. Also, once a digital command is inputted, it must be sent without an explicit confirmation from the user.

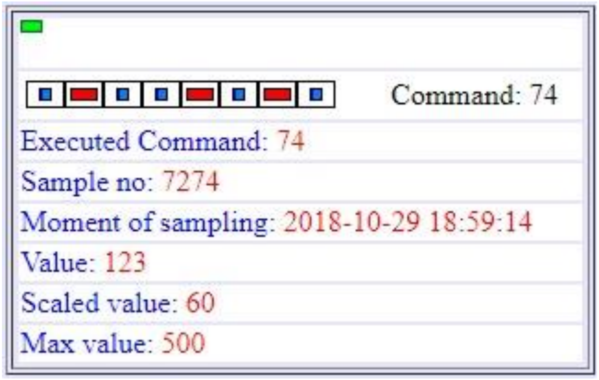

Fig. 6: The "digi-com" object

Continuous display of aquired values was possible through the use of Ajax technology that could continually bring content without reloading the web page that actually constitutes the client application.

The "digi_com" object contains also functionality to display the analog value "val_mas", taking into account the range of possible values ("max_value). See Figure 7.

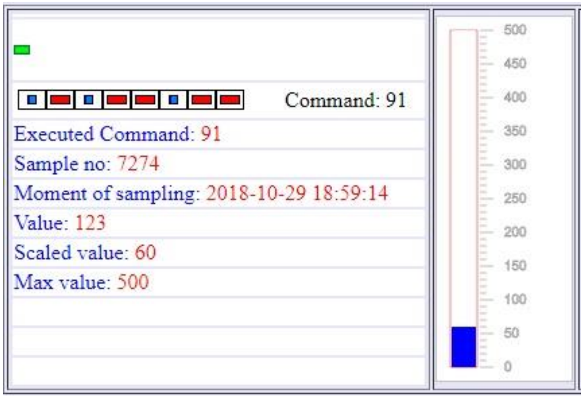

Fig. 7: The "digi-com" object for analogue values

There is also functionality to display trends showing the evolution of a certain value across time. We compared two methods: one on the client side, and one on the server side. The former proved to be the fastest. However, its main disadvantage is that it required a local database that had to be updated continuously. The server method made the opposite trade-off: it did not require running it continuously, but still ended up being slower.

7. Conclusions

This paper proposes a minimal implementation of the HTTP protocol in order to facilitate the communication between an IoT device and a server. We built a series of applications that connect IoT devices to a server. It would be ideal to build a single application that talks to multiple IoT clients concurrently. However, IoT devices are resource-limited and do not support complex protocols.

Our solution proposes an intermediary database that facilitates the communication between IoT devices and clients. The only way clients can interact with the IoT device is through this local database, which allows concurrent access. Additionally, the data collected by the IoT device is saved in a database, allowing data trends to be displayed.

References

[1] Shinde, S. and Phalle, V. (2017), *A Survey Paper on Internet of Things based Healthcare System*, International Advanced Research Journal in Science, Engineering and Technology, vol. 4, Special Issue 4, January 2017.

[2] Sven, B., Thomas, S., Daniel; S., Alexander, S., Ralf, A., and Michael, A. (2013), *A Service Infrastructure for the Internet of Things based on XMPP*, ser. Conference Paper · March 2013 DOI: 10.1109/PerComW.2013.6529522Zhang, S., Zhu, C., Sin, J. K. and Mok, P. K. T. (2002), A novel ultrathin elevated channel low-temperature poly-Si TFT, *IEEE Electron Device Lett.*, vol. 20, pp. 569–571.

[3] Antonio, A., Rodrigo, V., and Leonardo, M. (2011), *A Trustful Infrastructure for the Internet of Things Based on EPOSMote.*, Conference Paper · December 2011 DOI: 10.1109/DASC.2011.35 · Source: DBLP

[4] Ayesha H., Hafsa, A., Warda, A., Hareem A., Yasar A., and Hannu, T. (2017), *Internet-of-things based Smart Tracking*, Conference Paper · May 2017 DOI: 10.1109/C-CODE.2017.7918899.

[5] Catarinucci L. (2015), *An IoT-Aware Architecture for Smart Healthcare Systems*, in IEEE Internet of Things Journal, vol. 2, no. 6, pp. 515-526, Dec. 2015.

[6] Maheshwari and D., Umesh, I.(2017), *A Study on Internet of Things based Smart*, Article · August 2017.

[7] Whitmore, A. Agarwal, A. and Da Xu, L.(2014), *The Internet of Things - A survey of topics and trends*, Information Systems Frontiers, vol. 17, no. 2, pp. 261-274.

[8] Rahim, R., Yuhong, L., and Theo, K.(2015), *A Scalable Distriubuted Ledger for Internet of Things based on Edge Computing*, Conference Paper · October 2018.

[9] Athul, P. (2015), *Implementation of an IoT(Internet of Things) based system*, Article, April 2015.

[10] Lavanya, G., Deepika, N, Sangeetha, T., Maheshwari, R., Josephine, R (2018), *Internet of Things Based Notifications Using Smart Notice Board*, International Journal of Pure and Applied Mathematics, vol. 119, no. 10, pp. 1915-1920, ISSN 1311-8080.