Telemetry for domestic water consumption based on IoT and open standards

Sayonara A. C. Tavares, Ricardo J. B. V. M. Cavalcanti
Diego R. C. Silva, Marcelo B. Nogueira and Marconi C. Rodrigues
Escola de Ciências e Tecnologia
Universidade Federal do Rio Grande do Norte
Natal, Rio Grande do Norte 59078-275
Emails: sayonaracirilo@bct.ect.ufrn.br, ricavalcanti@bct.ect.ufrn.br,
diego@ect.ufrn.br, marcelonogueira@ect.ufrn.br, marconicamara@ect.ufrn.br

***Abstract*—In order to avoid wasting natural and financial resources, water consumption in homes and industries has been the target of consciousness and reduction policies. This paper describes the development of a microcontrolled system for telemetric measurement of water consumption using open sources and following the increasing trend of Internet of Things. The proposed system collects data from a water meter, which is equipped with a digital sensor, and sends such information over the internet to a server, dispensing manual reading of the water meter. This method allows live monitoring of water consumption, as well as providing identification of leaks and consumption patterns. The validation of the system was based on tests held at the laboratory, where it reproduced the failures of a real environment.**

I. Introduction

The constant development of the world's technologies and the way they become more accessible, enable a transformation in living standards of society. Nowadays, the world is raisin for the fourth industrial revolution [1], that impact the manufacture, work, and finally, the way of life. In this new era, where information has its origin in many types of sensors and travels on physical means with different peculiarities, the challenge is to efficiently process a large amount of data in order to extract useful information. This trend has been called Big Data in the literature [2]. Processing the data in real time and automatically is already possible, since sensors and actuators interact which one each other and exchange information through the wired or wireless network wherein times, to connect, the devices use the Internet Protocol (IP), that is called Internet of Things or IoT [3]. Given this context, advances in electronics and information technologies have been increasing its support to various areas and enabling improvements in efficiency of several productive sectors, where the use of electronic devices makes possible to optimize tasks that are still performed in a precarious and manual way.

Naturally, the technological advances take on the daily tasks gradually. However, for several reasons such as: lack of an appropriate solution; costs associated with modernization; and lack of automation support structure, some services are not yet automated. The current form of reading the domestic water consumption by water and sewage companies is an example of this. In addition to the human resources intended to collect data from each water meter, the quality of this reading can be compromised, since the process requires human precision and is a repetitive work. Also, this measuring system makes recognition of a possible leakage more difficult. The focus of this work is the description of the automation of this task, proposing a telemetry solution, which consists of remote data monitoring.

Faced with this proposal, it is possible to eliminate reading made in person, besides making this data available near real time. In addition, the user also can check the measurement history remotely, associated with other benefits such as: more frequent and reliable readings. The automated measurement system presented in this article uses a microcontroller that collects consumption data from a pulsed water meter. This information will be transmitted over the Internet to a server, where the data will be stored and analyzed, providing the total volume consumed dynamically for the consumer and the distributor. The diagram simplified in Figure 1 shows this system.

To implement the proposed system it is necessary:

- An energy outlet near the place of installation (future versions might use batteries);
- Water meter with pulsed output;
- Wireless access point near the microcontroller.

Similar works

The SMART AQUA METER-SAM introduce a smart electronic measurement system. It automates the billing system using a low power wireless technology, ZigBee. Water consumption is monitored in real time and the data is transmitted to the data central via ZigBee for storage, billing and maintenance purposes. A human machine interface is designed using NI LabVIEW [4].

A system of measurement and control of water consumption using wireless sensor network. The system aims to educate the user presenting the daily graph for separate rooms in houses or industries. In addition, the system can help detect leaks [5].

This paper proposes alternatives to promote the rational consumption of water. The project presents a solution that adopts: the **Wi-Fi** protocol - because it is a wireless communication network that has several advantages such as: mobility,

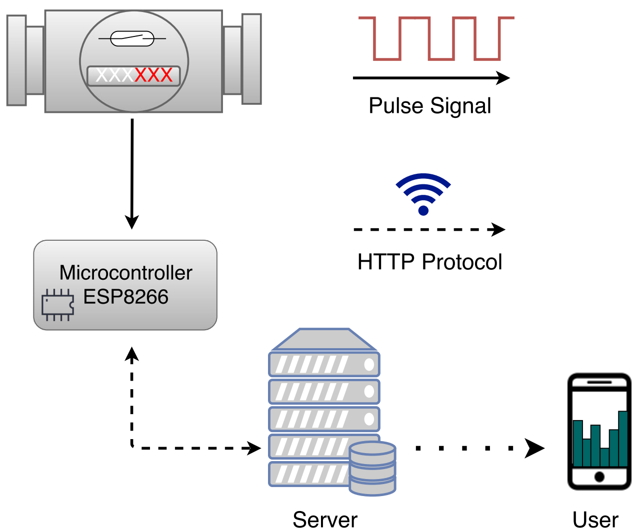

Figure 1. Microcontrolled system diagram

compatibility, sharing, accessibility, low maintenance cost, security and popularity [6]; and the protocol **HTTP** - for data transfer. The system presented in [4] differs from this since it is a solution to monitor water consumption in time based on ZigBee technology as well as [5]

II. Methodology

Usually, the monitoring of domestic water consumption is done by reading measuring equipment called water meters responsible for measuring the volume of water flowing through a pipe. This equipment totalizes the volume of water consumed and its analyze is carried out by water and sewage companies. Considering that with traditional water meters, the data reading should be done in person by an employee through a periodically visit to each residence, and in some cases, water meters are installed in places with difficult access (small, humid and poorly lighted), inadequate for this type of reading, the use of this measuring system makes recognition of a possible leak more difficult. It also compromises the quality of the reading, since the process requires human precision and it is a repetitive work, contributing to the increase of water waste. Hence, it is perceived that there will be gains in automating this process. For this purpose, traditional meters can be connected to other devices in order to process and send the information eliminating repetitive presential work.

In this work a water meter with a Reed Switch [7] sensor was used as shown in the Figure 2. The sensor is activated by a magnetic field applied on it (a simple magnet, fixed to a disc that rotates as water flows), generating an electrical signal (pulse) that is used in the electronic measurement of the volume of water. To capture, process and send the pulses, the ESP8266 microcontroller, developed by Espressif Systems [8] was chosen, since it has an IEEE 802.11 b/g/n (Wi-Fi) standard communication interface, which is used to communicate and send generated data for the server through a Web Service (see section II-D), since the standard allows a connection to access points, dispensing the need for a direct connection between the devices.

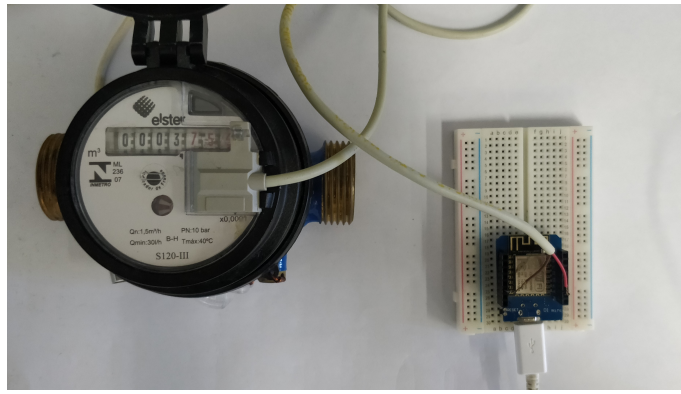

Figure 2. Water meter with a Reed Switch (left) connected to the ESP8266 microcontroller (right).

In addition, the ESP8266 used (Wemos D1 mini) has ports that work with the interrupt technology to read the pulses, a flash memory of 4M bytes for the storage of connection information and data that have not yet been sent to the database, which can be used to ensure the integrity of the same and the information registered at the first access to the network, even after a power outage or chip shutdown. Another fact that influenced the choice of the microcontroller was its low cost compared with other competitors such as the Arduino [9], which has lower performance and less memory, and does not have integrated Wi-Fi.

A. *Pulse analysis*

The sensor will generate a pulse for each cubic decimeter of volume passed through the water meter. Physically, a magnet, which is coupled to the tenth of a liter gear of the water meter by default, attracts the iron blades present in the sensor causing a short circuit, consequently generating an electric pulse. So, the duration of this pulse depends directly on the water flow on the system. When doing the study of the signal generated by the sensor on a digital oscilloscope, it is observed that it is subject to noise, called the bouncing effect (Figure 3). Often encountered in pushing keys as buttons, this effect is caused by undue oscillations at the closing of the mechanical contact of the key, which also occurs for the Reed Switch. In view of such a problem, the value corresponding to the minimum duration of a pulse is found, in which only the maximum flow rate that a water meter can operate satisfactorily for a short period of time without deterioration is considered. In this case the Elster, type S12-III of class B, has a maximum flow rate of $3m^3/h$ 0.83 l/s, in other words, when the device works at maximum flow a liter requires 1200ms. Knowing that the magnet has the size of $1/10$ of a liter gear perimeter, we can compute the minimum pulse witdth:

$$\Delta t = 120ms \tag{1}$$

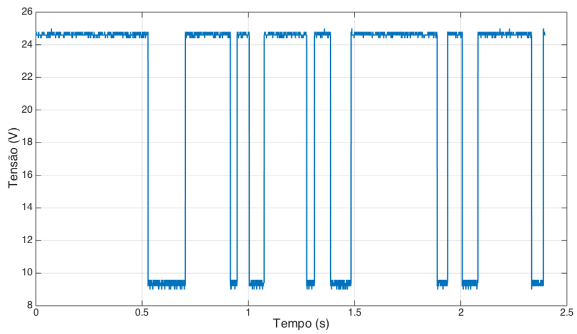

Figure 3. Pulse signal generated by sensor

In addition, using a debouncing technique, a software filter is implemented in the microcontroller firmware. In this sense, the technique used consists of comparing the time elapsed between the call of two interruptions (used to detect the pulses, as will be detailed in Section II-B) with a predetermined time Δt, which in this case will be the minimum pulse duration at the maximum flow, documented for each class of water meter, with a safety margin up. So, the counter of the number of pulses will only be incremented if the elapsed time has been greater than Δt.

B. Pulse reading

The microcontroller has two standard functions: the setup, where normally the initial settings are found that will run only once when powering up; and the main loop that executes infinitely. To read the pulses, the use of the interrupt was seen as the best way, because with this feature you can call the pulse counting function as a priority. This mechanism interrupts the current processing of the microcontroller and performs a specified task (called interrupt service routine or ISR); once the ISR is complete, the processor resumes execution of the loop from the point where it was interrupted. There are two types of interruption, internal (activated by timers) and external (external signals), which is our case.

External interrupts can be activated when the external signal rises, lowers or in both cases. We used the last one, called change mode (interruption in the rise and fall of the pulse) because there is noise in both cycle's parts of the pulse. If the rising mode was used (interruption in the rise of the pulse), for example, the noise present in the falling of the signal would cause a rise in the signal that could not be filtered. This means that we count 2 pulses per liter. The treatment of this double counting is done at the time of sending the data to Web Service (see section II-D).

C. Data structure - Queue

The packages with data related to the water meter are sent periodically. After a predefined time interval, one of the programmed internal interrupts in the microcontroller calls the function that sends the data to the server. However, in preliminary tests, the loss of information referred to connection problems was observed. In this way, the best solution found for the problem was the implementation of a queue in the firmware. This data structure enables to store and remove objects in a way that when there is a removal, the oldest object that is in the queue is removed. So every time the object is created it is inserted into a queue, where removal of it happens only when the server responds to the microcontroller that the information was received. Another application found for implementation of a queue is to use it to group the possible causes of errors without losing data. So if the ESP8266 is offline; reconnect; the queue is large; cannot communicate with the server; among others, a code that identifies the type of error, the date and time that the event occurred, is put in a specific queue of error and it will be sent when the connection is available.

The Flowchart in Figure 4 it is possible to visualize the logic implemented in the firmware setup function, in which the implementation of the error queue is observed. So, when restarting, the microcontroller tries to connect to the available Wi-Fi; if successful it requests to the server the current date and time, else it will try to connect again. When the date and time are updated an error message that says the microcontroller has been initialized is inserted in the error queue, as well the date and time that event occurred. The data that is in the queue is sent to the server and if the request is successful the data is removed from the queue, else it will try to send up three times and continues processing the code, in other words, goes to the loop function. This function is responsible for monitoring the flags altered by the time interrupts. The Flowchart in Figure 5 shows the implemented logic. When entering the function loop the state of the first flag is verified. If it is true it calls the function of synchronize the date/time, in which it requests to the Server the current date/time, following to the next verification. In case the flag_create_json is true it calls the function that creates the package containing the counted pulses along with the date/time of the creation and pushes to the pulses queue. Finally, if the flag_send_pulse is true, the functions that send the elements of the queues to the Server are called (if the queue is not empty).

D. Web Service

Web Service is an architecture that allows applications to send and receive data in XML or JSON format by default. This technology enables different applications to communicate with each other over the Internet using different resources. For the sharing of information in the Web service there are several standards, this work uses the Representational State Transfer (REST), that consists of a set of principles that uses the four methods of the Hyper Text Transfer Protocol based on the request-response architecture [10].

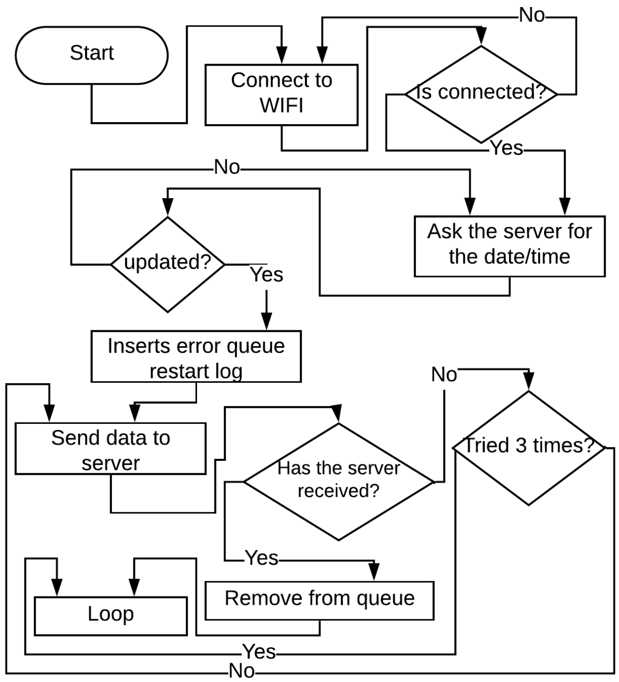

Figure 4. Logic implemented in the setup function

The chosen format to send and receive data via HTTP protocol was JSON, since its structure consists of key/value type pairs, which facilitates understanding and simplifies the generation. Thus, the generated format has attributes: the water meter identification, the number of pulses counted at a given time, the date and time it was generated.

Since the JSON structure to be sent must contain information of date and hour, the GET method (used to recover resources) was used to receive them, in JSON format. Such information is used by a microcontroller library that provides timing functionality.

The sending of the queue elements to a Web Service is done by a new interruption by time. This will call a function that makes a request using the POST method (used to create resources).

III. Results

Initially, part of the software responsible for counting pulses was isolated for testing. The test simulated the reproduction of a pulsed signal similar to that generated by the sensor present in the water meter, with noises. In this case, the counting was perfect in all the experiments performed. Subsequently, the tests toke place in a simulation closer to the real system, which the sending and receiving data tests are started. Within the robotics laboratory of the UFRN (LARS), a water meter equipped with the pulse emitting technology described in Section II was placed in a container with a pump, which generates a flow of water into the same reservoir. The flow telemetry could be monitored in near real time through the web

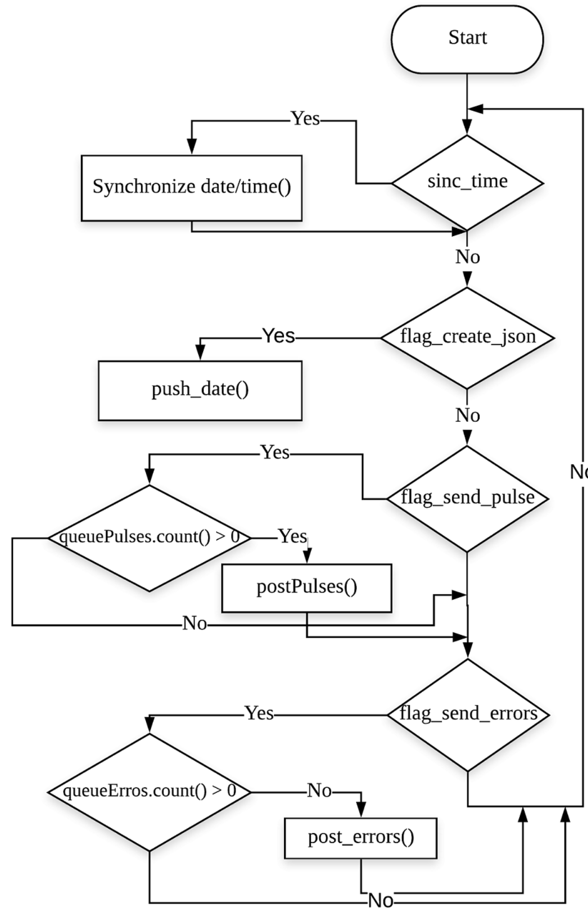

Figure 5. Logic implemented in the loop function

application and, similarly, in the serial port of the computer to which the microcontroller was connected. This allowed to observe the consistency between the data marked by the external meter of the water meter, what was read and passed by serial port and the sent data, presented in the application.

After validating the system as a whole, the described experiment spent 3 months under analysis. A total of 228470 pulses was counted, 20 less than the actual value, with a relative error of approximately 0.009%.

Such error can be associated with several factors:

- Unfiltered noise;
- Electromagnetic interference (generating fake pulses);
- Pulse count at microcontroller restart owing by a blackout or microcontroller malfunctioning;
- Resetting the microcontroller before the accumulated pulse value is recorded in non-volatile memory.

A. *Water meter pulse count*

For initial tests the time of interruption which inserting the water meter data in the queue was set for each minute; so every minute the number of pulses counted in this time window was inserted in the queue as well as the date and time. The size of the queue is important since it has a limitation and if more elements were stored than its capacity, the ESP8622 was forced to restart. This way, tests were performed to verify the number of elements that the memory of the microcontroller can store. With each element of the queue containing 55 characters, we were able to store 315 elements. Thus, with an element formed every minute, one can stay for up to five hours storing data.

B. Sending data

As explained before, we used an interrupt by time to send data, so that at each determined time interval the elements of the queue would be sent to the server until the queue presented no elements. Each time an element is sent, the server returns an HTTP status code. Because each code has a meaning, it is possible to identify causes of failures in sending and receiving of data. To monitor the behavior of the pulses queue with the microcontroller connected and disconnected to the internet, a test was performed with the ESP8266 sending data over a period of approximately 4080 seconds, where every 15 seconds a packet is inserted in the queue and in every 60 seconds the microcontroller tries to send the current data in queue to the server. Figure 6 shows the behavior of the queue during this experiment (they are on different scales). In the first 480 seconds, a Wi-Fi network was available for connection. During this stage, every time the queue had four elements, they were sent so to empty the queue, as shown in Figure 6a. Then the internet was disconnected for 3600 seconds and the behavior of the queue is shown in Figure 6b. Since there is no connection and the data cannot be sent, the size of the queue increases until the moment in which the internet connection was re-established. Finally, the microcontroller is reconnected and spends a few seconds sending all its packets, as shown in Figure 6c. After all elements were sent, chart 6c behaves as chart 6a.

C. Data visualization

The information regarding the total volume of water consumed is made available as the server receives data. In this way, the user can monitor their consumption within a daily, monthly or annual time window. Such data is shown graphically in a web application where the ordinate axis represents the liters and the abscissa the chosen consumption scale; a graph is made available as well as errors occurring in the system, shown in Figure 8. As previously described, this tool would also be useful for the concessionaire responsible for the distribution of water, since they could have access to the accumulated total and calculate the amount to be paid by the consumer without the need for a presential reading.

Figure 7 shows the data obtained by the experimental system installed in the laboratory, described in Section III. With the modification of the flow, it was possible to visualize the change in the graph that shows the number of accumulated pulses per hour. Above the bars, the total number of pulses collected can be observed, in which it is expected that this value will be identical with the one shown externally in the hydrometer.

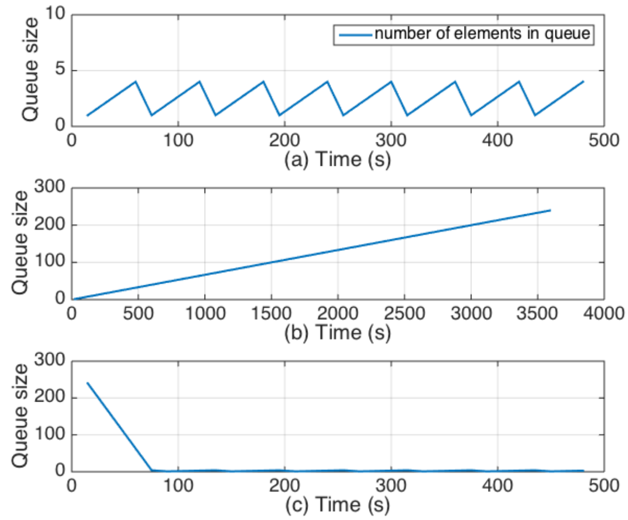

Figure 6. Behavior of the queue during an experiment. In (a) we see the normal behavior of the queue when there is internet available. The queue size increases as elements are inserted and decreases as they are sent to the server. In (b), when the internet connection is lost, the queue size increases. When the connection is restored, the queue is emptied and returns to its normal behaviour as in (a).

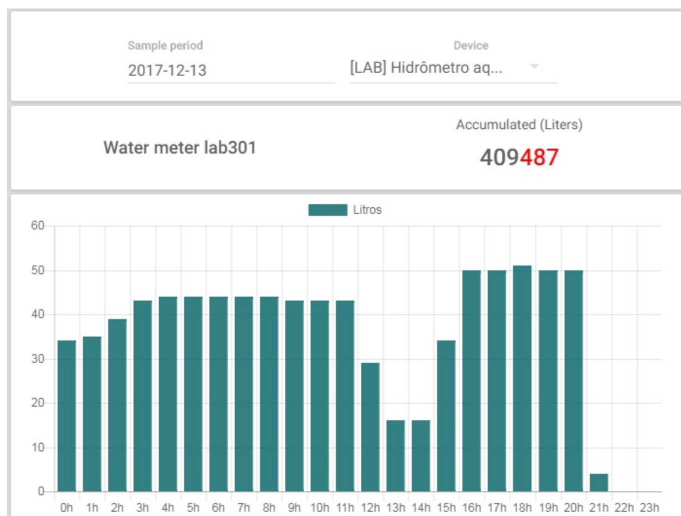

Figure 7. Daily Consumption Chart

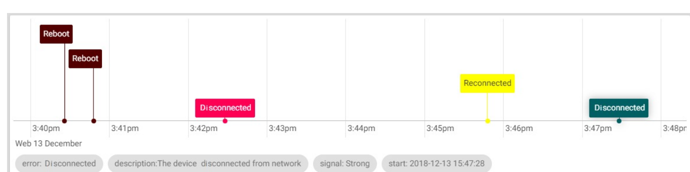

Figure 8. Errors occurred

IV. Conclusions

The proposed work dealt with the development of a microcontrolled system for measuring individual domestic water consumption. For this purpose, validation tests were performed in an experimental environment, where the system presented an accuracy of 99.99% (described in section III) in its measurement. In this context, it can be concluded that this technology is effective for reading the flow of a water meter, enabling telemetry; however, it is necessary to improve the filter, since this error is cumulative according to the flow and can generate future inconsistencies. The primary goal has been achieved since the consumer can have graphs reporting the total volume consumed as shown in the III section, obtaining more reliable and frequent readings. The concessionaire may also have access to this information to calculate the amount to be paid by the user without the need for a presential reading, eliminating repetitive tasks, reducing the occurrence of reading errors and making it easier to identify possible anomalies. Other technologies also need to be studied for implementation in order to favor users who do not have a Wi-Fi network near the place where the water meter is installed, as an alternative for communication at long distances with minimum energy consumption. In [11], the autors propose the LoRa (Long Range), a radio frequency technology, as a good candidate to solve IoT-based problems.

REFERENCES

[1] H. Lasi, P. Fettke, H.-G. Kemper, T. Feld, and M. Hoffmann, "Industry 4.0," *Business & Information Systems Engineering*, vol. 6, no. 4, pp. 239–242, 2014.

[2] A. McAfee, E. Brynjolfsson, T. H. Davenport, D. Patil, and D. Barton, "Big data," *The management revolution. Harvard Bus Rev*, vol. 90, no. 10, pp. 61–67, 2012.

[3] B. P. Santos, L. A. Silva, C. S. Celes, J. B. Borges, B. S. P. Neto, M. A. M. Vieira, L. F. M. Vieira, O. N. Goussevskaia, and A. A. Loureiro, "Internet of things: from theory to practice."

[4] V. Sharath, S. Suhas, B. S. Jain, S. V. Kumar, and C. P. Kumar, "Smart aqua meter," in *Advances in Electronics, Computers and Communications (ICAECC), 2014 International Conference on*. IEEE, 2014, pp. 1–5.

[5] M. F. C. da Silva and D. P. Pereira, "System for measurement and control of water consumption using wireless sensor network," in *Computing System Engineering (SBESC), 2011 Brazilian Symposium on*. IEEE, 2011, pp. 105–107.

[6] D. Araujo, J. Costa, D. R. C. Silva, M. B. Nogueira, and M. C. Rodrigues, "Arquitetura geral de um sistema para automação residencial utilizando plataformas abertas," in *XIII Simpósio Brasileiro de Automação Inteligente, 2017 Brazilian Symposium on*. Universidade Federal do Rio Grande do Sul, 2017, pp. 1–6.

[7] B. E. Shlesinger Jr. and C. D. Mariner, "Magnetic reed switch," Aug. 2 1977, uS Patent 4,039,985.

[8] N. Kolban, "Kolban's book on esp8266," *an introductory book on ESP8266*, 2015.

[9] M. McRoberts, "Arduino basic," *São Paulo: Novatec*, 2011.

[10] R. F. Da Silva and P. R. Gonçalves, "Web services–uma análise comparativa," *Revista das Faculdades Integradas Claretianas–Nº5–janeiro/dezembro de*, p. 8, 2012.

[11] U. Raza, P. Kulkarni, and M. Sooriyabandara, "Low power wide area networks: A survey," *arXiv preprint arXiv:1606.07360*, 2016.