An IoT-Based Culvert Monitoring System for Urban Flood Prevention

Ladson Gomes, Mariana Barros, Thiago Silva, and Edna Barros

***Abstract*—Urban floods cause several damages to both the cities and the citizens. The rainwater triggers the floods, which is aggravated by the impermeable soil, forcing the water to be released mainly by the drainage system. However, the solid residues on the streets are dragged into the culverts, obstructing it. As a result, the culvert becomes a breeding site for mosquitoes. The drainage system cleaning is expensive and time-consuming. To mitigate this problem, we propose a system to monitor culvert obstruction, which includes three subsystems: The basket collector, the gateway for data flow control and analysis, and the web server.**

The developed system uses solar power for energetic autonomy, and presented sensor accuracy on the culvert environment and good reliability during tests. The IoT approach for the system's design favors the scalability for a more precise analysis on the real world.

***Index Terms*—Internet of Things, Monitoring System, Embedded, Urban Floods, Solar Energy, Smart Cities.**

I. Introduction

URBAN floods are one of the major issues in the big metropolis of Brazil and have been an increasing topic over the last ten years. It happens mainly due to the impermeable soil, which forces the rainwater to be released only through the drainage system[1]. The water drags trash and other residues from the streets into the culverts, the entrance of the drainage system. When the garbage accumulates inside the culvert, it may clog the pipes and hold the water, becoming a breeding site for mosquitoes. The urban floods cause several problems to the city's infrastructure, workforce and the citizen's health.

In São Paulo, a metropolis with over 11 million people (IBGE - 2010), it is estimated a decrease of 13,7% on the culvert cleaning, according to data about the first four months of the year between 2013 and 2015. According to the City Hall, the sanitation periodicity varies based on the culverts that are in potentially floodable areas. The cleaning costs in São Paulo are about R$70 million a month [2]. In Goiania, (population of over 1,296 million according to IBGE 2010), the cleaning of each culvert costs about R$53,80. The annual expenses with the cleaning of every culvert at least once a year is over R$16 million, and are necessary at least five people to unblock each culvert.

The high cost of culvert cleaning is due to the laborious work of removing the residues from the pipes. Redesigning a city and its vital systems is not a possible task, but we may use microelectronics for optimizing and making our cities smarter.

From the problem analysis, a system was developed to monitor residues level and manage the cleaning of culverts throughout a city. The system, named EccoBin, is divided into three modules, designed for maximizing the scalability, sustainability, and efficiency. The first module is responsible for collecting data, acting, and retaining residues on the culvert. The second module can manage the data flow from up to six culverts, analyzing the data and monitoring the rain. And the third module is responsible for storage, display, control, and intelligence concerning the data collected by the system.

This paper presents the design and development of an Internet of Things system, from the embedded components to the application on the web. The system architecture and the modules' description are presented in the following section. The section "Implementation" details the operation and technologies used, respectively, in the Basket Collector, in the Gateway in the Web Server. Section 3 describes the metrics used for the system validation and the results obtained during the tests. Section 4 names other researches and systems related to the same area. Section 5 concludes the article.

L. Gomes, M. Barros and T. Silva are undergraduate students on Computer Engineering at the Center of Informatics at UFPE. Respective e-mails: (lgs3, msb4, tms3)@cin.ufpe.br

E. Barros is a titular professor at the Center of Informatics at UFPE. E-mail: ensb@cin.ufpe.br

II. Related Works

In this section, some related works are presented and compared with the monitoring system proposed in this paper.

The "ECCO Filter" [3] is a device developed to collect and detect the presence of residues in a culvert. It is composed of a perforated basket that collects the garbage to ease the cleaning. It uses an ultrasonic sensor to measure if there is garbage in the basket. The data is sent through wireless technology to the "ECCO Manager", that displays this information to the user. However, the system doesn't measure the presence of standing water in the culvert, neither takes measures to avoid the mosquitoes' procreation. Also, it is not described how the system is powered.

In [4], a device is described to detects if there is an obstruction in the culvert, in order to mitigate the occurrence of water flows in the big cities. The system is composed of an ultrasonic sensor that detects if there are residues in the culverts. The sensor is connected to a microprocessor, and the data is sent to a database through 3G technology. The device is powered by a solar panel attached to a battery. However, the system described in the paper doesn't use a basket, and the problem of the pipes' obstruction remains. Also, the system does not detect the presence of standing water in the culvert, neither takes measures to avoid the mosquitoes' procreation.

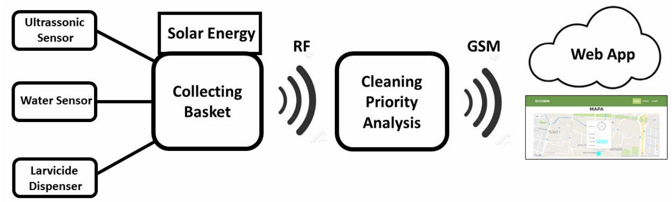

Fig. 1. System Architecture

III. THE ECCOBIN SYSTEM

A. System Architecture

The system is divided into three subsystems: the Basket Collector, the Gateway for analysis and control and the Web Server, as shown in figure 1. The three modules communicate with each other through wireless technology, following a protocol designed to avoid data loss and deadlocks.

On the Basket Collector subsystem, powered by a solar panel and a battery, a basket allows collecting the garbage that gets into the culverts, letting the water pass through. The module contains sensors that measure the volume of trash collected, and read sensors to verify the occurrence of standing water in the basket. Upon request, the module sends those data to the Gateway subsystem. The communication is made through a nRF24l01 radio frequency module. The subsystem also pours larvicide powder in the culvert to avoid the mosquitoes larvae growth. The Gateway subsystem, located in a lamp post nearby, requests, receives and stores the data from the Basket Collector, verifies the communication between the modules and also the proper behavior of the sensors. The module also receives data from a pluviometric sensor, and, along with the data collected from the culvert, calculates the probability of flooding, defining a cleaning priority level for it. A GSM module sends the analysis data to the central server. The Web Server module receives the information from up to 6 Basket Collector modules and stores them in a database. Then, the data is displayed on the web application in a map containing the current report of every culvert. The module sends an email alert to the responsible authorities in the case of high flooding risk, failure on the communication or on the sensors. The citizens can also contribute with information about the culverts on the city by sending a report. It allows the change on the periodicity of data collection through the web app. The figure 1 below shows the system architecture.

B. Description of the modules

1) Basket Collector: The Basket Collector includes a perforated basket located next to the culvert's opening. It is responsible for collecting the garbage dragged into the place, in order to ease cleaning and allow the water drainage. The basket measures the volume occupied by the retained residues and verifies if there is water in the culvert.

The module communicates with the Gateway using a transmitter, which sends the requested information. If there is clogged water, it will be treated for avoiding mosquitoes. After sending the data, the module enters standby mode to save power, which lasts for the set period. The module design

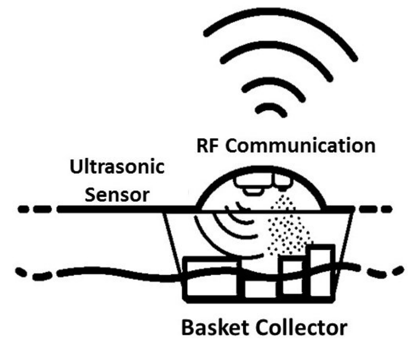

Fig. 2. Basket Collector Design

is shown in figure 2.

2) Gateway: The Gateway is able to communicate with the Basket Collector module, following a communication protocol designed specifically for this situation. The module requests and receives data from them, check the communication and sensors and measures the rain amount in the neighborhood. With these data, the module performs an analysis and defines the culvert cleaning priority, taking into account the garbage volume in the culvert and the information about rain on the neighborhood.

The Gateway module sends the verification and analysis results to the Web Server and receives back the time interval that the Basket Collector module should be in the energy economy mode. This information is then forwarded to the Basket Collector.

3) Web Server: The Web Server receives, through the Internet, the measured data and the analysis result. It includes the information concerning the communication status and sensors checking. The server is responsible for storing, in a database, all the information related to each culvert, in addition to showing, in a map, each one's location and recent values. When a culvert receives a high cleaning priority, or when malfunctioning is detected, the server sends an email alert to the responsible authorities.

The module also allows the residents to make a report of the flooded culverts and offers the possibility to change the data collecting periodicity through the web application. Lastly, the server also defines an optimized cleaning route from the database information.

C. Implementation

1) Basket Collector: The Basket Collector module was implemented using an Arduino Uno, that is responsible for receiving the information scanned by the sensors, sending it to the Gateway module and activating the mechanism of overthrowing the larvicide powder in the culvert. It responds to the communication check and receives the period that will remain in power saving mode. When requested, the module enters this mode for a certain time to ensure energy efficiency. The module is powered by a solar panel connected to a battery, which provides its operation even when there is not

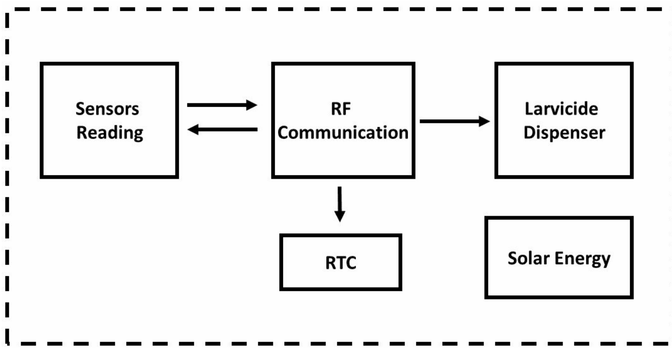

Fig. 3. Basket Collector Module

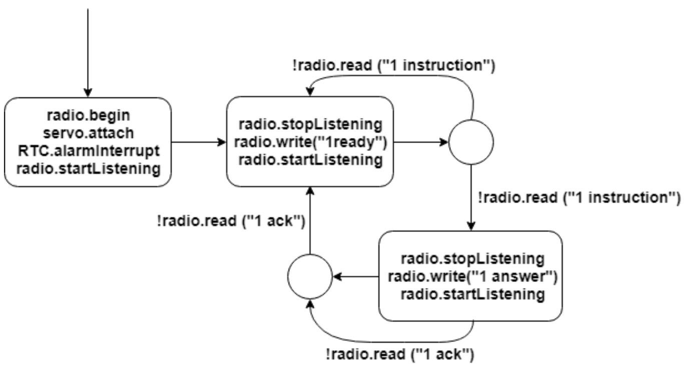

Fig. 4. Basket Collector Module State Machine

the presence of sunlight. Figure 3 below shows the architecture of the Basket Collector module.

The Arduino is connected to two ultrasonic sensors, which measure the distance to the closest obstacle and, using the basket volume as a parameter, calculates the percentage of the basket occupied by garbage. A water sensor attached to the side of the basket measures the level of retained water in it. The Arduino then joins this information into a string and sends it, by an RF transmitter, to the Gateway module. In response, it receives a confirmation that the information was received correctly and, depending on the situation, the command to activate the mechanism of pouring the larvicide powder. Also, the module responds to the request from the Gateway to verify either the communication or the sensors. One RTC (Real-time Clock) connected to the module performs the task of, when requested, put the module in power save mode, during a period which is also informed through RF by the Gateway. The module is powered by the solar panel, which is also used for charging a battery across the board EH-01. That way, there is better energy efficiency, making the module energetically self-sustainable.

Figure 4 above shows the state machine that implements the control of the Basket Collector. In the first state, the RF transmitter is initialized, opens the channels for the module to receive and send messages, and begins to listen. The servo motor and the RTC are also initialized. Then, in the second state, the module sends to the Gateway through RF a message saying that it is ready to work. While it does not receive a message with the instruction coming from the Gateway, it continues sending the message of "ready". When the message arrives, the module proceeds following the command received and sends the appropriate response to each situation. If the instruction is to check the RF communication, the module responds by saying that the communication interface is working. If the instruction is requesting data, the module measures the percentage of volume occupied and the presence of water in the basket and sends through the RF transmitter. In case the statement is warning that there is standing water in the culvert, the analysis module triggers a mechanism that pours larvicide powder in the culvert to avoid the breeding of mosquitoes larvae. And, if the instruction is requesting to enter in the power saving mode, the Basket Collector reads the amount of time to be in the power saving mode and then sends a message confirming the action before entering in this mode. Then the Basket Collector expects a message from the Gateway to acknowledge that the requested data has been successfully received. While it doesn't receive the message, the module sends the data again through the RF transmitter. Then, if necessary, the module enters into power-saving mode.

2) Gateway: The Gateway, responsible for the system analysis and control, was implemented using the Galileo Gen 2 board. It is responsible for storing and analyzing the data from the pluviometric sensor and the Basket Collector module, and, with the analysis result, defining a cleaning priority level for the culverts. It is also responsible for checking the communication and sensors from the Basket Collector module, and sending the commands to get it the standby mode and pours larvicide powder. The analysis result and the collected information are transmitted, through a GSM module, to the Web Server. The Gateway module also sets the collecting data periodicity, which can be calculated by itself or manually defined in the web application. Figure 5 below shows the Gateway architecture.

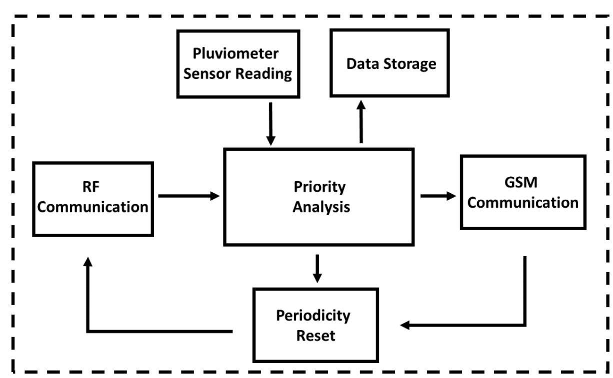

Fig. 5. Gateway Architecture

The communication between the Basket Collector and the Gateway follows a protocol to avoid data loss and deadlocks. The protocol is based on the sent and received messages so that while the module does not receive the message it is supposed to, it keeps asking for the message repetitively. If a timeout is reached and the expected message still hadn't arrived,

TABLE I
MESSAGES SET

Code	Message
1	Data Request
2	Communication Checking
3	Larvicide Dispenser
4	Sensors Checking
5	Low Consumption Mode

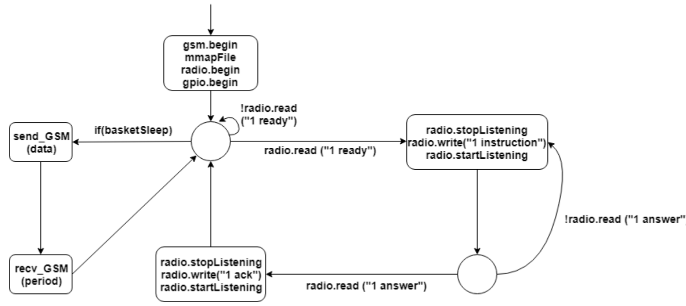

Fig. 6. Gateway State Machine

the gateway will notify that there may be a communication problem in the next time it sends a message to the Web Server.

The module has an instruction list that is transmitted to the Basket Collector module. The instruction that is being computed at the moment is initially decoded, as shown in table 1, and then sent through the RF transmitter. Instruction number "2" requests the RF communication checking, that is repeated during a certain number of times to assure the verification. Instruction number "4" requests the sensors collected data from the Basket Collector. This instruction also repeated a certain number of times, has the purpose of acquiring data to check the operation of the sensors. Instruction number "1" requests the data sensors again. These data are studied along with the pluviometric sensor data, and from them is defined the culvert cleaning priority level. In case there is standing water on the culvert, instruction number "3" requests the Basket Collector to actuate the mechanism to pour larvicide powder. Instruction number "5" sends the command for the Basket Collector to get in the energy economy mode, also sending the time interval that it should remain in this mode. When the analysis module receives the message saying that the Basket Collector left the standby mode, the cycle restarts.

The priority computation done in the module is based on the rainfall amount collected along with the garbage volume in the culvert. According to various tests performed and their results, the system is divided into five levels:

- Level 1: it is not raining, and the garbage percentage was smaller than 20%.
- Level 2: it is not raining, and the garbage percentage was greater than or equal to 20% and smaller than 50%.
- Level 3: it is not raining, and the garbage percentage was greater than or equal to 50% and smaller than 80%.
- Level 4: it is not raining, and the garbage percentage was greater than 80%, or it is raining, and the garbage percentage was smaller than 80%.
- Level 5: it is raining, and the garbage percentage was greater than 80%.

When the level is defined, the module stores and transmits it, through the GSM module, to the Web Server. Through the same module, the Gateway receives from the web server the information concerning the way the periodicity will be defined. In the case it is set as automatic, the analysis module itself defines it, from the collected data. Otherwise, the periodicity manually defined in the web application is received through the GSM module.

Figure 6 below illustrates the Gateway state machine. In the first state, the GSM module, the pluviometric sensor, and the RF transmitter are initialized. Then, the Gateway keeps waiting for the message from the Basket Collector, claiming that it is ready to work. When the expected message arrives, the module sends the instruction to the Basket Collector and keeps waiting for the answer. When it is received, the Gateway sends a message confirming that the response has arrived. This cycle is repeated until the instructions list is over. When the Basket Collector gets in the energy economy mode, the Gateway communicates with the web server. Initially, it sends the collected data and the priority analysis, and, subsequently, it receives from the web server the data collecting periodicity. When the Basket Collector sends again the message claiming that it is ready to work, the cycle is restarted.

3) Web Server: The central server module is implemented as a web application, which receives information from the system through the Internet from the GSM module connected to the Gateway.

The Basket Collector module, using the Galileo, sends the information received from the system to the server through an HTTP POST request, encoding the information in a string on the x-www-URL-encoded format. The server receives the data through the HTTP request and verifies if the data is valid, if so, the data received will be stored on the database with the date and time it got to the system.

The received data is stored in a database, and the most recent entry for each culvert will be on display on a map, to improve the user interface. On the trace route section, the fastest route for cleaning the culverts in need is calculated according to the selected priority and then displayed. Using the Google Maps API, it is possible to easily map, present, and estimate the fastest route from point A to point B, passing through all the culverts marked.

The server is built using the Django framework for web development and is hosted on Microsoft Azure (ec-cobin1.azurewebsites.net). The database uses sqlite3 to store the system data. The front-end used Bootstrap 3, jQuery and Google Maps and Google Charts APIs to display the data. The periodicity in which the system retrieves new data can be reconfigured on the website, by the system manager. There is a field to provide the number of minutes for the data to be updated periodically, and a field in which you choose if the periodicity inserted shall be used, or if the Gateway will decide what is the best periodicity base on if it is raining or not. The system alerts the responsible for the maintenance by email, in case it receives an entry of priority 5, communication error or sensor malfunction, informing the culvert's id, address and possible causes.

The local community can also help on the data collection, reporting the status of a culvert with a picture, a short form and the address of the culvert. This data is going to be on display for the system manager.

IV. Results

A. Validation

We developed a prototype for the system with the specified characteristics, to test on a real sized prototype culvert. In the event of not being able to check a culvert in the street, we tried to simulate the environment on its real-life conditions. In order to validate the system; it was used the unit test technique. The project modules were separately tested and subsequently connected, for validating the complete path that the data will go through.

The Basket Collector module was tested with different amounts of garbage, for the ultrasonic sensors validation. The module was also tested with and without water, for the water sensor and the larvicide dispenser validation. The data transmission using the RF module was tested in different distances, with and without obstacles. The module was tested using the solar panel and the battery, to confirm that it is self-sustaining.

For validating the Gateway, the pluviometric sensor was tested with different amounts of water. The cleaning priority level computation was calculated from the measurement of the water draining time. It was also tested the necessary time for the module to send and receive messages from the server.

The Web Server was validated by testing different data transmission through the GSM module. It was measured the necessary time for the web application to update the data to be sent. It was also tested the different functionalities from the web application: sending an email alert, defining an optimized cleaning route, updating the map and the database, analyzing the residents' reporting and changing the priority.

B. Measurements

The power consumption in the Basket Collector module is 387.5mW, including the consumption of the Arduino board, the NRF24L01 module, the ultrasonic sensor and the power sensor. The solar panel produces 3W of energy, and the battery supplies about 1.8W.

It was necessary around 40 minutes for the total charge of the battery on a sunny day, while the system was working. The battery kept the system working correctly for 10 hours after the solar panel stopped producing electricity, simulating a night environment, while the basket was in sleep mode for 30 minutes before measuring new data. The developed system got to have a 91% uptime, using solar power for energetic autonomy, or about 22h a day, using the lithium battery.

The system response total time was calculated by summing up the time for reading the sensors to passing through all stages, the time for doing verifications and the time for getting on display on the web server. The measured total time was always less than 1 minute. The data collection and transmission to the Gateway, verifying the sensors measurements took 10 seconds on average, and the GSM transmission took between 20 and 45 seconds.

TABLE II
Basket Collector Budget

Components	Price
RF Module	U$3.9
Arduino Uno Board	U$12.9
Servomotor	U$3.9
Ultrasonic Sensor	U$3.9
Battery	U$10.3
Solar Panel	U$9
Water Sensor	U$3.1
Total	U$47

TABLE III
Gateway Budget

Components	Price
Intel Galileo Gen2 Board	U$129
RF Module	U$3.9
Pluviometric Sensor	U$38.7
GSM Module	U$67.2
Total	U$238.8

The Gateway was able to control the data flow of the system, managing two data collectors at the same time, analyze the data flow, and monitor the rain sensor in parallel. In this situation, the response time was the same for each culvert. As the communication was sequential, each Basket Collector would wait for the other one to finish.

The Basket Collector developed as a prototype measures 0,021 m^3. The Basket collector was tested when the basket was empty, and when it had 25%, 50%, 75% and 100% from its volume occupied by trash. In each situation, the system was also tested with water, and it was able to correctly measure the level of solid residues and detect the presence of liquid in the culvert.

We tested the communication range between the collector basket and the gateway, and the maximum range was about 150 meters on the open. This range may be shorter if there are obstacles between the modules, other than the wood culvert cover. When the RF range was measured with a wall as an obstacle, the maximum range where we could still communicate was 20 meters. The measured range for the nRF24l01 greatly differs from the seller's[5] estimated range of 1km in ideal circumstances. The nRF24l01 was chosen due to the possibility of using multiple channels for transmission and listening to more than one channel at a time, reducing the interference on the communication.

The cost of implementing the Basket Collector was about U$47. For the gateway, the implementation cost was about U$238. Table 2 details the electronic components used in the Basket Collector. Table 3 describes the components used to implement the Gateway.

The basket collector module was implemented on 350 lines of code. The Gateway was implemented in 960 lines of code. Figure 7 shows the prototype of the Culvert and the Web

Server.

Fig. 7. Evaluation Prototype: (left to right) the perforated basket, the culvert, the culvert cover(top and bottom).

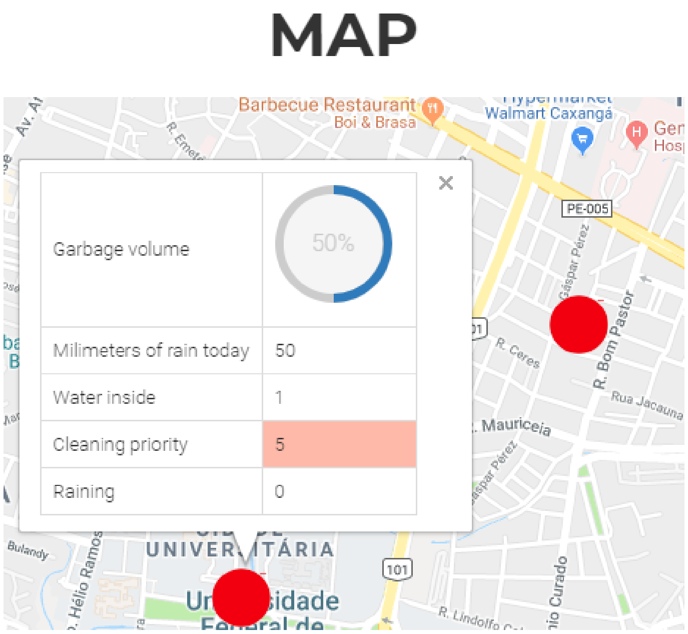

Fig. 8. Evaluation Prototype: The mapping interface on the website.

In comparison to the current options to mitigate the urban floods problem, the proposed system brings the following advantages:

- Energy Efficiency: The solar panel and the battery attached to the Basket Collector make this module self-sustainable, assuring that it will have enough power to work, despite the environment.
- Intelligent data collecting periodicity: The system allows the user to choose if the periodicity for data collecting will be defined by itself or by the gateway, according to the measurements and analysis results.
- Low response time and real-time information update: Due to the code simplicity and optimizations, the system shows a low response time. When it is not in the standby mode, the Basket Collector performs all the measurements and immediately sends the data to the gateway. This feature, in turn, is measuring the rain amount constantly, which assures the real-time data.

V. Conclusion

In this paper, we presented the architecture and implementation of a system for monitoring the drainage system, based on embedded systems and Internet of Things.

Eccobin is a cleaning management and monitoring system for culverts that brings a practical, sustainable and lasting solution to mitigate difficulties on the drainage system of urban centers. It consists of three subsystems communicating among themselves wirelessly for data exchange. The developed prototype covers the proposed functionalities for the system. The Basket Collector module is energetically self-sustainable and was designed in a way to measure the garbage volume, detect water on the basket, pour larvicide powder on the culvert, and communicate with the Gateway to exchange instructions and data through radio frequency. The Gateway measures the amount of rain and calculates the cleaning priority for each culvert with the available data, and is responsible for transmitting data to both the collector basket modules and the web server through the GSM module. The Web Server is a web application for culvert data storage and display, being able to register new culverts, display the information on a map and send alert messages when necessary.

The proposed system architecture was designed to assure scalability, reliability, and robustness for enduring the environmental conditions. The monitoring system was developed to manage the maintenance of culverts and to decrease expenses with human resources.

VI. Future Works

A sketchy prototype currently implements the EccoBin for a concept proof, but an integrated circuit is on the horizon if there is demand. For a final product, a less expensive standard board could replace the Galileo in the Gateway, e.g. a Raspberry, Beaglebone, or even an ESP32. In all of the mentioned boards, it is possible to use multiple threads and have an additional memory for code. The Arduino Uno on the Basket Collector, for instance, could be replaced by an Arduino Pro Mini, which could reduce the price and consumption.

To lower the costs of the system, a higher range wireless module for the communication among the basket collectors and the gateway is necessary. So fewer gateways would be needed. A less expensive alternative solution can also replace the GSM module, e.g. SigFox and LoRa. The HTTP protocol used on the communication could be replaced for an IoT-focused protocol, as the now popular MQTT, or another alternative that may arise in the following years that mitigates the problems of vulnerability in IoT networks. In the case of a long-range, low power and low-cost communication as SigFox, the need for the Gateway would be minimum, and the Basket Collector could send the data straight to the server for analysis.

For future research, data provided by this system in a real environment can lead to flood prediction and prevention. Also, with a better data analysis, the system would have the possibility to help in the management of the city cleaning system. The analyzed data could be used in researches related to urban planning, as these data could provide insights into the flow of people throughout the city.

References

[1] Tomaz, Plinio. Curso de manejo de aguas pluviais: Capitulo 7 - Bueiros ou Travessias [Rainwater Management Course: Chapter 7 - Culverts or Crossings]. 2009.

[2] Mazzo, Aline; Sena, Stephane. Prefeitura de SP diminui o ritmo de limpeza de bueiros na cidade [SP City Hall slows down the cleaning of manholes in the city]. Folha de São Paulo. 2015. URL: http://www1.folha.uol.com.br/cotidiano/2015/07/1656388-prefeitura-de-spdiminui-o-ritmo-de-limpeza-de-bueiros-na-cidade.shtml. Access in (June 2017).

[3] Ecco Sustentável. 2012. URL: http://http://eccosustentavel.com/. Access in (June 2017).

[4] S. A. Ribeiro; A. D. de Carvalho; A. M. de Oliveira; S. T. Kofuji. Modelo conceitual de um sensor microcontrolado 3G para automação do controle de saturação de dispositivos de drenagem urbana (bueiro) aplicado a cidades inteligentes. Congresso Científico da Semana Tecnológica –IFSP, 2014.

[5] Open Impulse. URL:https://www.openimpulse.com/blog/products-page/product-category/long-range-nrf24l01-pa-lna-rf-transceiver/. Access in (July 2018).