Wireless Sensor Network for Monitoring Environmental Factors in Industrial Installations

Lauro Manoel Lima da Gama, Joo Batista Hidaka de Oliveira Gaia, Antonio de Padua Soares Junior, and Msc, Almir Kimura Junior

***Abstract*—Abstract**
This paper proposes a wireless sensor network for measuring environmental factors on industry surroundings. Industrial environment is regulated in varying degrees across the world by many environmental and safety policies. The adherence of these policies makes necessary a continuous and distributed measurement of different factors like temperature, noise levels and the presence of contaminants. The analysis of this data may help prevent violations in work conditions and environmental damage. Low worker productivity and mood disorders are also often related to to bad work environments and be combated by analysis of the environment. The proposal of a low cost, easy interface and customizable wireless sensor network to be used in industrial environment will help improve work conditions.

***Index Terms*—Wireless sensor networks, Internet of Things, Environmental factors**

I. Introduction

WITH the increasing importance and subsequent regulation of the impact the industrial Environment has on the health of workers and denizens of nearby industrial areas, arises the necessity of monitoring environment factors.

In the United States the agency responsible for regulation work conditions is the Occupational Safety & Health Administration(OSHA) [1].It cites many cases of work related injuries and diseases that could be prevented with the correct environment control.

To analyse correctly a region the best approach is through geographically distributed data collection. To achieve this collection is necessary the implementation of a sensor network. The dynamic nature of industries present a challenge to wired networks and the pressures of competition and cost efficiency apply many constraints to development.

Wireless sensor networks have advantages over wired networks because the sensor nodes are may be installed without changes in existent infrastructure by a relative small price [2].

II. Research Methodology

This is descriptive research. The source of information is secondary data from different sources like websites & books mentioned at the end.

A. Sensor network

In recent years, wireless sensor networks (WSNs) have gained worldwide attention for use in different applications. Sensor nodes are spatially distributed across a large area of interest to sense, measure, and gather information and transmit the data to the user. The nodes are typically equipped with radio transceivers, micro-controllers, and batteries. These sensor nodes are in general equipped with one or more sensors (e.g. mechanical, thermal, biological) [3]. They are small in size, inexpensive, and could be deployed in large numbers. They can be used in applications such as military target tracking and surveillance, natural disaster relief, biomedical health monitoring , and industrial automation. [4] Each of these nodes can sense, measure, and gather information from the environment and, based on some local decision process, they can transmit the sensed data to the user [3]. This article proposes a sensor network capable of collecting data from sensors spread in a industrial environment and transmit this information to a central hub to be further analysed.

B. Design and implementation

1) Network topology: The target for the proposed network are indoor buildings with air conditioning and artificial lightning. These types of building normally range to up to hundred of meters of built area with many partitions between the areas. The nodes don?t need to communicate between themselves, only with the server. Short-range wireless technologies such as IEEE 802.15.4 in mesh network configuration are widely considered to be cost-effective solution for use in industrial settings [4]. These characteristics make the star network topology the most adequate to be implemented.

In this topology each sensor node connects directly with the central hub, sending and receiving information. This communication is made over WIFI where each node connects with the hub and receives an IP address. Each node has an internal identification in hash format that is sent with each message. The message payload from each sensor may change according with the hardware so the message format must be flexible. The presented architecture is able to be scaled to a high number of sensor nodes connected to only one hub. This is possible because of a efficient message protocol that is able to process hundreds of messages per second [5].

2) Sensor Node: The sensor nodes are raspberry pi 2 model B [6] running a Raspbian [7] distribution composed of a micro-controller receiving signals from sensors. Each signal must be converted to an appropriated scale and packaged in a message to the Central Hub. The following subjects are

This article was submitted on July 27th, 2015. This work was supported in part by Denso Industrial da Amazônia LTDA and Paulo Feitoza Foundation.

Lauro Manoel Lima da Gama is with Paulo Feitoza Foundation, Manaus, AM, BRA (e-mail: lauro.gama@gmail.com).

Joo Batista Hidaka de Oliveira Gaia is with Paulo Feitoza Foundation, Manaus, AM, BRA (e-mail: hidaka.gaia@gmail.com).

Antonio de Padua Soares Junior, is with Vortice Technology and Innovation, Manaus, AM, BRA (e-mail: antoniodepadua27@gmail.com).

Msc, Almir Kimura Junior is with the Electrical Engineering Department, Universidade do Estado do Amazonas, Manaus, AM, BRA, (e-mail: akimurajr@gmail.com).

being monitored: Temperature, humidity, noise, luminance, and flammable gas presence.

The nodes may be placed in hard to reach locations or in great quantities making a requirement of the network that the sensor node must be fault resilient and able to restart automatically in case of failure. Updates on the software will be made over the air through automatic processes. The sensor node has a program called *”client”* that is responsible for the minimum running processes as life cycle besides the data collection and transmission of information.

The client persists data from the sensors in a SQLite [8] database and transmits this information to the central hub. In case of failure of communications the data collected may be resent in a future date. A Lithium polymer battery is used to provide energy backup to the node in case of power shortage. in this scenario the data transmission will be halted and only the data collection will occur.

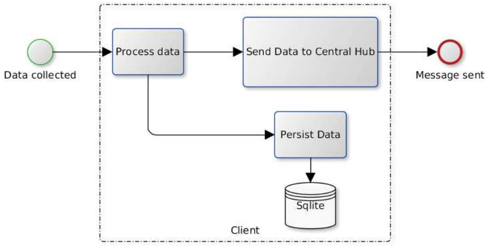

Fig. 1: Sensor Node Diagram

3) Central Hub: The central hub must be able to process a high volume of messages coming from the sensor nodes and to answer requisitions for information from network users.

The central hub was implemented also using raspberry pi 2 model B running a computer program based on Tornado [9], a RabbitMQ broker and a web server based on the framework flask [10].

Tornado is a Python web framework and asynchronous networking library. By using non-blocking network I/O, Tornado can scale to tens of thousands of open connections, making it ideal for long polling, WebSockets, and other applications that require a long-lived connection to each user. [9]

The server is responsible for listening for messages posted on the RabbitMQ broker and collect the relevant data and persist them in its own SQLite database. This database has information from all nodes composing the network.

The RabbitMQ broker is responsible for managing communications between nodes and the central hub.

The web server is an application with an REST API [11] responsible for providing users of the system with an method to visualize the data collected by the sensor network. It uses a lightweight server structure ideal for running in embedded systems.

The services implemented in the central hub may be divided between data collection and communication.

4) Message Protocol: The messages are sent from the sensor node to the central Hub using the MQTT protocol.

MQTT stands for MQ Telemetry Transport. It is a publish/subscribe, extremely simple and lightweight messaging protocol, designed for constrained devices and low-bandwidth, high-latency or unreliable networks. The design principles are to minimise network bandwidth and device resource requirements whilst also attempting to ensure reliability and some degree of assurance of delivery. These principles also turn out to make the protocol ideal of the emerging ?machine-to-machine? (M2M) or ?Internet of Things? world of connected devices, and for mobile applications where bandwidth and battery power are at a premium [12].

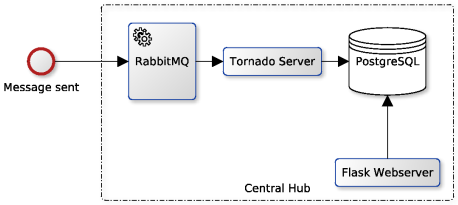

Fig. 2: Central Hub Diagram

MQTT is a publish/subscribe messaging system that allows clients to publish messages without concerning themselves about their eventual destination; messages are sent to an MQTT broker where they may be retained.

The messages’ payloads are just a sequence of bytes, up to 256MB, with no requirements placed on the format of those payloads and with the MQTT protocol usually adding a fixed header of two bytes to most messages.

Other clients can subscribe to these messages and get updated by the broker when new messages arrive. To allow for the variety of possible situations where MQTT can be put to use, it lets clients and brokers set a ”Quality of Service” on a per-message basis from ”fire and forget” to ”confirmed delivery”. MQTT also has a very light API, with all of five protocol methods, making it easy to learn and recall, but there’s also support for SSL-encrypted connections and username/password authentication for clients to brokers.

The broker utilized in the implementation was RabbitMQ through the python package Pika. Pika is a pure-Python implementation of the AMQP 0-9-1 protocol that tries to stay fairly independent of the underlying network support library [13]. The model takes into account the need for constant data analysis and transmission of high volumes of messages in a heavy electronic noise environments.

5) Message Format: The message payload is formatted using in the Json format. JSON (JavaScript Object Notation) is a lightweight data-interchange format [14]. Each information corresponds to a key-value par that can be read easily by both human and computer programs.

```
{
   "timestamp": "2015-07-16 15:40:49.218438",
   "data": {
      "ilumination": 210,
      "noise": 123,
      "carbon_monoxide": 170,
      "temperature": 51,
      "humidity": 127
   },
   "id": 307
}
```

C. Implementation and tests

To validate the proposed system a prototype was implemented to collect temperature, humidity and luminance data to monitor the work conditions of a computer data server room. These kinds of rooms normally need low temperatures and low humidity to maintain the proper function of the machines. In the other way the luminance levels need to meet the minimum required to not strain workers eyesight. The prototype was used to assure that all three factors were in a range satisfactory for workers and machines. As a result of this implementation a data sample of 19315 measurements was extracted that is described in table I. The samples comprehend the period from September, 3rd, 2015 at 17:25:37 to September, 04th, 2015 at 14:08:27.

No messages were lost during testing.

TABLE I: Data Server environment factors

Variable	Mean	STD	Min	Max
Temperature (°C)	27.097334	1.162900	25	31
Humidity (%)	35.382760	4.395421	26	45
Luminance (lux)	202.594512	445.754322	0	2055
Gas (ppm)	94.932747	17.561130	43	495

The python Package matplotlib [15] was used to create the following graphics:

On figure 3 it is possible to observe that the temperature in the room remains with little fluctuation with changes of only 6 degrees Celsius.

The graphic of the gas presence data contains 2 spikes that represent a test of the sensor in the beginning of the data collection and other more pronounced at approximately 21:45. The source of second spike is unknown.

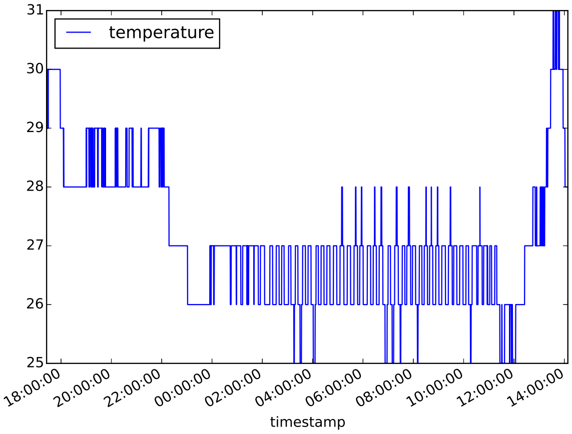

Fig. 3: Temperature

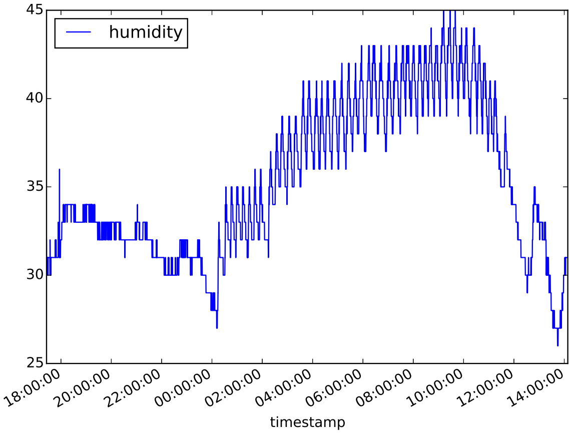

Fig. 4: Humidity

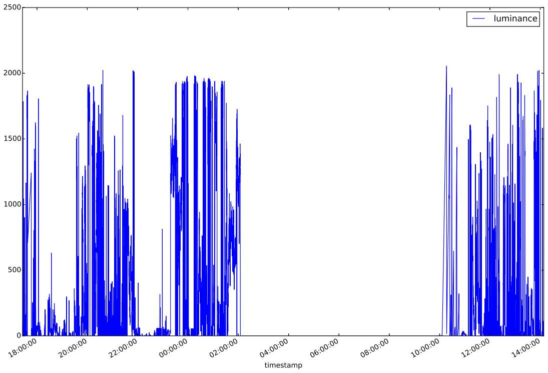

Fig. 5: Luminance

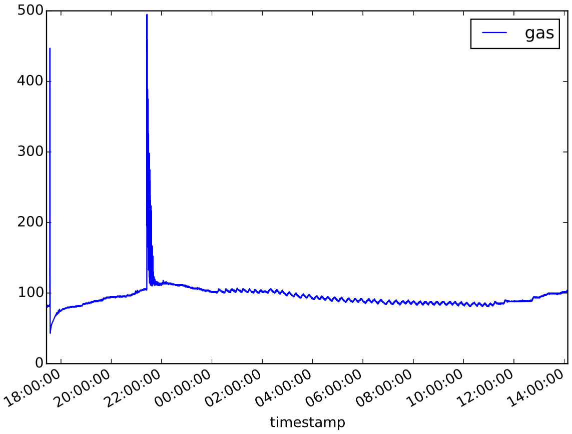

Fig. 6: Gas presence

D. Application of collected data

According with Woo [16] the workplace can make manifest latent mood disorders, destabilize, and aggravate symptoms and courses of mood disorders among workers. This is reinforced by Sarode [17] bad work conditions related to Noise, Lightning, air quality, temperature and humidity affected both the psychological and physiological welfare of the workers, causing such conditions as eye strain, fatigue, headache, back pain, and nausea.

The information collected by the WSN can be used to profile, make adjustments and improve work conditions in the monitored environment. These improvements may impact the productivity, quality and costs of workers.

The noise sensor information may be used to specify mufflers on combustion engines, the class of hear gear to be distributed to workers of a specific area or installation of acoustic insulation.

Luminance data may be used to adjust automatic levels of lighting.

Temperature and humidity data may be used to adjust Air conditioning systems.

III. Conclusion

In this article an Architecture for Wireless sensor networks for monitoring Environmental factors is presented. Environmental monitoring through sensor networks is a promising technology. With the advances in the miniaturization of sensors and low power micro-controller systems these networks will become more prevalent and gain more applications. In the future other sensor nodes will be deployed in an office ambient to also monitor the luminance and noise.

Others sensors may be used with this setup as hall sensors to monitor electrical energy consumption or flow rate sensors to monitor water consumption, river flow or sewers.

For long range monitoring further study will be necessary to choose the best network topology and technology with the best candidates being mesh networks over high frequency wireless.

Acknowledgment

The authors would like to thank their families for their love, care, kindness, patience, and unfailing trust.

References

[1] OSHA, “Occupational safety & health administration,” 2015.

[2] Pavithra L and A Angeline, “Survey on Industrial Wireless Sensor Networks,” pp. 3166–3171, 2015.

[3] R. F. d. S. Ribeiro, *Urubu : energy scavenging in wireless sensor networks*. PhD thesis, Polytechnic Institute of Porto School of Engineering, 2010.

[4] M. Cheffena, “Industrial wireless sensor networks : channel modeling and performance evaluation,” 2012.

[5] Scalagent, “Benchmark of MQTT servers,” Tech. Rep. January, 2015.

[6] R. P. Foundation, “What is raspberry pi?,” 2015.

[7] Raspbian, “Raspbian.org,” 2015.

[8] , “Sqlite,” 2015.

[9] , “Tornado,” 2015.

[10] “A python microframework,” 2015.

[11] R. T. Fielding, “Architectural Styles and the Design of Network-based Software Architectures,” *Building*, vol. 54, p. 162, 2000.

[12] “What is mqtt,” 2015.

[13] “Introduction to pika,” 2015.

[14] “Definition of json,” 2015.

[15] J. D. Hunter, “Matplotlib: A 2d graphics environment,” *Computing In Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.

[16] J. Woo and T. T. Postolache, “The impact of work environment on mood disorders and suicide: Evidence and implications,” *International Journal Disability Human Development*, vol. 7, no. 2, pp. 185–200, 2008.

[17] A. P. Sarode and M. Shirsath, “The Factors Affecting Employee Work Environment & Its Relation with Employee Productivity,” *International Journal of Science and Research (IJSR)*, vol. 3, no. 11, pp. 2735–2737, 2014.