

# Indoor Air Quality Monitoring Sensor for the Web of Things <sup>†</sup>

Daniel Ibaseta <sup>1</sup>, Julio Molleda <sup>2,\*</sup>, Fidel Díez <sup>1</sup> and Juan C. Granda <sup>2</sup>

<sup>1</sup> CTIC Technological Centre, 33203 Gijón, Spain; daniel.ibaseta@fundacionctic.org (D.I.); fidel.diez@fundacionctic.org (F.D.)

<sup>2</sup> Department of Computer Science and Engineering, University of Oviedo, 33204 Gijón, Spain; jcgranda@uniovi.es

\* Correspondence: jmolleda@uniovi.es; Tel.: +34-985-182-518

† Presented at 2nd International Research Conference on Sustainable Energy Engineering, Materials and Environment, Mieres, Spain, 25–27 July 2018.

Published: 14 November 2018

**Abstract:** Many Internet of Things platforms use dedicated software coupled with proprietary devices and interfaces, creating *silo* solutions with no interoperability. The Web of Things paradigm promotes using open Web standards to connect physical objects to the Internet through an application layer. In this paper, we propose a low-cost, indoor air quality monitoring sensor implementing a minimal servient building block recommended by the Web of Things Working Group of the World Wide Web Consortium. The proposed sensor runs a Web server on a low-power system-on-chip microcontroller, which provides temperature, relative humidity and carbon dioxide measurements to the Internet through a REST API. Any client on the Internet able to manage the HTTP protocol may access this sensor, making it compatible with any air quality monitoring platform that uses HTTP.

**Keywords:** IoT; WoT; REST API; air quality monitoring; SoC microcontroller

---

## 1. Introduction

Embedded computing and low-power wireless communication provide physical objects with Internet connectivity, forming the Internet of Things (IoT) [1]. The IoT offers services that lay at the core of many fields, such as smart homes, smart cities, smart industries and smart agriculture. Many products in the IoT use dedicated software tightly coupled with proprietary devices and interfaces, creating *silo* solutions that make the IoT suffer from a lack of interoperability. The Web of Things (WoT) paradigm uses open Web standards to connect things to the Internet and publish their data through an application layer. Reusing technologies and standards of the World Wide Web makes it possible to create interoperable infrastructures between the cyber-world and the physical-world [2].

One of the fields where the IoT plays a great role is air quality monitoring (AQM). Air quality refers to the concentration of several pollutants -such as carbon monoxide (CO), carbon dioxide (CO<sub>2</sub>), sulphur dioxide (SO<sub>2</sub>), nitrogen dioxide (NO<sub>2</sub>), and ozone (O<sub>3</sub>)- released by human activities and natural sources into air. Most of the IoT devices and services developed for AQM depend on proprietary platforms or technologies, making it hard to develop a wide accessible application.

In this paper we design and develop a low-cost, indoor air quality monitoring (IAQM) sensor following the recommendations of the World Wide Web Consortium (W3C) WoT Working Group. The main benefit of this sensor is the interoperability with other IoT-based AQM platforms. This sensor is worldwide accessible through a standard Web browser or any HTTP-based application.

## 2. The Web of Things

Two decades ago, Web technologies for embedded systems were envisioned to create intelligent devices worldwide connected and managed from Web browsers [3]. More recently, [4] report the changes on the Internet because of the IoT and state that the IoT should be considered as part of the overall Internet of the future. Modern solutions create proxies to bridge heterogeneous Web services from the Internet into IoT networks, as in the solution proposed by [5].

The Web of Things emerges from the application of Web standards and technologies to access information and services of physical objects in the IoT. Zeng et al. [2] survey architectural concepts and some key enabling technologies of the WoT. As in IoT each physical object possesses a digital counterpart referred as “Thing”, in the WoT each physical object is commonly referred as “Web Thing” [6]. A Web Thing commonly exposes an HTML or JSON representation, implements an API to access its properties, and defines an OWL-based (Web Ontology Language) semantic description.

In the WoT, all the things are abstracted as Web resources, and RESTful services is preferred because their low complexity and loose-coupling stateless interaction. The key characteristics of a RESTful design are: (i) identification of resources via URI; (ii) uniform interfaces to read and manipulate the resources, using four basic HTTP operations: GET, POST, PUT and DELETE; (iii) representation of the resources in different formats, such as HTML, JSON or XML; and (iv) stateless interaction.

Some organizations and standards focused on using Web technologies to counter IoT interoperability issues are:

*OGC Sensor Web Enablement.* The Open Geospatial Consortium (OGC), founded in 1994, is an international organization focused on open standards for the global geospatial community. The OGC’s Sensor Web Enablement (SWE) [7] is a suite intended to make all types of sensors, transducers and sensor data repositories discoverable, accessible and useable through the Web.

*Open Connectivity Foundation.* In February 2016, the Open Connectivity Foundation (OCF) was established as an industry group dedicated to ensuring secure interoperability through a standard communication platform, a bridging specification, an open source implementation and a certification program [8]. OCF sponsors IoTivity, an open source project hosted by the Linux Foundation focused on developing standard specifications, promoting interoperability guidelines and providing a certification program to enable IoT. In October 2016, this project was merged with AllJoyn, an open source framework to connect and discover devices.

*W3C Web of Things.* In December 2016, the W3C launched the Web of Things Working Group [9] to propose recommendations and develop a set of standards for the WoT, with the following goals: (i) counter the fragmentation of the IoT; (ii) reduce the cost of development of IoT solutions; and (iii) lessen the risks to both investors and customers. These recommendations are based on scripting languages like JavaScript, data encodings like JSON, and protocols like HTTP and WebSockets. The W3C WoT is devised as the application layer of the IoT, interconnecting existing IoT platforms and complementing available standards.

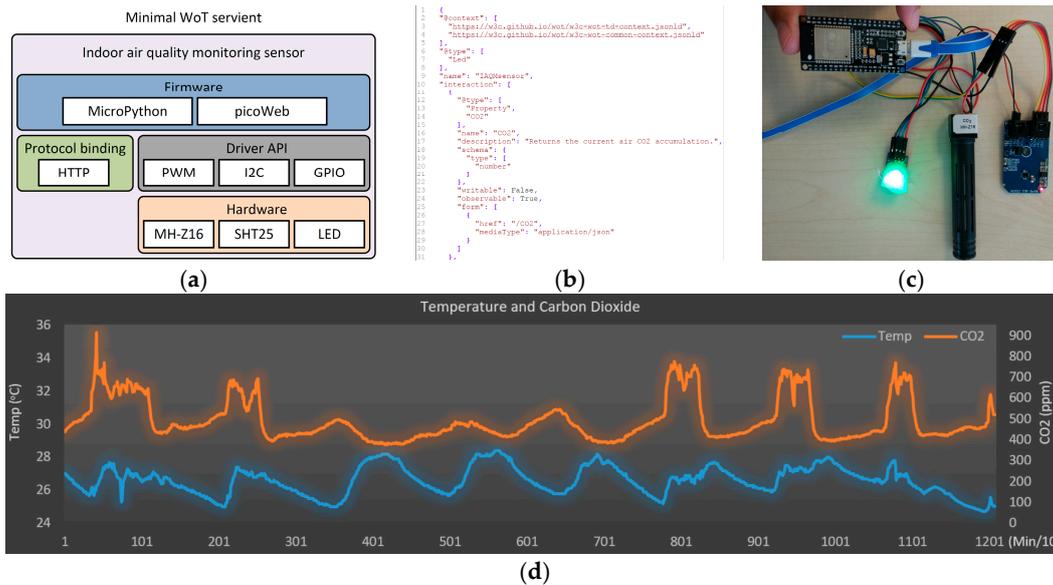
## 3. Indoor Air Quality Monitoring

Air quality monitoring is a common application of sensor networks and IoT. Postolache et al. [10] proposed a sensor network for AQM which is able to monitor air quality from different locations and publish it on the Web. Bhattacharya et al. [11] proposed a wireless sensor network based on Libelium nodes (composed of a gas sensor board and the Waspote processing board) to compute an air quality index used to command heating, ventilating and air conditioning systems. Abraham and Li [12] proposed an AQM wireless sensor network based on Arduino, XBee modules and low-cost micro gas sensors. Yu and Lin [13] used wireless sensor networks to develop an IAQM wireless system able to measure temperature, relative humidity and CO<sub>2</sub> concentration levels, which is able to analyze and predict air quality in an indoor working environment. Benammar et al. [14] designed a real-time IAQM system using wireless sensor networks, emphasizing the use of a gateway in processing collected air quality data and its reliable dissemination to end-users through a Web server. This system uses the Raspberry Pi 2 model B for the gateway, the open-source IoT platform Emoncms

for the Web server for live monitoring, and Libelium sensor nodes (composed of calibrated sensors, the Gas Pro sensor board and the Wasp mote processing board).

All the previously reviewed systems are capable of wirelessly monitoring the quality of the air of a given indoor environment. However, all of them create *silos* in the IoT ecosystem since they are conceived as standalone systems with restricted and/or proprietary communications among sensor boards and Web servers or gateways. Therefore, they are not able to communicate with each other, even though some of them use the same processing and gas boards. The sensors of a system can only communicate with the gateways of that specific system. If they were able to communicate among themselves, they would be able to lay the foundation of a global wireless network system for air quality monitoring.

The IAQM sensor designed in this work uses current Web standards and technologies to avoid interoperability issues. The proposed sensor is a low-cost node based on a low-power system-on-chip microcontroller, running a Web server. It is based on the W3C recommendations for WoT Architecture, implementing a minimal servient (a software stack that implements the W3C WoT building blocks). This sensor consists of: (i) Processing Board: the core of the servient is the Espressif ESP32 system-on-chip microcontroller. This system runs MicroPython, a full compiler and runtime of the Python 3 programming language optimized to run on microcontrollers. On top of it, a RESTful API is exposed by means of picoWeb, an asynchronous, non-threaded Web framework; and (ii) Sensor board: the gas sensor is chosen taking power consumption into account. We use the MH-Z16 infrared CO<sub>2</sub> sensor in the proposed IAQM sensor. Temperature and relative humidity are measured using the fully calibrated SHT25 sensor.



**Figure 1.** Proposed indoor air quality monitoring sensor: (a) Servient architecture of the proposed sensor; (b) Thing Description, according to the W3C WoT Architecture (because of space limitations, only the CO<sub>2</sub> section of the file is shown); (c) Sensor hardware and wiring; (d) Temperature and CO<sub>2</sub> output in an 8-day period (measurements taken every 10 min.).

The proposed sensor (see Figure 1) can be worldwide accessible from any standard Web browser using several URIs. For instance, requesting a temperature reading from the sensor would require accessing the following <http://WoTthing/temperature>, given that WoTthing links to the root directory of the Web server of the sensor.

#### 4. Conclusions

In this work we deal with the recommendations of the W3C WoT Working Group to counter the interoperability issues in IoT platforms. W3C WoT tries to augment or interface with existing standards and platforms. We designed and developed a low-cost, indoor air quality monitoring

sensor which publishes air quality measurements on the Internet through a REST API, accessible using standard Web browsers or HTTP-based applications. Regardless of the minimal servient architecture implemented in this system, the sensor can be worldwide accessible. This makes the sensor really accessible, compatible, and also transparent for programmers to use. The main benefit of the sensor proposed in this work is the interoperability with other IoT-based air quality monitoring platforms, since the sensor can provide a description about its features, and thus, any platform can retrieve measurements from this sensor despite the manufacturer.

Future work will involve creating more building blocks for the system, as defined by the W3C WoT Architecture.

**Author Contributions:** J.M. and D.I. designed the research plan. J.M. reviewed the literature and wrote the original manuscript. D.I. built the sensor, developed the software and revised the manuscript. All authors conceived and designed the testing, discussed and commented on the paper.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Al-Fuqaha, A.; Guizani, M.; Mohammadi, M.; Aledhari, M.; Ayyash, M. Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. *IEEE Commun. Surv. Tutor.* **2015**, *17*, 2347–2376. doi:10.1109/COMST.2015.2444095.
2. Zeng, D.; Guo, S.; Cheng, Z. The web of things: A survey. *J. Commun.* **2011**, *6*, 424–438. doi:10.4304/jcm.6.6.424-438.
3. Agranat, I. Engineering web technologies for embedded applications. *IEEE Int. Comput.* **1998**, *2*, 40–45. doi:10.1109/4236.683798.
4. Atzori, L.; Iera, A.; Morabito, G. The Internet of Things: A survey. *Comput. Netw.* **2010**, *54*, 2787–2805. doi:10.1016/j.comnet.2010.05.010.
5. Jin, W.; Kim, D. Development of virtual resource based IoT proxy for bridging heterogeneous web services in IoT networks. *Sensors* **2018**, *18*. doi:10.3390/s18061721.
6. W3C. Web of Things (WoT) Architecture. Available online: <https://www.w3.org/TR/wot-architecture/> (accessed on 17 April 2018).
7. OGC-SWE. Open Geospatial Consortium—SensorWeb Enablement. Available online: <http://www.opengeospatial.org/ogc/markets-technologies/swe> (accessed on 26 June 2018).
8. OCF. Open Connectivity Foundation. Available online: <https://openconnectivity.org/> (accessed on 25 June 2018).
9. W3C. Web of Things Working Group. Available online: <https://www.w3.org/WoT/WG/> (accessed on 25 May 2018)
10. Postolache, O.; Dias Pereira, J.; Silva Girão, P. Smart sensors network for air quality monitoring applications. *IEEE Trans. Instrum. Meas.* **2009**, *58*, 3253–3262. doi:10.1109/TIM.2009.2022372.
11. Bhattacharya, S.; Sridevi, S.; Pitchiah, R. Indoor air quality monitoring using wireless sensor network. In Proceedings of the International Conference on Sensing Technology (ICST), Kolkata, India, 18–21 December 2012; pp. 422–427. doi:10.1109/ICSensT.2012.6461713.
12. Abraham, S.; Li, X. A cost-effective wireless sensor network system for indoor air quality monitoring applications. *Procedia Comput. Sci.* **2014**, *34*, 165–171. doi:10.1016/j.procs.2014.07.090.
13. Yu, T.C.; Lin, C.C. An Intelligent Wireless Sensing and Control System to Improve Indoor Air Quality: Monitoring, Prediction, and Preaction. *Int. J. Distrib. Sens. Netw.* **2015**, *2015*. doi:10.1155/2015/140978.
14. Benammar, M.; Abdaoui, A.; Ahmad, S.; Touati, F.; Kadri, A. A modular IoT platform for real-time indoor air quality monitoring. *Sensors* **2018**, *18*. doi:10.3390/s18020581.

