

Article

A Software Framework for Heterogeneous Wireless Sensor Network Towards Environmental Monitoring

Qian Huang ^{1,*}  and Kane Rodriguez ²¹ School of Architecture, Southern Illinois University Carbondale, Carbondale, IL 62901, USA² Department of Electrical and Computer Engineering, Southern Illinois University Carbondale, Carbondale, IL 62901, USA; kaustez@siu.edu

* Correspondence: qhuang@siu.edu

Received: 7 January 2019; Accepted: 21 February 2019; Published: 28 February 2019



Abstract: A wireless sensor network (WSN) is typically composed of spatially distributed miniature sensors that help collect large amounts of real-time environmental data from buildings. These environmental data (e.g., temperature, humidity, CO₂ concentration) can assist a series of heating, ventilation, and air conditioning (HVAC) equipment to increase the building energy efficiency. From a system design perspective, heterogeneous wireless sensor networks need to address two challenges. First, sensor data acquisition, conversion, fusion, and packaging involve a series of software processing. Since each type of environmental sensor typically has unique processing requirements, it is difficult to develop an efficient software framework to combine the processing of multiple heterogeneous sensors. Second, during normal operation of a heterogeneous wireless sensor network, if users insert or remove some environmental sensors, the entire WSN system should operate normally. In this work, in order to solve the above two system design challenges, we have developed a low-power, low-cost, small form-factor WSN development platform, and its software framework can perform efficient data acquisition, conversion, fusion, and packaging for multiple heterogeneous sensors. Our proposed software framework enables easy and rapid WSN system deployment without affecting the overall functionality of each node. The proposed design differs from existing WSN platforms in that it emphasizes advanced high-level usability and reduces time to market without sacrificing low-level features. The proposed WSN system has been implemented and tested in an office building for indoor fire hazard detection. The experimental results show that our software framework can successfully complete data acquisition, conversion, fusion, and packaging tasks for three heterogeneous environmental sensors. In addition, we have verified that our software framework supports robust system operation when inserting or removing sensors from an existing heterogeneous WSN system.

Keywords: energy-efficient building; heterogeneous wireless sensor network; software framework

1. Introduction

More than 35% of the electricity in the United States is used in heating, ventilation, and air conditioning (HVAC) of buildings [1]. Researchers believe that intelligent control of HVAC operation can significantly reduce building energy consumption. For example, it has been reported that 10%–15% reduction in building energy is achieved using occupancy-driven adaptive HVAC control [2]. So far, intelligent HVAC control needs to analyze real-time environmental information in buildings, such as room temperature, humidity, CO₂ level, and room occupancy levels [3]. Nowadays, various wireless sensor networks have been developed to sense and collect building environmental information. In this decade, many types of wireless sensor networks (WSNs) have been developed for indoor environmental sensing. So far, the main optimization aspects of these WSN systems include sensing

accuracy, power consumption, form factor, and implementation cost. As the global WSN market is expected to exceed \$1.8 billion by 2024, the need for a standard WSN development platform is emerging [4]. Nowadays, existing WSN development platforms mainly emphasize the design aspects of power consumption [5,6], system cost [7–10], and network latency [11–13]. Yet, rare research efforts have been made to create and establish user-friendly design paradigms that allow ordinary users to easily deploy and configure heterogeneous WSNs, rather than WSN professionals or researchers. It is attractive to develop a low-power, low-cost, small form-factor WSN development platform that enables rapid system deployment without much knowledge of WSNs.

When multiple different types of environmental sensors are utilized in a WSN, it becomes a heterogeneous WSN. Depending on user requirements or preferences, these different types of environmental sensors may come from a sensor supplier or various sensor suppliers. From a system design perspective, heterogeneous wireless sensor networks WSNs need to address several challenges. First, sensor data acquisition, conversion, fusion, and packaging involve a series of software processing. For example, typical outputs of these environmental sensors are analog signals, which need to be converted to digital signals that allow for complex data processing (e.g., fusion, encryption) in microcontrollers. Moreover, as each type of environmental sensor typically has unique processing requirements, it is difficult to develop high-level software frameworks to efficiently conduct data processing for multiple heterogeneous sensors. Second, during normal operation of a heterogeneous wireless sensor network, if users insert or remove some environmental sensors, the entire WSN system should function properly. In this work, to solve the above two system design challenges, we have developed a low-power, low-cost, small form-factor WSN development platform whose software framework performs efficient data acquisition, conversion, fusion, and packaging for multiple heterogeneous sensors (i.e., CO₂, humidity, and light sensors). As this work is targeted for open-source WSN platforms, the source code in our software framework is written using Node.js, a widely supported open-source JavaScript. Therefore, our proposed software framework will benefit from community-supported applications that leverage the latest coding for each software object.

This work makes the following contributions: (1) The proposed software framework facilitates the connection of a flexible number of heterogeneous sensors into a WSN and customizes the network configuration interface. Thus, our proposed software framework supports robust and functional system operation when inserting or removing environmental sensors. (2) Our proposed software framework can collaboratively operate with a set of WSN hardware components, including heterogeneous environmental sensors, microcontrollers, and others. Our proposed high-level software framework does not compromise the overall functionality of each WSN hardware component. (3) After implementing the entire heterogeneous WSN system, we successfully conduct an indoor fire hazard test. The experimental results show that our software framework can successfully fulfill data acquisition, conversion, fusion, and packaging tasks for three heterogeneous environmental sensors. In addition, we have verified that our software framework supports robust and functional system operation when inserting or removing sensors in this implemented heterogeneous WSN.

The rest of this paper is organized as follows. Section 2 reviews the related literature on wireless sensor networks. Section 3 describes the details of proposed WSN system architecture, including physical components, software framework, and system integration. Section 4 will present and discuss experimental data for indoor fire hazard detection. Finally, Section 5 will conclude and describe future work.

2. Related Works in the Literature

In the past few years, research efforts have been made to optimize power consumption, deployment cost, and data sensing accuracy in WSN systems. For example, researchers from the University of California, Berkeley, have developed an ultra-low-power sensor module (Telos), which enters a low-power mode of operation until events (e.g., data acquisition or transmission) wake the sensor module from a low-power state [5]. In 2012, researchers proposed a wireless sensor network

that focuses on easy access to real-time data in a car pollution control setting [14]. This design includes an Atmega 328-PU microcontroller, several gas sensors, and global positioning system (GPS) modules. In addition, another low-power, small form-factor, WSN design (Sprouts) was proposed for general applications [15]. Sprouts is an open-source, multi-standard WSN platform that relies on Bluetooth communication and vibration energy harvesting for low-power operation. The WSN architecture of Sprouts is based on using ARM Cortex M3 microcontrollers for data acquisition and processing. To further reduce power consumption, the researchers use an MSP430 microcontroller and a CC2530 antenna to build an outdoor WSN platform [16]. This sensor design supports low power consumption, because the MSP430 microcontroller minimizes energy usage by entering a low-power mode, and the CC2530 antenna utilizes the ZigBeePRO stack solution for low-power wireless communication. Their focus is on avoiding the redundancy of custom WSN applications for outdoor environmental monitoring, but the architecture they propose does not provide a corresponding implementation. In 2014, researchers proposed a WSN dedicated to sustainable agriculture [12]. This WSN system consists of several sensor nodes around a base station node. The base station node has a wider transmission range than its adjacent sensor nodes, so the base station node is responsible for collecting time-stamped data from each surrounding sensor node and relaying the data to another base station node until the data is uploaded to the internet. A Raspberry Pi microcontroller is used in this sensor design. The choice of such a microcontroller greatly increases the power consumption of the overall design. Another disadvantage of this design is its high hardware cost (\$78 per microcontroller). Later, a traffic monitoring system was developed to adopt a ZigBee communication protocol [7]. However, in this design, the adopted microcontroller for sensor data acquisition is an Arduino Uno, which is relatively expensive and also consumes substantially high power.

In 2015, a study of energy-efficient lighting control in green buildings used an MSP430 microcontroller and a CC2530 digital antenna chip for WSNs [8]. Compared to the other WSN designs, this low-power design is cost-effective and also offers a compact form factor. In 2016, the researchers integrated CO₂ and light sensors into a wireless sensor platform, which also utilizes an MSP430 microcontroller and a CC2500 transceiver [9]. This design achieves higher sensing accuracy and also keeps low cost and power consumption. Later, in 2017, the researchers [10] proposed to assemble five sensing functions (i.e., CO₂ level, temperature, lighting, room occupancy, and fire detection) by using three heterogeneous environmental sensors (i.e., a temperature sensor, a light sensor, and a CO₂ sensor). Unfortunately, the study in [10] only demonstrated a prototype of hardware implementation and did not discuss the corresponding software framework that performs data acquisition, conversion, fusion, and packaging from multiple heterogeneous sensors.

From the above discussion, so far, none of these existing heterogeneous WSN systems have addressed the issue of creating high-level user-friendly software frameworks. As a result, it is worth investigating advanced software frameworks that enable low design complexity, low cost, robust system operation, and easy implementation or configuration by ordinary users without a strong technical background. This is the focus of this paper.

3. Proposed WSN System Architecture

This section describes details of physical components, software components, system integration, and wireless network topology of the proposed WSN development platform.

3.1. Physical Components

In this WSN system, there are two main physical components: a wireless node module and an application point module. The wireless node module consists of an EZ430-RF2500 board, which is developed by Texas Instruments Inc. (Dallas, TX, United States). As shown in Figure 1, this board integrates an MSP430 microcontroller from Texas Instruments Inc. (Dallas, TX, United States), a CC2500 antenna chip from Texas Instruments Inc. (Dallas, TX, United States), and several signal input/output (I/O) pins for connecting heterogeneous environmental sensors. The board is powered by a battery

pack that provides a supply voltage of 3 V. The application point module differs from the wireless node module in that it is powered by a USB port that provides a direct physical link to a computer.



Figure 1. Hardware components of a wireless sensor module and an application point module.

Next, we introduce the details of this microcontroller and antenna chip. The MSP430 microcontroller enables low power operation between data acquisition and transmission events. It embeds an analog-to-digital converter (ADC) that collects analog data from any signal I/O pin on the board. The analog-to-digital conversion involves internal voltage references for voltage comparison. In addition, the microcontroller facilitates wireless data communication between nodes through the CC2500 antenna chip. The antenna chip offers very low-power communications and operates on radios below 1 GHz and 2.4 GHz. Moreover, it provides preloaded open-source code for Advanced Encryption Standard (AES)-128 security.

3.2. Software Framework

As an indispensable block in WSN systems, a software framework includes software components in Figure 2, such as Simple-Device, Simple-Message, Simple-Communication, Application-Point-Server, Data-Management-Application, and SimpliciTi Application Programming Interface (API). In this section, we will introduce and describe each of these software components in detail.

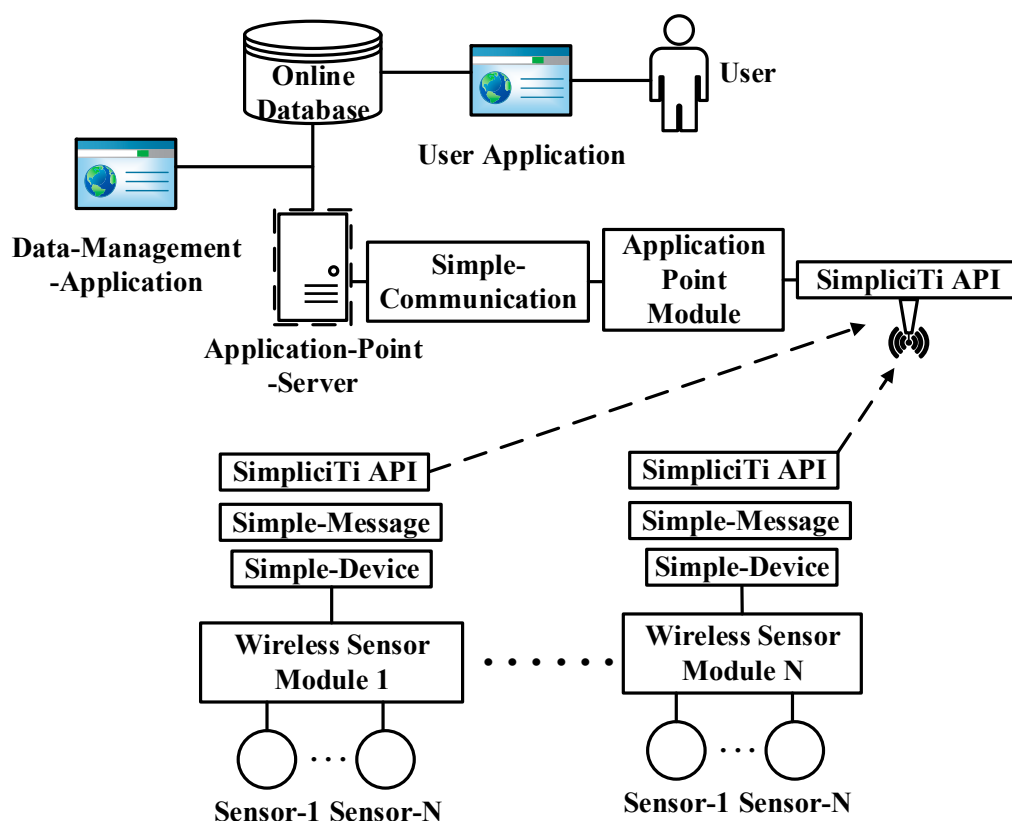


Figure 2. Block diagram of system software framework. API: Application Programming Interface.

Simple-Device object: It stores all relevant node device information into a single packaged object. This object contains device information related to device voltage, device temperature, and ADC conversion for each signal I/O pin. The reference voltages for ADC conversion are dynamically selected in the software. Moreover, after the device reads its battery voltage, this battery voltage will be used as the upper bound reference voltage for every ADC conversion. This reference voltage is always updated before new data is acquired from each pin.

Simple-Message object: It leverages an underlying SimpliciTi API to facilitate the transfer of device data. The Simple-Device object is passed to a Simple-Message object, which wraps all device data before transfer. After sending a data packet, the receiving device unpacks the packet into usable data using a Simple-Message object.

Simple-Communication object: It allows easy communication with the host operating system via a UART port. Data is transferred using a JSON format and allowed to be easily parsed in the host operating system. This software module is located on the application point node, where the node is connected to the host operating system through a physical UART port.

Application-Point-Server host: It uses a Node.js server to listen for incoming data transfers from an application point network, which is connected to a host computer. Once an incoming transfer request is received, the data is transformed according to user-specified conversion and data label mapping. Later, this data will be posted in an online database.

Data-Management-Application: This is a setting application that allows users to configure data conversion functions for each environmental sensor. In addition, the received data can be mapped to the desired range via this type of a user-friendly interface. This module also allows users to view recently collected data from each sensor in the network.

SimpliciTi Application Programming Interface (API): A low-complexity software has been provided by Texas Instruments to facilitate communication between CC2500 chips. In this software framework, the API supports node communications in a peer-to-peer or star network topology. In addition, if an end device node is too far away to communicate with the application point, repeaters may be brought into the network for the signal relay.

3.3. Overall System Integration and Operation

This proposed software framework provides an abstraction level for low-level microcontrollers in the network to collect data for users. With the Data-Management-Application module, users can easily modify collected data into forms that can be easily used with any other existing application. As shown in Figure 3, the system operation and the life cycle of acquired data are described as follows: (1) The output pins of environmental sensors (e.g., humidity and temperature sensors) are connected to the user I/O pins of wireless node module. Yet, each environmental sensor provides its analog output, referring to the supply voltage and ground voltage of the sensor node. Therefore, when an analog output is captured by a wireless node module, it is necessary to convert the analog output to an appropriate voltage with respect to the reference voltages of a wireless node module. In the proposed software framework, all sensor output pin voltages, device temperature, and device voltage sets are packaged in a Simple-Device object. Then, this object is compressed into a Simple-Message object, which is wrapped into a SimpliciTi API package. (2) This package is wirelessly transmitted to an application point module, which decompresses a Simple-Message object back into a Simple-Device object. Then, a Simple-Communication object is used to send received environmental sensor data to a host computer in JSON format via the UART port. On the host computer, an Application-Point-Server host listens for any incoming transmission on this specified UART port. Once a Simple-Communication event is detected, the received data is modified based on the user specifications defined in the Data-Management-Application. For example, when an application point module sends data serially to an Application-Point-Server host, the data can be converted from “devicePin3Voltage: 2.33” to “CO₂ (ppm): 1200”. This conversion is achieved by mapping the received data to specified user data. (3) After the host computer obtains the captured sensor data, it will upload the data to an online

database for visualization or analysis in the future. According to the above discussion, the proposed system supports a “Plug and Play” style of system operation and data acquisition. Once a user connects the output pin of an environmental sensor to a wireless sensor module, the specific sensor data will be captured and converted via the proposed software framework, and then the data collected by this environmental sensor will be uploaded to an online database. The disconnection of an environmental sensor from this WSN does not affect the operation of other sensors in this system.

1. Connect Environmental Sensors to a Wireless Sensor Module

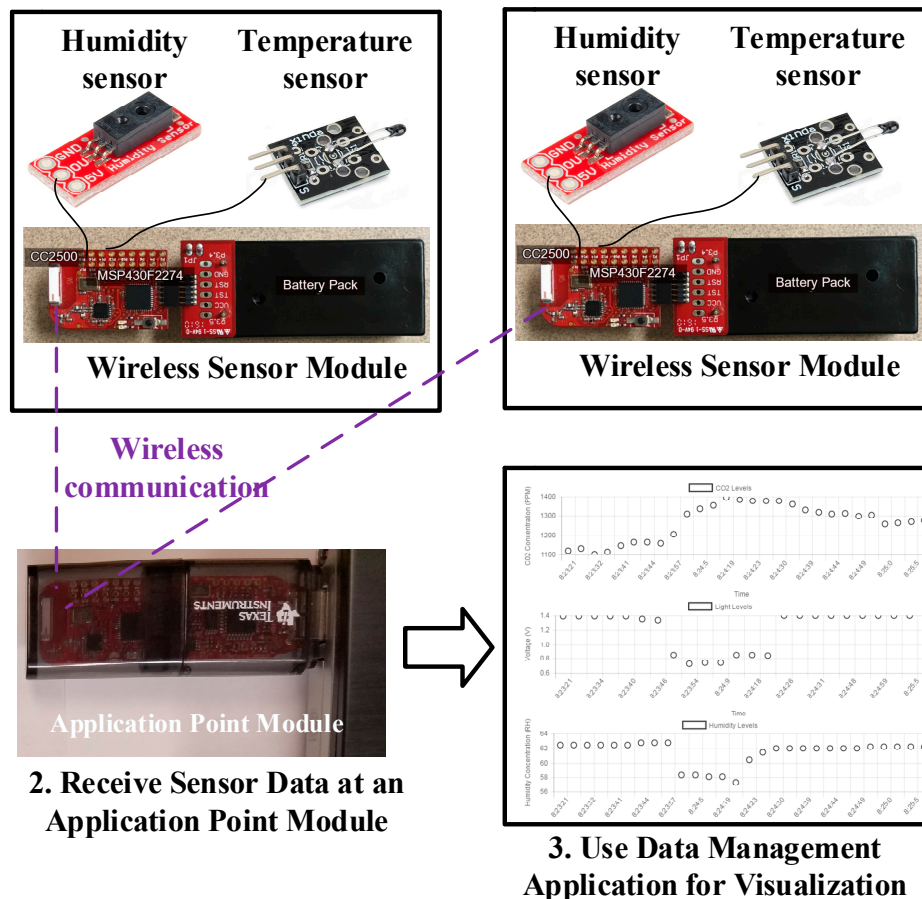


Figure 3. Illustration of the entire wireless sensor network (WSN) system operation.

In Figure 3, the entire WSN system utilizes a star network topology. In a star topology, an application point module is typically surrounded by all wireless sensor modules. Other network topologies (e.g., tree, ring, or fully connected) are also available and applicable for our WSN system platform. Particularly, if a wireless sensor module is too far away from the application point module, a repeater can be placed between the two wireless sensor modules for the signal augment and relay.

4. Experimental Validation and Measurement Results

In the literature, a variety of WSN systems have been widely adopted in fire hazard detection and prediction [17–21]. Therefore, in this work, in order to validate the functionality of our proposed software framework, we applied our developed WSN system to detect indoor fire hazards in an office building.

As shown in Figure 4, the experimental setup consists of two wireless sensor modules, an application point module, a host computer, a fire source (i.e., lighter), and three environmental sensors (i.e., a light sensor, a CO₂ sensor, and a humidity sensor). The CO₂ and humidity sensors are connected to wireless sensor module 1, while the light sensor is connected to wireless sensor module

2. After initializing the proposed WSN system to collect data for approximately 25 s, the lighter was ignited to produce a flame in this room close to both wireless sensor modules. Then, after keeping it for around 25 s, the lighter flame was extinguished. Figures 5–8 plot how the collected sensor data in our deployed heterogeneous WSN system vary with time.

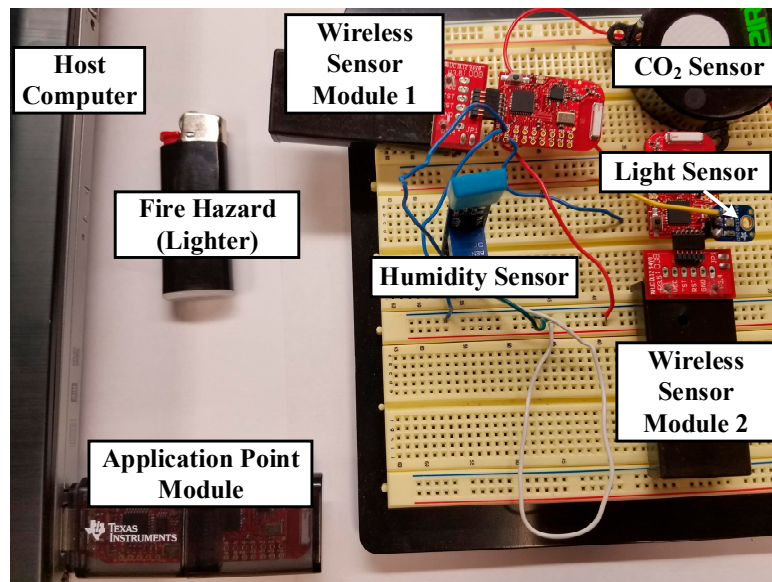


Figure 4. Experimental setup for indoor fire hazard detection system.

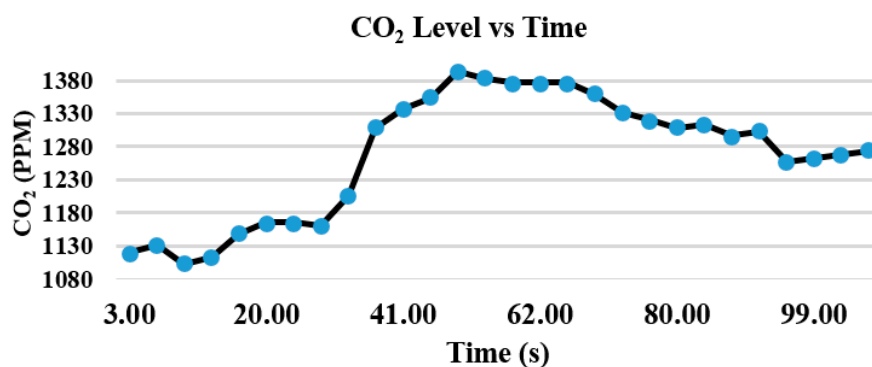


Figure 5. The recorded CO₂ level resultant from this fire hazard event.

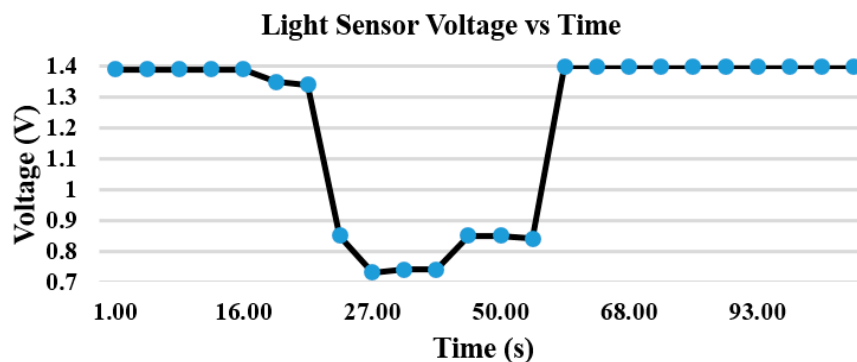


Figure 6. The recorded light sensor voltage level resultant from this fire hazard event.

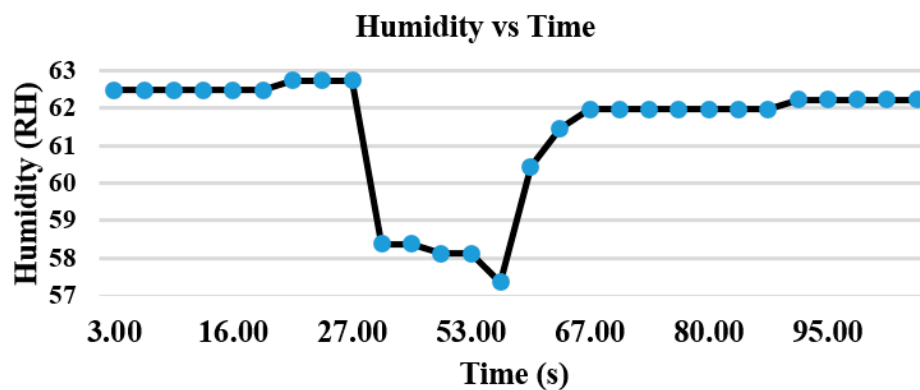


Figure 7. The recorded humidity level resultant from this fire hazard event.

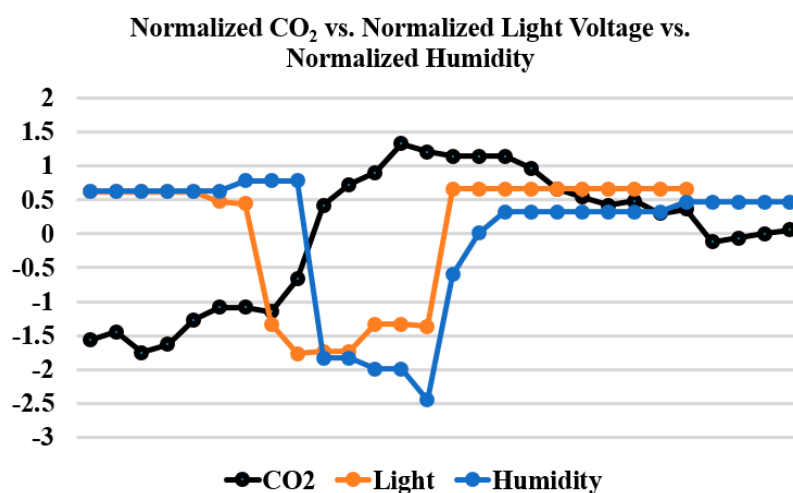


Figure 8. Comparison of the normalized sensor data from three heterogeneous sensors.

Figure 5 shows a sharp rise in the carbon dioxide level after the ignition of the lighter. We can see that the CO₂ level gradually reached a peak value of 1380 ppm. After extinguishing the lighter, the flame disappeared, so the CO₂ level began to drop slowly with time. In addition, as shown in Figure 6, the output voltage of this light sensor exhibited a valley value during the presence of the flame. An almost immediate voltage drop appeared at around $T = 25$ s, and a corresponding voltage rise appeared at around $T = 55$ s. Figure 7 plots the relative humidity (RH) values sensed by the humidity sensor in Figure 4. We can see that the RH level dropped sharply at around $T = 26$ s, and started to recover at around $T = 55$ s. The RH level is expected to be higher during the presence of a nearby flame because the byproducts of a combustion event are carbon dioxide and water. However, since the adopted RH sensor does take temperature into account when determining the RH value, the temperature increase caused by the flame adversely affected the sensor output. As a result, the measured RH value became even lower in this experiment.

These experimental results show that our software framework can successfully fulfill data acquisition, conversion, fusion, and packaging tasks for three heterogeneous environmental sensors. Figure 8 plots the comparison of collected data from the three sensors after normalization, when a nearby fire hazard event occurs. To avoid false positive detections, the integrated microcontroller will make decisions based on the captured three simultaneous pulses in Figure 8. In this study, we also verified that the detection behavior of this WSN system is normal when a new environmental sensor is inserted or an existing environmental sensor is removed from the two wireless sensor modules in Figure 4. The observed robust system operation demonstrates that our high-level software framework supports convenient heterogeneous WSN configurations while the system is running. From the user's

point of view, this is a very useful feature because ordinary users without a strong background or WSN knowledge can freely configure or adjust WSNs without worrying about system crashes.

5. Conclusions and Future Work

Due to the high design complexity and lack of a high-level user-friendly software framework, existing designs of heterogeneous WSN systems require professionals to build or deploy them. Therefore, ordinary users cannot easily configure or modify heterogeneous WSNs. To date, there are few studies in the literature that address the issue of creating high-level user-friendly software frameworks. To solve this problem, we have proposed an advanced user-friendly software framework for heterogeneous WSN systems. The block diagram and basic objects of the proposed software framework are introduced and described. To the best of our knowledge, our proposed software framework is the first work in the literature to perform efficient and user-friendly data acquisition, conversion, fusion, and packaging for multiple heterogeneous sensors. It emphasizes ease of use and rapid deployment time without sacrificing low-level system features such as low power, low cost, and small form factor. The proposed software framework has been implemented in a WSN system. Measurement results in a fire hazard detection indicate that our proposed software framework can successfully perform data acquisition, conversion, fusion, and packaging tasks for three heterogeneous environmental sensors. This experiment also shows that our software framework supports convenient WSN configurations (i.e., insertion or removal of environmental sensors while a heterogeneous WSN is running). In the future, we will investigate the integration of expert systems into microcontrollers to check for redundant information before broadcasting data, thus further reducing the network traffic and power consumption of heterogeneous WSNs.

Author Contributions: Q.H. conceived and designed the experiments; K.R. performed the experiments; Q.H. and K.R. analyzed the data and wrote the paper.

Conflicts of Interest: The authors declare no conflict of interest. The founding sponsors had no role in the design of the study; in the acquisition, analyses, or interpretation of data; in the writing of the manuscript; and in the decision to publish the results.

References

1. Lombard, L.; Ortiz, J.; Pout, C. A review on buildings energy consumption information. *Energy Build.* **2008**, *40*, 394–398. [[CrossRef](#)]
2. Agarwal, Y.; Balaji, B.; Gupta, R.; Lyles, J.; Wei, M.; Weng, T. Occupancy-driven energy management for smart building automation. In Proceedings of the ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Building, Zurich, Switzerland, 2–5 November 2010; pp. 1–6.
3. Huang, Q.; Rodriguez, K.; Whetstone, N.; Habel, S. Rapid Internet of Things (IoT) prototype for accurate people counting towards energy efficient buildings. *J. Inf. Technol. Construct.* **2019**, *24*, 1–13.
4. Harrop, P.; Das, R. *Wireless Sensor Network (WSN) 2014–2024: Forecasts, Technologies, Players*; IDTechEx: Boston, MA, USA, 2014.
5. Polastre, J.; Szewczyk, R.; Culler, D. Telos: Enabling ultra-low power wireless research. In Proceedings of the International Symposium on Information Processing in Sensor Networks, Los Angeles, CA, USA, 24–27 April 2005; pp. 364–369.
6. Bonivento, A.; Carloni, A.; Vincentelli, A. Platform-based design of wireless sensor networks for industrial applications. In Proceedings of the Design Automation & Test in Europe Conference, Munich, Germany, 6–10 March 2006; pp. 1–6.
7. Lee, J.; Zhong, Z.; Du, B.; Gutesa, S.; Kim, K. Low-cost and energy-saving wireless sensor network for real-time urban mobility monitoring system. *J. Sens.* **2015**, *2015*, 685786. [[CrossRef](#)]
8. Magno, M.; Polonelli, T.; Benini, L.; Popovici, E. A low cost, highly scalable wireless sensor network solution to achieve smart LED light control for green buildings. *IEEE Sens. J.* **2014**, *15*, 2963–2973. [[CrossRef](#)]
9. Huang, Q.; Mao, C. Occupancy estimation in smart buildings using hybrid CO₂/light wireless sensor network. *J. Appl. Sci. Arts* **2016**, *1*, 1–13.

10. Huang, Q.; Mao, C.; Chen, Y. A compact and versatile wireless sensor prototype for affordable intelligent sensing and monitoring in smart buildings. In Proceedings of the International Workshop on Computing in Civil Engineering, Seattle, WA, USA, 25–27 June 2017; pp. 155–161.
11. Selavo, L.; Wood, A.; Cao, Q.; Sookoor, T.; Liu, H.; Srinivasan, A.; Wu, Y.; Kang, W.; Stankovic, J.; Young, D.; et al. LUSTER: Wireless sensor network for environmental research. In Proceedings of the International Conference on Embedded Networked Sensor Systems, Sydney, Australia, 6–9 November 2007; pp. 1–14.
12. Concepcion, A.; Stefanelli, R.; Trincherro, D. A wireless sensor network platform optimized for assisted sustainable agriculture. In Proceedings of the IEEE Global Humanitarian Technology Conference, San Jose, CA, USA, 10–13 October 2014; pp. 159–165.
13. Fu, Y.; Guo, Q.; Chen, C. A-LNT: A wireless sensor network platform for low-power real-time voice communications. *J. Electr. Comput. Eng.* **2014**, *2014*, 394376. [[CrossRef](#)]
14. Reshi, A.; Shafi, S.; Kumaravel, A. VehNode: Wireless sensor network platform for automobile pollution control. In Proceedings of the 2013 IEEE Conference on Information & Communication Technologies, Thuckalay, India, 11–12 April 2013; pp. 963–966.
15. Kouche, A. Towards a wireless sensor network platform for the Internet of Things. In Proceedings of the IEEE International Conference on Communications, Ottawa, ON, Canada, 10–15 June 2012; pp. 632–636.
16. Lee, C. Towards a general wireless sensor network platform for outdoor environment monitoring. In Proceedings of the IEEE Conference on Sensors, Taipei, Taiwan, China, 28–31 October 2012; pp. 1–5.
17. Sabit, H.; Anbuky, A.; Hosseini, H. Wireless sensor network based wildfire hazard prediction system modeling. *Procedia Comput. Sci.* **2011**, *5*, 106–114. [[CrossRef](#)]
18. Huang, Q.; Cox, R.; Shaurette, M.; Wang, J. Intelligent building hazard detection using wireless sensor network and machine learning techniques. In Proceedings of the International Conference on Computing in Civil Engineering, Clearwater Beach, FL, USA, 17–20 June 2012; pp. 485–492.
19. Serna, M.; Casado, R.; Bermudez, A.; Pereira, N.; Tennina, S. Distributed forest fire monitoring using wireless sensor networks. *Int. J. Distrib. Sens. Netw.* **2015**, *11*, 964564. [[CrossRef](#)]
20. Saeed, F.; Paul, A.; Rehman, A.; Hong, W.; Seo, H. IoT-based intelligent modeling of smart home environment for fire prevention and safety. *J. Sens. Actuator Netw.* **2018**, *7*, 7010011.
21. Carlotti, T.; Silvani, X.; Innocenti, E.; Morandini, F.; Bulte, N.; Dang, T. An OCARI-based wireless sensor network for heat measurements during outdoor fire experiments. *Sensors* **2019**, *19*, 158. [[CrossRef](#)] [[PubMed](#)]



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).